

PROCEEDINGS OF

**The 59th Conference on
Simulation and Modelling (SIMS 59)**

26-28 September 2018

Oslo Metropolitan University, Norway



Lars Erik Øi, Tiina Komulainen, Robin T. Bye, and Lars O. Nord

Editors

The conference is organized by Oslo Metropolitan University, in cooperation with University of South-Eastern Norway, NTNU - Norwegian University of Science and Technology, Norwegian Society of Electric and Automatic Control (NFEA), and Scandinavian Simulation Society (SIMS).

www.scansim.org

Proceedings of the 59th Conference on Simulation and Modelling (SIMS 59)

Oslo Metropolitan University, Norway, 26-28 September 2018

Editors:

Lars Erik Øi, Tiina Komulainen, Robin T. Bye, and Lars O. Nord

Published by:

Scandinavian Simulation Society and Linköping University Electronic Press

ISBN: 978-91-7685-494-5

Series: Linköping Electronic Conference Proceedings, No. 153

ISSN: 1650-3686

eISSN: 1650-3740

DOI: <https://doi.org/10.3384/ecp181531>

Organized by:

Oslo Metropolitan University

c/o Tiina Komulainen

Pilestredet 46

NO-0167 Oslo

Norway

in cooperation with:

University of South-Eastern Norway

Postboks 235, NO-3603 Kongsberg, Norway

NTNU – Norwegian University of Science and Technology

NTNU, NO-7491 Trondheim, Norway

Norwegian Society of Electric and Automatic Control (NFEA)

Kjøita 25, NO-4630 Kristiansand S, Norway

Scandinavian Simulation Society (SIMS)

Malardalen University (chairman), Box 883, 72123 Västerås, Sweden

Cover photo:

Oslomet - Storbyuniversitetet

© 2018 Scandinavian Simulation Society. All rights reserved.

Welcome to SIMS 59

Oslo Metropolitan University (OsloMet) in collaboration with University of South-Eastern Norway (USN), NTNU – Norwegian University of Science and Technology, the Norwegian Society of Electric and Automatic Control (NFEA), and the Scandinavian Simulation Society are proud to arrange the 59th International Conference on Simulation and Modelling (SIMS 59) in Oslo, Norway.

This 1000-year-old Norwegian capital sits at the head of the Oslo Fjord and has a municipal population of about 670 000. The city offers visual delights such as the Vigeland Sculpture Park, the Viking Ship Museum, the Munch Museum and Holmenkollen, as well as more challenging content such as the new Nobel Peace Centre and the Holocaust Center. The Oslo Pass allows free travel on public transport, free parking and many museum and sight admissions.

The Scandinavian Simulation Society consists of members from five Nordic countries: Denmark, Finland, Norway, Sweden, and Iceland. The goal of SIMS is to develop further the science and practice of modelling and simulation in all application areas and to be a forum for information interchange between professionals and non-professionals in the Nordic countries.

The SIMS 59 conference will cover broad aspects of simulation, modeling and optimization in engineering, especially in energy applications. Central themes include:

- modeling and simulation for design, planning, optimization, control, monitoring and diagnostics
- modeling tools, numerical methods for simulation, simulation tools, and technology
- visualization of modeling and simulation results

Application areas include: Automotive energy systems, Bio energy and biofuels, Biosystems and medical systems, Carbon capture and storage, Chemical process plants, Electric power systems, Energy storage, Energy and climate, Environmental impact, Fuel cells, Geothermal systems, Health and Welfare, Heat pumps, Hybrid and electrical vehicles, Hydro power, Industrial energy systems, Marine energy systems, Mechatronic energy systems, Oil and gas exploration and production, Photovoltaics, processing plants, Power stations, Refrigeration systems, Solar thermal systems, Thermal power plants, Wave and tidal power, Wind power.

A key component of the SIMS conference is the opportunity to socialize and make new connections. The conference offers several possibilities for networking including a conference dinner at the restaurant Olivia at Aker Brygge, a conference study visit to the VEAS water treatment plant, and last but not least several coffee and lunch breaks.

On behalf of the organizers, we wish to thank all participants, authors, keynote speakers and presenters for their contribution to this conference! We also want to acknowledge the support we have received from the conference board, the program committee, NFEA, and the SIMS board.

Tiina Komulainen, Lars Erik Øi, Robin T. Bye, and Lars O. Nord

Organizing committee:

Tiina Komulainen, Oslo Metropolitan University, Norway

Lars Erik Øi, University College of Southeast Norway, Norway

Robin T. Bye, NTNU – Norwegian University of Science and Technology, Norway

Lars O. Nord, NTNU – Norwegian University of Science and Technology, Norway

National Organizing Committee:

Tiina Komulainen, Oslo Metropolitan University

Evi Zouganeli, Oslo Metropolitan University

Alex Alcocer, Oslo Metropolitan University

Alfredo Carella, Oslo Metropolitan University

Anis Yazidi, Oslo Metropolitan University

Laura Marcano (PhD committee chair), Oslo Metropolitan University

Lars Erik Øi, University College of Southeast Norway

Bernt Lie, University College of Southeast Norway

Finn Haugen, University College of Southeast Norway

Britt Moldestad, University College of Southeast Norway

Samee Maharjan (PhD committee chair), University College of Southeast Norway

Robin T. Bye, NTNU – Norwegian University of Science and Technology

Lars Nord, NTNU – Norwegian University of Science and Technology

Luca Riboldi, NTNU – Norwegian University of Science and Technology

Dipo Olaitan, NTNU – Norwegian University of Science and Technology

Ottar L. Osen, NTNU – Norwegian University of Science and Technology

Ibrahim A. Hameed, NTNU – Norwegian University of Science and Technology

David Cameron, University of Oslo

Barbro Berg Bakken, Norwegian Society of Electric and Automatic Control (NFEA)

International Programme Committee:

Shobhana Singh, Aalborg University, Denmark

Brian Elmegaard, Technical University of Denmark, Denmark

Hamid Reza Shaker, University of Southern Denmark, Denmark

Weria Pezeshkian, University of Southern Denmark, Denmark

Esko Juuso, Oulu University, Finland

Mika Ruusunen, Oulu University, Finland

Kaj Juslin, Enbuscon Ltd., Finland

Kai Zenger, Aalto University, Finland

Magnus Jonsson, University of Skövde, Iceland

Silja Rán Sigurðardóttir, National Energy Authority, Iceland

Erik Dahlquist, Mälardalen University, Sweden

Lars Eriksson, Linköping University, Sweden

Conference location:

Oslo Metropolitan University, Pilestredet 46, NO-0167 Oslo, Norway

Opening ceremony and plenary session:

Conference opening

Monday 26 September 2018 at 09:00.

Conference opening speech:

Astrid Oust Janbu, Head of the Department of Mechanical, Electronics and Chemical Engineering, Oslo Metropolitan University, Norway

Keynote 1:

Can one do it all? The coming of unified system simulation tools including steady state, dynamics and optimization features

Jens Pålsson (PhD), Senior Application Engineer, Modelon, Sweden

Wednesday 26 September 2018 at 9:20

Keynote 2:

Process modelling and design as precursor to 3D printing in the chemical engineering field

Carlos Grande (PhD), Senior Research Scientist, SINTEF Industry – Process Technology

Thursday 27 September 2018 at 9:20

Conference study visit:

Study visit organized by Finn A. Haugen

VEAS water treatment plant

Friday 28 September 2018 at 9:00

Social program:

Conference dinner

Olivia Aker Brygge

Wednesday 26 September 2018 at 19:30

Papers

Exergy Analysis for Combined Heat and Power (CHP) Plants <i>Jairo Rúa and Lars O. Nord</i>	1
Oil and Gas digital twins after twenty years. How can they be made sustainable, maintainable and useful? <i>David Cameron, Arild Waaler and Tiina Komulainen</i>	9
CFD study on the effect of Archimedes number and heating rate on the thermal stratification of a ventilated office <i>Mehrdad Rabani, Habtamu Bayera Madessa and Natasa Nord</i>	17
CFD Simulation of Solidification of non-Newtonian Fluid Flowing in a Complex Geometry Pipeline in Turbulent Flow Regime <i>Ludmila Vesjolaja, Jakub M Bujalski and Knut Vaagsaether</i>	24
Simulation of condensation in compressed raw biogas using Aspen HYSYS <i>Lars Erik Øi and Jon Hovland</i>	31
FMI4j: A Software Package for working with Functional Mock-up Units on the Java Virtual Machine <i>Lars Ivar Hatledal, Houxiang Zhang, Arne Styve and Geir Hovland</i>	37
Model Predictive control of a district heating system <i>Frode Lie-Jensen, Andreas Aannø, Elena Aleksandrova, Anders Westli, Morten Nielsen and Tiina M. Komulainen</i>	43
Process Design of a Novel Low Temperature Methanol Synthesis Process Using an Air-blown Autothermal Reformer <i>Christian Ahoba-Sam, Lars Erik Øi and Klaus-Joachim Jens</i>	51
Simulation and Economic Optimization of Amine based CO ₂ Capture Process using Excess Heat at a Cement Plant <i>Hassan Ali, Lars Øi and Nils Henrik Eldrup</i>	58
Deriving the Beam Equation using the Minimum Total Potential Energy Principle and Solving the Equation Numerically <i>Magnus Komperød</i>	65
Experimental Design and Modeling for Propylene Oxide - CO ₂ -Poly (Propylene Carbonate) Solutions <i>Umesh Pandey, Kai Arne Sætre, Jostein Mathiassen, Sara Ronasi, Siw Bodil Fredriksen and Carlos F. Pfeiffer</i>	73
Study of Effectiveness of Operator Training Simulators in the Oil and Gas Industry <i>Ravikanth Kallakuri, Prakash Chandra Bahuguna, Donald Glaser and Sanjay Shivalkar</i>	79
Dynamic model of a heat pump based house heating system <i>Shobhana Singh and Kim Sørensen</i>	87

Population balance modelling for fertilizer granulation process <i>Ludmila Vesjolaja, Bjørn Glemmestad and Bernt Lie</i>	95
Convective Melting Modeling Approach for Phase Change Materials with Variable Boundary Heating <i>Donato Rubinetti, Daniel A. Weiss, Arnab Chaudhuri and Dimitrios Kraniotis</i>	103
Shock propagation and diffraction through cavity <i>Arnab Chaudhuri</i>	111
Comparison of different state estimator algorithms applied to a simulated anaerobic digestion reactor <i>Shadi Attar and Finn Aakre Haugen</i>	118
FVM-Modeling of Continuity-Coupled Electrical Charge Submitted to Incompressible Flow with OpenFOAM® <i>Donato Rubinetti and Daniel A. Weiss</i>	126
Using the concept of data enclosing tunnel as an online feedback tool for simulator training <i>Laura Marcano, Anis Yazidi, Davide Manca and Tiina Komulainen</i>	132
Numerical Solution of the Beam Equation for Beams subject to Large Deflections <i>Magnus Komperød</i>	140
Modeling and Simulation of Triclosan Kinetics and Distribution in Humans Using PBPK Model <i>Vincenza Cascella, Monica Andreassen, Trine Husøy, Hubert Dirven and Bernt Lie</i>	148
A Data-Driven Sensitivity Analysis Approach for Dynamically Positioned Vessels <i>Xu Cheng, Robert Skulstad, Guoyuan Li, Shengyong Chen, Hans Petter Hildre and Houxiang Zhang</i>	156
Web Enabled High Fidelity Drilling Computer Model with User-Friendly Interface for Education, Research and Innovation <i>Robert Ewald, Jan Einar Gravdal, Dan Sui and Roman Shor</i>	162
Machine Learning in Python for Weather Forecast based on Freely Available Weather Data <i>Erik Boye Abrahamsen, Ole Magnus Brastein and Bernt Lie</i>	169
Comparison of simulation tools for dynamic models <i>Sveinung M. Sund, Marianne Plouvier and Bernt Lie</i>	177
Comparison of linear controllers for nonlinear, open-loop unstable reactor <i>Mohammad Khalili and Bernt Lie</i>	185
Second Order KP Scheme for the Solution of flow in a Venturi Channel <i>Susantha Dissanayake, Roshan Sharma and Bernt Lie</i>	193
The effects of fan and opening of a sliding door on a cold storage room: a numerical study <i>Azada Ayarmal, Ole Melhus and Arnab Chaudhuri</i>	201
Developing Simulation Tools for Interdisciplinary Modelling <i>Sigve Karolius and Heinz Preisig</i>	210

Linearization for Analysis of a Hydropower Model using Python API for OpenModelica <i>Liubomyr Vytvytskyi and Bernt Lie</i>	216
Decision Trees for Human Activity Recognition in Smart House Environments <i>Veralia Gabriela Sanchez and Nils-Olav Skeie</i>	222
Comparison of simulation tools to fit and predict performance data of CO ₂ absorption into monoethanol amine at CO ₂ Technology Centre Mongstad (TCM) <i>Lars Erik Øi, Kai Arne Sætre and Espen Steinseth Hamborg</i>	230
Occupancy and daily activity event modelling in smart homes for older adults with mild cognitive impairment or dementia <i>Flávia Dias Casagrande and Evi Zouganeli</i>	236
Model-order selection of output-error models – BSM1 as case study <i>Christian Wallin and Jesús Zambrano</i>	243
Distribution of Solids in a Fluidized Bed Operated without a Gas Distributor <i>Cornelius Agu and Britt Moldestad</i>	248
Simulation of CO ₂ storage in the North Sea <i>Arne O. Torsen, Harris J. Smistad, Håkon Tveit, Ole C. Hansen, Vegard G. Bjørtuft, Nora C. I. Furuvik and Britt M. E. Moldestad</i>	255
Improving bench scale pilot reactor design for polymerisation of green CO ₂ -polymers <i>Thomas S. Larsen, Kai A. Sætre, Sara Ronasi, Hildegunn H. Haugen and Britt M. E. Moldestad</i>	263
Study of fluidized bed regimes using Computational Particle Fluid Dynamics <i>Rajan Jaiswal, Cornelius E. Agu, Britt M. E. Moldestad and Rajan Kumar Thapa</i>	271
Simulation Model for Borehole Heat Exchangers <i>Petri Hietaharju and Tobias Boström</i>	277
A Mean Value Model for Unsteady Gas Flows and Heat Transfer in Pipes <i>Olov Holmer and Lars Eriksson</i>	284
Validation of an Open-Source Mean-Value Heavy-Duty Diesel Engine Model <i>Viktor Leek, Kristoffer Ekberg and Lars Eriksson</i>	290
Simulations and real applications of PI and MPC averaging level control in a water resource recovery facility <i>Finn Aakre Haugen</i>	297
Simulation of the Pyrolysis Process: from Plastic Waste to Environmental Friendly Fuel <i>Marius Andersen, Carlos Pfeiffer, Kai Arne Sætre and Siw Fredriksen</i>	303
Intelligent Multimodel Simulation in Decomposed Systems <i>Esko Juuso</i>	308
Experimental and Computational study of Chemical Looping Combustion <i>Rajan K. Thapa and Britt M. E. Moldestad</i>	316
Optimization of Bioreactor Volumes in Steady-State – A Simulation Study <i>Hanna Molin, Jesús Zambrano and Bengt Carlsson</i>	322

System Development for On-line Monitoring using Raman Spectroscopy for CO ₂ Absorption by MEA <i>M.H. Wathsala N. Jinadasa, Kahawalage Amila Chandra and Maths Halstensen</i>	328
Sensitivity Analysis and Effect of Simulation parameters of CPFD Simulation in Fluidized Beds <i>Janitha Bandara, Henrik Nielsen, Britt Moldestad and Marianne Eikeland</i>	334
The impact of the weather data file on the energy performance certificate, the case of Norway <i>Alex Gonzalez, Tor Arvid Vik and Dimitrios G. Zengin</i>	342

Exergy Analysis of Combined Heat and Power (CHP) Plants

Jairo Rúa Lars O. Nord

Department of Energy and Process Engineering, NTNU - Norwegian University of Science and Technology,
Trondheim - Norway, jairo.r.pazos@ntnu.no

Abstract

Combined heat and power plants are an efficient technology for waste heat recovery applications where both heat and power are demanded. Steam Rankine cycles are normally utilized when high temperature exhaust streams are available. Different heat demands and temperatures might be expected depending on the application. Thus, designing a power system capable of meeting the heat demand and producing the maximum possible power supposes a challenge. This work aims at providing some guidelines to properly select, from a second law perspective, a CHP plant configuration depending on the heat demand and temperature. Two plant configurations are considered and broad ranges of heat demand and temperature are studied. A software developed in MATLAB was utilized for all thermodynamic property calculations. Optimization of the thermodynamic cycle was carried out for each set of conditions and the best possible scenario was always evaluated. An exergy analysis of the components integrating the power plant was then performed, pointing out where irreversibilities were more predominant and where there was more potential for improvements. Results showed that extracting process heat in the condenser was more beneficial than utilizing a supplementary heat exchanger in series with the heat recovery steam generator.

Keywords: steam Rankine cycle, waste heat recovery, gradient-based optimization, Second law analysis.

1 Introduction

Uncontrolled and prolonged emissions of anthropogenic greenhouse gases (GHG) have increased by 40% the atmospheric concentrations of CO₂ with respect to pre-industrial levels. Combustion of fossil fuels is the main source of these anthropogenic GHG emissions, which accounts for 78% of the total emission increase in the period from 1970 to 2010. Accordingly, there is a scientific consensus that this irresponsible human behaviour is one of the major drivers of global warming and climate change (IPCC, 2014). However, power generation and industry are expected to rely on fossil fuel combustion until the end of the century (IEA, 2016). Efficiency improvement is considered a promising mitigation alternative that aims at reducing the CO₂ emissions on these sectors (IEA, 2016).

Waste heat recovery from both power generation systems and energy intensive industries is expected to lead to higher efficiencies and lower CO₂ emissions (Nord and

Bolland, 2013; Quoilin et al., 2013). Combined heat and power (CHP) plants are suitable alternatives for waste heat applications where there is a demand of both heat and power. Nevertheless, the design of these systems is a major challenge due to the broad range of exhaust gas temperatures and the different heat demand and temperature.

Exergy analysis accounts for the irreversibilities that take place in each component of the cycle and leads to designs where maximum use of the available exergy is achieved. Many studies analyse complex systems in order to identify such irreversibilities. However, some of them do not motivate, economically or thermodynamically, the selection of a specific plant configuration. From an engineering perspective, both adequate thermodynamic performance and plant structure are fundamental choices that determine the efficient operation of a power or CHP plant. Consequently, it is not possible to ensure peak performance of any power system if only one of these two factors is motivated. Ertesvåg (2007) defined a parameter to compare the exergy improvements that are achieved using a CHP plant with the separate production of heat and power. Results showed that industrial CHP plants with gas turbines, bypass in the condenser for heat production, and back-pressure steam turbines are beneficial from an exergy perspective. Wang et al. (2009) carried out an exergy optimization and analysis of different types of cycles and configurations for waste heat recovery in a cement plant. They concluded the Kalina cycle leads to the highest efficiency, although the simple flash steam cycle leads to good performance with a simpler cycle configuration. Ahmadi and Dincer (2010) performed an exergo-economic analysis of an existing CHP plant where heat demand and temperature were defined. The effect of power demand and fuel cost on different parameters of the cycle was studied through a sensitivity analysis. It was found that economic saving may be achieved if the suitable turbomachinery and an adequate CHP plant design are selected according to a expected power demand.

This work aims at providing some guidelines for the selection of optimal CHP plant configurations for waste heat recovery applications at high temperature. In Section 2 two different CHP plants for producing a wide range of temperature and heat demands are described, the exergy analysis methodology is presented, and the optimization procedure for maximizing the power output is detailed. Results for both CHP plant configurations are shown and compared in Section 3.

2 Analytical Procedure

2.1 Case Study

The case study analysed in this work is a combined cycle where a flow of exhaust gas at 550 °C is utilized to produce the steam that drives the turbine in a simple Rankine cycle and the process heat required. Water is selected as working fluid due to the high temperature of the hot fluid (Hung et al., 1997). Waste heat recovery applications, such as those on offshore platforms or ships are normally subjected to weight and space limitations. Therefore, a single pressure once-through steam generator is utilized as a trade-off between compactness and heat exchange efficiency (Nord and Bolland, 2012, 2013). Two different configurations, whose layouts are shown respectively in Figure 1 and Figure 2, are studied:

1. Process heat is generated in the condenser with the heat rejected by the steam during its condensing before being pumped to restart the cycle. In this scenario, a back-pressure steam turbine may be utilized in order to leave some energy in the steam that enables the production of the demanded amount of heat at the required temperature. As process heat is produced in the condenser, the exhaust flow is entirely utilized to generate the maximum amount of steam for a certain temperature at the inlet of the turbine.
2. A supplementary heat exchanger is installed after the heat recovery steam generator (HRSG) to produce the required process heat. Therefore, it is not possible to employ all the exhaust gas energy for the generation of steam as there must be some available for the production of process heat. However, in this case, the generation of power is not limited by the back-pressure of the steam turbine as no useful energy is going to be obtained from the condenser. Maximum possible expansion is hence occurring in the steam turbine.

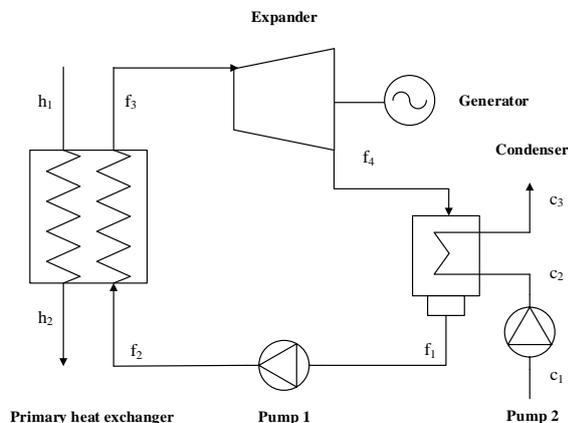


Figure 1. Simple Rankine cycle layout with heat extraction in the condenser.

Table 1. Specification assumed for the case study.

Variable	Nomenclature	Value
Ambient Temperature	T_0	10 °C
Ambient Pressure	p_0	1 atm
Heat Source Temperature	T_{h1}	550 °C
Heat Source Mass Flow Rate	\dot{m}_h	80 kg/s
Heat Source Outlet Pressure	p_{h2} or p_{h3}	1 atm
Process Heat Fluid Temperature	T_{c1} or T_{p1}	25 °C
Process Heat Fluid Outlet Pressure	p_{c3} or p_{p2}	2.5 atm
Process Heat Fluid Inlet Pressure	p_{c1} or p_{p1}	1.5 atm
Cooling Fluid Temperature	T_{c1}	10 °C
Heat Source Composition	CO_2	5.23 mol-%
	H_2O	4.29 mol-%
	O_2	15.24 mol-%
	N_2	75.24 mol-%

Almost all exhaust gases originated from the combustion of fossil fuels contain sulphurs that may condensate and cause corrosion when the temperature is excessively low. Thus, in practice, a minimum temperature at the outlet of the heat exchanger is fixed in order to avoid such problems. In this case, a "clean" gas stream was assumed. Limitations related to practical operation issues were not considered and hence a exhaust gas low temperature limit of 40°C was used. Pressurized water at 2.5 bar was utilized as both fluid media in the process heat generation and as cooling fluid in the condenser for configuration 2. Process heat demand and temperature were varied during this study so the effect of these variables on the exergy utilization and efficiency may be observed. A summary of the specifications assumed in this work is in Table 1.

2.2 Exergy Analysis

Steady-state mass and energy conservation laws are implemented in every component so the inlet and outlet states of every component may be calculated. The REFPROP library (Lemmon et al., 2013) is utilized to determine the required thermodynamic variables in every state

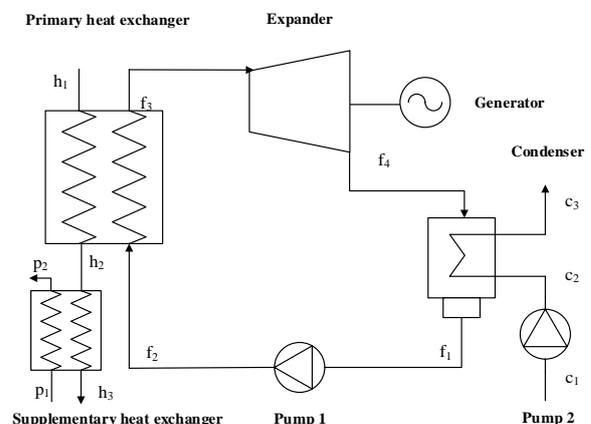


Figure 2. Simple Rankine cycle with supplementary heat exchanger layout.

(IAPWS formulation for the thermodynamic properties of water and steam). Once the thermodynamic states that specify the components' operation are known, exergy calculations can be performed. Exergy content is calculated in every state and thus, by means of exergy balances, the exergy transfer and destruction that occurs in the components integrating the cycle are calculated. More details about the implementation of this procedure in MATLAB can be found in the work done by Agromayor and Nord (2017).

Thermo-mechanical exergy was exclusively considered. When exergy balances are considered for every component, the chemical exergy between two different states cancels out. Point h_1 is the only state where it could be considered, being included in the exergy losses associated to the exhaust gas that is released to the atmosphere and thus reducing the second law efficiency of the plant. However, special ideal membranes that allow the individual expansion of the components integrating the exhaust gas stream would be needed (Kotas, 2012). This is unrealistic for a power plant and therefore chemical exergy is not utilized in the proposed case. In addition, kinetic and potential energies, and its contribution in the exergy calculations, were neglected.

2.3 Optimization

Second law plant efficiency, defined in Eq. 1, was selected as the objective function to be maximized in the Rankine cycle optimization. The numerator represents the net exergy extracted from the turbine in the process, which accounts for the net power produced in the bottoming cycle. Process heat, i.e. the thermo-mechanical exergy contained in the process heat stream (\dot{E}_{c3} for configuration 1 and \dot{E}_{p2} for configuration 2), is considered as a constraint in the optimization problem since power plant operators normally prefer to produce electricity rather than heat. The denominator is the thermo-mechanical exergy contained by the heat source at the inlet of the primary heat exchanger, \dot{E}_{h1} . Selecting this parameter as denominator allows to know how well the exergy content of the hot stream is utilized for the desired purposes, which is one of the main objectives of a waste heat recovery application.

$$\eta_{II,plant} = \frac{\dot{W}_{net}}{\dot{E}_{h1}} \quad (1)$$

Net power is defined in Eq. 2 as the difference between the power produced by the steam turbine and the power consumed by the pumps. Process heat exergy, \dot{E}_{heat} , accounts for the work that could be obtained if the fluid was taken reversibly to restricted equilibrium (Kotas, 2012) (see Eq. 3).

$$\dot{W}_{net} = |\dot{W}_{turb}| - |\dot{W}_{pump_{cycle}}| - |\dot{W}_{pump_c}| - |\dot{W}_{pump_{sup}}| \quad (2)$$

$$\dot{E}_{heat} = \dot{m}_{heat} e_{heat} = \dot{m}_{heat} [(h - h_0) - T_0 (s - s_0)] \quad (3)$$

During the optimization many variables are kept constant. A summary of the values of these variables is presented in Table. 2.

Table 2. Fixed variables during the Rankine cycle optimization.

Variable	Nomenclature	Value
HRSG Pressure Drop Heat Source Side	$\Delta p_{h,HRSG} [\%]$	2
HRSG Pressure Drop Working Fluid Side	$\Delta p_{w,HRSG} [\%]$	2
Condenser Pressure Drop Cooling Fluid Side	$\Delta p_c [\%]$	2
Condenser Pressure Drop Working Fluid Side	$\Delta p_{w,cond} [\%]$	1
Supplementary HE Pressure Drop Working Fluid Side	$\Delta p_{w,sup} [\%]$	1
Supplementary HE Pressure Drop Heat Source Side	$\Delta p_{h,sup} [\%]$	1
Expander Polytropic Efficiency	$\eta_{turb} [\%]$	80
Pump Polytropic Efficiency	$\eta_{pump} [\%]$	70
HRSG Minimum Temperature Difference	$\Delta T_{min,HRSG} [^{\circ}C]$	10
Condenser Minimum Temperature Difference	$\Delta T_{min,c} [^{\circ}C]$	10

In the model employed in this work, five degrees of freedom are enough to sequentially calculate all the thermodynamic states without solving any system of algebraic equations. These degrees of freedom are:

1. Outlet temperature of the heat source.
2. Pressure at the inlet of the turbine.
3. Pressure at the outlet of the turbine.
4. Enthalpy at the inlet of the turbine.
5. Enthalpy at the inlet of the HRSG.

3 Results and Discussion

Simulations for process heat demand varying between 30 and 42 MW, and temperature requirements ranging from 50 to 90 °C were performed. The temperature interval was selected based on a low quality heat demand where the produced heat will be used for low temperature applications. The choice of the process heat demand range was based on an optimization of the cycle without any constraint on the heat demand and an outlet temperature of 50 °C at the outlet of the condenser. It was observed that the heat flow rate in the condenser was 25 MW and hence higher values had to be tested. An upper limit to the maximum process heat that could be produced was found close to 44 MW, where the cycle needed to consume power instead of producing it, i.e. a negative second law efficiency was obtained. The selected range was 30 - 42 MW in order to ensure that sufficient heat (limited by the lower bound) and power (limited by the upper bound) were produced in all cases.

3.1 Simple Rankine Cycle

Second law efficiency variation with heat demand and temperature may be observed in Figure 3. As it could be expected, increasing temperature and heat demand have a negative effect on the second law efficiency. From Figure 4 can be observed that two opposite tendencies are found in the cycle. Irreversibility becomes larger in the HRSG and the condenser as the heat demand increases, but less exergy is destroyed in the turbine and the pump (not shown due to small scale) when the demand of heat is

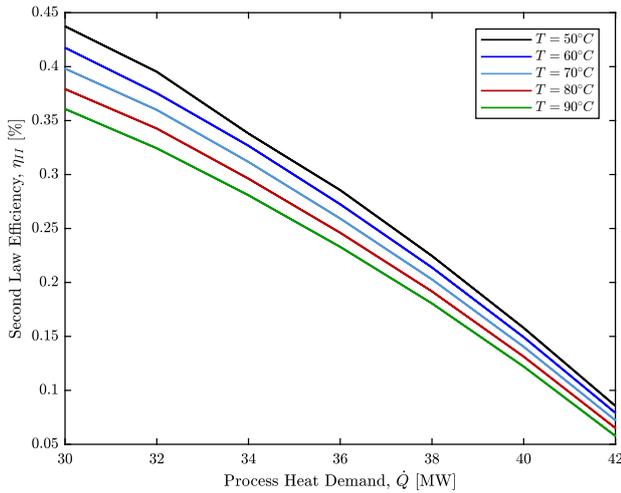


Figure 3. Second law efficiencies for simple Rankine cycle.

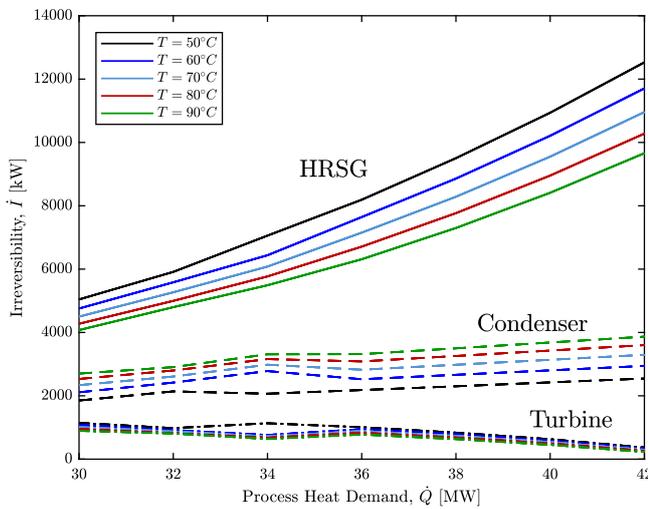


Figure 4. Irreversibility variation of the components of the simple Rankine cycle.

raised. The increase of both heat exchanger's irreversibility with the heat demand outweighs the reduction in exergy destruction that occurs in the turbomachinery and hence the overall effect of the heat demand on the second law efficiency is negative, i.e. it is reduced as the heat demand raises.

Mathematically, the connection between the reduction in the turbine's irreversibility with the increase in the process heat can be proven. Combining the polytropic efficiency of the turbine η_p shown in Eq. 4, and Eq. 5, where subscript s stands for isentropic,

$$\eta_p = \frac{dh}{dh_s} \quad (4)$$

$$T ds = dh - v dp \quad \text{and} \quad dh_s = v dp \quad (5)$$

the following expression is obtained:

$$\eta_p = \frac{T ds + v dp}{v dp} = 1 + \frac{T ds}{v dp} \quad (6)$$

If ideal gas behaviour of the steam is assumed (only for qualitative analysis purposes), Eq. 6 becomes:

$$ds = R(\eta_p - 1) \frac{dp}{p} \quad (7)$$

Integrating between the inlet and the outlet of the turbine:

$$\Delta s = R(1 - \eta_p) \ln\left(\frac{p_3}{p_4}\right) \quad (8)$$

It is possible to observe that entropy increment, and hence irreversibility (see Eq. 8), depend on the pressure ratio in the turbine and the polytropic efficiency. Since the polytropic efficiency is considered constant, the exergy destruction in the turbine only depends on the pressure ratio. From Figure 5 it can be seen that the pressure ratio decreases with increasing heat demand. Therefore, it is shown that the irreversibility in this case decreases with higher heat requirements.

In the pump's case the ratio between the specific volume and temperature in Eq. 6 is constant, and hence the following expression is obtained:

$$\Delta s = K(1 - \eta_p)(p_2 - p_1) \quad (9)$$

with K being a constant that represents the ratio between the specific volume and temperature, where the former is constant because water is an incompressible fluid and the variation in the latter is neglected since it is small. Thus, it is shown that, as it occurred in the turbine, the irreversibility in the pump decreases as the process heat demand increases.

From a second law analysis of a heat transfer process between a hot and a cold stream it can be shown that the entropy generation $\dot{\sigma}$, and thus the irreversibility, are proportional to the temperature difference and the heat exchanged. This can be proven assuming a heat exchange process between constant hot and cold flows without friction. The control volume is set so the heat transfer process occurs at T_H , and all the entropy generation is included. For the cold fluid, the first and second law, and the T-s equations are, respectively:

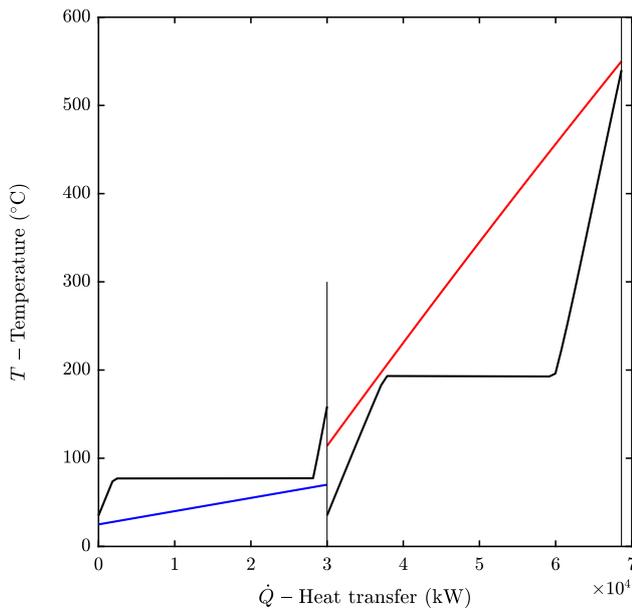
$$0 = \delta \dot{Q} + \dot{m}_C dh_C \quad (10)$$

$$0 = \frac{\delta \dot{Q}}{T_H} + \dot{m}_C ds_C + \delta \dot{\sigma} \quad (11)$$

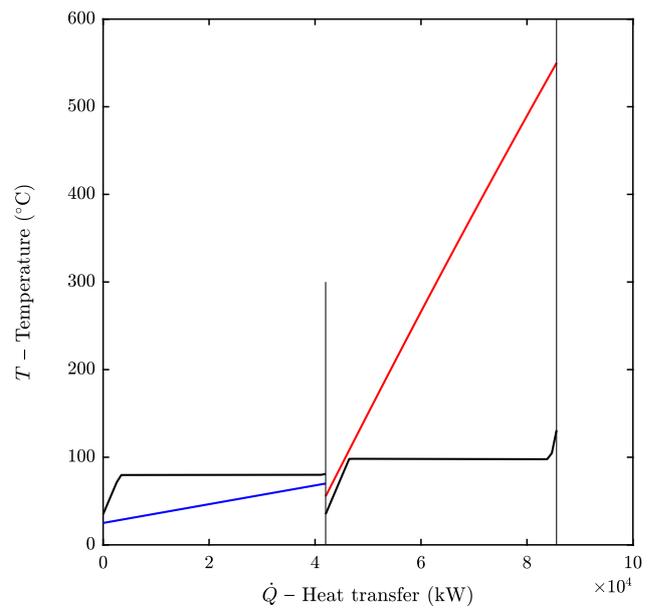
$$T_C ds_C = dh_C \quad (12)$$

Combining these three equations the following expression showing the relation between the entropy production and the existence of a finite temperature difference is obtained:

$$\delta \dot{\sigma} = \frac{T_H - T_C}{T_H T_C} \delta \dot{Q} \quad (13)$$

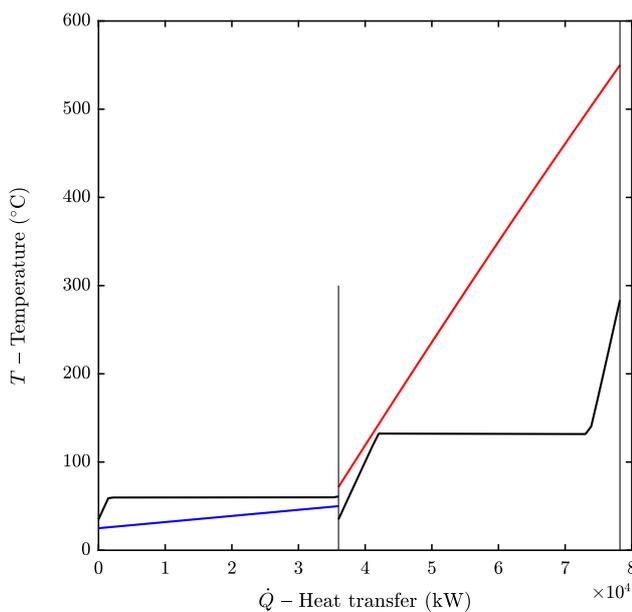


(a) T-Q diagram for $T = 70^{\circ}\text{C}$ and $Q = 30\text{ MW}$.

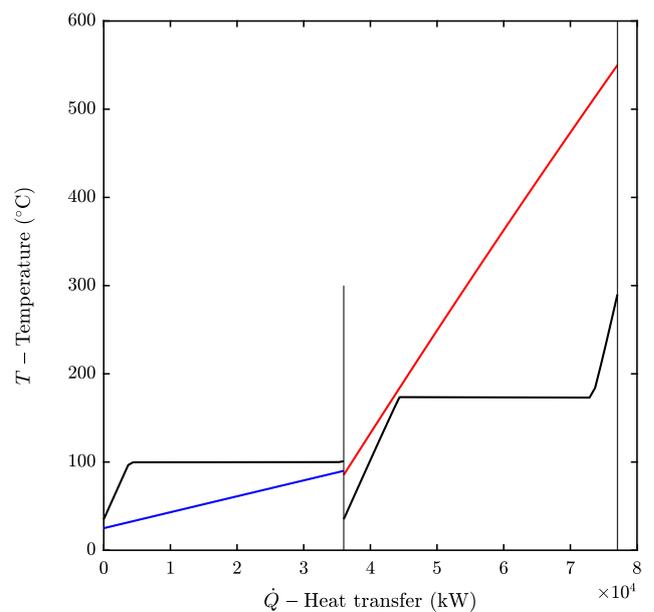


(b) T-Q diagram for $T = 70^{\circ}\text{C}$ and $Q = 42\text{ MW}$.

Figure 5. Effect of process heat demand on the optimal cycle configuration for the simple Rankine cycle.



(a) T-Q diagram for $T = 50^{\circ}\text{C}$ and $Q = 36\text{ MW}$.



(b) T-Q diagram for $T = 90^{\circ}\text{C}$ and $Q = 36\text{ MW}$.

Figure 6. Effect of process heat temperature on the optimal cycle configuration for the simple Rankine cycle.

Consequently, for the heat recovery steam generator, the heat transferred increases with the process heat demand and so does the irreversibility. In addition, high process heat demand implies less pressure ratio, which increases the temperature difference between the hot stream of exhaust gases and the pressurized working fluid along the HRSG, leading to higher entropy generation. Lastly, smaller pressure ratios mean that the overall cold stream temperature in the heat recovery steam generator is reduced, and hence the denominator in Eq. 13 is smaller for larger process heat demands.

The irreversibility in the condenser is also increased with the process heat demand because the heat transferred is larger. This effect may be observed in Figure 4, where there is a linear tendency for the condenser irreversibility lines. Different back-pressures at the outlet of the turbine originated by different process heat demands may lead to different hot temperatures along the condenser, which would imply different temperature differences and modifications in the denominator's product. However, these changes balance each other and the linear tendency due to the heat transferred dominates the exergy destruction.

Temperature influence in the second law efficiency of the power plant can also be understood if irreversibility in each component is analysed. High temperatures lead to smaller pressure ratios in the cycle for a given process heat demand (see Figure 6). Therefore, the irreversibility in the turbine and the pump decreases as the required temperature increases (see Eq. 8 and Eq. 9, respectively).

The effect of temperature in the heat recovery steam generator and the condenser can be clearly observed in Figure 4. The temperature in the process heat stream has a strong effect in the high and low pressures of the cycle. These pressures influence the temperature difference between the hot and cold streams in each heat exchanger and hence affect the exergy destruction taking place in both components.

In case of the heat recovery steam generator, the high pressure increase associated to the raise in the temperature of the process heat stream leads to a reduction of the irreversibility. Larger high pressures imply higher saturation temperature and hence the overall temperature difference along the component decreases, leading to smaller exergy destruction. The saturation temperature increase in the heat recovery steam generator also indicates larger overall cold temperature along the component, which enhances the reduction in entropy generation (see Eq. 13).

High temperatures in the outlet of the cold stream of the condenser lead to larger temperature differences along the component. This effect outbalances the increase of the denominator product associated to the increase of the overall temperature of the cooling fluid. Therefore, the entropy generation is boosted by the increase in the temperature requirements for the process heat stream.

From this analysis, it could be thought that the temperature raise of the process heat stream could lead to an improvement of the second law efficiency as the irreversibility of the heat recovery steam generator, the component with larger effect on this parameter, decreases for higher temperatures. However, in addition to an exergy destruction analysis, it is important to take into account the utilization of the exhaust gas stream and the power produced in the turbine. Exergy flow to the cycle and power generation in the turbine decrease as the temperature is increased. Thus, albeit the smaller irreversibility in the HRSG, high temperature requirements in the heat stream lead to lower values of the second law efficiency because a larger fraction of exergy leaves with the exhaust gases without being recovered by the cycle. The exergy content of the exhaust gases is simply destroyed in the ambient without being utilized.

3.2 Rankine Cycle with Supplementary Heat Exchanger

Second law efficiency variation with process heat demand and temperature are shown in Figure 7. Similarly to what occurred with the simple Rankine cycle configuration, second law efficiency decreases as the process heat demand is raised. However, temperature has no effect on this

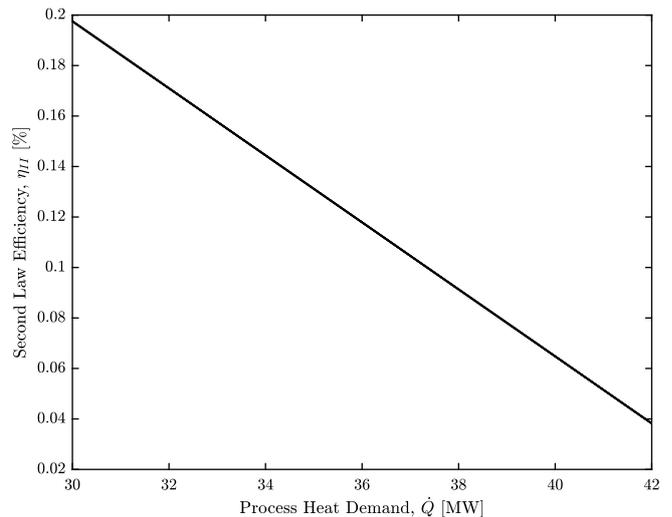


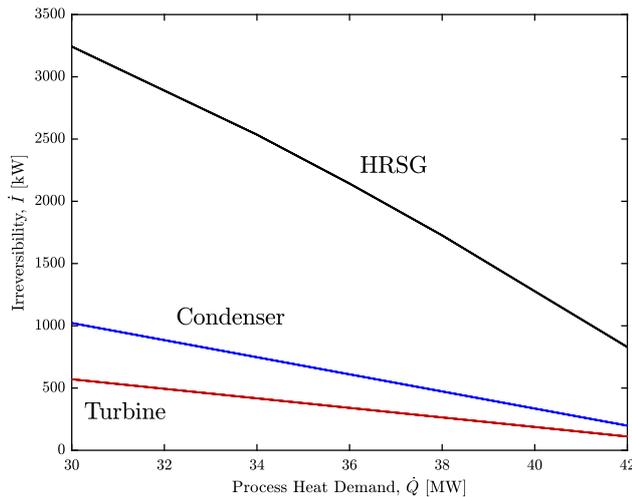
Figure 7. Second law efficiency variation of the Rankine cycle with Supplementary HE.

parameter as it can be clearly observed. In addition, and as it could be expected from the non-variation of the second law efficiency with process heat temperature, irreversibility in each of the cycle's components is also independent from the process heat temperature. This behaviour may be observed in Figure 8a.

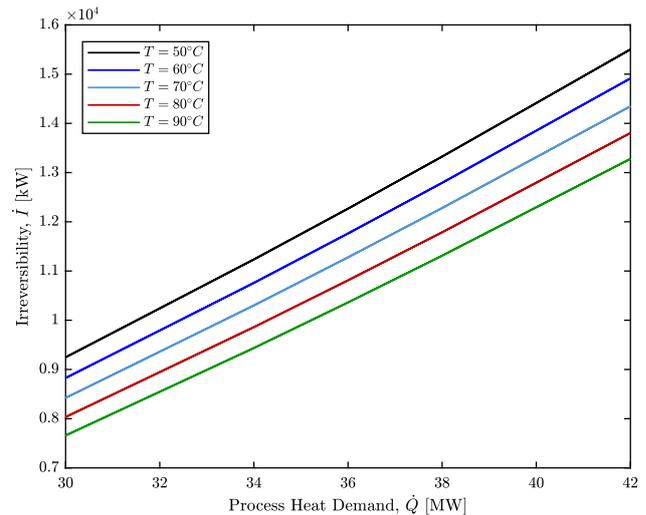
Invariance of the second law efficiency with process heat temperature can be explained from an optimization perspective. Described before, during the optimization procedure, the outlet temperature required for the process heat stream is fixed while the amount of demanded heat is set as an equality constraint. When the required temperature is the lowest, i.e. 50 °C, a maximum for the efficiency is found with an associated temperature at the outlet of the HRSG of the exhaust gases close to or above 400 °C. Thus, when the temperature requirements are increased, the same optimum point, which is the best possible point for the established conditions, is achieved since the energy contained in the exhaust gases at the outlet of the HRSG is enough to produce the demanded heat at higher temperatures. The obtained results lead to the same thermodynamic cycle. Consequently, as the objective function is the produced power in the turbine divided by the exergy content of the hot stream at the inlet of the HRSG (see Eq. 1), the second law efficiency does not change.

Identical thermodynamic cycles for different process heat temperatures imply that the pressure ratio in the turbine and the pressure difference in the pump do not change. Therefore, according to Eq. 8 and Eq. 9, exergy destruction in both components does not vary with the temperature of the process heat stream.

On the contrary, the irreversibility in the supplementary heat exchanger *does* change with the process heat temperature as it can be observed in Figure 8b. Different temperature in the process heat stream implies that, for the same hot inlet temperature in the supplementary heat exchanger, distinct temperature differences occur along the



(a) Irreversibilities of HRSG, condenser and turbine.



(b) Irreversibilities of supplementary heat exchanger with process heat and temperature.

Figure 8. Irreversibility variation of the components of the Rankine cycle & supplementary HE with process heat demand and temperature.

component. Hence, the higher the process heat temperature requirements the smaller the "overall" temperature difference and the irreversibility in the component. This behaviour is explained mathematically by Eq. 13.

Heat demand has a clear linear influence in the second law efficiency of the power plant and in the irreversibility in each component. On the contrary to what occurred for the simple Rankine cycle, exergy destruction in all cycle components decreases with the increase in the heat demand. Big changes in the high and low pressures of the cycle do not occur and hence the irreversibility is reduced as the heat demand is raised. Heat transferred in the heat recovery steam generator and the condenser decreases as the heat demand increases. Eq. 13 shows the linear relation between heat transferred and entropy generation (and consequently exergy destruction), and hence it can be understood why the irreversibility in these heat exchangers is decreased as the process heat demand increases. The "overall" hot temperature and thus the temperature difference are higher for larger heat demands, however, this effect is clearly outbalanced by the changes in the heat transfer. The weight of the heat transfer increase in the entropy generation and the linear behaviour it produces is easily observable in Figure 8.

3.3 Configuration Comparison

From the previous discussions in the sections above two distinguishable behaviours may be observed. The simple Rankine cycle is affected by both process heat demand and temperature since this heat is produced in the condenser and consequently affects the thermodynamic cycle directly. On the other hand, having a supplementary heat exchanger after the HRSG together with the low temperature requisites allows the second configuration to be independent of the process heat temperature. This feature

could be regarded as an advantage for the second configuration but, due to the requirements imposed during the optimization, the cycle design is strongly penalized, reaching much lower efficiencies (both first and second law) than the simple Rankine configuration.

From the efficiency definitions and the results presented it can be deduced that the simple Rankine configuration produces more electric power than the configuration with supplementary heat exchanger for an equal set of process heat requirements. Larger irreversibilities are found in the components of the first cycle than in the second one for this reason, as more working fluid is circulating, more power is produced in the turbine, more power is consumed by the pump and larger temperature differences are encountered along the HRSG and the condenser. Exergy destruction is thus concentrated in the supplementary heat exchanger in the second configuration, being its magnitude almost as large as all the irreversibilities of the cycle's components in the simple Rankine cycle configuration.

4 Conclusions

In this work two different configurations of a combined heat and power plant were proposed for waste heat recovery applications. The first configuration produced the process heat in the condenser by utilizing a back-pressure steam turbine. In the second alternative, a supplementary heat exchanger after the heat recovery steam generator was considered. An optimization procedure of each cycle for different process heat demands and temperature was done in order to obtain the best possible design for each case. The second law efficiency was employed as the objective function and, in addition to the thermodynamic states, the irreversibility was calculated in each component.

Both qualitative and quantitative analyses from a second law perspective were done for both power plant configurations. It was found that in both *cycles* the HRSG design was the main source of entropy generation due to the large heat that was transferred and the big temperature differences. However, when the second power plant configuration was analysed, the largest irreversibility occurred in the supplementary heat exchanger as a consequence of the large heat transferred and the temperature difference. Heat demand, and not process heat temperature, was the process heat parameter that had the largest effect on both efficiencies, being the variable that limits the most the power generation in the steam turbine.

The power plant with supplementary heat exchanger was independent of the process heat temperature due to the nature of the optimal solution, leading to poor efficiencies and low power production. As a consequence, the simple Rankine cycle is considered as the best of the two proposed alternatives because it does not only have better first and second law efficiencies but it is also able to generate more power for a specific process heat.

References

- R. Agromayor and L. O. Nord. Fluid selection and thermodynamic optimization of organic Rankine cycles for waste heat recovery applications. *Energy Procedia*, 129:527–534, 2017. doi:10.1016/j.egypro.2017.09.180.
- P. Ahmadi and I. Dincer. Exergoenvironmental analysis and optimization of a cogeneration plant system using Multimodal Genetic Algorithm (MGA). *Energy*, 35(12):5161–5172, 2010. doi:10.1016/j.energy.2010.07.050.
- I. S. Ertesvåg. Exergetic comparison of efficiency indicators for combined heat and power (CHP). *Energy*, 32(11):2038–2050, 2007. doi:10.1016/j.energy.2007.05.005.
- T. C. Hung, T. Y. Shai, and S. K. Wang. A Review of Organic Rankine Cycles (ORC) for the Recovery of Low-Grade Waste Heat. *Energy*, 22(7):661–667, 1997. doi:10.1016/j.energy.2010.07.050.
- IEA. World Energy Outlook 2016. , OECD/IEA, Paris, France, 2016.
- IPCC. Climate Change 2014: Synthesis Report. Contribution of Working Groups I, II and III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change [Core Writing Team, R.K. Pachauri and L.A. Meyer (eds.)], IPCC, Geneva, Switzerland, 2014.
- T. J. Kotas. *The Exergy Method of thermal Plant Analysis*. Exergon Publishing Company UK Ltd, 2012. ISBN 978-1-908341-89-1.
- E. W. Lemmon, M. L. Huber, and M. O. McLinden. NIST Standard Reference Database 23: Reference Fluid Thermodynamic and Transport Properties-REFPROP, Version 9.1, National Institute of Standards and Technology, 2013.
- L. O. Nord and O. Bolland. Steam bottoming cycles offshore - Challenges and possibilities. *Journal of Power Technologies*, 92(3):201–207, 2012. doi:10.1016/j.energy.2010.07.050.
- L. O. Nord and O. Bolland. Design and off-design simulations of combined cycles for offshore oil and gas installations. *Applied Thermal Engineering*, 54(1):85–91, 2013. doi:10.1016/j.applthermaleng.2013.01.022.
- S. Quoilin, M. Van Den Broek, S. Declaye, P. Dewallef, and V. Lemort. Techno-economic survey of Organic Rankine Cycle (ORC) systems. *Renewable and Sustainable Energy Reviews*, 22:168–186, 2013. doi:10.1016/j.rser.2013.01.028.
- J. Wang, Y. Dai, and L. Gao. Exergy analyses and parametric optimizations for different cogeneration power plants in cement industry. *Applied Energy*, 86(6):941–948, 2009. doi:10.1016/j.apenergy.2008.09.001.

Oil and Gas digital twins after twenty years. How can they be made sustainable, maintainable and useful?

David B. Cameron¹ Arild Waaler¹ Tiina M. Komulainen²

¹SIRIUS Centre, Department of Informatics, University of Oslo, Norway, {davidbc, arild}@uio.no

²Faculty of Technology, Art and Design, Oslo Metropolitan University, Norway, timako@oslomet.no

Abstract

The digital twin offers a potentially powerful way of using simulation to support business and change the way industrial operations are done. The idea of the digital twin is not new but recent changes in information technology make implementation of digital twins a natural next step in the application of simulation technologies. Simulation practitioners will find that their models are increasingly embedded in complex systems that combine simulations with operational data to solve a business problem. However, the successful adoption of this approach is challenging. This paper asks the question: “How can digital twins be made sustainable, maintainable and useful?”. We focus primarily on the development of twins in the oil and gas industry. Most academic work in this area has been done in the manufacturing industries. We review this literature and propose a simple model of digital twins. This allows us to identify challenges with current implementations and propose a research agenda that will allow future twins to be sustainable, maintainable and usable.

Keywords: digital twin, semantics, integration, on-line simulation, Industrie 4.0, hybrid analytics.

1 Introduction

The digital twin is one of the buzzwords of the last year. For example, Gartner group identified digital twins as one of the top ten strategic technology trends for 2017 (Panetta, 2016) and 2018 (Panetta, 2017). As Gartner themselves note, the idea of a digital twin is not new, as the discipline of engineering simulation is at least thirty years old. Klostermeier, Haag & Benlian (2018) attribute the twin concept to researchers in NASA, with roots back into the Apollo program. They also note that the digital twin may be seen as a fourth milestone in the application of simulation technology, as shown in Figure 1. This model is useful, but it obscures work done in the last decades where simulations were aligned with operational data to solve localized control and monitoring tasks.

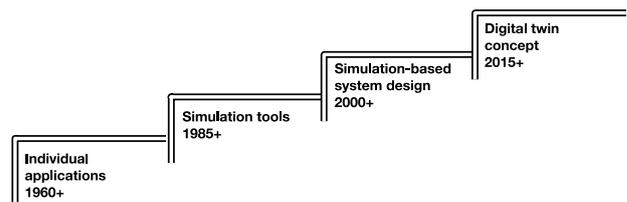


Figure 1. Milestones in simulator application (from Klostermeier et al. (2018))

For example, in the oil and gas sector, it is now twenty years since the first implementation of an on-line flow assurance system (Hyllseth & Cameron, 2003). This provided a digital twin of a multiphase pipeline from the Oseberg field in the North Sea.

As simulation and automation engineers, we should welcome this new, high level focus on using simulation to support operational decisions. However, we see that there are important issues related to sustainability, maintainability and usability that must be addressed if digital twins are to realize their potential in industrial operations. This paper presents these challenges and proposes a program of research, standardization and best practice that addresses them.

We propose the idea of a semantic backbone that supports the integration of several different types of digital twin application around a shared understanding of a facilities design. To be successful, this backbone will need to find a pragmatic balance between comprehensiveness and maintainability. It will also need to build on standards in a way that prevents reinvention of the wheel and allows modular construction of semantic models.

A successful digital twin also requires the correlation of structured data from designs, measurements and simulations with unstructured data from logs and documents. In addition, there remain many issues to be resolved around hybrid modelling: the linkage between data science - applied statistics and empirical modelling - and physical and engineering simulations. Our conclusion is that useful digital twins require the building of an interdisciplinary collaboration between computer scientists, data scientists and their engineering and operational colleagues.

2 Materials and Methods

This paper builds on the authors' own and diverse experiences in the field of integration of complex technical systems and online simulation. Each author brings a different perspective to the problem. One of us is a computer scientist, while the other two are chemical engineers with experience from developing and implementing on-line simulation systems. The analysis is supported by a literature search performed around *Industrie 4.0* and digital twins (in English and German) using ScienceDirect, OnePetro and Web of Science, with a focus on recent papers.

Digital twins have been identified as a core research topic by the SIRIUS Centre for Research-Based Innovation. This centre held a workshop on digital twins, with participants from oil companies, system vendors and consultancies on 20th March 2018. This paper is a partial summary of discussions at this meeting.

3 What is a Digital Twin?

3.1 Defining a Digital Twin

As noted above, the digital twin term had its origin in the American aerospace industry. Thus, the Defense Acquisition University defines a digital twin to be “An integrated multi-physics, multi-scale, probabilistic simulation of an as-built system, enabled by Digital Thread that uses the best available models, sensor information, and input data to mirror and predict activities/performance over the life of its corresponding physical twin.” (DAU, 2018a). This definition references another concept, namely the *digital thread*. This is defined as “An extensible, configurable and component enterprise-level analytical framework that seamlessly expedites the controlled interplay of authoritative technical data, software, information, and knowledge in the enterprise data-information-knowledge systems, based on the Digital System Model template, to inform decision makers throughout a system's life cycle by providing the capability to access, integrate and transform disparate data into actionable information.” (DAU, 2018b). This definition then points to a specific data model, namely the Digital System Model, which is “A digital representation of a ... system, generated by all stakeholders that integrates the authoritative technical data and associated artefacts which define all aspects of the system for the specific activities throughout the system lifecycle” (DAU, 2018c).

We can work this definition into the framework shown in Figure 2. In this figure we use concepts from the oil and gas industry to identify the parts of a twin.

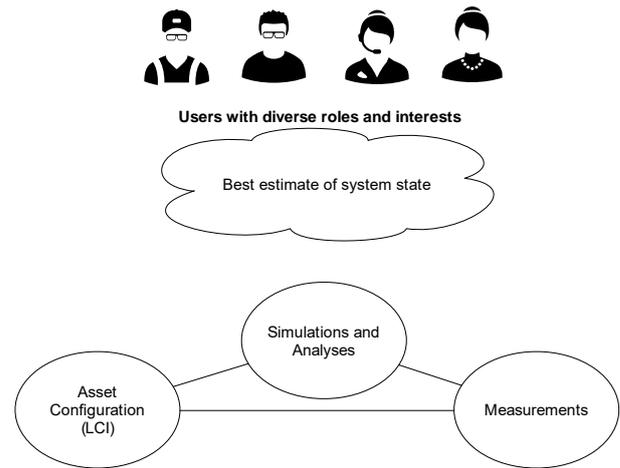


Figure 2. A conceptual framework for a digital twin, with concepts from the process industries.

Three types of data must be coordinated in a digital twin: (1) measurements of a system and its surroundings, (2) a description of the system's configuration and construction (the asset configuration) and (3) simulations and analyses of the system. We combine these three data sources to produce information about the system state that is useful for one or more users of the system.

For an oil platform or process plant, the asset configuration is contained in a Life-cycle Information (LCI) system. This allows a digital twin to exploit the virtual reality possibilities of a detailed 3D geometric model. Measurements come from the facility's control and automation system, specialized monitoring systems and the laboratory information management system. Many types of simulation and analysis are possible. We need to ask, what are we simulating? Are we interested in structural integrity? Then we need a structural simulation. Are we interested in process behaviour? Then we need a dynamic process simulator. Are we interested in the state of rotating equipment? Then we need an analysis tool that finds events in vibrational, temperature and tribology data.

3.2 Digital Twins in Oil and Gas

Explicit references to digital twins in the oil and gas industry are few and are often high-level, non-technical presentations. This may be due to the term only having come into use in the last few years, as a commercial term used to transfer technology from the “digital sector”. Poddar (2018) defines a twin as a “virtual and simulated model or a true replica of a physical asset”. What a true replica is, is left undefined. A list of design, operations and optimisation applications is given without examples and commercial integration platforms are listed. Sharma et al. (2017) give a longer but similar presentation. They describe a twin of a supply boat that can be used to optimise operations and maintenance. They also note

that a twin can support real-time barrier analysis for risk assessment.

A recent textbook on digital oil and gas technologies (Carvajal et al., 2018) does not mention digital twins. However, the concept is implicit in the on-line use and updating of an integrated asset model (p236ff) and in introducing the not-yet-realized concept of a closed loop asset model (p336). They point out challenges in implementing such a framework. We will return to this in section 4.

An early description of on-line integrated asset surveillance is given by Deaton et al. (2007). Their list of challenges is still valid: disparate data sources, non-integrated and unfiltered data. They note that data problems can be overcome, but a successful system requires embedding in the operators work processes.

Ten years later, Sankaran et al. (2017) describe Anardarko Petroleum's implementation of an integrated performance and optimization system that integrates control data and telemetry with simulation and optimization to support operational decisions. They noted that care and system design was needed to ensure that the application remained scalable and maintainable.

Digital twins for specific disciplines, such as structural and marine performance of floating platforms (Renzi et al., 2017), drilling fluids in a well (Mayani et al., 2017) and flow assurance in a gas pipeline (Lunde et al., 2013) emphasise further that the idea of the digital twin is not new. What is new is the idea of the comprehensive, interdisciplinary twin.

3.3 Digital twins in manufacturing

As noted above, the digital twin had its origin in manufacturing, in particular aerospace. It is therefore worth looking at the literature from this domain in considering how digital twins can work in the oil and gas domain. Digital twins in manufacturing focus on a *product* and its *lifecycle*, from concept generation and to recycling or disposal (Tao et al. 2017). A digital twin can make a product smart, personalized, sustainable, green, and service-oriented in an interconnected and interactive way (Tao and Zhang 2017). In oil and gas, we rather focus on the asset and its lifecycle, from concept through engineering, procurement and construction to operations and, eventually decommissioning.

The data for a product twin comes from the design and from the production process. Production data includes equipment states, materials and work-in-process. It is collected in real-time from RFID tags, sensors, actuators, controllers, lasers, vision systems, scanning systems and coordinate measuring machines. (Tao and Zhang 2017, Grieves 2014).

This product focus is supported by product life cycle management (PLM) systems. This approach was

first proposed by Grieves in 2003 (Mayani et al., 2018; Grieves 2014; Grieves and Vickers 2017).

Industrie 4.0 is a German initiative in developing and applying cyber-physical systems in manufacturing (Lu 2017, Trappey et al. 2017). The digital twin is a core component in these systems (Uhlmann, Hohwieler, and Geisert 2017, Parrott and Warshaw 2017). These authors note that two types of twins can be built: a *product* twin, that simulates the product's form and function, and a *process* twin that simulates how a product is made. These twins converge for the digital twin or an oil and gas platform.

According to Tao et al. (2017), a digital twin is a real-time reflection of a physical system in virtual space, as per the DAU definition above. It can interact and converge with historical data, interacts seamlessly with real-time data and updates its models continuously (self-evolution). In the conceptual design phase, a digital twin can give quick overview of the different product-related data and customer's feedback, enable communication between customers and designers, and thereby, help the designers to determine the product functionalities. At the detailed design phase, the digital twin enables repeated simulation tests and ensure that the desired performance is achieved for the product design scheme and the production line. At the verification stage the digital twin allows virtual verification of the different aspects of the product, and thereby cut costs for extensive physical verification and testing.

A wide variety of architectures have been proposed for digital twins (Tao and Zhang 2017, Alam and Saddik 2017, Uhlemann, Lehmann, and Steinhilper 2017, Yun, Park, and Kim 2017, Rosen et al. 2015, Parrott and Warshaw 2017). It is difficult to find a consensus architecture, but we can see that digital twin is placed in the automation hierarchy between level 3 and 4, as shown in Figure 3.

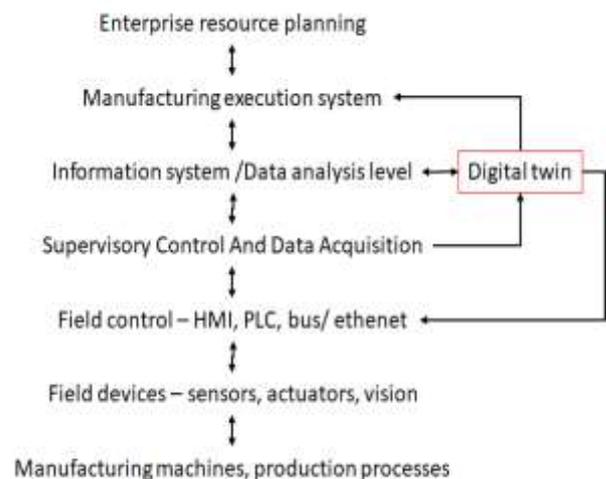


Figure 3: Digital twins in the automation hierarchy.

The digital twin acquires physical plant data through the SCADA system or the process database: what is now the Industrial Internet of Things (IIoT). This data is

compared with design data and is used to drive simulations of product and process behaviour. Results from these analyses are sent to the information and data analysis system of the manufacturing process to provide recommendations to operators and management.

Peer-reviewed papers on digital twins seem to focus on the mechanical behaviour of single products. Thus, Schroeder et al. (2016) report a very simple digital twin for a valve. The different components of the valve including sensors and actuator, were modelled using an AutomationML tool. The data was made available for the user applications via FIWARE middleware. Cai et al. (2017) presented a digital twin for a CNC machine. The graphical simulation of the machine was CAD modelled in SolidWorks software, transferred into STL format and the graphical simulation was created with OpenGL library. The manufacturing data and sensory data were stored in a PostgreSQL database. Analytical tools were used to monitor the operation conditions, extract machining features and to predict product characteristics.

The opposite extreme of a manufacturing digital twin is the F35 digital twin. West and Blackburn (2017) describe the ambitions of Lockheed and the US government to build an on-line structural and functional model of each F35 fighter and its components. West and Blackburn's analysis is deeply sceptical to the feasibility of this approach. It is simply too complex and too computationally intensive. We will return to these challenges in the next section.

So, what can we learn from the manufacturing industry? Manufacturers of the components in an oil and gas platform – instruments, valves, pumps, compressors, transformers – will offer twins of these components that will support capture of operation data for improving designs and optimising maintenance. These individual product twins will need to be incorporated into the oil company's facility or process twin. The facility twin, on the other hand is more than the sum of its components. It will need to simulate the processing of oil and gas, the electrical, mechanical and structural behaviour of the facility. This integration of diverse and complex components into a complex system is difficult.

4 Challenges in digital twins

We noted above West and Blackburn's justified but slightly polemic scepticism to the F35 digital twin. We share some of their fears. A digital twin of an oil and gas facility is a complex system. It requires the integration of a patchwork of different solutions from multiple vendors.

In this section we will describe eight challenges that need to be addressed to allow digital twins to fulfil their promised potential.

1. Business Models, Security and Confidentiality.

The first challenge is to find a business model that is sustainable. Who benefits from a digital twin? We believe that all participants in the supply chain *can* benefit from digital twins. However, this requires clarity in the sharing of data and new models for procurement of facilities and services. An operator has a strong incentive to have a comprehensive twin for monitoring and optimization. It is linked to their license to operate. They are responsible for the asset and its data. Vendors can benefit from using operational data in their product twins. This sharing and collaboration for simulation and analytics is a good thing. However, mechanisms must be found for fair rewards, allocating responsibility, ensuring secure access and protecting IPR.

2. Work practices. The digital twin can change work practices, but only if it offers the users tangible and measurable benefits. The user must be convinced that the system gives benefits, is safe and is usable. The twin must help, not hinder. Carvajal et al. (2018, p236) also pointed to this challenge: the digital twin must support the engineering and management organization.

3. Scope. A digital twin that aims to do everything is likely to do nothing well. We need to use fit for purpose modelling, and the principle of parsimony (i.e. just enough for functionality, but no more). However, we cannot do this at the cost of capturing necessary interactions and interdisciplinary effects. We need to support models with different granularity, that allow a user zoom in or out on the digital twin as needed. We also need different, but consistent models for processes with widely varying time constants, e.g. compressors (milliseconds) compared with a multiphase pipeline (days) or a reservoir (weeks).

4. Usability. How do I, as a user, get the information I need without being distracted by the information I do not need? How can I do this quickly and easily? A comprehensive digital twin has multiple concerns and can produce massive amounts of data. A specialized user must be able to find the information she needs quickly, easily and without distraction.

5. Integration. A digital twin will consist of many data sources and simulation models. How can we connect different models together without creating a "point-to-point nightmare"? Many vendors are offering platforms for data integration (Poddar, 2018). Commercial platforms are offered by vendors as diverse as GE, SAP, Siemens, Kongsberg, DNV GL, Schlumberger and Cognite (just to focus on the Norwegian market). An operating company will have to cope with several vendor platforms, both proprietary and open access, in addition to their own, internal platform. Carvajal et al. (ibid.) note that data is stored in "disparate locations, are in structured and unstructured formats, are not linked to a common data base, span long production and acquisition times, and have not been adequately and consistently quality checked".

6. Maintenance. A digital twin is a complicated software and hardware system. How can the maintenance of the necessary data, software and hardware be cost-effective? As noted above, a twin combines life-cycle information, measurements of the asset state and simulations. All these must be maintained so that they reflect the as-built state of the facility. They must be included in the facility's management of change.

7. Computational overload, edge and cloud. A comprehensive digital twin will require extensive computational resources. These resources will likely be distributed over a hybrid cloud that combines private clouds with vendor platforms and HPC resources. This system needs good design and operation to work. This design will need to carefully define what should be done at the edge of the system and what is done in the cloud.

8. Uncertainty, Validation and Data Science. Finally, a digital twin is only as good as the data and models used in the system. Data must be cleaned and reconciled. Models must be validated and tuned to ensure that they follow the state of the facility. Even "first principles" models must be tuned. Setting up effective methods for tackling uncertainty and model alignment should be a fruitful area for collaboration between control and process engineers and the data science community.

5 A research program for digital twins

How can digital twins be made sustainable, maintainable and useful despite these challenges? We believe that a solution will be collaboration between computer scientists, control and simulation engineers, data scientists and end-user technical specialists. Here we will present the computer science and data science parts of solution. This research program combines the sub-disciplines of knowledge representation, natural language technologies, formal methods, scalable computing and data science. This knowledge of technologies must be informed by the deep domain knowledge that is embedded in the digital twins' simulation models and is owned by the facility's engineers – chemical, petroleum, mechanical, electrical and control – and managers.

Challenges related to scope, usability, integration and maintenance can be addressed if we adopt a *semantic backbone* that supports the integration of several different types of digital twin application around a shared understanding of a facility's design. To be successful, this backbone will need to find a pragmatic balance between comprehensiveness and maintainability. It will also need to build on standards in a way that prevents reinvention of the wheel and allows modular construction of semantic models.

Recent advances in the construction of ontologies using templates (Forsell, 2017) promise to allow this.

Simulation providers can use these ontologies to allow exchange of model configurations with engineering databases and the exchange of calculated results with the monitoring and optimization layers of the digital twins. Point-to-point connections through tag cross-reference lists can be replaced with declarative mappings – where data in the simulation results or configuration is mapped to items in the semantic backbone.

Pragmatic standards are needed. Unfortunately, semantic standards such as ISO15926 have not lived up to their promise as tools for integration and data exchange. This is partly due to a lack of tool support and to a choice of technology, Express, that became dated and hampered progress. We also believe that the standard development attempted to be comprehensive at the expense of solving smaller, realistic problems along the way. Semantic models used by our group aim to take the best from ISO15926 where it is possible. We also will take account of the industry initiatives in DEXPI (www.dexpi.org) and CFIHOS (<http://uspi-global.org/>).

Good semantic models can also address the usability problem. Mapping data to concepts that are used by the end-user allows automatic generation of graphical interfaces that meet a specific user's needs. The OptiqueVQS (Soylu et al., 2018) framework is just such a tool. Our research group is working further with this approach to implement faceted search (Klungre & Giese, 2017).

A successful digital twin also requires the correlation of structured data from designs, measurements and simulations with unstructured data from logs and documents. This requires the structured data to be interpreted and transformed into structured data in the process model. There are now many commercial and academic tools for parsing and processing text. The IBM Watson framework is one such commercial offering (Chen et al. 2016) and can be used to parse and process texts. However, language algorithms that are trained on general data sets do not perform well when confronted with oil and gas terminology. A process of domain adaption is needed to improve performance. This process is made difficult by the smaller corpuses of data that are available for training. The SIRIUS centre is working on this domain adaption challenge, with promising results and good performance in solving standard challenge problems (Nooralahzadeh, Øvrelid and Lønning, 2018).

The challenges of maintenance and computational overload can also be addressed through using formal methods to design and monitor the *deployment* of a digital twin. As we noted above, a digital twin is a collection of interacting computational components, deployed across one or more cloud platforms and including edge devices. The behaviour of this system is

difficult to predict, especially at design time. However, simulation tools for the computer systems themselves can be used to test different deployment plans and resolve challenges. The same model can also be used as a monitoring tool for the deployed system – a digital twin of the digital twin (Johnsen, Lin & Yu, 2016).

Finally, there remains the challenge of uncertainty, validation and data science. The digital twin is built on models. To quote George Box (Box, 1979), all these models are wrong, but some are useful. A digital twin will contain many models. Some will be based on physical principles: structural, geometrical and process simulations. Others will be purely empirical, based on machine learning. These models must be validated against observed facility behaviour and aligned so that they mirror observed normal behaviour. Aligning models to observed data is difficult and remains an art. Finding out whether a discrepancy is due to an error in data, a wrong parameter, poor model structure or an actual malfunction in the facility requires a good understanding of the facility and well-developed judgment. This is true whether the models a rigorous physical model or a machine-learnt empirical representation.



Figure 4. A research agenda for digital twins in the oil and gas industry.

A maintainable digital twin will contain structured tools that allow validation and tuning of all the models in the system. We believe that hybrid analytics – the combination of data science with physical and engineering simulations – is a valuable and fruitful area of research. Machine learning can benefit from being constrained by the laws of physics, while the laws of physics contain parameters that are uncertain or expensive to measure. Good statistical practice is needed in the engineering communities and engineering knowledge is needed among data scientists.

We are working on two problems related to data science for digital twins. The first of these is related to data access. Data science projects in industry are currently not scalable. Each new implementation needs to start from scratch, finding data, checking it and making it available. Our proposed semantic backbone will allow data science solutions to be

rapidly transferred to similar sites in an organization (Kharmalov et al., 2017).

The second area of research is related to the use of sensor data in data science. When a data scientist talks about streaming data, they usually mean a sequence of discrete event records – like tweets or sales transactions. The stream of data from analogue sensors is subtly different. The underlying signal from a sensor is continuous. The process of digitization itself introduces uncertainty and error in the calculation. Filtering and data compression provide further sources for error. Common data science frameworks expect data in vectors at common times. Production of this from a time series data base requires interpolation. All these details increase the cost and decrease the usefulness of data science work. A well-defined semantics and query tool for time-series data from sensors could solve many of these challenges.

Companies in the oil and gas sector are installing digital twins now, using commercially available platforms and siloed applications. This provides academics with an opportunity to engage with the observed problems of our colleagues in operations and maintenance. This means that we need a research program that engages with operations and today’s digital twins through pilots.

Each pilot has a narrow enough focus to be tractable. The companies we are collaborating with have linked these installations to a well-defined business case. Current pilots are ambitious: if successful they will bring previously unachieved levels of interdisciplinarity, effectiveness and access to data in design, operation and maintenance. At the same time, the pilots are focused on one specific business problem. By working with existing pilots and proposing new pilots we plan to establish a virtuous cycle, where shortcomings in today’s technology and methods can be filled with research-driven innovations.

6 Conclusion

This paper has attempted to give a synthetic overview of an important, if hyped, element of today’s digitalization landscape. The digital twin is, in many ways a rebranding of several generations of on-line systems for simulation and decision support.

Marketing and vendor communication is raising expectations about digital twins. We welcome this interest and believe that it gives an opportunity to integrate operations technology and information technology. However, we also believe that implementing digital twins is inherently difficult. Solving the challenges that we have identified will require the best efforts of many computer scientists, data scientists, engineers and managers. The role and importance of simulation can only be expected to grow in the next decade. Our research agenda can help ensure that we do not disappoint our managers.

Acknowledgements

The SIRIUS centre is funded by Norwegian Research Council, project number 237898. It is co-funded by partner companies: Statoil, Schlumberger, DNV GL, IBM, SAP, Computas, OSIsoft, Numascale, Dolphin Interconnect Solutions, Kadme, Evry and Fluid Operations AG.

We thank the participants from SIRIUS partner companies and others at the workshop on digital twins held in Oslo on 20th March 2018.

References

- Alam, K.M. & A. El Saddik. 2017. "C2PS: A Digital Twin Architecture Reference Model for the Cloud-Based Cyber-Physical Systems." *IEEE Access* 5:2050-2062. <https://doi.org/10.1109/ACCESS.2017.2657006>.
- Box, G. E. P. 1979, "Robustness in the strategy of scientific model building", in Launer, R. L.; Wilkinson, G. N., *Robustness in Statistics*, Academic Press, pp. 201–236.
- Cai, Yi, B. Starly, P. Cohen, & Yuan-Shin Lee. 2017. "Sensor Data and Information Fusion to Construct Digital-twins Virtual Machine Tools for Cyber-physical Manufacturing." *Procedia Manufacturing* <https://doi.org/10.1016/j.promfg.2017.07.094>.
- Carvajal, G., Maucec, M., & Cullick, S. (2018). *Intelligent digital oil and gas fields: concepts, collaboration, and right-time decisions*. Cambridge, MA: Gulf Professional Publishing.
- Chen, Y., Elenee Argentinis, J., & Weber, G. 2016. "IBM Watson: How Cognitive Computing Can Be Applied to Big Data Challenges in Life Sciences Research." *Clinical Therapeutics*, 38(4), 688–701. <https://doi.org/10.1016/j.clinthera.2015.12.001>
- DAU. 2018a "DAU Glossary: Digital Twin," <https://www.dau.mil/glossary/pages/3386.aspx>, accessed 18th March 2018.
- DAU. 2018b "DAU Glossary: Digital Thread," <https://www.dau.mil/glossary/pages/3385.aspx>, accessed 18th March 2018.
- DAU. 2018c "DAU Glossary: Digital System Model," <https://www.dau.mil/glossary/pages/3384.aspx>.
- Deaton, D. F., & Kloosterman, J. T., 2007. "Success Stories in Onshore Production Surveillance and Optimization." *SPE Annual Technical Conference and Exhibition*, 9-11 October, San Antonio, Texas, USA. <https://doi.org/10.2118/110574-MS>
- Grievies, M., & J. Vickers. 2017. "Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems (Excerpt)." In *Trans-Disciplinary Perspectives on System Complexity*, edited by F.-J. Kahlen, S. Flumerfelt and A. Alves, 85-114. Springer.
- Grievies, M.W. 2014. *Digital Twin: Manufacturing Excellence through Virtual Factory Replication*. LLC.
- Hyllseth, M., & Cameron, D. (2003). Operator training and operator support using multiphase pipeline models and dynamic process simulation: Sub-sea production and on-shore processing. In A. Kraslawski & I. Turunen (Eds.), *Computer Aided Chemical Engineering* (Vol. 14, pp. 425–430). Elsevier. [https://doi.org/10.1016/S1570-7946\(03\)80152-9](https://doi.org/10.1016/S1570-7946(03)80152-9)
- Johnsen, E. B., Lin, J.-C., & Yu, I. C. 2016. "Comparing AWS Deployments Using Model-Based Predictions". In *Leveraging Applications of Formal Methods, Verification and Validation: Discussion, Dissemination, Applications* (pp. 482–496). Springer. https://doi.org/10.1007/978-3-319-47169-3_39
- Kharlamov, E., Mailis, T., Mehdi, G., Neuenstadt, C., Özçep, Ö., Roshchin, M., ... Waaler, A., 2017, "Semantic access to streaming and static data at Siemens.", *Web Semantics: Science, Services and Agents on the World Wide Web*, 44, 54–74. <https://doi.org/10.1016/j.websem.2017.02.001>
- Klostermeier, R., Haag, S. & Benlian, A. *Digitale Zwillinge – Eine explorative Fallstudie zur Untersuchung von Geschäftsmodellen* HMD, (2018), 55: 297. <https://doi.org/10.1365/s40702-018-0406-x>
- Klungre, V.N., Giese, M. 2017 *A Faceted Search Index for OptiqueVQS*. *CEUR Workshop Proceedings*.
- Lee, Jay, Hung-An Kao, and Shanhu Yang. 2014. "Service Innovation and Smart Analytics for Industry 4.0 and Big Data Environment." *Procedia CIRP* 16:3-8. <https://doi.org/10.1016/j.procir.2014.02.001>.
- Lu, Yang. 2017. "Industry 4.0: A survey on technologies, applications and open research issues." *Journal of Industrial Information Integration* 6:1-10. doi: <https://doi.org/10.1016/j.jii.2017.04.005>.
- Lunde, G. G., Rudrum, G., Angelo, P., Holmas, K., & Setyadi, G. R. 2013. "Ormen Lange Flow Assurance System (FAS) - Online Flow Assurance Monitoring and Advice". *Offshore Technology Conference*. <https://doi.org/10.4043/24297-MS>
- Mayani, M. G., Svendsen, M., & Oedegaard, S. I. (2018). *Drilling Digital Twin Success Stories the Last 10 Years*. Presented at the SPE Norway One Day Seminar, SPE: Society of Petroleum Engineers. <https://doi.org/10.2118/191336-MS>
- Nooralahzadeh, F., Øvrelid, L., & Lønning, J. T., 2018. "SIRIUS-LTG-UiO at SemEval-2018 Task 7: Convolutional Neural Networks with Shortest Dependency Paths for Semantic Relation Extraction and Classification in Scientific Papers." *ArXiv:1804.08887 [Cs]*. Retrieved from <http://arxiv.org/abs/1804.08887>
- Panetta, Kasey 2017, *Gartner Top 10 Strategic Technology Trends for 2018*, <https://www.gartner.com/smarterwithgartner/gartner-top-10-strategic-technology-trends-for-2018/>, accessed 23rd April 2018.

- Panetta, K., 2016, Gartner Top 10 Strategic Technology Trends for 2017., <https://www.gartner.com/smarterwithgartner/gartners-top-10-technology-trends-2017/> , accessed 23rd April 2018.
- Parrott, A., & L. Warshaw. 2017. "Industry 4.0 and the digital twin - Manufacturing meets its match." Deloitte University Press.
- Poddar, T. 2018. "Digital Twin Bridging Intelligence Among Man, Machine and Environment". Presented at the Offshore Technology Conference Asia, OTC: Offshore Technology Conference. <https://doi.org/10.4043/28480-MS>
- Renzi, D., Maniar, D., McNeill, S., & Del Vecchio, C. (2017). Developing a Digital Twin for Floating Production Systems Integrity Management. Presented at the OTC Brasil, OTC: Offshore Technology Conference. <https://doi.org/10.4043/28012-MS>
- Rosen, R., G. von Wichert, G. Lo, & K.D. Bettenhausen. 2015. "About the Importance of Autonomy and Digital Twins for the Future of Manufacturing." IFAC-PapersOnLine 48 (3):567-572. doi: <https://doi.org/10.1016/j.ifacol.2015.06.141> .
- Sankaran, S., Wright, D., Gamblin, H., & Kumar, D. (2017). "Creating Value by Implementing an Integrated Production Surveillance and Optimization System – An Operator’s Perspective." Presented at the SPE Annual Technical Conference and Exhibition, Society of Petroleum Engineers. <https://doi.org/10.2118/187222-MS>
- Schroeder, G.N., C. Steinmetz, C.E. Pereira, and D.B. Espindola. 2016. "Digital Twin Data Modeling with AutomationML and a Communication Methodology for Data Exchange." IFAC-PapersOnLine 49 (30):12-17. <https://doi.org/10.1016/j.ifacol.2016.11.115>.
- Sharma, P., Hamedifar, H., Brown, A., & Green, R., 2017. The Dawn of the New Age of the Industrial Internet and How it can Radically Transform the Offshore Oil and Gas Industry. Presented at the Offshore Technology Conference, OTC: Offshore Technology Conference. <https://doi.org/10.4043/27638-MS>
- Soylu, A., Kharlamov, E., Zheleznyakov, D., Jimenez-Ruiz, E., Giese, M., Skjæveland, M. G., ... Horrocks, I., 2018. "OptiqueVQS: A visual query system over ontologies for industry". Semantic Web, Preprint, 1–34. <https://doi.org/10.3233/SW-180293>
- Tao, Fei, Jiangfeng Cheng, Qinglin Qi, Meng Zhang, He Zhang, & Fangyuan Sui. 2017. "Digital twin-driven product desing, manufacturing and service with big data." International Journal of Advanced Manufacturing Technology 94 (9-12):3563–3576. <https://doi.org/10.1007/s00170-017-0233-1>
- Tao, F., and M. Zhang. 2017. "Digital Twin Shop-Floor: A New Shop-Floor Paradigm Towards Smart Manufacturing." IEEE Access 5:20418-20427. <https://doi.org/10.1109/ACCESS.2017.2756069> .
- Trappey, Amy J. C., Charles V. Trappey, Usharani Hareesh Govindarajan, Allen C. Chuang, and John J. Sun. 2017. "A review of essential standards and patent landscapes for the Internet of Things: A key enabler for Industry4.0." Advanced Engineering Informatics 33:208-229. doi: <https://doi.org/10.1016/j.aei.2016.11.007>.
- Uhlemann, Thomas H. J., Christian Lehmann, and Rolf Steinhilper. 2017. "The Digital Twin: Realizing the Cyber-Physical Production System for Industry 4.0." Procedia CIRP 61:335-340. doi: <https://doi.org/10.1016/j.procir.2016.11.152> .
- Uhlmann, Eckart, Eckhard Hohwieler, and Claudio Geisert. 2017. "Intelligent production systems in the era of industrie 4.0 - changing mindsets and business models." Journal of Machine Engineering 17 (2):5-24.
- West, T.D., and M. Blackburn. 2017. "Is Digital Thread/Digital Twin Affordable? A Systemic Assessment of the Cost of DoD’s Latest Manhattan Project." Procedia Computer Science 114:47-56. doi: <https://doi.org/10.1016/j.procs.2017.09.003> .
- Yun, S., J. H. Park, and W. T. Kim. 2017. "Data-centric middleware based digital twin platform for dependable cyber-physical systems." 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN), 4-7 July 2017.

CFD study on the effect of Archimedes number and heating rate on the thermal stratification of a ventilated office

Mehrdad Rabani¹ Habtamu B. Madessa¹ Natasa Nord²

¹ Department of Civil Engineering and Energy Technology, OsloMet – Oslo Metropolitan University, Norway,
{Mehrdad.Rabani, Habtamu-Bayera.Madessa}@oslomet.no

² Department of Energy and Process Engineering, Norwegian University of Science and Technology, Norway,
natasa.nord@ntnu.no

Abstract

This paper dealt with simulating a typical occupied office equipped with displacement ventilation using CFD method. The STAR-CCM+ commercial software was employed for performing the simulations in order to analyze the trend of the indoor air temperature profile in the office occupied space. Understanding this trend is a key parameter to ensure that the occupants are comfortable and thereby further conclusions on energy efficiency could be extracted. The simulations were carried out for incompressible, turbulent, and constant property air flow in the steady state condition. The results showed that increasing the number of occupants, while Archimedes number was constant, would increase the throw length of the incoming cold jet leading to an overall lower temperature profile for the case with five occupants.

Keywords: Ventilation, CFD, Archimedes number, Temperature stratification

1 Introduction

Ventilation system effectiveness plays a prominent role in the improvement of indoor air quality (IAQ) of buildings (Sundell, 1988). In modern societies, people spend a lot of their time (on average more than 90%) in indoor environments (Bruinen de Bruin, 2008), therefore the need for good IAQ is considered as imperative. In addition, ventilation stands for 20% to 40% of total building energy use in modern countries (Pérez-Lombard, 2008). This can undeniably justify the importance of special care for mechanical ventilation system, especially in countries characterized with cold climate such as Norway, where using natural ventilation is seldom possible.

There are two common methods implemented in ventilation systems for replacing the used air with fresh air: mixing and displacement ventilation. The mixing ventilation has the capability to remove considerable amounts of excess heat without risk of draft and the air

temperature is almost well mixed (Hestnes and Eik-Nes, 2017).

In displacement ventilation, the system delivers fresh air directly to the occupied zone leading to an efficient removal of pollutants and excess heat. However, this results in a thermal stratification of air in the occupant zone that in turn may cause discomfort if the gradient exceeds a certain level (Nielsen, 2000). Therefore, understanding the air temperature profiles and the influencing parameters are crucial to assess the comfort conditions in displacement ventilation systems.

Modelling of temperature and velocity flow fields is challenging in ventilation systems. It requires a knowledge of both 1) impact of each room component on the room physics, and 2) the interaction of all of these effects simultaneously. Nevertheless, Computational Fluid Dynamics (CFD) is a well-established method and has proven its competence in simulating the ventilation system performance (Nielsen, 2009; Menchaca-Brandan et al, 2017).

As a common practice in most of CFD studies, a set of dimensionless parameters are defined to better investigate the effect of a parameter on the flow field and thermal stratifications. Using the dimensionless parameters will make the results functional for all the parameter combinations as long as they do not influence the physics of the flow in the room, e.g. they do not change the turbulence regime to laminar and vice versa etc. (Sonin, 2001).

The dimensionless parameters that are often considered for studying the flow field characteristics, e.g. vertical temperature gradient and velocity distribution are the Archimedes, Reynolds, and Prandtl numbers. Previous numerical and experimental studies demonstrated that the effect of Reynolds and Prandtl numbers are trivial on the temperature variation especially in displacement ventilation (Nielsen, 1988). However, the Archimedes number has shown a great potential to affect the thermal stratification considerably (Nielsen, 1988; Corgnati et al, 2009). The Archimedes number is defined as Eq. (1).

$$Ar = \frac{g\beta\Delta Th}{u^2} \quad (1)$$

where the g is the gravity acceleration, β is the air volume expansion coefficient, u is the supply velocity, ΔT is the temperature difference between outlet and inlet, and h is the height where the supply terminal is placed (Nielsen, 1988). In order to better predict the thermal stratification (Menchaca-Brandan, 2012) considered the following model:

$$Ar = \frac{g\beta\Delta Th^3}{u^2 A_s} \quad (2)$$

where A_s is the area of supply terminal.

This paper examined the effect of the Archimedes number on the air temperature stratification and velocity distribution in a typical occupied office equipped with displacement ventilation using CFD. The most important contribution of this study was to indicate how CFD could be used to investigate the combined effect of different heating rate due to occupancy and Archimedes numbers (for high buoyancy and high supply momentum) on the air temperature stratification and velocity distribution. The aim was to test the capability of the simulation software in capturing the phenomena happen in extreme condition.

2 Methodology and case study

2.1 Geometry specifications

In order to study the effect of different parameters on the temperature stratification, the office building model proposed by (Menchaca-Brandan, 2012) was used. The building is an open plan office. Due to symmetry, only one repeating part of the open plan office was considered for simulations in order to reduce the computational cost. The office was ventilated using the displacement method.

Figure 1 shows the geometry and dimensions of the model used for CFD simulations. The room has two inlets and one outlet. The outlet area was fixed (2 m width and 0.2 m height) and its upper edge was placed 0.1 m from the ceiling. The inlets were placed 1 m from the floor with constant total area of 0.06 m².

The heating source was a seated occupant modelled as a cylinder with round-top shape having 1.13 m height and 0.62 m diameter and located centrally in the office (Menchaca-Brandan and Glicksman, 2011). Three and five (Figure 1) occupants were considered for simulations. In both cases, the distance of the first occupant was 2 m from the inlet wall and the distance between each one is 2 m and 4 m in the case of five and three occupants, respectively.

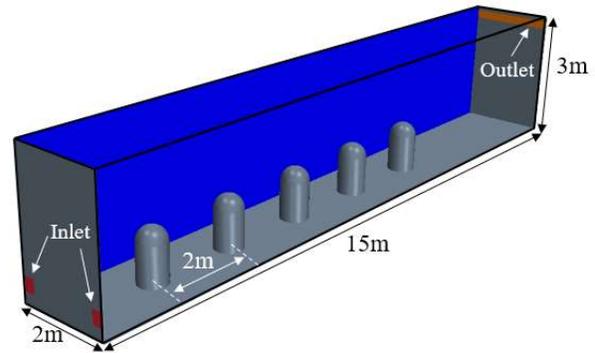


Figure 1. Dimensions and geometry of case study

2.2 CFD model

The commercial code STAR-CCM+ (Version 12.04.010) was used for the numerical solution of the RANS (Reynolds-Averaged Navier-Stokes) and energy equations. The software has In-Situ Adaptive integral Tabulations and a number of other methods to speed up computation. The whole CFD workflow including CAD, meshing, solving, optimization, and post-processing can be automated in one macro. The software solves the RANS equations in their integral form, by means of Finite Volume (FV) method. The simulation cases were run in steady state condition. The segregated solver was chosen that uses the SIMPLE pressure-velocity coupling algorithm where the mass conservation constraint on the velocity field is fulfilled by solving a pressure-correction equation. It should be noted that the segregated solver has its roots in constant-density flows (STAR-CCM+ User Guide, 2017).

In order to check if the steady state condition was reached, the mass flow rate difference between outlet and inlet as well as the outlet temperature were monitored during the simulations, which the first one should be approximately zero and the latter should be nearly equal to the outlet temperature calculated from energy conservation equation.

2.2.1 Boundary conditions

Figure 2. indicates the boundary conditions of the CFD model implemented in STAR-CCM+. At inlets, a constant air velocity boundary condition was defined. The inlet temperature was 19°C. The outlet was modelled as pressure outlet with zero gauge pressure. Due to symmetrical conditions, the side wall surfaces of the office were modelled with symmetry boundary condition as shown in Figure 2. Ceiling was considered as the surface with constant heat flux of 8 W/m² (around 25% of total heat flux in a typical office) representing heat gain due to lighting (Menchaca-Brandan, 2012). Other walls and surfaces were considered adiabatic. This study focused on the ventilation during working hours. The heat gain due to occupant was modelled as a heating source emitting a fixed amount of 120 W, which

is associated with a seated person with metabolic rate 1 Met (Fanger, 1972).

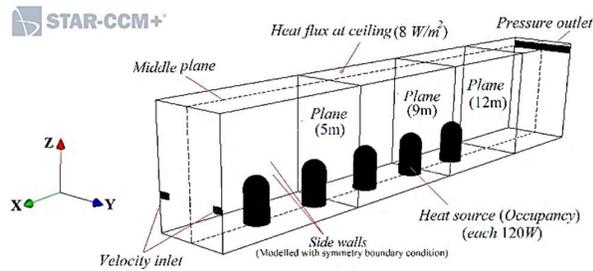


Figure 2. Boundary conditions and measured planes of CFD models in STAR-CCM+

2.2.2 Simulation specification

The flow regime was considered fully turbulent. Therefore, the Shear Stress Transport k - ω (SST k - ω) turbulence model was adopted for simulations due to its decent performance regarding supply jet behavior, prediction of buoyancy turbulence interaction, and temperature stratification in ventilated buildings (Gilani et al, 2016). The computational domain discretization was obtained by means of a polyhedral-shaped mesh (Figure 3a). It is numerically more stable, less diffusive,

and more accurate than an equivalent tetrahedral mesh (STAR-CCM+ User Guide, 2017). In order to perform a grid independence study, four meshes with 2033458, 1203867, 526027, and 119742 cells were investigated. The mesh with 1203866 cells was chosen, giving lesser computational time, because the maximum discrepancy between the temperature value for this mesh and the largest mesh with 2033458 cells was less than 2.5%, as shown in Figure 3b. An inflation layer around the surface was also used to capture high gradients generated within a small region near the walls and surfaces accurately with a $y^+ \leq 5$. Figure 3c illustrates the frequency of y^+ near the heating source surfaces. Maximum frequency of y^+ occurs at $y^+ \sim 0.023$, which is much less than 5 indicating the mesh properly resolved the viscosity-affected region near the wall and recaptured the heat transfer from surfaces to the flow in the human thermal plume. The radiation was modelled using surface to surface (S2S) model behaving the air as transparent to radiation. The emissivity of all walls was defined to be 1 owing to typical materials used in an office have usually an emissivity approximately 1 (Menchaca-Brandan, 2012).

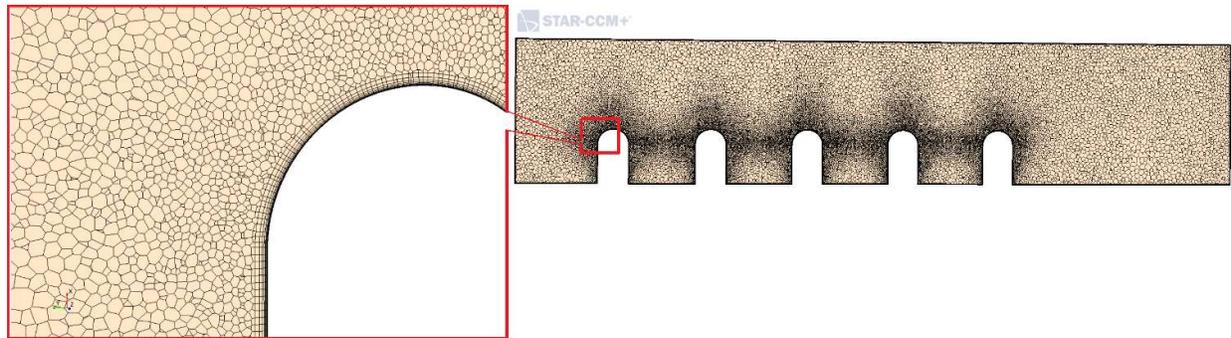


Figure 3a. Mesh generation used in this study with inflation layer around surface

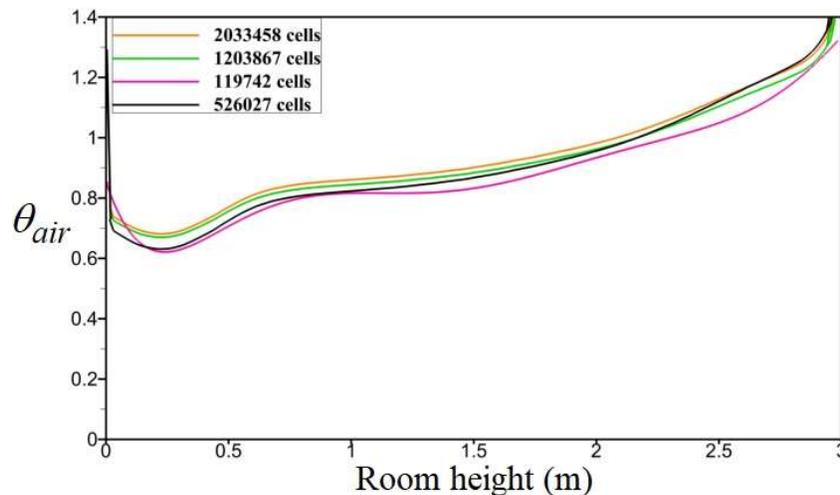


Figure 3b. Vertical temperature profile for different mesh cell numbers

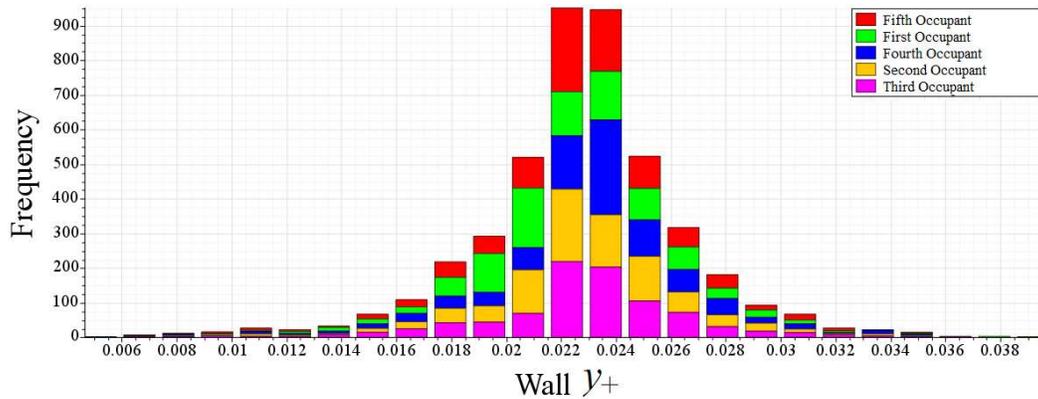


Figure 3c. Wall y^+ frequency on the thermal heating walls

2.3 Validation and verification

The method was validated by previous experimental and numerical studies. The first simulation was performed in order to replicate the experimental results of an air flow in a rectangular room in the presence of radiation (Figure 4a) (Olson et al. 1990).

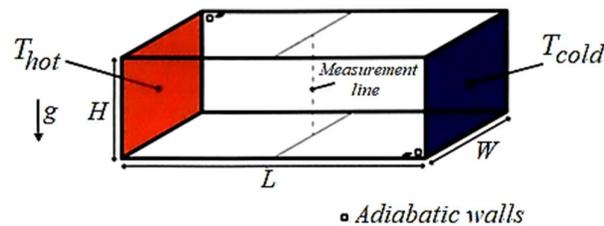


Figure 4a. Configuration of rectangular enclosure. The rectangle in dotted lines corresponds to the location where measurements were performed (Olson et al, 1990).

The experimental setup used for validation purpose has been shown in Table 1. The Rayleigh number, which is a dimensionless number indicating whether the natural convection boundary layer is laminar or turbulent, was defined as follows:

$$Ra_L = \frac{g\beta(T_{hot} - T_{cold})L^3}{\nu\alpha} \quad (3)$$

Where L , ν , α , and β are the length of the room, kinematic viscosity, thermal diffusivity, and thermal expansion coefficient of the air, respectively. According to the calculated Rayleigh number in Table 1, the flow was considered turbulent.

Table 1. Experimental setup used by (Olson et al, 1990)

Room Length (L)	7.9 (m)
Room Width (W)	3.9 (m)
Room Height (H)	2.5 (m)
T_{hot} (Temperature of cold wall)	36.6 ($^{\circ}\text{C}$)
T_{cold} (Temperature of hot wall)	19.9 ($^{\circ}\text{C}$)
Rayleigh number (Ra_L)	6.98×10^{11}

Figure 4b shows the comparison of present numerical simulation with previous experimental and numerical works. As it can be noted, the present numerical method fairly replicates the experimental results, even better than previous numerical study especially for the points near the ceiling. This shows that STAR-CCM+ can be used to study displacement ventilation in an office building, where both the radiative and convective heat transfers are present.

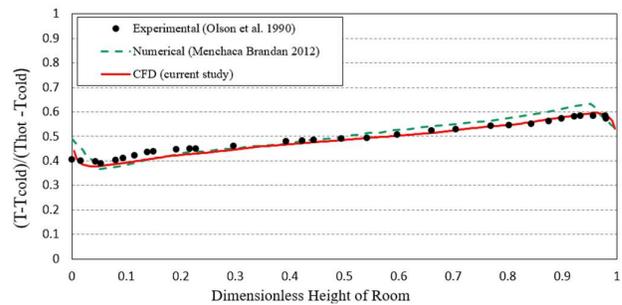


Figure 4b. Comparison of dimensionless temperature variation on the vertical line

2.4 Archimedes number analysis

Four cases were considered for simulations in terms of combination effects of two different Archimedes number and two different heating rates. Two sets of three and five number of occupants were considered. Furthermore, two Archimedes numbers of 60 and 0.5 were selected. In order to better interpret these combined effects, the Archimedes number was rewritten using energy conservation equation in the space as follows:

$$Ar = \frac{g\beta\dot{Q}h^3}{\rho c_p u^3 A_s^2} \quad (4)$$

where \dot{Q} is the total heat gains in the room. The geometric properties in Eq. 4 (h and A_s) were considered constant. Therefore, with regard to the selected Archimedes number, the inlet velocity in each simulation case was chosen so that the supply volumetric air flow rate was always kept higher than the

minimum standard air amount for Norwegian non-residential buildings (TEK17).

Temperature profiles were measured at three different measuring planes, located at 5 m, 9 m, and 10 m from the inlet wall respectively (see Figure 2). Temperature gradients were compared using the following dimensionless form:

$$\theta_{air} = \frac{T - T_{in}}{T_{out} - T_{in}} \quad (5)$$

3 Results and discussion

Figure 5 indicates the effect of different Archimedes number and different number of heating rates on the dimensionless temperature variation. As can be seen from Figure 5, at high Archimedes number (cases (c) and (d)), there almost existed a two-structured zone with lower stratified temperature profile in the occupancy zone and upper destratified temperature profile formed due to convective effect of human thermal plume. However, at the low Archimedes number (cases (a) and (b)), the higher velocity of supply jet mixes the cold supply air with thermal plume of heating rate causing the uniform temperature zone to be distorted and the temperature near the ceiling to increase. Furthermore, a higher heating rate (more occupants) would result in a lower air temperature value in the space when the Archimedes number is kept constant (compare the cases (c) and (d)).

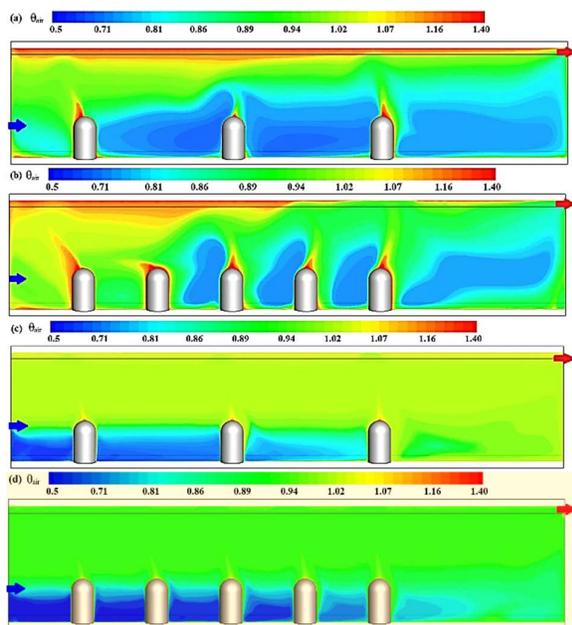


Figure 5. Temperature distribution contour for cases: (a) $Ar=0.5$, $N=3$, (b) $Ar=0.5$, $N=5$, (c) $Ar=60$, $N=3$, and (d) $Ar=60$, $N=5$. Results were given along the middle plane. Inlet and outlet have been specified with blue and red arrows, respectively.

Figure 6 illustrates the combined effect of heating rate and the Archimedes number on the velocity distribution in the office space. At the low Archimedes number, the inlet supply air moving across the space reached the outlet wall and rises to higher heights and then return to the inlet until it is entrained by the supply air flow. This phenomenon is less pronounced at the high Archimedes number, since the lower velocity of supply jet returning from the outlet wall cannot overcome the human thermal plume rise. It is worth noticing that the simulation tool could properly capture the recirculating zones created due to interaction of the human plume and the cold supply jet.

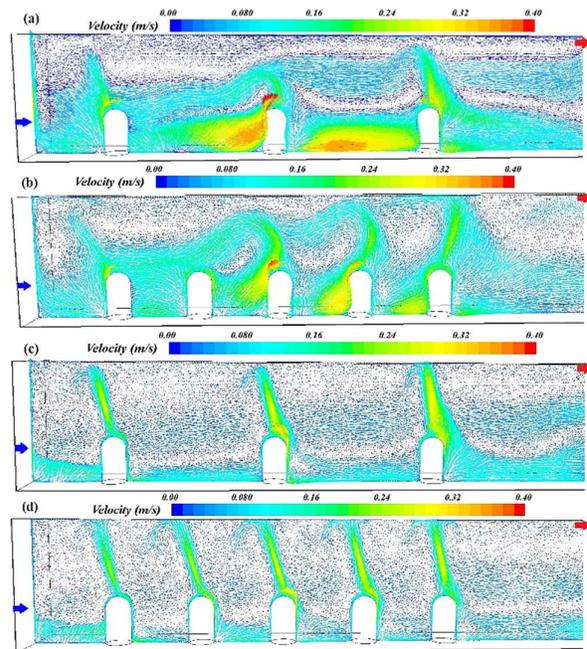


Figure 6. Air velocity distribution and vectors along the middle plane: (a) $Ar=0.5$, $N=3$, (b) $Ar=0.5$, $N=5$, (c) $Ar=60$, $N=3$, and (d) $Ar=60$, $N=5$. Inlet and outlet have been specified with blue and red arrows, respectively.

Figure 7 indicates the development of temperature profile for different Archimedes numbers and heating rates. Due to high velocity of the supply jet, the low Archimedes number shows a larger overall vertical temperature variation than the high Archimedes number. Increasing the number of occupants, while Archimedes number was constant, would increase the throw length of the incoming cold jet leading to an overall lower temperature profile for the cases with five occupants. In addition, the air temperature near the floor was higher than its neighbor region due to radiation from the ceiling.

Figure 8 indicates the vertical temperature variation in three measured planes in one specific case. Temperature near the floor was lower at $x=5$ m compared to the other cases due to the effect of the cold supply jet. Likewise, the temperature near the ceiling was higher at $x=5$ m owing to the fact that the cold

supply jet after reaching the outlet wall returned to the inlet and was mixed with the thermal plume near the ceiling and was accumulated near the inlet wall until it was entrained by the supply jet. The temperature variation, in general, in the occupancy zone was less at $x=9$ m and 12 m compared to $x=5$ m, because it was less influenced by the incoming cold jet at these planes.

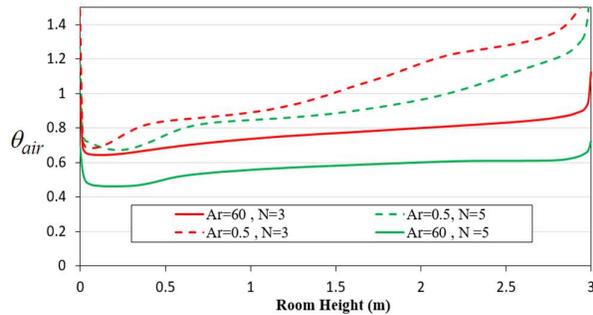


Figure 7. Dimensionless temperature profile at the plane $x=5$ m

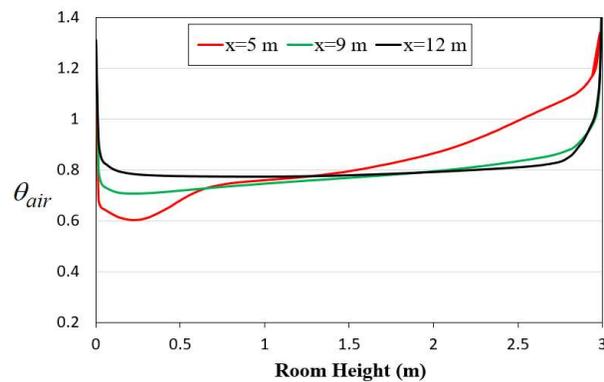


Figure 8. Dimensionless temperature profile at the planes $x=5$ m, 9m, and 12m for $Ar=0.5$, $N=5$

The effect of cold supply jet for different Archimedes numbers and heating rates is illustrated in Figure 9. At the low Archimedes number (cases (a) and (b)), the high momentum of the cold jet results in longer throw length and larger vertical temperature variation. The throw length of the cold jet in cases (a) and (b) was longer due to higher inlet velocity in these cases. However, at the high Archimedes number (cases (c) and (d)), the incoming jet fallen and reached the floor near the inlet due to strong buoyancy force of the warm air in the proximity. Consequently, a stratified layer was formed in the occupancy zone, while a convective recirculating zone was formed due to buoyancy effect of human thermal plumes in the upper part.

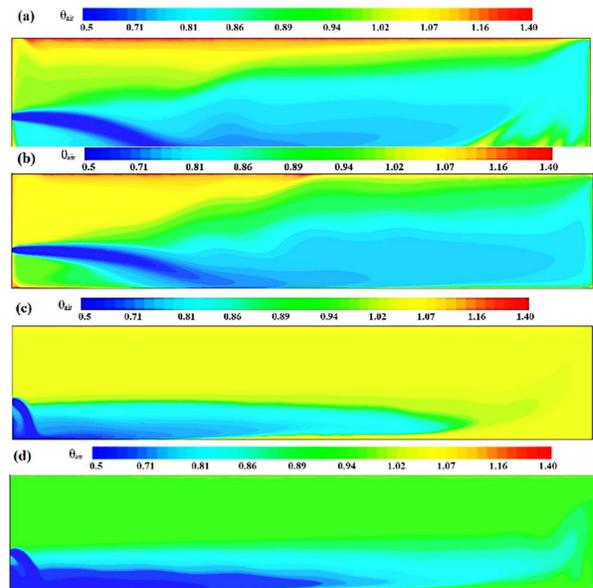


Figure 9. Temperature distribution contour for cases: (a) $Ar=0.5$, $N=3$, (b) $Ar=0.5$, $N=5$, (c) $Ar=60$, $N=3$, and (d) $Ar=60$, $N=5$. Results were given along the symmetry plane.

4 Conclusion

In this study, four cases of combinations of the Archimedes number and the heating rates were investigated by means of CFD method. Simulations were performed using STAR CCM+ software. The results indicated that the high Archimedes number resulted in the dominant phenomenon of the buoyancy of convective flow induced by the human thermal plume. Therefore, two structured areas were formed in the space: a lower zone with stratified temperature profile due to interaction of the incoming cold jet and the thermal plume and an upper recirculating zone with the destratified temperature profile due to convective effect of human thermal plume. For the low Archimedes number, high momentum of the incoming jet disrupts the flow field resulting in mixing of the cold jet with convective flow of the thermal plume in the space. Increasing the number of the heating sources decreased the overall vertical temperature. Furthermore, the results indicated that the software properly captured the convective recirculating regions created by thermal plume as well as the vertical temperature gradient in both buoyant and high momentum flows.

It is worth mentioning that the present results in this study showed the capability of STAR CCM+ for numerical study of buildings ventilated by displacement ventilation. As further study, STAR CCM+ will be used to investigate the effect of the high and low Archimedes numbers in mixing ventilation systems.

References

- Y. Bruinen de Bruin, K. Koistinen, S. Kephelopoulos, O. Geiss, S. Tirendi and D. Kotzias (2008): Characterisation of urban inhalation exposures to benzene, formaldehyde and acetaldehyde in the European Union: comparison of measured and modelled exposure data. *Environmental Science and Pollution Research*, 15(5):417–430.
- S.P. Corgnati, M. Perino, G.V. Fracastoro and P.V. Nielsen (2009): Experimental and numerical analysis of air and radiant cooling systems in offices. *Building and Environment*, 44:801–06.
- P.O. Fanger (1972): Thermal Comfort, *McGraw-Hill*, New York, USA.
- S. Gilani, H. Montazeri and B. Blocken (2016): CFD simulation of stratified indoor environment in displacement ventilation: Validation and sensitivity analysis. *Building and Environment*, 95:299–313.
- A.G. Hestnes and N.L. Eik-Nes (2017): Zero emission buildings. Bergen. Fagbokforlaget, 107–121.
- M.A. Menchaca-Brandan and L.R. Glicksman (2011): How to replicate computationally a human plume with a simple geometry, and how not to, in: *Proceedings of RoomVent, Trondheim, Norway*.
- M.A. Menchaca-Brandan (2012): Study of Airflow and Thermal Stratification in Naturally Ventilated Rooms. *PhD thesis*, Massachusetts Institute of Technology.
- M.A. Menchaca-Brandan, F.A. Dominguez Espinosa and L. R. Glicksman (2017): The influence of radiation heat transfer on the prediction of air flows in rooms under natural ventilation. *Energy and Buildings*, 138:530–38.
- P.V. Nielsen (1988): Displacement Ventilation in a Room with Low-Level Diffusers. *Aalborg: Department of Building Technology and Structural Engineering, Aalborg University. Indoor Environmental Technology*, R8836(10).
- P.V. Nielsen (2000): Velocity distribution in a room ventilated by displacement ventilation and wall-mounted air terminal devices. *Energy and Buildings*, 31:179–87.
- P.V. Nielsen (2009): CFD in Ventilation Design: a new REHVA Guide Book. *Aalborg: Department of Civil Engineering, Aalborg University*.
- D.A. Olson, L.R. Glicksman and H.M. Fern (1990): Steady-state natural convection in empty and partitioned enclosures at high Rayleigh numbers. *Journal of Heat Transfer (Transactions of the ASME (American Society of Mechanical Engineers), Series (C))*, 112(3):109–111.
- L. Pérez-Lombard, J. Ortiz and C. Pout (2008): A review on buildings energy consumption information. *Energy and Buildings*, 40:394–398.
- A.A. Sonin (2001): The Physical Basis of Dimensional Analysis. *Technical Report*, Massachusetts Institute of Technology.
- J. Sundell (2004): On the history of indoor air quality and health. *Indoor Air*, 14:51–58.
- STAR-CCM+ User Guide (2017), Version 12.04.010, CD-adapco.
- Veiledning om tekniske krav til byggverk (2017). *Byggeteknisk forskrift- TEK17*.

SIMS2018 CFD Simulation of Solidification of non-Newtonian Fluid Flowing in a Complex Geometry Pipeline in Turbulent Flow Regime

Ludmila Vesjolaja¹ Jakub M. Bujalski² Knut Vaagsaether¹

¹ Department of Process, Energy and Environmental Technology, University of South-Eastern Norway, Norway, {ludmila.vesjolaja, knut.vagsaether}@usn.no

² Process Modeling and Control Department, Yara Technology Centre, Norway, jakub.bujalski@yara.com

Abstract

In this CFD (Computational Fluid Dynamics) study, the turbulent flow of a non-Newtonian fluid through an industrial scale transportation pipeline is modelled in Ansys Fluent[®], with a focus on the fluid solidification due to heat transfer on the pipe walls. The turbulence was modelled using two different turbulence models: a standard low-Reynolds-number $k-\varepsilon$ turbulence Chan-Hsieh-Chen (CHC) model and a modified Malin's turbulence model. Simulations were performed with fluid viscosity depending both on the shear rate as well as on the temperature. However, according to the simulation results, as long as the inlet fluid velocity is maintained sufficiently high (turbulent flow), the occurrence of fluid solidification is not significantly affected by the viscosity dependence on the temperature. All turbulence models show fluid solidification on the pipe walls, and not inside the pipe itself. The standard CHC model shows more pipe wall zones that are solidified, while the modified Malin's turbulence model shows a more diffusive behavior. The latter model has an effect on the velocity distribution across the pipeline in such a way that the fluid flow between the pipelines become more evenly distributed. The simulation results of pipe insulation and liquid flow rate, on the fluid solidification were used to give recommendations of improvements to avoid blockages in the transportation pipelines in the industrial process. According to the simulation results, the use of pipe insulation can minimize the occurrence of fluid solidification on the pipe walls.

Keywords: non-Newtonian fluid, turbulent pipe flow, solidification, Malin turbulence model, CFD

1 Introduction

The solidification of fluids in pipe flows is an important topic in many practical engineering problems, especially in manufacturing industries where material phase-change may occur. The change of phase of the fluid may cause damages to the pipelines due to blockages that may eventually lead to unforeseen plant shut downs and

additional cleaning procedures. Due to heat transfer the phase-change of the fluid usually occurs first on the pipe wall where a solid phase develops and increases its radial size with time causing possible pipeline blockages (Conda et al., 2004; Wei and Güceri, 1988). The modelling of this phenomenon is very challenging since they are time-dependent and factors such as flowrate and temperature directly affect its formation.

Despite the problems caused by fluid solidification in various engineering processes, very few research works about fluid solidification have been published in literature. Early studies devoted to the solidification phenomena were performed by Hirschberg (1962) and by Zerkle and Sunderland (1969). In these works, the solidification of the fluid was studied by assuming a laminar flow regime at the steady-state (Hirschberg, 1962; Zerkle and Sunderland, 1969). Wei and Güceri (1988) conducted another significant study where an attempt was made to develop a numerical model for describing the solidification in fully developed internal pipe flows. Almost 20 years later, Conde et al. (2004) have developed a 2D numerical model for describing the solidification of water, olive oil and aluminum in cylindrical ducts. Interestingly, this is the only available study where solidification in internal pipe flows was conducted using Ansys Fluent[®]. During the past ten years only few works were devoted to the solidification in internal flows, and these mostly focus on the enhancement of phase-change in heat pipes (Motahar and Khodabandeh, 2016; Sharifi et al., 2014). To our knowledge, only Myers and Low (2013) has published on the solidification of the non-Newtonian fluid flows in pipes. They have developed a mathematical model for the solidification of the Power-Law fluids (shear-thinning) in narrow pipes and have assumed a laminar flow regime as well as using MATLAB to solve their model equations (Myers and Low 2013).

This study investigates the solidification during fluid transport around a final stage of fertilizer particulation process. One of the challenges faced in the manufacturing of complex fertilizers is the efficient transportation of the process fluids from one stage of the process operation to the next. This relatively simple

operation can be a major issue as the fluid has non-Newtonian behavior. The complex behavior of the process fluid affects the efficiency of pumps causing pipe blockages due to pre solidification in the pipes, leading to loss of production. The understanding of this rheological behavior of the non-Newtonian fluid is of fundamental importance for proper operation of the plant. Detailed knowledge of non-Newtonian fluid solidification can also be useful for designing layout of the pipelines and for proper selection of pipe insulation to reduce the risk of pipe blockages. Hence, in this paper, the authors have contributed with the modeling and numerical simulations of the solidification of a highly non-Newtonian fluid in turbulent flow regime in 3D. The focus is on the flow of an industrial case non-Newtonian fluid through complex geometry pipelines in turbulent flow regime.

2 Model development

2.1 Solidification of the fluid

In this study, “Enthalpy-Porosity” formulation is used to model fluid solidification. This approach is based on the studies by Voller and Prakash (1987) and the method is also available in Ansys Fluent® User’s guide (2006). According to this technique, the melt interface is computed implicitly. In this method, a liquid fraction that is linked with every cell in the computational domain is used to track the interface. The liquid fraction indicates the fraction of the cell that is in the liquid state. Ansys Fluent® uses the “mushy zone” which is modelled as a “pseudo-porous media” in which the porosity (or liquid fraction) ranges from one to zero. When the porosity is equal to one, the fluid is in fully liquid-state, and when it is equal to zero, the fluid is in solid-like state with zero velocity (Ansys Fluent®, 2006).

The corresponding energy equation and enthalpy formulations solved in Ansys Fluent® (2006) are represented with Equation 1 and Equations 2 to 4 respectively. The liquid fraction is calculated using Equation 5 and the solution for the temperature is found iteratively using Equations 1 to 5:

$$\frac{\partial}{\partial t}(\rho H) + \text{div}(\bar{v}\rho H) = \text{div}(k \text{ grad } T) + S_t \quad (1)$$

$$H = h + \Delta H \quad (2)$$

$$h = h_{ref} + \int_{T_{ref}}^T c_p dT \quad (3)$$

$$\Delta H = \beta L \quad (4)$$

$$\beta = \begin{cases} 0, & \text{if } T < T_{solid} \\ 1, & \text{if } T > T_{liquid} \\ \frac{(T - T_{solid})}{(T_{liquid} - T_{solid})} & \end{cases} \quad (5)$$

where ρ is density, H is total enthalpy, \bar{v} is velocity vector, k is turbulent kinetic energy, T and is absolute

temperature, S_t is shear rate component, h is sensible enthalpy, ΔH is latent heat, and c_p is specific heat.

2.2 Rheology of the fluid

The particular fluid used in this study is a shear-thinning slurry whose viscosity depends on both the shear rate and the temperature. The rheological properties of the fluid taken into consideration were obtained experimentally using Anton Paar Modular Compact Rheometer 302. The experimentally obtained flow curve was fitted to different rheological models. The

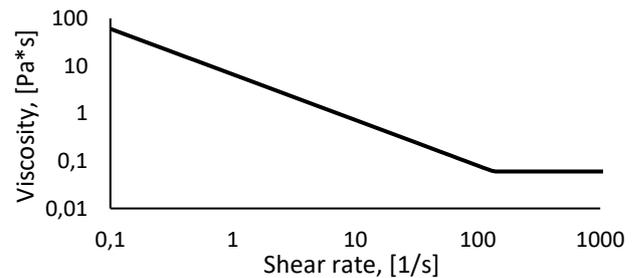


Figure 1. Fluid viscosity dependence on the applied shear rate (log-log scale).

best fit was obtained with the Power-Law model for non-Newtonian fluids. The fitted flow curve used in this study is illustrated in Figure 1.

The effect of the temperature on the viscosity of the fluid was characterized using cooling curve illustrated in Figure 2. It indicates that the temperature significantly affects the viscosity of the given fluid. The higher the temperature, the lower is the apparent viscosity of the fluid. Therefore, to obtain a better model of the given process, the changes in the viscosity due to the temperature, in combination with its dependence on the shear rate, should be included when modelling solidification.

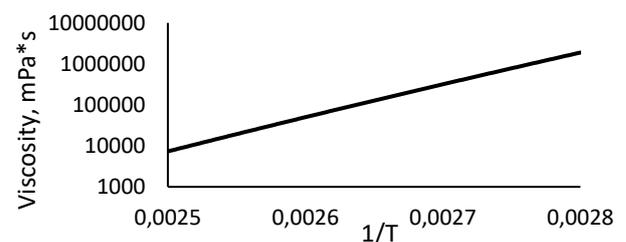


Figure 2. Fluid viscosity dependence on temperature.

It is possible to model the effect of both the temperature and the shear rate on the viscosity of the fluids in Ansys Fluent®. For this, an activation energy constant (also known as activation temperature) and a reference temperature should be known. In this study, the activation temperature was estimated using the Least Square fitting method. Least Square fitting was performed by using the modified Arrhenius law (Equation 6) showed in Rojas et al. (2008) and was fitted to the cooling curve polynomial (Figure 2) obtained

experimentally. The estimated value of the activation temperature (α) is listed in Table 2.

$$\mu = \mu_{\infty} \exp\left(\frac{E}{RT}\right) \quad (6)$$

where μ is apparent viscosity, μ_{∞} is infinite viscosity, E is energy, and R is universal gas constant.

2.3 Turbulence model

The standard k - ε turbulence model is widely used in modelling of internal turbulent pipe flows due to its simplicity and applicability. However, the standard k - ε turbulence model does not account for drag reduction effect and may have unsatisfactory results and predictions at the near wall zones where eddy viscosity changes rapidly with the distance from the pipe wall (Mathur and He, 2013; Versteeg and Malalasekera, 2007). This might have a crucial role in modelling solidification since the solidification of the fluid is expected to occur on pipe walls. One solution to overcome this problem is to use low-Reynolds-number k - ε turbulence models that are specifically developed to account for near-wall phenomena. However, the fluid taken into consideration exhibits a highly non-Newtonian behaviour. To account for this phenomena, in this paper Malin's turbulence model for Power-Law fluids has been modified (Vesjolaja, 2016).

Malin (Malin, 1997; Malin, 1998) has developed a model where the damping functions are specially treated to describe the non-Newtonian fluid flow in the turbulent flow regime (Reynolds number (Re) up to 10^5). Malin's model is based on low-Reynolds-number k - ε turbulence models like the Lam-Bremhost (LB) model (Lam and Bremhost, 1981). The only and the most important difference between these two models is in the way the eddy/turbulent viscosity is calculated. For both models, the transport equation for the kinetic energy is formulated as in the standard k - ε turbulence models and can be founded in earlier works (Malin, 1997; Lam and Bremhost, 1981). However, the eddy dissipation rate formulation that carries turbulent viscosity term differs between the two models. With the Malin's turbulence model, the formulation of eddy viscosity includes the power-law index (that carries the

non-Newtonian characteristics) in the damping function term, while this is not present in the Lam-Bremhost model.

The transport equation for the eddy dissipation rate is given by Equation 7, meanwhile the turbulent viscosity is calculated by Equation 8 using damping functions (f_1 , f_2 and f_{μ}) defined in Table 1 (Malin, 1997).

$$\frac{\partial(\rho\varepsilon)}{\partial t} + \text{div}(\rho\varepsilon U) = \text{div}\left(\frac{\mu_t}{\sigma_{\varepsilon}} \text{grad } \varepsilon\right) + f_1 C_1 \frac{\varepsilon}{k} 2\mu_t S \cdot S - f_2 C_2 \rho \frac{\varepsilon^2}{k} \quad (7)$$

$$\mu_t = C_{\mu} f_{\mu} k^2 / \varepsilon \quad (8)$$

where ε is turbulent dissipation rate, U is velocity component, μ_t is turbulent viscosity, and C_1 , C_2 and C_{μ} are turbulence adjustable constants.

Malin's turbulence model is not readily available in Ansys Fluent[®]. It is possible to implement Malin's turbulent model by using User Defined Functions (UDF), however the computational time for the convergence of the solution is prolonged significantly. The increased computational time is even more pronounced when modelling thermal effects (solidification in this case). Hence, the Malin's turbulence model was implemented in a robust way by enabling and modifying the built-in low-Reynolds-number k - ε model. For this, Malin's eddy viscosity formulation was coupled to the built-in CHC (Chang et al., 1995) turbulence model using a UDF. The CHC model was preferred to the LB model as well to other turbulence models available in Ansys Fluent[®] e.g. Launder-Sharma, Abid, Yang-Shih and the Abe-Kondoh-Nagano models. This choice is based on the fact, also supported by previous studies (Vesjolaja, 2016) that the CHC model exhibits more stable solutions during the simulations. Both the standard CHC and the modified Malin's models were compared in this study (see Equations 7, 8 and Table 1 for the algebraic equations used in different models).

Table 1. Turbulence model damping functions used in Equations 7 and 8.

Turbulence Model	f_1	f_2	f_{μ}	Wall boundary conditions
Malin original (Malin, 1997)	$1 + \left(\frac{0.05}{f_{\mu}}\right)^3$	$1 - \exp(-Re_t^2)$	$(1 - \exp(-0.0165Re_k/n^{1/4}))^2 * (1 + 20.5/Re_t)$	$\frac{\partial\varepsilon}{\partial y} = 0$
CHC original (Chang et al., 1995)	1.0	$(1 - 0.01 \exp(-Re_t^2)) * (1 - \exp(-0.0631Re_k))$	$(1 - \exp(-0.0215Re_k))^2 * (1 + 31.66/Re_t^{5/4})$	$\varepsilon_w = \nu \left(\frac{\partial^2 k}{\partial y^2}\right)$
Modified Malin's (Vesjolaja, 2016)	1.0	$(1 - 0.01 \exp(-Re_t^2)) * (1 - \exp(-0.0631Re_k))$	$(1 - \exp(-0.0165Re_k/n^{1/4}))^2 * (1 + 20.5/Re_t)$	$\varepsilon_w = \nu \left(\frac{\partial^2 k}{\partial y^2}\right)$

3 Fertilizer process as a case study

A case study has been performed where the modified Malin’s model and the CHC model described in Section 2.2 is used to simulate the flow of a non-Newtonian fluid in 3D. The case study consists of the flow through a bended pipeline as shown in Figure 3. Geometry of the pipelines for the process taken into consideration. Figure 3 and is a part of a process from a fertilizer production plant.

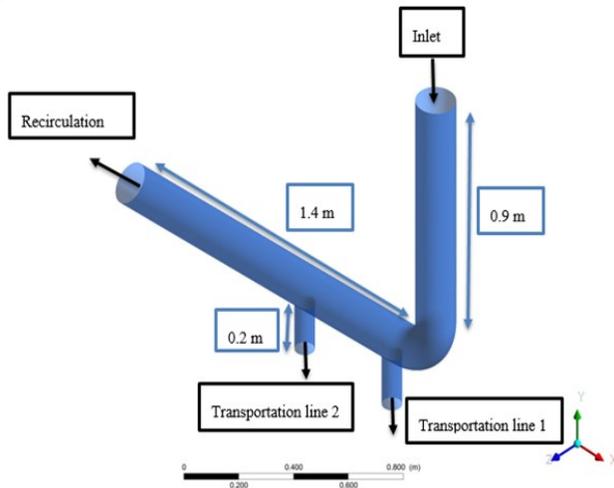


Figure 3. Geometry of the pipelines for the process taken into consideration.

The pipework consists of the main pipeline which has a bend of 90° and two additional pipelines (denoted as “Transportation line 1” and “Transportation line 2” in Figure 3) that are attached right after the main pipeline bend. The diameter of the main pipeline is 150 mm and that of the two transportation lines is 75 mm each. The thickness of the pipe walls is 5 mm and the pipe material is stainless steel. The characteristics of the pipe wall material are listed in Table 2.

The process fluid is pumped through the main pipeline inlet (denoted as “Inlet” in Figure 3) and is transported to the next stage of operation through the Transportation lines 1 and 2. The remaining process fluid is recirculated using the “Recirculation” line as denoted in Figure 3.

During the operation of the fertilizer plant, reduced pump capacity at the inlet of the pipe was observed. It is suspected that this could be due to the solidification of the fluid in the pipelines. This study thus focuses on understanding as to where and how such solidification

occurs in the pipelines. For this, a suitable model capable of taking into account the rheology of the non-Newtonian fluid is simulated in Ansys Fluent®. Detailed 3D simulations have been performed to observe where and how such solidification occurs for a given operating condition of the process giving a result for steady state under the selected boundary conditions.

4 Methods and materials

General settings: The simulation of the given process were performed using commercial CFD tool, Ansys Fluent® Academic Research, Release 16.2. The geometry was designed using Ansys DesignModeler and mesh was generated using sweep mesh method with the element size of 5 mm and 10° curvatures (1681179 cells in the simulation domain). The simulations were performed for the 3D steady-state regime using pressure based solver and absolute velocity formulation.

Momentum Boundary Conditions: The inlet velocities were varied from 3.0 m/s to 5.0 m/s in accordance to the different simulation case studies. The inlet velocities were defined to be the same over the whole cross sectional area of the pipe inlet. The pipeline outflows were defined as pressure outlets and gauge pressure was set to zero.

The estimation of the Reynolds-number for the internal pipe flow under consideration is challenging due to the non-Newtonian behavior of the fluid. In this study, the “generalized Reynolds-number” has been calculated using the Dodge and Metzner correlation (Dodge and Metzner, 1959) for the Power-Law fluids (Equation 9). For the calculation of the Reynolds-number, the inlet velocity (v_{bulk}) was taken as a bulk velocity.

$$Re = \frac{(\rho * v_{bulk}^{2-n} * D^n)}{K (0.75 + \frac{0.25}{n})^{n+8n-1}} \quad (9)$$

where D is diameter of the pipe, n is non-Newtonian index, K is consistency index.

Turbulence model: Turbulent flow was modelled using the standard CHC turbulence model and the modified Malin’s model as described in Section 2.3. The modified Malin’s turbulence model was implemented using UDF.

Thermal Boundary Conditions: The inlet fluid temperature as well as the temperature for the backflow

Table 2. Thermo-physical properties of the materials used in this study.

Material	ρ (kg/m ³)	C_p (J/(kg·K))	k_{th} (W/(m·K))	T_{liq} (K)	T_{sol} (K)	T_{ref} (K)	α (K)
Fluid	1400	1470	0.3	403	393	303	18100
Pipe Wall (stainless steel)	7900	515	16.6	Not used	Not used	Not used	Not used
Wall insulation (calcium silicate)	210	840	0.08	Not used	Not used	Not used	Not used

of the fluid at the outlets was set to 413 K. The wall heat transfer was modelled using “Thin Wall approach” and the wall temperature was set to 296 K.

Solution methods: The simulations were performed in a steady-state regime. The Semi-Implicit Method for Pressure Linked Equations (SIMPLE) method was used as a pressure-velocity coupling scheme. The first order upwind scheme was chosen as a spatial discretization method. The standard solution initialization was used for the standard CHC turbulence model. However, when simulating with the modified Malin’s turbulence model, the solution was initialized with the end results of the corresponding CHC model (simulated with the standard settings). The mesh sensitivity study was performed and the solution was assumed to be converged when the scaled residuals become constant. In addition, mass flowrates were checked and the solution was assumed to be converged if the mass flowrate imbalance was less than 10^{-4} . Mass outflow rates were computed using Surface Integrals available in Ansys Fluent®.

Material properties: The non-Newtonian behavior of the fluid was characterized with the Power-Law model for non-Newtonian fluids ($n = 0.05$). The thermo-physical properties of the fluid and the wall material (including wall insulation) used for simulations are summarized in Table 2.

5 Results and Discussion

5.1 Effect of temperature on fluid’s viscosity

To have a better understanding of how the temperature affects the viscosity of the non-Newtonian fluid and how this further influences the fluid solidification, simulations were modelled using standard CHC turbulence model. To start with, the fluid flow rate corresponding to the transient flow regime (with $Re = 6796$ and the inlet velocity of 3.0 m/s) was considered. Two cases were studied: (a) viscosity dependent only on the shear rate and, (b) viscosity dependent on both the temperature and the shear rate, and results are shown in Figure 4.

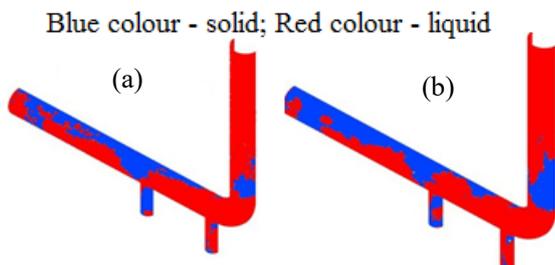


Figure 4. Solidification profiles using CHC model with 3 m/s inlet velocity for fluid with: (a) shear rate dependent viscosity; (b) both shear rate and temperature dependent viscosity.

Both simulations showed fluid solidification on the pipe walls, meanwhile no solidification occurred inside the pipe itself for both cases. Moreover, at the pipe

walls, observable differences in the solidification profiles were seen for the two cases. For the first case (viscosity not affected by temperature), smaller solidification zones were observed (Figure 4a) compared to the results obtained for the latter case (see Figure 4b).

When the flow was fully turbulent in nature ($Re = 18497$ with inlet velocity 5 m/s), the solidification profiles between the two cases did not reveal any significant differences (Figure 5). This might be due to the generation of higher turbulence inside the pipe and hence higher eddy viscosity. Therefore, changes in the molecular viscosity will not affect the solidification results significantly. Hence, if there are no perturbations in the actual process inlet velocity and is maintained at 5 m/s, the fluid solidification will not be significantly affected by the viscosity dependency on the temperature. Furthermore, for performing the simulations, the computational time is significantly smaller for case (a) than for case (b). Due to these reasons, the only case (a) will be studied further on and case (b) will not be discussed in the remaining part of this paper.

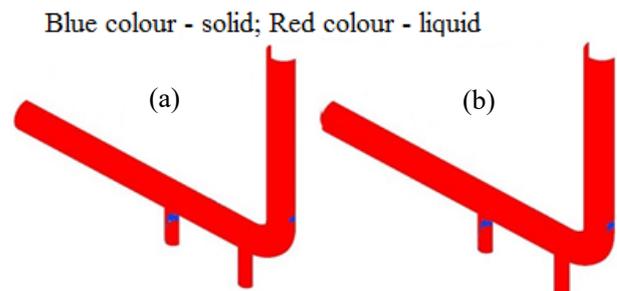


Figure 5. Solidification profiles using CHC model with 5 m/s inlet velocity for fluid with: (a) shear rate dependent viscosity; (b) both shear rate and temperature dependent viscosity.

5.2 Different turbulence models for solidification

The solidification of the fluid was studied using the standard CHC and the modified Malin’s turbulence models. The viscosity dependency on the temperature was ignored in the simulations due to the reasons described in Section 5.1.

Both the standard CHC and the modified Malin’s turbulence models showed fluid solidification on the pipe walls, meanwhile no solidification occurred inside the pipe itself. However, the location of the solidification zones and their sizes were different for these two models.

The standard CHC model showed more pipe wall zones that are susceptible to fluid solidification. The “risk zones”, i.e. the zones where the probability of the occurrence of fluid solidification is higher are the pipe wall towards the main pipe outlet, the transportation

pipeline walls and the backside of the pipe wall just before the main pipe bend (denoted as “zone 1”, “zone 2” and “zone 3” respectively in Figure 6). The occurrence of the fluid solidification on “zone 1” can be explained with the presence of a larger surface area of the wall that is exposed to the ambient air temperature. The larger the surface area, the larger will be the heat transfer between the ambient air and the fluid and hence more fluid will be solidified. Solidifications on “zone 2” and “zone 3” can be explained with the occurrence of lower fluid velocities at these zones. The lower the velocity, the slower the fluid molecules are moving inside the pipe and hence these molecules have comparatively more time to exchange heat with the ambient air through the pipe walls.

The modified Malin’s turbulence model showed different and smaller “risk zones” for solidification compared to the CHC model. The solidification of the fluid occurred after the main pipe bend and near the outlet of the transportation Pipelines 1 and 2 (denoted as “zone 4” and “zone 5” respectively in Figure 7). From the velocity profiles it can be seen that these zones are the low-velocity zones or the “dead zones”.

The occurrence of solidification on these zones can also be explained with a longer residence time of the fluid molecules (due to lower velocity) in these zones and hence more transfer of heat from the fluid to the ambient air through the pipe walls. Interestingly in contrast to the standard CHC model, the modified

Malin’s model can capture these “dead zones” which are the “risk zones” for fluid solidification. The occurrence of less solidification along the pipeline walls can be explained with the diffusive behaviour of the modified Malin’s model (see velocity profiles on Figure 6a and Figure 7a). The higher the turbulence, the higher is the velocity and hence the less is the residence time of the fluid molecule inside the pipeline. Therefore, this causes fewer fluid molecules to solidify on the pipe walls. The modified Malin’s model shows more evenly distributed mass outflow rates and hence evenly distributed velocities along the pipelines.

5.3 Effect of pipe wall insulation on solidification

To investigate if the insulation of the pipelines would minimize the occurrence of fluid solidification, the simulations were also performed with the changes in the thickness of the pipe wall insulation. The properties of the insulation layer are summarized in Table 2. The solidification was simulated using both the standard CHC and the modified Malin’s turbulence models.

Both turbulence models showed similar simulation results revealing that no solidification occurred when the pipe wall was insulated with a 50 mm-thick insulation layer jacket. This behavior was observed for the higher fluid velocity (5 m/s) as well as for the lower fluid velocity (3 m/s). However, without pipe insulation, the fluid solidifies at different places along the pipeline

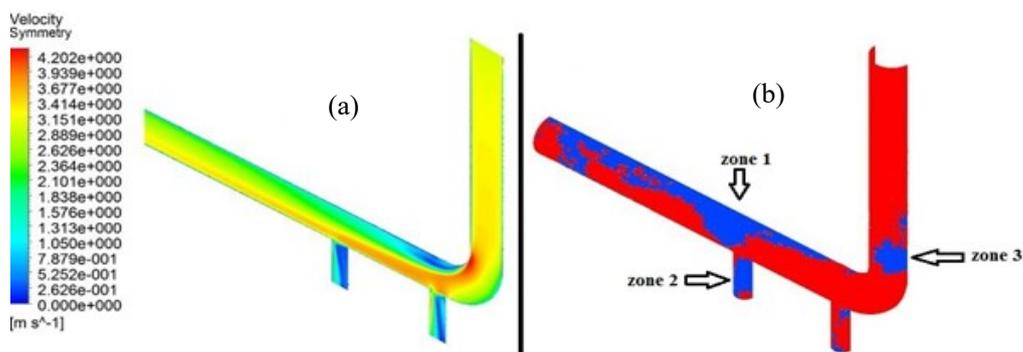


Figure 6. Simulation results with the CHC model (3 m/s inlet velocity): (a) velocity profile; (b) fluid solidification.

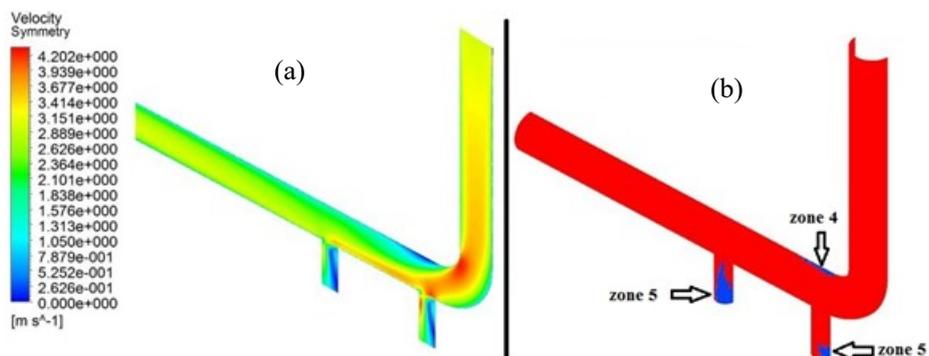


Figure 7. Simulation results with the modified Malin’s model (3 m/s inlet velocity): (a) velocity profile; (b) fluid solidification.

as described in the previous sections. Thus, the absence of the pipe wall insulation could be a probable reason for the detected reduced pump capacity during the operation process.

6 Conclusions

The solidification model was coupled to the non-Newtonian turbulence model in Ansys Fluent[®] by using UDFs. The “Enthalpy-Porosity” formulation was used to model fluid solidification, meanwhile modified Malin’s model was used to model turbulent fluid flow. The Malin’s eddy viscosity model was combined with the low-Reynolds-number $k-\varepsilon$ turbulence Chan-Hsieh-Chen model to account for the non-Newtonian behaviour of the process fluid.

As long as the inlet fluid velocity is maintained sufficiently high (turbulent flow), the occurrence of fluid solidification is not significantly affected by the viscosity dependence on the temperature. According to the simulation results, both the CHC and modified Malin’s turbulence models showed fluid solidification on the pipe walls, and not inside the pipe itself. The location and the size of fluid solidification zones were different for these two models. Modified Malin’s model showed more diffusive behaviour. It is difficult to conclude which of these models better represents the real process. Validation of the simulation results should be considered as a potential future work.

According to the simulation results, the use of pipe insulation can minimize the occurrence of fluid solidification on the pipe walls. Without insulation, solidification occurred on the pipe walls. Thus, the use of pipe insulation is beneficial and is highly recommended for the investigated process.

Acknowledgements

Authors kindly thank the department of Process Modelling and Control at Yara International ASA for providing the knowledge about the process. We would also like to thank the NPK department at Yara specially Carole Allen for her contributions in performing experiments.

References

Chang, K., Hsieh, W., & Chen, C. (1995). A modified low-Reynolds-number turbulence model applicable to recirculating flow in pipe expansion. *Journal of fluids engineering*, 117(3), 417-423.

Conde, R., Parra, M., Castro, F., Villafruela, J., Rodríguez, M., & Méndez, C. (2004). Numerical model for two-phase solidification problem in a pipe. *Applied Thermal Engineering*, 24(17), 2501-2509.

Dodge, D., & Metzner, A. (1959). Turbulent flow of non-Newtonian systems. *AIChE Journal*, 5(2), 189-204.

Ansys Fluent[®] 6.3 User's Guide, (2006).

Hirschberg, H. (1962). Freezing of piping systems. *Kaltetechnik*, 14, 314-321.

Lam, C., & Bremhost, K. (1981). A modified form of the $k-\varepsilon$ model for predicting wall turbulence. *ASME J. Fluids Eng*, 103, 456-460.

Malin, M. (1997). Turbulent pipe flow of power-law fluids. *International communications in heat and mass transfer*, 24(7), 977-988.

Malin, M. (1998). Turbulent pipe flow of Herschel-Bulkley fluids. *International communications in heat and mass transfer*, 25(3), 321-330.

Mathur, A., & He, S. (2013). Performance and implementation of the Launder–Sharma low-Reynolds number turbulence model. *Computers & Fluids*, 79, 134-139.

Motahar, S., & Khodabandeh, R. (2016). Experimental study on the melting and solidification of a phase change material enhanced by heat pipe. *International communications in heat and mass transfer*, 73, 1-6.

Myers, T., & Low, J. (2013). Modelling the solidification of a power-law fluid flowing through a narrow pipe. *International Journal of Thermal Sciences*, 70, 127-131.

Rojas, M. A., Castagna, J., Krishnamoorti, R., & Han Dh, T. A. (2008). Shear thinning behavior of heavy oil samples: laboratory measurements and modeling. Paper presented at the SEG Annual Meeting, Las Vegas.

Sharifi, N., Bergman, T. L., Allen, M. J., & Faghri, A. (2014). Melting and solidification enhancement using a combined heat pipe, foil approach. *International Journal of Heat and Mass Transfer*, 78, 930-941.

Versteeg, H. K., & Malalasekera, W. (2007). An introduction to computational fluid dynamics: the finite volume method. Pearson Education.

Vesjolaja, L. (2016). Flow Patterns of Highly Non-Newtonian Fluids in Industrial Applications. Master Thesis in Process Technology, Department of Process, Energy and Environmental Technology, University College of Southeast Norway.

Voller, V. R., & Prakash, C. (1987). A fixed grid numerical modelling methodology for convection-diffusion mushy region phase-change problems. *International Journal of Heat and Mass Transfer*, 30(8), 1709-1719.

Wei, S.-S., & Güçeri, S. I. (1988). Solidification in developing pipe flows. *International Journal of Heat and Fluid Flow*, 9(2), 225-232.

Zerkle, R. D., & Sunderland, J. (1968). The effect of liquid solidification in a tube upon laminar-flow heat transfer and pressure drop. *Journal of Heat Transfer*, 90(2), 183-189.

Simulation of Condensation in Compressed Raw Biogas Using Aspen HYSYS

Lars Erik Øi¹ Jon Hovland²

¹Department of and Process, Energy and Environmental Technology, University College of Southeast Norway

²SINTEF Tel-Tek, SINTEF Industry, Porsgrunn, Norway

lars.oi@usn.no

Abstract

Raw biogas typically contains 60 % methane, 40 % CO₂, small amounts of other components and is saturated with water. It is a question whether raw biogas can be compressed to high pressures without condensation. The aim of this work is to calculate the condensation limit under different conditions with varied temperature, pressure and gas composition using different equilibrium models. Traditionally, gas mixtures of methane, CO₂ and water are calculated in a process simulation program with standard models like Peng-Robinson (PR) and Soave-Redlich-Kwong (SRK). PR and SRK with the α -function replaced with a Twu α -function were also evaluated. For mixtures with only methane and CO₂ (dry biogas) all the models gave similar results. Under normal ambient temperatures (above 0 °C), a dry mixture with more than 40 % methane will not give any condensation. For biogas saturated with water, the different models gave similar results up to about 70 bar when binary coefficients were included, but above this pressure there were significant deviations between the models. The PR and SRK with standard or Twu α -function gave reasonable results for the dew-point predictions, but above about 70 bar the uncertainty increases.

Keywords: CO₂, methane, water, biogas, phase envelope, Aspen HYSYS

1 Introduction

Biogas is a mixture of methane, CO₂ and usually small amounts of other components like water and H₂S. Bio-methane (purified biogas) contains typically 97 % methane, and raw biogas typically contains 60 % methane, 40 % CO₂ and is saturated with water. Purified biogas is transported in cylinders under high pressure (typically 230 bar) or is injected into natural gas pipelines. It is a question whether raw biogas can be compressed to such high pressures because of the possibility of unwanted condensation. CO₂ and water in

the liquid phase is very corrosive, and may lead to operating problems. Hovland (2017) has concluded that condensation will normally be avoided when compressing dry biogas, but that it is dependent on the water concentration when compressing raw biogas.

Traditionally, gas mixtures of methane, CO₂ and water are calculated in a process simulation program with standard models like PR (Peng and Robinson, 1976) and SRK (Soave, 1972). It is known that these models simulate the gas phase and the condensation point quite accurately at least below the critical point (46 bar for methane and 74 bar for CO₂), but the resulting liquid phase may be questionable. Equilibrium models like HV (Huron and Vidal, 1979) and TST (Twu et al., 2005) have more parameters, and these models have been shown to give good results also for the liquid composition in CO₂ and water condensate in a mixture with methane. Other more complex models like SAFT-VR (Al Ghafri et al., 2014) and CPA (Austegard et al., 2006) have been used to describe this system.

There are a number of articles available studying the calculations and models for vapour/liquid equilibrium in the methane/CO₂/water-system (Austegard et al., 2006; Privat and Jaubert, 2014; Al Ghafri et al., 2017; Legoix et al., 2017). Austegard et al. conclude that a simple equation of state like SRK is satisfactory to describe the vapour phase, but only more complex models like e.g. SRK combined with a HV model is necessary to describe the liquid phase.

Water solubility in CO₂ gas or a mixture of CO₂ and methane shows a minimum for a constant temperature between 50 and 100 °C at a pressure in the range of the critical pressures (Austegard et al., 2006; Aasen et al., 2017; Privat and Jaubert, 2014). For this system, a minimum solubility is equivalent to a maximum dew point temperature. The water solubility in pure CH₄ is close to constant over a large pressure range close to the critical pressure (Privat and Jaubert, 2014).

Circone et al. (2003) have studied hydrate formation in CO₂/water mixtures and compared with methane hydrates. Hydrates in equilibrium in this system have been observed up to 13 °C (Al Ghafri et al., 2014) but will probably not be a practical problem above 0 °C.

There are several authors studying models for the system CO₂/water (Spycher et al., 2003; Longhi 2005;

Aasen et al., 2017). Aasen et al. have compared several different models including combinations of different equilibrium models. They conclude that a PR model in combination with HV and a volume shift gives the best results when also prediction of phase compositions and densities should be included.

The first aim of this work is to calculate the condensation limit for dry biogas and raw biogas under different temperature, pressure and gas composition using different equilibrium models. The last aim is to evaluate whether the selected models are satisfactory to predict the condensation limits for biogas compression and condensation.

2 Simulation Programs and Models

Commercial process simulation programs which have been used for calculating gas and liquid properties including condensation for mixtures of methane, CO₂ and water are Aspen Plus, Aspen HYSYS, Pro/II and ProMax. Process simulation programs are useful for simulation of such processes because several vapour/liquid equilibrium models are available in the programs.

In Aspen HYSYS, the equilibrium models SRK (Soave, 1972), PR (Peng and Robinson, 1976) and TST (Twu et al., 2005) are available for systems containing methane, CO₂ and water. In the SRK-Twu and PR-Twu models, the original α -function is replaced by a function from Twu et al. (1991). The PR model has only one adjustable parameter for each binary component pair while TST has 5 adjustable parameters for each binary pair. In Aspen Plus, the SRK model combined with HV is available, but this model is not available in Aspen HYSYS.

The equations for the SRK equation of state are shown in equations 1-8.

$$p = \frac{RT}{v-b} - \frac{a}{v(v+b)} \quad (1)$$

$$b = \sum_{i=1}^N x_i b_i \quad (2)$$

$$b_i = \frac{0,08664RT_c}{p_c} \quad (3)$$

$$a = \sum_{i=1}^N \sum_{j=1}^N x_i x_j (a_i a_j)^{0,5} (1 - k_{ij}) \quad (4)$$

$$a_i = a_{ci} \alpha_i \quad (5)$$

$$a_{ci} = \frac{0,42748R^2 T_c^2}{p_c} \quad (6)$$

$$\alpha_i = \left[1 + m_i \left(1 - T_r^{1/2} \right) \right]^2 \quad (7)$$

$$m_i = 0,48 + 1,574\omega_i - 0,176\omega_i^2 \quad (8)$$

P, T, v and R are the pressure, temperature, molar volume and universal gas constant, respectively.

T_c is the critical temperature, ω is the acentric factor and T_r is the reduced temperature defined as the ratio between T and T_c. The binary interaction parameter k_{ij} (equal to k_{ji}) is a constant that may be fitted for a binary component pair and x_i is the mole fraction for component i. In the PR equation, equation 1, 3, 6 and 8 are replaced by equation 9, 10, 11 and 12.

$$p = \frac{RT}{v-b} - \frac{a}{v(v+b)+b(v-b)} \quad (9)$$

$$b_i = \frac{0,077796RT_c}{p_c} \quad (10)$$

$$a_{ci} = \frac{0,457235R^2 T_c^2}{p_c} \quad (11)$$

$$m_i = 0,37464 + 1,54226\omega_i - 0,26992\omega_i^2 \quad (12)$$

In the SRK-Twu and PR-Twu equations, the α -function is replaced by equation 13. Fitted values for the parameters L, M and N for CO₂ and water can be found in (Twu et al., 2005).

$$\alpha_i = (T_r^{N(M-1)}) e^{L(1-T_r^{NM})} \quad (13)$$

The TST equation of state (equation 14) use the Twu α -function. Equation 3 and 6 in the SRK equation are similar for the TST equation except for the numerical constants which are 0.07407 (for b_i) and 0.47051 (for a_i).

$$p = \frac{RT}{v-b} - \frac{a}{(v-0,5b)(v+3b)} \quad (14)$$

The TST equation can be combined with the NRTL activity coefficient model to include non-ideal components. In the case of calculating the dew point of only methane, CO₂ and water, the use of NRTL coefficients is not necessary. A combined TST and NRTL model is used in Aspen HYSYS when modelling glycol dehydration (Twu et. al, 2005) including the highly non-ideal binaries between glycol and other components. The TST model must have model parameters fitted to experimental values to be accurate, especially when it is combined with a liquid model like NRTL.

In the standard version of SRK and PR, k_{ij} is a constant for each binary pair. When utilizing the default k_{ij} values in Aspen HYSYS, the k_{ij} values are constant for all component pairs except for water/CO₂ where it is a temperature dependent function. In literature, different optimized values for the k_{ij} values can be found because the parameters may be optimized for different conditions, e.g. for accurate prediction of either the gas phase or the condensate phase. For the calculation of dew points, it is reasonable to use binary interaction coefficients optimized for the gas phase.

3 Process Description and Simulation Specifications

3.1 Process description of raw biogas compression

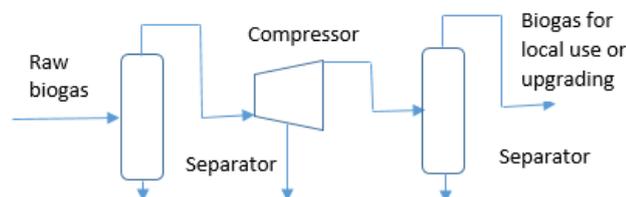


Figure 1. A traditional raw biogas compression process

A simplified diagram for a traditional raw biogas compression process is shown in Figure 1. Raw biogas is produced by anaerobic digestion where the temperature will typically be in the range 37 - 55 °C. Typical concentrations are 50-70 mol-% CH₄ (Pettersson and Wellinger, 2009). The rest is mainly CO₂, but traces of H₂S and organic components containing nitrogen and sulphur can be expected. The raw biogas will normally be saturated with water.

When the raw biogas production is above 100 Nm³/h, it can be reasonable to upgrade it on-site. However, with lower volumes it can be too expensive to have on-site upgrading. For volumes below 100 Nm³/h, Hovland (2017) suggests that it may be of interest to compress the gas to a high pressure, typically above 100 bar, and transport it to a facility for upgrading to biomethane (> 97 mol-% methane). Before or during compression, the raw biogas may be treated to remove some of the impurities. To reduce the amount of water, it can be cooled to remove some of the water as condensate before compression.

Possible removal options for different components in biogas cleaning and upgrading can be found in Pettersson and Wellinger (2009).

Condensed water formed after compression will normally be separated from the compressed gas. If the compression is performed in several stages with cooling between the stages, condensate may be removed after each stage. In most cases, condensation during compression is regarded to be a problem, and should be avoided.

3.2 Simulation specifications

Process simulations are performed for 4 conditions relevant for biogas production. For most of the cases the models PR, SRK, TST, PR-Twu and SRK-Twu are used. For all the conditions, calculations with the default parameters (especially the k_{ij} for water) are used. For some conditions other k_{ij} values are also used. In some cases, phase envelopes are calculated. In the dry gas

cases, the HYSYS 2-phase option was selected. In the cases including water, the ComThermo 3-phase option was selected. The 4 cases for the different conditions are specified as case A to D.

- A) Dry biogas with 60 mol-% methane and 40 mol-% CO₂ starts at 37 °C and 1 bar, is cooled to 10 °C and is compressed in three stages to 64 bar. The specifications are listed in Table 1.
- B) Dry biogas with 40 mol-% methane and 60 mol-% CO₂ starts at 37 °C and 1 bar, is cooled to 10 °C and is compressed in three stages to 64 bar.
- C) 60 kmol/h methane, 40 kmol/h CO₂ and 10 kmol/h water is mixed at 37 °C and 1 bar and cooled to 10 °C, then the liquid phase is removed and it is then compressed in three stages to 64 bar. The resulting water concentration was then approximately 1.2 %.
- D) 59.9 kmol/h methane, 40 kmol/h CO₂ and 0.1 kmol/h water is mixed at 37 °C and 1 bar, cooled to 10 °C, and then compressed in three stages to 64 bar.

Table 1. Specifications for the case A simulation

Parameter	Value
Inlet gas temperature	37 °C
Temperature before compression	10 °C
Inlet gas pressure	1 bar
Pressure after compression	64 bar
Inlet CH ₄ flow	60 kmol/h
Inlet CO ₂ flow	40 kmol/h

4 Process Simulation, Results and Discussion

4.1 Simulation of compression of dry methane/CO₂ mixture (Case A and B)

The Aspen HYSYS flow-sheet model for the base case simulation is presented in Figure 3.

Table 2. Dew point at 64 bar, cricondeterm and cricondenbar for a mixture of 60 mol-% methane and 40 mol-% CO₂ (Case A)

Model	T _{DEW} (°C)	T _{CRIC} (°C)	P _{CRIC} (bar)
PR	-23.6	-22.5	83.2
SRK	-23.3	-22.1	82.6
TST	-21.5	-20.3	79.6
PR-Twu	-24.7	-23.6	83.0
SRK-Twu	-24.0	-22.9	83.5

All the models calculate that condensation in dry gas with 60 % methane does not appear above $-20\text{ }^{\circ}\text{C}$. This is below normal process and operating conditions. In a cold climate temperatures may be lower than $-20\text{ }^{\circ}\text{C}$. The possibility for such low temperatures is also possible under depressurization, e.g. through a valve. It is expected that all the models in Table 2 are capable of calculating condensing conditions for dry biogas under depressurization reasonably accurate.

Case B is of interest because a 40 % methane and 60 % CO_2 has a dew point close to $0\text{ }^{\circ}\text{C}$. Based on earlier evaluations mainly based on measurements and calculations from Yang et al. (2015), Hovland (2017) concluded that below 58 % CO_2 , no condensation would appear if the temperature is kept above $-3\text{ }^{\circ}\text{C}$.

Table 3. Dew point at 64 bar, cricondeterm and cricondenbar for a mixture of 40 mol-% methane and 60 mol-% CO_2 (Case B)

Model	T_{DEW}	$T_{\text{CRIC}}\text{ (}^{\circ}\text{C)}$	$P_{\text{CRIC}}\text{ (bar)}$
PR	-5.4	-1.7	89.5
SRK	-5.2	-1.3	88.4
TST	-3,9	-0.5	82.6
PR-Twu	-6,3	-2.7	90.0
SRK-Twu	-5,8	-1.8	90.5

The calculated cricondeterms with different models have a maximum deviation of $2.2\text{ }^{\circ}\text{C}$. From this it is concluded that the results can be expected to be fairly accurate for all the models evaluated. No condensation will appear above $0\text{ }^{\circ}\text{C}$ in a dry biogas with less than 60 mole-% CO_2 . This conclusion is consistent with the conclusion from Hovland (2017) that no condensation should appear above $-3\text{ }^{\circ}\text{C}$ in a mixture with less than 58 mole-% CO_2 .

It was checked whether changing the k_{ij} parameter for methane/ CO_2 would change the results. It was found that the results in Table 3 were only slightly influenced by varying the k_{ij} parameter.

A phase envelope from Aspen HYSYS is shown in Figure 2. The important part for the evaluation of condensation is the dew point curve to the right. The point with the highest temperature is the cricondeterm. The point with the highest pressure is the cricondenbar. In the critical point for the mixture, slightly to the left of the cricondenbar, the compositions in both phases are equal. The calculated envelopes are similar up to about 70 bar, but above 70 bar, the decrease in dew point is different dependent on the model and dependent on the model parameters, especially the binary interaction parameters.

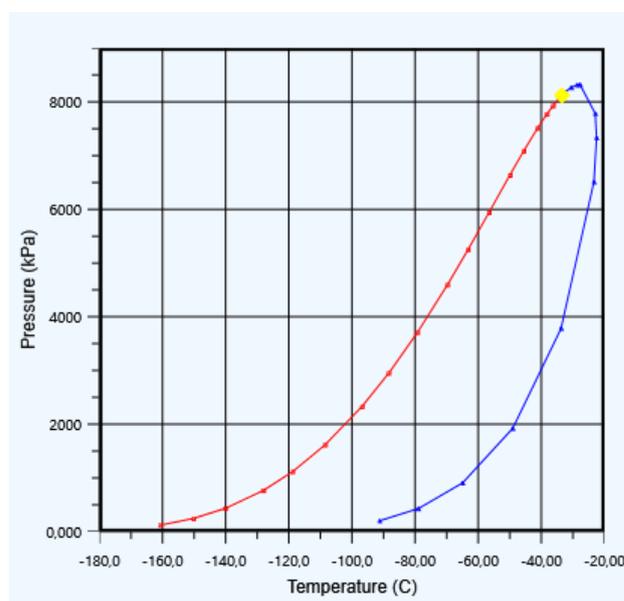


Figure 2. Phase envelope, Peng-Robinson, $\text{CH}_4=0.6$, $\text{CO}_2=0.4$, default $k_{ij}=0.1$

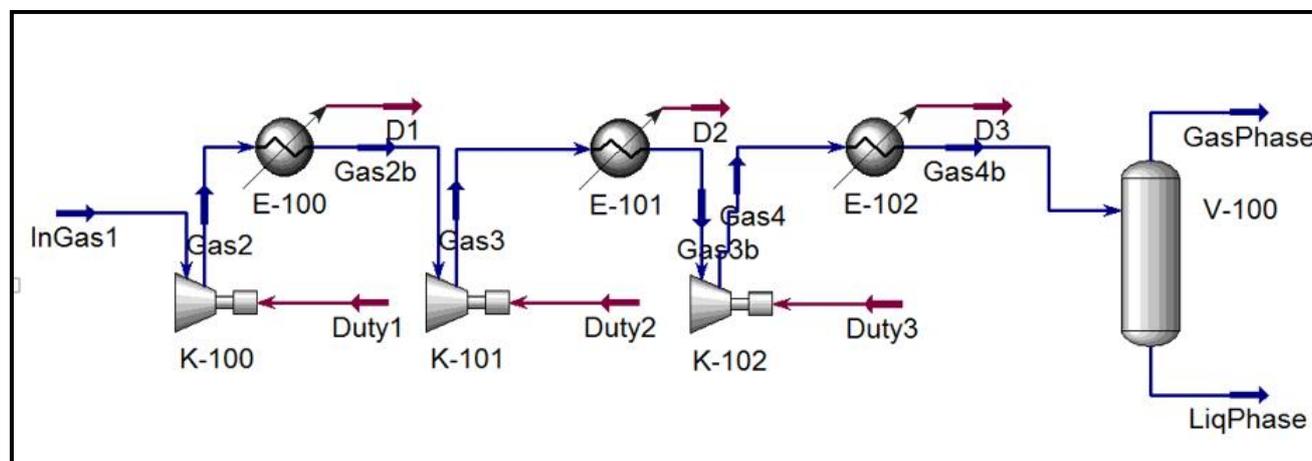


Figure 3. Aspen HYSYS flow-sheet for compression with intercooling and separation

4.2 Simulation of compression of a raw biogas including water, Case C and D

In Case C and D, the process was simulated with water included. In Case C, biogas saturated with water at 10 °C is simulated. The stream then contains approximately 1.2 mol-% water before the compressor. In Table 4, the dew point, cricondeterm and the pressure at the cricondeterm are shown using different models with default values from Aspen HYSYS. The first results for TST, PR-Twu and SRK-Twu in Table 4 were performed without k_{ij} for the water binaries. When the option including k_{ij} 's for water binaries was used, the dew point temperatures were much closer to the PR and SRK models.

Table 4. Dew point at 64 bar, cricondeterm and pressure at cricondeterm for a mixture of 60 mol-% methane and 40 % CO₂ saturated with water at 10 °C (Case C)

Model	T _{DEW} (°C)	T _{cric}	P _{cric} (bar)
PR	82.9	-	-
SRK	82.9	-	-
TST	76.6	78.8	96
PR-Twu	76.6	78.9	99
SRK-Twu	77.0	79.7	101
TST+k _{ij}	85.6	-	-
PRTwu+k _{ij}	85.5	-	-
SRKTwu+k _{ij}	85.6	-	-

A phase envelope from the calculations in Aspen HYSYS is shown in Figure 4. The curve to the right is the dew point curve. The other lines are phase boundary lines without importance for the condensate limit.

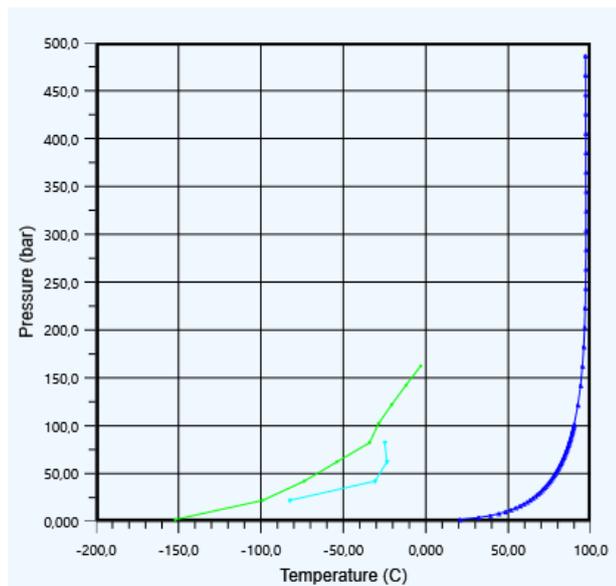


Figure 4. Phase envelope for PR, 60 kmol CH₄, 40 kmol CO₂, saturated with water at 10 °C. Default k_{ij} values.

The phase envelope calculated with SRK was similar to the envelope in Figure 4. However, with the other models, especially those calculated without k_{ij} values for water binaries, the dew point decreased significantly when the pressure increased above 70 bar. Using other k_{ij} values for water/CO₂, the envelope curve changed considerably above 70 bar. The reason for different binary coefficients is that they may be optimized for either the gas phase or the condensate phase. The CO₂ concentration in the condensed water phase was compared for the different models, and the difference was up to a factor of two. The difference between the calculated results from the different models above 70 bar are significant, so the uncertainty in this region must be regarded as large. There are few experimental points for the three component system in this region (Al Ghafri et al., 2014).

In Case D, the water mole fraction was specified to 0.001. This water concentration is possible to obtain if condensate is removed after intercooling steps in the compressor. Results are shown in Table 5. The phase envelope for PR is shown in Figure 5.

Table 5. Dew point at 64 bar, cricondeterm and pressure at cricondeterm for a mixture of 60 mol-% methane and 40 mol-% CO₂ with 0.1 % water (Case D)

Model	T _{DEW} (°C)	T _{CRIC} (°C)	P _{CRIC} (bar)
PR	26.5	27.6	89.7
SRK	26.9	28.0	89.2
TST+k _{ij}	28.8	32.1	122
PRTwu+k _{ij}	28.5	32.1	121
SRKTwu+k _{ij}	28.8	32.2	122

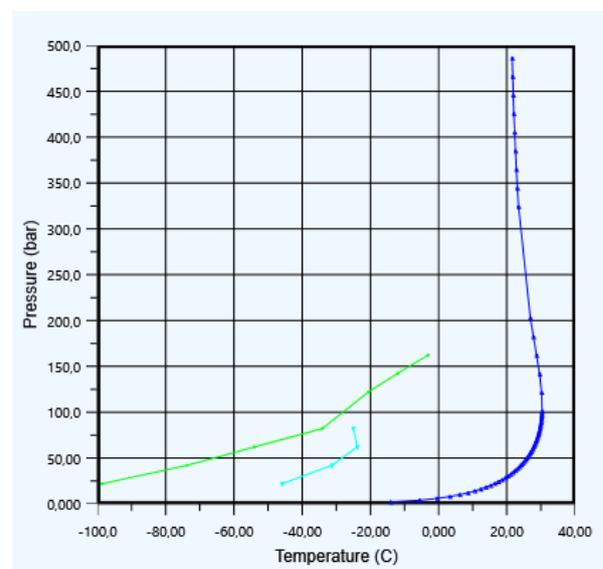


Figure 5. Phase envelope for PR model, 59,9 mol% CH₄, 40 mol% CO₂, 0.1 mol% water: Default k_{ij} for water/CO₂

The envelopes in Case D are similar for the different models up to about 70 bar, but above 70 bar, the decrease in dew point is different dependent on the model and dependent on the model parameters, especially the k_{ij} for water and CO₂. As for the case with a higher water concentration, the difference between the models above 70 bar is significant, so that the uncertainty above 70 bar must be regarded as large. The deviation is more than 5 °C for the calculation of the cricondenterm and more than 30 bar for the calculation of the pressure at the cricondenterm.

When using an equation of state, it is reasonable that the non-ideality and uncertainty increases when the pressure increases, and also when the mixture is close to condensation and close to the critical point which is order of magnitude 70 bar. The prediction of vapour/liquid equilibrium becomes more uncertain when water is added to CO₂ and methane because the physical interactions become more complex. The binary parameters which are meant to adjust for non-ideality are normally fitted in the region of vapour/liquid equilibrium which is below order of magnitude 70 bar.

5 Conclusion

The condensation limit for dry and raw biogas under different conditions with varied temperature, pressure and gas composition and using different equilibrium models were calculated.

For dry biogas, all the models Peng-Robinson (PR), Soave-Redlich-Kwong (SRK), PR-Twu, SRK-Twu and Twu-Sim-Tassone (TST) gave similar results. Biogas with 60 mol-% CH₄ and 40 mol-% CO₂ will have a condensation temperature less than -20 °C. Under normal ambient temperatures (above 0 °C), a mixture with more than 40 % methane will not give any condensation.

A process is simulated where raw biogas is cooled to 10 °C to remove water before compression. The results with biogas saturated with water at low pressure, the different models gave similar results up to about 70 bar, but above this pressure, different models gave different results. The results were dependent on the chosen value of the water/CO₂ binary interaction coefficient. The deviation in dew point temperature was about 6 K.

Both the standard PR and SRK models and the PR and SRK with the Twu α -function and with water/CO₂ binary coefficients included, gave reasonable results for the dew-point and to predict the conditions where it should be safe to avoid condensation.

For the calculation of dew points, it is recommended to use binary interaction coefficients optimized for the gas phase. If accurate calculations of the liquid composition after condensation is needed, a more advanced model like TST or HV with fitted parameters is recommended.

References

- A. Aasen, M. Hammer, G. Skaugen, J. P. Jakobsen and Ø. Wilhelmsen. Thermodynamic models to accurately describe the PVTxy-behaviour of water/carbon dioxide mixtures, *Fluid Phase Equilibria*, 442:125-139, 2017.
- S. Z. S. Al Ghafri, E. Forte, G. C. Maitland, J.J. Rodriguez-Henriquez and J. P. M. Trusler. Experimental and Modeling Study of the Phase Behaviour of (Methane + CO₂ + Water) Mixtures. *Journal of Physical Chemistry*, 118:14462-14478, 2014.
- A. Austegard, E. Solbraa, G. de Koeijer and M. J. Mølnvik. Thermodynamic models for calculating mutual solubilities in H₂O-CO₂-CH₄ mixtures. *Trans IChemE, Part A, Chem. Eng. Res. Des.*, 84(A9):781-7946, 2006.
- S. Circone, L. A. Stern, S. H. Kirby, W. B. Durham, B. C. Chakoumakos, C. J. Rawn, A. J. Rondinone and Y. Ishii. CO₂ Hydrate: Synthesis, Composition, Structure, Dissociation Behaviour, and a Comparison to Structure I CH₄ Hydrate. *Journal of Physical Chemistry B*, 2003. doi:10.1021/jp027391j
- J. Hovland. Compression of raw biogas – A feasibility study. Tel-Tek report 2217020-1, 2017. Available on <https://www.biogas2020.se/wp-content/uploads/2017/06/2217020-1compressionrawbiogas.pdf>
- M. J. Huron and J. Vidal. New mixing rules in simple equations of state for representing vapour-liquid equilibria of strongly non-ideal mixtures. *Fluid Phase Equilibria*, 3:255-271, 1979.
- L. N. Legoux, L. Ruffine, J. P. Donval and M. Haeckel. Phase Equilibria of the CH₄-CO₂ Binary and the CH₄-CO₂-H₂O Ternary Mixtures in the Presence of a CO₂-Rich Liquid Phase. *Energies*, 10(2034):1-11, 2017. Doi:10.3390/en10122034.
- J. Longhi. Phase equilibria in the system CO₂-H₂O I: New equilibrium relations at low temperatures. *Geochimica et Cosmochimica Acta*, 69, No. 3: 529-539, 2005.
- D. Peng and D. B. Robinson. A New Two-Constant Equation of State. *Ind. Eng. Chem. Fundam.*, 15(1):59-646, 1976.
- A. Petersson and A. Wellinger. Biogas upgrading technologies – developments and innovations. IEA Bioenergy Task 37, 2009. Available on www.iea-biogas.net
- R. Privat and J. N. Jaubert, *Predicting the Phase Equilibria of Carbon Dioxide Containing Mixtures Involved in CCS Processes Using the PPR78 Model*. InTech, 2014. Available on <http://dx.doi.org/10.5772/57058>.
- G. Soave. Equilibrium constants from a modified RedlichKwong equation of state. *Chemical Engineering Science*, 27:1197-1203, 1972.
- N. Spycher, K. Pruess and J. Ennis-King. CO₂-H₂O mixtures in the geological sequestration of CO₂. I. Assessment and calculation of mutual solubilities from 12 to 100 °C and up to 600 bar. *Geochimica et Cosmochimica Acta*, 67 (16):3015-3031, 2003.
- C. H. Twu, D. Bluck, J. R. Cunningham and J. E. Coon. A Cubic Equation of State with a New Alpha Function and a New Mixing Rule. *Fluid Phase Equilibria*, 69:33-50, 1991.
- C. H. Twu, V. Tassone, W.D. Sim and S. Watanasiri. Advanced equation of state method for modeling TEG-water for glycol gas dehydration, *Fluid Phase Equilibria*, 228-229:213-221, 2005.
- Z. Yang, M. Gong, Y. Zhou, X. Dong, X. Li, H. Li and J. Wu. Vapor-liquid equilibria of CH₄ + CO₂: A comparison between the molecular simulation and equation of state. *Science China Technological Sciences*, 58(4):650-658, 2015.

FMI4j: A Software Package for working with Functional Mock-up Units on the Java Virtual Machine

Lars Ivar Hatledal¹ Houxiang Zhang¹ Arne Styve² Geir Hovland³

¹Department of Ocean Operations and Civil Engineering, NTNU, Norway, {laht, hozh}@ntnu.no

²Department of ICT and Natural Sciences, NTNU, Norway, asty@ntnu.no

³Department of Engineering Sciences, UiA, Norway, geir.hovland@uia.no

Abstract

This paper introduces FMI4j, a software package for working with Functional Mock-up Units (FMUs) on the Java Virtual Machine (JVM). FMI4j is written in Kotlin, which is 100% interoperable with Java, and consists of programming APIs for parsing the meta-data associated with an FMU, as well as running them. FMI4j is compatible with FMI version 2.0 for Model Exchange (ME) and Co-Simulation (CS). Currently, FMI4j is the only software library targeting the JVM supporting ME 2.0. In addition to provide bare-bones access to such FMUs, it provides the means for solving them using a range of bundled fixed- and variable-step solvers. A command line tool named FMU2Jar is also provided, which is capable of turning any FMU into a Java library. The source code generated from this tool provides type-safe access to all FMU variables explicitly through the API (Application Programming Interface). Additionally, the API is documented with key information retrieved from the FMU meta-data, allowing essential information such as the description, causality and start value of each variable to be seamlessly exposed to the user through the Integrated Development Environment (IDE).

Keywords: FMI, Co-Simulation, Model Exchange, JVM

1 Introduction

Recently, several research projects at NTNU Aalesund (Hatledal et al., 2015; Chu et al., 2017, 2018) and others (Skjong et al., 2017; Sadjina et al., 2017) involve co-simulation and virtual prototyping. Virtual prototyping refers to a vision where models, or *virtual prototypes*, of complex systems can be developed, tested, and amended with a trial-and-error approach. As computer technology develops it becomes possible to make an increasing part of the necessary tests based on simulations. However, as complex models often require components from several different domains, perhaps developed in separate domain-specific tools, a standard is required to fit them all together.

FMI (Blochwitz et al., 2011, 2012) is a tool independent standard to support both Model Exchange (ME) and Co-Simulation (CS) of dynamic models. The first version of the standard, FMI 1.0, was released in 2010. In 2014, version 2.0 was released, which merged the two standards

and incorporated some major enhancements compared to the initial release. As such, version 2.0 is not backwards compatible with version 1.x.

A model implementing the FMI standard is known as a Functional Mock-up Unit (FMU), and is distributed as a zip-file with the extension *.fmu*. It contains:

- An XML-file containing meta-data of the packaged model, named *modelDescription.xml*.
- C-code implementing a set of functions defined by the FMI standard.
- Other optional resources required by the implementation.

FMI4j, the software package introduced in this paper, aims to simplify interaction with FMUs, and consists of easy to use software APIs for parsing and simulating FMUs on the JVM, as well as a tool for wrapping FMUs into Java libraries, named FMU2Jar. Kotlin was chosen as the implementation language as it is 100% interoperable with Java, while offering several language improvements such as null safety and less boilerplate code. From a usability perspective, invoking FMI4j code from Java feels no different than calling any other Java library.

The source code is published online under the permissive open-source MIT license and can be accessed through GitHub¹. Here, pre-compiled FMU2Jar binaries are also available. The APIs are available on maven central². Only version 2.0 and upwards are planned to be supported.

The rest of the paper is organized as follows. First some related work is given, followed by a presentation of the FMI4j software package. Finally, a conclusion and future work are given.

2 Related work

Since the release of the FMI standard, several software libraries implementing the standard have been published. An overview of such libraries for importing/invoking FMUs is given in Table. 1.

¹<https://github.com/SFI-Mechatronics/fmi4j>

²<http://mvnrepository.com/artifact/no.mechatronics.sfi.fmi4j>

Table 1. Software libraries providing FMI import

Name	Language				FMI support				Version	License
	C	C++	Java	Python	CS		ME			
					v1.0	v2.0	v1.0	v2.0		
FMI Library	x				x	x	x	x	2.0.3	BSD
FMU SDK		x			x	x	x	x	2.0.4	BSD
FMI++		x	x ^a	x ^a	x	x	x ^b	x ^b	-	BSD
PyFMI				x	x	x	x ^b	x ^b	2.4	LGPLv3
FMPy				x	x	x	x ^b	x ^b	0.2.5	BSD
JFMI			x		x		x		1.0.2	MIT
JavaFMI			x		x	x			2.24.5	LGPLv3

^a Through SWIG

^b Can solve ME FMUs

The FMI Library (FMIL) (JModelica, 2017) and FMU SDK (QTronic, 2014), written in C and C++ respectively, provide basic access to low-level FMI functions and is often used as base for creating more high-level libraries.

FMI++ (Widl et al., 2013) is a high level utility package for FMI based on FMIL for ME and CS, written in C++, that aims to bridge the gap between the basic FMI specification and the typical requirements of simulation tools. Interfaces for Python and Java can be generated using the Simplified Wrapper and Interface Generator (SWIG). While the Python interface for Windows comes pre-built, other packages must be built from source.

PyFMI (Andersson et al., 2016) is a high-level python library for interacting with FMUs, maintained by Modelon AB. It contains co-simulation masters for simulation of weakly coupled systems and provides a connection to the simulation package Assimulo (Andersson et al., 2015), a Python package for solving first or second order explicit ordinary differential equations (ODEs) or implicit ordinary differential equations (DAEs). PyFMI is available as a stand-alone package or as part of the JModelica.org distribution.

FMPy (Dassault Systems, 2017) is a free python library from Catia Systems for simulating FMUs. FMPy supports both FMI 1.0 and 2.0 for ME and CS. Using solvers from the Sundials package, FMPy can be used to solve ME FMUs. It also features both a command line utility and a GUI for running and presenting simulation results. FMPy and PyFMI may seem very similar, however there is a major difference in that FMPy is implemented in pure Python, whereas PyFMI acts as a wrapper for FMIL, with additional high-level features.

JFMI (Broman et al., 2013b) is a low-level wrapper for FMI 1.0 for CS and ME. The latest version, 1.0.2, was released in 2013. Although the library supports both FMI-CS and FMI-ME, a flexible solving mechanism for FMI-ME is not provided.

JavaFMI (Cortes Montenegro, 2014) is a set of components for working with the FMI standard using Java, developed by SIANI institute (Las Palmas University). JavaFMI is still actively maintained and offers cross plat-

form support for FMI version 1.0 and 2.0 for CS. A neat feature of JavaFMI is the ability to export Java code as CS FMUs.

While several FMI implementations exist, also for the JVM. Only JavaFMI is maintained, however it lacks FMI for ME support. It could be argued that FMI++ is available on the JVM by means of SWIG bindings, however, the library must be built from source which is not straightforward and requires a number of native dependencies.

As such, it can be argued that there is still room for an alternative, easy to use FMI implementation for the JVM that supports both CS and ME FMUs.

3 FMI4j

This section introduces FMI4j, a software package for working with Functional Mock-up Units on the JVM, developed by researchers at NTNU Aalesund. It is implemented from scratch in Kotlin and provides a high-level API for interacting with FMUs on the JVM (Java, Scala, Groovy, Kotlin etc.) that implements FMI 2.0 for CS and/or ME. When provided with an solver, FMI4j is able to solve ME FMUs. Such instances share a common interface with ordinary CS FMUs, that expose the most important FMI functions related to stepping a FMU forward in time.

Furthermore, FMI4j through the FMU2Jar tool is, to the best of the authors knowledge, the only publicly available software that utilizes the provided meta-data in an FMU in order to generate a high-level API tailored towards it. E.g provide type-safe and documented access to named variables directly through the API.

The different components available in the package is:

1. *fmi-modeldescription* - A library for parsing the meta-data found in the *modelDescription.xml* located within an FMU.
2. *fmi-import* - A library for loading and running FMUs on the JVM. Supports FMI 2.0 for CS and ME.
3. *FMU2Jar* - A command line tool for turning an FMU into a Java library (.jar).

FMU2Jar is dependent on `fmi-import`, which again depends on `fmi-modeldescription`. Artifacts from both libraries are hosted on *The Central Repository*³ hosted by Sonatype. A collection of the most notable FMI4j classes are shown in Figure. 1, some of which are described in more detail in the following sections.

3.1 fmi-modeldescription

`fmi-modeldescription` is a lightweight API for parsing the meta-data found in the `modelDescription.xml` located inside an FMU. Useful when only static information about the FMU is required. For instance if you only want to display static information about the FMU in a web-app or when generating source code tailored towards a particular FMU, as in the case for FMU2Jar.

FMU4j can parse the model-description given both a file and URL reference to the FMU location. It can also handle raw XML input. Usage is demonstrated in Listing. 1. For brevity, code snippets are provided in Kotlin.

As seen in Figure. 1 there are several different interfaces representing the model-description. The *CommonModelDescription* interface represents common meta-data found in both CS and ME FMUs, while the *SpecificModelDescription* interface contain additional common information found in the `<ModelExchange>` and `<CoSimulation>` XML elements for ME and CS FMUs respectively. Furthermore, the *ModelExchangeModelDescription* and *CoSimulationModelDescription* interfaces contains type-specific information located within the same entries.

Listing 1. Parsing the model-description file from an FMU.

```
File fmuFile = File("path/to/fmu.fmu")

//includes common FMI entries only
val md = ModelDescriptionParser.parse(
    fmuFile)

//includes also CS specific entries
val cs_md = md.
    asCoSimulationModelDescription()

//includes also ME specific entries
val me_md = md.
    asModelExchangeModelDescription()
```

3.2 fmi-import

`fmi-import` is responsible for loading and simulating FMUs. It relies on `fmi-modeldescription` for parsing and Java Native Access (JNA) for invoking the native FMI functions written in C. For integration of differential equations, it relies on the Apache Commons Math library (Apache, 2017).

The API for reading and writing variables is given in Listing. 2 and 3 respectively. For convenience, FMU variables can be accessed through the *ScalarVariable* instance representing the variable entry from the XML. As seen, variables can also be accessed in a more FMI idiomatic

way using the *variableAccessor* handle found within an FMU implementation.

Listing 2. Read API.

```
val instance: FmiSimulation = ...
val speed: Double
    = instance.variableAccessor
        .readReal("speed")
```

Listing 3. Write API.

```
val instance: FmiSimulation = ...
val speedVariable: RealVariable = ...
val status: FmiStatus
    = speedVariable.write(1.0)

// or
val status: FmiStatus
    = instance.variableAccessor
        .writeReal("speed", 1.0)
```

A description of some of the most notable classes found within the `fmi-import` module are given below.

- *Fmu* - Represents an FMU file on disk. Responsible for extracting the FMU, and acts as a factory for new FMU instances. This allows extracted FMU content to be re-used across instances. On JVM shutdown, it will handle any necessary clean-up related to previously instantiated FMU instances and will also delete the extracted FMU contents.
- *FmuInstance* - Represents a generic FMU instance, exposing some of the most common functions.
- *FmiSimulation* - Extends the *FmuInstance* interface with time stepping. Common interface for CS FMUs and self-integrating ME FMUs.
- *AbstractFmuInstance* - Base class for all implemented FMU classes. Wraps the model description and a handle to the underlying native code belonging to the loaded FMU. Also contains common boilerplate code.
- *CoSimulationFmuInstance* - Represents a CS FMU instance. Implements the *FmiSimulation* interface. Example usage is given in Listing. 4. Implements the FMI extension for predictable step sizes proposed in (Broman et al., 2013a), enabling step-size negotiation between FMUs. More specifically, the extension adds the capability flag *canProvideMaxStepSize* and a CS specific C procedure, *fmiGetMaxStepSize*, which is an upper bound on the step-size that the FMU can accept.
- *ModelExchangeFmuInstance* - A bare-bones class for interacting with instances of ME FMUs. The responsibility of solving the FMU is left to the user, as the class simply provides access to the underlying FMU functions. Instantiated as seen in Listing. 5.
- *ModelExchangeFmuStepper* - Wraps an instance of a *ModelExchangeFmuInstance*, while implementing

³<https://search.maven.org/>

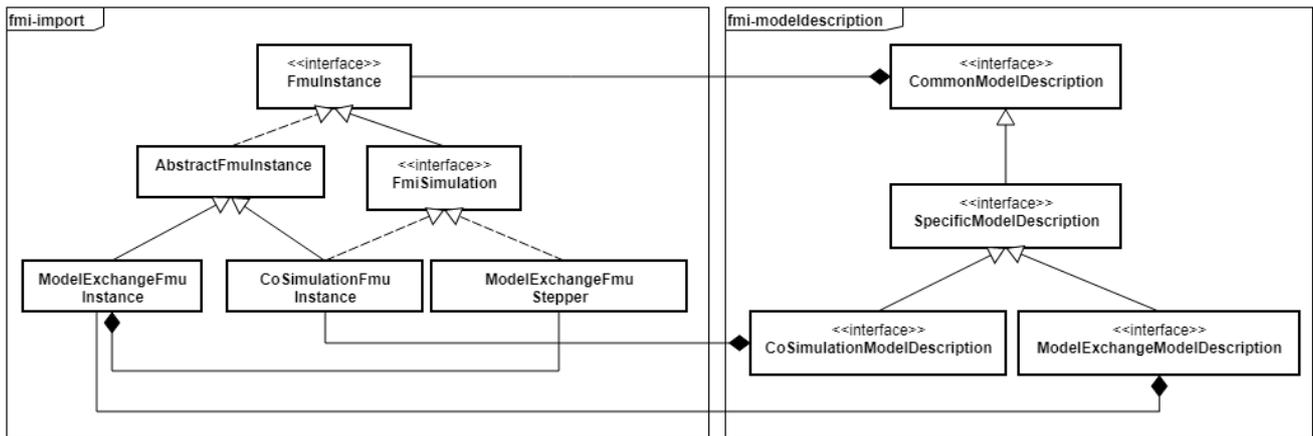


Figure 1. Simplified UML view of core FMI4j classes.

the *FmiSimulation* interface. Allows ME FMUs to be treated similar to CS FMUs. As seen in Listing. 6, it is instantiated very similarly to the *ModelExchangeFmu*, although a solver is required. For this purpose, FMI4j comes bundled with the Apache Commons Math package, which includes a range of both fixed and adaptive step-size solvers. A complete overview of the available solvers is given in Table. 2 and 3.

Listing 4. Loading and running an CS FMU

```

val fmuFile = File("path/to/fmu.fmu")
val slave = Fmu.from(fmuFile)
                .asCoSimulationFmu()
                .newInstance()

// assign start values here

slave.init() //throws on error

val dt = 1.0/100
val stop = 10.0
while (slave.currentTime < stop) {
    slave.doStep(dt)
}
slave.terminate()

```

Listing 5. Instantiating an ME FMU.

```

val file = File("path/to/fmu.fmu")
val slave = Fmu.from(file)
                .asModelExchangeFmu()
                .newInstance()

```

Listing 6. Instantiating a self-integrating ME FMU.

```

...
val solver = EulerIntegrator(1E-3)
val slave = Fmu.from(file)
                .asModelExchangeFmu(solver)
                .newInstance()

```

Table 2. Fixed-step solvers available in the Apache Commons Math package.

Name	Integration Order
Euler	1
Midpoint	2
Classical Runge-Kutta	4
Gill	4
3/8	4
Luther	6

Table 3. Adaptive step-size solvers available in the Apache Commons Math package.

Name	Order	Error Estimation Order
Higham and Hall	5	4
Dormand-Prince 5	5	4
Dormand-Prince 8	8	5 and 3
Gragg-Bulirsch-Stoer	variable	variable
Adams-Bashforth	variable	variable
Adams-Moulton	variable	variable

3.3 FMU2Jar

FMU2Jar is a command line tool for packaging an FMU into a Java library, allowing the FMU to be used as any other Java library. The generated library also exposes all variables from the FMU through a type-safe API. That is, named functions for getting and setting typed variable values are generated for each accessible variable in the FMU. These are documented with information retrieved from the associated entry in the model-description. This makes it easier to use the FMU, as all variables and associated documentation can be browsed from within an IDE, as seen in Figure. 2. Also, variables are grouped by causality for easier look-up. Both CS and ME FMUs are supported, with ME FMUs being wrapped as CS FMUs and subsequently solved using the solver provided on initialization, as seen in Listing. 7.

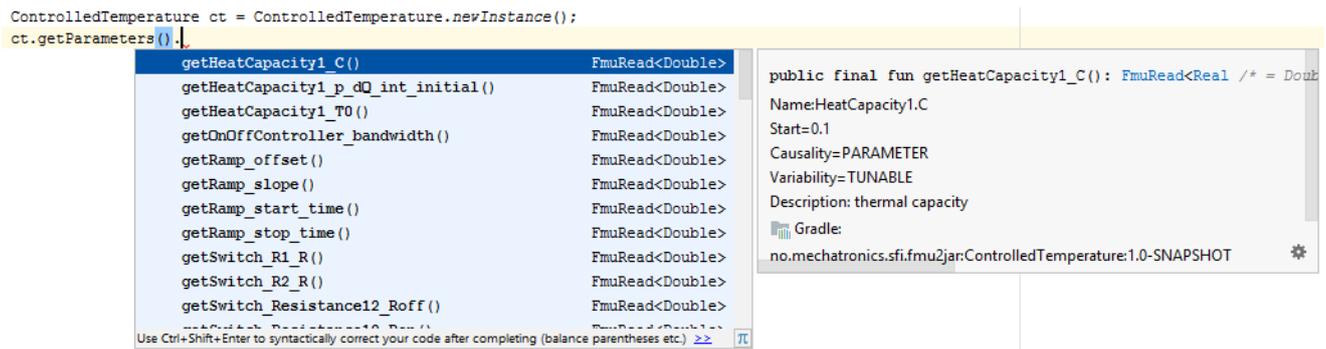


Figure 2. FMU named *ControlledTemperature* wrapped as a Java library using FMU2jar, then imported into IntelliJ IDE. From within the IDE, the user can browse and read documentation on all available variables.

Listing 7. Instantiating both a CS and a self-integrating ME FMU generated by FMU2Jar.

```
// Given an FMU that supports
// both CS & ME:
// First instantiate a CS FMU
ControlledTemperature
    .newInstance()
    .use { fmu ->
        ...
    }

// and then a self-integrating ME FMU
val solver = EulerIntegrator(1E-3)
ControlledTemperature
    .newInstance(solver)
    .use { fmu ->
        ...
    }
```

The Command Line Interface (CLI) is shown in Listing 8. When supplying *-mavenLocal* as an argument, a maven artifact is published to the local maven repository (.m2 folder). This allows the user to easily include the library in a software project using a build system such as Apache Maven or Gradle. The user may also save the generated .jar into a specified folder and reference it explicitly.

Listing 8. FMU2Jar CLI

```
-fmu <arg>    Path to the FMU
-help        Prints this message
-mavenLocal  Should the library be
             published to maven local?
-out <arg>   Specify where to copy the
             generated .jar
```

FMU2Jar is most useful when working with FMUs programmatically, as its advantages such as variable look-up, type-safe variable access and in-IDE documentation has little to no function in common GUI based simulation environments such as OpenModelica, SimulationX, etc.

3.4 Performance

Table 4 shows how FMI4j compares to some of the other FMI libraries in terms of performance. The table shows the time required in order to step two different test FMUs

Table 4. Performance comparison

Library	Execution time [ms]	
	<i>bouncingBall.fmu</i>	<i>TorsionBar.fmu</i>
FMIL	4	2801
JavaFMI	54	5843
FMI4j	53	5979
FMPy	60	9662

forward in time. Both FMUs implements the CS standard and was downloaded from the official SVN repository for test FMUs. A step-size of $1E-2$ and target time equal to 100 seconds is used for the *bouncingBall.fmu* exported from FMUSDK, while a step-size of $1E-5$ and target time equal to 12 seconds was used for the *TorsionBar.fmu* exported from 20Sim. For each time-step, a read call on a real-valued output variable is performed. The tests were performed on a i7-4770 CPU running Windows 10. From the results we see that the native FMIL library is faster than FMI4j by a good margin. This is to no surprise as there is some overhead related to calling native functions from Java (Kurzyniec and Sunderam, 2001). Performance wise, FMI4j and JavaFMI are practically identical as they both relies on JNA to handle native code execution. FMPy, which runs in an interpreted language, is slower in both test cases.

4 Conclusion and Future Work

This paper presents a high-level software package for working with FMUs on the JVM platform. It includes both a library for parsing the model-description file and also for running the FMUs, as well as a tool for wrapping FMUs as Java libraries, named FMU2Jar. Both FMI 2.0 for Co-simulation and Model-Exchange is supported. Currently, it is the only library implemented for the JVM to support version 2.0 of the ME standard. Using one of the bundled solvers from the Apache Commons Math library, such FMUs can be solved directly by the library.

The FMU2Jar tool makes it easier to work with a specific FMU by wrapping it as a Java library, and generate maven artifacts for it, which facilitates easy integration

with popular build tools such as *Maven* and *Gradle*. Furthermore, variables are exposed through the API as type-safe method calls with documentation retrieved from the model-description.

Recently, the FMI steering committee released a feature list for version 3.0 of the FMI standard (FMI steering committee, 2018). As a future work, we aim to support this standard some time after it has been officially released.

In the future FMI4j may also include the option to export FMUs from Java byte-code.

A request to list FMI4j on the official FMI tools page has been submitted, and is pending approval. If or when new features are added to the software, the capabilities shown in this entry will be updated accordingly.

5 Acknowledgement

The research presented in this paper is supported by the Norwegian Research Council, SFI Offshore Mechatronics, project number 237896.

References

- Christian Andersson, Claus Führer, and Johan Åkesson. As-simulo: A unified framework for ode solvers. *Mathematics and Computers in Simulation*, 116:26–43, 2015.
- Christian Andersson, Johan Åkesson, and Claus Führer. Pyfmi: A python package for simulation of coupled dynamic models with the functional mock-up interface. *Technical Report in Mathematical Sciences*, 2016(2), 2016.
- Apache. Apache commons math, 2017. URL <http://commons.apache.org/proper/commons-math/>. (Date accessed 23-June-2018).
- Torsten Blochwitz, Martin Otter, Martin Arnold, Constanze Bausch, H Elmqvist, A Junghanns, J Mauß, M Monteiro, T Neidhold, D Neumerkel, et al. The functional mockup interface for tool independent exchange of simulation models. In *Proceedings of the 8th International Modelica Conference; March 20th-22nd; Technical Univeristy; Dresden; Germany*, number 063, pages 105–114. Linköping University Electronic Press, 2011.
- Torsten Blochwitz, Martin Otter, Johan Åkesson, Martin Arnold, Christoph Clauss, Hilding Elmqvist, Markus Friedrich, Andreas Junghanns, Jakob Mauss, Dietmar Neumerkel, et al. Functional mockup interface 2.0: The standard for tool independent exchange of simulation models. In *Proceedings of the 9th International MODELICA Conference; September 3-5; 2012; Munich; Germany*, number 076, pages 173–184. Linköping University Electronic Press, 2012.
- David Broman, Christopher Brooks, Lev Greenberg, Edward A Lee, Michael Masin, Stavros Tripakis, and Michael Wetter. Determinate composition of fmus for co-simulation. In *Proceedings of the Eleventh ACM International Conference on Embedded Software*, page 2. IEEE Press, 2013a.
- David Broman, Christopher Brooks, Edward A. Lee, Thierry S. Noudui, Stavros Tripakis, and Michael Wetter. Jfmi - a java wrapper for the functional mock-up interface, 2013b. URL <https://ptolemy.eecs.berkeley.edu/java/jfmi/>. (Date accessed 23-June-2018).
- Yingguang Chu, Lars Ivar Hatledal, Houxiang Zhang, Vilmar Æsøy, and Sören Ehlers. Virtual prototyping for maritime crane design and operations. *Journal of Marine Science and Technology*, pages 1–13, 2017.
- Yingguang Chu, Lars Ivar Hatledal, Vilmar Æsøy, Sören Ehlers, and Houxiang Zhang. An object-oriented modeling approach to virtual prototyping of marine operation systems based on functional mock-up interface co-simulation. *Journal of Offshore Mechanics and Arctic Engineering*, 140(2):021601, 2018.
- Johan Sebastian Cortes Montenegro. Javafmi una librería java para el estándar functional mockup interface. 2014.
- Dassault Systems. Fmpy, 2017. URL <https://github.com/CATIA-Systems/FMPy>. (Date accessed 23-June-2018).
- FMI steering committee. Fmi version 3.0: Status, 2018. URL <https://fmi-standard.org/downloads/>. (Date accessed 23-June-2018).
- Lars Ivar Hatledal, Hans Georg Schaathun, and Houxiang Zhang. A software architecture for simulation and visualisation based on the functional mock-up interface and web technologies. In *Proceedings of the 56th Conference on Simulation and Modelling (SIMS 56), October, 7-9, 2015, Linköping University, Sweden*, number 119, pages 123–129. Linköping University Electronic Press, 2015.
- JModelica. Fmi library, 2017. URL <http://www.jmodelica.org/FMILibrary>. (Date accessed 09-December-2017).
- Dawid Kurzyniec and Vaidy Sunderam. Efficient cooperation between java and native codes—jni performance benchmark. In *The 2001 international conference on parallel and distributed processing techniques and applications*. Citeseer, 2001.
- QTronic. Fmu sdk, 2014. URL <http://www.qtronic.de/de/fmusdk.html>. (Date accessed 23-June-2018).
- Severin Sadjina, Lars T Kyllingstad, Martin Rindarøy, Stian Skjong, Vilmar Æsøy, Dariusz Eirik Fathi, Vahid Hassani, Trond Johnsen, Jørgen Bremnes Nielsen, and Eilif Pedersen. Distributed co-simulation of maritime systems and operations. *arXiv preprint arXiv:1701.00997*, 2017.
- Stian Skjong, Martin Rindarøy, Lars T Kyllingstad, Vilmar Æsøy, and Eilif Pedersen. Virtual prototyping of maritime systems and operations: applications of distributed co-simulations. *Journal of Marine Science and Technology*, pages 1–19, 2017.
- Edmund Widl, Wolfgang Müller, Atiyah Elsheikh, Matthias Hörtenhuber, and Peter Palensky. The fmi++ library: A high-level utility package for fmi for model exchange. In *Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES), 2013 Workshop on*, pages 1–6. IEEE, 2013.

Model predictive control of district heating system

Frode Lie-Jensen¹ Andreas Aannø¹ Elena Aleksandrova¹ Anders Westli¹ Morten Nielsen²
Tiina Komulainen¹

¹Faculty of technology, Arts & Design, Department of Mechanical, Electronic & Chemical engineering, OsloMet, Norway, timako@oslomet.no

²Fortum Oslo Varme AS, morten.nielsen.luraas@fortum.no

Abstract

District heating system (DHS) is a widely used and increasingly popular energy source in cities. The uncertainty in the heat load (HL) due to customer demand fluctuations makes unit commitment (UC) and heat production unit (HPU) control a complex task. This case study of the DHS at Fortum Oslo Varme AS (FOV) aims to find a strategy to optimize and fully automate UC and HPU. Our results suggests this can be accomplished by using model predictive control (MPC) to control HPU power and flow rate, mixed integer linear programming (MILP) optimization to solve UC problem, and multiple linear regression (MLR) model to predict the HL. We also show that the fuel cost can be reduced significantly.

Keywords: district heating, model predictive control, system identification, unit commitment problem, heat load prediction

NOMENCLATURE

Symbol	Explanation
<i>Plant variables</i>	
$K1 - 4$	HPU 1-4
T_{kt}	Temperature, outlet HPU
T_{nt}	Temperature, outlet heat exchanger
T_{kr}	Temperature, inlet HPU
T_{nr}	Temperature, inlet heat exchanger
\dot{V}	Volume flow, consumer side of heat exchanger
\dot{Q}	Power input to HPU
<i>Heat load prediction</i>	
α	constant
y	predicted value, dependent variable
x_i	independent variables
β_i	regression constants
ξ	error term
<i>Unit commitment optimization</i>	
T_h	Time horizon
N	Number of HPU
C_n^t	Energy cost for HPU n at time t
P_n^t	Power production HPU n at time t
P_N^i	Initial power HPU n
P_e^t	Predicted combined heat load

P_n^{min}	Minimum power output for HPU n
P_n^{max}	Maximum power output for HPU n
P_n^r	Maximum ramp rate for HPU n
δ_n^t	Binary variable for HPU n at time t (on/off)
δ_n^i	Initial binary state of HPU n (on/off)
SU_n	Start up cost for HPU n
T_n^d	Minimum down-time for HPU n if currently stopped
T_n^u	Minimum on-time for HPU n if currently on
<i>Model predictive control</i>	
k	time step
T	Prediction horizon
y	Measured output at sample time $k = 1$
\hat{y}_k	Estimated output at sample time k
y_k^{ref}	Set point process state at time k
x_k	state vector at time k
u_k	input vector at time k
A	state matrix
B	input matrix
C	output matrix
Q	output weighting matrix
R	input weighting matrix
n	number of states
m	number of inputs
p	number of outputs

1 Introduction

District heating is a widely used technology for supplying hot water and space heating to consumers. The basic District Heating System (DHS) consists of Heat Production Units (HPU) applying energy to a closed system of circulating water. The energy is transported through the DHS to the customer substation that exchanges energy with the internal customer network, Figure 1. Early district heating networks were typically fueled by fossil fuels and supplied high temperature water or steam to the network. The last decades the focus has shifted towards renewable heat sources and maximizing operating efficiency (Lake, Rezaie, and Beyerlein 2017). In 2016 the distribution of energy sources for Norway was 50.1% waste incineration, 28.3% wood byproducts, 13.0% electricity, 5% fossil fuels and 3,6% from other sources. (Statistisk Sentralbyrå 2017).

Fortum Oslo Varme AS (FOV) owns and operates the DHS in Oslo and provides annually 1.7 TWh to over 4000 consumers produced by 36 HPUs situated at 11 different Heat Production Plants (HPP). FOV provides 36% of the national combined power by DHS, making it the largest in Norway (Fortum Oslo Varme 2018).

FOV is continuously optimizing the DHS to minimize human resources, reduce fuel cost and improve operational reliability. One aspect that can contribute to this is to fully automate HPU commitment and control. FOV executes the HPU commitment problem and control manually by human resources, and due to the system complexity, the resulting commitment plan and HPU control is occasionally not optimal.

Considering the size of the FOV's DHS and number of HPUs, taking the entire system into account when solving these problems would be a large task. Therefore this paper aims to develop a strategy for automatic HPU control and commitment of a minor subsection of the DHS. The subsection selected for this project is separated from the rest of the DHS by a heat exchanger, and can easily be considered a separated system, Figure 1.

To optimize and automate HPU control and commitment, there are several aspect that need to be taken into consideration. The UC problem is dependent on knowing the future HL, which is a variable that is customer controlled and fluctuates heavily depending on several factors such as season, temperature, wind, sunlight, time of day among others. A HL prediction is therefore necessary for achieving optimal UC results. Typically the UC problem in DHS operates with an hourly resolution and operators and simple controllers will handle any deviation from predicted value. The pitfall of this strategy is that it relies on operators executing the actual plan of the UC algorithm and making the right decision of which HPU to increase or decrease power. To eliminate this, a MPC can use the UC algorithm output to control all HPU on a shorter horizon based on the feedback from differential pressure at the substations.

2 Heat load prediction

In order to properly use predictive control and solving the UC problem, an estimate of future heat load is needed. Hagalid (2018) has as a part of his project thesis in collaboration with FOV developed a HL prediction based on multiple linear regression (MLR) for FOV's DHS. Other methods for predicting HL were not considered. A MLR model can be expressed as:

$$y = \alpha + \sum_{k=1}^n \beta_k x_k \quad (1)$$

Hagalid (2018) shows that, by selecting HL as dependent variable (y) and air temperature, month of the year as 12 categorical variables, hour of the week as 168 categorical variables, heat production 24 hours prior, air temperature multiplied by month of the year, and air temperature

multiplied by hour of the week as the independent variables (x_k), results in robust and good performing model.

Hagalid kindly provided the MATLAB scrips used to create the MLR model. With minor adjustments the script was adapted to create a MLR model for the subsection. To find the the regression coefficients, the function *fitlm* was fed historical temperature and power data, The outdoor temperature is recorded at HPP 1 and power data is the hourly mean power transferred from HPP 1 and 2 to the subsection's substations.

3 Unit commitment

For district heating systems, involving more than one HPU, the state and how much power they should produce needs to be optimized. This will typically be influenced by variable fuel cost, restrictions on up/down-time, startup costs and ramp restrictions. Mixed integer linear programming (MILP) can be used to formulate a cost function to be minimized along with a number of constraints on the solution. Several strategies for solving this can be found in (Saravanan et al. 2013). Mixed integer problems are NP-hard and an efficient solver is therefore necessary.

Yalmip(Löfberg 2004) is a free MATLAB add-on which allows easier mathematical formulation of optimization problems. Optitoolbox (Currie and Wilson 2012) add-on provides an easier interface to MILP solver SCIP (Gleixner et al. 2017).

The cost function in (2) was implemented in MATLAB 2017b through the Yalmip/Optitoolbox add-ons using SCIP as solver. Constraints were applied to the solution in order to ensure correct initial conditions (3,5), that power demand is met (6), power output is between HPU limits (7), ramp rate restrictions (8) and that up/down-time restrictions are met (9,10).

$$\min_c J = \sum_{t=1}^{T_h} \sum_{n=1}^N C_n^t P_n^t + \sum_{t=2}^{T_h} \sum_{n=1}^N \delta_n^t S U_n \quad (2)$$

$$P_n^t = P_n^{t=1} \quad \forall n \in N \quad (3)$$

$$\delta_n^t \in \{0, 1\} \quad \forall n \in N, t \in T_h \quad (4)$$

$$\delta_n^t = \delta_n^{t=1} \quad \forall n \in N \quad (5)$$

$$\sum_{n=1}^N P_n^t \geq P_e^t \quad \forall t \in T_h \quad (6)$$

$$\delta_n^t P_n^{min} \leq \delta_n^t P_n^t \leq \delta_n^t P_n^{max} \quad \forall t \in T_h, \forall n \in N \quad (7)$$

$$P_n^t \geq |P_n^t - P_n^{t-1}| \quad \forall t \in T_h, \forall n \in N \quad (8)$$

$$\delta_n^t \geq \delta_n^t - \delta_n^{t-1} \quad \forall t \in T_h, \forall n \in N \quad (9)$$

$$\delta_n^t = [\delta_n^t, \delta_n^{min(T_h, t+T_n^u-1)}]$$

$$\delta_n^d \leq 1 - \delta_n^{t-1} - \delta_n^t \quad \forall t \in T_h, \forall n \in N \quad (10)$$

$$\delta_n^d = [\delta_n^d, \delta_n^{min(T_h^h, t+T_d-1)}]$$

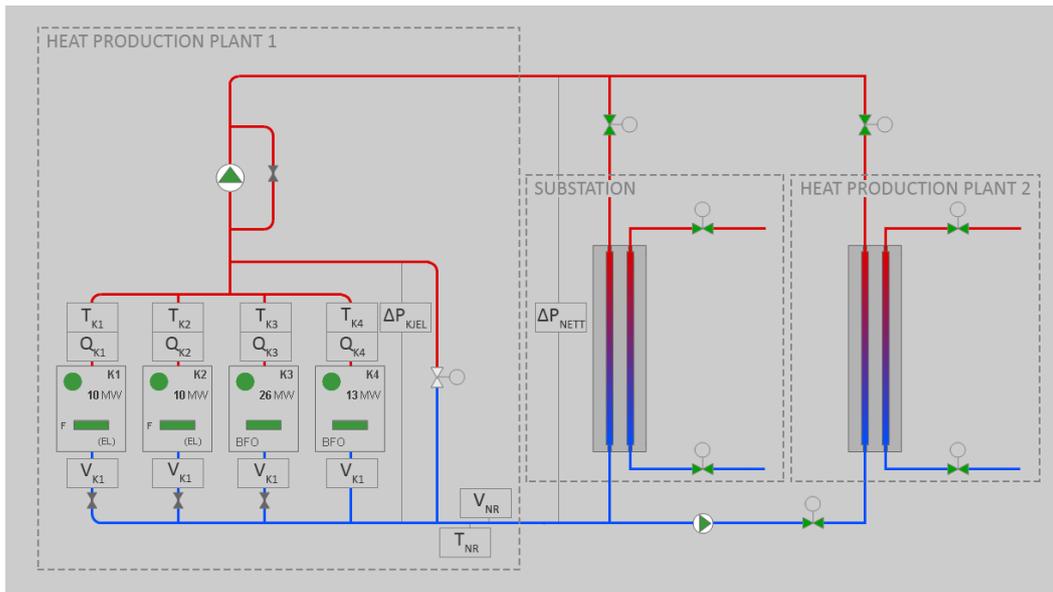


Figure 1. Operational sketch of the subsection of FOV's DHS. All substations are represented as one large. HPP 1, containing four HPU(K1-K4), is the controlled plant. The subsection is separated from the rest of the DHS at HPP 2. Although not shown here, the HPUs have an internal loop that exchanges heat with the DHS

4 System dynamics and identification

The MPC is dependent on a model for predicting future system states. The models need to describe the dynamics of each HPU in the subsection. Although the HPUs use different fuel and are structurally diverse, they are all uniformly connected to the DHS, illustrated by Figure 2. This allows us to use the same model structure for all HPUs.

The variables of importance is the supply line temperature (T_{nr}) and flow (\dot{V}_n) to the DHS, hence selected as the model output. The flow in the internal loop is constant and is therefore omitted. The HPU outlet temperature (T_{kt}) is not of interest of controlling to a given value. However MPC can provide predicted value for all outputs for the length of the simulation horizon, and T_{kt} can be useful to calculate the HPU inlet temperature (T_{kr}). Therefore T_{kt} is also selected as model output. The variables affecting the selected outputs, hence selected as inputs, are the supplied HPU power (\dot{Q}_k), T_{kr} , the return line temperature from the DHS (T_{nr}) and the reference flow value (\dot{V}_{ref}) for the DHS flow via the flow controlled valve.

There are multiple model types that fulfills our requirements. We have chosen to model our system as a process model transfer function (PMTF), defined as $H(s) = \frac{ke^{-ts}}{(\tau_1s+1)(\tau_2s+1)\dots(\tau_ns+1)}$ for n th order model with system gain k , time delay t and time constants τ_n . The resulting model structure for a HPU gives a multi-variable PMTF matrix is described in (11)

$$\begin{bmatrix} T_{kt}(s) \\ T_{nr}(s) \\ \dot{V}_n(s) \end{bmatrix} = \begin{bmatrix} H_{11}(s) & H_{12}(s) & 0 & 0 \\ H_{21}(s) & H_{22}(s) & H_{23}(s) & H_{24}(s) \\ 0 & 0 & 0 & H_{34}(s) \end{bmatrix} \begin{bmatrix} \dot{Q}_k(s) \\ T_{kr}(s) \\ T_{nr}(s) \\ \dot{V}_{ref}(s) \end{bmatrix} \quad (11)$$

MATLAB System Identification Toolbox was used for model identification. Data with 2 second sample time was selected from the period September 2016 to April 2017. Emphasis was put on finding periods with varied operation where the HPU parameters showed sufficient dynamics. The HPUs have different dynamics during start-up and shut-down, due to by pass valves are open. These periods were avoided as estimation and validation data. Low order process models were sought, typically first order.

5 Model predictive control

MPC is a widely used concept. The basic principles involve calculating future response of outputs from a set of inputs based on a dynamic model of the plant. Both inputs and outputs are subject to constraints such as value and ramp rates. Measured or estimated disturbances can be incorporated in the system. MPC has the benefit of being able to keep inputs and outputs within specified operating ranges while keeping input variation to a minimum.

There are numerous ways to implement MPC, but it is most commonly applied as a linear quadratic cost optimization problem. Given a system described by the state space model $\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k$, a quadratic cost optimization problem can be formulated as

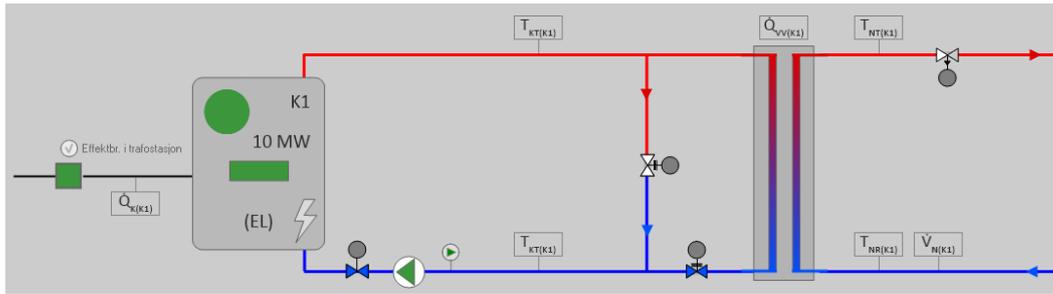


Figure 2. A operational sketch for HPU K1. The figure shows the internal loop of the HPU that exchanges heat to the DHS

$$\min_{\mathbf{u}, \Delta \mathbf{y}} J = \sum_{k=1}^{T_h} \Delta \mathbf{y}_k^T \mathbf{Q} \Delta \mathbf{y}_k + \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k \quad (12)$$

$$\begin{aligned} s.t. \quad & \mathbf{x}_{k+1} = \mathbf{A} \mathbf{x}_k + \mathbf{B} \mathbf{u}_k \\ & \hat{\mathbf{y}}_k = \mathbf{C} \mathbf{x}_k \\ & \mathbf{x}_{min} \leq \mathbf{x} \leq \mathbf{x}_{max} \\ & \mathbf{u}_{min} \leq \mathbf{u} \leq \mathbf{u}_{max} \\ & \Delta \mathbf{u}_{min} \leq \mathbf{u}_k - \mathbf{u}_{k-1} \leq \Delta \mathbf{u}_{max} \end{aligned} \quad \forall k \in [1, T_h]$$

$$\begin{aligned} for \quad & \Delta \mathbf{y}_k = \mathbf{y}_k^{ref} - \hat{\mathbf{y}}_k \\ & \hat{\mathbf{y}}_{k=1} = \mathbf{y} \\ & \mathbf{y} \in \mathbb{R}^p, \quad \mathbf{x} \in \mathbb{R}^n \\ & \mathbf{u} \in \mathbb{R}^m, \mathbf{A} \in \mathbb{R}^{n \times n} \\ & \mathbf{B} \in \mathbb{R}^{n \times m}, \quad \mathbf{C} \in \mathbb{R}^{p \times n} \end{aligned}$$

The model for one HPU consist of four inputs, however only \dot{Q}_k and \dot{V}_{ref} are controllable and set as manipulated variables (MV). T_{kr} and T_{nr} is set as measured disturbance (MD). A measured disturbance is often set to the present measured value for the whole simulation horizon. For T_{nt} this is sufficient, as the temperature is stable with little variation. T_{kr} on the other hand fluctuates heavily, which might reduce the MPC performance. To compensate for this, as mentioned in section 4, future T_{kr} values can be calculated using the predicted values T_{kt}, T_{nt} , the MD T_{nr} , the energy equation $\dot{Q} = c_p \dot{V} \Delta T$ and the assumption $\dot{Q}_{HEin} = \dot{Q}_{HEout}$ as shown in (13)

$$T_{kr} = -\frac{\dot{V}_n (T_{nt} - T_{nr})}{\dot{V}_{int}} + T_{kt} \quad (13)$$

The HPUs are parallel to each other, meaning the total flow $\dot{V}_T = \dot{V}_{K1} + \dot{V}_{K2} + \dot{V}_{K3} + \dot{V}_{K4}$. By combining the individual HPU flow outputs the resulting system model is a 16 by 9 PMTF matrix.

The MPC controller was made using MATLAB *mpc*-function and simulated on the internal plant using *mpc-move*. Weights for the measured variables were tuned to allow considerable slack in T_{kt} while keeping T_{nt} and $dotV$ as close to setpoint as possible. In order to handle the continuously fluctuating flow and at the same time foresee the largest time constants in the system, the sample time is set to 20s. With the simulation and control horizon set to respectively 100 and 50 samples, the MPC foresee 2000 seconds, about 33 minutes.

6 System control strategy

The simulation horizon of 33 minutes is sufficient to control the HPUs, but too short to make decision regarding UC, hence a control hierarchy with differing time ranges is proposed. The prediction algorithm will run hourly and return predicted total power demand. The UC algorithm uses the predicted HL as input and returns operational state of each HPU to the model predictive controller. The UC algorithm can be run every 10-60 minutes or called when necessary from MPC. To integrate the UC and MPC algorithm, a function is needed, illustrated by Figure 3.

The function calculates $Ref_{\dot{V}}$ for the length of the simulation horizon using \dot{Q}_{pred} and $\dot{V} = \frac{\dot{Q}_{pred}}{c_p (T_{nt} - T_{nr})}$, knowing the present and predicted values of T_{nt} and T_{nr} . The calculated $Ref_{\dot{V}}$ value might deviate from real time measurements of \dot{V} . The function creates a stipulated trajectory by regression towards \dot{V}_{ref} . The function also updates the MV cost and constraints to force the MPC to follow the UC algorithm plan.

The MPC is dependent on feedback to correct any deviation in predicted HL. There are no direct real time measurement of the HL available, but the differential pressure over the substations can work as a substitute. The customer controlled valve in the substations will open and increase the flow through the substation HE if the customer require more power. If all substations combined open their valves, and the total flow and temperature in the DHS are static, the differential pressure will drop. To compensate for this, one can either change the supply line temperature or the flow. Changing the supply line temperature implies large transport delays up to several hours, as the water flow average speed is 2-5m/s. Changing the flow leads to a quicker response as the pressure wave in the pipes travels with speed up to 1200 m/s. To counter the deviations in the predicted HL, the total flow reference for HPP 1 can be adjusted by integrating the deviation from a set value for the substations differential pressure. The MPC will decide which of the enabled HPUs to increase power and flow. Consequently, the UC algorithm is not controlling the HPU power is directly, but decides which HPU to be active, start up or shut down.

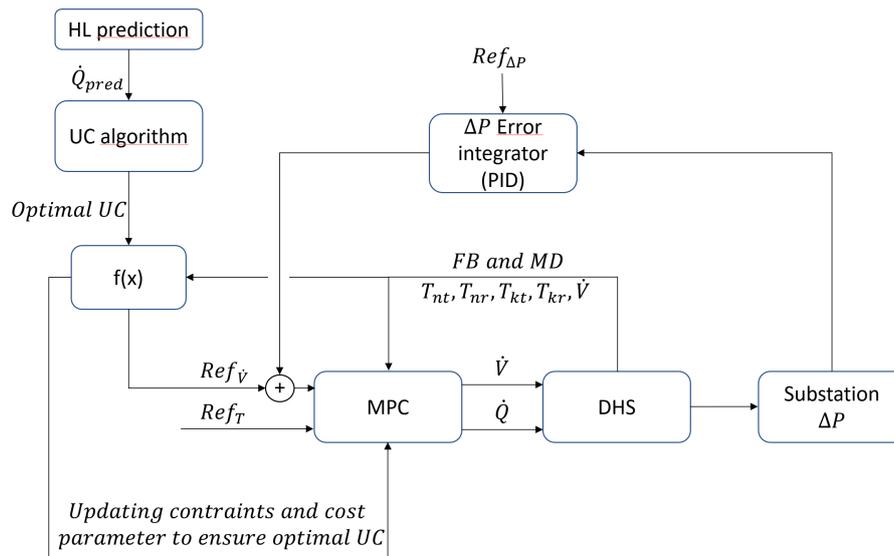


Figure 3. Figure show proposed system control hierarchy

7 Results

7.1 Heat load prediction

The HL predictions were made for three separate periods and compared to actual historical data. Plots of predicted values and actual data for the period 12-16. Jan 2016 is shown in Figure 4. Errors were calculated using *Root Mean Square Error (RMSE)*, *Mean Average Percent Error (MAPE)* and *Mean Error (ME)*, shown in Table 1.

From the error data from June 2017-February 2018 the mean error is negligible, this indicates that on average the model performs well. For shorter periods(16-20 Jan 2018) a significant bias is seen which might impair predictive control of the network. The MAPE and RMSE values are quite high but it is important to note that very short term deviations without bias can be tolerated in a district heating system. The short time deviations might not be model errors, but actually caused by variations in the operating mode of the HPU's so that the training data is flawed. A better model could probably be made by using consumer energy use instead of produced energy as inputs to the model.

Table 1. Errors for heat load prediction

Period	RMSE [MW]	MAPE [%]	ME [MW]
Jun 17-Feb 18	1.5	7.1	0.04
12-16 Jul 17	0.8	9.5	-0.09
16-20 Jan 18	1.7	4.9	-0.8

7.2 Unit commitment

The unit commitment algorithm was tested against actual data from 20-24 Jan 2016 using the method described in section 3. Real, time dependent fuel prices were used.

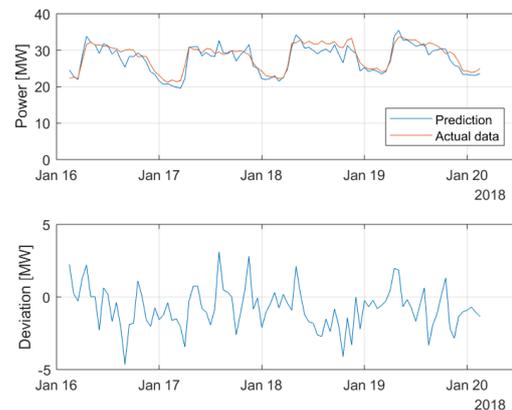


Figure 4. Predicted and actual power output for the period between 16-20 January 2018. Absolute deviation from actual data is shown in the lower part of the plot.

Since no data for consumer power usage were available power demand was calculated by summing the power output of all HPU's from historical data. The algorithm was also tested with the same numerical results, but slightly slower processing time using CBC (Lougee-Heimer 2003) and MATLAB's *intlinprog*-solver.

Power output of each HPU from the algorithm and actual historical data are shown in Figure 5. Fuel prices are shown in the same figure for reference.

Fuel costs were calculated for actual data and the proposed solution. A potential fuel cost saving of 20.5% was seen for this period. However this was a known problematic period, similar savings are not to be expected routinely. The UC algorithm is powerful and returns good results, however this subsection of the district heating network is quite simple; only two types of HPU's of which none have difficult operating constraints nor any signifi-

Table 2. Parameters for unit commitment optimization

HPU (<i>n</i>)	K1	K2	K3	K4
P_n^{max} [MW]	10	10	26	13
P_n^{min} [MW]	0.5	0.5	0.5	0.5
P_n^r [MW/hr]	48	48	60	60
T_n^u [hr]	1	1	1	1
T_n^d [hr]	1	1	1	1
SU [NOK]	200	200	500	500

cant start up costs. It might seem that the UC algorithm is overcomplicated for this use, but is chosen keeping in mind that it can be extended to handle UC for the entire DHS.

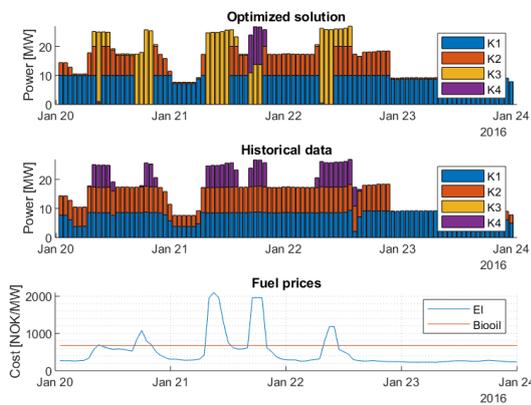


Figure 5. Top: Proposed solution from algorithm for the period 20-24 Jan 2016. Mid: Actual power output from historical data for the same period. Bottom: Actual fuel prices from the same period.

7.3 System identification

All four HPU loops were modeled separately as described in section 4, for brevity complete data will just be presented for HPU K1. Data for estimation was collected from the period 8-11 Feb 2017 and the resulting model was validated against data from 12-16 Jan 2016. The Normalised Root Mean Square Error (NRMSE) of 0.932 for T_{kt} and 0.787 for T_{nt} indicates a good fit between identified model and actual HPU dynamics. Plots of simulated response from model and validation data are shown in Figure 6 and 7 for respectively T_{kt} and T_{nt} .

All models were process models as shown in (11), maximum model order was set to 2. Parametrization of all non-zero transfer functions for HPU 1 is shown in Table 3.

Table 4 shows a summary of NRMSE fits from system identification of all four HPUs. The electrical HPU (K1-2) yielded a good fit between model and actual data but larger deviations was seen for the bio oil HPUs (K3-4). The bio oil HPUs are rarely in use, and when in operation they are set at a static power level yielding poorer dynamic

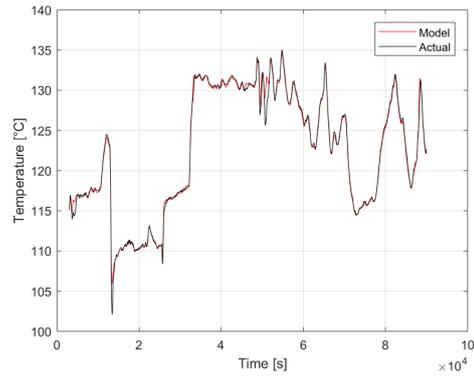


Figure 6. Plot of simulated response from identified model and actual data for T_{kt} . NRMSE: 0.932.

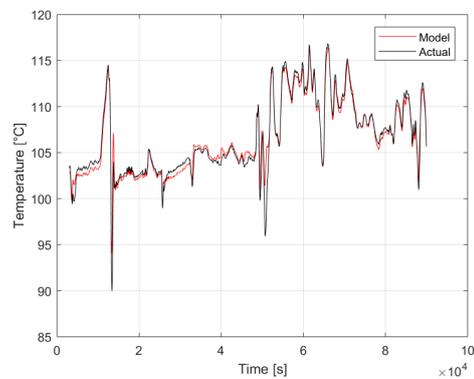


Figure 7. Plot of simulated response from identified model and actual data for T_{nt} . NRMSE: 0.787.

Table 3. Parameters for process model matrix

Parameter	Transfer function
$H_{11}(s)$	$\frac{3.56}{1+122s} e^{-5.2s}$
$H_{12}(s)$	$\frac{0.996}{1+122s} e^{-0.7s}$
$H_{21}(s)$	$\frac{1.464}{(1+129s)(1+0.017s)} e^{-16.8s}$
$H_{22}(s)$	$\frac{1.042}{1+93s}$
$H_{23}(s)$	$\frac{0.360}{1+6.5 \cdot 10^7 s} e^{-0.5s}$
$H_{24}(s)$	$\frac{-0.046}{(1+22s)(1+0.038s)} e^{-24s}$
$H_{34}(s)$	$\frac{1}{1+22.6s}$

quality of the estimation data. There might also be a larger nonlinear dynamic due to combustion process which our models does not represent properly. Further investigation into this should be done.

Table 4. NRMSE fits for all identified HPU models.

HPU	NRMSE [%]	
	T_{kt}	T_{nt}
K1	93.2	78.7
K2	94.4	80.8
K3	64.4	-9.6
K4	36.0	45.5

7.4 Model predictive control

Figure 8 and 9 show the MPC performance compared to the existing PID cascade. The MPC use the identified model presented in section 7.3 as internal and external plant. The simulation use the historic T_{nr} as MD. The historic \dot{V} is set as $Ref_{\dot{V}}$. The simulation shows that MPC better able to keep T_{nt} to the reference temperature. Figure 9 shows outputs and disturbances for HPU K1 in the upper section and controlled variables \dot{V} and \dot{Q} for HPU K1 in the bottom plot.

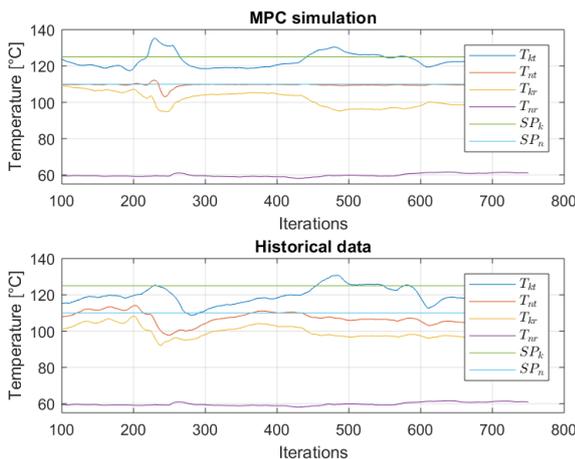


Figure 8. Comparison of closed loop control performance between MPC simulation and present PID cascade of K1 in respect to temperature.

8 Discussion

The load prediction algorithm worked quite well. Significant deviations from actual values are seen but averaged over longer periods the deviation is canceled. It is believed that a better prediction could be made if power used by consumers was used instead of power produced at the plants. Power production data may be influenced by the operators and may not correspond to actual consumer demand. There are also other methods using neural network that look promising, as shown by (Verrilli et al. 2017).

The unit commitment algorithm works well but may not be necessary for this system seeing as the HPUs have fast ramp rates and few restrictions. A combined optimization/MPC-scheme would likely be the best solu-

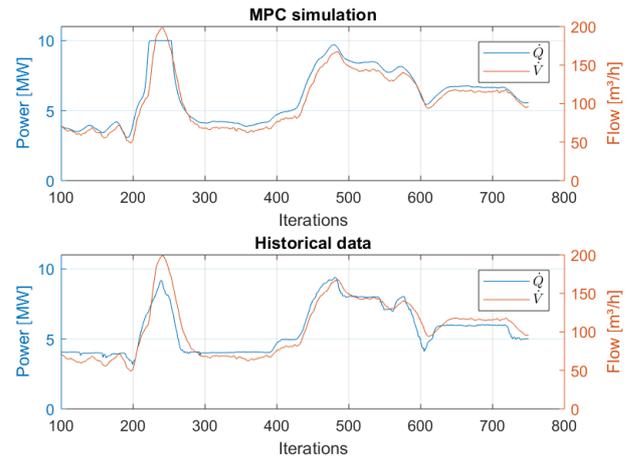


Figure 9. Comparison of closed loop control performance between MPC simulation and present PID cascade of K1 in respect to \dot{Q} and \dot{V} .

tion, but would require writing a custom non-linear MPC algorithm.

Good correlation between model and plant was seen for the electrical HPUs. For the bio oil HPUs somewhat poor correlation was seen. This is mainly caused by lack of, and less dynamic estimation data which is detrimental for model estimation. There might also be differences in operation or some nonlinear dynamics in the combustion process which is not reflected in the estimation variables.

The MPC simulation shows that the temperature reference tracking may be improved by implementing MPC to control the HPU. However, the simulations was carried out using the identified models as internal and external plant, meaning there is 100% fit between the MPC's predicted response and the simulated system response. The MPC's actual performance is expected to be poorer.

Simulating the complete control hierarchy was difficult for a number of reasons. Currently the closed loop feedback used to indicate power demand is differential pressure across select customer heat exchangers. This is controlled by the customers and can therefore not easily be simulated without modeling the entire DHS. The relation between differential pressure and power demand is also not very clear. A simulation of the complete control hierarchy has therefore not yet been performed.

Further work is needed to complete the project, and our suggestions is to develop a nonlinear model of the HPUs and the DHS, which allows simulation of the control hierarchy and provides a more realistic MPC results.

9 Conclusion

Methods for load prediction and unit commitment performed quite well and are thought to be useful in a real implementation of the control hierarchy. The model predictive controller is likely to improve temperature reference tracking. The complete control hierarchy has not

been tested and is necessary to complete the project.

10 Acknowledgements

The authors would like to thank Marius Hagalid for supplying the load prediction model and helping with our implementation. Karine Huuse, Ingeborg Sauge Torpe and Patrik Myrendal at Fortum Oslo Varme have aided our understanding of district heating systems in general and provided specific data necessary to complete the project.

References

- Currie, Jonathan and David I. Wilson (2012). “OPTI: Lowering the Barrier Between Open Source Optimizers and the Industrial MATLAB User”. In: *Foundations of Computer-Aided Process Operations*. Ed. by Nick Sahinidis and Jose Pinto. Savannah, Georgia, USA.
- Fortum Oslo Varme (2018). *Fjernvarme i Oslo*. Retrieved 15.04.2018. URL: <https://www.fortum.no/fjernvarme-i-oslo>.
- Gleixner, Ambros et al. (2017). *The SCIP Optimization Suite 5.0*. eng. Tech. rep. 17-61. Takustr.7, 14195 Berlin: ZIB.
- Hagalid, Marius Jean Bischof (2018). “Development of Models for Automatic Control of the Facilities of Fortum Oslo Varme in Regard to Reliability and Optimal Operation”. In: *Unpublished preliminary report*.
- Lake, Andrew, Behnaz Rezaie, and Steven Beyerlein (2017). “Review of district heating and cooling systems for a sustainable future”. In: *Renewable and Sustainable Energy Reviews* 67, pp. 417–425. ISSN: 1364-0321. DOI: <https://doi.org/10.1016/j.rser.2016.09.061>. URL: <http://www.sciencedirect.com/science/article/pii/S1364032116305585>.
- Löfberg, J. (2004). “YALMIP : A Toolbox for Modeling and Optimization in MATLAB”. In: *In Proceedings of the CACSD Conference*. Taipei, Taiwan.
- Lougee-Heimer, Robin (2003). “The Common Optimization INterface for Operations Research”. English. In: *IBM Journal of Research and Development* 47(1), pp. 57–66.
- Saravanan, B. et al. (2013). “A solution to the unit commitment problem—a review”. In: *Frontiers in Energy* 7.2, pp. 223–236. ISSN: 2095-1698. DOI: [10.1007/s11708-013-0240-3](https://doi.org/10.1007/s11708-013-0240-3). URL: <https://doi.org/10.1007/s11708-013-0240-3>.
- Statistisk Sentralbyrå (2017). *Fjernvarme og fjernkjøling i Norge*. Retrieved 17.01.2018. URL: <https://www.ssb.no/fjernvarme>.
- Verrilli, F. et al. (2017). “Model predictive control-based optimal operations of district heating system with thermal energy storage and flexible loads”. English. In: *IEEE Trans. Autom. Sci. Eng. (USA)* 14.2, pp. 547–57. ISSN: 1545-5955. URL: <http://dx.doi.org/10.1109/TASE.2016.2618948>.

Process Design of a Novel Low Temperature Methanol Synthesis Process Using an Air-blown Autothermal Reformer

Christian Ahoba-Sam, Lars Erik Øi, Klaus-Joachim Jens

Department of and Process, Energy and Environmental Technology, University College of Southeast Norway,
Norway
lars.oi@usn.no

Abstract

Methanol (MeOH) synthesis at low temperature (100 °C) presents an opportunity for full syngas conversion per pass. This presents a cheaper alternative for MeOH synthesis using an air-blown autothermal reformer (ATR) rather than the conventional high temperature (>250 °C) MeOH synthesis approach which requires an expensive cryogenic O₂-blown ATR. The aim of this work was to use the process simulation program Aspen HYSYS to simulate and optimize the reactor conditions for a complete MeOH process design using an air-blown ATR. Our results revealed that, while syngas produced from 'normal' air-blown ATR (syngas composition 0.20CO:0.40H₂:0.39N₂) required 100 bar to obtain full conversion per pass, syngas produced from enriched air-blown ATR (syngas composition 0.31CO:0.62H₂:0.07 N₂) required 60 bar total syngas pressure to achieve the same. Even though the energy generated in both processes was enough to cover the heating demand in the total process with surplus, the enriched air-blown system provides a better energy recovery if the surplus energy is not used for extra power generation. The total process energy demand due to compression was estimated to be 2270 and 983 MJ/ton MeOH product for the normal air-blown and enriched air-blown systems respectively. A process design was proposed based on the optimized conditions for the enriched air-blown process.

Keywords: Air-blown reformer, syngas, methanol, low temperature, simulation, Aspen HYSYS

1 Introduction

Methanol (MeOH) is a multi-purpose molecule widely used as a base chemical, and for storage of energy and CO₂ (Olah, 2005). MeOH can be used as a fuel blend or directly converted to valuable hydrocarbons such as gasoline over acidic microporous materials (Olsbye et al., 2012), thereby providing an alternative source of petrochemical feedstock used today.

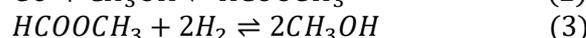
The current technology for MeOH synthesis is based on conversion of syngas and operates around 250-300 °C and 50-100 bar (Hansen & Højlund Nielsen, 2008). While this technology is highly optimized, the relatively high operating temperature limits conversion to barely 20 % per pass due to thermodynamics of the

exothermic MeOH synthesis reaction (Equation (1)). As a result, the current process requires several recycling steps to optimize production (Lange, 2001).

Furthermore, the low conversion per pass of this process requires the use of a rather 'pure' syngas other than a N₂-diluted syngas for the MeOH synthesis. Such pure syngas production alone accounts for more than half of the total operation and capital cost in current methanol processes (Marchionna et al., 1998). This is due to the use of expensive cryogenic O₂ for partial oxidation of hydrocarbons. The lowest cost of syngas production is by the use of either an ordinary air or an O₂ enriched air rather than a cryogenic O₂-blown autothermal reformer (ATR) (Hansen & Højlund Nielsen, 2008). The highly exothermic syngas conversion to MeOH requires a relatively low temperature to achieve a full conversion per pass. Hence, such a low temperature process will allow for the use of N₂-diluted syngas for MeOH production, with reduced capital and operation cost as there will be no need for recycling.



Alternatively, a low temperature MeOH synthesis (LTMS) process which proceeds rapidly in a liquid medium at about 100 °C presents the possibility for full syngas conversion per pass (Christiansen, 1919). The LTMS process is known to occur in two steps (Equations (2) and (3)), via a methyl formate intermediate (Ohyama & Kishida, 1998). Typically alkali alkoxide is known to catalyse the carbonylation step (Equation 1) and Cu based materials catalyse the hydrogenolysis step (Equation 2). Cu nanoparticles due to largely exposed surface area accelerate the hydrogenolysis step in the LTMS process (Ahoba-Sam, Boodhoo, et al., 2018). For example, in diglyme solvent, Cu nanoparticles in combination with sodium methoxide led to up to 92 % conversion per batch with 20 bar syngas composed of 0.33CO:0.67H₂, at 100 °C (Ahoba-Sam, Olsbye, et al., 2018).



In this paper, our focus was to design a complete process for the LTMS reaction using Aspen HYSYS simulation

program. Considering that the LTMS process can tolerate N₂ diluent, the use of ordinary air and other O₂ enriched air can be simulated for the syngas production, and subsequently be used to design a complete MeOH plant. Even though the concept has been described as a ‘dream reaction’ (Hansen & Højlund Nielsen, 2008), the current work demonstrates that the suggested concept is feasible. The specific aim of this work was to simulate and optimize the reactor conditions in order to propose a complete design of the LTMS process using an air-blown ATR. Different parameters such as chemical compositions, temperature and pressure in the reactors were varied to optimize the process parameters.

2 Process Description

2.1 Principles of the Syngas Production

The syngas production process involved partial oxidation of methane (CH₄) using air, illustrated in Equation (4). The Figure 1 shows a syngas production process using air as the source of O₂. The process consists of a reactor (ATR), compressor and heat exchangers. Compressed air was used to make-up for the stoichiometry between CH₄ to O₂. The CH₄ feed used was assumed to be pure, without any sulphur or heavier hydrocarbon present, while the air feed consisted of only O₂ and N₂ to simplify the simulation. The compressed air together with the CH₄ feed were pre-heated and fed directly into the reactor.

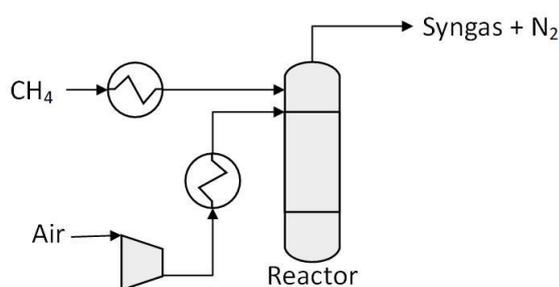
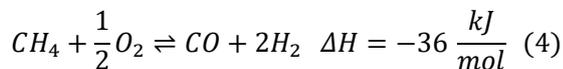


Figure 1. Principle for the syngas production process

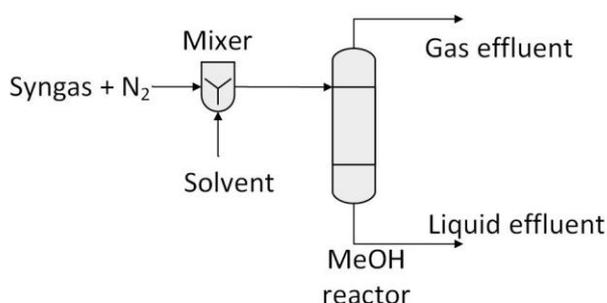


Figure 2. Principle for the low temperature MeOH synthesis process

2.2 Principles of the MeOH Synthesis Process

Figure 2 shows the MeOH synthesis process using N₂ containing syngas as feed. The LTMS process involved a MeOH reactor and a mixing unit. Since this step is highly exothermic (see Equation (1)) and requires lower operating temperature, the process is often carried out in liquid medium to absorb excess heat to minimize adiabatic rise in temperature beside other solvent’s polarity role for the catalysis (Ahoba-Sam et. al., 2017). The syngas feed is mixed with a liquid solvent and fed into the MeOH reactor. For simplicity of the model, we have taken MeOH as a solvent and due to the 100 % thermodynamic syngas conversion, we have neglected the effect of MeOH solvent in the equilibrium calculations.

3 Models

All the simulations were performed using the Aspen HYSYS (version 8.6) program. Gibbs reactors were employed for both the syngas production and LTMS processes. A Gibbs reactor calculates the composition with the theoretical free energy minimum, which is the theoretical equilibrium composition. The Peng-Robinson equation of state (Peng & Robinson, 1976) fluid package was used in all the simulations. The equilibrium for the partial oxidation of CH₄ was calculated for the process in the ATR reactor, while CH₄ was assumed to be inert in the MeOH reactor. After the individual reactors were optimized, an overall system was designed.

Separating units were added in the overall system to represent, (i) a pressure swing adsorber (PSA) and (ii) H₂O/CO₂ absorber. The PSA was included to help regulate the O₂/N₂ composition that is fed into the ATR. The H₂O/CO₂ absorber was used to separate the H₂O and CO₂ from the syngas effluent before the MeOH reactor. After establishing reasonable pressure and temperature conditions in the individual reactors, the total compression, and heating requirements were simulated, to find out whether additional heating was necessary for the total process.

4 Process Simulations

4.1 Simulation and Optimization of the ATR for Syngas Production

To optimize the feed composition, different mole fractions of the CH₄: air (containing O₂ and N₂) were fed into the ATR reactor. The O₂/N₂ ratio was kept constant at air composition of 21/79. The Figure 3 shows the effect of the ratio of CH₄/O₂ on the syngas produced at 600 °C. The highest amount of CO + H₂ coupled with the least H₂O and CO₂ side product (3 and 1 % respectively) was observed at CH₄/O₂=2. While H₂O and CO₂ side products increased below the CH₄/O₂=2

ratio due to increase in oxidation, lower CH₄ conversion was observed at higher CH₄/O₂ ratio as the amount of O₂ became limiting. When the ratio of CH₄/O₂ was kept constant and O₂/N₂ ratios were varied (not shown), no variation was observed in the composition of the products. Overall CH₄/O₂=2 was chosen as a reasonable composition as this gave the highest amount of syngas (CO+H₂) with H₂/CO=2.

The temperature was varied to determine a reasonable temperature required for the feed inlet. Figure 4 shows the effect of temperature on the syngas produced between 600 to 1600 °C. This was done at 0.30CH₄:0.15O₂:0.55N₂ feed composition set to 20 bar. Generally, the overall CH₄ conversion increased while side products decreased with increasing temperature. After 1200 °C, subtle changes were observed in the main products such that both the amount of syngas and H₂/CO=2 were similar. The amount of H₂O and CO₂ side product at 1200 °C decreased from 0.36 and 0.06 % to 0.06 and 0.01 % respectively at 1600 °C. Nevertheless, considering the exothermic nature of the process and its significance on the reactor material, 1200 °C was the temperature of choice for the syngas production.

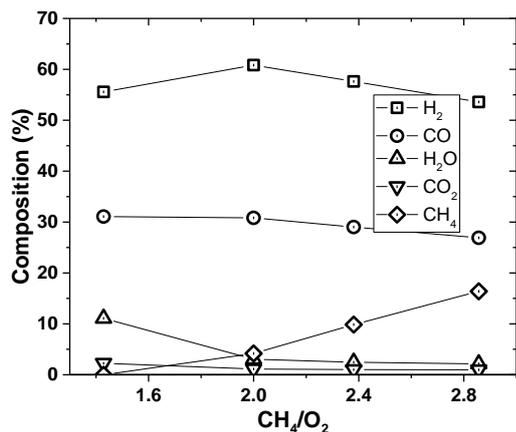


Figure 3. Effect of feed composition on the syngas production, at 600 °C and 10 bar.

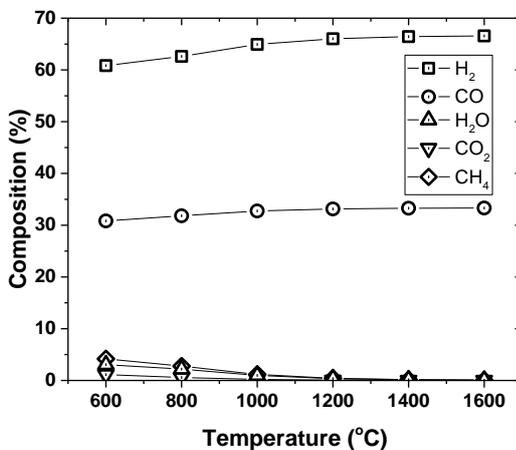


Figure 4. Effect of temperature on the syngas production, 20 bar

Furthermore, the total inlet pressure was varied to determine a reasonable pressure required for the optimal feed inlet. Figure 5 shows the effect of pressure on the syngas produced between 10 to 100 bar. This was done using similar syngas composition as was done for the temperature (0.30CH₄:0.15O₂:0.55N₂) at 1200 °C. The overall CH₄ conversion increased while side products decreased with decreasing pressure. After 30 bar, no significant changes were observed as both the amount of syngas produced and H₂/CO=2 remained the same. Therefore 20 bar was a reasonable pressure of choice for the syngas production.

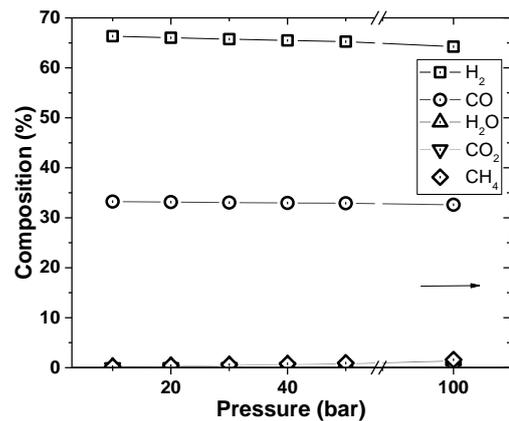


Figure 5. Effect of pressure on the syngas production

4.2 Simulation and Optimization of the MeOH Synthesis Process

The operating conditions for MeOH production was simulated to optimize the LTMS process. Figure 6 shows the effect of temperature on conversion at 20 and 100 bar syngas pressure. This was calculated using syngas ratio of 0.21 H₂ : 0.41 CO : 0.39 N₂. The syngas conversion rose exponentially from 300 °C to 120 °C and then increased slightly with decreasing temperature. The optimum temperature however depends on the operating pressure as the 20 and 100 bar syngas pressures showed similar trend but different conversions.

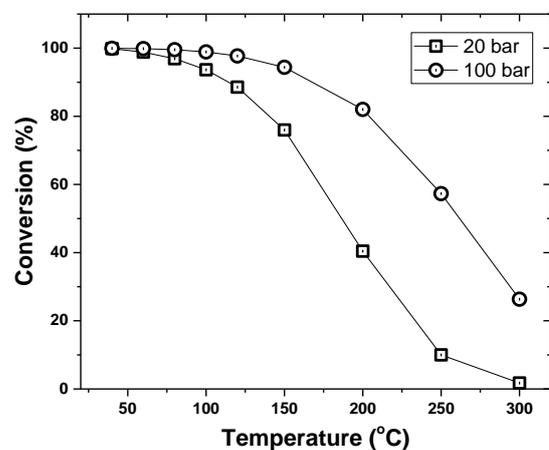


Figure 6. Effect of temperature on the LTMS process

Figure 7 shows the effect of pressure with and without N₂ at 100 °C. The conversion with N₂ was calculated using 0.21H₂:0.41CO:0.39 N₂ syngas ratio while the conversion without N₂ was calculated based on 0.67H₂:0.33CO composition. The syngas without N₂ showed more than 99 % conversion from 5 to 100 bar total syngas pressure. The syngas with N₂ however exponentially increased with pressure such that about 99 % conversion was achieved at 100 bars. This indicated the importance of N₂ diluent on the partial pressures of the syngas composition required for optimal conversion. Nevertheless, a reasonable pressure chosen for achieving optimal conversion in the presence of 39 % N₂ syngas diluent was 100 bar.

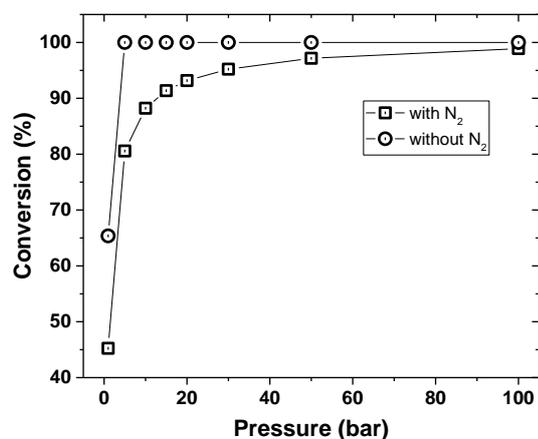


Figure 7. Effect of pressure on the LTMS process

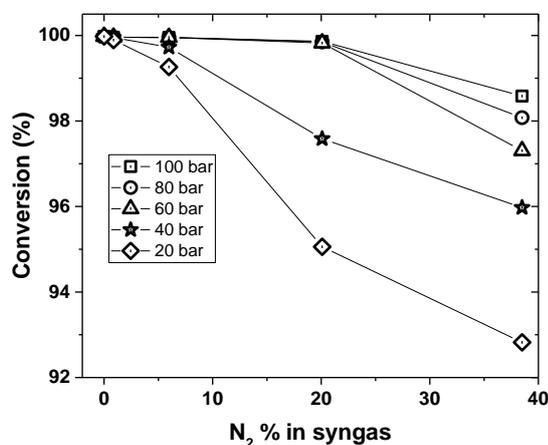


Figure 8. Effect of N₂ diluent concentration on LTMS process at different pressures

Furthermore, the importance of N₂ diluent was determined by varying its composition in the syngas. Figure 8 shows the effect of N₂ composition in the MeOH synthesis at 100 °C at different syngas total

pressures. Syngas conversion increased with decreasing the amount of N₂ diluent in the syngas. Interestingly, a slight decrease of the amount of N₂ in the syngas from 39 % (from normal air composition) to 20 %, leads to full conversion even at 60 to 100 bar. Further N₂ reduction below 7 % in syngas will thermodynamically allow more than 99 % conversion at 100 °C and 20-100 bar. PSA for example can easily be used to enrich air up to 90 % O₂ content in air (Rao & Muller, 2007). Hence for the enriched air, 7 % N₂ in syngas, which can achieve full conversion per pass at 60 bar was chosen for the LTMS process.

4.3 Simulation of the Overall LTMS Process

The optimized operation conditions for the two reactors were put together as an overall LTMS process. Two scenarios were considered; one involving ‘normal’ air-blown ATR (0.21 O₂:0.79 N₂) and the other involving an O₂ enriched air-blown-blown (0.70 O₂:0.30 N₂) system. Figure 9 shows the Aspen HYSYS flow-sheet for the overall standard LTMS process for an enriched air-blown ATR. The set-up in the Figure 9 differs from the normal air-blown system by the inclusion of a PSA unit for enriching the air.

The details of the selected operating conditions used for the calculation is tabulated in Table 1. The partial oxidation was carried out at 1200 °C and effluent from the ATR cooled down to 30 °C in both systems. The 20 bar N₂ containing syngas produced was compressed to either 60 or 100 bar in the compressor where adiabatic efficiencies were specified to 75 %. Starting with 3990 kmol/h CH₄ flow, the normal air-blown system yielded 3842 kmol/h MeOH at 100 bar syngas pressure while that with the enriched air-blown yielded 3919 kmol/h MeOH at 60 bar.

The heat/energy flow for the two systems is shown in Table 2. The negative signs in the table represent heating demands, while the positive represented heat release. For the normal air-blown system, a surplus (after recovery) of 7.68x10⁸ kJ/h heat was released as calculated from the heating and cooling. The total energy demand for the compressors was estimated to be 2.80x10⁸ kJ/h or 2270 MJ/ton MeOH product for the air-blown system. For the enriched air-blown system, a surplus (after recovery) of 6.10x10⁸ kJ/h heat was released when the heating and cooling streams were considered. The energy demand due to compression was estimated to be 1.23x10⁸ kJ/h or 983 MJ/ton MeOH product required for the enriched air-blown system. Overall the heat demand is covered by the surplus heat in both air-blown ATR systems.

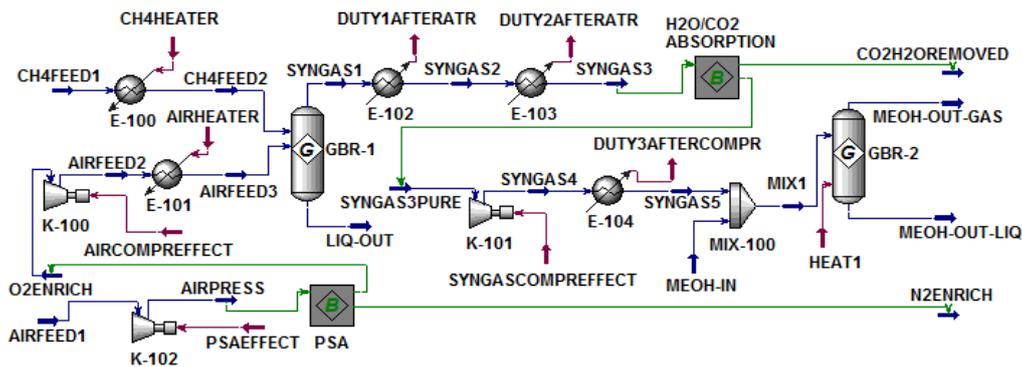


Figure 9. Aspen HYSYS flow-sheet for the overall standard LTMS process

Table 1. Specifications of Operation Conditions for the Overall LTMS Process as used in Figure 9

Parameter	Air-blown ATR system				Enriched air-blown ATR system			
	Molar flow (kmol/h)	Pressure (bar)	Temperature (°C)	Main composition	Molar flow (kmol/h)	Pressure (bar)	Temperature (°C)	Main Composition
CH ₄ Feed	3990	20	20	1 CH ₄	3990	20	20	1 CH ₄
CH ₄ Feed2	1200	1200	..
AirFeed	9500	1.01	20	0.21 O ₂ : 0.79 N ₂	14250	1.01	20	0.21 O ₂ :0.79 N ₂
AirPress	-	-	-	-	..	1.50	66.5	..
O ₂ Enrich	-	-	-	-	2850	1.10	67	0.70 O ₂ :0.30 N ₂
AirFeed2	9500	20	515	0.21 O ₂ : 0.79 N ₂	..	20	595	..
AirFeed3	1200	1200	..
Syngas1	19375	..	1356	0.20 CO : 0.40 H ₂ : 0.39 N ₂	12727	..	1439	0.31 CO:0.62 H ₂ :0.07 N ₂
Syngas2	652	687	..
Syngas3	30	30	..
Syngas3pure	-	-	-	-	12678	20	33.5	..
Syngas4	19375	100	265	0.20CO:0.40H ₂ :0.39N ₂	..	60	186	..
Syngas5	30	60	100	..
Mix1
MeOH out (Liquid+Gas)	11642	..	100	0.33CH ₃ OH: 0.64N ₂	4839	0.81CH ₃ OH: 0.18 N ₂

Table 2. Heat/Energy flow for the overall LTMS process in reference to Figure 9

Heating	Air-blown (10 ⁶ kJ/h)	Enriched-air (10 ⁶ kJ/h)
CH ₄ HEATER (-)	306	306
AIRHEATER (-)	219	60.2
DUTY1AFTERATR (+)	451	315
DUTY2AFTERATR (+)	366	252
DUTY3AFTERCOMPR (+)	141	32.2
HEAT1 (+)	335	377
Compression		
PSAEFFECT	-	19.4
AIRCOMPREFEFFECT	144	47.6
SYNGASCOMPREFEFFECT	135	56.4

5 Discussion of the Process

The air-blown ATR process for LTMS technology differ from conventional MeOH synthesis technology by the inclusion of N₂ diluent. The use of normal air for syngas production presents a cheaper alternative rather than a cryogenic O₂ which is more capital intensive. Alternatively, cheaper O₂ enriched air can be produced either by PSA or membrane separation technologies.

PSA was preferred for the model since membrane separation is usually economical only at small scale (i.e. < 20 ton/day). When an enriched air is used, the cost of production using a PSA will be cheaper than the use of a cryogenic air separation (Rao & Muller, 2007) considering the 70 % O₂ (in air) purity estimated for the partial oxidation in this work.

The exothermic partial oxidation reaction for the syngas production generates excess energy. The reaction specified at 1200 °C indicates outlet temperatures between 1350-1440 °C. This can raise concerns about the choice of the reactor material. Nevertheless, a typical ATR reactor has a burner operating above 2000 °C (Dybkaer, 1995), and hence a 1440 °C stream outlet with a good heat transfer does not pose extra danger compared to the existing reactors. It is also important to note that the partial oxidation process is already a commercial process in ammonia plants (York et. al., 2003) and considering the high operating temperature, the conversions obtained are practically close to equilibrium.

Furthermore, the difference in energy input and output due to heating indicated surplus of 7.68x10⁸ and 6.10x10⁸ kJ/h released in the normal air-blown and the

enriched air-blown processes respectively. This indicated that heat generated in the process was enough to cover the heating demand in the total process. If the surplus energy is not recovered, the heat lost will be higher in the normal air-blown system. The surplus energy can however be used for power production for example in steam turbines (Ganapathy & Faulkner, 2002). The energy demand will therefore be due to feed compression, which can be reduced by the power generated from the surplus energy.

Thermodynamically, a full conversion per pass can be attained at lower temperatures below 120 °C. This however, depended on the partial pressure of the syngas components. When 39 % N₂ (from normal air-blown) in syngas was used as feed, full syngas conversion was possible at 100 bar. One disadvantage is the need for larger reactor volumes due to the space occupied by the N₂ diluent. However, with the enrichment of the air, same conversion could be attained with 7 % N₂ in syngas at 60 bar. The energy demand from compression relative to MeOH production was therefore estimated to 2270 and 983 MJ/ton MeOH product for the normal air-blown and the enriched air-blown systems respectively.

The optimized LTMS process was put together and a complete process design proposed. The Figure 10 shows a simplified diagram of the proposed air-blown LTMS process. Even though full syngas conversion can be achieved at 100 °C and 60 bar, there are a few experimental drawbacks. H₂O and CO₂ for example are catalyst poisons, and are required to be less than 10 ppm (Liu et. al., 1988; Ohyama, 2003). The amount of methoxide diminishes by reacting with H₂O and CO₂ to produce hydroxide and carbonate respectively. As a result, there is a need to have an absorbing unit to remove H₂O and CO₂ from the syngas. Moreover, as MeOH is separated from the product stream some of the catalyst system which has undergone recycling can be reintroduced into the reactor. Overall, the air-blown ATR for a complete LTMS process design is a promising process for cheaper MeOH production.

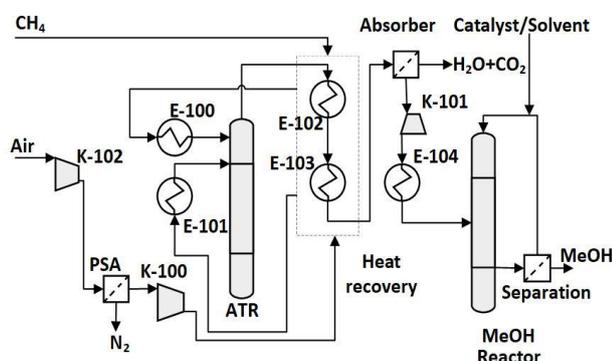


Figure 10. Simplified flow diagram of a complete LTMS process design

6 Conclusion

Simulations and optimizations of air-blown ATR and MeOH synthesis were performed to design a complete LTMS process. A normal air and an O₂ enriched air-blown ATR were optimized for syngas production in the low temperature MeOH synthesis process. Overall, the air-blown system containing 39 % N₂ (from air) in the syngas required about 100 bar to achieve full conversion, while the enriched air-blown system which contained 7 % N₂ in syngas could achieve same conversion at 60 bar. In both cases, the energy generated in the process was enough to cover the heating demand in the total process. When the surplus energy is not recovered, the heat lost will be higher in the normal air-blown system than the enriched air-blown system. The energy required for compression was therefore estimated to be 2270 and 983 MJ/ton MeOH product for the normal air and enriched air-blown systems respectively. An overall design was proposed based on the optimized conditions for the air-blown process.

References

- C. Ahoba-Sam, Boodhoo, K., Olsbye, U., & Jens, K.-J. 2018. Tailoring Cu Nanoparticle Catalyst for Methanol Synthesis Using the Spinning Disk Reactor. *Materials*, 11(1), 154.
- C. Ahoba-Sam, Olsbye, U., & Jens, K.-J. 2017. The Role of Solvent Polarity on Low-Temperature Methanol Synthesis Catalyzed by Cu Nanoparticles. *Frontiers in Energy Research*, 5(15). doi:10.3389/fenrg.2017.00015
- C. Ahoba-Sam, Olsbye, U., & Jens, K.-J. 2018. Low temperature methanol synthesis catalyzed by copper nanoparticles. *Catalysis Today*, 299, 112-119. doi:<https://doi.org/10.1016/j.cattod.2017.06.038>
- J. A. Christiansen. (1919). U.S. Patent 1,302,011.
- I. Dybkjaer. 1995. Tubular reforming and autothermal reforming of natural gas — an overview of available processes. *Fuel Processing Technology*, 42(2), 85-107. doi:[https://doi.org/10.1016/0378-3820\(94\)00099-F](https://doi.org/10.1016/0378-3820(94)00099-F)
- V. Ganapathy, & Faulkner, L. 2002. *Industrial Boilers and Heat Recovery Steam Generators*. Boca Raton: CRC Press.
- J. B. Hansen, & Højlund Nielsen, P. E. 2008. Methanol Synthesis. In G. Ertl, H. Knozinger, F. Schuth, & J. Weitkamp (Eds.), *Handbook of Heterogeneous Catalysis* (Vol. 6, pp. 2920-2949). Weinheim, Germany: Wiley-VCH Verlag GmbH & Co. KGaA.
- J.-P. Lange. 2001. Methanol synthesis: a short review of technology improvements. *Catalysis Today*, 64(1), 3-8. doi:[https://doi.org/10.1016/S0920-5861\(00\)00503-4](https://doi.org/10.1016/S0920-5861(00)00503-4)
- Z. Liu, Tierney, J. W., Shah, Y. T., & Wender, I. 1988. Kinetics of two-step methanol synthesis in the slurry phase. *Fuel Processing Technology*, 18(2), 185-199. doi:[http://dx.doi.org/10.1016/0378-3820\(88\)90095-1](http://dx.doi.org/10.1016/0378-3820(88)90095-1)
- M. Marchionna, Di Girolamo, M., Tagliabue, L., Spangler, M. J., & Fleisch, T. H. 1998. A review of low temperature methanol synthesis. *Studies in Surface Science and Catalysis, Volume 119*, 539-544. doi:[http://dx.doi.org/10.1016/S0167-2991\(98\)80487-4](http://dx.doi.org/10.1016/S0167-2991(98)80487-4)
- S. Ohyama. 2003. Low-Temperature Methanol Synthesis in Catalytic Systems Composed of Copper-Based Oxides and Alkali Alkoxides in Liquid Media: Effects of Reaction Variables on Catalytic Performance. *Topics in Catalysis*, 22(3), 337-343. doi:10.1023/A:1023500725571

- S. Ohyama, & Kishida, H. 1998. Physical mixture of CuO and Cr₂O₃ as an active catalyst component for low-temperature methanol synthesis via methyl formate. *Applied Catalysis A: General*, 172(2), 241-247. doi:[http://dx.doi.org/10.1016/S0926-860X\(98\)00135-5](http://dx.doi.org/10.1016/S0926-860X(98)00135-5)
- G. A. Olah. 2005. Beyond Oil and Gas: The Methanol Economy. *Angewandte Chemie International Edition*, 44(18), 2636-2639. doi:10.1002/anie.200462121
- U. Olsbye, Svelle, S., Bjørgen, M., Beato, P., Janssens, T. V. W., Joensen, F., . . . Lillerud, K. P. 2012. Conversion of Methanol to Hydrocarbons: How Zeolite Cavity and Pore Size Controls Product Selectivity. *Angewandte Chemie International Edition*, 51(24), 5810-5831. doi:10.1002/anie.201103657
- D.-Y. Peng, & Robinson, D. B. 1976. A New Two-Constant Equation of State. *Industrial & Engineering Chemistry Fundamentals*, 15(1), 59-64. doi:10.1021/i160057a011
- P. Rao, & Muller, M. 2007. *Industrial Oxygen, Its Generation and Use*. Paper presented at the Energy Efficiency in Industry.
- A. P. E. York, Xiao, T., & Green, M. L. H. 2003. Brief Overview of the Partial Oxidation of Methane to Synthesis Gas. *Topics in Catalysis*, 22(3), 345-358. doi:10.1023/A:1023552709642

Simulation and Economic Optimization of Amine-based CO₂ Capture using Excess Heat at a Cement Plant

Hassan Ali¹ Lars Erik Øi¹ Nils Henrik Eldrup¹

¹Department of Process, Energy and Environmental Technology, University College of Southeast Norway, Norway
Corresponding author's email address: hassan.ali@usn.no

Abstract

In order to remove CO₂ from power or process industry, a well-known method is absorption in monoethanolamine (MEA) followed by desorption, and this technology has been in operation for decades. The major challenge is the high energy demand for CO₂ desorption. In many industrial cases, a limited amount of cheap waste heat is available and this makes partial CO₂ capture an interesting option. It is not obvious whether a high removal efficiency from a part of the exhaust or a low removal efficiency from the total exhaust is the optimum solution. In this work, simulations of traditional amine-based CO₂ capture processes are performed with full-flow and part-flow of flue gas. The cost of CO₂ capture is estimated using a detailed factor method and a Lang factor method. It is found that a full-flow alternative is the energy optimum while a part-flow alternative treating 80% of the exhaust gas is the cost optimum. This work shows that the calculated optimum is dependent both on the criteria used and on the selected method.

Keywords: CO₂ capture at cement plant, Aspen HYSYS simulation, Partial capture, Cost estimation

1 Introduction

Global warming due to increased greenhouse gas emissions, especially CO₂ emissions has become a major environmental issue. CO₂ emissions have been tripled from fossil fuel, cement industry and flaring since 1970 (IPCC, 2014). The cement industry accounts for around 5% of anthropogenic greenhouse gas emissions. CCS (Carbon capture and storage) is urgently needed along with other energy efficiency measures to reduce the industrial emissions to a level that will meet the 2°C goal (IEAGHG, 2013). United Nations has set this long term goal to limit the global average temperature to well below 2°C above pre-industrial levels since this would reduce the risks and impacts of climate change (IEA, 2015).

1.1 Aim

The aim of this work is to investigate the energy optimum and cost optimum conditions for CO₂ capture from a cement plant with the use of limited excess heat available from the process. Besides, a task is to compare two cost estimation methods, i.e., detailed factor method and Lang factor method.

The subsequent challenge is to perform a cost-benefit analysis of different cases to evaluate whether it is cost optimum to treat all the exhaust gas or only a part of it. Some previous studies (Park, 2016; Øi et al, 2017) have concluded both that a part-flow alternative is optimum and that a full-flow alternative is optimum. The objective of this work is to analyze whether the calculated optimum is dependent both on the criteria used and on the selected method.

1.2 Literature

There have been numerous studies that perform techno-economic analysis of different CO₂ capture concepts, not only for power industry but also for process industries (Rao et al, 2002; Kuramochi et al, 2012) but detailed studies that investigate waste or excess heat potential from process industries to power post combustion CO₂ capture plants are rare.

(Dong et al, 2012) performed a study of the possibility to utilize waste heat from a cement plant to capture CO₂ effluent from the plant. Up to 78 % capture could be achieved using only waste heat by integrating heat recovery with CO₂ capture.

A techno-economic analysis of an oil refinery with amine-based carbon capture plant has been performed (Andersson et al., 2016). In this work, excess heat from the refinery was shown to decrease specific cost of carbon capture.

The (NORDICCS, 2017) project has evaluated the potential of using waste heat from cement industry to cover the reboiler duty of the stripper for an amine-based CO₂ capture plant and concluded that utilisation of waste heat is necessary in order to lower the cost of CO₂ capture. The CO₂stCap project (Skagestad et al, 2017) is in progress in Norway and Sweden to evaluate different possibilities for partial CO₂ capture from industrial sources.

At the University College of Southeast Norway there have been performed simulations of possible CO₂ capture from Norcem cement plant in Brevik (Svolsbru, 2013). (Park, 2016) simulated partial CO₂ capture and concluded that in case of partial CO₂ capture of approximately 40 % of the CO₂ in the flue gas from a cement plant, treating all the flue gas would probably be more cost optimum compared to treat only a part of the flue gas. (Øi et al, 2017) have performed partial CO₂ capture on a traditional amine-based process and a vapour recompression process and concluded that the process with a low absorption column treating the total exhaust gives the lowest cost per ton CO₂ captured.

1.3 Process description

Figure 1 shows a standard process for CO₂ absorption into an amine-based solvent. It comprises an absorption column, a stripping column including a reboiler and condenser, circulating pumps and heat exchangers.

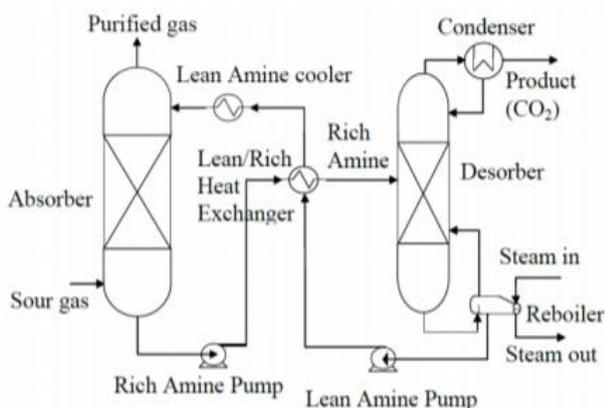


Figure 1. Process flow diagram of a standard amine-based CO₂ capture process (Aromada and Øi, 2015)

A sketch of a general post-combustion partial CO₂ capture process is presented in Figure 2. The whole or a part of a flue gas is sent to an absorber where CO₂ is absorbed in a solvent. The solvent is regenerated by releasing the CO₂ in a desorber and the regenerated solvent is sent back to the absorber.

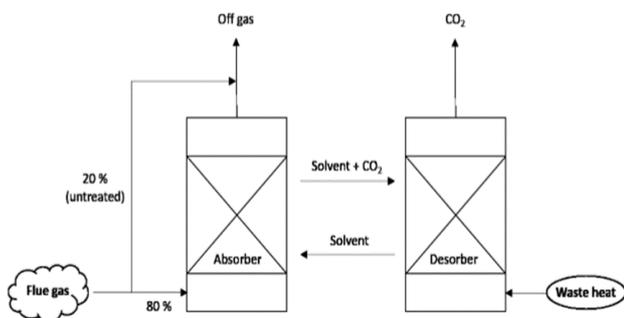


Figure 2. A schematic of partial CO₂ capture (Park, 2016)

2 Methodology

Four case studies are analysed for partial CO₂ capture using only excess heat as mentioned in Table 1.

Table 1. Case studies description

Case study	Description	Flow type
C100	All the flue gas from String 1 goes to the CO ₂ capture plant	Full-flow
C80	80% of flue gas from String 1 goes to the CO ₂ capture plant	Part-flow
C60	60% of flue gas from String 1 goes to the CO ₂ capture plant	Part-flow
C40	40% of flue gas from String 1 goes to the CO ₂ capture plant	Part-flow

The cost and energy optimum alternative from the above four case studies was selected for two more case studies, one with lower reboiler temperature (115 °C)

and the other with a plate & frame heat exchanger to be used as lean/rich heat exchanger. The case studies in this work are performed in two parts:

1. Simulation of amine-based CO₂ capture plant
2. Dimensioning and cost estimation of CO₂ capture plant

2.1 Specifications and simulation of standard CO₂ capture process

All case studies were simulated for a standard process as in Figure 1 using Aspen HYSYS version 8.6 by selecting the Kent-Eisenberg vapour/liquid equilibrium model. Aspen HYSYS is a commercial general purpose process simulation program from AspenTech. It contains several equilibrium models, process unit operation models and flow-sheeting calculation alternatives.

The specifications for the full flow case simulation (case C100) are presented in Table 2. The flue gas (string 1) data are from a cement plant, and the excess heat is assumed to be constant 24.5 MW (NORDICCS, 2017). The absorption and desorption columns are simulated with equilibrium stages including a stage efficiency.

Table 2. Aspen Model parameters and specifications for the full flow alternative (Case study: C100)

Simulation parameter	Value
Flue gas (string 1) temperature from process	80 °C
Inlet flue gas temperature to absorber	40 °C
Inlet gas pressure to absorber	1.1 bar
Inlet flue gas molar flow rate	5788 kmol/h
CO ₂ in inlet flue gas	22.1 mol-%
Lean MEA temperature	40 °C
Lean MEA pressure	1.01 bar
Lean MEA mass flow rate	527500 kg/h
MEA content in Lean MEA	29.0 mass-%
CO ₂ in Lean MEA	5.5 mass-%
Number of stages in absorber	15
Murphree efficiency in absorber stages	0.11 – 0.21
Temperature in amine before desorber	101.2 °C
Number of stages in desorber	10
Murphree efficiency in desorber stages	0.5
Reflux ratio in desorber	0.3
Desorber pressure	2.0 bar
Reboiler temperature	120 °C
Reboiler Power (only excess heat)	24.5 MW
Pressure increase across Lean amine pump	3 bar
Pump efficiency	0.75
ΔT _{min} in Lean/Rich heat exchanger	10 °C

Murphree efficiencies for CO₂ in the absorption column stages are specified; efficiency is constant at 0.21 for the first five stages and then decreases linearly down to 0.11 for stage 15 (Øi, 2012). Murphree efficiency for CO₂ in the desorption column is constant at 0.5. The Murphree efficiency for a stage is defined by the change in mole fraction CO₂ from a stage to another divided by the change on the assumption of equilibrium. Pumps were simulated with an adiabatic efficiency of 0.75.

Figure 3 shows the representation of the standard amine-based absorption-desorption process in the simulation program Aspen HYSYS. The calculation sequence is similar to earlier works (Øi, 2007; Aromada and Øi, 2015). First the absorption column T-100 is calculated from the inlet gas and the lean amine (which is first guessed). The rich amine from the bottom of the absorption column passes through the pump P-100 and the main rich/lean heat exchanger E-102 and gains heat from the lean amine from the desorption column. The heated rich amine is entering the desorption column T-101 which calculates the hot lean amine leaving the desorption column. The hot lean amine leaving from bottom of desorber is being pumped to a higher pressure via lean amine pump P101 and passes through the lean/rich heat exchanger E-102 and is then further cooled in the lean cooler E-101. Then this lean amine is checked in a recycle block RCY-1. It is checked whether the recycled lean amine is sufficiently close to the earlier guessed lean amine stream, which may be changed by iteration. This is completing the loop.

2.2 Dimensioning and cost estimation calculations

2.2.1 Scope analysis

The cost analysis is limited to the equipment listed in the flow-sheet Figure 3 excluding the flue gas cooler. No

pre-treatment like inlet gas purification or cooling is considered. And no treatment after stripping like compression, transport or storage of CO₂ is considered. The cost estimate is limited to installed cost of listed equipment. It does not include e.g. land procurement, preparation, service buildings or owners cost.

2.2.2 Dimensioning of equipment

The dimensions of the process equipment are estimated based on typical dimension factors. The absorption column diameter is based on a gas velocity of 2.5 m/s and the desorption column is based on a gas velocity of 1 m/s (Park and Øi, 2017). The packing height of the absorption and desorption column is 1 meter per stage with a specified stage efficiency. The total height of the absorption column and desorption column is assumed to be 40 m and 22 m respectively. The calculation of absorber height includes packing, liquid distributors, water wash, demister, gas inlet & outlet and sump while calculation of desorber height includes inlet for condenser, packing, liquid distributor, gas inlet and sump.

The heat transfer areas of the heat exchangers are calculated based on duties and temperature conditions obtained from simulations. Overall heat transfer coefficient values have been assumed, for lean/rich heat exchanger 500 W/(m²K), lean amine cooler 800 W/(m²K), reboiler 800 W/(m²K) and condenser 1000 W/(m²K) (Øi, 2012). Shell and tube heat exchangers were mainly considered for case studies but for one alternative study plate & frame heat exchanger was also considered.

Centrifugal pumps are selected for the rich amine and lean amine pump. Volumetric flow rate and pump power are required in order to calculate equipment cost for pumps, which is available from the simulations.

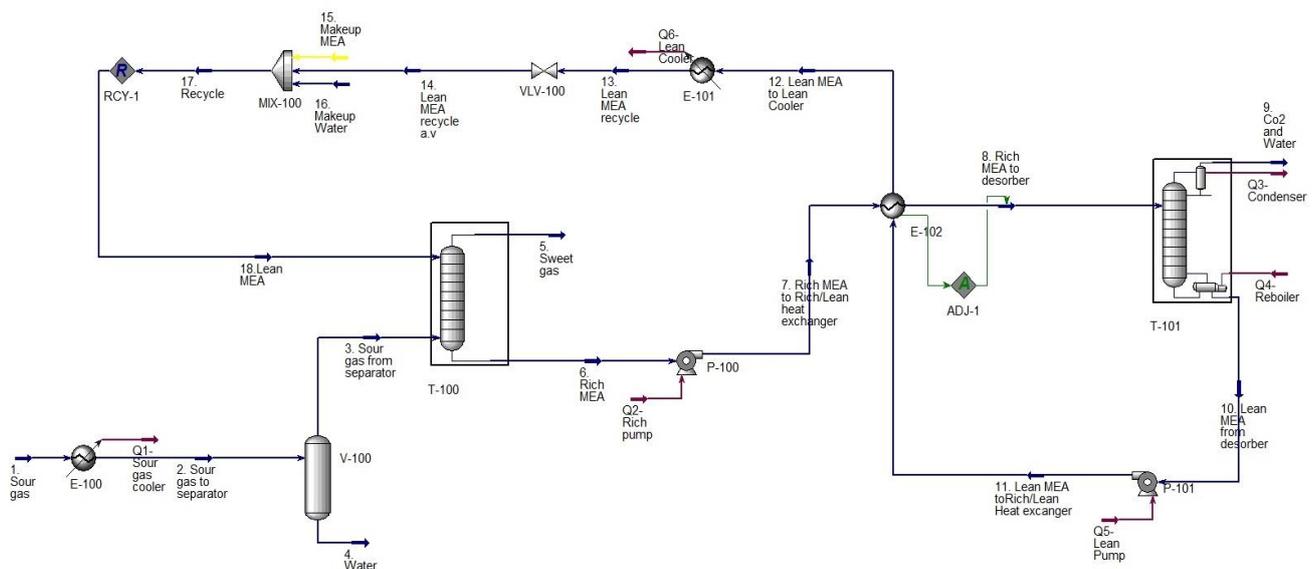


Figure 3. Aspen HYSYS flow-sheet of a standard amine-based CO₂ capture process

2.2.3 Capital cost estimation methods used

The equipment costs are taken from the Aspen In-plant Cost Estimator (v.10), which gives the cost in Euro (€) for Year 2016 (1st Quarter). A generic location that has good infrastructure and easy access to a workforce and materials, e.g. Rotterdam, is assumed. Stainless steel (SS316) with a material factor of 1.75 was assumed for all equipment units. To calculate capital cost, two methods were used.

In the detailed factor method, each equipment cost (in carbon steel) was multiplied with its individual installation factor to get equipment installed cost, as in earlier works (Øi, 2012; Park, 2016). The total capital cost was then calculated by adding all the individual equipment installed costs. The detailed installation factor is a function of the site description, equipment type, materials, size of equipment and includes direct costs (such as the costs for erection, instruments, civil, piping, electrical, insulation, steel and concrete), engineering costs, administration costs and the costs for commissioning and contingency. The updated installation factors for year 2016 (Eldrup, 2016) were used that decreases with increasing equipment cost. This cost estimate is expected to have an accuracy of ±40%.

In the Lang factor method (named after Hans J. Lang in 1947) the idea is to have overall installation factors, called Lang factors, depending upon the type of process plant. In this study, a Lang factor for a fluid process plant which is 4.74 (Turton et al, 2013) has been multiplied with the sum of all equipment costs to estimate the total capital cost.

2.2.4 Operational cost calculation

The electricity cost is set to 0.12 €/kWh. The cooling water cost is set to 0.02 €/m³, and the excess heat is specified to be free although the excess/waste heat always comes with a cost. The annual maintenance cost was set to 4 % of the equipment installed cost. Annual operator cost is added on basis of shift work (6 operators). One operator is assumed to cost 77000 €/year which includes salary as well as employer's expenses. The yearly operating time was 8000 hours, the calculation time was set to 25 years (2 years construction) and the interest was set to 7.5 %.

2.2.5 Capture efficiency and cost calculation

The CO₂ capture efficiency is calculated using equation (1) and the CO₂ capture cost is calculated using equation (2) shown below.

$$\text{Capture efficiency} = \frac{CO_2 \text{ sour gas} - CO_2 \text{ sweet gas}}{CO_2 \text{ sour gas}} \times 100 \quad (1)$$

$$CO_2 \text{ capture cost} \left(\frac{\text{€}}{\text{ton } CO_2} \right) = \frac{CAPEX + OPEX (\text{€/yr})}{CO_2 \text{ captured (ton/yr)}} \quad (2)$$

3 Results and Discussion

For the main four case studies of partial CO₂ capture, Figure 4 shows the plot between captured CO₂ from full flow (C100) to 40% flow (C40) and the cost of capture per ton CO₂. The lowest cost is obtained for C80 with the detailed factor method. The cost results for the Lang factor method has a higher cost per ton CO₂ captured than with the detailed factor method for all the cases. The reason for this is the fact that in the detailed factor method, each equipment gets different installation factor and when the installation factors for all the equipment are combined, that was found to be less than the Lang factor (4.74) used for this study.

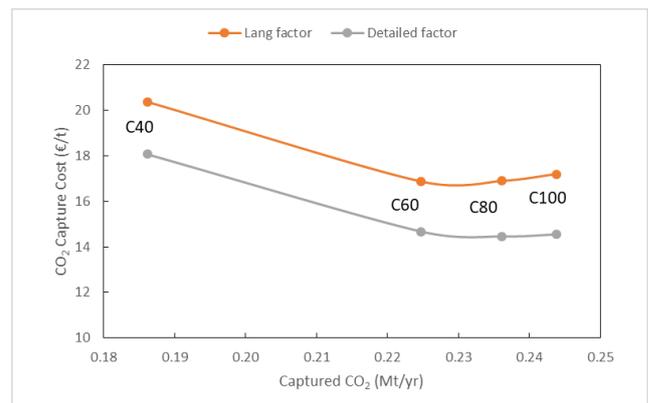


Figure 4. CO₂ capture cost plotted against captured CO₂ for full-flow and part-flow case studies

The curve in Figure 4 also indicates that the cost of CO₂ capture initially goes down when the amount of CO₂ capture decreases from 0.245 Mt/yr to around 0.23 Mt/yr but then the cost increases sharply as the captured amount decreases further.

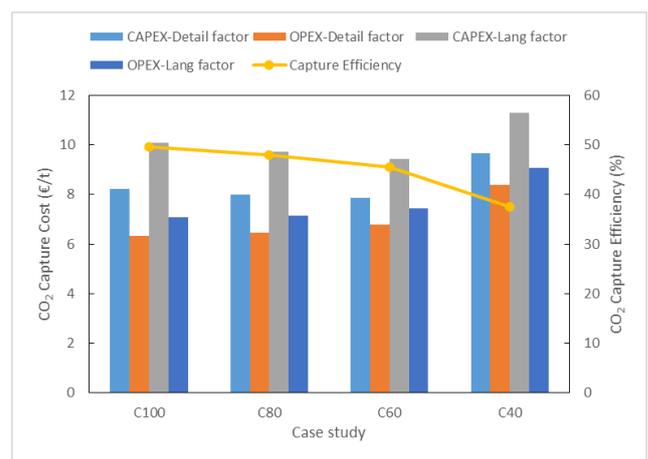


Figure 5. Overall cost analysis of four case studies

Detailed cost analysis and capture efficiency for the main four case studies is shown in Figure 5. The CAPEX dominates in all the case studies. The best capture efficiency is for case study C100 but the capture efficiency does not fall down drastically from C100 to C60 (49.6 to 45.5%). While for C40, the efficiency falls down to 37% and this case study has also the highest

capture cost as well. The energy optimum case study proves to be C100. The cost optimum case study when it comes to Lang factor is C60 (16.87 €/t) but capture cost of C80 (16.90 €/t) is not far away from the lowest. With the detailed factor method, the lowest capture cost comes for the case study C80 (14.46 €/t) while capture cost for C100 (14.54 €/t) is close to that of C80. Hence, the case study C80 with the detailed factor method is cost optimum and selected for further analysis.

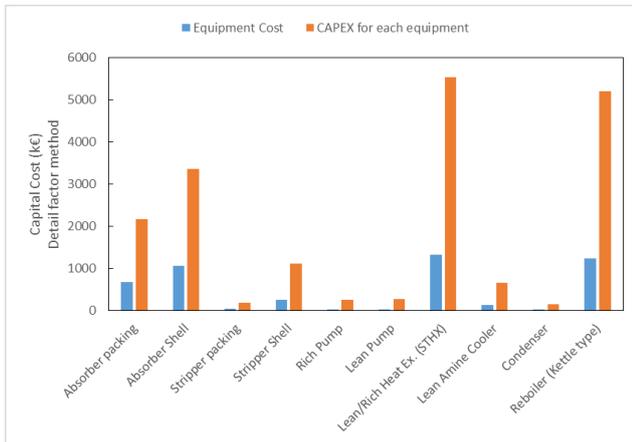


Figure 6. Capital cost overview of case study C80 (STHX, Shell & tube heat exchanger)

Since CAPEX dominates the capture cost, it will be worthwhile to have a detailed look on the capital cost of case study C80 that helps in optimization, which is shown in Figure 6. There are four major equipment, lean/rich heat exchanger, reboiler, absorber shell and packing that are contributing significantly and the efforts should be directed to reduce this cost.

An alternative to reduce the lean/rich heat exchanger capital cost is to replace the shell and tube heat exchanger (STHX) with a plate and frame heat exchanger (PFHX) (Marcano, 2015). That has also been performed for all the case studies, with the name PFHX and the results are presented in Figure 7.

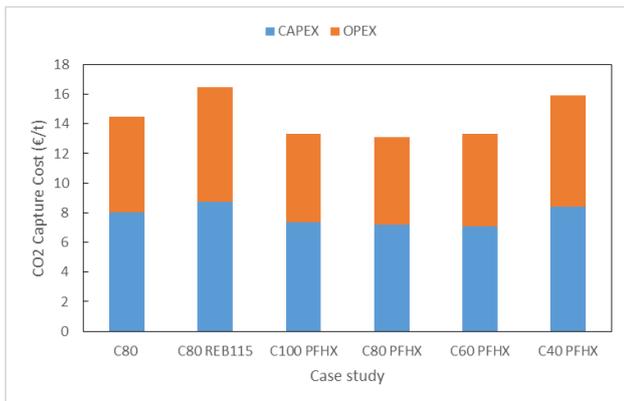


Figure 7. Cost overview of alternatives with PFHX, Plate & frame heat exchanger and with lower reboiler duty

The results clearly indicates that by replacing shell & tube heat exchanger with plate & frame heat exchanger (for lean/rich heat exchanger), the capture cost further

decreases for all the cases. The lowest capture cost in this scenario remains to be case study C80 that decreased from 14.46 €/t (with STHX) to 13.11 €/t (with PFHX).

In another alternative on case study C80, the reboiler temperature has been decreased from 120 °C to 115 °C. By doing this, more excess heat can be available and it might help in reducing the capture cost. The results of this new case study C80 REB115 is also presented in Figure 7. For this case study, excess heat was increased to 25.1 MW since we can utilize further excess heat of 5 °C from hot exhaust gas. Table 3 contains some important input parameters and outputs for case studies C80 and C80 REB115.

Table 3. Input Parameters and results for case study C80 and C80REB115

Parameter	Unit	Case Study	
		C80	C80REB115
Flue gas flow rate	Kmol/h	4630	4630
Excess heat to reboiler	MW	24.5	25.1
Lean MEA flow rate	kg/h	535000	845900
Lean loading		0.26	0.35
Rich loading		0.51	0.50
CO ₂ capture efficiency	%	47.9	46.3
CO ₂ removed per year		0.236	0.228
Reboiler energy demand	MJ/kg CO ₂	3.27	3.47

The results in Figure 7 shows that the capture cost has increased from 14.46 €/t to 16.46 €/t for case study C80 REB115 even though the excess heat has been increased. Besides the capture cost, reboiler energy demand has also increased for this lower reboiler temperature case study, while the capture efficiency and CO₂ removed per year decreases as shown in Table 3.

3.1 Sensitivity analysis

The sensitivity analysis has been performed on capital cost, specifically on the installation factors of the four most costly equipment identified i.e., lean/rich heat exchanger, reboiler, absorber shell and packing. Installation factors for these equipment have been decreased by 50% to see the impact they have on capture cost of main four case studies.

Another analysis has been performed on civil installation sub-factor. This sub-factor of the detailed installation factor is expected to cover additional cost due to equipment cost (and size). This sub-factor has also been decreased by 50% for all the equipment installation factors and its effect on capture cost has been analysed.

The results are presented in Table 4, which shows that by decreasing the installation factors for absorber packing, the full flow case C100 becomes the cost optimum case although the lowest cost 12.82 €/t is achieved for case C80 when installation factor for lean/rich heat exchanger is reduced. For all other scenarios, case C80 continues to give lowest cost per ton

when the installation factor or civil sub-factor is decreased by 50%. The greatest impact on capture cost is by the lean/rich heat exchanger and the reboiler, the capture cost goes down significantly from 1.4 – 2.5 €/t for all the cases. The lowest impact is by the civil sub-factor where the capture cost decreases by only 0.13 €/t for cases C100 to C60 apart from for the C40 case where the increase is 0.96 €/t.

Table 4. Effect of installation factors (IF) and civil sub-factor (factors decreased by 50%) on capture cost

Case study	C100	C80	C60	C40
Capture Cost, €/t	14.54	14.46	14.67	18.06
IF-Abs. Packing, €/t	13.77	13.82	14.17	17.66
IF-Abs. Shell, €/t	13.47	13.46	13.79	17.15
IF-Reboiler, €/t	13.07	12.92	13.05	16.10
IF-l/r heat exch., €/t	12.86	12.82	13.02	15.48
Civil sub-factor, €/t	14.41	14.33	14.54	17.10

In a more detailed analysis for cost optimization, the number of stages in the absorber should be optimized but this is not included in the scope of this study.

3.2 Comparisons with earlier work

(Dong et al, 2012) calculated that it was possible to capture 78 % CO₂ in a cement case under other conditions. The amount captured was dependent on the degree of integration. (Park, 2016) concluded that the lowest total cost per ton CO₂ captured was calculated for the standard full-flow process with 5 absorption stages. This conclusion was however based on the assumption that transport and treating of the gas before or after CO₂ capture was not considered. (Øi et al, 2017) worked on partial capture from flue gas of cement industry and concluded that the energy optimum case and the lowest total cost per ton CO₂ captured was calculated for the standard full-flow process with a low number of absorption stages.

4 Conclusion

Different case studies from full flow of the flue gas from String 1 to part flow for partial CO₂ capture in a cement industry were simulated with only excess heat using the process simulation tool Aspen HYSYS. These case studies were cost estimated using the Aspen In-plant cost estimator along with two cost estimation methods i.e., detailed factor method and Lang factor method.

The highest CO₂ removal efficiency is obtained for the full flow alternative which is regarded as the energy optimum process with a reboiler energy demand around 3.2 MJ/kg CO₂. The cost optimum case was with 60% of the flue gas flow into the capture plant, when the Lang factor method was used. When using the detailed factor method, the case with 80% of the flue gas flow is the cost optimum alternative. This is valid for all the different case studies performed via detailed factor method with the exception when the installation factor for absorber packing was decreased, the full flow

alternative becomes the cost optimum. This clearly shows that the selection of the cost estimation method and the assumptions made have a great impact on the results.

The greatest impact on capture cost was by the capital cost, specifically by the lean/rich heat exchanger, reboiler, absorber shell and packing. The capture cost can be reduced by selecting a plate and frame heat exchanger as the lean/rich heat exchanger.

Acknowledgments

The authors wish to thank the partners in the CO₂stCap project: The University College of Southeast Norway, Gassnova, Energimyndigheten, Rise Bioeconomy, SINTEF, Chalmers, Swerea MEFOS, SSAB, GCCSI, IEAGHG, Elkem AS, Norcem Brevik AS and AGA Gas AB.

References

- V. Andersson, P. Å. Franck, and T. Berntsson. Techno-economic analysis of excess heat driven post-combustion CCS at an oil refinery. *International Journal of Greenhouse Gas Control*, 45: 130-138, 2016. doi: <https://doi.org/10.1016/j.ijggc.2015.12.019>
- S. A. Aromada and L. E. Øi. Simulation of Improved Absorption Configurations for CO₂ Capture. *Linköping Electronic Conference Proceedings SIMS 56*, 21-29, 2015. doi: <http://dx.doi.org/10.3384/ecp1511921>
- R. Dong, Z. Zhang, H. Lu, Y. Yu. Recovery of waste heat in cement plants for the capture of CO₂. *Frontiers of Chemical Science and Engineering*, 6(1): 104-111, 2012. doi: <https://doi.org/10.1007/s11705-011-1166-0>
- N. H. Eldrup. *Installation factor sheet - Project Management and Cost Engineering*. Master's Course. University College of Southeast Norway, Porsgrunn, 2016.
- IEA (2015): Carbon Capture and Storage: The solution for deep emission reductions. *International Energy Agency*, Paris, France.
- IEAGHG (2009): Partial Capture of CO₂. *IEA Greenhouse Gas R&D Programme*, 2009/TR2, Cheltenham, UK.
- IEAGHG (2013): Deployment of CCS in the Cement Industry. *International Energy Agency*, 2013/19, December, 2013.
- IPCC (2014): Climate Change 2014: Synthesis Report. Contribution of Working Groups I, II and III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change [Core Writing Team, R.K. Pachauri and L.A. Meyer (eds.)]. *IPCC*, Geneva, Switzerland, 151 pp.
- T. Kuramochi, A. Ramirez, W. Turkenburg and A. Faaij. Comparative assessment of CO₂ capture technologies for carbon-intensive industrial processes. *Progress in Energy and Combustion Science*, 38: 87-112, 2012. doi: <https://doi.org/10.1016/j.peccs.2011.05.001>
- L. A. Marcano. *Design and evaluation of heat exchangers used in post-combustion CO₂ capture plants*. Master's Thesis. Telemark University College, Porsgrunn, 2015.
- NORDICCS (2017): Industrial implementation of Carbon Capture in Nordic industry sectors. *NORDICCS*, Technical Report D4.2.1501/D18.
- K. Park. *Optimization of partial CO₂ capture*. Master's Thesis, Telemark University College, Porsgrunn, 2010.
- K. Park and L. E. Øi. Optimization of gas velocity and pressure drop in CO₂ absorption column. *Linköping Electronic*

- Conference Proceedings SIMS 58*, 292-297, 2017. doi: 10.3384/ecp17138292
- A. B. Rao and E. S. Rubin. Technical, Economic and Environmental Assessment of Amine-based CO₂ Capture Technology for Power Plant Greenhouse Gas Control. *Environmental Science and Technology*, 36: 4467-4475, 2002. doi: 10.1021/es0158861
- R. Skagestad, F. Normann, S. Ó. Garðarsdóttir, M. Sundqvist, M. Anheden, N. H. Eldrup, H. Ali, H. A. Haugen and A. Mathisen. CO₂stCap - Cutting Cost of CO₂ Capture in Process Industry. *Energy Procedia*, 114: 6303-6315, 2017. doi: <https://doi.org/10.1016/j.egypro.2017.03.1767>
- C. B. Svolsbru. *CO₂ capture alternatives in cement plant*. Master's Thesis, Telemark University College, Porsgrunn, 2013.
- R. Turton, R. C. Bailie, W. B. Whiting, J. A. Shaeiwitz and D. Bhattacharyya (2013). Analysis, Synthesis, and Design of Chemical Processes. *Pearson Education International*, Fourth edition. New York.
- L. E. Øi. Aspen HYSYS simulation of CO₂ removal by amine absorption a gas based power plant. *The 48th Scandinavian Conference on Simulation and Modelling (SIMS 2007)*, Göteborg, Sweden, 2007.
- L. E. Øi. *Removal of CO₂ from exhaust gas*. PhD Thesis, Telemark University College, Porsgrunn. (TUC 3: 2012)
- L. E. Øi, E. Sundbø and H. Ali. Simulation and Economic Optimization of Vapour Recompression Configuration for Partial CO₂ Capture. *Linköping Electronic Conference Proceedings SIMS 58*, 298-303, 2017. doi: 10.3384/ecp17138298

Deriving the Beam Equation using the Minimum Total Potential Energy Principle and Solving the Equation Numerically

Magnus Komperød¹

¹Technological Analyses Centre, Nexans Norway AS, Norway, magnus.komperod@nexans.com

Abstract

The beam equation describes the deflection of a beam subject to point loads and / or distributed loads, while being supported at both ends. The beam equation is commonly derived in the scientific literature using force- and moment balances, which lead to a boundary value problem. The present paper derives the beam equation using the minimum total potential energy principle and solves the optimization problem numerically. The motivation behind this work is to ease future extensions of the beam equation into larger deflections and nonlinear materials. These future extensions are necessary to model subsea power cables and umbilicals during bending stiffness tests which is the author's final goal.

Keywords: Beam Equation, Bending Stiffness Test, Minimum Total Potential Energy Principle, Numerical Analysis, Subsea Power Cable, Umbilical.

1 Introduction

The offshore oil and gas industry, as well as power grid companies, require increasingly accurate analyses and physical testing of subsea power cables and umbilicals to be able to install them in deeper waters, in colder environments, and during harsher weather conditions. This challenges engineers, scientists, and software developers working in the cable manufacturing industry and among service- and software providers.

Subsea power cables and umbilicals are complex structures consisting of a large number of individual elements, such as the umbilical shown in Figure 1. For some cable designs the number of individual elements exceeds 200. The relative displacements between these elements and complex nonlinear material characteristics are among the main challenges in modeling cables' mechanical properties. As both the cable geometry and the material properties are highly complex, there will inherently be uncertainties in the cable models. It is therefore essential to validate the models through physical testing.

The scientific literature on physical bending stiffness tests of cables is very sparse. Hence, there is very limited information on various test rig designs; Maioli (2015) and Tarnowski (2015) both present results based on rigs with a design as sketched in Figure 2. In Maioli (2015) the arrangement is horizontal, i.e. the force is applied in the horizontal plane, while in Tarnowski (2015) the force

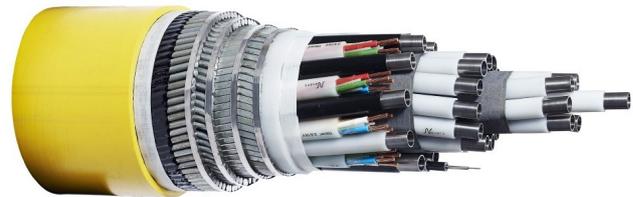


Figure 1. Umbilical with steel tubes, electric- and fiber optic signal cables, and armor wires.

is applied vertically.

The author and his colleagues have developed a bending stiffness rig which is also based on the principle illustrated in Figure 2. This work is presented in Jordal et al. (2017) and Komperød et al. (2017). The rig is vertical and the force can be applied in both directions, i.e. up and down, in a sinusoidal manner, making the cable oscillate.

In Komperød et al. (2017) the beam equation is used to calculate the cable's bending stiffness based on measurements of the cable's deflection and the force required to achieve this deflection. This equation is derived for beams under the assumptions of small deflections and linear, elastic material in the beam. The exact same equation is also used in Maioli (2015). Tarnowski (2015) presents another model that intends to handle larger deformations. However, also this approach considers the cable as one solid object, rather than a compound object consisting of a large number of individual cable elements.

When cables are bent, there are relative displacements between the cable elements. A large number of scientific publications consider the effect of shear forces between cable elements due to friction, for example Lutchansky (1969) and Kebabze (2000), and show that this effect has major impact on the cables' bending stiffness. Over the last few years, also shear forces due to bitumen-coating on the armor wires have got attention in the scientific literature. Among the publications on this topic are Hedlund (2015), Komperød (2016a,b), and Martindale et al. (2017).

Analyses of cables' mechanical properties are commonly (i) assuming constant cable bending curvature

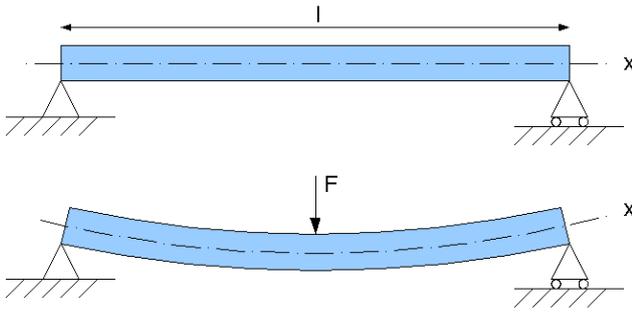


Figure 2. Sketch of bending stiffness rig. Illustration from www.wikimedia.org.

along the cable's length direction and (ii) neglecting the boundary conditions at the cable ends. However, to the author's knowledge, it is not possible to perform a physical bending stiffness test of a cable in such a way that the cable's physical boundary conditions will not influence the test results. Also, the condition of constant bending curvature is challenging to meet in real-life test rigs. Hence, the bending stiffness calculated in analyses and the bending stiffness identified from physical tests are not directly comparable because they are based on different assumptions.

To close the gap between analyses and physical testing, the author develops models of cables subject to the loads and the boundary conditions of the bending stiffness rig. These models can then be directly compared to, and hence validated against, the results of the physical tests. When the models have been validated, changing their loads and boundary conditions allows the same models to be used in the case of constant curvature and no boundary conditions as discussed above. This way the author intends to bridge the differences between analyses and physical testing which are currently preventing the analyses from being verified against physical testing in a consistent way. Due to the high complexity of the overall problem, the work will be split into 4-6 milestones.

This paper presents the results of the first milestone, which is to derive the beam equation using the minimum total potential energy principle and to solve this equation numerically. Using the minimum total potential energy principle, rather than force- and moment balances, is believed to simplify future extensions of the beam equation into large deflections, nonlinear materials, and correct modeling of shear forces between the cable elements. It is further believed that it will be impossible to reach an analytical solution to the overall problem. It is therefore desirable to develop a numerical solution during the first milestone which can later be extended when working towards future milestones.

2 Nomenclature

Table 1 presents the main nomenclature used in this paper. Variables without a specific physical meaning are defined in the main text where they first appear. Vectors are denoted with lower case, bold font. Matrices are denoted with upper case, bold font. Square bracket are used to denote parts of a vector or a matrix. For example $\mathbf{A}[1 : 4, 3]$ means the first four rows of the third column of \mathbf{A} . Exponent notation applied to vectors, for example \mathbf{x}^2 , means element-by-element exponent.

Table 1. Nomenclature.

$\bar{\mathbf{c}}$	Vector with Clenshaw-Curtis quadrature weights.
\mathbf{D}_1	Chebyshev first-derivative matrix.
\mathbf{D}_2	Chebyshev second-derivative matrix.
EI	Bending stiffness [$\text{Nm}/(\text{m}^{-1})$].
g	Acceleration of gravity [m/s^2].
L	Length between beam supports [m].
M	Bending moment [Nm].
\bar{m}	Mass per unit length [kg/m].
N	Number of discretization points [-].
\bar{N}	Number of discretization points over a half beam [-].
P	Total potential energy [J].
P_g	Gravitational energy [J].
P_s	Strain energy [J].
s	Beam length parameter [m].
u	Vertical deflection of beam [m].
u_p	Vertical deflection of piston [m].
\mathbf{u}	Vector containing discrete points of u .
x	Coordinate along x -axis [m].
κ	Bending curvature [m^{-1}].
λ_i	Lagrange multiplier no. i .

Figure 3 shows the Cartesian coordinate system used in this paper.

3 Assumptions and Simplifications

The mathematical derivation in this paper is subject to the following assumptions and simplifications:

1. The beam is subject to infinitesimal deflections only.
2. The beam is made of a linear, elastic material and has identical cross section over its entire length.
3. Only the beam length between the supports shown in Figure 2 is considered. That is, possible beam length outside these supports is disregarded.

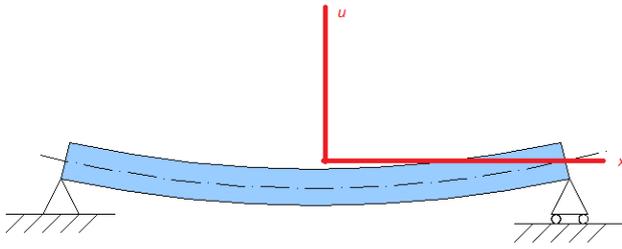


Figure 3. The Cartesian coordinate system used in this paper. Based on illustration from www.wikimedia.org.

4. The beam is assumed not to move horizontally at the center of its length, i.e. where the F arrow points in Figure 2.
5. The developed model is quasi-static, i.e. inertia and kinetic energy are disregarded.
6. The beam's height is much smaller than the length between the supports, L .

4 Mathematical Model

This section derives the mathematical model of the total potential energy of the beam, as well as the constraints that follow from the physical test rig and the beam's physical properties. How to formulate the model and the constraints numerically, and how to numerically solve for the minimum total potential energy are derived in Section 5.

4.1 Strain Energy in the Beam

The bending stiffness of the beam, EI , is by definition

$$EI \stackrel{\text{def}}{=} \frac{M}{\kappa}. \quad (1)$$

Solving Eq. 1 for the moment, M , and integrating w.r.t. the curvature, κ , gives the beam's strain energy, P_s , per unit length, i.e.

$$\begin{aligned} \frac{\partial P_s}{\partial s} &= \int_0^{\kappa} M d\kappa \\ &= \frac{1}{2} EI \kappa^2. \end{aligned} \quad (2)$$

Hence, the total strain energy in the beam is

$$\begin{aligned} P_s &= \int_{\text{beam length}} \frac{1}{2} EI \kappa^2 ds \\ &= \frac{1}{2} EI \int_{\text{beam length}} \kappa^2 ds, \end{aligned} \quad (3)$$

where $\kappa = \kappa(s)$. The integration limits will be discussed in more detail in Section 5.

4.2 Gravitational Energy

When choosing $u = 0$ as the reference level for calculating the gravitational energy, P_g , the gravitational energy per unit length is

$$\frac{\partial P_g}{\partial s} = \tilde{m}gu. \quad (4)$$

Hence, the gravitational energy over the entire beam is

$$\begin{aligned} P_g &= \int_{\text{beam length}} \tilde{m}gu ds \\ &= \tilde{m}g \int_{\text{beam length}} u ds, \end{aligned} \quad (5)$$

where $u = u(s)$.

4.3 Total Potential Energy

The beam's total potential energy, P , is the sum of the strain energy and the gravitational energy, i.e.

$$\begin{aligned} P &= P_s + P_g \\ &= \frac{1}{2} EI \int_{\text{beam length}} \kappa^2 ds \\ &\quad + \tilde{m}g \int_{\text{beam length}} u ds. \end{aligned} \quad (6)$$

4.4 Constraints

The purpose of finding the total potential energy in Eq. 6 is to minimize this expression w.r.t. the beam's deflection $u(s)$ over its length, s , which will give its total potential energy and its deflection profile. However, the beam is not completely free to move and deform; it is constrained by the test rig and by its own physical properties. In mathematical terms this translates into constraints that apply when minimizing Eq. 6 w.r.t. $u(s)$.

The bending stiffness rig developed by the author and his colleagues is as sketched in Figure 2. As the rig is designed to oscillate between positive and negative values of u (i.e. up and down), the supports are made to both prevent the beam from being pushed down and from being lifted up. Similarly, the force F is generated by a piston which can both push the beam down and lift it up. Hence, the constraints are

$$u(\text{left end}) = 0, \quad (7)$$

$$u(\text{center}) = u_p, \quad (8)$$

$$u(\text{right end}) = 0, \quad (9)$$

where u_p is the piston's position.

From beam theory it follows that the function $u(s)$ and its first- and second derivatives are continuous. This is also a constraint which can mathematically be expressed as

$$u(s) \in C^2, \quad (10)$$

where C^2 is the set all functions which are continuous and have continuous first- and second derivatives (and possibly continuous higher order derivatives). The mathematical importance of the latter constraint will be further explained in Section 5.

5 Numerical Solution

Section 4 derives a mathematical model of the total potential energy in the beam, as well as the associated constraints. The present section solves the constrained optimization problem, i.e. finds the minimum total potential energy, using numerical mathematics.

5.1 Implications of Assumptions and Simplifications

Assumptions 1 of Section 3 implies that the beam's length parameter s will be identical to the x coordinate. Assumptions 1 and 4 combined mean that the beam will not move horizontally at any point. Hence, the beam's length exactly spans the interval $[-\frac{L}{2}, \frac{L}{2}]$. It is then convenient to replace s by x , and consider the beam over $x \in [-\frac{L}{2}, \frac{L}{2}]$. It then follows that the integration limits of Eqs. 3, 5, and 6 are $-\frac{L}{2}$ and $\frac{L}{2}$.

From the mathematical literature, the curvature of a graph $u(x)$ is known to be

$$\kappa = \frac{\frac{d^2u}{dx^2}}{\left(1 + \left(\frac{du}{dx}\right)^2\right)^{\frac{3}{2}}}. \quad (11)$$

Under assumption 1, $\frac{du}{dx}$ is very small compared to unity. Eq. 11 can then be approximated by

$$\kappa = \frac{d^2u}{dx^2}. \quad (12)$$

5.2 Calculating the Strain Energy

The overall purpose of the numerical solution of the beam equation is to find the $u(s)$ that minimizes the total potential energy subject to the constraints. To the author's knowledge, numerical methods are not able to find the exact function $u(s)$. What these methods instead do is (i) to find $u(s)$ at discrete points which allows subsequent interpolation, as well as numerical differentiation and quadrature (numerical integration), or (ii) to find the coefficients of a polynomial or a series, for example a Fourier series or a Chebyshev series, which then serves as an approximation to $u(s)$. In this paper it has been chosen to base the calculations on discrete points that are Chebyshev-distributed along x , because this distribution gives excellent convergence properties both in interpolation, differentiation, and quadrature.

The constraint of Eq. 10 requires $u(s)$ to be continuous and to have continuous first- and second derivatives. However, $u(s)$ does not have continuous third derivative at the point where the piston, i.e. the force F , pushes or pulls the beam. This is because the force gives an abrupt change in the beam's shear force. To prevent the abrupt third derivative from disturbing the excellent convergence properties of Chebyshev series for smooth functions, the function $u(s)$ is split in two halves; one half to the left of the force F , i.e. in the interval $[-\frac{L}{2}, 0)$, and one half to the right of the force, i.e. the interval $(0, \frac{L}{2}]$. Eq. 10 then imposes constraints on how the two halves should be connected.

The numbers of Chebyshev nodes over the left half and the right half are the same, namely \bar{N} . Then the total number of nodes is

$$N = 2\bar{N} - 1, \quad (13)$$

because the center node is common for both halves. The nodes are organized in a column vector $\mathbf{u} \in \mathbb{R}^N$.

The following text explains how to calculate the strain energy in the left half of the beam as function of the discrete-point deflections of the vector \mathbf{u} . Let $\bar{\mathbf{u}}$ be the first \bar{N} nodes of \mathbf{u} , i.e.

$$\bar{\mathbf{u}} \stackrel{\text{def}}{=} \mathbf{u}[1 : \bar{N}] \in \mathbb{R}^{\bar{N}}. \quad (14)$$

Let $\bar{\mathbf{k}}$ be the vector with curvature values, κ , at the Chebyshev nodes. The curvature can be approximated by the second derivative as stated in Eq. 12. Hence, the curvatures can be calculated as

$$\bar{\mathbf{k}} = \mathbf{D}_2 \bar{\mathbf{u}}, \quad (15)$$

where \mathbf{D}_2 is the Chebyshev second-derivative matrix of dimension $\bar{N} \times \bar{N}$. The structure of this matrix is derived in Reid (2014).

From Eq. 3 and the reasoning of Section 5.1, the strain energy in the left half of the beam can be approximated by

$$P_s(\text{left half}) = \frac{1}{2}EI \int_{-L/2}^0 \kappa^2 dx. \quad (16)$$

Eq. 15 gives that the strain energy per unit length at the Chebyshev nodes is $\frac{1}{2}EI(\mathbf{D}_2\bar{\mathbf{u}})^2$, where the superscript means element-by-element exponent. The strain energy in the left half of the beam is then given by Clenshaw-Curtis quadrature, i.e.

$$P_s(\text{left half}) = \left(\frac{L}{4}\bar{\mathbf{c}}\right)^T \left(\frac{1}{2}EI(\mathbf{D}_2\bar{\mathbf{u}})^2\right), \quad (17)$$

where $\bar{\mathbf{c}}$ is the vector containing the Clenshaw-Curtis quadrature weights for the standard quadrature interval $[-1, 1]$. In the first parenthesis of Eq. 17, a factor $\frac{L}{2}$ is the quadrature interval width, while a factor $\frac{1}{2}$ is to cancel the width 2 of the standard interval. Clenshaw-Curtis quadrature is explained in Reid (2014).

Eq. 17 can be rewritten to

$$P_s(\text{left half}) = \frac{EIL}{8} \bar{\mathbf{u}}^T \mathbf{D}_2^T \text{diag}(\bar{\mathbf{c}}) \mathbf{D}_2 \bar{\mathbf{u}}, \quad (18)$$

where the operator $\text{diag}(\cdot)$ returns a diagonal matrix with the argument vector's elements on the diagonal. It is now convenient to define the symmetric matrix $\bar{\mathbf{W}}$ as

$$\bar{\mathbf{W}} \stackrel{\text{def}}{=} \frac{EIL}{4} \mathbf{D}_2^T \text{diag}(\bar{\mathbf{c}}) \mathbf{D}_2 \in \mathbb{R}^{\bar{N} \times \bar{N}}. \quad (19)$$

Eq. 18 can then be written as

$$P_s(\text{left half}) = \frac{1}{2} \bar{\mathbf{u}}^T \bar{\mathbf{W}} \bar{\mathbf{u}}. \quad (20)$$

Eq. 20 gives an expression for the strain energy in the left half of the beam which is easy to handle, and in particular easy to differentiate. Differentiation is important for later finding the minimum. However, it is not straight forward how to extend the equation to also include the right half of the beam, while keeping it easy to differentiate, because the center node is included in the calculations of both halves. The following presents a simple solution. A block matrix \mathbf{W}_L is defined as

$$\mathbf{W}_L \stackrel{\text{def}}{=} \begin{bmatrix} \bar{\mathbf{W}} & \mathbf{0}_{\bar{N} \times (\bar{N}-1)} \\ \mathbf{0}_{(\bar{N}-1) \times \bar{N}} & \mathbf{0}_{(\bar{N}-1) \times (\bar{N}-1)} \end{bmatrix}, \quad (21)$$

where the subscript L means left, and $\mathbf{0}$ means the zero matrix of the dimension indicated by the subscripts. The

strain energy in the left half of the beam can then be written as

$$P_s(\text{left half}) = \frac{1}{2} \mathbf{u}^T \mathbf{W}_L \mathbf{u}. \quad (22)$$

Please note that \mathbf{u} , rather than $\bar{\mathbf{u}}$, is used in Eq. 22.

Making a similar reasoning as Eqs. 14-22 for the strain energy of the right half of the beam gives

$$P_s(\text{right half}) = \frac{1}{2} \mathbf{u}^T \mathbf{W}_R \mathbf{u}, \quad (23)$$

where

$$\mathbf{W}_R \stackrel{\text{def}}{=} \begin{bmatrix} \mathbf{0}_{(\bar{N}-1) \times (\bar{N}-1)} & \mathbf{0}_{(\bar{N}-1) \times \bar{N}} \\ \mathbf{0}_{\bar{N} \times (\bar{N}-1)} & \bar{\mathbf{W}} \end{bmatrix}. \quad (24)$$

In Eq. 24 the subscript R means right, and $\bar{\mathbf{W}}$ is the matrix defined in Eq. 19.

The strain energy over the entire beam is found by adding the two halves, i.e.

$$\begin{aligned} P_s &= \frac{1}{2} \mathbf{u}^T \mathbf{W}_L \mathbf{u} + \frac{1}{2} \mathbf{u}^T \mathbf{W}_R \mathbf{u} \\ &= \frac{1}{2} \mathbf{u}^T (\mathbf{W}_L + \mathbf{W}_R) \mathbf{u} \\ &= \frac{1}{2} \mathbf{u}^T \mathbf{W} \mathbf{u}, \end{aligned} \quad (25)$$

where

$$\mathbf{W} \stackrel{\text{def}}{=} \mathbf{W}_L + \mathbf{W}_R. \quad (26)$$

Eq. 25 gives an expression for the strain energy in the entire beam which is easy to differentiate w.r.t. \mathbf{u} .

5.3 Calculating the Gravitational Energy

Numerical calculation of the gravitational energy resembles numerical calculation of the strain energy, but it is simpler, partly because it depends on $u(s)$ rather than its second derivative, and partly because it is linear rather than quadratic.

Using Eq. 5 and Clenshaw-Curtis quadrature, the gravitational energy of the left half of the beam can be approximated by

$$P_g(\text{left half}) = \left(\frac{L}{4}\bar{\mathbf{c}}\right)^T (\tilde{m}g\bar{\mathbf{u}}), \quad (27)$$

where $\bar{\mathbf{u}}$ is as defined in Eq. 14, and $\bar{\mathbf{c}}$ is the vector which elements are the Clenshaw-Curtis quadrature weights. Eq. 27 can be rewritten to

$$P_g(\text{left half}) = \mathbf{q}_L^T \mathbf{u}, \quad (28)$$

for the vector \mathbf{q}_L defined as

$$\mathbf{q}_L \stackrel{\text{def}}{=} \frac{\tilde{m}gL}{4} \begin{bmatrix} \bar{\mathbf{c}} \\ \mathbf{0}_{(\bar{N}-1) \times 1} \end{bmatrix}. \quad (29)$$

Similarly, the gravitational energy of the right half of the beam can be expressed as

$$P_g(\text{right half}) = \mathbf{q}_R^T \mathbf{u}, \quad (30)$$

for

$$\mathbf{q}_R \stackrel{\text{def}}{=} \frac{\tilde{m}gL}{4} \begin{bmatrix} \mathbf{0}_{(\bar{N}-1) \times 1} \\ \bar{\mathbf{c}} \end{bmatrix}. \quad (31)$$

The gravitational energy over the entire beam is then given by adding Eq. 28 and Eq. 30, i.e.

$$\begin{aligned} P_g &= \mathbf{q}_L^T \mathbf{u} + \mathbf{q}_R^T \mathbf{u} \\ &= (\mathbf{q}_L + \mathbf{q}_R)^T \mathbf{u} \\ &= \mathbf{q}^T \mathbf{u}, \end{aligned} \quad (32)$$

where

$$\mathbf{q} \stackrel{\text{def}}{=} \mathbf{q}_L + \mathbf{q}_R. \quad (33)$$

5.4 Calculating the Total Potential Energy

The total potential energy is calculated by adding Eq. 25 and Eq. 32. Hence the total potential energy is

$$P = \frac{1}{2} \mathbf{u}^T \mathbf{W} \mathbf{u} + \mathbf{q}^T \mathbf{u}. \quad (34)$$

5.5 Constraints

The constraints of Eqs. 7-9 are straight forward to convert to the vector notation used in the numerical calculation. The constraints become

$$\mathbf{u}[1] = 0, \quad (35)$$

$$\mathbf{u}[\bar{N}] = u_p, \quad (36)$$

$$\mathbf{u}[N] = 0. \quad (37)$$

The constraint of Eq. 10 must be handled carefully: In the open intervals $(-\frac{L}{2}, 0)$ and $(0, \frac{L}{2})$, $u(x)$ is described

by the polynomials which interpolates $\{u[0], \dots, u[\bar{N}]\}$ and $\{u[\bar{N}], \dots, u[N]\}$, respectively. Please recall that the corresponding x -values are Chebyshev-distributed, which means that the interpolating polynomials are guaranteed to converge to the true $u(x)$ as \bar{N} and $N - \bar{N}$ increase. As $u(x)$ is described by polynomials in these open intervals, it meets C^∞ , and hence also C^2 , in the intervals.

At the left endpoint, $u(-\frac{L}{2})$ and its right-sided limit will be equal, because the limit is given by a polynomial which interpolates through the point $(-\frac{L}{2}, u(-\frac{L}{2}))$. Hence, $u(x)$ is at least C^0 at the left endpoint. As $u(x)$ is not defined for $x < -\frac{L}{2}$, derivatives of any order only exist on the right side of $x = -\frac{L}{2}$, where they are given by the polynomial interpolation. Hence, $u(x)$ is C^∞ at the left endpoint. A similar reasoning applies for the right endpoint as well, which concludes that also this endpoint is C^∞ .

At $x = 0$, two polynomials that both interpolates through the point $(0, u(0))$ meet. Hence, $u(0)$ and the limits from both sides will be equal, which means that $u(x)$ is C^0 at $x = 0$. However, there is no inherent effect that forces the two interpolating polynomials to have the same derivatives at $x = 0$. This is true for derivatives of any orders. Therefore, for $u(x)$ to be C^2 at $x = 0$, equal first- and second derivatives at both sides of $x = 0$ must be explicitly enforced as constraints, i.e.

$$\lim_{x \rightarrow 0^-} \frac{\partial u}{\partial x} = \lim_{x \rightarrow 0^+} \frac{\partial u}{\partial x}. \quad (38)$$

$$\lim_{x \rightarrow 0^-} \frac{\partial^2 u}{\partial x^2} = \lim_{x \rightarrow 0^+} \frac{\partial^2 u}{\partial x^2}. \quad (39)$$

In terms of the discrete-point representation of $u(x)$, Eq. 38 can be approximated by

$$\mathbf{D}_1[\bar{N}, :] \mathbf{u}[1 : \bar{N}] = \mathbf{D}_1[1, :] \mathbf{u}[\bar{N} : N], \quad (40)$$

where \mathbf{D}_1 is the Chebyshev first-derivative matrix of dimension $\bar{N} \times \bar{N}$. Eq. 40 can be written as

$$\mathbf{p}^T \mathbf{u} = 0, \quad (41)$$

where the vector \mathbf{p} is defined by

$$\mathbf{p}_L \stackrel{\text{def}}{=} [\mathbf{D}_1[\bar{N}, :] \quad \mathbf{0}_{1 \times (\bar{N}-1)}]^T, \quad (42)$$

$$\mathbf{p}_R \stackrel{\text{def}}{=} [\mathbf{0}_{1 \times (\bar{N}-1)} \quad \mathbf{D}_1[1, :]]^T, \quad (43)$$

$$\mathbf{p} \stackrel{\text{def}}{=} \mathbf{p}_L - \mathbf{p}_R. \quad (44)$$

A similar reasoning for the second derivatives gives

$$\mathbf{r}^T \mathbf{u} = 0, \quad (45)$$

where

$$\mathbf{r}_L \stackrel{\text{def}}{=} [\mathbf{D}_2[\bar{N}, :] \quad \mathbf{0}_{1 \times (\bar{N}-1)}]^T, \quad (46)$$

$$\mathbf{r}_R \stackrel{\text{def}}{=} [\mathbf{0}_{1 \times (\bar{N}-1)} \quad \mathbf{D}_2[1, :]]^T, \quad (47)$$

$$\mathbf{r} \stackrel{\text{def}}{=} \mathbf{r}_L - \mathbf{r}_R. \quad (48)$$

5.6 Minimization subject to Constraints

The total potential energy given by Eq. 34 should be minimized w.r.t. \mathbf{u} subject to the constraints given in Eqs. 35, 36, 37, 41, and 45. An intuitive approach is to use the five constraints to eliminate five elements from the vector \mathbf{u} , and then minimize for the remaining $N - 5$ elements of \mathbf{u} . This is indeed a decent and feasible approach. However, the elimination process will disturb the simple formulation of Eq. 34 and complicate the solution process somewhat. From the author's point of view, implementing the constraints using Lagrange multipliers is a simpler and more elegant approach that does not obscure the formulation of Eq. 34. Also, the Lagrange multipliers are themselves interesting variables, many with physical interpretations.

The Lagrange function of the constrained optimization problem is

$$\begin{aligned} L(\mathbf{u}, \lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5) = & \frac{1}{2} \mathbf{u}^T \mathbf{W} \mathbf{u} + \mathbf{q}^T \mathbf{u} \quad (49) \\ & - \lambda_1 \mathbf{u}[1] \\ & - \lambda_2 (\mathbf{u}[\bar{N}] - u_p) \\ & - \lambda_3 \mathbf{u}[N] \\ & - \lambda_4 \mathbf{p}^T \mathbf{u} \\ & - \lambda_5 \mathbf{r}^T \mathbf{u}. \end{aligned}$$

The expression of Eq. 49 is quadratic. Hence, it can be written on the form

$$L(\mathbf{v}) = \frac{1}{2} \mathbf{v}^T \mathbf{A} \mathbf{v} + \mathbf{b}^T \mathbf{v}, \quad (50)$$

where the vector \mathbf{v} is the vector \mathbf{u} augmented with the five Lagrange multipliers, i.e.

$$\mathbf{v} \stackrel{\text{def}}{=} \begin{bmatrix} \mathbf{u} \\ \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \end{bmatrix} \in \mathbb{R}^{N+5}. \quad (51)$$

Further, the matrix \mathbf{A} of Eq. 50 is the block matrix

$$\begin{aligned} \mathbf{A} \stackrel{\text{def}}{=} \begin{bmatrix} \mathbf{W} & -\mathbf{i}_1 & -\mathbf{i}_{\bar{N}} & -\mathbf{i}_N & -\mathbf{p} & -\mathbf{r} \\ -\mathbf{i}_1^T & 0 & 0 & 0 & 0 & 0 \\ -\mathbf{i}_{\bar{N}}^T & 0 & 0 & 0 & 0 & 0 \\ -\mathbf{i}_N^T & 0 & 0 & 0 & 0 & 0 \\ -\mathbf{p}^T & 0 & 0 & 0 & 0 & 0 \\ -\mathbf{r}^T & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (52) \\ \in \mathbb{R}^{[(N+5) \times (N+5)]}, \end{aligned}$$

where \mathbf{i}_j is the j th column of the $N \times N$ identity matrix. The vector \mathbf{b} of Eq. 50 is

$$\mathbf{b} \stackrel{\text{def}}{=} \begin{bmatrix} \mathbf{q} \\ 0 \\ u_p \\ 0 \\ 0 \\ 0 \end{bmatrix} \in \mathbb{R}^{N+5}. \quad (53)$$

The solution of the constrained optimization problem is the \mathbf{v} vector that gives a stationary point to the Lagrange function. The derivative of the Lagrange function is

$$\frac{dL}{d\mathbf{v}} = \mathbf{A} \mathbf{v} + \mathbf{b}. \quad (54)$$

Hence, the solution of the constrained optimization problem is the solution of the linear system with $N + 5$ equations and $N + 5$ unknowns

$$\mathbf{A} \mathbf{v} = -\mathbf{b}. \quad (55)$$

6 Results and Interpretations

The numerical calculations derived in this paper, which are based on the minimum total potential energy principle, have been compared to the formulas based on force- and moment balances. The two approaches give identical results for $\bar{N} \geq 5$, while they differ for lower values of \bar{N} . This conclusion is to be expected because the force- and moment-balances approach gives $u(x)$ as a fourth order polynomial, and it requires five interpolation points from the numerical calculations to give a fourth order polynomial.

Figure 4 presents a comparison of the two approaches for a steel tube with bending stiffness 124.1 kNm² and specific mass 8.257 kg/m. The length between the supports is 3.000 m and the vertical deflection of the piston is $u_p = -3.200 \times 10^{-2}$ m. These input values are from a steel tube tested in the physical bending stiffness rig. The physical test is presented in Jordal et al. (2017).

The first N elements of the \mathbf{v} vector found from Eq. 55 is \mathbf{u} , i.e. the nodes shown in Figure 4. The latter five

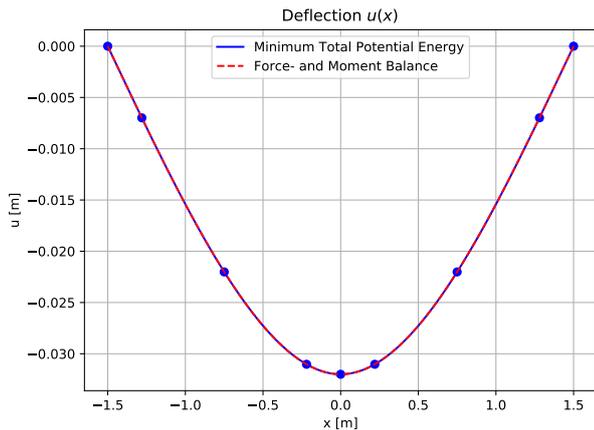


Figure 4. Comparison of $u(s)$ found from (i) the minimum total potential energy principle as derived in this paper, and (ii) from the approach based on force- and moment balances. The blue dots represent the values calculated by the former method, and the blue, solid line represents the polynomial interpolations between these points. $\bar{N} = 5$ is used in the example.

elements of \mathbf{v} are the five Lagrange multipliers defined in Eq. 49. The physical interpretation of each Lagrange multiplier is the derivative of the total potential energy, P , w.r.t. the constraint associated with the multiplier. Hence, λ_1 and λ_3 are the forces from the left and right beam supports, respectively, toward the beam, as shown in Figure 2. Similarly, λ_2 is the force from the piston toward the beam, i.e. the force F in the figure. Multiplier λ_4 can be shown to be the beam's bending moment at $x = 0$.

The last multiplier, λ_5 , expresses the total potential energy's sensitivity to discontinuities in the bending curvature at $x = 0$. To the author's knowledge, this formulation does not correspond to a well-known physical variable. λ_5 is a factor 10^{16} smaller than the other multipliers, measured in absolute value. Hence, λ_5 is assumed to be zero in the case of infinite arithmetic precision, which means that the corresponding constraint would have been met even if it was not enforced.

7 Conclusions and Further Work

The present paper derives numerical calculation of the beam equation based on the minimum total potential energy principle under the assumptions of linear material and small deflections. The calculation gives identical results as the derivation based on force- and moment balances which is commonly presented in the literature.

The work presented in this paper is the first milestone towards the author's final goal of modeling subsea power cables and umbilicals during bending stiffness tests. The future milestones are to include the effects of large deflections, nonlinear materials, and shear forces between the cable elements.

It is believed that the minimum total potential energy principle is more suitable for the future extensions than

force- and moment balances. It is further believed that it will be impossible to reach an analytical solution of the overall problem due to its complexity, which is the reason for using numerical mathematics.

References

- J. Hedlund. Modelling of viscoelastic dynamic bending stiffness for VIV analysis of submarine cables. In *Proceedings of the 9th International Conference on Insulated Power Cables (Jicable '15)*, 2015.
- L. Jordal, R. Slora, E. Vermeer, and M. Komperød. A novel bending stiffness rig for identification of subsea cables' and umbilicals' sensitivity to temperature under sinusoidal curvature oscillations. In *Proceedings of the Twenty-seventh International Ocean and Polar Engineering Conference (ISOPE 2017)*, San Francisco, California, USA, 2017.
- E. Kebabdzé. *Theoretical modelling of unbonded flexible pipe cross-sections*. PhD thesis, South Bank University, 2000.
- M. Komperød. A frequency-domain model of bitumen-coated armor wires in subsea power cables, umbilicals, and power umbilicals. In *Proceedings of the 35th International Conference on Ocean, Offshore and Arctic Engineering (OMAE 2016)*, Busan, South Korea, 2016a.
- M. Komperød. Analytical calculation of capacity for bitumen-coated armor wires in subsea cables and umbilicals. In *Proceedings of the Twenty-sixth International Ocean and Polar Engineering Conference (ISOPE 2016)*, Rhodes, Greece, 2016b.
- M. Komperød, J. I. Juvik, G. Evenset, R. Slora, and L. Jordal. Large-scale tests for identifying the nonlinear, temperature-sensitive, and frequency-sensitive bending stiffness of the NordLink cable. In *Proceedings of the 36th International Conference on Ocean, Offshore and Arctic Engineering (OMAE 2017)*, Trondheim, Norway, 2017.
- M. Lutchansky. Axial stress in armor wires of bent submarine cables. *Journal of Engineering Industry*, 91(3):687 – 693, 1969.
- P. Maioli. Bending stiffness of submarine cables. In *Proceedings of the 9th International Conference on Insulated Power Cables (Jicable '15)*, 2015.
- H. Martindale, S. Rossiter, T. Sheldrake, and R. Langdon. Improved dynamic structural modelling for subsea power cables with bitumen coated armour wires. In *Proceedings of the 36th International Conference on Ocean, Offshore and Arctic Engineering (OMAE 2017)*, Trondheim, Norway, 2017.
- H. Reid. Massachusetts Institute of Technology - Lecture notes 18.330 Introduction to numerical analysis, 2014.
- J. Tarnowski. Improved method of determining bending stiffness of underground cables. In *Proceedings of the 9th International Conference on Insulated Power Cables (Jicable '15)*, 2015.

Experimental Design and Modeling for Propylene Oxide - CO₂- Poly (Propylene Carbonate) Solutions

Umesh Pandey¹, Kai Arne Sætre², Jostein Mathiassen², Sara Ronasi², Siw Bodil Fredriksen², Carlos F. Pfeiffer¹

(mecha.umesh@gmail.com, kai.saetre@norner.no, jostein.mathiassen@norner.no, sara.ronasi@norner.no, siw.fredriksen@norner.no, carlos.pfeiffer@usn.no)

¹University College of Southeast Norway, Norway

²Norner Research, Norway

Abstract

In this research, experimental design was used to formulate the empirical models of viscosity and density of poly(propylene carbonate) (PPC), propylene oxide (PO), and carbon dioxide (CO₂) solutions by designing experiments at key values of the process variables; concentration of PPC between 0 to 34% (% w/w), temperature in the reactor between 50 to 75°C, and gas phase manometric CO₂ pressure between 20 to 40 bar. A bench scale reactor (2000 ml) comprising an external circulation loop equipped with in-line viscosity and density measurement devices was used to carry out the tests. The results show that the equilibrium viscosity and density of the solution increased with the concentration of PPC and decreased with the pressure and temperature in the reactor. The density model has $R_{adjusted}^2$ value close to unity indicating that the model can predict the variation in the density with very high accuracy. In comparison, the viscosity model has a lower $R_{adjusted}^2$ value indicating a need for additional experiments to improve the model. However, both empirical models predict the general trends of the density and viscosity characteristics in the selected range and can be used as a viable alternative to thermodynamic models.

Keywords: Carbon dioxide polymers, poly(propylene carbonate), experimental design, statistical modeling

1 Introduction

1.1 Background

With recent development in the field of polymer science, CO₂ as a feedstock in a synthesis of the biopolymer, poly(propylene carbonate) (PPC), has been viewed as an attractive alternative. The research community is further motivated to explore and optimise the copolymerization of CO₂ and propylene oxide (PO) to produce PPC, as, CO₂ is relatively cheaper, abundant, and environmentally friendly in comparison to petroleum-based feedstocks (Arakawa *et al.*, 2001). Till date, the studies have specifically focused on understanding the effect of catalyst on the selectivity of the copolymer product and optimizing copolymerization

in a batch reactor (Narang *et al.*, 2016; Meng *et al.*, 2016; Pan *et al.*, 2014; Wang *et al.*, 2012; Qin *et al.*, 2003). However, understanding of flow characteristics (density and viscosity) of product streams with the variation in process parameters (pressure, temperature, and concentrations) is crucial for optimal product yield and economic opportunity for industrial synthesis of PPC. Prior attempts of formulating analytical models of the density and viscosity based on the thermodynamic theories by this research group at Norner Research were unsuccessful due to the complexity of polymer characteristics thereby causing difficulties in obtaining precise measurements of the concentration of the phases involved. In this research, the design of experiments (DOE) was used to plan, conduct and analyse the experiments and then formulate a statistical model of viscosity and density of PO-CO₂-PPC solutions.

1.2 Experimental Design

Experimental design is a systematic approach to perform experiments and discover the effect of controllable factors on the response variables. Experimental factors are variables which have a direct impact on the response variables. They assume a set of discrete values known as levels of the factors. There are diverse types of experimental design based on the objective, levels, resolution, and the total number of runs. The general full factorial design is one of the key design where tests are performed at all the possible combinations of levels of experimental factors. It is a suitable approach to discover the general behaviour of the processes, formulate an empirical model and validate analytical models (Davim, 2016; Montgomery, 2001; Montgomery *et al.*, 2007).

In this study, a general full factorial design of experiments with three factors (initial concentration of PPC, the temperature in the reactor and gas phase CO₂ pressure in the reactor) at three levels is used to study the viscosity and the density of the solution and formulate second-order empirical models with 9 parameters.

2 Experimental

2.1 Chemicals

Table 1 is a list of the chemicals used in the tests. PPC was purchased from Empower Materials with the product name QPAC® 40 and a range of molecular weights between 100-300 kmol/g. PO with a purity of 99.5% w/w was supplied by Acros Organics. Acetone and heptane from Sigma Aldrich were used to wash up the set-up.

Table 1. List of chemicals.

Chemical	Symbol
Propylene oxide	PO
Heptane	C ₇ H ₁₆
Acetone	(CH ₃) ₂ CO
Carbon dioxide	CO ₂
Poly (propylene carbonate)	PPC

2.2 Set-up

Figure 1 shows a bench-scale batch reactor set-up used in the tests. The reactor was equipped with temperature control, stirrer to maintain homogeneity of the chemical mixture and inlet for pressure controlled CO₂ feeding. A pump was used to ensure continuous flow in the circulation unit (CU). The unit was equipped with inline viscometer to measure viscosity and density of the solution. Furthermore, the unit was equipped with a flow meter to measure flow in the circulation unit.

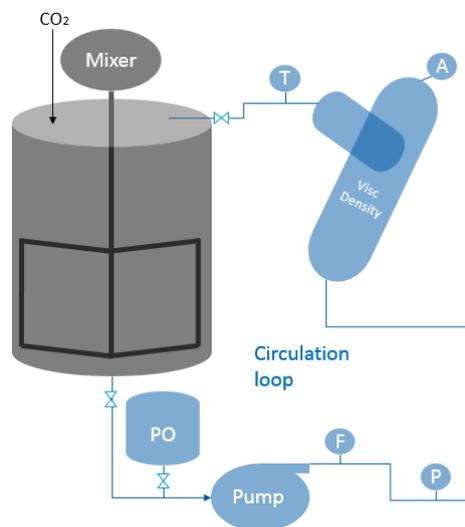


Figure 1. A bench scale reactor equipped with in-line measurement devices for density and viscosity characterisation of the reactor solution.

2.3 Design Space

Table 2 shows the design space of factorial experimental design. The design space includes the relevant operating window for the polymer synthesis.

Table 2. Design space for the experimental design

Factors	Max	Min
Initial concentration of PPC (C) (% w/w)	34.0	0.0
Temperature in the reactor (T) (°C)	75.0	50.0
Gas phase CO ₂ pressure (P) (bar)	45.0	20.0

2.4 Factorial Experimental Design

Table 3 is a full factorial design implemented to study the flow characteristics of the PO-CO₂-PPC solutions.

Table 3. Full factorial experimental design.

Test runs	C [% w/w]	T [°C]	P [bar]
1	0	50.0	20.0
2	0	50.0	32.5 (30.0)
3	0	50.0	45.0 (40.0)
4	0	62.5	20.0
5	0	62.5	32.5
6	0	62.5	45.0
7	0	75.0	20.0
8	0	75.0	32.5
9	0	75.0	45.0
10	17	50.0	20.0
11	17	50.0	32.5
12	17	50.0	45.0
13	17	62.5	20.0
14	17	62.5	32.5
15	17	62.5	45.0
16	17	75.0	20.0
17	17	75.0	32.5
18	17	75.0	45.0
19	34	50.0	20.0
20	34	50.0	32.5
21	34	50.0	45.0
22	34	62.5	20.0
23	34	62.5	32.5
24	34	62.5	45.0
25	34	75.0	20.0
26	34	75.0	32.5
27	34	75.0	45.0

2.5 Procedures

The test runs were separated into three distinct groups based on the concentration of PPC (0 % w/w, 17 % w/w, 34 % w/w). Before performing tests of each group, the set-up was cleaned with heptane/acetone and dried with N₂ gas/ vacuum.

After the equipment was washed and dried, the reactor was loaded with the required amount of PPC.

The reactor was then assembled, and N₂ was purged out followed by feeding of PO into the reactor, stirring and increasing the reactor temperature to the set-point. Viscosity and density responses were measured at equilibrium (steady state) in distinct combinations of gas-phase CO₂ pressure and temperature. At the end of tests of each group, the set-up was cleaned and prepared as described above for the next loading of PPC with the accompanying group of tests.

3 Results and Discussions

3.1 Summary of Experimental Observations

Table 4 shows an overview of viscosity (μ) and density (ρ) responses of the solution after the complete test runs (according to full factorial design, test runs 1-27), preliminary tests at 0% PPC concentrations (test runs 28-30) and the replication tests at 17% PPC concentration (test 31-38). The experimental factor values were controlled according to the factorial design, but there were minor deviations between design values and experimental values due to random noise.

Table 4. Viscosity (μ) and density (ρ) responses.

<i>Test runs</i>	<i>C</i> % w/w	<i>T</i> °C	<i>P</i> bar	ρ kg/m ³	μ cP
1	0.0	50.0	19.7	811.0	0.6
2	0.0	50.0	29.6	816.0	0.6
3	0.0	50.0	40.0	822.0	0.8
4	0.0	62.5	20.0	791.0	0.7
5	0.0	62.5	32.5	797.0	0.6
6	0.0	62.5	40.0	799.0	0.7
7	0.0	75.0	20.0	772.0	0.7
8	0.0	75.0	32.5	774.0	0.4
9	0.0	75.0	40.0	776.0	0.8
10	17.0	50.0	20.0	862.0	4.5
11	17.0	50.7	32.5	858.0	2.0
12	17.0	50.0	38.5	883.0	2.0
13	17.0	62.5	20.0	846.0	4.5
14	17.0	62.5	30.0	845.0	3.1
15	17.0	62.5	40.0	846.0	1.2
16	17.0	75.0	20.0	830.0	4.7
17	17.0	75.0	30.0	828.0	3.3
18	17.0	75.0	40.0	827.0	2.3
19	34.0	50.0	20.3	916.0	80.5
20	34.0	50.0	32.5	918.0	44.0
21	34.0	50.0	45.0	923.0	180.0
22	34.0	62.5	20.7	900.0	80.0
23	34.0	62.5	32.5	899.0	48.0
24	34.0	62.0	45.6	888.0	35.0
25	34.0	75.0	20.0	882.0	76.0

<i>Test runs</i>	<i>C</i> % w/w	<i>T</i> °C	<i>P</i> bar	ρ kg/m ³	μ cP
26	34.0	75.0	32.5	883.0	50.0
27	34.0	75.0	45.0	881.0	34.0
28	0	50.0	20.0	810.0	0.6
29	0	50.0	32.5	817.0	0.6
30	0	50.0	40.0	820.0	0.72
31	17%	50	20	856	3.4
32	17%	50	19	858	4.6
33	17%	50	29.3	860	5
34	17%	62.5	19.6	840	4.5
35	17%	62.5	29.6	845	3.3
36	17%	62.5	39.3	856	3.4
37	17%	75	19.5	825	4.5
38	17%	75	31	829	3.3

As regression coefficients of statistical model formulated using coded factors can be directly compared to determine the relative significance of coefficients, factors values were transformed to coded value using Equation (1).

$$X_{Coded} = 2 \frac{X_{Actual} - X_{Min}}{X_{Max} - X_{Min}} - 1 \quad (1)$$

Where,

- X_{Coded} Factor values in the coded form
- X_{Actual} Actual factor values in Table 4
- X_{Min} Minimum factor values in Table 2
- X_{Max} Maximum factor values in Table 2

3.2 Density Model

Equation (2) is a density model formulated using coded factor values.

$$\rho = 846.5 + 51.5C - 19.6T + 2.8P + 0.62C^2 + 1.2T^2 + 0.9P^2 + 1.6TC + 3.3TP + 2.6PC \quad (2)$$

In statistical modelling, analysis of variance (ANOVA) and goodness of fit is used to test the significance of a model. The goodness of fit (R^2) measures the error in fitting experimental data in the regression model. Its values are between 0 and 1, and higher values mean better fit and vice versa. However, for a certain set of data used in the formulation of a regression model, R^2 value is entirely model dependent and increases with the inclusion of additional parameters even if the parameters are insignificant (overestimation). To deal with this problem, $R^2_{adjusted}$ is used. It measures the percentage of variation due to the significant parameters and decreases with the inclusion of additional insignificant model parameters. Together with the determination of R^2 value, ANOVA

is performed to test that the selected parameters in the model are jointly responsible for the variation in response variable (Montgomery, 2001; Montgomery *et al.*, 2007).

Table 5 presents the analysis of variance (ANOVA) and summary of statistics in the formulation of density model. As Significance F-test value (P value of F test) is 0.0, it can be concluded with 95% confidence that the model parameters in the density model are the primary cause of variation in density responses of the mixture. $R_{adjusted}^2$ value is closed to unity, which indicates that the error in fitting experimental density responses in the regression model is negligible and the predicted and measured density responses are statistically identical. This is also evident in the root mean squared prediction error $\sigma = 4.51 \text{ kg/m}^3$ which is merely 0.53% of the mean density response. Based on ANOVA and $R_{adjusted}^2$ value it can be statistically concluded that the model can explain most of the variations in experimental density responses and it can be used to predict the density responses in the vicinity of the design space.

Table 5. ANOVA analysis and summary of statistics of density model.

Description	Values
Multiple R	0.995
R-squared	0.990
Adjusted R-squared	0.987
F-value	323.9
Significance-F (P-value)	0.0
Variance (σ^2)	20.4

Figure 2 shows a plot of density residuals and Figure 3 shows the normal probability plot of density residuals. Both plot are symmetrical about X-axis except few outliers indicating that model predicted density values and measured density values are statistically identical. However, the residuals are relatively higher at higher PPC concentrations.

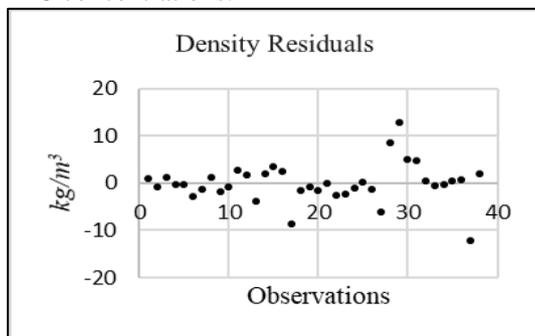


Figure 2. Residual plots of density model.

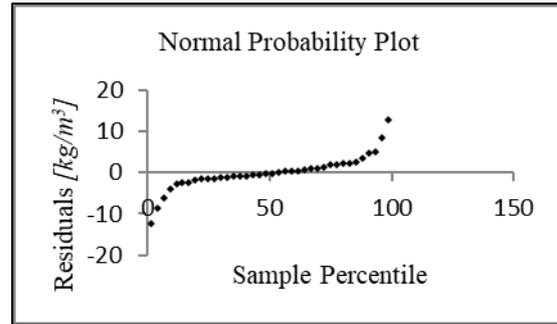


Figure 3. Normal probability plot of density model

Table 6 presents a significance test (*t-test*) of regression coefficients in the density model. P-values of regression coefficients in the *t-test* shows that linear terms (*C*, *T*, *P*) and a second-order term (*T·P*) have a significant impact on the density responses while other second-order terms do not contribute significantly in the model.

Table 6. *t-test* of coefficients in density model.

Coefficients	<i>t-stat</i>	<i>P-value</i>
<i>C</i>	48.4	0.0
<i>T</i>	-20.5	0.0
<i>P</i>	2.2	0.0
<i>C</i> ²	0.4	0.7
<i>T</i> ²	0.7	0.5
<i>P</i> ²	-0.4	0.7
<i>T·C</i>	1.3	0.2
<i>T·P</i>	-2.5	0.0
<i>P·C</i>	-1.8	0.1

Equation (3) is reduced density model considering only significant terms with 95% confidence (P value less than 0.05) in the model.

$$\rho = 846.5 + 51.5C - 19.6T + 2.8P - 3.3TP \quad (3)$$

The reduced density model shows that the density varies linearly with the factors. Among the factors, density increases linearly with the concentration of PPC and density decreases with an increase in the temperature. However, pressure contribution to the variation in density responses is relatively low in comparison to other factors.

3.3 Viscosity Model

Equation (4) is the viscosity regression model formulated using coded factor values.

$$\begin{aligned} \mu = & -5.9 + 32.5C - 9.5T + 3.67P + 29.7C^2 \\ & + 6.6T^2 + 13.3P^2 \\ & - 10.6TC - 13.8TP \\ & - 0.8PC \end{aligned} \quad (4)$$

Table 7 presents ANOVA and summary of statistics in the formulation of viscosity model. Similar to the density model, Significance F-test value is 0.0, and it can be concluded with 95% confidence that the model

parameters in the viscosity model are the primary cause in a variation of viscosity responses of the mixture. On the other hand, Adjusted R-squared value (0.710) is relatively lower, and the root mean square prediction error ($\sigma = 19.3 \text{ cP}$) is considerably high in comparison to most of the viscosity responses (all viscosity responses at 17% PPC and 0% PPC are equal or lower than 5.0 cP). In order to formulate model with better fit and lower prediction error, additional tests with smaller PPC concentration changes at higher PPC concentration is essential as the residuals are substantially higher at higher PPC concentrations. However, the model can still sufficiently explain and predict the general trends of the viscosity responses of the mixture in the selected operating range.

Table 7. ANOVA and summary of statistics of viscosity model.

Description	Values
Multiple R	0.884
R-squared	0.781
Adjusted R-squared	0.710
F-value	11.10
Significance-F (P-value)	0.00
Variance (σ^2)	372.8

Figure 4 shows a plot of viscosity residuals and Figure 5 shows the normal probability plot of viscosity residuals. Both plots consist of outliers with higher PPC concentration indicating that the model is unable to represent viscosity responses at higher PPC concentrations.

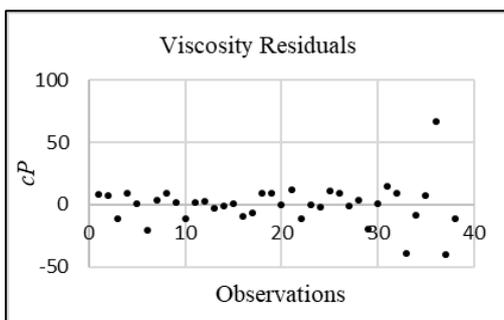


Figure 4. Residual plot of viscosity model.

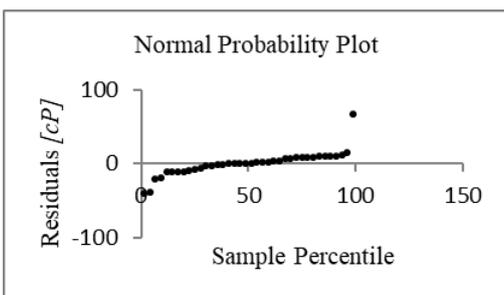


Figure 5. Normal probability plot of viscosity residuals.

Table 8 presents a significance test (t-test) of regression coefficients in the viscosity model. Linear terms (C, T), as well as second order term (C^2 , T·C, T·P), have a significant impact (95% confidence) on the viscosity responses with P-value less than 0.05. The remaining terms (P , T^2 , and P^2) do not contribute significantly to the model.

Table 8. t-test of coefficients in viscosity model.

Coefficients	t-stat	P-value
C	7.2	0.0
T	-2.3	0.0
P	0.7	0.5+
C^2	4.5	0.0
T^2	1.0	0.3
P^2	1.5	0.1
T·C	-2.1	0.0
T·P	-2.5	0.0
P·C	-0.1	0.9

Equation (3) is a reduced density model excluding insignificant terms with 95% confidence in Equation (4).

$$\mu = -5.9 + 32.5C - 9.5T + 29.7C^2 + 13.3P^2 - 10.6TC - 13.8TP \quad (5)$$

The reduced viscosity model shows that the viscosity primarily depends on the concentration of PPC in the mixture. It increases quadratically with the concentration of PPC and decreases with temperature in the reactor. On the other hand, viscosity changes quadratically with the pressure at constant temperature and concentrations. Besides, the pressure-temperature combination has an inverse effect on the viscosity response of the mixture.

4 Conclusions

The full factorial design of experiments is an efficient approach to formulate empirical models of the viscosity and density of a mixture of PPC, PO and CO_2 under varying concentration of PPC, temperature in the reactor and gas phase CO_2 pressure. The reduced empirical/regression model of density obtained from statistical analysis is highly efficient in the prediction of density response of the mixture in the design space. The density decreases insignificantly with increase in the gas phase CO_2 pressure at the higher temperature and increases at low temperatures due to second order effect of (TP), and increases linearly with PPC concentrations. On the other hand, the viscosity of the mixture predominantly depends on the concentration of PPC in the mixture. The viscosity increases with an increase in the PPC concentration and decreases with an increase in the temperature of the reactor. The residual analysis of

viscosity shows that more experiments are required to predict the impact of the selected factors on the viscosity to obtain lower root mean squared prediction error ($\sigma < 19.3 \text{ cP}$). However, both models explain the impact of pressure, concentration, and temperature in the viscosity and density responses of CO₂-PO-PPC solutions in a relevant process operation window and will be useful in the process development in the future.

Acknowledgement

The Research Council of Norway is gratefully acknowledged for the support under grant no 228157/WFI to this project.

References

- H. Arakawa, M. Aresta, J. N. Armor, M. A. Barteau, E. J. Beckman, A. T. Bell, J. E. Bercaw, C. Creutz, E. Dinjus, D. A. Dixon, K. Domen, D. L. DuBois, J. Eckert, E. Fujita, D. H. Gibson, W. A. Goddard, D. W. Goodman, J. Keller, G. J. Kubas, H. H. Kung, J. E. Lyons, L. E. Manzer, T. J. Marks, K. Morokuma, K. M. Nicholas, R. Periana, L. Que, J. R. Nielson, W. M. H. Sachtler, L. D. Schmidt, A. Sen, G. A. Somorjai, P. C. Stair, B. R. Stults, and W. Tumas. Catalysis Research of Relevance to Carbon Management: Progress, Challenges and Opportunities. *Chemical Reviews*, 101(4):953-956, 2001. doi: 10.1021/cr000018s
- J. P. Davim. *Design of Experiments in Production Engineering*. Springer. 2016. doi: 10.1007/978-3-319-23838-8
- Q. Meng, R. Cheng, J. Li, T. Wang, B. Liu. Copolymerization of CO₂ and propylene oxide using ZNGA/DMC composite catalyst. *Journal of CO₂ Utilization*, 16:86-96, 2016. doi: 10.1016/j.jcou.2016.06.011
- D. C. Montgomery. *Design and analysis of experiments*, John Wiley and Sons. 2001. doi: 10.1002/qre.458
- D. C. Montgomery and G. C. Runger. *Applied statistics and probability for engineers*. John Wiley and Sons. 2007.
- S. Narang, D. Berek, S. N. Upadhyay and R. Mehta. Effect of electron density on the catalysts for copolymerization of propylene oxide and CO₂. *Journal of Polymer Research*, 23:96, 2016. doi: 10.1007/s10965-016-0994-5
- X. Pan, Z. Liu, R. Cheng, D. Jin, X. He, and B. Liu. Experimental and theoretical studies on CO₂ and propylene oxide (PO) copolymerization catalyzed by ZnEt₂-glycerine-Y(CCl₃COO)₃ ternary catalyst. *Journal of Organometallic Chemistry*, 753:63-71, 2014. doi: 10.1016/j.jorganchem.2013.12.001
- Z. Qin, C. M. Thomas, S. Lee, and G. W. Coates. Cobalt-Based Complexes for the Copolymerization of Propylene Oxide and CO₂: Active and Selective Catalysts for Polycarbonate Synthesis. *Angewandte Chemie International Edition*, 42:5484-5487. doi: 10.1002/anie.200352605
- Z. Wang, Z. Bu, T. Cao, T. Ren, L. Yang, and W. Li. A novel and recyclable catalytic system for propylene carbonate synthesis from propylene oxide and CO₂. *Polyhedron*, 23(1):86-89, 2012. doi: 10.1016/j.poly.2011.07.002.

Study of Effectiveness of Operator Training Simulators in the Oil and Gas Industry

Kallakuri Ravikanth¹, Dr. P.C. Bahuguna², Donald C. Glaser³, Dr. Sanjay Shivalkar⁴

¹Research Scholar- PhD (Oil & Gas Management), University of Petroleum and Energy Studies, Dehradun, India & Training Specialist, Dolphin Energy Limited, Doha, State of Qatar. E-Mail: foralldata@hotmail.com ² Associate Professor, University of Petroleum and Energy Studies, Dehradun, India. E-Mail: bahuguna@ddn.upes.ac.in

³President, Simulation Solutions Inc., New Jersey, USA. E-Mail: dglaser@simulation-solutions.com ⁴Senior Training Specialist, Dolphin Energy Limited, Doha, State of Qatar. E-Mail: mitayash@hotmail.com

Abstract

Control room operators remotely monitor the process data coming from the field, guide field operators and take corrective actions using highly automated and complex distributed control systems. Operators have to intervene during abnormal conditions in the process and bring operations to safe conditions. Human errors remains the top contributor for industrial losses and the challenge is to prevent or address them by training and competency development. Operator training simulators are widely used in the industry to provide hands-on training to operators. Carefully analyzing simulator training effectiveness by predefining standard evaluation models in advance is one method to address the success of the training. Simulator training provides an understanding of the different process logical relationships in order to be able to accurately identify and quickly respond to problems before they escalate. In the current low oil price environment, organizations are facing number of challenges in developing exceptional workforce to sustain the business. The findings of this study re-emphasizes the need to develop training evaluation models integrated with the simulator configuration to effectively analyze the desired outcomes of each training session. This might further contribute to the reduction of operational errors in the hydrocarbon industry.

Key words: Control room operators, Human errors, Simulation training, Operator training simulator, Training effectiveness.

1. Introduction

The oil and gas industry is meeting the world's growing energy needs in an environmentally and socially responsible way in partnership with different governments and local communities as per

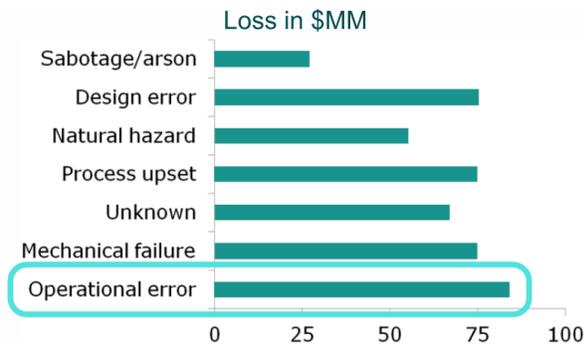
international, local and environmental regulations (British Petroleum Outlook, 2017). An expanding population, economic growth, new technology development and scope of regulations transforming the energy landscape of the world. Industrial growth, retiring and ageing workforce, an influx of new workers mean that the organizations must get new employees trained and competent as quickly as possible to sustain in the current low price environment.

1.1 Human errors in process control

Process plant operators control, monitor, operate and maintain the plant machinery and equipment during the production and different unit operations. They are able to navigate through piles of DCS (Distributed Control Systems) graphic pages, monitor hundreds of process parameters, make corrective actions and handle abnormal conditions. The ever increasing automation in the industry leading to more sophisticated processes that are more complex for the operators to understand. Consequently, the more sophisticated processes have a tendency to reduce human reliability (Wiener et.al, 1989). The implementation of new technology and automation brings changes to the interface design in control rooms and in the field. This brings to fore possible acclimatization issues faced by workers who may have a high level of familiarity with legacy systems from their experience.

The Center for Chemical Process Safety (CCPS) of American Institute of Chemical Engineers (AIChE), observed that in many incidents, the employees are not fully prepared for a major emergency or incident at the facility and they are not having any formal training on the plant control systems (www.hydrocarbonprocessing.com). The Marsh/McLennan compilation report for business impact shown in **Figure 1**. shows the losses in

hydrocarbon sector breaking up into different reasons. It points out that, over 28% are from operational errors and about 40% of abnormal operations are caused by human errors.



Source: Chemical Safety Topical Committee sponsored by the Department of Energy (DOE)

Figure 1. Causes & Losses in the Industry

The “abnormal” situations, in particular during the start-up and turn-around, or unscheduled shut-down, are subject to a very substantial part of the significant incidents occurring in many plants. The recurring accidents in the process industry are due to lack of learning from earlier incidents and applying the lessons learnt from the accidental data and reports.

1.2 Role of operator training simulator

Simulators are used for instruction and training in areas such as commercial and military aviation, battle field management, chemical industries, building construction and first responder agencies. They provide a safe alternative in replicating the high-risk, real-world training scenarios. Real world renditions of these situations are too costly and dangerous to provide opportunities for staff to repeatedly practice in the high stress workplace situations (Tichon et al., 2003).

Training simulators are used in the oil and gas industries to train the operators for both routine operations, startup, shutdown and abnormal situations related to the process. The benefits are repeated training on rare events, process upsets, validate or test new procedures, practice startup etc., which in turn increases confidence, enhances their effectiveness and decreases work-related stress. The difference between the operator response with and without appropriate training compiled is in **Figure 2**. It shows that the response skills are dropping with time after the initial training and practice, typically during the commissioning of the units (www.invensys.com). Response skills of the operators can be sustained with proper training of the

simulated scenarios and practice them repeatedly using a suitable training simulator.

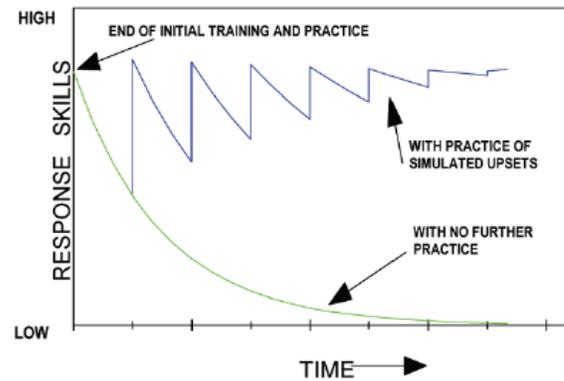


Figure 2. Operator response with and without appropriate training

The confidence in simulators might be due to the fact that most research on their effectiveness has been based on the subjective evaluation of trainees rather than on objective performance data (Salas-2006). The overall goal of training via simulation is to build confidence in staff in their own ability to perform under adverse conditions (Stetz et al., 2006). Due to the advent of modern automation and control systems, operators have less autonomy because they are driven by routine equipment operations, performance and further operators have less opportunity to experience abnormal situations and those situations can be more complex than is the case for manual operation (Hans-2015). A major perception among the managers is that a simulator is a luxury that cannot be justified in financially hard times.

To know that the training is effective, the program needs clear evaluation criteria and process for evaluating the impact on the performance of specific individuals by testing and observing the feedback using defined methodology (Peter-1997). With many similarities to the military, medical and aviation industry, the performance and effectiveness research in these sectors can be applied to the oil and gas industry process control (Kindervarter-2014). The evaluation of training effectiveness is the final stage of developing an effective simulation training program. Process training specialists did not know the impact of their simulator training programs on the effectiveness of individuals or operations teams (Peter-1997) as most of them are not formally trained on evaluation techniques and behavioral based methods.

2. Summary of Literature Review

Training is defined as the planned learning experience that trains the employees how to perform the current and future jobs. Workplace learning improve the profitability of different departments while cultivating more positive attitudes toward profit orientation across an organization. Mostly the training is focused on what needs to be known, rather than the application of skills. Further technological support is one of the most critical support factors to the transfer of learning (Yamnill-2001). The objectives of training determine the most appropriate criteria for assessing the effectiveness of training (Winfred et.al, 2003). The trainee's capabilities, personality, motivational factors, previous work experience will affect the learning process. Training of non-technical skills may help reduce the risk of human error and trained personnel are more prepared to react in case of an emergency situation. The main goal of training is to prepare participants for the tasks they are going to perform on their jobs (Barnard et.al, 2001).

Learning is the process of acquiring the ability to respond adequately to a situation which may or may not have been previously encountered. This means it is a change that occurs in our behaviour (more specifically in our work behaviour) as a result of some experience or practice. The training must focus on this change in behaviour, which gets reflected in performance (Raja-2010) Therefore any training (technical or behavioral) must focus on making the experience a learning one rather than just an enjoyable event. They must be motivated and willing to engage completely in the learning process through a "reflective self-discovery" in order to recognize weaknesses in decision-making and clinical reasoning skills and to be humble enough to correct those weaknesses with the guidance of the instructor (S. Yamnill-2001).

Evaluation is considered as an ongoing process, rather than done only at certain intervals. A number of barriers exist that prevent organizations from evaluating operators at various levels. The three major reasons to evaluate training programs are i) to improve the program, ii) to maximize transfer of learning and achieving subsequent organizational goals and iii) to demonstrate the value of training to the organization, which is effectiveness. In addition to monetary benefits, most training programs will have intangible, non-monetary benefits and the key challenge is to monitor and identify them. When the trainee experiences success or a sense of mastery, it translates into being less aroused physiologically, less distracted and more likely to focus on the task.

Variable-priority training led to better multitasking performance, and a trend for a reduction in the automation complacency effect (Raja-2010).

2.1 Effectiveness of OTS Training

Measuring effectiveness is the process of determining whether or not the desired results have been produced. Opinion-based surveys, questionnaires, ratings and checklists are best used in combination with objective measurements (e.g., time, speed, error) of trainee (learning) performance. There is significant change in seafarer attitude, skill level and knowledge due to the usage of simulation training in the shipping industry for different tasks in operations and Kirkpatrick's model is most suitable model for measuring the simulator training effectiveness (Surender-2015). A simulator does not train; it is the way the simulator is used that yields the benefit and the desired results. What is more important than the simulator technology is that how educational methodology is applied and whether it increases training effectiveness significantly or not (Drown-1995).

The simulation literature lacks significant research how the different elements of simulation training can be used to create a learning environment that encourages active learning and engagement in the work place. The training evaluation and effectiveness on the individuals or across operations teams is not studied in the perspective of an operator across the industry using available models. With many similarities with the military, medical, transport and aviation industries, the simulation training effectiveness research in these sectors can be applied in the oil & gas industry.

3. Research Methodology and Design

For the purpose of training assessment, the cause-and-effect relationships of interest are those between the process, learning outcome and its impact on training. In these relationships, the process variables (e.g., training methods and mediums used) are indicators of the outcomes, for example, knowledge gained among trainees, learning from incidents and competency etc. The key to identify the essential elements of effective training lies in understanding the correlation of these variables with the intended impact of training (Cohen & Colligan-1998).

The research objective is to measure the effectiveness of simulation training using proven methods from the literature and apply the methodology to some training sessions in the industry to validate the findings. Based on the literature review and subsequent gap and factor analysis, different

research questions are formulated as, what is the effectiveness of the operator training using simulators, how to evaluate the effectiveness and really is there a change in knowledge, attitude and skill levels after training and the knowledge acquired being used in work place. The first step to show the value of an training program is to create a program that has organizational value. Based on the review of literature and study of different models Donald Kirkpatrick's Four Level Training Evaluation Model, shown in **Figure 3**, is adapted as the base model. Kirkpatrick's framework consists of four levels of evaluation. Reaction (Level 1) is a measure of how participants react to the training program and it is a measure of customer satisfaction. Learning (Level 2) is measuring the knowledge gained during the program. Job behavior (Level 3) is measuring how well the participant applies the new knowledge or skills back on the job. This level of evaluation is important as it addresses the issue of training transfer. Level 3 evaluations often show that even though learning took place (Level 2), the skills are seldom fully applied back on the job (Robinson, 1998).



Figure 3. Kirkpatrick's Training Evaluation Model (<https://www.kirkpatrickpartners.com/Our-Philosophy/The-Kirkpatrick-Model>).

Conducting a Level 3 evaluation can help uncover the reasons that participants do not apply the new skills on the job. Results (Level 4) reflects the evaluation of training's impact on the organization's business results. At this level of evaluation, questions regarding improvement on the organization are answered. The model also assumes that each level is important and that all levels are interlinked. Most importantly organizations change in many ways, and behaviors and results change depending on these as well as on training.

Survey method is used as the primary data gathering tool by using structured questionnaire. Different questionnaires are prepared based on the four levels of Kirkpatrick's evaluation model and the simulator training participants are asked to respond accordingly. The questions are validated by peer review and focused group discussion between five different training specialists in the oil and gas

industry. The participants experience is ranging from 10-30 years with an age bracket of 30 to 56 years. Some of the questions have multiple answers and participants are asked to choose what is relevant to them. The central tendency is analyzed to derive the results (P.A. Bishop -2015). Some of the questions are marked using five point Likert's scale, consisting of the scale range is 1= strongly disagree, 2= disagree, 3= neither agree nor disagree, 4= agree and 5= strongly agree with the same weighted score. The weighted mean score of the data is calculated as below.

$$\frac{[(\text{number of people who selected response 1}) * (\text{weighting of response 1}) + (\text{number of people who selected response 2}) * (\text{weighting of response 2}) + \dots + (\text{Number of people who selected response n}) * (\text{weighting of response n})]}{(\text{total number of respondents})} \quad (1)$$

The survey is carried out during the months February-March 2018 at three sites consisting of around fifty (corresponding to an population of 100, precision level of $\pm 10\%$, (Smith, 1983)) respondents using two different training simulators supplied by two different OTS. The simulated processes include Heat Exchangers, Distillation, and Natural Gas processing units with the different processing equipment modelled along with the process shutdown systems similar to the operating plant. The demographic data is kept confidential along with the names of the companies as per the internal agreement. The results are analyzed and presented in the next section and the corresponding questions are presented in sub-sections.

4. Findings from the Survey

Training effectiveness will depend on a number of factors, such as time spent, motivation, learning efficacy and cognitive ability. Behavioral based standard assessment of operators is more important to mitigate the risk of human error and needs to be integrated with the modern simulators as a standard option, which will reduce the dependency on the instructor and same can be applied to all training sessions using the same simulator. The four levels of evaluation is summarized as below.

4.1 Reaction

Based on the survey results in **Figure 4**, we can conclude that prebriefing is necessary to give a brief idea of planned training sessions and the limitations of the training models used for training. Prebriefing

increases the confidence of trainees and it was overall beneficial to them.

4.1.1 Questionnaire

- Q1. Prebriefing increased my confidence and was beneficial to my overall learning experience
- Q2. Prebriefing is necessary to know what is planned in the exercise and the simulator limitations.
- Q3. I have lost interest during the prebriefing as we already know what we have to do during the exercise.
- Q4. Prebriefing is not required as we already know what is in the simulation training.
- Q5. We need to update the simulators continuously in tandem with the real plant to have updated model for training.

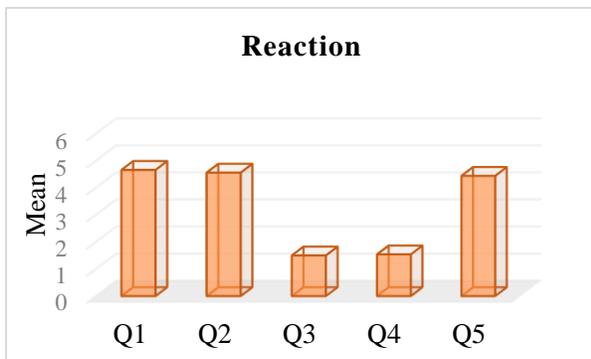


Figure 4. Reaction level

Some of the operators mentioned that prebriefing is not required, as they already know what is expected of them during the training. More than 90% of operators responded that the skills learned in the simulation training reflected in the actual happenings in the job and repetitive training helped them to handle the situations. Some of the low scores are related directly to the low level of experience of some operators responding to the survey. Majority of the respondents agreed that the OTS models to be updated simultaneously with the plant so that they can practice before the actual implementation of control loops in the OTS and refresh their skills.

4.2 Learning

All the questions related to learning are rated highly by the participants as captured in Figure 5. The simulator training directly related to their job responsibilities and participants are motivated to learn new things due to this similarity. Some of them mentioned that they were confident of troubleshooting and understanding of the complex

control loops better than the theoretical sessions in the class room. This concludes that the current training helped them to develop a better understanding of the process control and exposure to the critical operations. OTS also gave an opportunity to trainees to practice their decision making skills in a controlled atmosphere and gain confidence in executing the tasks.

4.2.1 Questionnaire

- Q1. I am better prepared & respond to changes in process conditions.
- Q2. I developed better process control and exposure to critical operations.
- Q3. I gain more confidence in trouble shooting.
- Q4. I can make critical interventions using complex control loops.
- Q5. I am more confident to contribute for process optimization.
- Q6. I had the opportunity to practice my decision making skills.
- Q7. I am more confident to prioritize operations.
- Q8. I learned from observing my peers and actively involved due to collective learning.



Figure 5. Learning level

The results from learning level emphasizes the trainee's motivation, behavioral strength and confidence in executing different tasks during the actual operations after the training. Interaction with other peers during the training helped them to use the OTS as an observational tool to gain confidence and motivated to make their workplace better.

4.3 Behavior

There is significant change in the behavior of the participants during and after the training as noted by the instructor and is further envisaged from the feedback. The results compiled in Figure 6. provided an opportunity to introspect individual performance.

4.3.1 Questionnaire

- Q1. Debriefing contributed to my learning and was valuable in helping me improve my judgment.
- Q2. Debriefing allowed me to verbalize my feelings before focusing on the next scenario.
- Q3. Debriefing provided opportunities to introspect my performance during simulation.
- Q4. I am more confident in communicating with my supervisors & engineers during critical operations.
- Q5. I am more confident to train & can be able to guide others in the control room operations
- Q6. Debriefing was a process of constructive evaluation of the simulation session.

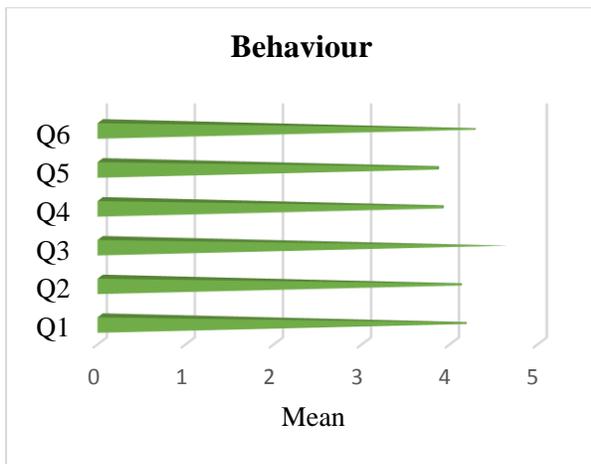


Figure 6. Behavior level

The discussions during simulation sessions helped them to communicate with their superiors in more efficient way and work as a team to further consolidate their learning. During critical operations they are more confident to act and handle the process upsets and might seek opinion with support staff to achieve desired results. The participants considered debriefing as a constructive evaluation of the simulation session. The confidence gained during the training is useful to guide others in the panel operations after practicing repeated sessions using a simulator.

4.4 Results

Most of the participants believe that simulation training helped them to understand the process in a better way. Further the practice sessions helped their ability to integrate the theoretical knowledge in to practical experience. The OTS investment can be supported by the reduction of human errors, production quality improvements, performance improvement and the most rated being the startup &

shutdown without incidents. The questionnaire and their corresponding answers **Figure 7.** are compiled as below.

4.4.1 Questionnaire

- Q1. This training can be used in all process operator trainings.
- Q2. The knowledge acquired is used in the workplace effectively.
- Q3. I am more confident in using evidence based practice and hands on approach in control room operations.
- Q4. I am more confident in carrying out interventions that foster operational safety.
- Q5. There is significant change in application of skills, attitude and confidence levels before and after simulation training.
- Q6. I am more confident in critical control loops, advanced process control and optimization due to experience gained from simulated operations.
- Q7. Simulator training is the most effective training for operators among all other types of training available in the Oil & Gas Industry.
- Q8. Simulation evaluation can be effectively used as a competency testing tool for promotions and career planning.

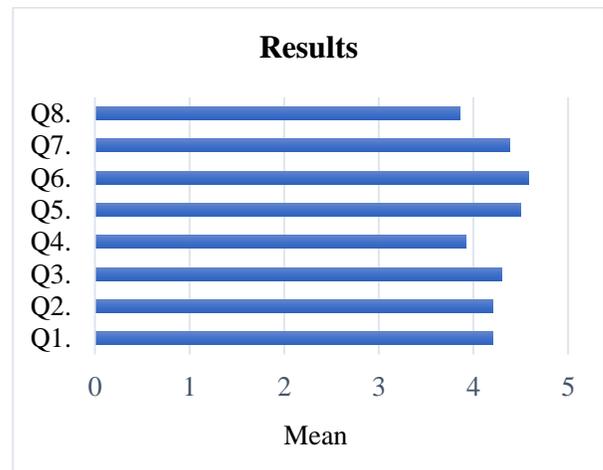


Figure.7. Results level

As per the results, the simulation sessions should be planned soon after the theoretical class room sessions to have effective learning and hands on application. Well planned simulation sessions using updated simulator voted in **Figure 8.** For maximum utilization of the OTS. The qualified instructors preferably from the same domain experience can help the trainees to run scenarios, which are actually happened in their process operations. They can be

taken from the incident reports and corrective action reports or similar case studies developed in the simulator, as an exercise preferably by the OTS supplier.

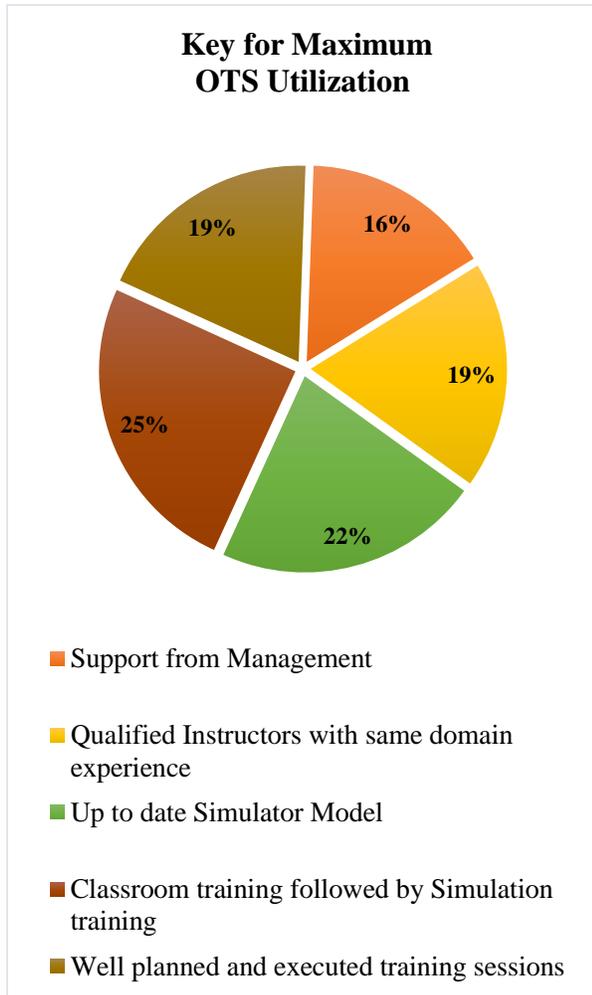


Figure 8. The Key for successful utilization of simulation training

The operators are better prepared to handle the process upsets, reduction of human errors **Figure 9**. By replicating the lessons learnt during actual panel operations and, safer startups. This will effectively justifies the OTS investment for the organization. The OTS training sessions helped participants to analyze the process better and improved their ability to take corrective actions during emergencies. This will further improve operational safety and helps them to gain more confidence in executing the complex tasks. Over the course of time one can see the reduction of human errors due to repeated training and more than 30% of response's from the participants validate the same.

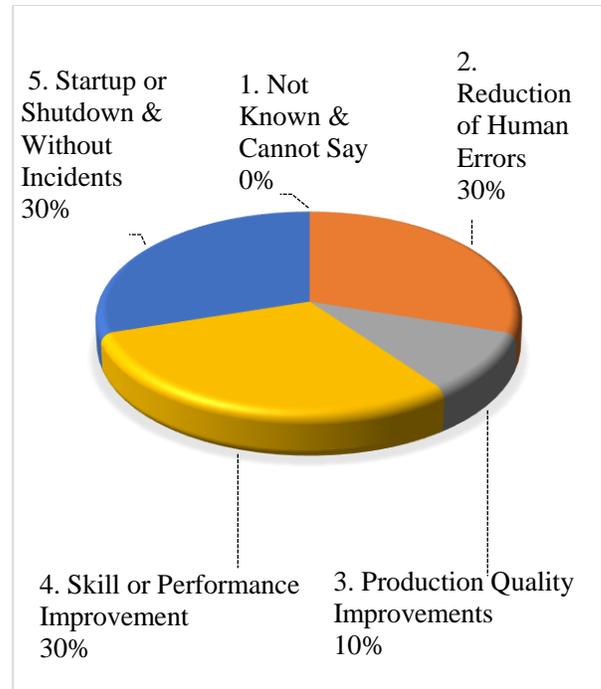


Figure 9. Supporting the OTS investment

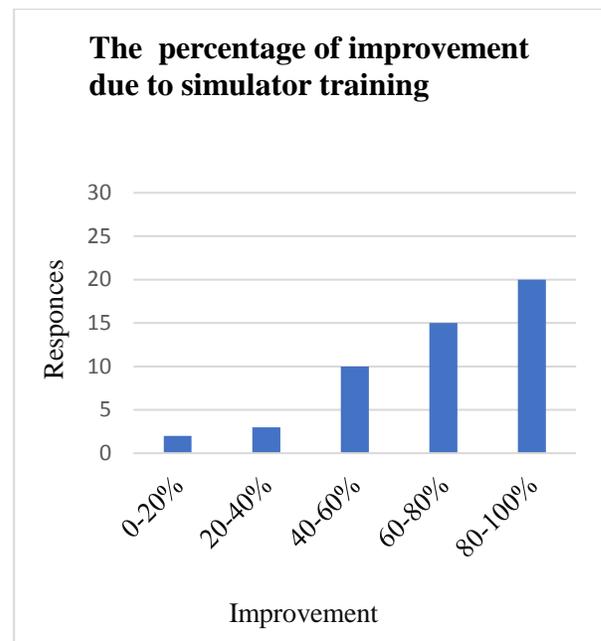


Figure 10. Results level findings

More than 80% of the participants, as per **Figure 10**, agrees that there is significant improvement in actual handling of operations after the training. The knowledge acquired from the training is effectively used at the work place and the OTS training is the most effective training across the industry to the operators.

Based on the findings, we can conclude that the simulator training enhance operator skills, behaviour , performance, helped to reduce downtime, faster and more efficient startup and finally improve overall competency of staff.

5. Limitations, Conclusions & Future Outlook

The survey is limited to a small group of participants and is relatively small size for proper statistical analysis. The results are analyzed using the data captured before and immediately after the training. Post training survey data after a certain period of time is not considered due to very low response for emails sent to the individuals. Further the analysis is not capturing the capability of the OTS process models in executing the tasks and the competency of instructors.

The simulator can be used to verify the procedures and they can then be optimized, rewritten to insist on critical points. The operators are confident in executing the complex tasks, leading to product quality, skill improvement and one might see a reduction in human errors due to the competency based approach. By integrating the training evaluation models with the OTS, the instructor can easily measure each and every session without being depend on questionnaires and infer the results. This methodology can optimize the trainings delivered and improve future sessions.

A combination of skilled instructors, immersive simulation, cloud technologies coupled with up-to-date high-fidelity simulators and pedagogical course material are essential for ensuring the best possible training for the new and experienced operators in the oil and gas industry. The operators have to take objective based decisions based on real time data, strive to contribute process changes, and needs to be competent for the ever changing new cyber security protocols. The instructor role is changing due to new virtual & augmented technologies, automated and objective based decisions. They also need to be trained to update with the technology, which is often missing due to various organizational reasons. There are number of challenges in the development of exceptional workforce to sustain the business. Staggered industrial growth, retiring and ageing workforce, an influx of new workers forcing the organizations to get new hires quickly trained and competent in the current low oil price environment and simulation training continues to be the best among the available trainings for operators and most cost effective way.

6. References

- ARC Insights-2007, T. Fiske, ARC Advisory Group, Maine. (www.arcweb.com/market-studies/)
- Bishop, R.L. Herron, 2015. Use and misuse of the Likert item responses and other ordinal measures.
- Down, D.F., & R. Mercer. "Applying marine simulation system to improve mariner professional development". PIANC, Port `95, TAMPA Florida, March 1995.
- E. Salas., 2006. Design, delivery and evaluation of training systems. Handbook of human factors and ergonomics. Hoboken, NJ: Wiley and Sons, 472–512.
- G. Tichon Jennifer & Guy M. Wallis, Stress training and simulator complexity, why sometimes more is less by, The School of Human Movement Studies, University of Queensland, Australia.
- Meshkati, International Journal of Occupational Safety and Ergonomics (JOSE) - 2006, Vol. 12, No. 1, Page 79–93.
- Raja-2010, Complacency and Bias in Human Use of Automation: DOI: 10.1177/0018720810376055.
- Smith, 1983. Sampling Considerations in Evaluating Cooperative Extension Programs, University of Florida.
- Surender kumar, Effectiveness of seafarers training, Doctoral thesis-2015, COMES, LSCM, UPES, Dehradun, India.
- Stetz. M., Wiederhold, B.K. and Wildzunas, R., 2006. Usefulness of virtual reality stress inoculation training for military medical females. Cyber psychology and Behaviour.
- Taro Yamane, 1967. Statistics, an Introductory Analysis, 2nd Ed., New York: Harper and Row.
- T. Bullemer Peter -1997, A New Training Strategy: Design the Work Environment for Continuous Learning, Honeywell Advanced Solutions centre.
- Tichon et al., 2003. Proceedings of the World Conference on Educational Multimedia, Honolulu, Hawaii.
- Winfred, Winston & Pamela, Journal of Applied Psychology, The American Psychological Association Inc. 2003, Vol. 88, No. 2, Pages 234–245.
- Wiener, 1989; Bainbridge, 1983; Wiener & Curry, 1980, Flight-deck automation: promises and problems. www.sciedu.ca/jnep (Designing the simulation learning environment: Journal of Nursing Education and Practice)
- www.aveva.com/engineer-procure-construct/simulation-and-training/operator-training-simulators
- www.kirkpatrickpartners.com/Our-Philosophy/The-Kirkpatrick-Model
- [www.taproot.com/archives/699 & 6410](http://www.taproot.com/archives/699&6410)
- Yamill. S. & McLean, G. N. (2001), Theories supporting transfer of training, Human Resource Development Quarterly, pages 195-208.
- Y.F. Barnard, Veldhuis, G.J., & Van Rooij, J.C.G.M., (2001), "Evaluation in practice: Identifying factors for improving transfer of training in technical domains", Studies in Educational Evaluation, 27:269-290.

Dynamic model of a heat pump based house heating system

Shobhana Singh Kim Sørensen

Department of Energy Technology, Aalborg University, Denmark
{ssi,kso}@et.aau.dk

Abstract

Heat pumps have gained a huge popularity as an efficient indoor heating system including several other applications due to relatively low installation cost and energy consumption, and high potential mitigation of global warming than conventional heating systems. The present work aims to develop a modeling tool to investigate dynamic behavior of heat pumps for heating applications. A vapor-compression, air-source HP house heating system with a zeotropic refrigerant R-407c is developed using MATLAB[®]-based Simscape[™] language. The system components such as the evaporator, compressor, condenser, expansion valve, house and the controller are modeled using a combination of Simscape[™] foundation libraries. Each component model has a set of tunable parameters that can be adjusted by user preferences in order to simulate the real-time operation. The dynamics of the system and its response to variable ambient temperature is investigated and discussed. The model depicts the expected behaviour under the simulated operating conditions and is capable of maintaining a comfortable user defined heating temperature inside the house. The model developed will be used to gain insight into performance characteristics of heat pump systems used for local production in the district heating. *Keywords: Dynamic modeling, vapor-compression heat pump, Simscape, two-phase fluid, R-407c, coefficient of performance, house heating*

1 Introduction

Heat pumps (HPs) are becoming one of the highly sustainable systems around the world as they can be operated in both, centralized and decentralized power generation. They are highly efficient and economical component of modern heating and cooling systems which not only use less energy but also mitigate greenhouse gas emissions. Due to the availability of a variety of external energy sources, HP based systems are widely used for both domestic and industrial applications. Air-source HPs are often used for air-conditioning in moderate climatic conditions. They absorb heat from the outside air at low temperatures into a fluid, which passes through a compressor, where its temperature is increased, and transfers heat at its highest temperature to the heating and hot water circuits.

In the European Union, space and water heating

account for approximately 79% of the total energy consumption (192.5 Mtoe¹) out of which exclusively 64.7% comes from the residential sector (Europa 2016). Reduction in the energy consumption, in buildings and industries, can be achieved by several means such as innovative construction and design, energy efficient heating systems, and smart control strategies. However, HP based heating can solely reduce the electricity consumption by approximately 50% than the conventional heating sources such as gas or oil boilers and direct electric heaters. Increasing popularity of HP systems for space heating applications has impelled the need to investigate HPs and their dynamic behavior in response to the varying load in order to make improvements in system design and control.

Mathematical modeling and simulation are one of the best tools for analyzing system's behavior and controlling it for the given operating conditions. Dynamic modeling of vapor-compression systems has been a topic of broad research interest since last three decades. Several modeling efforts, transient and steady-state, have been made to simulate HP and its components, and to approximate their behavior for heating applications. The transient response of evaporating and condensing flows was first investigated by Wedekind and Stoecker (Wedekind and Stoecker 1968). Later, more research focused mainly on two main modeling approaches, the moving boundary (MB) and finite volume (FV), was conducted to predict the accurate evolution of dynamics in the systems (Wedekind, Bhatt, and Beck 1978; Beck and Wedekind 1981). A detailed review of both approaches for dynamic modeling of vapor compression systems is provided by Bendapudi et al. (Bendapudi, Braun, and Groll 2008). It has also been reported that the FV formulation is more robust for start-up and load-change transients, however, computationally slower than MB method (Bendapudi, Braun, and Groll 2008). Rasmussen and Bhaskar (Rasmussen 2012; Rasmussen and Bhaskar 2012) reviewed the dynamic modeling of vapor-compression systems in detail and presented examples of physical models of the system components. Koury et al. (Koury, Machado, and Ismail 2001) presented two models to simulate the steady-state and transient behavior of water-water type refrigeration system where the condenser and evaporator were

¹million tonnes of oil equivalent

modeled as control volumes (CVs). Haberschill et al. (Haberschill, Guitari, and Lallemand 2007) developed a model to simulate the operation of a CO₂ air-to-water HP under transient conditions. Dynamic modeling of vapor-compression systems and maintaining the right balance between the complexity and accuracy is challenging. However, the evolution of advanced simulation softwares has significantly reduced the modeling effort.

The use of object-oriented modeling approach has increased during past few years due to the ease of implementation, computationally efficient and reasonable robustness (Rasmussen 2012). The most common simulation platforms available for dynamic modeling of complex physical systems are Dymola, Modelica, MATLAB-Simulink[®], Simscape^{™2}. Torge and Gerhard (Torge and Gerhard 2004) developed a steady-state model of CO₂-refrigeration cycle with two different types of heat exchangers in Modelica and showed a fair agreement of the results with measured data. Gräber et al. (Graber et al. 2010) used an object-oriented thermodynamic library written in Modelica to model novel HP systems for domestic hot water supply. Bin and Alleyne (Bin and Alleyne 2010) developed a dynamic model in Simulink[®] to describe the transient behavior of heat exchangers (condenser/evaporator) for vapor compression cycle systems used in Air Conditioning and Refrigeration. They validated the model with the experimental system and concluded that the model well predicts the system dynamics in shut-down and start-up transients. Mortada et al. (Mortada et al. 2012) developed a dynamic model of a HP system in the Dymola environment and validated each component by comparison with test results of a heat pump prototype. In addition, Chamoun et al. (Chamoun et al. 2012) adopted both MB and FV approaches to develop a high temperature HP model in Modelica using water vapor as a refrigerant.

2 Problem definition and approach

A physical modeling approach provides several benefits among other approaches available for modeling multi-domain systems. It enables the design engineer to evaluate multiple system configurations and design options, predict system performance, and cost-effectively test control strategies. Models based on physical approach also provide a virtual prototype which can be scaled relatively faster than other conventional approaches. The sub-system or component models developed for one system can also be utilized in multi-domain system models of different application. The objective of the present work is to develop a platform to simulate the dynamic behavior of typical air-to-air vapor compression HP using a physical modeling

²Simscape is a toolbox in Simulink which supports modeling and simulation of multi-domain physical systems.

approach. Development of the system model in the chosen simulation environment for a house heating application is presented. Results from the model are discussed with the concluding remarks in the end. The main difference between the previous studies and the present work is mainly the choice of modeling platform and the controlling approach.

3 Dynamic modeling

The primary objective of developing a physical model is to capture the physical behavior of the HP system and its dynamic response to the heat consumption application. For the purpose here, air-to-air vapor-compression HP is simulated with R-407c as a refrigerant. The simulated application is useful as it captures the standard household utility of space heating.

3.1 Methodology

The modeling platform chosen in the present work is Simscape[™] (MATLAB 2018a) which is based on the physical modelling approach. Simscape offers a much stronger simulation environment in combination with MATLAB and Simulink functionalities. It strengthens the model capabilities and enables users to reuse the model in multiple physical domains and applications. For example, the physical system and controllers can be optimized simultaneously in one environment. Additionally, the physical system model can be converted into C-code which can be deployed for other tasks such as hardware-in-the-loop (HIL) testing. In the chosen approach, the whole system is simulated as a set of components (sub-systems) represented by physical blocks where the dynamic equations are expressed as partial differential equations. The model is developed by assembling the system components (sub-systems) of different domains using physical connections represented by lines and control signals into a single schematic. The output of each component is generated based on input parameters, states at a given time and the connection between them. The graphical scheme of the system and the interconnections between components can be seen in Figure 1.

The components of the HP system, i.e., evaporator, compressor, condenser expansion valve, along with the indoor space and the outdoor environment are created using the 'Two-phase fluid (2P)' foundation library of Simscape. The model includes necessary sensors to measure the temperature, pressure, and mass flow rate at the desired locations. Simscape blocks that are used in developing the different components are also shown in Table 1. The description of each Simscape block with corresponding governing equations can be found in the reference (MATLAB 2018b). Table 2 illustrates the parameters used in developing the current model.

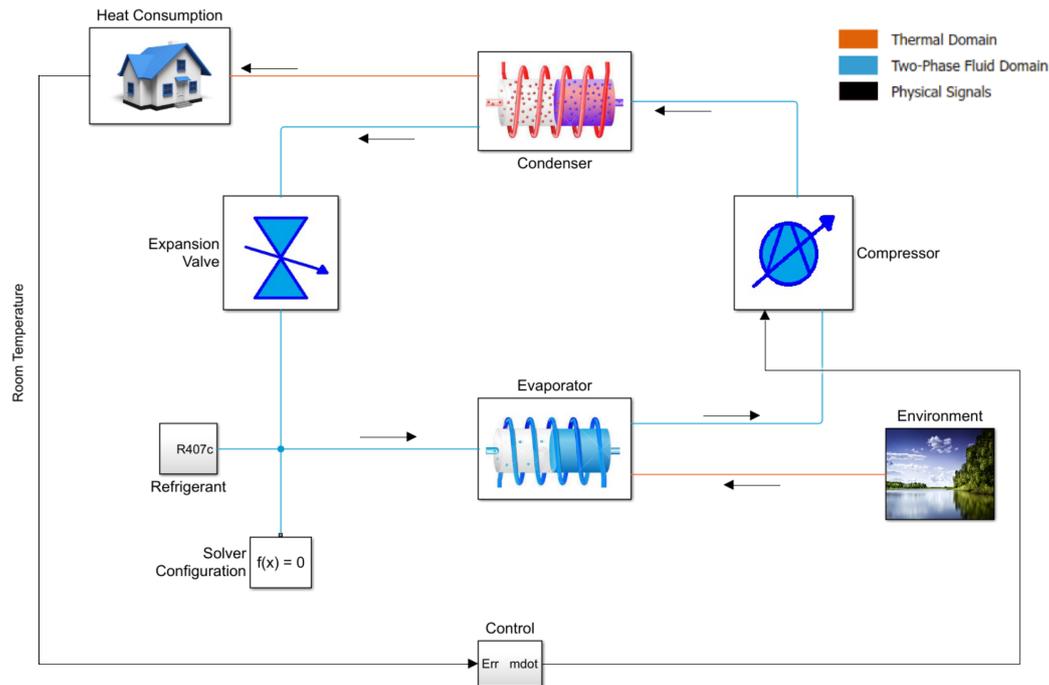


Figure 1. The graphical scheme of the system model in Simscape.

3.2 Physical component modeling

This section describes the modeling approach adopted to model different physical components of the heat pump system, the physical blocks adopted from the Simscape library, and the assumptions made to simulate system's dynamic behavior.

3.2.1 Evaporator

Evaporators are heat exchangers that are used in HPs to absorb heat from the low-temperature source and transfer it to the refrigerant. In the present study, outdoor ambient air is the low-temperature source and is simulated as the environment). For air-to-air systems, fin and tube type of heat exchangers are well suited evaporators as they have larger surface area and thus provide better heat transfer.

In the present work, a finned copper tube of the constant circular cross-sectional area is modeled using the 'Pipe (2P)' block. The block models the flow dynamics of a two-phase fluid due to friction losses and the convective heat transfer within the rigid pipe wall. The evaporator

model is described by the geometrical data as shown in Table 2. The following assumptions are considered in the model:

- Two control volumes
- The liquid and vapor phase are in equilibrium
- Fully developed flow
- Negligible gravitational force
- Heat transfer at constant pressure

The refrigerant enters the evaporator in two-phase and leaves in the form of superheated vapor. To capture this dynamic change in the phase, the evaporator is modeled as two fluid CVs with a set of differential equations based on temperature and phase of the refrigerant in each segment (lumped model) which approximated a linear behavior. The illustration of the evaporator dynamics is shown in Figure 2. The pressure and temperature evolve according to the compressibility and thermal capacity of the fluid volumes. The model also accounts for heat convection from the ambient air i.e., the environment to the outer pipe wall, heat conduction through the pipe wall and heat convection from the evaporator pipe wall to the refrigerant. This approach satisfactorily describes the dynamics inside the heat exchanger.

3.3 Compressor

The compressor is a principal pressure source in HPs. When the superheated vapor leaves the evaporator, the

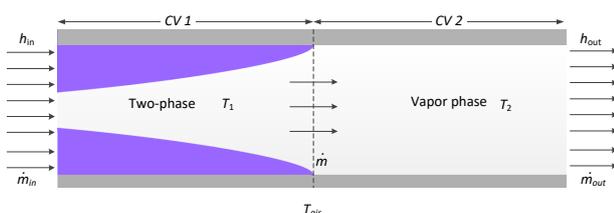
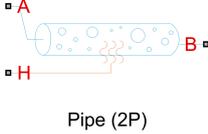
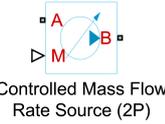
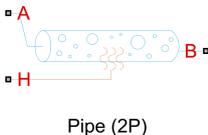
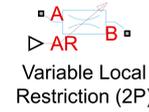
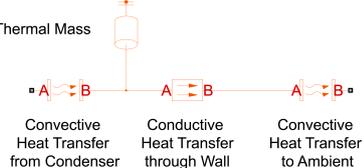
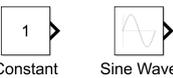


Figure 2. Illustration of the evaporator.

Table 1. Description of Simscape™ blocks.

Physical component	Simscape™ block
Evaporator	 <p>Pipe (2P)</p>
Compressor	 <p>Controlled Mass Flow Rate Source (2P)</p>
Condenser	 <p>Pipe (2P)</p>
Expansion valve	 <p>Variable Local Restriction (2P)</p>
House	 <p>Thermal Mass</p> <p>Convective Heat Transfer from Condenser Conductive Heat Transfer through Wall Convective Heat Transfer to Ambient</p>
Refrigerant	 <p>Two-Phase Fluid Properties (2P)</p>
Environment	 <p>Constant Sine Wave</p>

compressor performs an isentropic work on it, which means that entropy at the evaporator outlet is equal to the entropy at the inlet of the condenser. It conveys superheated vapor of the refrigerant from a low pressure and low temperature state to a high pressure and high temperature state. In the present model, the compressor is modeled by using ‘Controlled Mass Flow Rate Source (2P)’ block, which is based on the following assumptions:

- Negligible friction losses
- Isentropic compression
- No heat exchange with the surroundings, i.e. adiabatic

The compressor is simulated as an ideal mechanical source that maintains the desired mass flow rate

Table 2. Details of model parameters.

<i>Evaporator</i>	
Pipe length	30 m
Pipe diameter	0.01 m
Pipe thickness	0.005 m
Fin area	1 m ²
Convective heat transfer coefficient	150 W/m ² K
Phase-change time constant	1 s
<i>Condenser</i>	
Pipe length	30 m
Pipe diameter	0.01m
Pipe thickness	0.005m
Convective heat transfer coefficient	20W/m ² K
Phase-change time constant	1 s
<i>Expansion valve</i>	
Minimum restriction area	0.1mm ²
Maximum restriction area	1.5mm ²
<i>House</i>	
Height	4 m
Width	5 m
Length	5 m
Wall thickness	0.15 m
Set temperature	294.15K
Convective heat transfer coefficient from air to wall	20 W/m ² K
Convective heat transfer coefficient from wall to atmosphere	30 W/m ² K
<i>Environment</i>	
Environment temperature	288.15 K
Atmospheric pressure	0.101325 MPa
<i>Refrigerant</i>	
Refrigerant type	R-407c
Initial pressure	0.6 MPa
Initial vapor quality	0.05
Mass flow rate	0.005 kg/s
<i>Material properties</i>	
Specific heat of air	1005.4 J/kgK
Density of air	1.2250 kg/m ³
Specific heat of Copper	390 J/kg.K
Thermal conductivity of copper	400 W/m.K
Density of copper	8940 kg/m ³
Thermal conductivity of brick wall	0.03 W/m.K

regardless of the pressure difference. The mass flow rate is provided as a physical input signal. It is worth to mention that the time-constant of the mass flow rate through the compressor is smaller than that for the heat exchangers. Since the start-up and shut-down period dynamics is not the focus of the present modeling, time-constant of 0.1 s is assumed sufficient to simulate the temperature variation through the compressor.

3.4 Condenser

The condenser is similarly modeled as the evaporator. However, necessary adjustments in parameters and construction modifications are made. The corresponding geometrical data is keyed into the 'Pipe (2P)' block. The assumptions are similar to the evaporator but the condenser is simulated with one fluid CV (lumped). The dynamics of the condenser modeled is illustrated in Figure 3. The superheated high-pressure vapors enter the heat exchanger and condense at constant pressure leaving it in two-phase. The heat rejected by the condenser is transferred to house for heating consumption.

3.5 Expansion valve

The two-phase refrigerant from the outlet of condenser passes through the expansion-valve which reduces the refrigerant pressure to a level that maintains the superheating in the evaporator. The expansion is isenthalpic, which means the enthalpy at the condenser outlet is equal to the enthalpy at the evaporator input. The 'Variable Local Restriction (2P)' block is used to simulate the expansion valve in the present model. The block models the pressure loss across the restriction by controlling the restriction area as a physical signal. The model is based on the following assumptions:

- Negligible friction losses
- Isenthalpic expansion
- No heat exchange with the surroundings i.e., adiabatic

3.6 House

The house model is simplified in order to simulate a space heating consumption. The house is therefore simulated as a single, small-sized square room which needs to be heated up to a user desired temperature. The

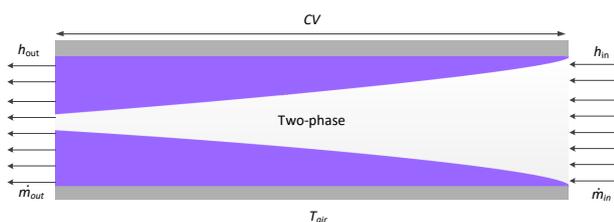


Figure 3. Illustration of the condenser.

model is characterized by the thermal mass of air and thermal properties of the materials. It is assumed that the house is empty and that no internal heat transfer takes place (i.e., no heat generation). However, losses to the ambient are considered. The indoor air exchanges heat with the environment (ambient air) which is simulated as a combination of thermal convection, and thermal conduction through exterior walls. The house model is developed using the 'Thermal (T)' library of Simscape.

3.7 Environment

The environment is modeled as a low-temperature heat source of infinite capacity to the heat pump with a defined ambient temperature. In the present work, blocks named 'Constant' and 'Sine Wave' are used to emulate ambient air temperature profiles.

3.8 Refrigerant properties

R-407c is preferred as a refrigerant to simulate the heating system as it is commonly used in both residential and commercial HPs. It is a non-flammable zeotropic blend of three refrigerants namely difluoromethane (R-32), pentafluoroethane (R-125), and 1,1,1,2-tetrafluoroethane (R-134a) with a mass ratio of 23%, 25% and 52% respectively (ASHRAE 2017). It is environmentally acceptable with zero Ozone Depletion Potential (ODP) however has a considerable Global Warming Potential (GWP) of 1530. The property tables of both phases, liquid and vapor, are generated using the REFPROP database developed by National Institute of Standard and Technology (NIST). The 'Two-Phase Fluid Properties (2P)' block is used to import the thermophysical properties of a two-phase refrigerant in the modeling platform and use during the simulation.

3.9 Operation mode and control

Since the present work focuses only on heat pump operation for space heating application, the critical feature of the model is heat flow control, which ensures the effective use of heat pump and its heating output. Two control strategies are utilized to simulate the desired behavior and are described in the following sections:

3.9.1 On/off control

In the model, the heating system is controlled by an on/off controller which keeps the indoor space temperature at a desired level by turning the system on and off to satisfy the load requirements. The on/off control is chosen because the heat pump system which will be later used to validate the model has the same controller. This is also the most common variation of thermostats for space heating. The temperature set-point is modeled using a 'Constant' block which specifies the temperature that must be maintained inside the house. The set-point is 294.15 K with a differential of $\pm 1^\circ$ in the present simulation. When the temperature in the house falls below 293.15 K, the controller turns on and the

compressor runs with full power; when the temperature exceeds 295.15 K, the controller turns off the compressor.

3.9.2 Expansion valve control

The compressor in the heat pump is a core component, and also expensive which needs to be chosen carefully for the given application. Therefore, it must be controlled against any kind of failure under the operational limits. It is critical to control the temperature to ensure that only superheated vapors exit the evaporator to avoid malfunctioning and mechanical damage to the compressor. The control checks the evaporator temperature using a temperature sensor, and continuously feeds it back to control the restriction area of the valve desired to maintain the superheat.

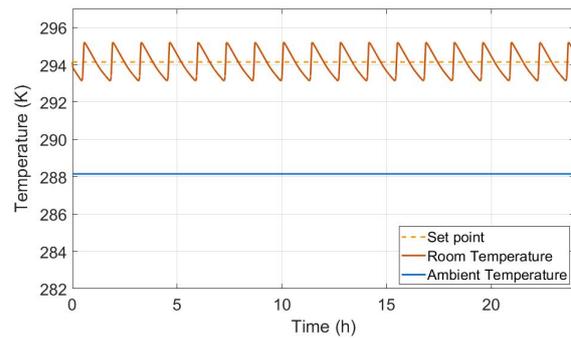
3.10 Solver

The complete model of the system is obtained by combining the sub-models of the components. Simulation time, step size, and solver type are defined as global model configuration parameters. The system is simulated for 24 hours (~ 86400 s) with a time step of 1 s. The implicit fixed-step solver ODE14X is selected as the solver for the simulation system. The present model simulates known physical system, such as heat exchanger and house designs. Whereas, unknown parameters like initial values, heat transfer coefficients are tuned to achieve the desired response and convergence.

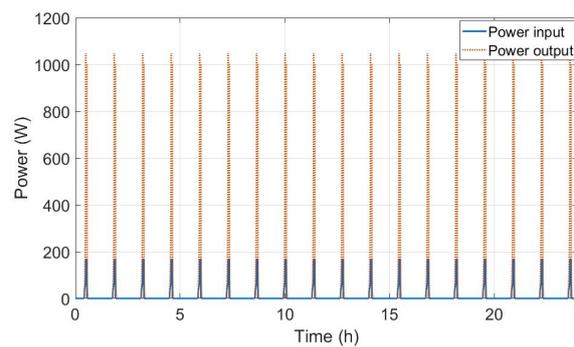
4 Simulation results

In this section, the output results based on the simulated system design are presented and discussed. Since the ambient temperature has a significant influence on the thermal behavior of the house, results with two ambient temperature profile are analyzed.

Figure 4(a-b) presents the performance results of the system at constant ambient temperature of 288.15 K. Figure 4(a) shows the house temperature (represented by solid line '—') as a function of time and the ambient temperature (represented as solid line '—') as predicted by the model. It can be seen that as the house temperature falls 1° below the user set temperature level, which is 294.15 K (represented by dashed line '- - -'), the HP starts and runs until the house temperature reaches 1° above the set-point, and then it stops until the temperature drops 1° below the set-point. The heating performance of the HP is assessed by the total heating effect produced and the amount of power consumed in producing that effect. Figure 4(b) shows the heat flow from the condenser to the house, i.e. power output (represented by dotted line '.....') and the corresponding power input (represented as solid line '—'). The results show that approximately 1040 W heating effect is produced at a power consumption of 170 W by the system. The system behaves similarly over the simulated time and thus the heat consumption profile remains same for a period of 24 hours due to constant ambient



(a) Temperature variation of the house.

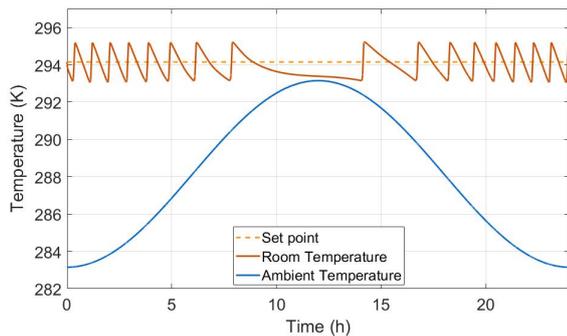


(b) Heat pump power input and output.

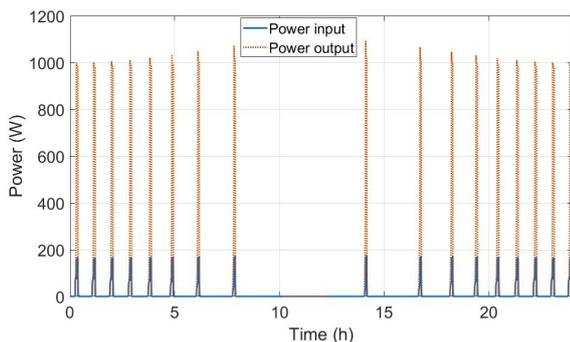
Figure 4. Performance characteristics at constant ambient temperature.

temperature.

The system's response is also analyzed with variable ambient temperature. To serve the purpose here, the sinusoidal temperature profile is chosen to emulate the variable ambient temperature. Figure 5(a-b) shows the corresponding results predicted by the model. The variation in the house temperature (represented by solid line '—') with time is shown in Figure 5(a) where it is seen that when the ambient temperature (represented by solid line '—') is higher, the HP does not run or runs relatively less frequent than at lower ambient temperature. This effect is attributed to lower heat loss from the house due to higher ambient temperature which in turn reduces the house heating demand. As expected, the house temperature is maintained within 1° of the set temperature (represented by dotted line '- - -') even when the outdoor temperature varies sinusoidally. The impact of varying ambient temperature on the power input (represented by solid line '—') and output (represented by dotted line '.....') is shown in Fig. 5(b). It is observed that the heating effect, i.e. power output is slightly lower, approximately 1000 W, at low ambient temperatures and increases gradually to approximately 1070 W as the ambient temperature increases leading to no flow of heat at the peak. The results confirm that the model developed satisfactorily predicts the dynamic behavior of the HP system, thus can be utilized at varying loads.



(a) Temperature variation of the house.



(b) Heat pump power input and output.

Figure 5. Performance characteristics at time varying ambient temperature.

5 Limitations and future work

The model developed demonstrates the dynamic behavior of a typical air-to-air vapor-compression HP for domestic space heating application. However, the results are yet to be verified by the laboratory experiments to ensure the reliability. The laboratory scale set-up of the HP system to test the accuracy of the model and various thermal effects based on the model are currently under construction. Once built, the model will be verified and used in developing and testing new control strategies, and the overall system design to achieve better performance and energy consumption results. It will further be extended using the same approach to simulate the dynamics at various load profiles. The model developed is considered useful in determining HPs feasibility to the heating demands in existing or new district heating system.

6 Conclusion

The present work focused on physical modeling of the HP based heating system dynamics. The model simulates the performance outputs such as evaporator, condenser and heating temperature, compressor, expansion valve operation as well as the thermal behavior of the heat consumption. The results obtained show that SimscapeTM is a robust tool for both evaluating and controlling the dynamic behavior of the system, however,

limited to the design simulated and the input parameters used. The developed model provides a framework to build multiple model strategies with different available sources and integrated renewable energy systems (buffer) that will deliver better heating performance and flexibility of a HP at a lower cost. Moreover, different scenarios of heating applications can be investigated by adopting different control designs in the model which can positively contribute to the district heating systems and power grid balancing.

Acknowledgement

The work presented in the paper is a part of a research project titled ‘Local heating concepts for power balancing’ funded by Innovation fund Denmark (IFD) under the ForskEL programme, Project grant no. 12539.

References

- ASHRAE (2017). *Journal American Society of Heating, Refrigeration and Air-conditioning Engineers, ASHRAE Handbook-Fundamentals-I-P Edition, Chapter-30.*
- Beck, B. T. and G. L. Wedekind (1981). “A Generalization of the System Mean Void Fraction Model for Transient Two-Phase Evaporating Flows”. In: *ASME Journal of Heat Transfer* 103.1, pp. 81–85. DOI: 10.1115/1.3244436.
- Bendapudi, S., J. E. Braun, and E. A. Groll (2008). “A comparison of moving-boundary and finite-volume formulations for transients in centrifugal chillers”. In: *International Journal of Refrigeration* 31.8, pp. 1437–1452. ISSN: 0140-7007. DOI: <https://doi.org/10.1016/j.ijrefrig.2008.03.006>.
- Bin, L. and A. G. Alleyne (2010). “A dynamic model of a vapor compression cycle with shut-down and start-up operations”. In: *International Journal of Refrigeration* 33.3, pp. 538–552. ISSN: 0140-7007. DOI: <https://doi.org/10.1016/j.ijrefrig.2009.09.011>.
- Chamoun, M. et al. (2012). “Dynamic model of an industrial heat pump using water as refrigerant”. In: *International Journal of Refrigeration* 35.4, pp. 1080–1091. ISSN: 0140-7007. DOI: <https://doi.org/10.1016/j.ijrefrig.2011.12.007>.
- Europa (2016). *European Commission. An EU Strategy on Heating and Cooling, 51 Final. Available online: , accessed April 11, 2018.* Tech. rep. DOI: https://ec.europa.eu/energy/sites/ener/files/documents/1_EN_ACT_part1_v14.pdf.
- Graber, M. et al. (2010). “Modelling of heat pumps with an object-oriented model library for thermodynamic systems”. In: *Mathematical and Computer Modelling*

- of Dynamical Systems* 16.3, pp. 195–209. DOI: 10.1080/13873954.2010.506799.
- Haberschill, P., I. Guitari, and A. Lallemand (2007). “Comportement dynamique d’une pompe ffdfffd chaleur au CO₂ en cycles sous critique et transcritique”. In: *International Journal of Refrigeration* 30.4, pp. 732–743. ISSN: 0140-7007. DOI: <https://doi.org/10.1016/j.ijrefrig.2006.09.002>.
- Koury, R. N. N., L. Machado, and K. A. R. Ismail (2001). “Numerical simulation of a variable speed refrigeration system”. In: *International Journal of Refrigeration* 24.2, pp. 192–200. ISSN: 0140-7007. DOI: [https://doi.org/10.1016/S0140-7007\(00\)00014-1](https://doi.org/10.1016/S0140-7007(00)00014-1).
- MATLAB (2018a). *Simscape™*. URL: <https://se.mathworks.com/products/simscape.html>.
- (2018b). *Simscape Blocks in Simscape-By Category*. URL: https://se.mathworks.com/help/physmod/simscape/simscapeblocklist.html?s_cid=doc_ftr (visited on 06/14/2018).
- Mortada, S. et al. (2012). “Dynamic modeling of an integrated air-to-air heat pump using Modelica”. In: *International Journal of Refrigeration* 35.5, pp. 1335–1348. ISSN: 0140-7007. DOI: <https://doi.org/10.1016/j.ijrefrig.2012.03.009>.
- Rasmussen, B. P. (2012). “Dynamic modeling for vapor compression systems-Part I: Literature review”. In: *HVAC&R Research* 18.5, pp. 934–955. DOI: 10.1080/10789669.2011.582916.
- Rasmussen, B. P. and S. Bhaskar (2012). “Dynamic modeling for vapor compression systems-Part II: Simulation tutorial”. In: *HVAC&R Research* 18.5, pp. 956–973. DOI: 10.1080/10789669.2011.582917.
- Torge, P. and S. Gerhard (2004). “Modelling and transient simulation of CO₂-refrigeration systems with Modelica”. In: *International Journal of Refrigeration* 27.1, pp. 42–52. ISSN: 0140-7007. DOI: [https://doi.org/10.1016/S0140-7007\(03\)00098-7](https://doi.org/10.1016/S0140-7007(03)00098-7).
- Wedekind, G. L., B. L. Bhatt, and B. T. Beck (1978). “A system mean void fraction model for predicting various transient phenomena associated with two-phase evaporating and condensing flows”. In: *International Journal of Multiphase Flow* 4.1, pp. 97–114. ISSN: 0301-9322. DOI: [https://doi.org/10.1016/0301-9322\(78\)90029-0](https://doi.org/10.1016/0301-9322(78)90029-0).
- Wedekind, G. L. and W. F. Stoecker (1968). “Theoretical Model for Predicting the Transient Response of the Mixture-Vapor Transition Point in Horizontal Evaporating Flow”. In: *ASME Journal of Heat Transfer* 90.1, pp. 165–174. DOI: 10.1115/1.3597448.

Population balance modelling for fertilizer granulation process

Ludmila Vesjolaja¹ Bjørn Glemmestad² Bernt Lie¹

¹Department of Electrical Engineering, IT and Cybernetics, University of South-Eastern Norway, Norway,
{ludmila.vesjolaja,bernt.lie}@usn.no

²Process Modeling and Control Department, Yara Technology Center, Norway, bjorn.glemmestad@yara.com

Abstract

Few granulation plants are operated optimally. It is common to operate granulation plants below their maximum design capacity, and in many cases, periodic instabilities may also occur. From a process control and optimization point of view, it is desirable to develop a dynamic model that can show the dominating dynamics of a granulation process and can be used for design of optimal operation of the granulation plant. In this paper, a dynamic model of a drum granulator is developed using population balance (PB). Different simulation scenarios are used to analyze various granulation mechanisms that are characteristic to drum granulators. Simulation results show that for the drum granulator, the particle agglomeration has a greater impact on the change in particle size distribution (PSD) compared to the particle growth due to layering. In addition, coarser particles are produced when a size-dependent agglomeration kernel is used in the granulator model. For combined processes, i.e., processes where the particle growth due to layering and agglomeration are considered simultaneously, coarser particles with a wider PSD are obtained with the size-dependent agglomeration kernel.

Keywords: granulation, population balance, agglomeration, layering

1 Introduction

Granulation is a particle enlargement process during which fine particles and/or atomizable liquids are converted into granules via a series of complex physical processes. In a typical granulation plant, the main operational goal is to produce granules with improved properties compared to their ungranulated form, and therefore, to meet product quality requirements (e.g., produce granules with the desired PSD, moisture content, porosity, etc.). Granulation processes are used in a wide range of industrial applications, such as in pharmaceuticals, chemicals, and fertilizer industries (Litster and Ennis, 2004). However, the operation of granulation plants in an industrial scale can be challenging. Many granulation plants have a high recycle-to-product ratio, and it is common to operate granulation plants below their maximum design capacity. In addition, periodic instability associated with the operation of the granulation circuit have been reported (Radichkov et al., 2006; Heinrich et al., 2003). This causes the PSD of

the particles flowing out of the granulator to oscillate, thus making it difficult to maintain the desired product quality. An increase in the production of off-spec particles (oversized and undersized) gives rise to a higher recycle-to-product ratio, and the plant does not operate in an optimal manner. One way to address these problems is to develop a mathematical model of the granulator that can be used to study and understand various dynamics occurring in the granulator. The model can further be used to design optimal control structures to increase the efficiency of the plant.

The most widely used approaches for modeling granulation processes include Discrete Element Modeling, as well as PB modeling. In this study, since the focus is on the development of a dynamic model suitable for control purposes, PB modeling has been used to develop a mathematical model of a drum granulator. A rich literature related to PB modeling of a granulation process is available (Randolph and Larson, 1962; Wang et al., 2006; Wang and Cameron, 2007). In these works, (i) the effect of different granulation mechanisms to the PSD of the granulator outflow is neglected, (ii) the numerical scheme (Hounslow discretization method) used for solving the population balance equation (PBE) is relatively inaccurate (shows overprediction as also mentioned in (Kumar, 2006; Kumar et al., 2006)), and (iii) many of the processes are only batch processes. In this paper, a dynamic model of a continuous drum granulator using the PB framework is developed. The resulting integro-differential PBEs are discretized using an accurate numerical scheme, namely the cell average technique (Kumar, 2006) and the flux limitation scheme (Koren, 1993). The developed model is simulated to understand and analyze how different granulation mechanisms affect the PSD of the granules formed in the granulator.

2 Granulation Mechanisms

According to (Iveson et al., 2001), the granulation process is divided into three basic mechanisms: (i) nucleation and wetting, (ii) growth and consolidation, and (iii) breakage and attrition. Different granulation mechanisms are predominant depending on the type of the granulator being used. For continuous drum granulation with recycling, effects of nucleation, breakage, and attrition mechanisms are believed to be negligible compared to particle growth due to layering and agglomeration (Fig. 1). Particle nu-

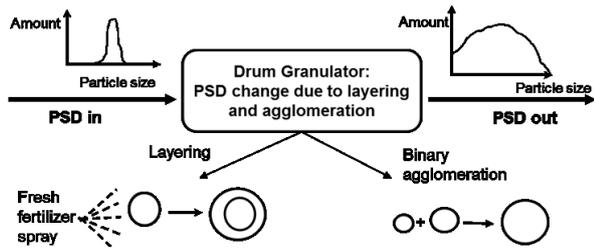


Figure 1. Main granulation mechanisms characteristic to drum granulators.

cleation is insignificant for this particular process as it is a continuous process in which the recycle feed acts as seeds for the granulator. Particle breakage is mainly important in a high shear granulators, e.g., granulation mechanism due to breakage can be significant in pharmaceutical industries where high shear granulators are typically used. As to the attrition, this granulation mechanism might give significant changes in PSD only when high velocities (e.g., fluidized bed spray) granulators are used (Litster and Ennis, 2004). Layering occurs due to a successive coating of a liquid phase onto a granule. As a result, the granule grows in its mass, and the volume increases, but the number of granules in the system remains unchanged. No collision between granules is assumed during this particle growth (Litster and Ennis, 2004). Layering is a continuous process (differential growth), and an assumption of size-independent linear growth rate is common in the PB modeling of granulation processes. This simplification implies that each granule has the same exposure to a new fertilizer spray feed material, and a volumetric growth rate is proportional to a projected granule surface area (Litster and Ennis, 2004). Binary agglomeration refers to a particle growth mechanism that occurs due to successful collision of two particles, resulting in the formation of a larger composite particle. Agglomeration is a discrete (sudden) process that changes the total number of particles: two particles *die*, and a new particle is *born* as a result of collision of two particles. Thus, the agglomeration results in a reduction in the total number of particles, while the total mass remains conserved (Litster and Ennis, 2004).

3 Model Development

3.1 Population Balance Principles

Balance laws such as mass and energy balances are often used in process modeling to describe dynamics of different physical and chemical processes. With particulate processes, PB is frequently used to describe dynamics of particle property distribution. A detailed derivation and explanation of the PBE can be found in Ramkrishna (Ramkrishna, 2000). The general form of a PBE with particle

diameter (L) as the internal coordinate is represented as,

$$\frac{\partial n(L,t)}{\partial t} = -\frac{\partial}{\partial L} [Gn(L,t)] + B(L,t) - D(L,t) - \frac{\partial}{\partial z} \left[\frac{dz}{dt} n(L,z,t) \right], \quad (1)$$

where $n(L,t)$ is the number density function. The first term on the right hand side represents the particle growth due to layering, the second and the third terms stand for particle birth and death respectively, and the last term represents a continuous process and gives the flow of particles through the granulator. G is the growth rate and z represents the distance along the axial direction of the drum granulator. The birth and the death terms usually include integrals which make the solution of the population balance equation complicated. In this paper, a plug flow along the axial direction of the drum granulator has been assumed. For simplifying the model complexity, a concept of *output equivalent* (perfect mixing) inside the granulator can be assumed. Thus, Eq. (1) can be simplified to

$$\frac{\partial n(L,t)}{\partial t} = -\frac{\partial}{\partial L} [Gn(L,t)] + B(L,t) - D(L,t) + \dot{n}_i \gamma_i - \dot{n}_e \gamma_e. \quad (2)$$

Here, \dot{n}_i is the number flowrate of particles entering the granulator (influent), \dot{n}_e is the number flowrate of particles leaving the granulator (effluent), γ_i is the size distribution function of the inlet flow of the particles (influent), γ_e is the size distribution function of the outlet flow of the particles (effluent).

In addition, for the PBE of Eq. (2), the following assumptions are made:

- The concept of *perfect mixing* inside the granulator is applied: particle property (size) inside the granulator is the same as at the outlet of the granulator.
- Particle breakage is neglected since the drum granulator is operating at low shear forces. Thus, the birth (B) and death (D) rates are only due to binary agglomeration.
- Particle size reduction due to attrition is neglected since the granulation drum does not operate at high velocities.

3.2 Growth Rate for Layering

The formulation of the particle growth rate for layering (G) is based on combination of the work of (Mörl, 1981) and (Mörl et al., 1977), as summarized in (Drechsler et al., 2005). This model assumes linear size-independent growth rate, meaning that a small particle gets less slurry per unit time than a larger particle, but the growth rate (the change of particle diameter over time) is constant for all particle sizes. Thus, the growth rate due to layering depends on a slurry rate (fresh fertilizer spray rate, \dot{m}_{s1}),

moisture fraction in the slurry ($X_{sl,i}$), and the total surface area of the particles ($A_{p,tot}$) as given by Eqs. (3) and (4).

$$G = \frac{\partial L}{\partial t} = \frac{2\dot{m}_{sl}(1 - X_{sl,i})}{\rho A_{p,tot}}, \quad (3)$$

$$A_{p,tot} = \pi n \int_{L=0}^{L=\infty} L^2 dL. \quad (4)$$

3.3 Particle Agglomeration

Particle agglomeration is a discrete event, which is challenging to model. One of the most widely used formulations of the agglomeration process was introduced by (Kapur and Fuerstenau, 1969). The general form of a length-based agglomeration is represented by Eqs. (5) and (6),

$$B(L, t) = \frac{L^2}{2} \times \int_0^L \frac{\beta \left[(L^3 - \lambda^3)^{\frac{1}{3}}, \lambda \right] n \left[(L^3 - \lambda^3)^{\frac{1}{3}}, t \right] n(\lambda, t)}{(L^3 - \lambda^3)^{\frac{2}{3}}} d\lambda, \quad (5)$$

$$D(L, t) = n(L, t) \int_0^\infty \beta(L, \lambda) n(\lambda, t) d\lambda. \quad (6)$$

Here, β is the agglomeration (coalescence) kernel. The agglomeration kernel is a key parameter that controls the overall rate of agglomeration. Despite more than 50 years of research, only empirical and semi-empirical agglomeration kernels are available. Thus, these should be fitted to experimental data. Some of the most frequently used agglomeration kernels for two colliding particles with volumes v and w in granulation processes are summarized in Table 1. Here, β_0 is the part of the agglomeration kernel which usually depends on the operating conditions of the granulator such as the drum speed, bed depth and the moisture content in the particles. In this paper, the value of β_0 has been taken to be a constant (however tunable) for simplifying the model development. For a more detailed analysis, β_0 should be expressed as a function of process parameters and fitted with the experimental data. This has been left as a potential future work.

Table 1. Overview of agglomeration kernels.

Agglomeration kernel	References
$\beta = \beta_0$	Random kernel
$\beta = \beta_0 \times (v + w)$	Golovin (Golovin, 1963)
$\beta = \beta_0 \times \left(\frac{(v+w)^a}{(vw)^b} \right)$	Kapur (Kapur, 1972)

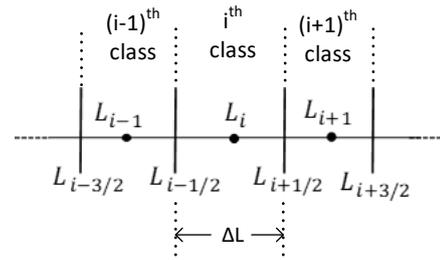


Figure 2. Size discretization into classes (cells).

4 Numerical Solution

Various discretization techniques/schemes can be used to discretize the continuous PBE of Eq. (2) into a set of ordinary differential equations (ODEs) which can then be solved with an appropriate ODE solver. In this work, the particle size is represented by the diameter of the particle. To obtain the particle size distribution, the particles are classified into $i \in 1, 2, \dots, N_c$ classes or cells as shown in Figure 2. Here, i represents the i^{th} particle class, L_i is the diameter of the particle of the i^{th} class, $L_{i\pm\frac{1}{2}}$ is the left and the right boundary of the i^{th} class and $\Delta L = L_{i-\frac{1}{2}} - L_{i-\frac{1}{2}}$ is the size of the classes. The dots in each class represent the cell center. As was discussed in Section 2, the particle size change inside the drum granulator can be considered to be due to

- pure layering (agglomeration mechanism ignored),
- pure agglomeration (layering mechanism ignored),
- combined process (both layering and agglomeration considered).

In this paper, all the three cases of granulation mechanisms are considered separately and described in detail in the subsequent subsections.

4.1 Pure Layering

If the growth of particles is considered to be only due to layering, Eq. (2) reduces to,

$$\frac{\partial n(L, t)}{\partial t} = -\frac{\partial}{\partial L} [Gn(L, t)] + \dot{n}_i \gamma_i - \dot{n}_e \gamma_e. \quad (7)$$

The PDE represented by Eq. (7) can be discretized into a system of ODEs using a finite volume scheme. In this paper, a high resolution scheme, based on the flux limiting approach, is chosen as the numerical scheme for particle size discretization. Particularly, the Koren flux limiting method (Koren, 1993) is used in this paper. The high resolution flux limiting methods attain higher accuracy than the first order upwind scheme. In addition, these methods also avoid spurious oscillations by applying a high order flux in the smooth regions and a low order flux near

discontinuities. Equation (7) can be discretized with the Koren scheme as

$$\frac{dN_i}{dt} = Gn\left(t, L_{i-\frac{1}{2}}\right) - Gn\left(t, L_{i+\frac{1}{2}}\right) + \dot{N}_i\gamma_i - \dot{N}_e\gamma_e, \quad (8)$$

where,

$$n\left(t, L_{i-\frac{1}{2}}\right) \approx \frac{1}{\Delta L} \left[N_{i-1} + \frac{1}{2}\phi\left(\tilde{\theta}_{i-\frac{1}{2}}\right)(N_{i-1} - N_{i-2}) \right], \quad (9)$$

$$n\left(t, L_{i+\frac{1}{2}}\right) \approx \frac{1}{\Delta L} \left[N_i + \frac{1}{2}\phi\left(\tilde{\theta}_{i+\frac{1}{2}}\right)(N_i - N_{i-1}) \right]. \quad (10)$$

Here, ϕ is the limiter function defined as

$$\phi\left(\tilde{\theta}\right) = \max\left[0, \min\left(2\tilde{\theta}, \min\left(\frac{1}{3} + \frac{2\tilde{\theta}}{3}, 2\right)\right)\right]. \quad (11)$$

Parameter $\tilde{\theta}$ is defined as

$$\tilde{\theta}_{i-\frac{1}{2}} = \frac{N_i - N_{i-1} + \varepsilon}{N_{i-1} - N_{i-2} + \varepsilon}, \quad \tilde{\theta}_{i+\frac{1}{2}} = \frac{N_{i+1} - N_i + \varepsilon}{N_i - N_{i-1} + \varepsilon}, \quad (12)$$

with a very small constant ε to avoid division by zero.

In an industrial application, it is relatively easier to work with mass-based population balance equations (PBEs) instead of number-based PBEs due to: (i) PSD in a real plant is typically measured by sieving and weighting, and (ii) mass-based PBE is more convenient to use from a numerical point of view (huge number of particles compared to their masses). To convert the number-based formulation given by Eqs. (8)-(12) to a mass-based formulation, Eq. (13) is applied,

$$N_i = \frac{6M_i}{\pi\rho L_i^3}. \quad (13)$$

Equation (13) was derived assuming that all particles are ideal spheres with constant density. After rearranging, the growth due to layering in a mass-based PBE can be represented as

$$\frac{dM_i}{dt} = L_i^3 \left[Gm\left(t, L_{i-\frac{1}{2}}\right) - Gm\left(t, L_{i+\frac{1}{2}}\right) \right] + \dot{M}_i\gamma_i - \dot{M}_e\gamma_e, \quad (14)$$

where,

$$m\left(t, L_{i-\frac{1}{2}}\right) \approx \frac{1}{\Delta L} \left\{ \frac{M_{i-1}}{L_{i-1}} + \frac{1}{2}\phi\left(\theta_{i-\frac{1}{2}}\right) \times \left(\frac{M_{i-1}}{L_{i-1}^3} - \frac{M_{i-2}}{L_{i-2}^3} \right) \right\}, \quad (15)$$

$$m\left(t, L_{i+\frac{1}{2}}\right) \approx \frac{1}{\Delta L} \left\{ \frac{M_i}{L_i} + \frac{1}{2}\phi\left(\theta_{i+\frac{1}{2}}\right) \times \left(\frac{M_i}{L_i^3} - \frac{M_{i-1}}{L_{i-1}^3} \right) \right\}, \quad (16)$$

with

$$\theta_{i-\frac{1}{2}} = \frac{\frac{M_i}{L_i^3} - \frac{M_{i-1}}{L_{i-1}^3} + \varepsilon}{\frac{M_{i-1}}{L_{i-1}^3} - \frac{M_{i-2}}{L_{i-2}^3} + \varepsilon}, \quad \theta_{i+\frac{1}{2}} = \frac{\frac{M_{i+1}}{L_{i+1}^3} - \frac{M_i}{L_i^3} + \varepsilon}{\frac{M_i}{L_i^3} - \frac{M_{i-1}}{L_{i-1}^3} + \varepsilon}. \quad (17)$$

Here M_i is the total mass of the particle in the i^{th} class. The growth rate G is considered to be size-independent as described in more detail in Section 3.2. The growth rate due to layering is modeled using Eqs. (3) and (4). In addition, if T_R is the retention time, then, $\dot{M}_e\gamma_e = \frac{M_i}{T_R}$.

4.2 Pure agglomeration

If the change in the particle size is considered to be due to agglomeration only, Eq. (2) reduces to

$$\frac{\partial n(L, t)}{\partial t} = B(L, t) - D(L, t) + \dot{n}_i\gamma_i - \dot{n}_e\gamma_e. \quad (18)$$

Analytical solutions of the pure agglomeration problems can be found in some simplified cases. Thus, numerical techniques are needed to solve the resulted PBEs. However, the discretization of agglomeration terms (B, D) is more challenging compared to the growth due to layering. Agglomeration is a discrete event and the birth and death of particles can be considered to be source and sink terms, respectively. A suitable numerical scheme that is simple to implement and produce exact numerical results of some selected moments is the cell averaging technique (Kumar, 2006; Kumar et al., 2006). The cell average scheme is referred to as a sectional method, and assigns all the newborn particles within a cell more precisely compared to other sectional methods. Using the cell average scheme, Eq. (18) can be discretized with respect to the particle size as

$$\begin{aligned} \frac{dN_i}{dt} = & B_{i-1}\lambda_i^-(\bar{L}_{i-1})H(\bar{L}_{i-1} - L_{i-1}) \\ & + B_i\lambda_i^-(\bar{L}_i)H(L_i - \bar{L}_i) + B_i\lambda_i^+(\bar{L}_i)H(\bar{L}_i - L_i) \\ & + B_{i+1}\lambda_i^+(\bar{L}_{i+1})H(L_{i+1} - \bar{L}_{i+1}) \\ & - N_i \sum_{k=1}^{N_c} \beta_{ik}N_k + \dot{N}_i\gamma_i - \dot{N}_e\gamma_e. \end{aligned} \quad (19)$$

Here, N_c is the total number of particle size classes or cells. B_i is the birth of particles in the i^{th} cell due to binary agglomeration of two particles from the j^{th} and k^{th} cell respectively, and can be expressed as

$$B_i = \frac{1}{2} \sum_{j=1}^i \sum_{k=1}^i \beta_{jk}N_jN_k, \quad (20)$$

where condition $L_{i-\frac{1}{2}} \leq (L_j^3 + L_k^3)^{\frac{1}{3}} \leq L_{i+\frac{1}{2}}$ should be fulfilled. β_{jk} is the agglomeration kernel for binary agglomeration of particles from the j^{th} and the k^{th} cells. \bar{L}_i is the

average diameter of all the new-born particles in the i^{th} cell, and is given as

$$\bar{L}_i = \left[\frac{\sum_{j=1}^i \sum_{k=1}^i \beta_{jk} N_j N_k (L_j^3 + L_k^3)}{\sum_{j=1}^i \sum_{k=1}^i \beta_{jk} N_j N_k} \right]^{\frac{1}{3}}, \quad (21)$$

with dimensionless term $\lambda_i^\pm(L)$ given as

$$\lambda_i^\pm(L) = \frac{L^3 - L_{i\pm 1}^3}{L_i^3 - L_{i\pm 1}^3} \quad (22)$$

The Heaviside step function H is defined as

$$H(\tilde{L}) = \begin{cases} 1, & \text{if } \tilde{L} > 0 \\ \frac{1}{2}, & \text{if } \tilde{L} = 0 \\ 0, & \text{if } \tilde{L} < 0. \end{cases} \quad (23)$$

The cell average technique can be used to preserve any two moments. Here, we have chosen to preserve the zeroth moment (total number of particles conserved) and the third moment (total mass conserved) taking the diameter-based formulation. Using Eq. (13), the mass based form of the PBE can be written as

$$\begin{aligned} \frac{dM_i}{dt} = & L_i^3 [B_{i-1} \lambda_i^- (\bar{L}_{i-1}) H(\bar{L}_{i-1} - L_{i-1}) \\ & + B_i \lambda_i^- (\bar{L}_i) H(L_i - \bar{L}_i) + B_i \lambda_i^+ (\bar{L}_i) H(\bar{L}_i - L_i) \\ & + B_{i+1} \lambda_i^+ (\bar{L}_{i+1}) H(L_{i+1} - \bar{L}_{i+1})] - M_i \sum_{k=1}^{N_c} \beta_{ik} \frac{M_k}{L_k^3} \\ & + \dot{M}_i \gamma_i - \dot{M}_e \gamma_e, \quad (24) \end{aligned}$$

where the birth of the particles B_i are given as

$$B_i = \frac{1}{2} \sum_{j=1}^i \sum_{k=1}^i \beta_{jk} \frac{M_j}{L_j^3} \frac{M_k}{L_k^3}, \quad (25)$$

and the average diameter of all the new-born particles in the i^{th} class is

$$\bar{L}_i = \left[\frac{\sum_{j=1}^i \sum_{k=1}^i \beta_{jk} \frac{M_j}{L_j^3} \frac{M_k}{L_k^3} (L_j^3 + L_k^3)}{\sum_{j=1}^i \sum_{k=1}^i \beta_{jk} \frac{M_j}{L_j^3} \frac{M_k}{L_k^3}} \right]^{\frac{1}{3}}. \quad (26)$$

With T_R being the retention time, $\dot{M}_e \gamma_e = \frac{M_i}{T_R}$.

In this paper, agglomeration kernels (β_{jk} and β_{ik}) are defined using the Kapur model (Kapur, 1972) with $a = \frac{1}{3}$ and $b = 0$; this is one of the most widely used kernels for drum granulation. With the diameter-based formulation, the agglomeration kernels are given as

$$\beta_{xy} = \left(\frac{6}{\pi} \right)^{\frac{2}{3}} \frac{1}{\rho} \beta_0 K_{xy} = \left(\frac{6}{\pi} \right)^{\frac{2}{3}} \frac{1}{\rho} \beta_0 (L_x^3 + L_y^3)^{\frac{1}{3}}. \quad (27)$$

The term $\left(\frac{6}{\pi} \right)^{\frac{2}{3}} \frac{1}{\rho}$ arises during the conversion from the number-based formulation to the mass-based formulation of PBEs. Subscript xy means either jk or ik . β_0 is the particle size independent part of the agglomeration kernel. K_{jk} and K_{ik} are the parts of the agglomeration kernel which are particle size dependent as shown in Eq. (27).

4.3 Combined Process

In the case of the combined process, a change in the particle size is a result of both particle growth due to layering, and particle binary agglomeration. The number-based PBE for the combined process is represented by (2). For conversion to the mass-based PBE, (13) is used. Size discretization for the growth term (G) is performed using the Koren flux limiting scheme as discussed in Section 4.1. Particle birth (B) and death (D) terms are size discretized using the cell averaging technique as was discussed in detail in Section 4.2. The resulting size discretized mass-based PBE for the combined process is written as

$$\begin{aligned} \frac{dM_i}{dt} = & L^3 \left[Gm \left(t, L_{i-\frac{1}{2}} \right) - Gm \left(t, L_{i+\frac{1}{2}} \right) \right] \\ & + L_i^3 [B_{i-1} \lambda_i^- (\bar{L}_{i-1}) H(\bar{L}_{i-1} - L_{i-1}) + B_i \lambda_i^- (\bar{L}_i) H(L_i - \bar{L}_i) \\ & + B_i \lambda_i^+ (\bar{L}_i) H(\bar{L}_i - L_i) + B_{i+1} \lambda_i^+ (\bar{L}_{i+1}) H(L_{i+1} - \bar{L}_{i+1})] \\ & - M_i \sum_{k=1}^{N_c} \beta_{ik} \frac{M_k}{L_k^3} + \dot{M}_i \gamma_i - \dot{M}_e \gamma_e, \quad (28) \end{aligned}$$

where all symbols in (28) are described in previous Sections 4.1 and 4.2.

5 Simulation Results and Discussion

5.1 Simulation Setup

The discretized PBEs for a continuous drum granulation process described by Eqs. (14), (24), and (28) are solved using a 4th order Runge-Kutta method with fixed time step. Dynamic simulations are performed using MATLAB (MATLAB, 2017). Simulations for continuous drum granulation are performed using parameters summarized in Table 2.

Table 2. Simulation setup parameters.

Parameter	Value
Range of L [mm]	0-8
Number of classes	80
ρ [$kg \cdot m^{-3}$]	1300
β_0 [s^{-1}]	$8.5 \cdot 10^{-11}$
T_R [s]	360
$\dot{m}_{sl,i}$ [$kg \cdot s^{-1}$]	250
$X_{sl,i}$	0.1
Time step for RK4 [s]	10
Simulation time [h]	2.5

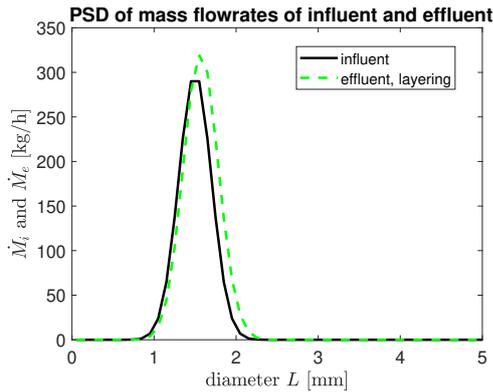


Figure 3. Influent and effluent PSD of the drum granulator for pure layering.

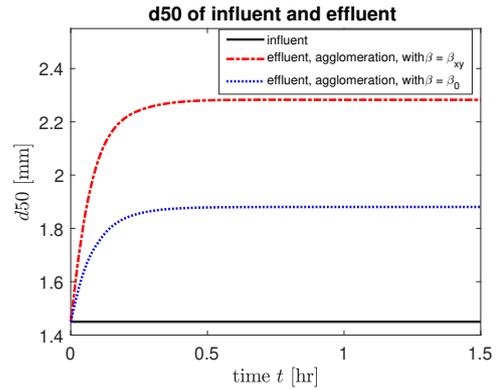


Figure 5. Change of the average particle size for pure agglomeration.

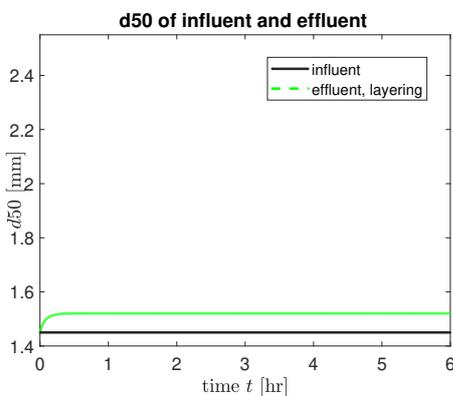


Figure 4. Change of the average particle size for pure layering.

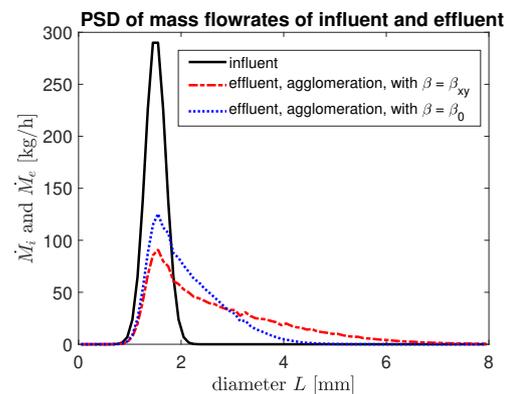


Figure 6. Influent and effluent PSD of the drum granulator for pure agglomeration.

5.2 Simulation Results for Pure Layering and Pure Agglomeration

In this paper, simulation results are compared by analyzing the PSD at the inlet (Gaussian distribution) and the outlet of the drum granulator. In addition, the evolution of the average size of the particles represented by their d_{50} diameter (median diameter that corresponds to intercept for 50% of cumulative mass) are also studied. Figure 3 compares the PSD of the inlet flow and the outlet flow from the granulator after the system has reached the steady state. The only granulation mechanisms affecting the PSD is layering. Clearly, the PSD at the outlet of the drum granulator has changed and has become slightly wider compared to the inlet distribution. The fraction of coarser particles increases due to layering, and, thus more of large particles are produced. Figure 4 shows that the average size of the particles has increased from 1.45 mm to 1.52 mm when only the layering is the driving mechanism for particle size change. Thus, in average, particles have grown by $\sim 5\%$ using the model parameters summarized in Table 2.

However, when agglomeration was chosen as a granulation mechanism, the average particle size has grown by $\sim 30\%$ compared to its initial value (Figure 5). This increase in particle size (d_{50} to 1.85 mm) was observed

when agglomeration was modeled using a constant (size-independent) agglomeration kernel. The inclusion of particle size-dependency on the agglomeration rate has increased the average particle size even more (Figure 5). The d_{50} has grown from 1.45 mm (at the inlet) to 2.28 mm at the outlet of the granulator (with size-dependent agglomeration kernel). This gives $\sim 58\%$ increase in the average particle size. As expected, the same trend is observed in PSDs of the inlet and outlet mass flow rates (Figure 6).

Granulation produces larger particles when the agglomeration rate is assumed to be dependent on particle size compared to size-independent agglomeration rate. As shown in Figure 6, agglomeration with the size-dependent kernel has produced particles whose size are as large as 5 mm, while no particles with this size are produced with the size-independent kernel.

5.3 Simulations Results for Combined Process

To simulate the combined process, simultaneous particle binary agglomeration and particle growth due to layering is considered to be taking place in the drum granulator. In Figure 7, a comparison of the PSDs between pure agglomeration (with constant agglomeration kernel) and combined process is shown.

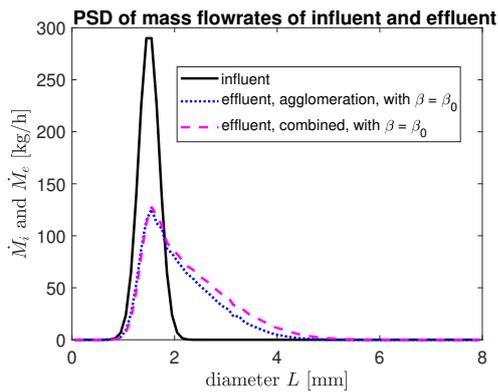


Figure 7. PSDs for pure agglomeration and combined process with size-independent kernel.

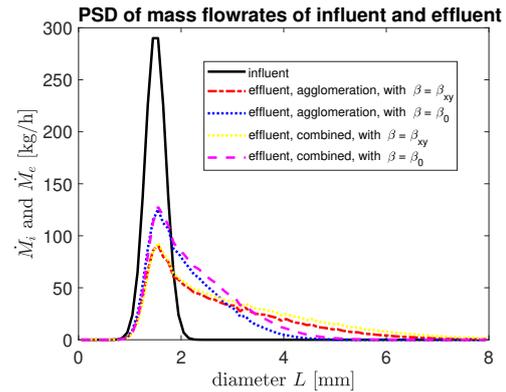


Figure 9. Comparison of PSD for pure agglomeration and combined process.

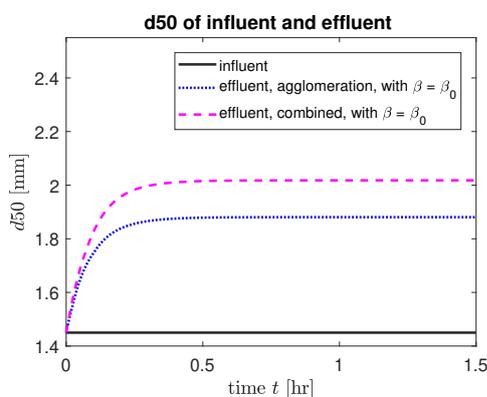


Figure 8. Change in d_{50} for pure agglomeration and combined process with size-independent kernel.

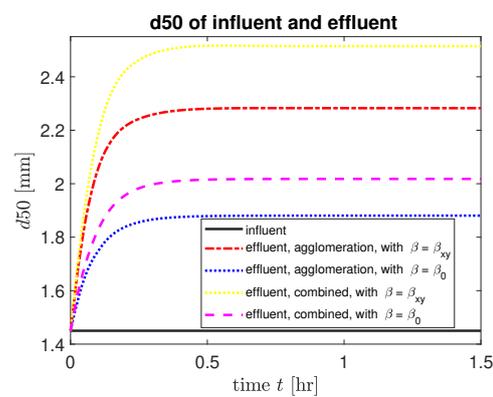


Figure 10. Comparison of d_{50} for pure agglomeration and combined process.

For the combined process, the PSD at the outlet is wider compared to the pure agglomeration granulation process. The mass fractions of coarser particles (>2 mm) become larger, and hence, larger particles are produced with the combined process. The comparison of these two granulation processes with the d_{50} plot (Figure 8) confirms the PSD shown in Figure 7. With the combined process, a higher value of d_{50} is obtained as compared to the pure agglomeration.

The d_{50} of the particles has increased by $\sim 7\%$ (with a constant agglomeration kernel) for the combined process compared to the pure agglomeration process. A similar trend of the particle size change is observed for processes when a size-dependent agglomeration kernel is used in the simulations (Figures 9 and 10). The PSD is wider, and larger particles are produced when the combined process is simulated (Figure 9). The latter is also reflected in the d_{50} plots (Figure 10). Interestingly, the value of d_{50} has grown from 2.28 mm for the pure agglomeration process to 2.52 mm for the combined process (Figure 10). This gives $\sim 10\%$ difference in average particle size for the pure and the combined process compared with the size-dependent kernel. This difference is $\sim 7\%$ when simulations are performed with the constant agglomeration kernel. Thus, particle enlargement and hence the total change

in PSD for the combined process is more intensive if particle agglomeration is driven by a size-dependent agglomeration kernel.

The plots of PSDs (Figure 9) reveal the same pattern (larger particles are produced with the combined processes compared to pure agglomeration). The PSDs of different processes start to deviate from each other when particle size fractions are larger than 1.2 mm. Granulation processes that are simulated with the constant agglomeration kernel produce more particles that are in the range of [1.2, 3.2] mm of size, compared to those processes that are simulated with the size-dependent agglomeration kernel. In contrast, the processes that are simulated by assuming size-dependent agglomeration kernels, result in a larger amount of coarse particles (≥ 3.2 mm), e.g., simulations with the size-dependent kernel produces particles with sizes as high as 6 mm, while no particles with such size is produced when the size-independent kernel is used (true for both pure agglomeration and combined process).

Based on the simulation results discussed above, the particle growth in drum granulators due to layering seems to play a minor role compared to the granulation mechanism for the particle binary agglomeration. This trend was indicated in others works (Wang et al., 2006; Wang and Cameron, 2007).

The agglomeration kernel is indeed an important parameter for modeling drum granulation processes. Not only the proper formulation of the size-independent part is needed, but also the dependency of agglomeration rate to particle size should be analyzed in order to obtain a proper model of the real plant.

6 Conclusions

In this paper, a comparative study on various model forms for representing a drum granulation process is given. Different granulation mechanisms are compared based on simulation results represented by particle size distributions and the d_{50} diameter (to reflect the average size of particles) at the influent and the effluent of the drum granulator. For the drum granulator under consideration, the simulation results lead to the following conclusions:

- Particle growth due layering has very small effect on the change of the particle sizes compared to particle binary agglomeration.
- Inclusion of the particle size dependency on the agglomeration kernel affects the mass distribution function, i.e., particles with a wider PSD and larger particles are produced compared to simulations with a constant agglomeration kernel.
- The combined process increases the growth of particles by $\sim 7\%$ (with size-independent kernel) and by $\sim 10\%$ (with size-dependent kernel) compared to a pure agglomeration process.

The choice of the agglomeration kernel directly affects the PSD of the particles. The size-independent part of the kernel should be calculated by taking into account the operational parameters of the actual drum granulator.

7 Acknowledgment

The economic support from The Research Council of Norway and Yara Technology Centre through project no. 269507/O20 'Exploiting multi-scale simulation and control in developing next generation high efficiency fertilizer technologies (HEFTY)' is gratefully acknowledged.

References

J. Drechsler, M. Peglow, S. Heinrich, M. Ihlow, and L. Mörl. Investigating the dynamic behaviour of fluidized bed spray granulation processes applying numerical simulation tools. *Chemical Engineering Science*, 60(14):3817–3833, 2005.

A.M. Golovin. The solution of the coagulation equation for raindrops, taking condensation into account. *Soviet Physics-Doklady*, 8(2):191–193, 1963.

S. Heinrich, M. Peglow, M. Ihlow, and L. Mörl. Particle population modeling in fluidized bed-spray granulation - analysis of the steady state and unsteady behavior. *Powder Technology*, 130:154–161, 2003. doi:10.1016/S0032-5910(02)00259-0.

S.M. Iveson, J.D. Litster, K. Hapgood, and B.J. Ennis. Nucleation, growth and breakage phenomena in agitated wet granulation processes: a review. *Powder technology*, 117(1-2):3–39, 2001.

PC Kapur. Kinetics of granulation by non-random coalescence mechanism. *Chemical Engineering Science*, 27(10):1863–1869, 1972.

P.C. Kapur and D.W. Fuerstenau. Coalescence model for granulation. *Industrial & Engineering Chemistry Process Design and Development*, 8(1):56–62, 1969.

B. Koren. A robust upwind discretization method for advection, diffusion and source terms. In C. B. Vreugdenhil and B. Koren, editors, *Numerical Methods for Advection-Diffusion Problems, Notes on Numerical Fluid Mechanics*, pages 117–138. 1993.

J. Kumar. *Numerical approximations of population balance equations in particulate systems*. PhD thesis, Otto-von-Guericke-Universität Magdeburg, Universitätsbibliothek, 2006.

J. Kumar, M. Peglow, G. Warnecke, S. Heinrich, and L. Mörl. Improved accuracy and convergence of discretized population balance for aggregation: The cell average technique. *Chemical Engineering Science*, 61(10):3327–3342, 2006.

J. Litster and B. Ennis. *The science and engineering of granulation processes*, volume 15. Springer Science & Business Media, 2004.

MATLAB. 2017a. The MathWorks, Inc., Natick, Massachusetts, United States., 2017.

L. Mörl. *Anwendungsmöglichkeiten und Berechnung von Wirbelschichtgranulationstrocknungsanlagen*. PhD thesis, Technische Hochschule Magdeburg, 1981.

L. Mörl, M. Mittelstrab, and J. Sachse. Zum kugelwachstum bei der wirbelschichttrocknung von suspensionen oder losungen. *Chemical Technology*, 29(10):540–541, 1977.

R. Radichkov, T. Müller, A. Kienle, S. Heinrich, M. Peglow, and L. Mörl. A numerical bifurcation analysis of continuous fluidized bed spray granulator with external product classification. *Chemical Engineering and Processing*, 45:826–837, 2006. doi:10.1016/j.cep.2006.02.003.

D. Ramkrishna. *Population balances: Theory and applications to particulate systems in engineering*. Academic press, 2000.

A.D. Randolph and M.A. Larson. Transient and steady state size distributions in continuous mixed suspension crystallizers. *AIChE Journal*, 8(5):639–645, 1962.

F.Y. Wang and I.T. Cameron. A multi-form modelling approach to the dynamics and control of drum granulation processes. *Powder Technology*, 179(1-2):2–11, 2007.

F.Y. Wang, X.Y. Ge, N. Balliu, and I.T. Cameron. Optimal control and operation of drum granulation processes. *Chemical Engineering Science*, 61(1):257–267, 2006.

Convective Melting Modeling Approach for Phase Change Materials with Variable Boundary Heating

D. Rubineti¹ D. A. Weiss¹ A. Chaudhuri² D. Kraniotis²

¹Institute of Thermal- and Fluid-Engineering, University of Applied Sciences and Arts Northwestern Switzerland

²Department of Civil Engineering and Energy Technology, OsloMet — Oslo Metropolitan University, Norway

Abstract

The present study conceives a numerical model for phase change materials following the apparent heat capacity method where the phase change occurs within a chosen temperature interval. A multiphysical modeling approach to satisfy the coupled momentum, energy and continuity conservation equations whilst avoiding numerical singularities is applied. By means of a 2D test-case geometry with variable boundary heating the influence of natural convection within the melted liquid zone is visualized. Corresponding non-dimensional governing equations are analysed to quantify the dominant contributing terms. It turns out that for sufficiently small Grashof number, or consequently small Rayleigh numbers the influence of natural convection can be neglected, thus simplifying the problem substantially. The modeling approach has been adapted to a 2D-axisymmetric geometry within the scope of experimental validation. The simulation results and experimental data show reasonably good agreement.

The model is numerically stable and suitable to facilitate design of latent heat storage systems.

Keywords: Phase Change Materials, Fluid Flow, Heat Transfer, Phase Transition, Natural Convection, Melting, Latent Heat Storage

1 Introduction

Phase change material (PCM) finds an increasingly wide range of applications in thermal energy management due to their energy storage capabilities. The choice of a suitable PCM should satisfy physical, technical and economic requirements. In essence, these consist of materials with a characteristically large enthalpy of fusion in a temperature range smeared over a phase transition region. In order to appropriately design latent heat storage applications using PCMs, it is important to predict the complex behavior of the materials well enough. In the present study, a numerical model for implementation in COMSOL MultiphysicsTM is conceived. Natural convection is accounted for by the Boussinesq approximation which allows the fluid to be treated as incompressible for inertial forces and compressible for buoyancy forces. The chosen material in this study is *n*-eicosane as specified in table 1, whose thermophysical properties come with great deal of experimental coverage (Muhammad et al., 2015;

Jones et al., 2006; Sparrow and Broadbent, 1982).

2 Physical Model

To model the physical nature of PCMs it is necessary to consider the disciplines of heat transfer and fluid flow. The strong coupling occurs due to natural convection and phase change effects constrained by the PCM container geometry as illustrated by figure 1.

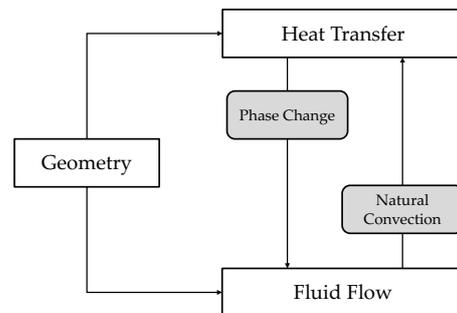


Figure 1. Coupled Physics.

The chosen geometry is a 2D square enclosure with 1 cm side-length (H). Figure 2 shows the computational setup. On the right wall a Dirichlet boundary condition for three different temperatures is applied: $T_R = 40/50/70^\circ\text{C}$. The three remaining boundaries represent thermal insulation. Boundaries are also prescribed as no-slip walls wherever applicable.

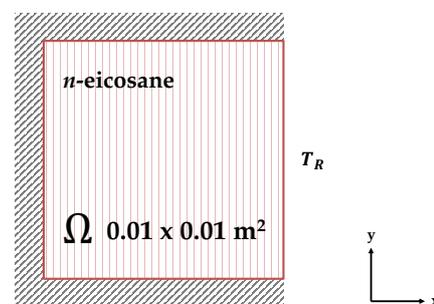


Figure 2. Geometry of the 2D model.

The melting process occurs within a time-range of $0 < t < 1000$ s for all temperatures.

NOMENCLATURE

A_m	mushy zone coefficient	$\text{kg m}^{-3} \text{s}^{-1}$
C_p	specific heat capacity at constant pressure	$\text{J kg}^{-1} \text{K}^{-1}$
D	Gaussian distribution	-
F	volumetric force on fluid	N m^{-3}
g	gravitational acceleration	m s^{-2}
H	domain length	m
k	thermal conductivity	$\text{W m}^{-1} \text{K}^{-1}$
L	latent heat / heat of fusion	J kg^{-1}
p	pressure	Pa
R	radius	m
r	radial coordinate	m
T	temperature	K
ΔT	phase transition temperature range	K
T_m	melting temperature	K
T_R	wall temperature	K
t	time	s
u	velocity	m s^{-1}
u_0	characteristic velocity	m s^{-1}
x	x-coordinate	m
y	y-coordinate	m
z	z-coordinate	m
α	thermal diffusivity	$\text{m}^2 \text{s}^{-1}$
β	thermal expansion coefficient	K^{-1}
ε	Carman-Kozeny constant	-
μ	dynamic viscosity	Pas
ν	kinematic viscosity	$\text{m}^2 \text{s}^{-1}$
Φ_v	dissipation function	s^{-2}
ρ	density	kg m^{-3}
θ	melt fraction	-
Br	Brinkman number	-
Gr	Grashof number	-
Pr	Prandtl number	-
Ra	Rayleigh number	-
Re	Reynolds number	-
Super- and subscripts		
l	liquid	
ref	reference quantity	
s	solid	
\sim	dimensionless quantity	

Table 1. Properties of *n*-eicosane

	<i>Solid</i>	<i>Liquid</i>
ρ	910	769
k	0.423	0.146
C_p	1926	2400
β	-	8.161×10^{-4}
T_m	36.4	-
L	248000	-

3 Numerical Model

Modeling of phase change phenomena can be achieved by splitting the computation domain Ω into a subdomain for the solid fraction and a subdomain for the liquid fraction. A major numerical drawback hereof is the difficulty of tracking the solid/liquid interface which require delicate adaptive meshing methods to guarantee both convergence and reasonable representation of the physical processes (Lewis et al., 2004). The predominant state-of-art approach for the modeling of phase change materials is based on enthalpy methods, *i.e.* the enthalpy-porosity formulation (Dutil et al., 2011). The enthalpy-porosity formulation requires only one mathematical model. By means of temperature-dependent material laws the material is considered both as solid and liquid while the phase transition region is denoted as mushy region.

3.1 Material description

The enthalpy-porosity method bases all material properties such as melt fraction, heat capacity, density and thermal conductivity to the current temperature T of the material. The enthalpy-porosity formulation quantifies the melt fraction $\theta(T)$ in terms of a temperature-dependent ramp function from 0 to 1 over a chosen phase transition temperature interval ΔT . It is centered around the melting threshold temperature T_m (Kheirabadi and Groulx, 2015) as,

$$\theta(T) = \begin{cases} 0, & \text{for } T < T_m - \Delta T/2 \\ \frac{T - (T_m - \Delta T/2)}{\Delta T}, & \text{for } T_m - \Delta T/2 < T < T_m + \Delta T/2 \\ 1, & \text{for } T > T_m + \Delta T/2. \end{cases} \quad (1)$$

A graphical representation of the melt fraction function is shown in figure 3a. To account for the contribution of latent heat over the phase transition region, a Gaussian distribution function $D(T)$ ensures energy conservation over the chosen temperature interval as illustrated in figure 3b. $D(T)$ is given by,

$$D(T) = \frac{e^{-\frac{(T - T_m)^2}{(\Delta T/4)^2}}}{\sqrt{\pi(\Delta T/4)^2}}. \quad (2)$$

The definitions according to equation (1) and (2) allow the modification of the heat capacity C_p of the PCM as,

$$C_p(T) = C_{p,s} + \theta(T)(C_{p,l} - C_{p,s}) + D(T)L. \quad (3)$$

Note that, the heat capacity increases notably in the mushy region due to the latent heat contribution as seen in figure 3c. Temperature-dependent thermal conductivity and density are defined analogously like the heat capacity without latent heat contribution term.

In numerical point of view, the choice of the temperature transition range ΔT requires careful consideration. The

sharp gradients of the modified heat capacity may lead to convergence issues if ΔT is small (*i.e.* ~ 0.1 K) and the mesh resolution at the interface does not cover the imposed range. A small ΔT means greater accuracy coming with greater computational effort. A qualitative initial choice in the range of 5 K may be used (Murray and Groulx, 2011). A reasonable trade-off between accuracy and computational effort in the test-cases of the present study is the chosen value of 2 K. The work is carried out with the commercial tool COMSOL MultiphysicsTM which offers a suitable user-input environment for implementing the enthalpy-porosity method.

3.2 Governing Equations

The governing equations to describe the physics include the laminar Navier-Stokes equations, the continuity equation and the energy conservation equation. Here the set of equations for a successful implementation in COMSOL MultiphysicsTM is presented.

3.2.1 Continuity equation

The continuity equation assuming constant density is defined as follows

$$\nabla \cdot \mathbf{u} = 0. \quad (4)$$

3.2.2 Momentum equation

The incompressible momentum equation assumes constant density yielding

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p + \nabla \cdot (\mu(T)\nabla\mathbf{u}) + S(T)\mathbf{u} + \mathbf{F}, \quad (5)$$

where $S(T)\mathbf{u}$ is an additional modeling term to account for the flow within the mushy zone. Its component $S(T)$ is described by the Carman-Kozeny equation as

$$S(T) = A_m \frac{(1 - \theta(T))^2}{\theta(T)^3 + \varepsilon}, \quad (6)$$

with the modeling constants $A_m = 10^6 \text{ kg m}^{-3} \text{ s}^{-1}$ and $\varepsilon = 10^{-3}$ (Samara et al., 2012; Kheirabadi and Groulx, 2015). The function $S(T)$ forms a sink-term in the momentum equation. For the solid fraction, the momentum equation (5) shall return the trivial $\mathbf{u} = 0$ solution to ensure immobility. As illustrated in figure 3d, the $S(T)$ affects the solid fraction as well as the solid-dominated part of the mushy zone ($T < T_m$). The modeling term $S(T)$ overrides every other terms in the momentum equation in the solid region. To achieve convergence in the model, however, it is recommended to additionally modify the viscosity (Kheirabadi and Groulx, 2015) in COMSOL MultiphysicsTM as below

$$\mu(T) = \mu_l(1 + S(T)), \quad (7)$$

with the empirical viscosity-temperature relation (Muhammad et al., 2015) for *n*-eicosane,

$$\mu_l = (9 \times 10^{-4}T^2 - 0.6529T + 119.94) \times 10^{-3}. \quad (8)$$

where T is given in [K] and the dynamic viscosity μ_l in [Pa s]. Consequently the dissipation term in the momentum equation overrides the other terms within the solid regime. That way it is guaranteed that flow is induced only in the liquid fraction within the computational domain despite of solving the momentum equation for both solid and liquid fraction.

The force term accounting for gravity and natural convection is given by the Boussinesq approximation. This is implemented as (Bird et al., 2007),

$$\mathbf{F} = \rho_l(1 - \beta(T - T_m))\mathbf{g}. \quad (9)$$

3.2.3 Energy equation

The energy equation in terms of the temperature distribution T is defined by

$$\rho(T)C_p(T)\frac{\partial T}{\partial t} + \rho(T)C_p(T)\mathbf{u} \cdot \nabla T = \nabla \cdot (k(T)\nabla T). \quad (10)$$

3.3 Mesh

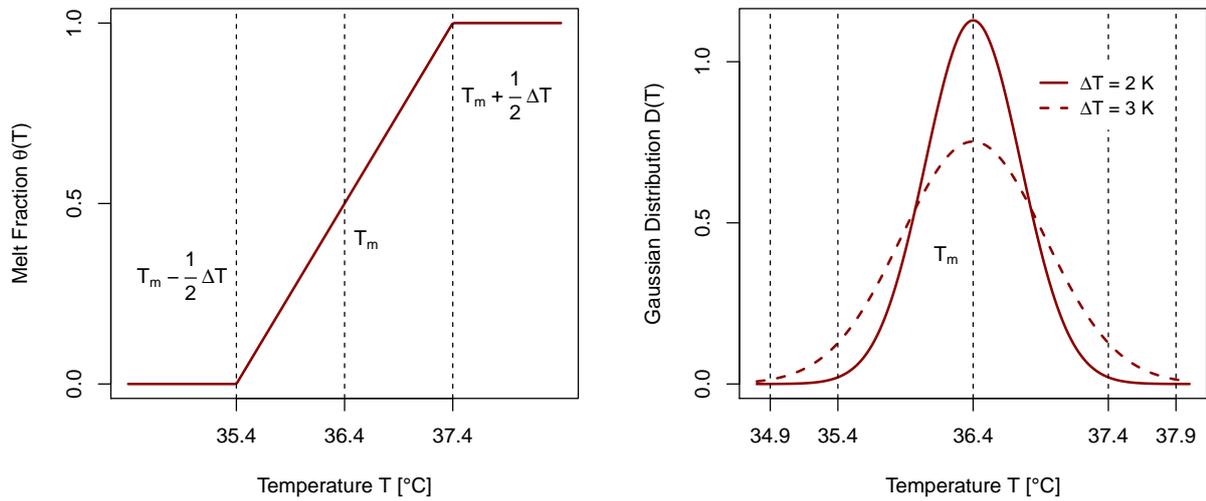
The mesh for the test-case geometry is a mapped mesh with 40000 quadrilateral elements. The mesh is strategically refined towards the heated right boundary and on the top boundary to tackle potential complications with no-slip walls during early stages of melting when the velocities within the fluid fraction are close to zero.

On a high-performance cluster with 16 CPUs and 94 GB RAM the computational runtime amounts to approximately 7.5 hours for a simulation time of $t = 1000\text{s}$. To ensure convergence it is recommended to use a fully coupled direct solver configured for the Newton nonlinear method.

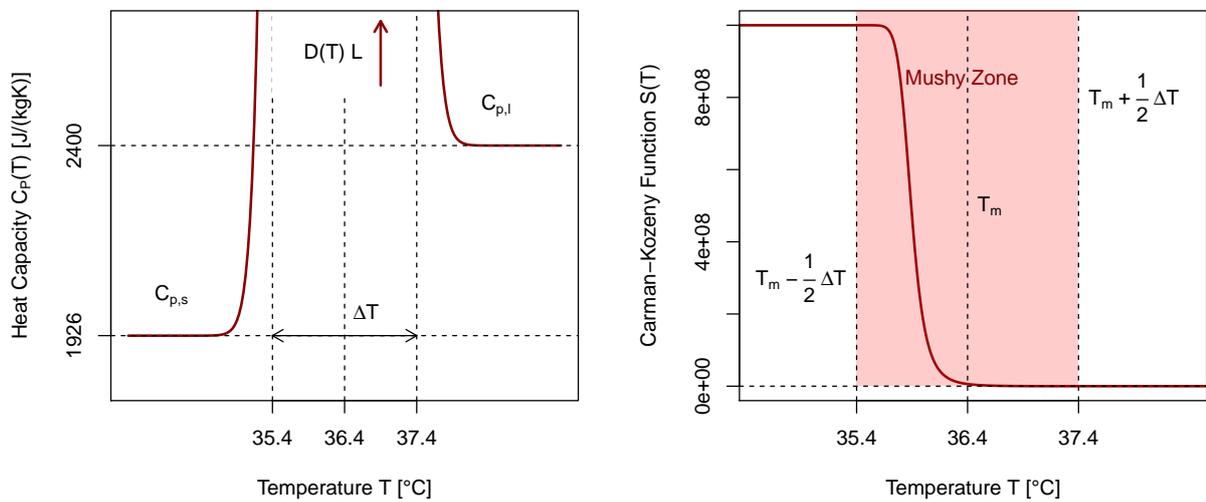
4 Results

The results for all test cases are summarized in figure 4. It appears that the melt fraction has a strong dependence on temperature, indicating that the natural convection plays an important role. To quantify the importance of natural convection the non-dimensional forms of the governing equations for non-isothermal flow are investigated. A temperature-independent liquid viscosity of $\mu_l = 0.008$ Pa s (Kheirabadi and Groulx, 2015) in the simulation and the following non-dimensional variables are used (Bird et al., 2007)

$$\begin{aligned} \tilde{x} &= \frac{x}{H} & \tilde{y} &= \frac{y}{H} & \tilde{p} &= \frac{p - p_{ref}}{\rho u_0^2} \\ \tilde{t} &= \frac{u_0 t}{H} & \tilde{\mathbf{u}} &= \frac{\mathbf{u}}{u_0} & \tilde{T} &= \frac{T - T_{ref}}{T_R - T_{ref}} \\ \tilde{\Phi}_v &= \left(\frac{H}{u_0}\right)^2 \Phi_v & \tilde{\nabla} &= H \nabla & \frac{D}{D\tilde{t}} &= \left(\frac{H}{u_0}\right) \frac{D}{Dt}. \end{aligned}$$



(a) Melt fraction function $\theta(T)$, implemented as piecewise function. (b) Gaussian distribution function $D(T)$, shown for different ΔT s.



(c) Heat capacity function $C_p(T)$, rising up to a maximum value of $D(T)L = 272000 \text{ J kg}^{-1} \text{ K}^{-1}$. (d) Carman-Kozeny function to model the sink-term and viscosity of the solid fraction.

Figure 3. Temperature-dependent modeling functions as presented by (Kheirabadi and Groulx, 2015) for the chosen PCM *n*-eicosane.

with $u_0 = v/H$ being the characteristic velocity. Thus the continuity equation yields

$$\tilde{\nabla} \cdot \tilde{\mathbf{u}} = 0.$$

The momentum equation is expressed in terms of the total derivative and contains two dimensionless groups

$$\frac{D\tilde{\mathbf{u}}}{D\tilde{t}} = -\tilde{\nabla}\tilde{p} + \left[\frac{\mu}{u_0\rho H} \right] \tilde{\nabla}^2\tilde{\mathbf{u}} - \left[\frac{g\beta(T_R - T_m)H}{u_0^2} \right] \left(\frac{\mathbf{g}}{g} \right) (\tilde{T} - \tilde{T}_m),$$

whereas the energy equation features a dissipation term and writes

$$\frac{D\tilde{T}}{D\tilde{t}} = \left[\frac{k}{u_0\rho HC_p} \right] \tilde{\nabla}^2\tilde{T} + \left[\frac{\mu u_0}{\rho HC_p(T_R - T_m)} \right] \tilde{\Phi}_v.$$

The values of the four dimensionless groups for the present test-case are summarized in table 2 with the following relevant dimensionless numbers

Brinkman	$Br = \frac{\mu u_0^2}{k\Delta T},$
Grashof	$Gr = \frac{g\beta\Delta TH^3}{\nu^2},$
Prandtl	$Pr = \frac{\nu}{\alpha},$
Rayleigh	$Ra = \frac{g\beta\Delta TH^3}{\alpha\nu} = GrPr,$
Reynolds	$Re = \frac{\rho u_0 H}{\mu}.$

As can be seen from the evaluation the sharpest increase with temperature is the buoyancy term in the momentum equation. Comparing with the simulation results in figure 4, it is safe to assume that natural convection can be neglected for $Gr/Re^2 < 266$ which reduces the model complexity substantially as the energy equation alone is sufficient to investigate the phase change without CFD coupling.

4.1 Validation

The implemented model is validated using experimental data of (Jones et al., 2006). Their experimental setup with *n*-eicosane in a cylindrical enclosure as illustrated in figure 5 with the wall temperature $T_R = 45^\circ\text{C}$ and simulation time $t = 18000$ s. Additionally we set a subcooling wall at the bottom with $T_B = 32^\circ\text{C}$ with the effect that a part of the enclosure shall remain solid according to (Jones et al., 2006).

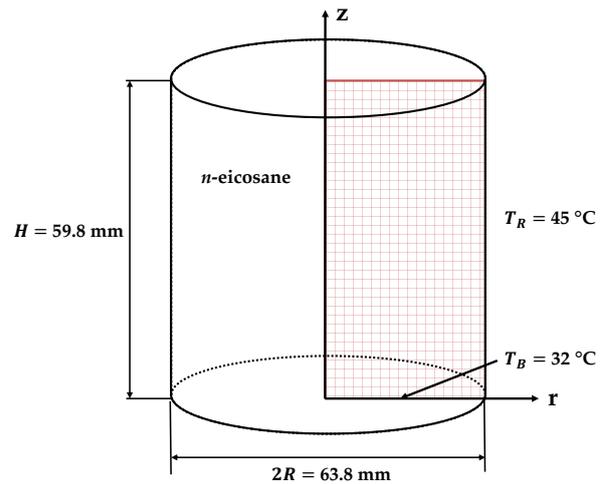


Figure 5. 2D axisymmetric geometry for experimental validation

The computational cost on the same high-performance cluster increases for the validation case to approximately 25 hours.

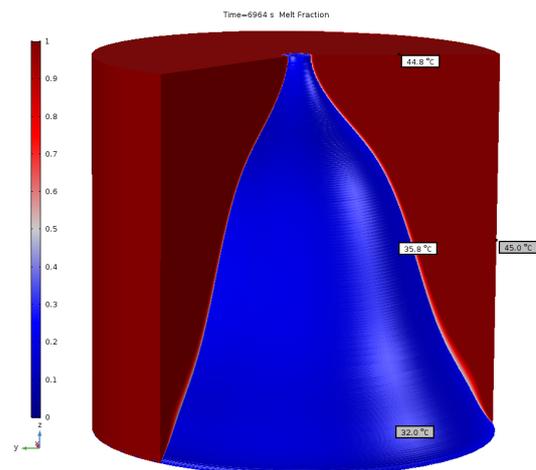


Figure 6. Melt fraction at $t = 6964$ s with temperature labels.

The evolution of the melting front show reasonably good agreement with the experimental results as shown in figure 7.

4.2 Discussion

The dimensionless estimation of the terms as indicated in table 2 can be of useful guidance when setting up a phase change related model. In fact, for low *Gr* and *Ra* numbers, the model might deliver acceptable results solving the energy equation only. Low *Gr* and *Ra* numbers are achieved when the temperature difference $T_R - T_m$ has approximately the same order of magnitude as the phase transition temperature range ΔT . The analysis with dimensionless numbers presented here targets the liquid fraction only. Within the mushy zone a different behavior can be expected, *i.e.* the dissipation term in the energy equation will override the other terms due to a sharp increase in vis-

Table 2. Dimensionless groups in non-isothermal flow equations

Dimensionless group	T_R		
	40 °C	55 °C	70 °C
$\left[\frac{\mu}{u_0 \rho H} \right] = \frac{1}{\text{Re}}$	1	1	1
$\left[\frac{g\beta(T_R - T_m)H}{u_0^2} \right] = \frac{\text{Gr}}{\text{Re}^2} = \frac{\text{Ra}}{\text{PrRe}^2}$	266	1376	2486
$\left[\frac{k}{u_0 \rho H C_p} \right] = \frac{1}{\text{RePr}}$	0.008	0.008	0.008
$\left[\frac{\mu u_0}{\rho H C_p (T_R - T_m)} \right] = \frac{\text{Br}}{\text{RePr}}$	1.25×10^{-10}	2.42×10^{-11}	1.34×10^{-11}

cosity due to the Carman-Kozeny porosity function $S(T)$ as seen in equation (7).

Nonetheless, the modeling approach can seamlessly be adapted to other materials. The validation case shows reasonably good agreement between simulation and experimental results. The deviations can be explained by uncertainties in the retrieval of experimental data (Jones et al., 2006) and in the choice of modeling parameters, e.g. ΔT . Smearing out the phase transition over a temperature range ΔT is questionable in terms of physical accuracy as its purpose and necessity is the handling of numerical singularities. Further points to consider for model improvement include

- mesh independence study
- quantification of viscous dissipation in the mushy zone
- volumetric expansion during melting
- aging of the material.

5 Conclusion

This work provides a modeling guideline for computing phase change phenomena on a stationary grid, coupling heat transfer and fluid dynamics. From the results it can be seen that the increasing influence of natural convection also increases to the amount of molten material. While the dimensionless estimation delivers reliable reference values for an estimation, the simulation provides visuals to spot further design improvements. Of particular interest are the local (grid) Gr resp. Ra numbers which indicate the exact region of influence within the geometry.

The presented model for the implementation of phase change phenomena in COMSOL MultiphysicsTM is numerically stable and physically accurate enough to facilitate the design of latent heat storage systems.

References

- Byron R. Bird, Warren E. Stewart, and Edwin N. Lightfoot. *Transport Phenomena*. Wiley, New York, second edition edition, 2007.
- Yvan Dutil, Daniel R. Rousse, Nizar Ben Salah, Stéphane Lasue, and Laurent Zalewski. A review on phase-change materials: mathematical modeling and simulations. *Renewable and sustainable Energy reviews*, 15(1):112–130, 2011. doi:10.1016/j.rser.2010.06.011.
- Benjamin J. Jones, Dawei Sun, Shankar Krishnan, and Suresh V. Garimella. Experimental and numerical study of melting in a cylinder. *International Journal of Heat and Mass Transfer*, 49(15-16):2724–2738, 2006.
- Ali C. Kheirabadi and Dominic Groulx. Simulating phase change heat transfer using comsol and fluent: Effect of the mushy-zone constant. *Computational Thermal Sciences: An International Journal*, 7(5-6), 2015. doi:10.1615/ComputThermalScien.2016014279.
- Roland W. Lewis, Perumal Nithiarasu, and Kankanhalli N. Seetharamu. *Fundamentals of the finite element method for heat and fluid flow*. John Wiley and Sons, 2004.
- M.D. Muhammad, O. Badr, and H. Yeung. Validation of a cfd melting and solidification model for phase change in vertical cylinders. *Numerical Heat Transfer, Part A: Applications*, 68(5):501–511, 2015. doi:10.1080/10407782.2014.994432.
- Robynne E. Murray and Dominic Groulx. Modeling convection during melting of a phase change material. In *Proceedings of the COMSOL Conference*, 2011.
- Farid Samara, Dominic Groulx, and Pascal H. Biwolé. Natural convection driven melting of phase change material: comparison of two methods. In *Excerpt from the Proceeding of the COMSOL Conference*, 2012.
- E.M. Sparrow and J.A. Broadbent. Inward melting in a vertical tube which allows free expansion of the phase-change medium. *Journal of Heat Transfer*, 104(2):309–315, 1982. doi:10.1115/1.3245089.

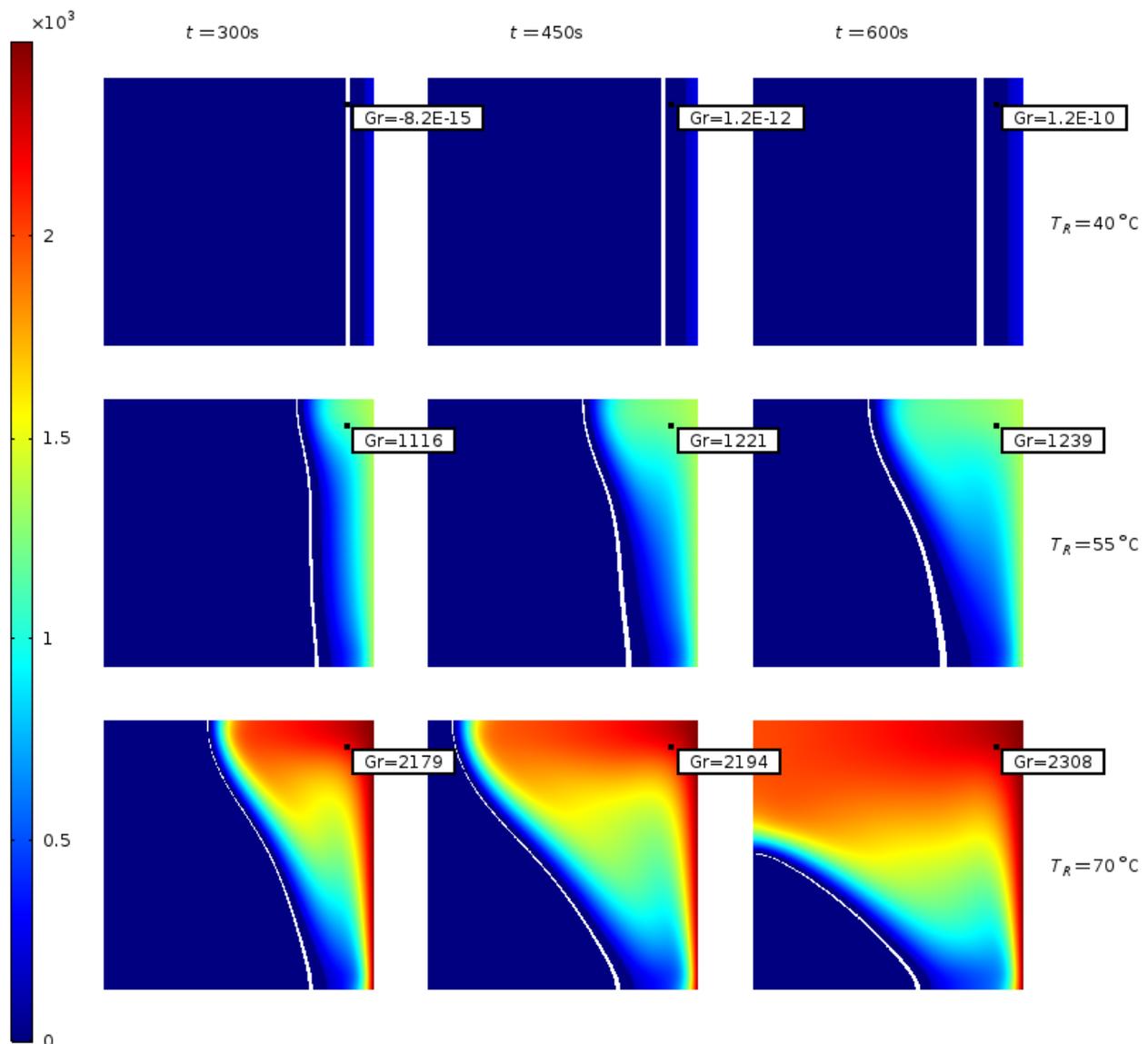
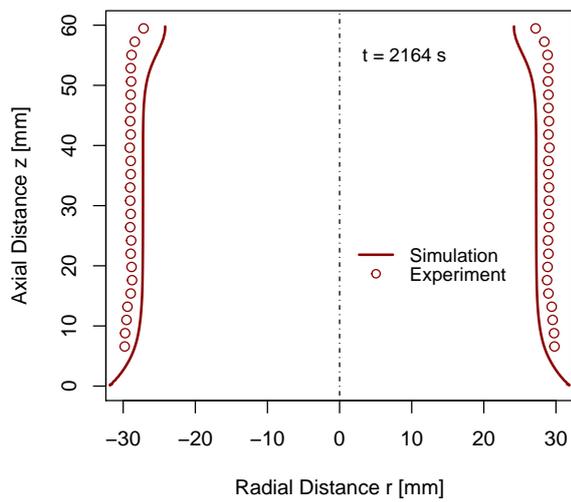
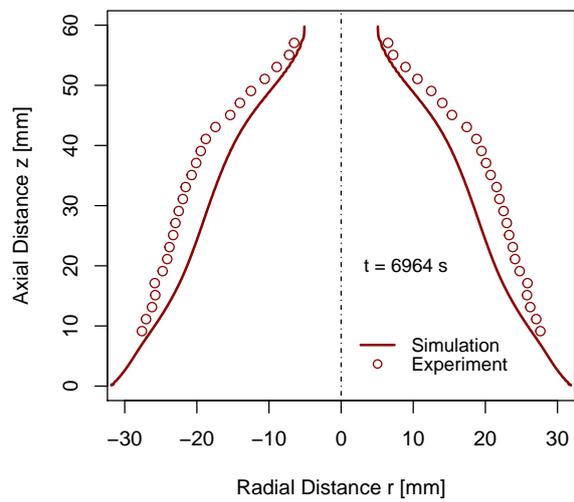


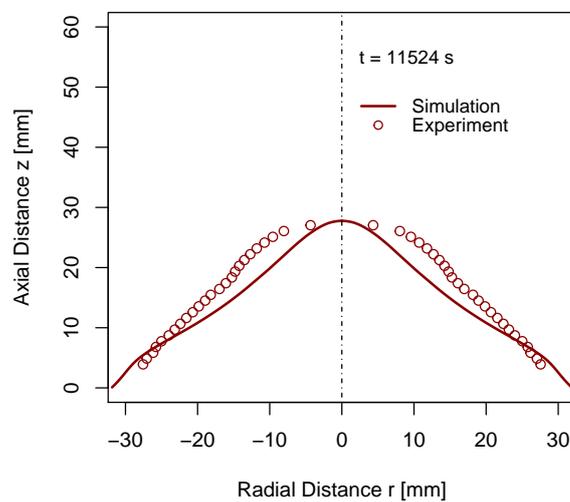
Figure 4. Simulation result of the grid Grashof number with white isoline tracking the melting front. As expected from the computed values in table 2, Gr increases strongly with temperature. Due to the constant viscosity μ_l and thermal diffusivity α the Gr number is directly linked to Rayleigh number by a scaling factor of $1/Pr = \alpha\rho_l/\mu_l$. Thus, values of Gr can be used as reference to measure the intensity of buoyancy resp. driving force of natural convection. The simulation results visualize the main regions of influence of the convective flow; for $T_R = 40\text{ °C}$ the front moves parallel away from the heated boundary insinuating no contribution from the momentum equation. On the other extreme, the $T_R = 70\text{ °C}$ case clearly shows how the phase interface bends under influence of convection.



(a) Interface location at $t = 2164$ s.



(b) Interface location at $t = 6964$ s.



(c) Interface location at $t = 11524$ s.

Figure 7. Comparison of simulation results with experimental data (Jones et al., 2006).

Shock propagation and diffraction through cavity

Arnab Chaudhuri

Department of Civil Engineering and Energy Technology
OsloMet — Oslo Metropolitan University, Oslo, Norway, arnab.chaudhuri@oslomet.no

Abstract

This work presents a numerical analysis of a planar moving shock wave with Mach number $M_s = 1.3$, travelling through a square cavity geometry with rigid boundaries. A high-order artificial viscosity based Discontinuous Spectral Element Method (DSEM) is used for this purpose. The explicit numerical scheme utilizes entropy generation based transport coefficients to solve the conservative form of the viscous compressible fluid flow equations. Numerical prediction of the shock propagation and diffraction is found to be in excellent agreement with the experimental results of the literature. The stable numerical scheme resolves the detail of the complex flow dynamics for varying reference Reynolds number (Re_f). The range of values of the artificial coefficients and the relative contribution of the components of the artificial energy dissipation rate are investigated and compared for different cases. Artificial energy dissipation is less for low Re_f . The dilatational dissipation dominates over other components till the incident shock wave resides in the flow domain.

Keywords: shock propagation, shock diffraction, shock reflection, DSEM, artificial viscosity

1 Introduction

Shock/Blast wave mitigation is an important research domain for disaster management. Shock wave propagation through solid structures are associated with complex shock-shock/shock-vortex/shock-boundary layer interactions. Understanding the fundamental features of shock wave propagation and diffraction is important for design of suitable mechanism for shock wave/ blast wave attenuation. Attenuation can be achieved by various means, e.g. utilising solid obstacles, foams, textiles, porous materials, granular filters, metallic grids, perforated plates, branched/bend duct, duct with rough walls, etc. Investigation related to shock/blast wave interaction with various configurations is consequently remaining as an active research field.

Apart from the experimental measurements, numerical approach can predict much more detailed flow evolution of such applications. Igra et al. summarizes both experimental and numerical works for many of these approaches (Igra et al., 2013). A detailed computational study of shock wave attenuation through solid obstacles are presented in (Chaudhuri et al., 2013). Application of abrupt changes in tunnel geometry is shown in (Igra et al., 1996,

1998, 2001). Most of the numerical simulations of shock diffraction problems of literature relied on inviscid simulations. However, viscous interactions become significant for shock-boundary layer/boundary layer-shear layer interactions and for long term shock-vortex dynamics. For example, the classical shock diffraction over 90° sharp corner requires the consideration of no-slip walls in numerical approach to predict the secondary vortex and long-time evolution of flow dynamics (Sun and Takayama, 2003; Skews et al., 2012; Law et al., 2014; Chaudhuri and Jacobs, 2017).

It is desirable to use high-order numerical schemes together with stable shock capturing property to resolve the shocked flow dynamics. Numerical studies exploiting high-order schemes successfully reveal the general wave characteristics and complex flow dynamics of various phenomena associated with shock wave interaction with rigid bodies/boundaries like shock diffraction, reflection, focusing etc. (Chaudhuri et al., 2012; Glazer et al., 2011; Shadloo et al., 2014; Soni et al., 2017). The present work is aimed to perform viscous simulations and detailed analysis of shock propagation through the cavity like geometries/bend duct with cavities. A numerical analysis of a planar shock ($M_s = 1.3$) travelling through a two dimensional square cavity is carried out. A high-order artificial viscosity based DSEM is used for this purpose (Chaudhuri et al., 2017). A particular focus on the estimation of entropy generation based artificial viscosity coefficients and artificial viscous energy dissipation for various flow Reynolds number has been made.

The paper is organized as follows. A brief description of the governing equations is given in section 2. The overall numerical approach is presented in section 3. In section 4, the problem setup for the present study is illustrated. Comparison of the findings with experimental results and flow analysis are discussed in section 5. Finally, conclusions are drawn in section 6.

2 Governing equations

Shock wave propagation through a gaseous medium is governed by the compressible Navier-Stokes (NS) system of equations. The non-dimensional form of the governing equations with artificial transfer coefficients is given by,

$$\frac{\partial \mathbf{Q}}{\partial t} + \nabla \cdot \mathcal{F}^a(\mathbf{Q}) - \frac{1}{Re_f} \nabla \cdot \mathcal{F}^v(\mathbf{Q}) = 0, \quad (1)$$

and

$$\mathbf{Q} = \begin{pmatrix} \rho \\ \rho \mathbf{v} \\ \rho E_t \end{pmatrix}, \mathcal{F}^a(\mathbf{Q}) = \begin{pmatrix} \rho \mathbf{v} \\ \rho \mathbf{v} \mathbf{v} + p \overline{\overline{\delta}} \\ (\rho E_t + p \overline{\overline{\delta}}) \mathbf{v} \end{pmatrix},$$

$$\mathcal{F}^v(\mathbf{Q}) = \begin{pmatrix} 0 \\ \overline{\overline{\tau}} \\ \overline{\overline{\tau}} \mathbf{v} + \frac{\kappa_{\text{eff}}}{(\gamma-1)M_f^2 \text{Pr}_f} \nabla T \end{pmatrix},$$

where \mathbf{Q} is the conservative solution vector, \mathcal{F}^a and \mathcal{F}^v are the inviscid and viscous flux respectively. ρ is the density, \mathbf{v} is the velocity vector, and E_t is the total internal energy. p is the static pressure and T is the temperature. γ is the ratio of specific heats. $\overline{\overline{\delta}}$ is the Kronecker delta tensor. Here, M_f , Re_f and Pr_f are the reference Mach number, Reynolds number, and Prandtl number, respectively.

Assuming Stokes' hypothesis with zero bulk viscosity the viscous shear stress tensor can be written as, $\overline{\overline{\tau}} = 2\mu_{\text{eff}}\mathbf{S} - \frac{2}{3}\mu_{\text{eff}}(\nabla \cdot \mathbf{v})\overline{\overline{\delta}}$, where $\mu_{\text{eff}} = \mu_h \text{Re}_f + \mu$ is the effective dynamic viscosity and $\mathbf{S} = \frac{1}{2}[(\nabla \mathbf{v})^T + \nabla \mathbf{v}]$ is the symmetric part of the velocity gradient tensor. The superscript "T" designate a transpose. Similarly, $\kappa_{\text{eff}} = \kappa_h \text{Re}_f + \kappa$ is the effective thermal conductivity. Note that, μ_h and κ_h are yet to be determined artificial transfer coefficients. For NS system of equations $\mu = \kappa = 1$. In this study, $\gamma=1.4$, $M_f = 1$ and $\text{Pr}_f = 0.72$ are prescribed. The ideal gas equation of state, $p = \rho T / (\gamma M_f^2)$, closes Eqn. 1.

3 Numerical approach

The staggered Chebyshev collocation method is used to approximate the compressible NS system of equations with explicit time marching algorithms. In the nodal collocation formulation of the DSEM, the physical domain is subdivided with hexahedral physical elements. Each physical element is then mapped to a unit cubic computational element in the computational domain by an isoparametric transformation. The solution vector is collocated at the Chebyshev-Gauss quadrature points while the fluxes are collocated at the Chebyshev-Lobatto quadrature points. The detail of the discretization procedure can be found in (Kopriva, 1998; Jacobs et al., 2005).

The numerical scheme utilizes entropy generation based artificial transport coefficients to address Gibb's oscillation. Artificial viscosity coefficients are scaled with viscous and thermal entropy generation terms of the entropy transport equation. A shock sensor based switch is used to optimize the artificial coefficients. Also, the upper bound of artificial coefficients ensures that the inviscid time step remains smaller than the viscous time step. The basic idea of an artificial viscosity method is to explicitly add even-order dissipation term to stabilize the scheme. However, artificial viscosity methods require arbitrary model constants as flow-dependent tuning parameters for the optimal solution. A brief description of the method of calculation of the artificial viscosity coefficients is presented

here again for better clarity. The artificial momentum and thermal conductivity are linked with the viscous and conductive, entropy generating terms Φ and Γ respectively. In non dimensional form this yields the following expression for the artificial viscosity coefficients: (Spurk and Aksel, 2008)

$$\mu_h = C_\mu \frac{\rho(\Delta h)^2}{\|\rho s - \overline{\rho s}\|_\infty} \left[\frac{\Phi}{T} \right], \quad (2)$$

and

$$\kappa_h = C_\kappa \frac{\rho(\Delta h)^2}{\|\rho s - \overline{\rho s}\|_\infty} \left[\frac{1}{\text{Pr}_f, (\gamma-1)M_f^2} \frac{\Gamma}{T} \right], \quad (3)$$

where C_μ and C_κ are model parameters, Δh is the mesh size, $\rho s = \frac{\rho}{\gamma(\gamma-1)M_f^2} \ln\left(\frac{p}{\rho^\gamma}\right)$ and $\overline{\rho s}$ is the spatial average of ρs . Here $\|\rho s - \overline{\rho s}\|_\infty$ is the globally computed supremum based on the global average entropy. The artificial viscosity coefficients defined above ensure the positivity (thus dissipative behavior) and μ_h and κ_h scale with the grid spacing and vanish as $\Delta h \rightarrow 0$.

A shock sensor θ (Ducros et al., 1999) is used to control the artificial viscosity coefficients as $\mu_h \theta \mathcal{H}(-\nabla \cdot \mathbf{v})$, and $\kappa_h \theta$. This in turn reduces the dissipation in rotation-dominated regions. The purpose of the Heaviside function $\mathcal{H}(-\nabla \cdot \mathbf{v})$ is to ensure that the dissipation is small in regions of isentropic expansion fans and contact discontinuities. The coefficients are controlled so that the inviscid time step $\Delta t_{\text{inv}} = \text{CFL}_{\text{inv}} \Delta h / (|\mathbf{v}| + \sqrt{T})$, is smaller than the viscous time step. Here, CFL_{inv} is the corresponding CFL number for the inviscid time step. The upper limit of the artificial coefficient becomes $\mu_{h,m} = C_m \rho \Delta h (|\mathbf{v}| + \sqrt{T})$, where $C_m \propto \text{CFL}_{\text{vis}} / \text{CFL}_{\text{inv}}$ represents another model parameter for the upper bound of μ_h . Note that, κ_h is also bounded by the $\mu_{h,m}$. The complete description of the scheme is reported in (Chaudhuri et al., 2017).

4 Problem setup

The geometry of the cavity flow configuration is taken similar to that presented in the experimental and the invis-

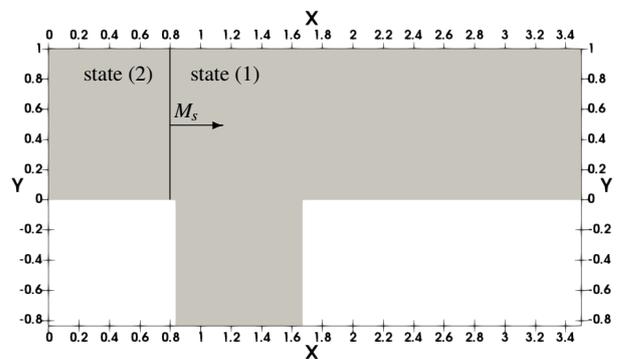


Figure 1. Schematic description of the cavity.

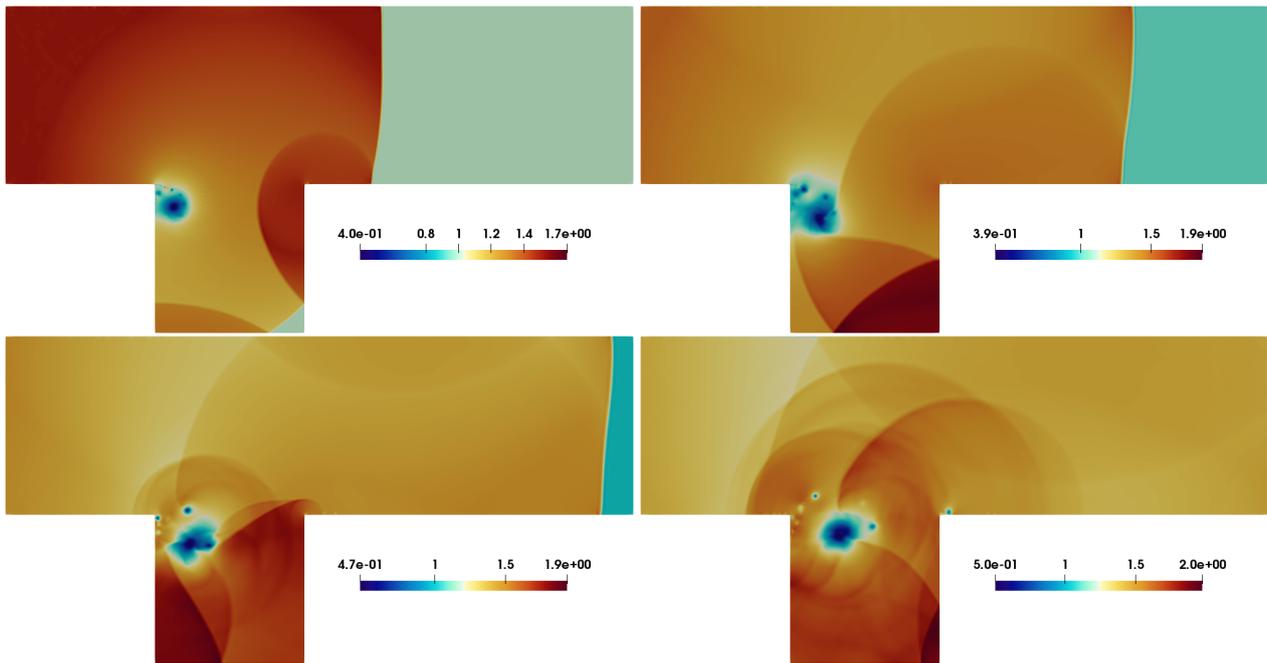


Figure 2. Density contours for $Re_f = 10^6$, top left: $t = 1.0$, top right: $t = 1.5$, bottom left: $t = 2.0$, and bottom right: $t = 2.5$.

cid study of (Igra et al., 1996). Figure 1 shows the setup for the non-dimensional physical domain in the two-dimensional (2D) $x - y$ plane. A shock wave with a Mach number $M_s = 1.3$ is allowed to pass through the cavity geometry. The initial shock location is at $x_s = 0.8$ and the conditions are prescribed using the Rankine-Hugoniot relations for stagnant state (1) and shocked gas state (2). The left and right boundaries are set by the initial states. The other boundaries (top and bottom) are assigned with no-slip wall conditions. The reference state ϕ_f , is taken as the stagnant state conditions ϕ_1 , to satisfy $M_f = 1$ and the reference length is taken as the height of the shock-tube of the experiment (Igra et al., 1996) $L_f = 60$ mm. 50,000 P3 (fourth order) elements with a total of 0.8×10^6 degrees of freedom are considered in the domain. In the present study, the value of the CFL number is taken as 0.9 and the model constants for artificial viscosity are set as: $(C_\mu, C_\kappa, C_m) = (0.5, 0.25, 0.15)$ like our previous studies. A very weak filter is used, filter order of 64 (Chaudhuri et al., 2017) to enhance the stability of the method. The simulations are performed on Supermicro X9DRT compute nodes (dual Intel E5-2670). A typical test case utilizes 12 nodes consuming about 860 CPU hours.

5 Results and Discussion

In subsection 5.1, first a discussion the flow evolution associated with the shock propagation and diffraction through the cavity geometry will be given. The analysis of artificial viscosity based DSEM methodology in light of the artificial dissipation will be given in the following subsection 5.2. Simulations are carried out with varying Re_f , namely 10^6 , 10^5 , and 10^4 . The same mesh in all cases has been used.

5.1 General flow evolution

Figure 2 shows the density contours for $Re_f = 10^6$ at different time instants. The incident shock (IS) is first gets diffracted in the left corner of the cavity. Subsequently, the corner primary vortex and shear layer instability evolves while incident shock propagates further downstream. The primary vortex grows and moves away from the corner of the cavity. The secondary vortices arising due to viscous effects on the no-slip wall, interact with shear layer and influence in the evolution of the unstable shear layer. This secondary vortex interaction is similar in nature to those reported for classical $M_s = 1.5$ flow over 90° diffraction corner, see e.g. (Sun and Takayama, 2003; Skews et al., 2012; Law et al., 2014; Chaudhuri and Jacobs, 2017). The incident shock further gets diffracted in the right corner of the cavity and produces the reflected wave which rebounds back and forth in the cavity section and intensely perturbs the growing primary vortex and the shear layer. Several transverse waves arise from the multiple reflections/diffractions within the cavity and the top wall of the flow configuration. The incident shock leaves the domain at $t \approx 2.1$. Note that, the present viscous simulations capture the viscous effects of no-slip boundary interaction, together with the complex shock dynamics. To resolve these, refined elements are generated using standard stretching functions (Chung, 2010) near the wall regions. Artificial viscosity based DSEM scheme is capable of resolving all the features of the complex flow dynamics.

Figure 3a shows the evolution of the pressure profile along the flow direction at $y = 0.5$. The initial pressure jump associated with $M_s = 1.3$ is $P_2/P_1 = 1.805$. Incident shock propagates with inviscid speed while leaving the domain boundary with a little lower intensity of $P/P_1 \approx 1.68$. The

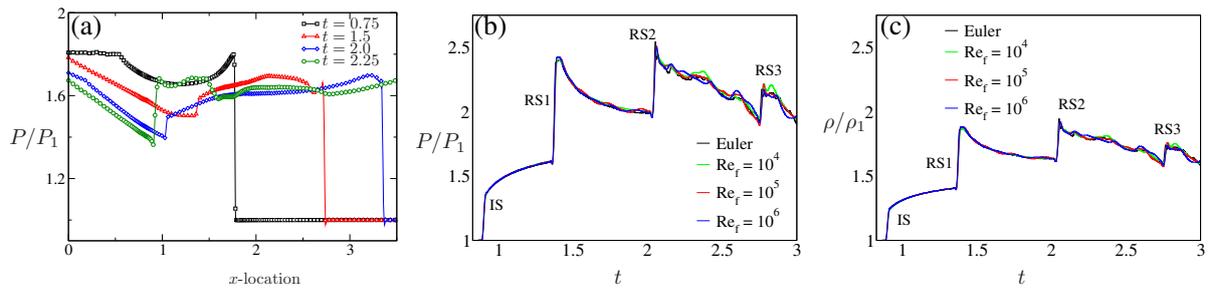


Figure 3. (a): Pressure evolution for $Re_f = 10^6$ at $y = 0.5$, (b) and (c): probe signals for different cases.

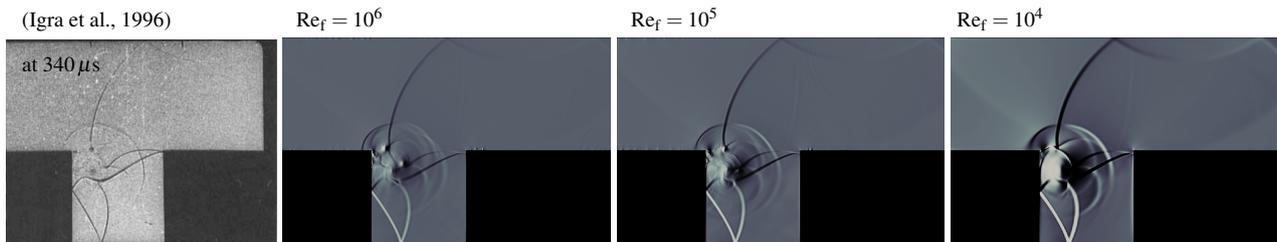


Figure 4. Comparison of experimental Shadowgraph (Igra et al., 1996) with numerical Schlieren at $t = 1.9$.

propagation and diffraction through the cavity geometry exhibit an attenuation effect. Flow quantities are recorded at a probe location situated at the middle of the bottom wall of the cavity. Figures 3b and 3c show the pressure and density evolution at this probe location. The signature of the reflected waves passing through this probe is clearly seen in this figure. The signals show the incident shock and several reflected shocks (RS's). The RS1 from the right wall of the cavity bounces between the cavity walls and produces the subsequent RS's. The pressure peaks agree very well with the experimental and the inviscid numerical work of (Igra et al., 1996; Igra and Igra, 2016). Note that, this shock dynamics remains essentially similar irrespective of the Re_f . This is expected, as the shock dynamics is very fast compare to the slow viscous effects. Only little variations are observed once the incident shock is out of the domain. The overall shock dynamics are found to be in excellent agreement with the experimental (Igra et al., 1996) results (see Figure 4). Based on reference length $L_f = 60\text{mm}$, the Re_f for the experimental setup of (Igra et al., 1996) lies in the range of 10^6 . Figure 4 clearly depicts that the experimental shadowgraph is essentially similar to the numerical Schlieren pictures. The resemblance is most for $Re_f = 10^6$. For lower Re_f viscous effects stabilize the shear layer and the secondary vortex interaction is less dramatic compared to the cases with higher Re_f . Evidently, the shock structures are much thicker for $Re_f = 10^4$.

5.2 Artificial dissipation of DSEM

In the artificial viscosity based DSEM formulation μ_h and κ_h are calculated explicitly (see section 3). These are designed to achieve optimal dissipation in shock dominated regions of the flow-field by the use of a shock sensor and the Heaviside function (see section 3). Figure 5 shows

that the range of values of μ_h (see left column) remains similar in all cases. The contribution of μ_h and κ_h predominantly lie in the regions with shocks/shocklets. The contours are in accordance with the shock structure of the flow-field. For higher Re_f , one can notice shock thicknesses are much less (in accordance with Figure 4). The contours of the local mesh Reynolds number, Re_h (defined as $Re_h = \Delta h |\mathbf{v}| \rho Re_f$) are also shown in Figure 5. Note that, here the mesh remains the same for three cases. These contours look very similar in nature with expected difference in order of magnitude. For $Re_f = 10^6$, the $Re_h \lesssim 2800$, for $Re_f = 10^5$, the $Re_h \lesssim 280$ and for $Re_f = 10^4$, the $Re_h \lesssim 26$. As mentioned in (Chaudhuri et al., 2017; Chaudhuri and Jacobs, 2017), the inverse of the μ_h is a measure of the local effective Reynolds number. In previous studies, it is observed that, for high Re_f the maximum of the added artificial viscosity lies in the order of the inverse of the local Re_h . This is being illustrated in the contours of $\mu_h Re_h$ in Figure 5. For cases with lower Re_f the maximum of the $\mu_h Re_h$ gets reduced. This clearly depicts the fact that the added dissipation is optimal in all cases. The relative contribution of artificial dissipation is less for low Re_f . Note that the range of values of κ_h always remains smaller than μ_h (not shown). The artificial viscosity based DSEM is thus capable of adding a controllable amount of dissipation for capturing shocks and yields stable solutions.

A further investigation of the components of the artificial energy dissipation rate associated with the transport equation of the instantaneous kinetic energy. The energy dissipation rate \mathcal{E} associated with μ_h consists of rotational \mathcal{E}_r , non-homogeneous \mathcal{E}_{nh} and dilatation dissipations \mathcal{E}_d as:

$$\mathcal{E} = \mathcal{E}_r + \mathcal{E}_{nh} + \mathcal{E}_d. \quad (4)$$

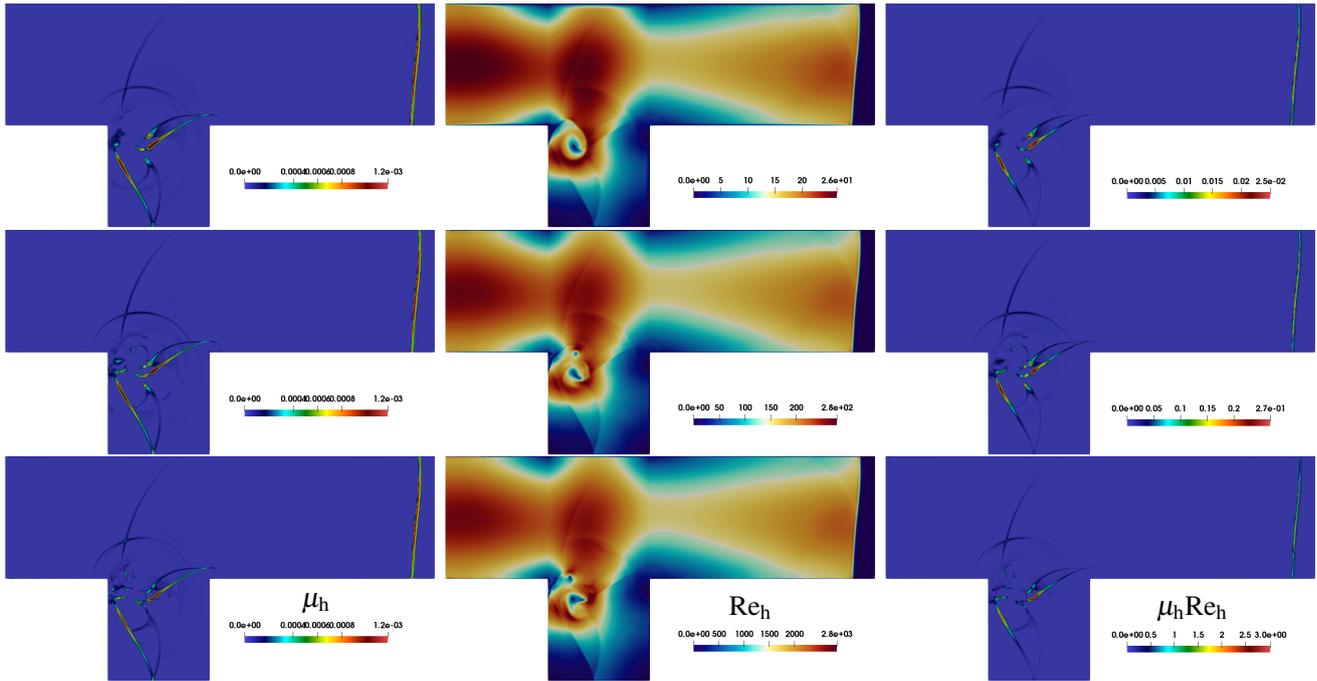


Figure 5. Contours of μ_h , Re_h and $\mu_h Re_h$ at $t = 2$, top row: $Re_f = 10^4$, middle row: $Re_f = 10^5$ and bottom row: $Re_f = 10^6$.

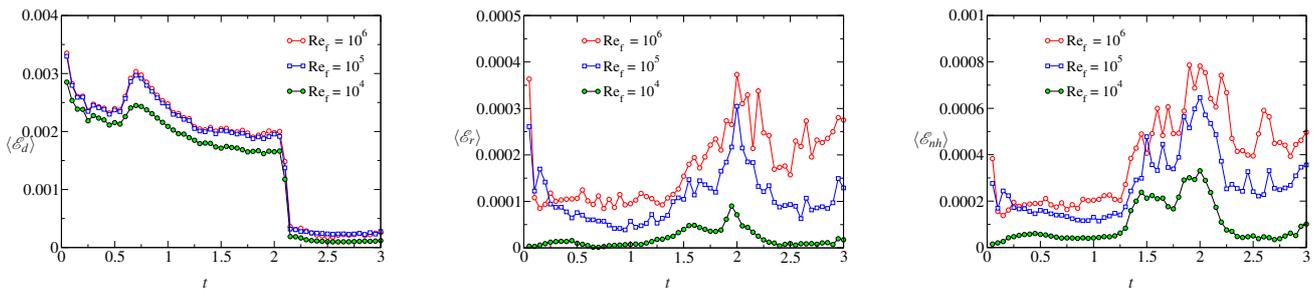


Figure 6. Components of the artificial energy dissipation rate.

where, $\mathcal{E}_r = \mu_h \Omega$, $\mathcal{E}_{nh} = 2\mu_h [(\nabla \mathbf{v})^T \nabla \mathbf{v} - (\nabla \cdot \mathbf{v})^2]$, and $\mathcal{E}_d = \frac{4}{3}\mu_h (\nabla \cdot \mathbf{v})^2$. Here, Ω is the enstrophy. Figure 6 shows the area-weighted (entire domain) dissipation rates as a function of time. For any property ϕ , the area weighting is defined as $\langle \phi \rangle = \frac{\int \phi dA}{\int dA}$. It can be seen that $\langle \mathcal{E} \rangle$ is lower for lower Re_f . Each contributing term follows the same behavior. Clearly, the dominant contributing term for total kinetic energy dissipation rate is \mathcal{E}_d till the incident shock remains in the flow domain i.e., $t \lesssim 2.1$. Evidently, the component terms become comparable for $t > 2.1$ (see Figure 7 for different Re_f). One can notice that the, $\langle \mathcal{E}_{nh} \rangle$ becomes significant in all cases $\approx 30\% - 50\%$. On the other hand, \mathcal{E}_r remains $\lesssim 20\%$ in all cases. Note that, the values of $\langle \mathcal{E} \rangle$ remain typically two-order of magnitude lower than the pressure dilatational term as found in (Chaudhuri and Jacobs, 2017).

6 Conclusions

In this work a study of shock propagation and diffraction of a planar shock ($M_s = 1.3$) travelling through a square cavity is performed. The geometry is taken similar to a

previous literature. An artificial viscosity based DSEM is used for this purpose. The findings can be summarized as:

- The study with 50,000 P3 (fourth order) elements resolved all flow features well. The results are in excellent agreement with experimental results of literature. Accounting the viscous effects, the boundary and shear layer interaction with shock wave is well predicted by the present simulations. The pressure signal at the mid location of the bottom wall of the cavity shows multiple shock reflections on the cavity walls. The signals are similar irrespective of Re_f . The pressure peaks agree very well with (Igra et al., 1996) and the recent work (Igra and Igra, 2016).
- Estimation of artificial viscosity coefficients for various Re_f reveals the consistency of the entropy generation based formulation. The time evolutions of area weighted average quantities of the components of the artificial energy dissipation rate are in accordance of the shocked flow physics.

Dissipations dictated by entropy generation in the pre-

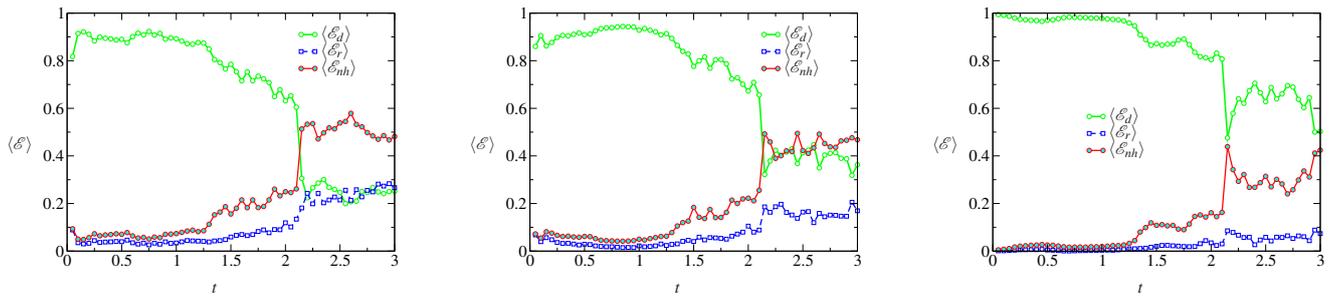


Figure 7. Normalized components of the artificial energy dissipation rate, left: $Re_f = 10^6$, middle: $Re_f = 10^5$, right: $Re_f = 10^4$.

sent artificial viscosity based methodology in a way account the effects of turbulence models. In the future, studies with 3D Large eddy simulations will be performed for further, deeper analysis related to the role of the artificial viscosity coefficients.

Acknowledgement

This work utilized the supercomputing facility of Abel computer cluster (University of Oslo) under the project NN9542K at Notur, UNINETT Sigma2 AS (National infrastructure for scientific computational science in Norway). The author greatly acknowledges the collaboration with G.B. Jacobs of San Diego State University, W. S. Don of Ocean University of China and F. Mashayek of University of Illinois at Chicago.

References

- A Chaudhuri and GB Jacobs. Dynamics of shock wave diffraction over sharp splitter geometry using entropy-based artificial viscosity method. *Shock Waves*, pages 1–15, 2017.
- A Chaudhuri, A Hadjadj, O Sadot, and E Glazer. Computational study of shock-wave interaction with solid obstacles using immersed boundary methods. *International Journal for Numerical Methods in Engineering*, 89(8):975–990, 2012. doi:<https://doi.org/10.1002/nme.3271>.
- A Chaudhuri, A Hadjadj, O Sadot, and G Ben-Dor. Numerical study of shock-wave mitigation through matrices of solid obstacles. *Shock Waves*, 23(1):91–101, 2013. doi:<https://doi.org/10.1007/s00193-012-0362-2>.
- A. Chaudhuri, G.B. Jacobs, W.S. Don, H. Abbassi, and F. Mashayek. Explicit discontinuous spectral element method with entropy generation based artificial viscosity for shocked viscous flows. *Journal of Computational Physics*, 332:99 – 117, 2017. ISSN 0021-9991. doi:<http://dx.doi.org/10.1016/j.jcp.2016.11.042>. URL <http://www.sciencedirect.com/science/article/pii/S0021999116306349>.
- TJ Chung. *Computational fluid dynamics*. Cambridge university press, 2010.
- F Ducros, V Ferrand, Franck Nicoud, C Weber, D Daracq, C Gacherieu, and Thierry Poinsot. Large-eddy simulation of the shock/turbulence interaction. *Journal of Computational Physics*, 152(2):517–549, 1999. doi:<https://doi.org/10.1006/jcph.1999.6238>.
- E Glazer, O Sadot, A Hadjadj, and A Chaudhuri. Velocity scaling of a shock wave reflected off a circular cylinder. *Physical Review E*, 83(6):066317, 2011.
- Dan Igra and Ozer Igra. Planar shock-wave diffraction into a square cavity filled with different gases. *AIAA Journal*, pages 1798–1802, 2016.
- O Igra, J Falcovitz, H Reichenbach, and W Heilig. Experimental and numerical study of the interaction between a planar shock wave and a square cavity. *Journal of Fluid Mechanics*, 313: 105–130, 1996.
- O Igra, L Wang, J Falcovitz, and W Heilig. Shock wave propagation in a branched duct. *Shock Waves*, 8(6):375–381, 1998.
- O Igra, X Wu, J Falcovitz, T Meguro, K Takayama, and W Heilig. Experimental and theoretical study of shock wave propagation through double-bend ducts. *Journal of Fluid Mechanics*, 437:255–282, 2001.
- O Igra, J Falcovitz, L Houas, and G Jourdan. Review of methods to attenuate shock/blast waves. *Progress in Aerospace Sciences*, 58:1–35, 2013.
- Gustaaf B Jacobs, David A Kopriva, and Farzad Mashayek. Validation study of a multidomain spectral code for simulation of turbulent flows. *AIAA Journal*, 43(6):1256–1264, 2005.
- David A Kopriva. A staggered-grid multidomain spectral method for the compressible Navier–Stokes equations. *Journal of Computational Physics*, 143(1):125–158, 1998.
- C Law, AO Muritala, and BW Skews. Unsteady flow with separation behind a shock wave diffracting over curved walls. *Shock Waves*, 24(3):283–294, 2014. doi:<https://doi.org/10.1007/s00193-013-0486-z>.
- MS Shadloo, A Hadjadj, and A Chaudhuri. On the onset of postshock flow instabilities over concave surfaces. *Physics of Fluids (1994-present)*, 26(7):076101, 2014. doi:<http://dx.doi.org/10.1063/1.4890482>.
- Beric Skews, Craig Law, Adam Muritala, and Sebastian Bode. Shear layer behavior resulting from shock wave diffraction. *Experiments in fluids*, 52(2):417–424, 2012. doi:<https://doi.org/10.1007/s00348-011-1233-9>.
- V Soni, A Hadjadj, A Chaudhuri, and G Ben-Dor. Shock-wave reflections over double-concave cylindrical reflectors. *Journal of Fluid Mechanics*, 813:70–84, 2017.

Joseph H. Spurk and Nuri Aksel. *Fluid Mechanics*. Springer-Verlag Berlin Heidelberg, 2008. ISBN 978-3-540-73537-3. doi:<https://doi.org/10.1007/978-3-540-73537-3>.

M Sun and K Takayama. A note on numerical simulation of vortical structures in shock diffraction. *Shock Waves*, 13(1): 25–32, 2003. doi:<https://doi.org/10.1007/s00193-003-0195-0>.

Comparison of different state estimator algorithms applied to a simulated anaerobic digestion reactor

Shadi Attar¹ Finn Aakre Haugen²

^{1,2} Faculty of Technology, Natural Sciences and Maritime Sciences, University College of Southeast Norway, Norway
shadi.attar@usn.no, Finn.Haugen@usn.no

Abstract

This study deals with a simulator-based comparison of different state estimators of an anaerobic digestion process. A simulated biogas reactor based on the AM2 model is considered. Extended Kalman Filter, Unscented Kalman Filter, Particle Filter and Moving Horizon Estimator are four state estimators studied. The investigation is on both states and parameters estimation. The maximum number of parameters can be estimated equals the number of the measurement.

Keywords: Anaerobic digestion, state estimation, Kalman filter, particle filter, moving horizon estimator, simulation

1 Introduction

Anaerobic digestion (AD) is a biological process to produce biogas consisting of methane and carbon dioxide in the absence of oxygen. There are several biogas models to represent the behaviour of the AD reactor. In this study, the Anaerobic Digestion Model number 2 (AM2) (Bernard, et al., 2001; Dochain, 2008) with 6 states and 13 parameters is applied. The model is implemented in MATLAB.

The main goal of this study is states and parameters estimation for a simulated AD reactor based on the AM2 model. Estimation of states and parameters is useful for monitoring and control purposes. Four states estimation methods consisting of Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF), Particle Filter (PF) and Moving Horizon Estimator (MHE) are considered.

State estimation for large-scale wastewater treatment plants is studied in (Busch, et al., 2013). UKF state estimation is surveyed for a biogas reactor (Haugen, et al., 2014). A state estimator is developed using a calibrated simulation model of a full-scale biogas plant, which is based on the Anaerobic Digestion Model N0.1. (Gaida, et al., 2012). A number of software sensor design methods, including extended Kalman filters, receding-horizon observers is studied in (Bogaerts & Vande Wouwer, 2003).

The outline of the article is as follows Section 2 includes the process description and the AM2 model

equations and observability analysis method. Section 3 explains about state estimation consists of four methods and comparison their results. Parameter estimation are described in Section 4 regarding different number of measurement for estimating both environmental parameters and model parameters. Section 5 presents and discusses about the simulation results. Conclusions are given in Section 6, and future work is suggested in Section 7.

2 Materials and methods

2.1 Process description

The biogas reactor model applied in this article is the AM2 model with parameter values as presented in (Bernard, et al., 2001). The reactor has the form of a cylindrical tank with effective volume of 948 litres. The temperature of reactor will be constant due to a temperature control system.

2.2 Mathematical model

The mathematical model which is the basis of this simulator, is generated from mainly mass balance and bacterial kinetics equations. Moreover, ionic balance, affinity constant, bacterial kinetics, ideal gas law and Henry's law are considered in the model (Dochain, 2008).

2.2.1 Material balances

In Anaerobic digestion process, organic material is converted by microorganisms in two phases are called acidogenesis and methanization. In the first phase, the acidogenic bacteria (X_1) consume the organic substrate (S_1) and produce CO_2 and volatile fatty acids (S_2). In the second phase, the population of methanogenic bacteria (X_2) uses for produce the methane. Z and C are the total alkalinity and total inorganic carbon, respectively.

Mass balances gives the following differential equations, which constitute a state space model of the AD reactor:

$$\frac{dX_1}{dt} = [\mu_1 - \alpha D]X_1 \quad (1)$$

$$\frac{dX_2}{dt} = [\mu_2 - \alpha D]X_2 \quad (2)$$

$$\frac{dZ}{dt} = D(Z_{in} - Z) \quad (3)$$

$$\frac{dS_1}{dt} = D(S_{1in} - S_1) - k_1\mu_1X_1 \quad (4)$$

$$\frac{dS_2}{dt} = D(S_{2in} - S_2) + k_2\mu_1X_1 - k_3\mu_2X_2 \quad (5)$$

$$\frac{dC}{dt} = D(C_{in} - C) - q_C + k_4\mu_1X_1 + k_5\mu_2X_2 \quad (6)$$

D is the dilution rate defined as:

$$D = \frac{F_{feed}}{V} \quad (7)$$

where F_{feed} is the infeed flowrate and V is the effective volume of medium in the reactor. D represents a normalized flow. $Z_{in}, S_{1in}, S_{2in}, C_{in}$ in (3) - (6) are respectively concentration of the inflow of alkalinity, substrate, VFA and dissolved inorganic carbon.

q_C represents the flow of inorganic carbon from liquid phase to gas phase is calculated accordance to (8) - (10):

$$q_C = k_L a (C + S_2 - Z - K_H P_C) \quad (8)$$

where $k_L a$ and K_H are liquid-gas transfer constant and Henry's constant, respectively. P_C is partial carbon dioxide pressure and it can be calculated as:

$$P_C = \frac{\phi - \sqrt{\phi^2 - 4K_H P_T (C + S_2 - Z)}}{2K_H} \quad (9)$$

with

$$\phi = C + S_2 - Z + 2K_H P_T + \frac{k_6}{k_L a} \mu_2 X_2 \quad (10)$$

Methane flow is directly related to the methanogenic rate (μ_2):

$$q_m = k_6 \mu_2 X_2 \quad (11)$$

The model equation for the pH is:

$$pH = -\log\left(K_b \frac{C + S_2 - Z}{Z - S_2}\right) \quad (12)$$

where K_b is an affinity constant.

2.2.2 Bacterial kinetics

The models to describe the growth of microorganism are assumed Monod-type and Haldane-type (Dochain, 2008). The growth of acidogenic bacteria, μ_1 , is considered based on Monod type kinetics:

$$\mu_1 = \mu_{1max} \frac{S_1}{S_1 + K_{S_1}} \quad (13)$$

The growth of methanogenic bacteria, μ_2 , is assumed based on Haldane kinetics:

$$\mu_2 = \mu_{2max} \frac{S_2}{S_2 + K_{S_2} + \frac{S_2^2}{K_{I_2}}} \quad (14)$$

where μ_{1max} and μ_{2max} are respectively the maximum growth rate of acidogenic biomass and methanogenic biomass.

2.3 Observability

The standard method to check the observability of a linear model is used in this study. At first, the AM2 model is linearized and the model converted to (15):

$$\begin{aligned} \frac{dx}{dt} &= Ax + BD \\ y &= Cx + ED \end{aligned} \quad (15)$$

The observability matrix for (15) is computed according to (16)

$$OB = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} \quad (16)$$

n is the number of elements in the state vector. If the rank of the observability matrix is less than n , the system is non-observable, i.e. there are state variables that are non-observable.

3 State estimation

3.1 Introduction

In the practical systems, we are not able to measure some of variables which have effects on control and monitoring purpose. To have reliable monitoring and control, we need to estimate these variables as far as possible. A suitable state estimator is essential for this aim. As shown in Figure 1, the state estimation is used in the control system. However, the controller is not described in the present article.

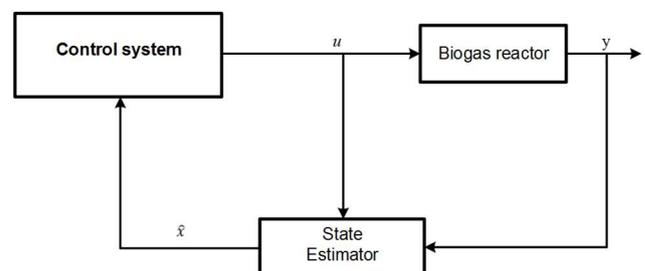


Figure 1. A state estimator scheme.

The state vector of the AM2 model is:

$$x = [X_1, X_2, Z, S_1, S_2, C] \quad (17)$$

It is decided to estimate some of parameters such as S_{1in}, S_{2in} as the environmental variables and k_6 as the model parameter in the parameter estimation section. Four states estimator methods consisting of EKF, UKF, PF (Simon, 2006) and MHE (Boegli, 2014) are implemented based on the AM2 model.

The dilution rate, D is as an input variable, and q_m is considered as process measurement.

$$y = q_m \quad (18)$$

Where y is measurement vector.

The observability of the state vector is tested regarding the standard method of observability analysis and only four state variables are observable.

Initial values of the states for the simulated model are based on a steady state condition and shown in Table 1. Initial value of all state estimators are determined with considering a large initial estimation errors in X_1, X_2, Z to check the convergence rate of the methods. See Table 1.

We need to determine P, Q and R which are state estimation error covariance, process noise covariance and measurement noise covariance, respectively. They are computed as:

$$\begin{aligned} P &= \text{diag}(K_P x_{init}^2) \\ Q &= \text{diag}(K_Q x_{init}^2) \\ R &= \text{diag}(K_y y_{init}^2) \end{aligned} \quad (19)$$

where x_{init} is assumed as an initial values of augmented state vector, y_{init} is an initial values of measurement vector. K_P, K_Q and K_y are coefficients equal 0.1, 0.01 and 1 respectively, select by trial and error.

Table 1. The initial values for the simulated model and the state estimators

Initial values (x_{init})	Simulated model	State estimators	Unit
X_{1init}	0.39	1.39	g/L
X_{2init}	0.85	1.85	g/L
Z_{init}	62.2	72.2	mmol/L
S_{1init}	1.17	3.17	g/L
S_{2init}	2.78	2.78	mmol/L
C_{init}	67.9	70.9	mmol/L

The amount of biogas depends on the dilution rate. During the simulation, dilution rate would be changed to check its effect and it is assumed that the dilution rate is always known.

The model parameters are known and considered based on (Bernard, et al., 2001).

Noise is added to the simulated measurements.

3.2 Extended Kalman Filter

This method is based on linearizing the non-linear system around a nominal state set. The linear model is only used for calculation of Kalman gain.

The procedure of implementation of EKF method in the simulation follows the principle of the discrete-time extended Kalman filter presents in (Simon, 2006). The outcomes of this method for 15 days is shown in Figure 2 with the magenta curves.

3.3 Unscented Kalman Filter

The EKF is based on linearization to propagate the mean and covariance of states. UKF is an enhancement of the Kalman filter to reduce the linearization error of the EKF.

The procedure of implementation of UKF method in the simulation follows the principle of the discrete-time unscented Kalman filter present in (Simon, 2006). Figure 2 presents the results of UKF state estimator for all six state variable in AM2 model with the black curves.

3.4 Particle Filter

This method is based on statistical approach. The particle filter is a probability-based estimator as an enhancement on Bayesian estimator (Simon, 2006). The number of particles is assumed to be 100. The results of particle filter in the same condition on UKF and EKF are shown in Figure 2 with the green curves.

3.5 Moving Horizon Estimator

This method is an optimization problem over the specific horizon, N , which is shifted in terms of time.

The procedure of implementation of MHE method in the simulation follows the principles in (Boegli, 2014). The horizon, N , in this simulation equals to 10.

The objective function to minimize is:

$$\begin{aligned} \min_x \quad & \sum_{k=t-N}^t \|x_{k+1} - f(x_k, D_k)\|_Q^2 \\ & + \sum_{k=t-N}^t \|y_k - h(x_k, D_k)\|_R^2 \\ & + \|x_{t-N} - \hat{x}_{t-N}\|_P^2 \end{aligned} \quad (20)$$

subject to: $C - S_2 - Z > 0$

y_k is the process measurements.

Since $C - S_2 - Z$ equals to CO_2 , this term should be positive. The first to third sums in (20) represents dynamical system mismatch, measurement mismatch and arrival cost, respectively.

The arrival cost term represents the error of the state estimate at the “end” of the horizon. In our implementation, the state at $k = t - N$ is estimated, thereby implicitly minimizing the arrival cost term. Therefore, we have omitted the arrival cost term as an explicit term.

The results of MHE are shown in Figure 2 with the red curves.

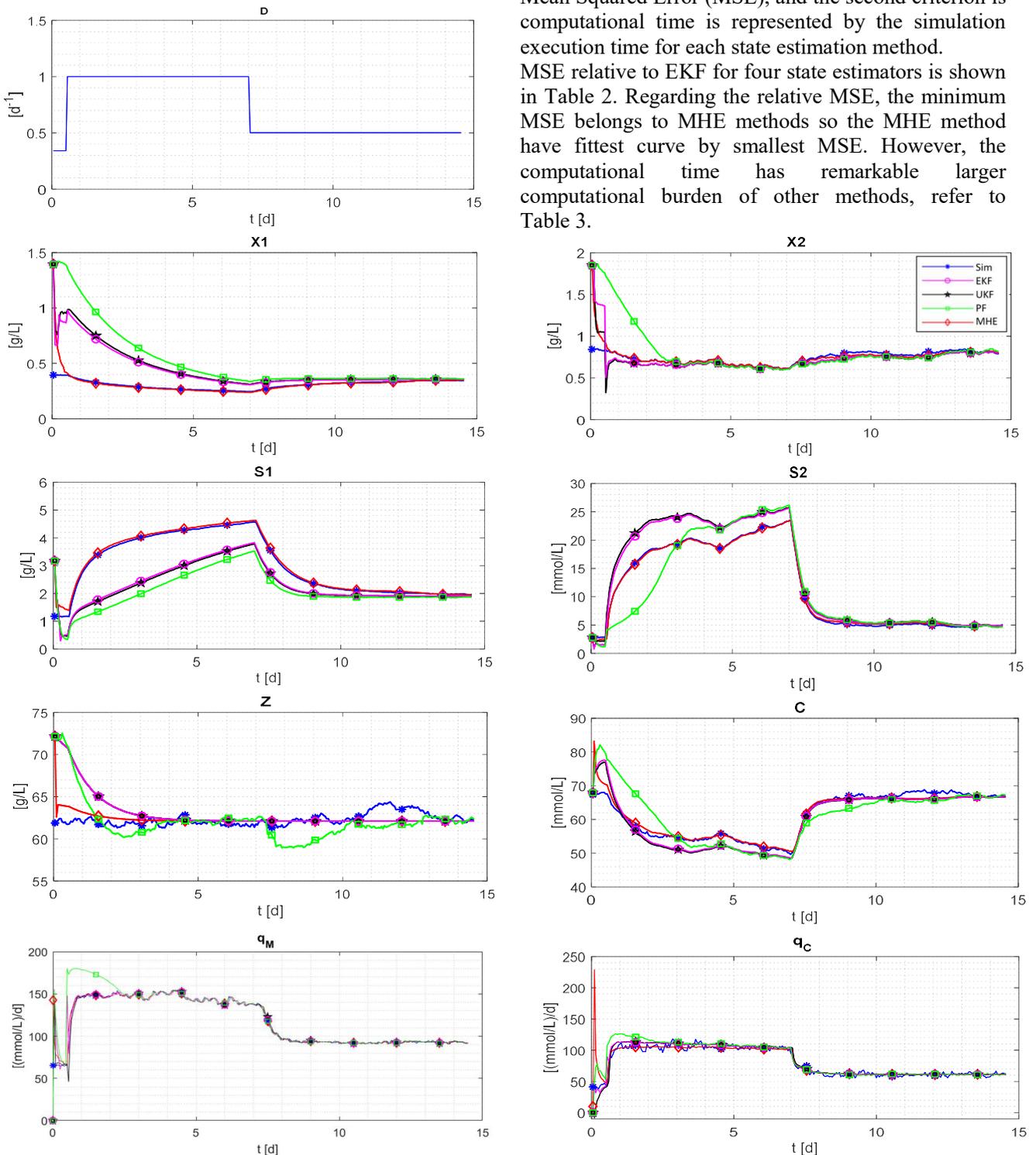


Figure 2. The results of four state estimation strategies when augmented vector consists of all six state variables and the methane gas flow rate is the only measurement.

3.6 Comparison of the various state estimators

All four state estimators are executed on the same conditions and with the same initial values. It is assumed that dilution rate, D , changes after one day from 0.34 to $1 d^{-1}$ and after 7.5 days, D is changed to $0.5 d^{-1}$.

Two criteria are supposed for comparison among of the state estimation methods. The first criterion is the Mean Squared Error (MSE), and the second criterion is computational time is represented by the simulation execution time for each state estimation method.

MSE relative to EKF for four state estimators is shown in Table 2. Regarding the relative MSE, the minimum MSE belongs to MHE methods so the MHE method have fittest curve by smallest MSE. However, the computational time has remarkable larger computational burden of other methods, refer to Table 3.

The convergence rate for the MHE strategy is fastest comparing with other methods in all estimated states. EKF has faster computation but the Jacobians need to be calculated.

Table 2. MSE relative to EKF for four state estimation methods regarding one measurement and six state variables

Relative MSE	EKF	UKF	PF	MHE
X_1	1	1.48	4.10	0.16
X_2	1	0.5	3.82	0.27
Z	1	1	1.02	0.54
S_1	1	1.28	2.06	0.03
S_2	1	1.35	1.60	0.02
C	1	1	2.25	0.44
q_m	1	0.55	3.08	0.48
q_c	1	1.36	2.83	0.83

Table 3. The relative computational burden to EKF measured with simulation time.

	EKF	UKF	PF	MHE
Computational time	1	13	62	14892

4 Parameters estimation

Some parameters are not known nor measured while they are essential parameters for control and monitoring. Such parameters should therefore be estimated.

4.1 One measurement and one estimated parameter

The organic substrate concentration of the inflow, S_{1in} , is considered as a first parameter for estimation. The augmented state vector is:

$$x = [X_1, X_2, Z, S_1, S_2, C, S_{1in}] \quad (21)$$

The parameter state variables are here denoted augmentation state variables.

Initial values of S_{1in} for the simulated model and state estimators are 9.5 and 14.5 g/L, respectively. The methane gas flow rate is assumed as measurement, so measurement vector is (18). All conditions are the same as mentioned in Section 3. When one measurement is considered, two states are non-observable.

Table 4 shows the relative MSE to the EKF for all four methods. Based on the relative MSE, the MHE method is not reliable approach to estimate the parameters.

Since the error is remarkably large with MHE, this method is ignored to be able to compare other estimators. We focus on the result for S_1 and S_2 state

variables and the assumed parameter variable in Figure 3. However, there are two non-observable states based on the rank test, EKF, UKF and PF methods can estimate the states. It is shown that standard observability analysis are not effective for non-linear systems.

Table 4. MSE relative to EKF for four state estimation methods for one measurement and six state variables plus S_{1in} as the estimated parameter

Relative MSE	EKF	UKF	PF	MHE
X_1	1	9.82	9.54	4
X_2	1	2.43	9.73	0.87
Z	1	1	1.02	7.48
S_1	1	2.06	2.58	718.758
S_2	1	0.67	4.44	31.84
C	1	1	1.14	7.55
S_{1in}	1	0.67	1.48	2.34
q_m	1	0.867	1.829	8.280
q_c	1	1.099	1.99	58.13

Table 4 shows the EKF and the UKF have better results for the parameters estimation. In the following sections, we focus on just the EKF and UKF.

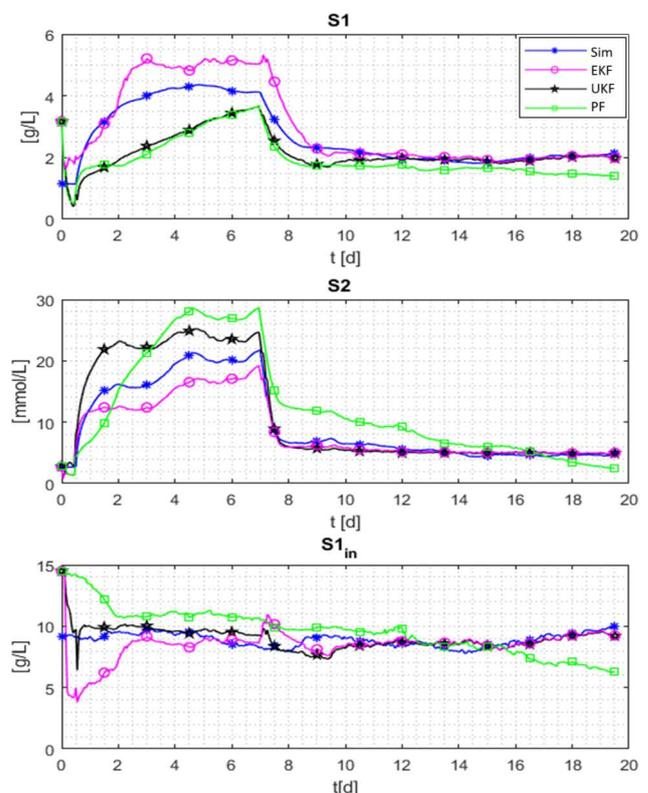


Figure 3. The results of the state estimators consisting of EKF, UKF and PF for an augmented state vector with 7 elements and one measurement.

4.2 One measurement and two estimated parameters

The VFA concentration of the inflow, S_{2in} , is considered as second parameter for estimation. So the augmented state vector is:

$$x = [X_1, X_2, Z, S_1, S_2, C, S_{1in}, S_{2in}] \quad (22)$$

All conditions are the same as mentioned in Section 4.1 plus the initial values of S_{2in} for the simulated model and state estimators are 93.5 and 133.5 mmol/L, respectively. Three states are non-observable when there is just one measurement.

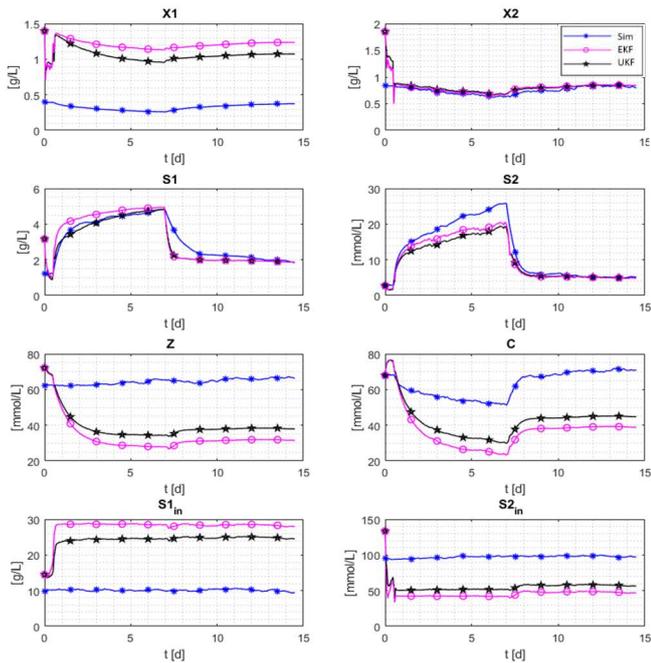


Figure 4. The results of the state estimators consisting of EKF, UKF for an augmented state vector with 8 elements and one measurement.

The results states that it is not possible to estimate two parameters with one measurement. This lack the measurement ruins also the state estimation, refer to Figure 4. The states X_1 , Z , C are not possible to estimate.

In the Section 4.5, it will be checked the possibility of the estimation of two parameters with two measurement.

4.3 Two measurements and one estimated parameter

It is of interest to investigate the case when the number of measurement is greater than the number of estimated parameters. Carbon dioxide gas flow rate is considered as the second process measurement. so the measurement vector is :

$$y = [q_m, q_c] \quad (23)$$

One state is non-observable based on the rank test. The results are shown in Figure 5.

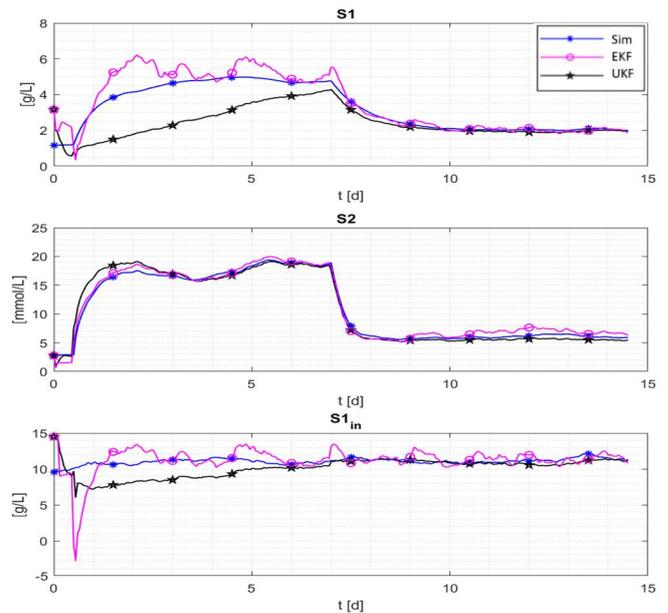


Figure 5. The results of the EKF, UKF state estimators for an augmented state vector with 7 elements and one measurement.

4.4 Three measurements and one estimated parameter

All conditions are the same as the mentioned condition in Section 4.3 except considering pH of the reactor as third measurement. The measurement vector is:

$$y = [q_m, q_c, pH] \quad (24)$$

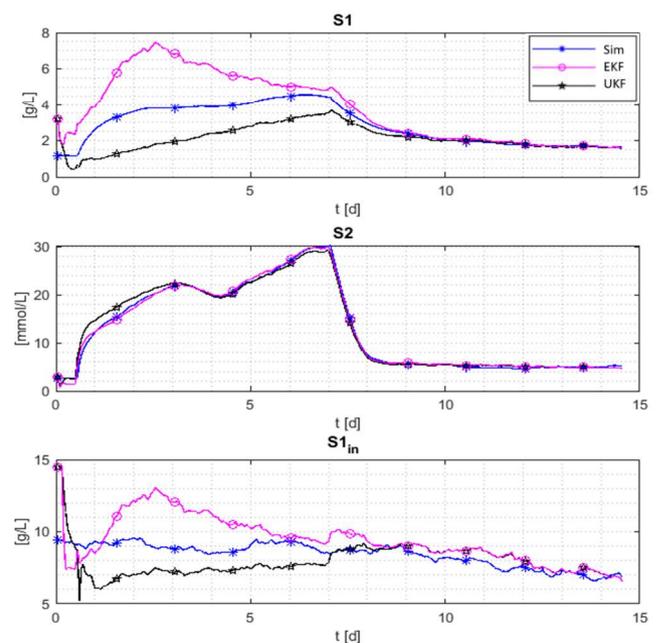


Figure 6. The results of the EKF, UKF state estimators for an augmented state vector with 8 elements and three measurement.

All state become observable. The results are shown in Figure 6 for the EKF and the UKF.

Regarding the results in Section 4.3 and 4.4, we can estimate parameters if the number of measurements is larger than the number of the estimated parameters.

4.5 Two measurements and two estimated parameter

It is supposed the augmented state vector and the measurement vector are according to equation (22) and (23), respectively. There is one non-observable state based on the rank test of the observability matrix.

The results of the simulation are shown in Figure 7. Regarding to the results, two measurements are required at least to estimate two parameters.

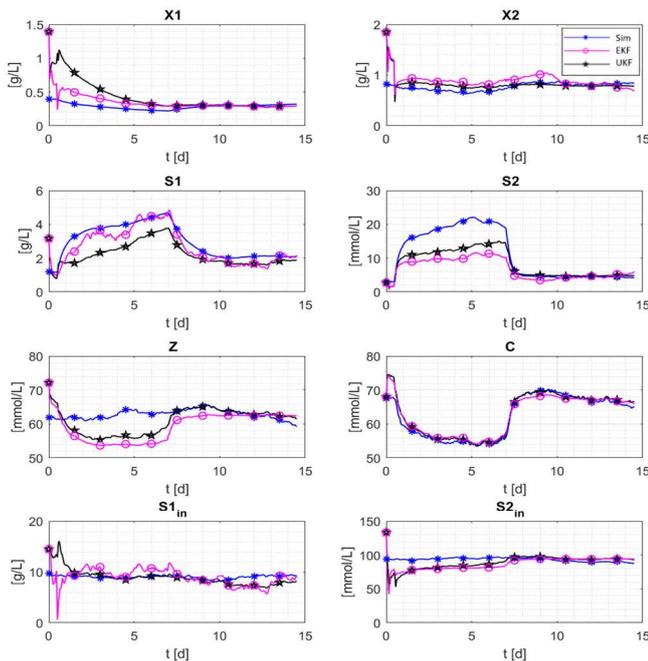


Figure 7. The results of the EKF, UKF state estimators for an augmented state vector with 8 elements and two measurement.

4.6 Two measurements and three estimated parameter

Until now, the environmental parameters are estimated, namely inflow concentrations S_{1in} and S_{2in} . It is assumed that third estimated parameter is selected among of model parameters. One of the most important model parameters which has a direct effect on the methane production is k_6 . The augmented state vector is

$$x = [X_1, X_2, Z, S_1, S_2, C, S_{1in}, S_{2in}, k_6] \quad (25)$$

The initial values of k_6 for the simulated model and state estimators are 435 and 300 mmol/L. the measurement vector is assumed as (23). All states are observable. The results are shown in Figure 8. Since the number of measurements is less than the number of estimated parameters, it is not possible to estimate the parameters.

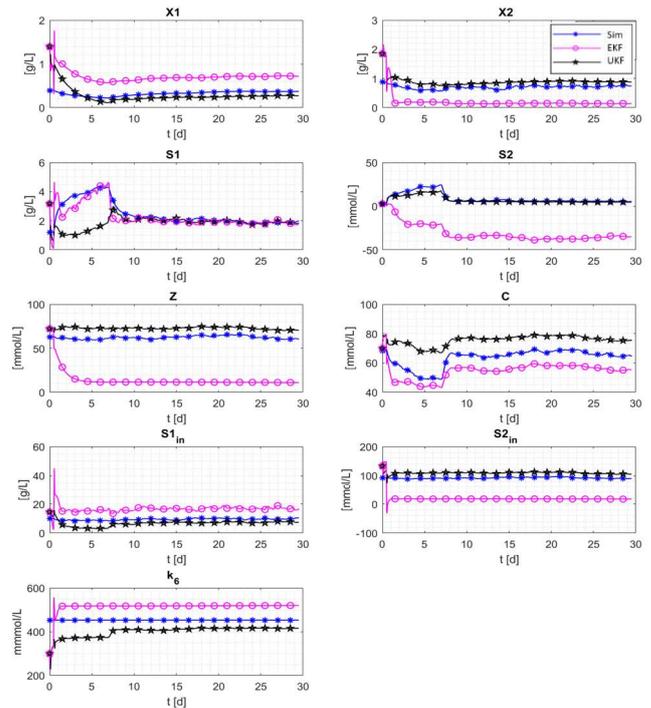


Figure 8. The results of the EKF, UKF state estimators for an augmented state vector with 9 elements and two measurement.

4.7 Three measurement and three estimated parameter

The augmented state vector and the measurement vector are according to equation (25) and (24), respectively.

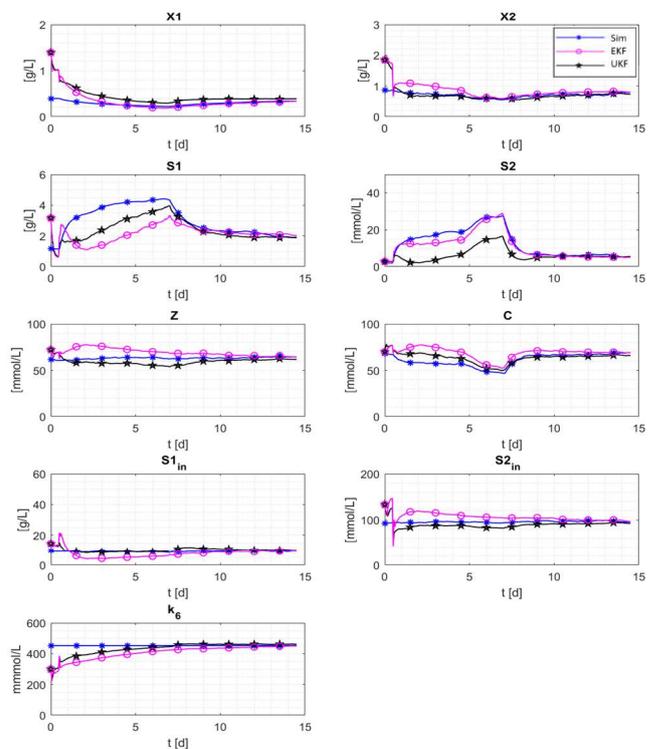


Figure 9. The results of the EKF, UKF state estimators for an augmented state vector with 9 elements and three measurements.

All state are observable based on the rank test on observability matrix.

The results shows in Figure 9 and presents that three parameters can estimate with three measurements.

5 Discussions

The results shows that all four methods can estimate all states of AM2 model for a biogas reactor.

In this case, the MHE is the best state estimation method to estimate the all six states, but computational times is excessive comparing with other methods. However, the MHE is not suitable for estimating the parameters, we assume this is because the optimization problem is not well conditioned. We have calculated the Hessian of the objective function during the estimation. The Hessian is positive definite, but with a large ratio of its eigenvalues, and hence the MHE has found a minimum. However, this may well be a local minimum.

Both the EKF and the UKF give a good estimation with lower implementation effort. Both methods are good for state and parameter estimation. For the EKF, there is an extra computation demand related to calculation of the Jacobians needed for calculation of the Kalman gain. While the UKF straightforward approach and Jacobian free algorithm.

In this study, PF can estimated both states and parameters, but the accuracy is lower comparing to EKF and UKF.

In EKF and UKF, the maximum number of parameter can be estimated is related to the number of the measurements. The maximum number of parameters can be estimated equals the number of the measurement. See Table 5.

Table 5. Assessment of the state estimation based on the number of measurements and estimated parameters

Meas#	1	1	1	2	2	3	2	3
Orig. states#	6	6	6	6	6	6	6	6
Aug. states #	0	1	2	1	2	1	3	3
Non-ob. state#	2	2	3	1	1	0	0	0
Assessment	P	P	F	P	P	P	F	P

*P= Pass, F=Fail

6 Conclusions

In this study, the four model-based state estimation methods regarding the AM2 model are investigated. These methods are EKF, UKF, PF and MHE which are evaluated in the simulation case study.

Further parameter estimations are verified regarding to different number of measurement and different number of parameter to estimate.

7 Future work

Plans for future work are: Observability analysis for nonlinear models; Analysis of robustness of the estimators; Application to real AD reactor data.

Acknowledgements

Thanks to USN for financial supporting.

Nomenclature

C	total inorganic carbon concentration (mmol/L)
D	dilution rate (d ⁻¹)
k ₁	yield for substrate degradation
k ₂ , k ₃	VFA yields (mmol/L)
k ₄ , k ₅	CO ₂ yields (mmol/L)
k ₆	yield for CH ₄ production (mmol/L)
K _b	Affinity constant (mol/L)
K _H	Henry's constant (mmol/L per atm)
k _{1,a}	liquid-gas transfer constant (d ⁻¹)
K _{I2}	inhibition constant (mmol/L)
K _{S1}	half-saturation constant (g/L)
K _{S2}	half-saturation constant (mmol/L)
P _T	total pressure (atm)
q _C , q _m	CO ₂ and CH ₄ flow rates (mmol/L per d)
S ₁	organic substrate concentration (g/L)
S ₂	volatile fatty acids concentration (mmol/L)
X ₁ , X ₂	concentration of acidogens and methanogs (g/L)
V	effective volume of medium in the reactor (L)
Z	total alkalinity (mmol/L)
α	fraction of bacteria in the liquid phase
μ _{max}	maximum specific growth rate (d ⁻¹)

References

- Bernard, O. et al., 2001. Dynamical model development and parameter identification for an anaerobic wastewater treatment process. *Biotechnology and bioengineering*, 75(4), pp. 424-438.
- Boegli, M., 2014. *Real-Time Moving Horizon Estimation, Application to Friction State and Parameter Estimation*, Leuven: s.n.
- Bogaerts, P. & Vande Wouwer, A., 2003. Software sensors for bioprocesses. *ISA Transaction*, Volume 42, pp. 547-558.
- Busch, J., Elixmann, D., Kuhl, P. & Gerken, C., 2013. State estimation for large-scale wastewater. *Water research*, Volume 47, pp. 4774-4787.
- Dochain, D., 2008. *Bioprocess Control*. London: Wiley.
- Gaida, D., Wolf, C., Meyer, C. & Stuhlsatz, A., 2012. State estimation for anaerobic digesters using the ADM1. *water Science & Technology*, Volume 66, pp. 1088-1095.
- Haugen, F. A., Bakke, R. & Lie, B., 2014. State estimation and model-based control of pilot Anaerobic digestion reactor. *Journal of Control Science and Engineering*.
- Simon, D., 2006. *Optimal state estimation*. Hoboken, New Jersey: Wiley Interscience.

FVM-Modeling of Continuity-Coupled Electrical Charge Submitted to Incompressible Flow with OpenFOAM®

D. Rubinetti¹ D. A. Weiss¹

¹Institute of Thermal- and Fluid-Engineering , University of Applied Sciences and Arts Northwestern Switzerland

Abstract

This study conceives a FVM-based model for ionization processes coupling the incompressible Navier-Stokes equations to Maxwell's equations adjusted for electrostatics. Modeling instructions for the opensource tool OpenFOAM™ are presented within the scope of defining a customized solver which calculates the distortion of electrical current by an external fluid flow.

By means of a simplified test-case the negligible impact of external convection on the distribution of space charge for many engineering applications is confirmed while pointing out the limitations of negligible convection. The study shows that peak velocities beyond 100 ms^{-1} within the flow field might have an impact on the current distribution, thus requiring careful evaluation of the modeling assumptions.

The approach described in the modeling guide proves to be numerically stable. The results for the electric field distribution have been analytically verified.

Keywords: Corona Discharge, Fluid Flow, Electrodynamics, Space Charge Density, Convection

1 Introduction

Corona discharge describes the effect where a non-conducting gaseous carrier such as air is ionized in proximity of an electrode with strong electrical field. On molecular level the air molecule receives electrons from the electrode (negative polarity) or loses electrons to the electrode (positive polarity). In either case the ionized molecule is accelerated away from the electrode towards the grounded electrode. The sum of all ionized molecules traveling approximately on electric field lines are described as a continuum and as such it is subject to macroscopic transport phenomena. Ionization processes are widely used *i.e.* to reduce the environmental impact of combustion-based pollution. To assist the design of such applications numerical models prove to be a cost-effective and powerful tool. Among numerous considerations for the numerical modeling of Corona discharge and ionization processes as such, the assumption of negligible external convection is prevalent. This study tackles the widespread convention of excluding convection by means of a geometrically simplified test-case run by a customized solver which includes fluid dynamics and electrostatics in a coupled way.

NOMENCLATURE

b	ion mobility	$\text{m}^2 \text{V}^{-1} \text{s}^{-1}$
E_0	Corona onset field strength	V m^{-1}
\hat{E}	dimensionless electric field	-
f_r	fatigue factor	-
g	gravitational acceleration	m s^{-2}
I	electric current per unit length	A m^{-1}
\hat{I}	dimensionless electric current	-
j	current density	A m^{-2}
m_s	relative gas density	-
r	radius	m
r_1	charging electrode radius	m
r_2	ground electrode radius	m
p	pressure	Pa
p_0	standard pressure	Pa
T_0	standard temperature	K
u	fluid velocity field	m s^{-1}
u_r	radial fluid velocity	m s^{-1}
V	volumetric flow per unit length	$\text{m}^2 \text{s}^{-1}$
\hat{V}	dimensionless volumetric flow	-
δ	boundary cell layer height	m
ϵ_0	vacuum permittivity	F m^{-1}
ϵ_r	relative permittivity	-
ν	kinematic viscosity	$\text{m}^2 \text{s}^{-1}$
ϕ	electric potential	kV
ρ_{el}	space charge density	C m^{-3}

2 Physical Model

The physical nature of electrical charge in a gaseous environment is best explained by electrostatic precipitation - a widely used application area for the control of airborne particle emissions. In practice, a high-voltage electrode (typically 10-50 kV) is immersed in exhaust gases from *i.e.* cement plants. By ionizing the surrounding air, soot particles are being electrically charged and accelerated out of the main exhaust gas flow. Thus, cleaning the gas from toxic components. Figure 1 illustrates the multiphysical coupling of such systems.

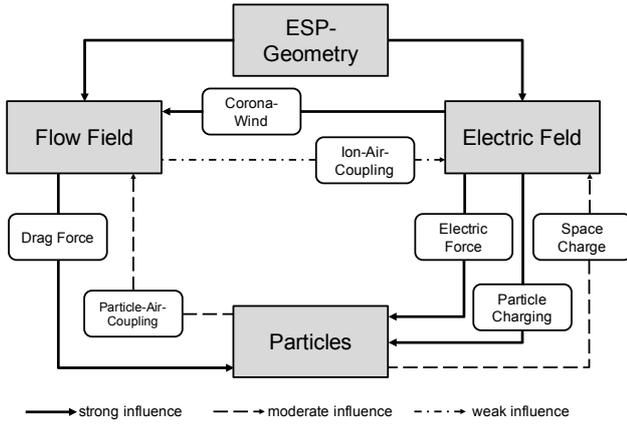


Figure 1. Multiphysics coupling of electrostatic precipitators (Rubinetti et al., 2017)

In this study the interdependence of flow field and electric field is studied. The production and acceleration of air molecules leads to a strong influence of the electric field onto the flow field via Corona wind. Hereby, an initially stagnating fluid ($u = 0$ m/s) undergoes an acceleration itself. The reverse effect, when the flow field distorts the propagation of ions, is often neglected due to its assumed weak influence (Rubinetti et al., 2015). As a characteristic scale on molecular level it can be stated that ions move with a velocity of ~ 1000 m s⁻¹ on electric field lines while the surrounding fluid in most applications has a significantly lower characteristic velocity.

The study is carried out with two coaxially arranged cylinders of $r_1 = 0.002$ m and $r_2 = 0.2$ m as shown in figure 2. Hereby the inner cylinder represents the wire as the charging electrode and the outer cylinder is the grounded electrode. The coaxial geometry is reduced to a quarter of a cylinder slice with one mesh cell in axial direction, thus scaling the problem from 3D to 2D suppressing alterations in axial direction.

Table 1 lists the assumptions and definitions used in the study. In this theoretical case the charging electrode $\partial\Omega_1$ is assumed to carry a given space charge density for the electrostatic part and simultaneously act as inlet for the fluid flow part with a velocity of $w = 100$ m s⁻¹ radially distributed.

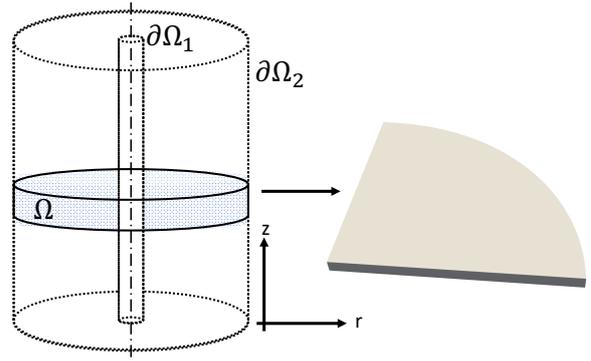


Figure 2. Simplified coaxial geometry

The Corona onset field strength in [V m⁻¹] is given by the empirical correlation (White, 1963)

$$E_0 = 3 \times 10^6 f_r \left(m_s + 0.03 \sqrt{\frac{m_s}{r_1/1[\text{m}]}} \right) \quad (1)$$

with the fatigue factor f_r being a measure for the usage of the electrode ($f_r = 0.6$ for practical use, $f_r = 1$ for a new electrode). The relative gas density takes into account the surrounding fluid, which, at standard conditions (T_0, p_0) has the value of $m_s = 1$.

3 Numerical Model

The numerical model considers the incompressible Navier-Stokes equations and Maxwells equations for electrostatic applications without magnetic influence. A modelling approach for the definition of a customized solver for OpenFOAM® is given.

3.1 Governing equations

For the fluid flow part the continuity equation yields

$$\nabla \cdot \mathbf{u} = 0 \quad (2)$$

while the incompressible Navier-Stokes equation writes

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} - \nu \nabla^2 \mathbf{u} = -\frac{1}{\rho} \nabla p + \mathbf{g}. \quad (3)$$

As for the electrostatics part, neglecting magnetic influences in Maxwells equations, the electric field can be expressed in terms of the gradient of the electric potential

$$\mathbf{E} = -\nabla \phi \quad (4)$$

which leads us to the Poisson equation for electric potential with a source term including the space charge density ρ_{el}

$$\nabla^2 \phi = -\frac{\rho_{el}}{\epsilon_0}. \quad (5)$$

Equations (4) and (5) are subject to the conservation of electrical charges such that

Table 1. Definition of model parameters

	Charging Electrode $\partial\Omega_1$	Grounded Electrode $\partial\Omega_2$	Domain Ω
Radius r [m]	0.002	0.2	
Electric potential ϕ [kV]	50	0	
Space charge density ρ_{el} [Cm ⁻³]	0.001	-	
Fluid velocity u_r [ms ⁻¹]	50	-	
Corona onset field strength E_0 [Vm ⁻¹]	5.01×10^6		
Fatigue factor f_r	1		
Relative gas density m_s [-]			1
Ion mobility b [m ² V ⁻¹ s ⁻¹]			0.001
Vacuum permittivity ϵ_0 [Fm ⁻¹]			8.85×10^{-12}
Relative permittivity ϵ_r [-]			1
Standard temperature T_0 [K]			298.15
Standard pressure p_0 [atm]			1

$$\frac{\partial \rho_{el}}{\partial t} + \nabla \cdot \mathbf{j} = 0. \quad (6)$$

While the current density \mathbf{j} is given by

$$\mathbf{j} = \rho_{el}(\mathbf{u} + b\mathbf{E}) \quad (7)$$

which forms the core of the present investigation due to its convective part $\rho_{el}\mathbf{u}$ which is often neglected alongside with diffusion.

3.2 Implementation

The following description points at implementing electrostatics into OpenFOAM® syntax. The customization of the solver can be achieved by compiling the built-in *icoFoam* structure with additional lines of code for the electrostatics part. Firstly, the governing equation for Poissons equation (5)

```
solve
(
    fvm::laplacian(phi_e) + rho/epsilon0
);
```

with a newly defined variable `phi_e` for the electric potential. Second point in the iteration procedure is the convective transport of the space charge, as $b\mathbf{E} + \mathbf{u}$ as seen in equation (7)

```
rhoFlux = -b*mesh.magSf()*fvc::snGrad(phi_e)
+phi;
```

where `phi` accounts for the fluid-flow velocity field \mathbf{u} . The variable `phi` is pre-defined by *icoFoam* and shall not be confused with the newly defined variable for the electric field `phi_e`. Finally, to ensure charge conservation as given by equation (6) the following step shall be added as

```
solve
(
    fvm::ddt(rho) + fvm::div(rhoFlux, rho)
);
```

For postprocessing purposes it proves useful to include an optional command which writes a separate solution file for the electric field, that is

```
Efield == -fvc::grad(phi_e);
```

To write out an additional solution file to the electric field changes must be included also in the *createFields.H* file. Moreover, in the "0" folder an *Efield* file must be specified. Because the value of the electric field is determined by the solution variable for the electric potential, it is recommended not to override the value of the electric field at the boundaries of the electrode. Instead, the boundary condition `extrapolatedCalculated` in the electrical field file for both electrode boundaries shall be used to ensure that the electrical field variable is written out after and based on the electric potential variable during each iteration taken by the solver.

3.3 Mesh

For the calculations a blockmesh is used with 150 cells in radial direction and 80 cells in azimuthal direction, while one cell is present in axial direction. The radial distribution of the cells is set with `simpleGrading(400)`, resulting in a first cell width to radius ratio of 0.02 according to figure 3. The high resolution of the radial component in proximity of the charging electrode is needed to ensure the correct solution of high gradients of the electric field.

The total number of cells amounts to 12'000 Finite Volumina which on a 2.6 GHz CPU with 8 GB RAM require 244 seconds of CPU runtime. The simulation time goes from 0 s to 0.05 s with a timestep of 5×10^{-6} s.

4 Analytical Verification

The coaxial cylinder test-case geometry allows the formulation of an analytical solution for the electric field. As no changes in axial direction are expected, the volumetric flow per unit length can be expressed in terms of the radial velocity as:

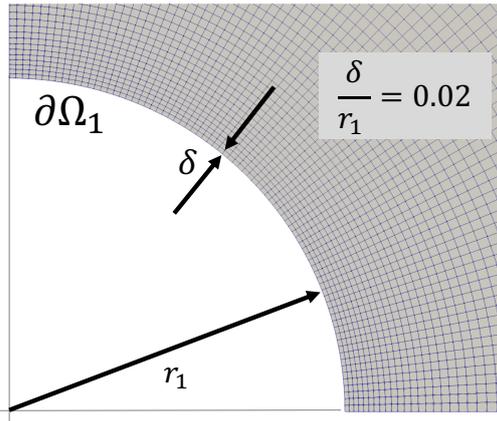


Figure 3. Cutout of the mesh used in the calculations with focus on the charging electrode

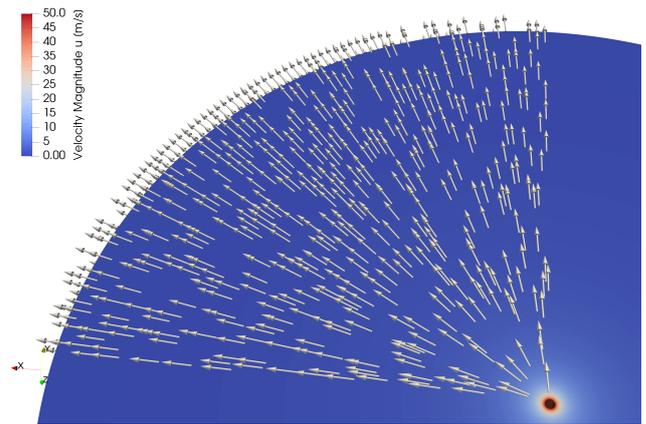


Figure 4. Visualization of the velocity field

$$V = 2\pi r u_r \quad (8)$$

whereas the electric current is quantified in a similar way including the current density in radial direction

$$I = 2\pi r j_r = 2\pi r (\rho_{el}(u_r + bE_r)) \quad (9)$$

To work around integration constants, the volumetric flow and electric current are non-dimensionalized using the quantities listed in table 1 as

$$\hat{V} = \frac{V}{2\pi b r_1 E_0} \quad (10)$$

$$\hat{I} = \frac{I}{2\pi \epsilon_0 \epsilon_r b E_0^2} \quad (11)$$

With the non-dimensional radius $\hat{r} = r/r_1$ the dimensionless electric field takes the form

$$\hat{E} = \frac{-\hat{V} \pm \sqrt{\hat{I}(\hat{r}^2 - 1)(\hat{V} + 1)^2}}{\hat{r}} \quad (12)$$

for the case of positive convection, that is, when the electric current and fluid flow have the same direction. In countercurrent the term $(\hat{V} + 1)^2$ becomes $(\hat{V} - 1)^2$.

5 Results

5.1 Velocity field

The velocity field is shown in figure 4 with unscaled velocity vectors. As expected, the flow starts from the electrode with an intensity of 50 m s^{-1} . Radially, the velocity decreases rapidly, due to large spatial increase and dispersion.

5.2 Electric field

The resulting electric field is shown in figure 5 and figure 6. It can be seen that a fine mesh resolution around the charging electrode is necessary to reliably represent the sharp gradients of the electric field.

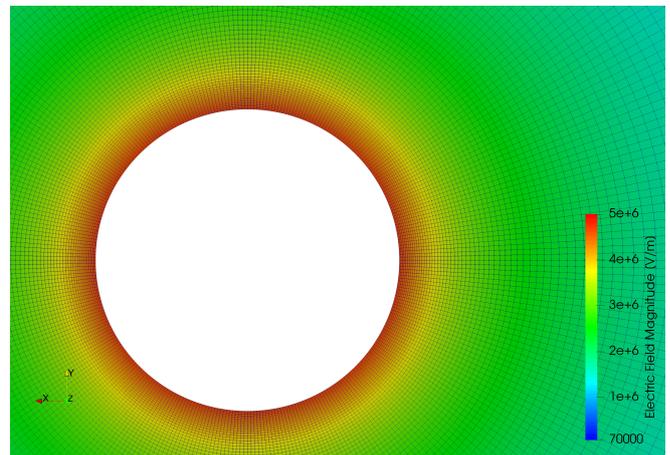


Figure 5. Close-up view of the electric field

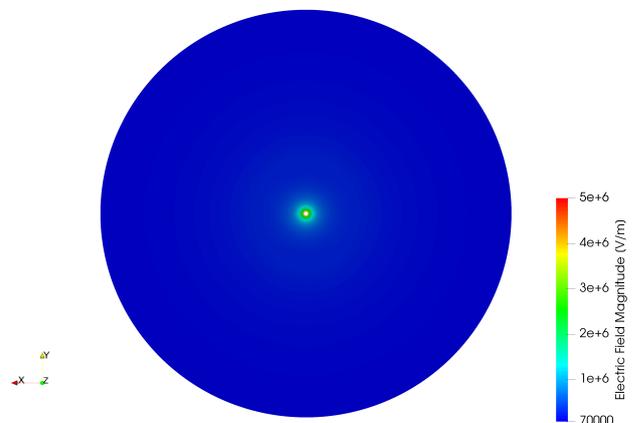


Figure 6. Global view of the electric field

For the given setup the solution for the radial distribution of the coaxial geometry is analytically verified as shown in figure 7.

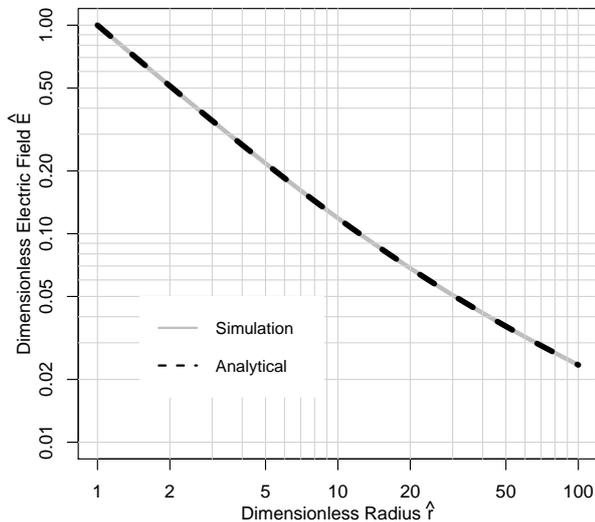


Figure 7. Verification of the electrical field along the radius

5.3 Discussion

In this hypothetical test-case the results for the electrostatics are analytically verified which consequently confirms the validity of the presented modeling procedure. For this geometry with the given model parameters for air there is no noticeable influence of convection. In many engineering applications where ionization processes are part of a larger construct the assumption of negligible convection holds.

However, as illustrated by the sensitivity analysis in figure 8 under certain flow patterns and regimes the influence of convection might be of considerable importance, *i.e.* for supersonic velocities the electric field experiences an increase further away from the electrode. In practice, this effect can be understood as deviation of current, where to the most extreme extent it could be possible to reverse the direction of electrical current flow merely by external convection.

For the test-case and the verification of the modeling strategy a constant value for the initial space charge density ρ_{el} is appropriate. Future work will target a more complex FVM modeling methodology aiming at dynamically computing the value of the space charge density within the interactions taken by the solver. The interested reader may be referred to (Rubinetti et al., 2017) for an FEM-based approach.

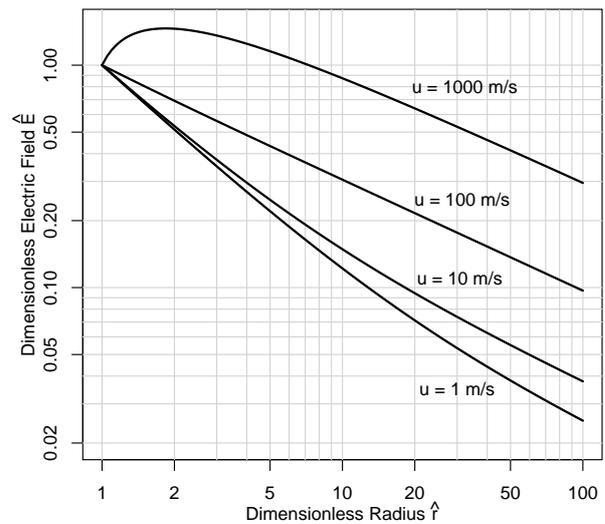


Figure 8. Sensitivity analysis for the electric field with different velocities assuming ion mobility $b = 10^{-6} \text{m}^2 \text{V}^{-1} \text{s}^{-1}$

6 Conclusion

The presented test-case and the corresponding modeling methodology for OpenFOAM® is numerically stable and shows the suitability of FVM for the multiphysical coupling of electrostatics and fluid dynamics. For the case of air ionization and an overlaying velocity field with 50ms^{-1} peak velocity convection has no noticeable influence - thus confirming the wide-spread modeling assumption of negligible external convection for most engineering applications involving Corona discharge. The results have been analytically verified including a sensitivity analysis showing that for certain flow regimes convection might need to be considered to ensure physical accuracy.

Acknowledgements

Gernot Boiger and Dirk Wilhelm are gratefully acknowledged for the fruitful discussions conducted during the development of the customized OpenFOAM® solver in this work.

References

- Ulrich Kogelschatz, Walter Egli, and Edgar A. Gerteisen. Advanced computational tools for electrostatic precipitators. *ABB review*, (4):33–42, 1999.
- Donato Rubinetti, Daniel A Weiss, and Walter Egli. Electrostatic precipitators—modelling and analytical verification concept. In *University of Applied Sciences Northwestern Switzerland, Windisch, Switzerland, COMSOL CONFERENCE*, 2015.
- Donato Rubinetti, Daniel A. Weiss, and Walter Egli. Corona

discharge-a fully coupled numerical approach verified and validated. *International Journal of Multiphysics*, 11(4), 2017.

Harry J. White. Industrial electrostatic precipitation. In *Industrial electrostatic precipitation*. Addison-Wesley, 1963.

Using the concept of data enclosing tunnel as an online feedback tool for simulator training

Laura Marcano^{1,2} Anis Yazidi³ Davide Manca⁴ Tiina Komulainen¹

¹ Department of Electronic Engineering, OsloMet – Oslo Metropolitan University, Norway

² Department of Electrical Engineering, University College of Southeast Norway (USN), Norway

³ Department of Computer Science, OsloMet – Oslo Metropolitan University, Norway

⁴ Department of Chemical Engineering, Politecnico di Milano, Italy

{Laura.Marcano, Anis.Yazidi, Tiina.Komulainen}@oslomet.no
davide.manca@polimi.it

Abstract

Feedback is one of the key factors that makes industrial simulator training an effective learning tool. Usually, the trainees receive feedback from the instructor, who guides them through the simulation tasks. However, nowadays the availability of expert instructors is scarce while the training demand is increasing. Therefore, there is a need for new simulator training practices that could allow the trainees to be more independent and decrease the need to rely so often on the instructor. This could be achieved by offering the trainees online automated feedback. This article presents a method for developing a tool meeting those requirements is presented. Simulation data were gathered representing different execution paths of the same scenario. Data were then analyzed and clustered using different clustering techniques. Interestingly, “good” and “bad” performances are shown to be separable using different techniques for clustering multivariate time series. Furthermore, we introduce the concept of enclosing data tunnel representing the trajectory of well-behaving execution paths in a reduced dimensional space. By conditioning the mal-behaving performances to be less than 20 % of the total simulation time inside the tunnel, an accuracy on 68 % was obtained. Being more flexible and allowing the mal-behaving performances to be inside the tunnel for a maximum of 35 % of the total simulation time, the accuracy of the enclosing tunnel was increased to 84 %.

Keywords: simulator training, online feedback, data clustering, enclosing tunnel, execution path

1 Introduction

A number of studies point out the importance of feedback during simulator training (Darken, 2009, Dozortsev, 2013, Håvold et al., 2015, Kluge et al., 2014, Kluge et al., 2009, Salas et al., 2012, Tichon and Diver, 2010). Feedback is a very effective learning mechanism that can be used to guide the trainees towards the development of a better performance. According to Salas et al. (2012) “Practice is most powerful when

combined with timely, constructive, and diagnostic feedback”. Usually, trainees receive feedback from an expert instructor who guides them through the simulation scenarios. Commonly, expert instructors are experienced operators who have accumulated a great knowledge of the system through years. This dependency on expert instructors has raised concern in different industrial sectors given that many of the experienced operators have retired or will retire in the near future (Alamo and Ross, 2017, Dozortsev, 2013, Manca et al., 2012). Consequently, the availability of expert instructors is continuously decreasing while the demand for operator training continues increasing. Therefore, there is need for new (tutoring) methods and techniques that could help the current instructors to cope with the training demands by allowing the trainees to be more independent. One way trainees can be more independent is by offering them online automated feedback about their performance during the training scenarios.

The topic of automated feedback for simulator training has been mentioned in several studies, some of them indicate that the use of instructional tools embedded in the simulator can improve the efficiency of training (Bell et al., 2008, Malakis and Kontogiannis, 2012). Further, there are studies that present a method (Manca et al., 2014) or an already developed tool (Dozortsev, 2013) to give automated feedback.

In this work, a procedure for developing an automated feedback tool for a simulator training is presented. The procedure developed is based on the analysis of the data collected from different variations of the same training scenario. The analysis of the data allows defining a good performance reference. This work builds on previous ideas from our position paper (Marcano et al., 2017b).

In order to be able to provide online automated feedback, it is necessary to know the state of the system at all times. The state of the system can be determined based on some key variables that together give a suitable overview of the process. These variables can be compared to the defined good performance reference.

Based on the results of the comparison the trainees can be informed whether they should reconsider the actions taken and try a different approach to solving the training task.

The case study we considered is a training scenario developed in K-Spice, a dynamic process simulation tool from Kongsberg Oil and Gas Technologies (Kongsberg, 2009). K-Spice resorts to a generic oil and gas production model. The goal of the studied training scenario is to increase the overall oil production flow with respect to the starting point.

In the next section, the procedure followed for developing the online feedback tool is described in detail, then the results obtained are presented, followed by the discussion, and finally some conclusions are drawn.

2 Methodology

The case study consists of a training scenario developed with the generic oil and gas production model integrated into the simulation tool K-Spice. Aim of the training scenario is to increase, in 30 min, the oil production flow with respect to the one given in the initial conditions of the simulation. The trainee must fulfill the goal without compromising the correct functioning of the process. In order to develop an online feedback tool for the case study, the following steps were followed.

2.1 Selection of variables

Figure 1 shows an overview of the generic oil and gas production model. The sections with the most relevant information of the process are the wells, the high-pressure separator (HP-separator), the export pump and the gas export compressor, the oil and gas export sections, and the high pressure flare (HP-flare). The variables studied were taken from these sections, being: 1) The total sum of outlet flows from the wells; 2) Inlet flow of the HP-separator; 3) Pump power consumption; 4) Compressor power consumption; 5) Oil export flow; 6) Gas export flow; and 7) HP-flare flow.

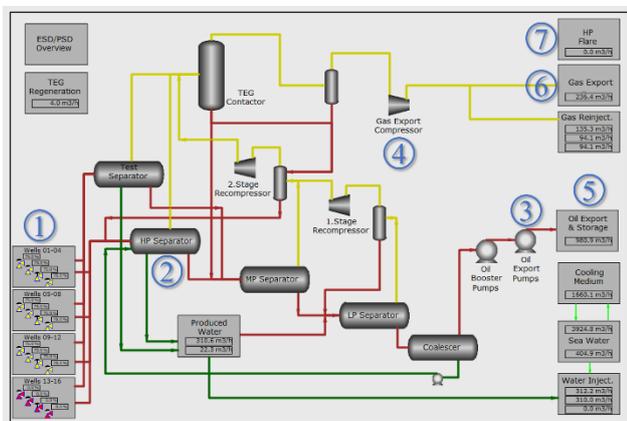


Figure 1. Overview of the generic oil and gas production process.

2.2 Data generation

In order to gather relevant data, a method to generate variations of the case study was developed. Each process variation was a random selection of five possible actions. The actions were defined based on the observations and results gathered from the simulator training sessions mentioned in Marcano et al. (2017a). During the development of the research (Marcano et al., 2017a), it was possible to extract knowledge about the students' understanding of the process and the probable actions they could execute. Based on this distilled knowledge, it was decided that a maximum of three actions could take place per variation of the case study, with a certain delay between them. The delay was set to 15 s, 45 s, 60 s, 120 s, and 180 s. These delays were chosen because during the simulator training sessions (Marcano et al. (2017a) we noticed that the trainees usually did not wait longer than 3 min to make changes in the simulation. The construction of one variation occurs as follows; first, a random action is chosen, and then, depending on the chosen first action there are some conditions that will determine the following random actions to choose among if any. The defined actions and the conditions triggering them are explained below.

1. Opening the choke valve from a well.

Opening a choke valve is the right execution path to follow when trying to increase the oil production in the process. Therefore, it is assumed that if the first decision of the trainee is to open a choke valve, then, if there are possible following actions, these will also be opening a choke valve. How much the choke is going to be opened is decided randomly between two options, being 85 % and 100 %. All choke valves in the simulation that are predetermined open, are open up to 75 %.

2. Closing the choke valve of a well.

Closing a choke valve is a wrong action to execute if the oil production needs to be increased. If the trainee is confused and closes a valve by mistake, then the next actions could, unfortunately, be to close even more valves. However, it could also happen that the student notices the mistake and tries to fix it by reopening the closed valve and opening an extra one. Then we randomly decide whether we will perform a sequence of 1 or 2 next actions. In case of choosing only one subsequent action, that second action could be either closing another valve or reopening the one closed. In case we choose two subsequent actions, then, the following two actions are to reopen the closed valve and open an extra one. How much the choke is going to be closed is decided randomly between two options, being 0 % and 65 %. For the opening, the same conditions explained in the first action are applied.

3. Opening the pulse-controlled valve to the test separator.

During the simulator training sessions carried out in Marcano et al. (2017a), it was noticed that few students

opened a pulse-controlled valve thinking that it was a choke valve. This mistake was also noticed during the simulator training sessions performed later in 2017. There were just a few students who made the mistake but it seems to be common to happen. Consequently, it was decided to take it into account. However, given that opening a pulse-controlled valve is a rare mistake if this action takes place first, then, it will be the only action to be executed and there will be no subsequent actions.

4. Opening the outlet control valve of the HP-separator.

Opening the outlet valve of the HP-separator might occur due to the misconception that by increasing the outlet flow from the HP-separator the oil production would increase as well. The next step is to choose whether to proceed with a sequence of one subsequent action or two subsequent actions. If we randomly chose to follow with two actions, then, these were set to be the opening of choke valves. If only one action is following, then, this could be either opening a choke valve or a pulse-controlled valve.

5. Increasing the pressure set point of the HP-separator.

Increasing the pressure of the HP-separator leads the system to switch on the high-pressure flare. This action was defined to ensure the possibility of analyzing execution paths with a negative environmental impact. If only one more action follows this one, then, it could be either opening a choke valve or a pulse-controlled valve. If two actions follow, then, both will be to open a choke valve.

2.3 Data clustering

The data gathered in this work consists of multivariate time series. It is necessary to identify from the data what corresponds to well-behaving performances and what corresponds to mal-behaving performances. This way it is possible to make balanced groups for training and validation. In order to cluster the data, it is necessary to use a notion of similarity. This can be done by calculating the distance between every possible combination of pairs of execution paths. In this work, three methods for distance calculation were evaluated namely: Euclidean distance, dynamic time warping (DTW), and symbolic aggregate approximation (SAX). This was done in order to determine and select the most accurate method among such distances.

Euclidean distance

Given two time series X and Y of the same length N , (1) defines the Euclidean distance between them. Figure 2a shows an intuitive representation of the Euclidean distance (Lin et al., 2003).

$$D(X, Y) \equiv \sqrt{\sum_{i=1}^N (x_i - y_i)^2} \quad (1)$$

Dynamic Time Warping (DTW)

The objective of DTW is to compare two (time-dependent) sequences $X := (x_1, x_2, \dots, x_N)$ of length $N \in \mathbb{N}$ and $Y := (y_1, y_2, \dots, y_M)$ of length $M \in \mathbb{N}$. These sequences may be discrete signals (time-series) or, more generally, feature sequences sampled at equidistant points in time (Müller, 2007). Being \mathcal{F} a feature space, $x_n, y_m \in \mathcal{F}$ for $n \in [1:N]$ and $m \in [1:M]$. To compare two different features $x, y \in \mathcal{F}$, a local cost measure is needed, also referred to as local distance measure, which is defined to be a function $c: \mathcal{F} \times \mathcal{F} \rightarrow \mathbb{R}_{\geq 0}$ (Müller, 2007).

Typically, $c(x, y)$ is small (low cost) if x and y are similar to each other, and otherwise $c(x, y)$ is large (high cost). Evaluating the local cost measure for each pair of elements of the sequences X and Y , the cost matrix $C \in \mathbb{R}^{N \times M}$ defined by $C(n, m) := c(x_n, y_m)$ is obtained. Then the goal is to find an alignment between X and Y having minimal overall cost (Müller, 2007). Each matrix element (i, j) corresponds to the alignment between the points x_i and y_j . A warping path is created, which consists of a contiguous set of matrix elements that defines a mapping between X and Y (Keogh and Ratanamahatana, 2005). The signal with an original set of points $X(\text{original}), Y(\text{original})$ is transformed to $X(\text{warped}), Y(\text{original})$. However, even though DTW measures a distance-like quantity between two given sequences, it does not guarantee the triangle inequality to hold (Müller, 2007).

SAX (Symbolic Aggregate approXimation)

SAX allows a time series of arbitrary length N to be reduced to a string of arbitrary length w , ($w < N$, typically $w \ll N$). The alphabet size is also an arbitrary integer a , where $a > 2$. SAX uses an intermediate representation between the raw time series and the symbolic strings. First, the data is transformed into the Piecewise Aggregate Approximation (PAA) representation and then symbolize the PAA representation into a discrete string (Lin et al., 2003).

In dimensionality reduction via PAA a time series X of length N can be represented in a w -dimensional space by a vector $\bar{X} = \bar{x}_1, \dots, \bar{x}_w$. The i^{th} element of \bar{X} is calculated as follows (Lin et al., 2003):

$$\bar{x}_i = \frac{w}{N} \sum_{j=\frac{N}{w}(i-1)+1}^{\frac{N}{w}i} x_j \quad (2)$$

Having transformed a time series database into PAA, a further transformation to obtain a discrete representation can be applied. SAX uses a discretization technique that produces symbols with equiprobability (Lin et al., 2003). If the original subsequences in the Euclidean distance are transformed into PAA representations, \bar{X} and \bar{Y} , using (2), then a lower bounding approximation of the Euclidean distance between the original

subsequences can be obtained (3), this is illustrated in Figure 2b (Lin et al., 2003).

$$DR(\bar{X}, \bar{Y}) \equiv \sqrt{\frac{N}{w} \sum_{i=1}^w (\bar{x}_i - \bar{y}_i)^2} \quad (3)$$

If the data is further transformed into the symbolic representation, a MINDIST function that returns the minimum distance between the original time series of two words can be defined by (4), which is illustrated in Figure 2c (Lin et al., 2003).

$$MINDIST(\hat{X}, \hat{Y}) \equiv \sqrt{\frac{N}{w} \sum_{i=1}^w (dist(\hat{x}_i, \hat{y}_i))^2} \quad (4)$$

The *dist()* function can be implemented using a table lookup as shown in Table 1. Table 1 is for an alphabet of cardinality 4. The distance between two symbols can be read off by checking the corresponding row and column. For example, *dist(a,c) = 0.67* (Lin et al., 2003).

Table 1. A lookup table used by the MINDIST function. This table was taken from Lin et al. (2003).

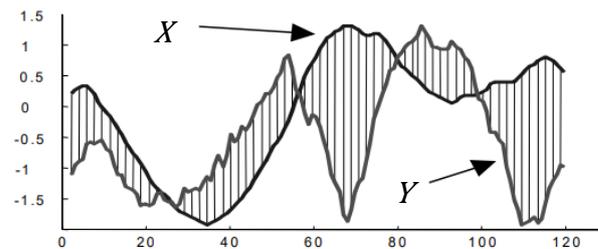
	a	b	c	d
a	0	0	0.67	1.34
b	0	0	0	0.67
c	0.67	0	0	0
d	1.34	0.67	0	0

Hierarchical clustering

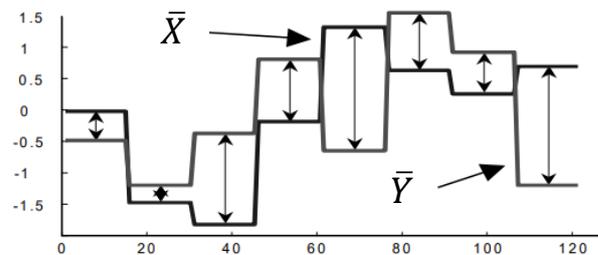
Hierarchical clustering groups data over a variety of scales by creating a cluster tree. The tree is not a single set of clusters, but rather a multilevel hierarchy, where clusters at one level are joined as clusters at the next level. This allows deciding the level or scale of clustering that is most appropriate for the application required (MathWorks, 2018).

2.4 Samples selection

For this study, 75 out of 1145 possible variations were generated the way described in Section 2.2, this represents 6.5 % of all the possible combinations. The data were classified using the methods described in Section 2.3. Of the 75 variations, 2/3 of the data corresponding to the well-behaving execution paths, and 2/3 of the data corresponding to the mal-behaving execution paths were used for training i.e. 50 variations in total. The rest of the data was used for validation i.e. 25 variations (1/3 of the good performances, and 1/3 of the bad performances). For each time series, one sample was taken every 12 s during 30 min i.e. 150 data points, plus one additional point at time zero.



(a)



(b)

$$\hat{X} = \mathbf{b a a b c c b c}$$

$$\hat{Y} = \mathbf{b a b c a c c a}$$

(c)

Figure 2. a) Euclidean distance between two time series. b) Distance measure defined for the PAA. c) SAX representations of two time series. This figure was taken from Lin et al. (2003).

2.5 Introducing the concept of enclosing tunnel

2.5.1 Data processing and dimensionality reduction

The training data was processed using PCA. In order to compare with each other all the variations in the training data, a PCA analysis was done for different time slots of the time series. For each time slot, the average of the data within the range was taken. The time moving average was calculated using the sliding window algorithm, with a window size of 35 samples. Which means that the first PCA calculated describes all the variations in the time span between the first and the 35th sample. The second PCA describes all the variations in the time span between the second and the 36th sample, and so forth until the entire time-range of 30 min (151 samples) is covered. After testing different sizes for the sliding window algorithm, a size of 35 samples was selected given that it was the one that resulted in the smoothest graphical representation of the data. The variance results show that the first and second principal components (PC1 and PC2, respectively) are the most representative, as shown in Figure 3. Figure 3a shows the first 19 sliding windows, it can be seen that the first two principal components describe approximately 70 to

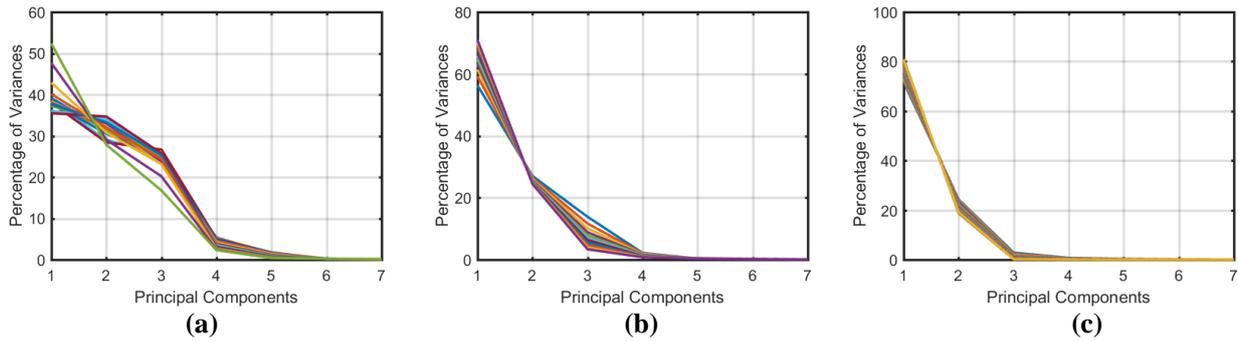


Figure 3. Variances. a) Variances of the first 19 sliding windows. b) Variances from the 20th to the 30th sliding window. c) Variances from the 31st to the 117th sliding window.

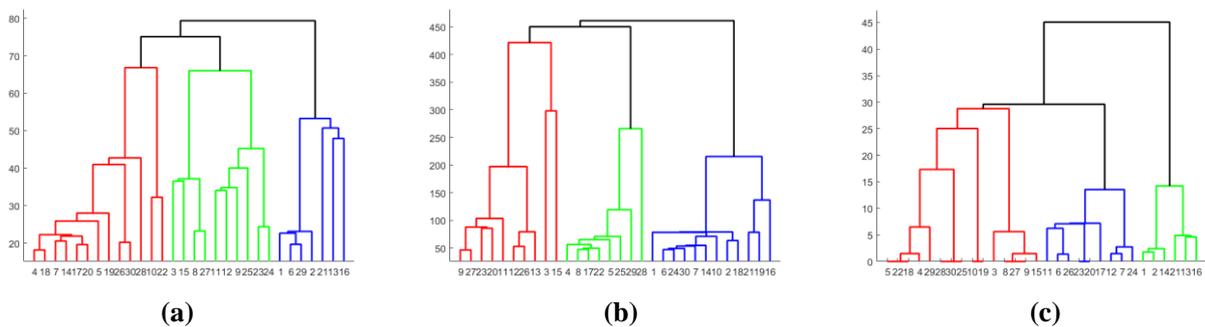


Figure 4. Hierarchical clustering based on time series distances. a) Euclidean b) DTW c) SAX.

80 % of the data. Further, Figure 3b and Figure 3c show that the first two principal components describe approximately 90% of the data. Therefore, all analyses made are based on the first two principal components.

2.5.2 Data delimitation using an enclosing tunnel

The structure of the enclosing data tunnel is based on five main circles. The enclosing circles were calculated based on the data projected on the 2D plane formed by the data scores from PC1 vs the scores from PC2. In order to frame the data projected on this plane we resort to the minimal enclosing circle problem. The minimal enclosing circle problem consists of finding the circle of smallest radius that contains a given set of points in its interior or on its boundary. Jung’s theorem states that every finite set of points with geometric span d has an enclosing circle with radius no greater than $d/\sqrt{3}$ (Weisstein, 2018). Each circle is located in a moment in time in which the well-behaving execution paths show a significant tendency of change. The data enclosing tunnel was constructed by creating a surface that connects each of the sections formed between the minimal circles.

3 Results

We started by clustering the raw data to separate the well-behaving and mal-behaving performances. The

data clustering was carried out implementing the three methods described in the methodology. Once the distances were calculated with the three methods, they were clustered using hierarchical clustering. Figure 4 shows the three clustering trees obtained with each of the methods. In general, it can be seen that there are three main clusters formed by the data, given that three main groups (green, red and blue) were obtained with each method. However, the two main branches of the cluster tree obtained with the SAX method (Figure 4c) are more noticeable than the two main branches of the other two methods, Euclidean and DTW (Figure 4a, Figure 4b), which indicates that the clusters formed by the SAX method are defined more clearly. Numerically this is checked with the cophenetic correlation coefficient, which resulted to be 0.9347 for the clustering tree calculated with the Euclidean distances, 0.8769 for the clustering tree calculated with the method DTW, and 0.9392 for the clustering tree calculated with the method SAX. Therefore, the results obtained with SAX were the one used for classifying the data as good and bad performances.

Figure 6 shows the results of the data processing using PCA. Figure 6a depicts a 3D representation of the variation along time, of the scores obtained with the first two principal components. It can be noticed that there are three different patterns in the data. Figure 6b corresponds to an upper view of the previous one. It

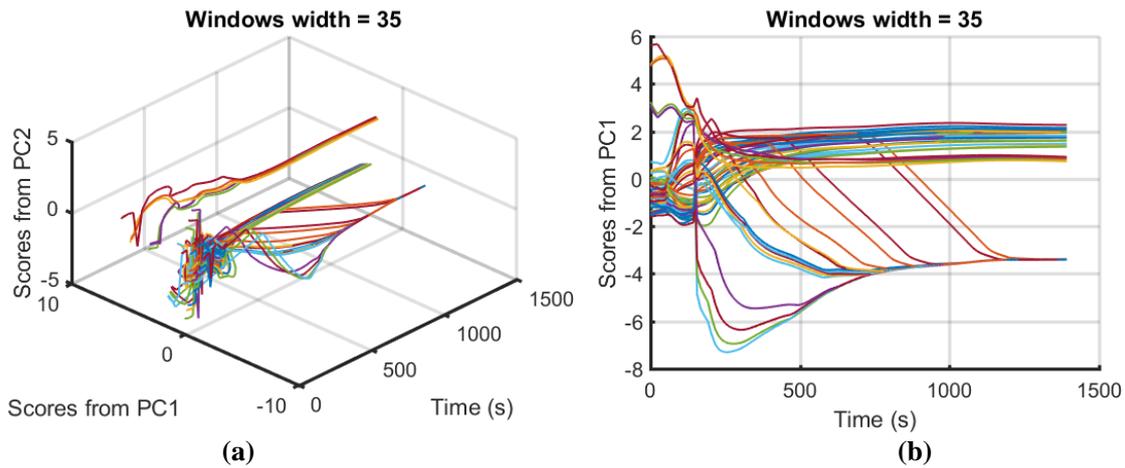


Figure 5. a) 3D view of the enclosing tunnel. **b)** 2D view of the enclosing tunnel.

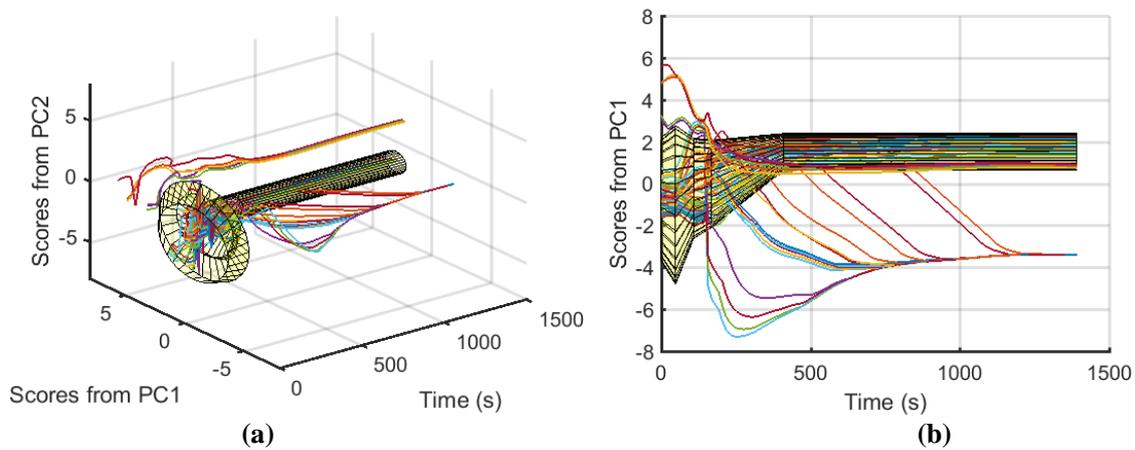


Figure 6. a) Data scores from PC2 vs Data scores from PC1 vs Time. **b)** Data scores from PC1 vs Time.

shows the variation of scores obtained with the first principal component versus time, and three different tendencies of the data can also be appreciated.

Figure 5 shows a 3D and 2D view of the tunnel enclosing the data that correspond to the good performances. The tunnel consists of five different circular sections that correspond to the minimal enclosing circle of the well-behaving execution paths in each section. All data that do not fall inside the tunnel corresponds to a bad performance. The trends that start being inside the tunnel but then go totally outside correspond to those actions where the outlet controlled valve of the HP-separator is opened. The trends that are above the tunnel correspond to those actions where the pressure set point of the HP-separator was increased.

In order to test the accuracy of the tunnel, the data left aside for validation was used. First, the validation data was projected on the PCA space calculated with the training data. Next, the processed validation data was plotted with the tunnel, as shown in Figure 7. Finally, the accuracy of the tunnel was determined by calculating the total amount of time that each good and bad

performance spent inside the tunnel. In the case of the well-behaving execution paths, it was expected that they would be inside the tunnel at least 80 % of the total simulation time. While in the case of the mal-behaving execution paths, it is expected that they wouldn't be inside of the tunnel more than 20 % of the total simulation time. Based on these limits the accuracy calculated for the tunnel resulted to be 68 %, as shown in the confusion matrix presented in Figure 8a. The diagonal of the confusion matrix (green squares) represents the correct classifications. Figure 8a shows that 12 execution paths were correctly classified as "good", and 5 execution paths were correctly classified as "bad". On the other hand, the red squares show the incorrect classifications, and it can be seen that 8 execution paths were wrongly classified as "good", these are false positives. In order to improve the accuracy of the tunnel, the tolerance of the bad performances inside of the tunnel was increased to 35 % of the total simulation time. This way the false positives were reduced from 8 to 4, given as a result an improved accuracy of 84 % as can be seen from Figure 8b.

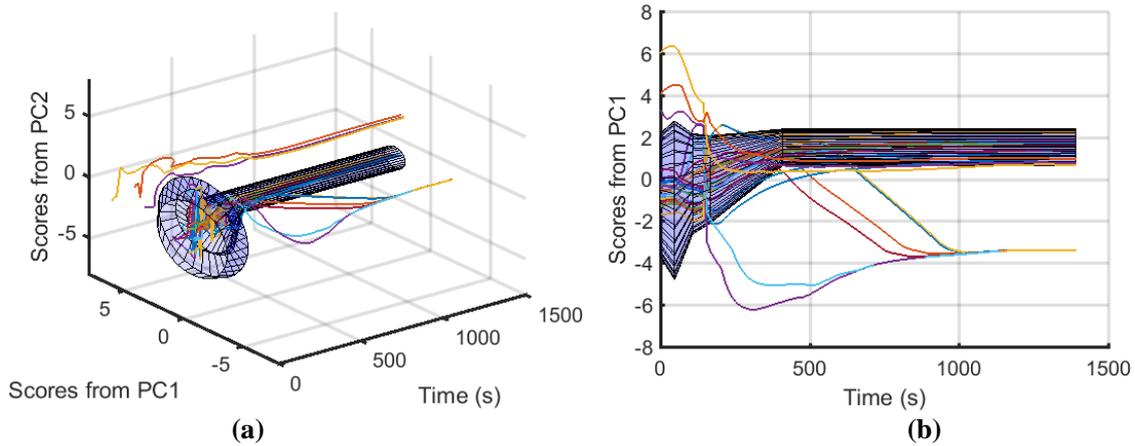


Figure 7. a) Validation data and 3D view of the enclosing tunnel. b) Validation data and 2D view of the enclosing tunnel.

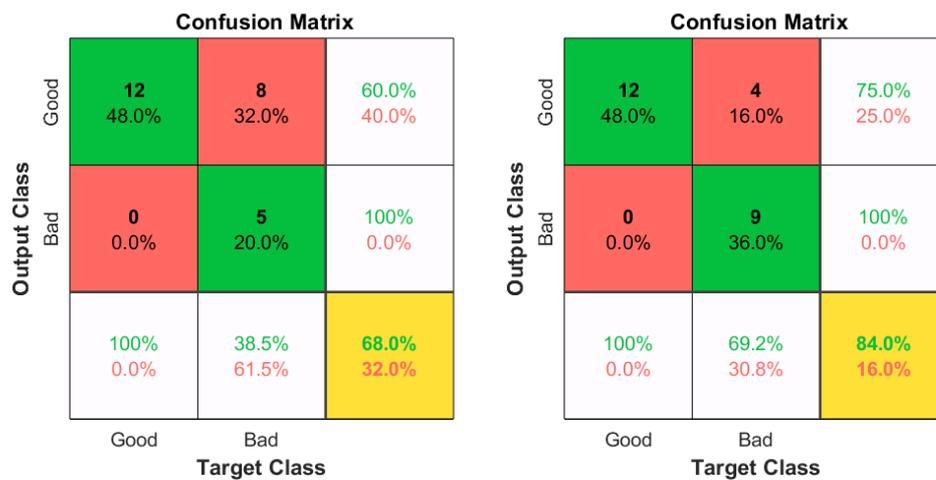


Figure 8. a) Confusion matrix: mal-behaving execution paths inside of the tunnel for 20 % of the total simulation time. b) Confusion matrix: mal-behaving execution paths inside of the tunnel for 35 % of the total simulation time.

4 Discussion

In this work, the construction of the online feedback tool was based on the well-behaving execution paths. Consequently, it was necessary to find methods that could ease the laborious task of clustering data and identifying their typology. The results obtained with the data clustering techniques show that these are effective methods for finding similarities among data. Which is very useful when handling a large amount of data, such as those produced from simulator training scenarios.

The proposed enclosing tunnel could be used as an effective tool for generating online feedback. The data of a new trainee could be monitored, for instance, every two minutes. The first set of data should be projected on the PCA space, and later compared with the tunnel, if it is observed that the execution path is outside the tunnel a warning can be given to the trainee. If the execution path is inside the tunnel no warning should be generated. Later, in the next two minutes, the data of the last four minutes could be analyzed, and once again depending

on the data trend it is decided if a warning should be given to the operator or not. This sequence should be repeated online over the total duration of the simulation scenario. Further, depending on the data behavior we could also determine the type of mistake made by the trainee, and more detailed feedback could be generated. This refers particularly to the cases in which the outlet valve of the HP-Separator is opened, and when the pressure set point of the HP-Separator is increased. These two cases present a very differentiated behavior around the tunnel, therefore it could be easy to identify them. However, the trends for the cases where the outlet control valve of the HP-Separator is opened may take several minutes before leaving the tunnel. These are the cases that were classified correctly by allowing them to be inside the tunnel 35 % of the total time.

In general, this method could be used for different training scenarios. This procedure shows that an enclosing tunnel, based on good performances, can be designed for any kind of scenario, thus online feedback can be offered to the operators, giving them more training independence.

5 Conclusion and future work

The data clustering methods implemented, Euclidean distance, DTW, and SAX showed to be effective for finding similarities among data. Of the three methods, SAX is shown to be the most effective of all with a cophenetic correlation coefficient of 0.9392. The clustering of the data helped to identify among the entire data set the well-behaving execution paths, which were used to design the online feedback tool for simulator training. The online feedback tool designed consists of an enclosing data tunnel. The tunnel developed has in principle an accuracy of 68 %. This value was calculated by allowing the mal-behaving execution paths to be inside the tunnel no more than 20 % of the total simulation time. However, with a more flexible tolerance (bad performances allowed to be inside the tunnel 35 % of the total simulation time) the total accuracy of the tunnel could be increased up to 84 %. It was demonstrated that it is possible to develop a method that can be used to generate automated online feedback, thus opening the possibility of more independent simulator training sessions.

Future work includes improving the accuracy of the tunnel without increasing the tolerance for mal-behaving execution paths. This could be done by increasing the amount of training data, so more differences can be noticed among the time series. Additionally, the method should also be improved so that it can detect if more specific requirements have been fulfilled by the operator. Furthermore, enclosing tunnels constructed for different training scenarios should be compared to each other in order to determine if a single generic tunnel could be designed to be used for different training scenarios.

References

- Alamo, J. & Ross, M. 2017. Solve operator training in a challenging refining industry. *Hydrocarbon Processing*, 96, 67-69.
- Bell, B. S., Kanar, A. M. & Kozlowski, S. W. J. 2008. Current issues and future directions in simulation-based training in North America. *The International Journal of Human Resource Management*, 19, 1416-1434. 10.1080/09585190802200173
- Darken, R. P. 2009. Identifying the limits of training system effectiveness through taxonomies of human performance. *Theoretical Issues in Ergonomics Science*, 10, 231-243. 10.1080/14639220802151625
- Dozortsev, V. M. 2013. Methods for computer-based operator training as a key element of training systems (Present-day trends). *Automation and Remote Control*, 74, 1191-1200. 10.1134/S0005117913070102
- Håvold, J. I., Nistad, S., Skiri, A. & Ødegård, A. 2015. The human factor and simulator training for offshore anchor handling operators. *Safety Science*, 75, 136-145. <http://dx.doi.org/10.1016/j.ssci.2015.02.001>
- Keogh, E. & Ratanamahatana, C. A. 2005. Exact indexing of dynamic time warping. *Knowledge and Information Systems*, 7, 358-386. 10.1007/s10115-004-0154-9
- Kluge, A., Nazir, S. & Manca, D. 2014. Advanced Applications in Process Control and Training Needs of Field and Control Room Operators. *IIE Transactions on Occupational Ergonomics and Human Factors*, 2, 121-136. 10.1080/21577323.2014.920437
- Kluge, A., Sauer, J., Schüler, K. & Burkolter, D. 2009. Designing training for process control simulators: a review of empirical findings and current practices. *Theoretical Issues in Ergonomics Science*, 10, 489-509. 10.1080/14639220902982192
- Kongsberg. 2009. *K-Spice: A new and powerful dynamic process simulation tool* [Online]. Available: https://www.kongsberg.com/en/kongsberg-digital/news/2009/june/0625_kpice/ [Accessed April 30 2018].
- Lin, J., Keogh, E., Lonardi, S. & Chiu, B. 2003. A symbolic representation of time series, with implications for streaming algorithms. *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*. San Diego, California: ACM. 10.1145/882082.882086
- Malakis, S. & Kontogiannis, T. 2012. Refresher Training for Air Traffic Controllers: Is It Adequate to Meet the Challenges of Emergencies and Abnormal Situations? *The International Journal of Aviation Psychology*, 22, 59-77. 10.1080/10508414.2012.635127
- Manca, D., Nazir, S., Colombo, S. & Kluge, A. 2014. Procedure for Automated Assessment of Industrial Operators. *CHEMICAL ENGINEERING*, 36, 391-396.
- Manca, D., Nazir, S., Lucernoni, F. & Colombo, S. 2012. Performance Indicators for the Assessment of Industrial Operators. In: IAN DAVID LOCKHART, B. & MICHAEL, F. (eds.) *Computer Aided Chemical Engineering*. Elsevier. <http://dx.doi.org/10.1016/B978-0-444-59520-1.50143-3>
- Marcano, L., Komulainen, T. M. & Haugen, F. A. 2017a. Implementation of performance indicators for automatic assessment. In: ANTONIO ESPUÑA, M. G. & LUIS, P. (eds.) *Computer Aided Chemical Engineering*. Elsevier. <https://doi.org/10.1016/B978-0-444-63965-3.50497-9>
- Marcano, L., Yazidi, A., Ferati, M. & Komulainen, T. 2017b. Towards Effective Automatic Feedback for Simulator Training. *Proceedings of the 58th Conference on Simulation and Modelling (SIMS 58) Reykjavik, Iceland, September 25th – 27th, 2017*. Linköping University Electronic Press, Linköpings universitet.
- Mathworks. 2018. *Hierarchical Clustering* [Online]. Available: <https://se.mathworks.com/help/stats/hierarchical-clustering.html> [Accessed 25 April 2018].
- Müller, M. 2007. Dynamic Time Warping. *Information Retrieval for Music and Motion*. Berlin, Heidelberg: Springer Berlin Heidelberg. 10.1007/978-3-540-74048-3_4
- Salas, E., Tannenbaum, S. I., Kraiger, K. & Smith-Jentsch, K. A. 2012. The Science of Training and Development in Organizations: What Matters in Practice. *Psychological Science in the Public Interest*, 13, 74-101. 10.1177/1529100612436661
- Tichon, J. & Diver, P. 2010. Plant operator simulation: benefits and drawbacks for a construction training organization. *Cognition, Technology & Work*, 12, 219-229. 10.1007/s10111-010-0140-9
- Weisstein, E. W. 2018. *Minimal Enclosing Circle* [Online]. Available: <http://mathworld.wolfram.com/MinimalEnclosingCircle.html> [Accessed 24 April 2018].

Numerical Solution of the Beam Equation for Beams subject to Large Deflections

Magnus Komperød¹

¹Technological Analyses Centre, Nexans Norway AS, Norway, magnus.komperod@nexans.com

Abstract

The bending stiffness of subsea cables and umbilicals can be identified from physical tests where the cable is supported at both ends and then pushed or pulled at the center. The beam equation can be used to calculate the bending stiffness from the measured force and the measured deflection. However, a study performed by Nexans Norway AS shows that the beam equation oversimplifies the cables' properties and thereby disregards significant effects. This paper extends the beam equation into the case of large deflections, which is the second milestone of the work on deriving an accurate model. Within a realistic range of cable deflections, the force calculated by the novel model differs with up to 20% compared to the previous, oversimplified model. From the author's point of view, this large difference justifies the somewhat increased complexity and computation time of the novel model.

Keywords: Beam Equation, Bending Stiffness, Large Deflection, Minimum Total Potential Energy Principle, Numerical Analysis, Subsea Power Cable, Umbilical.

1 Introduction

Mathematical models and finite element analyses of subsea power cables' and umbilicals' mechanical properties are used to calculate the mechanical stresses that arise when the cable is subject to external loads such as axial tension, bending, and twisting. Mathematical models of cables have been presented in the scientific literature at least since the 1960s; for example Lutchansky (1969) which is still relevant today. The field has matured since then by improving the models and including additional effects. Among the most comprehensive publications on models of cables' mechanical properties is the excellent PhD thesis of Keadze (2000).

Physical testing of cables are necessary to validate the mathematical models and the finite element analyses. However, the test rigs themselves inherently introduce new, often complex, properties to the overall system, which make it difficult to separate the cable properties from the rig properties in the measurement data. Further, the rig sensors do not measure the desired cable properties directly. Hence, mathematical models have to be developed for calculating the desired cable properties based on the sensor signals. For some kinds of physical cable tests, including bending stiffness tests, developing such models

are complex and challenging tasks.

Unfortunately, the scientific literature is sparse on physical tests of cables, and in particular sparse on models for calculating the desired cable properties from the sensor signals. Maioli (2015) presents results on traditional deflection tests as sketched in Figure 1. This publication uses beam theory derived for small deflections to calculate the cables' bending stiffness, despite the actual deflections may be rather large. Tarnowski (2015) presents results from a similar test, but with a bending stiffness calculation model that intends to handle large deflections. Also Ekeberg and Dhaigude (2016) and Dhaigude and Ekeberg (2016) present bending tests. However, these papers are sparse on the details of the calculation model.

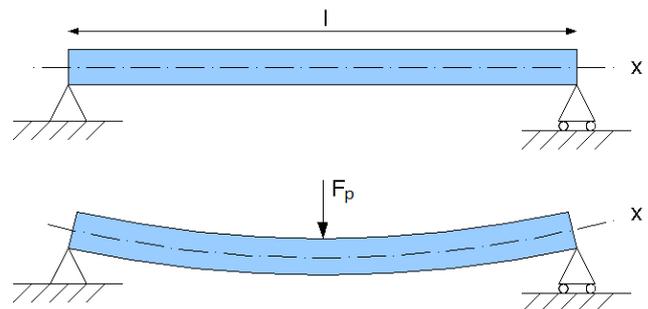


Figure 1. Sketch of bending stiffness rig. Illustration from www.wikimedia.org.

Nexans Norway AS is currently performing several R&D activities on improving modeling and physical testing of subsea power cables' and umbilicals' mechanical properties. As part of this work a novel bending stiffness rig has been developed. The rig is based on the principle sketched in Figure 1 and its details are presented in Jordal et al. (2017). Initially the cables' bending stiffness was calculated from the rig's sensors as presented in Komperød et al. (2017), which is essentially the beam equation derived for small deflections, followed by filtering and postprocessing to improve the signal quality. However, a more detailed study of cables' behavior under bending revealed that this approach is an oversimplification. Nexans Norway AS has therefore started an R&D work which aims to correctly model the cable under bending stiffness tests.

The R&D work is split into milestones for easier han-

ding the complexity of the overall modeling problem. Komperød (2018) presents the first milestone which is to rewrite the beam equation using the minimum total potential energy principle and solve this equation numerically. The motivation behind the paper is that the minimum total potential energy principle is believed to be more suitable for extending the beam equation than the force- and moment balances which are commonly used in the scientific literature for deriving this equation. Also, the paper solves the beam equation numerically because it is assumed to be impossible to solve the extended beam equation analytically.

The present paper derives and presents the results of the second milestone, which is to consistently handle large deflections. This is an important step towards correctly modeling the cables' bending stiffness based on the sensors of the bending stiffness rig, because the physical tests are often performed with large deflections. The improvement of the novel model will be demonstrated by comparing it to the model of Komperød (2018).

2 Nomenclature

Table 1 presents the main nomenclature used in this paper. Variables without an intuitive physical meaning are defined in the main text where they first appear.

Table 1. Nomenclature.

\bar{c}	Vector with Clenshaw-Curtis quadrature weights.
D_1	Chebyshev first-derivative matrix.
D_2	Chebyshev second-derivative matrix.
EI	Bending stiffness [Nm/(m ⁻¹)].
F_p	Deflection force [N].
g	Acceleration of gravity [m/s ²].
L	Length between supports [m].
M	Bending moment [Nm].
\bar{m}	Mass per unit length [kg/m].
N	Number of discretization points [-].
\bar{N}	Number of discretization points over a half beam [-].
P	Total potential energy [J].
P_g	Gravitational energy [J].
P_s	Strain energy [J].
s	Beam length parameter [m].
u	Vertical deflection of beam [m].
u_p	Vertical deflection of piston [m].
\mathbf{u}	Vector containing discrete points of u .
x	Coordinate along x -axis [m].
κ	Bending curvature [m ⁻¹].
λ_i	Lagrange multiplier no. i .

Vectors are denoted with lower case, bold font. Matrices are denoted with upper case, bold font. Square bracket are used to denote parts of a vector or a matrix. For example $\mathbf{A}[1 : 4, 3]$ means the first four rows of the third column of \mathbf{A} .

The symbol \bullet means vector dot product. The symbol \cdot applied to vectors and matrices, for example $\mathbf{a} \cdot \mathbf{b}$, means element-by-element multiplication. Exponent notation applied to a vector, for example \mathbf{x}^2 , means element-by-element exponent.

Figure 2 shows the Cartesian coordinate system used in this paper.

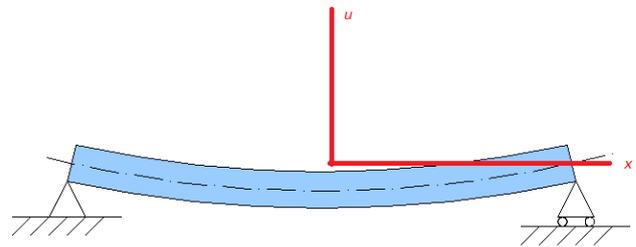


Figure 2. The Cartesian coordinate system used in this paper. Based on illustration from www.wikimedia.org.

3 Assumptions and Simplifications

The mathematical derivation in this paper is subject to the following assumptions and simplifications:

1. The beam is made of a linear, elastic material and has identical cross section over its entire length.
2. Only the beam segment between the supports shown in Figure 1 is considered. That is, possible beam segments outside these supports are disregarded.
3. The beam is assumed not to move horizontally at the center of its length, i.e. where the F arrow points in Figure 1. During the physical bending stiffness tests this constraint is enforced by the design of the bending stiffness rig.
4. The developed model is quasi-static, i.e. inertia and kinetic energy are disregarded.
5. The height of the beam is small compared to the length between the beam supports, L .

4 Total Potential Energy and Constraints

In Komperød (2018) the total potential energy, P , was found to be

$$P = \frac{1}{2}EI \int_{\text{beam length}} \kappa^2 ds \quad (1)$$

$$+ \tilde{m}g \int_{\text{beam length}} u ds.$$

The upper right term of Eq. 1 is the beam's strain energy due to bending, while the lower right term is the beam's gravitational energy. In the equation EI is the beam's bending stiffness and κ is its bending curvature, while s is the cable length parameter. Further, \tilde{m} and g are the beam's mass per unit length and the acceleration of gravity, respectively, and u is the beam's deflection. u is a function of s , while κ is a function of the first- and second derivatives of u .

Also the constraints were found in Komperød (2018). They are

$$u(\text{left end}) = 0, \quad (2)$$

$$u(\text{center}) = u_p, \quad (3)$$

$$u(\text{right end}) = 0, \quad (4)$$

$$u(s) \in C^2. \quad (5)$$

The constraints of Eqs. 2, 3, and 4 follow directly from the rig design as sketched in Figure 1. In Eq. 3, u_p is the position of the piston which applies the force F_p .

In Eq. 5, C^2 is the set all functions which are continuous and have continuous first- and second derivatives (and possibly continuous higher order derivatives). Hence, the practical interpretation of this constraint is that the beam's deflection, deflection angle, and curvature are continuous. The third derivative is known to be discontinuous at the point load F_p of Figure 1. These results are all well known from beam theory.

5 Numerical Solution

The overall problem is to minimize the total potential energy, i.e. Eq. 1, w.r.t. the constraints given by Eqs. 2 - 5. This section solves this problem using numerical mathematics.

5.1 Calculating the Total Potential Energy

The total potential energy is given by Eq. 1. In order to be calculated numerically, the integration limits must be expressed numerically, and the curvature κ must be expressed in terms of the deflection u .

It is more convenient to use the horizontal coordinate x , rather than the cable length parameter s , as the integration variable. The arc length formula relates x and s , i.e.

$$ds = \left[1 + \left(\frac{du}{dx} \right)^2 \right]^{\frac{1}{2}} dx. \quad (6)$$

Using x as the integration variable and the Cartesian coordinate system of Figure 2 gives that the integration limits are $-\frac{L}{2}$ and $\frac{L}{2}$, where L is the distance between the beam supports. The curvature is related to the deflection by

$$\kappa = \frac{d^2u}{dx^2} \left[1 + \left(\frac{du}{dx} \right)^2 \right]^{-\frac{3}{2}}. \quad (7)$$

Hence, Eq. 1 can be written as

$$P = \frac{1}{2}EI \int_{-L/2}^{L/2} \left(\frac{d^2u}{dx^2} \right)^2 \left[1 + \left(\frac{du}{dx} \right)^2 \right]^{-\frac{5}{2}} dx \quad (8)$$

$$+ \tilde{m}g \int_{-L/2}^{L/2} u \left[1 + \left(\frac{du}{dx} \right)^2 \right]^{\frac{1}{2}} dx.$$

Similar to in Komperød (2018), the unknown u that minimizes the total potential energy can not be calculated directly. Instead u will be calculated at discrete, Chebyshev-distributed points which can later be interpolated. The points will be Chebyshev-distributed over the left half of the beam, and Chebyshev-distributed over the right half of the beam, rather than being Chebyshev-distributed over the entire beam. This is done to avoid that the discontinuity of the third derivative at the center point disturbs the excellent convergence properties of Chebyshev series for functions in C^∞ .

Let \mathbf{u} be a vector with the N discrete points of u , where N is an even number. It is convenient to introduce

$$\bar{N} = N/2, \quad (9)$$

$$\bar{\mathbf{u}}_L = \mathbf{u}[1 : \bar{N}], \quad (10)$$

$$\bar{\mathbf{u}}_R = \mathbf{u}[\bar{N} + 1 : N]. \quad (11)$$

Hence, $\bar{\mathbf{u}}_L$ is the discrete points of u on the left half of the beam, and $\bar{\mathbf{u}}_R$ is the discrete points of the right half. Note that the last element of $\bar{\mathbf{u}}_L$ must be equal to the first element of $\bar{\mathbf{u}}_R$, because they both equals the piston deflection, u_p . This will later be enforce through a constraint.

The first and second derivative of u at the Chebyshev nodes can be calculated using the Chebyshev first-derivative and second-derivative matrices \mathbf{D}_1 and \mathbf{D}_2 . Once the total potential energy per unit length is calculated at the Chebyshev nodes, the total potential energy

over the beam can be calculated using Clenshaw-Curtis quadrature. Both the Chebyshev derivative matrices and Clenshaw-Curtis quadrature are explained in Reid (2014). Hence, the total potential energy over the beam is

$$\begin{aligned}
 P = & \frac{EIL}{8} \left((\mathbf{D}_2 \bar{\mathbf{u}}_L)^2 \cdot [\mathbf{1}_{\bar{N} \times 1} + (\mathbf{D}_1 \bar{\mathbf{u}}_L)^2]^{-\frac{5}{2}} \right) \bullet \bar{\mathbf{c}} \quad (12) \\
 & + \frac{EIL}{8} \left((\mathbf{D}_2 \bar{\mathbf{u}}_R)^2 \cdot [\mathbf{1}_{\bar{N} \times 1} + (\mathbf{D}_1 \bar{\mathbf{u}}_R)^2]^{-\frac{5}{2}} \right) \bullet \bar{\mathbf{c}} \\
 & + \frac{\tilde{m}gL}{4} \left(\bar{\mathbf{u}}_L \cdot [\mathbf{1}_{\bar{N} \times 1} + (\mathbf{D}_1 \bar{\mathbf{u}}_L)^2]^{\frac{1}{2}} \right) \bullet \bar{\mathbf{c}} \\
 & + \frac{\tilde{m}gL}{4} \left(\bar{\mathbf{u}}_R \cdot [\mathbf{1}_{\bar{N} \times 1} + (\mathbf{D}_1 \bar{\mathbf{u}}_R)^2]^{\frac{1}{2}} \right) \bullet \bar{\mathbf{c}}.
 \end{aligned}$$

In Eq. 12 all vectors are of dimension $\bar{N} \times 1$ and all matrices are of dimension $\bar{N} \times \bar{N}$. The vector $\bar{\mathbf{c}}$ is the vector of the Clenshaw-Curtis quadrature weights for the standard integration interval $[-1, 1]$. The vector $\mathbf{1}$ is the vector which all elements are 1 and its dimension is given by the subscript. On the right side of Eq. 12 the first row is the strain energy of the left half of the beam, the second row is the strain energy of the right half of the beam, the third row is the gravitational energy of the left half of the beam, and the fourth row is the gravitational energy of the right half of the beam.

5.2 Constraints

The overall problem to be solved is to minimize Eq. 12 subject to the constraints of Eqs. 2 - 5. This section considers how to implement these constraints, while the subsequent sections present how to solve the optimization problem.

The constraints of Eq. 2 and Eq. 4 are straight forward to implement; they become

$$\mathbf{u}[0] = 0, \quad (13)$$

$$\mathbf{u}[\bar{N}] = 0. \quad (14)$$

Because two vector elements of \mathbf{u} represent the deflection at center, i.e. at $x = 0$, the constraint of Eq. 3 can be implemented in three different ways: (i) $\mathbf{u}[\bar{N}] = u_p$ and $\mathbf{u}[\bar{N} + 1] = u_p$, (ii) $\mathbf{u}[\bar{N}] = u_p$ and $\mathbf{u}[\bar{N} + 1] = \mathbf{u}[\bar{N}]$, or (iii) $\mathbf{u}[\bar{N}] = \mathbf{u}[\bar{N} + 1]$ and $\mathbf{u}[\bar{N} + 1] = u_p$. All choices give the same information, and hence the same results. However, the latter two choices give more straight forward interpretation of the Lagrange multipliers. The second choice is used in this paper. The constraints are then

$$\mathbf{u}[\bar{N}] = u_p, \quad (15)$$

$$\mathbf{u}[\bar{N}] - \mathbf{u}[\bar{N} + 1] = 0. \quad (16)$$

As explained in Komperød (2018), the constraint of Eq. 5 will be obeyed if and only if u and its first- and second derivatives are continuous at $x = 0$. u is continuous by Eq. 16. The first- and second derivatives are continuous by enforcing

$$\mathbf{D}_1[\bar{N}, :] \bar{\mathbf{u}}_L - \mathbf{D}_1[1, :] \bar{\mathbf{u}}_R = 0, \quad (17)$$

$$\mathbf{D}_2[\bar{N}, :] \bar{\mathbf{u}}_L - \mathbf{D}_2[1, :] \bar{\mathbf{u}}_R = 0. \quad (18)$$

The constraints are implemented in the order Eq. 13, Eq. 15, Eq. 16, Eq. 14, Eq. 17, and Eq. 18. Although the total potential energy, i.e. Eq. 12, is nonlinear, all constraints are linear. Hence, the constraints can be written in matrix form as

$$\mathbf{A}\mathbf{u} = u_p \mathbf{b}, \quad (19)$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{i}_1^T \\ \mathbf{i}_{\bar{N}}^T \\ (\mathbf{i}_{\bar{N}} - \mathbf{i}_{\bar{N}+1})^T \\ \mathbf{i}_{\bar{N}}^T \\ \mathbf{D}_1[\bar{N}, :], -\mathbf{D}_1[1, :] \\ \mathbf{D}_2[\bar{N}, :], -\mathbf{D}_2[1, :] \end{bmatrix} \in \mathbb{R}^{6 \times N}, \quad (20)$$

$$\mathbf{b} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \in \mathbb{R}^{6 \times 1}. \quad (21)$$

In Eq. 20, \mathbf{i}_j is the j th column of the $N \times N$ identity matrix. Note that in the third row of the matrix \mathbf{A} two vectors of dimension N are subtracted, while in the fifth and sixth rows two vectors of dimension \bar{N} are concatenated into one vector of dimension N .

5.3 Transforming to an Unconstrained Optimization Problem

The vector \mathbf{u}^* that minimizes the constrained optimization problem of Eq. 12 and Eq. 19 must obey Eq. 19. The matrix \mathbf{A} of Eq. 20 has full rank, i.e. rank 6, otherwise the constraints would have been redundant or contradictory. Hence, \mathbf{A} has a nullspace of dimension $N - 6$. That is, there are $N - 6$ independent vectors that solves the homogeneous equation

$$\mathbf{A}\mathbf{u} = \mathbf{0}_{6 \times 1}, \quad (22)$$

where $\mathbf{0}$ is the vector which all elements are zero and its dimension is given by the subscript.

Let $\mathbf{U} \in \mathbb{R}^{N \times (N-6)}$ be a matrix which columns are an orthonormal basis of the nullspace of \mathbf{A} . Then the homogeneous equation can be written

$$\mathbf{A}\mathbf{U}\mathbf{w} = \mathbf{0}_{6 \times 1}, \quad (23)$$

where $\mathbf{w} \in \mathbb{R}^{(N-6) \times 1}$ is an arbitrary vector. Further, let \mathbf{u}_s be a particular solution of the equation

$$\mathbf{A}\mathbf{u} = \mathbf{b}. \quad (24)$$

Then, multiplying Eq. 24 by u_p and adding Eq. 23 gives

$$\mathbf{A}(u_p \mathbf{u}_s + \mathbf{U}\mathbf{w}) = u_p \mathbf{b}. \quad (25)$$

Hence, the deflection vector \mathbf{u} given by

$$\mathbf{u} = u_p \mathbf{u}_s + \mathbf{U}\mathbf{w} \quad (26)$$

obeys all constraints for any \mathbf{w} . In other words, the $N - 6$ elements of \mathbf{w} are the $N - 6$ degrees of freedom for minimizing the total potential energy, P , given by Eq. 12. Expressed mathematically, the constrained optimization problem is transformed into the unconstrained optimization problem

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} P(\mathbf{u}(\mathbf{w})). \quad (27)$$

5.4 Initial Values for Iterative Optimization

Because $\mathbf{U}\mathbf{w}$, for any \mathbf{w} , added to a particular solution \mathbf{u}_s gives a new particular solution, there are infinite number of particular solutions that satisfy Eq. 24. An intuitive approach is to calculate a particular solution from Eq. 24 using linear algebra. However, many optimization methods, including Newton's method and Quasi-Newton methods, depend on sufficiently good initial values in order to converge to the global minimum. A solution of Eq. 24 found from linear algebra may be a very poor initial value for the optimization method.

A better method for finding a particular solution that is also a good initial value for the optimization method is to use the analytical solution of the beam equation. Although the analytical solution is valid only for infinitesimal deflections, it is still a decent initial value. For most real-life bending stiffness tests of cables the effect of gravity is small. Hence, the gravity term can be excluded from the analytical solution when calculating the initial values. The analytical solution then becomes

$$u = \begin{cases} -\frac{4}{L^3}x^3 - \frac{6}{L^2}x^2 + 1 & , \quad x < 0 \\ \frac{4}{L^3}x^3 - \frac{6}{L^2}x^2 + 1 & , \quad x \geq 0 \end{cases} \quad (28)$$

It is trivial to verify that Eq. 28 obeys the constraints of Eqs. 2 - 5 for $u_p = 1$. Hence, the discrete-point deflection

vector \mathbf{u} calculated by evaluating Eq. 28 at the Chebyshev points is a particular solution of Eq. 24 and a good initial value for the optimization method.

A most favorable property of Eq. 28 is that it does not depend on the deflection force, F , or the beam's bending stiffness, EI. This is because the force and the bending stiffness cancel in the calculation of the deflection, u . Actually, calculation of the deflection depends only on the length between the beam supports, L , which is trivial to measure prior to the bending stiffness tests.

If gravity is to be taken into consideration, the particular solution will be somewhat more complex and it will depend on the beam's specific weight, \bar{m} , and its bending stiffness, EI. In other words, an initial guess of the bending stiffness will then be necessary. Please note that even if gravity is neglected when calculating the particular solution, gravity will still be included during the optimization, and hence be included in the final calculation of the deflection.

5.5 Iterative Optimization

The approach presented in Section 5.4 provides good initial values, which significantly ease the iterative optimization process. The results presented in this paper will in future work be extended with (i) nonlinear materials and (ii) shear forces of helical cable elements. Because these extensions may significantly change the characteristics of the optimization problem, the choice of optimization method has not yet been settled.

Also the choice of number of nodes, N , will need further attention. Too few nodes will not provide sufficient flexibility to represent the beam's or cable's true behavior. However, a large number of nodes increase the computation time and may increase the risk that the optimization algorithm get trapped in a local minimum or fails due to accumulations of arithmetic errors.

The *preliminary* conclusions based on the work done so far are:

1. Good choices of N seems to be in the range 14 to 20, i.e. \bar{N} in the range 7 to 10. Higher values of N give no or insignificant reduction of the total potential energy, P . Lower values of N gives somewhat increased P , which indicates insufficient flexibility to resemble the beam's true deflection profile. The proposed range of N gives rapid convergence and no issues with local minima or arithmetic errors have been observed.
2. Newton's method converges very rapidly and no failures have been observed in the proposed range of N . The method converges also for $N = 50$, while it fails for $N = 100$. The intermediate values have not been tested. Calculation of the elements of the Hessian matrix and the gradient vector are done numerically using finite differences, because analytical calculations will not be an option after implementing non-

linear materials and shear forces. After implementing these improvements, calculation of the Hessian matrix may be expensive in terms of CPU load.

3. The gradient decent method performs poorly; it converges very slowly for small values of N , and it fails to reach the global minimum for larger N .
4. The Broyden–Fletcher–Goldfarb–Shanno (BFGS) method can roughly be considered a hybrid between the gradient decent method and Newton’s method; it starts off identically to the gradient decent method and then uses the information gained during the iterations to build an estimate of the Hessian matrix, which makes it gradually resemble the properties of Newton’s method.

Similar to Newton’s method, the BFGS method performs very well within the proposed range of N . Actually, the BFGS method converges to the global minimum for $N = 100$, where the two former methods both fail. The BFGS method fails for $N = 200$, while the intermediate values have not been tested.

The BFGS method avoids the expensive computation of the Hessian matrix. However, the method requires a larger number of iterations than Newton’s method, and may require several calculations of the total potential energy, P , to get the correct step length within each iteration. Hence, further study must be performed in order to conclude whether Newton’s method or the BFGS method is the most efficient for this particular optimization problem.

6 Calculating Deflection Force

In the physical bending stiffness rig the piston’s deflection, u_p , and deflection force, F_p , are measured. Hence, comparisons between the physical tests and the cable model will focus on these variables. In the model the deflection is given as a constraint, while the force is to be calculated. The force is given as the derivative of the total potential energy w.r.t. the deflection, i.e.

$$F_p = \frac{\partial P}{\partial u_p}. \quad (29)$$

Based on Eq. 29 the force can be calculated using finite differences or the Chebyshev first-derivative matrix \mathbf{D}_1 .

The motivation behind the work presented in this paper is to improve the calculations derived in Komperød (2018), because the latter depend on simplifications which are valid only for infinitesimal deflections. It is then of interest to compare the two calculation methods to answer whether they differ significantly or not. Figure 3 presents the calculated deflection force, F_p , as function of the deflection, u_p , for the two approaches. The calculations are done for a steel tube with bending stiffness 124.1 kNm^2

and specific mass 8.257 kg/m . The distance between the supports are 3.000 m . The interval $u_p \in [-0.4, 0.4]$ is the range of the piston on the bending stiffness rig. The BFGS method with $N = 20$ was used.

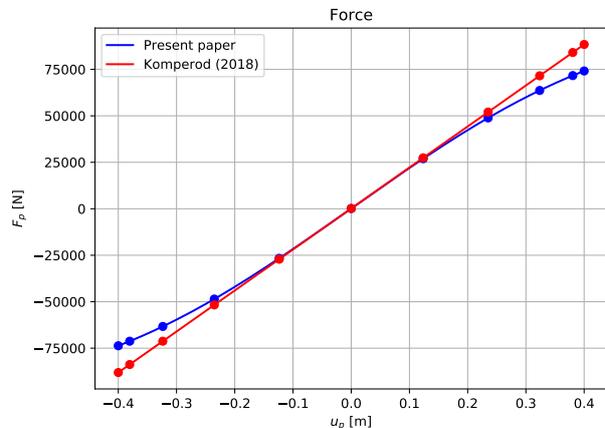


Figure 3. Deflection force calculated from the present paper and from Komperød (2018). The dots represent Chebyshev-distributed points and the lines represent interpolations between these points. For the present paper the force is calculated using Eq. 29 and the Chebyshev first-derivative matrix \mathbf{D}_1 . For Komperød (2018) the force is calculated from Lagrange multipliers as explained in that paper.

As seen from Figure 3, the two approaches give near identical results for small deflections, i.e. deflections close to zero, while the model of Komperød (2018) becomes increasingly inaccurate for larger deflections. Figure 4 shows the relative error of Komperød (2018) compared to the calculations derived in the present paper. As seen from the figure, the relative error resembles a parabola quite well. Hence, the relative error increases approximately by the deflection squared. The reason behind this relationship has not been studied. The largest relative error shown in the figure is almost 20%.

Based on Figures 3 and 4, it is concluded that the approach of Komperød (2018) gives high accuracy for small deflections, but should be avoided for larger deflections. This result was expected because Komperød (2018) is based on the assumption of infinitesimal deflections. Please note that the deflection range for which the latter approach will give high accuracy will depend on the length between the supports, L . Therefore, the error indicated by Figures 3 and 4 should not be generalized to arbitrary values of L .

7 On Lagrange Multipliers

In the present paper, the author chooses to use the constraints to reduce the number of variables and hence improve the convergence of the optimization problem. In Komperød (2018) the constraints are handled using Lagrange multipliers, which give additional information as the majority of the Lagrange multipliers have physical in-

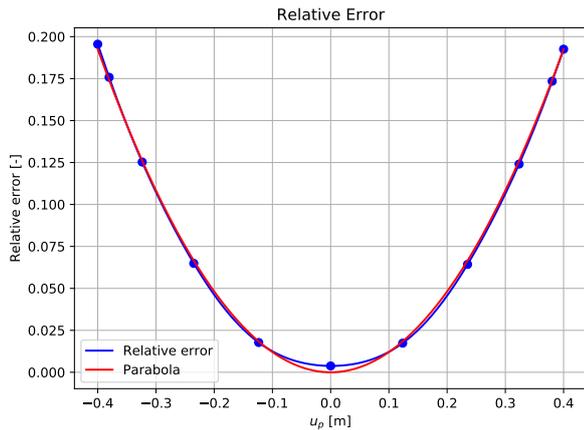


Figure 4. Relative error of Komperød (2018) compared to the calculations derived in the present paper. The figure also shows a parabola on the form au_p^2 , where a is a parameter set so that the parabola and the relative error graph match at $u_p = 0.4$.

terpretations.

Because the Lagrange multipliers are the sensitivities of the optimized value, in this case the minimum total potential energy, w.r.t. infinitesimal perturbations of the constraints, the Lagrange multipliers can easily be found from the approached derived in the present paper at the expense of additional iterations. That is, infinitesimal perturbations are added to the right hand side of Eq. 19 and the total potential energy is then re-optimized. The author has tested this method and found that it gives correct results. Further details and demonstrations of the method are beyond the scope of this paper.

8 Further Work

The work presented in this paper is the second milestone toward modeling subsea power cables and umbilicals during bending stiffness tests. The remaining milestones are (i) modeling nonlinear materials and (ii) modeling shear forces between the cable elements. The final model will be validated against the results from the physical bending stiffness tests. Once the model is completed and validated, it can be used for other calculations and simulations only by changing its loads and the boundary conditions.

This modeling work is part of Nexans Norway AS' continuous work on improving modeling and physical testing of subsea power cables and umbilicals. This work enables the company to install cables and umbilicals in ever deeper waters, in lower temperature areas, and under harsher weather conditions.

9 Conclusions

This paper improves the work of Komperød (2018) to correctly handle large deflections. Comparing the previous model and the improved model shows that the relative error of the previous model increases roughly by the deflec-

tion squared. For the example presented in this paper the largest relative difference is near 20%.

From the author's point of view, the only significant disadvantage of the improved model is that it depends on iterative optimization, while the previous model is solved explicitly. However, as shown in this paper, the analytical solution of the beam equation can be used as initial values for the iterative optimization. Using these initial values, both Newton's method and the BFGS method solve the optimization problem quickly, while the gradient decent method is very slow and may fail to converge to the correct values.

The author concludes that the modeling work presented in this paper is successful. This opens up for the next milestone, which is to include nonlinear materials in the model. The final goal of the overall R&D activity is to develop models which are feasible compromises between accuracy, complexity, and computation time, and which are validated against physical bending stiffness tests.

References

- M. M. Dhaigude and K. I. Ekeberg. Validation of the loxodromic bending assumption using high-quality stress measurements - high tension case. In *Proceedings of the Twenty-sixth International Ocean and Polar Engineering Conference (ISOPE 2016)*, Rhodes, Greece, 2016.
- K. I. Ekeberg and M. M. Dhaigude. Validation of the loxodromic bending assumption using high-quality stress measurements. In *Proceedings of the Twenty-sixth International Ocean and Polar Engineering Conference (ISOPE 2016)*, Rhodes, Greece, 2016.
- L. Jordal, R. Slora, E. Vermeer, and M. Komperød. A novel bending stiffness rig for identification of subsea cables' and umbilicals' sensitivity to temperature under sinusoidal curvature oscillations. In *Proceedings of the Twenty-seventh International Ocean and Polar Engineering Conference (ISOPE 2017)*, San Francisco, California, USA, 2017.
- E. Kebabzde. *Theoretical modelling of unbonded flexible pipe cross-sections*. PhD thesis, South Bank University, 2000.
- M. Komperød. Deriving the beam equation using the minimum total potential energy principle and solving the equation numerically. In *Proceedings of the 59th International Conference of Scandinavian Simulation Society (SIMS 2018)*, Oslo, Norway, 2018.
- M. Komperød, J. I. Juvik, G. Evenset, R. Slora, and L. Jordal. Large-scale tests for identifying the nonlinear, temperature-sensitive, and frequency-sensitive bending stiffness of the NordLink cable. In *Proceedings of the 36th International Conference on Ocean, Offshore and Arctic Engineering (OMAE 2017)*, Trondheim, Norway, 2017.
- M. Lutchansky. Axial stress in armor wires of bent submarine cables. *Journal of Engineering Industry*, 91(3):687 – 693, 1969.

- P. Maioli. Bending stiffness of submarine cables. In *Proceedings of the 9th International Conference on Insulated Power Cables (Jicable '15)*, 2015.
- H. Reid. Massachusetts Institute of Technology - Lecture notes 18.330 Introduction to numerical analysis, 2014.
- J. Tarnowski. Improved method of determining bending stiffness of underground cables. In *Proceedings of the 9th International Conference on Insulated Power Cables (Jicable '15)*, 2015.

Modeling and Simulation of Triclosan Kinetics and Distribution in Humans Using PBPK Model

Vincenza Cascella¹, Monica Andreassen², Trine Husøy², Hubert Dirven², Bernt Lie¹

¹University of South-Eastern Norway, Bernt.Lie@usn.no;

²Department of Toxicology and Risk Assessment; Norwegian Institute of Public Health

Abstract

Triclosan (TCS) has been used as an antibacterial additive in several personal care products (PCPs) for more than 40 years. In animals studies, it is observed that TCS in rats has adverse effects on the endocrine function and thyroid hormone homeostasis, and intensify antibiotic resistance. In human studies, significant levels of TCS are detected in human plasma and breast milk. In this study, a Physiologically Based Pharmacokinetic or Toxicokinetic (PBPK/TK) model is developed to describe the concentration of triclosan in human organs after exposure through the skin (dermal) or orally. Several studies have been conducted on toxicokinetics of TCS, but there is a lack of information on parameters for PBPK models. In this paper, focus is on finding parameters for TCS to be used in the PBPK model. In a first case, the PBPK model was based on data for TCS and the structurally related chemical Bisphenol A (BPA), in a second case, partition coefficients were found using Poulin's method. The simulations were carried out using MATLAB, and the results for the two cases are compared.

Keywords: Triclosan, PBPK, Model parameters

1 Introduction

In everyday life people are surrounded by products which contain chemicals that might have negative effects on the human body. Many chemical compounds can be intentionally or unintentionally inhaled, ingested, or dermally absorbed. Various risks that could incur upon exposure are investigated (Fang et al., 2010). In animals studies, it is observed that *triclosan* (TCS) in rats has adverse effects on the endocrine function and thyroid hormone homeostasis, and intensify antibiotic resistance. The widespread application of TCS in numerous personal care and household products such as disinfectants, soaps, detergents, toothpastes, mouthwashes, fabrics, deodorants, shampoo, plastic additives in addition to several veterinary and industrial products (Dann and Hontela, 2011), has led to concerns regarding human health risks (Vingskes and Spann, 2018). TCS, an organic compound categorized as a polychloro phenoxyphenol, is a chlorinated aromatic compound that has functional groups representative of both ethers and phenol. Invented more than 40 years ago, and used increasingly over the past 25 years, regulations were even-

tually invoked in 2010 by the European Community Cosmetic Directive on the use of TCS, limiting the concentration of TCS at 0.1-0.3% of product weight in personal care products. This percentage is considered safe in everyday products defined as toothpaste, hand soaps, shower gels, and deodorants (Dann and Hontela, 2011). The effects of TCS as an environmental chemical is also of concern, but this is not considered further here.

Physiologically based pharmacokinetic (PBPK) modeling and simulation can be used to predict the pharmacokinetic behavior of drugs in humans using clinical data (Zhao et al., 2011). The purpose for developing a PBPK for TCS is the wide exposure to the substance and lack of information about its toxicity on long-term use. Another reason, not to be underestimated, is the reduction of experiments on animals, for ethical reasons. PBPK modeling is a quantitative attempt to model the drug distribution in the body in compartments and to predict tissue concentration within specific organs (liver, kidney), tissue (skin, muscle), and body fluids (blood, Cerebrospinal fluid (CSF) and urine).

Here, we report the absorption, distribution, metabolism and excretion (ADME) of TCS after exposure via the dermal and/or oral route. Section 2 gives an overview of the system, the dynamic model is presented in Section 3, simulation results are presented and discussed in Section 4, and some conclusions are drawn in Section 5.

2 System Overview

2.1 Organs and Topology

Organs, blood flow, topology, and intake of TCS is illustrated in Figure 1.

The model equations follow the principles of mass transport, fluid dynamics, and biochemistry in order to simulate the evolution of a substance in the body (Campbell et al., 2012). PBPK models can predict the timescales and concentration in which TCS appear and disappear from various organs of the body, for setting dosing guidelines and risk assessment. Each compartment is described by capillary blood flow and capillary-tissue volume, as well as partitioning between capillary-tissue.

Each tissue is defined with assumptions of either perfusion-rate-limited or permeability-rate-limited.

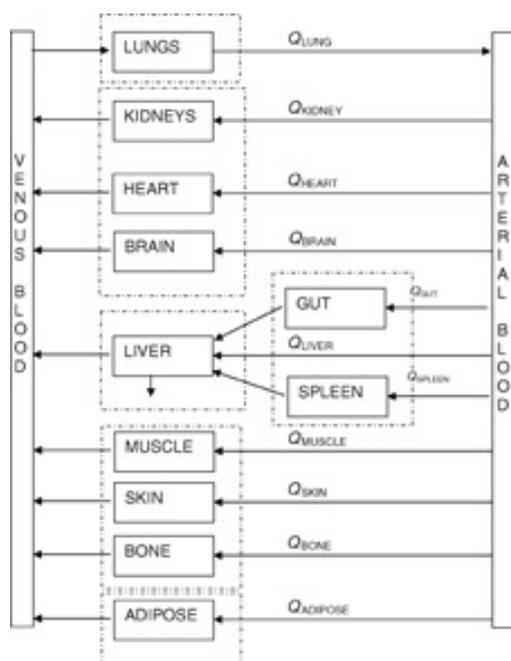


Figure 1. Flowchart of organ topology.

Perfusion-rate-limited kinetics tends to exist for small lipophilic molecules where the blood flow to the tissue proved to be the limiting process of the absorption. Permeability-rate-limited kinetics occurs for more hydrophilic and larger molecules where the permeability across the cell membrane becomes the limiting process of absorption. (Zhuang and Lu, 2016).

2.2 Triclosan and Data

Triclosan is used as an anti-bacterial component in toothpaste, deodorant, etc. The anti-bacterial operation of Triclosan is as follows¹. The cell membrane of bacteria is held together by a lipid bi-layer. In order for bacteria to live and grow, this lipid bi-layer must be able to grow. This growth takes place when a regulatory gene, named FabI, encodes an enzyme called enoyl-acyl carrier protein reductase (ENR) such that lipids can attach to the active site of ENR and then be inserted into the bi-layer. If Triclosan is present, Triclosan is instead adsorbed to the active site of ENR and blocks the uptake of lipids. This effectively blocks the growth of bacteria cells.

Triclosan has chemical formulae $C_{12}H_7Cl_3O_2$ (CAS² number 3380-34-5), with chemical structure as indicated in Fig. 2.

Triclosan is a colorless to off-white crystalline powder with a slightly aromatic odor, and is tasteless. Triclosan has low solubility in water (0.1 g/L) but much higher solubility in lipids (fatty acids). Physical-chemistry data for TCS are limited, and it may be necessary instead to use

¹<https://illum.in.usc.edu/68/what-makes-antibacterial-soap-antibacterial/>

²Chemical Abstracts Service)

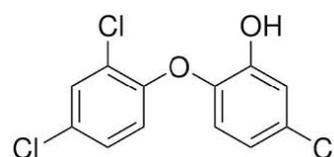


Figure 2. Molecular structure of triclosan (TCS). From (Fang et al., 2010).

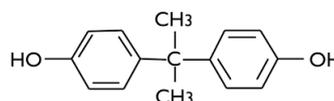


Figure 3. Molecular structure of bisphenol A (BPA).

data for structurally related molecules.

Bisphenol A (BPA; 2,2-bis(4-hydroxyphenyl)propane) is a chemical compound found in many products (Konieczna et al., 2015), hence important data are readily available. BPA is an organic compound composed of two phenol rings connected by a methyl bridge, with two methyl functional groups attached to the bridge, Fig. 3.

BPA is structurally related molecule to TCS, Figures 2 and 3. Although they have differences, important properties such as the octanol/water coefficient are in the same range, Table 1.

In addition, major metabolism through sulphatation and glucuronidation is the same. It can be assumed that TCS and BPA have a comparable tendency of to be solubilized in tissue.

2.3 Triclosan Metabolism

Triclosan metabolizes to form conjugates *triclosan sulfate* and *triclosan glucuronide*, Fig. 4.

The operation of triclosan metabolism is as follows. A family of enzymes known as SULT (sulfotransferase) catalyze the transfer of a sulfonate group ($-SO_3^-$) from PAPS (3'-phosphoadenosine-5'-phosphosulfate) to triclosan, forming the conjugate *triclosan sulfate* with the release of PAP (3-phosphoadenosine-5-phosphate). Denoting triclosan by T and triclosan sulfate by T_S we can write the overall reaction as



Similarly, an enzyme known as UGT (Uridine 5'-diphospho-glucuronosyl transferase = UDP-glucuronosyl transferase) catalyzes the transfer of the glucuronic acid

Table 1. Molecular mass (M_w) and *n*-octanol-water solubility parameter $\log K_{ow}$ for TCS and BPA. Taken from <https://chem.nlm.nih.gov/chemidplus/rn/3380-34-5> (TCS) and <https://chem.nlm.nih.gov/chemidplus/name/bisphenol> (BPA).

	TCS	BPA
M_w	289.53 g/mol	228.9 g/mol
$\log K_{ow}$	4.76	3.32

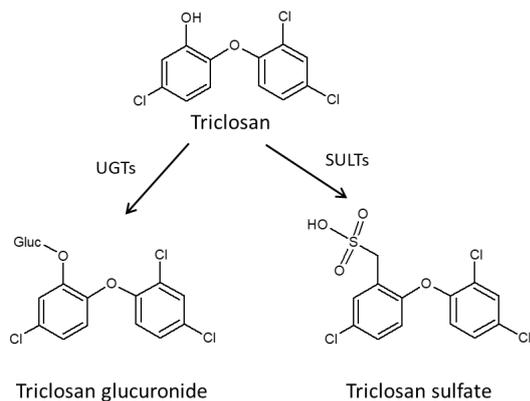


Figure 4. Metabolism of triclosan into glucuronides and sulfates. After (Fang et al., 2010).

component of UDP-glucuronic acid to triclosan, forming the conjugate triclosan glucuronide as well as a residual of UDP-glucuronic acid. Denoting UDP-glucuronic acid by G, the conjugate *triclosan glucuronide* by T_G, and the residual of UDP-glucuronic acid by \bar{G} , we can write the overall reaction as



In both cases, triclosan (T, substrate) is adsorbed on the enzyme surface and covers/is inserted into the active site of the enzyme. Provided that triclosan has been adsorbed into the active site, we assume that a co-substrate (PAPS or G) also is adsorbed at the active site, forming a substrate/co-substrate complex. Next, a reaction takes place in this complex formed at the active site, and conjugated triclosan (T_S or T_G) is formed together with a co-product (PAP or \bar{G}) and these then desorb from the active site. Under the conditions of constant enzyme concentration and constant co-substrate concentration, the overall reaction rate referring to triclosan can be written as:

$$\hat{r}_j = \hat{r}_j^{\max} \frac{c_T}{c_T + K_T^j} \quad (3)$$

where $[\hat{r}_j]$ is given in mole/(time and protein mass), and $j \in \{S, G\}$. The individual reaction rates are thus:

$$\hat{r}_S = \hat{r}_S^{\max} \frac{c_T}{c_T + K_T^S} \quad (4)$$

$$\hat{r}_G = \hat{r}_G^{\max} \frac{c_T}{c_T + K_T^G}. \quad (5)$$

The conjugation of triclosan has been demonstrated *in vitro* in the human liver microsomes or cytosol (Wang et al., 2004), and the parameters taken from this study are shown in Table 2.

From Table 2 it should be observed that the molar rates of generation are given as

$$\dot{n}_j^g = \hat{r}_j \cdot m_{p,j} \quad (6)$$

Table 2. Michaelis-Menten reaction parameters for TCS metabolism in human liver (Wang et al., 2004).

	K_T^j	$\frac{\mu\text{mol}}{\text{L}}$	\hat{r}_j^{\max}	$\frac{\text{pmol}}{\text{min mg Protein}}$
Sulfonation	8.5 ± 3.2		95.9 ± 28.4	
Glucuronidation	107 ± 22.2		739 ± 163	

where $m_{p,j}$ is the relevant mass of protein in the liver. It should be noted that the available protein mass differs in sulfonation and glucuronidation: (Wang et al., 2004) indicate that sulfonation takes place in the human liver cytosol while glucuronidation takes place in the liver microsomes. Ideally, we would like to relate $m_{p,j}$ to the mass m_h of the liver (hepatic mass) as an organ instead, in other words to find ξ_j satisfying

$$m_{p,j} = \xi_j m_h. \quad (7)$$

In (Cubitt et al., 2009), the hepatic cytosolic scaling factor was found in the range $\xi_S^h \in [45, 134]$ mg protein in liver *cytosol* per g of liver, with a mean weighted value $\xi_S^h = 80.7 \times 10^{-3}$. They also report a typical fraction protein in intestine cytosol per g intestine to be $\xi_S^i = 18 \times 10^{-3}$. Observe that the rate numbers given in Table 2 do not necessarily carry over from liver to intestine.

In (Barter et al., 2007), experimental results are cited in the range $\xi_G^h \in [19, 77]$ mg protein in liver *microsomes* per g of liver. The value probably varies among individuals, with age, and by gender. A possible useful value may be $\xi_G^h = 30 \times 10^{-3}$.

2.4 Partition Coefficients

Most organs include blood flow with TCS through capillaries, and diffusion of TCS from the capillaries into the surrounding tissue. With capillary concentration $c_c(t)$ of the drug, the tissue concentration is $c_t'(t, x)$ — in a simplified description the tissue concentration is $c_t^*(t)$ at the interface, and $c_t(t)$ in the bulk of the tissue; $c_t = \lambda c_t^*$ where λ is the *partition coefficient* (PC). The driving force for diffusion is $\Delta c = c_c - c_t^*$. For organs where the tissue only interacts with the capillary, at equilibrium, $\Delta c_{\text{eq}} = 0$, thus $c_{t,\text{eq}} = \lambda c_{t,\text{eq}}^* = \lambda c_{c,\text{eq}}$. Accurate partition coefficients is important for getting realistic results.

Finding PCs from *in vivo* experiments takes time and is costly. Some progress has been made in developing correlation formulas for PCs, where the PC is correlated with knowledge of tissue and blood (capillary) composition in terms of neutrolipids, phospholipid, and water fractions. Results in general demonstrate that the correlation formulas can provide a good approximation for partition coefficients (Kuepfer et al., 2016). One such correlation expression is the algorithm of Poulin and Krishnan. Let S be the solubility in g solute per 100g of solvent. Specifically, we consider the solubility of drug (TCS) in water S_w and drug in *n*-octanol S_o . The solubilities are taken to be proportional,

$$S_o = K_{ow} S_w. \quad (8)$$

Table 3. Partition coefficient λ calculated from (Poulin and Krishnan, 1995a,b).

Organ	λ BPA	λ TCS
Fat, fa	0.7	0.2536
Bone, bo	0.5	0.7392
Brain, br	4.4	1.2941
Kidney, ki	4.4	1.2774
Liver, li	5.7	1.2409
Lung, lu	5.5	1.3099
Muscle, mu	0.8	1.2446
Skin, ski	5.7	1.1174

Typically, $\log K_{ow}$ is given — here, we use $\log K_{ow} \approx 4.8$ (Fang et al., 2010). With $S_w = 10^{-3}$ at 25°C , S_o is thus known.

Next, we assume that the drug is associated with neutral lipids (N), phospholipids (P), and water (W) in both tissue (t) and capillaries (blood, b). The fraction of these are denoted χ_j where $j \in \{N, P, W\}$; values for these fractions are found in (Ye et al., 2016) for each organ. By observing that at equilibrium,

$$\lambda = \frac{c_{t,eq}}{c_{c,eq}} = \frac{c_{t,eq}}{c_{b,eq}}, \quad (9)$$

an estimate of the PC can, e.g., be given as described in (Poulin and Krishnan, 1995a,b):

$$\lambda = \frac{\chi_{N,t}S_o + \chi_{P,t}(0.7S_w + 0.3S_o) + \chi_{W,t}S_w}{\chi_{N,b}S_o + \chi_{P,b}(0.7S_w + 0.3S_o) + \chi_{W,b}S_w}. \quad (10)$$

Here, it has been assumed that neutral lipids are insoluble in water, and that phospholipids are 70% soluble in water and 30% soluble in *n*-octanol.

Table 3 shows the PCs for BPA, compared to the PCs for TCS as found from Eq. 10.

3 Dynamic Model

3.1 General Balance

As pharmacokinetic models are used for assessing concentration of chemicals in different organism/tissue of a body, the mass balance equations are transferred into differential equations in the form of number of moles. The general species balance in molar form is

$$\frac{dn_j}{dt} = \dot{n}_j^i - \dot{n}_j^e + \dot{n}_j^g, \quad (11)$$

where:

n_j : number of moles of species j accumulated in the volume,

\dot{n}_j^i : influent in moles per unit time of species j ,

\dot{n}_j^e : effluent flow in moles per unit time of species j ,

\dot{n}_j^g : generated number of moles per time unit of species j .

3.2 Compartments

Organs are often represented by 2-compartment models consisting of capillary volume and tissue volume. Assuming well mixed volumes and not accumulation of blood, without metabolism, the two volumes can be described by

$$\frac{dc_c}{dt} = \frac{\dot{V}}{V_c} (c_c^i - c_c) - \frac{\Theta_{c2t}}{V_c} (c_c - c_t^*) \quad (12)$$

$$\frac{dc_t}{dt} = \frac{\Theta_{c2t}}{V_t} (c_c - c_t^*) \quad (13)$$

where $c_t = \lambda c_t^*$. Assuming that Θ_{c2t} is “small” (permeation limited),

$$c_c \approx c_c^i, \quad (14)$$

and the model reduces to

$$\frac{dc_t}{dt} = \frac{\Theta_{c2t}}{V_t} \left(c_c^i - \frac{1}{\lambda} c_t \right). \quad (15)$$

On the other hand, if Θ_{c2t} is “large” (perfusion limited), we will have a quasi steady state where

$$c_c \approx c_t^* = \frac{1}{\lambda} c_t, \quad (16)$$

thus

$$\frac{dc_c}{dt} = \frac{\dot{V}}{V_c} (c_c^i - c_c) \quad (17)$$

and $c_t = \lambda c_c$.

Four organs need special treatment: metabolism takes place in the skin (dermal uptake) and in the liver; for these, metabolism reactions must be included. Furthermore, the GI tract/the intestine (oral intake) or the skin (dermal intake) need to include a transport model for the drug. Finally, the kidney has an additional “leakage” of drug from the tissue to urine.

3.3 Non-metabolic Organs

Whether the system is permeation limited or perfusion limited (or a mixture), the models in Section 3.2 need to be applied to each of free triclosan, triclosan sulfate, and triclosan glucuronide.

3.4 Metabolic Organs

Free triclosan can metabolize (react) into its conjugates; the balances are:

$$\frac{dn_{T,k}}{dt} = \dot{V} (c_{T,k}^i - c_{T,k}^*) + \dot{n}_{T,k}^g + \dot{n}_{d2s} \quad (18)$$

$$\frac{dn_{TG,k}}{dt} = \dot{V} (c_{TG,k}^i - c_{TG,k}^*) + \dot{n}_{TG,k}^g \quad (19)$$

$$\frac{dn_{TS,k}}{dt} = \dot{V} (c_{TS,k}^i - c_{TS,k}^*) + \dot{n}_{TS,k}^g \quad (20)$$

Here, index k refers to the organ where metabolism takes place — either liver or skin. The term denoted \dot{n}_{d2s} is the

quantity of triclosan applied on the skin or taken in orally³. Excretion is through the kidney in the urine, and can be expressed as:

$$\dot{n}_{j,k2u} = \Theta_{k2u} c_{j,ins}^* \quad (21)$$

3.5 Flow Junction

The flow topology has one junction at the inlet to the blood pool. For this junction, no mass is accumulated. Steady state species balances for this junction lead to:

$$\begin{aligned} \dot{n}_{k,bp}^e &= \dot{n}_{k,ins}^e + \dot{n}_{k,br}^e + \dot{n}_{k,mu}^e + \dot{n}_{k,sk}^e + \dot{n}_{k,ki}^e \\ &+ \dot{n}_{k,bo}^e + \dot{n}_{k,fa}^e + \dot{n}_{k,li}^e \end{aligned} \quad (22)$$

$$\dot{n}_{k,bp}^e = \dot{n}_{k,lu}^i \quad (23)$$

Here, subscript k denotes the possible substrate, i.e., free triclosan, glucuronide, and sulfate.

4 Simulation

4.1 Parameters

Once model equations are written, their parameters must be derived from experiments found in the literature, or estimated from correlation formulas such as in Eq. 10. Parameters for BPA were used in the PBPK model for TCS where values for TCS were not found.

Triclosan has been shown to be sulfonated and glucuronidated in human liver, and parameters of TCS are used regarding metabolism of the liver as shown in Table 2. Specific data for metabolism in human skin is lacking. Although the absorption mechanisms are different, metabolism parameters in the skin are lower than those for the liver; it was assumed that 20 percent of the corresponding parameters for the liver can be used for the skin reactions.

4.2 Operating Condition

The following operating condition are used in the model:

- Initial state: all are zero.
- Body Weight (BW): 0.25 kg for rats and 73 kg for human.
- Chemical dose: 0.1 g/kg BW is supplied continuously for 1 hour in rats and 2 hours in humans.
- Consideration: relevant mass of protein in the skin is equal to the relevant mass of the protein of the liver $m_{p,sk} = m_{p,h}$.

4.3 Simulation Results

The simulations are carried out using MATLAB for both dermal and oral applications, using partition coefficient for BPA, but also using partition coefficients for TCS derived from Poulin's method. The results for TCS sulfate

follow the same pattern as those for TCS glucuronide, and only the TCS glucuronide results are reported due to space constraints.

Figures 5 and 6 show the levels of TCS and TCS glucuronide using BPA coefficient and parameters of TCS found in the literature, or assumed. Organs like small intestine, muscles, and kidney show the same trend: a marked peak in the start due to the intake, while the elimination rate differs for these organs. The concentration of free triclosan is high in the liver and in the skin, compared to the other organs (lungs, fat, bones) because free triclosan is added in these two organs. In Figure 6 for both oral and dermal application, the trend of concentrations of glucuronide in organs is similar to that of free triclosan. The concentrations of the metabolites take relatively long time to settle down. Higher peaks of concentration of metabolites are present in the small intestine, in the liver and in the skin, where the transformation of triclosan takes time.

These results can be compared with the PBPK model for BPA in plasma from (Fowler, 2013) through oral exposure. Their study indicates a rapid increase in concentrations due to intake of the chemical, which goes to zero after some 8 hours. The maximum peak of concentration has the same order of magnitude for free triclosan in the case in Figure 5 for the small intestine and muscle which is the organs where most of the BPA is accumulated (Fowler, 2013). The time axis is not directly comparable, due to a different amount of intake of the chemical. In (Völkel et al., 2002), metabolites of BPA are formed when higher doses of bisphenol are taken in. The results of (Völkel et al., 2002) show that after oral application of low doses of BPA in humans, only a very small percentage of the dose of BPA is available for other bio transformation pathways, due to the rapid glucuronidation.

Similarly, Figures 7 and 8 show the level of TCS and TCS glucuronide based on the Poulin formulae for partition coefficient estimation (Poulin and Krishnan, 1995b). In Figure 7, the concentration in the kidney is higher than that of the liver in the free triclosan of using PCs of BPA; the elimination of TCS takes approximately 10 days. Also, the amount of free triclosan in bones shows a considerable difference from using the PCs of BPA. The time for elimination of the concentrations for other organs seems to take in the order of 2 days. In Figure 8, bones and fat have the highest concentration of TCS glucuronide. It is also interesting to compare the elimination time for TCS glucuronide. The results of simulation using Poulin's formula for PCs cannot be compared with results in the literature, for the lack comparable results. A high concentration of triclosan and TCS glucuronide is visible in the kidney and the small intestine, both with dermal and oral application.

5 Conclusions

In this study, a PBPK model for triclosan and its conjugates was developed, which indicates their concentration

³Assuming no transport limitation.

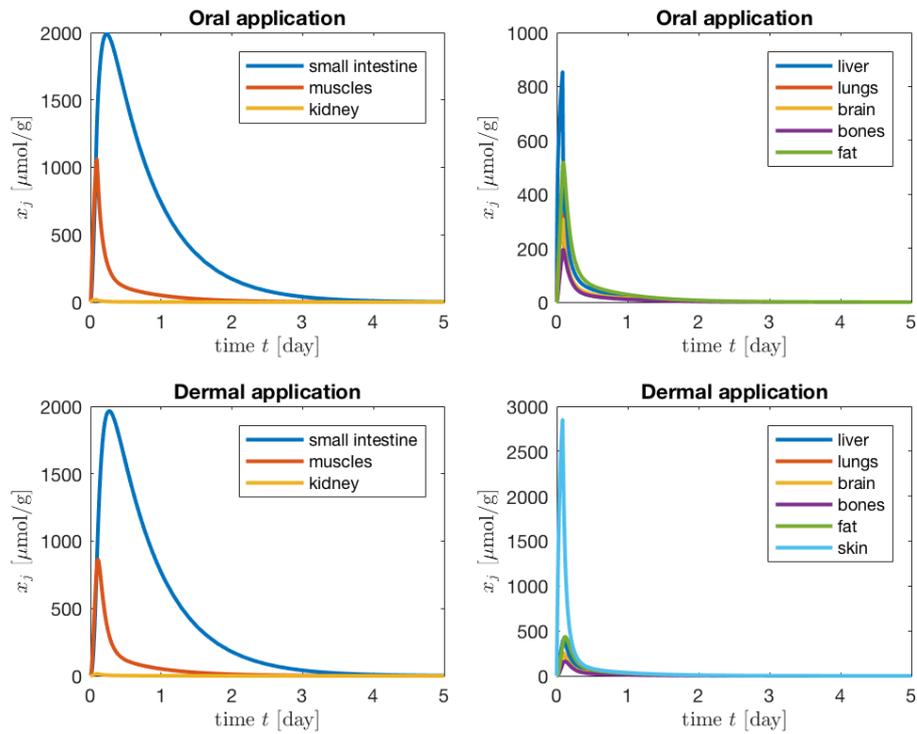


Figure 5. Free Triclosan concentration simulated using BPA PCs.

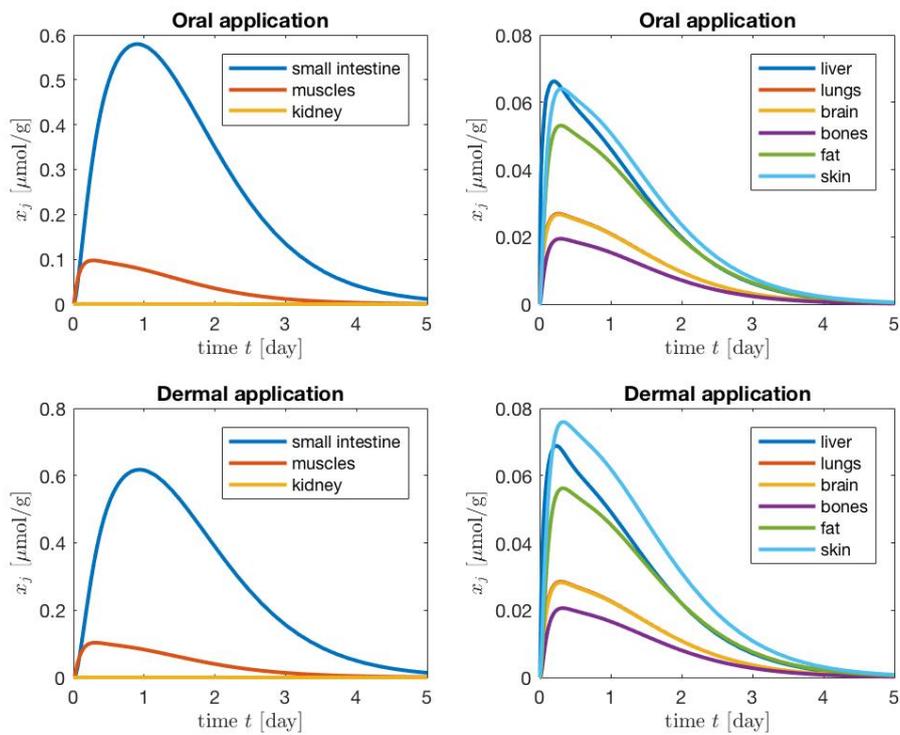


Figure 6. Glucuronide concentration of TCS simulated using BPA PCs.

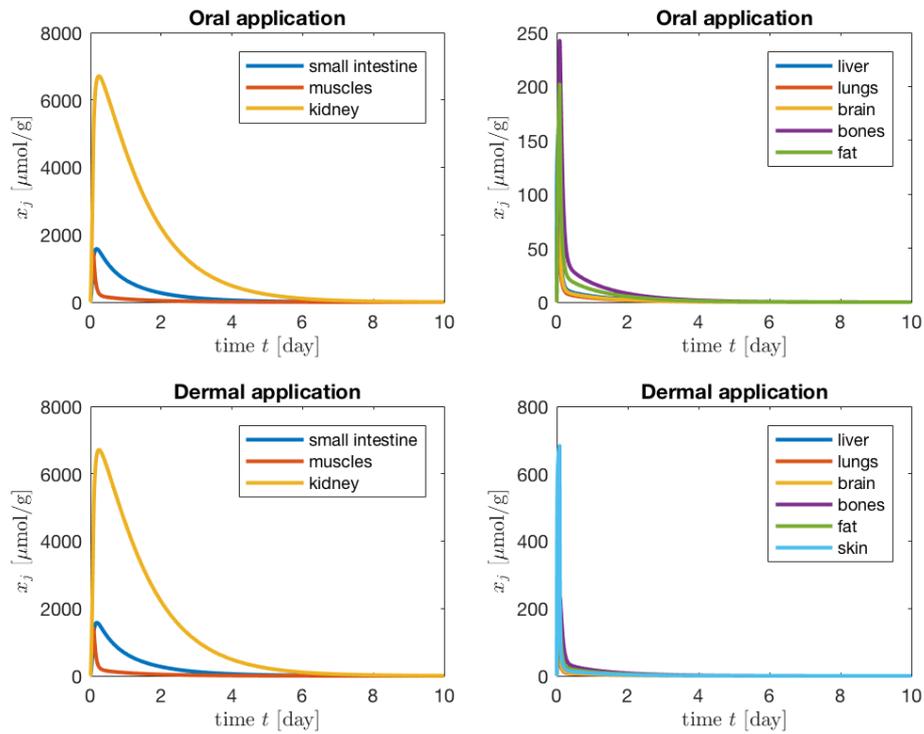


Figure 7. Free Triclosan concentration simulated using formula of Poulin for PCs.

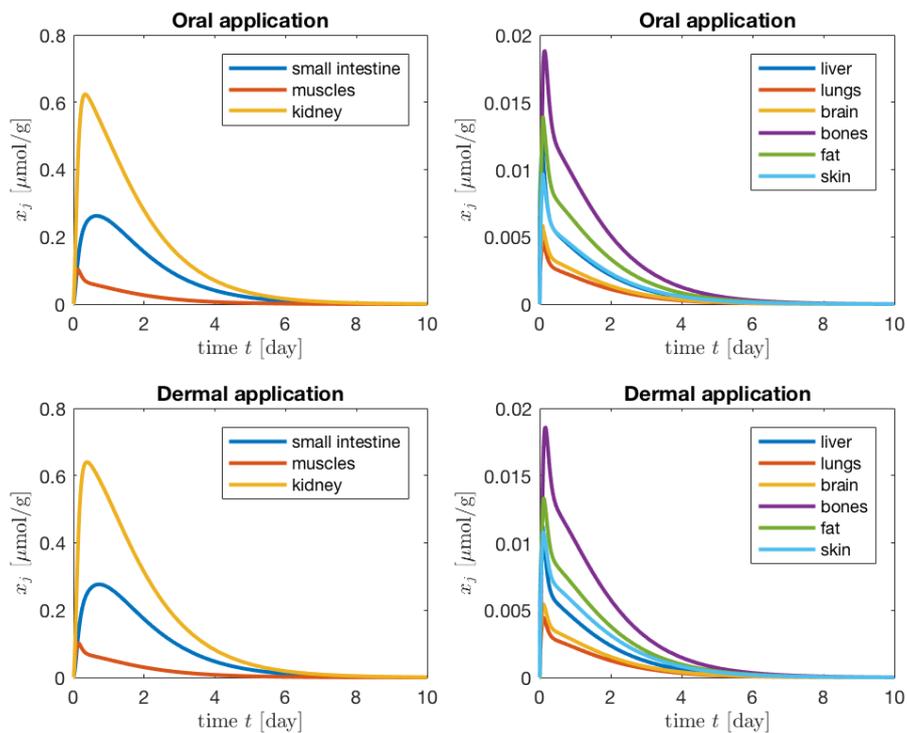


Figure 8. Glucuronide concentration of TCS simulated using formula of Poulin for PCs.

in organs of the human body. Organs such as muscles, fat, kidney, and liver have higher concentrations than the other organs, which seems reasonable and is in line with (Fang et al., 2010). The unavailability of parameters for Triclosan and simulation/experimental results for Triclosan in the literature makes it difficult to verify the results further at the moment.

References

- Z. E. Barter, M. K. Bayliss, P. H. Beaune, A. R. Boobis, D. J. Carlile, R. J. Edwards, J. B. Houston, B. G. Lake, J. C. Lipscomb, O. R. Pelkonen, G. T. Tucker, and A. Rostami-Hodjegan. Scaling factors for the extrapolation of in vivo metabolic drug clearance from in vitro data: reaching a consensus on values of human microsomal protein and hepatocellularity per gram of liver. *Current Drug Metabolism*, 8(1): 33–45, January 2007.
- Jerry L Campbell, Rebecca A Clewell, P Robinan Gentry, Melvin E Andersen, and Harvey J Clewell. Physiologically based pharmacokinetic/toxicokinetic modeling. In *Computational Toxicology*, pages 439–499. Springer, 2012.
- Helen E Cubitt, J Brian Houston, and Aleksandra Galetin. Relative importance of intestinal and hepatic glucuronidation — impact on the prediction of drug clearance. *Pharmaceutical research*, 26(5):1073, 2009.
- Andrea B Dann and Alice Hontela. Triclosan: environmental exposure, toxicity and mechanisms of action. *Journal of applied toxicology*, 31(4):285–311, 2011.
- Jia-Long Fang, Robin L Stingley, Frederick A Beland, Wafa Harrouk, Debbie L Lumpkins, and Paul Howard. Occurrence, efficacy, metabolism, and toxicity of triclosan. *Journal of Environmental Science and Health, Part C*, 28(3):147–171, 2010.
- Bruce A Fowler. *Computational toxicology: methods and applications for risk assessment*. Academic Press, 2013.
- Aleksandra Konieczna, Aleksandra Rutkowska, and D Rachon. Health risk of exposure to bisphenol a (bpa). *Roczniki Państwowego Zakładu Higieny*, 66(1), 2015.
- L. Kuepfer, C. Niederal, T. Wendl, J.-F. Schlender, S. Willmann, J. Lippert, M. Block, T. Eissing, and D. Teutonico. Applied concepts in pbpk modeling: How to build a pbpk/pd model. *CPT: Pharmacometrics & Systems Pharmacology*, 5(10):516–531, 2016.
- Patrick Poulin and Kannan Krishnan. A biologically-based algorithm for predicting human tissue: blood partition coefficients of organic chemicals. *Human & experimental toxicology*, 14(3):273–280, 1995a.
- Patrick Poulin and Kannan Krishnan. An algorithm for predicting tissue: Blood partition coefficients of organic chemicals from n-octanol: Water partition coefficient data. *Journal of Toxicology and Environmental Health, Part A Current Issues*, 46(1):117–129, 1995b.
- Anna Katharina Vingskes and Nicole Spann. The toxicity of a mixture of two antiseptics, triclosan and triclocarban, on reproduction and growth of the nematode *Caenorhabditis elegans*. *Ecotoxicology*, pages 1–10, 2018.
- Wolfgang Völkel, Thomas Colnot, György A Csanády, Johannes G Filser, and Wolfgang Dekant. Metabolism and kinetics of bisphenol a in humans at low doses following oral administration. *Chemical research in toxicology*, 15(10): 1281–1287, 2002.
- Li-Quan Wang, Charles N Falany, and Margaret O James. Triclosan as a substrate and inhibitor of 3'-phosphoadenosine 5'-phosphosulfate-sulfotransferase and udp-glucuronosyl transferase in human liver fractions. *Drug Metabolism and Disposition*, 32(10):1162–1169, 2004.
- Min Ye, Swati Nagar, and Ken Korzekwa. A physiologically based pharmacokinetic model to predict the pharmacokinetics of highly protein-bound drugs and impact of errors in plasma protein binding. *Biopharmaceutics & Drug Disposition*, 37(3):123–141, 2016.
- P Zhao, L Zhang, JA Grillo, Q Liu, JM Bullock, YJ Moon, P Song, SS Brar, R Madabushi, TC Wu, et al. Applications of physiologically based pharmacokinetic (pbpk) modeling and simulation during regulatory review. *Clinical Pharmacology & Therapeutics*, 89(2):259–267, 2011.
- Xiaomei Zhuang and Chuang Lu. Pbpk modeling and simulation in drug research and development. *Acta Pharmaceutica Sinica B*, 6(5):430–440, 2016.

A Data-Driven Sensitivity Analysis Approach for Dynamically Positioned Vessels

Xu Cheng^{1,2} Robert Skulstad² Guoyuan Li² Shengyong Chen¹ Hans Petter Hildre² Houxiang Zhang²

¹School of Computer Science and Technology, Tianjin University of Technology, China

chengxutjut@gmail.com, csy@tjut.edu.cn

²Department of Ocean Operations and Civil Engineering, Norwegian University of Science and Technology, Norway
{xu.cheng, robert.skulstad, guoyuan.li, hans.p.hildre, hozh}@ntnu.no

Abstract

For safety-critical marine operations, the dynamically positioned (DP) vessel should maintain a predetermined heading and position for varying environmental conditions using the thrusters. Studying the effect of each thruster to the capability of DP vessels is significance but challenging. This paper presents a data-driven and variance-based sensitivity analysis (SA) approach that can dig into the ship sensor data to estimate the influence of each thruster for DP operations. Considering high-computational cost of variance-based SA, an Extreme Learning Machine (ELM)-based SA is proposed. To apply the SA to sensor data, an ANN is built and trained on the basis of ship sensor data and then employed as a surrogate model to generate Monte Carlo (MC) samples. A benchmark test shows the correctness of the proposed approach. A case study of SA in DP operation is conducted and the experimental results show that the proposed approach can rank and identify the most sensitive factors. The proposed approach highlights the application of variance-based SA in data-driven modeling for ship intelligence.

Keywords: dynamical positioning, sensitivity analysis, thrust analysis, data-driven modeling

1 Introduction

With the development of oil and gas exploration and other operations in the deep sea, DP vessels have obtained more and more attention. A DP vessel is defined as a vessel that can maintain a specified position and heading automatically by employing its thrusters to guarantee the continuous offshore operation (Sørensen, 2011). The safety of operation of the DP vessel is often the first consideration to be paid attention to, especially in those safety-critical operations such as drilling, oil production, and off-loading, where the positioning and heading accuracy are very high, regardless of the environmental conditions. To satisfy the critical requirements, it is necessary to understand the effect of each thruster to against the environmental factors (Xu et al., 2015; Pivano et al., 2012).

However, it is not an easy task to study the influence of each thruster in DP vessels. Mahfouz and El-Tahan de-

veloped a software program with the aim of selecting and configuring the thrusters for the newly designed DP vessels (Mahfouz and El-Tahan, 2006). A program which is named as DPCAP is developed by the Maritime Research Institute Netherlands (MARIN) with the purpose of analyzing the thruster capacity of the DP vessels. The main drawback of the DPCAP is that there is no quantitative method to compare the different thruster configurations. To improve the ability of DPCAP, Xu et al. proposed a novel thrust sensitivity analysis on the basis of a newly defined synthesized positioning capability criterion for marine vessels (Xu et al., 2015). The main limitation of this method is that it employs local sensitivity analysis (LSA), which can only reflect the characteristics of the synthesized capability at some fixed points in the input space.

To address the limitation of Xu's method to study the effect of each thruster in DP vessels, a data-driven and variance-based SA approach is proposed. The approach utilizes the historical data of DP to analyze the influence of each thruster. In addition, in order to be able to apply the proposed method to most marine operations to mine useful information, this paper also proposes a framework for data analysis in marine operations.

Our on-going project aims to develop intelligent systems to support decision making in various maritime operations. A new integrated platform including data analysis tools, and data-driven modeling technique is designed, which will serve the maritime industry for improving operational effectiveness and safety. In this paper, only data analysis tools part would be focused on. The variance-based SA would be employed to understand the DP data (Fernández-Navarro et al., 2017). In the case of only sensor data available, it is still a challenge to employ the variance-based SA directly to the sensor data of the DP vessels. Thus, a fitted and validated surrogate model representing the motion of DP vessel is constructed on the basis of sensor data that are observed in a simulated vessel. To accelerate the SA, an ELM is adapted. The contribution of this paper includes (1) introducing a framework of data analysis for offshore operations, and (2) providing a data-driven SA method that is capable of coping with the SA on the DP data.

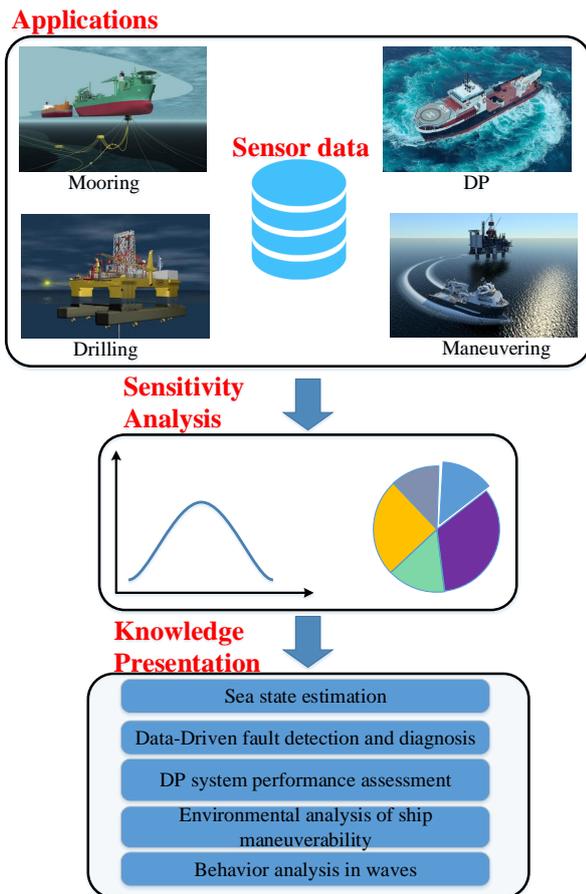


Figure 1. Framework of data analysis in offshore operations.

The rest of the paper is organized as follows. Section 2 shortly introduces the overall structure of the data analysis framework. In Section 3, the ELM-based sensitivity analysis is described in detail. Section 4 presents case studies and evaluation results. Conclusion and future work are shown in Section 5.

2 Framework of data analysis for offshore operations

Due to the significant uncertainties and various operating conditions, offshore operations are complex and hazardous. Operational safety is a major issue and is easily challenged by harsh marine environments, complex geological conditions, and human and equipment factors. The conventional model-based solutions require an in-depth knowledge of the offshore operations, which are impractical for complicated offshore operations. Fortunately, with the rapid development of technologies such as data collection, data mining and artificial intelligence, the extraction of useful information has been significantly improved in the offshore operations (Li et al., 2016, 2017). Hence, data-driven technology is an alternative that can be used for efficient operational monitoring and data analysis (Cheng et al., 2017; Wang et al., 2016).

A considerable amount of sensor data has been accumu-

lated in all kinds of offshore operations. Analyzing these data has the following difficulties: 1) the sensor data to be analyzed are usually too large and high dimensional; 2) they usually contain measurement-induced noise and redundant information, which makes it difficult to analyze accurately; 3) It is not easy to intuitively interpret the data from multiple sensors. Furthermore, there are many uncertainties in these data, which are influenced by a variety of factors, such as weather conditions, human factors, and so on. How to characterize and reduce that uncertainty is becoming more and more popular in engineering research. As an aspect of uncertainty quantification, the SA is defined as the investigation of “how uncertainty in the output of a model (numerical or otherwise) can be apportioned to different sources of uncertainty in the model input factors” (Saltelli, 2002). SA has been widely used for industrial applications with different purposes, mainly including assessing the uncertainty, calibrating the model, and making robust decision (Pianosi et al., 2016). In general, SA could be implemented in either a local or a global manner. The difference is that the former analyzes the effect of a single input on the output and treats the other inputs as deterministic values; whereas the latter examines the sensitivity from the perspective of the entire range of each input’s variation.

Figure 1 illustrates a possible scheme of the data analysis for offshore operations. Note that the sensitivity analysis module is the core of the scheme. It takes the sensor data from numerous offshore applications as input and the designated metric, e.g., the capability of DP vessels, as output, to quantify how much the input contributes to the output. The result can benefit both the DP system performance assessment and behavior analysis in waves. For example, if one thruster accounts for one of the main factors for DP system, this element will be given more attention, since, if it fails, the remaining thrusters may not possible produce sufficient forces to keep the vessel on the predetermined position and heading. The information from the sensitivity analysis can also be obtained to establish an estimator for sea state identification. Based on the framework, it is possible to implement an intelligent system that utilizes data analysis method, artificial intelligent algorithms and advance control theory to achieve ship intelligence for various purposes ranging from sea state estimation, risk assessment, manoeuvring evaluation, sensor diagnosis to behavior analysis.

3 Proposed Approach

To test the feasibility of the introduced data analysis framework, a data-driven sensitivity analysis approach for DP operations is proposed, as shown in Figure 2. The process of the proposed approach is as follows: (1) Training the artificial neural network (ANN) to generate a surrogate model on the basis of DP sensor data. (2) Validating the model. (3) Performing variance-based SA on the well-trained surrogate model. In order to accelerate

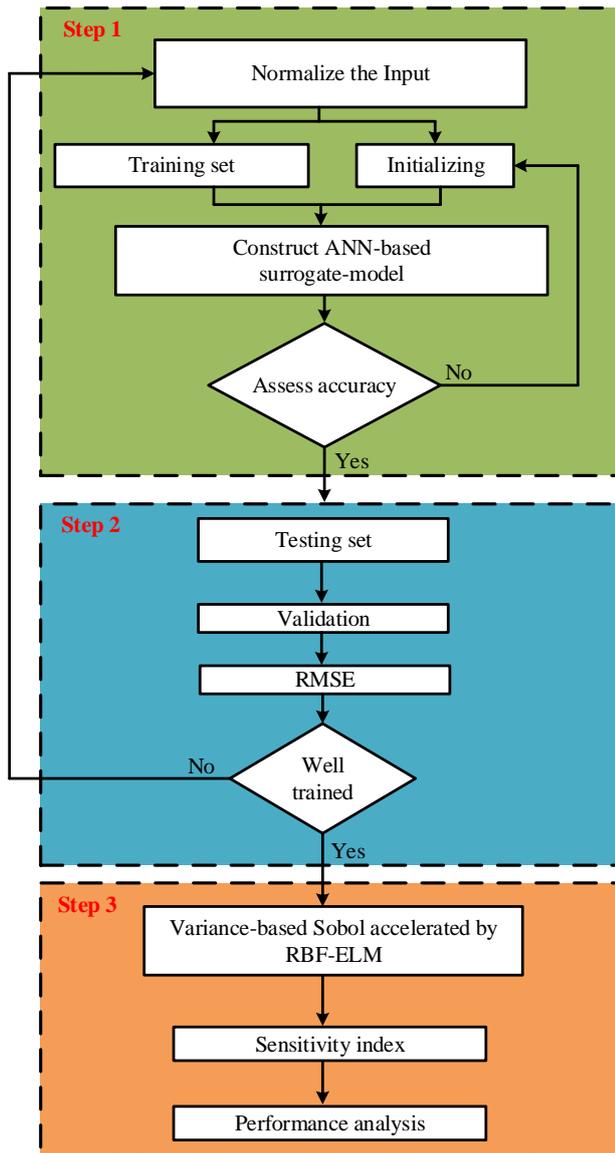


Figure 2. Workflow of the proposed approach.

Sobol' method, Radial basis function (RBF)-based ELM is integrated into Sobol'.

3.1 ANN-based surrogate model

The data-driven ANN is an essential part of the scheme. It has multiple roles: First, it provides a bridge between the modeler and the sensor data to better simulate and understand the ship's behavior. Second, the data-driven ANN model is employed to generate a number of input parameters using Latin Hyper-cube Sampling (LHS) technique and to calculate the corresponding output. In this paper, a three-layer feed-forward neural network is established for the proposed method. The Back-Propagation (BP) algorithm is used as a learning algorithm. Sigmoid functions and linear functions are used as activation functions for the hidden and output layers, respectively. During the training phase, 80% of sensor data is used for training, and the remaining 20% is used for testing and validation.

3.2 ELM-Sobol SA method

Assuming the model form is $f(\mathbf{X}) = f(x_1, \dots, x_M)$, where $\mathbf{X} = (x_1, \dots, x_M)$ represents the model input which contains M independent parameters. The model output can be decomposed as follows (Saltelli and Sobol', 1995; Li et al., 2006):

$$f(\mathbf{X}) = f_0 + \sum_{i=1}^M f_i(x_i) + \sum_{1 \leq i < j \leq M} f_{ij}(x_i, x_j). \quad (1)$$

Eq.(1) is known as ANOVA-representation (Analysis of Variance). f_0 is the mean of output. f_i and f_{ij} are the first- and second-order decomposition of output, respectively.

The variance of the model output can be obtained as follows:

$$V_y = E(f(\mathbf{X})^2) - f_0^2 \quad (2)$$

The main sensitivity index can be defined as follows:

$$S_i = \frac{V_i}{V_y} = \frac{V(E(f(\mathbf{X})|x_i))}{V_y} \quad (3)$$

The Sobol' method employs the MC methods to calculate the sensitivity index which brings high computational complexity. In this paper, the idea of Wu et al. (Wu et al., 2016) is adapted to reduce the computational complexity of Sobol'.

Assuming the model output can be represented by the RBF-based ELM, which formulation is shown as follows:

$$f(\mathbf{X}) = \sum_{i=1}^N \omega_i \varphi_i(\mathbf{X}) \quad (4)$$

where φ_i is the i -th basis function and ω_i is the coefficient of i -th basis function. According to Eq.(1), f_0 is the expectation of the model output.

$$f_0 = \int_0^1 f(\mathbf{x}) d\mathbf{x} = \sum_{i=1}^N \int_0^1 \omega_i \varphi_i(\mathbf{x}) d\mathbf{x} = \sum_{i=1}^N \omega_i \prod_{j=1}^m \psi_i^j \quad (5)$$

The ψ_i^j can be calculated using the Gaussian cumulative distribution function.

$$\psi_i^j = c_i \sqrt{\pi} \left[\Phi \left\{ \sqrt{2} \frac{1-x_i^j}{c_i} \right\} + \Phi \left\{ \sqrt{2} \frac{x_i^j}{c_i} \right\} - 1 \right] \quad (6)$$

where Φ is the Gaussian distribution function, and c_i is the center of i -th basis function. The mean square of the model output can be computed as:

$$\begin{aligned} E(f(\mathbf{x})^2) &= \int_0^1 f^2(\mathbf{x}) d\mathbf{x} = \sum_{k=1}^N \sum_{i=1}^N \int_0^1 \omega_k \omega_i \varphi_k(\mathbf{x}) \varphi_i(\mathbf{x}) d\mathbf{x} \\ &= \sum_{k=1}^N \sum_{i=1}^N \omega_k \omega_i \prod_{j=1}^m \psi_{ki}^j \end{aligned} \quad (7)$$

where

$$\psi_{ki}^j = e^{-\frac{x_i^j - x_k^j}{c_{ik}^2}} c_{ik} \sqrt{\pi} \left[\Phi \left\{ \sqrt{2} \frac{1 - x_{ik}^j}{c_{ik}} \right\} + \Phi \left\{ \sqrt{2} \frac{x_{ik}^j}{c_{ik}} \right\} - 1 \right] \quad (8)$$

where $c_{ik}^2 = \frac{c_k^2 c_i^2}{c_k^2 + c_i^2}$, $x_{ik}^j = \frac{c_k^2 x_i^j + c_i^2 x_k^j}{c_k^2 + c_i^2}$

Considering the input factor x_d , the first order term is

$$\begin{aligned} f_d(x_d) &= E(y|x_d) - f_0 = \int_0^1 \int_0^1 f(\mathbf{x}) \prod_{j=1, j \neq d}^m dx_j - f_0 \\ &= \sum_{i=1}^N \omega_i \varphi_i(x_d) \prod_{j=1, j \neq d} \psi_i^j - f_0 \end{aligned} \quad (9)$$

Then the partial variance of x_d can be calculated as follows:

$$\begin{aligned} V[E(y|x_d)] &= E(f_d^2(x_d)) - f_0^2 \\ &= \sum_{i=1}^N \sum_{k=1}^N \omega_i \omega_k \psi_{ki}^d \prod_{j=1, j \neq d} \psi_i^j \psi_k^j - f_0^2 \end{aligned} \quad (10)$$

One can see that the Sobol' method can be computed by the coefficients of RBF.

4 Experiment

This section mainly refers to three relevant experiments which are used in verifying the feasibility of the method we proposed.

4.1 Benchmark function

The first subsection is dedicated to the validation of the proposed approach. The widely used benchmark Sobol' function (Kersaudy et al., 2015) in variance based GSA is employed to test the performance of the proposed approach. The expression of Sobol' function is shown as follows:

$$f(\mathbf{x}) = \prod_{i=1}^M \frac{|4x_i - 2| + a_i}{1 + a_i} \quad \mathbf{x} \in [0, 1]^M \quad (11)$$

In this paper, $M = 7$ is the number of input parameters. The $a_1 - a_7$ are set to 1, 2, 5, 10, 20, 50, 100. First, the original samples are set big enough to generate a high accuracy ANN model, and 10 ANN models are created, and the best model is chosen from the 10 models based on the performance test using RMSE. The sensitivity results are listed in Table 1.

As the Table 1 shows, the proposed approach is competent to improve the computational efficiency but not sacrificing on precision, compared with the Monte Carlo method. The proposed approach significantly reduces the computational complexity regarding the number of ANN

Table 1. First-order sensitivity index of each input.

Input	Samples of Proposed approach			Samples of Monte Carlo	Analytical
	128	256	512	5000	
x_1	0.334	0.570	0.595	0.609	0.604
x_2	0.294	0.240	0.265	0.272	0.268
x_3	0.041	0.061	0.063	0.071	0.067
x_4	0.015	0.019	0.015	0.019	0.020
x_5	0.007	0.008	0.011	0.011	0.010
x_6	0.005	0.005	0.006	0.001	0
x_7	0.009	0.007	0.002	0.001	0

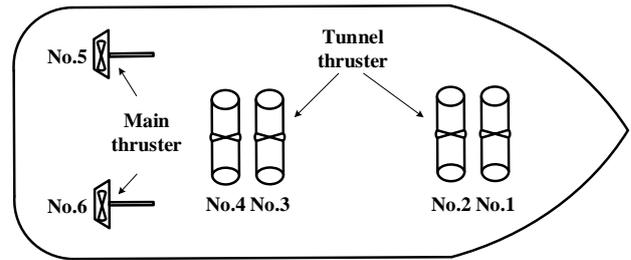


Figure 3. Thruster configuration

samples. Considering the complexity of a model, the proposed approach does not raise the computational complexity of conducting the variance-based GSA. Therefore, the more complex the model is, the more computation time would be reduced.

4.2 Sensitivity result of the DP thrusters

A sensitivity analysis for a DP vessel is conducted to assess how effectively the analysis determined the most sensitive thruster. The case ship model used is equipped with two tunnel thrusters in the bow, two tunnel thrusters and two main propellers at the stern, which is shown as Figure 3. The vessel is equipped with DP controller and the ship motion data and thruster data are collected.

For better analysis, we assume uniform, independent distributions for all parameters and initial conditions. The definition of input variability space then only requires defining the range of variation of each input factor. In this study, we chose to use ranges as wide as possible. Some parameters have the range according to the physical meaning of the parameters; the range of the others are confined by the minimum and maximum observed values of that variable over the entire data set.

As the performance of DP vessel is strongly related to the wind and wave, this section focuses on the influence of each thruster to the ship heading under different environmental effects. Therefore, ship heading was chosen as output variable, the output thrust of the six thrusters are considered as the input variables of the ANN model. The DP operation is simulated in the Offshore Simulator Centre AS (OSC), with the height of the wave assumed to remain constant at 2 m (ITTC spectrum and 15s period). The wind increases from 4 m/s to 12 m/s. Three scenarios with the wind and wave come from three different direc-

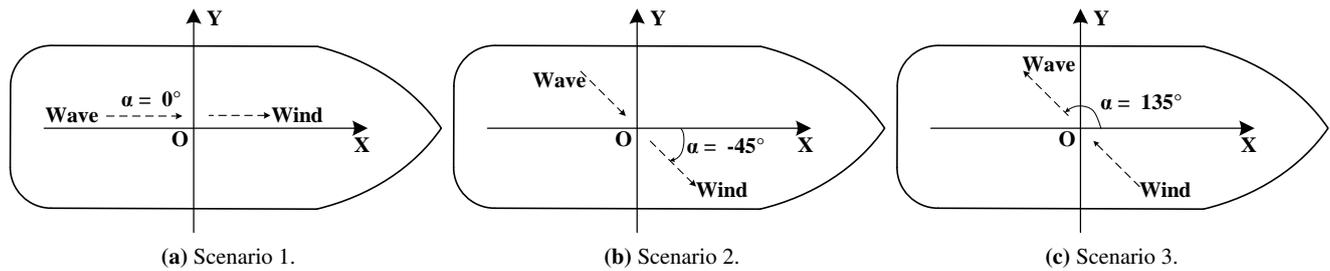


Figure 4. Three scenarios with different environmental factors.

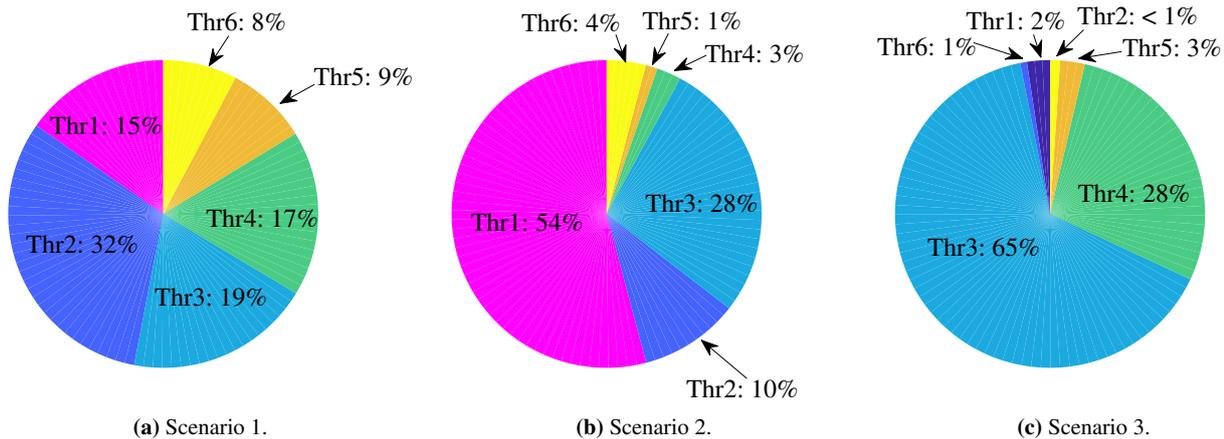


Figure 5. Sensitivity results of three scenarios.

tions are conducted, as shown in Figure 4.

Figure 5a, Figure 5b and Figure 5c show the sensitivity results of the three scenarios, respectively. In scenario 1, it can be seen that there are four parameters (thruster 1 ~ thruster 4) which are important to heading. The possible explanation for scenario 1 is that the transverse force and moment are highly depends on the four thrusters, and the wave and wind have no influence on the transverse force and moment. In scenario 2, it is obvious that the most sensitive thruster is thruster 1, and the thruster 3 follows. In order to overcome the transverse force generated by wind and waves, thruster 1 and thruster 3 supply sufficient forces in transverse direction. As showed in Figure 5c, it is clear that the most sensitive thruster is the thruster 3. This is also recognized in the greater sensitivity result in thruster 4. As showed in Figure 4, the direction of wave and wind are the same. Hence, the expected thrust would be provided by those thrusters with orientation close to the direction of the environmental force. It can be concluded that, to improve the ability of DP in scenario 3. It is most effective to increase the propulsion capability of thruster 3.

4.3 Knowledge understanding of DP

In this paper, a dual-layer DP controller was used, including the motion controller and thruster force allocation controller. This paper mainly focuses on whether the thrust allocation is reasonable under different environmen-

tal disturbances and whether it can maintain the ship's position and heading. When the wind and waves come from the longitudinal direction, the main thruster will generate strong thrust to counteract the influence of wind and waves. Note that the two main thrusters are operated using the same command and therefore does not contribute any torque about the transverse direction. In this case, the tunnel thruster with a large moment of force should have a greater influence on the heading angle. From the experimental results, as showed in Figure 5a, the thruster 2 has a large influence on the heading angle and the main thruster (thruster 5 and thruster 6) has relative lower influence. Therefore, the force allocation is reasonable under such circumstances. When the wind and waves come from the side, the thrust generated by the tunnel thruster near the environmental force should have a greater influence on the ship heading. The reason is that those thrusters close to environmental forces need to generate large thrusts to counteract environmental forces. From the experimental results of scenario 2 and scenario 3, when the force by the wind and wave is close to the orientation of thruster 1 and 2, the influence of thruster 1 and 2 should be greater. When the force by the wind and wave is close to the orientation of thrusters 3 and 4, thrusters 3 and 4 should have a higher influence.

5 Conclusion

In this paper, a data analysis framework for offshore operations is introduced. An approach integrated ELM-based SA with ANN is proposed that can be applied for DP sensor data. This approach can be employed to quantify the influence of each thruster of DP vessels. To improve the capability of the DP vessels, the most effective way is to increase the thrust when the most sensitive thruster is identified. In order to verify the feasibility of this approach, a benchmark test is conducted. The benchmark test validated the feasibility of the proposed method used for conducting SA. Another experiment of SA on heading modeling in DP operation was conducted. The result shows that this approach can identify those influential factors that have an effect on the heading of the DP vessel. It is seen that the sensitivity analysis of thruster depends crucially on the environmental disturbances. Thus, the sensitivity analysis is of value in the preliminary design of the thrust system of DP vessels.

The sensitivity analysis of the thruster is highly time-dependent. However, in this study, the surrogate model is ANN which cannot represent the dynamic of the DP vessel completely. Therefore, our future work will focus on investigating how to extend the proposed method to adapt to such external disturbances.

6 Acknowledgement

This research is partially supported by the project “SFI MOVE” funded by Norway Research Council, Norway (Project No: 237929). The authors would like to thank to the Offshore Simulator Centre AS for technical support. The author Xu Cheng would like to thank the sponsorship of the Chinese Scholarship Council for funding his research at Norwegian University of Science and Technology.

References

Offshore simulator centre. <http://www.offsim.no/>.

Xu Cheng, Shengyong Chen, Chen Diao, Mengna Liu, Guoyuan Li, and Houxiang Zhang. Simplifying neural network based model for ship motion prediction: A comparative study of sensitivity analysis. In *ASME 2017 36th International Conference on Ocean, Offshore and Arctic Engineering*, pages V001T01A016–V001T01A016. American Society of Mechanical Engineers, 2017.

Francisco Fernández-Navarro, Mariano Carbonero-Ruz, David Becerra Alonso, and Mercedes Torres-Jiménez. Global sensitivity estimates for neural network classifiers. *IEEE transactions on neural networks and learning systems*, 28(11):2592–2604, 2017.

Pierric Kersaudy, Bruno Sudret, Nadège Varsier, Odile Picon, and Joe Wiart. A new surrogate modeling technique combining kriging and polynomial chaos expansions—application to uncertainty analysis in computational dosimetry. *Journal of Computational Physics*, 286:103–117, 2015.

Genyuan Li, Jishan Hu, Sheng Wei Wang, Panos G Georgopoulos, Jacqueline Schoendorf, and Herschel Rabitz. Random sampling-high dimensional model representation (rs-hdmr) and orthogonality of its different order component functions. *The Journal of Physical Chemistry A*, 110(7):2474–2485, 2006.

Guoyuan Li, Houxiang Zhang, Bikram Kawan, Hao Wang, Otmar L Osen, and Arne Styve. Analysis and modeling of sensor data for ship motion prediction. In *OCEANS 2016-Shanghai*, pages 1–7. IEEE, 2016.

Guoyuan Li, Bikram Kawan, Hao Wang, and Houxiang Zhang. Neural-network-based modelling and analysis for time series prediction of ship motion. *Ship technology research*, 64(1): 30–39, 2017.

Ayman B Mahfouz and Hussein W El-Tahan. On the use of the capability polar plots program for dynamic positioning systems for marine vessels. *Ocean engineering*, 33(8-9):1070–1089, 2006.

Francesca Pianosi, Keith Beven, Jim Freer, Jim W Hall, Jonathan Rougier, David B Stephenson, and Thorsten Wagener. Sensitivity analysis of environmental models: A systematic review with practical workflow. *Environmental Modelling & Software*, 79:214–232, 2016.

Luca Pivano, Øyvind Notland Smogeli, and Bjørnar Vik. Dyncap—the next level dynamic dp capability analysis. *Marine Cybernetics AS*, 2012.

Andrea Saltelli. Sensitivity analysis for importance assessment. *Risk analysis*, 22(3):579–590, 2002.

Andrea Saltelli and Il’ya Meerovich Sobol’. Sensitivity analysis for nonlinear mathematical models: numerical experience. *Matematicheskoe Modelirovanie*, 7(11):16–28, 1995.

Asgeir J Sørensen. A survey of dynamic positioning control systems. *Annual reviews in control*, 35(1):123–136, 2011.

Hao Wang, Sindre Fossen, Fang Han, Ibrahim A Hameed, and Guoyuan Li. Towards data-driven identification and analysis of propeller ventilation. In *OCEANS 2016-Shanghai*, pages 1–6. IEEE, 2016.

Zeping Wu, Donghui Wang, Patrick Okolo, Fan Hu, and Weihua Zhang. Global sensitivity analysis using a gaussian radial basis function metamodel. *Reliability Engineering & System Safety*, 154:171–179, 2016.

Shengwen Xu, Xuefeng Wang, Lei Wang, Shuai Meng, and Bo Li. A thrust sensitivity analysis based on a synthesized positioning capability criterion in dpcap/dyncap analysis for marine vessels. *Ocean Engineering*, 108:164–172, 2015.

Web Enabled High Fidelity Drilling Computer Model with User-Friendly Interface for Education, Research and Innovation

Robert Ewald¹, Jan Einar Gravdal¹, Dan Sui², Roman Shor³

¹IRIS - International Research Institute of Stavanger, Norway, ²University of Stavanger, Dept. of Petroleum Technology, ³University of Calgary, Dept. of Chemical and Petroleum Engineering, Canada

Abstract

Next generation intelligent software for drilling control systems together with automated monitoring and analysis systems is expected to save costs for the drilling industry. However, the transition from monitoring a process, which today is controlled manually, to automating the process requires a step-change in education of personnel as well as in infrastructure for development and testing new technology. The lack of high quality field data from drilling and well operations is a major problem in research and innovation projects within the oil & gas and geothermal drilling sector, as well as in education within these areas. Since 2015, IRIS and the University of Stavanger have developed a web enabled high fidelity drilling simulator as part of the OpenLab Drilling project¹.

This paper describes the objectives of the project, the technical solutions of the web enabled drilling simulator, and the results obtained during the first year after deployment to the users.

Keywords: Process modeling, Well construction, Process control, Simulator, Web Interface

1 Introduction

During a drilling operation large amount of data is collected. However, the amount of information that can be retrieved from the data is often sparse, and the efforts to use the data for analysis and process control purposes is very challenging (Arnaout et al 2013; Ashok, et al 2016; Maidla et al 2018; Cayeux et al, 2013a). The lack of high-quality data and information about the data (metadata) also hinders technology development and demonstration of innovative solutions in realistic environments. However, technology can be tested and demonstrated in a simulated environment up to a certain technology readiness level. IRIS, together with Statoil and Sekal² has experience with the DrillTronics™ system where advanced drilling control software has been tested extensively through many years in a

simulated environment (Cayeux et al 2012). Worldwide, there are several drilling simulators available that include both transient hydraulics- and drill string mechanic computations. In Norway, research and development at Rogaland Research (later known as IRIS) in the 1980's and 90's paved way for the well flow model RF-Kick (Petersen et al, 1998), later commercialized in DrillBench (which later became part of a Schlumberger Software suite). The RF-Kick was verified by data collected from Ullrigg in Stavanger and validated through various test campaigns both at Ullrigg and other wells (Rommetveit et al 1991). The model was used for planning and special studies for drilling operations. From 2001, Rogaland Research initiated a large program to improve their well flow model and prepare for real-time (online) simulations. A new numerical solver was developed (Lorentzen et al 2002) and the well flow model was integrated with other models such as a torque & drag model (Yi et al 2004) and transient cuttings transport model (Cayeux et al 2013b), and recently a transient torque & drag model (Cayeux et al 2018). This simulation model is commonly known as WeMod. Another simulation model was developed during the 2000s by SINTEF (Petersen et al 2008). This model, together with a torque & drag model is now used in software for training and real-time monitoring (Ødegård et al 2013) now commercialized by eDrilling³. Another renown well flow model is the simplified well flow model developed at the Norwegian University of Science and Technology and Statoil by Kaasa et al. (2012). A thorough review of these and other models are given by Aarsnes et al (2014).

Common for software based on simulation models developed at the research communities is that the user interface is designed for scientists and engineers skilled to set up the cases (configure the model), run the software and analyze the results. This has the benefit that the complexity of the models can be exploited to their full potential. However, this is at the expense of

¹ OpenLab Drilling (formerly known as Virtual Arena) is an infrastructure project funded by the Research Council of Norway: www.openlabdrilling.com

² Sekal is a Norwegian company owned by Statoil, Saudi Aramco and IRIS: www.sekal.com

³ eDrilling is a Norwegian Company owned by Teresoft.

user friendliness, and makes it difficult for non-experts to utilize such models.

2 Project objective

The overall objective of the OpenLab Drilling project is to stimulate education, research and innovation and to increase the acceptance and uptake of new technologies within automated drilling. One of the secondary objectives is to create a user-friendly interface to high fidelity models so that realistic (but artificial) data can be available for students, lecturers, and analysts (researchers and engineers). This is the motivation behind the development of the web enabled drilling simulator.

3 OpenLab simulator

The simulator uses IRIS WeMod as the core model, with its high fidelity transient well flow and drill string vibrational model. Because of the advantages over desktop applications, a web-based solution has been preferred. Among the benefits that have been considered are:

- Cost effective development
- Accessible anywhere
- Easily customizable
- Accessible from a range of devices
- Improved interoperability
- Easier installation and maintenance
- Adaptable to increased workload
- Increased Security when deployed on dedicated servers
- Promotes e-learning and thereby more efficient and environment friendly collaboration which reduces the need for travel.

The motivation by developing the web-based architecture is that a software available through a web application programming interface (API) has the potential to become an easily accessible software for students, lecturers and analysts. To reach out to as many users as possible the OpenLab Drilling project has focused on user friendliness for students, lecturers and analysts, and for those who are not experienced with drilling simulation software. The graphical user interface (GUI) has therefore been developed in close collaboration with drilling engineers as well as unexperienced students and lecturers of drilling courses and experienced researchers.

4 User groups

To guide the design of the user interaction it was early decided which user groups to support and how these should be prioritized. This has guided the development team to make decisions when conflicting solutions have been proposed. Each user group has specific needs and skills, which have been assessed through interviews and user tests.

Students

- are not familiar with the industrial way of presenting drilling data
- are familiar with web interaction paradigms
- need to learn how inputs to the drilling system affect drilling operations

Lecturers

- desire simple to use but physically accurate scenario-based simulations
- prefer that students spend time understanding the scenario rather than the simulation system
- benefit from systems which allow easy collaboration or knowledge sharing
- demonstrate drilling simulations in different scenarios for classroom teaching
- design projects, tasks & assignments in an easy way
- need for the teaching with digital technologies

Analysts/Engineers

- collaborate to a larger extent with other users.
- access various configurations and simulations.

Although these user groups and their individual users will have different needs and expectations to a drilling simulator, there are some common requirements that needed to be met. These are mainly related to user friendliness in configuration and simulation, to uptime and robustness of the software, and to access and storage of configurations and results.

5 User friendliness

The complex configuration needed to run high-fidelity simulators is a barrier to entry. Such models require a huge amount of configuration parameters to be set. A configuration in the Web Enabled Drilling Simulator is a description of the different elements and components that affect the circulation system and drillstring mechanics. In OpenLab Drilling this challenge has been addressed with a few complementing design decisions: Adequate defaults, deliberate choice of interaction components, and extensive input validation. Selection of default configurations have a big impact to lower the barrier of entry. A simple choice of preconfigured wells which cover different use cases is therefore presented for the user at entry. Each of these is ready to be simulated interactively. The user is guided through the configuration of casing architecture, trajectory, drill string design, drilling fluid parameters, geo-pressures and geothermal gradient. All of which comes with default configuration parameters. This allows learning the application in a step-by-step fashion while remaining sufficient flexibility for experienced users to input complex scenarios.



Figure 1: Simulation results are available in graphs from which data can be downloaded.

6 Input Validation

Although adequate defaults are provided, the user may adjust many interacting configuration options and setpoints. Without any validation of configurations and simulation setpoints, inconsistent options lead to crashing the simulator. This is unacceptable since the user is not informed about the inner workings of the simulator. To remedy this, a validation layer is included which catches conflicting configuration parameters which might crash the simulator. Nevertheless, the default configurations can be changed completely based on the user's own specifications. A change in a configuration may lead to validation errors that comes together with a message describing precisely which variables that need to be changed to regain validity. A configuration is not approved for simulation until all validation errors are removed by the user.

The validation and the quick feedback it enables, teaches the user how drilling data is related. With a valid configuration it is guaranteed that a simulation can be initialized and started.

7 Simulations

OpenLab Drilling is flexible for a user to simulate a wide range of scenarios such as well control, hole cleaning problems, buckling, and drill string vibrations to mention a few. To run a new simulation, a few initializations parameters need to be entered. The initial depth of the drill bit defines a starting point for the simulation. When running a simulation several setpoints (manipulated variables) can be edited at any time to simulate a drilling process. These are related to (1) fluid flow, (2) drill string movements, and (3) choke openings.

The setpoints are constrained by upper and lower bounds to prevent the user from entering unphysical values or values that may crash the simulator. The validation of configuration parameters and setpoints guarantees that a simulation will start and run. However, as for real drilling operations the user (or driller) can end up in a situation where only careful adjustment of the drilling parameters will prevent the drill string to become stuck in the borehole, or severe gas kicks to destabilize the well. A simulation can be run into a state where the models can no longer produce a reliable result and eventually crash. This can e.g. be the

case when large amounts of cuttings accumulate in the well because of insufficient hole cleaning, and eventually creates a completely blocking of the annulus. It should be emphasized that each simulation is an individual process and a crashed simulation will of course not affect other simulations running concurrently.

In simulation mode, all relevant time- and depth-based plots are available in the web browser on the user's demand and can suit the scenario at hand. During simulation, the user controls all the main drilling parameters at 1 Hertz time resolution. The simulator can also run in fast-forward which typically implies 10 times real time dependent on the client computer's performance and the network latency. The simulation process can be paused, perform a single step, or completed and shut down. A simulation can also run in sequential mode in which a predefined selection of setpoints is entered in a table. The user can then specify changes in any setpoint at a given time and construct a sequence. When executing a sequence, the user cannot interact with the simulator and change any of the setpoints. When the simulator runs in sequence mode, the simulation speed is typically around 15 times real time. The benefit by the sequential mode is that the user can run through the same scenario on different configurations and explore how a certain procedure will evolve for different wells, drilling fluids, etc.

Simulation results can be downloaded at any time during or after a simulation from the simulation GUI. For simplicity all results are downloaded to CSV format for easy processing in e.g. Excel. As for the configurations, the simulations can be shared. Unlike other desktop applications, a web enabled simulator allows other people from anywhere in the world to monitor and interfere with the simulation. By sending the URL a user can invite others to monitor an ongoing simulation or explore the results on completed simulations. The receiver of a URL can also take control and run a simulation on behalf of the owner of the simulation.

In addition to the browser interface, a WebAPI is available, making the simulator accessible to Matlab™ and Python, among others. These interfaces require special knowledge about the particular programming language and a description of these interfaces is not part of this paper.

8 Architecture and concept of operation

The system architecture of the OpenLab Drilling simulation system needs to account for performance and security. For this reason, the following system components are used:

- Webservice which serves the WebAPI
- Watchdog service per simulation server

- Relational database
- Clients: Webbrowser, Matlab client or Python client. Other clients may be developed by external users of the WebAPI.

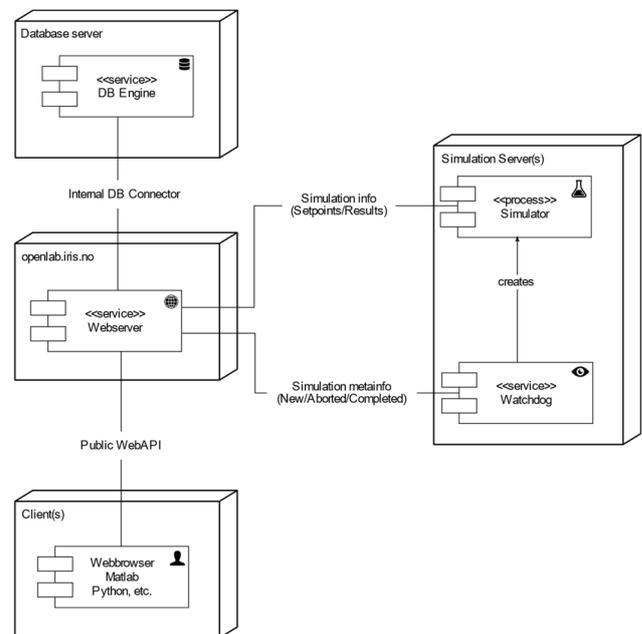


Figure 2: System communication architecture

To support a continuous development process, a Build server and a Deployment server has been established. By this approach a documented and consistent deployment is possible which makes it easy to release small improvements relatively often.

Immediate feedback for users is an important enabler of learning. Consequently, keeping the latency of the system low is important. The load on the system depends on the number of simulations which run concurrently and the number of users monitoring the results. The system works as follows: Watchdog processes monitor the WebAPI for new simulation requests. One watchdog then claims responsibility for this request and starts a simulation process. If the cores of the simulation machine are busy, the watchdog will respond slower to a simulation request and other watchdog processes with less utilized cores will pick up the simulation request. By this approach an automated load balancing is achieved without explicitly assigning work to different machines. In addition, a new simulation machine can be connected to the WebAPI and immediately assist when starting new simulations.

When a simulation process has been created and is initialized, it picks step requests from a queue and processes them as quickly as it can. The request contains the boundary conditions called setpoints. When the timestep is simulated the results are sent to the WebAPI which is responsible to persist them in the database.

The WebAPI is responsible for connecting user facing clients and simulation clients. It also persists

configurations, setpoints and results to make simulations reproducible. The WebAPI enforces authorization as well.

Since the system is exposed to different users with different trust levels, the architecture addresses security concerns. Malicious data may be used to subvert parts of the system and threaten assets that need to be protected. A main tool to mitigate threats is isolation of components using operating system tools such as process isolation and firewalls.

The most complex component of the system is the simulation which may crash when input data are invalid. This makes various exploits possible and a malicious user may subvert the simulation process for other purposes.

This risk is mitigated by validating data, and also by isolating each simulation process and allowing the communication of results only to the WebAPI.

9 The WeMod simulation model

The IRIS WeMod simulation model can simulate a wide range of downhole effects with high fidelity. The simulator consists of a set of fully integrated numerical models:

- A transient multi-phase flow model that solves mass and momentum balance to estimate pressure distribution inside the wellbore. (Lorentzen et al 2002 and 2014). The flow model has been improved, tested and validated over nearly 20 years. Dynamic effects such as pump accelerations, drill-string movements (surge-swab), gelling, and the presence of several drilling or formation fluids in the well are simulated. Flow of hydrocarbon to and from the formation is calculated by an integrated near-well reservoir model.
- A transient cuttings transport model that estimates the distribution of the cuttings inside the annulus and determines whether the drilled solids are in suspension in the drilling fluid or accumulate in cuttings bed (Cayeux et al 2013b). Transport of cuttings by bed erosion is also simulated.
- A torque and drag model that computes the tension and torque distribution along the drill-string (Yi et al 2004). In the current version a soft string model is used, however during 2018 a new transient torque and drag model will be available.
- A heat transfer model, which computes the temperature evolution inside the wellbore and in the formation near the well (Corre et al, 1984). Forced convection, heat conduction, and convective heat transfer are accounted for in the simulations.

The above-mentioned models are integrated in such a way that the hydraulic model uses the temperature profile generated by the heat transfer model to estimate pressures, densities and velocities in a numerical grid. The results are in turn used by the torque and drag model for buoyancy calculations and by the cuttings transport model for estimation of the transport capabilities. For density and rheology estimations, the hydraulic model uses the proportion of cuttings in suspension generated by the cuttings transport model.

10 Future extensions

OpenLab Drilling is currently providing simulated drilling data. The applications are hydrocarbon and geothermal drilling. However, for most users of OpenLab Drilling the access to real drilling data will be of high value. For analysts and engineers, the ability to test a prototype software at an early stage on real data will be of high importance and potentially affect the robustness and usability of the final product greatly. For students, access to real data will give valuable insight to the actual precision of the drilling data and provide a much more realistic view of the challenges the industry is facing when evolving towards drilling automation. A natural extension of OpenLab Drilling is therefore to offer an interface to providers and users of real drilling data.

It is also the intention to extend OpenLab to other disciplines than drilling, such as reservoir simulation and production data. The architecture developed in the OpenLab project is suitable for such use and can facilitate more knowledge transfer between the disciplines.

11 Project results

The drilling industry's increased focus on cost reduction has paved the way for automation in terms of process control rather than machine control. Process control in drilling requires technology developed to control a complex, non-linear and highly transient process. Fast and accurate simulation models that can predict downhole conditions accurately are therefore becoming more important. Until now, the user interface of simulators like WeMod and other high-fidelity simulators has been overwhelming for students and other novice users and erects a high barrier to onboard new users. With OpenLab Drilling an interface is made available for all who has a need for realistic but artificial drilling data provided by a high-fidelity simulator. The user interface allows users with very limited experience to drilling to take it into use. OpenLab Drilling is now being used as part of regular drilling courses at the University of Stavanger and the University of Calgary. In addition, Master and PhD students at other universities in Norway and abroad are using the simulator. Two of the major oil & gas operators in

Norway are using the software. During the first months when the software has been available in 2018 nearly 100 new users have taken the software into use.

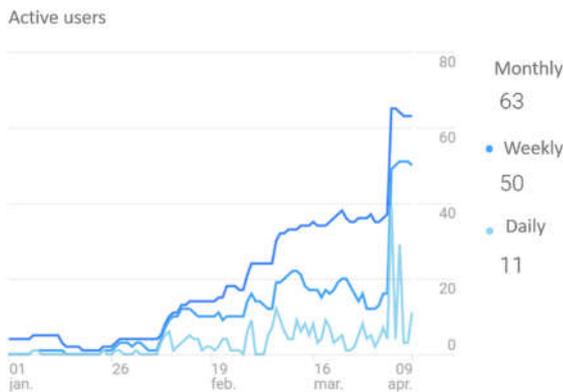


Figure 3: Registered user of OpenLab from January 1st to April 9th 2018. Ref: Google Analytics.

12 Conclusions

In the OpenLab Drilling project a drilling simulator on the web has been developed for students, educators and analysts with a low barrier of entry. The web enablement has made it possible to deploy the software to many users during the development phase to obtain valuable feedback at an early stage. Thanks to this, the user interface and prioritized functionally is continuously developing to meet the users' needs. The system is now deployed at universities in Norway and abroad, and to major oil & gas operators. Close collaboration with the user groups has been a key to succeed in developing a user interface that fits the need from students, researchers and lecturers.

Acknowledgements

The authors acknowledge the Research Council of Norway for financing OpenLab Drilling through the National Financing Initiative for Research Infrastructure (INFRASTRUKTUR). The authors are also very grateful to Miles AS in Stavanger, and CloudService AS for providing IT competence to the project.

References

Aarsnes, U.J.F, NTNU, Di Meglio, F., Mines ParisTech, Aamo, O.M., NTNU, Kaasa, G-O.Statoil. Fit-for-purpose Modeling for Automation of Underbalanced Drilling Operations. *Paper SPE/IADC 168955. SPE/IADC Managed Pressure Drilling & Underbalanced Operations Conference & Exhibition*, 8-9 April, Madrid, Spain, 2014.

Arnaut, A., Zoellner, P., and Thonhauser, G., TDE Thonhauser Data Engineering GmbH, Johnstone, N. "Intelligent Data Quality control of Real-time Rig Data." SPE 167437. *SPE Middle East Intelligent Energy*

Conference and Exhibition, 28-30 October, Manama, Bahrain, 2013.

- Ashok, P., Ambrus, A., Ramos, and D., Lutteringer, J., Sentinel Real time LLC, and Behounek, M., Yang, Y. L., Thetford, T., and Weaver, T., Apache Corp. "A Step by Step Approach to Improving Data Quality in Drilling Operations: Field Trials in North America." SPE 181076., *SPE Intelligent Energy International Conference and Exhibition*, 6-8 September, Aberdeen, Scotland, UK, 2016.
- Cayeux, E., Daireaux, B., Dvergsnes, E.W., and Leulseged, A., IRIS, Bruun, B., Statoil, and Herbert, M., ConocoPhillips Norway. "Advanced Drilling Simulation Environment for Testing New Drilling Automation Techniques". SPE 150941. *IADC/SPE Drilling Conference and Exhibition*, 6-8 March, San Diego, California, USA, 2012.
- Cayeux, E., Daireaux, B., Dvergsnes, E. W., IRIS. Florence, F., National Oilwell Varco. "Toward Drilling Automation: On the Necessity of Using Sensors That Relate to Physical Models". Paper SPE 163440. *SPE/IADC Drilling Conference*, 5-7 March, Amsterdam, The Netherlands, 2013.
- Cayeux, E., IRIS, Mesagan, T., Tanripada, S., Zidan, M., Statoil, Fjelde, K. K., University of Stavanger. "Real-Time Evaluation of Hole Cleaning Conditions Using a Transient Cuttings Transport Model". Paper SPE 163492. *SPE/IADC Drilling Conference*, 5-7 March, Amsterdam, The Netherlands, 2013.
- Cayeux, E. IRIS. "On the Importance of Boundary Conditions for Real-Time Transient Drill-String Mechanical Estimations". Paper SPE 189642. *IADC/SPE Drilling Conference and Exhibition*, 6-8 March, Fort Worth, Texas, USA, 2018.
- Corre, B., Eymard, R. and Gounet, A. "Numerical computation of Temperature Distribution in a wellbore while drilling". *Paper SPE 132080 presented at the SPE Annual Technical Conference and exhibition*, Houston, 16-19 September, 1984.
- Kaasa, G-O, Statoil, Stamnes, Ø.N., Aamo, O.M. and Imsland, L.S., Norwegian University of Science and Technology. Paper SPE 143097. *SPE Drilling & Completion Journal*. Vol. 27, Issue 01. March 2012.
- Lorentzen, R.J., RF-Rogaland Research. "Higher Order Numerical Methods and Use of Estimation Techniques to Improve Modeling of Two-Phase Flow in Pipelines and Wells". *Doctor thesis*, University of Bergen, April 2002.
- Lorentzen, R.J., Stordal, A. and Nævdal, G., SPE, IRIS, and Karlsen, H.A. and Skaug, H.J., University of Bergen, Department of Mathematics. Estimation of Production Rates with Transient Well-Flow Modeling and the Auxiliary Particle Filter. Paper SPE 165582, *SPE Journal*, pages 172-180, February 2014.
- Maidla, E., Maidla, W., Rigg, J., Crumrine, M., Wolf-Zoellner, P., ProNova - TDE Petroleum Data Solutions, Inc. "Drilling Analysis Using Big Data has been Misused and Abused". Paper SPE 189583. *IADC/SPE Drilling Conference and Exhibition*, 6-8 March, Fort Worth, Texas, USA, 2018.
- Petersen, J. and Rommetveit, R., RF-Rogaland Research, Tarr, B., Mobil Technology Company. "Kick with Lost Circulation Simulator, a Tool for Design of Complex Well

- Control Situations”. Paper SPE 49956. SPE Asia Pacific Oil & Gas Conference and Exhibition, Perth, Australia, 12-14 October 1998.
- Petersen, J., Rommetveit, R., Bjorkevoll, K.S., Froyen, J., SINTEF Petroleum Research. “A General Dynamic Model for Single and Multi-Phase Flow Operations during Drilling, Completion, Well Control and Intervention”. Paper SPE 114688. IADC/SPE Asia Pacific Drilling Technology Conference and Exhibition, 25-27 August, Jakarta, Indonesia, 2008.
- Rommetveit, R., Vefring, E.H. RF-Rogaland Research. «Comparison of Results from and Advanced Gas Kick Simulator with Surface and Downhole Data From Full Scale Gas Kick Experiments in an Inclined Well”. Paper SPE 22558. SPE Annual Technical Conference and Exhibition, Dalla, TX, USA, 6-9 October 1991.
- Yi, Y., Lund, B., Aas, B., He, X., Rommetveit, R., Lazaro, D., Stokka, S., RF-Rogaldn Research. Bottazzi, F., ENI. “An Advanced Coiled Tubing Simulator for Calculations of Mechanical and Flow Effects; Model Advancements and Full-Scale Verification Experiments.” Paper SPE 89455, *SPE/ICoTA Coiled Tubing Conference and Exhibition*, Houston, TX, USA, 23-24 March 2004.
- Ødegard, S.I, eDrilling, Risvik, B.T., Statoil, Bjorkevoll, K.S., SINTEF, Mehus, Ø., Oiltec Solutions, Rommetveit, R and Svendsen, M., eDrilling. “Advanced Dynamic Training Simulator for Drilling as well As Related Experience from Training of Drilling Teams with Focus on Realistic Downhole Feedback”. Paper SPE 163510. *SPE/IADC Drilling Conference*, 5-7 March, Amsterdam, The Netherlands, 2013.

Machine Learning in Python for Weather Forecast based on Freely Available Weather Data

E. B. Abrahamsen, O. M. Brastein, B. Lie*

Department of Electrical Engineering, Information Technology and Cybernetics
University of South-Eastern Norway, N-3918 Porsgrunn,
*(Bernt.Lie@usn.no)

Abstract

Forecasting weather conditions is important for, e.g., operation of hydro power plants and for flood management. Mechanistic models are known to be computationally demanding. Hence, it is of interest to develop models that can predict weather conditions faster than traditional meteorological models. The field of machine learning has received much interest from the scientific community. Due to its applicability in a variety of fields, it is of interest to study whether an artificial neural network can be a good candidate for prediction of weather conditions in combination with large data sets. The availability of meteorological data from multiple online sources is an advantage. In order to simplify the retrieval of data, a Python API to read meteorological data has been developed, and ANN models have been developed using TensorFlow.

Keywords: Weather prediction, Auto-regressive neural networks, Meteorological data

1 Introduction

1.1 Background

The forecasting of weather conditions and in particular the prediction of precipitation is important for hydro-power operation and flood management. Mechanistic meteorology prediction models based on 3D CFD/Navier Stokes equations (Thibault and Senocak, 2009) is extremely demanding wrt. computing power. Generating a 14 day weather forecast can easily take 12 hours even on fast computers. Machine Learning (ML), Big Data, and use of Internet of Things (IoT) are receiving increased interest from the industry. It is well known that large amounts of data coupled with novel ML methods can produce results on par with traditional physics based models.

Due to an interest in weather monitoring in the general public, today a large number of weather stations are connected to the internet, and are thus available as cheap, distributed sensors. Additionally, several organizations that are involved in collection of meteorological data offer online data servers with accessible Application Programming Interfaces (API) such as the HTTP based GET/REST protocols. In order to simplify experimentation with sev-

eral sources of meteorological data it is of interest to develop a unified API, hence facilitating the extraction of data from different sources. With large quantities of data, both historical and current measurements, it is an attractive solution to use machine learning in order to predict weather conditions based on these relatively simple data sources. Using a large amount of data together with novel machine learning algorithms can then compensate for lack of complex meteorological models and yield usable forecasts with less computing time.

Simple ML models would base predictions on auto regressive (AR) structures, where, say the current temperature in a location is correlated with several past temperatures in the same location. In a slightly more advanced auto regressive structure, a set of properties, e.g., the tuple (temperature, humidity, and precipitation) could be correlated with several past values of the same tuple. An even more advanced structure is the auto regressive structure with exogenous input (ARX). In such a model, the current (local) set of properties is correlated with both past values of the same (local) set, but also with other values from the same location or values of the same properties from other locations at *current time*. Finally, in ARMAX structures, exogenous inputs at *different times* (= moving average) are used in the correlation.

1.2 Previous Work

(Hayati and Mohebi, 2007) studied multi layer perceptron (MLP) neural networks trained and tested on ten years of meteorological data (1996-2006). The network structure consisted of three layers with a logistic sigmoid activation function in hidden layers and linear functions in the output layer. Seven weather variables were used in the study: dry temperature, wet temperature, wind speed, humidity, pressure, sunshine, and radiation. The inputs were normalized and used to predict dry air temperature in intervals of 3 hours for a total of 8 predictions pr day. The error was calculated using mean absolute error (MAE). In (Smith et al., 2006), the authors focused on developing artificial neural network (ANN) models to forecast air temperature at hourly intervals from one to 12 hours ahead. Thirty models were calibrated for each interval, in order to study the effect of randomized initial weights on test set prediction accuracy. The network structure consisted of three

fully connected hidden layers that used Gaussian, Gaussian complement, and hyperbolic tangent activation functions. The input data was linearly transformed to the range [0.1, 0.9] and consisted of five weather variables: temperature, relative humidity, wind speed, solar radiation and rainfall. Later, seasonal variables were introduced as inputs which improved model accuracy. A recent machine learning (ML) approach, based on a hybrid model including both ANNs, decision trees (DT), and Gaussian process modeling (GP) is presented in (Grover et al., 2015). They concluded that while previous attempts at weather modeling using ML have had limited success, their hybrid model approach surpasses the NOAA¹ benchmarks. A review on the use of machine learning methods for weather prediction is presented in (Chauhan and Thakur, 2014).

Meteorological data from a number of sources are available today, e.g., from the Norwegian Meteorological Institute data service `frost.met.no`, and from *Netatmo*². These and others are potential “Big Data” sources.

A number of high quality ML tools have become available the last decade, also as packages in free computer languages such as Python. One possible ML tool which runs in Python is Google’s TensorFlow. AR, ARX, and ARMAX models for linear systems are routinely used in system identification, e.g. (Ljung, 2002, 1999; Johansson, 1993).

1.3 Overview of Paper

Weather prediction is a convenient case for studying machine learning. By developing APIs for accessing available data from meteorological institutes and other weather stations, this gives access to an abundance of data. Weather data is something that most people can relate to in their daily life, but is also important for energy systems, flood prediction, etc. Good physical based meteorological models are available, which makes it easy to compare the quality of machine learning models. In this paper, we have focused on a new Python API for collecting weather data, and given simple, introductory examples of how such data can be used in machine learning. Weather data from `frost.met.no` have been collected using a newly developed Python API. These data have been used to train and tune several auto-regressive artificial neural networks (AR-ANN) by using TensorFlow from Python. The resulting models have been used to predict the temperature in Porsgrunn with prediction horizons of 1, 3, 6, and 12 hours. The example ANN is then extended with precipitation data and compared to the initial AR-ANN..

This paper is organized as follows; Section 1 provides the necessary background information, and a short review of previous work relevant to the project. Section 2 gives some theoretical details regarding ANNs and the developed APIs for collection of weather data. Section 3 discusses the obtained results, before the work is concluded

¹National Oceanic and Atmospheric Administration, <http://www.noaa.gov>

²<https://www.netatmo.com>

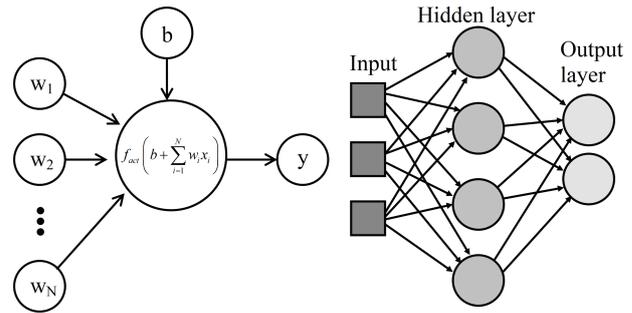


Figure 1. Illustration of a single neuron (left) and an example of an artificial neural network (right).

in Section 4 together with suggestions for future work.

2 Materials and Methods

2.1 Artificial Neural Networks

Artificial neural networks (ANN) have existed in various forms since the 1940s (McCulloch and Pitts, 1943; Goodfellow et al., 2016), but have received renewed interest in recent years (Goodfellow et al., 2016). An ANN is a collection of neurons, which are small computational units that superficially mimic the way neurons work in nature. A single neuron is simply a weighted sum of a set of inputs, plus a bias, with an applied activation function, Fig. 1 (left).

A non-linear activation function $f_{act}(\cdot)$ is important for success in applying ANNs, otherwise the resulting model output is simply a linear combination of the inputs. The equation for a single neuron can be written as:

$$y_k = f_{act}(b + x_i w_i) \quad (1)$$

The power of ANNs comes from connecting many neurons together in a network. The simplest network structure is a feed forward network, as shown in Figure 1 (right). Neurons are connected in simple layered structures where the inputs of each neuron are connected to all the outputs of the previous layer. If we describe the inputs x_i and weights w_i in matrix form, we can write a whole layer of neurons as:

$$y = f_{act}(Wx) \quad (2)$$

where the bias is included as $w_0 = b$ by adding an artificial constant input $x_0 = 1$.

A feed forward ANN is built by connecting multiple layers together. The inputs to the network are connected to the inputs of the first hidden layer. The first hidden layer can then be connected to more hidden layers. The last hidden layer connects to the output layer. The output of the ANN is given by this output layer. We can then write a single non-linear matrix equation for the whole network. An example equation for an ANN with three hidden layers is:

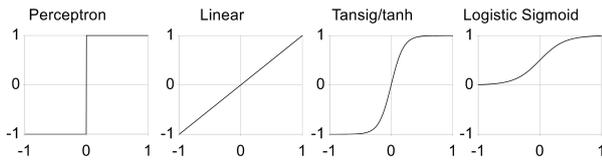


Figure 2. Typical activation functions used in ANNs.

$$y^{\text{out}} = f_{\text{act}}^{\text{out}} \left(W^{\text{out}} f_{\text{act}}^{(2)} \left(W^{(2)} f_{\text{act}}^{(1)} \left(W^{(1)} x \right) \right) \right) \quad (3)$$

Equation (3) shows that an ANN is simply a non-linear matrix equation with a large number of coefficients. Each $W^{(j)}$ matrix can be large, thus allowing the ANN model to fit complex non-linear systems. The descriptive power that comes from this complexity is the reason why ANN models is able to adapt to such a large variety of systems.

Many choices for activation function are possible, as shown in Figure 2. Initially in the history of ANNs, a simple sign operator $y = \text{sgn}(x)$ was often used as the activation function. These simple neurons were called *perceptrons*. The perceptron term survives to this day in the ANN community; indeed, a deep learning network is often referred to as a Multi Layer Perceptron (MLP). In this work, both tanh and logistic sigmoid $\frac{1}{1+\exp(-x)}$ activation functions (Goodfellow et al., 2016) are used. Training of an ANN is essentially a parameter optimization of the network weights such that the output of the network minimizes a chosen loss function. However, since an ANN model has a high number of parameters to optimize, there are a large number of local solutions. The optimization of the training weights is performed iteratively, where each training iteration is called an *epoch* (Goodfellow et al., 2016).

Globally optimal solutions for ANN training are in general considered very hard to find (Goodfellow et al., 2016; Bishop, 2013). Instead, the focus is on finding a solution that is “good enough”. During training, there is always a risk that a particular training session can get stuck in a local minimum which is far from optimal. If the same ANN hyper-parameter configuration is trained multiple times, it is usually clear if one or more iterations are indeed giving sub-optimal performance due to this local minima problem. It is also important to note that there has been much research in improving training performance in the presence of local minima. Hence, there exists a large number of training algorithms which seek to improve ANN training performance. For more details on the development of ANNs see, e.g., (Goodfellow et al., 2016).

A model with high descriptive power is prone to over-fitting. The term over-fitting is used in empirical modeling to describe what happens when a model adapts to random variations in the training set which does not generalize well to new data. This effect is apparent in all forms of empirical modeling, from simple curve fitting to complex ANNs. Since the ANNs have such a large num-

ber of coefficients, the over-fitting problem is particularly important. The simplest way to reduce the risk of over-fitting is to increase the amount of training data, either by collecting more data or artificially creating more training data through some form of transformation on the original data (Ciresan et al., 2010). Another way to reduce the risk of over-fitting is to apply a regularization method (Goodfellow et al., 2016; Kuhn and Johnson, 2013). The subject of regularization is a research field in itself, which involves methods that prevents the training algorithms for ANNs from adapting to random variations in the training data. The simplest form of regularization is to have a large amount of data. Since this work is based on retrieving weather data from online databases, the cost of obtaining data is relatively low, hence a large amount of training data is readily available. One common regularization method is the use of weight decay (Goodfellow et al., 2016). Weight decay adds a penalty to the loss function which is proportional to the training weights w_i themselves. This forces the weights to stay “small” (Goodfellow et al., 2016):

$$J_{\min} = \alpha \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2 + (1 - \alpha) \sum_{i=1}^M w_i^T w_i \quad (4)$$

In Eq. (4) the loss measure is the mean square error (mse) between the predictions \hat{y}_i and the references/observations y_i . The L_2 (2-norm) weight decay penalty is the last term in the J_{\min} loss function. A hyper parameter α is used to decide how strong the weight decay regularization should be.

2.2 Timeseries modeling

A common method for modeling discrete timeseries data is the use of auto-regressive models with exogenous inputs (ARX). Using the time-shift operator q^{-k} to indicate a quantity being shifted k time-steps back in time, these models can be expressed on the form

$$y_k (1 + a_1 q^{-1} + \dots + a_n q^{-n}) = u_k (b_1 q^{-1} + \dots + b_m q^{-m}) \quad (5)$$

That is, the output at time k is a function of both the inputs and the output at previous times. If all b_i coefficients are zero, e.g., there are no exogenous inputs, the model is called an AR model (Ljung, 1999). A nonlinear ARX model can be formulated as

$$y_k = f(y_{k-1}, \dots, y_{k-n}, u_{k-1}, \dots, u_{k-m}) \quad (6)$$

Traditional ARX models are linear models as illustrated in Eq. (5), thus $f(\cdot)$ in Eq. (6) forms a linear combination of past inputs and outputs. When ANNs are applied to timeseries modeling, function $f(\cdot)$ in Eq. (6) is replaced by the ANN, such that:

$$y_k = \text{ANN}(y_{k-1}, \dots, y_{k-n}, u_{k-1}, \dots, u_{k-m}) \quad (7)$$

Hence, the term auto-regressive neural network (AR-ANN) is used to denote an ANN which predicts a time-series variable based on previous measurements of the same variable, e.g. the inputs and outputs to the network are the same variables but at different times. Similarly, an ARX-NN is a network which in addition to previous measurements of the output variable also has additional measurements as its inputs.

2.3 Python API for Data Collection

A Python API wrapper is an easy way to obtain free weather data from APIs and open data. A wrapper was designed to support multiple weather data suppliers, so it is possible to add more suppliers in the future. The API does not support the use of multiple suppliers at the same time. Currently the Norwegian Meteorological Institute data service `frost.met.no` and *Netatmo* are supported. The API will request hourly data for a given date, either at the station nearest to the specified latitude and longitude coordinate or within a specified rectangle as specified in kilometers centered on a given latitude and longitude coordinate. The wrapper uses `HTTP GET` requests to obtain the data from the data suppliers and returns a list where each element is a 3 item list with `stationID`, `timestamp`, and measured value. The returned data can then be saved to a file or database.

2.4 Experimental Data

The data used in this work was collected from `frost.met.no` using the mentioned Python API. The data consists of hourly temperature and precipitation measurements in the period `01.01.2016 T00:00` to `31.12.2017 T23:00` from weather station `SN30255` at latitude: `59.091` and longitude: `9.66` in Porsgrunn, Norway. Due to downtime on the station the first month of 2017, data starts at `01.02` in the 2017 part of the data set, hence for consistency the first month of 2016 was also removed. For the first experiment only temperature data was used. In the second experiment, temperature data and precipitation data was used. For all experiments, the data was split into three independent sets: 60% used for training, 20% for hyper parameter tuning (validation), and 20% for testing the prediction accuracy of the models.

3 Results and Discussion

The goal of this work is to predict the temperature using an artificial neural network (ANN). Four cases have been studied in both experiments, using prediction horizons of 1, 3, 6 and 12 hours. In each experiment, four separate models were created, one for each case. The input data was normalized, and the output was denormalized to get predictions in degrees Celsius. To test the different models, 48 consecutive hours is set to be predicted by the different models.

All ANNs use rectified linear unit (ReLU) as the activation function in the hidden layers and a linear activation function in the output layer.

3.1 Experiment 1 - AR Neural Network

The first experiment used only temperature data. This constitutes an auto-regressive neural network (AR-NN) model. Figure 3 shows the results of predicting the test set using each of the four models, together with the reference measurements.

There is a sudden change in measured temperature in the time interval 36 to 42 in the test set. The models with prediction horizons 1 and 3 hours show an oscillating response to this rapid change. However, they are still closer to measured data than the models with longer prediction horizons. At time 8 to 20, the 12 hour model is significantly poorer than the 1, 3, and 6 hour models. A plausible reason for this might be that the models respond to data given and the 12 hour model uses data that is 12 hours prior to the measurement. So, at time 7 the 12 hour model started increasing, and at time 15 it flattens out, probably because the algorithm used the data at time 3 and measured that the data started to turn, thus a prediction model should “slow down”. All the models show significant deterioration in predictions when the temperature changes rapidly.

Table 1 shows the hyper-parameters with test error summarized. The error is calculated from normalized test data, hence the error is *not* presented in units of Celsius. The number of layers for the 12 hour prediction model is two times that in the other models. Further, the regression horizon (i.e., the amount of past data points used in the predictions) is 169 hours (7 days) for the 12 hour model. Compared to a regression horizon of 48 hours for the 6 hour prediction model, this is a significant difference. The learning rate is also significantly lower for the 12 hour prediction model. The low number of epochs and low learning rate was found necessary for the 6 and 12 hour prediction models to generalize properly and avoid over-fitting. With these hyper-parameters, training is slowed down.

Figure 4 shows the errors in degree Celsius for the predictions on the test set. These results are calculated as the difference between measured value and predicted value. Hence, the error that is negative is overshooting and error that is positive is undershooting. Observe from Figure 4, that due to the rapid change in measured data in time interval 36 to 42, the error is significantly higher for all the models.

Table 2 shows a selection of tuning of the ANN hyper-parameters for 1 hour prediction. At first, a 3 hour regression horizon was used on the assumption that a human observer would likely be able to predict the temperature one hour ahead of time based on a small amount of data. Later, a 24 hour regression horizon was tested against the same structure. The longer regression horizon was found to improve prediction performance on the validation set. Selecting a good neural network structure is important. The choice of network structure depends on the type of application. According to (Heaton, 2017), one hidden layer can approximate any continuous function, while two hidden

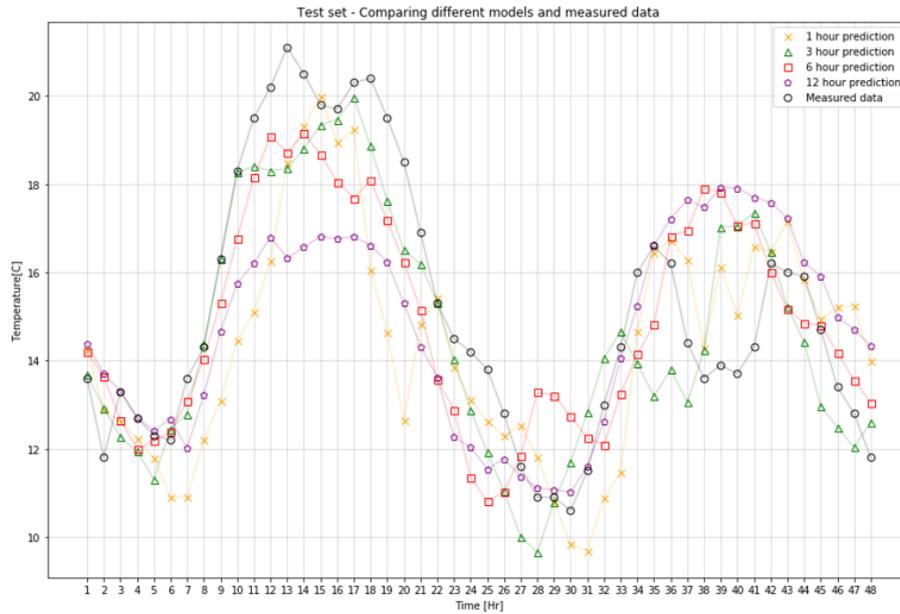


Figure 3. Shows the measurements and outputs of the 4 different models.

Table 1. Hyper-parameters for each model.

<i>Model</i>	<i>Test error</i>	<i>Epochs</i>	<i>Layer structure</i>	<i>Regression horizon</i>	<i>Learning rate</i>
1 hour	0.0101	810	17, 12	24	0.01
3 hour	0.0318	150	30, 20	48	0.01
6 hour	0.0608	1000	38, 24	48	0.001
12 hour	0.0894	500	112, 75, 50, 34	169	0.0001

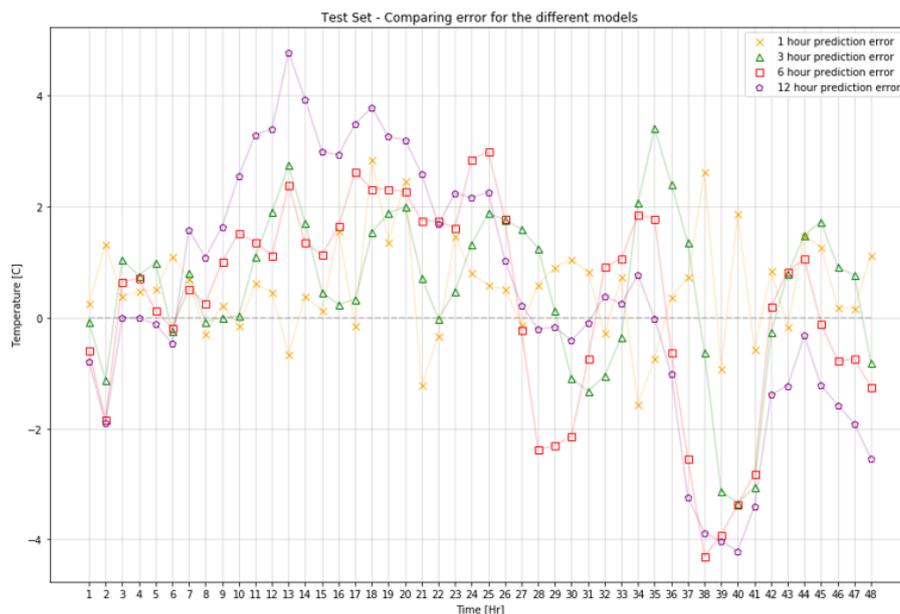


Figure 4. Deviation in degrees Celsius each point for each AR model was off for the 48 hours predicted. The deviation is calculated as the difference between measured value and predicted value, thus a negative error implies overshooting and a positive error implies undershooting.

layers can approximate any arbitrary function. Due to having more descriptive power, the two layer networks also tends to adapt faster to the patterns in the training data, thus learning the input-output relationships faster. Hence, from one to three hidden layers were tested as shown in Table 2. Three layers give approximately the same loss as two layers, hence a two-layer model is chosen. The width of each layer was chosen according to the following three rules suggested by (Heaton, 2017):

1. The number of hidden units in each layer should be between the number of inputs and number of outputs.
2. The number of hidden units in each layer should be $\frac{2}{3} \times (\text{number of inputs} + \text{number of outputs})$.
3. The total number of hidden neurons in all layers should be less than $2 \times (\text{number of inputs})$.

3.2 Experiment 2 - ARX Neural Network

In Experiment 2, the neural network input is extended to include precipitation data. Observe that the error in the time interval 36 to 42 is reduced for the shorter models (1 and 3 hour). The 1 hour model predicts the test set with satisfactory accuracy. However the 3 hour model still has significant prediction errors, in particular at higher temperatures. Observe that, as expected, the prediction accuracy of each model deteriorates with longer prediction horizons. This is particularly apparent in the time interval 8 to 20.

Table 3 shows the hyper-parameters with test set prediction errors. The test errors on the ARX models are higher than on the AR models. This is unexpected and the reason is most likely poor tuning of the models. Two hidden layers were used for all four models, as suggested by (Heaton, 2017). The hidden structures depend largely on the choice of regression horizon, hence the structures for each model is similar, except for the 6 hour model, which achieved better tuning with hidden layer structure (33,12) instead of (17,12).

The error for each model predicted on the 48 hour test set is shown in Figure 5. Observe that the error due to the mentioned rapid change in measured data at time 36 to 42 in the 1 hour prediction model is reduced.

4 Conclusions and Future Work

In this work, artificial neural networks are used to predict temperature. Four separate models were trained to predict the temperature 1, 3, 6, and 12 hours ahead. In the first experiment, only temperature was used as input to the networks. This constitutes an auto-regressive neural network (AR-NN). In the second experiment, precipitation data was introduced into the network, forming an auto-regressive neural network with exogenous input (ARX-NN). After extensive tuning of hyper parameters for all eight models, the prediction results of the models were compared. Introducing precipitation as an input in the

ARX model was shown to slightly improve the prediction performance. Hence, it may be interesting to extend the model with other inputs. Mainly, it is of interest to study whether introduction of data from other geographical locations can improve the prediction results. Based on knowledge of how the jet stream moves and influences the weather, together with local pressure variations, it would be natural to add weather information from, e.g., Kristiansand, Oslo, etc. as exogenous inputs. This will be a topic for future research.

References

- C.M. Bishop. *Pattern Recognition and Machine Learning: All "just the Facts 101" Material*. Information science and statistics. Springer, 2013. ISBN 9788132209065. URL <https://books.google.no/books?id=HL4HrgEACAAJ>.
- Divya Chauhan and Jawahar Thakur. Data mining techniques for weather prediction: A review. *International Journal on Recent and Innovation Trends in Computing and Communication*, 2(8):2184–2189, 2014.
- Dan Claudiu Ciresan, Ueli Meier, Luca Maria Gambardella, and Jürgen Schmidhuber. Deep big simple neural nets excel on handwritten digit recognition, 2010. *Cited on*, 80, 2010.
- Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.
- Aditya Grover, Ashish Kapoor, and Eric Horvitz. A deep hybrid model for weather forecasting. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 379–386. ACM, 2015.
- Mohsen Hayati and Zahra Mohebi. Application of artificial neural networks for temperature forecasting. *World Academy of Science, Engineering and Technology*, 28(2):275–279, 2007.
- Jeff Heaton. The number of hidden layers, 2017. URL <http://www.heatonresearch.com/2017/06/01/hidden-layers.html>.
- R. Johansson. *System Modeling and Identification*. Information and system sciences series. Prentice Hall, 1993. ISBN 9780134823089. URL <https://books.google.no/books?id=FZ7gAAAAMAAJ>.
- Max Kuhn and Kjell Johnson. *Applied predictive modeling*, volume 26. Springer, 2013.
- L. Ljung. *System Identification: Theory for the User*. Prentice Hall information and system sciences series. Prentice Hall PTR, 1999. ISBN 9780136566953. URL <https://books.google.no/books?id=nHFoQgAACAAJ>.
- Lennart Ljung. Prediction error estimation methods. *Circuits, Systems and Signal Processing*, 21(1):11–21, Jan 2002. ISSN 1531-5878. doi:10.1007/BF01211648. URL <https://doi.org/10.1007/BF01211648>.
- Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

Table 2. 1 hour prediction, selected hyper-parameter search.

<i>Exp. No.</i>	<i>Epochs</i>	<i>Regression horizon</i>	<i>Layer structure</i>	<i>Learning rate</i>	<i>Training set loss</i>	<i>Validation set loss</i>
1	1000	3	3	0.001	0.0144	0.0109
2	1000	24	3	0.001	0.0125	0.0105
3	1000	24	17	0.01	0.0116	0.0101
4	1000	24	17,12	0.01	0.0113	0.0100
5	1000	24	17,12,9	0.01	0.0113	0.0101
6	810	24	17,12	0.01	0.0113	0.0100

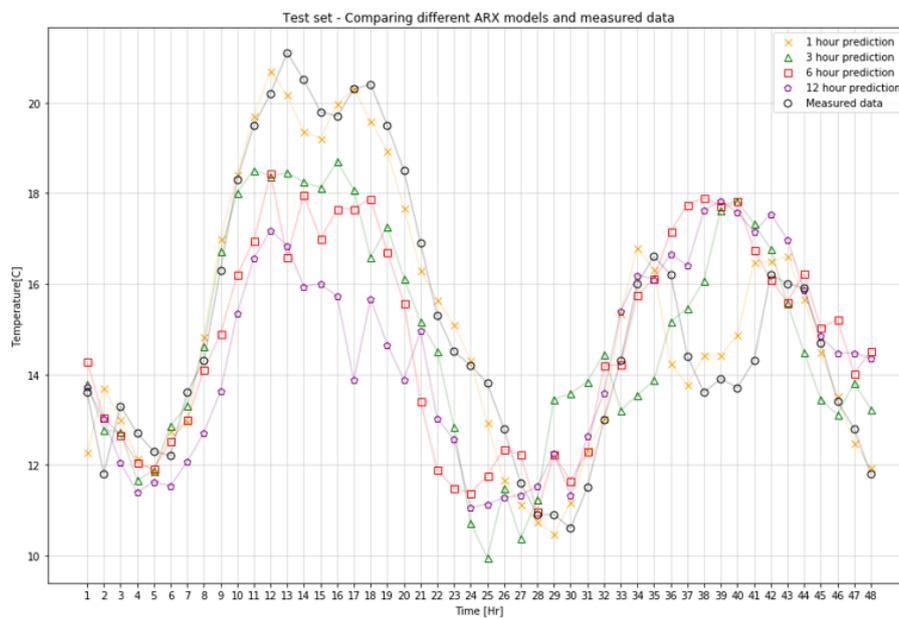


Figure 5. Measured value and the outputs of 4 different ARX models.

Table 3. Hyper-parameters for each ARX model.

<i>Model</i>	<i>Test error</i>	<i>Epochs</i>	<i>Layer structure</i>	<i>Regression horizon</i>	<i>Learning rate</i>
1 hour	0.0133	500	17, 12	24	0.01
3 hour	0.0653	500	33, 23	48	0.001
6 hour	0.0946	500	17, 12	48	0.01
12 hour	0.0991	500	17, 12	24	0.001

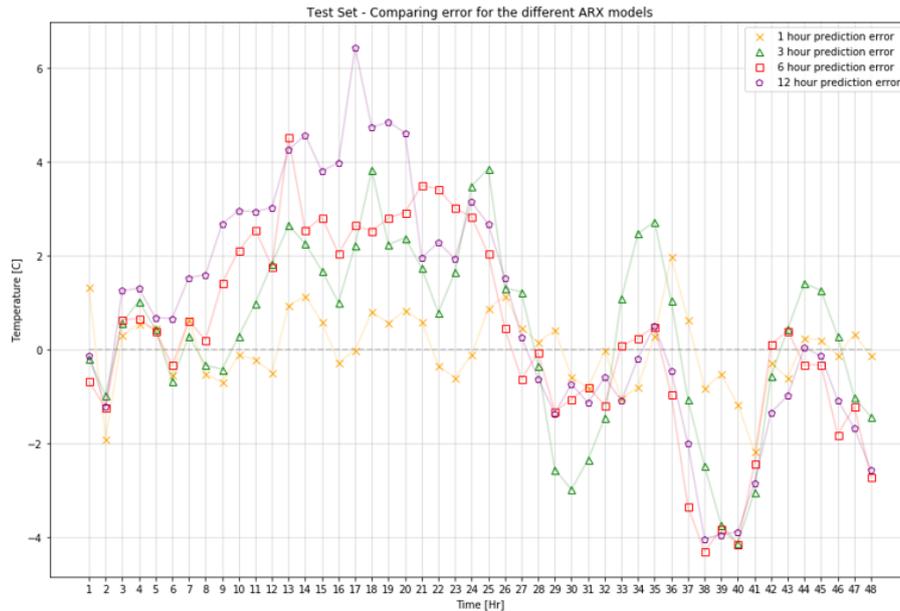


Figure 6. Error in Celsius for the ARX model over the 48 hour test set. The error is calculated as the difference between measured value and predicted value.

Brian A Smith, Ronald W McClendon, and Gerrit Hoogenboom. Improving air temperature prediction with artificial neural networks. *International Journal of Computational Intelligence*, 3(3):179–186, 2006.

Julien Thibault and Inanc Senocak. Cuda implementation of a navier-stokes solver on multi-gpu desktop platforms for incompressible flows. In *47th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition*, page 758, 2009.

Comparison of Simulation Tools for Dynamic Models

Sveinung M. Sund¹, Marianne Plouvier², and Bernt Lie¹

¹University of South-Eastern Norway, Bernt.Lie@usn.no, ²IMT Mines Albi, France

Abstract

Macroscopic models are used extensively in process engineering, and can often be posed as DAE (Differential Algebraic Equation) models. Three generic tools for solving such DAEs are compared: OpenModelica, Julia, and MATLAB. To make the comparison concrete, a simple non-linear process model from the literature was extended by removing simplifying assumptions; the more complex model was posed as DAEs. Some implementation details of DAE models in OpenModelica, Julia, and MATLAB are given. Selected simulation results are given, with resulting execution time. The three tools gave identical simulation results. The tools are then compared wrt. cost, ease of use, documentation, numeric quality, Eco-system, and possibility for reuse of models/library. Overall, Julia appears may appear as the best choice. However, Modelica is found to be easier to use, so an ideal solution would probably be some tight integration of Modelica with Julia. *Keywords:* process modeling, dynamics, DAEs, simulation tool.

1 Introduction

1.1 Background

Macroscopic models are used extensively in process engineering, and can in general be posed as a set of differential equations stemming from conservation/balance laws with the addition of algebraic relations describing thermodynamics, transport laws, reaction engineering, etc. The result is a set of Differential Algebraic Equations (DAE) with a relatively simple structure. Such DAE models can be transformed/reduced to Ordinary Differential Equations (ODEs) with some effort. The resulting ODEs are often simpler than the DAEs, thus more efficient solvers may be available. At the same time, reducing DAEs to ODEs also eliminates many variable which may be of interest to study.

In education, as well as in engineering practice, it is important to choose simulation tools for doing experiments on such models. Important aspects are, e.g.: cost, ease of use, how well the tool is documented, numeric quality, Eco-system (packages/functions for plotting, analysis, random numbers, control packages, access to time series database, etc.), possibility of reuse of the model (model library, extract information for control design, etc.).

Dynamic models are used for design and operation of systems, and it is therefore of interest to fit such mod-

els to experimental data, to reuse such models for process design, stability analysis, control design, estimator design, etc. The various simulation tools have different support for and capabilities wrt. analysis and synthesis. An important question is then: is it wise to aim at a one-language/tool solution, or is it better to use different tools for simulation and design/synthesis as long as the languages can interact?

1.2 Previous Work

Specialized process engineering tools are well developed, e.g., from AspenTech¹, Process Systems Enterprise², etc. More generic tools are also popular, such as Modelica based tools (OpenModelica³, JModelica.org⁴, Dymola⁵, etc.), other high level tools (DAE Tools⁶, APMonitor⁷), script based tools (MATLAB⁸, Python⁹, Julia¹⁰, etc.), and computer algebra systems (Mathematica¹¹, Maple¹², etc.). The possibility to integrate OpenModelica with Python to enable more extended analysis was studied in (Lie et al., 2016).

A simple model of a nonlinear, open loop unstable reactor is given in (Seborg et al., 2011), and is used in various control studies (Henson and Seborg, 1997). In a student project/exam paper at University College of South-east Norway¹³, this model has been extended into a DAE model by removing some of the simplifying assumptions. Both of these models are suitable for testing basic capabilities of simulation tools for handling small-scale models.

1.3 Organization of Paper

In this paper, we compare the suitability of selected tools (OpenModelica, Julia, MATLAB) for solving macroscopic DAE models in process engineering. The paper

¹www.aspentech.com

²www.psenderprise.com/products/gproms

³<https://openmodelica.org/>

⁴www.jmodelica.org

⁵www.3ds.com/products-services/catia/products/dymola

⁶<http://daetools.com>

⁷<http://apmonitor.com>

⁸<https://mathworks.com>

⁹www.anaconda.com

¹⁰<https://julialang.org>

¹¹www.wolfram.com

¹²www.maplesoft.com

¹³Course FM1015 Modelling of Dynamic Systems, www.usn.no/academics/study-and-subjectplans/#/subjects/FM1015_2018H_1

is organized as follows. In Section 2, an overview of a suitable DAE model structure is given, with details of the case study model. In Section 3, some general characteristics and specific implementation details of the models in the chosen languages are given. In Section 4, selected simulation results are given, together with some resulting execution times. In Section 5, the findings are discussed, before some conclusions are drawn in Section 6. An appendix gives some model details.

2 Model Overview

2.1 Mechanistic Model Structure

Mechanistic models at the macroscopic level can typically be described using balance laws of form

$$\frac{dx}{dt} = f(x, z, u; \theta) \quad (1)$$

where differential variable x is the “balanced” property (amount, momentum, energy), while z is some auxiliary algebraic variable, u is some input variable and θ is some model constant/parameter. Such models need to be complemented with transport/thermodynamic/reaction laws (constitutive laws) of form

$$0 = g(x, z, u; \theta). \quad (2)$$

Together, Eqs. 1 and 2 form a set of differential-algebraic equations, DAE. The set of differential variables and algebraic variables is referred to as the model *descriptor*, (x, z) . Such DAE models are conceptually simple to formulate, but may contain relatively many descriptor variables and many parameters θ .

DAE models can be manipulated into sets of ordinary differential equations, ODE, of form

$$\frac{d\tilde{x}}{dt} = \tilde{f}(\tilde{x}, u; \tilde{\theta}) \quad (3)$$

where \tilde{x} is a state of the system. For DAE models of index 1, $\dim x \geq \dim \tilde{x}$, while for DAE models of index larger than 1, it is possible that $\dim x \leq \dim \tilde{x}$. Normally $\dim \theta \geq \dim \tilde{\theta}$.

Tools for analyzing ODE models (solvers, stability concepts, etc.) are better developed than tools for DAE models. ODE models are smaller and faster to solve, but manual model simplification may introduce (model) errors. DAE models, on the other hand, hold more information (the algebraic variables), and are simpler to formulate, so there are important advantages in keeping models in DAE form.

It is of interest to compare how easy it is to formulate DAE models and solve them in popular modeling/simulation tools. For such a comparison, it is convenient to introduce a case study.

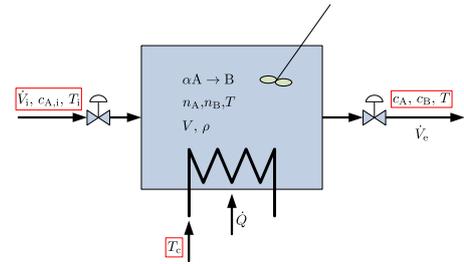


Figure 1. Cooled liquid reactor with reaction $aA \rightarrow B$.

2.2 Process Overview

We consider a liquid reactor of constant volume V , with influent volumetric flow rate \dot{V}_i , influent concentration $c_{A,i}$ of reactant A carried via an inert solvent S,¹⁴ and influent temperature T_i , Fig. 1.

It is of interest to convert species A into species B through an exothermic reaction



the products are carried out of the reactor via solvent S. To keep control of reactor temperature T , heat rate \dot{Q} is added by flowing a liquid at temperature T_c through the tube side of a coil/heat exchanger. With a high flow rate of the cooling liquid, T_c is constant through the heat exchanger, and the heat rate can be described as

$$\dot{Q} = \mathcal{U}A(T_c - T). \quad (5)$$

where $\mathcal{U}A$ is a parameter. If $T_c < T$, $\dot{Q} < 0$ and the reactor is cooled.

The rate of reaction r is given as

$$r = kc_A^a \quad (6)$$

where a is the reaction order and k is given by the Arrhenius expression

$$k = k_0 \exp\left(-\frac{E/R}{T}\right). \quad (7)$$

The operation of the reactor is influenced by inputs \dot{V}_i , $c_{A,i}$, T_i and T_c , and it is of interest to study how these influence the outputs c_A , c_B and T .

Although the case study has inputs $u = (\dot{V}_i, c_{A,i}, T_i, T_c)$ and outputs $y = (c_A, c_B, T)$, in a control problem one may choose to control the temperature $y = T$ by manipulating the input $u = T_c$. In that case, the additional inputs $(\dot{V}_i, c_{A,i}, T_i)$ may be considered disturbances.

For practical purposes, the cooling liquid temperature $T_c \in [4, 90]^\circ\text{C}$ or so.¹⁵ In reality, T_c is not directly controllable. Instead, an influent “cooling” temperature $T_{c,i}$ could be varied while temperature T_c in the heat exchanger is varied via manipulating the cooling liquid flow rate.

¹⁴Inert implies non-reacting.

¹⁵If vaporized, higher temperatures may be achieved.

2.3 Balance Equations

For the given process, the following balance equations are relevant,

$$\frac{dm_S}{dt} = \dot{m}_{S,i} - \dot{m}_{S,e} \quad (8)$$

$$\frac{dn_A}{dt} = \dot{n}_{A,i} - \dot{n}_{A,e} + \dot{n}_{A,g} \quad (9)$$

$$\frac{dn_B}{dt} = \dot{n}_{B,i} - \dot{n}_{B,e} + \dot{n}_{B,g} \quad (10)$$

$$\frac{dU}{dt} = \dot{H}_i - \dot{H}_e + \dot{Q} \quad (11)$$

where \dot{x} indicates flow of variable x (in general, $\dot{x} \neq \frac{dx}{dt}$), m is mass, n is amount in mole, U is internal energy, and H is enthalpy; from thermodynamics, $U \triangleq H - pV$ and $\dot{U} \triangleq \dot{H} - p\dot{V}$. The differential variable is $x = (m_S, n_A, n_B, U)$.

Indices i, e, and g indicate influent, effluent, and generation, respectively.

2.4 Model Complexity

Here, we will operate with two levels of model complexity.

1. At first, we will simply assume an *ideal solution*. The resulting model will occasionally be referred to with index is. For this model, the reactor composition influences both reactor density and energy properties.
2. In a simpler model, we will both assume constant overall density due to domination of solvent S, and that the composition does not influence the energy properties. If we also assume a first order reaction ($a = 1$), these are the assumptions behind the original model as presented in Example 2.5 of (Seborg et al., 2011). The *original* model is in state space form, with c_A and T as states. This model will occasionally be referred to with index org.

2.5 Ideal Solution Model

We assume a perfectly mixed tank volume V . We will also assume an *ideal solution*. Let superscript \bullet indicate a quantity representing a hypothetically pure substance. Assuming an ideal solution implies:

$$V = V_S^\bullet + V_A^\bullet + V_B^\bullet \quad (12)$$

$$H = H_S^\bullet + H_A^\bullet + H_B^\bullet \quad (13)$$

where

$$V_j^\bullet = \frac{m_j}{\rho_j^\bullet} \quad (14)$$

$$m_j = n_j M_j \quad (15)$$

$$H_j^\bullet = m_j \hat{H}_j^\bullet = n_j \tilde{H}_j^\bullet \quad (16)$$

Similarly, for the influent flow, we find

$$\dot{V}_i = \dot{V}_{S,i}^\bullet + \dot{V}_{A,i}^\bullet; \quad (17)$$

$\dot{V}_{B,i}^\bullet \equiv 0$ (no feed of species B). Equations 12, 13, and 17 are based on the assumption of ideal solution, and are not generally valid. Equations 14–16, on the other hand, are valid by definition.

In addition to the ideal solution assumption, we need a model of liquid specific enthalpy. For an ideal (incompressible) liquid with temperature independent specific heat capacity:

$$\hat{H}_S^\bullet = \hat{H}_S^\circ + \hat{c}_{p,S}^\bullet (T - T^\circ) \quad (18)$$

and similar for species A and B.¹⁶ Here, $(\hat{H}_S^\circ, T^\circ)$ are specific enthalpy at standard state and a chosen reference temperature T° . Assuming ideal solution implies that

$$m_S = \rho_S^\bullet \left(V - \frac{n_A M_A}{\rho_A^\bullet} - \frac{n_B M_B}{\rho_B^\bullet} \right) \quad (19)$$

which gives an algebraic constraint between balanced variables m_S, n_A, n_B , hence only two of the three amount balances in Section 2.3 are independent — and we really only have 3 states in the system. Similarly (no influent of species B),

$$\dot{m}_{S,i} = \rho_S^\bullet \left(1 - \frac{c_{A,i} M_A}{\rho_A^\bullet} \right) \dot{V}_i. \quad (20)$$

In total, the ideal solution DAE model has 18 constants/parameters θ , 4 differential variables x , 41 algebraic variables z , and 4 input variables u . The number of state variables is 3; these are necessary to specify the system at initial time. Model parameters are given in Table 3, while operating conditions are given in Appendix A, Table 4.

2.6 Original ODE Model

The ideal solution DAE may be reduced to ODE form. Converting the species balances to differential equations for the concentrations, the result is rather complicated, and can be expressed in implicit form as

$$M \cdot \frac{d}{dt} \begin{pmatrix} c_A \\ c_B \end{pmatrix} = \begin{pmatrix} \frac{c_{A,i} \dot{V}_i - arV}{V} - \frac{c_A \dot{m}_{S,i} / (\rho_S^\bullet V)}{1 - \frac{c_A M_A}{\rho_A^\bullet} - \frac{c_B M_B}{\rho_B^\bullet}} \\ r - \frac{c_B \dot{m}_{S,i} / (\rho_S^\bullet V)}{1 - \frac{c_A M_A}{\rho_A^\bullet} - \frac{c_B M_B}{\rho_B^\bullet}} \end{pmatrix} \quad (21)$$

where M is the “mass matrix”

$$M = \begin{pmatrix} 1 + \frac{c_A M_A / \rho_A^\bullet}{1 - \frac{c_A M_A}{\rho_A^\bullet} - \frac{c_B M_B}{\rho_B^\bullet}} & \frac{c_A M_B / \rho_B^\bullet}{1 - \frac{c_A M_A}{\rho_A^\bullet} - \frac{c_B M_B}{\rho_B^\bullet}} \\ \frac{c_B M_A / \rho_A^\bullet}{1 - \frac{c_A M_A}{\rho_A^\bullet} - \frac{c_B M_B}{\rho_B^\bullet}} & 1 + \frac{c_B M_B / \rho_B^\bullet}{1 - \frac{c_A M_A}{\rho_A^\bullet} - \frac{c_B M_B}{\rho_B^\bullet}} \end{pmatrix}, \quad (22)$$

and the reaction rate is as in Eqs. 6–7. The internal energy balance can be simplified to

$$C_p \frac{dT}{dt} = \frac{\dot{V}_i}{V} C_{p,i} (T_i - T) + (-\Delta_r \tilde{H}) rV + \dot{Q} \quad (23)$$

¹⁶For species A and B, molar enthalpy is used.

where

$$C_p = c_A V \tilde{c}_{p,A}^{\bullet} + c_B V \tilde{c}_{p,B}^{\bullet} + m_S \tilde{c}_{p,S}^{\bullet} \quad (24)$$

$$C_{p,i} = V \rho_S^{\bullet} \tilde{c}_{p,S}^{\bullet} + V c_{A,i} \tilde{c}_{p,A}^{\bullet} \quad (25)$$

with m_S as in Eq. 19, while the reaction enthalpy $\Delta_r \tilde{H}$ is

$$\Delta_r \tilde{H} = \tilde{H}_B^{\circ} - a \tilde{H}_A^{\circ} + (\tilde{c}_{p,B}^{\bullet} - a \tilde{c}_{p,A}^{\bullet}) (T - T^{\circ}) \quad (26)$$

and added heat rate is as in Eq. 5.

If we assume that the solvent totally dominates the mixture, the ideal solution model simplifies further to an ODE with identical structure as that of Example 2.5 in (Seborg et al., 2011). If, furthermore, the reaction order is set to $a = 1$ and specific heat capacities are chosen as $\tilde{c}_{p,B}^{\bullet} \equiv \tilde{c}_{p,A}^{\bullet}$, then the model becomes identical to that of (Seborg et al., 2011). The original, constant density model is presented in (Khalili and Lie, 2018).

3 Implementation Details

3.1 Basic Language Characteristics

This paper considers modeling languages Modelica, Julia, and MATLAB. For Modelica, tool OpenModelica is used.

Modelica is equation based and object oriented. Equality symbol $=$ represents true mathematical equality, thus equations can appear in arbitrary order, and equations can be implicit. Constants and (default) parameters are named and given value within the model class, and input variables are supported. Modelica is object oriented, thus statement `Real T = 273.15` instantiates and gives value to a quantity named `T` in class `Real`. Similarly, if we create a model class `CSTR`, statement `CSTR mCSTR(R=8.31)` instantiates a model object `mCSTR` from class `CSTR` and sets class parameter `R = 8.31`.

When running Modelica models, they are first translated to assignments. This implies that a symbolic reordering and simplification of the model equations take place, before C code is generated and compiled to an executable file. If debugging is needed, this complicates matters because the actually executed order of the equations may differ from the order in the Modelica code.

Both Julia and MATLAB are assignment based. Thus, symbol $=$ represents assignment. In assignments, the order of the statements is important. Typically, the current value of the model descriptor is passed as an input argument, possibly together with parameter values, and current time. There is no direct support for system inputs. The current descriptor and the parameter are passed with the abstract names (e.g., `x` and `par`), and may be given a physically recognizable name within the function body. Examples: if the first element in the descriptor is the mass of solvent m_S , we could name it as `m_S = x(1)`. Similarly, if the first element of the parameter is $\mathcal{U}A$, we could name it as `UA = par(1)`. This would be needed in order to use recognizable label names in the model formulation

— for ease of understanding the code. Also, initial values of the full descriptor and parameter needs to be given outside of the model function.

Being assignment based, debugging is relatively easy because the statements are executed in the order they appear in the model.

3.2 Modelica Formulation

In Modelica (Fritzson, 2015), models are classes; the Ideal Solution model is named `ModSeborgCSTRis` and is enclosed between statements

```
model ModSeborgCSTRis
...
end ModSeborgCSTRis;
```

The model body is composed of sections, given by a section statement which ends with line feed. Within sections, statements can be multi-line, and are ended by a semi-colon `;`. The model quantities are usually defined in the first section. As an example, parameter ρ_S^{\bullet} , variable m_S with consistent initial value `mS0`, and input variable T_c are defined by

```
parameter Real rhoS_o = 1e3;
Real mS(start = mS0);
input Real Tc;
```

After the declaration of quantities, an equation section is declared by section statement **equation**. As an example, equations $\frac{dn_A}{dt} = \dot{n}_{A,i} - \dot{n}_{A,e} + \dot{n}_{A,g}$ and $V = V_S + V_A + V_B$ can be stated as

```
der(nA) = ndAi - ndAe + ndAg;
V = VS + VA + VB;
```

For balanced models, all model constants/parameters should be specified within the model class, and the model should hold the same number of equations as variables — with the exception of input variables which are defined outside of the model class.

Models without input variables can be instantiated directly from the model class, and simulated. On the other hand, models with input variables must be instantiated in another model class, and be given an input value in this other model class. Several models can be wrapped within a *package* and put in a file with the same name as the package. Here, we use package name `SeborgCSTR`, and the package is enclosed between statements

```
package SeborgCSTR
...
end SeborgCSTR;
```

To define the inputs to model `ModSeborgCSTRis`, we thus create a second model `SimSeborgCSTR` which is enclosed statements as required for model classes. Within this second model, we instantiate object `srIS` in the declaration section by statement `ModSeborgCSTRis srIS;` — the class name followed by the object name, similar to the statement `Real mS(start = mS0);`. It is possible to let the instantiated object `srIS` over-rule a parameter specified in class `ModSeborgCSTRis` just as object

mS has over-ruled a (default) parameter in class Real. Finally, in the equation section of class SimSeborgCSTR, we can give an input value to the instantiated object srIS with a statement such as `srIS.Tc = 300;`

To run the Modelica model, we can import the package into OpenModelica¹⁷, select the main class that we want to solve (SimSeborgCSTR), check the model for errors, and set up simulation details (solver, simulation length, tolerance, etc.), and simulate the system. OpenModelica has simple facilities for plotting results and saving the plots to file.

3.3 Julia Formulation

To solve differential equations in Julia¹⁸, it is necessary to add package DifferentialEquations (Rackauckas and Nie, 2017) from GitHub, which is straightforward. To activate this package within a Julia session, issue statement

```
julia> using DifferentialEquations
```

With this package, DAEs are posed as a DAEProblem. As an example, specifying the Ideal Solution model in function `seborg_is` with descriptor `x`¹⁹ and initial value guesses for $\frac{dx}{dt}|_0$ and $x|_0$ given by `dxdt0` and `x0`, simulation time span `tspan`, and parameter vector θ given in `par_is`, the DAE problem named `probis` is set up by command:

```
julia> probis = DAEProblem(seborg_is, dxdt0, x0,
    , tspan, par_is, differential_vars=diff_vars
    )
```

where keyword argument `differential_vars` is given value `diff_vars`; `diff_vars` is a Boolean vector (1D array) with true value for elements corresponding to differential variables in `x`, and false value for elements corresponding to algebraic variables.

The DAE problem is then solved by issuing command

```
julia> solis = solve(probis, IDA())
```

where `IDA` is the name of the chosen solver code (Hindmarsh et al., 2005). The solution is stored in type (object) `solis`.

The model has been implemented in function `seborg_is`, which is defined with arguments `seborg_is(err, dxdt, x, par, t)`. Here, variable `err` contains the errors in the equations: if e is this error, then the DAE formulation is rephrased into $e_d = -\frac{dx}{dt} + f(x, z, u; \theta)$ and $e_a = g(x, z, u; \theta)$, respectively for the differential and algebraic equations; the solver then attempts to make the error `err` ($= e_d, e_a$) as close to zero as possible. In order to operate with variables within the Julia function which are physically descriptive, it is necessary to rename the abstract names like `dxdt`, `x`, and `par` into `dmSdt = dxdt[1]`, etc. The model is then specified via statements for `err`, e.g.,

```
err[1] = -V + VS + VA + VB
...
err[42] = -dmSdt + mdSi - mdSe
```

etc. Input variables must be specified within the model function.

The Julia solver is more restrictive wrt. initial conditions than OpenModelica: for Julia, decent guesses for both `dxdt0` and `x0` must be supplied in order for the solution to be found, while for OpenModelica, initial conditions for the *state* suffice.

3.4 MATLAB Formulation

The MATLAB²⁰ formulation is written in a script named `seborg_is`. There are several ways to formulate DAE models in MATLAB, e.g., via the Mass Matrix. Here, we have instead used the Symbolic Math Toolbox to convert the system to a form suitable for numeric solvers.²¹

Quantities are declared using the `syms` function, with a distinction between variables and constants. As an example, the dependent variable n_A is written as `nA(t)` and the parameter V as `V`. Further, the Symbolic Math Toolbox functionality allows for the algebraic relations to be specified as mathematical equalities. As an example, equations $\frac{dn_A}{dt} = \dot{n}_{A,i} - \dot{n}_{A,e} + \dot{n}_{A,g}$ and $n_A = c_A V$ can be written as

```
eqn1 = diff(nA(t), 1) == ndAi - ndAe(t) + ndAg
    (t);
...
eqn5 = nA(t) == cA(t)*V;
```

These equations are stored in a row vector and the variables in a column vector.

In the case of ideal solution, it is necessary to reduce the differential index by invoking the `reduceDifferentialOrder` function. This function takes the equations and variables as input and creates new equations and variables to replace derivatives.

Next, input variables and parameters must be assigned a value and formulated as function handles suitable for MATLAB solvers. MATLAB is very sensitive regarding the order of assignment of variables and initial values.

It is also necessary to set initial values for every variable, including the differential variables created by `reduceDifferentialOrder`, which can be complicated to initialize. The `decic` function can be called to find consistent initial values that satisfy the equations from initial guesses. This function is sensitive to the input tolerances, and may produce false solutions if they are set too low.

Finally, the system is solved by integrating over a specified time span with the implicit solver `ode15i`.

¹⁷<https://openmodelica.org/>

¹⁸<https://julialang.org/>

¹⁹Julia doesn't distinguish between differential and algebraic variables.

²⁰<https://mathworks.com/>

²¹https://se.mathworks.com/help/symbolic/solve-differential-algebraic-equations.html?s_tid=gn_loc_drop

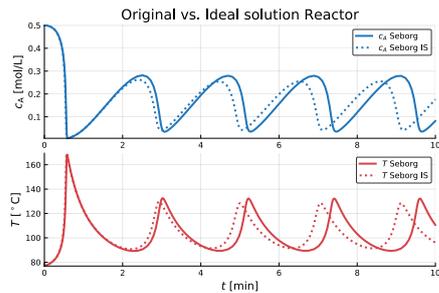


Figure 2. Temporal evolution of concentration c_A and temperature T for original model (solid) and ideal solution model (dotted).

Table 1. Average Scaled Execution Times over 1000 runs; basis is 5.3 ms for solving Ideal Solution (IS) model in Julia.

	OM*	Julia	MATLAB
ORG	11.5	0.36	0.3 (6.0)
IS	12.4	1	11.1

4 Simulation Results

4.1 Simulation Results

The simulation results are the same, whether OpenModelica, Julia, or MATLAB are used. Figure 2 illustrates the temporal evolution for the original model and the ideal solution model, as computed and plotted by Julia running in IJulia notebook²² when the cooling temperature jumps from $T_c = 300\text{ K}$ to $T_c = 305\text{ K}$ at initial time.

4.2 Execution Time

It is very difficult to compare various languages wrt. the execution time for solving models. Important considerations are:

- Are the same algorithms used?
- Is the same solution tolerance used?
- Is the execution time problem dependent?
- Do all languages have the same initial information, e.g., the same initial guess if iterations are needed?
- Do all languages have the same overhead in finding the solution?

Table 1 reports some execution times for OpenModelica as run from Python, for Julia, and for MATLAB — for the original model of (Seborg et al., 2011) and the DAE formulation of the ideal solution model. The results are indicative for assessing the languages, at best.

A few comments on the results in Table 1.

1. In all cases, different (but comparable?) solvers are used (OpenModelica: DASSL for ORG/IS; Julia: default for ORG, IDA for DAE; MATLAB:

²²= Jupyter notebook

Table 2. Comparison of Simulation Tools for solving DAE models. OM = (Open)Modelica, grading A–F where A is best.

	OM	Julia	MATLAB
Cost	A	A	E
Ease of use	B	C	D
Documentation	C	D	A
Numeric quality	A	A	A
Debugging	B	C	A
Eco-system	D	B	A
Library facility	A	C	D

ode45(ode15s) for ORG, ode15i for IS). For DAE problems, the tolerances have been tuned to find a solution for Julia and MATLAB.

2. For the DAE model, OpenModelica and Julia need to iterate to find initial value for the entire descriptor, while this has been precomputed for MATLAB.
3. For the ODE model, MATLAB solves the problem quickly when using the `ode45` solver, but 20 times slower when using the `ode15s` solver. This must be due to considerable overhead when using a stiff solver on a non-stiff problem.
4. OpenModelica code has been optimized, translated to C and compiled into an executable file. Thus, OpenModelica code should execute fast. The reason why OpenModelica is relatively slow must therefore lie in some inefficiency in the Python API.

5 Discussion

Table 2 gives a *subjective* comparison of computer languages/tools MATLAB, Julia, and OpenModelica.

With excellent support for libraries, similarity between theoretic model and implementation, support for input variables, and inclusion of named quantities within classes, Modelica is made for DAEs and is easy to use. It suffices to specify initial states; this adds to the ease of use. Julia package `DifferentialEquations.jl` gives good support for DAEs. MATLAB has good built-in support for ODEs; for solving DAEs, some complexity is involved in combining additional toolboxes such as the Symbolic Toolbox. Julia and MATLAB require good initial guesses for the entire descriptor and its derivative, which adds to their complexity.

MATLAB documentation is excellent, Modelica has a handful of introductory books, while the Julia language is still in a flux prior to v. 1.0²³ with partially outdated and simple books. All three languages have excellent solvers, still Julia has by far the largest set of available solvers.

MATLAB has a useful visual debugger with breakpoints. Being a young language, Julia currently has somewhat poor support for debugging. Because both languages

²³Julia v.1.0 is scheduled for August 6, 2018.

are assignment based, debugging is relatively easy. Modelica is equation based, thus equations are reordered and modified during translation to executable code without user intervention. It is therefore difficult for the user to keep track of which assignment comes first — and therefore which equation causes the problem. Modelica does have decent syntax checking and the user can keep track of undeclared/unused quantities, and whether the number of equations matches the number of unknowns. Also, there is a debugger which keeps track of the re-ordering of equations, but some expertise is needed to use this debugger efficiently.

MATLAB comes with good plotting capabilities, random number generators, etc., and has a rich Eco-system. Julia has a large set of available packages of very good quality, including support for Automatic Differentiation, which conceptually can be used for automatic linearization of models. Modelica has a more limited Eco-system, and lacks good support for random number generation, good plotting capabilities, etc. However, OpenModelica has support for automatic linearization, optimal control, etc., which can be integrated with Python via a Python API.

Although MATLAB and Julia have tools for solving DAEs, they lack good support for libraries and flow-sheeting. Modelica, on the other hand, excel in these areas.

6 Conclusions

When it comes to ease of use, direct translation from mathematical model to computer code, and support for libraries, Modelica is the clear winner. Coding DAE models is simple in Julia, too, but the notation is more abstract, and more importantly: a decent initial value for the entire descriptor is required. MATLAB is somewhat more complex in use than Julia again, and is also demanding wrt. initial value for the descriptor.

Regarding documentation and debugging, MATLAB is the clear winner. Julia is a very new language; v. 1.0 is still in the pipeline, and it is natural that the documentation is lacking. However, it is expected that documentation for Julia, as well as debugging facilities, will improve rapidly. Modelica has good documentation and decent debugging facilities.

The MATLAB Eco-system is extensive; Julia's is smaller, but surpasses that of MATLAB in some areas. The Modelica/OpenModelica Eco-system is more limited, with only rudimentary tools for plotting and analysis.

Regarding numeric quality, the languages are comparable although Julia is richest wrt. solvers. For execution speed, OpenModelica (compiled C code) and Julia (low level JIT compiler) should have an edge over MATLAB. Limited results indicate that, indeed, Julia is very fast. However, OpenModelica is comparable to MATLAB. The reason for this may lie in overhead in the Python API.

The ideal tool would be a one-language solution; this

would enable application of the entire Eco-system on the model, e.g., automatic differentiation, mixing model and optimization, structural analysis of the model, etc. At the moment, Julia is perhaps the best/most promising one-language tool.

However, recent work enables operation of OpenModelica within Python; support for integration of OpenModelica in both Julia and MATLAB are in the works. Although these are two-language solutions, the combinations reduce OpenModelica's disadvantage wrt. Eco-system. It would be better with an even tighter integration to eliminate interface overhead; this could, e.g., be achieved by compilation of Modelica into a script language instead of into C.

A Parameters and Operating Conditions

For the DAE model, choose the following parameters for the model, Table 3. The operating conditions (initial states, inputs) are defined in Table 4.

References

- Peter Fritzson. *Principles of Object-Oriented Modeling and Simulation with Modelica 3.3: A Cyber-Physical Approach*. Wiley-IEEE Press, Piscataway, NJ, second edition, 2015. ISBN 978-1-118-85912-4.
- Michael A. Henson and Dale E. Seborg. *Nonlinear Process Control*. Prentice Hall, Upper Saddle River, New Jersey, 1997.
- A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, and C. S. Woodward. Sundials: Suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software*, 31(3):363–396, 2005. Also available as LLNL technical report UCRL-JP-200037.
- Mohammad Khalili and Bernt Lie. Comparison of linear controllers for nonlinear, open-loop unstable reactor. In *Proceedings, SIMS 2018*, Oslo Metropolitan University, September 2018. SIMS, Linköping University Press.
- Bernt Lie, Sudeep Bajracharya, Alachew Mengist, Lena Buffoni, Arun Kumar, Martin Sjölund, Adeel Asghar, Adrian Pop, and Peter Fritzson. API for Accessing OpenModelica Models from Python. In *Proceedings of EuroSim 2016*, Oulu, Finland, 2016, September 2016.
- Christopher Rackauckas and Qing Nie. Differentialequations.jl — a performant and feature-rich ecosystem for solving differential equations in julia. *Journal of Open Research Software*, 5(15), 2017. DOI: <http://doi.org/10.5334/jors.151>.
- Dale E. Seborg, Thomas F. Edgar, Duncan A. Mellichamp, and III Doyle, Frank J. *Process Dynamics and Control*. John Wiley & Sons, Hoboken, NJ, third edition edition, 2011. ISBN 978-0-470-12867-1. ISBN 978-0-470-12867-1.

Table 3. Parameters for ideal solution liquid reactor.

Quantity	Symbol	Value
Density of pure species S	ρ_S^\bullet	1000 g/L
Density of pure species A	ρ_A^\bullet	1500 g/L
Density of pure species B	ρ_B^\bullet	2500 g/L
Molar mass of species A	M_A	50 g/mol
Molar mass of species B	M_B	$a \cdot M_A$
Reactor volume	V	100L
Stoichiometric constant	a	1
Activation energy per gas constant	E/R	8750 K
Reaction constant	k_0	$\exp\left(\frac{E/R}{350}\right) \approx 7.2 \cdot 10^{10} \text{ min}^{-1}$
Standard state temperature	T°	293.15 K
Standard state pressure	p°	$1.01 \times 10^5 \text{ Pa}$
Specific enthalpy of species S at (T, p°)	\hat{H}_S°	0 J/g
Molar enthalpy of species A at (T, p°)	\tilde{H}_A°	$5 \times 10^4 \text{ J/mol}$
Molar enthalpy of species B at (T, p°)	\tilde{H}_B°	0 J/mol
Specific heat capacity of species S	$\hat{c}_{p,S}^\bullet$	0.239 J/(g K)
Molar heat capacity of species A	$\hat{c}_{p,A}^\bullet$	$5 \frac{\text{J}}{\text{mol K}}$
Molar heat capacity of species B	$\hat{c}_{p,B}^\bullet$	$5 \frac{\text{J}}{\text{mol K}}$
Heat transfer parameter	$\mathcal{U}A$	$5 \times 10^4 \text{ J}/(\text{min K})$

Table 4. Operating conditions for ideal solution liquid reactor. Superscript * for inputs indicates nominal inputs.

Quantity	Symbol	Value
Initial value, concentration of A	$c_A _{t=0}$	0.5 mol/L
Initial mole number of species B	$n_B _{t=0}$	0 mol
Initial temperature	$T _{t=0}$	350 K
Initial mole number, A	$n_A _{t=0}$	$= c_A _{t=0} \cdot V$ in mol
Initial mass, species A	$m_A _{t=0}$	$n_A _{t=0} \cdot M_A$ in g
Initial mass, species B	$m_B _{t=0}$	$n_B _{t=0} \cdot M_B$ in g
Initial volume, pure species A	$V_A _{t=0}$	$\frac{m_A _{t=0}}{\rho_A^\bullet}$ in L
Initial volume, pure species B	$V_B _{t=0}$	$\frac{m_B _{t=0}}{\rho_B^\bullet}$ in L
Initial volume, pure species S	$V_S _{t=0}$	$V - V_A _{t=0} - V_B _{t=0}$ in L
Initial mass, species S	$m_S _{t=0}$	$V_S _{t=0} \cdot \rho_S^\bullet$ in g
Initial specific enthalpy of species S	$\hat{H}_S^\bullet _{t=0}$	$\hat{H}_S^\circ + \hat{c}_{p,S}^\bullet (T _{t=0} - T^\circ)$ in J/g
Initial molar enthalpy of species A	$\tilde{H}_A^\bullet _{t=0}$	$\tilde{H}_A^\circ + \hat{c}_{p,A}^\bullet (T _{t=0} - T^\circ)$ in J/mol
Initial molar enthalpy of species B	$\tilde{H}_B^\bullet _{t=0}$	$\tilde{H}_B^\circ + \hat{c}_{p,B}^\bullet (T _{t=0} - T^\circ)$ in J/mol
Initial enthalpy of species S	$H_S^\bullet _{t=0}$	$m_S _{t=0} \hat{H}_S^\bullet _{t=0}$ in J
Initial enthalpy of species A	$H_A^\bullet _{t=0}$	$n_A _{t=0} \tilde{H}_A^\bullet _{t=0}$ in J
Initial enthalpy of species B	$H_B^\bullet _{t=0}$	$n_B _{t=0} \tilde{H}_B^\bullet _{t=0}$ in J
Initial total enthalpy of ideal mixture	$H _{t=0}$	$H_A^\bullet _{t=0} + H_B^\bullet _{t=0} + H_S^\bullet _{t=0}$ in J
Initial internal energy	$U _{t=0}$	$H _{t=0} - p^\circ V \times 10^{-3}$ in J
Influent volumetric flow rate	\dot{V}_i^*	100 L/min
Influent concentration of species A	$c_{A,i}^*$	1 mol/L
Influent temperature	T_i^*	350 K
Cooling liquid temperature	T_c^*	300 K

Comparison of Linear Controllers for Nonlinear, Open-loop Unstable Reactor

Mohammad Khalili and Bernt Lie

University of South-Eastern Norway, Bernt.Lie@usn.no

Abstract

Many industrially important reactors are operated over a regime where they exhibit nonlinear behavior. Exothermic reactors are often open-loop unstable. For such reactors, safe operation with good performance requires good control. We have considered linear controllers and tested these on a simple chemical engineering non-linear, open loop unstable model and an extension of the model in order to do a basic test of model uncertainty. Specifically, a PI controller has been tuned and tested, the operation of an Extended Kalman Filter (EKF, KF) has been tuned and tested. Based on feedback from estimated states in the EKF, a linear quadratic controller with integral action (LQG+I) has been tuned and tested; the study has been carried out using MATLAB for KF and LQG+I tuning; the remaining study has been carried out in a Jupyter Notebook using Python in tandem with Modelica. The PI controller lead to negative cooling water temperature upon a step change in temperature reference. When constraining the input to liquid water with anti-windup, PI control gives considerable undesirable overshoot in the reactor temperature. The LQG+I controller performs much better wrt. temperature overshoot. Overall, the reported work has demonstrated how a modern simulation set-up (OpenModelica + Python) can be used for model based control analysis and design.

Keywords: nonlinear reactor, model uncertainty, control design, PI control, LQG control

1 Introduction

1.1 Background

Many industrially important reactors are operated over a regime where they exhibit nonlinear behavior. Exothermic reactors are often open-loop unstable. For such reactors, safe operation with good performance requires good control.

It is of interest to compare how modern tools can be used to efficiently and accurately assess the performance of both standard PI controllers and more advanced multivariable controllers. In the simplest case, classical output feedback linear quadratic reactors with state estimation (LQG) could be considered, possibly extended with constraint handling as in linear model predictive control (MPC). A further extension could involve nonlinear MPC and nonlinear state estimation, but also nonlinear

feedback controllers such as passivity based controllers and nonlinear observers.

To make the comparison concrete, it is useful to introduce a representative model which is nonlinear and open loop unstable, with constraints in inputs and outputs. It is also of interest to consider systems which are multivariable input with multivariable outputs (MIMO), and also systems with stable or unstable zero dynamics.

1.2 Previous Work

Classical control is well developed (Åström and Murray, 2008), (Seborg et al., 2011); the same is true for linear quadratic optimal control (Anderson and Moore, 1989). Model predictive control is newer (Maciejowski, 2002), (Rawlings et al., 2017). State estimation is described, e.g., in (Simon, 2006). Basic nonlinear control is described in (Henson and Seborg, 1997).

A nonlinear, open loop unstable single input - single output (SISO) reactor model is described in (Seborg et al., 2011); an extension of the model is described in (Sund et al., 2018). The Modelica modeling language is useful for encoding dynamic models (Fritzson, 2015), and extended analysis can be achieved by combining Modelica with Python (Lie et al., 2016).

1.3 Organization of Paper

Section 2 describes the chosen illustrative process model, together with the control problem. In Section 3, a PI controller for the open-loop unstable process model is developed, including a description of an anti-windup algorithm that is easy to implement in Modelica. Further, continuous LQ control with integral action is described, and a controller is tuned. Next, a discrete time constant gain state estimator is discussed. In Section 4, the state estimator is tested for both the original model and an extended model, then the PI controller is tested for both models. Finally, the LQG+I controller is tested. In Section 5, the results are discussed, and some conclusions are drawn. In three appendices, the extended model is described, a nominal linear model is given, and the extended LQ+I system is given.

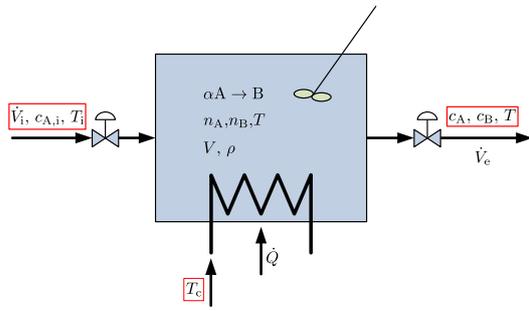


Figure 1. Cooled liquid reactor with reaction $aA \rightarrow B$.

2 Problem Description

2.1 Process Overview

We consider a liquid reactor of constant volume V , with influent volumetric flow rate \dot{V}_i , influent concentration $c_{A,i}$ of reactant A carried via an inert solvent S,¹ and influent temperature T_i , Fig. 1.

It is of interest to convert species A into species B through an exothermic reaction



the products are carried out of the reactor via solvent S. To keep control of reactor temperature T , heat rate \dot{Q} is added by flowing a liquid at temperature T_c through the tube side of a coil/heat exchanger. With a high flow rate of the cooling liquid, T_c is constant through the heat exchanger, and the heat rate can be described as

$$\dot{Q} = \mathcal{U}A(T_c - T). \quad (2)$$

where $\mathcal{U}A$ is a parameter. If $T_c < T$, $\dot{Q} < 0$ and the reactor is cooled.

The rate of reaction r is given as

$$r = kc_A^a \quad (3)$$

where a is the reaction order and k is given by the Arrhenius expression

$$k = k_0 \exp\left(-\frac{E/R}{T}\right). \quad (4)$$

The operation of the reactor is influenced by inputs \dot{V}_i , $c_{A,i}$, T_i and T_c , and it is of interest to study how these influence the outputs c_A , c_B and T .

Assuming that the reaction mixture constitutes an ideal solution, a DAE formulation of the system model is discussed in (Sund et al., 2018). Appendix A suggests how the DAE model can be manipulated into an implicit ODE, and furthermore how assuming first order reaction ($a = 1$), constant reactor density, and constant thermal parameters leads to the model in (Seborg et al., 2011). A Linear Time Invariant (LTI) approximate model in form

$$\frac{dx^\delta}{dt} = Ax^\delta + Bu^\delta \quad (5)$$

$$y^\delta = Cx^\delta + Du^\delta \quad (6)$$

is given in Appendix B where superscript δ indicates deviation from a nominal solution, based on the conditions suggested in (Sund et al., 2018) and computed using a Python API for OpenModelica (Lie et al., 2016).

2.2 Control Problem

Although the case study has inputs $u = (\dot{V}_i, c_{A,i}, T_i, T_c)$ and outputs $y = (c_A, c_B, T)$, in a control problem one may choose to control the temperature $y = T$ by manipulating the input $u = T_c$. In that case, the additional inputs $(\dot{V}_i, c_{A,i}, T_i)$ may be considered disturbances.

Realistic controllers must utilize output feedback; internal states are not known in practice, thus state feedback is infeasible. We compare the operation of linear controllers such as proportional (P) and linear quadratic (LQ) and linear quadratic with state estimation (LQG), as well as their versions with integral action (PI, LQ+I, LQG+I). Some key differences are that P+PI and LQG, LQG+I are output feedback controllers, while LQ, LQ+I are unrealistic state feedback controllers. The main difference between P+PI and LQ(G), LQ(G)+I controllers are that P+PI controllers do not use much information about the system dynamics, while LQ(G), LQ(G)+I controllers include a dynamic model².

With digital controllers, measurements are available at given sample times, and it is common to use zero-order hold of the control input between each sample. In modern control systems, also P+PI controllers are usually implemented as digital controllers.

3 Controller Design

3.1 PI Controller

Using the linearized model in Appendix B with T_c as control input, the closed loop matrix A_{cl} with a proportional controller (P controller) is

$$A_{cl} = A - K_p B_{:,1} C \quad (7)$$

where $B_{:,1}$ is the first column of B . Looping through $K_p \in [-1, 8]$ leads to the closed loop eigenvalues as depicted in Figure 2.

The P-controller stabilizes the system for $K_p \gtrsim 1.2$; $K_p = 5.7$ gives two real, closed loop eigenvalues/poles at approximately $\lambda \approx -5$, which implies closed loop time constants $\tau_j \approx \frac{1}{5} = 0.2$.

For a proportional + integral controller, it is reasonable to let the reset time (= integral time) be, say, 10 times larger than the closed loop time constants of a P controller. Thus, the PI controller

$$T_c(s) = T_c^* + K_p \frac{1 + T_{int}s}{T_{int}s} \cdot e(s) \quad (8)$$

$$e(s) = T_{ref}(s) - T(s) \quad (9)$$

¹Inert implies non-reacting.

²Essentially, LQ controllers give “intelligent” derivative action.

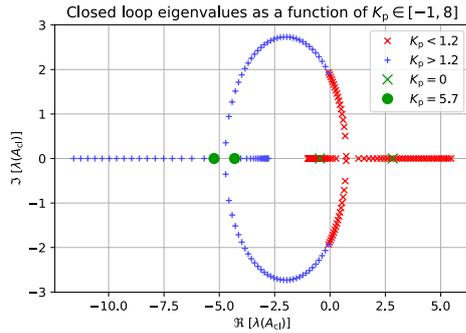


Figure 2. Root locus plot $\lambda(A_{cl}; K_p)$ for $K_p \in [-1, 8]$.

with $K_p = 5.7$ and $T_{int} = 2$ may be an acceptable choice.³ Nominal input T_c^* is not needed with integral action, but is useful to avoid an initial “kick” in the control action. T_{ref} is the reference temperature. If we let $T_{int} \rightarrow \infty$, the controller becomes a P controller.

In the time domain, we can express the PI controller as

$$T_c - T_c^* = K_p e + \tilde{T}_c \quad (10)$$

$$\frac{d\tilde{T}_c}{dt} = \frac{K_p}{T_{int}} e. \quad (11)$$

To handle constraints for $T_c \in [4, 96]^\circ\text{C}$, if $T_c = K_p e + T_c^* + \tilde{T}_c$ breaks this constraint, we set T_c equal to the constraint and $\frac{d\tilde{T}_c}{dt} = 0$ to avoid controller wind-up.

3.2 LQ+I Controller

We consider an infinite horizon optimal linear quadratic (LQ) controller with cost function

$$\mathcal{J} = \frac{1}{2} \int_0^\infty (e_y^T Q_y e_y + e_u^T R e_u) dt \quad (12)$$

where matrix $Q_y > 0$ puts weight on *quality* deviation $e_y = y - y_{ref}$ and matrix $R > 0$ puts weight on the *regulating* input $e_u = u - u^*$. Postulating a reference state x_{ref} such that $y_{ref} = C x_{ref}$, this gives the standard LQ cost function

$$\mathcal{J} = \frac{1}{2} \int_0^\infty (e_x^T Q e_x + e_u^T R e_u) dt \quad (13)$$

where $e_x = x - x_{ref}$ with $x_{ref} = \begin{pmatrix} T_{ref} & c_{A,ref} \end{pmatrix}^T$ and $c_{A,ref}$ is found by solving Eq. 36 in steady state:

$$c_{A,ref} = \frac{\dot{V}_i/V}{k_0 \exp\left(-\frac{E/R}{T_{ref}}\right) + \dot{V}_i/V} c_{A,i}, \quad (14)$$

while $Q = C^T Q_y C \geq 0$. The solution is

$$e_u = -K_c e_x \Rightarrow u = u^* - K_c e_x \quad (15)$$

where K_c can be found, e.g., using the MATLAB control toolbox as `>> Kc = lqr(A, B, Q, R)`. Here, it is required

³The integral time is denoted T_{int} in order to make a distinction between integral time and influent temperature, T_i .

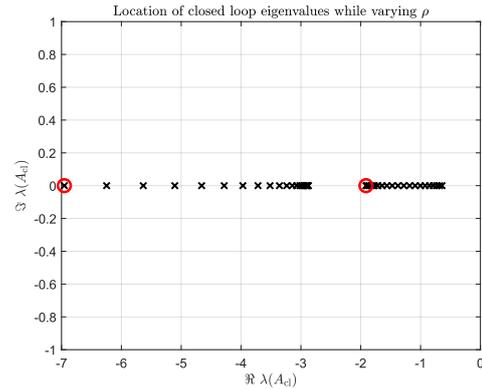


Figure 3. Variation of closed loop eigenvalues $\lambda(A_{cl})$ with ρ varying logarithmically over $[10^1, 10^{-2}]$; red circles indicate “cheap control”.

that (A, B) is stabilizable and (A, C) detectable. The closed loop eigenvalues are given by

$$A_{cl} = A - BK_c. \quad (16)$$

See, e.g., (Anderson and Moore, 1989) for a classical treatment.

A relatively common rule-of-thumb for choosing weight matrices is to let them be diagonal with elements

$$Q_{y,j,j} = 1/\varepsilon_{y,j}^2 \quad (17)$$

$$R_{j,j} = \rho/\varepsilon_{u,j}^2 \quad (18)$$

where $\varepsilon_{y,j}$ and $\varepsilon_{u,j}$ are some acceptable deviations in $e_{y,j}$ and $e_{u,j}$, respectively, while ρ is some scalar tuning parameter for R which is manipulated to get acceptable closed loop eigenvalues. We could choose, say, approximately 1% of nominal values:

$$Q_y = \frac{1}{32}, \quad R = \rho \frac{1}{32}. \quad (19)$$

If $\rho \rightarrow 0$, this should give a fast closed loop system since this means that the control input is “cheap”. Figure 3 shows the closed loop eigenvalues $\lambda(A_{cl})$ as ρ varies logarithmically over $[10^1, 10^{-1}]$, and illustrates the well known result that one eigenvalue approaches the (“stable”) zero at $\lambda = -2$ when $\rho \rightarrow 0$ (Kwakernaak and Sivan, 1972). From an input-output perspective, the closed loop pole at $\lambda \approx -2$ will approximately cancel out the zero; see transfer function in Appendix B.

A decent compromise would be $\rho = 0.2$, leading to eigenvalues at $\lambda(A_{cl}) = (-5.1761, -1.8246)$ where the fast eigenvalue would be similar to that achieved via P control. The resulting state feedback gain is

$$K_c = \begin{pmatrix} 4.4838, & 107.4579 \end{pmatrix}. \quad (20)$$

This standard LQ controller gives steady deviation in e_y . One way to achieve zero steady deviation in e_y is to augment the system with state

$$\frac{dz}{dt} = e_y \quad (21)$$

and extend the cost function to

$$\tilde{\mathcal{J}} = \mathcal{J} + \frac{1}{2} \int_0^{\infty} z^T Q_z z dt : \quad (22)$$

unless $e_y \rightarrow 0$, $\tilde{\mathcal{J}}$ will be unbound and can not be considered optimal.⁴ The extended control problem with integrator is solved as above, with augmenting the system into state $\tilde{x} = (x, z)$, and likewise for \tilde{Q} , see Appendix C.⁵

A possible rule-of-thumb for Q_z is $Q_z = Q_y/\rho_z$ where first Q_y and R are chosen to give decent closed loop poles of the original system, and then ρ_z is tuned to give reasonably fast integral action. Setting $\rho_z = 4$ leads to

$$\tilde{K}_{cl} = (4.7073, 106.0020, 1.1180) \quad (23)$$

and closed loop eigenvalues $\lambda(\tilde{A}_{cl}) = (-5.1568, -0.5011, -1.8105)$.

Using a digital controller with fixed sample time, a discrete version of the LQ controller can easily be found by (i) discretizing the system $(A, B, C, D) \xrightarrow{t_s} (A_d, B_d, C, D)$ with sample time t_s using, e.g., MATLAB function `c2d(sys, ts)`, and related optimal control function `>> Kc = dlqr(Ad, Bd, Q, R)`.

3.3 State Estimator

Because measurements normally are available in discrete time with sample time t_s , it is more convenient to operate with a discrete time model. Thus, we assume a linearized model

$$x_{t+1} = A_d x_t + B_d u_t + G_d w_t \quad (24)$$

$$y_t = C x_t + D u_t + v_t \quad (25)$$

where B_d is discretized input matrix $B_{:,1}$, G_d is discretized input matrix $B_{:,2:4}$, w_t is process disturbance and v_t is measurement noise. We will assume that w and v are uncorrelated, zero mean white noise with a Gaussian distribution, and co-variance matrices W and V , respectively. Thus

$$E \{ w_t w_{t+\tau}^T \} = W \cdot \delta_\tau \quad (26)$$

$$E \{ v_t v_{t+\tau}^T \} = V \cdot \delta_\tau \quad (27)$$

$$E \{ w_t v_{t+\tau}^T \} = 0 \quad (28)$$

where δ_τ is the (single argument) Kronecker delta. With input T_c used as control variable, inputs $(T_i, \dot{V}_i, c_{A,i})$ would be natural disturbances. With *a priori* estimates $\hat{x}_{t|t-1}$ given by solving the non-disturbed, nonlinear model over sample time t_s with initial value $\hat{x}_{t-1|t-1}$, and *a priori* measurement $\hat{y}_{t|t-1} = g(\hat{x}_{t|t-1}, u_{t-1})$, we can find the *a posteriori* estimate $\hat{x}_{t|t}$ from Kalman's expression as

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_e (y_t - \hat{y}_{t|t-1}). \quad (29)$$

⁴Even if $e_y \rightarrow 0$, most likely $z \neq 0$ for infinite time leading to infinite value for $\tilde{\mathcal{J}}$; still, some optimum may be found. However, if e_y is different from zero at infinite time, no optimum can be found.

⁵Seemingly, integral action can be achieved in a simpler way by replacing weight on u in \mathcal{J} with weight on \dot{u} . However, this requires an unbiased model to work, with added complexity of estimating the bias in a state estimator.

Here, a constant Kalman gain K_e can be found, e.g., by using MATLAB's control toolbox, `>> Ke = dlqe(Ad, Gd, C, W, V)`. It is required that (A_d, C) is detectable and (A_d, G_d) is stabilizable in order for the estimator to work.

In practice, W and V are not known accurately. A possible rule-of-thumb for tuning the state estimator could be as follows. Let $\hat{\zeta}_{w,j}$ be assumed variation in w_j over time horizon τ_j . Then choose diagonal elements of W as $W_{j,j} = (\hat{\zeta}_{w,j} \frac{t_s}{\tau_j})^2$.⁶ For the measurement noise, assume instantaneous standard deviation $\hat{\sigma}_{v,j}$, but we set diagonal elements to $V_{j,j} = \mu \hat{\sigma}_{v,j}$ where μ is a tuning parameter to give suitable closed loop performance of the estimator: if $\mu \rightarrow 0$, we assume little measurement noise and the estimator should become fast. Ideally, μ should be unity.

We could choose, say, approximately 1% of nominal values for $\hat{\zeta}_{w,j}$ in the course of $\tau_{w,j} = \tau$, and assume a 1% measurement standard deviation:

$$W = \left(\frac{t_s}{\tau} \right)^2 \begin{pmatrix} \left(\frac{300}{100} \right)^2 & 0 & 0 \\ 0 & \left(\frac{100}{100} \right)^2 & 0 \\ 0 & 0 & \left(\frac{1}{100} \right)^2 \end{pmatrix}, \quad V = \mu \cdot \left(\frac{300}{100} \right)^2. \quad (30)$$

To simplify matters, we will assume direct disturbance to each state, $w_t' = G_d w_t$, where

$$W' = G_d W G_d^T. \quad (31)$$

Choosing $t_s = \frac{1}{10}$, $\tau = t_s$, and $\mu = 1$ leads to Kalman gain

$$K_e = \begin{pmatrix} 0.45942 \\ -0.0030453 \end{pmatrix} \quad (32)$$

and estimator closed loop eigenvalues at $\lambda = (0.74733, 0.91768)$ which are well within the unit circle, hence stable.

The fast mode in the *controller* has a time constant of approximately $\tau_j \approx \frac{1}{5} = 0.2$, which implies that a step change is damped to $\exp(-1) \approx 0.37$ in 0.2 min, or to approximately 0.74 in $t_s = 0.1$ min. This means that the fast mode of the controller in a *discretized* system is approximately $\lambda_d \approx 0.74$. In other words: the Kalman Filter gain in Eq. 32 makes the estimator *slower* than the controller. If we tune μ to $\mu = 10^{-2}$ instead, we find

$$K_e = \begin{pmatrix} 0.75947 \\ -0.0018525 \end{pmatrix} \quad (33)$$

with eigenvalues at $\lambda = (0.37741, 0.80854)$. Reducing μ further, the second eigenvalue is stuck at a value around 0.8; the first eigenvalue, is, however, twice as fast as the closed loop controller eigenvalue.

3.4 LQG+I Controller

For linear systems, the controller and the estimator (with Gaussian noise, LQG) can be tuned independently. We

⁶The assumed discrete time standard deviation would be $\hat{\sigma}_{w,j} = \hat{\zeta} \frac{t_s}{\tau_j}$.

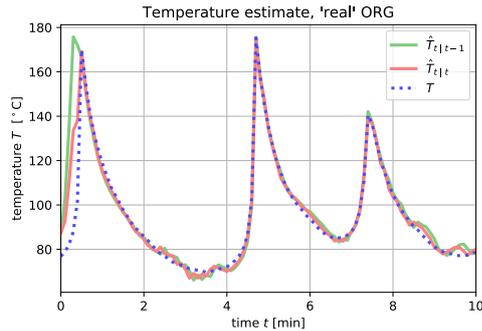


Figure 4. Real temperature T , a priori estimate $\hat{T}_{t|t-1}$, and a posteriori estimate $\hat{T}_{t|t}$ when estimator model is identical to the real system — except initial states.

will assume that this is true here, too. Because it doesn't make sense to assume “disturbance” on the integral state, we will base the estimator on the original model with state x , and not on state \tilde{x} .

4 Simulation Results

4.1 Operation

The model as given in Appendix A is used for all designs (P+PI controller tuning, LQ(G)+I design); this original model is referred to as the `org` model. A slightly more complex model (ideal solution assumption, `is`) is used. White noise disturbances, additive to the nominal values of $(T_i, \dot{V}_i, c_{A,i})$ in (Sund et al., 2018), are used with values as discussed in Section 3.3 for all simulations. For testing the state estimator, the measurement noise indicated in Section 3.3 is used with parameter $\mu = 1$ in the Kalman Filter tuning, together with a 3% deviation in the initial states of the estimator. Because a continuous time PI controller is implemented in Modelica, it is difficult to have continuous measurement noise: for the controller comparisons, the measurement noise is set to zero, and the tuning parameter of the Kalman Filter is set to $\mu = 10^{-2}$.

4.2 State Estimation

Figure 4 shows the estimate of T when the “real” system equals the `org` model.

Figure 5 shows the resulting estimate of T when the “real” system equals the `is` model.

Figures 6 and 7 show the similar estimates of c_A , respectively.

4.3 Proportional + Integral Control

Figure 8 shows the use of PI controller to keep reactor temperature T close to a reference T_{ref} . The PI controller tuned for the `org` model, is also applied to the `is` model.

The result indicates that the controller easily handles the model difference between the two models. Figure 9 shows the applied control input T_c as well as the integral state \bar{T}_c in the controller for the two model cases.

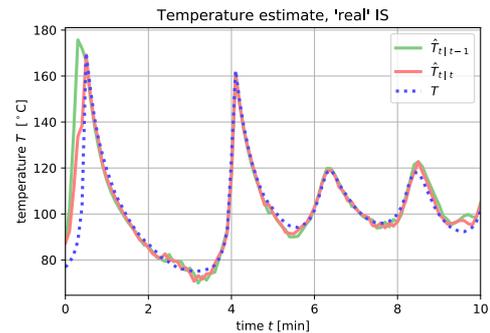


Figure 5. Real temperature T , a priori estimate $\hat{T}_{t|t-1}$, and a posteriori estimate $\hat{T}_{t|t}$ when estimator model differs from the real system, including different initial states.

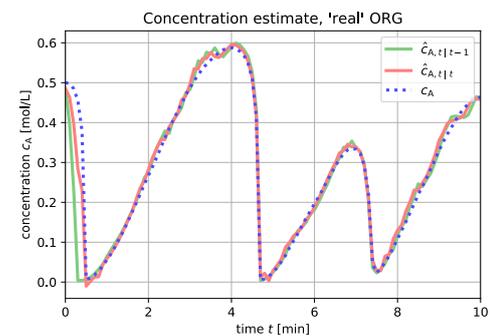


Figure 6. Real temperature c_A , a priori estimate $\hat{c}_{A,t|t-1}$, and a posteriori estimate $\hat{c}_{A,t|t}$ when estimator model is identical to the real system — except initial states.

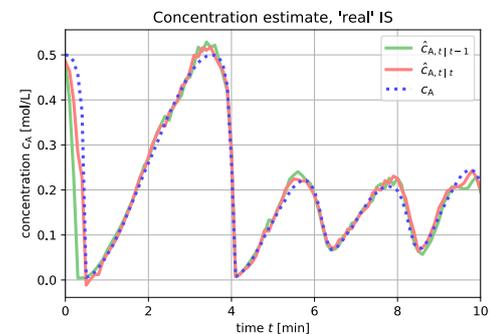


Figure 7. Real temperature c_A , a priori estimate $\hat{c}_{A,t|t-1}$, and a posteriori estimate $\hat{c}_{A,t|t}$ when estimator model differs from the real system, including different initial states.

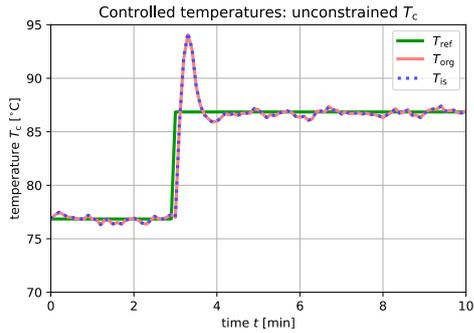


Figure 8. Output T as controlled with PI controller tuned for org model, and applied to org and is model.

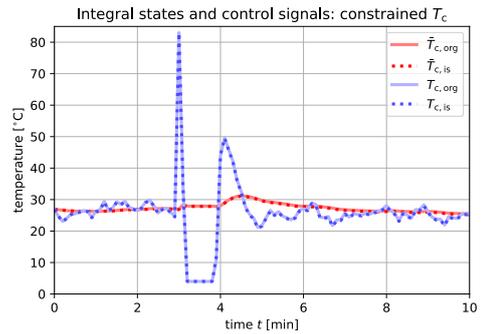


Figure 11. PI control signal T_c and integrator state \bar{T}_c for org and is models: control input T_c is constrained to $[4, 96]$ °C and anti-windup is applied.

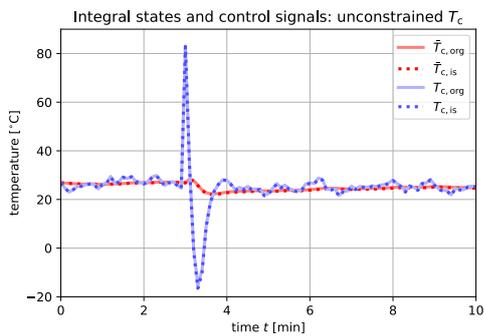


Figure 9. PI control signal T_c and integrator state \bar{T}_c for org and is models.

Figure 9 clearly shows a problem for the controller: the cooling water can not take on negative temperatures T_c . We therefore add the constraint that $T_c \in [4, 96]$ °C, which together with anti-windup leads to the results in Figures 10 and 11 for output T and controller T_c , respectively.

4.4 Linear Quadratic Control

Figure 12 shows the use of LQG+I controller to keep reactor temperature T close to a reference T_{ref} .

Figure 13 shows the applied control input T_c .

Figures 12 and 13 should be compared with Figures 8

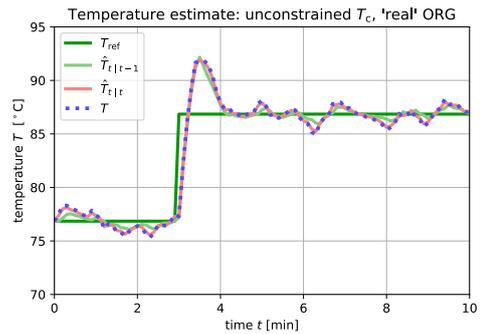


Figure 12. Output T as controlled with LQG controller tuned for org model, and applied to org model.

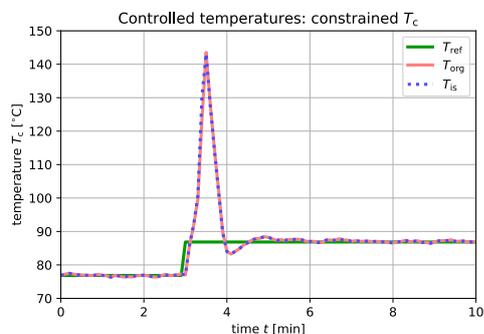


Figure 10. Output T as controlled with PI controller tuned for org model, and applied to org and is model: control input T_c is constrained to $[4, 96]$ °C and anti-windup is applied.

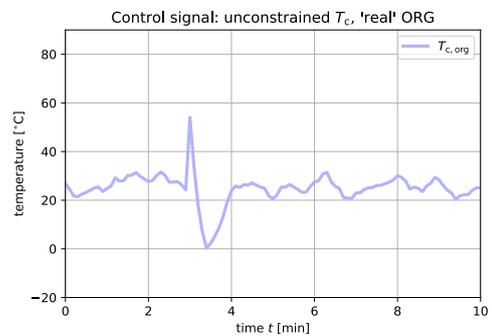


Figure 13. Control signal T_c from LQG controller.

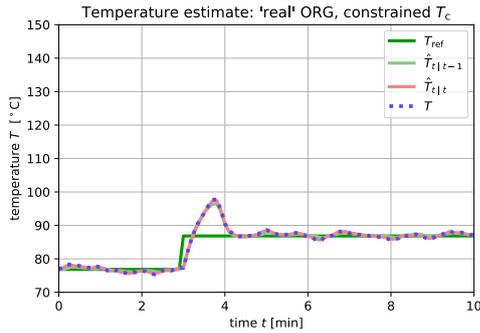


Figure 14. Output T as controlled with LQG controller tuned for `org` model, and applied to `org` model: control input T_c is constrained to $[4, 96]$ °C and anti-windup is applied.

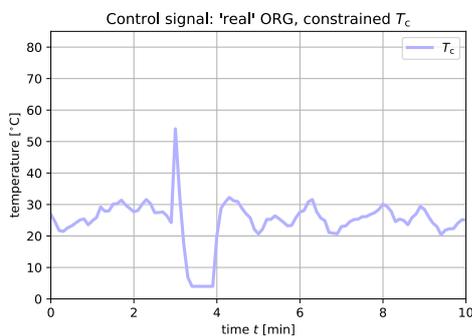


Figure 15. Control signal T_c from LQG controller: control input T_c is constrained to $[4, 96]$ °C and anti-windup is applied.

and 9, respectively.

Adding the same constraint and anti-windup in the LQG+I controller leads to the results in Figures 14 and 15 for output T and controller T_c , respectively. These should be compared to Figures 10 and 11, respectively.

5 Discussion and Conclusions

Practical procedures for tuning a PI controller for an open loop unstable system is described, together with some continuous time implementation details with anti-windup action suitable for implementation in Modelica. A disadvantage of a continuous time implementation is that the chosen combination (Python + Modelica) does not allow for continuous time measurement noise. Details of tuning a constant gain Extended Kalman Filter is discussed, with an algorithm suitable for implementation in a combination of Python and Modelica. Finally, design of a linear LQG controller is described, and a controller is tuned. The method is extended to include integral action.

The developed control algorithms are applied to the nonlinear models, and tested through simulation. The EKF estimator works well in a highly non-linear regime, even for a more complex model. The developed PI controller is tuned; the example shows the need to take control input constraint into account. This leads to an unfortunate overshoot in the controlled variable (tempera-

ture). The LQG controller gives superior control with a considerable reduction in temperature overshoot, but gives slightly less disturbance attenuation; this problem would have been solved by setting KF parameter μ even smaller. To further reduce the temperature overshoot, utilizing the nonlinearity of the system in the controller design should be attempted.

The design of the LQG controller was carried out using the MATLAB Control Toolbox⁷. All other parts of the design and testing was carried out in a Jupyter Notebook⁸ where Python code was combined with Modelica code.

The main message of this work is not so much the design of a controller, but rather to demonstrate how modern modeling and simulation tools such as OpenModelica in tandem with a script based language such as Python can ease and streamline control design, analysis, and testing through simulation.

A Model details

The ideal solution extension of the model in (Seborg et al., 2011) is presented with some details in (Sund et al., 2018): model parameters and nominal operating conditions are given, together with a manipulation of the model into an implicit ODE model of form $M \cdot \frac{dx}{dt} = f(x, u; \theta)$. The sequel discusses how this implicit ODE model can be simplified to the model in (Seborg et al., 2011).

When assuming that solvent S totally dominates in the mixture wrt. density, “mass matrix” M in Appendix A of (Sund et al., 2018) simplifies to the identity matrix, $M \rightarrow I$. Furthermore, the total heat capacities simplify to

$$C_p = m_S \hat{c}_{p,S}^* = C_{p,i}. \quad (34)$$

With this assumption, information about species B becomes irrelevant — unless we are particularly interested in c_B . Thus, when c_B is of no particular interest, the model can be simplified to

$$\frac{dc_A}{dt} = \frac{\dot{V}_i}{V} (c_{A,i} - c_A) - a \cdot r \quad (35)$$

$$\frac{dT}{dt} = \frac{\dot{V}_i}{V} (T_i - T) + \frac{(-\Delta_r \tilde{H}) r}{\rho_S^* \hat{c}_{p,S}^*} + \frac{\dot{Q}}{\rho_S^* \hat{c}_{p,S}^* V}, \quad (36)$$

where r and \dot{Q} are given by Eqs. 3 and 2, respectively, while $\Delta_r \tilde{H}$ is given by

$$\Delta_r \tilde{H} = \tilde{H}_B^{\circ} - a \tilde{H}_A^{\circ} + (\tilde{c}_{p,B}^* - a \tilde{c}_{p,A}^*) (T - T^{\circ}). \quad (37)$$

Consider now the Continuous Stirred Tank Reactor (CSTR) in Example 2.5 of (Seborg et al., 2011): this model is identical to the simplified model in Eqs. 35–36⁹ provided that $a \equiv 1$ and $\Delta_r \tilde{H}$ is constant wrt. temperature. Assuming that the standard state specific enthalpies are

⁷www.mathworks.com

⁸www.jupyter.org

⁹The model notation is changed from that of (Seborg et al., 2011).

temperature independent, i.e., \hat{H}_j° is temperature independent, this requires that $\tilde{c}_{p,B}^\bullet - a\tilde{c}_{p,A}^\bullet \equiv 0$ — or $\tilde{c}_{p,B}^\bullet \equiv \tilde{c}_{p,A}^\bullet$. The parameters and operating conditions in Tables 3-4 of (Sund et al., 2018) are judiciously chosen to ensure that the solvent-dominating model is identical to the model in (Seborg et al., 2011).

B Model linearization

For the model, with $x = (T, c_A)$, $u = (T_c, T_i, \dot{V}_i, c_{A,i})$, and $y = T$, the linearized model is:

$$\frac{dx^\delta}{dt} = Ax^\delta + Bu^\delta \quad (38)$$

$$y^\delta = Cx^\delta + Du^\delta \quad (39)$$

where for any z , $z^\delta \triangleq z - z^*$ and asterisk $*$ indicates nominal value. Using the Python API described in (Lie et al., 2016) together with OpenModelica, let `sr_org` be a Python object of the original reactor in (Seborg et al., 2011) (but with n_A replacing c_A as state). We can then linearize the model in Python with the statement:

```
>>> A, B, C, D = sr_org.linearize()
```

The following system matrices are found:

$$A = \begin{pmatrix} 4.3796 & 209.205 \\ -0.035714 & -2 \end{pmatrix} \quad (40)$$

$$B = \begin{pmatrix} 2.09205 & 1 & 0 & 0 \\ 0 & 0 & 0.005 & 1 \end{pmatrix} \quad (41)$$

$$C = (1 \ 0) \quad (42)$$

$$D = (0 \ 0 \ 0 \ 0). \quad (43)$$

The open loop eigenvalues of A are $\lambda = (2.83388381, -0.45432613)$, hence the system is open loop unstable. The open loop transfer function from T_c^δ to T^δ is

$$T^\delta(s) = 2.092 \frac{s+2}{s^2 - 2.38s - 1.288} \cdot T_c^\delta(s) \quad (44)$$

which implies that the system has a “stable” zero at $s = -0.5$ at the nominal operating point.

C Extended LQ+I system

The extended LQ problem with integral action is given as follows. The extended system is¹⁰

$$\underbrace{\frac{d}{dt} \begin{pmatrix} x \\ z \end{pmatrix}}_{\tilde{x}} = \underbrace{\begin{pmatrix} A & 0 \\ C & 0 \end{pmatrix}}_{\tilde{A}} \tilde{x} + \underbrace{\begin{pmatrix} B \\ 0 \end{pmatrix}}_{\tilde{B}} u \quad (45)$$

while the extended cost function $\tilde{\mathcal{J}}$ given as

$$\tilde{\mathcal{J}} = \frac{1}{2} \int_0^\infty (e_{\tilde{x}}^T \tilde{Q} e_{\tilde{x}} + e_u^T R e_u) dt \quad (46)$$

where

$$\tilde{Q} = \begin{pmatrix} C^T Q_y C & 0 \\ 0 & Q_z \end{pmatrix}. \quad (47)$$

¹⁰ T_{ref} acts as a disturbance

References

- Brian D. O Anderson and John B. Moore. *Optimal Control: Linear Quadratic Methods*. Prentice-Hall, Inc., 1989.
- Karl Johan Åström and Richard M. Murray. *Feedback Systems. An Introduction for Scientists and Engineers*. Princeton University Press, Princeton, NJ, 2008. ISBN 978-0-691-13576-2.
- Peter Fritzson. *Principles of Object-Oriented Modeling and Simulation with Modelica 3.3: A Cyber-Physical Approach*. Wiley-IEEE Press, Piscataway, NJ, second edition, 2015. ISBN 978-1-118-85912-4.
- Michael A. Henson and Dale E. Seborg. *Nonlinear Process Control*. Prentice Hall, Upper Saddle River, New Jersey, 1997.
- Huibert Kwakernaak and Raphael Sivan. *Linear Optimal Control Systems*. Wiley-Interscience, New York, 1972.
- Bernt Lie, Sudeep Bajracharya, Alachew Mengist, Lena Buffoni, Arun Kumar, Martin Sjölund, Adeel Asghar, Adrian Pop, and Peter Fritzson. Api for accessing openmodelica models from python. In *Proceedings of EuroSim 2016*, Oulu, Finland, 2016, September 2016.
- Jan M. Maciejowski. *Predictive Control with Constraints*. Prentice Hall, Harlow, England, 2002.
- James B. Rawlings, David Q. Mayne, and Moritz M. Diehl. *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, LLC, second edition, 2017.
- Dale E. Seborg, Thomas F. Edgar, Duncan A. Mellichamp, and III Doyle, Frank J. *Process Dynamics and Control*. John Wiley & Sons, Hoboken, NJ, third edition edition, 2011. ISBN 978-0-470-12867-1. ISBN 978-0-470-12867-1.
- Dan Simon. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley-Interscience, Hoboken, New Jersey, 2006.
- Sveinung M. Sund, Marianne Plouvier, and Bernt Lie. Comparison of simulation tools for dynamic models. In *Proceedings, SIMS 2018*, Oslo Metropolitan University, September 2018. SIMS, Linköping University Press.

Second Order Semi-Discrete Scheme for the Solution of Flow in a Venturi Channel

Susantha Dissanayake¹ Roshan Sharma¹ Bernt Lie¹

¹Department of Electrical Engineering, IT and Cybernetics, University of South-Eastern Norway, Porsgrunn, Norway, {roshan.sharma,bernt.lie}@usn.no

Abstract

In many different applications, a Venturi channel is used as a tool to compute fluid flow rates. The Saint-Venant equation is a hyperbolic type Partial Differential Equation (PDE) which can be used to model fluid flows through a Venturi channel. The suitability of the 2nd order Kurganov-Petrova (KP) scheme to solve the hyperbolic PDE for fluid flow in the Venturi channel is studied. A laboratory Venturi rig established at the University of South-Eastern Norway (USN) is used to measure the Steady State (SS) fluid levels along the channel. In this paper, the simulated results are compared with the experimental data. In addition, the simulation results obtained with the second order scheme for solving the Saint-Venant equations are compared with a 1st order numerical scheme. The Froude number for the flow is calculated to check the flow regime changes: from a subcritical flow to a supercritical flow in the Venturi section of the channel. The 2nd order KP scheme is found to be a suitable numerical scheme which can be used to discretize hyperbolic PDEs.

Keywords: Semi-discrete KP scheme, Venturi rig, drill mud flow

1 Introduction

The Kurganov-Petrova (KP) scheme is a 2nd order numerical scheme, which is developed to discretize Hyperbolic Partial Differential Equations (PDEs) in spatial directions (Kurganov and Petrova, 2007). In the development of the KP scheme, the local speed of discontinuity propagation is taken into account [2]. Suitability of the KP scheme to simulate flows of water in a reach of a river has been tested by the authors of this paper (Sharma, 2015; Vytvytskyi et al., 2015; Dissanayake et al., 2016). Here, we consider the usefulness of the 2nd order KP scheme to solve the Saint-Venant equation for fluid flow through a Venturi channel. The simulated results are compared to the experimental data obtained from a Laboratory Venturi channel established at the University of South-Eastern Norway (USN). Moreover, the simulation results with the 2nd order KP scheme is compared with a 1st order scheme. Based on the case study, the usefulness of the KP scheme to solve hyperbolic PDEs for fluid flow through the venturi channel is assessed.

In the Finite Volume Methods (FVM), the properties of the flux are averaged in each Control Volume (CV) (Versteeg and Malalasekera, 2007). A linear reconstruction based on the flux property average of CVs, causes discontinuity at the cell interfaces giving rise to the Riemann problem (Kurganov and Tadmor, 2000); the discontinuity causes two different property values for a single interface of a CV and the uniqueness of solution is lost. Subsequently, this may lead to oscillation in the final solution. In order to deal with such discontinuity problems, several numerical techniques have been developed. Various Riemann solvers can be used to address the problem but they usually consist of a number of intermediate calculations to compute a single value for the CV interface (Kurganov and Petrova, 2007). Due to the number of intermediate calculations involved, Riemann solvers are computationally expensive with slow convergence. The 2nd order KP scheme is a Riemann free solver, which is recognized as an appropriate scheme to address such discontinuity problems with reasonable accuracy and fast convergence (Kurganov and Petrova, 2007).

Use of a Venturi channel as a tool to compute flow rates is well known in many industries, e.g., in oil drilling. The downhole pressure of an oil wellbore is considered as a crucial parameter to be controlled in the drilling operations. Abrupt changes of the pressure of an oil-well might cause minor to major fractures in the wellbore. Such a situation might result in an uncontrolled reduction of oil-well pressure. Subsequently, if the oil well pressure decreases below the formation pore pressure, an unwanted flow of formation fluid into the wellbore occurs. Such a phenomenon is referred to as a kick (Berg et al., 2015). A critical escalation of a kick condition may lead to blow-out from oil wells (Hauge and Øien, 2012). Hence, the kick is a key parameter to observe when drilling is in progress.

Kick is directly related to the drilling fluid circulation. Thus, kick can be identified by applying mass balance to the drilling fluid circulation (Berg et al., 2015). This balance of drilling fluid inflow and outflow of a wellbore can be expressed as follows,

$$\frac{dm}{dt} = \dot{m}_i - \dot{m}_e. \quad (1)$$

Here, \dot{m}_i is the mass flow rate of drilling fluid injected into the well (influent), \dot{m}_e is the mass flow rate of the fluid

flowing out of the well (effluent) and m is the mass of fluid in the well. For the detection of the kick and loss, it is important to accurately measure the flow rate of the fluid flowing out of the well. In oil industries, the use of a measuring paddle in an open channel is a typical/conventional way of measuring the flow rate out of the well. One of the alternatives is to use an open channel system such as the Venturi flume/Venturi channel for return drill fluid flow measurement (Berg et al., 2015).

In this paper, the 2nd order KP scheme is used to discretize the Saint-Venant equation spatially. The objective is to check the suitability of the 2nd order KP scheme to solve hyperbolic type PDEs based on the case study of the solution of the flow in the Venturi channel. The considered laboratory Venturi channel has level sensors in fixed positions which provides the SS fluid level for a known volumetric flow rate. The set of experimental data is compared to the simulation results with the 2nd order KP scheme. In the latter part of the study, both experimental data and the 2nd order simulation results are compared to a set of 1st order simulation results which have been published previously (Agu, 2014).

This paper is arranged as follows. The Venturi channel and its use for flow rate estimation is discussed in section 2. A detailed description of the laboratory Venturi rig is given in the same section, with a set of ground floor tanks, upstream pipe, and the Venturi channel itself. A basic introduction to the Saint-Venant equation together with the 2nd order KP scheme is given in section 3. The model of the laboratory Venturi channel is simulated in section 4. Simulation parameters, variables, and quantities are tabulated in the same section. Section 5 provides a discussion of results: how the simulated results compare with the experimental data. Moreover, simulation results of a 1st order numerical scheme (Agu et al., 2015) are discussed, and the percentage errors of the two different methods are given. Finally, some conclusions are drawn in section 6.

2 The Venturi Channel

The Venturi channel or Venturi flume is an open channel system. The Venturi channel holds a section where the width narrows; the Venturi section. When fluid flows through a Venturi channel, the fluid level is decreased in the Venturi section due to an acceleration of the flow velocity. In the SS fluid flow, a correlation between volumetric flow rate and the fluid height can be derived. Consequently, either volumetric flow rate or fluid height is computed. Typically, the fluid flow rate is calculated based on the level measurements. The Venturi channel has two different names based on applications. The Venturi tube is used for relatively smaller flow (e.g. mm^3/s), while the Venturi channel is used for large scale flows with up to millions of cubic meter per hour units. The Venturi tube was developed by G.B. Venturi based on the Bernoulli principle,

$$\frac{v^2}{2} + gh + \frac{p}{\rho} = \text{constant}. \quad (2)$$

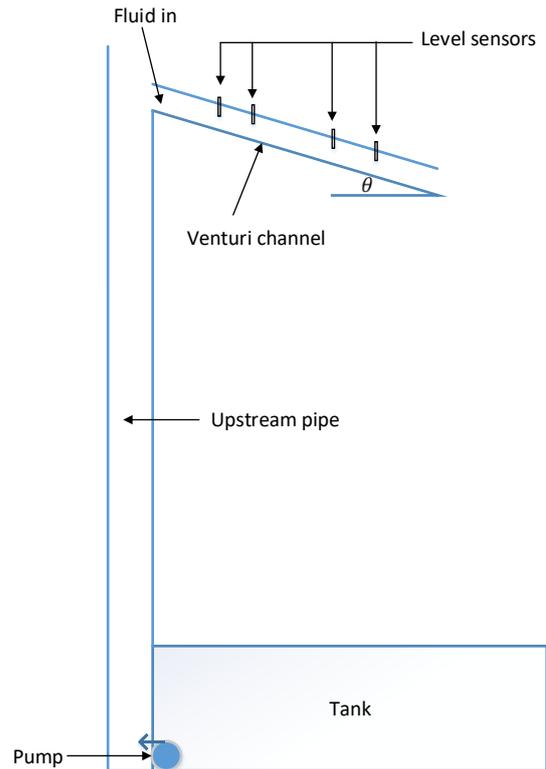


Figure 1. Laboratory Venturi rig.

Here, v denotes the velocity of the fluid, g is the acceleration of gravity, h is the height elevation from the reference plane, p is the pressure of the fluid, and ρ denotes the density of the fluid.

In the laboratory Venturi rig, the volumetric flow rate is an adjustable variable. The SS fluid height is measured by level sensors at fixed positions. The USN Venturi rig is designed for doing experiments on the flow of drill mud: more precisely, experiments on flow of artificial drill mud. An artificial drill mud is a substitute fluid with properties that match with the real drill mud.

The complete set-up has ground tanks for storing artificial drill mud. The system has an upstream pipe which pumps fluid from the tanks and up to some elevation (2nd floor). At the end of the upstream pipe, the Venturi channel is attached. A schematic diagram of the complete set-up of the Venturi rig at USN is illustrated in Figure 1.

A detailed sketch of the Venturi channel is given in Figure 2. The dimensions of the laboratory Venturi channel are tabulated in Table 1. The total length of the Venturi channel is 3.7 m.

Table 1. Dimensions of the laboratory venturi channel

Symbol	Description	Values
B_T	Top width of entrance to the converging section	0.455 [m]
b_0	Bottom width of entrance to the converging section	0.2 [m]
b_{*0}	Bottom width of exit from the converging section	0.1 [m]
H	Total height of the channel	0.35 [m]
α	Angle of the inclined sides of the channel	70°
L_a	Length of the upstream section	2.95 [m]
L_b	Length of the converging section	0.15 [m]
L_c	Length of the throat section	0.2 [m]
L_d	Length of the diverging section	0.15 [m]
L_e	Length of the downstream section	0.25 [m]
θ	Angle of channel bed	0.08°

3 Governing Equations and the 2nd Order KP Scheme

3.1 Governing Equations

The Saint-Venant equation/Shallow Water Equation is a hyperbolic type PDE, which has versatile use in fluid dynamics: many different fluid dynamic applications such as fluid flows in open channels, water flow in the rivers to estimate wave propagation, compute tsunami wave, etc. (Sharma, 2015; Vytvytskyi et al., 2015; Dissanayake et al., 2016). The Saint-Venant equation is given in Equation 3.

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} = S \quad (3)$$

Here, U stands for fluid property such as the flow rate and fluid level, $F(U)$ represents physical fluid flux and S is a source term. x and t are spatial and time coordinates respectively. More precisely, the Saint-Venant equation with source term is given as (Sharma, 2015),

$$\frac{\partial U(x,t)}{\partial t} + \frac{\partial F(x,t,U)}{\partial x} = S(x,t,U), \quad (4)$$

$$U = (A, Q)^T, \quad (5)$$

$$F = \left(Q, \frac{Q^2}{A} + gI_1 \cos(\theta) \right)^T \quad (6)$$

$$S = (0, -gI_2 + gA(S_0 - S_f)). \quad (7)$$

Here, A is the cross-sectional area, Q is the volumetric flow rate and g is acceleration due to gravity. The S term reflects the source terms such as the expressions of friction terms which give resistance against the fluid flow. I_1 and I_2 in Equations 6 and 7 are functions of the geometry of the channel. The term gI_1 gives the hydrostatic force in the cross-sectional area of the flow and I_1 is expressed as,

$$I_1 = h^2 \left(\frac{W}{2} + h \frac{S_L}{3} \right). \quad (8)$$

Here W is the width of the base of the channel and S_L is the slope of the side wall (the laboratory Venturi rig is trapezoidal). If the channel is rectangular, then S_L become zero. h is fluid level. I_2 in Equation 7 represents the pressure force in the fluid volume, which occurs due to channel width and slope variations along the axial direction.

$$I_2 = h^2 \left(\frac{1}{2} \frac{dW}{dx} + h \frac{dS_L}{3 dx} \right). \quad (9)$$

S_f in Equation 7 is friction term and S_0 stands for bed slope which is expressed as,

$$S_0 = -\frac{\partial B}{\partial x}. \quad (10)$$

Here B stands for the bottom elevation. For Newtonian fluids, the friction term S_f in Equation 7 is calculated as (Agu et al., 2015),

$$S_f = v|v|n_M^2 R_h^{-4/3}. \quad (11)$$

Here n_M is Manning's roughness coefficient. v is the velocity of the fluid and is calculated as,

$$v = \frac{Q}{A} \quad (12)$$

R_h in Equation 11 refers to hydraulic radius and is expressed as in Equation 13.

$$R_h = \frac{A}{P_{\text{wet}}}. \quad (13)$$

P_{wet} is the wetted perimeter which is illustrated in Figure 3.

P_{wet} in Equation 13 is the sum of all lengths of wetted surface by the fluid, which is expressed as,

$$P_{\text{wet}} = \sum_{i=0}^{\infty} l_i. \quad (14)$$

Here l_i is the length of each surface which has contact with the fluid flow in the channel.

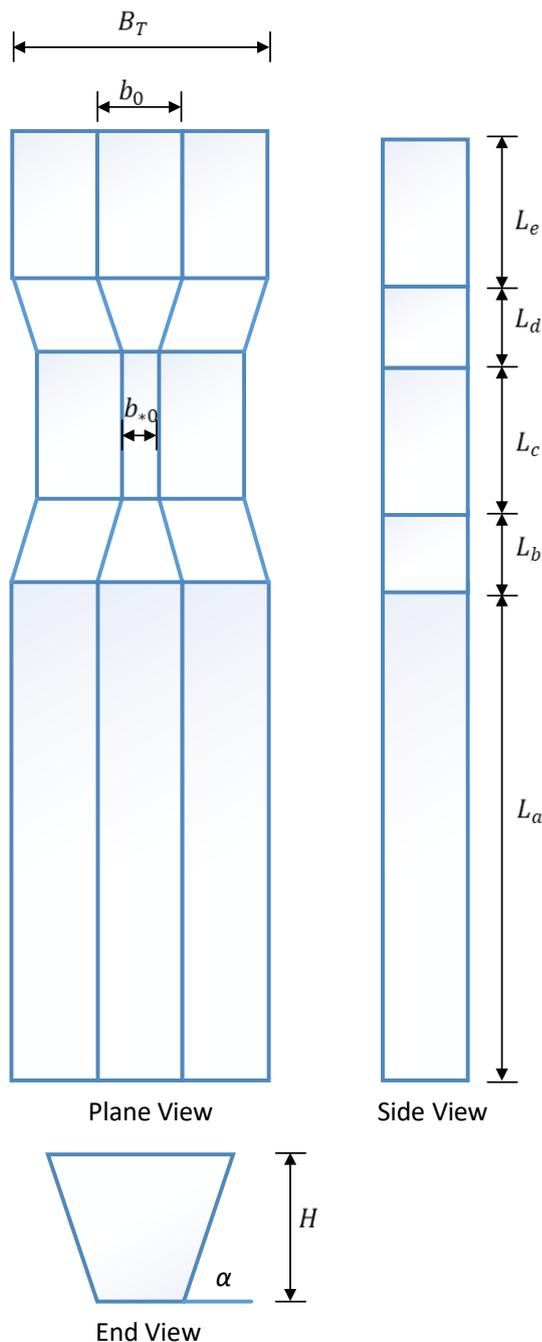


Figure 2. Illustration of the Laboratory Venturi channel.

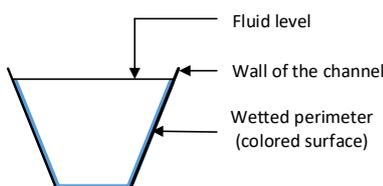


Figure 3. The wetted perimeter of a trapezoidal channel.

Many common fluids are assumed as Newtonian, e.g., water. A Newtonian fluid is a fluid for which the viscous stresses developed due to the flow of fluid are linearly proportional to local strain rates or rate of change of deformation (Versteeg and Malalasekera, 2007). The drilling fluid however is a non-Newtonian fluid.

3.2 Non-Newtonian Fluid

Non-Newtonian fluids are fluids for which the viscous stresses developed due to the flow of fluid are not linearly proportional to the rate of deformation. Polymers, paints, and Oobleck¹ are common examples of non-Newtonian fluids. Drill mud consist of fines of drill cuttings in a mixture of liquid, and is exhibits a non-Newtonian fluid behaviour.

Therefore, fluids used in our simulations are considered as non-Newtonian fluid. The friction term in the Saint Venant equation is modified to account for more types of fluids. Subsequently, the Venturi rig simulation is extended to the non-Newtonian fluid.

Shear stress, τ , is a component of stress applied on the material cross-sectional area. Shear stress is written as,

$$\tau = \frac{F}{A}. \quad (15)$$

Here, F refers to forces applied on the cross sectional area A . For a flowing fluid, the shear stress becomes,

$$\tau = \mu \left(\frac{\partial v}{\partial x} \right). \quad (16)$$

Here μ is the dynamic viscosity of the fluid. The term $\frac{\partial v}{\partial x}$ is the velocity gradient. x signifies the spatial coordinate.

For non-Newtonian fluid, shear stress is derived by applying the momentum balance. Then τ is written as in Equation 17, the well-known Herschel-Buckley model.

$$\tau = \rho g S_f (R_h - h) = \tau_h + K \left(\frac{\partial u}{\partial x} \right)^n. \quad (17)$$

In Equation 17, ρ signifies the density of the fluid and g denotes the acceleration due to gravity. h is height above the reference plane. K is the consistency index, u in the equation stands for the velocity component in the positive x direction, τ_h is yield stress and n is the flow index. For $n < 1$, the fluid is shear thinning. If $n > 1$, the fluid is shear thickening. If $n = 1$ and $\tau_h = 0$, Equation 17 reduces to Newtonian model (Agu, 2014).

For non-Newtonian fluids, internal frictional shearing stress has a major impact. Then from Equation 17, S_f is derived as (Agu et al., 2015),

$$S_f = \frac{K}{4\rho g R_h} \left(\frac{|v|}{h} \frac{1+2n}{n} \right)^n. \quad (18)$$

¹Oobleck is corn flour in water. The name originates from a story-book: Bartholomew and the Oobleck by Dr. Seuss (Theodor Geisel).

3.3 The 2nd Order KP Scheme

The 2nd order KP scheme is used to discretize the Saint-Venant equation. The 2nd order KP scheme is semi-discrete in nature, and is used in the spatial discretization of the model equations (Kurganov and Petrova, 2007). Spatial discretization of the Saint-Venant equation results in a set of Ordinary Differential Equations (ODEs). Different time integrators can be used to solve such ODEs. However, the chosen time integrator should essentially be of order equal to or lower than the order of the spatial discretization to attain fast convergence. Also, (Dissanayake et al., 2016) found that fixed step length solvers are more suitable for solving such a set of ODEs. The ODEs resulting from the spatial discretization are of 2nd order. Therefore, a 2nd order Runge-Kutta (RK2) method is used as a time integrator.

The set of ODEs generated by the spatial discretization of the Saint-Venant equation using the KP scheme is written as (Sharma, 2015),

$$\frac{d}{dt} \bar{U}_j = - \frac{H_{j+\frac{1}{2}}(t) - H_{j-\frac{1}{2}}(t)}{\Delta x}, \quad (19)$$

$$H_{j+\frac{1}{2}}(t) = \frac{a_{j+\frac{1}{2}}^+ F(U_{j+\frac{1}{2}}^-, B_{j+\frac{1}{2}}) - a_{j+\frac{1}{2}}^- F(U_{j+\frac{1}{2}}^+, B_{j+\frac{1}{2}})}{a_{j+\frac{1}{2}}^+ - a_{j+\frac{1}{2}}^-} + \frac{a_{j+\frac{1}{2}}^+ a_{j+\frac{1}{2}}^-}{a_{j+\frac{1}{2}}^+ - a_{j+\frac{1}{2}}^-} (U_{j+\frac{1}{2}}^+ - U_{j+\frac{1}{2}}^-), \quad (20)$$

$$H_{j-\frac{1}{2}}(t) = \frac{a_{j-\frac{1}{2}}^+ F(U_{j-\frac{1}{2}}^-, B_{j-\frac{1}{2}}) - a_{j-\frac{1}{2}}^- F(U_{j-\frac{1}{2}}^+, B_{j-\frac{1}{2}})}{a_{j-\frac{1}{2}}^+ - a_{j-\frac{1}{2}}^-} + \frac{a_{j-\frac{1}{2}}^+ a_{j-\frac{1}{2}}^-}{a_{j-\frac{1}{2}}^+ - a_{j-\frac{1}{2}}^-} (U_{j-\frac{1}{2}}^+ - U_{j-\frac{1}{2}}^-), \quad (21)$$

where $a_{j\pm\frac{1}{2}}^\pm$ are the one-sided local speeds of wave propagation and $U_{j\pm\frac{1}{2}}^\pm$ are property fluxes at indexed positions.

4 Simulation Setup

The model of the laboratory Venturi channel is simulated for the flow of artificial drill mud. Important simulation parameters are summarized in Table 2. MATLAB² is used as a simulation software.

5 Results and Discussion

Width variation along the length of the Venturi channel is plotted in Figure 4. According to the dimensions of the laboratory Venturi channel (Table 1), the width of the

Table 2. Simulation Parameters.

Length of the Venturi channel	3.7 [m]
Density of the fluid	1109 [kgm ⁻³]
Acceleration due to gravity	9.81 [ms ⁻²]
Channel side angle	70 ⁰
Channel bed angle	0.08 ⁰
Volumetric flow in	0.00451 [m ³ s ⁻¹]
Volumetric flow out	0.00451 [m ³ s ⁻¹]
Number of CVs	50
Time step (Δt)	0.02 [s]

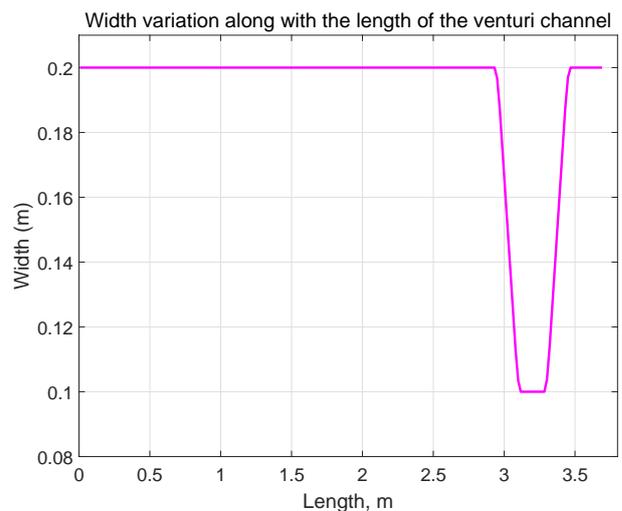


Figure 4. Width variation in the Venturi rig as a function of the length of the channel.

channel varies gradually between 0.2 m and 0.1 m at the throat section of the channel. The SS fluid level along the whole length of the Venturi channel obtained using the KP scheme is shown in Figure 5.

According to Figure 5, the initial fluid level is about 0.076 m from the reference plane (which is the bottom of plane of the channel). The SS fluid level decreases in the Venturi section: after 3 m along the channel, the fluid level starts to decrease and continues to decrease until position ca. 3.4 m along the Venturi channel. According to the dimensions of the Venturi channel (Table I), the Venturi section starts at position 2.95 m. In other words, the width of the channel is gradually reducing from 0.2 m to 0.1 m after 2.95 m from the starting point of the Venturi channel. Downstream from the Venturi section: after 3.45 m, the channel width is back again to 0.2 m. Downstream from the Venturi section, the level of the fluid remains constant: more or less steady level throughout the rest of the length: $3.45\text{m} \leq \text{Length} \leq 3.6\text{m}$. Visual observation and level sensor readings support these changes in flow behavior. Experimental data from the Venturi rig is plotted with the simulated SS fluid level, in order to check the accuracy of the simulation.

²MATLAB R2014a

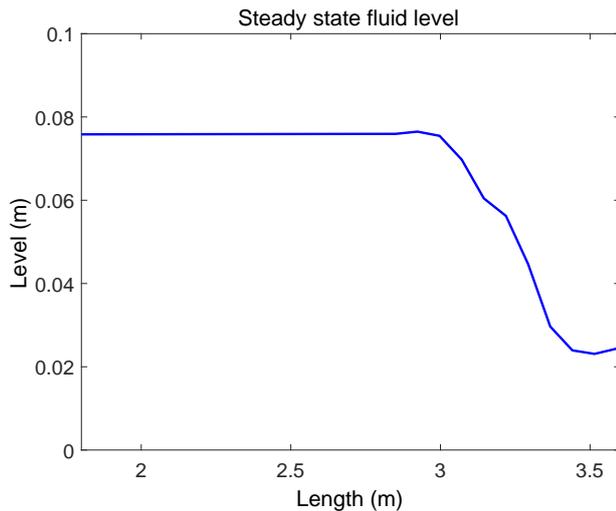


Figure 5. Steady State fluid level in the Venturi channel using KP scheme.

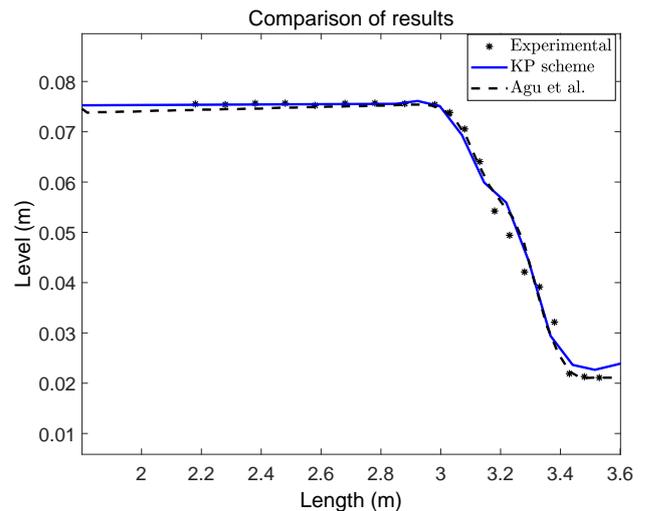


Figure 7. Comparison of the simulation results from 2nd order KP scheme and 1st order scheme with experimental data.

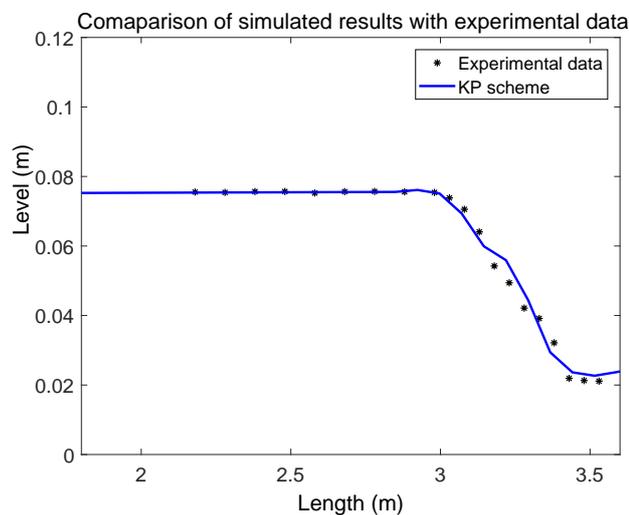


Figure 6. Comparison of simulation results with KP scheme with experimental data.

5.1 Comparison of Simulated Results With Experimental Data

The laboratory Venturi rig has a few level sensors which can measure the fluid level at several positions in the Venturi channel. Simulated data and experimental data have been plotted and compared in this section. Figure 6, shows the simulation results together with the experimental data.

In a previous work by (Agu et al., 2015), a similar study of the exact same Venturi channel was carried out. In their study, the authors derived the ODEs by the spatial discretization of the Saint-Venant equation with a 1st order scheme. Both the 1st order simulation results [11], and the 2nd order simulation results are plotted together with the experimental data in Figure 7. (Agu et al., 2015) used MATLAB ode solver ode15s in their simulations.

From Figure 7, both the 1st order and the 2nd order

scheme produce similar results at the steady state. In the Venturi section, both schemes follow the similar pattern. However, both schemes show a small deviation from the experimental data. After the Venturi section, the 1st order model approximation is closer to the experimental data; the 2nd order KP scheme shows a small deviation.

5.2 Error Computation

Simulated results have only a minor deviation from the experimental data. Fluid level readings have been taken at fixed positions along the Venturi channel. These fixed positions do not necessarily fall at the centers of the grid cells of the FVM. The simulated results have thus been interpolated to compute the values of the fluid level such that they correspond to these fixed positions (where level sensors are placed) and then only compared with the experimental data. These interpolated values and the experimental data are plotted together with the simulated results in Figure 8.

In a similar way, interpolated values and the experimental data are plotted for the study of (Agu et al., 2015) where a first order scheme for spatial discretization is used. Figure 9 shows the interpolated points, experimental data and the simulation of (Agu et al., 2015).

The difference between the interpolated values and the experimental data are the error of the simulated results. Percentage error of the 2nd order KP scheme and the 1st order scheme of (Agu et al., 2015) are plotted in Figure 10.

According to Figure10, the 2nd order KP scheme has negligible percentage error upstream from the Venturi section: $2.2\text{m} \leq \text{Level} \leq 3\text{m}$. The simulation results of (Agu et al., 2015) exhibit a small deviation for this section of the channel. In the Venturi section both simulated results show more or less similar behavior. However, the simulation of (Agu et al., 2015) has a slightly larger overshoot and undershoot in the Venturi section. The simula-

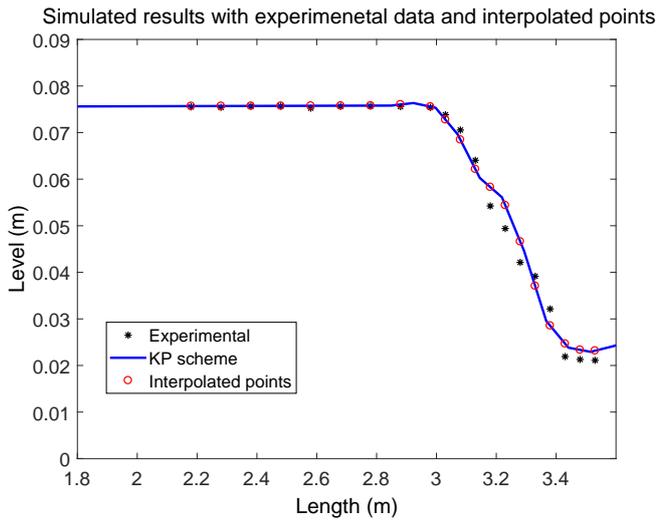


Figure 8. Simulation results with the 2nd order KP scheme, interpolated points, and experimental data.

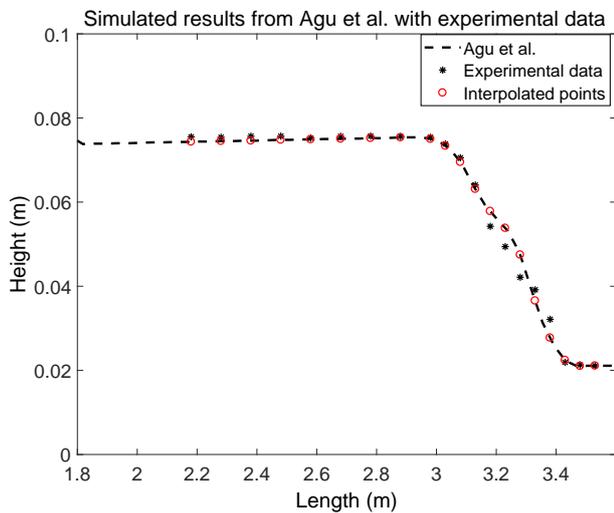


Figure 9. Simulated results of (Agu et al., 2015), experimental data and interpolated points.

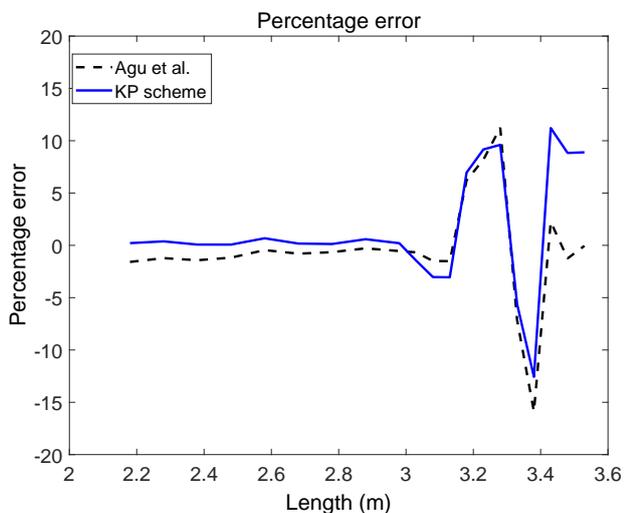


Figure 10. Percentage error for 2nd order KP scheme and the 1st order scheme of (Agu et al., 2015).

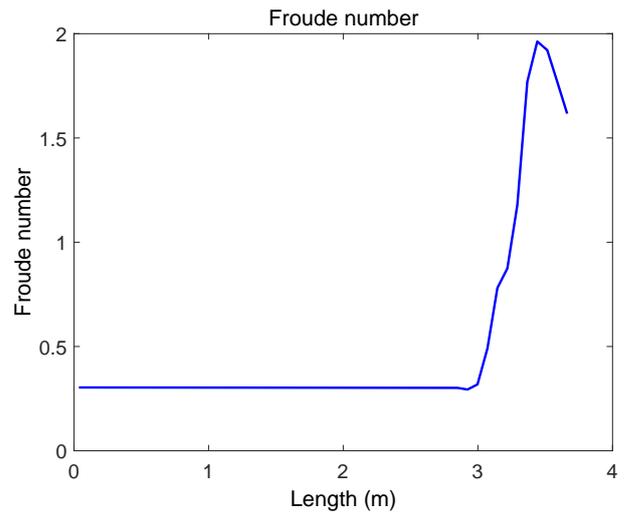


Figure 11. Froude number along the length of the venturi channel.

tion of (Agu et al., 2015) produces a slightly less percentage error after the Venturi section.

As a whole, the KP scheme produces acceptable results for the whole length of the Venturi channel apart from the length: $3.4\text{m} \leq \text{Length}$ (the part which is not useful and thus not used for the estimation of fluid flow rate).

5.3 Flow Regimes

Flow velocity changes along the length of the Venturi channel can be found by computing the Froude number (Fr) which is a dimensionless number that explains the speed-length ratio, and is written as,

$$\text{Fr} = \frac{u_0}{\sqrt{g_0 l_0}}. \quad (22)$$

Here u_0 is characteristic flow velocity, g_0 is characteristic of the external field: more precisely, acceleration of gravity. l_0 stands for characteristic length. Fr for the current scenario can be written as,

$$\text{Fr} = \frac{u_0}{v}. \quad (23)$$

Here v is the characteristic fluid propagation velocity and is written as,

$$v = \sqrt{gh}, \quad (24)$$

where g is gravitational constant and h is fluid level. The Froude number computed along the Venturi channel is plotted in Figure 11. When Fr increases and becomes greater than 1, this means that the flow changes from subcritical to supercritical. In the Venturi section of the channel, flow becomes supercritical.

5.4 Flow Rate Calculation

When the fluid flow is at SS, fluid level is measured. When the flow rate increases, the level increases. Such increment of the height of the flowing fluid increases the wetted

perimeter. The wetted perimeter has a correlation with the hydraulic radius (Equation 13). According to Equation 2, total energy in the system is preserved. Based on the known parameters, back calculation can be used to compute volumetric flow rate of the fluid.

6 Conclusions

Based on this study, the KP scheme is recognized as a suitable numerical scheme to discretize the Saint-Venant equations, which is a hyperbolic PDE. ODEs resulting from the spatial discretization has a 2nd order of accuracy. Hence when the KP scheme is compare with the 1st order scheme, the KP scheme shows increment in the accuracy. According to the percentage error comparison, it is concluded that increment of the order of the spatial discretization improve the accuracy. Flow regime changes along the Venturi channel is observed with the Frode number. The KP scheme successfully recognized the flow regime change from subcritical to supercritical. For the Venturi section both schemes show deviation from the experimental results. However, the 1st order scheme shows slightly larger overshoot and undershoot. Hence the KP scheme can be used to solve the Saint-Venant equations for the flow through a venturi channel.

Acknowledgements

The economic support from The Research Council of Norway and Statoil ASA through project no. 255348/E30 “Sensors and models for improved kick/loss detection in drilling (Semikidd)” is gratefully acknowledged. Mr. Cornelius Emeka Agu supplied materials, simulation results, experimental data, etc. This is gratefully acknowledged.

Kindly convey sincere thanks to the other project group members: Janitha Chandimal, Junyang Mao, and Obianuju Ezuka.

References

- C. E. Agu. Model based estimation of drilling mud flow using a venturi channel. Master’s thesis, Faculty of Technology, Telemark University College, Porsgrunn, Norway, 2014.
- C. E. Agu, B. Lie, and G. Elseth. Simulation of transcritical flow in hydraulic structures. In *Proceedings of the 56th Conference on Simulation and Modelling (SIMS 56)*, number 119, pages 369–375, Linköping University, Sweden, October 7-9 2015. ISBN 978-91-7685-900-1. doi:10.3384/ecp15119369.
- C. Berg, A. Malagalage, C.E. Agu, G.-O. Kaasa, and B. Lie. Model-based drilling fluid flow rate estimation using venturi flume. *2nd IFAC Workshop on Automatic Control in Offshore Oil and Gas Production*, 48(6):171–176, 2015.
- S. Dissanayake, R. Sharma, and B. Lie. Semi discrete scheme for the solution of flow in river tinnelva. In *Proceedings of EUROSIM 2016*, pages 134–139, Oulu, Finland, September 13-16 2016. IEEE. ISBN ISBN 978-1-5090-4119-0.
- S. Hauge and K. Øien. Deep water horizon: Lessons learned for the norwegian petroleum industry with focus on technical aspects. *Chemical Engineering Transactions*, 26:621–626, 2012. ISSN 1974-9791. doi:10.3303/CET1226104.
- A. Kurganov and G. Petrova. A second-order well-balanced positivity preserving central-upwind scheme for the saint-venant system. *Communications in Mathematical Science*, 5:133–60, 2007.
- A. Kurganov and E. Tadmor. New high-resolution central schemes for nonlinear conservation laws and convection-diffusion equations. *Journal of Computational Physics*, 160(1):241 – 282, 2000. ISSN 0021-9991. doi:10.1006/jcph.2000.6459.
- R. Sharma. Second order scheme for open channel flow. Technical report, USN Open Archive, University of South-Eastern Norway, 2015. URL <http://hdl.handle.net/11250/2438453>.
- H. K. Versteeg and W. Malalasekera. *An introduction to computational fluid dynamics*. Pearson Education, Upper Saddle River, United States, 2nd edition, 2007. ISBN 9780131274983.
- L. Vytvytskyi, R. Sharma, and B. Lie. Model based control for run-of-river system. part 1: Model implementation and tuning. *Modeling, Identification and Control*, 36(4):237–249, 2015. doi:10.4173/mic.2015.4.4.

The effects of fan and door opening on a cold storage room: a numerical study

Azada Ayarmal Ole Melhus Arnab Chaudhuri

Department of Civil Engineering and Energy Technology
OsloMet — Oslo Metropolitan University, Oslo, Norway.

azada@me.com, ole.melhus@oslomet.no, arnab.chaudhuri@oslomet.no

Abstract

We present numerical analysis of a cold storage room with a product load, a fan, and a sliding door by solving compressible form of the conservative 3D Navier-Stokes equation (without Boussinesq approximation) together with the energy and mass fraction equations. The commercial computer program StarCCM+ is used for this purpose. The sliding door is treated with an overset mesh on a suitable background mesh. A rigid body translation is applied by a field-function to model the opening and closing of the door. The preliminary study captures essential features of the effect of the energy transport and the cooling of the product load by the fan and the hot, moist air exchange through the doorway. The airflow behavior and heat/mass transfer mechanisms are studied and presented for several numerical experiments.

Keywords: Cold storage room, Natural convection, Forced convection, Numerical simulations

1 Introduction

Cold storage rooms are designed for preservation of perishable commodities at low temperatures and are widely used in food/pharmaceutical/photographic industries. Multiple coupled interactions, involving heat-mass-momentum transfer and phase change of constituent components play important roles during the operating condition of a cold store in such applications. Temperature and moisture control are the key parameters for optimal operation and energy consumption. Numerous guidelines are available for successful design and implementation for cold storage rooms. Among several other factors, infiltration of hot/moist air through open doors may become crucial for the performance of the rooms (infiltration load). Study of air infiltration is also very important for analyzing indoor air pollutant behavior, smoke control, design of clean room, hospital operating theatres, paint booths, restaurant kitchen etc.

In their study, (Wilson and Kiel, 1990) made experimental measurements in density driven counterflow through an open door. With a simple quasi-steady inviscid model, they provided a set of practical equations for estimation of the flow through an open door in a sealed room. A doorway opening possibly can result in trans-

port of infectious air out of the isolation room, leading to a breakdown in isolation conditions (Tang et al., 2005) in negative-pressure isolation room. In this context, the study of different types of doors (variable speed single and double, sliding and hinged doors) with moving human figure are presented in (Tang et al., 2013) by a 1:10 scale water tank. They reported that double-hinged doors are associated with the greatest risk of leakage into or out of the room. The decreasing order risk of leakage reported as follows: double-hinged > single-hinged > double sliding > single sliding doors. Apart from experimental approach, computational fluid dynamics tools are more cost effective alternatives to study the fundamental processes involved during the operation of a cold storage room. Nevertheless, resolving complex coupled interactions via realistic numerical models is also challenging. Large-eddy simulation (LES) is performed by (Choi and Edwards, 2012) to study the human-induced contaminant transport in room-compartments. The effects of human and door motions and vent system activity in vestibule connected compartments (contaminated and clean room) are reported. According to their study, swinging door motion induces up to six times more compartment-to-compartment transport than sliding door motion. The human wake effect enhances the compartment-to-compartment transport. Experiments and unsteady Computational Fluid Dynamics (CFD) simulations are presented by (Chang et al., 2016) to investigate the contaminant inleakage produced by the hinged door opening and closing. The indoor air flow and mass exchange induced by the rotating motion of a hinged door separating two rooms is investigated by (Papakonstantis et al., 2017).

Experimental studies of air flow, temperature and humidity patterns in cold rooms can be found in (Duret et al., 2014; Hoang et al., 2015; Scaar et al., 2017). A simplified heat transfer modeling in a cold room with loads is presented by (Laguerre et al., 2015). CFD studies of airflow inside cold stores with/without loads, usually accounts local pressure loss source terms, to model fan and cooler unit (Hoang et al., 2000; Nahor et al., 2005; Akdemir and Bartzanas, 2015). In their study, (Wang et al., 2015), simulated the air flow and temperature variation in an ice drink refrigeration cabinet, during an automated dynamic cycle of switching on and switching off. They used a porous-jump model for thin perforated plate and pressure drop of

the fan for air circulation.

The literature review presented above reveals that the detailed numerical study of volume exchange with sliding doors or studies dealing with the infiltration via doors for cold store setup are not abundant in literature. Only a few and very recent works report such attempts, for a cold room (Foster et al., 2003; Carneiro et al., 2017) and for refrigerated vehicles (Zhang et al., 2017). The present study is aimed to perform a numerical analysis by solving compressible form of the conservative 3D laminar Navier-Stokes equation (without Boussinesq approximation) together with the energy and mass fraction equations for a cold storage room with a product load, a fan, and a sliding door. The commercial computer program StarCCM+ is used for this purpose.

The paper is organized as follows. In section 2 we present the governing equations and methodology for the study of cold storage room. The problem setup is given in section 3, followed by the results and discussion in section 4. Finally, the conclusions are drawn in section 5.

2 Method

2.1 Governing equations

The governing equations for the gaseous phase (inside the cold storage room) is the 3D compressible Navier-Stokes system of equations together with mass, species conservation and energy conservation equations. The general transport equation for any conserved property ($\rho\phi$) can be expressed as in the following standard (Versteeg and Malalasekera, 2007) form:

$$\frac{\partial \rho\phi}{\partial t} + \vec{\nabla} \cdot \rho\phi\vec{v} = \vec{\nabla} \cdot \Gamma\vec{\nabla}\phi + S_\phi. \quad (1)$$

The left hand side consists of the temporal term and the advection term. On the right hand side, the terms signify the diffusion term and the generation term respectively. The solid phase continuum (a block of solid ice as a load in the cold storage room) is governed by the energy conservation equation without any advection or source term. We solve the unsteady Reynolds averaged Navier-Stokes (RANS) formulation. Realizable $\kappa - \epsilon$ model is used for turbulence. The gravity term in the momentum equations is treated directly without Boussinesq approximation. Newtonian fluid and ideal gas law assumptions serve as the constitutive relation and the equation of state. These close the system of equations with appropriate boundary conditions.

The finite volume method (FVM) based commercial computer program StarCCM+ is used to solve the governing equations in the computational domain. For the present study, the gas continuum consists of two species, namely air and water (H₂O) vapor inside the cold storage room. We solve conjugate heat transfer problem by setting the link between the solid phase and the surrounding

gaseous medium of the room. The energy and momentum source terms are assigned to represent the effect of the cooler fan in the room without modelling the detail of the cooler unit. The cases with sliding door of the room are handled with the overset mesh setup of StarCCM+. The background mesh (room region) and the overset mesh (door with a surrounding region of choice) are generated for this purpose. A time varying rigid body translation is applied setting the motion parameters associated with the door geometry for the operation of the sliding door opening and closing. Unsteady simulations are performed accordingly. The overall setup of the different test cases is described in section 3. The brief description of the numerical schemes used in the present study is as follows. Second order schemes for convection, hybrid Gauss least-squared gradient method based 2nd order schemes for diffusion and Venkatakrisnan limiter function are chosen in solver setup. First order Euler implicit scheme is used for time integration with a time step $\Delta t = 0.1$ s. Algebraic Multi-Grid (AMG) techniques are also invoked with the setup mentioned above.

3 Problem setup

3.1 Opened/closed door and sliding door

A simplified cold storage room with a cooler fan and a sliding door has been considered (see the schematic in figure 1). Table 1 summarizes the Length (L), Breadth (B) and Height (H) of different geometries for the test cases with the fixed door (either fully open or fully closed). The dimension of the door is 1m×2m. All walls of the room are considered as adiabatic. Pressure outlet conditions are set at the door when it is kept open with a prescribed pressure and temperature of the environment. The initial condition of room is set as: temperature, $T = 243$ K, the mass fraction of air $Y_{\text{air}} = 0.9$, the mass fraction of moisture $Y_{\text{H}_2\text{O}} = 0.1$, velocity field $\vec{v} = 0$, pressure $P = 0$. The solid ice is kept at: temperature $T = 255$ K and no-slip wall is satisfied with an appropriate contact interface (with zero resistance) for conjugate heat transfer calculations. At the opened door surface: environment temperature $T_{\text{env}} = 293$ K, $Y_{\text{air}} = 0.7$, $Y_{\text{H}_2\text{O}} = 0.3$, $P = 0$. The cooler fan specifications are prescribed with an energy source -100 W, a momentum source $= 2 \text{ N/m}^3$ and a momentum source velocity derivative $= 2 \text{ kg/m}^3 \cdot \text{s}$.

Table 1. Dimensions for problem setup.

Geometry	H (m)	L (m)	B (m)	Area (m ²)	Volume (m ³)
Cold room	2.3	3.7	3.1	11.5	26.4
Ice block	2.0	1.0	1.0	1.0	2.0
Cooler fan	0.3	0.5	0.3	0.15	0.045

Table 2. Test cases.

Case 1	Opened door with fan-off
Case 2	Opened door with fan-on
Case 3	Closed door with fan-on

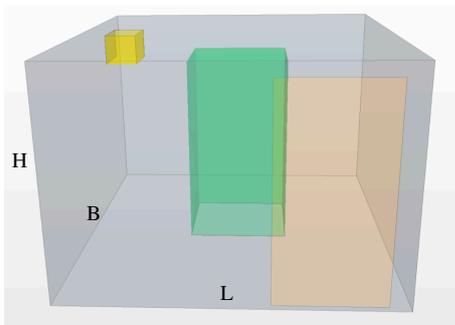


Figure 1. Computational domain of the storage room.

We will first present the results of unsteady simulations with following cases (see table 2) with either closed door or opened door situations (subsection 4.1). The cases with sliding door setup will be presented subsequently in subsection 4.2.

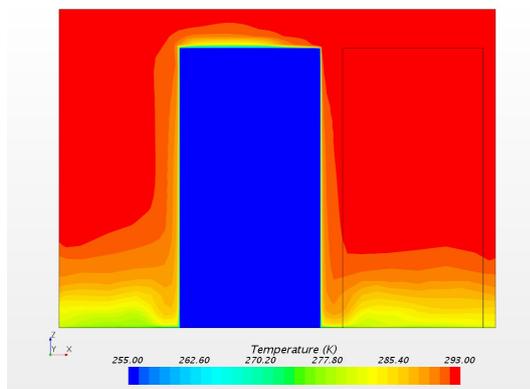


Figure 2. Case 1: contours of T on a $z-x$ plane approximately mid location of the load.

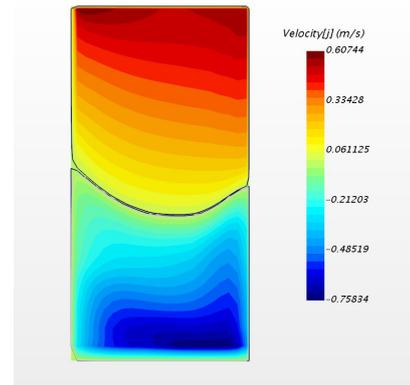


Figure 3. Case 1: contours of y -component of velocity through the door. Black curves represent zero velocity.

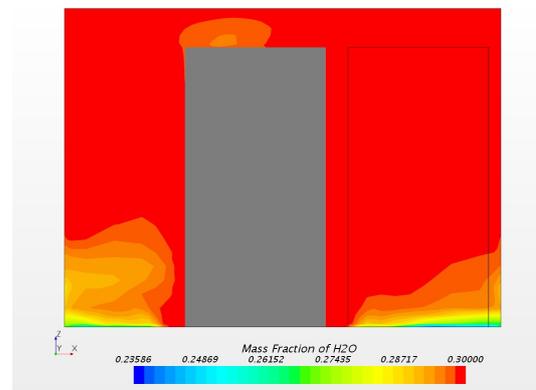


Figure 4. Case 1: contours of Y_{H_2O} .

4 Results and discussion

4.1 Room with fully opened/closed door

For Case 1, the door is suddenly set as fully opened. The unsteady simulation is performed (till $t = 120s$) according to the boundary condition mentioned in section 3. During this time, the air exchange occurred through the door. The room temperature increases as the hot and moist air flows in, through the upper section of the door. Around this time the major portion of the room appears to reach the outdoor temperature (see Fig 2). However, the cold air leaves through the lower section of the door. This is evident from the lower temperature around floor level of the compartment. Figure 3 shows the velocity distribution through the door. The hot air gets cooled as it circulates around the cold product block. From the velocity distribution (see Figure 3), it is clear that the maximum inflow velocity at the upper level of the doorway is lower than the maximum outflow velocity at the bottom level of the doorway. The mass fraction of the water vapor also rapidly gets increased to reach the outdoor condition by this exchange. Figure 4 shows the distribution of the water vapor in the room. The temperature on the surface of the product block is shown in figure 5. As expected, due to the proxi-

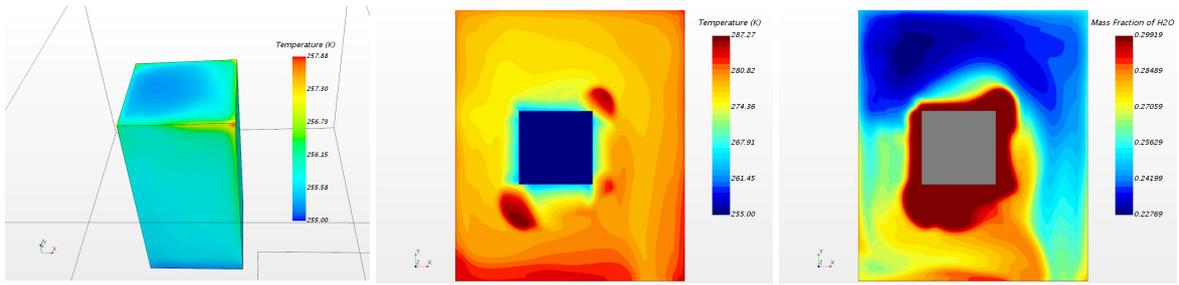


Figure 5. Case 1: T contours at the product surfaces (left), contours of T and Y_{H_2O} at the floor level (middle and right).

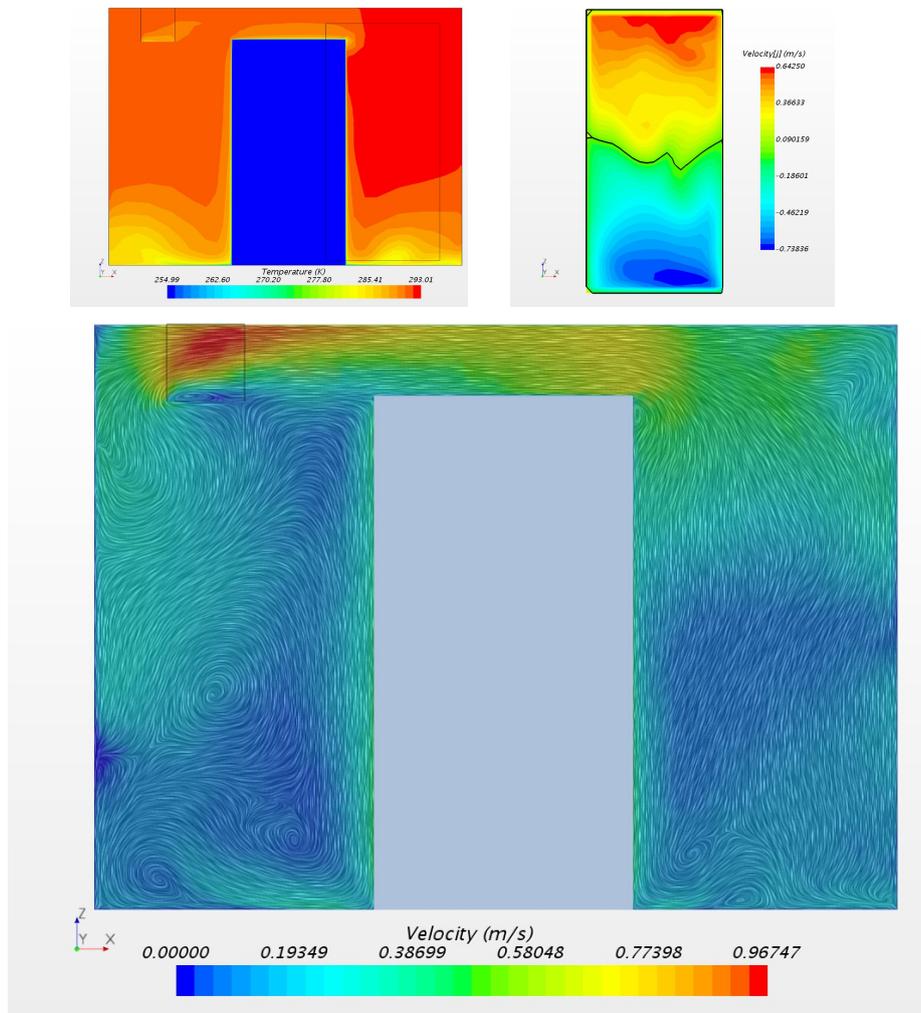


Figure 6. Case 2: contours of T (top left), contours of y -component of velocity through the door (top right) and line integral convolution (LIC) of \vec{v} (bottom). Black curves represent zero velocity.

mity and the exposure to the doorway, the temperature of the nearest surfaces (upper portions) of the product block is found to be higher than the other surfaces. Note that, the temperature of the lower portions of the near-doorway surfaces is lower. This is in accordance with the distribution of temperature in the room as well as near the floor level (see figure 5). Also, note that, the distribution of Y_{H_2O} reveals the nature of advection and diffusion around the ice block. More and more moist air are dragged along the surfaces of the block downwards due the downward bulk movement of air associated with the air exchange process incepted by the opened door.

Case 2 is simulated with similar boundary conditions as Case 1 with the added effect of a cooling fan unit. The energy and momentum source terms are invoked to model the cooler fan as mentioned in section 3. From figure 6, it can be seen that the a more asymmetric distribution of the temperature field in this plane compared to Case 1. The overall distribution reveals that the temperature of the room increases. The cooler capacity (set by the energy source) is insufficient to cool the room during the 120 seconds time interval. It naturally means the increased cooling load, and can be handled by altering the cooling capacity. For the present case, the airflow induced by the fan actually enhances the mixing of the hot and moist air intake from the opened door. This explains the observed overall temperature distribution for this case. The velocity distribution at the doorway and in the same sample $z - x$ plane is also shown in figure 6. The overall distribution is much different compared to Case 1. This corroborates with the effect of mixing due to the presence of the fan. The fan induced mixed-air flows around the product block also influences the heat transfer through the solid surfaces differently compared to Case 1. Note that, we observed a net mass flow rate out for both Case 1 and Case 2 through the door at $t = 120s$. However, it is found to be $\approx 38\%$ less mass flow rate out for Case 2 with the cooler fan-on mode compared to Case 1 with fan-off mode.

Figure 7 shows the different contours associated with fan alone when the door is closed (Case 3). The air flow pattern is evidently different compared to the other cases with open door. The cold air circulation from the cooler fan is meant to enhance the freezing of the product load. Figure 7 shows a clear trace of air flow from the fan that contributes to convective hypothermia.

4.2 Room with sliding door

In this section, we present the results for a scaled down cold storage room (dimensions are given in table 3) with sliding door of size $6cm \times 12cm$. Figure 8 shows the setup of the computational domain. Here we consider only air as a working fluid. An overset mesh region is built around the sliding door associated with a suitable background mesh. The sliding door is opened and closed with a rigid body motion via a suitable field function of the StarCCM+ program. We solve unsteady laminar Navier-Stokes system of equations for this preliminary setup. For these test ca-

Table 3. Dimensions for sliding door setup.

Geometry	H (m)	L (m)	B (m)
Cold room	0.18	0.19	0.12
Ice block	0.14	0.13	0.05
Cooler fan	0.02	0.01	0.05

ses, the cold ice block is kept at a constant temperature ($T = 255K$) and the energy equation within the ice block is not solved. The fan is modeled in such a way that, it supplies $0.002kg/s$ of air at a temperature of $T = 250K$. The outside temperature is prescribed as $T = 300K$. The door is opened during the first $0.8s$ and closed during the next $0.8s$. We present the results at a 55% open door situation during the opening phase and closing phase of the sliding door. The results are compared for cases with fan-off and fan-on modes. Figure 9 shows the wall normal velocity contours at 55% open door situations. We found a net mass flow rate out when the fan is off. The net mass flow rates out of the room are almost identical during the opening and closing stages for the fan-off mode. On the other hand, we observed that, the net mass flow rates are different at those stages when the fan is on. The fan induced fluid-flow inside of the room influences the air exchange through the door. During the opening phase, we noticed a net mass flow rate out, while during the closing phase we observed a net mass flow rate in. The net mass flow rate in is about 4.8 times more than the mass flow rate out. Also note that, the net mass flow rate in for the case with the fan-on, is about 42% less than the net mass flow rate out with fan-off situation. The temperature distributions at a representative plane perpendicular to the door are illustrated in figure 10 and figure 11. These contours corroborate with the above mentioned air exchange behavior.

5 Conclusions

We carried out a numerical study of hot, moist air exchange through a sliding door in a cold storage room. It has been observed that, the energy transport and the cooling of the product load is affected by the fan and the doorway. The air exchange and air mixing patterns differ noticeably in the presence of a fan. The increased demand of the cooling due to the hot and moist air exchange is being noticed. Additionally, a preliminary analysis with transient sliding door setup for a scaled room with one cycle of opening and closing is presented. Without the fan, the air exchange appeared almost identical during the opening or closing phase of the sliding door at 55% opening. On the other hand, we found a lower air exchange rate with fan-on mode. In future, a detailed study can be opted, taking into account the more realistic condensation of the moist air intake within the framework of a full scale cold storage room with a transient sliding door.

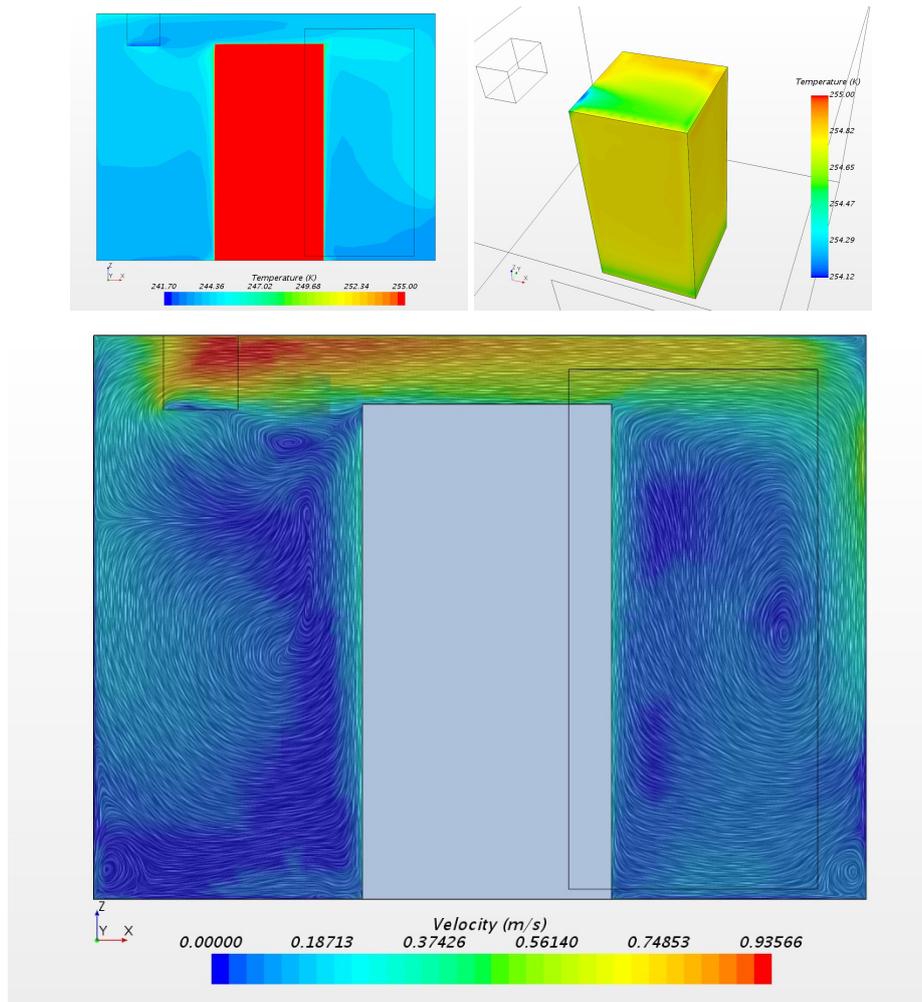


Figure 7. Case 3: contours of T (top left), the T contours (top right) at the product surfaces and LIC of \vec{v} (bottom).

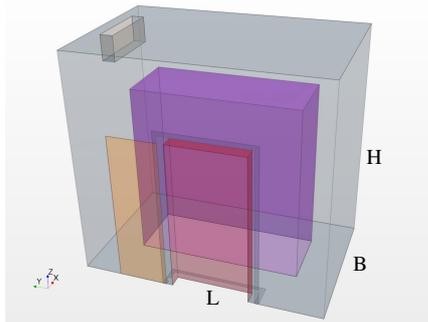


Figure 8. Computational domain of the scaled storage room with sliding door.

6 Acknowledgement

Authors greatly acknowledge the commercial computer program StarCCM+, used in this study. The results are obtained by the licensed version 12.06.010 of StarCCM+ on a stand alone desktop at OsloMet — Oslo Metropolitan University.

References

- Serap Akdemir and Thomas Bartzanas. Numerical modelling and experimental validation of a cold store. *Tarım Bilimleri Dergisi*, 21(4):606–619, 2015.
- Rui Carneiro, PD Gaspar, and PD Silva. 3d and transient numerical modelling of door opening and closing processes and its influence on thermal performance of cold rooms. *Applied Thermal Engineering*, 113:585–600, 2017.
- Le Chang, Xu Zhang, Shengji Wang, and Jun Gao. Control room contaminant inleakage produced by door opening and closing: Dynamic simulation and experiments. *Building and Environment*, 98:11–20, 2016.
- J-I Choi and Jack R Edwards. Large-eddy simulation of human-induced contaminant transport in room compartments. *Indoor air*, 22(1):77–87, 2012.
- S Duret, H-M Hoang, Denis Flick, and O Laguerre. Experimental characterization of airflow, heat and mass transfer in a cold room filled with food products. *International journal of refrigeration*, 46:17–25, 2014.
- AM Foster, MJ Swain, R Barrett, and SJ James. Experimental verification of analytical and cfd predictions of infiltration through cold store entrances. *International Journal of Refrigeration*, 26(8):918–925, 2003.
- Hong-Minh Hoang, Steven Duret, Denis Flick, and Onrawee Laguerre. Preliminary study of airflow and heat transfer in a cold room filled with apple pallets: Comparison between two modelling approaches and experimental results. *Applied Thermal Engineering*, 76:367–381, 2015.
- My Lan Hoang, Pieter Verboven, Josse De Baerdemaeker, and BM Nicolai. Analysis of the air flow in a cold store by means of computational fluid dynamics. *International Journal of Refrigeration*, 23(2):127–140, 2000.
- O Laguerre, S Duret, HM Hoang, Laurent Guillier, and Denis Flick. Simplified heat transfer modeling in a cold room filled with food products. *Journal of Food Engineering*, 149:78–86, 2015.
- HB Nahor, ML Hoang, Pieter Verboven, Martine Baelmans, and BM Nicolai. Cfd model of the airflow, heat and mass transfer in cool stores. *International Journal of Refrigeration*, 28(3):368–380, 2005.
- Ilias G Papakonstantis, Elizabeth Abigail Hathway, and Wernher Brevis. An experimental study of the flow induced by the motion of a hinged door separating two rooms. *Building and Environment*, 2017.
- Holger Scaar, Ulrike Praeger, Klaus Gottschalk, Reiner Jerdemann, and Martin Geyer. Experimentelle und numerische analyse der luftströmung in obst-und gemüselagern. *LANDTECHNIK–Agricultural Engineering*, 72(1), 2017.
- Julian W Tang, Andre Nicolle, Jovan Pantelic, Christian A Klettner, Ruikun Su, Petri Kalliomaki, Pekka Saarinen, Hannu Koskela, Kari Reijula, Panu Mustakallio, et al. Different types of door-opening motions as contributing factors to containment failures in hospital isolation rooms. *PloS one*, 8(6):e66663, 2013.
- JW Tang, I Eames, Y Li, YA Taha, P Wilson, G Bellingan, KN Ward, and J Breuer. Door-opening motion can potentially lead to a transient breakdown in negative-pressure isolation conditions: the importance of vorticity and buoyancy airflows. *Journal of Hospital Infection*, 61(4):283–286, 2005.
- Henk Kaarle Versteeg and Weeratunge Malalasekera. *An introduction to computational fluid dynamics: the finite volume method*. Pearson Education, 2007.
- Limin Wang, Lin Zhang, and Guoping Lian. A cfd simulation of 3d air flow and temperature variation in refrigeration cabinet. *Procedia engineering*, 102:1599–1611, 2015.
- DJ Wilson and DE Kiel. Gravity driven counterflow through an open door in a sealed room. *Building and Environment*, 25(4):379–388, 1990.
- Xiang Zhang, Jia-Wei Han, Jian-Ping Qian, Yi-Zhong Wang, Lin Wang, and Xin-Ting Yang. Computational fluid dynamic study of thermal effects of open doors of refrigerated vehicles. *Journal of Food Process Engineering*, 2017.

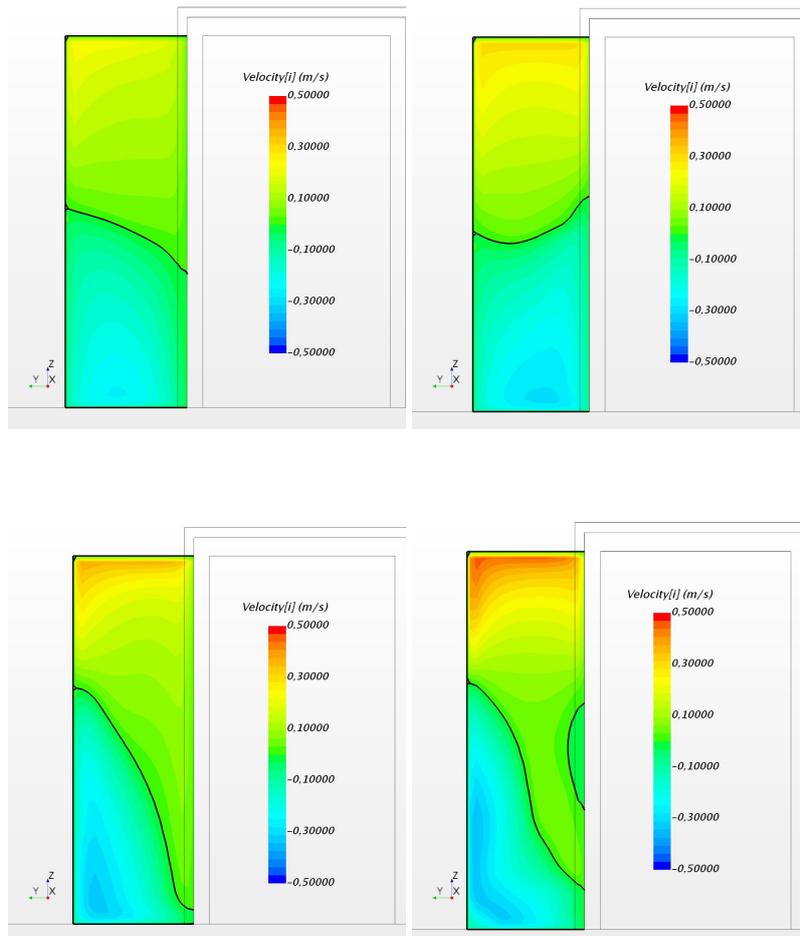


Figure 9. Door normal velocity contours at $t = 0.44\text{s}$ (left column) and at $t = 1.16\text{s}$ (right column). Top row: with fan-off mode, bottom row: with fan-on mode. Black curves represent zero velocity.

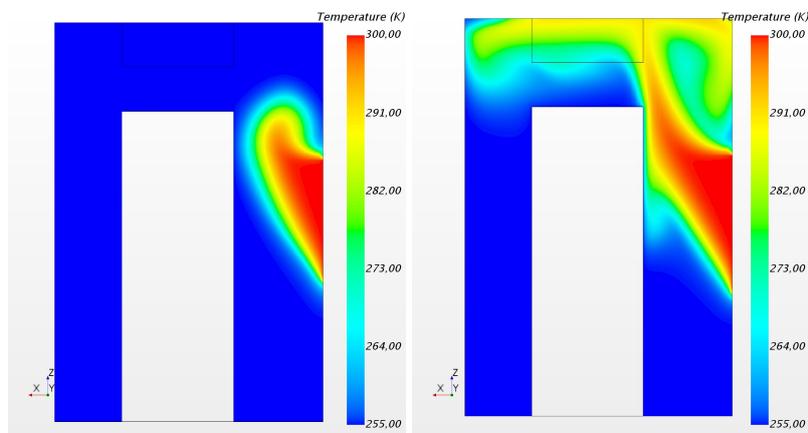


Figure 10. Temperature contours in a representative plane perpendicular to the door at $t = 0.44\text{s}$ (left) and at $t = 1.16\text{s}$ (right) with fan-off mode.

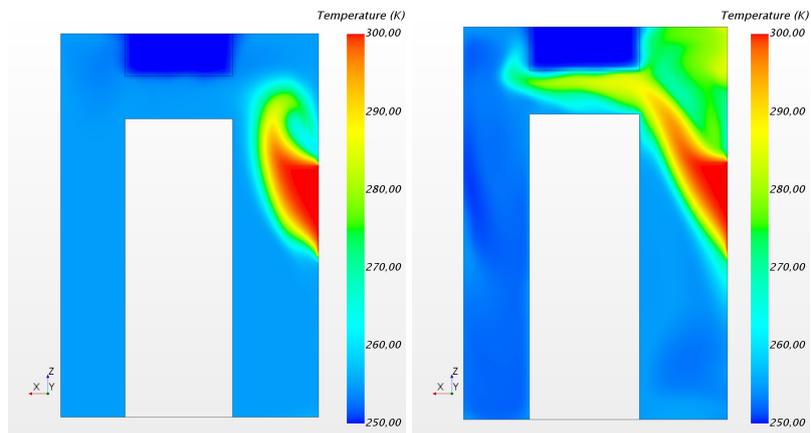


Figure 11. Temperature contours in a representative plane perpendicular to the door at $t = 0.44\text{s}$ (left) and at $t = 1.16\text{s}$ (right) with fan-on mode.

Developing Software Tools Facilitating Interdisciplinary Modelling

Sigve Karolius¹ Heinz Preisig¹

¹Chemical Engineering, Norwegian University of Science and Technology, Norway, sigve.karolius@ntnu.no, Heinz.Preisig@chemeng.ntnu.no

Abstract

Computer-aided modelling has focused on developing domain-specific frameworks. Despite being powerful stand-alone tools they can be challenging to incorporate into a multi-scale model whose inherent interdisciplinary nature leads to a heterogeneous set of languages and tools. However, tools and models can be made easier to adopt into future projects by making conscientious development choices based on software development techniques. This paper shares the experiences and software design options that were considered and employed in the development of the MoDeNa multi-scale modelling framework.

Keywords: multi-scale simulations, software design

1 Introduction

The focus on interdisciplinary collaboration is likely to increase within the computer-aided modelling community as the demand for simulation-guided design and discovery requires increasingly accurate predictive models. The trend is already clear within material-science where both the European Materials Modelling Council (EMMC) and National Aeronautics and Space Administration (NASA) have formulated long-term strategies aimed at developing solutions facilitating software interoperability for domain-specific simulation platforms, as can be seen in the technical report (Liu et al., 2018).

This is obviously not the first time that the need for standardisation has been realised and acted upon. One example from chemical engineering is the CAPE-OPEN project presented in (Belaud and Pons, 2002). The goal of the project was to create a common interface for process models to facilitate interoperability between well entrenched proprietary simulation environments, such as the Advanced System for Process Engineering (ASPEN) outlined in (Evans et al., 1979), as well as developing third-party extensions. Another example is the Modelica project, described in (Erik Mattsson et al., 1998), which aimed to provide a generic language for developing physics-based models in any domain.

The challenge in the integrated materials computational engineering (ICME) domain is that the models requires intertwining descriptions of systems on multiple scales and are therefore based on different physical principles. The project Morphology Development of Micro- and Nano-structures (MoDeNa) was one effort aimed at creating a platform coupling models together, but used simpli-

fied versions in-place of the detailed models. The project was a major software-development effort that involved implementing the software framework in collaboration with model-developers. This paper describes software design principles that were used in order to achieve coupling between a large number of models presented in (Karolius et al., 2017), as well as extensions that would have to be made in order for the framework to also support developing and simulating fully coupled models.

2 Constituent Parts and Coupling

All modelling activities involve describing systems and the relationship between them. In the multi-scale simulation domain the focus is mainly on coupling models that describe physical systems on vastly different scales as illustrated in Figure 1.

In order to develop tools that support the development of multi-scale systems it is necessary to identify the relationships between the elements that make up the overall multi-scale model as well as the type of coupling that must be supported. The project Morphology Development of Micro- and Nano-structures (MoDeNa) took aim at one area of multi-scale simulation, namely sequentially coupled models.

2.1 Ontology

The *ontology* provides a formal description of the constituent parts and the relationship between them. This is currently one of the major efforts of the materials modelling experts in the EMMC, but the topic is still relevant for the software design.

The work by (Yang and Marquardt, 2009) provides an extensive abstract ontology for multi-scale models from an engineering perspective. It could be of particular interest to study the overlap between ontologies from different domains and attempt to create a correlation between them for the purpose of facilitating communication between ontologies.

An alternative approach to defining an ontology and demanding developers to adhere to it is to develop a graphical modelling environment, such as (Elve and Preisig, 2017), that strictly adheres to a pre-defined ontology. Consequently, the model developer has no choice but adhering to the ontology.

Considering that the relationships between the entities in the ontology is directly related to scale-bridging in a multi-scale model the approach could be of great aid in de-

signing a software framework capable of handling a wide variety of scale-coupling strategies.

2.2 Scale-bridging Strategies

There are several approaches for coupling the scale-specific simulation models in a multi-scale model, both rigorous theoretical frameworks and practical approaches can be found in recent literature such as (Weinan, 2011).

2.2.1 Fully Coupled Models

The overall multi-scale model can, seen from the top-down perspective, be described as an onion. At the engineering-level one is therefore always making implicit assumptions about the scales below. Bearing in mind that

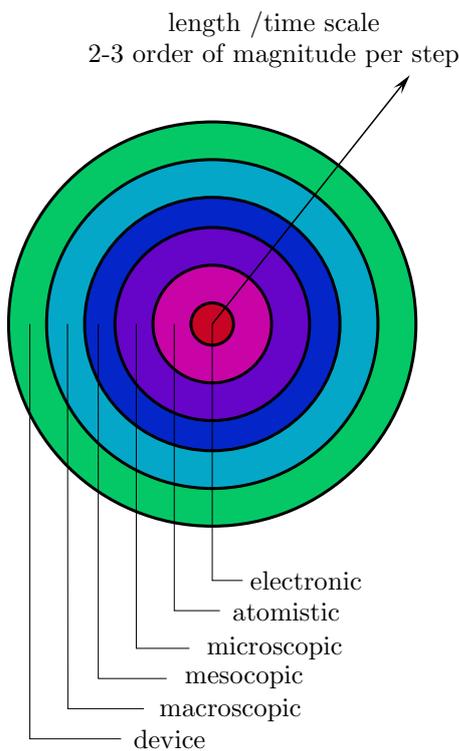


Figure 1. Illustration of a multi-scale model as the layers of an onion, there each layer represents a modelling domain.

continuum-based models have been successfully used for decades it could be tempting to question the need for the scale-coupling illustrated in Figure 2. However, the scale-



Figure 2. Illustration of a model in which the dynamics of the scale-specific models overlap. The multi-scale description consequently requires that the models are fully coupled.

coupling is often important, such as when considering with surface effects in low-density systems, such as rarefied gases (Docherty et al., 2014). Moreover, when the coupling does occur it quickly becomes a dominating factor that characterises the dynamics of the system.

Fully coupling models is therefore necessary in order to capture physical phenomena that arises as a direct consequence of the multi-scale nature of the system. However, the computational cost associated with running concurrent simulations limits the number of models that can be integrated in this manner before the computational effort outweighs the benefit.

2.2.2 Scale-Separated Models

The scale-separation assumption is an argument that decouples the dynamics of the individual scale-specific models as illustrated in Figure 4. This is the assumption on which the MoDeNa software framework was based. The use of surrogate-models in the MoDeNa project is shown in Figure 3, they are used in-place of the detailed model for the lower scale. In this way the scale-separation is

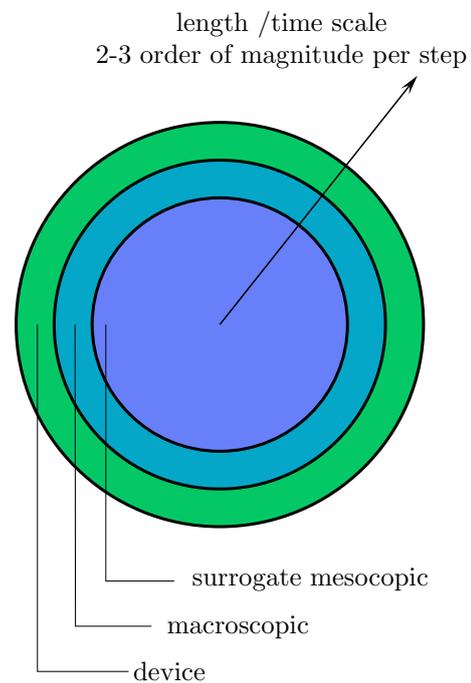


Figure 3. Illustration of a multi-scale model as the layers of an onion, there each layer represents a modelling domain.

exploited, typically for the purpose of speeding up the higher scale model. As mentioned in the previous section, scale separation is very common and the use of low-level models to calculate material properties for exotic alloys is highly interesting for engineers. In contrast to the fully

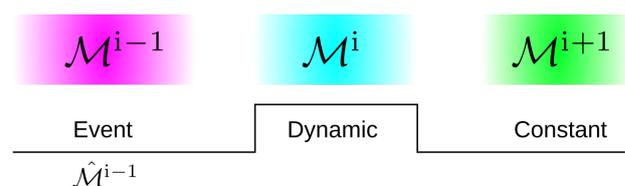


Figure 4. Illustration of the relationship between the dynamics of a model on a specific scale to that of the scale below and above.

coupled models in the previous section one can use a lot

of surrogate-based scale-separated models as long as they are never used outside of the domain in which the parameters were validated.

3 Software Design Concepts

The process of developing generic software is intrinsically iterative and different modelling paradigms often have significantly different requirements. However, it can be considerably simplified by making conscious decisions that clearly separate individual components of the software and define to what extent it supports generic concepts.

3.1 Abstraction

For the past two decades so-called *object-oriented*, or data-centric, design has been the prevailing principle for software development. The earliest formal representation of an object-oriented paradigm is often accredited to the SIMULA languages, whose history is presented in (Nygaard and Dahl, 1981), which was designed to facilitate development of discrete event systems (DESs).

Nowadays, most modern programming languages support object-oriented (OO) programming to a greater or lesser extent. It is common for modelling frameworks to take advantage of the abstraction capabilities that object-orientation provides, which in practice means to design well-defined generic implementations that are capable of describing special cases of an appropriate type. The advantage often lies in that multiple solution strategies can be defined for the generic description and any of them can be imposed on a particular problem.

The Open Field Operation and Manipulation (OpenFOAM) project founded by (Weller et al., 1998) is an excellent example of the power of OO design principles. At its core the OpenFOAM framework is a generic implementation of finite volume method (FVM) and provides abstract data-types designed specifically for the purpose of making the computer implementation of partial differential equations (PDEs) mimic mathematical notation. The listing 1 shows an excerpt from the source code for the OpenFOAM solver "laplacianFoam", an algorithm for solving the heat equation.

Listing 1. A slightly modified excerpt from the source code of the `laplacianFoam` solver in the OpenFOAM project. The code block is the entire functional part of the main loop of the solver and illustrates how well-designed abstraction of operators and data-containers simplifies the implementation.

```
while ( simple.loop() ) {
    while ( simple.correctNonOrthogonal() ) {
        solve(
            fvm::ddt(T) - fvm::laplacian(DT, T)
        );
    }
}
```

When comparing the code-block with the colour coding in

equation 1 the resemblance is striking.

$$\frac{\partial T}{\partial t} - \nabla [D \nabla T] = 0 \quad (1)$$

However, even though the abstraction of the differential operators makes the overall implementation look deceptively simple, there is obviously a lot of software engineering taking place in the background.

This raises the question of what happens to the terse implementation if the solver should be modified to include temperature dependency of the thermal diffusivity $D(T)$ by employing a simulation-model from a lower scale, e.g. using molecular dynamics (MD) and the Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS) described in (Plimpton, 1995).

There is effectively two possible choices at this point:

Keep the original abstraction

In the case of OpenFOAM it is possible to implement a specialised thermo-physical model which handles the coupling with the lower-scale model; thus, the original solver remains largely unchanged.

Modify the solver

The easiest strategy is to implement the coupling to the MoDeNa framework directly in the main loop of the solver. This will render the original implementation virtually unrecognisable, but also clearly show the coupling.

There are positive and negative aspects to both strategies. As an example consider the sample code for implementing the MoDeNa software framework directly into the `laplacianFoam` solver.

Listing 2. Excerpt from the source code of the `modenaLaplacianFoam` solver. The main loop effectively split into two parts where one is a loop that requests information about the thermal diffusivity from the MoDeNa software framework before continuing to the solver equivalent to the original `laplacianFoam` implementation in Listing 1

```
while ( simple.loop() ) {
    while ( simple.correctNonOrthogonal() ) {
        try { // ----- MoDeNa Block
            forAll(T, celli) {
                DT[celli] = model(Tpos, T[celli]);
            }
        } catch(const modenaException& e) {
            return e.errorCode();
        } // ----- Call Solver
        solve(
            fvm::ddt(T) - fvm::laplacian(DT, T)
        );
    }
}
```

When compared to the original solver in Listing 1 the modified implementation in Listing 2 has major changes. However, the changes are fairly straight forward to implement for someone who is somewhat familiar with the programming language. Moreover, since the implementation

interrupts the main loop it would be possible to perform synchronization with an external application; thus, potentially allowing for concurrent computational fluid dynamics (CFD) and MD simulations that exchange data on-the-fly.

3.2 Modularity

In the overall picture of a simulation that employs multiple models it is intuitive to consider each individual simulation-model as a software module. The module is commonly further divided into components that perform independent tasks, such as pre/post processing or providing abstraction, which is the case for the operators in the example in Listing 1.

3.2.1 Coupling

In general the term *coupling* refers to the relationship between software modules in a project. The idea motivating the focus on coupling in software design is that internal changes within one module should affect other modules to the least extent possible.

Loose coupling

Loose coupling refers to a situation where there is little interaction between the modules that makes up the software. Loosely coupled models can for instance be running a simulation in order to generate initial conditions for a subsequent simulation.

Tight coupling

Modules that are tightly coupled are highly intertwined and the connection may require in-depth knowledge of the internal components of a different module. Simulation models that requires run-time exchange of information, maybe even synchronised execution, are tightly coupled.

For the purpose of maintainability it is desirable to strive towards achieving loose coupling, illustrated in Figure 5, between modules. Loose coupling simplifies both de-

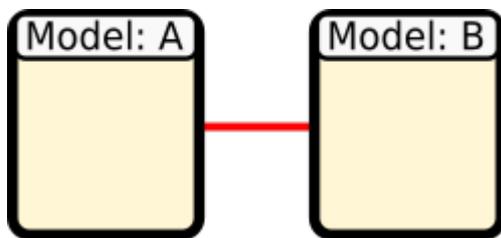


Figure 5. Illustration of two loosely coupled models where the interface effectively decouples the inner-workings of each model. This makes it possible for one model to make changes to internal components as-necessary without requiring the other model to change.

velopment and simulation of models compared to tightly coupled models illustrated in 6. In the case of interdisciplinary modelling efforts tight coupling is particularly problematic. The reason for this is that it is important that

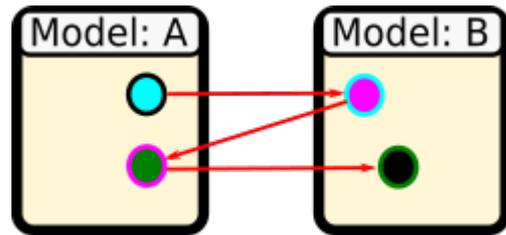


Figure 6. Illustration of tight coupling between two models where both models need some information about individual components of the other. Consequently, internal changes in one model may cause the other model to adapt accordingly.

models can be tested and validated in relative isolation, i.e. without connections to other models.

However, the question still remains with how to deal with models that are not loosely coupled by nature and requires exchange of information at run-time. The strategy here is to employ an external framework, such as the MoDeNa software framework presented in (Karolius et al., 2016), in order to decouple the connection between the individual models as shown in Figure 7.

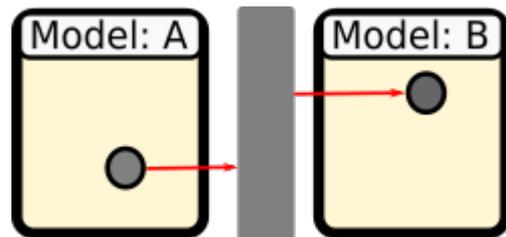


Figure 7. Illustration of a situation where two models that are inherently coupled, but the use of an external framework is used for the purpose of decoupling the connection.

3.2.2 Cohesion

Cohesion is an intra-module concept and refers to the relationship between the internal components of a module. It is a measure of the degree to which the internal components of a module focuses on a single task.

It is common to refer to the degree of cohesion within a module as *low* or *high*.

Low cohesion

Low cohesion effectively implies that the implementation of the software module, or simulation model, is disorganised. In a module that suffers from low cohesion each module may try to perform several unrelated tasks and still require importing functionality from other components in the module.

High cohesion

Models that have high cohesion are organised into components that focus on specific tasks with little or no interaction with the other components in the module.

Since cohesion is an intra-module concept the practical importance of cohesion becomes less relevant if the coupling between modules is loose. However, as comparing the illustrations of modules with high and low cohesion, in Figure 8 and respectively, it can be argued that the former will lead to fewer sources for errors and less maintenance. In contrast to coupling there is not much that an external

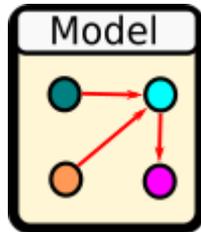


Figure 8. Illustration of a module with high cohesion.

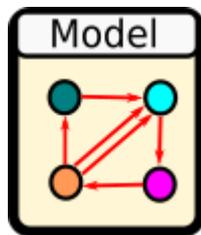


Figure 9. Module whose functionality is implemented in multiple components that require cross- and circular-imports in the module.

framework can do in order to enforce cohesion within a simulation model. However, it is possible to promote cohesion by making it possible to pre-define a workflow that should be executed when running the application.

3.3 Workflows

Most scientific simulations follow a pre-defined workflow: *initialisation*, *execution* and *post-processing*. However, for the purpose of exploring a large design space for simulation-guided studies it necessary to automate the execution and tracking of a large number of simulation runs.

In its simplest form a computational workflow consists of *atomic*, i.e. non-interruptable, tasks. Software tools, such as FireWorks (Jain et al., 2015) and are treats the workflow as a directed acyclic graph (DAG) is designed to execute whatever it can until there are no more tasks left in the queue.

3.3.1 Adaptivity and Distributed Tasks

In order to make workflows that can be employed in a machine-learning concept it is necessary to incorporate some adaptivity into the execution of the computational workflow. In practice this idea is rather simple: make the simulation-models capable of telling the workflow generator what to do in a particular context.

The MoDeNa software framework used the features of FireWorks to implement an adaptive workflow that would

perform run-time parameter estimation and validation of using model-based design of experiments (Franceschini and Macchietto, 2008).

Some simulations are simply too complex to be executed in a normal desktop environment and requires the workflow to migrate to facilities capable of running applications that require heavy computing.

3.3.2 Concurrency

As long as simulation tasks are *atomic* concurrency is not a big issue and parallel execution is easily facilitated using DAG s. However, when simulations have to exchange information while they are running there is a problem, namely how to ensure that the processes are synchronised.

One approach could be to employ the message passing interface (MPI) toolkit, which is a standardised way of handling inter-process communication. However, with this approach one could end up attempting to run multiple MPI processes inside the main MPI process. An alternative way of modelling concurrency is to use Petri-nets (Peterson, 1977), as illustrated in Figure 10 A basic model

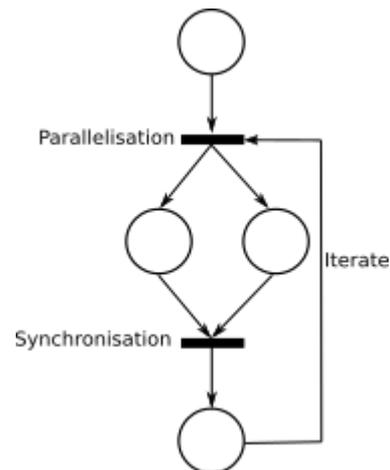


Figure 10. Simple illustration of a naive concurrency model for two models that runs in parallel and synchronises between every iteration allowing for information to be exchanged between the models.

for concurrency would not require implementing complex MPI-style communication mechanisms.

4 Conclusion

Designing a generic software framework that facilitates development of multi-scale models requires making pragmatic choices in the software design. The main focus of the framework is to support the modelling and simulation effort, not necessarily providing framework that uses abstraction techniques to hide the implementation from the users.

Instead, the design should aim at enforcing modularity of the scale-specific models and enabling flexible and distributed computational workflows. There are also significant lessons to be learned from

the use of *semantic interoperability* within web-development as well as the *gene ontology* project.

Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° [604271].

References

- Jean-Pierre Belaud and Michel Pons. Open software architecture for process simulation: The current status of cape-open standard. In Johan Grievink and Jan van Schijndel, editors, *European Symposium on Computer Aided Process Engineering-12*, volume 10 of *Computer Aided Chemical Engineering*, pages 847 – 852. Elsevier, 2002. doi:[https://doi.org/10.1016/S1570-7946\(02\)80169-9](https://doi.org/10.1016/S1570-7946(02)80169-9). URL <http://www.sciencedirect.com/science/article/pii/S1570794602801699>.
- Stephanie Y. Docherty, Matthew K. Borg, Duncan A. Lockerby, and Jason M. Reese. Multiscale simulation of heat transfer in a rarefied gas. *International Journal of Heat and Fluid Flow*, 50:114 – 125, 2014. ISSN 0142-727X. doi:<http://dx.doi.org/10.1016/j.ijheatfluidflow.2014.06.003>. URL <http://www.sciencedirect.com/science/article/pii/S0142727X14000836>.
- Arne Tobias Elve and Heinz A. Preisig. From ontology to executable program code. In Antonio Espuña, Moisés Graells, and Luis Puigjaner, editors, *27th European Symposium on Computer Aided Process Engineering*, volume 40 of *Computer Aided Chemical Engineering*, pages 2317 – 2322. Elsevier, 2017. doi:<https://doi.org/10.1016/B978-0-444-63965-3.50388-3>. URL <http://www.sciencedirect.com/science/article/pii/B9780444639653503883>.
- Sven Erik Mattsson, Hilding Elmqvist, and Dynasim Ab. Mod-elica - an international effort to design the next generation modeling language. 30, 01 1998.
- L.B. Evans, J.F. Boston, H.I. Britt, P.W. Gallier, P.K. Gupta, B. Joseph, V. Mahalec, E. Ng, W.D. Seider, and H. Yagi. Aspen: An advanced system for process engineering. *Computers & Chemical Engineering*, 3(1):319 – 327, 1979. ISSN 0098-1354. doi:[https://doi.org/10.1016/0098-1354\(79\)80053-8](https://doi.org/10.1016/0098-1354(79)80053-8). URL <http://www.sciencedirect.com/science/article/pii/0098135479800538>.
- Gaia Franceschini and Sandro Macchietto. Model-based design of experiments for parameter precision: State of the art. *Chemical Engineering Science*, 63(19):4846 – 4872, 2008. ISSN 0009-2509. doi:<http://dx.doi.org/10.1016/j.ces.2007.11.034>. URL <http://www.sciencedirect.com/science/article/pii/S0009250907008871>. Model-Based Experimental Analysis.
- Anubhav Jain, Shyue Ping Ong, Wei Chen, Bharat Medasani, Xiaohui Qu, Michael Kocher, Miriam Brafman, Guido Petretto, Gian-Marco Rignanese, Geoffrey Hautier, Daniel Gunter, and Kristin A. Persson. Fireworks: a dynamic workflow system designed for high-throughput applications. *Concurrency and Computation: Practice and Experience*, 27(17):5037–5059, 2015. ISSN 1532-0634. doi:[10.1002/cpe.3505](https://doi.org/10.1002/cpe.3505). URL <http://dx.doi.org/10.1002/cpe.3505>. CPE-14-0307.R2.
- Sigve Karoliuss, Heinz A. Preisig, and Henrik Rusche. Multi-scale modelling software framework facilitating simulation of interconnected scales using surrogate-models. In Zdravko Kravanja and Miloš Bogataj, editors, *26th European Symposium on Computer Aided Process Engineering*, volume 38 of *Computer Aided Chemical Engineering*, pages 463 – 468. Elsevier, 2016. doi:<http://dx.doi.org/10.1016/B978-0-444-63428-3.50082-5>.
- Sigve Karoliuss, Heinz A. Preisig, and Henrik Rusche. Sequential multi-scale modelling concepts applied to the polyurethane foaming process. In Antonio Espuña, Moisés Graells, and Luis Puigjaner, editors, *27th European Symposium on Computer Aided Process Engineering*, volume 40 of *Computer Aided Chemical Engineering*, pages 487 – 492. Elsevier, 2017. doi:<https://doi.org/10.1016/B978-0-444-63965-3.50083-0>. URL <http://www.sciencedirect.com/science/article/pii/B9780444639653500830>.
- Xuan Liu, David Furrer, Jared Kusters, and Jack Holmes. Vision 2040: A roadmap for integrated, multiscale modeling and simulation of materials and systems. Technical Report 20180002010, National Aeronautics and Space Administration (NASA), 03 2018.
- Kristen Nygaard and Ole-Johan Dahl. The development of the simula languages. In Richard L. Wexelblat, editor, *History of Programming Languages I*, pages 439–480. ACM, New York, NY, USA, 1981. ISBN 0-12-745040-8. doi:[10.1145/800025.1198392](https://doi.org/10.1145/800025.1198392). URL <http://doi.acm.org/10.1145/800025.1198392>.
- James L. Peterson. Petri nets. *ACM Computing Surveys*, 9:223–252, 1977.
- Steve Plimpton. Fast parallel algorithms for short-range molecular dynamics. *Journal of Computational Physics*, 117(1):1–19, 1995. ISSN 0021-9991. doi:<https://doi.org/10.1006/jcph.1995.1039>. URL <http://www.sciencedirect.com/science/article/pii/S002199918571039X>.
- E. Weinan. *Principles of Multiscale Modeling*. Cambridge University Press, 1 edition, 2011. ISBN 978-1-1070-9654-7.
- H. G. Weller, G. Tabor, H. Jasak, and C. Fureby. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Comput. Phys.*, 12(6):620–631, November 1998. ISSN 0894-1866. doi:[10.1063/1.168744](https://doi.org/10.1063/1.168744). URL <http://dx.doi.org/10.1063/1.168744>.
- Aidong Yang and Wolfgang Marquardt. An ontological conceptualization of multiscale models. *Computers & Chemical Engineering*, 33(4):822 – 837, 2009. ISSN 0098-1354. doi:<https://doi.org/10.1016/j.compchemeng.2008.11.015>. URL <http://www.sciencedirect.com/science/article/pii/S0098135408002524>.

Linearization for Analysis of a Hydropower Model using Python API for OpenModelica

Liubomyr Vytvytskyi Bernt Lie

Department of Electrical engineering, Information Technology and Cybernetics, University of South-Eastern Norway, Porsgrunn, Norway, {Liubomyr.Vytvytskyi,Bernt.Lie}@usn.no

Abstract

Even though almost all processes in the real world are described by nonlinear models, nonlinear theory for analysis of these models is far less developed than the theory for linear models. Therefore model linearization is important in order to make efficient analysis tools for these models.

This paper describes the possibility of automatic linearization in Python for a hydropower system modeled in OpenModelica using our in-house hydropower Modelica library *OpenHPL*. Linearization is made using a Python API. Simple uses of the linearized model for analysis and synthesis are indicated.

Keywords: linearization, hydropower, Python API, OpenModelica

1 Introduction

1.1 Background

A transition towards more renewable energy sources is currently taking place in Europe and all over the world. This situation leads to increase in the use of flexible hydropower plants to compensate the highly changing production from intermittent energy sources such as wind and solar irradiation. A high head hydropower system is considered for this study, because it takes up the main part of all hydropower plants in Norway.

The possibility for modelling and simulating this hydropower system takes an important role in order to make efficient analysis tools for testing a designed controller for stability and performance in different operating regimes. One of such analysis tools can provide automatic linearization; an approximate linear model allows for the use of linear theory for analysis and synthesis which is much better developed than nonlinear theory.

1.2 Previous Work

Basic mathematics and control theory needed to model, analyze, and design feedback systems are provided in (As-tröm and Murray, 2010). Based on these methods, the linearized hydropower model can be further tested and analyzed for control purposes using a Python¹ package — *python-control* (The Python Control Systems Library)².

python-control is a Python module, where basic operations for analysis and design of feedback control systems are implemented.

A Python API³ for OpenModelica⁴ already exists that provides possibilities for controlling simulations of the OpenModelica models via Python (Lie et al., 2016). Python in turn gives much wider possibilities for plotting, analysis, and optimization (e.g., using Python packages *matplotlib*, *numpy*, *scipy*, etc.).

Some work on modeling a waterway for the high head hydropower system together with a generator, the Francis turbine, and a governor, has already been done using OpenModelica (Vytvytskyi and Lie, 2017, 2018). Unit models have been assembled in our in-house Modelica⁵ library *OpenHPL*⁶.

1.3 Overview of Paper

In this paper, the main contribution is investigation of how modern computer tools can make the workflow of analysis and design, including linearization and linear control analysis/design. This is the first paper that demonstrates how linearization can be done using the Python API for “non-academic” hydropower models of different complexity. Model implementation is done in OpenModelica using the *OpenHPL* library.

The paper is structured as follows: Section 2 gives a system description of a high head hydropower system. Section 3 gives an overview of the modeling tools and a presentation of the hydropower model. Then automatic linearization and a simple PI controller design are described in Sections 4 and 5. Finally, discussion and conclusions are given in Section 6.

2 System Description

High head plants typically collect and store water in reservoirs in mountains, with tunnels leading the relatively small flow of water down a considerable height difference to the aggregated turbine and generator. The electricity, produced by the generator, is then transferred through power lines to consumers. A typical structure for the high

³<https://goo.gl/Qyjqq2>

⁴<https://openmodelica.org>

⁵<https://www.modelica.org>

⁶Open Hydro Power Library is developed by the first author within his PhD study.

¹<https://www.python.org>

²<https://goo.gl/MtbYtf>

head hydropower plant is depicted in Figure 1 (Vytvytskyi and Lie, 2017).

For simulations in this paper, the data from the Sundsbarm hydropower plant in Telemark, Norway is used with data provided in (Winkler et al., 2011), see Table 1 and 2.

Table 1. The waterway geometry of Sundsbarm hydropower plant.

Waterway unit	Height difference, m	Length, m	Diameter, m
Reservoir	48	—	—
Conduit	23	6600	5.8
Penstock	428.5	600	3
Surge tank	120	140	3.4
Discharge race	0.5	600	5.8
Tail water	5	—	—

Table 2. The turbine geometry of Sundsbarm hydropower plant.

Turbine type	Nominal head, m	Nominal flow rate, m ³ /s	Nominal power, MW
Francis	460	24.3	104.4

3 Modeling

3.1 Modeling Tools

All modeling is done in OpenModelica, which is an open-source Modelica-based modeling and simulation environment intended for industrial and academic usage⁷.

For modeling the hydropower system, library *OpenHPL* is used. This is an in-house hydropower library, where different parts of the waterway components, such as reservoir, conduit, surge tank and turbine, have been assembled. In this library, different waterway components of the hydropower system are described by both mass and momentum balance, and could include compressible/incompressible water or elastic/inelastic pipe walls. A better overview of the mathematical models and methods used in this library is given in (Vytvytskyi and Lie, 2017; Splavska et al., 2017).

In addition, our hydropower library can also be connected with other open source Modelica libraries such as *OpenIPSL*⁸ (Open-Instance Power System Library), where a much wider variety of power system components are presented. Together, the *OpenHPL* and *OpenIPSL* libraries give a possibility to develop a model for the whole hydropower system that starts from the water in the reservoir and ends with the different electrical loads. Linearization also works for more complex/detailed models than

⁷Some tutorials exist for Modelica — <http://book.xogeny.com>, and OpenModelica — <https://goo.gl/76274H>

⁸<http://openipsl.readthedocs.io/en/latest>

used here (e.g., a model for the whole hydropower system), but space limitations restrict our presentation to simpler cases.

3.2 Model Presentation

In this study, two cases of complexity for this system are considered:

1. Simplified system with incompressible water and inelastic pipe.
2. More complex system that includes water compressibility and pipe shell elasticity in the penstock.

Both these cases are straightforward to implement in OpenModelica using the *OpenHPL* library. A block diagram that is relevant for both cases of the hydropower system is presented in Figure 2. For simplicity, the water levels in reservoir and tail water are considered to be constant.

In both cases, the model has one input — turbine gate opening — u_{tr} , and one output — turbine volumetric flow rate — \dot{V}_{tr} .

4 Linearization

4.1 Overview

The Python API (Lie et al., 2016) provides a linearization function that allows approximation of nonlinear DAE models in OpenModelica to linear state space models in Python.

First, the Modelica model is instantiated in Python using the *OMPpython* package and the following command:

```
hps_s=ModelicaSystem("OpenHPL.mo", "
    OpenHPL.Tests.HPLiniazation", "
    Modelica") // for simpler model
hps_kp=ModelicaSystem("OpenHPL.mo", "
    OpenHPL.Tests.HPLiniazationKP", "
    Modelica") // for complex model
```

After this, the input signal and simulation options are set in Python for the simulation. Before linearization, the model parameters are set to steady state values. Automatic linearization is done from Python, where the matrices for the general state-space representation of a linear system are given using the following command:

```
As,Bs,Cs,Ds = hps_s.linearize() // for
    simpler model
Akp,Bkp,Ckp,Dkp = hps_kp.linearize() // for
    complex model
```

4.2 Simple Model

First, the simple hydropower model is linearized. Through linearization, the state-space matrices A , B , C and D are generated:

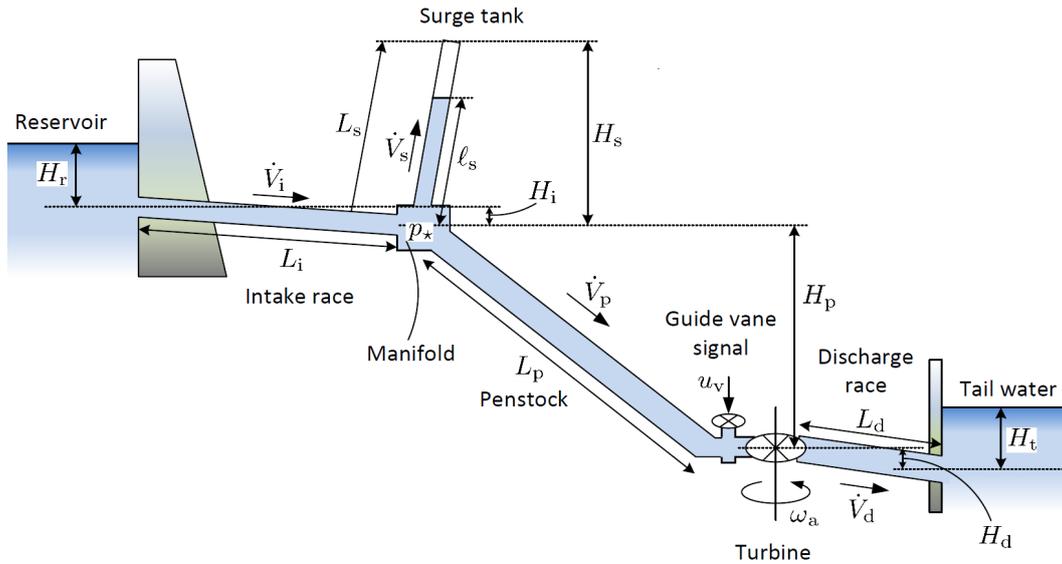


Figure 1. Structure of the high head hydropower plant.



Figure 2. Model of the hydropower system.

$$A = \begin{bmatrix} -4.4 & 3.4 \cdot 10^{-4} & 7.7 \cdot 10^{-6} \\ 4.2 & -4.0 \cdot 10^{-3} & -1.1 \cdot 10^{-5} \\ 0.0 & 997 & 0.0 \end{bmatrix} \quad (1)$$

$$B = \begin{bmatrix} 110.45 \\ -106.62 \\ 0.0 \end{bmatrix} \quad (2)$$

$$C = [1.0 \ 0.0 \ 0.0] \quad (3)$$

$$D = [0.0] \quad (4)$$

Information about the state, input and output variables and their order for the linearized simple model can be checked in Python using the following commands:

```
hps_s.getLinearInputs() // for inputs
hps_s.getLinearOutputs() // for outputs
hps_s.getLinearStates() // for states
```

As mentioned above, this model has one input $u = u_{tr}$ and one output $y = \dot{V}_{tr}$. The linearization algorithm gives

a state vector with 3 elements: $x = [\dot{V}_p, \dot{V}_s, m_s]^T$. Here, \dot{V}_p and \dot{V}_s are the volumetric flow rates in the penstock and surge tank respectively, m_s is the water mass in the surge tank. The linearization algorithm has actually considered two more states (the water masses in the reservoir and tail water). However, due to assumption of the constant water level in those compartments, their rows in the A matrix are zero vectors and can be neglected.

As seen, for this simpler case the linear model is of low order. It is also known that the system is asymptotically stable if all eigenvalues of the A matrix have negative real parts. Using the following command from the *numpy* package, we find the eigenvalues:

```
linalg.eig(As)
```

The eigenvalues of A matrix are as follows:

$$\text{eig}(A) = \begin{bmatrix} -4.367 \\ -0.003 + 0.06j \\ -0.003 - 0.06j \end{bmatrix} \quad (5)$$

4.3 Complex Model

Next, the more complex hydropower model has been linearized in the same way as was presented for the simpler case. The state-space matrices are presented below in simplified form due to their shape:

$$A = \begin{bmatrix} 4.17 & \dots & 0.0 \\ \vdots & \ddots & \vdots \\ 0.0 & \dots & 0.0 \end{bmatrix} \in \mathbb{R}^{22 \times 22} \quad (6)$$

$$B = \begin{bmatrix} 0.0 \\ \vdots \\ 0.0 \end{bmatrix} \in \mathbb{R}^{22 \times 1} \quad (7)$$

$$C = [0.0 \ \dots \ 0.0] \in \mathbb{R}^{1 \times 22} \quad (8)$$

$$D = [0.036] \quad (9)$$

The inputs, outputs and states for the linearized complex model are also provided using the following commands:

```
hps_kp.getLinearInputs() // for inputs
hps_kp.getLinearOutputs() // for outputs
hps_kp.getLinearStates() // for states
```

The input and output are the same as for the simpler case. However in this case, the model consists of 22 states that make it more space demanding, $x = [\dot{m}_{p,i}, p_{p,i}, \dot{V}_s, m_s]^T$. Here, two states are also relevant for the surge tank: the volumetric flow rate, \dot{V}_s , and the water mass, m_s . On the other hand, the penstock now is described by 20 equations — 10 for the mass flow rate — $\dot{m}_{p,i}$ and 10 for the pressure — $p_{p,i}$ (here, i is a cell number in range from 1 to n , where n is a number of discretization points of the penstock). This is due to using the Finite Volume method for the discretization of the more complex model with compressible water and elastic pipe walls (the penstock is divided in ten cells here).

In the same way as it was done for the previous simpler case, the eigenvalue analysis of A matrix could be performed. We found that this more complex system is also asymptotically stable.

4.4 Bode Plot Comparison

After the hydropower model has been linearized and the (A, B, C, D) matrices for the general state-space representation are defined for the two cases, some further analysis for the linearized system might be done. For control purposes, the frequency response of a system (Bode plot) can be interesting.

To plot this frequency response, the following commands from the *python-control* package in Python can be used:

```
sys = ss(A, B, C, D)
mag, phase, omega = bode_plot(sys, dB=True)
```

As an alternative, the transfer function $H(s)$ of the system could be found from:

$$H(s) = C(sI - A)^{-1}B + D \quad (10)$$

Here, s is the Laplace operator and for the frequency response, define $s = j\omega$, where ω is frequency in radians. After this, the Bode plot for the linearized hydropower system can be plotted. The Bode diagram for the two cases of the linearized hydropower model are shown in Figure 3.

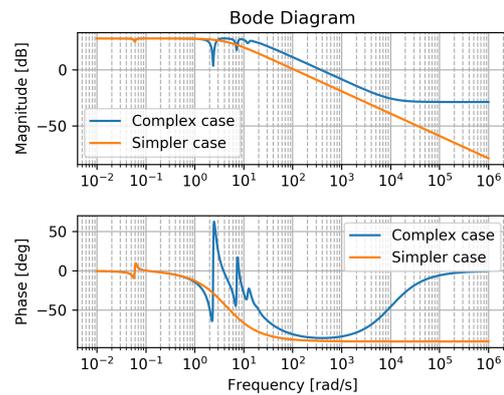


Figure 3. Comparison of Bode plot for two cases of the hydropower system.

5 Design of PI Controller

Using the *python-control* package in Python, a simple PI controller for the linearized hydropower models could be designed and tuned.

First, the step response of the control signal for the linearized hydropower model is found for the two cases, using the following command:

```
sys_s = ss(As, Bs, Cs, Ds) // simple system
youts, Ts = step(sys_s)
sys_kp = ss(Akp, Bkp, Ckp, Dkp) // complex
system
youtkp, Tkp = step(sys_kp)
```

The results of the control signal step response for both the simple and the complex linearized models are shown in Figure 4.

After this, a PI controller $C_r(s) = \frac{K_p s + K_i}{s}$ is tuned.

Then, the controller transfer function is defined in Python using the control package and connected to the hydropower system via feedback using the following commands:

```
CrPI = tf([[Kp, Ki]], [[1., 0.]])
Trs = feedback(sys_s*CrPI, 1)
youts, Ts = step(Trs)
Trkp = feedback(sys_kp*CrPI, 1)
youtkp, Tkp = step(Trkp)
```

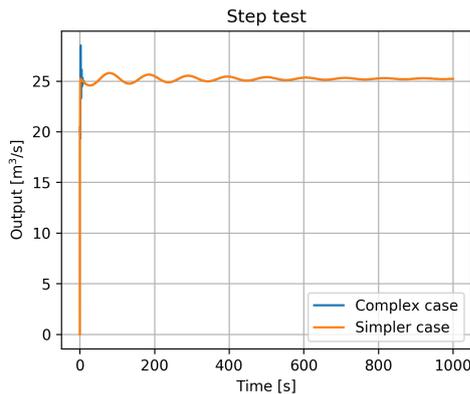


Figure 4. Comparison of the step response for the simple and complex linearized models.

The results of the step response for the reference value for the PI controller that control the hydropower system for the two cases are shown in Figure 5 and Figure 6. For the two cases, the step response is done for two sets of controller parameters.

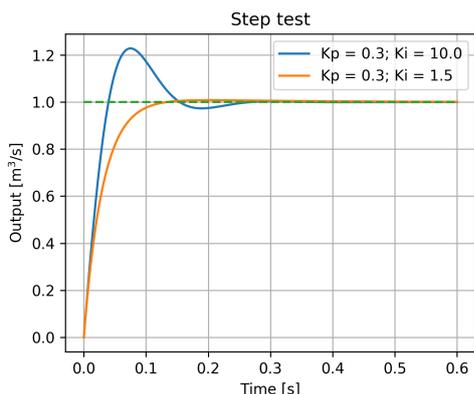


Figure 5. Step response for the simpler model with PI controller.

Finally, the designed and tuned PI control could be checked on the original (nonlinear) hydropower model in OpenModelica. The results of the step test for the output and input are shown in Figure 7 — for the simpler case and Figure 8 — for the more complex case.

6 Discussion and Conclusions

The possibility of automatic linearization of OpenModelica models through Python using the Python API has been presented in this paper.

Two cases with different model complexity for the hydropower system have been linearized in order to show the linearization capability of the Python API. Despite the model complexity, the linearization algorithm finds the state space matrices A , B , C , D .

After linearization, linear theory could be further used for the model analysis and synthesis. Examples of analysis has been presented by creating a Bode plot and designing

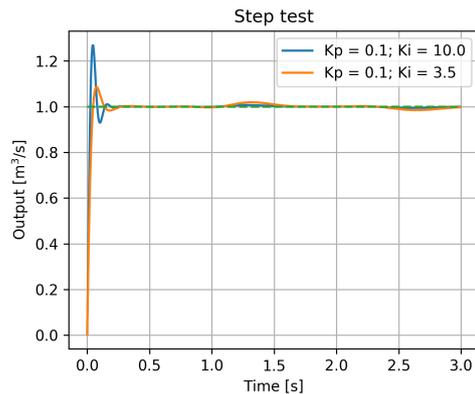


Figure 6. Step response for the complex model with PI controller.

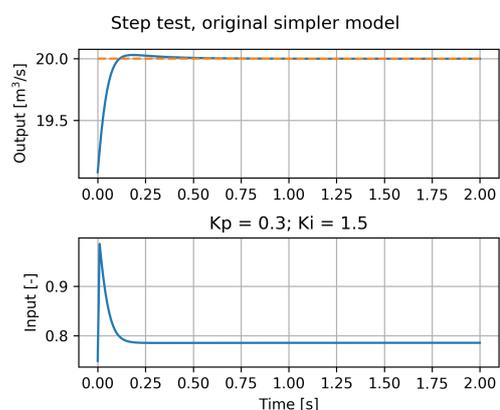


Figure 7. Step test for the nonlinear hydropower model, simpler case.

a simple PI controller, using the already exist packages in Python (*numpy* or *python-control*). The Bode diagram has been plotted for two cases of the hydropower system in order to show the frequency response of the models. Then the possibility of PI controller design has been shown for the two cases. The designed PI controller has also been tested for the original (nonlinear) models in OpenModelica.

Besides the presented examples of linear analysis, many more other possibilities for analysis and synthesis of the linearized model might be used, such as sensitivity or stability analyses, etc.

References

- Karl J. Aström and Richard M. Murray. *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, 2010.
- Bernt Lie, Sudeep Bajracharya, Alachew Mengist, Lena Bufoni, Arun Kumar, Martin Sjölund, Adeel Asghar, Adrian Pop, and Peter Fritzson. API for Accessing OpenModelica Models From Python. *Proceedings of EuroSim 2016, Oulu, Finland*, 2016.
- Valentyna Splavska, Liubomyr Vytvytskyi, and Bernt Lie. Hy-

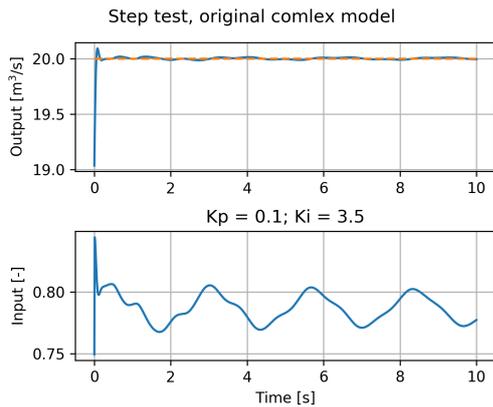


Figure 8. Step test for the nonlinear hydropower model, complex case.

dropower Systems: Comparison of Mechanistic and Table Look-up Turbine Models. *Proceedings of 58th SIMS Conference, Reykjavik, Iceland, 2017.*

Liubomyr Vytvytskyi and Bernt Lie. Comparison of elastic vs. inelastic penstock model using OpenModelica. *Proceedings of 58th SIMS Conference, Reykjavik, Iceland, 2017.*

Liubomyr Vytvytskyi and Bernt Lie. Mechanistic model for Francis turbines in OpenModelica. *Proceedings of Mathmod conference, Wien, Austria, 2018.*

Dietmar Winkler, Hege M. Thoresen, Ingvar Andreassen, Magamage A. S. Perera, and Behzad R. Sharefi. Modelling and Optimisation of Deviation in Hydro Power Production. In *Modelica Conference, 2011.*

Decision Trees for Human Activity Recognition Modelling in Smart House Environments

Veralia Gabriela Sánchez Nils-Olav Skeie

Department of Electrical Engineering, Information Technology and Cybernetics, University of South-Eastern Norway (USN), Porsgrunn, Norway, {veralia.g.sanchez, nils-olav.skeie}@usn.no

Abstract

Human activity recognition in smart house environments is the task of automatic recognition of physical activities of a person to build a safe environment for older adults or any person in their daily life. The aim of this work is to develop a model that can recognize abnormal activities for assisting people living alone in a smart house environment. The idea is based on the assumption that people tend to follow a specific pattern of activities in their daily life. An open source database is used to train the decision trees classifier algorithm. Training and testing of the algorithm is performed using MATLAB. The results show an accuracy rate of 88.02% in the activity detection task.

Keywords: *intelligent environment, behaviour modelling, pattern recognition, probabilistic model, predictive model, Norway*

1 Introduction

Human activity recognition modelling (HAM) in smart environments is an important area of research. Smart houses are being developed to improve and ease the life of the inhabitant. The idea of implementing HAM is to recognize the activities of a person in order to adapt the house to its user (Reaz, 2013; Vainio et al., 2008).

A smart house is defined as any living environment that has been carefully designed to support its inhabitant in carrying out daily activities, as well as to promote independent lifestyles (Chan et al., 2008).

People tend to follow a pattern in their daily live (Alam et al., 2010; Bourobou and Yoo, 2015). Therefore, it is possible to recognize the activities of daily life (ADL) a user performs, such as eating, toileting, bathing, dressing, etc. This recognition task is also known as human activity recognition (HAR).

Once the ADL recognition task is done, HAM can use the output from it to learn the pattern of the user and model the user's activities. The modelling has the potential to detect any deviation from the usual pattern.

Detecting abnormal activities has several applications including assisting older adults. In Norway, 38.5% of households with people aged 65 and over are living alone (sentralbyraa, 2018). Hence, a smart house can help the older adult to remain living in their own home for as long as possible (Sanchez et al., 2017).

In this work, HAR is implemented using an open source database. The output of the HAR is used for the HAM. HAM generally refers to the task of modelling the person activity pattern together with time. Therefore, accurate activity recognition is a crucial part for good HAM.

Decision trees are used to develop the HAM. Decision trees are a probabilistic algorithm that is able to predict the next step or value by learning from data. An open dataset is used for training the model.

2 Related Work

Decision tree is a supervised learning method. This method has been used for several tasks in the field of pattern recognition and machine learning as a predictive model. The main goal is to predict the next value given several input variable.

Previous studies on pervasive environment using decision trees have been successfully implemented (McBurney et al., 2008).

In smart house environments, an 80% accuracy was achieved using decision trees on 20 everyday activities in a research by Bao and Intille (2004). Another research based on decision tree with good result for ADL is the work by Fan et al. (2014).

3 Design and Methods

Figure 1 shows the methodology flow in this work.

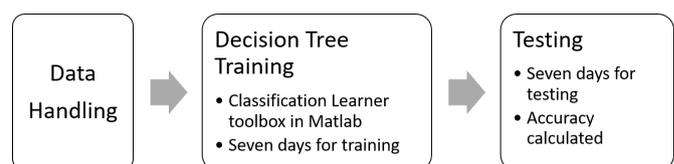


Figure 1. Methods

3.1 Decision trees

Decision trees is a hierarchical model also known as classification and regression trees. They have the property of predicting response from data.

The attributes of the decision trees are mapped into nodes. The edges of the tree represent the possible output values. Each branch of the tree represents a classification rule, from the root to the leaf node (Lara and Labrador, 2013).

3.2 Software

MATLAB is used in this study for developing the model. The classification learner toolbox was specifically used for training the tree. The code from the classification learner toolbox was exported and saved for later use with any other dataset. Testing was also coded in MATLAB.

Finally, the program Wolfram Mathematica is used for the results plots presented in this article.

3.3 Dataset

An open dataset is used in this study. The dataset has been previously used in other research and is known in the HAR field (Ordóñez et al., 2013). The dataset is named "Activities of Daily Living (ADLs) Recognition Using Binary Sensors Data Set" and is available for download at (Ordóñez, 2013). The purpose of using an open dataset is to obtain unbiased results.

The dataset consists of annotated ADLs collected by two different users living on a daily basis in a smart house. The activities in the dataset were manually labelled by the users. Table. 1 presents the dataset attributes.

Table 1. ADLs Database

<i>Name</i>	<i>Value</i>
Setting	Apartment
Number of Rooms	4 Rooms + Hall/Entrance
Number of labelled days	14 days
Labels (ADLs included)	Leaving, Toileting, Showering, Sleeping, Breakfast, Lunch, Dinner, Snack, Spare Time/TV, Grooming
Number of sensors	12 sensors
Sensors	PIR: Shower, Basin, Cooktop Magnetic: Maindoor, Fridge, Cabinet, Cupboard Flush: Toilet Pressure: Seat, Bed Electric: Microwave, Toaster

Two instances of data exist corresponding to each user living in the smart house. One dataset of 14 days (OrdóñezA), and the second dataset of 21 days (OrdóñezB). The first dataset data is depicted in Fig. 2. The first dataset

is used this work for creating and testing the model. The second dataset (OrdóñezB) is implemented later in order to test the model with a different dataset.

3.3.1 Data Handling

The variables used from the dataset are "Date", "Time", "Activity", and "Room". Another variable named "position" was added to improve the recognition task. This variable *position* correspond to one of the three following values: *laying*, *sitting*, *standing*.

Table 2 depicts the first day from the dataset. The dataset is in a text file format.

Table 2. Day 1 example of the dataset

<i>Date</i>	<i>StartTime</i>	<i>EndTime</i>	<i>Activity</i>	<i>Room</i>
28-11-11	02:27:59	10:18:11	Sleeping	Bedroom
28-11-11	10:21:24	10:23:36	Toileting	Bathroom
28-11-11	10:25:44	10:33:00	Showering	Bathroom
28-11-11	10:34:23	10:43:00	Breakfast	Kitchen
28-11-11	10:49:48	10:51:13	Grooming	Bathroom
28-11-11	10:51:41	13:05:07	Spare Time	Livingroom
28-11-11	13:06:04	13:06:31	Toileting	Bathroom
28-11-11	13:09:31	13:29:09	Leaving	Hall
28-11-11	13:38:40	14:21:40	Spare Time	Livingroom
28-11-11	14:22:38	14:27:07	Toileting	Bathroom
28-11-11	14:27:11	15:04:00	Lunch	Kitchen
28-11-11	15:04:59	15:06:29	Grooming	Bathroom
28-11-11	15:07:01	20:20:00	Spare Time	Livingroom
28-11-11	20:20:55	20:20:59	Snack	Kitchen
28-11-11	20:21:15	02:06:00	Spare Time	Livingroom

In order to model the decision tree, a sample was drawn from the dataset. All the 14 days in the dataset were stopped when the activity *leaving* was found. Seven days were used for training and seven days were used for testing.

The dataset text values were coded to numbers in order to develop the MATLAB code. Table 3 shows the *rooms* with their respective codes.

Table 3. House rooms and their code

<i>Name of Room</i>	<i>Number Assigned</i>
Bedroom	1
Bathroom	2
Kitchen	3
Livingroom	4
Hall	5

Numbers were also assigned to the *activities* to make the learning and decoding process more feasible. Table 4 shows the *activities* with the assigned codes.

Table 5 shows the coding used for the *position* values.

A total of 9 *activities*, 5 *rooms*, and 3 *positions* are used.

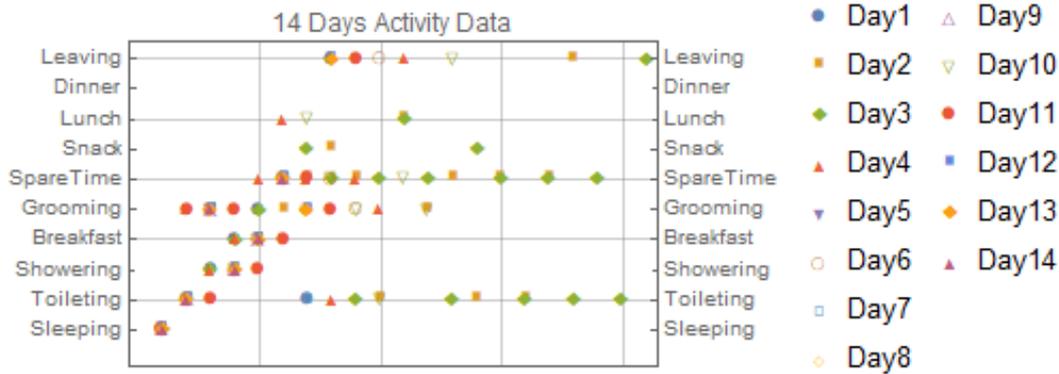


Figure 2. Activities Dataset Graph

Table 4. Activities numbers

Name of Activity	Number Assigned
Sleeping	1
Toileting	2
Showering	3
Breakfast	4
Grooming	5
Spare time/TV	6
Snack	7
Lunch	8
Leaving	9

and their code

Table 5. Position numbers and their code

Name of Activity	Number Assigned
Lying	1
Sitting	2
Standing	3

3.4 HAR Modelling (HAM)

HAM refers to the modelling of the behaviour or activity of the person. Behaviour is regarded as an activity with duration, i.e, the time elapsed from start to end of an activity and time of day (Pfeiffer et al., 2016). For example, a behaviour can be having breakfast, which consist of opening the refrigerator, cooking, sitting and eating breakfast. This set of activities are given in a time span (time elapsed from start to end), and usually in the morning (time of day).

Normal and abnormal activity and behaviour can be detected in a smart house by analysing both, the activity and the time. Abnormal activity detection main purpose is to warn a member of the family or caretaker whether something is wrong with the person. This can be regarded as anomaly. "Anomaly detection refers to the problem of finding patterns in data that do not conform to expected behavior" (Chandola et al., 2009).

4 Experiments

The dataset used in this study is available in a text file only. Therefore, the dataset was exported to an excel file. The dataset contains 14 days of data in total. Two files were created, one for training and one for testing. Random numbers was generated in MATLAB with the *randperm* function to randomly select seven days for training. The numbers selected according to the random generator were days: 12, 10, 5, 14, 1, 7, and 6. Hence, these days were use for training. The remaining days (2, 3, 4, 8, 9, 11, 13) were used for testing.

A new variable called *duration* was added. The variable *duration* was calculated using the *time* data from the dataset and consist of the time spent in each activity, from start to end of each activity. The *duration* value was calculated in seconds.

In the excel file, the text values of the dataset were coded to numbers. The variables *room*, *position* and *activities* were coded as explained in section 3.3.1. The *room* values were coded to numbers from 1 to 5. The *position* values coded to numbers from 1 to 3. The *activity* values were coded to numbers from 1 to 9.

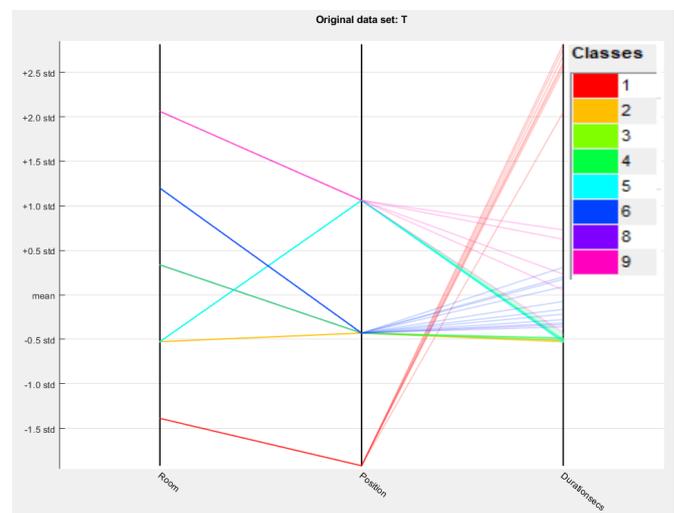


Figure 3. Parallel coordinated plots

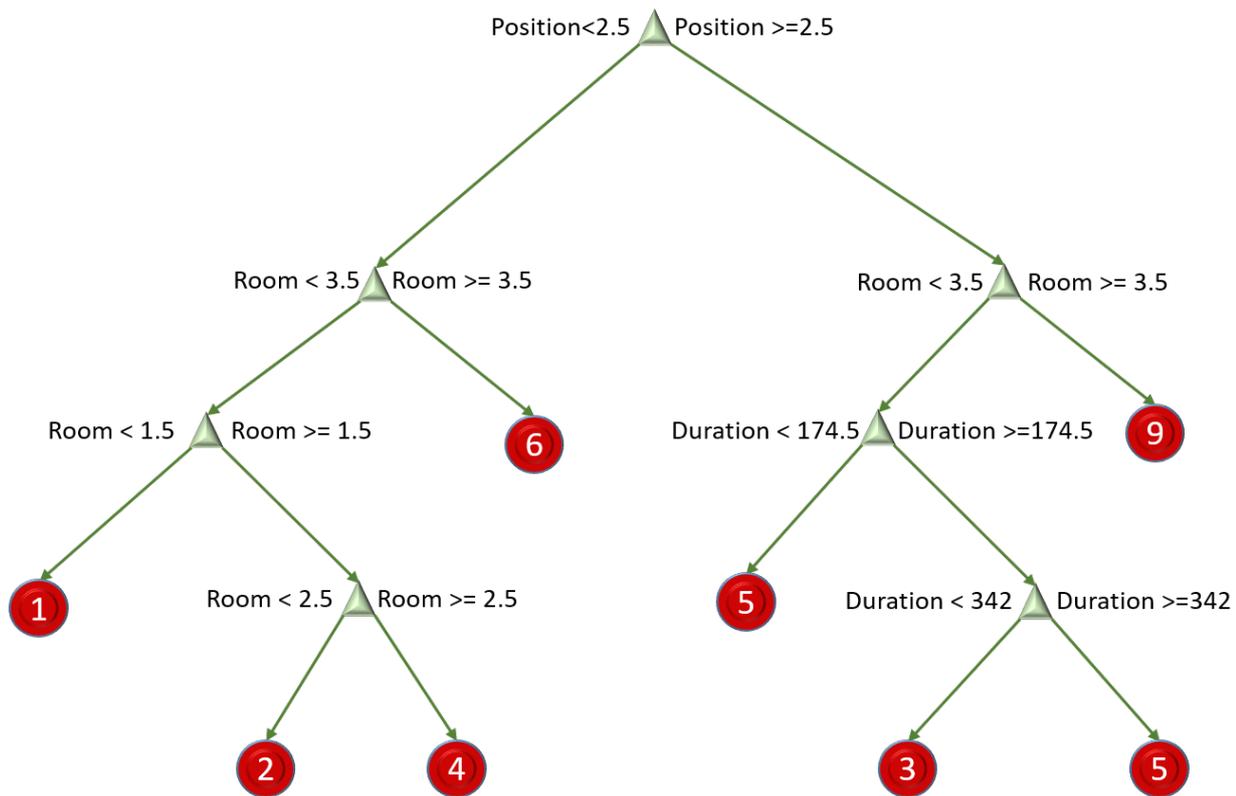


Figure 4. Trained Decision Tree. Red circles represent the activities

Both excel files were imported to MATLAB as table data type. The training was performed using the integrated classification learner toolbox. The variables used for training the decision trees are the *room*, *position*, and *duration*. The output variable is the *activity* data.

Figure 3 shows the parallel coordinated plots of the data. The variables *room*, *position*, and *duration* are plotted to show the relationship between them. According to Figure 3, it is possible to see that activities 3 and 5 (*showering and grooming*) follow almost the same path line in the graph. Also, activities 4, and 8 (*breakfast and lunch*) almost follow the same path line, with the *duration* barely different for each of the two activities. Activity 7, *snack*, was not found in the training dataset.

Once the tree is trained, testing is performed with the remaining seven days of the dataset: days 2, 3, 4, 8, 9, 11, 13. The testing consists on using the variables *room*, *position*, and *duration* as input data. The response or output is the *activity* value. Each day from the testing dataset was tested and compared to the real data.

A new fictional test set was created in order to test the model with abnormal data, as showed in table 6. The test set consists of a fictional single day. The table shows that the *duration* of some of the activities were exaggerated. In addition, the *position: lying* of the first activity in the *hall* room should qualify as abnormal behaviour.

Result plots were obtained using the Mathematica software. The actual data and the predicted data for each of

Table 6. Added test day

<i>Room</i>	<i>Position</i>	<i>Durationsecs</i>	<i>Activity</i>
Hall	Lying	10000	-
Bathroom	Sitting	15000	toileting
Bathroom	Standing	450	grooming
Living-room	Sitting	9000	spare time
Living-room	Sitting	9500	spare time
Hall	Standing	412	leaving

the testing days was copied to Mathematica and plots were coded to visually present the results.

Finally, the total computational time was measured.

5 Results

Figure 4 shows the trained decision tree. The decision tree model was able to classify seven out of nine activities in the dataset.

Figure 5 shows the number of observations for each of the activities. The true class is in the y-axis and the predicted class is in the x-axis. It is possible to see that there are no observation in the training dataset for activity number seven (*snack*), and only one observation for activity number eight (*lunch*).

Most error counts in figure 5 occurred in activities that are performed in the same *room*, such as *showering and grooming* (3 and 5). However, this number of observations

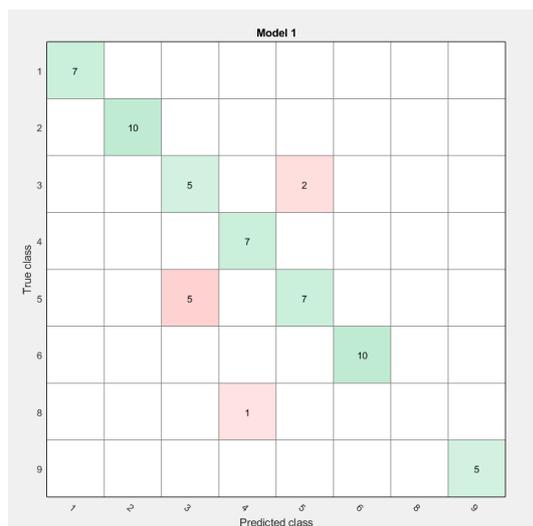


Figure 5. Number of observations

with errors is low.

Figure 6 shows in percentage of success and errors prediction in the training data, called "positive predictive values false discovery rate". The highest false discovery is 50% in the activity *grooming* (5). The model classified the activities *showering* (3) half of the times instead of the true class *grooming* (5).

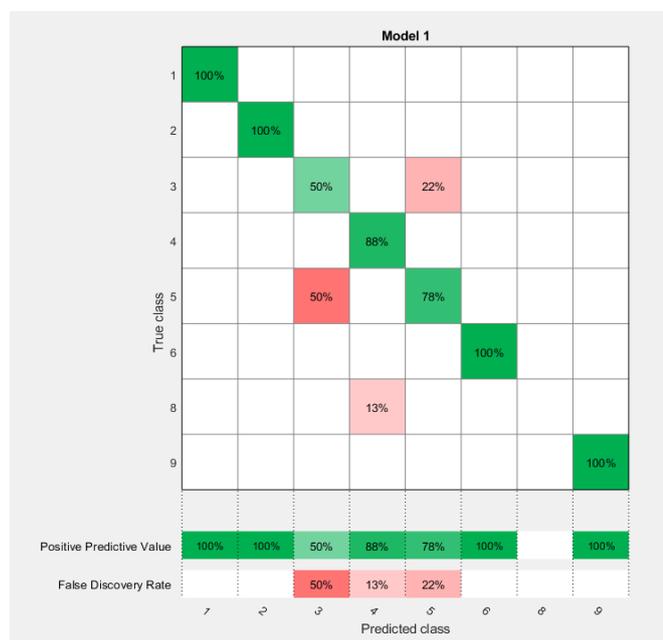


Figure 6. Positive predictive values and false discovery rates

For the activity *grooming* (5), the model had a false discovery rate of 22%, classifying the activity *showering* (3) instead of *grooming* (5).

For the activity *lunch* (8), the model classified the activity *breakfast* (4) with a false discovery of 13% .

Figure 7 shows the results of the test. Days 2, 3, 4, 8, 9, 11, and 13 were used for testing. Some prediction

errors were found when comparing the actual data with the estimated data.

Most of the errors were found between the *grooming* and *showering* activity, and between the *breakfast* and *snack* activities. A possible explanation for these prediction errors is that both of these activities are performed in the same room, *bathroom* and *kitchen*, respectively.

Figure 7h shows the results of the added fictional day with abnormal behaviour data. The model predicted the activity *Spare Time* instead of finding an abnormal behaviour in the first activity. The true positive-false posi-

Table 7. True positive \false positive rate for each activity

Activity	True Positive	False Positive
Sleeping	100%	0
Toileting	100 %	0
Showering	77%	23%
Breakfast	98%	2%
Grooming	94%	6%
Spare time	100 %	0
Lunch	36%	64%
Leaving	100 %	0

tive rate for each fo the predicted activities are shown in table 7.

Finally, the total accuracy of the activity recognition task is 88.02%. The computational time of the model, consisting of training and testing is around 3 seconds.

5.1 Test on second dataset

The model was tested on the second dataset (OrdonezB) consisting of 21 days, also open source as described in section 3.3. The purpose of this second test is to verify that the model works with any dataset.

In this test, the entire dataset was used, without sampling. The dataset was also processed as described in section 4. The values were coded to numbers. A total of 10 activities, 5 rooms and 3 positions were used.

The results showed that the model worked as well as in the experimental work (dataset Ordonez A). Like in the experimental work, minor mistakes were found in the prediction task corresponding to activities made in the same room. Namely *bathroom* and *kitchen*. Therefore, the model presented here is able to work with any dataset.

6 Discussion

In this work, decision trees are researched to perform human activity recognition modelling.

The decision tree classified seven out of nine activities. This is because there are no observation of activity *snack*, and only one observation for activity *lunch* in the training dataset. Therefore, the model could only classify seven activities in total.

Some predictions presented minor error rates. One possible reason for the these error rates is that there are rooms

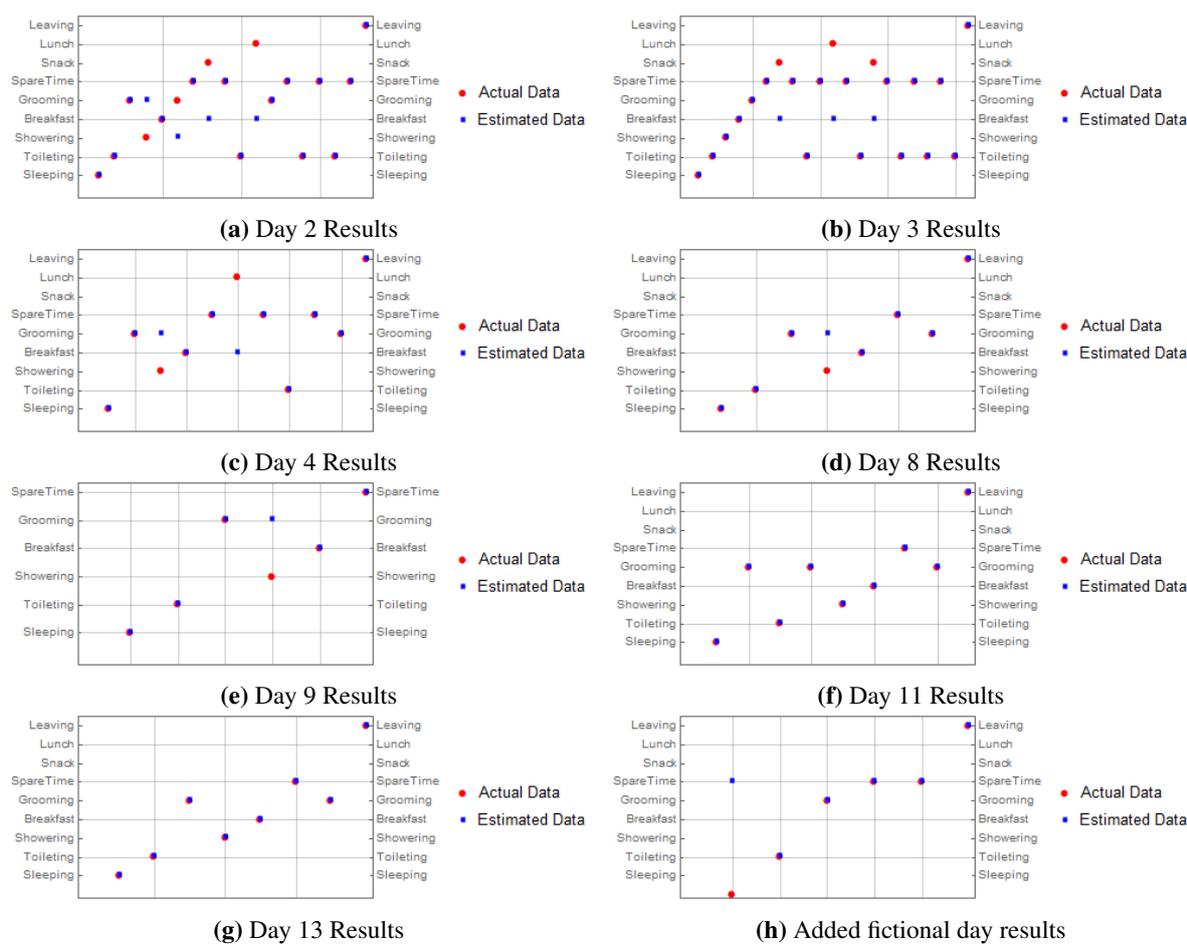


Figure 7. Results

that allow different types of activities. Hence, the recognition task is more difficult. For example, in the room *bathroom*, three different activities are performed: *showering*, *toileting* and *grooming*. Thus, the model tends to predict the highest probability of the *activity* given the room *bathroom*.

This is the same case for the room *kitchen*, where activities *breakfast*, *lunch* and *snack* are performed. The activity *lunch* has an error rate of 64%. From figure 5 it is possible to see that there is only one observation of the activity *lunch*. Therefore, the model would hardly predict this activity. Instead, the model predicts the activity *breakfast*, because it has the highest probability.

Decision trees are probabilistic algorithm and thus produces some errors in the prediction task. As any probabilistic algorithm, decision trees will always chose the highest probability according to the trained data.

In general, the finding suggest that decision trees are a good tool for HAR with 88.02% accuracy. However, for the HAM, the model does not detects abnormal behaviour as well as it does HAR. When a fictional single day test set was created to check the performance on an abnormal day, the model did not meet the expectations in the anomaly detection task.

The most like reason for this, is that decision trees does

not always enforce to check every variable before estimating a results. Consequently, in the added fictional test day with room *hall* and position *lying*, the tree predicted the activity *spare time*. Thus, the tree does not check for yhe other variables of *room*, nor *duration*. The model should have detected an abnormal behaviour in this scenario, since *lying* in the *hall* is not a normal activity, but a possible fall.

Another reason for the model not detecting abnormal behaviour is that abnormal situations need to be trained in decision trees. This means that all possible abnormal scenarios need to be learned *a priori*. As a result, the finding suggest that decision trees are not the best option for detecting abnormal activities or behaviour.

The model was also tested with the second dataset available (OrdenezB) to verify that the model is able to work with any dataset. The results obtained were similar to the experimental work. Minor mistakes were found in activities performed in the same room.

Possible solutions for improving the model are more research on how to enforce the decision trees to check every single parameter. Another option could be to combine decision trees with another probabilistic method to increase the accuracy of the model.

Finally, HAM would ideally keep the activity history of

the user in order to model the behaviour of the person. For example, if the user has followed the pattern *wake up, toiletting, grooming, showering and breakfast*, the most normal behaviour would be not to repeat any of those activities again within a given frame time.

7 Ethics in Smart Houses

Smart house technology, like any other type of technology, can carry many ethical challenges. Therefore, a separate study has been carried at USN to address this topic. We consider that the ethical aspects are an important part of our research in smart house technology.

Among the main challenges found that smart houses presents are cost-effectiveness, privacy, autonomy, informed consent, dignity, safety, and trust (Sánchez et al., 2017).

These challenges are central to keep in mind when developing a smart house system. Developers need to be aware of these challenges in order to provide a safer and dignified environment for the users. Nevertheless, it is important to acknowledge that smart house systems, at some point, cannot solve all the problems that are related to ageing, disabilities and diseases. There are needs that people develop as they age and smart house technology cannot help them any more (Sánchez et al., 2017).

8 Conclusion and Future work

In this study, activity recognition modelling (HAM) is researched. The goal is to find the normal and abnormal behaviour of the person living in a smart house. Decision trees have been used to perform activity recognition because they can predict responses to data. The output from the activity recognition task is used as an input for the modelling task.

The input data for the decision trees learning task are the *rooms, duration* and *position*. The responses are the *activities*. A total accuracy of 88.02% was achieved for activity recognition using decision trees. Thus, decision trees can be a good tool for activity recognition. However, HAM did not meet the expected results.

The reason for this is that decision trees does not enforces to verify every single input variable before calculating a result. Therefore, more research on how to check every variable before estimating a results needs to be studied. Alternatively, combining decision tree algorithm with another probabilistic model could be a possible solution for HAM.

References

MR Alam, MBI Reaz, M Ali, SA Samad, FH Hashim, and MK Hamzah. Human activity classification for smart home: A multiagent approach. In *Industrial Electronics & Applications (ISIEA), 2010 IEEE Symposium on*, pages 511–514. IEEE, 2010.

Ling Bao and Stephen S Intille. Activity recognition from user-

annotated acceleration data. In *Pervasive computing*, pages 1–17. Springer, 2004.

Serge Thomas Mickala Bourobou and Younghwan Yoo. User activity recognition in smart homes using pattern clustering applied to temporal ann algorithm. *Sensors*, 15(5):11953–11971, 2015.

Marie Chan, Daniel Estève, Christophe Escriba, and Eric Campo. A review of smart homes’ present state and future challenges. *Computer methods and programs in biomedicine*, 91(1):55–81, 2008.

Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

Xiaohu Fan, Hao Huang, Changsheng Xie, Zhigang Tang, and Jing Zeng. Private smart space: Cost-effective adls (activities of daily livings) recognition based on superset transformation. In *Ubiquitous Intelligence and Computing, 2014 IEEE 11th Intl Conf on and IEEE 11th Intl Conf on and Autonomic and Trusted Computing, and IEEE 14th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UTC-ATC-ScalCom)*, pages 757–762. IEEE, 2014.

Oscar D Lara and Miguel A Labrador. A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys and Tutorials*, 15(3):1192–1209, 2013.

Sarah McBurney, Eliza Papadopoulou, Nick Taylor, and Howard Williams. Adapting pervasive environments through machine learning and dynamic personalization. In *2008 IEEE International Symposium on Parallel and Distributed Processing with Applications*, pages 395–402. IEEE, 2008.

Ordonez. Activities of daily living (adls) recognition using binary sensors data set, 2013. URL <https://archive.ics.uci.edu/ml/datasets/Activities+of+Daily+Living+28ADLs29+Recognition+Using+Binary+Sensors>. Accessed: 2017-05-01.

Fco Javier Ordóñez, Paula de Toledo, and Araceli Sanchis. Activity recognition using hybrid generative/discriminative models on home environments using binary sensors. *Sensors*, 13(5):5460–5477, 2013.

Carlos F Pfeiffer, Veralia Gabriela Sánchez, and Nils-Olav Skeie. A discrete event oriented framework for a smart house behavior monitor system. In *Intelligent Environments (IE), 2016 12th International Conference on*, pages 119–123. IEEE, 2016.

Mamun Bin Ibne Reaz. Artificial intelligence techniques for advanced smart home implementation. *Acta Technica Corviniensis-Bulletin of Engineering*, 6(2):51, 2013.

Veralia Gabriela Sanchez, Carlos F Pfeiffer, and Nils-Olav Skeie. A review of smart house analysis methods for assisting older people living alone. *Journal of Sensor and Actuator Networks*, 6(3):11, 2017.

Veralia Gabriela Sánchez, Ingrid Taylor, and Pia Cecilie Bing-Jonsson. Ethics of smart house welfare technology for older

adults: A systematic literature review. *International Journal of Technology Assessment in Health Care*, pages 1–9, 2017. doi:10.1017/S0266462317000964.

Statistisk sentralbyrå. Key figures for the population, 2017. <https://www.ssb.no/en/befolkning/nokkeltall/population>, 2018. Accessed 2017-12-10.

Antti-Matti Vainio, Miika Valtonen, and Jukka Vanhala. Proactive fuzzy control and adaptation methods for smart homes. *Intelligent Systems, IEEE*, 23(2):42–49, 2008.

Comparison of Simulation Tools to Fit and Predict Performance Data of CO₂ Absorption into Monoethanol Amine at CO₂ Technology Centre Mongstad (TCM)

Lars Erik Øi^{1,*} Kai Arne Sætre¹ Espen Steinseth Hamborg²

¹Department of and Process, Energy and Environmental Technology, University College of Southeast Norway

²CO₂ Technology Centre Mongstad DA

lars.oi@usn.no

Abstract

In this work, several sets of experimental data from the amine based CO₂ capture process at CO₂ Technology Centre Mongstad (TCM) have been compared with simulations of different equilibrium based models and a rate-based model. The equilibrium models (in Aspen Plus and Aspen HYSYS) were fitted by adjusting the Murphree efficiency for each stage and the rate-based model (in Aspen Plus) was fitted by adjusting the interfacial area factor. Aspen Plus (using the Electrolyte-NRTL model) and Aspen HYSYS (using Kent-Eisenberg and Li-Mather models) gave almost identical results for the capture rate and small deviations for the temperature profiles. There are however deviations both between the measured temperatures at a specified column height and between measured temperatures and the simulated temperatures. Equilibrium based models are less fundamental than rate-based models, but for the conditions in this study, the rate-based models still lack accurate input parameters like the interfacial area. The results from this study show that equilibrium and rate-based models perform equally well in both fitting performance data and in predicting performance at changed conditions.

1 Introduction

Developing robust and predictable process simulation tools for CO₂ capture is an important step in improving carbon capture technology to reduce man-made carbon emissions. Examples of available process simulation tools for CO₂ absorption into amine solutions are the equilibrium based models in Aspen Plus and Aspen HYSYS and the rate-based model in Aspen Plus. Equilibrium based absorption models are based on the assumption of equilibrium at each stage. The model can be extended by introducing a Murphree efficiency (the ratio of the change in mole fraction from a stage to the next divided by the change assuming equilibrium). Rate-based models are based on rate expressions for chemical reactions, mass transfer and heat transfer.

At CO₂ Technology Centre Mongstad (TCM) there is an absorption column with a rectangular cross section of 3.55 times 2 meter which is equivalent to a packing diameter of 3 meter, and a packing height up to 24 meter. At TCM, performance tests of CO₂ absorption from flue gas into 30 wt-% monoethanol amine (MEA) have been run in 2013 (Thimsen et al., 2014; Hamborg et al., 2014) and in 2015 (Gjernes et al., 2017). Figure 1 shows a simplified process diagram of the amine based CO₂ absorption and desorption facility at TCM.

Keywords: CO₂, amine, absorption, simulation

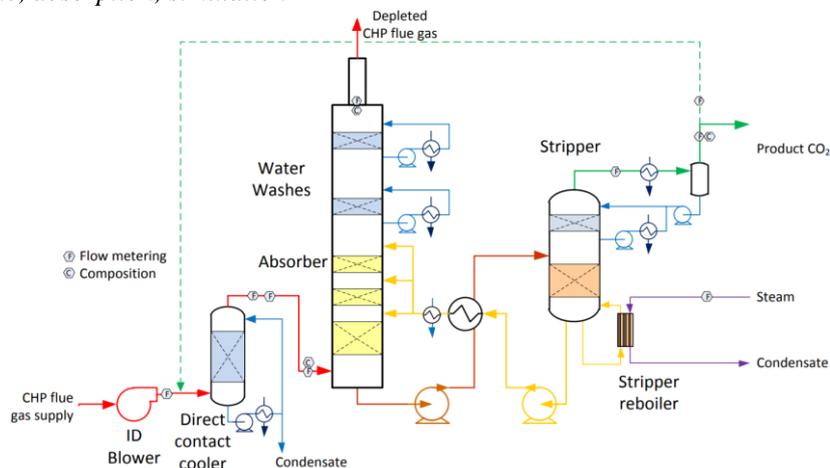


Figure 1. Simplified process diagram of the amine based CO₂ capture plant at TCM (Thimsen et al., 2014)

A performance test at TCM is normally run at constant conditions for a long period of time to obtain steady state conditions. In this work, the emphasis is on the absorber part of the process. Especially the total CO₂ capture rate (in % of incoming CO₂) in the absorption section and the temperature profile from top to bottom of the absorption section are the evaluated parameters.

The aim of this work is to compare results from simulations with performance data for CO₂ absorption into 30 wt-% MEA at TCM using different simulation tools. In this work, 4 sets of experimental data (scenarios) from the amine based CO₂ capture process at TCM have been compared with simulations of different equilibrium based models and a rate-based model. The work is based on the Master Thesis of Kai Arne Sætre (2016). Some of the results from the Master Thesis were presented as a non-published Poster at the TCCS-9 conference in Trondheim in June 2017. In the original work (Sætre, 2016) the equilibrium models (in Aspen Plus and Aspen HYSYS) were fitted to one specific scenario by adjusting the Murphree efficiency for each stage, and the rate-based model (in Aspen Plus) was fitted to another scenario by adjusting the interfacial area factor. In this work, the Murphree efficiency for each stage was adjusted in the other scenarios to achieve a good fit to the temperature profile. The interfacial area was constant in all the calculations for the rate-based model, and this gave a good fit to the CO₂ capture rate and reasonably good fit to the temperature profile.

2 Available Equilibrium and Kinetic Models

There are several equilibrium models available for the MEA/water/CO₂ system describing the relations between the vapour and liquid phase at equilibrium. Aspen HYSYS has an amine package with the Kent-Eisenberg (1976) and the Li-Mather (1994) equilibrium models. Aspen Plus has an Electrolyte-NRTL equilibrium model which is based on Austgen et al. (1989). The column models in both Aspen HYSYS and Aspen Plus can be specified with Murphree efficiencies.

In the Master Thesis work of Zhu (2015) and Sætre (2016), a Murphree efficiency for each stage (meter of packing) was estimated for a set of TCM data (Hamborg et al., 2014). Zhu (2015) found that a Murphree efficiency fitted to 0.09 for all stages obtained good agreement between measured and simulated CO₂ capture rate. Using different Murphree efficiencies for each stage, it was possible to achieve also good agreement between the measured and simulated temperature profile.

In Aspen Plus, there are several models for heat transfer, mass transfer and kinetics which can be included in a rate-based calculation. A rate-based example file for CO₂ removal using MEA is available with the Aspen Plus program. The parameters in this file

are mostly based on the work of Zhang et al. (2009) who fitted Aspen Plus simulations to experimental runs at a CO₂ absorption pilot plant at the University of Texas. The rate-based models used at TCM have been developed during several years, and different models have been used in the Master Thesis works of Larsen (2014), Desvignes (2015) and Sætre (2016).

There have been published very few comparisons between different simulation tools for CO₂ absorption. One reference comparing different simulation programs for CO₂ removal from atmospheric gas is Luo et al. (2009). They tested Aspen RadFrac, ProTreat, ProMax, Aspen RateSep, CHEMASIM from BASF and CO2SIM from SINTEF/NTNU and compared with pilot plant data. They claimed that all models were capable of fitting the capture rate, but the temperature and concentration profiles were less well predicted. Øi (2012) has performed comparisons between the programs Aspen HYSYS and Aspen Plus. He claimed that there were small differences between the tested equilibrium models, and that a rate-based and equilibrium based model with estimated Murphree efficiencies gave similar results.

In literature, different models and tools are mentioned. In this work a simulation tool is meant as a complete collection of models for equilibrium, rate expressions and efficiencies including defined parameter values for all the models. It should be noted that most of the models used are very dependent on the parameter values.

3 Material, Methods and Specifications

3.1 Performance Data from TCM

Performance data for this work have been taken from 4 sets of conditions (scenarios) at TCM. They are from campaigns in 2013 and 2015 for approximately 30 wt-% MEA in water. 24 meter of packing height (the maximum available) was used in these scenarios. These scenarios were all run with anti-foam to avoid operating problems due to foaming. Only the performance of the absorber part of the process was evaluated in this work. The conditions in each scenario are mainly defined by the conditions of the inlet gas stream and the inlet amine stream to the absorption section of the absorption column. The performance test data cover measurements from all the process units in Figure 1. Each of the scenarios were run for several days to obtain measurements for steady state conditions.

The data for the 4 scenarios are listed in Table 1, Table 2, Table 3 and Table 4 (from Sætre, 2016). The data are from scenarios documented in Hamborg (2014) and Gjernes (2017), but some of the data are converted to different units to make them suitable for input to simulation programs.

The 4 scenarios which have been selected in this work are named H14 and 6w from 2013 (Hamborg, 2014) and 2B5 and Goal1 from 2015 (Gjernes, 2017). The names have been used internally at TCM, except the H14 scenario which is based on a temperature profile with mean values from the 4 locations for each packing height of one meter. The 4 scenarios were run with amine concentrations close to 30 wt-% MEA in water. The measured MEA concentrations were 30.0 (H14), 30.4 (6w), 31.6 (2B5) and 32.4 wt-% (Goal1).

Table 1 Scenario H14 experimental input data for process simulations.

Input data to the simulations			
Amine inlet		Flue gas inlet	
Flow rate [kg/h]	54900	Flow [kmol/h]	2022
Temperature [°C]	36.5	Temperature [°C]	25.0
MEA [mol%]	10.94	CO ₂ [mol%]	3.7
H ₂ O [mol%]	86.54	H ₂ O [mol%]	2.95
CO ₂ [mol%]	2.52	O ₂ [mol%]	13.6
Pressure [bara]	1.0313	N ₂ [mol%]	79.75
		Pressure [bara]	1.063

Table 2 Scenario 6w experimental input data for process simulations.

Input data to the simulations:			
Amine inlet		Flue gas inlet	
Flow rate [kg/h]	54915	Flow [kmol/h]	2005
Temperature [°C]	36.9	Temperature [°C]	25
MEA [mol%]	11.13	CO ₂ [vol%]	3.57
H ₂ O [mol%]	86.37	H ₂ O [vol%]	3.0
CO ₂ [mol%]	2.5	O ₂ [vol%]	13.6
Pressure [bara]	1.0313	N ₂ [vol%]	79.83
		Pressure [bara]	1.063

Table 3 Scenario 2B5 experimental input data for process simulations.

Input data to the simulations			
Amine inlet		Flue gas inlet	
Flow rate [kg/h]	49485	Flow [kmol/h]	2022
Temperature [°C]	36.8	Temperature [°C]	28.2
MEA [mol%]	11.67	CO ₂ [mol%]	3.57
H ₂ O [mol%]	85.65	H ₂ O [mol%]	3.7
CO ₂ [mol%]	2.68	O ₂ [mol%]	14.6
Pressure [bara]	1.0313	N ₂ [mol%]	78.08
		Pressure [bara]	1.063

Table 4 Scenario Goal1 experimental input data for process simulations.

Input data to the simulations			
Amine inlet		Flue gas inlet	
Flow rate [kg/h]	44391	Flow [kmol/h]	2017
Temperature [°C]	28.6	Temperature [°C]	25
MEA [mol%]	12.04	CO ₂ [mol%]	3.62
H ₂ O [mol%]	85.19	H ₂ O [mol%]	3.1
CO ₂ [mol%]	2.77	O ₂ [mol%]	14.3
Pressure [bara]	1.0313	N ₂ [mol%]	79
		Pressure [bara]	1.063

The results from the performance data scenarios to be compared with simulations, are the total CO₂ capture rate and the temperature profile measured for every meter of the packing. The temperature was measured at different locations for each meter of packing, and the different locations were named A, B, C and D. For the H14 scenario, a mean value for each meter of packing was specified in the temperature profile (Hamborg, 2014).

3.2 Specifications for the Equilibrium Based Simulation Tools

When using Aspen HYSYS version 8.0, the Amine package with the Kent-Eisenberg model was used with non-ideal vapor phase. In the work of Sætre (2016) also the Li-Mather model was used, but the results with the Li-Mather model are not included in this work.

When using Aspen Plus version 8.0, the Electrolyte-NRTL (Non-Random-Two-Liquid) was used. The sample file available from Aspen Plus, was used as the basis for the Aspen Plus simulations.

In the Master Thesis work of Zhu (2015) and in Sætre (2016), a Murphree efficiency for each of the 24 stages (meter of packing) was estimated for the TCM data set (Hamborg 2014). Zhu (2015) found that a Murphree efficiency of 0.09 for all stages gave a good fit to the capture rate. This approach was used by Sætre (2016) in his work, but these results are not presented in this work. Zhu (2015) also found that a linear Murphree efficiency profile gave a better fit to the temperature profile. A Murphree efficiency of 0.23 at the top stage and then reduced linearly to 0.09 for stage 14 and constant equal to 0.01 (close to 0) for the bottom stages fitted the temperature profile very well for the H14 scenario (Zhu, 2015). This Murphree efficiency was used in all the scenarios in the work of Sætre (2016). In this work, this was found to be satisfactory only for the scenarios H14 and Goal1. The capture rate and temperature profile in the scenarios 6w and 2B5 scenarios were found to be well predicted by a linear efficiency profile with $E_M = 0.192$ at the top stage and 0.008 at the bottom. This was specified in this work for 6w and 2B5 in both the Aspen HYSYS and Aspen Plus simulation tool.

3.3 Specifications for the Rate-Based Tool

The specifications in the rate-based Aspen Plus simulation tool at TCM have been developed during several years and different versions have been used (Larsen, 2014; Desvignes, 2015; Sætre 2016). Especially the parameters in the Electrolyte-NRTL model are not the same in all versions. In this work, the parameters in the sample file from Aspen Plus version

8.0 (Rate_Based_MEA_Model) were used. Most of these specifications are based on the work by Zhang et al. (2009) where Aspen Plus rate-based simulations were fitted to pilot scale experiments of CO₂ absorption at the University of Texas. The specifications for the rate-based model (in the APRB_TCM2016 file) is shown in Table 5.

Table 5. Specifications for the rate-based model in the APRB_TCM2016 file

SPECIFICATION	
Calculation type	Rate-based
Number of stages	50
Efficiency type	Vaporization efficiencies
Reaction ID	MEA-NEW
Holdup	0.0001 stage 1 to 50
Reaction condition factor	0.9
Packing type	Koch metal 2x
Section diameter [m]	3
Section packed height [m]	24
Flow model	Countercurrent
Interfacial area factor	0.55 (0.5 to 1)
Film Liquid phase	Discrxn
Film Vapor phase	Film
Mass transf coeff method	Bravo et al. 1985
Heat transf coeff method	Chilton and Colburn
Interfacial area method	Bravo et al. 1985
Holdup method	Bravo et al. 1992
Add. Discretize points liquid	5

Detailed documentation of the rate-based model can be found in the Aspen Plus program documentation. References to the mass transfer, interfacial area and hold-up models are Rocha et al. (1985) and Rocha et al. (1993), and for the heat transfer coefficient method Chilton and Colburn (1935).

4 Results

4.1 General Results

The results from the simulations are mainly the capture rate and the temperature profile. There are 3 simulation tools used, and no parameters are changed in the simulation tools from scenario to scenario except for the Murphree efficiencies. Simulation results and performance data are compared for each scenario. In general, the simulated capture rate was reasonably close to the measured capture rate for all scenarios except for the Goal1 scenario. Because of that, emphasis is in this work on comparison of temperature profiles.

4.2 Scenario H14

Measured capture rate was 88.5 %. Aspen HYSYS achieved 86.9 %, Aspen Plus equilibrium based 86.9 and Aspen Plus rate-based 88.5 %. Comparison between measured and simulated temperature profile is shown in Figure 2.

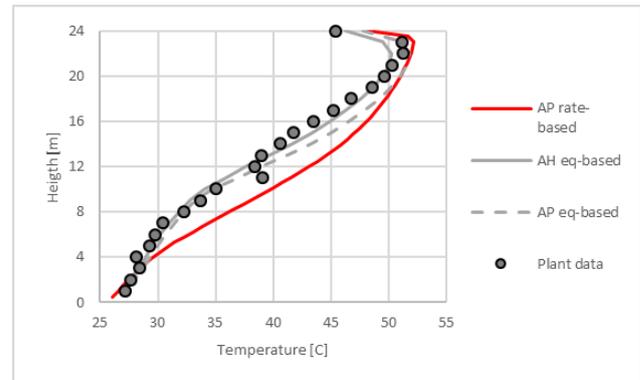


Figure 2. Comparison of plant data scenario H14 and simulated temperature profiles.

The deviation is less than 2 °C for the equilibrium models and less than 6 °C for the rate-based model. It must be noted that the Murphree efficiencies in the equilibrium based models were actually fitted to the plant data.

4.3 Scenario 6w

Measured capture rate was 88.5 %. Aspen HYSYS achieved 87.2 %, Aspen Plus equilibrium based 87.5 and Aspen Plus rate-based 86.1 %. Comparison between measured and simulated temperature profile is shown in Figure 3.

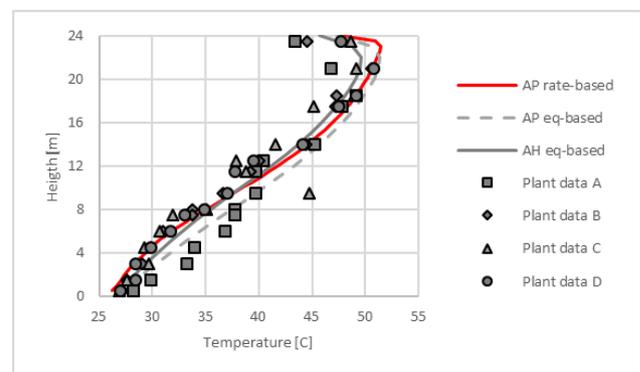


Figure 3. Comparison of plant data scenario 6w and simulated temperature profiles.

With the exception of the plant data A and one outlier temperature from plant data C, the deviations between measured and simulated temperatures were less than 3 °C. In this case the Murphree efficiencies were fitted as a linear profile from top to bottom. It must be noted that the interfacial area factor was fitted (to 0.55) for the Aspen Plus rate-based model to fit the capture rate and the temperature profile for this scenario.

4.4 Scenario 2B5

Measured capture rate was 87.2 %. Aspen HYSYS achieved 87.3 %, Aspen Plus equilibrium based 87.5 and Aspen Plus rate-based 86.0 %. Comparison between measured and simulated temperature profile is shown in Figure 4.

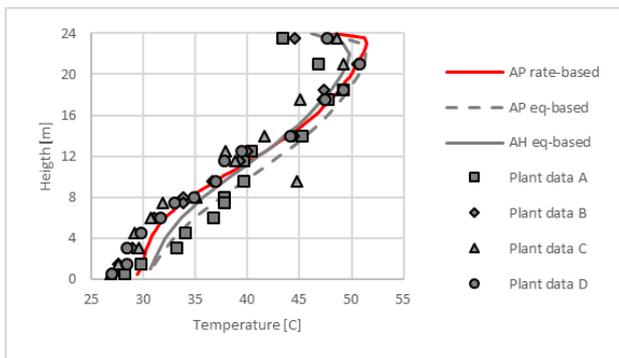


Figure 4. Comparison of plant data scenario 2B5 and simulated temperature profiles.

With the same exception of the plant data A and one temperature from plant data C, the deviations between measured and simulated temperatures were less than 2 °C. Also in this case the linear Murphree efficiency profile gave a good fit to the temperature profile. The equilibrium based temperatures were slightly less than the measured temperatures, while the rate-based temperatures were close to the measured.

4.5 Scenario Goal1

Measured capture rate was 90.1 %. Aspen HYSYS achieved 86.2 %, Aspen Plus equilibrium based 82.7 and Aspen Plus rate-based 78.9 %. This is the only case where there is a significant deviation between performance data and capture rate. The deviation is 4 %-points for the Aspen HYSYS equilibrium based model, 7 %-points for Aspen Plus equilibrium based and 11 %-points for the Aspen Plus rate-based model. Comparison between measured and simulated temperature profile is shown in Figure 5.

The deviations between measured and simulated temperatures were less than 3 °C. The Murphree efficiency profile from Zhu (2015) gave a reasonably good fit for the equilibrium based models with less than 2 °C difference between the models.

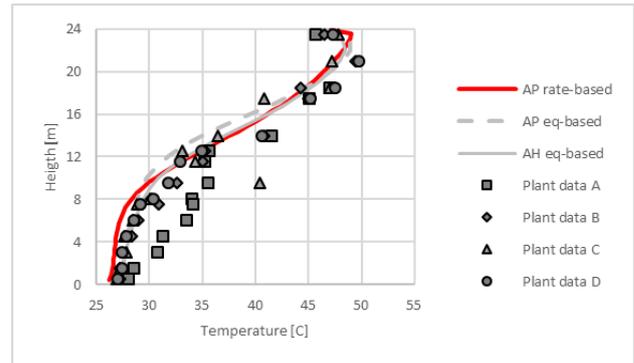


Figure 5. Comparison of plant data scenario Goal1 and simulated temperature profiles.

5 Discussion

There is very little difference in the results from different equilibrium models like Kent-Eisenberg and Electrolyte-NRTL. Sætre (2016) also simulated the 4 scenarios in this work with the Li-Mather model. The Li-Mather and Kent-Eisenberg models gave very similar results, and this has also been experienced earlier (Øi, 2012). There is a difference of approximately 2 °C in the maximum temperature in the upper part of the column between Kent-Eisenberg and Electrolyte-NRTL, but this does not seem to give differences in the removal rate. When simulating the Goal1 scenario, the maximum temperature in the simulation tools were very close to each other.

All the tools are close to similar when it comes to prediction of the total absorption rate. Fitting the models to achieve the measured absorption rate can be performed for one scenario by adjusting the Murphree efficiency for the equilibrium based models or the interfacial area factor for the rate based models. For the equilibrium based models, two sets of Murphree efficiencies were used to achieve reasonable results for all 4 scenarios. The ability to predict performance at other conditions was about the same for the different simulation tools. Luo et al. (2009) have tested Aspen RadFrac, ProTreat, ProMax, Aspen RateSep, CHEMASIM and CO2SIM. As in this work, they concluded that basically all the codes were capable of giving reasonable predictions on overall CO₂ absorption rate.

The figures show that measured temperature profiles and simulated temperature profiles are reasonably close. There are however deviations both between the measured temperatures at different locations at a specified column height and between measured temperatures and the simulated temperatures. It is not obvious that any of the simulation tools predicts the measured temperatures better than the other models.

Some references (Zhang et al., 2009; Larsen, 2014; Desvignes, 2015) have compared also the concentration profiles when comparing performance data and simulation tools. This is not done in this work. In the work of Larsen (2014) the concentration data values were too uncertain to make the comparisons significant because the uncertainty in the measured liquid concentrations were higher than the simulation uncertainties.

It has been claimed (Zhang et al., 2009) that rate-based models are superior to equilibrium based models because the rate-based models are capable of describing more detailed mechanisms. It is however factors and parameters in the rate-based models which are not known, especially specifications for fluid flow, heat transfer and mass transfer conditions in structured packings. A rate-based model has several parameters which can be used to fit the model to performance data. The most reasonable parameter to vary to fit capture efficiency is the interfacial area factor. It could also be possible to divide the column in sections with different interfacial area factors. The ability to predict performance at other conditions is however a difficult challenge. One assumption for the rate-based model is that there is ideal flow in axial direction and perfect mixing in radial direction. The large deviations in temperature measurements at different locations for each meter of packing (location A, B, C and D) indicate that this assumption is optimistic.

It is an important question whether the simulation tools with fitted parameters are able to give a predictable simulation at different conditions. Sætre (2016) tried to predict the capture rate and the temperature profiles for scenarios with 40 wt-% MEA (Desvignes, 2015) with the 3 simulation tools used in this work (which is fitted for 30 wt-% scenarios). The predictions were rather poor.

In this work, only performance data with 24 meter of packing was used. With such a high packing height, the capture rate becomes very high, and will approach equilibrium. This is indicated by the Murphree efficiencies which give the best temperature fit for scenario H14 and Goal1 were almost zero for the lowest stages. There are also available performance data for lower packing heights, e.g. 18 meter (Hamborg, 2014; Gjernes, 2017). When comparing performance data with simulation tools, comparisons using lower packing height would probably give interesting results.

6 Conclusions

It is possible to fit a rate-based model by adjusting the interfacial area factor, and to fit an equilibrium model by adjusting the Murphree efficiency for each stage. Equilibrium based models are less fundamental than rate-based models, but for the conditions in this study,

the rate-based models still lack accurate input parameters like the interfacial area. The results from this study show that equilibrium and rate-based models perform equally well in both fitting performance data and in predicting performance at changed conditions. The models with fitted parameters will give a predictable simulation only when there are small changes in the process parameters.

References

- D. M. Austgen, G. T. Rochelle, X. Peng and C. Chen. Model of Vapor-Liquid Equilibria for Aqueous Acid Gas-Alkanolamine Systems Using the Electrolyte-NRTL Equation. *Industrial & Engineering Chemistry Research*, 28:1060-73, 1989.
- T. H. Chilton and A. P. Colburn. Distillation and absorption in packed columns - A convenient design and correlation method. *Industrial and Engineering Chemistry*, 27:255-260, 1935.
- C. Desvignes. *Simulation of post-combustion CO₂ capture process with amines at CO₂ Technology Centre Mongstad*. Master Thesis, CPE Lyon, France, 2015.
- E. Gjernes, S. Pedersen, T. Cents, G. Watson, B. F. Fostås, M. I. Shah, G. Lombardo, C. Desvignes, N. E. Flø, A. K. Morken, T. de Cazanove, L. Faramarzi and E. S. Hamborg. Results from 30 wt% MEA performance testing at the CO₂ Technology Centre Mongstad. *Energy Procedia*, 114:1146-1157, 2017.
- E. S. Hamborg, V. Smith, T. Cents, N. Brigman, O. Falk-Pedersen, T. de Cazanove, M. Chagnal, J. K. Feste, Ø. Ullestad, H. Ulvatn, O. Gorset, I. Askestad, L. K. Gram, B. F. Fostås, M. I. Shah, A. Maxson and D. Thimsen. Results from MEA testing at the CO₂ Technology Centre Mongstad. Part II: Verification of baseline results. *Energy Procedia*, 63:5994-6011, 2014.
- R. L. Kent, B. Eisenberg. Better data for Amine Treating. *Hydrocarbon Processing*, 55(2):87-90, 1976.
- I. M. S. Larsen. *Simulation and validation of CO₂ mass transfer processes in aqueous MEA solutions with Aspen plus at TCM*. Master Thesis, Telemark University College, Norway, 2014.
- Y. Li, A. E. Mather. Correlation and Prediction of the Solubility of Carbon Dioxide in a Mixed Alkanol Solution. *Industrial & Engineering Chemistry Research*, 33:2006-2015, 1994.
- X. Luo, J. N. Knudsen, D. de Montigny, T. Sanpasertparnich, R. Idem, D. Gelowitz et al. Comparison and validation of simulation codes against sixteen sets of data from four different pilot plants. *Energy Procedia*, 1:1249-56, 2009.
- J. A. Rocha, J. L. Bravo, J. R. Fair. Mass Transfer in Gauze Packings, *Hydrocarbon Processing*, 64(1):91, 1985.
- J. A. Rocha, J. L. Bravo, J. R. Fair. Distillation Columns Containing Structured Packings: A Comprehensive Model for Their Performance. 1. Hydraulic Models. *Industrial and Engineering Chemistry Research*, 32:641-651, 1993.
- K. A. Sætre. *Evaluation of process simulation tools at TCM*. Master Thesis. University College of Southeast Norway, 2016.
- D. Thimsen, A. Maxson, V. Smith, T. Cents, O. Falk-Pedersen, O. Gorset and E. S. Hamborg. Results from MEA testing at the CO₂ Technology Centre Mongstad. Part I: Post-Combustion CO₂ capture testing methodology. *Energy Procedia*, 63:5938-5958, 2014.
- Y. Zhang, H. Chen, C. Chen, J. Plaza, R. Dugas, G. T. Rochelle. Rate-based Process Modelling Study of CO₂ Capture with Aqueous Monoethanolamine Solution. *Industrial & Engineering Chemistry Research*, 48:9233-46, 2009.
- Y. Zhu. *Simulation of CO₂ capture at Mongstad using Aspen HYSYS*. Master Thesis. Telemark University College, Norway, 2015.
- L. E. Øi. Comparison of Aspen HYSYS and Aspen Plus simulation of CO₂ absorption into MEA from atmospheric gas. *Energy Procedia*, 23: 360-369, 2012.

Occupancy and Daily Activity Event Modelling in Smart Homes for Older Adults with Mild Cognitive Impairment or Dementia

Flávia Dias Casagrande Evi Zouganeli

Department of Electronic Engineering, OsloMet – Oslo Metropolitan University, Norway.

{flavia.casagrande, evi.zouganeli}@oslomet.no

Abstract

In this paper we present event anticipation and prediction of sensor data in a smart home environment with a limited number of sensors. Data is collected from a real home with one resident. We apply two state-of-the-art Markov-based prediction algorithms – Active LeZi and SPEED – and analyse their performance with respect to a number of parameters, including the size of the training and testing set, the size of the prediction window, and the number of sensors. The model is built based on a training dataset and subsequently tested on a separate test dataset. An accuracy of 75% is achieved when using SPEED while 53% is achieved when using Active LeZi.

Keywords: smart home, prediction models, sensor data, occupancy modelling, event modelling

1 Introduction

We present results from the Assisted Living project, an interdisciplinary project that aims to develop assisted living technology (ALT) to support older adults with mild cognitive impairment or dementia (MCI/D) live a safe and independent life at home. The project is carried out by experts in the field of nursing and occupational therapy, ethics, and technology (Zouganeli et al., 2017). MCI and dementia involve cognitive decline, which can affect attention, concentration, memory, comprehension, reasoning, and problem solving. Smart homes can potentially include a number of intelligent functions that can provide valuable support to older adults with MCI/D, such as prompting support e.g. in order to assist or encourage, diagnosis support tools, as well as prediction, anticipation and prevention of hazardous situations. Activity recognition and prediction is a prerequisite and a necessary tool for achieving the majority of these.

We present our first results on prediction of binary sensor data in a smart home environment. Several algorithms have been reported in the literature for this purpose. However, to the extent of our knowledge, such prediction algorithms have not yet been tested in a real home, nor have they been proven to be accurate enough to be implemented in real homes. In addition, there is no comprehensive study comparing the different available algorithms or providing guidelines as to which application areas they are best suited for. In this paper we apply two algorithms on

data from a real home, compare their performance, and shed some light regarding their application areas.

2 Related Work

Data prediction algorithms have been extensively researched on in the literature (Wu et al., 2017). Event or activity prediction can for example lead to an improved operation of automation functions (e.g. turn on the heater sufficient time prior to the person arriving at home); facilitate useful prompting systems (e.g. prompt the resident in case the predicted next activity is not performed) (Holder and Cook, 2013); or detect changes/ anomalies in certain behaviour patterns (e.g. movement, everyday habits, etc.) and hence assist to indicate the onset or the progress of a condition (Riboni et al., 2016). The Active LeZi (ALZ) algorithm has been extensively applied for prediction on sequential data (Gopalratnam and Cook, 2007). The algorithm was tested on the Mavlab testbed dataset and was shown to achieve a 47% accuracy. Some of the ideas of ALZ have been used in the implementation of a new algorithm, the sequence prediction via enhanced episode discovery (SPEED)(Alam et al., 2012). SPEED was tested on the same dataset as ALZ and achieved an accuracy of 88.3% when the same dataset was used both for training and for testing. These algorithms are based on Markov models, where at any given point in time the next state depends solely on the previous one (Rabiner and Juang, 1986). Hence, the most probable next event can be estimated based on the current state.

Besides probabilistic algorithms, neural networks have also been used for event prediction. A root square mean error (RMSE) of 0.05 using Echo State Network (ESN) and Non-linear Autoregressive Network (NARX) was reported by using a number of input/output configurations (Lotfi et al., 2012; Mahmoud et al., 2013). Other relevant research includes prediction of the time when a certain activity will happen using decision trees (Minor and Cook, 2016) or time series (Moutacalli et al., 2015). Prediction of the next activity as well as the time, location, and day it would occur has also been reported (Nazerfard and Cook, 2015).

In this paper, we use the Active LeZi and SPEED algorithms for the prediction of the next sensor to be activated/ deactivated in an event sequence obtained from a real home with one resident.

3 Field Trial

Our field trial involves ten independent one-bedroom apartments within a community care facility for people over 65 years old. Each apartment comprises a bedroom, a living room, open kitchen area, a bathroom, and an entrance hall (Figure 1).

The purpose of the trial and the sensor system to be deployed have been decided upon in close collaboration with the residents (Zouganeli et al., 2017). A minimal number of binary sensors has been deployed in our trial in order to both minimize surveillance of the residents in their private homes, and comply with the technical and economic constraints imposed by the research project this work is a part of. The set of sensors has subsequently been chosen so that it can enable the realization of useful functions for older adults with MCI/D as these were indicated after dialogue cafes with the users (Zouganeli et al., 2017). We chose to include sensors that indicate occupancy patterns (movement around the apartment) and some daily activities like eating/ drinking, dressing, sleeping, and leisure activities (reading, watching TV, listening to radio). Hence, the system comprises motion, magnetic, and power sensors. A motion sensor (Pyroelectric/Passive Infrared – PIR) detects motion through the change of the infrared radiation in its field of view. It sends a message ‘1’ when a motion is detected. Magnetic sensors indicate whether doors/ windows/ drawers are open or closed, by sending messages ‘1’ and ‘0’, respectively. Power sensors measure the electricity usage of a certain appliance, and can therefore indicate whether it is turned on or off, and send messages ‘1’ and ‘0’ respectively. Figure 1 shows a schematic of the apartment. There are 15 sensors installed in total: seven motion sensors (one in each area of the apartment and two over and by the bed to indicate whether the person is in bed); four magnetic sensors (back and entrance doors, wardrobe, and cutlery drawer); and four power sensors on appliances (nightstand lamp, coffee machine, TV, and living room/ reading lamp).

The sensors are connected wirelessly through Z-Wave and xComfort protocols to a Raspberry Pi 3, which receives the data and transfers it for storage in a secure server (TSD). The data comprises timestamp (date and time with precision up to seconds), sensor ID, and sensor message (binary) – see example in Table 1.

Table 1. Binary sensors data.

Timestamp	Sensor ID	Sensor message
01.09.2017 17:58:05	4	1
01.09.2017 17:58:40	6	1
01.09.2017 17:59:02	10	1
01.09.2017 17:59:05	10	0

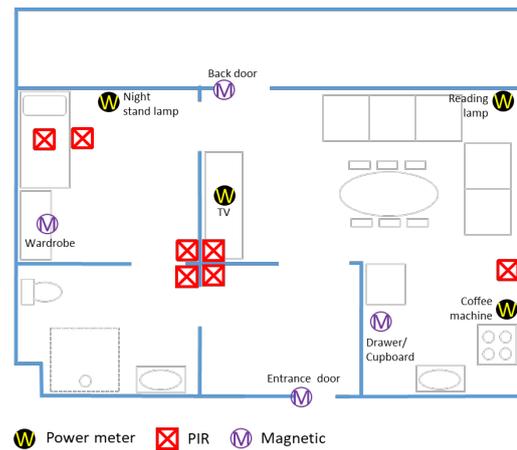


Figure 1. Sensors system in the field trial apartment.

4 Prediction Algorithms

Both ALZ and SPEED translate the data acquired from the sensors into a sequence of letters and identify patterns that occur frequently, so-called contexts. The contexts and their frequency of occurrence are used to generate a tree, which is then used to calculate the next most probable event to occur. This last step is performed by the Prediction Partial Matching algorithm (PPM) (Cleary and Witten, 1984; Cleary et al., 1997). Table 2 presents a possible scenario in a smart home of performed actions by the resident and the corresponding sensors being triggered. For ALZ and SPEED, each sensor is assigned with a letter, as shown in Table 3.

Table 2. Actions scenario.

Action performed	Activated sensor
Wake up	PIR bedroom (on)
Go to living room	PIR living room (on)
Turn on TV	Power TV (on)
Go to kitchen	PIR kitchen (on)
Turn on coffee machine	Power coffee machine (on)
Go to living room and watch TV while coffee is being made	PIR living room (on)
Go to kitchen	PIR kitchen (on)
Turn off coffee machine	Power coffee machine (off)
Go to living room	PIR living room (on)

4.1 Active LeZi

ALZ is a sequence prediction algorithm based on a text compression algorithm (Gopalratnam and Cook, 2007). The input in ALZ consists of a sequence of lower case letters, where each letter represents event from one sensor. For example, the sequence corresponding to the scenario described in Table 2 would be "abcdebdeb". ALZ uses the

Table 3. Assignment of letters to sensors.

Sensor	Letter
PIR bedroom	a/A
PIR living room	b/B
Power TV	c/C
PIR kitchen	d/D
Power coffee machine	e/E

idea from the LZ78 text compression algorithm to generate patterns that occur in a sequence and create a tree with these and their frequencies (Ziv and Lempel, 1978).

A given sequence x_1, x_2, \dots, x_i is parsed into n_i subsequences w_1, w_2, \dots, w_{n_i} such that for all $j > 0$ the prefix of the subsequence w_j is equal to some w_i for $1 < i < j$. For example, if we have the sequence "abcdebdeb", the dictionary would have the following words "a", "b", "c", "d", "e", "bd", "eb". These words correspond to contexts derived from the sequence. ALZ generates more contexts from their suffixes, if possible. For example, "bd" would also generate "d", and "eb" would generate "b". This accounts for contexts that were not perceived by the LZ78 algorithm and that are possibilities in a smart home environment. This increases the convergence rate of the model (Gopalratnam and Cook, 2007).

When the sequence is parsed completely and the contexts are derived from it, their frequency of occurrence is counted. An order- $k-1$ Markov tree is then constructed based on the contexts and their frequencies, where k corresponds to the longest word found in a training sequence. Then PPM is used to calculate the next most probable event. The generated tree for the example scenario with sequence "abcdebdeb" is shown in Figure 2.

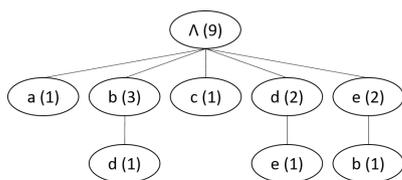


Figure 2. Tree generated by ALZ from sequence "abcdebdeb".

4.2 SPEED

SPEED is a sequence prediction algorithm that is based on the occurrence of frequent patterns in home environments (Alam et al., 2012). It assumes that human activity is predictable since usually certain patterns are repeated daily. SPEED defines an episode as the sequence between an initial and ending point of an activity. For example, the moment a coffee machine is turned "on" is the initial point of a coffee making episode, which lasts until the coffee machine is turned "off". An "off" event cannot happen unless an "on" event has happened before. Therefore "off" events always happen after an "on" event of the same activity (or sensor), and vice-versa.

The data received from the sensors in the smart home are represented as a sequence of letters, where upper case letters represent a sensor's "on" event and lower case letters represent a sensor's "off" event. For the example scenario presented in Table 1, the sequence would be "AaBCbDEdBbDedB".

The main idea of the SPEED algorithm is to extract episodes from a sequence of data and derive contexts from them. These contexts are used to generate a decision tree that keeps track of the learned episodes and their frequencies. The height of the tree is the length of the longest episode found in the sequence, defined as the maximum episode length. For every event in a sequence, the algorithm searches for its opposite event in the window and if it exists, an episode was found. In the previous sequence, the first episode found is "Aa", the contexts generated from it would be "A", "a" and "Aa". We keep track of these and count their occurrences to generate an order- $k-1$ Markov model, where k is the maximum episode length. A tree for the example sequence is presented in Figure 3. Finally, the PPM algorithm is used for prediction.

4.3 PPM Algorithm

PPM calculates the probability distribution of each possible event based on a given sequence by taking into consideration the different order Markov models with different weights (Cleary and Witten, 1984; Cleary et al., 1997). The weights are given by the escape probability, which allows the model to go from a higher-order to a lower one. The advantage of PPM is that it assigns a greater weight to the probability calculated in higher-order models if the symbol being predicted is actually found in the tree (Gopalratnam and Cook, 2007). The predicted symbol is the one with the highest probability.

ALZ and SPEED use slightly different strategies of PPM. ALZ uses the exclusion strategy, which means the prediction is performed with the suffixes of the given sequence, except the sequence itself. Therefore, in the case of the sequence "eb", the contexts used to calculate the probability of each letter being the next would be "e" and the null context. Suppose we want to calculate the probability of having an "e" after "eb" using ALZ, based on the tree in Figure 2. The probability would be given by Equation 1: in an order-2 model, the probability of having an "e" after an "e" is $0/2$ and we escape to the order-1 with $1/2$ probability. In order 1, the probability of having an "e" after a null context is $2/9$.

In the case of SPEED, the contexts used for calculating probabilities after a certain sequence would be all the suffixes, including the sequence itself. Suppose we have the sequence "dB". We would use contexts "dB", "d" and the null context. The probability of having a "b" after this sequence based on the tree in Figure 3, would be given by Equation 2: we start in order 2 model, where the probability of having a "b" after "dB" is $1/2$ and escape to the lower order with probability $1/2$. In order-1, the probability of having a "b" after "d" is $0/4$ and we escape to the

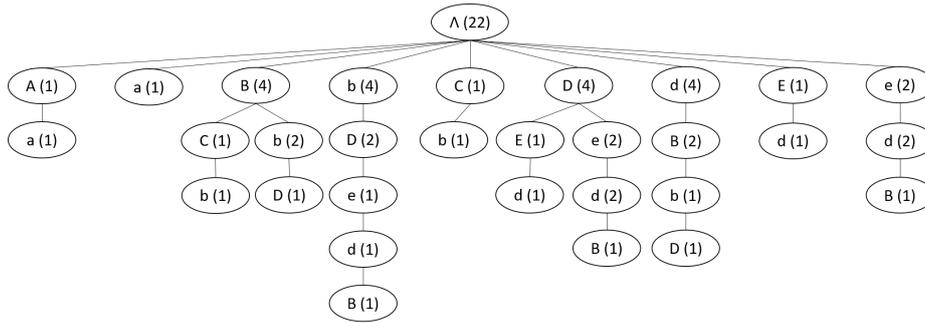


Figure 3. Tree generated by ALZ from sequence "AaBCbDEdBbDedB".

lower order with probability $2/4$. Finally, in the lowest order, the probability of "b" after a null context is $4/22$.

$$p(e, eb) = \frac{0}{2} + \frac{1}{2} \left(\frac{2}{9} \right) = 0.111 \quad (1)$$

$$p(b, dB) = \frac{1}{2} + \frac{1}{2} \left(\frac{0}{4} + \frac{2}{4} \left(\frac{4}{22} \right) \right) = 0.545 \quad (2)$$

5 Results and Discussions

Data has been collected from the apartment described in section 3 over a period of two weeks. In total, there are 6182 raw sensor events. The data was translated to the format required by ALZ and SPEED, which resulted in 4629 and 9062 events respectively. In the SPEED sequence, we performed noise removal such that "on" events only come after "off" events of the same sensor, or vice-versa. We ended up with 9044 events. In the SPEED algorithm, the next event is predicted based on the last sequence of size equal to the maximum episode length (Alam et al., 2012). Firstly, we reproduced the results using the same dataset and method as reported in that paper (Alam et al., 2012). Subsequently we modified the testing procedure somewhat by calculating the optimal number of last events to base the prediction on, i.e. the window that leads to the maximum overall prediction accuracy, which we refer to as the optimal window. Window sizes up to the maximum episode length are considered.

In order to be able to compare our results to the performance of the same algorithms when they are used on the data from the Mavlab testbed (Alam et al., 2012), we firstly compute the prediction accuracy that is attained when using the same dataset for both the training and the testing, as performed in their work. Figure 4 presents the results when training and testing using the same sequence of n events, where $n = \{100, 200, \dots, 2000\}$.

In this test, SPEED had an optimal window of five and ALZ of six events, when the training and testing sets consisted of 2000 events. An accuracy of 82% and 73% is achieved by SPEED and ALZ respectively. Clearly training and testing with the same dataset leads to overfitting. As a result, the apparent accuracy may keep increasing

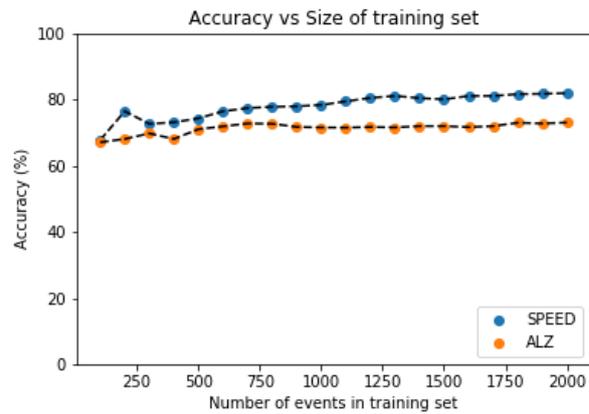


Figure 4. SPEED and ALZ prediction accuracy vs. the size of training set.

when we increase the dataset size. For a dataset size equal to 700, as in the Mavlab dataset, SPEED and ALZ attain 77% and 73% prediction accuracy respectively when used on our data.

In order to evaluate the actual prediction accuracy of the algorithms, our data is split into a training set, a validation set, and a testing set. The training set is used to construct the tree, the validation set is used to find the optimal window, and the testing set is used to calculate the prediction accuracy.

We first analyse the importance of choosing the optimal window to predict events from. Figure 5 shows the prediction accuracy for different sizes of the window, and for four different sizes of the training dataset when using the SPEED algorithm. Similarly, Figure 6 shows the effect of the window size in the case of the ALZ algorithm. The validation set comprised 1000 events in all cases.

We notice that smaller window sizes (1-4 events) provide better accuracy, for both algorithms. The accuracy deteriorates very quickly with increasing window size. This behaviour is as expected in particular for a setup with a small number of sensors, since long sequences of events are not bound to be repeated frequently. In the case of SPEED, for example, bathroom activities would be maximum two-events long ("on-off" bathroom motion sensor). These graphs are in addition a manifestation of the fact

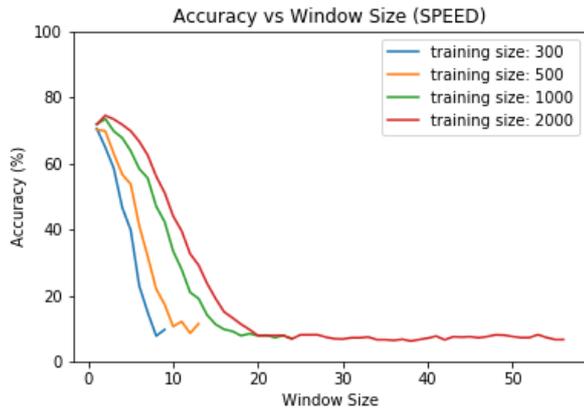


Figure 5. SPEED prediction accuracy vs. the window length, for several training set sizes.

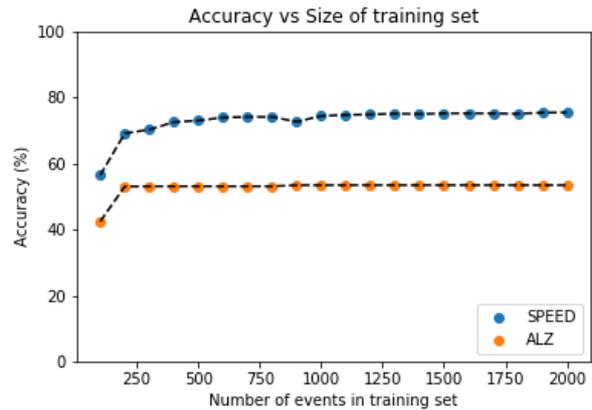


Figure 7. SPEED and ALZ prediction accuracy vs. the size of training set.



Figure 6. ALZ prediction accuracy vs. the window length, for several training set sizes.

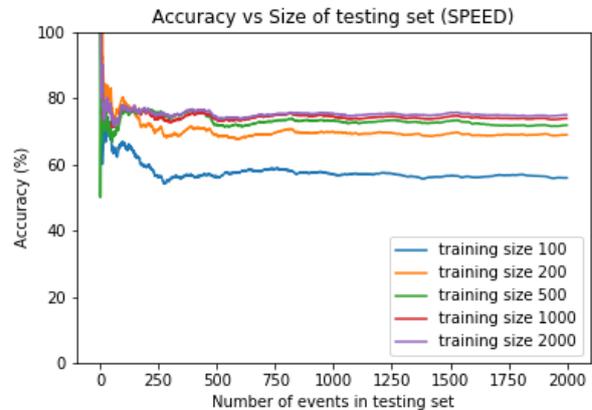


Figure 8. SPEED prediction accuracy vs. the size of testing set.

that SPEED creates a tree of much longer height than ALZ does. The tree height corresponds to the longest episode in SPEED, whereas in ALZ it corresponds to the longest context. This is evident from Figures 3 and 2 where the respective trees are shown for the same example scenario.

Once the optimal window was calculated from a validation set of 1000 events, we computed the accuracy for different number of training events. We trained the algorithms with a number of events $i = \{100, 200, 300, \dots, 2000\}$. The prediction accuracy was computed based on a testing set of 1500 events. Figure 7 shows the results for both SPEED and ALZ.

SPEED achieved an accuracy of 75% and ALZ an accuracy of 53%, with optimal windows of two and one respectively. We observe that this maximum accuracy is achieved with SPEED for training sets larger than about 800 events, while ALZ reaches a maximum accuracy for a training set of 300 events or more. Hence, ALZ converges to its maximum accuracy faster than SPEED, however, it achieves a much poorer prediction accuracy than SPEED. Using a larger number of events for the training does not increase the accuracy significantly for neither of the algo-

rithms.

At this point, we can associate some of these results to the trees generated for both algorithms for the example scenario in sections 4.1 and 4.2. The height of the tree is significantly larger in SPEED for the same performed actions. It can also be noted from Figure 3 that SPEED collects a significantly higher number of contexts and frequencies, which may be the reason why SPEED leads to better accuracy.

In the following we examine the dependence of the prediction accuracy on the size of test dataset. Figure 8 shows the prediction accuracy attained by SPEED as a function of the size of the testing dataset for different sizes of the training dataset. Figure 9 shows the same results for ALZ.

In the case of SPEED, the prediction accuracy is quite variable for a test dataset of up to about 250 events due to the small number of predicted events. The maximum accuracy is achieved for test set sizes larger than about 500 events when the training is performed based on a set with 500, 1000 and 2000 events. This confirms that the algorithm is quite robust. ALZ shows similar behaviour and achieves its maximum prediction accuracy for test datasets larger than about 200 events.

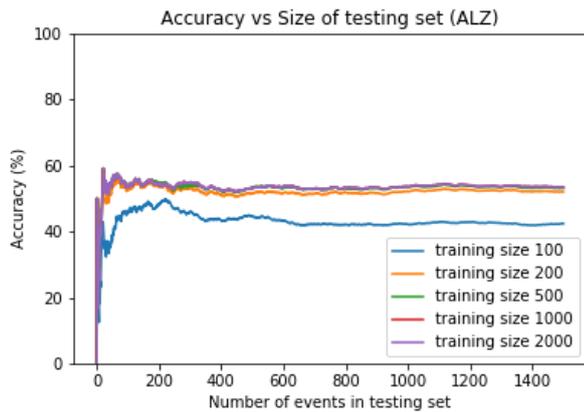


Figure 9. ALZ prediction accuracy vs. the size of testing set.

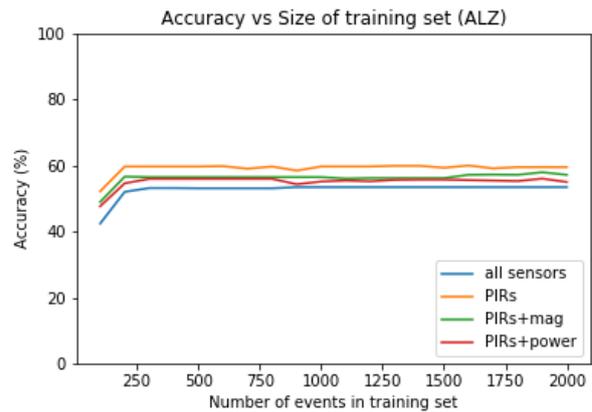


Figure 11. ALZ prediction accuracy vs. the size of training set for different sets of sensors.

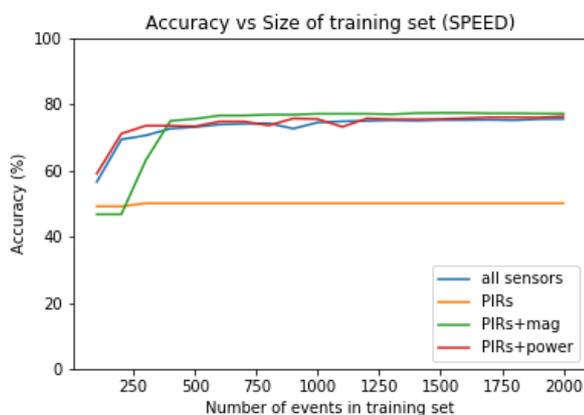


Figure 10. SPEED prediction accuracy vs. the size of training set for different sets of sensors.

A last test was performed to reveal the dependence of the prediction accuracy on the number and type of sensors. Four alternatives were investigated based on our current data: all sensors (15), only PIR sensors (7), only PIR and magnetic sensors (11), and only PIR and power sensors (11). The last two sets have the same number of sensors, however, magnetic and power sensors can affect accuracy differently. The results are shown in Figure 10 for SPEED and Figure 11 for ALZ.

Both algorithms show relatively good robustness with respect to the number of sensors. The accuracy is not significantly dependent upon the number of sensors in the dataset, in most of the cases. A clear exception is the case when only PIRs are used for prediction using SPEED. The prediction accuracy is very poor in this case. Note that in this case the longest episode will be two events. For example, if the resident would go from the bedroom to the living room and then to the kitchen, the resulting sequence would be "AaBbCc". There is no context connecting the living room to the bedroom, or the kitchen to the living room. Hence, while the "off" events are easily correctly predicted, the prediction of the next sensor to be activated will often be quite inaccurate in this case. Note that in

this case the tree created by SPEED will have a maximum length of two. On the other hand, ALZ is better suited to such cases where events are not highly interweaved. When SPEED is used the remaining sensor sets achieved a prediction accuracy that is similar to that achieved by the full set of sensors. The alternative where the power sensors are not included provides slightly better results indicating that events related to appliances are more difficult to predict.

In the case of ALZ, the best accuracy is achieved when fewer sensors are used. This is a result of the fact that the average probability of occurrence for each event increases when the number of possible events decreases. The prediction accuracy of events that involve magnetic sensors is relatively high as doors and drawers are often closed right after they have been opened, thus making this a relatively easy pattern to predict. On the other hand, power sensors can occur somewhat randomly with many other events happening in between, thus making the prediction of the associated events more inaccurate.

6 Conclusions and Future Work

Activity recognition and prediction in a smart home environment with binary sensors has received a lot of attention in recent years. Most of the reported work is carried out in testbeds and lab environments where users are often asked to execute pre-scripted activities. Such smart-home testbeds typically include a quite large number of sensors, e.g. the CASAS testbed utilized around 50 sensors (Gopalratnam and Cook, 2007).

In this paper we have presented preliminary results on event prediction based on data from a real home collected using just 15 binary sensors. We have used two prediction algorithms, ALZ and SPEED, to predict the next sensor event in a sequence. To the extent of our knowledge, this is the first time these algorithms are used on a dataset obtained from a real home. We compare the prediction accuracy of the two models and examine the dependence of their performance on a number of parameters – the size of the training dataset, the size of the testing dataset, and the

size of the window used for the prediction. We reached an accuracy of 75% with SPEED and 53% with ALZ when training with a dataset of 2000 events and testing on a separate dataset of 1500 events. Increasing the number of events in either the training or the testing dataset, did not improve the attained accuracy. In addition, we examined the dependence of the prediction accuracy on the number of sensors for both algorithms. Our results show that robust prediction accuracy can be attained by a relatively low number of sensors.

However, a much higher prediction accuracy is required before such algorithms are applicable to real homes. Future work will include the time component in order to improve the accuracy of our models as this has been indicated to lead to a considerable improvement (Marufuzzaman et al., 2015).

Acknowledgements

The authors wish to thank the residents and the housekeepers at the seniors' care unit Skøyen Omsorg+; Torhild Holthe and Dr. Anne Lund (OsloMet) for recruiting participants for the trial and communicating with the residents throughout; Dejan Krunic and Øyvind Width (Sensio AS) for installations of the sensors, and the rest of the participants of the Assisted Living Project for a fruitful interdisciplinary collaboration; Dr. Muhammad Raisul Alam for clarifications regarding the SPEED algorithm and the way it was applied in his work; Prof. Jim Tørrensen, University of Oslo, for his support. The project is financed by the Norwegian Research Council under the SAMANSVAR programme (247620/O70).

References

- Muhammad Raisul Alam, M. B.I. Reaz, and M. A. Mohd Ali. SPEED: An inhabitant activity prediction algorithm for smart homes. *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, 42(4):985–990, 2012. ISSN 10834427. doi:10.1109/TSMCA.2011.2173568.
- John G. Cleary and Ian H. Witten. Data Compression Using Adaptive Coding and Partial String Matching. *IEEE Transactions on Communications*, 32(4):396–402, 1984. ISSN 00906778. doi:10.1109/TCOM.1984.1096090.
- John G. Cleary, W. J. Teahan, and Ian H. Witten. Unbounded length contexts for PPM. In *Proceedings DCC '95 Data Compression Conference*, pages 52–61, 1997. ISBN 0-8186-7012-6. doi:10.1109/DCC.1995.515495.
- Karthik Gopalratnam and Diane J. Cook. Online Sequential Prediction via Incremental Parsing: The Active LeZi Algorithm. *IEEE Intelligent Systems*, 22(1), 2007.
- Lawrence B. Holder and Diane J. Cook. Automated activity-aware prompting for activity initiation. *Gerontechnology*, 11(4):534–544, 2013. ISSN 15691101. doi:10.4017/gt.2013.11.4.005.00.
- Ahmad Lotfi, Caroline Langensiepen, Sawsan M. Mahmoud, and M. J. Akhlaghinia. Smart homes for the elderly dementia sufferers: Identification and prediction of abnormal behaviour. *Journal of Ambient Intelligence and Humanized Computing*, 3(3):205–218, 2012. ISSN 18685137. doi:10.1007/s12652-010-0043-x.
- Sawsan Mahmoud, Ahmad Lotfi, and Caroline Langensiepen. Behavioural pattern identification and prediction in intelligent environments. *Applied Soft Computing Journal*, 13(4):1813–1822, 2013. ISSN 15684946. doi:10.1016/j.asoc.2012.12.012.
- M Marufuzzaman, M. B. I. Reaz, M. A. M. Ali, and L. F. Rahman. A Time Series Based Sequence Prediction Algorithm to Detect Activities of Daily Living in Smart Home. pages 262–270, 2015. ISSN 0026-1270. doi:10.3414/ME14-01-0061.
- Bryan Minor and Diane J. Cook. Forecasting occurrences of activities. *Pervasive and Mobile Computing*, 2016.
- Mohamed Tarik Moutacalli, Abdenour Bouzouane, and Bruno Bouchard. Sensors activation time predictions in smart home. *Proceedings of the 8th ACM International Conference on Pervasive Technologies Related to Assistive Environments*, page 37, 2015. doi:10.1145/2769493.2769530.
- Ehsan Nazerfard and Diane J. Cook. CRAFT: An Activity Prediction Model based on Bayesian Networks. 33(4):395–401, 2015. ISSN 1527-5418. doi:10.1038/nbt.3121.ChIP-nexus.
- L. R. Rabiner and B. H. Juang. An Introduction to Hidden Markov Models. *IEEE ASSP Magazine*, 3(1):4–16, 1986. ISSN 07407467. doi:10.1109/MASSP.1986.1165342.
- Daniele Riboni, Claudio Bettini, Gabriele Civitarese, Zafar Haider Janjua, and Rim Helaoui. SmartFABER: Recognizing fine-grained abnormal behaviors for early detection of mild cognitive impairment. *Artificial Intelligence in Medicine*, 67:57–74, 2016. ISSN 18732860. doi:10.1016/j.artmed.2015.12.001.
- TSD. Services for sensitive data (TSD). <https://www.uio.no/english/services/it/research/sensitive-data/index.html>. Accessed: 2018-04-30.
- Shaoen Wu, Jacob B. Rendall, Matthew J. Smith, Shangyu Zhu, Junhong Xu, Honggang Wang, Qing Yang, and Pinle Qin. Survey on Prediction Algorithms in Smart Homes. *IEEE Internet of Things Journal*, 4(3):636–644, 2017. ISSN 23274662. doi:10.1109/JIOT.2017.2668061.
- Jacob Ziv and Abraham Lempel. Compression of Individual Sequences via Variable-Rate Coding. *IEEE Transactions on Information Theory*, 24(5):530–536, sep 1978. ISSN 15579654. doi:10.1109/TIT.1978.1055934.
- Evi Zouganeli, Flavia D. Casagrande, Torhild Holthe, Anne Lund, Liv Halvorsrud, Dag Karterud, Adele Flakke-Johannessen, Hilde Lovett, S.K. Mørk, J. Strøm-Gundersen, Erik Thorstensen, R. Norvoll, R.T. Meulen, M.-R. Kennedy, R.J. Owen, M. Ladikas, and Ellen-Marie Forsberg. Responsible development of self-learning assisted living technology for older adults with mild cognitive impairment or dementia. *ICT4AWE 2017 - Proceedings of the 3rd International Conference on Information and Communication Technologies for Ageing Well and e-Health*, (Ict4awe):204–209, 2017. doi:10.5220/0006367702040209.

Model-order selection of output-error models - BSM1 as case study

Christian Wallin^{1,2,*} Jesús Zambrano²

¹ABB AB, Power Generation, Västerås, Sweden.

²School of Business, Society and Engineering, Mälardalen University, Västerås, Sweden.

*Corresponding author e-mail: christian.wallin@se.abb.com

Abstract

Output-Error (OE) System Identification is used to estimate the nonlinear behavior of an activated sludge process (ASP) in a Wastewater Treatment Plant (WWTP). The aim is to identify dynamic models to reproduce the effect of different plant dynamics. How the dissolved oxygen concentration of the aerobic tank affect the effluent ammonia concentration and how the internal recirculation affect the nitrate concentration of the anoxic tank is studied. The best fit of the model is estimated by varying the model order through a trial-and-error approach. Three different scenarios are investigated: one Single-Input-Single-Output (SISO) and two Multiple-Input-Multiple-Output (MIMO) structures. In the SISO scenario only the oxygen to the effluent ammonia dynamics is investigated. Then for both the MIMO scenarios the internal recirculation to nitrate concentration dynamics in the anoxic tank is included and in the last scenario the influent flow rate is also included. The approach is evaluated using the Benchmark Simulation Model no.1 (BSM1).

*Keywords: Benchmark Simulation Model No. 1, Model Predictive Control, Output Error Model, System Identification.*strict

1 Introduction

The activated sludge process (ASP) in a wastewater treatment plant (WWTP) are large non-linear systems subject to perturbations and uncertainty in the influent composition. However, these process should operate continuously and following strict effluent regulations.

From the point of view of control, a system identification of the process is important, mainly because it will improve the control performance of the process, which is typically formed by PI controllers. Another reason is that the system identification can be used to carry out stability analysis of the closed-loop system (Chistiakova et al., 2017).

The system identification involves defining a model structure, mainly a black-box model, where the model parameters are adjusted to fit the data and do not reflect physical consideration Ljung (1999). These models involve a model-order definition with adjustable parameters. The definition of such a model order is still empirical. Typically, a certain model order is assumed, see for example Chistiakova et al. (2017) where an Output-Error

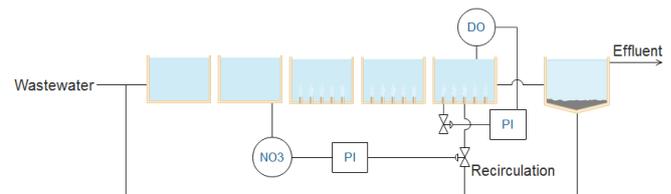


Figure 1. 1 basic layout where the influent water first passes 2 anoxic tanks followed by 3 aerated tanks and then passes a settler before being released. Two control loops are shown: One measures the nitrate concentration of the 2nd anoxic tank to control the internal recirculation and the other measures the oxygen concentration in the last aerated tank to control the air-flow rate.

(OE) model and nonlinear models were estimated, Ekman (2008) where a bilinear model is estimated and Vrečko et al. (2004) where a state-space model is estimated, those cases used an ASP as case study.

The aim of this work is to present a way to get an appropriate model-order in a system identification of the process. For the system identification, an OE model is used. An ASP was used case study using data from the Benchmark Simulation Model no. 1 (BSM1) (Alex et al., 2008).

2 The Benchmark Simulation Model

2.1 Description

Data from BSM1 is used for system identification, see the model layout in Figure 1. The BSM1 is a platform that defines a conventional ASP, and includes a simulation model, plant layout, default control systems, performance criteria and test procedures. The plant layout of the BSM1 is formed by a five-compartments ASP, consisting of two anoxic tanks followed by three aerobic tanks and a settler.

The process model is based on the Activated Sludge Model No. 1 (ASM1) (Henze et al., 1987) for the ASP compartments and the Takács model (Takács et al., 1991) for the secondary settler. The BSM1 kinetics and stoichiometric parameter values were kept as default.

The BSM1 includes a constant and a dynamic influent. The constant influent (150 days) was used for the system identification of the SISO case and the MIMO case without influent flow rate, whereas the dynamic influent (14 days) was used for the case MIMO* where influent flow rate is also considered.

The BSM1 includes two default control loops. One control loop is formed by a PI controller that controls the dissolved oxygen (DO) concentration in the last aerobic tank via air flow rate regulation which is shown in Figure 1 as a valve controlling the incoming air to the blowers. The other control loop is also formed by a PI controller which deals with the control of the nitrate concentration in the last anoxic tank via the internal recirculation flow rate. The sampling time was 15 minutes. The dry weather scenario was used as dynamic influent.

2.2 Case studies

Three different structures were studied for system identification. One structure (referred as SISO case) considers only the DO set-point in the last aerated tank ($S_{O_2}^{sp}$) as input signal, whereas the effluent ammonia concentration ($S_{NH,eff}$) was used as output signal. The DO was modified from the default constant value of 2 mg/l to a range between 0.8 and 2.4 mg/l with a minimum step-interval of the change of set-point set to 100 time-steps.

The other structure (referred as MIMO case) includes the input/output signals of SISO with the addition of the effect of the internal recirculation (Q_{int}) in the nitrate concentration of the second anoxic tank ($S_{NO,2}$).

The last structure (referred as MIMO* case) is the MIMO case including the effect of the influent flow rate (Q_{in}). Here the minimum step-interval is also changed to 10 time-steps because of the lower total simulation time.

3 Method

3.1 System Identification

The structure of the model includes a linear OE model (Ljung, 1999), given as follows

$$\hat{y}(t) = \frac{B(q^{-1})}{F(q^{-1})} u(t - nk) + e(t), \quad (1)$$

where $\hat{y}(t)$ is the output signal of the OE model, $u(t)$ is the input signal, $e(t)$ is the error, nk is a time delay. B and F have the form

$$B(q^{-1}) = b_1 + b_2 q^{-1} + \dots + b_{nb} q^{-nb+1}, \quad (2)$$

$$F(q^{-1}) = 1 + f_1 q^{-1} + \dots + f_{nf} q^{-nf}, \quad (3)$$

which are polynomials in the backward shift operator q^{-1} (i.e. $q^{-i}x(k) = x(k-i)$), where $b_i (i = 1, \dots, nb)$ and $f_i (i = 1, \dots, nf)$ are unknown parameters, nb and nf are the orders of the OE model.

Part of the system identification involves generating input signals and measurable outputs. The input signals were generated by multiplying a pseudo-random binary sequence (PRBS) with a uniformly distributed random factor. This gives a sequence where each constant value is multiplied with a uniformly distributed amplitude (Wigren, 2003). Figure 2 shows an example of data used for system identification in the MIMO case.

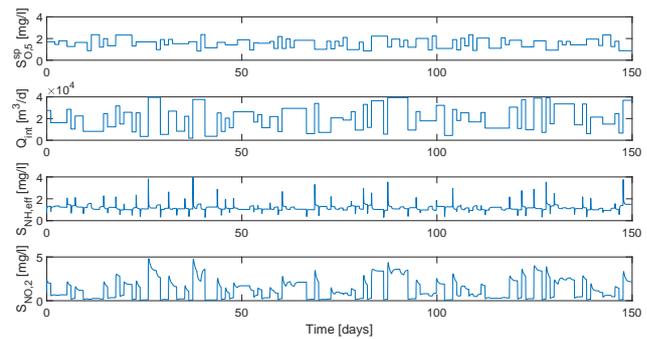


Figure 2. Input signals ($S_{O_2}^{sp}, Q_{int}$) and output signals ($S_{NH,eff}, S_{NO,2}$) used for system identification in MIMO case.

Multiple simulations are done and for each simulation multiple linear models are estimated by varying the order of the B and F polynomials (cf. (2)-(3)).

3.2 Model selection

The Akaike Information Criterion (AIC) (Akaike, 1974) is a way to compare and obtain a good model order. The AIC takes into account the number of parameters and the size of the data set in the following way

$$AIC = \log \left(\frac{1}{N} \sum_{i=1}^N \varepsilon_i^2 \right) + \frac{2n_p}{N}, \quad (4)$$

where ε is the error between the estimated model and the BSM1 model, n_p is the number of estimated parameters, and N is the size of the estimation data set.

In this work, the small sample-size corrected Akaike's Information Criterion ($AICc$) has been used, since it was especially developed for regression and autoregressive time series models (Hurvich and Tsai, 1989). The $AICc$ is defined as follows

$$AICc = AIC + \frac{2n_p(n_p + 1)}{N - n_p - 1}. \quad (5)$$

See in (5) that the number of estimated parameters is more relevant than the size of the data set. A model with the lowest $AICc$ is expected to be the model that better describes the data with the minimal number of parameters. The same principle applies to the AIC value.

3.3 Model validation

Since each set of input sequence gives a particular best model, the overall best model is obtained by checking how well the particular best models fit the other set of data, i.e. the model are cross validated with the different validation data. This cross validation is quantified using the Fit measure, which is the normalized root mean square error fitness value, defined as

$$Fit = 100 \times \left(1 - \frac{\|y - \hat{y}\|}{\|y - \bar{y}\|} \right), \quad (6)$$

where y is the output of the validation data, \hat{y} is the output given by the estimated model, and \bar{y} is the mean value of the validation data.

The full system identification and model validation process is shown in Figure 3. See that once one simulation is completed, the Input/Output data obtained is used to estimate an OE model in a loop where the orders of the OE model are varied up to a predefined maximum model order M . The best OE model is determined by the AIC_c value. This process is repeated for a new simulation. Finally, the best model from each simulation is cross-validated against the input and output data from each of the other simulations to determine which OE-model that has the best average *Fit* value against every simulation made.

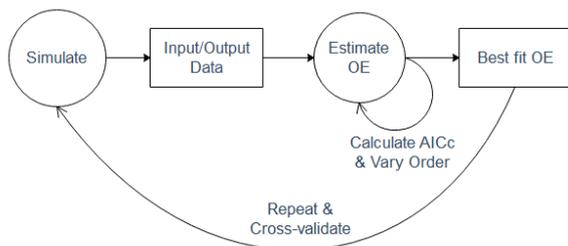


Figure 3. System identification process. A simulation is done and the data from this is collected and used to estimate a model. During model estimation different model orders are tried and each model is evaluated based on the AIC_c value. Multiple similar simulations are then done and the best model from each simulation is validated against input and output data from other simulations.

The BSM1 was simulated using the Matlab/Simulink[®] platform, version R2017a. Matlab was also used to estimate the coefficients of the OE-model running the `oe` command with default parameter settings.

4 Results

The best SISO case was estimated by running $i = 100$ cross-realizations of input-output data with model order $M = 10$. Figure 4(a) shows how good the input-output data fit for each of the different cross-fit realizations applied to the SISO case. In the boxplots, the median value is represented by a red line, the edges of the box are the 25th and 75th percentile, the end of the lines are the extreme points that the algorithm consider not to be outliers and outliers are plotted as a red + sign. The best model order obtained was an OE(6,10,1) from trial 93, which gave

an average fit of 71.4% with the following coefficients:

$$\begin{cases} B(q^{-1}) = -0.133q^{-1} + 0.429q^{-2} - 0.545q^{-3} \\ \quad + 0.371q^{-4} - 0.156q^{-5} + 0.034q^{-6}, \\ F(q^{-1}) = 1 - 3.556q^{-1} + 5.154q^{-2} - 4.172q^{-3} \\ \quad + 2.216q^{-4} - 0.686q^{-5} - 0.17q^{-6} \\ \quad + 0.601q^{-7} - 0.768q^{-8} + 0.523q^{-9} \\ \quad - 0.142q^{-10}. \end{cases} \quad (7)$$

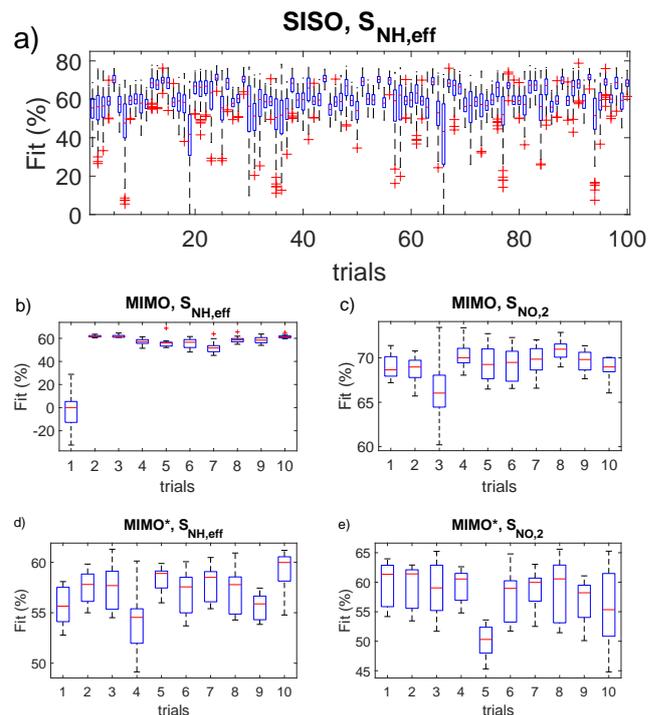


Figure 4. Boxplot of cross-validation for several trials in the SISO, MIMO and MIMO* cases. (a) SISO case, (b)-(c) Output $S_{NH,eff}$ and $S_{NO,2}$ of MIMO case, (d)-(e) Output $S_{NH,eff}$ and $S_{NO,2}$ of MIMO* case. Red line is the median value, the edges of the box are the 25th and 75th percentile, the end of the lines are the extreme points that the algorithm consider not to be outliers and outliers are plotted as a red '+' sign.

The same procedure was applied for the MIMO case. In this case, $i = 10$ and $M = 3$ which is lower than the SISO case due to the increasing processing time. Figure 4(b)-(c) show how good the input-output data fit for each of the different cross-fit realizations applied to the MIMO case. The best model order obtained was from trial 2, which gave a sum average fit of 65.2% and has the following form:

$$OE\left(\begin{bmatrix} 3 & 3 \\ 2 & 3 \end{bmatrix}, \begin{bmatrix} 3 & 3 \\ 1 & 3 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}\right),$$

with the coefficients for $\hat{y}_1(t)$ ($S_{NH,eff}$):

$$\begin{cases} B_1(q^{-1}) = -0.1197q^{-1} - 0.04353q^{-2} + 0.1633q^{-3}, \\ B_2(q^{-1}) = 2.767 \times 10^{-5}q^{-1} - 5.52 \times 10^{-5}q^{-2} \\ \quad + 2.754 \times 10^{-5}q^{-3}, \\ F_1(q^{-1}) = 1 - 0.9046q^{-1} - 0.2211q^{-2} + 0.1258q^{-3}, \\ F_2(q^{-1}) = 1 - 2.482q^{-1} + 2.021q^{-2} - 0.5382q^{-3}. \end{cases} \quad (8)$$

and for $\hat{y}_2(t)$ ($S_{NO,2}$):

$$\begin{cases} B_1(q^{-1}) = -0.1747q^{-1} + 0.1817q^{-2} \\ B_2(q^{-1}) = 4.156 \times 10^{-5}q^{-1} - 8.21 \times 10^{-5}q^{-2} \\ \quad + 4.054 \times 10^{-5}q^{-3}, \\ F_1(q^{-1}) = 1 - 0.9173q^{-1}, \\ F_2(q^{-1}) = 1 - 2.601q^{-1} + 2.213q^{-2} - 0.6124q^{-3}. \end{cases} \quad (9)$$

The same procedure was applied for the MIMO* case with influent flow rate. In this case, $i = 10$ and $M = 2$ which is even lower than the MIMO case due to the further increase of processing time. Figure 4(d)-(e) show how good the input-output data fit for each of the different cross-fit realizations applied to the MIMO* case. The best model order obtained was from trial 2, which gave a sum average fit of 58.5% and has the following form:

$$OE \left(\begin{bmatrix} 2 & 2 & 2 \\ 1 & 2 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 2 & 2 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \right),$$

with the coefficients for $\hat{y}_1(t)$ ($S_{NH,eff}$):

$$\begin{cases} B_1(q^{-1}) = -0.5266q^{-1} + 0.4534q^{-2}, \\ B_2(q^{-1}) = 1.513 \times 10^{-5}q^{-1} - 1.54 \times 10^{-5}q^{-2}, \\ B_3(q^{-1}) = -3.791 \times 10^{-5}q^{-1} + 5.487 \times 10^{-5}q^{-2}, \\ F_1(q^{-1}) = 1 - 0.9664q^{-1}, \\ F_2(q^{-1}) = 1 - 1.614q^{-1} + 0.6388q^{-2}, \\ F_3(q^{-1}) = 1 - 1.768q^{-1} + 0.8218q^{-2}, \end{cases} \quad (10)$$

and for $\hat{y}_2(t)$ ($S_{NO,2}$):

$$\begin{cases} B_1(q^{-1}) = 0.0091q^{-1}, \\ B_2(q^{-1}) = 2.735 \times 10^{-5}q^{-1} + 2.898 \times 10^{-6}q^{-2}, \\ B_3(q^{-1}) = -4.192 \times 10^{-5}q^{-1}, \\ F_1(q^{-1}) = 1 - 0.9977q^{-1}, \\ F_2(q^{-1}) = 1 - 0.7495q^{-1}, \\ F_3(q^{-1}) = 1 - 0.889q^{-1}. \end{cases} \quad (11)$$

In figure 5(a)-(e) each of the best models is validated against a new simulation. For SISO and MIMO validation, new random sequences are generated for the constant influent scenario, whereas for MIMO* validation new random sequences are generated for the dry influent scenario.

5 Discussions

When more input and output variables are added to the model the complexity increase and the fit goes down. But

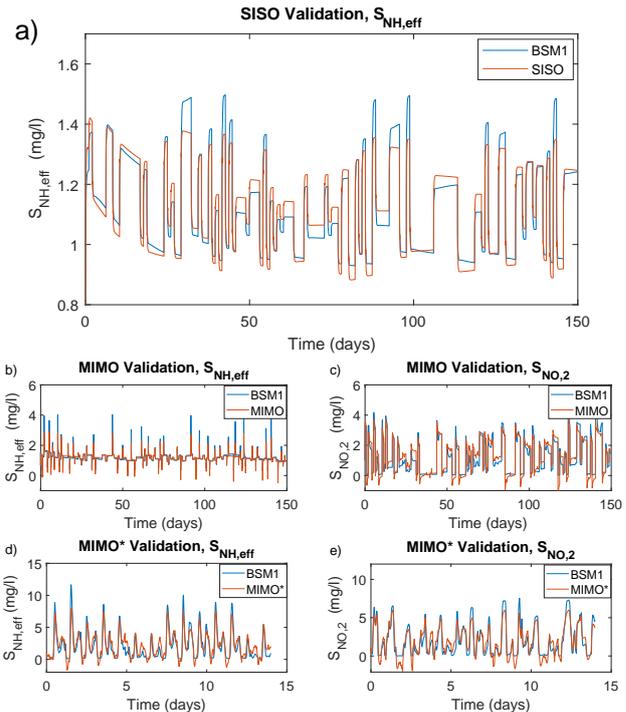


Figure 5. Validation of how well the output data from each of the winning models fit against the output of a new simulation of the same type as the ones used for system identification. Meaning that the input data from this new simulation is used as the input to the model and the output from the simulation is compared with the output of the simulation. (a) SISO case, (b)-(c) Output $S_{NH,eff}$ and $S_{NO,2}$ of MIMO case, (d)-(e) Output $S_{NH,eff}$ and $S_{NO,2}$ of MIMO* case.

since more data is considered and being taken into account in these more complex models, they should also be able to handle variations and changes of the data better. The more complex models could also be further improved by increasing the order of the models if higher processing power or more time was used for the system identification process. Another possibility would be to find correlations between the input variables to the OE model estimator namely nb and nf which could eliminate the need to try different orders of some variables and thus the time required to estimate the model. It would also be possible to have varying order of the different inputs variables since some of the output coefficients are very low while others are higher.

A potential use of system identification is from Model Predictive Control (MPC). Usually, the controllers installed in an ASP are based on a proportional-integral (PI) controllers which regulates the air flow rate in the aeration tanks using the feedback from the effluent ammonia concentration. MPC could enhance the response of a given process since it deals with multivariate constrained control problems in an optimal way. MPC has already been tested in ASP models with good results, see for example Foscoliano et al. (2016); Mulas et al. (2015).

The design of MPC involves a system identification of

the process, where the aim is to achieve a good model of the process in order to get a good control design (Foscoliano et al., 2016). Empirical models have been used for performing the system identification of ASPs, where a pre-defined model order is assumed, see some examples in Vrečko et al. (2004). System identification of simplified ASPs has been carried out by Chistiakova et al. (2017), dealing with linear and non-linear models.

Future studies will be to analyze how well the selected models work as models for MPC and how the fit of the models affects the MPC performance.

Another aspect to consider is different MIMO structures. For example, considering the effluent nitrate concentration. This might require a non-linear system identification of the process.

6 Conclusions

An OE model was used as a black-box model that would describe the input to output relationship of an ASP. A model with a good fit against several scenarios could be obtained by using the rather simple approach of calculating an OE model from input-output data and then cross-validate this model against several similar trials. A model for a SISO scenario could be calculated without much effort to a high model order, whereas the computation time using this method increases fast as the number of input and output variables increase. To reduce the computation time it was required to reduce the cross-validation trials and/or reduce the maximum order of the model which has a negative impact of the fit between the obtained model and the data. Some other black-box models would be investigated, however they might increase the number of parameters to identify.

7 Acknowledgments

This work has received funding from EU Horizon 2020, program SPIRE-2 (Sustainable Process Industry Research), under the project FUDIPO (Future Directions of Production Planning and Optimized Energy - and Process Industries).

References

- H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974. doi:10.1109/tac.1974.1100705.
- J. Alex, L. Benedetti, J. Copp, K.V. Gernaey, U. Jeppsson, I. Nopens, M.-N. Pons, L. Rieger, C. Rosen, J.P. Steyer, P. Vanrolleghem, and S. Winkler. Benchmark simulation model no. 1 (BSM1). Technical Report (TEIE-7229)/1-62/(2008), Dept. of Industrial Electrical Engineering and Automation. Lund University, 2008.
- T. Chistiakova, B. Carlsson, and T. Wigren. Nonlinear modelling of the dissolved oxygen to ammonium dynamics in a nitrifying activated sludge process. In *Instrumentation, Control and Automation - ICA2017*, pages 85–93, Quebec, Canada, 2017.

- M. Ekman. Bilinear black-box identification and MPC of the activated sludge process. *Journal of Process Control*, 18(7-8):643–653, 2008. doi:10.1016/j.jprocont.2007.12.006.
- C. Foscoliano, S. Del Vigo, M. Mulas, and S. Tronci. Predictive control of an activated sludge process for long term operation. *Chemical Engineering Journal*, 304:1031–1044, 2016. doi:10.1016/j.cej.2016.07.018.
- M. Henze, C. Grady, W. Gujer, G. Marais, and T. Matsuo. Activated sludge model no. 1 - scientific and technical report. Technical report, IAWQ, London, UK, 1987.
- C. M. Hurvich and C-L Tsai. Regression and time series model selection in small samples. *Biometrika*, 76(2):297–307, 1989. doi:10.1093/biomet/76.2.297.
- L. Ljung. *System Identification: Theory for the User (2nd Edition)*. Prentice Hall, 1999. ISBN 0136566952.
- M. Mulas, S. Tronci, F. Corona, H. Haimi, P. Lindell, M. Heinonen, R. Vahala, and R. Baratti. Predictive control of an activated sludge process: An application to the Viikmäki wastewater treatment plant. *Journal of Process Control*, 35:89–100, 2015. doi:10.1016/j.jprocont.2015.08.005.
- I. Takács, G.G. Patry, and D. Nolasco. A dynamic model of the clarification-thickening process. *Water Research*, 25(10):1263–1271, 1991. doi:10.1016/0043-1354(91)90066-y.
- D. Vrečko, N. Hvala, and S. Gerlšič. Multivariable predictive control of an activated sludge process with nitrogen removal. *IFAC Proceedings Volumes*, 37(3):505–510, 2004. doi:10.1016/s1474-6670(17)32632-0.
- T. Wigren. User choices and model validation in system identification using nonlinear wiener models. *IFAC Proceedings Volumes*, 36(16):837–842, 2003. doi:10.1016/s1474-6670(17)34864-4.

Distribution of Solids in a Fluidized Bed Operated without a Gas Distributor

Cornelius Agu* Britt M.E. Moldestad

Department of Process, Energy and Environmental Technology
University College of Southeast Norway, 3918 Porsgrunn, Norway, {Cornelius.e.agu,
britt.moldestad}@usn.no

Abstract

Despite several advantages of a gas distributor, there are also some challenges in its application. Using a gas distributor increases the fluidizing gas compression cost. Due to deposition of fine particles and product of chemical corrosion, the gas flow area across a distributor may also be reduced. This paper investigates the distribution of solids in a fluidized bed operated without a gas distributor. The gas supply is through two opposite points on the column wall. In this bed configuration, the fluidized bed behaviour simulated using the CPFD (Computational Particle-Fluid dynamics) Barracuda code is compared with an equivalent system where a uniform gas distribution is assumed. For different powders with mean size 480 – 710 μm , the results show that the axial distribution of pressure fluctuation in both types of bed configurations are similar, particularly near the bottom of the bed. The relative solid fraction fluctuation is lower than 0.2 around the central axis and it spans over 21 – 83% and 41 – 65% of the bed diameter, respectively. These results therefore show that stable bubbling behaviour and good distribution of solids can also be achieved in a bed operated without any distributor.

Keywords: Gas distributor, CPFD Barracuda, Fluidized bed, Pressure fluctuation, Solids fraction

1 Introduction

A gas distributor is usually used in fluidized bed operations. There are different types of gas distributors including porous plates, perforated plates and tuyere nozzles (Paiva et al., 2004; Kunii and Levenspiel, 1991). A gas distributor helps to ensure uniform gas distribution to achieve a stable operation. However, due to pressure drop across the distributor, gas-pumping cost through a fluidized bed can be high in operation. In chemical reactors, the gas passage (pores for porous plate or holes for perforated plate) can be clogged by fine particles, products of corrosion or sintered particles.

In addition, gas distributor design influences the performance of a fluidized bed. The selection of a suitable distributor usually depends on the operation and

particle size group (Kunii and Levenspiel, 1991). For a stable operation, Zuiderweg (1967) proposed that a ratio for a distributor pressure drop to the overall pressure drop can be selected from the range 0.2 – 0.4. Different literatures (Sobrino et al., 2009; Paiva et al., 2009) have investigated the effect of a distributor on a fluidized bed behaviour. Although, some studies employ larger particles at the bottom of a bed as a gas distributor, no available literature has described the performance of a bed without a gas distributor. The aim of this study is to investigate the distribution of solids in a fluidized bed operated without a gas distributor.

The behaviour of a bed without a distributor has been studied in a set of experiments conducted with three different particles with mean size in the range 480 – 710 μm . The pressure drop along the bed and axial distribution of relative pressure drop fluctuations for a system without any gas distributor are compared with those obtained from an equivalent bed fitted with a porous distributor plate. The results are briefly presented in this paper. The bed behaviour at the lower part of the bed was not investigated due to lack of experimental data. In this study, further investigation into the bed behaviour is carried out using the CPFD (Computation Particle-Fluid Dynamics) codes. This study therefore presents the simulated distribution of pressure fluctuation along a bed and distribution of solids fluctuation across the fluidized bed operated without a gas distributor. The results are compared with the simulated results for an equivalent system where a uniformly distributed gas flow is applied at entrance of the bed.

2 Materials and Methods

2.1 Experimental

Table 1 shows the properties of the particles and the minimum fluidization velocities obtained from the two different bed configurations. The experiments were performed in an 8.4 cm diameter cylindrical column and height 1.4 m with pressure sensors fitted at different locations along the column axis.

As can be seen in Table 1, the minimum fluidization velocity in both cases are very close for each of the

Table 1. Bed properties for different particles with the associated range of superficial air velocity.

Material	Sieve size range (μm)	Mean particle diameter (μm)	Density (kg/m^3)	Initial bed solids fraction (-)	U_{mf} (m/s)	
					With plate	Without plate
Limestone	450 - 1100	701	2837	0.478	0.382	0.390
Sand	400 - 1000	685	2650	0.544	0.320	0.330
Sand	100 - 700	484	2650	0.545	0.172	0.180

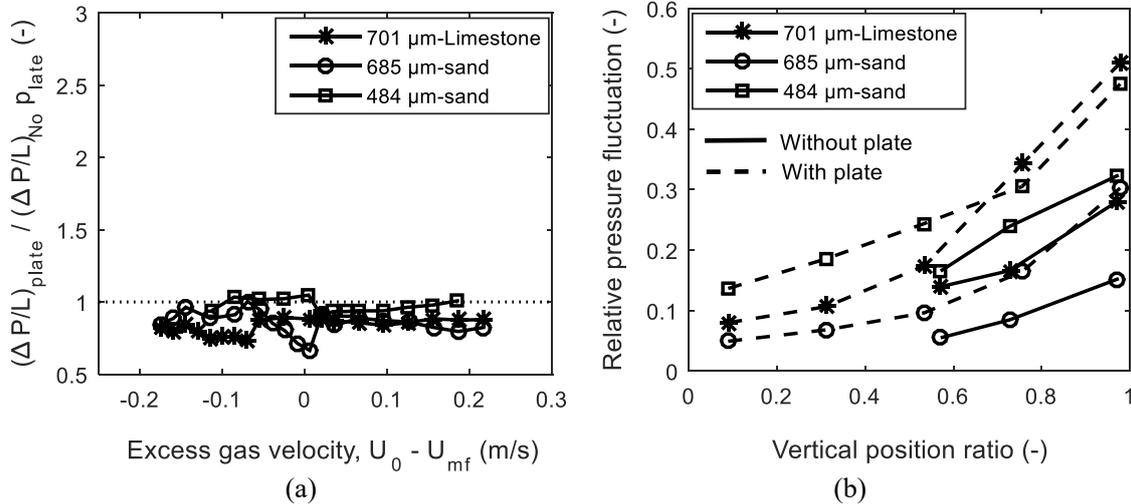


Figure 1. Experimental data (a) pressure drop ratio between a bed with porous plate and without any plate (b) axial distribution of pressure fluctuation comparing two bed cases. $U_0 = 0.571, 0.481$ and 0.361 m/s for the $701\ \mu\text{m}, 685\ \mu\text{m}$ and $484\ \mu\text{m}$ particles, respectively.

powders. In Figure 1 (a), the experimental study shows that the pressure drop in the bed operated with a distributor is relatively lower than that without any distributor, where $(\Delta p/L)_{plate} / (\Delta p/L)_{No\ plate} < 1$. Moreover, since $(\Delta p/L)_{plate} / (\Delta p/L)_{No\ plate} < 1$ when $U_0 - U_{mf} > 0$, it shows that there is no channeling effect in the bed without a distributor. Here, $\Delta p/L$ is the pressure drop per unit length measured across the bed height, U_0 is the superficial gas velocity and U_{mf} is the minimum fluidization velocity of the bed. For the same particle sizes in Table 1, the axial distribution of the pressure drop fluctuations measured above the middle of the bed also shows that a bed operated without a gas distributor is more or less stable than that with the distributor plate within the same section of the bed; see Figure 1(b).

However, the performance of a bed without the gas distributor is believed to have been influenced by the particle size and bed diameter. In the experimental setup, two 4 mm gas entry points are installed at two opposite sides near the bottom of the bed. With these gas introduction points in such a smaller diameter bed, uniform gas distribution may be achieved before the gas flows up to the middle of the bed.

2.2 Computational

The CPFD model is based on the two distinct phases between the fluid and the solids particles. The fluid

phase is modelled based on the Eulerian continuum approach while the particles motion is based on the Lagrangian particle tracing approach. For the fluid phase, the mass balance is expressed as

$$\frac{\partial(\varepsilon_g \rho_g)}{\partial t} + \nabla \cdot (\varepsilon_g \rho_g \mathbf{u}) = 0 \quad (1)$$

Here, $\mathbf{u}(u_x, u_y, u_z)$ is the gas velocity, where u_x, u_y and u_z are the respective velocity components in the x, y and z directions and ε_g is the volume fractions of the gas at any section of the bed. For any given volume within the bed, the model imposes the following relationship between the two phases.

$$\varepsilon_g + \varepsilon_s = 1 \quad (2)$$

where, ε_s is the volume fraction of the solid phase. The momentum balance in the gas phase is described by

$$\frac{\partial(\varepsilon_g \rho_g \mathbf{u})}{\partial t} + \nabla \cdot (\varepsilon_g \rho_g \mathbf{u} \mathbf{u}) = -\nabla p_g - \mathbf{f}_d + \varepsilon_g \rho_g \mathbf{g} + \nabla \cdot \varepsilon_g \boldsymbol{\tau}_g \quad (3)$$

Here, $\boldsymbol{\tau}_g$ is the gas shear tensor and $\mathbf{g}(0,0,-g)$ is the acceleration due to gravity. The momentum exchange rate per unit volume \mathbf{f}_d between the fluid and the individual particle can be obtained from

$$\mathbf{f}_d = \iint f m \left(\beta(\mathbf{u} - \mathbf{v}) - \frac{1}{\rho_s} \nabla p_g \right) dm dv. \quad (4)$$

For the solids phase, the momentum balance is based on the Lagrangian approach such that particles of similar

Table 2. Simulation parameters for the CPF D models.

Parameters	Description	With plate	Without plate
T (K)	Operating temperature	300	300
ρ (kg/m ³)	Air density	1.2	1.2
h_0 (m)	Inlet air flow	Mass flow rate	Mass flow rate
ϵ_{s0} (-)	Initial bed height	0.450	0.485
d_p	Initial solids volume fraction	Fixed bed solids fraction; see Table 1	Fixed bed solids fraction; see Table 1
$\epsilon_{s,max}$ (-)	Particle size	Sieve size range; see Table 1	Sieve size range; see Table 1
Grid	Close pack solids volume fraction	0.64	0.64
Δt (s)	Rectangular		
t_{sim} (s)	Time step	0.001	0.001
	Total simulation time	30	30

properties are grouped together and the group is treated as one particle. The grouping of particles is made on the basis of their sizes, densities and shapes. The individual particle in each group is assumed to be located at position $\mathbf{x}_s(x_s, y_s, z_s)$, where x_s , y_s and z_s are the position coordinate of the particle. The motion of the particles is described as in Eqs (5) and (6).

$$\frac{dv}{dt} + \nabla(\epsilon_g \rho_g \mathbf{u} \cdot \mathbf{u}) = \beta(\mathbf{u} - \mathbf{v}) - \frac{1}{\rho_s} \nabla p_g + \mathbf{g} - \frac{1}{\epsilon_s \rho_s} \nabla \cdot \boldsymbol{\tau}_s, \quad (5)$$

$$\frac{d\mathbf{x}_s}{dt} = \mathbf{v} \quad (6)$$

Here, $\mathbf{v}(v_x, v_y, v_z)$ is the solids velocity in the bed, β is the interphase drag coefficient and $\boldsymbol{\tau}_s$ is the inter-particle normal stress. The detailed descriptions of the CPF D model and its numerical scheme can be found elsewhere (Chen et al., 2013).

In the fluid-particle system, the drag model greatly influences the accuracy of the CFD simulation results. Several drag models can be found in the literature (Wen and Yu, 1966; Gibilaro et al., 1985; Gidaspow, 1994), and a number of these are included in the drag model library of the Barracuda software.

The CPF D model is simulated using the commercial software supplied by the Barracuda AS. The simulations are based on the two different geometries shown in Figure 2 for the two cases described in section 3. In the case with a distributor plate, the fluid flow at the inlet boundary is defined at the bottom of the column. The air velocity at the downstream of the distributor is assumed uniform across the cross-section of the bed. The pressure drop across the distributor and its influence on the hydrodynamic of the bed at different air velocities are not considered in this simulation. For the case without a distributor, the inlet airflow rate is defined at the two opposite sides of the column wall as shown in Figure 2(b).

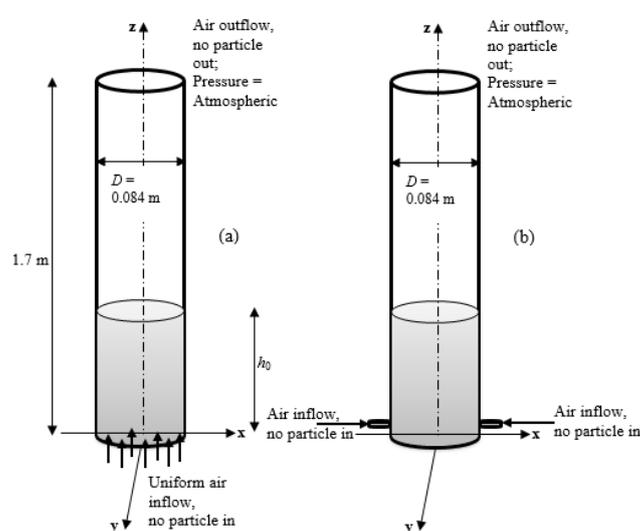


Figure 2. Computational domain for simulation of fluid-particle system with (a) uniformly distributed airflow from the bottom (b) airflow from the sidewalls.

By default, Barracuda generates uniform number of grids across a given cross-section within a computational domain. In this simulation, rectangular grids are used, and their sizes are as shown in Table 2. Also shown in the table are the parameters used in the simulations. The Wen and Yu (1966) drag model is used with the default settings as given in the Barracuda 17.05 version. In this study, the model simulations are used to qualitatively compare the solids distribution and movement between the two cases, and for this reason, all the parameters for particle-particle interactions are left at the default setting in the Barracuda software.

3 Results and discussions

Fluctuation of fluid pressure along a bed height and radial distribution of solid particles in a fluidized bed are simulated using the CPF D Barracuda code. With these simulated results, the behaviour of a bed operated

without any gas distributor is compared with its equivalent system operated with a gas distributor. These two different bed configurations are referred to as Case A and Case B, respectively in this discussion. For Case A, gas is introduced as a jet at two opposite sides of the computational domain as shown in Figure 2, and for Case B, a uniform gas flow distribution is introduced from the bottom.

Figure 3 shows the pressure drop per unit length simulated from the CPFD code for a bed of 685 μm sand particles at different gas velocities. Comparing with the experimental data, the results show that the code predicts the bed behaviour with a good accuracy. The trend of the pressure drop simulated agrees well with the overall pressure drop measured at plenum side of the experimental set up. For estimating the minimum fluidization velocity at the point of maximum pressure drop, the simulated result is compared with the pressure drop profile obtained in the middle of the bed. As can be seen, the simulated minimum fluidization velocity 0.33 m/s agrees very well with the experimental value 0.32 m/s.

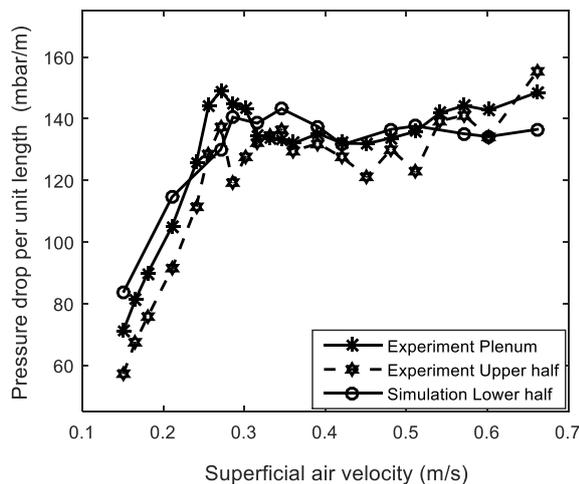


Figure 3. Variation of pressure drop with superficial air velocity in the bed of 685 μm sand particles, comparing the simulated with the experimental results at increasing gas flow rate.

3.1 Axial distribution of relative pressure fluctuation

Pressure fluctuation in a fluidized bed is mainly due to rising of bubbles in the bed (Bi, 1994). With an increase in the gas velocity, the fluctuation of pressure drop in a bed increases due to an increase in bubble size. Figure 4 shows the distribution of pressure fluctuation (computed as the standard deviation of the absolute pressure measured over a period) along the vertical axis in a bed of 685 μm sand particles. This result shows that the pressure fluctuation decreases along the bed height at a given gas velocity. The decreasing value in the pressure fluctuation is associated with a decrease in the fluid pressure up the bed height. The result also shows that the trend of the simulated pressure fluctuation using the

CPFD code agrees with that of the experiment, although quantitatively, there is a significant difference. Within the dense region of the bed, both results show that the fluctuation is slightly constant but decreases rapidly above this region. This also indicates that the fluid pressure fluctuation can be associated with the movement of bulk of particles in a fluidized bed.

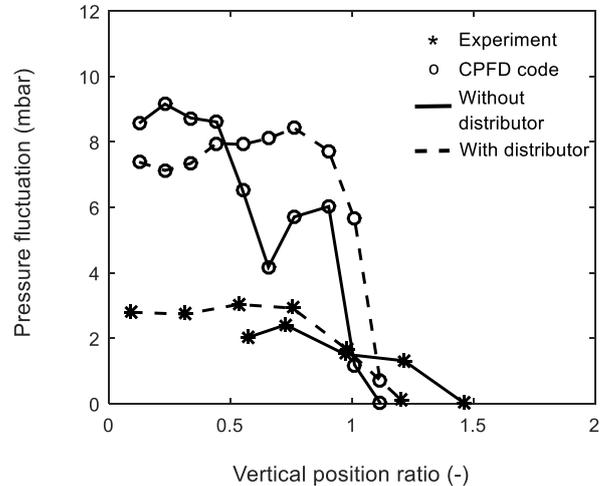


Figure 4. Distribution of absolute pressure fluctuation along the bed height for 685 μm sand particles at $U_0=0.481$ m/s.

However, Figure 4 shows that the absolute pressure fluctuation is more or less chaotic along the bed height due to uneven distribution of bubble activities (coalescence and splitting). With this chaotic behaviour, it will be difficult to compare the hydrodynamics of a bed in the two cases A and B. It can also be seen that the bed fluctuation associated with bubble rise is hidden. Bubbles often grow in size as they rise up a bed. Large bubbles carry significant amount of solid particles in their wakes and when they erupt near the surface of the bed, it results in a relatively higher bed fluctuation. Hence, normalizing the pressure fluctuation with the average fluid pressure will be a good tool for comparing bed behaviour in different fluidized bed systems of the same particle properties. Here, the normalized pressure fluctuation is referred to as the relative pressure fluctuation.

Figure 5 compares the relative pressure fluctuation simulated along the bed height for the two cases. As can be seen, the curve of relative pressure fluctuation increases smoothly along the bed. The relative fluctuation is higher in case B than in case A for both powders: 685 μm and 484 μm sand particles. A higher relative pressure fluctuation indicates a flow of a larger bubble volume. Comparing case A with B, the results in Figure 5 agree with the behaviour obtained in the experiment as shown in Figure 1. Closer to the surface of the bed, the fluctuations are much larger in case B (with plate) than in case A (without plate).

Moreover, similar to case B, Figure 5 shows that the pressure fluctuations in case A decrease smoothly along

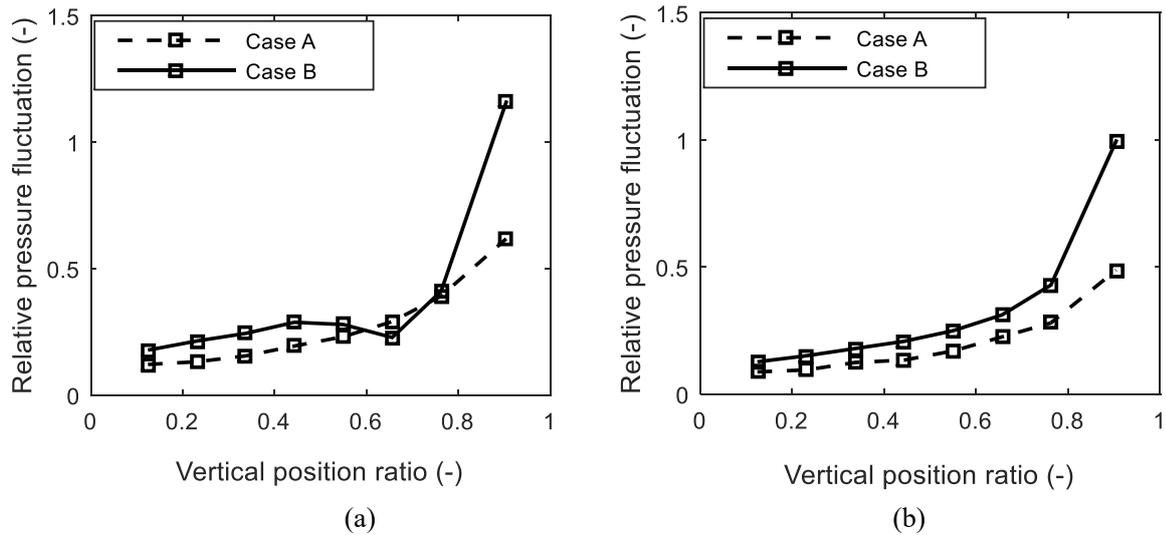


Figure 5. Distribution of relative pressure fluctuation along the bed height (a) 685 μm sand particles at $U_0 = 0.481$ m/s (b) 484 μm sand particles at $U_0 = 0.27$ m/s.

the negative bed axis, which shows that the behaviour at the lower part of the bed without a distributor is as stable as that operated with a gas distributor.

3.2 Radial distribution of solids fraction and axial velocity

Figure 6 shows the radial distribution of the relative solids fraction fluctuation obtained in the middle ($h = 22.5$ cm) of the bed for both cases A and B. The relative solid fraction fluctuation measures the degree of solid movement, and thus the solids mixing across a bed. Similar to the relative pressure fluctuation, the relative solid fraction fluctuation is obtained by dividing the standard deviation by the average solid fraction at each position in the bed radial direction. The results show that the solids movement is more pronounced around the column walls in both cases. Near the central axis of the bed, the degree of solids movement is higher in case B than in case A. The solids fluctuations also vary among the different powders at the same radial position. The variation is shown to be more dependent on the gas velocity above the minimum fluidization velocity than on the particle size. As $U_0 - U_{mf}$ increases, the solids fraction fluctuation increases. In Figure 6(b), the solids fraction fluctuations for the bed of 685 μm sand particles at any radial position and the corresponding values for the 701 μm limestone particles bed are closer to one another since the value of $U_0 - U_{mf}$ is almost the same for both beds. For the case in Figure 6(a), the relative solids fraction fluctuations are almost identical, irrespective of the bed material, particle size and the excess gas velocity.

However, solids fraction fluctuations do not absolutely indicate the direction of movement of the

particles. The fluctuations are brought about either by the upwards or downwards movement of the particles. Figure 7 shows the solids vertical velocities in both cases for the different particle types. The results show that the solid particles move upwards near the walls and downwards towards the centre of the bed. The speed of the solids along the central axis also increases with increasing value of $U_0 - U_{mf}$ as shown in the figures. Along the central axis, the solids speed in the bed of case B is higher than that in the bed of case A. Comparing Figures 6 and 7, it can be seen that the solids fraction fluctuations are greatly influenced by the upwards movement of the particles than by their downwards movement. Since the solids upwards movement is associated with the rise of bubbles, it shows that the solids fraction fluctuation and thus the mixing of solids, are influenced by the rising bubbles in the fluidized beds.

4 Conclusion

Including a gas distributor in a fluidized bed system can help to achieve a uniform gas distribution and stable operation. However, there are still some operational challenges in using a gas distributor. This study investigated the distribution of pressure fluctuation along a bed height and solids movement across the bed when it is operated without any gas distributor. The study was based on simulations using the CFPD Barracuda code.

For three different powders with mean size in the range 480 – 710 μm , the simulated axial pressure distribution was compared with the experimental data,

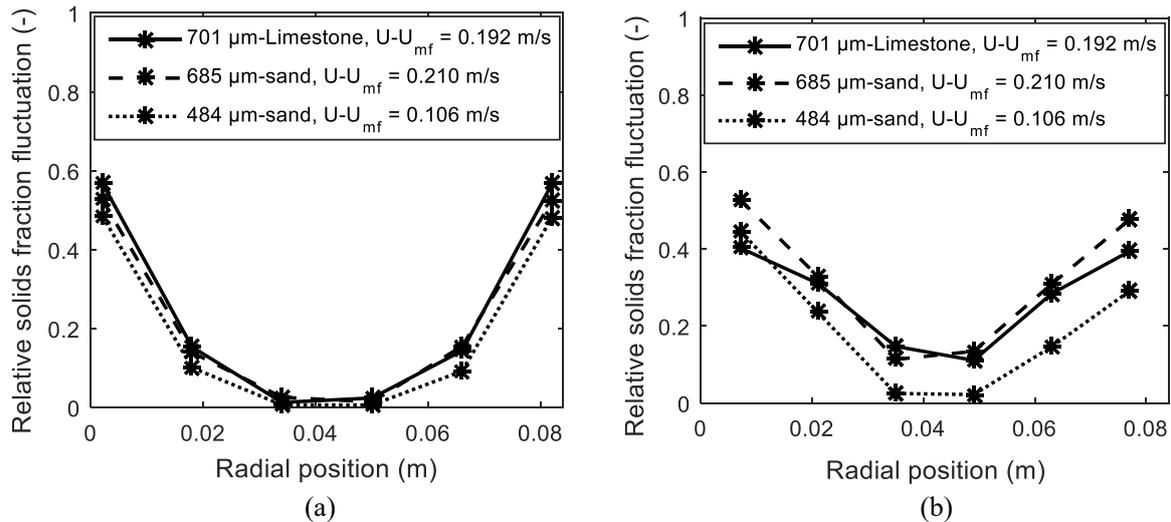


Figure 6. Radial distribution of the relative solids fraction fluctuations in the middle of the beds with (a) Case A (b) Case B.

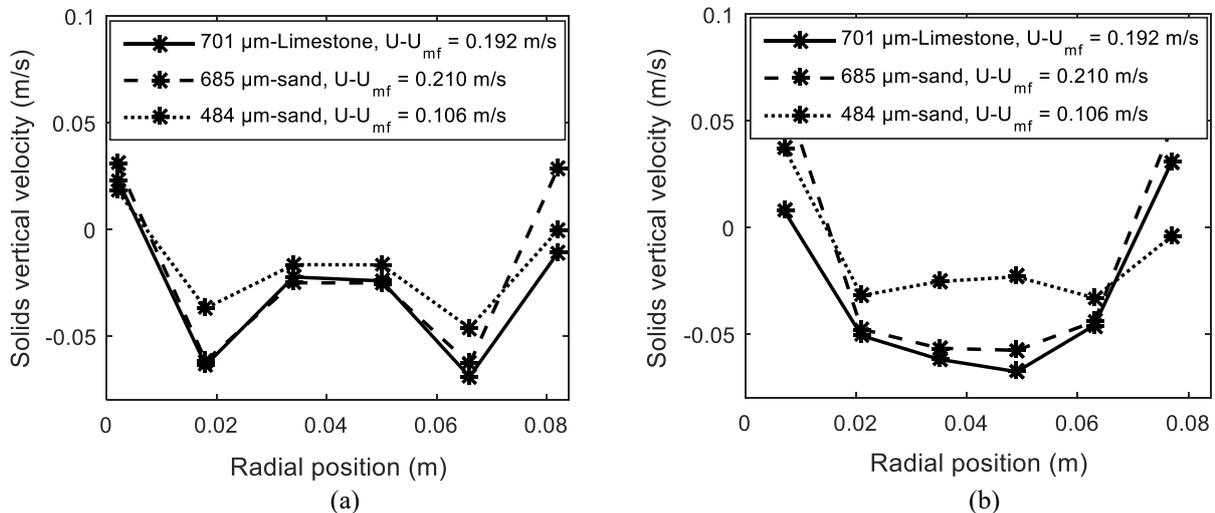


Figure 7. Radial distribution of the solids velocity in the middle of the beds with (a) Case A (b) Case B.

and the results showed that the model predicted the behaviour observed in the experimental data. The absolute pressure fluctuation is almost constant within the dense region of the bed, but decreases rapidly above this region. However, the pressure fluctuation normalized with the average pressure drop increases smoothly along the bed height, indicating an increase in the bubble size and bubble velocity along the bed. The results also show that within the lower region of the bed, the difference between the axial pressure fluctuation for a bed without a distributor and that where a uniform gas distribution is assumed is quite small, but significant at the upper region of the bed. As the relative pressure fluctuation decreases smoothly down the bottom of the bed, it also shows that the bubbling behaviour of the bed operated without a gas distributor is not affected due to absence of a distributor, particularly in the lower part of the bed. The relative solid fraction fluctuation is lower than 0.2 around the central axis and spans over 41 – 65% and 21 – 83% of the bed diameter for the cases with gas distributor and without gas distributor, respectively.

This indicates that there is better mixing behaviour with a uniformly distributed gas flow in the bed.

In all, further investigations are required to ascertain the optimum number of gas entering points and their distribution for efficient operation of a fluidized bed without a gas distributor.

References

- Paiva J.M., Pinho C. and Figueiredo R. The influence of the distributor plate on the bottom zone of a fluidized bed approaching the transition from bubbling to turbulent fluidization. *Chemical Engineering Research and Design*, 82: 25 - 33, 2004.
- Kunii D. and Levenspiel O. *Fluidization Engineering*, 2nd ed., Butterworth – Heinemann, Washington Street, USA, 1991.
- Zuiderweg F.J. Design Report on Fluidization. In *Proceedings of International Symposium on Fluidization*, A.A.H. Drinkenburg, ed., P. 739, Netherlands University Press, Amsterdam, 1967.
- Sobrinho C., Ellis N. and Vega M.D. Distributor effects near the bottom region of turbulent fluidized beds. *Powder Technology*, 189: 25 – 33, 2009.

- Paiva J.M., Pinho C. and Figueiredo R. Influence of the distributor plate and operating conditions on the fluidization quality of a fluidized bed. *Chemical Engineering Communications*, 196: 342 – 361, 2009.
- Chen C., Werther J., Heinrich S., Qi H-Y. and Hartge E-U. CFPD simulation of circulating fluidized bed risers. *Powder Technology*, 235: 238 – 247, 2013.
- Wen C.Y. and Yu Y.H. Mechanics of fluidization. *Chemical Engineering Progress. Symposium*, 62: 100 – 110, 1966.
- Gibilaro L.G., Felice R. Di and Waldram S.P. Generalized Friction Factor and Drag Coefficient for Fluid-Particle Interaction. *Chemical Engineering Science*, 40: 1817 – 1823, 1985.
- Gidaspo D. *Multiphase Flow and Fluidization: Continuum and Kinetics Theory Descriptions*, Academic Press Inc., San Diego, California, USA, 1994.
- Bi X. *Flow Regime Transitions in Gas-Solid Fluidization and Transport*, PhD Thesis, Department of Chemical Engineering, the University of British Columbia, 1994.

Simulation CO₂ storage in the North Sea

Arne O. Torsen, Harris J. Smistad, Håkon Tveit, Ole C. Hansen, Vegard G. Bjørtuft, Nora C. I. Furuvik, Britt M. E. Moldestad

Department of Process, Energy and Environmental Technology, University of Southeast Norway, Norway,
olechr95@gmail.com arne.torsen@hotmail.com vgb_95@hotmail.com tveit1994@gmail.com,
{152545,Nora.C.I.Furuvik,Britt.Moldestad}@usn.no

Abstract

Use of fossil fuel has resulted in emission of large amounts of CO₂ to the atmosphere, which increases the risk of global heating. There is a lot of research going on regarding CO₂ capture and storage. The Paris Agreement, a global agreement on the reduction of the climate change, was adopted in December 2015 and signed by 174 countries in April 2016. The agreement aims to keep the global temperature rise in this century well below two degrees Celsius. According to the International Energy Agency (IEA), about 120 billion tons of CO₂ have to be captured and stored in the period from 2015 to 2050 to ensure that the increase in temperature will be kept below this limit. This implies that more than 3000 carbon capture and storage plants must be operative within 2050. Norwegian Petroleum Directorate has estimated that the theoretical storage capacity on the Norwegian Continental Shelf is 80 billion tons of CO₂. As a comparison, Norwegian emission of CO₂ is about 50 million tons per year.

The purpose of this paper is to simulate the storage of CO₂ in an aquifer in the North Sea. Simulations using OLGA and Rocx are carried out on a limited part of the aquifer. Three different cases were run to study the distribution of CO₂ in the aquifer with time and the CO₂ storage. By comparing the cases, it was found that in all the cases CO₂ was evenly distributed in the aquifer with time independent on the injection area and the length of the production well compared to the length of the aquifer. The CO₂ storage was calculated to be 2590 ton, 3795 ton and 2560 ton for Case 0, Case 1 and Case 2 respectively. Case 2 had a CO₂ injection area of 21 m², whereas Case 0 and Case 1 had a simulated injection area of 10605 m² and 21210 m². Case 2, with the smallest injection area, is the most relevant case because real injection of CO₂ occurs through orifices in vertical or horizontal injectors.

Keywords: OLGA, Rocx, CO₂ storage, CO₂ phase diagram, CO₂ distribution, aquifer.

1 Introduction

When the Norwegian government signed the Kyoto agreement in 1997, they had negotiated an agreement

where Norway was allowed to increase the emissions of green house gases by 1% compared to the 1990 levels. In fact, the greenhouse gas emissions increased by 8% between 1990 and 2010

Due to this, Norway initiated actions to reduce the increase in emissions, and by 2016, the greenhouse gas emissions were reduced to 3% higher than 1990 levels (Statistics Norway, 2017).

The Norwegian government has the ambition to develop cost-effective CO₂ capture and storage technologies and to realize at least one full-scale CO₂ capture demonstration facility by 2020 (Stortinget, 2013). Gassnova has developed an idea study, "Exploration of possible full-scale CO₂ handling projects in Norway", where several sources of emissions and storage locations have been considered as current candidates. During the autumn 2015, the government decided to continue the project in a feasibility study. CO₂ capture studies have been conducted with three industrial actors, and according to these studies, it is concluded that it is technically possible to realize a CO₂ handling chain in Norway. Smeaheia, located just east of the Troll field, is especially drawn up as a good solution. Statoil concluded based on the study that the Smeaheia solution has the least risk, greatest operational flexibility and greatest potential for future capacity expansion. (Olje- og energidepartementet, 2016). The United Nations Inter-governmental Panel on Climate Change (IPCC) concluded that capture and storage of CO₂ may account for as much as one half of emission reductions in this century (Norwegian Petroleum Directorate).

The aim of this paper is to study the storage capacity and the distribution of CO₂ in an aquifer when different injection areas are used.

2 CO₂ storage in aquifers

There is significant technical potential for storing CO₂ in geological formations around the world. Saline aquifers are candidates for such storage. Environmentally sound storage of CO₂ is a precondition for a successful Carbon Capture and Storage (CCS) chain, and therefore mapping, qualification and verification of storage sites are important. Geological

formations on the Norwegian continental shelf are considered to have potential for storage for large quantities of CO₂. This paper is focused on the storage capacity in an aquifer east of the Troll field. (Norwegian Petroleum Directorate).

2.1 CO₂ storage capacity

Smeaheia is an aquifer located east of the Troll oil and gas field. A layer of the aquifer in Smeaheia, Sognefjord Delta East, is evaluated for CO₂ storage as part of a full cycle CCS pilot initiated by the Norwegian government (Riis et al., 2017). The Sognefjord Delta East aquifer, and has a permeability of 300 mD. The bulk volume is $5.54 \cdot 10^{11} \text{m}^3$ and the porosity is 0.210 (Lothe et al., 2014). The equation for calculation of the storage capacity is:

$$q = V_{Bulk} \cdot N/G \cdot \varphi \cdot \rho_{CO_2} \cdot E \quad (1)$$

where q is the mass of CO₂, V_{Bulk} is the total volume of the aquifer, N/G is the net to gross height of the aquifer, φ is the porosity of the rock, ρ_{CO_2} is the density of CO₂ at reservoir conditions and E is the storage efficiency. The storage efficiency is a function of mobility ratio, the connate water saturation and the trapping coefficient and is given by (Szulczewski and Juanes, 2009).

$$E = \frac{2 \cdot M \cdot \Gamma^2 (1 - S_{wc})}{\Gamma^2 + (2 - \Gamma)(1 - M + M \cdot \Gamma)} \quad (2)$$

The mobility ratio, M , is expressed by:

$$M = \frac{1}{\mu_{water}} \cdot \frac{\mu_{CO_2}}{k_{rg}^*} \quad (3)$$

where μ_{water} and μ_{CO_2} are the viscosity of water and CO₂ respectively, and k_{rg}^* is the end point relative permeability for CO₂. The trapping coefficient, Γ , can be calculated from:

$$\Gamma = \frac{S_{rg}}{1 - S_{wc}} \quad (4)$$

S_{rg} is the residual saturation of CO₂ and S_{wc} is the connate water saturation.

The storage efficiency, E , in the Sognefjord Delta East aquifer is 5.5% and the density of CO₂ is 0.69 ton/Rm³ (Lothe et al., 2014). Based on these values and the total volume and porosity, the CO₂ storage capacity of the Sognefjord Delta East is calculated to be 4.09 Gtons. By comparison, Europe has an annual CO₂ emission of 4.4 Gtons.

2.2 CO₂ injection in aquifers and reservoirs

In addition to CO₂ storage, injection of CO₂ into hydro carbon reservoirs is more and more used. Depleted oil reservoirs is used for storage of CO₂, whereas injection of CO₂ into active reservoirs is used for enhanced oil recovery (EOR) and storage. Figure 1 illustrates CO₂

storage in a hydrocarbon field with enhanced oil recovery (to the left) and CO₂ storage in an aquifer (to the right) (Université Recherche, 2018).

Injection of CO₂ in an aquifer below an oil reservoir, contributes to maintain the reservoir pressure and to reduce the interface tensions between oil and water resulting in increased oil mobility. CO₂ injection for EOR provides a great environmental benefit by reducing the emission of CO₂ to the atmosphere.

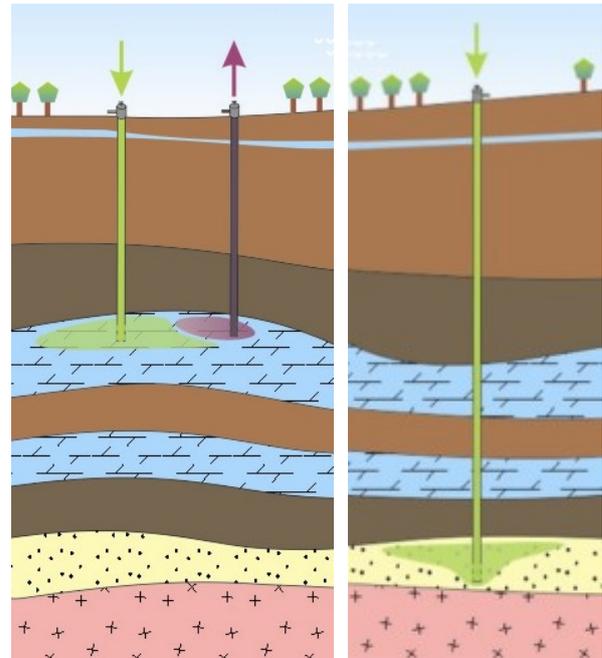


Figure 1. CO₂ storage in a reservoir with enhanced oil (Université Recherche, 2018).

CO₂ can be injected into an aquifer as liquid, gas or supercritical fluid. Figure 2 shows the phase diagram for CO₂. The critical point for CO₂ is 72.9 atm. and 31 °C, which implies that CO₂ very often is injected as supercritical fluid. The Sognefjord Delta East aquifer is located at an average depth of 1750 m (Lothe et al., 2014), and the pressure is ranging from about 170-200 bar. At these conditions, the CO₂ density is reported to be 690 kg/m³ and is in supercritical condition. (Lothe et al., 2014).

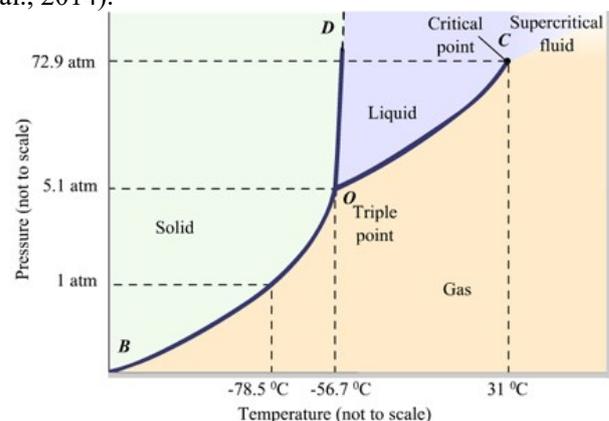


Figure 2. Phase diagram for CO₂.

A significant pressure depletion in the storage site of the formation, will bring CO₂ into the gas phase. If this happens, the storage capacity will be significantly decreased due to the density difference between the gas phase and the supercritical phase.

When injecting CO₂ for storage, it can be a challenge to obtain even distribution of CO₂ in the aquifer. In homogeneous aquifers, the anisotropic condition is important for the distribution of CO₂. Anisotropy is the ratio vertical to horizontal permeability. Vertical or horizontal wells are used for injection of CO₂ to an aquifer. The use of horizontal wells does not improve the storage efficiency significantly, and vertical wells provide higher storage efficiency than horizontal wells under strongly anisotropic conditions. Horizontal wells are preferable if the goal is to sequester a large amount of CO₂ in a short period. (Okwen et al., 2010)

3 OLGA and Rocx

OLGA is a software developed to simulate flow of oil, water and gas in pipelines and process equipment. Rocx is a near-well reservoir simulator, which can be combined with OLGA and enables the user to specify the reservoir properties in the near-well area. The coupling between OLGA and Rocx accounts for the dynamic wellbore/rock interactions, which is not directly possible by separate reservoir and well models. (OLGA Handbook). The inputs and set-up for Rocx and OLGA are presented in the following.

3.1 Rocx

Reservoir properties as porosity, saturation, permeability, relative permeability and capillary pressure are input parameters to Rocx. In addition fluid properties as viscosity, density, bubble point and gas/oil ratio are needed for the simulations. Initial and boundary conditions are set for the reservoir. Rocx gets information from OLGA regarding pressure and pressure drop in the well and through the inflow devices. Based on the reservoir and fluid information and the information given in OLGA, the production or injection rates are calculated.

The porosity is the storage capacity of an aquifer, and is defined as the ratio of the pore volume to the total rock volume. Pores can be interconnected with other pores or completely isolated. The different types of pores are shown in Figure 3. To describe the types of porosity, effective and absolute porosity are used. The effective porosity is the volume of the interconnected pores to the total volume of the rock, and is used in reservoir engineering calculations. The absolute porosity includes the total pore volume. (Tarek Ahmed, 2006).

Saturation is defined as the fraction of the pore volume occupied by a particular fluid. In an aquifer, the saturation is given as water and gas/CO₂ saturation (Tarek Ahmed, 2006).

Permeability is referred to as absolute permeability, effective permeability and relative permeability. The absolute permeability is a rock property, and is defined as the capacity and ability of a reservoir to transmit fluids. Absolute permeability is defined from Darcy's law as:

$$\dot{Q} = \frac{k \cdot A}{\mu} \cdot \frac{dP}{dL} \quad (5)$$

where \dot{Q} is the fluid volume flow, k is the permeability, μ is the viscosity of the flowing fluid, A is the cross section flow area and dP/dL is the pressure drop per unit length. The permeability is given in Darcy (D). (Tarek Ahmed, 2006).

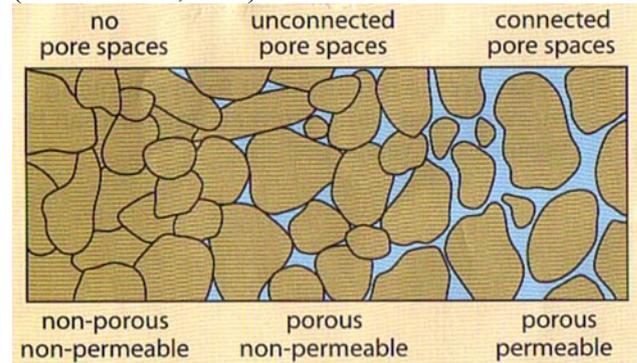


Figure 3. Different types of pores in a rock

Effective permeability is a function of the reservoir saturation and the characteristic of the rock, and is measured directly on small core samples in the laboratory. The effective permeability is measured for oil, water and gas/CO₂. The relative permeability is the ratio of the effective permeability of a fluid to the absolute permeability. The relative permeability of each phase depends on the saturation of the different phases in the reservoir. (Tarek Ahmed, 2006).

In this study, the relative permeability curves implemented in the Rocx model are based on experimental data from literature. The relative permeability curves can also be calculated by using the Corey model for CO₂ and water. The Corey model (Rocx Online Help, 2007) for water is expressed by:

$$K_{rw} = K_{rwoc} \left(\frac{S_w - S_{wc}}{1 - S_{wc} - S_{or}} \right)^{n_w} \quad (6)$$

where S_w is the water saturation, S_{wc} is the irreducible water saturation, S_{or} is the residual oil saturation, K_{rwoc} is the end point relative permeability for water at maximum water saturation, and n_w is the Corey exponent. (Rocx Online Help, 2007). Typical values for the Corey exponent for water are $n_w = 2-3$. When CO₂ is injected into an aquifer, the residual and endpoint relative permeability changes. Figure 4 shows an example of relative permeability curves for CO₂ and brine as a function of the CO₂ saturation (Paterson et al., 2011).

Rocx is calculating the aquifer or reservoir properties as a function of time. The main focus in this paper has been to study the changes in saturation of CO₂ and water

with time. Techplot has been used to plot the changes in fluid saturation in the aquifer.

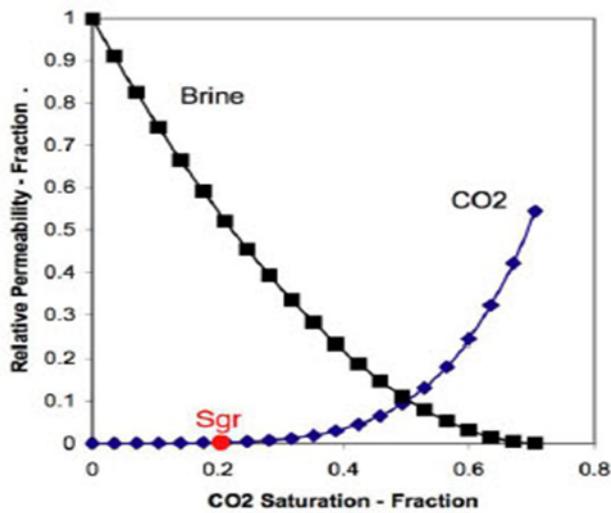


Figure 4. Relative permeability curves for CO₂ and water (Paterson et al., 2011)

3.2 OLGA

OLGA is a one-dimensional transient dynamic multi-phase simulator used to simulate flow in pipelines and connected equipment. The OLGA simulator is governed by conservation of mass, momentum and energy equations for the different phases (OLGA Handbook).

The well or injector design can be specified in OLGA. Figure 5 shows a set-up for injection of CO₂ for EOR in an oil reservoir tested by Vesjolaja et al. (Vesjolaja et al., 2018). Pressure sources are used for the injection points. However, it was concluded that injection of CO₂ directly to the reservoir, was not possible. The reason was that in the available OLGA/Rocx model, the only option was to use the black oil model, which was not compatible with CO₂ injection (Vesjolaja et al., 2018). The effect of CO₂ injection was therefore simulated by changing the relative permeability curves.

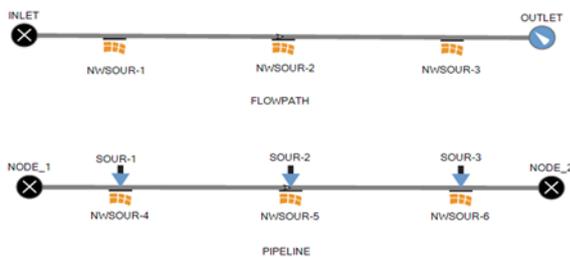


Figure 5. CO₂ injection design in OLGA (Vesjolaja et al., 2018).

In this study, the simulations are carried out by assuming that CO₂ with a constant pressure is injected continuously from the bottom of the aquifer. The constant CO₂ pressure is specified in Rocx as the pressure boundary condition.

In addition to the injector or well geometry, fluid properties and conditions, initial and boundary conditions, time steps and production/injection period are specified as input to OLGA. The results are plotted as trend plots (selected parameter as a function of time) or as profile plot (selected parameter as a function of location in a well/injector).

4 Results

Simulations have been carried out using the simulation tool Rocx in combination with OLGA.

4.1 Input data to OLGA and Rocx

The characteristics of the aquifer were chosen based on the available information about the Smeaheia aquifer. The aquifer is simulated as a homogeneous rock with an average porosity of 0.21 and horizontal permeability of 690 mD. The CO₂ is injected as supercritical fluid with density 711 kg/m³. The rock is defined as sandstone. The characteristics of the Sognefjord Delta East aquifer are listed in Table 1 together with the characteristics used in the simulations.

The simulated aquifer has a length, width and height of 105 m, 101m and 100 m respectively. The horizontal CO₂ injector is assumed located in the bottom of the aquifer, and is specified as a constant pressure boundary condition. To simulate the pressure difference between the injector and the aquifer, a well is located in the upper part of the aquifer. The boundary condition for the well is constant outlet pressure. Figure 5 presents a schematic overview of the simulated aquifer.

The CO₂ drive pressure from the bottom of the aquifer is 175 bar and the pressure at the outlet of the well is set to 130 bar. The number of control volumes in x-, y- and z-direction are 5, 21 and 10 respectively. It was assumed that the initial saturation of water in the aquifer was 100%.

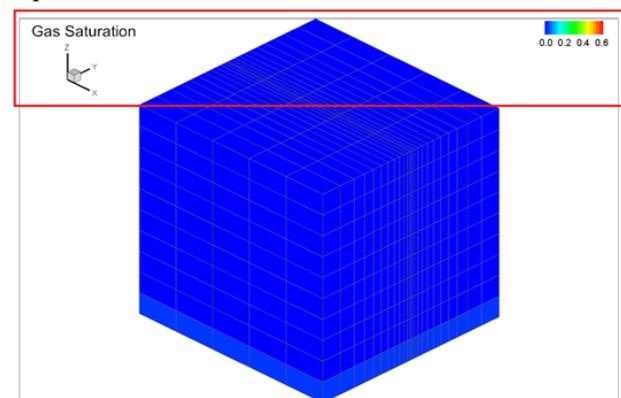


Figure 6. Schematic overview of the simulated aquifer.

The design of the well was developed in OLGA and is presented in Figure 7. The well includes connection to the aquifer via the sources (NWSOUR-1, NWSOUR-2, NWSOUR-3). The “leaks” (LEAK-1, LEAK-2, LEAK-3) represent the flow from the annulus (PIPELINE)

through the open valves (VALVE-A, VALVE-B, VALVE-C) to the production pipe (FLOWPATH). Packers are presented as closed valves (VALVE-1, VALVE-2) and are used to avoid annulus flow. The outlet pressure from the well is 130 bar, and is specified as a boundary condition in the pressure node.

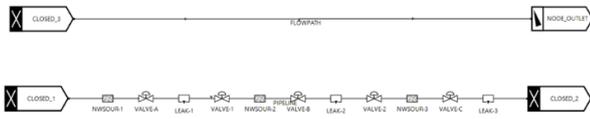


Figure 7. Well design including valves, packers, sources and leaks.

A summary of the input parameters to OLGA and Rocx are presented in Table 1, and the relative permeability curves used in the simulations are shown in Figure 8.

Table 1. Characteristics of Troll aquifer field and the simulated aquifer.

Parameter	Troll aquifer	Simulated aquifer
Water density	1023	Calculated at reservoir conditions
CO ₂ density	711 kg/Am ³	711 kg/Am ³
Porosity	0.21	0.21
Permeability	690 mD	x- and y- directions: 690 mD z-direction: 69 mD
Volume aquifer	5.54 · 10 ¹¹ m ³	1.06 · 10 ⁶ m ³ and 2.12 · 10 ⁶ m ³
Thickness		100 m
Aquifer pressure	175 bara	175 bara
Aquifer temperature	54 °C	54 °C
Initial water saturation	-	1
Irreducible water saturation	-	0.2
Residual CO ₂ saturation	-	0.2

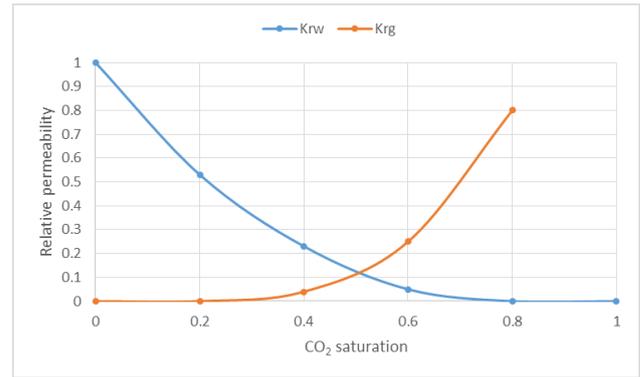


Figure 8. Relative permeability curves for water (blue) and CO₂ (red).

4.2 Results

Three different cases were simulated, Case 0 (Base case), Case 1 and Case 2. In Case 0 and Case 2, the length, width and height of the aquifer are 105m, 101m, 100m respectively. In Case 1, the length of the aquifer is increased to 210m, whereas the width and the height are the same as for Case 0 and Case 2. The well is located in the top of the aquifer, and has a length of 105m for all the cases. In Case 0 and Case 1, it was assumed that CO₂ was injected continuously from the whole bottom area of the aquifer. In Case 2 CO₂ was injected through only one control volume with area 105 m². The conditions for the cases are mainly selected to study the effect of the injection area and the aquifer length on CO₂ distribution and storage in the in the aquifer. The model is not validated to give exact results about CO₂ saturation and storage in the aquifer as a function of time. Also, the pressure drive (175-130 bar), which gives the injection rates, is set high to reduce the required simulation time. A grid resolution test is not performed in this study, but has earlier been performed by the research group for similar cases. The mesh is found to be acceptable to see whether the injection method and reservoir length have an effect on the distribution and storage of CO₂ in an aquifer. Figure 9 presents the distribution of CO₂ in the aquifer after 0, 20, 50 and 80 days. It can be seen that the CO₂ is equally distributed over the x-y area, and are moving gradually towards the top section with time. After 80 days, the average saturation of CO₂ in the aquifer is about 50%.

The results from simulation of Case 1 are presented in Figure 10. In this case, the length of the well (105 m) is half of the aquifer (210 m) and is located on the left side of the upper surface. Due to the location of the well, the CO₂ is not evenly distributed in the aquifer, and after 120 days, the water saturation in the upper right part of the aquifer is still about 100%.

Figure 11 shows the location of the CO₂ injection in Case 2, and Figure 12 gives the distribution of CO₂ in the aquifer after 30, 60, 90 and 105 days. As can be seen, the distribution of CO₂ with time is very different when the CO₂ is injected from a small area. The CO₂ is not

distributed equally with height in the aquifer, and large areas contain about 100% water.

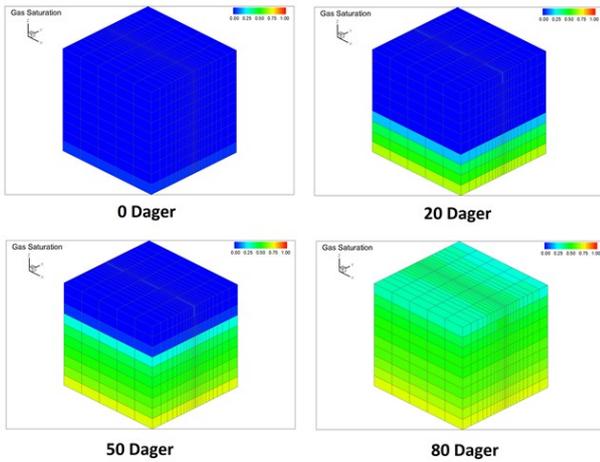


Figure 9. Distribution of CO₂ in the aquifer, Case 0.

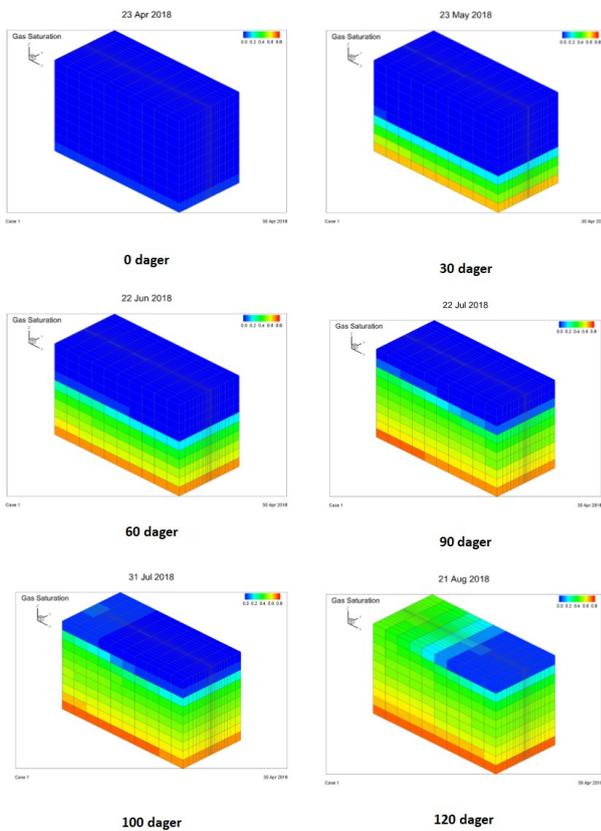


Figure 10. Distribution of CO₂ in the aquifer, Case 1.

However, after 105 days, CO₂ is about equally distributed up to a height of about 80 m. After 124 days, CO₂ breakthrough to the well occurs. The CO₂ storage for Case 2 is calculated based on 124 days.

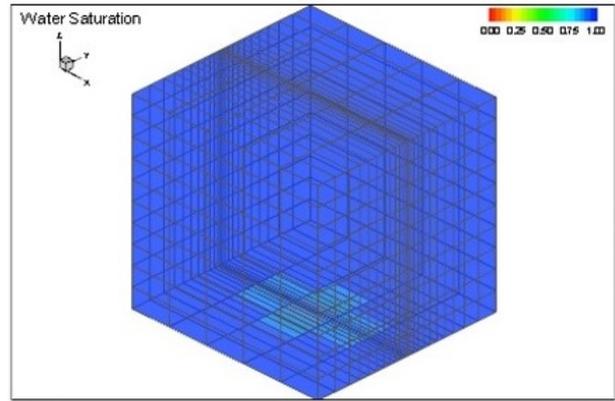


Figure 11. Injection of CO₂ from a limited area, Case 2.

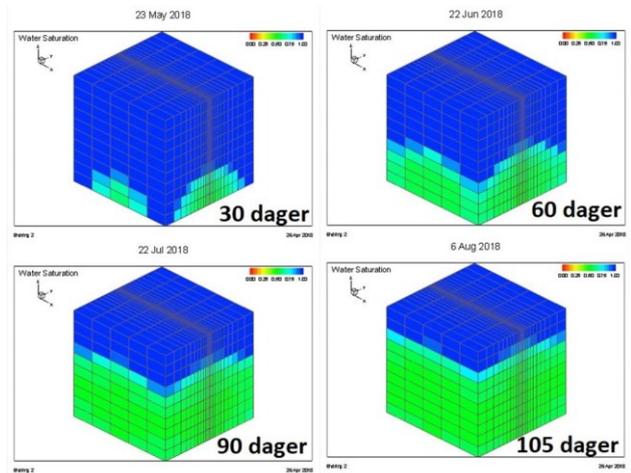


Figure 12. Distribution of CO₂ in the aquifer, Case 2.

The replaced mass of water for the three cases is presented in Figure 13. The storage of CO₂ in the aquifer is calculated from Equation (1) and summarized in Table 2. The calculations are based on the mass of displaced water until breakthrough of CO₂ to the well occurs. After breakthrough, the well is mainly producing CO₂, and these periods are therefore not considered in the calculations of CO₂ storage. To calculate the storage of CO₂, the density ratio of water and supercritical CO₂ is used. The displaced water is calculated from OLGA, and is the water produced in the period before breakthrough of CO₂ to the well. The breakthrough occurred after 82, 116 and 124 for Case 0, Case 1 and Case 2 respectively. The calculated storage of CO₂ for Case 0, Case 1 and Case 2 are 2590 ton, 3795 ton and 2560 ton. By comparing Case 0 and Case 1, the storage of CO₂ is significantly higher in Case 1 because the simulated aquifer is twice as big as in Case 0, and that the storage has occurred for a longer period. Case 2 gives about the same amount of stored CO₂ as Case 0, but the storage period is longer than in Case 0 and Case 1. This is due to the small injection area used in Case 2. The injection area was 105 m² in Case 2 compared to 10605 and 21210 m² in Case 0 and Case 1 respectively. However, for all the cases, a good distribution of CO₂ in the aquifer is obtained over time.

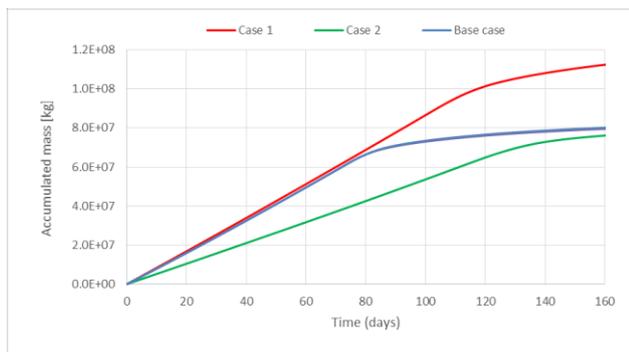


Figure 13. Displaced water during simulation of Case 0, Case 1 and Case 2.

This result is in good agreement with the results from the study performed by Okwen et al. (Okwen et al., 2010). The simulations with OLGA/Rocx managed to meet the objective of this study, which was to show the effect of injection area and aquifer volume on CO₂ distribution and storage. However, to produce more accurate results, effort has to be made to develop a robust model without the current limitations. Case 2 is the most relevant case and will be used as basis for further studies.

Table 2. CO₂ storage capacity calculated from Eq. 1.

Case	Pore volume of simulated aquifer [m ³]	CO ₂ [ton] storage
Case 0	222700	2590
Case 1	445400	3795
Case 2	222700	2560

5 Conclusion

Storage of CO₂ in an aquifer in the North Sea is considered. Simulations are carried out on a limited part of the aquifer. Available information about porosity, permeability, pressure and temperature in the aquifer are used as input to the simulations. OLGA and Rocx are used as the simulation tools. Due to problems with implementing a CO₂ injector in OLGA/Rocx, a model was developed where water was displaced by CO₂ and removed from the aquifer through a production well. After a period, CO₂ breakthrough to the well occurred. Three different cases were run to study the distribution of CO₂ in the aquifer with time and the CO₂ storage for the period before breakthrough. By comparing the cases, it was found that in all the cases CO₂ was evenly distributed in the aquifer with time independent on the injection area and the length of the production well compared to the length of the aquifer. The case with the smallest injection area is the most relevant case because real injection of CO₂ occurs through orifices in vertical or horizontal injectors. It was shown that the storage

capacity is dependent on injection period and volume of the aquifer, and not on the injection area. For further studies of CO₂ injection and distribution of CO₂ in aquifers a new model will be developed and validated against experimental core tests. In addition, a grid resolution test will be performed, and the driving pressure will be reduced. If possible, a CO₂ injector will be implemented in OLGA to obtain more realistic injection rates.

References

Tarek Ahmed, (2006) Reservoir Engineering Handbook, Third edition, Elsevier, Gulf Professional Publishing, ISBN 13: 978-0-7506-7972-5.

Ane Lothe, Benjamin Emmel, Per Bergmo, Gry Møl Mortensen, Peter Frykman, A first estimation of storage potential for selected aquifer cases, NORDICCS Technical Report D 6.3.1302 (D25), 2014.

Norwegian Petroleum Directorate, CO₂ storage atlas, Norwegian North Sea,

<http://www.npd.no/Global/Norsk/3-Publikasjoner/Rapporter/PDF/CO2-ATLAS-lav.pdf>

Roland Okwen, Mark Stewart, Jeffrey Cunningham, Effect of well orientation (vertical vs. horizontal) and well length on the injection of CO₂ in deep saline aquifers, *Transport in Porous Media*, 2010.

https://www.researchgate.net/publication/225451088_Effect_of_Well_Orientation_Vertical_vs_Horizontal_and_Well_Length_on_the_Injection_of_CO2_in_Deep_Saline_Aquifers

OLGA Handbook,

<https://www.scribd.com/document/114140977/Olga-Handbook-pdf>

Olje- og energidepartementet, Mulighetsstudier av fullskala CO₂-håndtering i Norge, 2016,

<https://www.regjeringen.no/globalassets/departementene/oen/pdf/mulighetsstudien.pdf>

L. Paterson, C. Boreham, M. Bunch, J. Ennis-King, B. Freifeld, R. Haese, C. Jenkins, M. Raab, R. Singh and L. Stalker, The CO₂CRC Otway stage 2B residual saturation and dissolution test: test concept, implementation and data Collected, Milestone report to ANLEC 2011, CO₂CRC Report No: RPT11-315, 2011.

<https://hub.globalccsinstitute.com/sites/default/files/sites/default/files/co2crc-otway-stage2b-residual-saturation-dissolution-test.pdf>

Fridtjof Riis, Nina Pedersen, Marthe A. Birkeland, Tom Hultgreen, Troll Area, Norwegian North Sea: Case Study of CO₂ Storage Sites in an Aquifer Under Depletion, International Conference and Exhibition, London, England, October 15-18, 2017.

[Rocx Online Help. Reservoir characteristics, Schlumberger, Rocx Reservoir Simulator, 1.2.5.0 edition, 2007](#)

Statistics Norway, 2017,

<https://www.ssb.no/klimagassn>

Store norske leksikon, Kyotoavtalen, 2016,

<https://snl.no/Kyotoprotokollen>

Stortinget, 2013, <https://www.stortinget.no/nn/Saker-og-publikasjoner/Sporsmal/Skriftlige-sporsmal-og-svar/Skriftlig-sporsmal/?qid=58394>

Michael Szulczewski, Ruben Juanes, 2009, A simple but rigorous model for calculating CO₂ storage capacity in deep saline aquifers at the basin scale, *Energy Procedia*, pp. 3307-3314, 2009.

Université Recherche, Research Group on Sedimentary Basin Energy Resources in Québec, 2018.

http://grrebs.ete.inrs.ca/en/csc/csc_stockage/

Ludmila Vesjolaja, Ambrose Ugwu, Arash Abbasi, Emmanuel Okoye, Britt M. E. Moldestad, Simulation of CO₂ for enhanced oil recovery, *Linköping Electronic Conference Proceedings* (ISSN: 1650-3686, eISSN: 1650-3740), 2018

Improving bench scale design for polymerization of green CO₂-polymers

Thomas S. Larsen¹, Kai A. Sætre², Sara Ronasi², Hildegunn H. Haugen¹, Britt M. E. Moldestad¹

¹Department of Process, Energy and Environmental Technology, University College of Southeast Norway, Norway, thomas.storesund.larsen@gmail.com hildegunn.h.haugen@usn.no britt.moldestad@usn.no

²Norner, Norway, kai.saetre@norner.no sara.ronasi@norner.no

Abstract

Plastic has become the most common material since the beginning of the 20th century and has become almost indispensable due to its durability, light weight and low costs. However, some of the characteristics also make the plastic problematic when it comes to the end of its life phase. Norner Research AS, a polymer research institute performing extensive independent or funded research in the plastic value chain from process technology to final product and application, has during the last ten years been active in developing biodegradable green plastic that can replace up to 40% of the fossil raw materials with industrial captured CO₂. The green plastic, poly(propylene carbonate) (PPC), is produced by reacting propylene oxide (PO) with CO₂. The aim of this paper is to study the effect of static mixer on production of PPC and improve the catalytic process by reducing the reaction time and increasing the yield. The process includes multiphase gas-liquid flow, and good mixing is one of the key factors to succeed in improving the process. The flow behavior of the components in the multi-phase bench scale reactor is studied in order to investigate the potential of reducing the reaction time by using static mixers to increase mass transfer. A static mixer is a device for the continuous mixing of fluids, and can be used to mix liquids, gases or to mix a gas into a liquid. The energy needed for mixing comes from a loss in pressure as the fluids flow through the static mixer.

Computational fluid dynamic (CFD) is an important tool in order to simulate and optimize the polymerization process. In this study, the CFD software Ansys/Fluent is used to investigate the multiphase flow through the static mixer. The static mixer is simulated to study the effect on the gas-liquid mixing. The simulations were performed with mixtures of 17% and 34% PPC in PO under CO₂ pressure. The simulations showed that the required number of mixing elements (L) to obtain a uniform mixing of CO₂, PO and PPC in the liquid phase is estimated to be 6.5L for the 17% PPC case and 5.25L for the 34% PPC case. Consequently, using more than the required number of mixing elements would increase the pressure drop without increasing the mixing. It was observed that the required mixer length decreased when the composition of PPC in the inlet liquid phase was increased from 17% to 34%.

Keywords: *Carbon dioxide based polymers, poly(propylene carbonate) green plastic, CFD, Ansys/Fluent, bench scale pilot reactor, static mixer*

1 Introduction

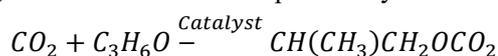
In recent years, the interest in green, sustainable materials to replace more common, unsustainable materials is becoming more and more relevant (Luinstra, 2008). One of the many reasons for this may be the lack of biodegradability of commonly used polymers such as polyethylene (PE) and polypropylene (PP). The stability of these polymers has a negative impact in nature, especially for marine life (Tokiwa et al., 2009). Another central topic nowadays is the CO₂ produced by modern activities and its contribution to the “greenhouse effect”. CO₂ is one of the largest resources of carbon readily available, and several journals suggest that reducing the CO₂ levels in the atmosphere can be beneficial towards halting climate change from human activities (Mikkelsen et al., 2010).

The process of synthesizing PPC from CO₂ was studied and published as far back as 1969 (Inoue et al., 1969). As time passed, varying levels of research was put into this topic. At the start of the 21st. century, climate change by human activities and its relation to increased CO₂ levels in the atmosphere as well as concerns on plastic waste management has sparked a new interest in using CO₂ based polymers (Luinstra, 2008). However, the main focus in these studies were on development of effective catalyst solutions and not much work is reported on the process development (Narang et al., 2016; Kember et al., 2011).

The use of static mixer in a tubular reactor was shown to be advantageous for similar process (Gürtler et al., 2013). In this study the multi phase flow through the static mixer is investigated.

2 Production of green polymer

PPC is produced by the co-polymerization between the greenhouse gas carbon dioxide (CO₂) and propylene oxide (PO) in the presence of a catalyst. Depending on the choice of the catalyst system, temperatures in the range of 50 to 100°C and pressures up to 60 bar are reported in literature (Luinstra, 2008; Coates et al., 2004). The reaction can be expressed by:



The process efficiency (reaction rate and yield) is one of the critical aspects that has to be investigated to obtain a successful production of poly(propylene carbonate). A simplified sketch of the bench scale polymerization process, which is featured in this paper, is illustrated in

Figure 1. The process consists of an autoclave reactor with an associated recycle loop. Valves and measurement sensors are not included in the drawing. However, viscosity, density, temperature, volume flow and pressure measurement sensors are installed in the bench scale rig. The existing bench scale rig does not include a static mixer, and the aim of this paper is to study the effect of installing a mixer. The plan is to mount the static mixer on the vertical pipe in the recycle loop, as shown in the figure, to improve the mixing between CO₂ and PO. The performance of a static mixer is studied in this paper by using computational fluid dynamics (CFD) modelling and simulations.

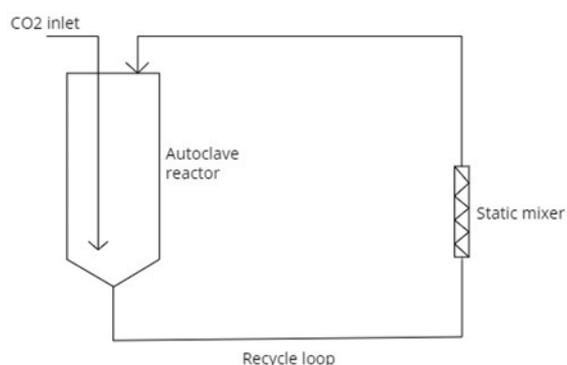


Figure 1. Draft of the bench scale rig for polymerization of poly(propylene carbonate).

3 Static mixer

The process of mixing different phases (gas/liquid, liquid/liquid, liquid/solids) with a static mixer is one of the more widely used mixing processes in chemical industries. Even so, the understanding of flow and mixing patterns within the mixers has been lacking until recently and has mainly been the property of static mixer producers. Several studies proclaim increased reaction yield and higher process efficiency when using in-line static mixers to disperse two or more fluid phases in the laminar regime (Fradette et al., 2006; Won and Kyu, 2004). Fortunately, the former ways of using mainly heuristics and know-how from experience to assess a static mixers performance is being replaced by more scientifically precise and quantifiable methods, as for example CFD (Zalc et al., 2002).

Static mixer will give an additional pressure drop in the polymerization process. The pressure drop in the static mixer can be validated by defining a dimensionless ratio Z based on the pressure drop over the elements

of an in-line mixer divided by the pressure drop of an open pipe of the same length as the mixer. This Z -value represents the increased energy required to push a fluid through a pipe with static mixer elements compared to one without such elements.

$$Z = \frac{\Delta P_{SM}}{\Delta P_{OP}} \quad (1)$$

ΔP_{SM} represents the pressure drop the fluid undergoes from the inlet of the first mixing element to the outlet of the last element in the pipe. ΔP_{OP} is the pressure loss over an open pipe with an equal length, and is calculated from the Hagen-Poiseuille equation:

$$\Delta P_{OP} = \frac{8\mu LQ}{\pi R^4} \quad (2)$$

where μ is the dynamic viscosity, L is the length of the pipe, Q is the volumetric flow rate and R is the pipe radius (Zalc et al., 2002).

In chemical processes where the temperature is a critical factor, installation of a static mixer gives a more uniform temperature distribution in addition to the improved mixing of the chemical components over the cross section area of the pipe. Non-uniform temperature distribution can potentially reduce the yield or degrade the product quality in a chemical reaction process.

There are many different static mixer designs available on the market, and the type of mixer has to be selected based on the fluids involved in the process. For mixing in a polymerization process, the most important is the capability of mixing non-Newtonian pseudo-plastic flow in the laminar regime. Figure 2 shows an image of a static mixer (Fusion Fluid Equipment).



Figure 2. Example of a static mixer (Fusion Fluid Equipment).

4 CFD set-up

This chapter will include the choices made with respect to mesh, model, method and schemes used in the simulations. The programs used for CFD in this paper are available from ANSYS and include ANSYS Design Modeler, ANSYS Meshing, ANSYS Fluent and ANSYS CFD Post. All these programs are combined into what is known as ANSYS Workbench.

4.1 Design and meshing of static mixer

ANSYS Design Modeler was used to import the STEP files of the static mixer. The files are CAD-drawings containing 3D data of the construction of the static mixer. After importing the STEP files to Design Modeler, the program was used to define a gas inlet, a liquid inlet and an outlet. This was done by defining a face at the inlet pipe and a face at the outlet pipe, effectively closing off the internal volume of the geometry. The face at the inlet pipe was split into two halves, defined as the gas inlet and liquid inlet respectively. The inlets were automatically defined as velocity inlets while the outlet was automatically defined as a pressure outlet. This is a default function in Design Modeler, which is applied by using the string “inlet” or “outlet” in the naming of the inlets and outlet respectively. After the inlets and outlet were defined, the fill function in Design Modeler was used to define the internal volume of the mixer. This internal volume represents where the fluids can be located inside the static mixer.

ANSYS Meshing was used to construct the mesh from the internal fluid domain extracted from the geometry in Design Modeler. Mostly, the default settings in ANSYS Meshing were used. CFD and Fluent were chosen as physics preference and solver preference respectively, while the element type was set to tetrahedrons. Tetrahedral elements were chosen over hexahedral and polyhedral because tetrahedrons are better suited for complex geometries. (Phoenix Analysis and Design Technologies, 2017). The size function was initially set to “adaptive” as default, but this was changed since the “adaptive” size function only generated 89000 elements, which resulted in a coarse mesh. The “proximity and curvature” size function generated 10.9 million elements, which was assumed too fine and computationally demanding. Increasing the minimum element size from the default value of 4.2906e-5 meters to 4.2906e-4 meters reduced the number of elements to 3252795 and number of nodes to 644440, which was assumed precise enough, since 3.5 million elements were used in (Zalc et al., 2002). For a mesh to be valid in representing a flow field, it must be of a certain quality. There are several properties, which indicate the quality of the mesh, depending on the element types. Some are general for all element types, such as node distribution and element smoothness. Node distribution is related to boundary layers and are important for areas of the flow where gradients are large, for example near walls. The lack of boundary layers is visible in Figure 3. This is likely one of the major contributors for potential erroneous calculations (numerical diffusion) in the mesh. Element smoothness is related to how large differences there are between the volumes of the elements for adjacent elements.

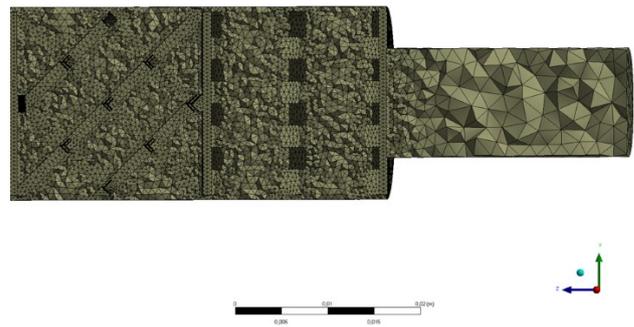


Figure 3. "Dissected" mesh

Ideally, this volume gradient between adjacent elements should be small enough to minimize potential truncation errors from the governing equations. For a tetrahedral mesh such as the one in this paper, element skewness and element aspect ratio are two important parameters. Skewness can be calculated in two different ways, but for tetrahedral elements, the equilateral-volume based skewness is calculated. The difference between the volume of an element and the volume of an equilateral element with the same circumradius is defined as the element skewness, and is expressed as: (ANSYS Inc., 2017)

$$Skewness = \frac{\text{optimal element size} - \text{element size}}{\text{optimal element size}} \quad (3)$$

Skewness is measured between 0 and 1 where 0 is equilateral and 1 is degenerate. There are several intervals between these numbers to categorize and decide if the elements comprising the mesh are good enough. These intervals are listed in Table 1.

Table 1. Skewness values and element qualities (ANSYS Inc.)

Value of skewness	Element quality	
1	Degenerate	
0.9 - < 1	Bad	
0.75 - 0.9	Poor	
0.5 - 0.75	Fair	
0.25 - 0.5	Good	
>0 - 0.25	Excellent	
0	Equilateral	

Degenerate elements will give faulty results as the equations being solved work under the assumption that the elements are approximately equilateral. Diverging calculations are typical because of such elements. A general rule says that a tetrahedral mesh should have a maximum skewness of less than 0.95 and an average skewness of less than 0.33 (ANSYS Inc., 2009). The maximum, minimum and average skewness of the mesh used in this study are presented in Table 2.

Table 2. Mesh skewness data

Minimum skewness	9.98 e-9 ≈ 0
Maximum skewness	0.84
Average skewness	0.24

The aspect ratio is the measure of how stretched an element is. The aspect ratio is defined as the ratio between the distance from an element centroid to a face centroid, and the distance from an element centroid to a node. The highest of these values is divided by the lowest to give the aspect ratio of an element. An aspect ratio of 1 means an equilateral tetrahedron and the higher the number gets, the more stretched out the tetrahedron will be. The aspect ratio of a tetrahedron, Q , is calculated as shown in Equation 4 (Cascade, 2017). The minimum, maximum and average aspect ratio of this mesh is listed in Table 3.

$$Q = \frac{h_{max}}{2*\sqrt{6}*r} \quad (4)$$

where h_{max} is the largest edge length and r is the radius of the tetrahedron.

Table 3. Mesh aspect ratio data.

Minimum aspect ratio	1.1575
Maximum aspect ratio	14.241
Average aspect ratio	1.8755

4.2 Models and input to ANSYS Fluent

After the mesh was validated with respect to skewness and aspect ratio, the next task was solving the discretized algebraic system of equations for the velocity and pressure fields throughout the mesh. This was done using ANSYS Fluent. Several choices had to be made with respect to solver, models, defining materials, boundary conditions, pressure-velocity coupling scheme and spatial discretization schemes. ANSYS Fluent has two different solvers, the pressure-based solver and the density-based solver. Generally, the pressure-based solver has been used for low-velocity incompressible flows while the density-based solver has been used for high velocity, compressible flows. The pressure-based solver was chosen for the simulations in this study. Steady state calculations were chosen because of time limitations and lack of polymerization reactions in Fluent. (Bakker, 2006) Two steady state cases were considered, which was the mixing of CO₂ and PO with 17 % and 34 % PPC respectively. The reason for using these compositions are that former reactor tests have been carried out with the same compositions, and the viscosity data from these tests were available. To different compositions were used to investigate the degree of mixing between CO₂ and PO as the amount of PPC increases in the mixture. Gravitation was included, and the gravitational acceleration was defined as -9.81 m/s² in the z-direction while the fluid flows in positive z-direction. This was

done to simulate that the static mixer is mounted vertically as is intended in the pilot reactor design. Models used in Fluent was the mixture multiphase model and the energy equation. Since both the gas inlet and the liquid inlet is defined with the same velocity, the Stokes number is assumed very low. Another reason is that the particles are gas bubbles, which tend to follow the fluid streamlines well. This should make the mixture multiphase model applicable for use in the simulations. Except for CO₂, which was predefined in the material database in Fluent, both PO and PPC had to be defined prior to the calculations. PO was defined with the properties presented in Table 4.

Table 4. PO properties (National Center for Biotechnology Information, 2018)

Density	830 kg/m ³
Specific heat	2153.93 J/kg*K
Thermal conductivity	0.134 W/m*K
Viscosity	0.00028 kg/m*s
Molecular weight	58.08 g/mole
Standard state enthalpy	-122.6 kJ/mole

Since molten PPC exhibits pseudoplastic/shear thinning characteristics, PPC was modelled with the non-Newtonian power law in Fluent:

$$\eta = k\gamma^{n-1}H(T) \quad (5)$$

n is a measure of deviation from Newtonian fluids. $n < 1$ is shear-thinning, $n = 1$ is Newtonian and $n > 1$ is shear-thickening. k is an average viscosity of the fluid, γ is the shear rate and $H(T)$ is Arrhenius law. Minimum and maximum viscosity for use in the non-Newtonian power law have to be included in Fluent. The power law inputs and other PPC properties are presented in Table 5.

Table 5. The power law inputs and other PPC properties

n	0.6 [19]
k	54.46 [7]
Density	1310 kg/m ³ [20]
Specific heat	1641.2 J/kg*K
Thermal conductivity	Same as for PO (no values were found)
Molecular weight	102 g/mole
Standard state enthalpy	Same as for PO (no values were found)

5 Results and discussion

The simulations were carried out using a mixture of CO₂, PO and PPC. The viscosity of PPC is significantly higher than the viscosity of CO₂ and PO and it is therefore important to include the PPC when simulating the static mixer. Two cases were simulated, one with 17% and one with 34% PPC in the liquid phase. The feed to the mixer included 17% and 34% PPC. The entire static mixer is comprised of 12 x-grid mixing elements. Each element has a length L , which gives the

entire mixing region a length of 12L. Figures 4, 5, 6 and 7 represent the results from the simulation with 17% PPC and show the fractions of CO₂ (to the left), PO (in the middle) and PPC (to the right) at position 1/4L, 1L, 2L and 5.25L respectively. The contour plot color scheme is explained in Table 6.

Table 6. Contour plot color scheme explanation

Color	Approximate volume fractions
Red	1
Yellow	0.75
Green	0.50
Light blue	0.25
Dark blue	0

The inlet to the static mixer is split into two equal parts defined as the gas inlet and liquid inlet respectively. CO₂ is fed into the mixer from the right, and can be observed moving from the right side of the cross-section towards the left. The PO and PPC can be seen emerging from the left side of the pipe cross section. The CO₂ is gradually mixed with the liquids along the length of the mixer.

Figure 6 shows an increase in mixing for CO₂ and PO. Both are still moving towards left and right respectively, but the largest rate of mixing for CO₂ and PO occurs from the top of the pipe towards the bottom and vice versa. This shows that the x-grid mixer mixes the fluids in both axial and radial directions. This gives an effective mixing, but also high pressure loss. The red PPC zones have all been reduced to green and blue, suggesting increased mixing compared to the mixing after 1L (Figure 5). The required mixing length to obtain a uniform mixing of CO₂, PO and PPC is estimated to be 5.25L as shown in Figure 7. Some changes in the mixing patterns from 6.5L towards 12L are observed, but they are minimal. The pressure drop through the mixer is proportional with the number of mixing elements, and to avoid high pressure drop, the number of mixing elements should not exceed the number of elements needed for the required degree of mixing. Using more than 7 mixing elements in this case would increase the pressure drop without increasing the mixing significantly. The pressure drop through the mixer in this case was 2676 Pa, whereas it could be reduced to half of 1561 Pa if a mixer with 7 elements were used.

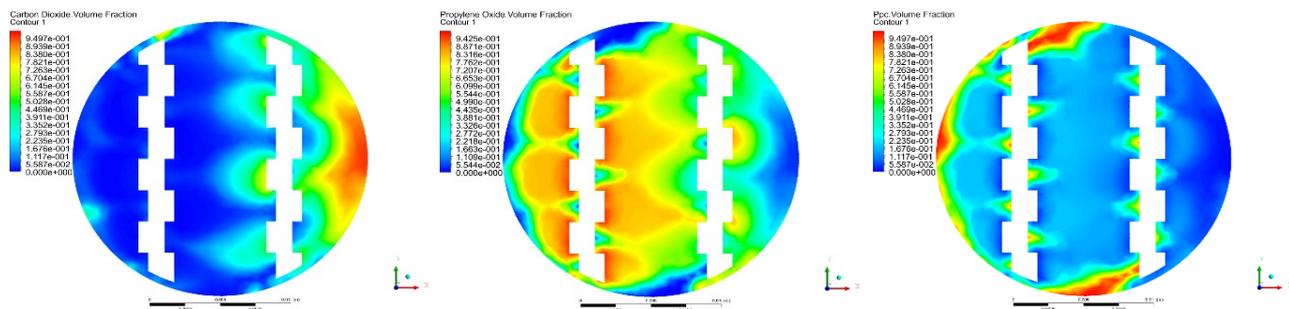


Figure 4. Contour plots of CO₂, PO and PPC respectively at 1/4 L. The feed mixture includes 17% PPC.

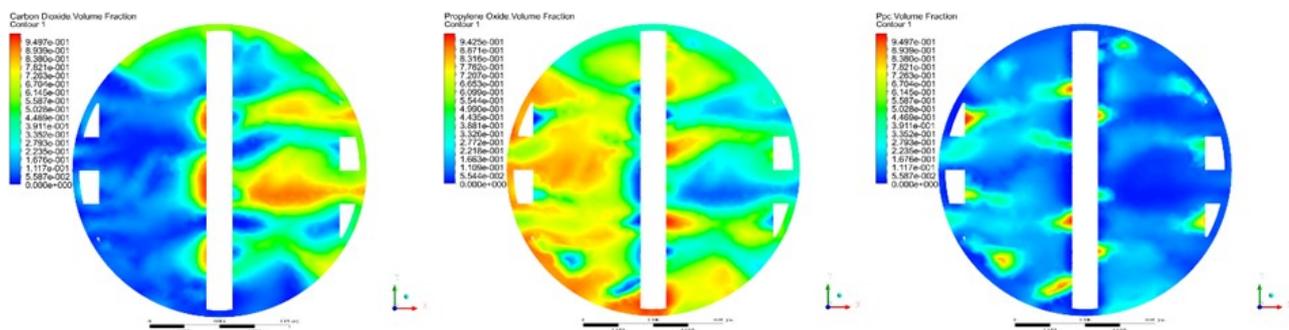


Figure 5. Contour plots of CO₂, PO and PPC respectively at 1 L. The feed mixture includes 17% PPC.

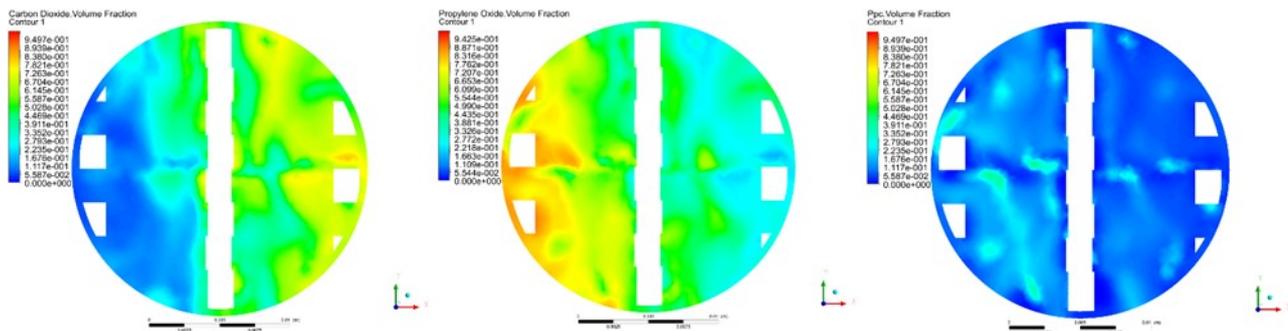


Figure 6. Contour plots of CO₂, PO and PPC respectively at 2L. The feed mixture includes 17% PPC.

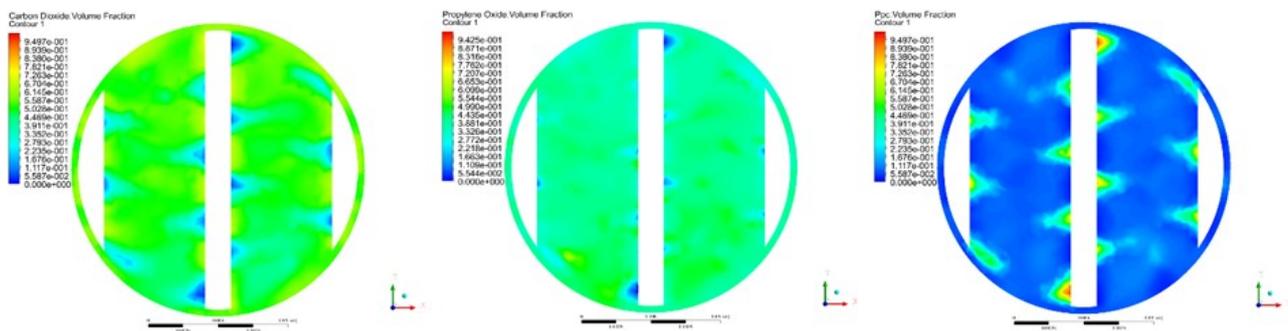


Figure 7. Contour plots of CO₂, PO and PPC respectively at 6.5L. The feed mixture includes 17% PPC

Figure 8, 9, 10 and 11 show the results from the simulation with 34% PPC in the liquid phase. The figures represent the volume fractions of the components CO₂, PO and PPC at the different locations (0.25L, L, 2L and 5.25L) in the static mixer.

As with the 17% PPC case (Figure 4), the 34% PPC case shown in Figure 8 starts off in the same way. CO₂ plumes emerging from the right towards the left. Figure 11 shows the mixer length required to get uniform mixing for the 34% PPC case. As in the case with 17% PPC there are some small differences in the mixing patterns from 5.25L to 12L, but they are too small to justify the increased pressure drop.

As mentioned earlier, the pipe cross section that leads into the static mixer was cut into two halves, the CO₂ inlet and the liquid inlet. This most likely prolongs necessary mixing length simulated in CFD compared to the real process, as both the gas and liquid phase will arrive at the static mixer premixed to some degree. This

ensures that the mixing length estimation for uniform mixing will be conservative. From all the figures, it can be observed that the pseudo-plastic PPC tends to “stick” to the pipe walls and mixing element edges. To conclude, the rate of PPC mixing into the CO₂ + PO mixture compared with the rate of CO₂ mixing with PO is low. The reasons for this can be multiple, but the major factors are probably the lack of polymerization reactions in the Fluent simulations, and the large variations in the viscosity of PPC compared to CO₂ and PO. However, the results show that good mixing will be obtained by reducing the mixing elements to five or six. The pressure drop through the mixer was calculated to 4084 Pa, and by reducing the number of mixing elements to six, the pressure drop will be reduced to 2042 Pa.

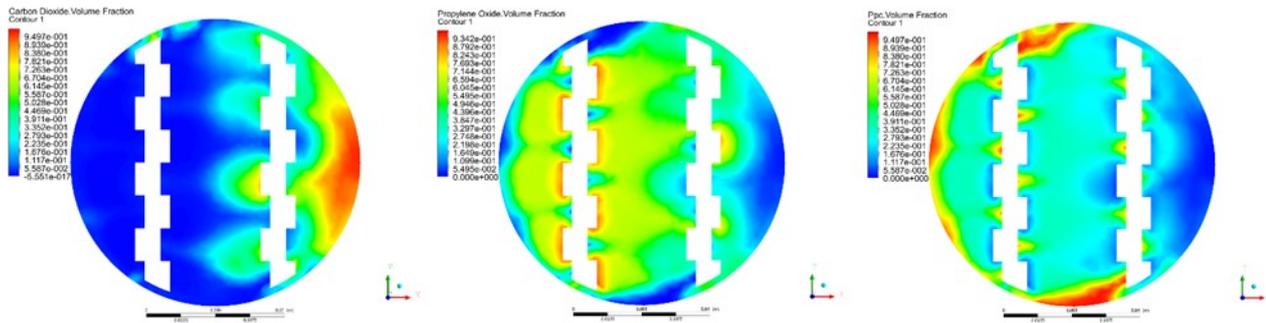


Figure 8. Contour plots of CO₂, PO and PPC respectively at 1/4 L. The mixture includes 34% PPC.

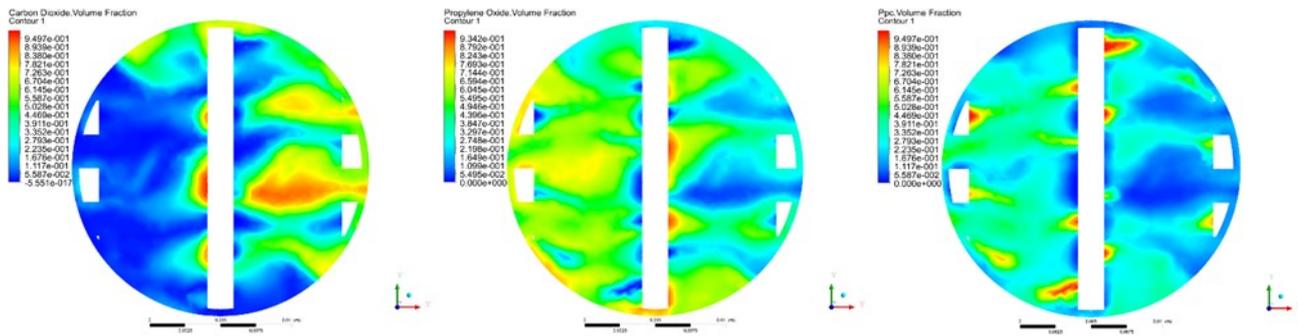


Figure 9. Contour plots of CO₂, PO and PPC respectively at L. The mixture includes 34% PPC.

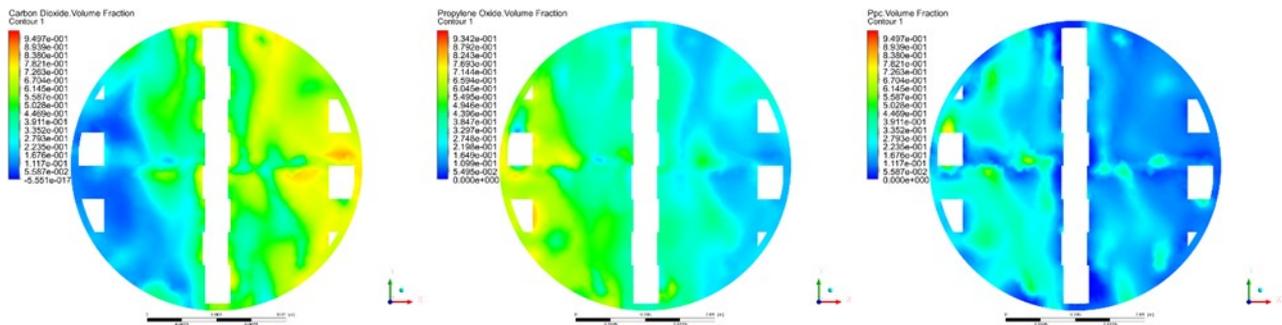


Figure 10. Contour plots of CO₂, PO and PPC respectively at 2 L. The mixture includes 34% PPC.

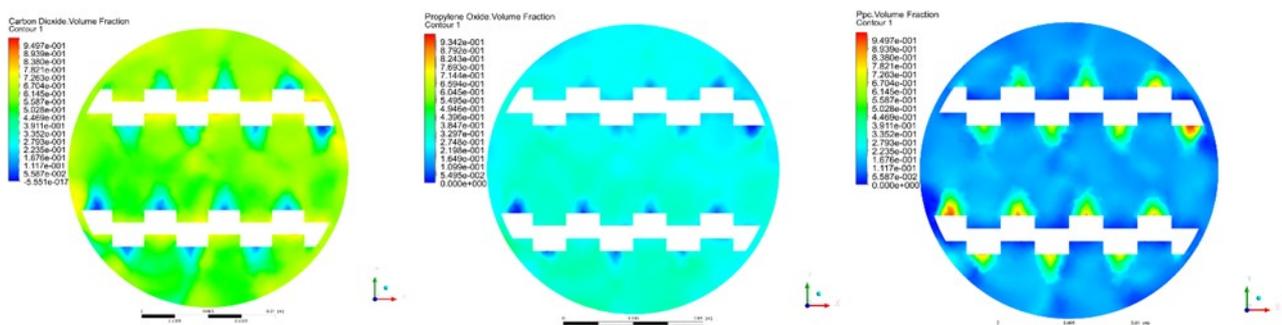


Figure 11. Contour plots of CO₂, PO and PPC respectively at 5.25 L. The mixture includes 34% PPC.

6 Conclusion

Green plastic (PPC) is produced by reacting propylene oxide with CO₂. The process includes multiphase gas-

liquid flow, and good mixing is one of the key factors to succeed in improving the process. The flow behavior of the components in the multi-phase bench scale reactor is studied in order to investigate the potential of reducing the reaction time by using a static mixer to increase mass

transfer. In this study, the CFD software ANSYS/Fluent is used to investigate the multiphase flow through a static mixer with twelve mixing elements. The static mixer is simulated to study the effect on the gas-liquid mixing. The simulations were performed with mixtures of PO and 17% and 34% PPC in the liquid phase and pure CO₂ in the gas phase. The required mixing length to obtain a uniform mixing of CO₂, PO and PPC is estimated to be 6.5L and 5.25L for the case with 17% and 34% PPC respectively. Some changes in the mixing patterns from 6.5L towards 12L are observed, but they are minimal. The pressure drop over the mixer is proportional to the length of the mixer. Using more than seven mixing elements would increase the pressure drop without increasing the mixing significantly. The pressure drop over the mixer was 2676Pa when the liquid feed consisted of 17% PPC and 4086Pa when 34% PPC was used in the liquid feed. A decrease in required mixing elements was observed when the composition of PPC in the inlet liquid phase was increased from 17% to 34%.

References

- ANSYS Inc., Skewness, 2017, https://www.sharcnet.ca/Software/Ansys/17.0/en-us/help/wb_msh/msh_skewness.html
- ANSYS Inc., Mesh Quality, 2009, <http://www.afs.enea.it/project/neptunius/docs/fluent/html/ug/node167.htm>
- Bakker André, Modeling chemical reactions with CFD, 2006, <http://www.bakker.org/dartmouth06/engs199/10-react.pdf>
- O. Cascade, Aspect ratio 3D, http://docs.salome-platform.org/latest/gui/SMESH/aspect_ratio_3d_page.html 2017.
- R. P. Chhabra, Non-Newtonian Fluids: An Introduction, Rheology of Complex Fluids, Eds. A. P. Deshpande, J. Murali Krishnan and P. B. Sunil Kumar & Springer, Munich, Chapter 1, 2010.
- G. W. Coates and D. R. Moore, "Discrete Metal-Based Catalysts for the copolymerisation of CO₂ and Epoxides: Discovery, Reactivity, Optimisation, and Mechanism". *Angew. Chem. Int.*, vol. 43, pp. 6618-6639, 2004.
- L. L. Fradette, Huai-Zhi, L. Choplin, and P. Tanguy, Gas/liquid dispersions with a SMX static mixer in the laminar regime, *Chemical Engineering Science*, 61, pp. 3506-3518, 2006.
- Fusion Fluid Equipment, <http://www.fusionfluid.com/FusionFluidEquipmentLLC/html>
- C. Gürtler, S. Grasser, A. Wolf. U.S. Patent 2013/0211042 A1.
- S. Inoue, H. Koinuma, and T. Tsuruta, "Copolymerization of carbon dioxide and epoxide," *J. Polym. Sci.*, 7, pp. 287-292, 1969.
- M. R. Kember, A. Buchard, C. K. Williams, Catalysts for CO₂/ epoxide copolymerization, *Chem. Commun*, 47, pp. 141-163, 2011.
- G. A. Luinstra, Poly(Propylene Carbonate) Old copolymers of Propylene Oxide and Carbon Dioxide with New Interests: Catalysis and Material Properties, *Polymer Reviews*, 48, pp. 192-219, 2008.
- M. Mikkelsen, M. Jorgensen, and F. C. Krebs, "The teraton challenge. A review of fixation and transformation of carbon dioxide," *Energy & Environmental Science*, vol. 3, pp. 43-81, 2010.
- S. Narang, D. Berek, S. N. Upadhyay and R. Mehta, Effect of electron density on the catalysts for copolymerization of propylene oxide and CO₂". *Journal of polymer research*, vol. 23, pp. 96, 2016, doi: 10.1007/s10965-016-0994-5.
- National Center for Biotechnology Information, *Propylene Oxide*, 2018, https://pubchem.ncbi.nlm.nih.gov/compound/Propylene_oxide
- National Institute of Standards and Technology, *Propylene Carbonate*, 2017, <https://webbook.nist.gov/cgi/cbook.cgi?ID=C108327Mask=2>
- Phoenix Analysis and Design Technologies. *Advanced Techniques in ANSYS Meshing*, 2017, https://www.padtinc.com/blog/wp-content/uploads/2017/04/Advanced-Techniques-in-ANSYS-Meshing_Blog.pdf
- Y. Tokiwa, B. P. Calabia, C. U. Ugwu, and S. Aiba, "Biodegradability of Plastics," *International Journal of Molecular Sciences*, vol. 10, pp. 3722-3742, 26/08/2009.
- S. Wang, Y. Huang, B. Liao, G. Lin, G. Cong, L. Chen, Structure and Properties of Poly(Propylene Carbonate), *International Journal of Polymer Analysis and Characterization*, Vol. 3, pp.131-143, 1997.
- J. Yoon Won and Y. Choi Kyu, "Polymerization of styrene in a continuous filled tubular reactor," *Polymer Engineering & Science*, vol. 36, pp. 65-77, 2004.
- J. M. Zalc, E. S. Szalai, and F. J. Muzzio, "Characterization of Flow and Mixing in an SMX Static Mixer," *AIChE Journal*, vol. 48, pp. 427-436, 2002.

Study of fluidized bed regimes using Computational Particle Fluid Dynamics

Rajan Jaiswal Cornelius E. Agu Rajan K. Thapa Britt M. E. Moldestad

Department of Process Energy and Environmental Technology
University College of Southeast Norway, Norway

rajanjaiswal357@gmail.com { cornelius.e.agu, rajan.k.thapa, [britt.moldestad](mailto:britt.moldestad@usn.no) }@usn.no

Abstract

The fluidization technology has a wide range of applications. In chemical synthesis, fluidized bed is used to enhance heat and mass transfer between the reacting species. More application of this technology can also be seen in pneumatic transport and circulation of solid particles. The different applications require different flow regimes. This study investigates the influence of initial bed height on the fluidized bed regime transition using the Computational Particle Fluid Dynamics (CPFD) software, Barracuda VR. The simulations are performed for a specific powder with a narrow particle size distribution. The results from the simulations are compared with the experimental data and correlations in the literature. The minimum fluidization velocity drop to a stable value and the bubbling velocity remains constant with an increase in the bed height. The gas velocity at onset of slugging decreases while that of turbulent increases to a stable value as the bed height increases.

Keywords: *Fluidized bed, transition to bubbling regime, pressure drop, bed height, minimum fluidization velocity, CPFD*

1 Introduction

1.1 Background

Fluidization technology has wide range of applications in many processes like chemical synthesis, pneumatic transportation, chemical regeneration, powder mixtures and even in the hospitals (Hargest and Artz 1969) for example, treatment of Ulcer patient. Each application requires different regimes. The types of flow regime that can be established in a fluidized bed depend on the parameters such as superficial gas velocity, particle properties, and bed dimension. The properties of fluidized bed based on these parameters have been widely studied by many researchers (Felipe and Rocha 2007; Escudero and Heindel 2011; Sarker, Rahman et al. 2012). However, the dynamic behavior of the different fluidization regimes still needs to be investigated.

Fluidization is the process where a bed of solid particles is transferred from a static solid-like state to a

dynamic fluid-like state. Fluidized beds can extend from loose bed to pneumatic conveying depending on the inlet superficial gas velocity. The superficial gas velocity at which the frictional force between the fluid and particles are counterbalanced by the weight of the bed is said to be the minimum fluidization velocity (u_{mf}). The pressure drop due to weight of the bed at this point is the maximum pressure drop (Kunii and Levenspiel 2013). With the increase in gas velocity, bubbles start to form and rise in the bed depending on the properties of particle. For a bed with smaller particles, the bed expands significantly before formation of bubbles occurs (Abrahamsen and Geldart 1980) while for a bed with larger particle diameter (Geldart B particle), the bubbles start to form as soon as the bed is fluidized. The onset of the bubbling regime occurs at the superficial gas velocity when the bubbles first appear in the bed and the corresponding velocity is called the minimum bubbling velocity (u_{mb}) (Dennis 2013). With further increase in the gas velocity, the bubble rise velocity and bubble size increase (Bauer, Werther et al. 1981). When the bubble diameter is $\sim(0.3 - 0.6) D$, the bed slugs. Where, D is the column diameter. Different types of slug can be observed, and these include axisymmetric, squared-nose and wall slug. The type of slug flow in a fluidized bed depends on particle type, particle size, bed diameter and the wall of the column (Dennis 2013; Yates 2013). Bubble and slug flow largely influence the gas and solid interaction in the fluidized bed, thus identifying onset of bubbling and slugging regime and their transition zone is crucial for the design of a fluidized bed reactor. The slug flow shifts into turbulent with further increase in gas velocity followed by random fluctuation of pressure drop. The turbulent regime is marked with the absence of bubbles and slugs in the bed and is followed by violent movement of elongated and distorted voids and particles. Increasing the superficial gas velocity beyond the bubbling velocity for fluidized beds of small particles, the fluctuation in pressure drop reaches a peak. The gas velocity at the peak pressure drop fluctuation is regarded as the critical velocity (u_c) as shown in Figure 1. Beyond u_c , the fluctuation decreases until it reaches a steady value. The minimum gas velocity at which the pressure fluctuation is relatively constant is denoted by

u_k (Yerushalmi and Cankurt 1979). Between the gas velocities u_c and u_k , the fluidized bed is in turbulent flow regime.

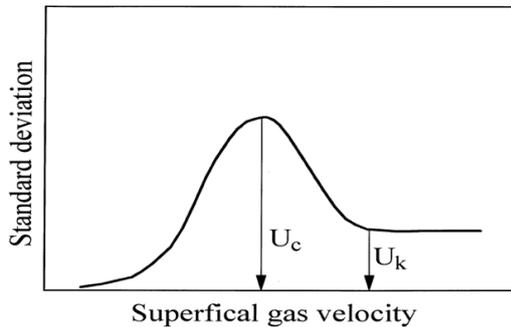


Figure 1. Amplitude of pressure Fluctuation with increase in gas velocity.

In the case of coarse particles, for instant Geldart D particles, large exploding bubbles are observed before it reaches the turbulent regime. Once the turbulent regime is established, severe gas channeling and then large-scale uniform circulation of bed material termed churning fluidization follow. Different regimes in a fluidized bed can be identified following different approaches. However, statistical method i.e. standard deviation of pressure fluctuation is the simplest and most economical (Vial, Camarasa et al. 2000).

In this work, minimum fluidization, bubbling, slug and turbulent regimes are obtained using a Computational Particle Fluid Dynamic (CPFD) model. The CPFD simulation is based on the Barracuda code, and the model is further validated using experimental results. The onset and transition of each regime are obtained from variation of pressure drop fluctuation (standard deviation) with gas velocity at five different aspect ratios. The standard deviation (σ) of pressure fluctuation is calculated from

$$\sigma = \sqrt{\frac{\sum_{i=1}^N (P_i - \bar{P})^2}{N-1}} \quad (1)$$

where, \bar{P} is the mean pressure and N is the number of data set. Similar to Figure 1, the change in standard deviation of pressure fluctuation is used to identify the onset or transition of one regime to another. The point where there is a change in the slope of the standard deviation curve marks the onset of one regime to another. This method has been used by different researches to identify the minimum fluidization velocity and quality of fluidization (PunČochÁŘ et al. 1985; Chong et al., 1987; Hong et al., 1990; Felipe and Rocha, 2007). Moreover, different fluidized bed regimes obtained using the standard deviation method have been compared with other techniques (Gourichl et al., 2006; Tchowa Medjiade et al., 2017). (Yerushalmi and

Cankurt, 1979) identified turbulent flow regimes with two transition velocities as shown in Figure 1. (Bi et al., 2000) identified transition of turbulent regime using the standard deviation method and then reported that the value of u_c for the onset of turbulent regime is higher for methods based on differential pressure measurement than those based on absolute pressure.

2 Experimental and Simulation Setup

2.1 Experimental Setup

The experimental set up used in this work consists of a 3D transparent cold bed column of height (h) 1.5 m and diameter (D) 0.084 m. A set of pressure transducers are connected to the pressure tapping points installed along the wall of the column. The distance between two consecutive pressure points along the column height is 10 cm. Compressed air at an ambient condition is supplied through an air supply hose fitted at the plenum below a porous plate distributor. The flow of air into the column is controlled by the air control valves attached to the rig. LabVIEW, a data acquisition software, is used to record the pressure drop along the column height. Figure 2 shows the experimental set up used in this work.

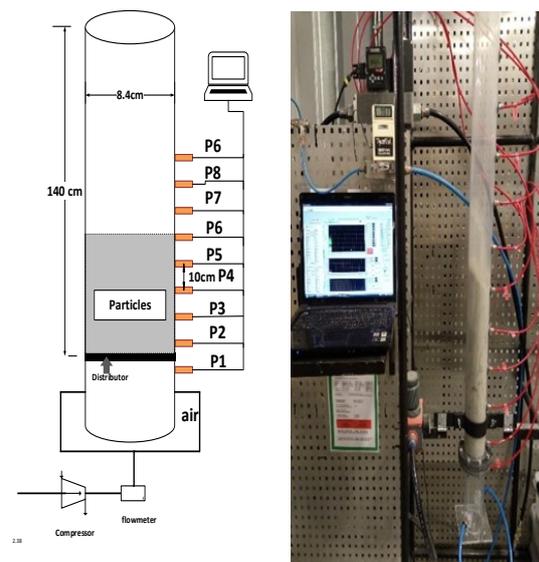


Figure 2. Experimental set-up for a cold fluidized bed.

2.2 Simulation Setup

Simulation tool such as the CPFD Barracuda code is not only used to optimize the cost, energy and time but can also be used to solve the major experimental challenges. Due to the limited height of the experimental set up, identifying the turbulent regime in the experiment is impossible. However, with the use of the Barracuda code in this study, all the regimes for sand particle based on five different aspect ratios (h_0/D): 0.7, 1, 1.5, 2, 2.5 are obtained. h_0 is the static bed height.

A cylindrical CAD geometry with the column height 300 cm and diameter 8.4 cm is imported into the

Barracuda VR software. Uniform grid of total 10000 cells is generated around the geometry for the simulations. The bottom of the column is set as inlet flow boundary condition while the top of the column is considered as the pressure boundary condition (as in the experimental set up). Thus, gas flow is uniform throughout the column with no boundary layer around the walls. The cells with volume fraction less than 0.04 and aspect ratio greater than 15:1 are removed since default grids are generated using default grid settings. The, pressure boundary conditions, CAD geometry, grid, flow, and transient data locations are shown in Figure 3.

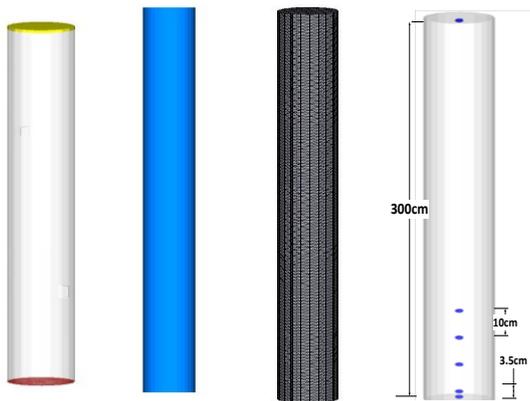


Figure 3. Simulation geometry showing from the left side the pressure boundary condition, the CAD geometry, the grid arrangement and the transient data locations.

The particle size distribution and close pack volume fraction used for the simulations are the same as in the experiments. The maximum momentum from redirection of particles collision were assumed to be 40% with the normal-to-wall and tangential-to-wall momentum retention as 0.3 and 0.99, respectively. The particle properties and operating conditions used in the simulations are summarized in Tables 2 and 3, respectively. The monitoring points were used at the middle of the column that resembles the pressure transducers height in the experiment.

Table 1. Operating conditions

Fluidizing gas	Air
Fluid temperature	Ambient (300K)
Superficial gas velocity	0.016 to 2 m/s
Static bed height	(0.7, 1, 1.5, 2, 2.5)·D
Outlet pressure	101325 Pa

Table 2. Particle and bed properties.

Sand Particles			
Particle mean diameter		234.74 μm	
Density		2650 kg/m^3	
Bulk density		1388 kg/m^3	
Solid void fraction		0.52	
Particle size distribution			
Sieve range (μm)	Mean diameter (μm)	Weight fraction	Cumulative frequency (%)
100-150	125	0.028	2.84
150-200	175	0.116	14.42
200-250	225	0.321	46.48
250-300	275	0.535	100

3 Results

3.1 Validation of CPFD model

To establish the valid model for further simulations, the simulations and experiments were carried out with sand particles of size 235 μm and aspect ratio (h_0/D) 2.5, and the minimum fluidization velocity obtained by plotting pressure drop versus superficial gas velocity are compared. Figure 4 shows the simulated pressure drop based on two different drag models: Wen and Yu and a combination of Ergun and Wen & Yu models. As compared with experimental data, the figure shows that the results from both models agree well with the experimental data when the bed is in fluidized state.

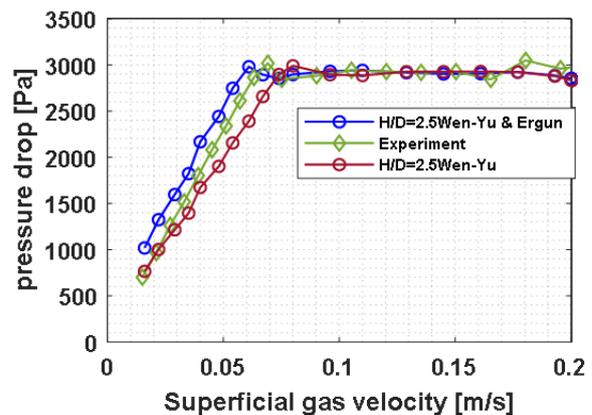


Figure 4. Pressure drop with superficial gas velocity for different bed aspect ratios.

However, while the Wen and Yu model over-predicts the gas velocity at the onset of fluidization, the combined Ergun and Wen & Yu model under-predicts the data but with a slightly lower deviation. The deviation using the Ergun and Wen & Yu and drag model is approximately 9%. Thus, the Ergun and Wen & Yu drag model is considered as acceptable to be used in the rest of the work.

The figure clearly shows that with increasing superficial gas velocity the pressure drop increases and reach a peak value identified as the maximum pressure

drop. At the maximum pressure drop, the minimum fluidization velocity U_{mf} is noted. Above U_{mf} , the bed pressure drop remains almost constant.

3.2 Onset of different fluidization regimes

For the bed aspect ratio 0.7, Figure 5 shows the variation of the simulated pressure fluctuation (standard deviation) at the middle of the bed. The pressure fluctuation remains almost zero until a superficial gas velocity denoted by U_{mb} , which describes the onset of bubbling in the bed. Beyond the value of U_{mb} , the fluctuation increases rapidly due to coalescence of bubbles. As gas velocity is increased, the bubble size increases. When the bubble grows to become a slug, the rate of change of the pressure fluctuation with the gas velocity decreases. The gas velocity at which the bed begins to slug is denoted by U_{ms} . With further increase in the gas velocity up to U_c , the gas slug explodes, leading to rapid fluctuation of the bed. For the gas velocity above U_c , the bed fluctuation decreases. The gas velocity U_c presents the onset of the turbulent flow regime. The decrease in the bed fluctuation with increasing gas velocity in the turbulent flow regime is associated with expansion of the bed where the particles are relatively far from each other. The particle separation increases with increasing gas velocity in this regime and may reach a constant value at very high velocity. For each of the regimes identified in Figure 5, the fluctuation of solids fraction in the bed is shown in Figure 6. Within the time interval 80 - 90 s for gas velocity of 0.064 m/s, it can be seen that the particle

solid volume fraction decreased from 0.52 to a value of about 0.5, marking the point of minimum fluidization condition.

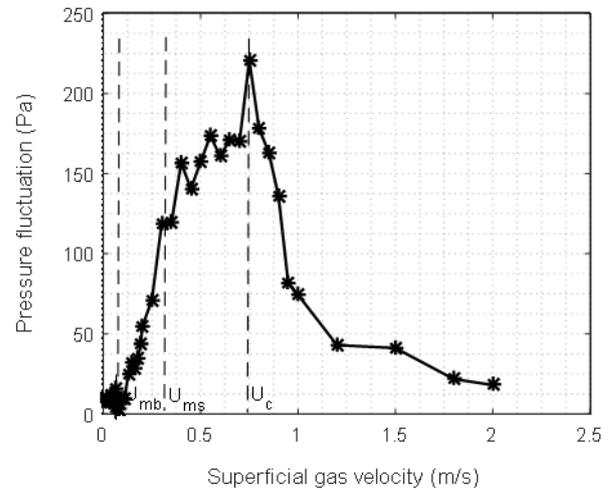


Figure 5. Pressure drop fluctuation showing the onset of different fluidized bed regimes.

The solid fraction remained constant up to 100 s and started to drop from 0.5 to 0.35 in the period of 100 - 110 s and flowrate 0.07, marking the onset of bubbling regime. Similarly, the onset of slugging regime is identified at the flowrate of 0.25 m/s and time period 190 - 200 s. The onset of turbulent regime can be confirmed at the flow rate of 0.75 m/s and at time 290 - 300 s. At this gas velocity, the fluctuation in solids fraction is vigorous, corresponding to the peak value of the pressure fluctuation in Figure 5.

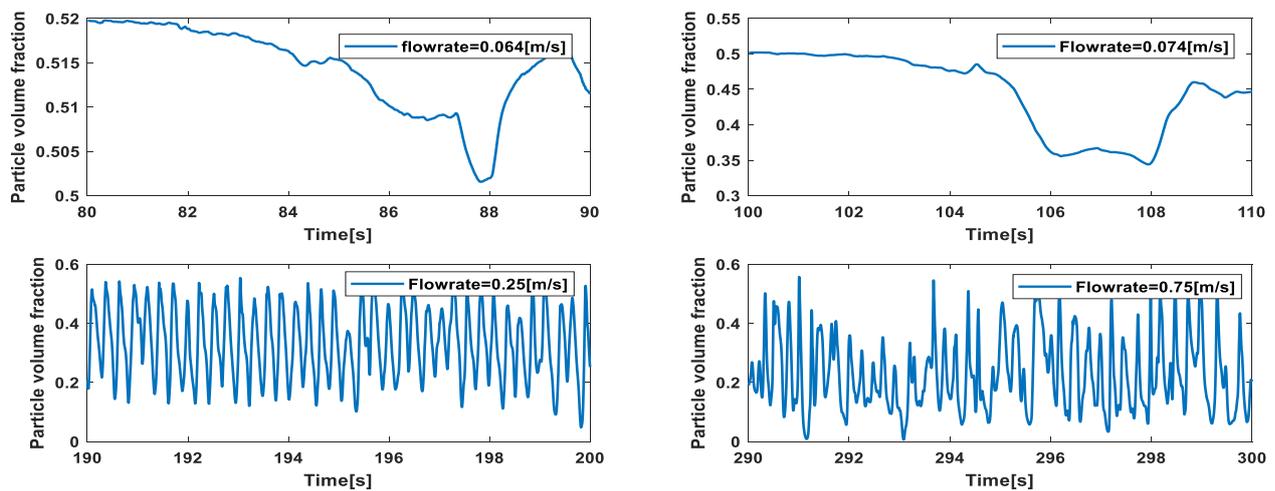


Figure 6. Change in particle volume fraction at the onset of different fluidized bed regimes (a) minimum fluidization velocity (B) minimum bubbling regime (C) minimum slugging regime (D) minimum turbulent flow regime.

3.3 Minimum fluidization and bubble velocity

The minimum fluidization velocities simulated at different bed aspect ratios are shown in Figure 7. The value of U_{mf} drops slightly from a value at a lower aspect ratio aspect ratio $h_0/D < 1.5$ to a value at a higher aspect ratio. A similar trend is also observed from the experimental results, but in a reverse order. Figure 7 also shows that the gas velocity at onset of bubbling is independent of the initial bed height. This indicates that when the bed is shallow $h_0/D < 2$, the bed begins to bubble as soon as it is fluidized. However, in a deeper bed, the result shows that the bed expands before bubbles begin to flow.

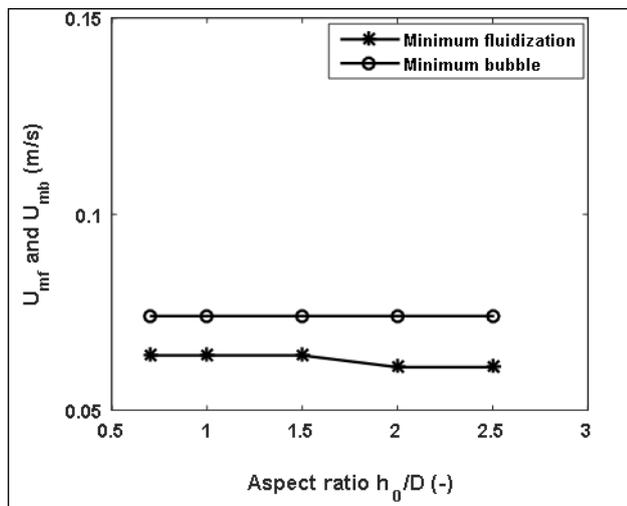


Figure 7. Variation of minimum fluidization velocity and bubbling velocity with different bed aspect ratios.

3.4 Minimum slugging velocity

The gas velocity at the onset of slugging regime is simulated for different aspect ratios. Figure 8 compares the minimum slugging velocity obtained from the CPF D simulations with the experimental data. The prediction of U_{ms} using different models are also shown. As can be seen, there is a good agreement between the simulated result and the experimental data for all aspect ratios. Similar to the experimental data, the predicted results show that the gas velocity at onset of slugging decreases with an increase in the aspect ratio.

The Baeyens and Geldart (1974) model under predicts the experimental data for all aspect ratios. The experimental data agree well with the Agu et al. (2018) model for $h_0/D < 1.5$. Particles with sphericity of 0.85 are assumed in the Agu et al. model.

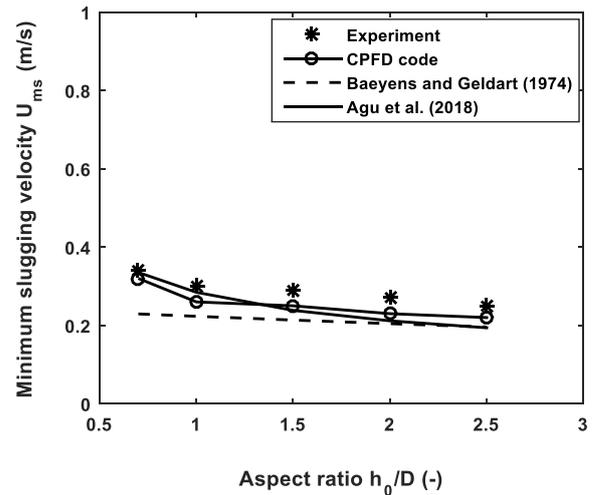


Figure 8: Variation of minimum slugging with different bed aspect ratios compared with the experimental data and different correlations

3.5 Onset of turbulent flow regime

Figure 9 shows a comparison of the simulated value of U_c with the result predicted based on Bi and Grace (1995). The simulation shows that at onset of turbulent regime, the gas velocity increases with an increase in the bed aspect ratio in the range $0 < h_0/D < 1.5$. In the deeper bed, U_c is constant, and this constant value agrees well with the results from the Bi and Grace model as can be seen.

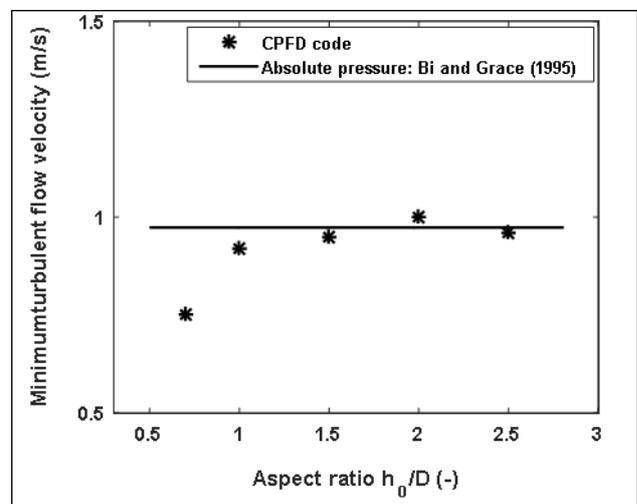


Figure 9. Variation of minimum turbulent flow velocity with different bed aspect ratios.

4 Conclusion

Fluidized bed operations are usually carried out within a given flow regime. Fluidized bed regime includes fixed bed regime, bubbling regime, slugging regime and turbulent flow regime. The static bed height may influence the transition from one regime to another, and it is therefore important to obtain how different bed

heights influences on the superficial gas velocity at the onset of each regime.

This study investigates the effect of bed height on the onset of the different fluidization regimes using the Computational Particle Fluid Dynamics (CPFD) software, Barracuda VR. The simulated results are based on sand particles with mean particle size 235 μm and are compared with experimental data and correlations available in the literature. The simulations are carried out in a 84 mm diameter bed using height to bed diameter ratios in the range of 0.7 – 2.5. In the study, the minimum fluidization velocity is obtained at the point of maximum pressure drop from the plot of pressure drop versus the gas velocity. The superficial velocities at the onset of bubbling, slugging and turbulent flow regimes for each aspect ratio are obtained from the plot of standard deviation of the pressure drop within the bed.

The result shows that the minimum fluidization velocity drop to a stable value and the bubbling velocity remains constant with an increase in the bed height. The gas velocity at onset of slugging decreases while that of turbulent increases to a stable value as the bed height increases. Comparing with the experimental data and different correlations, the agreement in the results show that the method employed in this study for identifying different fluidized bed regimes is satisfactory.

References

- A. R. Abrahamsen and D. Geldart (1980). Behaviour of gas-fluidized beds of fine powders part I. Homogeneous expansion. *Powder technology* **26**(1):35-46,1980. [https://doi.org/10.1016/0032-5910\(80\)85005-4](https://doi.org/10.1016/0032-5910(80)85005-4)
- C.E. Agu, C. Pfeifer, M. Eikeland, L. A. Tokheim and B.M.E. Moldestad. Models for predicting average bubble diameter and volumetric bubble flux in deep fluidized beds. *Industrial Engineering and Chemistry Research* **57**: 2658-2669, 2018. DOI: 10.1021/acs.iecr.7b04370
- J. Baeyens and D. Geldart. An investigation into slugging fluidized beds. *Chemical Engineering Science* **29**(1): 255-265, 1974. [https://doi.org/10.1016/0009-2509\(74\)85051-7](https://doi.org/10.1016/0009-2509(74)85051-7)
- W. Bauer, J. Werther and G. Emig. Influence of gas distributor design on the performance of fluidized bed reactor. *Ger. Chem. Eng* **4**: 291-298, 1981.
- H. T. Bi, N. Ellis, I. A. Abba and J. R. Grace. A state-of-the-art review of gas–solid turbulent fluidization. *Chemical Engineering Science*, **55**(21):4789-4825,2000. [https://doi.org/10.1016/S0009-2509\(00\)00107-X](https://doi.org/10.1016/S0009-2509(00)00107-X)
- Y. O. Chong, D. P. O'Dea, E. T. White, P. L. Lee and L. S. Leung. Control of the quality of fluidization in a tall bed using the variance of pressure fluctuations. *Powder Technology*, **53**(3):237-246, 1987. [https://doi.org/10.1016/0032-5910\(87\)80097-9](https://doi.org/10.1016/0032-5910(87)80097-9)
- J. S. Dennis. 3- Properties of stationary (bubbling) fluidised beds relevant to combustion and gasification systems. *Fluidized Bed Technologies for Near-Zero Emission Combustion and Gasification*, Woodhead Publishing: 77-148e,2013. <https://doi.org/10.1533/9780857098801.1.77>
- D. Escudero, and T. J. Heindel. Bed height and material density effects on fluidized bed hydrodynamics. *Chemical Engineering Science* **66**(16):3648-3655,2011. <https://doi.org/10.1016/j.ces.2011.04.036>
- C. A. S. Felipe and S. C. S. Rocha. Prediction of minimum fluidization velocity of gas–solid fluidized beds by pressure fluctuation measurements - Analysis of the standard deviation methodology. *Powder Technology* **174**(3):104-113,2007. <https://doi.org/10.1016/j.powtec.2007.01.015>
- B. C. Gourich, A. H. Vial, F. Essadki, M. Allam, B. Soulam and M. Ziyad. Identification of flow regimes and transition points in a bubble column through analysis of differential pressure signal-Influence of the coalescence behavior of the liquid phase. *Chemical Engineering and Processing: Process Intensification* **45**(3):214-223,2006. <https://doi.org/10.1016/j.ccep.2005.09.002>
- T. S. Hargest, and C. P. Artz . A New Concept in Patient Care: The Air-Fluidized Bed. *AORN journal* **10**(3): 50-53, 1969. [https://doi.org/10.1016/S0001-2092\(08\)70643-4](https://doi.org/10.1016/S0001-2092(08)70643-4)
- S. Hong, B. Jo, D. Doh and C. Choi. Determination of minimum fluidization velocity by the statistical analysis of pressure fluctuations in a gas-solid fluidized bed. *Powder Technology* **60**(3):215-221,1990. [https://doi.org/10.1016/0032-5910\(90\)80121-E](https://doi.org/10.1016/0032-5910(90)80121-E)
- D. Kunii and O. Levenspiel. *Fluidization engineering*, Elsevier, 2013
- M. Punčochář, J., Drahoš, J. Čermák and K. Selucký. EVALUATION OF MINIMUM FLUIDIZING VELOCITY IN GAS FLUIDIZED BED FROM PRESSURE FLUCTUATIONS. *Chemical Engineering Communications*. **35**(1-6):81-87,1985. <https://doi.org/10.1080/00986448508911219>
- R. Sarker, M. Rahman, N. Love and A. Choudhuri (2012). Effect of Bed Height, Bed Diameter and Particle Shape on Minimum Fluidization in a Gas-Solid Fluidized Bed. *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. DOI: 10.2514/6.2012-644
- W. M. Tchowa, A. R. Alvaro and A. Schumpe. Flow regime transitions in a bubble column. *Chemical Engineering Science* **170**:263-269, 2017. <https://doi.org/10.1016/j.ces.2017.04.010>
- C. Vial, E. Camarasa, S. Poncin, G. Wild, N. Midoux and J. Bouillard. Study of hydrodynamic behaviour in bubble columns and external loop airlift reactors through analysis of pressure fluctuations. *Chemical Engineering Science* **55**(15): 2957-2973, 2000. [https://doi.org/10.1016/S0009-2509\(99\)00551-5](https://doi.org/10.1016/S0009-2509(99)00551-5)
- J. Yates, (2013). *Fundamentals of Fluidized-Bed Chemical Processes: Butterworths Monographs in Chemical Engineering*, Butterworth-Heinemann.
- J. Yerushalmi, and N. T. Cankurt (1979). Further studies of the regimes of fluidization. *Powder Technology* **24**(2): 187-205. [https://doi.org/10.1016/0032-5910\(79\)87036-9](https://doi.org/10.1016/0032-5910(79)87036-9)

Simulation Model for Borehole Heat Exchangers

Petri Hietaharju¹ Tobias Boström²

¹Control Engineering, Faculty of Technology, University of Oulu, Finland, petri.hietaharju@oulu.fi

²Department of Physics and Technology, UiT, The Arctic University of Norway, Norway, tobias.bostrom@uit.no

Abstract

Seasonal thermal energy storage (STES) offers a solution to address the mismatch between production and consumption by storing the produced excess heat for later use. Borehole heat exchangers (BHEs) are one of the sensible STES technologies. In this paper, a long-term simulation model for BHEs was developed. A finite line-source model for the heat transfer outside the borehole and a quasi-3D model for the heat transfer inside the borehole were applied in two region simulation approach. Fast Fourier transformation technique together with a cubic spline interpolation method were used for faster simulation time with time varying loads and longer simulation periods. The simulation method was validated using experimental data. Results showed that the simulation model is able to accurately model ground and fluid temperature evolution.

Keywords: ground heat exchangers, boreholes, modelling, simulation, validation

1 Introduction

As renewable energy is gaining popularity, the storage of the energy is becoming increasingly important. In many times, the energy production from renewable energy sources is not aligned with the energy consumption. This situation arises for example with solar energy and waste incineration sources in cold climate conditions. Thermal energy storage (TES) systems offer a solution to address this mismatch between production and consumption by storing the produced excess heat for later use (Alva et al., 2018). Long-term TES systems are called seasonal thermal energy storages (STES) and can be further categorized into sensible, latent and chemical heat storages (Xu et al., 2014). Different sensible STES systems have been implemented in many large-scale projects since the 1980s and it is considered the most mature and reliable technology compared with the other options (Xu et al., 2014). In this work, simulation model for borehole heat exchangers (BHEs) is presented and validated. The analytical approach enables fast simulation times and flexibility to incorporate the model for system level simulations in different TES applications.

BHEs are one of the sensible STES technologies. They are the most commonly applied technology in ground source heat pump systems whose use has been significantly increasing in recent years (Lund and Boyd, 2016). To design, construct and operate BHE systems, modelling is needed (Koochi-Fayegh and Rosen, 2013). Determining the temperature of the fluid inside the borehole is the main goal of BHE modelling. Based on the fluid temperature range, the size and number of boreholes can be determined. Models can also be used to estimate the temperature rise in the ground. For reliable results, it is important that BHE model is validated against actual measurements. In many cases, field tests are used for validation, but laboratory scale tests can provide more controlled and reliable testing environment (Beier et al., 2011).

For modelling purposes, the heat transfer in BHE is usually divided into two regions: heat transfer inside the borehole and heat transfer outside the borehole. Models for these two regions are then linked by the borehole wall temperature. Many numerical and analytical approaches to modelling both regions have been suggested (Yang et al., 2010; Li and Lai, 2015). Li and Lai (2015) argue that analytical methods can be more useful than numerical methods for advancing GHE technology. Although numerical methods can be more accurate than the analytical ones, they can also be computationally inefficient and not suitable for being directly incorporated into a design or energy analysis programs. Analytical models include assumptions and simplifications that slightly reduce the accuracy of the results. However, the calculation time is also reduced, making them more suitable for long-term simulations and the algorithms are usually straightforward to implement in simulation programs.

In this work, a long-term simulation model for BHEs is developed and then validated against two different laboratory scale experimental data sets. Analytical models for heat transfer inside and outside of the borehole are applied and methods to reduce the calculation time are employed.

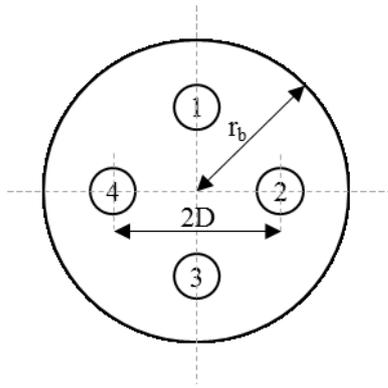


Figure 1. The cross section of a double U-tube borehole.

2 Methods

2.1 Heat Transfer Inside Boreholes

Zeng et al. (2003a; 2003b) presented a quasi-3D model for calculating the fluid temperature inside the borehole. Unlike the previous one- and two-dimensional models, this quasi-3D model takes into account the fluid temperature variation along the borehole depth. It is therefore a more accurate representation of the thermal process in the borehole heat exchanger. However, to keep the model concise and analytically manageable the convective heat flow in the grout and the ground in the axial direction is not considered.

Figure 1 shows the cross section of a borehole with a double U-tube configuration. The temperature excess in the four pipes of the U-tubes inside the borehole can be expressed as:

$$\begin{aligned} T_{f1} - T_b &= R_{11}q_1 + R_{12}q_2 + R_{13}q_3 + R_{14}q_4 \\ T_{f2} - T_b &= R_{21}q_1 + R_{22}q_2 + R_{23}q_3 + R_{24}q_4 \\ T_{f3} - T_b &= R_{31}q_1 + R_{32}q_2 + R_{33}q_3 + R_{34}q_4 \\ T_{f4} - T_b &= R_{41}q_1 + R_{42}q_2 + R_{43}q_3 + R_{44}q_4 \end{aligned} \quad (1)$$

where T_{f1} , T_{f2} , T_{f3} and T_{f4} are the fluid temperatures, T_b is the borehole wall temperature, q_1 , q_2 , q_3 and q_4 are the heat fluxes per unit length, R_{ii} ($i = 1, 2, 3, 4$) is the thermal resistance between the circulating fluid in a certain pipe and the borehole wall and R_{ij} ($i, j = 1, 2, 3, 4$) is the resistance between two pipes.

Assuming a symmetric configuration of the U-tubes $R_{ij} = R_{ji}$, $R_{ii} = R_{jj}$ and $R_{14} = R_{12}$ and so on. Based on the line-source approximation presented by Hellström (1991) three remaining resistances can be calculated as:

$$\begin{aligned} R_{11} &= \frac{1}{2\pi k_b} \left[\ln\left(\frac{r_b}{r_{p,o}}\right) - \frac{k_b - k}{k_b + k} \ln\left(\frac{r_b^2 - D^2}{r_b^2}\right) \right] + R_p \\ R_{12} &= \frac{1}{2\pi k_b} \left[\ln\left(\frac{r_b}{\sqrt{2}D}\right) - \frac{k_b - k}{2(k_b + k)} \ln\left(\frac{r_b^4 + D^4}{r_b^4}\right) \right] \\ R_{13} &= \frac{1}{2\pi k_b} \left[\ln\left(\frac{r_b}{2D}\right) - \frac{k_b - k}{k_b + k} \ln\left(\frac{r_b^2 + D^2}{r_b^2}\right) \right] \end{aligned} \quad (2)$$

where k_b and k are the thermal conductivity of the borehole and ground respectively, r_b and $r_{p,o}$ are the

radiuses of the borehole and U-tube pipe respectively, D is the distance from the U-tube pipe center to the borehole center. R_p is the heat transfer resistance from the fluid inside the U-tube and can be calculated as:

$$R_p = \frac{1}{2\pi k_p} \ln\left(\frac{r_{p,o}}{r_{p,i}}\right) + \frac{1}{2\pi r_p h_f} \quad (3)$$

where k_p is the thermal conductivity of the U-tube, subscripts o and i refer to the outer and inner radiuses of the U-tube pipe and h_f is the convective heat transfer coefficient for the fluid. First part of the Equation (3) refers to the conductive resistance of the pipe and the second part to the fluid convective resistance.

Nusselt number, which is needed for the calculation of h_f , can be calculated for turbulent flow using Dittus-Boelter equation:

$$Nu = 0.023Re^{0.8}Pr^{0.35} \quad (4)$$

where Re is the Reynolds number and Pr is the Prandtl number. Then, h_f can be calculated as:

$$h_f = \frac{k_f Nu}{2r_{p,i}} \quad (5)$$

where k_f is the thermal conductivity of the fluid.

The linear transformation of the Equation (1) leads to energy equilibrium equations for the fluid in individual pipes:

$$\begin{aligned} \pm M_f c_f \frac{dT_{f1}}{dz} &= \frac{T_{f1} - T_b}{R_{11}^\Delta} + \frac{T_{f1} - T_{f2}}{R_{12}^\Delta} + \frac{T_{f1} - T_{f3}}{R_{13}^\Delta} \\ &\quad + \frac{T_{f1} - T_{f4}}{R_{12}^\Delta} \\ \pm M_f c_f \frac{dT_{f2}}{dz} &= \frac{T_{f2} - T_{f1}}{R_{12}^\Delta} + \frac{T_{f2} - T_b}{R_{11}^\Delta} + \frac{T_{f2} - T_{f3}}{R_{12}^\Delta} \\ &\quad + \frac{T_{f2} - T_{f4}}{R_{13}^\Delta} \\ \pm M_f c_f \frac{dT_{f3}}{dz} &= \frac{T_{f3} - T_{f1}}{R_{13}^\Delta} + \frac{T_{f3} - T_{f2}}{R_{12}^\Delta} + \frac{T_{f3} - T_b}{R_{11}^\Delta} \\ &\quad + \frac{T_{f3} - T_{f4}}{R_{12}^\Delta} \\ \pm M_f c_f \frac{dT_{f4}}{dz} &= \frac{T_{f4} - T_{f1}}{R_{12}^\Delta} + \frac{T_{f4} - T_{f2}}{R_{13}^\Delta} + \frac{T_{f4} - T_{f3}}{R_{12}^\Delta} \\ &\quad + \frac{T_{f4} - T_b}{R_{11}^\Delta} \end{aligned} \quad (6)$$

where $0 \leq z \leq H$ and H is the borehole depth, M_f is the mass flow rate of the fluid, c_f is the heat capacity of the fluid and

$$\begin{aligned} R_1^\Delta &= R_{11} + R_{13} + 2R_{12} \\ R_{12}^\Delta &= \frac{R_{11}^2 + R_{13}^2 + 2R_{11}R_{13} - 4R_{12}^2}{R_{12}} \\ R_{13}^\Delta &= \frac{(R_{11} - R_{13})(R_{11}^2 + R_{13}^2 + 2R_{11}R_{13} - 4R_{12}^2)}{R_{13}^2 + R_{11}R_{13} - 2R_{12}^2} \end{aligned}$$

The sign on the left side of Equation (6) is positive for downward flow and negative for upward flow. Two conditions are necessary to solve the equations in Equation (6): when $z = 0$ the fluid temperature for the

downward flow is equal to the inlet temperature T_{in} and when $z = H$ the fluid temperature for the downward flow is equal to the fluid temperature for the upward flow. Applying these conditions, Zeng et al. (2003a; 2003b) used Laplace transform technique to solve the energy equilibrium equations in Equation (6) for the circulating fluid. As a result, the fluid temperature for the downward and upward flow in case of a single U-tube (pipes 1 and 3 or 2 and 4 in Figure 1) inside the borehole can be calculated as:

$$\begin{aligned} \theta_1(Z) &= \cosh(\beta Z) - \frac{1}{\beta S_{12}} \left[\left(\frac{S_{12}}{S_1} + 1 \right) \right. \\ &\quad \left. - \frac{\beta S_1 \cosh(\beta) - \sinh(\beta)}{\beta S_1 \cosh(\beta) + \sinh(\beta)} \right] \sinh(\beta Z) \\ \theta_2(Z) &= \frac{\beta S_1 \cosh(\beta) - \sinh(\beta)}{\beta S_1 \cosh(\beta) + \sinh(\beta)} \cosh(\beta Z) \quad (7) \\ &\quad + \frac{1}{\beta S_{12}} \left[1 - \left(\frac{S_{12}}{S_1} \right) \right. \\ &\quad \left. + 1 \right] \frac{\beta S_1 \cosh(\beta) - \sinh(\beta)}{\beta S_1 \cosh(\beta) + \sinh(\beta)} \sinh(\beta Z) \end{aligned}$$

where dimensionless parameters θ , Z , S_1 , S_{12} and β are defined as:

$$\begin{aligned} \theta &= \frac{T(z) - T_b}{T_{in} - T_b} \\ Z &= \frac{z}{H} \\ S_1 &= \frac{M_f c_f}{H} (R_{11} + R_{13}) \\ S_{12} &= \frac{M_f c_f}{H} \frac{R_{11}^2 - R_{13}^2}{R_{13}} \\ \beta &= \sqrt{\frac{1}{S_1^2} + \frac{2}{S_1 S_{12}}} \end{aligned}$$

The heat transfer rate per unit length into or from the ground can be calculated as:

$$q = \frac{(T_{in} - T_{out}) M_f c_f}{H} \quad (8)$$

In case of a double U-tube configuration inside borehole two cases for a parallel connection can be found. The fluid can flow through pipes 1 and 3 and pipes 2 and 4 (1-3, 2-4) or through pipes 1 and 2 and pipes 3 and 4 (1-2, 3-4). In both cases, Equation (7) can be used to calculate the fluid temperature but the dimensionless parameters S_1 and S_{12} are defined as

$$\begin{aligned} S_1 &= \frac{M_f c_f}{H} R_1^\Delta \\ S_{12} &= \frac{M_f c_f}{H} \frac{R_{12}^\Delta R_{13}^\Delta}{R_{12}^\Delta + R_{13}^\Delta} \end{aligned}$$

when the flow is (1-3, 2-4), and as

$$\begin{aligned} S_1 &= \frac{M_f c_f}{H} R_1^\Delta \\ S_{12} &= \frac{M_f c_f}{H} \frac{R_{12}^\Delta}{2} \end{aligned}$$

when the flow is (1-2, 3-4).

2.2 Heat Transfer Outside Borehole

Eskilson (1987) presented a numerical method to calculate g-functions that represent the dimensionless temperature response in the ground outside the boreholes by also taking into account the heat flow along the borehole depth. This was a major advancement from the previous infinite line-source model and cylindrical source model. However, the g-functions have to be calculated numerically for different borehole field configurations and is therefore time-consuming to implement in simulation programs. Based on Eskilson's (1987) model, Zeng et al. (2002) proposed an analytical method for calculating the ground temperature response. This finite line-source model was later modified by Lamarche and Beauchamp (2007) to faster calculate the ground temperature response using the integral mean borehole temperature. This model to calculate ground temperature response at time t and distance r from the borehole is shown in Equation (9) as presented by Marcotte et al. (2010).

$$\begin{aligned} \Delta T(r, t) &= \frac{q}{2\pi k_g} \left(\int_B^{\sqrt{B^2+1}} \frac{\text{erfc}(\omega z)}{\sqrt{z^2 - B^2}} dz - D_A \right. \\ &\quad \left. - \int_{\sqrt{B^2+1}}^{\sqrt{B^2+4}} \frac{\text{erfc}(\omega z)}{\sqrt{z^2 - B^2}} dz - D_B \right) \quad (9) \end{aligned}$$

Parameters B and ω in Equation (9) are defined as follows $B = r/H$, $\omega = \frac{H}{2\sqrt{at}}$ and D_A and D_B are given by:

$$\begin{aligned} D_A &= \sqrt{B^2 + 1} \text{erfc}(\omega\sqrt{B^2 + 1}) - \text{Berfc}(\omega B) \\ &\quad - \frac{e^{-\omega^2(B^2+1)} - e^{-\omega^2 B^2}}{\omega\sqrt{\pi}} \end{aligned}$$

and

$$\begin{aligned} D_B &= \sqrt{B^2 + 1} \text{erfc}(\omega\sqrt{B^2 + 1}) \\ &\quad - 0.5 \left[\text{Berfc}(\omega B) \right. \\ &\quad \left. + \sqrt{B^2 + 4} \text{erfc}(\omega\sqrt{B^2 + 4}) \right] \\ &\quad - \frac{e^{-\omega^2(B^2+1)} - 0.5(e^{-\omega^2 B^2} + e^{-\omega^2(B^2+4)})}{\omega\sqrt{\pi}} \end{aligned}$$

Marcotte and Pasquier (2008) applied fast Fourier transformation (FFT) to efficiently calculate the hourly ground temperature response with varying loads by applying the superposition principle in time (Yavuzturk, 1999). When the unit response calculated with Equation (9) ($q = 1$) and the incremental load is known at each time step, the ground temperature can be calculated by evaluating the convolution product of these two (Marcotte and Pasquier, 2008):

$$\Delta T(r, t) = \text{FFT}^{-1}(\text{FFT}(q_i)\text{FFT}(f)) \quad (10)$$

where FFT^{-1} is the inverse-FFT, q_i is the incremental load and f is the unit response calculated with Equation (9).

As can be seen from Equation (9), the calculation of the unit temperature response in the ground requires solving the integrals at each time step which can be time consuming for longer simulation periods. For 20-year simulation, Equation (9) would need to be solved $20 \times 8760 = 175,200$ times. However, Marcotte and Pasquier (2008) presented a faster method to calculate the unit temperature response for long simulation periods. In their method, the Equation (9) is only solved at certain time points that follow geometric progression and the solutions for all the other time points are obtained using cubic spline interpolant. Equation (9) is solved at times 1–48, 50, 54, 62, 78, 110, 174, 302, and so on up to, and including, the last time point. Marcotte and Pasquier (2008) reported 0.0007 °C maximum difference between the calculated and interpolated unit temperature response showing very good accuracy of the interpolation method.

2.3 Error Analysis

Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) were used to analyze the error between the measured experimental validation data and the simulation model:

$$\begin{aligned} \text{MAPE} &= \left[\frac{1}{N} \sum_{t=1}^N \left(\frac{|x(t) - y(t)|}{y(t)} \right) \right] \cdot 100\% \\ \text{MAE} &= \frac{1}{N} \sum_{t=1}^N (|x(t) - y(t)|) \\ \text{RMSE} &= \sqrt{\frac{1}{N} \sum_{t=1}^N (x(t) - y(t))^2} \end{aligned} \quad (11)$$

where x and y are the measured value and the output of the simulation model at time t respectively. Mean errors between the measured and simulated values are also reported to estimate the bias.

3 Simulation Model

The finite line-source model for the heat transfer outside the borehole and the quasi-3D model for the heat transfer inside the borehole presented in Section 2 were implemented in Matlab® environment. Parameters for the simulation model are listed in Table 1. Measured input was the inlet temperature (°C) or the heating/cooling load (W).

Water was used as default heat transfer fluid in the simulation model, but fluid properties can be easily changed if some other fluid than water is used. The heat capacity and dynamic viscosity of water were taken from tabular values based on water temperature rounded

Table 1. Parameters for the simulation model.

Parameter	Unit
Number of boreholes	-
Distance between boreholes	m
Borehole depth, H	m
Borehole radius, r_b	m
Ground thermal conductivity, k_g	W/m K
Ground thermal diffusivity, α	m ² /s
Initial ground temperature, T_0	°C
Grout thermal conductivity, k_b	W/m K
U-tube configuration	-
U-tube thermal conductivity, k_p	W/m K
U-tube inner radius, $r_{p,i}$	m
U-tube outer radius, $r_{p,o}$	m
Distance from U-tube center to borehole center, D	m
Fluid mass flow, M_f	kg/s

to the nearest integer. The thermal conductivity of the water was calculated as (Ramires et al., 1995):

$$\begin{aligned} k_f &= 0.6065 \left(-1.48445 + 4.12292 \left(\frac{273.15 + T_{in}}{298.15} \right) \right. \\ &\quad \left. - 1.63866 \left(\frac{273.15 + T_{in}}{298.15} \right)^2 \right) \end{aligned} \quad (12)$$

The simulation model can be used to calculate the outlet and ground temperature of the borehole field based on the known heating/cooling load or inlet temperature. The spatial superposition principle was applied to calculate the ground temperature response of the borehole field (Eskilson, 1987; Yavuzturk, 1999; Zeng et al., 2002). This means that the ground temperature response on a certain borehole wall is calculated by taking into account the effect of all the other boreholes. This requires solving Equation (9) for every distance between individual boreholes at each time instance. By default, for the ease and speed of computation, the simulation model is set up for a square or rectangular configuration of the borehole field where every borehole is at equal distance from each other. This allows taking advantage of the symmetry of some boreholes reducing the amount of calculations. However, the advantage of the analytical approach is that any regular or irregular borehole field configuration can be easily simulated with the model by providing borehole coordinates.

Based on the borehole field configuration, the mean unit temperature response of the ground at borehole wall for the whole borehole field is calculated with Equation (9) applying spatial superposition. For faster calculation, the cubic spline interpolation method presented in Section 2.2 is also applied. If the heating/cooling load is known, Equation (10) is used to calculate the hourly ground temperature response. Integrals are solved using *quadgk* function from Matlab®. Then an iterative search is executed to find the

proper inlet and outlet temperatures applying Equations (7) and (8). If the inlet temperature is known, an optimization algorithm *fminbnd* from Matlab® is used to find the proper heating rate and outlet temperature applying Equations (10), (7), and (8). The flow chart in Figure 2 presents the principle of the calculation procedure.

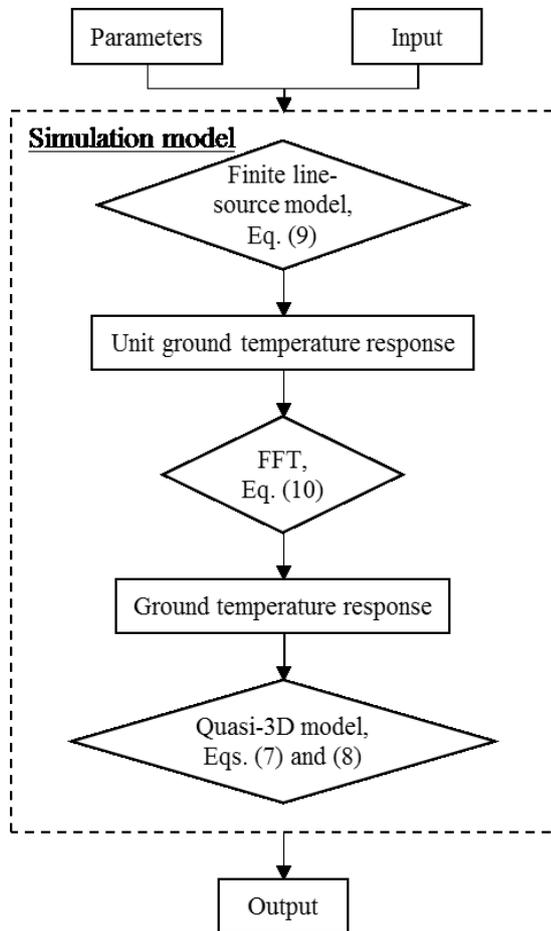


Figure 2. The principle of the calculation procedure for the BHE field.

4 Validation of the Modelling Methods

As there are no available experimental data sets for GHEs with multiple boreholes, two laboratory scale experimental data sets for single borehole systems were used for the validation of the modelling method. Data set 1 came from Shirazi and Bernier (2014), where the experimental apparatus was a small-scale laboratory tank with a single U-tube borehole. Data consisted of the measured ground temperature, supply and return temperature and flow rate recorded in ten second intervals for a 73-hour heat injection period. Data set 2 came from Beier et al. (2011), where the experimental facility was a large sandbox with a single U-tube borehole. Data set comprised minutely recorded values for the ground temperature, supply and return temperature and flow rate for 52 hours with constant

Table 2. Parameter values for data sets 1 and 2.

Parameter	Data set 1	Data set 2	Unit
H	1.23	18.3	m
r_b	0.038	0.064	m
k_g	0.29	2.82	W/m K
α	2.292×10^{-7}	8.8125×10^{-7}	m^2/s
T_0	23.5	22	$^{\circ}C$
k_b	0.35	0.73	W/m K
k_p	401	0.39	W/m K
$r_{p,i}$	0.00395	0.0137	m
$r_{p,o}$	0.00475	0.0167	m
D	0.014275	0.0265	m

heat input. The applied model parameters for both data sets are presented in Table 2. For data set 2, α was calculated using the volumetric heat capacity of 3.2×10^6 J/K m^3 as provided by Beier (2014).

First, the finite line-source model (Equation (9)) was tested to see if the model output follows the measured ground temperature. For data set 1, the ground temperature at borehole wall was calculated as a mean temperature of the sixteen temperature measurements provided at the borehole wall. The same was done with the data set 2 using four measured temperatures at the borehole wall. Equation (8) was used to calculate the

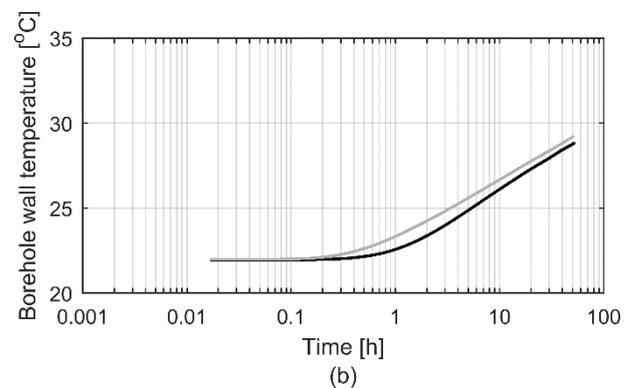
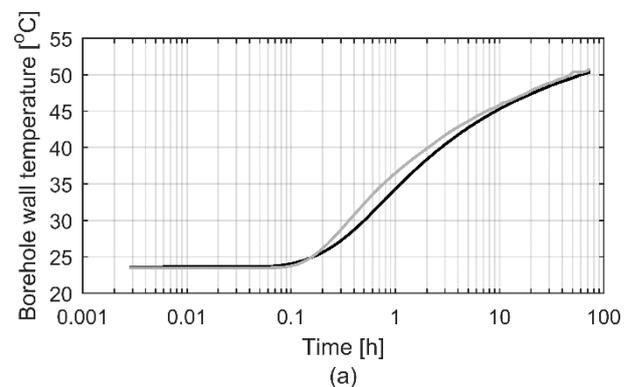


Figure 3. Modelling results on borehole wall temperature with (a) data set 1 and (b) data set 2. Black line is the mean borehole wall temperature calculated from the measured experimental data and grey line is the modelled borehole wall temperature.

Table 3. Modelling error for data sets 1 and 2.

	Data set 1	Data set 2
<i>Borehole wall temperature</i>		
MAPE (%)	1.16	1.81
MAE (°C)	0.53	0.49
RMSE (°C)	0.61	0.50
<i>Outlet temperature</i>		
MAPE (%)	0.27	0.78
MAE (°C)	0.17	0.28
RMSE (°C)	0.35	0.29

heating rate per unit length and then Equations (9) and (10) were applied to calculate the ground temperature response. The specific heat of water was assumed as 4200 J/kg K. Results are presented as a semi-log plot in Figure 3 and the modelling errors are presented in Table 3.

Mean errors for data sets 1 and 2 were -0.53 °C and -0.49 °C respectively, showing that the model slightly overestimates the ground temperature in both cases. This could be expected for the short simulation period that both data sets cover and it can be seen in Figure 3 that the error decreases in longer times. Maximum absolute errors for data sets 1 and 2 were 2.41 °C and 0.88 °C respectively. For both data sets, the maximum errors were related to early times in the modelling period. It is well known that finite line-source model is not accurate to model short term performance as the heat capacities inside the borehole are neglected by the model (Yang et al., 2010). As Eskilson (1987) presented, finite line-source model is valid approximately after time $5r_b^2/a$, which for data sets 1 and 2 were 8.8 and 6.5 hours respectively. Errors for data set 1 reach maximum around 0.5 hours and rapidly decrease after that and reaching absolute errors below 0.8 °C after 8.8 hours. For data set 2, the maximum errors occur around two hours and then slowly decrease. After 6.5 hours, absolute errors were below 0.7 °C.

Next, the modelled borehole wall temperature and measured inlet temperature and flow rate were used to calculate the outlet temperature applying the quasi-3D model (Equation (7)). As the inlet temperature was almost constant for data set 1, the dynamic viscosity and thermal conductivity for the water were assumed constants as 0.0004 Pas and 0.66 W/m K respectively. Results are presented as a semi-log plot in Figure 4 and the modelling errors can be found in Table 3.

For data set 1, the mean error was 0.16 °C. Although slightly underestimating, showing very good accuracy. For data set 2, the mean error was -0.28 °C, showing that there the model slightly overestimates the outlet temperature, although the modelling error decreases in longer times. Maximum absolute errors for data sets 1 and 2 were 30.35 °C and 1.11 °C respectively. Maximum errors for data set 1 were relatively big, but these maximum errors occurred only very early in the

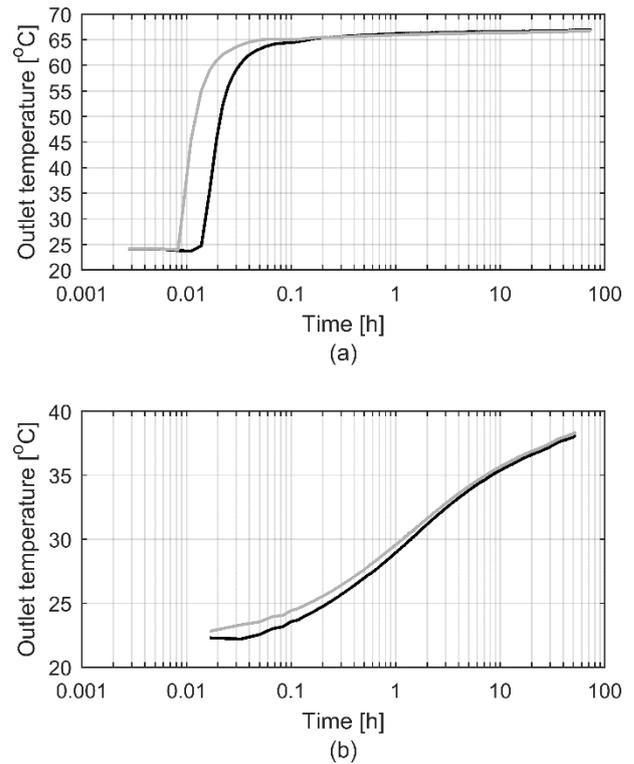


Figure 4. Modelling results on outlet temperature with (a) data set 1 and (b) data set 2. Black line is the measured outlet temperature from the experimental data and grey line is the modelled outlet temperature.

modelling period. Inlet temperature was increased rapidly in data set 1 at the start of the test and as the quasi-3D model neglects the heat capacity inside the borehole (Zeng et al., 2003b) the model overestimated the outlet temperature. However, after 9 minutes, which is much lower than the calculated validity time 8.8 hours, absolute errors were below 0.33 °C. For data set 2, absolute errors were below 0.33 °C after 6.5 hours.

The above results with two experimental data sets showed that the modelling errors were low. Taking into account the errors related to measurements, heating rate, thermal conductivities and heat capacities (Shirazi and Bernier, 2014; Li and Lai, 2013; Beier et al., 2011), it can be concluded that the selected methods for the simulation model can be applied to accurately model long-term borehole ground and fluid temperatures.

5 Conclusions

In this paper, simulation model for BHEs was developed applying finite line-source and quasi-3D models. For faster calculation time in case of time varying loads and longer simulation periods FFT technique and a cubic spline interpolation method were utilized. The simulation model was validated with two different laboratory scale experimental data sets. Results showed low modelling error with experimental data validating the correct performance of the simulation model.

The developed simulation model can be used for a long-term modelling of BHE fields consisting of multiple boreholes. It can also be applied for system level simulations by including heat pumps and different heat sources (e.g. solar collectors). Some shortcomings of the developed simulation model should be addressed in future work for wider applicability. These include adding the ability to take into account the borehole inclination and the efficient calculation of borehole fields with series connections.

References

- G. Alva, Y. Lin, and G. Fang. An overview of thermal energy storage systems. *Energy*, 144:341–378, 2018. doi:10.1016/j.energy.2017.12.037.
- R. A. Beier. Transient heat transfer in a U-tube borehole heat exchanger. *Applied Thermal Engineering*, 62(1):256–266, 2014. doi:10.1016/j.applthermaleng.2013.09.014.
- R. A. Beier, M. D. Smith, and J. D. Spitler. Reference data sets for vertical borehole ground heat exchanger models and thermal response test analysis. *Geothermics*, 40(1):79–85, 2011. doi:10.1016/j.geothermics.2010.12.007.
- P. Eskilson. *Thermal analysis of heat extraction boreholes*. PhD Thesis, University of Lund, 1987.
- G. Hellström. *Ground heat storage: thermal analyses of duct storage system*. PhD Thesis, University of Lund, 1991.
- S. Koochi-Fayegh and M. A. Rosen. A review of the modelling of thermally interacting multiple boreholes. *Sustainability*, 5(6):2519–2536, 2013. doi:10.3390/su5062519.
- L. Lamarche and B. Beauchamp. A new contribution to the finite line-source model for geothermal boreholes. *Energy and Buildings*, 39(2):188–198, 2007. doi:10.1016/j.enbuild.2006.06.003.
- M. Li and A. C. K. Lai. Analytical model for short-time responses of ground heat exchangers with U-shaped tubes: Model development and validation. *Applied Energy*, 104:510–516, 2013. doi:10.1016/j.apenergy.2012.10.057.
- M. Li and A. C. K. Lai. Review of analytical models for heat transfer by vertical ground heat exchangers (GHEs): A perspective of time and space scales. *Applied Energy*, 151:178–191, 2015. doi:10.1016/j.apenergy.2015.04.070.
- J. W. Lund and T. L. Boyd. Direct utilization of geothermal energy 2015 worldwide review. *Geothermics*, 60:66–93, 2016. doi:10.1016/j.geothermics.2015.11.004.
- D. Marcotte and P. Pasquier. Fast fluid and ground temperature computation for geothermal ground-loop heat exchanger systems. *Geothermics*, 37(6):651–665, 2008. doi:10.1016/j.geothermics.2008.08.003.
- D. Marcotte, P. Pasquier, F. Sheriff, and M. Bernier. The importance of axial effects for borehole design of geothermal heat-pump systems. *Renewable Energy*, 35(4):763–770, 2010. doi:10.1016/j.renene.2009.09.015.
- M. L. V. Ramires, C. A. Nieto de Castro, Y. Nagasaka, A. Nagashima, M. J. Assael, and W. A. Wakeham. Standard reference data for the thermal conductivity of water. *Journal of Physical and Chemical Reference Data*, 24:1377–1381, 1995. doi:10.1063/1.555963.
- A. S. Shirazi and M. Bernier. A small-scale experimental apparatus to study heat transfer in the vicinity of geothermal boreholes. *HVAC&R Research*, 20(7):819–827, 2014. doi:10.1080/10789669.2014.939553.
- J. Xu, R. Z. Wang, and Y. Li. A review of available technologies for seasonal thermal energy storage. *Solar Energy*, 103:610–638, 2014. doi:10.1016/j.solener.2013.06.006.
- H. Yang, P. Cui, and Z. Fang. Vertical-borehole ground-coupled heat pumps: A review of models and systems. *Applied Energy*, 87(1):16–27, 2010. doi:10.1016/j.apenergy.2009.04.038.
- C. Yavuzturk. *Modelling of vertical ground loop heat exchangers for ground source heat pump systems*. Master's Thesis, Technical University of Berlin, 1999.
- H. Y. Zeng, N. R. Diao, and Z. H. Fang. A finite line-source model for boreholes in geothermal heat exchangers. *Heat Transfer—Asian Research*, 31(7):558–567, 2002. doi:10.1002/htj.10057.
- H. Zeng, N. Diao, and Z. Fang. Efficiency of vertical geothermal heat exchangers in the ground source heat pump system. *Journal of Thermal Science*, 12(1):77–81, 2003a. doi:10.1007/s11630-003-0012-1.
- H. Zeng, N. Diao, and Z. Fang. Heat transfer analysis of boreholes in vertical ground heat exchangers. *International Journal of Heat and Mass Transfer*, 46(23):4467–4481, 2003b. doi:10.1016/S0017-9310(03)00270-9.

A Mean Value Model for Unsteady Gas Flows and Heat Transfer in Pipes

Olov Holmer Lars Eriksson

Vehicular Systems, Department of Electrical Engineering, Linköping University, Sweden
{olov.holmer,lars.eriksson}@liu.se

Abstract

Pipes are essential components in engines and therefore models of them are important. For example, the aftertreatment system for modern heavy-duty diesel engines consists of multiple components that are connected using pipes. The temperature in each of these components are important when determining the efficiency of the aftertreatment system and therefore models that accurately describe the temperature in the pipes between the components are important. Here, a dynamic pipe model that combines the adiabatic model of a control volume and that of a stationary one-dimensional flow with heat transfer in a pipe is developed and validated. The resulting model is a quasi-dimensional lumped parameter mean value model containing states for the temperature and pressure of the gas inside the pipe and the temperature of the pipe wall. The model uses the states and convective heat transfer models to calculate pressure at the inlet and outlet as well as temperature at the outlet, in a way that is physically correct under certain conditions. To validate the physical behavior of the model a detailed one-dimensional model is used, and to validate the practical applicability and accuracy of the model data from a passenger car gasoline engine is used to parameterize and validate the model.

Keywords: Pipe flow, Engine modeling, Heat transfer

1 Introduction

The complexity of modern powertrains are constantly increasing to meet the demands on better fuel efficiency and driveability while at the same time keep the emissions within the legislated levels. To achieve this, control systems that utilizes the complexity of the driveline are important. When designing control systems, especially for complex systems, models of the system can be very helpful when analyzing the problem as well as for speeding up testing procedures.

In engines, pipes are used to transport gas between the various components in the engine and therefore models of the are important as subcomponents in a larger model. For example, the gas conditions in the pipe connecting the engine with the aftertreatment system are important when determining the efficiency of the aftertreatment system. The efficiency of the aftertreatment system is most affected by the temperature, or energy, of the system.

There are two types of important processes, one is the heat transfer that occurs in a distributed manner along the flow and the other is compression and expansion of the gas inside the pipe. Currently there are accurate but complex distributed parameter models, GT-Power (Gamma Technologies, 2004), (Spring, 2006), etc. and there are compact control volume models for the gas compression and expansion (Hendricks, 2001; Eriksson and Nielsen, 2014) and there is a need for a compact model that can describe gas flow transients and heat transfer in pipe systems. Therefore, in this paper, a model that combines the adiabatic model of a control volume and that of a stationary one-dimensional flow with heat transfer, in a way that is energy consistent, is developed.

1.1 Method

The model is developed by combining a model for stationary one-dimensional flow from (Eriksson, 2002) with an adiabatic control volume described in (Eriksson and Nielsen, 2014). When combining the models, the different energy flows in a pipe are considered so that the energy inside the pipe is consistent with the energy flows in and out of the pipe.

To validate the model, measurements from a passenger car gasoline engine in a test stand as well as a detailed one-dimensional model are used.

1.2 Contributions

The main contribution is a new mean value model of a pipe and validation of this model using a detailed one-dimensional model and measurements from an engine.

1.3 Outline of the Paper

The paper is organized as follows: In Section 2 the different heat transfer modes in a pipe are discussed, in Section 3 a detailed one-dimensional model of a pipe is presented, in Section 4 a new mean value model is presented, in Section 5 validation of the new model is done, and finally in Section 6 conclusions are made.

2 Heat Transfer in Pipes

Heat transfer occurs in three different ways in a pipe: between the gas inside the pipe and the pipe wall, between the pipe wall and the surrounding air, and between the pipe and the components connected at the ends of the pipe. In

this section the different heat transfer modes are presented.

2.1 Gas to Wall Heat transfer

The internal heat transfer between the gas and wall, with temperatures T_g and T_w , respectively, is

$$\dot{Q}_e = h_{cv,i} A_i (T_g - T_w) \quad (1)$$

where $h_{cv,i}$ is the heat transfer coefficient. For fully developed turbulent flow the Nusselt number is often used in the following way

$$Nu = \frac{h_{cv,i} D_i}{\lambda} \quad (2)$$

where D_i is the inner diameter of the pipe and λ is the conductivity of the gas. Many empirical relations for the Nusselt number have been presented in literature (Wendland, 1993; Chen, 1993; Zhao and Winterbone, 1993; Eriksson, 2002), here the same as in (Eriksson, 2002) is used:

$$Nu = 0.48 Re^{0.5} \quad (3)$$

where Re is the Reynolds number, which for a pipe is

$$Re = \frac{4W}{\pi D \mu} \quad (4)$$

2.2 Wall to Ambient Heat Transfer

The external heat transfer from the wall to the ambient air consist both of convection and radiation in the following way

$$\dot{Q}_e = A_e (h_{cv,e} (T_w - T_{amb}) + F_v \varepsilon \sigma (T_w^4 - T_{amb}^4)) \quad (5)$$

where F_v is the viewing factor, ε is the emissivity, σ is the Stefan-Boltzmann constant, and $h_{cv,e}$ is the convective heat transfer coefficient. Since the engine is mounted in a test stand, where the ambient air is standing still, there is only natural convection. For natural convection the following relationship, from (Eastop and Mc Conkey, 1986), is used

$$h_{cv,e} = \begin{cases} 1.33 \left(\frac{T_w - T_{amb}}{D_e} \right)^{1/4}, & 10^4 < Gr \leq 10^9 \\ 1.25 (T_w - T_{amb})^{1/3}, & 10^9 < Gr < 10^{12} \end{cases} \quad (6)$$

where Gr is the Grashof number. In this case the Grashof number is calculated using

$$Gr = \frac{\beta g D_e^3 (T_w - T_{amb})}{\nu^2} \quad (7)$$

where β is the volume expansion coefficient, g is the gravitational acceleration, and ν is the kinematic viscosity of the fluid.

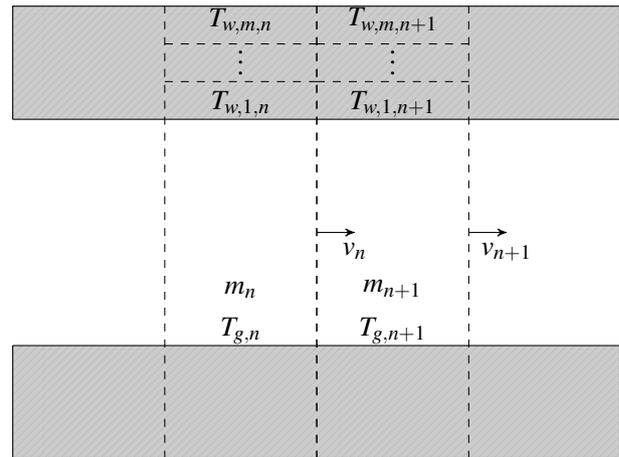


Figure 1. Illustration of how the pipe is split into segments in the one-dimensional model, some of the notation is also shown.

2.3 Heat Conduction in the Pipe Wall

Conduction occurs both radially and axially in the pipe wall. The radial conduction between two wall segments with length L_s diameter D_1 and D_2 , and temperatures $T_{w,1}$ and $T_{w,2}$ is (Holman, 1986)

$$\dot{Q}_{rad} = \frac{2\pi\lambda L_s}{\ln\left(\frac{D_2}{D_1}\right)} (T_{w,1} - T_{w,2}) \quad (8)$$

where λ is the conductivity of the wall material.

The axial conduction between two wall segment at distance L_s of each other and temperatures $T_{w,1}$ and $T_{w,2}$ is

$$\dot{Q}_{ax} = \frac{\lambda}{L_s} A_c (T_{w,1} - T_{w,2}) \quad (9)$$

where A_c is the cross section area between the segments.

3 One-Dimensional Model

In this section a detailed one-dimensional model is presented. In this model the pipe is split lengthwise into N segments. Each segment contains states for the temperature of the gas inside the segment $T_{g,n}$, mass of the gas, m_n , and the velocity of the gas flowing to the next segment, v_n . The pipe wall in the segment is split into M smaller segments with their own temperature, giving M states for the temperature of the pipe wall. An illustration of how the pipe is split into segments is shown in Figure 1, where also some of the notation is shown.

3.1 Mass Balance and Flow

In each segment, the ideal gas law

$$p_n V_s = m_n R T_{g,n} \quad (10)$$

gives us the pressure

$$p_n = \frac{m_n R T_{g,n}}{V_s} \quad (11)$$

and the density of the gas

$$\rho_n = \frac{m_n}{V_s} = \frac{p_n}{RT_{g,n}} \quad (12)$$

The mass flow from segment n to $n + 1$ is calculated by considering a plug moving from segment n to $n + 1$. The weight of the plug is

$$m_{v,n} = \frac{m_n + m_{n+1}}{2} \quad (13)$$

and the velocity, v_n , of the plug is given by

$$m_{v,n} \frac{dv_n}{dt} = A_c(p_n - p_{n+1} - \Delta p_f(v_n)) \quad (14)$$

where A_c is the cross section area of the pipe channel and $\Delta p_f(v)$ is the pressure drop due to friction in the pipe. In (Cengel et al., 2008) the relation for the pressure drop is given as

$$\Delta p_f(v) = f(Re)\rho v^2 \frac{L_s}{2D} \quad (15)$$

where f is the friction factor given by the Colebrook equation

$$\frac{1}{\sqrt{f}} = -2 \log_{10} \left(\frac{\epsilon/D}{3.7} + \frac{2.51}{Re\sqrt{f}} \right) \quad (16)$$

which can be approximated well with the following closed form expression

$$\frac{1}{\sqrt{f}} \approx -1.8 \log_{10} \left(\frac{6.9}{Re} + \left(\frac{\epsilon/D}{3.7} \right)^{1.11} \right) \quad (17)$$

Finally, the mass flow from segment n to $n + 1$ can be calculated as

$$W_n = A_c \rho_n v_n \quad (18)$$

3.2 Gas Temperature

The dynamics of the gas temperature is

$$c_v m_{g,n} \frac{dT_{g,n}}{dt} = W_{n-1}(c_v(T_{p,n-1} - T_{g,n}) + RT_{p,n-1}) - W_n(c_v(T_{p,n} - T_{g,n}) + RT_{p,n}) + \dot{Q}_i \quad (19)$$

where

$$T_{p,n} = \begin{cases} T_{g,n}, & w_{n-1} \geq 0 \\ T_{g,n+1}, & w_{n-1} < 0 \end{cases} \quad (20)$$

is the temperature of the moving plug.

3.3 Wall Temperature

The differential equations for the wall segments depend on the position of the segment. For the wall segments on the inner layer ($m = 1$) the equation is

$$\begin{aligned} \frac{dT_{w,1,n}}{dt} = & h_{cv,i,n}(T_{g,n} - T_{w,1,n}) \\ & + \dot{Q}_{ax}(T_{w,1,n+1}, T_{w,1,n}) \\ & + \dot{Q}_{ax}(T_{w,1,n-1}, T_{w,1,n}) \\ & + \dot{Q}_{rad}(T_{w,2,n}, T_{w,1,n}), \end{aligned} \quad (21)$$

for segments on the outer layer ($m = M$)

$$\begin{aligned} \frac{dT_{w,1,n}}{dt} = & h_{cv,e,n}(T_{amb} - T_{w,1,n}) \\ & + \dot{Q}_{ax}(T_{w,M,n+1}, T_{w,M,n}) \\ & + \dot{Q}_{ax}(T_{w,M,n-1}, T_{w,M,n}) \\ & + \dot{Q}_{rad}(T_{w,M-1,n}, T_{w,M,n}) \end{aligned} \quad (22)$$

and for segments that are fully inside the pipe

$$\begin{aligned} \frac{dT_{w,m,n}}{dt} = & \dot{Q}_{ax}(T_{w,m,n+1}, T_{w,m,n}) \\ & + \dot{Q}_{ax}(T_{w,m,n-1}, T_{w,m,n}) \\ & + \dot{Q}_{rad}(T_{w,m+1,n}, T_{w,m,n}) \\ & + \dot{Q}_{rad}(T_{w,m-1,n}, T_{w,m,n}). \end{aligned} \quad (23)$$

4 Mean Value Model

The mean value model combines a model of an adiabatic control volume with stationary heat transfer. The model contains three states: wall temperature, T_w , mean gas temperature, T_m , and the mass of the gas inside the pipe, m_g . In this section the details of the model are presented.

4.1 Wall Temperature

The dynamics of the pipe wall temperature is

$$c_w m_w \frac{dT_w}{dt} = \dot{Q}_i + \dot{Q}_e \quad (24)$$

where

$$\dot{Q}_{e,n} = A_o (h_{cv,e}(T_w - T_{amb}) + F_v \epsilon \sigma (T_w^4 - T_{amb}^4)) \quad (25)$$

is the external heat transfer, and

$$\dot{Q}_{i,n} = h_{cv,i} A_i (T_m - T_w) \quad (26)$$

is the internal heat transfer.

4.2 Mean Gas Temperature

The change of energy of the gas inside the pipe, if we consider the pipe as an adiabatic control volume (Eriksson and Nielsen, 2014), is

$$\dot{Q} = c_v (W_{in} T_{in} - T_m) + R(T_{in} W_{in} - W_{out} T_{out}) - \dot{Q}_i \quad (27)$$

and this means that the change in mean temperature, T_m , can be calculated as

$$c_p m_g \frac{dT_m}{dt} = c_v (W_{in} T_{in} - T_m) + R(T_{in} W_{in} - W_{out} T_{out}) - \dot{Q}_i \quad (28)$$

4.3 Mass Balance

The mass of the gas in the pipe m_g is calculated using

$$\frac{dm_g}{dt} = W_{in} - W_{out} \quad (29)$$

where W_{in} and W_{out} is the mass flow in and out of the pipe, respectively.

4.4 Temperature Drop Along the Pipe

During stationary conditions, the temperature of the gas at position l down the pipe is (Eriksson, 2002)

$$T(l) = T_w + \Delta T e^{-kl} \quad (30)$$

where $\Delta T = T_{in} - T_w$ and

$$k = \frac{\pi D_i h_{cv,i}}{W c_p} \quad (31)$$

here, because of the stationary conditions, $W = W_{in} = W_{out}$. During non-stationary conditions, however, $T(l)$ can essentially be arbitrary, depending on previous conditions, under the condition that the energy in the pipe must be consistent i.e.

$$c_p m_g T_m = \int_0^L c_p \frac{m_g}{L} T(l) dl \quad (32)$$

The assumption that $T(l)$ have the same exponential structure as in (30), but with a ΔT that makes (32) hold, results in

$$\Delta T = \frac{kL(T_m - T_w)}{1 - e^{-kL}}. \quad (33)$$

During non-stationary conditions it does not hold that $W = W_{in} = W_{out}$ and instead $W = \frac{W_{in} + W_{out}}{2}$ is used.

4.5 Transport Delay

The fact that the gas moves with a velocity

$$v = \frac{W}{\rho A_i} = \frac{WRT}{\rho A_i} \quad (34)$$

in a pipe of length L gives rise to a transport delay

$$\tau_d = \frac{L}{v} = \frac{LA_i p}{WRT} = \left/ \frac{LA_i = V}{pV = mRT} \right/ = \frac{m}{W}. \quad (35)$$

from the inlet to the outlet. This transport delay can be included in the model by using a time delayed version of T_m when calculating the output temperature.

5 Validation and Comparison

In this section the new mean value model is validated in three different ways: first the stationary cooling of the gas is validated using stationary measurements, then the gas mixing dynamics are validated using the one-dimensional model, and finally the model is validated using dynamic measurements from an engine. The engine on which the measurements were made is a turbocharged 2 liter inline 4-cylinder gasoline engine. The measurements in this paper have been done on the pipe between the compressor and the intercooler. This is a pipe that is approximately 2 m long, has a diameter of approximately 5 cm, and is made of rubber and plastics. The pipe is also equipped with temperature and pressure sensors at both ends.

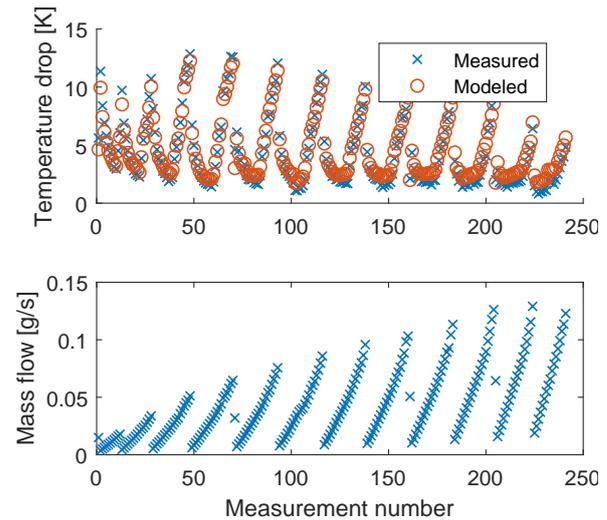


Figure 2. Measured and modeled stationary temperature drop between the compressor and intercooler. The upper figure shows the temperature drop and the lower shows the mass flow.

5.1 Stationary Cooling

In (Eriksson, 2002) the stationary cooling for exhaust pipes are validated and the principles for a pipe on the intake side are the same as on the exhaust pipes. However, the assumption of constant wall temperature might be less accurate since the pipes are made out of different materials with less conductivity. Therefore the model is here validated for pipes on the intake side of the engine. In Figure 2 the temperature drop in the pipe between the compressor and intercooler is shown for different mass flows. During stationary conditions the mass flow is the same throughout the whole intake side and therefore the mass flow is taken directly from the mass flow sensor, which is mounted slightly before the compressor. As can be seen the model predicts the pressure drop well, so it can be concluded that the model performs well in stationary conditions and the assumption of constant wall temperature is a reasonable assumption on the intake side as well.

5.2 Comparison with One-dimensional Model

The gas mixing has a time constant in the order of seconds. This makes it hard to measure it with temperature sensors, since they have a time constant that is similar or larger. Therefore, to validate these dynamics, simulations from the one-dimensional model are used. Two different simulations were done, one were a step in input temperature was made and one were a step in the input flow was made.

In Figure 3 the result from the step in temperature is shown. The step is made after 10 seconds and as expected the big change in output is delayed since it takes some time for the new temperature to propagate through the pipe. However, the one-dimensional model shows a small rise in temperature directly after the step, this is because the pressure in the pipe increases with the temperature and

this causes a temperature change that moves much quicker than the velocity of the gas, the same effect is present in the mean value model but is barely visible. It can also be seen that the mean temperature of the two models are the same before and after the step, but during the transient they differ. The reason for this is that the assumption of exponential decay of the temperature is not valid and therefore the heat transfer in the mean value model is not correct during transients, however, the difference is not that big.

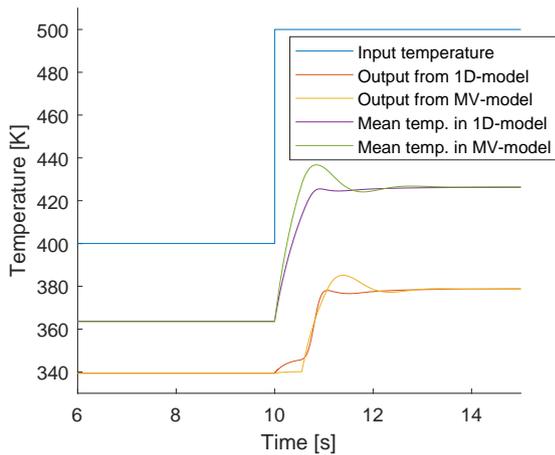


Figure 3. Output temperature and mean temperature from the one-dimensional and mean value model when a step is made in the input temperature.

Figure 4 shows the result when the step in input flow is made. Here it again can be seen that the temperature rises without a time delay, and again this is because the change in pressure is not affected by the transport delay. Here it is more evident that the mean value model does not describe this well, there is a small increase in temperature at the time of the step but most of the temperature change is delayed in the mean value model. The small initial increase in temperature in the mean value model is because of that the mass flow used to calculate the heat transfer along the pipe is the mean of the input and output flow, and since the input flow increases the mean flow increases which reduces the heat transfer to the pipe wall.

In the two examples above a step was used as input signal and that type of drastic change really tests the dynamic behavior of the model. Since this type of drastic change are not likely to occur in an engine, this could be considered to be outside the region where the mean value model is designed to be used. In Figure 5 the same step in temperature as in Figure 3, but now filtered with a time constant of 1 s, is used as input to the models. When using this input, with a smaller rate of change, it can be seen that the agreement between the models is better.

5.3 Dynamic Measurements

Here dynamic measurements where a step in wastegate position was made to increase the mass flow through the

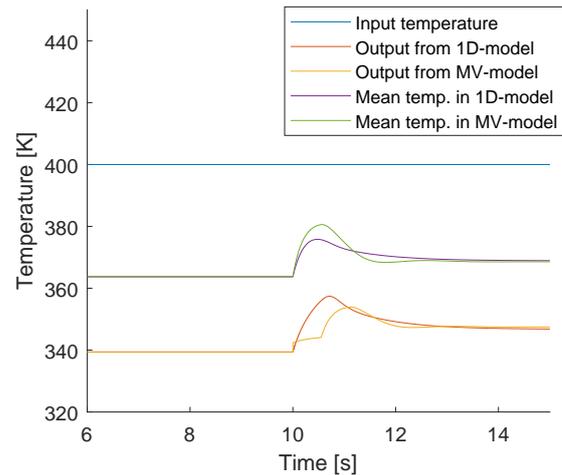


Figure 4. Output temperature and mean temperature from the one-dimensional and mean value model when a step is made in the input flow.

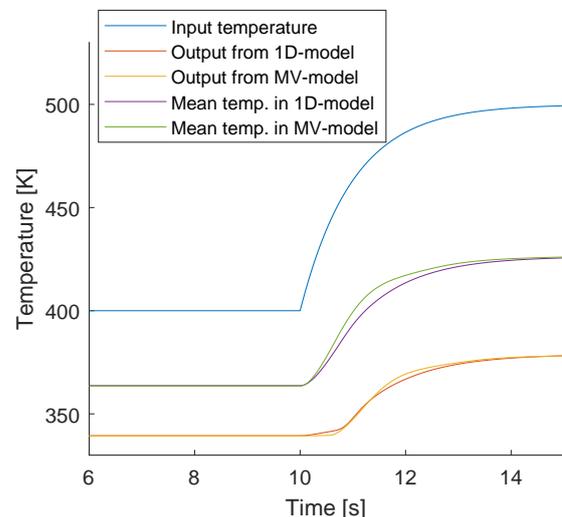


Figure 5. Output temperature and mean temperature from the one-dimensional and mean value model when the input temperature is a step filtered with a time constant of 1 s.

compressor and increase the temperature after the compressor are used to validate the model. The temperature after the compressor was used as input to the model and the temperature before the intercooler was used as validation data. During transients the mass flow sensor can not be used directly since it is mounted at some distance from the pipe. Therefore the mass flow sensor was only used to determine the stationary levels before and after the step, during the step the mass flow in and out of the pipe was assumed to move between these two levels in the following way: the input flow was assumed to increase linearly with the compressor angular velocity and the output flow was assumed to increase linearly with the pressure drop over the intercooler.

It was also discovered that there were some additional

dynamics, with a time constant of around 15 s, present in the measurements. These additional dynamics could come from the dynamics of the temperature sensor. However, the time constant might be somewhat too big to be explained solely by this, therefore further investigation is needed to investigate this phenomenon.

In Figure 6 the measured and modeled output when the sensor dynamics have been included by filtering the output from the model with the time constant of sensor is shown. As can be seen the model seems to agree well with the measurements.

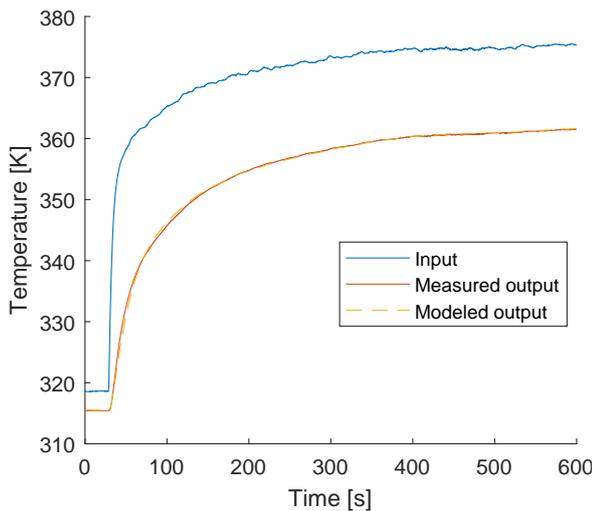


Figure 6. Dynamic measurements from the engine and the output from the model. Here the dynamics of the sensor is included by filtering the output from the model with a time constant of 15 s.

Figure 7 shows the same test as in Figure 6, but now one simulation without the wall temperature dynamics and one without the sensor dynamics is also shown. Here it becomes clear that both these dynamics are needed to capture the dynamics of the system, the sensor dynamics are needed to reduce the initial rate of change and the wall temperature dynamics are needed to capture the slower rate of change towards the stationary level.

6 Conclusions

A dynamic pipe model that combines the adiabatic model of a control volume and that of a stationary one-dimensional flow with heat transfer in a pipe has been developed and validated. The validation has been done using both measurements from an engine in a test stand as well as simulations from a detailed one-dimensional model. The model has shown to agree well with the measurements from the engine and the simulations from the one-dimensional model.

6.1 Future Work

Interesting future work would be to try to separate temperature changes in the inlet temperature from changes in

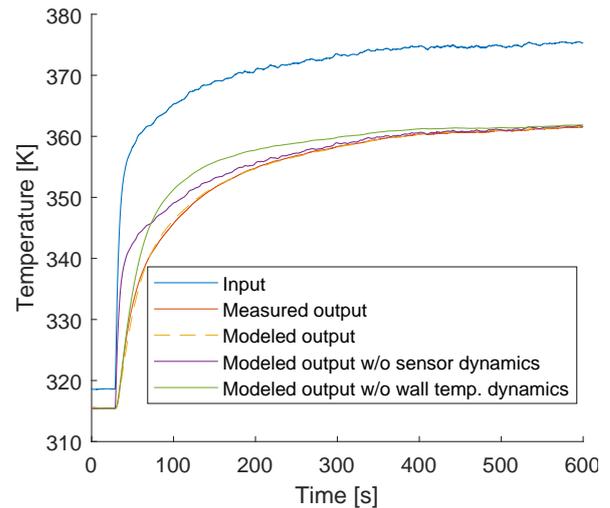


Figure 7. Dynamic measurements from the engine and the output from the model. Here the output with and without the sensor dynamics are shown as well as the output when the wall temperature dynamics are removed.

temperature due to compression of the gas, since temperature changes from compression are not affected by the transport delay that changes in inlet temperature are affected by.

References

- Yunus A Cengel, Robert H Turner, John M Cimbala, and Mehmet Kanoglu. *Fundamentals of thermal-fluid sciences*. McGraw-Hill New York, NY, 2008.
- David KS Chen. A numerical model for thermal problems in exhaust systems. Technical report, SAE Technical Paper, 1993.
- Thomas D Eastop and Allan Mc Conkey. *Applied thermodynamics for engineering technologies*. 1986.
- Lars Eriksson. Mean value models for exhaust system temperatures. Technical report, SAE Technical Paper, 2002.
- Lars Eriksson and Lars Nielsen. *Modeling and control of engines and drivelines*. John Wiley & Sons, 2014.
- Gamma Technologies. GT-Power User's Manual. *GT-Suite Version 6.1*, 2004.
- Elbert Hendricks. Isothermal vs. adiabatic mean value si engine models. *IFAC Proceedings Volumes*, 34(1):363–368, 2001.
- JP Holman. Heat transfer, 1986. *Mc Gran-Hill Book Company, Southern Methodist University*, 1986.
- Peter Spring. *Modeling and control of pressure-wave supercharged engine systems*. ETH Zurich, 2006.
- Daniel W Wendland. Automobile exhaust-system steady-state heat transfer. Technical report, SAE Technical Paper, 1993.
- Y Zhao and DE Winterbone. A study of warm-up processes in si engine exhaust systems. Technical report, SAE Technical Paper, 1993.

Validation of an Open-Source Mean-Value Heavy-Duty Diesel Engine Model

Kristoffer Ekberg Viktor Leek Lars Eriksson

Vehicular Systems, Department of Electrical Engineering, Linköping University, Sweden,
{kristoffer.ekberg, viktor.leek, lars.eriksson}@liu.se

Abstract

The pursuit for lower fuel consumption and stricter emission legislation has made a simulation- and optimization-based development methodology important to the automotive industry. The keystone in the methodology, is the system model. But for the results obtained using a model to be credible, the model has to be validated. The paper validates an open-source, mean-value engine model of a 12.7 liters CI inline 6 cylinder heavy-duty engine, and releases it as open-source.

Keywords: Mean-value engine model, 0-D modeling, Simulation, Optimal control

1 Introduction

In today's automotive industry, there is a drive for lower fuel consumption and better control of emissions. Simulation and numerical optimization are two tools that can be used to achieve that. The keystone in a simulation driven approach, is the system model. For it (and the results obtained using it) to be credible, it needs to be validated. This paper presents the model structure of an open-source engine model and validates it.

The engine is a 12.7 liters CI inline 6 cylinder Scania engine. The model structure is a mean-value engine model (MVEM) (Heywood, 1988), this type of structure does not model the piston movement explicitly. Instead the mean flow in and out of the cylinders is modeled. This makes the model computationally efficient, and suitable for control and optimization of the air and fuel path of the engine. It has a minimal number of states for efficient simulations, and is continuously differentiable for use with gradient-based optimization solvers.

The model is a continuation of previous modeling work by the authors, described in Ekberg et al. (2017). In contrast to the previous work, *indicated mean effective pressure* (IMEP), *pump mean effective pressure* (PMEP) and *friction mean effective pressure* (FMEP), have been remodeled, the compressor model has been changed for a more advanced model presented in Llamas and Eriksson (2017), the turbine has been remodeled to better describe the efficiency, and the model is validated as a complete system.

The purpose of this paper is to provide an open model to the research community, and to invoke confidence in the

model by validating it. The paper is outlined as follows. In Section 2 the data sets are presented, in Section 3 the estimation criterion is presented, in Section 4 the model equations are presented and the sub-models are validated, in Section 5 all submodels are connected and the complete modeled is tuned and validated against measurement data, and in Section 6 the model is simulated and its basic simulation properties are presented.

1.1 Contribution

The main contribution of this paper is a validated open-source MVEM of a Scania 12.7 liters 6 cylinder engine, downloadable from www.fs.isy.liu.se/Software/. Also, new component models for the engine torque, and an adaptation of an existing turbine model are introduced.

2 Data

Six datasets have been used for modeling and validation, see Table 1. Dataset A is dynamometer data of the engine, and is the primary modeling set. Dataset B is simulation data from a model of the same engine obtained using a detailed model implemented in GT-Power (Gamma Technologies, 2004). In B, the air-to-fuel ratio is varied for the operating points and is primarily used to develop the torque model. Dataset C is used to model the compressor, and dataset D is used for modeling the turbine. Dataset E is used to validate the complete model, and dataset F is used for throttle modeling.

Table 1. Datasets used to find model parameters.

Dataset	Signals	Samples
A. Stationary measurement data	24	235
B. GT Power Simulation	22	160
C. Compressor map	4	73
D. Turbine map	4	38
E. Dynamic measurement data	24	69598
F. Throttle effective area	1	11

3 Parameter Estimation

To parameterize the sub-models, the following criterion is used:

$$\theta^* = \arg \min_{\theta} \sum_{k=1}^K e_k^2(\theta) \quad (1)$$

where θ^* are the criterion minimizing parameters, e_k the model error at sample k , and K the number of samples.

Some of the submodels are not modeled using dataset A, which is the desired model behavior at steady-state. To compensate for this, the parameters need to be refitted to represent dataset A. This is done using a regularization technique, which is selected in order to preserve the model structure, according to:

$$\theta^* = \arg \min_{\theta} \left(\sum_{k=1}^K e_k^2(\theta) + C \sum_{i=1}^I (\mu_i \frac{\theta_i^* - \theta_i}{\theta_i^*})^2 \right) \quad (2)$$

where θ^* are the parameters refitted to dataset A, θ^* the parameters obtained using (1) and a dataset different than A, I the numbers of parameters to refit, μ_i the weight assigned to parameter i , and C the cost of changing the parameters.

To evaluate the model fit, the following measures are used:

$$\bar{e} = \frac{1}{K} \sum_{k=1}^K |e_k| \quad (3)$$

$$e_{\sigma} = \sqrt{V[|e_k|]} \quad (4)$$

$$\bar{e}_{\text{rel}} = \frac{1}{K} \sum_{k=1}^K \left| \frac{e_k}{s_k} \right| \quad (5)$$

$$e_{\text{rel},\sigma} = \sqrt{V[|e_k/s_k|]} \quad (6)$$

where \bar{e} is the mean absolute error, e_{σ} the variance of the absolute error, \bar{e}_{rel} the relative error, s_k the k :th sample, and $e_{\text{rel},\sigma}$ the variance of the relative error. The variance was calculated using the command `var` in Matlab R2017b.

4 Model

In the presentation of the validated model structure, the time dependence, (t) , is used to distinguish variables from constants, dependence on other variables is omitted for notational simplicity.

The model has four states $x(t)$:

$$x(t) = [p_c(t), p_{\text{im}}(t), p_{\text{em}}(t), \omega_{\text{tc}}(t)] \quad (7)$$

where p_c is the pressure after the compressor, p_{im} the intake manifold pressure, p_{em} the exhaust manifold pressure, and ω_{tc} the turbocharger angular velocity. The model further has three control inputs $u(t)$:

$$u(t) = [u_f(t), u_{\text{thr}}(t), u_{\text{wg}}(t)] \quad (8)$$

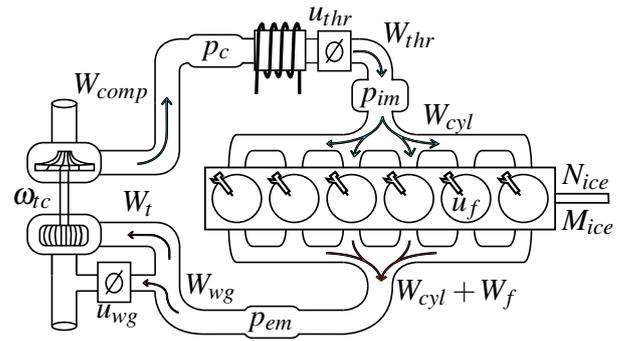


Figure 1. Model overview. Shown are the four states: pressure after the compressor p_c , intake manifold pressure p_{im} , exhaust manifold pressure p_{em} and turbocharger angular velocity ω_{tc} , and the three control inputs: Fuel injection per cycle u_f , throttle effective area u_{thr} , and wastegate effective area u_{wg} .

where u_f is the fuel mass injected per cycle, u_{thr} the throttle position, and u_{wg} the wastegate position. The engine speed, N_{ice} , is treated as an exogenous input into the system. Figure 1 shows an overview of the model, where the states and control inputs are visualized.

4.1 Control Volumes

The volume after the compressor, intake manifold and exhaust manifold are seen as thermodynamic control volumes, and modeled as dynamic states with filling and emptying dynamics. Using an isothermal model and by assuming mass conservation, and constant c_p and c_v , the models are described as:

$$\frac{d}{dt} p_c(t) = \frac{R_a T_c(t)}{V_c} (W_c(t) - W_{\text{thr}}(t)) \quad (9a)$$

$$\frac{d}{dt} p_{\text{im}}(t) = \frac{R_a T_{\text{im}}}{V_{\text{im}}} (W_{\text{thr}}(t) - W_{\text{cyl}}(t)) \quad (9b)$$

$$\frac{d}{dt} p_{\text{em}}(t) = \frac{R_e T_{\text{em}}(t)}{V_{\text{em}}} (W_{\text{cyl}}(t) + W_f(t) - W_t(t) - W_{\text{wg}}(t)) \quad (9c)$$

4.1.1 Parameters

There are four parameters to estimate: volume after the compressor V_c , intake manifold volume V_{im} , exhaust manifold volume V_{em} , and intake manifold temperature T_{im} .

4.1.2 Parametrization and Validation

In dataset A, the mean value of the measured temperature in the intake manifold is 304.00 K with a standard deviation of 1.11 K, indicating that a constant temperature in the intake manifold is an acceptable modeling assumption. The charge air cooler is assumed to be ideal, therefore the temperature in the intake manifold equals the surrounding temperature, $T_{\text{im}} = T_{\text{amb}}$. The sizes of the volumes V_c , V_{im} and V_{em} are tuned until the dynamics of the model comply with the measurements. Validation of the volume sizes is seen in section 6.

4.2 Throttle

In accordance with the throttle modeling approach in Eriksson and Nielsen (2014), the mass flow through the throttle is modeled as an isentropic compressible restriction, according to:

$$W_{\text{thr}}(t) = \frac{p_c(t)}{\sqrt{R_a T_c(t)}} C_{D,\text{thr}} A_{\text{thr,max}} u_{\text{thr}}(t) \Psi_{\text{thr}}(t) \quad (10)$$

where the air mass flow through the throttle is denoted by W_{thr} , the maximum throttle area by $A_{\text{thr,max}}$, the flow coefficient by $C_{D,\text{thr}}$, the temperature by T_c , and the specific gas constant of air by R_a . The throttle effective area is controlled directly via the control signal $u_{\text{thr}}(t)$. The flow parameter Ψ_{thr} is taken from (Shen and Ohata, 2011), and represented as in (Holmbom and Eriksson, 2018):

$$\Psi_{\text{thr}}(t) = \sqrt{\frac{\gamma_a + 1}{2\gamma_a} (1 - \Pi_{\text{thr}}(t)) \left(\Pi_{\text{thr}}(t) + \frac{\gamma_a - 1}{\gamma_a + 1} \right)} \quad (11a)$$

$$\Pi_{\text{thr}}(t) = \begin{cases} \frac{p_{\text{im}}(t)}{p_c(t)} & \text{if } \frac{p_{\text{im}}(t)}{p_c(t)} \geq \frac{1}{\gamma_a + 1} \\ \frac{1}{\gamma_a + 1} & \text{otherwise} \end{cases} \quad (11b)$$

where γ_a is the ratio of specific heats. The conditional equation (11b) does not have a continuous derivative at the switching point. The Logistic function is therefore used to emulate the switching (11b):

$$\Pi_{\text{thr}}(t) = \Pi^{\text{choke}} + c_{\text{switch}}(t)(\Pi(t) - \Pi^{\text{choke}}) \quad (12a)$$

$$c_{\text{switch}}(t) = \frac{1}{1 + e^{-c_{\Psi}(\Pi(t) - \Pi^{\text{choke}})}} \quad (12b)$$

$$\Pi^{\text{choke}} = \frac{1}{\gamma_a + 1} \quad (12c)$$

$$\Pi(t) = \frac{p_{\text{im}}(t)}{p_c(t)} \quad (12d)$$

where c_{Ψ} is the steepness parameter of the Logistic function. The implemented flow parameter $\Psi_{\text{thr}}(t)$ is modelled as (11a), where $\Pi_{\text{thr}}(t)$ is modelled according to (12).

4.2.1 Parameters

There is one parameter to estimate, $C_{D,\text{thr}}$. The throttle area $A_{\text{thr,max}}$ is known from dataset F.

4.2.2 Parametrization and Validation

A dataset for validating the throttle model was not obtained. $C_{D,\text{thr}}$ is seen as a tuning parameter in optimizing the steady-state levels of the model. A validation of Ψ_{thr} is depicted in Figure 2.

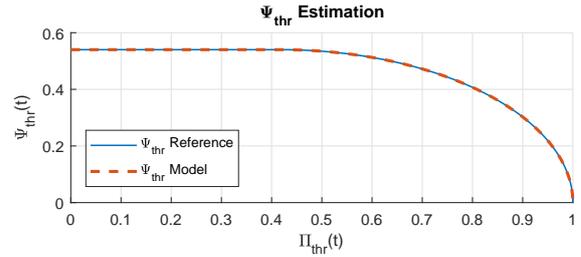


Figure 2. $\Psi_{\text{thr}}(t)$ -functions where the blue solid reference is described by (11), the red trace is when $\Pi_{\text{thr}}(t)$ is described according to (12).

4.3 Cylinder

The cylinder air mass flow is modeled with the help of the volumetric efficiency, η_{vol} , Heywood (1988). It is expressed using a single constant as in (Eriksson and Nielsen, 2014):

$$W_{\text{cyl}}(t) = \eta_{\text{vol}} \frac{V_D p_{\text{im}}(t) N_{\text{ice}}(t)}{n_r R_a T_{\text{im}}} \frac{1}{60} \quad (13)$$

The cylinder fuel mass flow is calculated from the fuel injection per cycle and engine speed:

$$W_f(t) = \frac{n_{\text{cyl}} N_{\text{ice}}(t) u_f(t)}{n_r} \frac{10^{-6}}{60} \quad (14)$$

The fuel-to-air equivalence ratio $\phi(t)$ is calculated as:

$$\phi(t) = \frac{W_f(t)}{W_{\text{cyl}}(t)} \text{AF}_s \quad (15)$$

where AF_s is the air-to-fuel stoichiometric ratio.

To model the engine torque $M_{\text{ice}}(t)$, it is broken down into the components: gross indicated torque $M_{\text{ig}}(t)$, pumping torque $M_{\text{pump}}(t)$, and friction torque $M_{\text{fric}}(t)$, and calculated as:

$$M_{\text{ice}}(t) = M_{\text{ig}}(t) - M_{\text{pump}}(t) - M_{\text{fric}}(t) \quad (16)$$

where the three components $M_{\text{ig}}(t)$, $M_{\text{pump}}(t)$ and $M_{\text{fric}}(t)$ are expressed in the normalized quantities IMEP, PMEP and FMEP (Eriksson and Nielsen, 2014).

The indicated torque is modeled according to:

$$M_{\text{ig}}(t) = \frac{V_D}{n_r 2\pi} \text{IMEP}(t) \quad (17a)$$

$$\text{IMEP}(t) = \eta_{\text{ig}}(t) \frac{q_{\text{HV}} u_f(t) 10^{-6} n_{\text{cyl}}}{V_D} \quad (17b)$$

$$\eta_{\text{ig}}(t) = \left(1 - \frac{1}{r_c^{\gamma_{\text{cyl}}(t)-1}} \right) \eta_{\text{cal}}(t) \quad (17c)$$

$$\gamma_{\text{cyl}}(t) = c_{\gamma,0} + c_{\gamma,1} \phi(t) + c_{\gamma,2} \phi^2(t) \quad (17d)$$

Where the operating point dependent losses, are modeled using the following polynomial structure:

$$\eta_{\text{cal}}(t) = c_{\text{cal},2}(t) \left(\frac{u_f(t)}{100} - c_{\text{cal},1}(t) \right)^2 + c_{\text{cal},0} \quad (18a)$$

$$c_{\text{cal},1}(t) = c_{\text{cal},10} + c_{\text{cal},11} \frac{N_{\text{ice}}(t)}{1000} \quad (18b)$$

$$c_{\text{cal},2}(t) = c_{\text{cal},20} + c_{\text{cal},21} \frac{N_{\text{ice}}(t)}{1000} + c_{\text{cal},22} \left(\frac{N_{\text{ice}}(t)}{1000} \right)^2 \quad (18c)$$

The pumping losses are modeled as:

$$M_{\text{pump}}(t) = \frac{V_D}{n_r 2\pi} \text{PMEP}(t) \quad (19a)$$

$$\text{PMEP}(t) = c_{\text{PMEP},0} + c_{\text{PMEP},1}(p_{\text{em}}(t) - p_{\text{im}}(t)) \quad (19b)$$

The losses which are not included in the pumping losses are lumped into the friction term, which is modeled as a polynomial in fuel injection and engine speed:

$$M_{\text{fric}}(t) = \frac{V_D}{n_r 2\pi} \text{FMEP}(t)$$

$$\text{FMEP}(t) = c_{f,0} + c_{f,1}N_{\text{ice}}(t) + c_{f,2}u_f(t) + c_{f,3}u_f(t)N_{\text{ice}}(t) \quad (20)$$

The temperature of the gas exiting the cylinders, T_e , is modeled based on calculations on an ideal cycle and adding the parameter η_{sc} to include non-ideal properties:

$$T_e(t) = \eta_{\text{sc}} \Pi_{\text{cyl}}^{1-1/\gamma_a}(t) r_c^{1-\gamma_a} \left(\frac{q_{\text{in}}(t)}{c_{p,a}} + T_{\text{im}} r_c^{\gamma_a-1} \right) \quad (21a)$$

$$q_{\text{in}}(t) = \frac{W_f(t)}{W_f(t) + W_{\text{cyl}}(t)} q_{\text{HV}} \quad (21b)$$

$$\Pi_{\text{cyl}}(t) = \frac{p_{\text{em}}(t)}{p_{\text{im}}(t)} \quad (21c)$$

To take the heat transfer from the exhaust manifold to the surroundings into account, the mean value exhaust gas temperature model from Eriksson (2002) is implemented:

$$T_{\text{em}}(t) = T_{\text{amb}} + (T_e(t) - T_{\text{amb}}) e^{-\frac{c_{\text{em,h}}}{(W_{\text{cyl}}(t) + W_f(t))c_{p,e}}} \quad (22)$$

4.3.1 Parameters

There are 18 parameters to estimate: η_{vol} , $c_{\gamma,0}$, $c_{\gamma,1}$, $c_{\gamma,2}$, $c_{\text{cal},0}$, $c_{\text{cal},10}$, $c_{\text{cal},11}$, $c_{\text{cal},20}$, $c_{\text{cal},21}$, $c_{\text{cal},22}$, $c_{f,0}$, $c_{f,1}$, $c_{f,2}$, $c_{f,3}$, $c_{\text{PMEP},0}$, $c_{\text{PMEP},1}$, η_{sc} , and $c_{\text{em,h}}$.

4.3.2 Parametrization and Validation

To preserve the properties observed in the data, the parametrization is carried out in steps. The cylinder massflow is estimated using dataset A. The error function is calculated as:

$$e_{k,\eta_{\text{vol}}} = (\eta_{\text{vol,data},k} - \eta_{\text{vol}})^2 \quad (23)$$

The resulting fit is shown in Table 2.

The indicated torque is estimated using data set B (Table 1). The parameterization was done by minimizing the following error function:

$$e_{k,\text{IMEP}} = (\text{IMEP}_{\text{data},k} - \text{IMEP}_{\text{model},k}(\theta_{\text{IMEP}}))^2 \quad (24a)$$

$$\theta_{\text{IMEP}} = [c_{\gamma,0}, c_{\gamma,1}, c_{\gamma,2}, c_{\text{cal},10}, c_{\text{cal},11}, c_{\text{cal},20}, c_{\text{cal},21}, c_{\text{cal},22}] \quad (24b)$$

The resulting fit is shown in Table 3.

The pumping torque is estimated using data set B (Table 1). The parameterization was done by minimizing the following error function:

$$e_{k,\text{PMEP}} = (\text{PMEP}_{\text{data},k} - \text{PMEP}_{\text{model},k}(\theta_{\text{PMEP}}))^2 \quad (25a)$$

$$\theta_{\text{PMEP}} = [c_{\text{PMEP},0}, c_{\text{PMEP},1}] \quad (25b)$$

The resulting fit is shown in Table 3.

The friction torque is estimated using data set B (Table 1). The parameterization was done by minimizing the following error function:

$$e_{k,\text{FMEP}} = (\text{FMEP}_{\text{data},k} - \text{FMEP}_{\text{model},k}(\theta_{\text{FMEP}}))^2 \quad (26a)$$

$$\theta_{\text{FMEP}} = [c_{f,0}, c_{f,1}, c_{f,2}, c_{f,3}] \quad (26b)$$

The resulting fit is shown in Table 3.

For the model describing the exhaust manifold temperature, the loss function is calculated as:

$$e_{k,T} = (T_{\text{em,data},k} - T_{\text{em,model},k}(\theta_{T_{\text{em}}}))^2 \quad (27a)$$

$$\theta_{T_{\text{em}}} = [\eta_{\text{sc}}, c_{\text{em,h}}] \quad (27b)$$

The resulting fit is shown in Table 2.

Table 2. Cylinder model fit to dataset A. \bar{e} is the mean absolute error, e_{σ} the variance of the absolute error, \bar{e}_{rel} the relative error, and $e_{\text{rel},\sigma}$ the variance of the relative error.

	\bar{e}	e_{σ}	\bar{e}_{rel}	$e_{\text{rel},\sigma}$
η_{vol}	0.84 %	0.61 %	0.92 %	0.69 %
T_{em}	5.20 K	3.40 K	0.78 %	0.49 %

Table 3. Cylinder model fit to dataset B. \bar{e} is the mean absolute error, e_{σ} the variance of the absolute error, \bar{e}_{rel} the relative error, and $e_{\text{rel},\sigma}$ the variance of the relative error.

	\bar{e}	e_{σ}	\bar{e}_{rel}	$e_{\text{rel},\sigma}$
IMEP	24.0 kPa	11.7 kPa	1.57 %	0.42 %
PMEP	99.2 kPa	105.1 kPa	23.0 %	13.6 %
FMEP	0.75 kPa	0.52 kPa	0.70 %	0.43 %

4.4 Turbocharger

The turbocharger dynamics is modeled from Newton's second law of motion according to:

$$\frac{d}{dt} \omega_{\text{tc}}(t) = \frac{P_t \eta_{\text{m}}(t) - P_c(t)}{\omega_{\text{tc}}(t) J_{\text{tc}}} \quad (28)$$

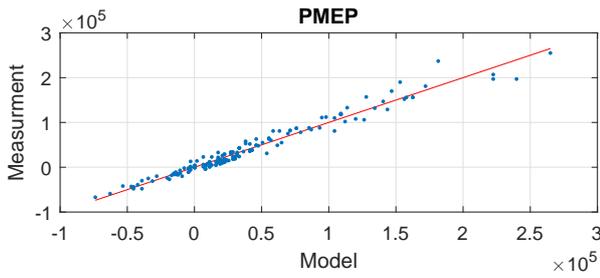


Figure 3. PMEP fit to dataset B. Red line shows the one-to-one ratio.

4.4.1 Parameters

There is one parameter to estimate: J_{tc} . The validation is seen in Section 6.

4.5 Turbine

The turbine power, including the mechanical efficiency of the turbocharger shaft is calculated according to:

$$P_t \eta_m(t) = W_t(t) c_{p,e} T_{em}(t) \eta_t(t) \left(1 - \Pi_t(t)^{1-1/\gamma_e}\right) \quad (29a)$$

$$\Pi_t(t) = \frac{p_{ats}}{p_{em}(t)} \quad (29b)$$

where p_{ats} is the pressure in the exhaust aftertreatment system. For the flow, the square root turbine flow model in (Eriksson and Nielsen, 2014) is adapted to describe dataset D:

$$W_{t,corr}(t) = k_0(t) \left(1 - \Pi_t(t)^{k_1(t)}\right)^{k_2(t)} \quad (30a)$$

$$k_0(t) = c_{00} + c_{02} N_{tc,corr,I}^2(t) \quad (30b)$$

$$k_1(t) = c_{10} + c_{11} N_{tc,corr,I}(t) \quad (30c)$$

$$k_2(t) = c_{20} + c_{21} N_{tc,corr,I}(t) + c_{22} N_{tc,corr,I}^2(t) \quad (30d)$$

$$N_{tc,corr,I}(t) = \frac{\omega_{tc}(t)}{\sqrt{T_{em}(t)}} \frac{30}{\pi} \frac{1}{1000} \quad (30e)$$

The turbine mass flow is calculated as:

$$W_t(t) = W_{t,corr}(t) \frac{p_{em} 10^{-3}}{\sqrt{T_{em}}} \quad (31)$$

For the turbine efficiency, the standard approach of modeling it from the blade-speed-ratio (BSR) is taken. However, since the BSR lines do not overlap in the BSR– η plane in dataset D, a speed dependence is in-

cluded. The model is described by:

$$\eta_t(t) = \eta_{t,max}(t) - k_\eta(t) (BSR(t) - BSR_{opt}(t))^2 \quad (32a)$$

$$BSR(t) = \frac{\omega_{tc}(t) D_t / 2}{\sqrt{2 c_{p,e} T_{em}(t) (1 - \Pi_t(t))^{1-1/\gamma_e}}} \quad (32b)$$

$$BSR_{opt}(t) = c_{BSR,0} + c_{BSR,1} N_{tc,corr,II}(t) + \quad (32c)$$

$$c_{BSR,2} N_{tc,corr,II}^2(t) \quad (32d)$$

$$\eta_{t,max}(t) = c_{\eta,0} + c_{\eta,1} N_{tc,corr,II}(t) \quad (32e)$$

$$k_\eta(t) = c_{max,0} + c_{max,1} N_{tc,corr,II}(t) \quad (32f)$$

$$N_{tc,corr,II}(t) = \frac{\omega_{tc}(t)}{\sqrt{T_{em}(t)}} \frac{30}{\pi} \frac{1}{10000} \quad (32g)$$

4.5.1 Parameters

There are 14 parameters to estimate: c_{00} , c_{02} , c_{10} , c_{11} , c_{20} , c_{21} , c_{22} , $c_{BSR,0}$, $c_{BSR,1}$, $c_{BSR,2}$, $c_{\eta,0}$, $c_{\eta,1}$, $c_{max,0}$, and $c_{max,1}$.

4.5.2 Parametrization and Validation

The parameterization is carried out by parameterizing the mass flow model and efficiency model separately using dataset D. For the mass flow, the loss function is calculated as:

$$e_{k,W_t} = (W_{t,data,k} - W_{t,model,k}(\theta_{W_t}))^2 \quad (33a)$$

$$\theta_{W_t} = [c_{00} \ c_{02} \ c_{10} \ c_{11} \ c_{20} \ c_{21} \ c_{22}] \quad (33b)$$

The model fit is shown in Table 4. For the efficiency, the following loss function is used:

$$e_{k,\eta_t} = (\eta_{t,data,k} - \eta_{t,model,k}(\theta_{\eta_t}))^2 \quad (34)$$

$$\theta_{\eta_t} = [c_{BSR,0} \ c_{BSR,1} \ c_{BSR,2} \ c_{\eta,0} \ c_{\eta,1} \ c_{max,0} \ c_{max,1}] \quad (35)$$

The model fit is shown in Table 4.

4.6 Wastegate

The wastegate is developed in the same way as the throttle (see section 4.2), apart from γ_a which is replaced by γ_e . The wastegate mass flow is described by:

$$W_{wg}(t) = \frac{p_{em}(t)}{\sqrt{R_e T_{em}}} C_{D,wg} A_{wg,max} u_{wg}(t) \Psi_{wg}(t) \quad (36)$$

where $\Psi_{wg}(t)$ is similar to (11a), but $\Pi_{thr}(t)$ is replaced by $\Pi_{wg}(t)$. $\Pi_{wg}(t)$ is defined as in Equation (12), where $\Pi(t)$ is replaced by $\Pi_t(t)$ in (29b).

Table 4. Turbine model fit to dataset D. \bar{e} is the mean absolute error, e_σ the variance of the absolute error, \bar{e}_{rel} the relative error, and $e_{rel,\sigma}$ the variance of the relative error.

	\bar{e}	e_σ	\bar{e}_{rel}	$e_{rel,\sigma}$
$W_{t,corr}$	$5.14 \cdot 10^{-4}$	$3.56 \cdot 10^{-4}$	1.30 %	0.95 %
	kg/s \sqrt{K} /kPa	kg/s \sqrt{K} /kPa		
η_t	0.92 %	0.99 %	1.34 %	1.47 %

4.6.1 Parameters

There is only one parameter to estimate, $C_{D,wg}$. The wastegate area $A_{wg,max}$ is determined from measuring the diameter. $C_{D,wg}$ is seen as a tuning constant. Since data was not available to parameterize the wastegate as a separate component, the same approach as for the throttle is taken. The modeling of the wastegate and throttle is similar due to both being controllable valves for restricting the gas flow.

4.7 Compressor

The parameterization of the compressor was done using dataset C and by using *LiU CPgui* (Llamas and Eriksson, 2018), which parameterizes a high-order control-oriented compressor model based on the *total least squares* algorithm. The fit to dataset A is shown in Table 5.

Table 5. Compressor model fit to dataset A. \bar{e} is the mean absolute error, e_σ the variance of the absolute error, \bar{e}_{rel} the relative error, and $e_{rel,\sigma}$ the variance of the relative error.

	\bar{e}	e_σ	\bar{e}_{rel}	$e_{rel,\sigma}$
W_c	$1.03 \cdot 10^{-4}$	$8.29 \cdot 10^{-4}$	0.13 %	1.51 %
	kg/s	kg/s		
η_c	0.92 %	0.99 %	1.34 %	1.47 %

5 Full System Parametrization

An approach taken in Wahlström and Eriksson (2011) and Sivertsson and Eriksson (2011) is to refit the parameters to the measurements when all model components are assembled in order to achieve good fit. A similar approach is taken here, but with a slightly different cost function (see equation 2) where a regularization technique is used in order to limit the changes in parameter values in order to preserve the model structure. This procedure is divided into two steps. The first step is to refit the torque model to dataset A. The second step is to tune all parameters influencing the steady state levels of the states when the complete model is fully assembled.

5.1 Torque model

For the torque model, the following parameters are refitted:

$$\theta_M = [c_{cal,0}, c_{cal,10}, c_{cal,11}, c_{cal,20}, c_{cal,21}, c_{cal,22}, c_{f,0}, c_{f,1}, c_{f,2}, c_{f,3}, c_{PMEP,0}, c_{PMEP,1}] \quad (37)$$

using criterion (2). The resulting fit is shown in Table 6

Table 6. Torque model refit to dataset A. \bar{e} is the mean absolute error, e_σ the variance of the absolute error, \bar{e}_{rel} the relative error, and $e_{rel,\sigma}$ the variance of the relative error.

	\bar{e}	e_σ	\bar{e}_{rel}	$e_{rel,\sigma}$
M_{ice}	8.42 Nm	6.59 Nm	1.81 %	3.05 %

5.2 Tuning of steady-state levels

In the same manner as for the torque model, using criterion (2), the parameters influencing the steady-state levels of the states are re-parameterized. To do this, the full model is simulated and the wastegate is controlled, using a *PID*-controller to minimize the error:

$$e_{wg} = W_c - W_{c,ref} \quad (38)$$

where the compressor massflow reference $W_{c,ref}$ is taken from the massflow measurement in dataset A. The resulting fit is shown in Table 7.

Table 7. Steady-state levels of states fit to dataset A for the fully parameterized model. \bar{e} is the mean absolute error, e_σ the variance of the absolute error, \bar{e}_{rel} the relative error, and $e_{rel,\sigma}$ the variance of the relative error.

	\bar{e}	e_σ	\bar{e}_{rel}	$e_{rel,\sigma}$
p_c	2.78 kPa	4.00 kPa	1.89 %	2.80 %
p_{im}	2.76 kPa	3.96 kPa	1.89 %	2.79 %
p_{em}	14.73 kPa	3.96 kPa	7.33 %	3.69 %
ω_c	1.98 kRPM	2.42 kRPM	4.88 %	7.93 %

6 Full System Validation

The complete system model is simulated with the control signals recorded from a dynamic engine test (part of data set E). The result from the simulated system, in comparison with the measurement data is displayed in Figure 4.

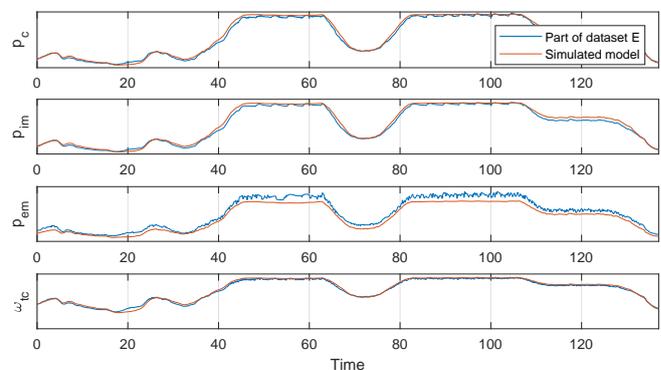


Figure 4. State validation, simulated engine model compared to dynamic measurement data. The control signals are taken from measurements.

7 Conclusions

A mean value engine model of a Scania 12.7 liters heavy-duty diesel engine has been developed and validated using both stationary and dynamic measurements. The results show good agreement with measurements, showing that both dynamics and steady-state levels are well represented indicating that the model is well suited for studying the engine dynamics and fuel optimal control. The model is open-source and downloadable from www.fs.isy.liu.se/Software/.

Acknowledgment

The work was financed by *Swedish Governmental Agency for Innovation Systems* under the program *Strategic Vehicle Research and Innovation*, grant FROST (2016-05380) and LINK-SIC. The authors would like to thank Scania, and especially Henrik Höglund and Björn Johansson at Scania, for providing the data and for modeling discussions.

References

- Kristoffer Ekberg, Viktor Leek, and Lars Eriksson. Optimal control of wastegate throttle and fuel injection for a heavy-duty turbocharged diesel engine during tip-in. In *58th Conference on Simulation and Modelling (SIMS 58)*, Reykjavik, Iceland, 2017.
- Lars Eriksson. Mean value models for exhaust system temperatures. *SAE 2002 World Congress & Exhibition*, 2002. ISSN 0148-7191. URL <https://doi.org/10.4271/2002-01-0374>.
- Lars Eriksson and Lars Nielsen. *Modeling and Control of Engines and Drivelines*. John Wiley and Sons Ltd, 2014.
- Gamma Technologies. GT-Power User's Manual. *GT-Suite Version 6.1*, 2004.
- John B. Heywood. *Internal Combustion Engine Fundamentals*. McGraw-Hill, 1988.
- Robin Holmbom and Lars Eriksson. Analysis and development of compact models for mass flows through butterfly throttle valves. In *WCX World Congress Experience*. SAE International, Apr 2018. URL <https://doi.org/10.4271/2018-01-0876>.
- Xavier Llamas and Lars Eriksson. Control-oriented compressor model with adiabatic efficiency extrapolation. *SAE International Journal of Engines*, 10(4), 2017.
- Xavier Llamas and Lars Eriksson. LiU CPgui: A toolbox for parameterizing compressor models. Technical Report LiTH-ISY-R-3102, Department of Electrical Engineering, Linköpings Universitet, SE-581 83 Linköping, Sweden, 2018.
- Tielong Shen and Akira Ohata. *Modeling and Control Design for Automotive Engines - with MATLAB Engine Simulator CD-ROM*. ISBN 978e-4-339-04610-6, 2011.
- Martin Sivertsson and Lars Eriksson. Modeling for Optimal Control: A Validated Diesel-Electric Powertrain Model. *SIMS 2014 - 55th International Conference on Simulation and Modelling*, Ålborg, Denmark, 2011.
- Johan Wahlström and Lars Eriksson. Modelling diesel engines with a variable-geometry turbocharger and exhaust gas recirculation by optimization of model parameters for capturing non-linear system dynamics. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 225(7):960–986, 2011.

Simulations and real applications of PI and MPC averaging level control in a water resource recovery facility

Finn Aakre Haugen¹

¹University of South-Eastern Norway, Norway, finn.haugen@usn.no

Abstract

The basin level control system in an existing wastewater treatment plant is designed for two different operation modes: compliant level control for smooth pump flow, and stiff level control. Both a model-based predictive controller (MPC) and a standard PI controller are implemented, and tested both on a dynamic simulator and on the real plant. The simulator, the MPC and the PI controller tuning are based on a mathematical process model derived from a material balance of the wastewater in the inlet basin.

Keywords: Water resource recovery facility, equalization magazine, level, MPC, Kalman Filter. PI control.

1 Introduction

This article reports results from a study where alternative controllers of the level of the equalization magazine upstream to the VEAS water resource recovery facility (wrrf)¹ at Slemmestad, Norway has been tested both on a dynamic simulator and on the real plant. This magazine comprises the part of the inlet tunnel being closest to the wrrf. VEAS is the largest wrrf in Norway, serving about 700,000 population equivalents (pe), treating in average about 3.5 m³/s. The biological treatment at the wrrf will benefit from a smoother hydraulic load than at present. One approach to this end is improving the level control system of the inlet basin, aiming at smoother pump flow from the basin to the treatment processes (Bolmstedt, 2004); (van Overloop et al., 2010).

The implementation of the simulator and the control system used in this study is in LabVIEW (National Instruments) with the MPC algorithm implemented in MATLAB code in LabVIEW's MATLAB Script node. The sampling time (time-step) of the various discrete-time algorithms of simulation, estimation, filtering, and control is 10 s.

The paper is organized in the following main sections: System description; Controller functions; Results; Conclusion; Abbreviations; Nomenclature; Acknowledgements; References.

¹"wrrf" is the terminology recommended by the International Water Association (IWA) and the Water Environment Federation (WEF)

2 System description

2.1 Geometrical design

Figure 1 depicts the principal geometrical design of the inlet part of the VEAS wrrf.

2.2 Operation modes of the tunnel and basin and requirements to the level control system

The tunnel and basin are operated in different modes:

- **Operation mode #1: Normally low load (tunnel flow) & Compliant level control:** The main aim of this mode is to obtain smooth pump flow. To this end, compliant level control is implemented: The level is allowed to vary between the soft limits of 1.5 m and 2.5 m, with 1.8 m as the nominal level setpoint. Lately, these limits have been changed to 1.6 m and 2.8 m, respectively, with setpoint 2.3 m. (Both these sets of specifications are used in this article.)
- **Operation mode #2: Normally low load & Stiff level control:** The main aim of this mode is to have the level close to a relatively low setpoint to ensure that the pump soaks up solid downfall from the wastewater accumulated in the basin during Operation mode #1. The duration of this operation mode is relatively short, approximately two hours, each second day. In this operation mode, the variations of the pump flow will, inevitably, be relatively large as they are almost the same as the variations of the net inflow to the basin.

In addition to Operation modes #1 and #2, there are operation modes concerning normally high load (at high precipitation) and to tunnel flushing (building up a volume of wastewater in a part of the tunnel, and then flushing the tunnel with this volume). This article covers only Operation mode #1.

2.3 Mathematical model of the inlet basin

The basis of both the simulator and the model-based predictive controller is a dynamic model of the liquid level of the basin, Eq. 1. The model stems from material balance assuming the sewage is water.

$$\dot{h}(t) = \frac{F_{in}(t) - F_{out}(t)}{A(h)} \quad (1)$$

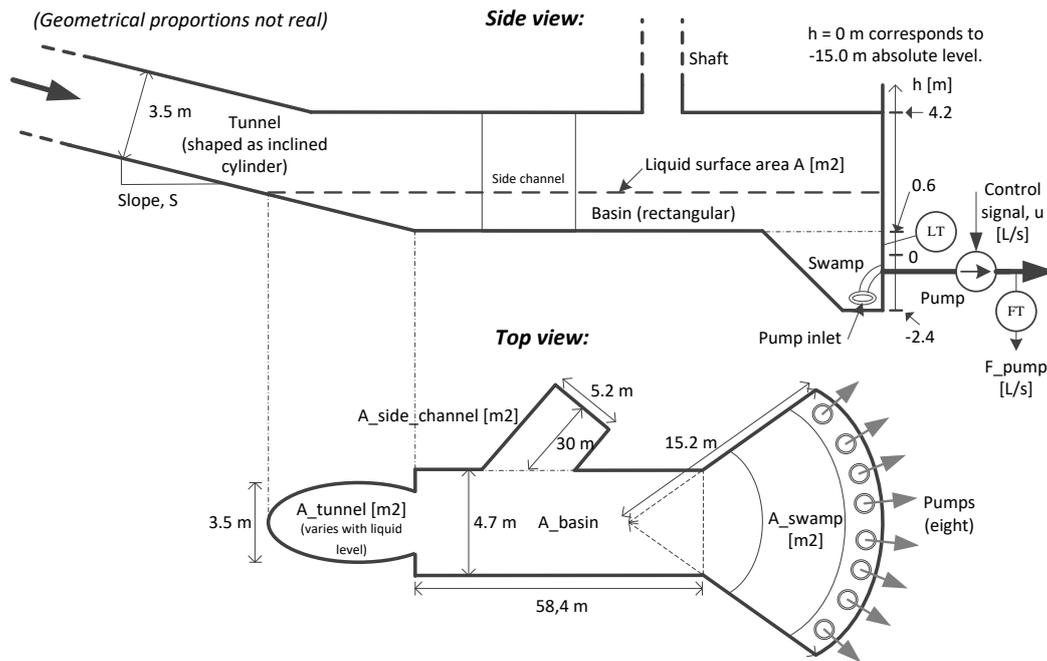


Figure 1. Principal design of inlet basin of VEAS wrff.

where

$$F_{out}(t) = F_{pump}(t) \quad (2)$$

and

$$F_{in}(t) = F_{in_{meas}}(t) + F_{in_{nonmeas}}(t) \quad (3)$$

where $F_{in_{meas}}$ are measured flows and $F_{in_{nonmeas}}$ are unmeasured flows.

The liquid surface area A in Eq. 1 is calculated from the assumed known geometry, and is therefore a function of the liquid level, h . The ellipsoidal liquid surface in the tunnel is continuously calculated by numerical integration. The dynamics of the pump is taken into account by the level controllers. In the real plant, an apparent time-delay of approximately 120 s is observed between the pump control signal u and the resulting (measured) pump flow F_{pump} :

$$F_{pump}(t) = u(t - T_{delay_{pump}}) \quad (4)$$

For conservative controller tuning, the pump dynamics is represented by a time-delay of $T_{delay_{pump}} = 120$ s.

2.4 Piping and Instrumentation Diagram (P&I diagram)

Figure 2 shows a P&I diagram of the level control system. The flow at Vækerø is measured. Vækerø is situated approximately 15 km upstreams the plant, The flow at Vækerø constitutes the main inflow component to the plant, counting for 70-80% of the total tunnel inflow. This flow arrives at the plant with a transportation time (time-delay) of approximately 3.0 h, but smoothed, so the transportation time is not well-defined. The smoothing is represented with a time constant low pass filter with an estimated time constant of 1 h. The Vækerø flow measurement is used in simulations. However, it is not used by the

controllers in the real implementations in this study due to the uncertain information about its contribution to the actual inflow to the basin, as pointed out above.

3 Controllers

Two different controllers are used in this study, namely (a) PI control and (b) MPC.

3.1 PI controller

A standard PI controller (Seborg et al., 2004) is used:

$$u(t) = K_c e(t) + \frac{K_c}{T_i} \int_0^t e(\theta) d\theta \quad (5)$$

The PI controller is tuned with the Skogestad method (Skogestad, 2003) for “integrator plus time-delay” process dynamics, but with a modification where the integral time, T_i , is reduced to obtain faster disturbance compensation (Haugen and Lie, 2013). In general, the following differential equation can represent such dynamics:

$$\dot{y} = K_i u(t - \tau) \quad (6)$$

By neglecting F_{in} , the process model, Eqs. 1 - 4, is on the form of Eq. 6 with

$$\begin{aligned} y &= h \\ u &= F_{pump} \\ K_i &= -\frac{1}{A} \end{aligned} \quad (7)$$

The Skogestad PI settings for the model Eq. 6 are:

$$K_c = \frac{1}{K_i(T_c + \tau)} = -\frac{A}{T_c + \tau} \quad (8)$$

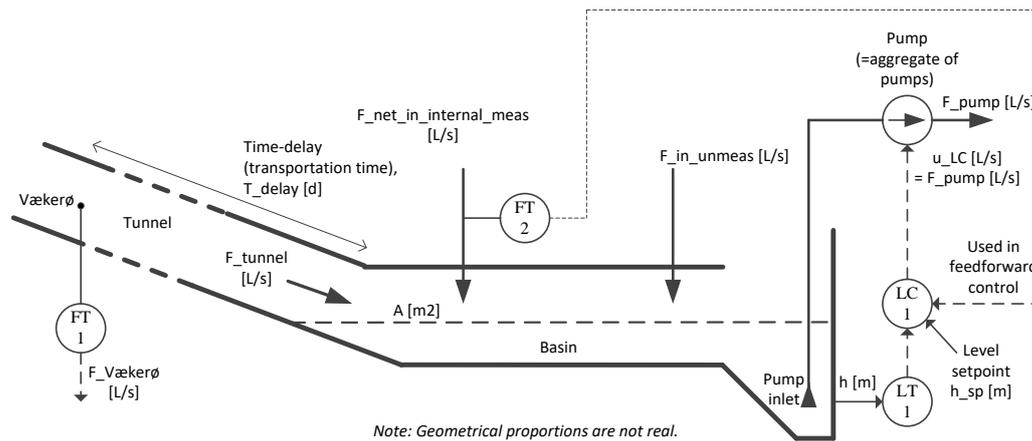


Figure 2. Piping and Instrumentation diagram (P&I D) of level control system of basin.

$$T_i = 2(T_c + \tau) \quad (9)$$

where T_c is the closed (control) loop time constant specified by the user. In general, the PI controller can be tuned for satisfactory performance using T_c as follows: By increasing T_c , the control systems typically becomes more sluggish: The control signal becomes smoother, and the control error becomes larger. On the other side, by decreasing the closed loop time-constant, the control systems typically becomes more aggressive (faster): The control signal varies more abruptly, and the control error becomes smaller.

3.2 MPC with Kalman Filter

A nonlinear MPC is used (Grüne and Pannek, 2011). The optimization (minimization) problem of the MPC used in this study is²

$$\min_u \left[J = \int_{t_0}^{t_0+T_p} C_1 e(t)^2 + C_2 \dot{u}(t)^2 dt \right] \quad (10)$$

The optimization or decision variable is the pump flow (control variable). $e = h_{sp} - h$ is control error. $\dot{u} = \dot{F}_{pump}$ is rate of change of pump flow (control variable). C_1 and C_2 are cost coefficients. T_p is the prediction horizon. t_0 is the present point of time. The MPC finds the sequence of sampled future pump flow values that gives the optimal balance or compromise between small control error and small rate of change of pump flow. To save the computational demand, control signal locking is used, i.e. the number of allowable values during the prediction horizon is set to N_p .

A time-constant filter with time constant T_{pump} was included in the calculation of the applied control signal to the pump to obtain improved pump flow smoothing.

The MPC uses a Kalman Filter estimate of the total unmeasured flows, see below.

²Although the objective function, J , is here on a continuous-time form, a corresponding discrete-time form is of course used in the computer implementation.

The MPC takes into account level constraints and control variable (pump flow) constraints. MATLAB's `fmincon` function is used as optimization function.

Kalman Filter. The estimate of the total of the unmeasured flows is calculated with an Extended Kalman Filter (EKF) (Simon, 2006) based on the process model, Eqs. 1 – 3 with total unmeasured flow, $F_{in_{nonmeas}}$ modelled as a “random walk” augmentation state variable. The time delay in Eq. 4 was neglected as it is assumed to have negligible effect on the estimates.

To summarize, the model used in the EKF comprises the following two differential equations:

$$\dot{h} = \frac{F_{in_{meas}} + F_{in_{nonmeas}} - F_{pump}}{A(h)} + w_1 \quad (11)$$

$$\dot{F}_{in_{nonmeas}} = 0 + w_2 \quad (12)$$

where w_1 and w_2 are random process disturbances.

The process output measurement used by the EKF to correct the predicted state estimate is the measured level.

The process measurement noise covariance was set to

$$R = [0.1] \quad (13)$$

and the process disturbance covariance was set to

$$Q = \begin{bmatrix} 0.1 & 0 \\ 0 & 10^{-6} \end{bmatrix} \quad (14)$$

In the tuning, R and $Q(1,1)$ were fixed to more or less random values, and then $Q(2,2)$ was used as the ultimate tuning parameter since it directly affects the estimate of $F_{in_{nonmeas}}$. $Q(2,2)$ was adjusted by trial-and-error to give a sufficiently fast while not too noisy flow estimate. R might have been set to the variance of representative time-series of the level measurement, but this was not implemented. In practice, it is mainly the ratio between the variances that determines the behaviour of the estimator.

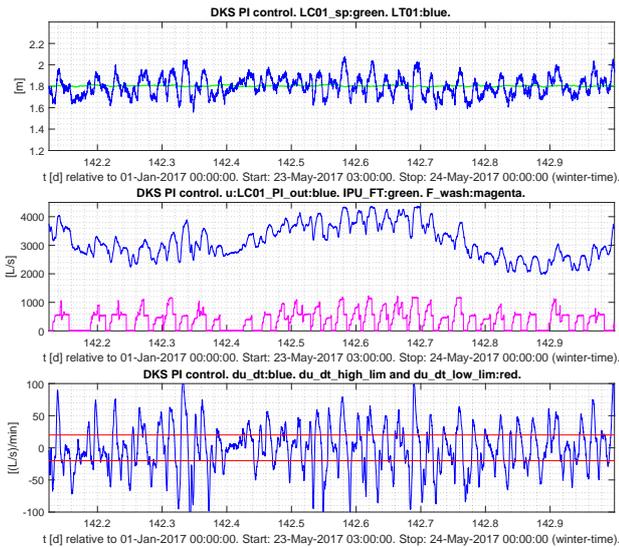


Figure 3. Results with real PI level control with original PI settings. (Time range: 21 h.)

4 Results

4.1 Real PI level control with original PI settings

Figure 3 shows real responses with the following original PI controller settings (before the retuning presented in this article):

$$K_c = -3200 \text{ (L/s)/m}, T_i = 1000 \text{ s} \quad (15)$$

which have been found with trial and error. The level setpoint is 1.8 m. The absolute value of the rate of change of the pump flow, $|\dot{u}|$, becomes substantially larger than the specified maximum of 20 (L/s)/min. The fraction of time that $|\dot{u}|$ is larger than 20 (L/s)/min is 57 %.

4.2 Simulated level control with retuned PI controller

The simulator was driven by real flow measurements. The PI controller was tuned with the Skogestad method with $T_c = 1000$ s adjusted by trial and error on the simulator to satisfy the specifications. The resulting PI settings are:

$$K_c = -2000 \text{ (L/s)/m}, T_i = 2000 \text{ s} \quad (16)$$

The simulated responses are shown in Figure 3 (together with responses with MPC). The specification to the rate of change of pump flow is met, see the lower plot in Figure 4. Also, the level is within the limits.

4.3 Simulated level control with MPC

The MPC was tuned with $C_1 = 1$ (fixed) and $C_2 = 0.05$ by trial and error. The time step was 120 sec, and the prediction horizon was 30 minutes, corresponding to 15 time steps. Control signal blocking with $N_p = 3$ blocks

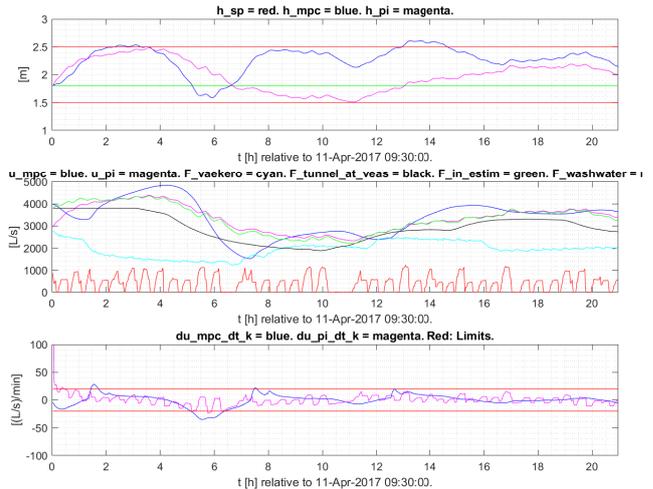


Figure 4. Results with simulated PI level control (magenta curves) and MPC level control (blue curves). (Time range: 21 h.)

of equal size (5 time steps) was implemented. The pump time-constant filter was set by trial and error to $T_{\text{pump}} = 700$ s.

The simulated responses are shown in Figure 4. The specification to the rate of change of pump flow is met except for a small time interval around time $t = 6$ h, see the lower plot in Figure 4. The level is within the limits except for a small time interval around $t = 14$ h.

4.4 Real level control with retuned PI controller

A retuned PI level controller was applied to the real plant. The retuning was based on the Skogestad method. It was decided to increase the normal level setpoint from originally 1.8 m to 2.3 m. This level increase implies that the liquid surface area, A , increases, thereby reducing the process gain. The reduction of the process gain allows for a more relaxed PI tuning (larger T_c).

The Skogestad tuning was based on the specification $T_c = 1500$ s. A time-delay of $T_{\text{delay,pump}} = 120$ s was included in the process model to account for pump control dynamics. The liquid surface area was fixed to $A = 2000$ m² in the tuning, which is approximately the area at the operating point of level $h = 1.8$ m.

The resulting tuning became

$$K_c = -1240 \text{ (L/s)/m}, T_i = 3240 \text{ s} \quad (17)$$

Figure 5 shows the results on the real plant. As seen from the lower plot, the specifications to the maximum rate of the pump flow is satisfied.

The PI settings above are now (as of May 2018) in ordinary use at the plant.

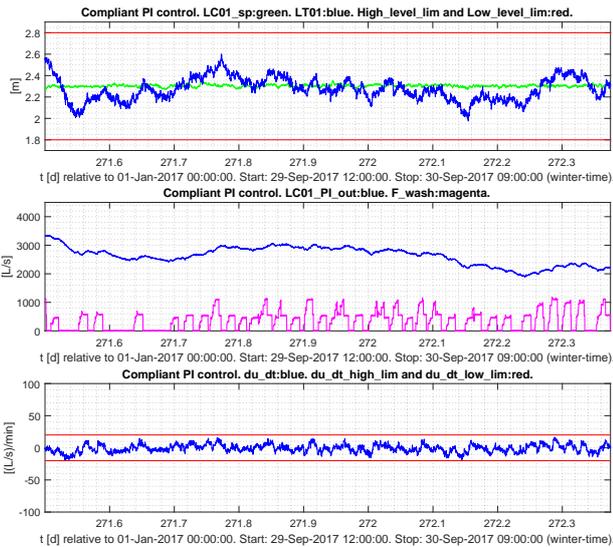


Figure 5. Results with PI control on the real plant. (Time range: 21 h.)

4.5 Real level control with MPC

Due to practical reasons, only initial results of MPC level control applied to the real plant can be shown in this article. Both a simulation study similar to the study presented above, and real tests were accomplished. Both these tests were accomplished without including the lowpass filter to smooth the pump flow. The MPC pump control signal showed abrupt changes at the points of time when the MPC decided to change the control signal, see Figure 6. The control signal resembles a piecewise constant signal. This behaviour is actually understandable since $\dot{u}(t)$ is zero except at the short transitions, making the rate of change term, $\dot{u}(t)^2$, in the MPC objective relatively small.

Several solutions were tried to mitigate the abrupt control signal changes. The best solution turned out to be the pump lowpass filter, but from these tests, we only have simulation results, cf. Section 4.3. Unfortunately, the equipment was not longer available for a practical test which might have confirmed the good simulation results with the improved MPC.

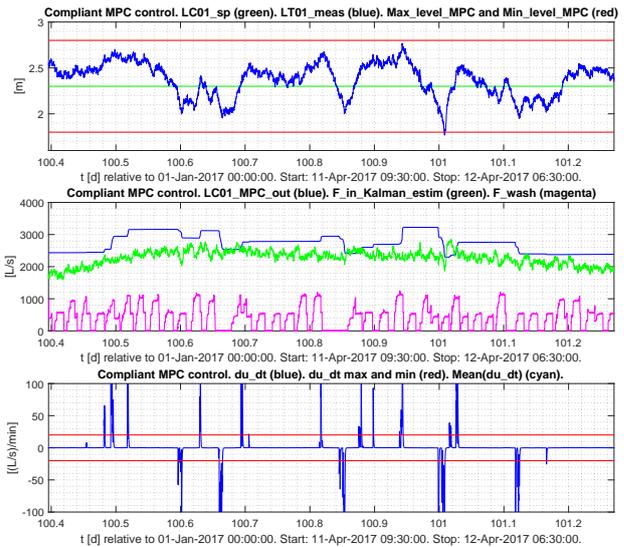


Figure 6. Results with MPC on the real plant. (Time range: 21 h.)

5 Conclusions

Successful averaging level control was obtained with a PI controller tuned with the Skogestad tuning method, both on the simulated plant and on the real plant.

The first implementation of MPC for averaging level control applied to the real plant was not successful as the pump flow pattern generated by the controller showed abrupt changes between the time intervals of piecewise constant flow. By including a lowpass filter on the control signal in the MPC model, successful control was eventually obtained with MPC – on the simulated plant. Unfortunately, due to practical reasons, this successful MPC implementation could not be tested on the real plant.

This study has demonstrated, to the author and to the staff at the plant, the power of using a dynamic simulator for tuning and testing control systems.

In the hindsight, one lesson to learn from this study is that enough time should be allocated to, hopefully, obtain satisfying simulation-based tests *before* testing the controller on the real plant.

Acknowledgements

The author is thankful for the support provided by the staff at VEAS wrrf, Slemmestad, Norway.

Nomenclature

Symbol	[Unit]	Description
A	[m ²]	Surface area of liquid
C_1, C_2	-	Cost coefficients of MPC
e	-	Control error
F_{in}	[L/s]	Total inflow to the basin
$F_{in_{meas}}$	[L/s]	Total measured inflows to the basin
$F_{in_{nonmeas}}$	[L/s]	Total unmeasured inflows to the basin
F_{out}	[L/s]	Total outflow of the basin
F_{pump}	[L/s]	Total basin pump flow
h	[m]	Water level of basin
h_{sp}	[m]	Setpoint of water level of basin
K_c	[(L/s)/m]	Gain of PI controller
K_i	-	Process integral gain
N_p	[1]	Number of MPC control signal blocks
T_p	[s]	Prediction horizon of MPC
t_0	[s]	Present point of time of MPC
T_c	[s]	Closed-loop time-constant (Skogestad)
$T_{delay_{pump}}$	[s]	Approximate time-delay of basin pump
T_i	[s]	Integral time of PI controller
u	[L/s]	Control signal to the pump
$ \dot{u} $	[(L/s)/min]	Rate of change of pump control signal

Abbreviations

Abbr.	Full name
EKF	Extended Kalman Filter
FT	Flow Transmitter (sensor)
LC	Level Controller
LT	Level Transmitter (sensor)
MPC	Model-Predictive Control
PID	Proportional + Integral + Derivative
wrrf	water resource recovery facility

References

- J. Bolmstedt. *Controlling the Influent Load to Wastewater Treatment Plants*. Lic. thesis, Lund University, Sweden, 2004.
- L. Grüne and J. Pannek. *Nonlinear Model Predictive Control*. Springer-Verlag, London, 2011. doi:10.1007/978-0-85729-501-9.
- F. Haugen and B. Lie. Relaxed Ziegler-Nichols Closed Loop Tuning of PI Controllers. *Modeling, Identification and Control*, 34(2):83–97, 2013. doi:10.4173/mic.2013.2.4.
- D. E. Seborg, Th. F. Edgar, and D. A. Mellichamp. *Process Dynamics and Control*. John Wiley and Sons, 2004.
- D. Simon. *Optimal State Estimation*. Wiley, 2006. doi:10.1002/0470045345.
- S. Skogestad. Simple analytic rules for model reduction and PID controller tuning. *Journal of Process Control*, 14, 2003. doi:10.1016/S0959-1524(02)00062-8.
- P. J. van Overloop, R. R. Negenborn, B. De Schutter B, and N. C. van de Giesen. Predictive control for national water flow optimization in the netherlands. *Intelligent Systems, Control and Automation, Springer*, 42, 2010.

Simulation of the Pyrolysis Process: from Plastic Waste to Environmental Friendly Fuel

Marius Andersen^a, Kai Arne Sætre^b, Siw Fredriksen^b, Carlos Pfeiffer^a

^aUniversity College of Southeast Norway, ^bNorner AS,

mariusandersen87@outlook.com,
carlos.pfeiffer@usn.no

kai.saetre@norner.no,

Siw.Fredriksen@norner.no,

Abstract

How to deal with plastic waste is an important question, as it may affect both the climate and environment. A method that may be beneficial to dispose some kinds of plastic waste that cannot be material-recycled is thermochemical conversion and, more specifically, pyrolysis. To be able to optimize such a process models are important. This paper describes the results of a study where the main aim was to identify and compare existing published models identifying the important variables regarding the pyrolysis of Polyethylene (PE) and Polypropylene (PP), published in open literature, and to compare these methods with experimental data. Several models were found, of which three were chosen for implementation and further evaluation.

Two of the methods propose the use of lumped kinetic models to predict the product-composition as a function of time and temperature, while the third method uses also the particle size of the polymer as an input-variable. Comparison with analytical data shows that the models performs well when the assumptions and implications behind them are taken into account.

Keywords: *Plastic Pyrolysis, Process Simulation, Feedstock Recycling*

1 Introduction

Since plastics appeared in the 1950s they have changed our world forever. They are cheap to produce, durable and have a wide array of uses. However, if not being recycled properly, plastics can also be a threat to the environment, not to mention a waste of energy and materials (Pinto *et al*, 1999).

An option proposed, and used to a certain degree, is to recycle plastics is pyrolysis (Beili *et al*, 2009). This is a process where the material is heated until decomposition in absence of oxygen, yielding a wide array of lighter hydrocarbons.

Experimental results have shown that the final composition of products is highly dependent on the purity and type of polymer(s), temperature and

residence time (A. Demirbaş 2005). Optimizing these relationships to give a good tradeoff between cost and product-value is important to make a process profitable, and to do this accurately and robust models are needed.

This paper describes the results of a thesis (Andersen, 2017) where one of the main aims was to find methods to model such a process, to implement and simulate these, and to compare them with experimental data. The experimental data were obtained from reported pyrolysis experiments using virgin HDPE and PP (Azuibuke, 2017).

1.1 Literature review

From a literature review several models were found and evaluated for use in the study (Csukas *et al*, 2013; Ding *et al*, 2012; Westerhout *et al*, 1997; Zhang *et al*, 2015). All the reported models were based on parameters derived experimentally. Pyrolysis is a complex process; thus, these models were based on different kinds of assumptions.

Two simplifications that were often found in the reported models were the assumption of a pure feed-material consisting of only one or two polymers, and the neglect of heat and mass transfer resistance.

The first simplification is invalid if waste-plastic is used, as there will be some food residue, labels and possibly non-desired polymers present.

Studies (Urionabarrenechea, 2011) suggest that different polymers and/or other impurities in the feed may have a strong impact on the reaction mechanism.

Polymers have, in general, a high heat transfer resistance (i.e., thermal conductivity is low). Therefore, it was decided to investigate the validity of the second assumption further. Heat transfer resistance effects may be negligible for small particles, however plastic waste is typically ground only to a suitable size for handling in bulk quantities. and the degree of grinding is a trade-off between cost and benefit.

For the reactions' pathways models, the kinetic parameters used in this article were derived on the assumption of a simple second order Arrhenius-mechanism.

2 Materials and Methods

2.1 Kinetic Models

The commonly used polyethylene and polypropylene are stable molecules, thus the overall-reaction breaking them down to smaller products will be an endothermic one. An equation for the reaction for the decomposition rate can be written as follows, assuming an Arrhenius mechanism:

$$\frac{dX_p}{dt} = -X_p \cdot A_0 \cdot e^{-\frac{E_a}{R \cdot T}} \quad (1)$$

where X_p is the mass fraction of the polymer, A_0 is the pre-exponential factor and E_a is the activation energy for the reaction. It is possible to determine the parameters for the total reaction this by measuring the decomposition rate at different temperatures to provide an Arrhenius-plot.

In order to model the production of valuable fuel components, that is, hydrocarbons with carbon numbers in the range of approx. C4 – C25 a more detailed approach is needed, due to the fuel products decomposing under the same conditions as the polymer. One equation for each of these reactions can be created, but this would be an enormous task due to the sheer number of species, thus reactions, involved.

2.2 Three Lump model for HDPE and PP/PE mix

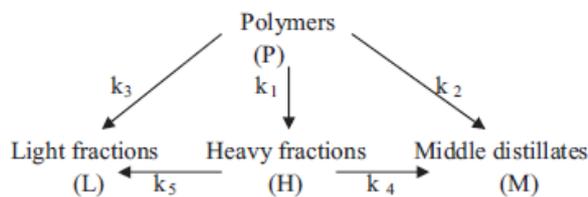


Figure 1 - Proposed reaction pathway by (Ding *et al*, 2012).

As can be seen in Figure 1 the model proposed by (Ding *et al*, 2012). is based on the assumption of a reaction pathway consisting of three parallel primary reactions (k_1 , k_2 , k_3), and two secondary reactions (k_4 , k_5), leading to the formation of three product-lumps.

Having defined the different bulks their rates of formation/consumption may be calculated using Equation (2-6) where X means mass fraction of polymer (P), light (L), medium (M) and heavy (H) product:

$$\frac{dX_p}{dt} = -X_p \cdot (k_1 + k_2 + k_3) \quad (2)$$

$$\frac{dX_H}{dt} = X_p \cdot k_1 - X_H \cdot (k_4 + k_5) \quad (3)$$

$$\frac{dX_M}{dt} = X_p \cdot k_2 + X_H \cdot k_4 \quad (4)$$

$$\frac{dX_L}{dt} = X_p \cdot k_3 + X_H \cdot k_5 \quad (5)$$

Where k_1 - k_4 are expressed as:

$$k_i = A_{0i} \cdot e^{-\frac{E_{a_i}}{R \cdot T}} \quad (6)$$

Assuming there is only polymer present initially, the initial conditions can be expressed as:

$$X_{P(0)} = 1, X_{H,0} = X_{M,0} = X_{L,0} = 0$$

In order to estimate the Arrhenius parameters, some experimentally-derived rate constants were published in the same article, and an Arrhenius-plot were done on these.

The results of these plots showed a variable degree of linearity. An explanation for this may a reaction order that varies as a function of conversion, as suggested by (Westerhout *et al*, 1997).

2.3 Six Step model for Polypropylene

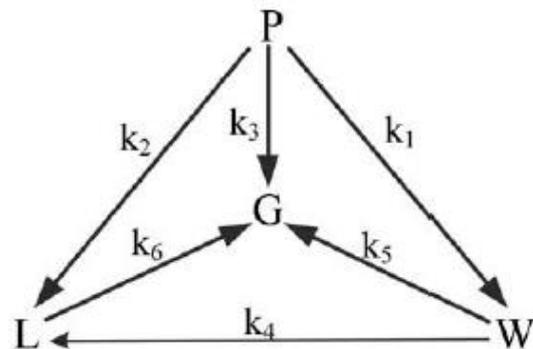


Figure 2 Reaction pathway for PP suggested by (Zhang *et al*, 2015)

This model, proposed by (Zhang *et al*, 2015) is based on the assumption of 3 products lumps, gas, liquid and wax, 3 primary reactions (k_1 , k_2 , k_3) and 3 secondary reactions (k_4 , k_5 , k_6).

This reaction pathway thus gives the following set of equations where X denotes mass fraction of the polymer (P), liquid (L), wax (W) and gas (G):

$$\frac{dX_p}{dt} = -X_p \cdot (k_1 + k_2 + k_3) \quad (7)$$

$$\frac{dX_L}{dt} = X_p \cdot k_2 + X_W \cdot k_4 - X_L \cdot k_6 \quad (8)$$

$$\frac{dX_W}{dt} = X_p \cdot k_1 - X_W \cdot (k_5 + k_6) \quad (9)$$

$$\frac{dX_G}{dt} = X_p \cdot k_3 + X_L \cdot k_6 + X_L \cdot k_5 \quad (10)$$

2.4 Heat transfer model

The third model implemented in this study is a heat transfer model (Westerhout *et al*, 1997). The model is based on an energy balance of a single particle, as can be seen below.

$$\rho_p c_{pp} \frac{\partial T_p}{\partial t} = k_p \frac{1}{x^\gamma} \frac{\partial(x^\gamma \frac{\partial T_p}{\partial x})}{\partial x} - \frac{\partial \xi}{\partial t} \rho_{p0} \Delta H_r \quad (11)$$

Where the meaning of the symbols is explained in Table 1:

Table 1 Explanation of symbols used in model

Symbol	Description [Unit]
ρ_p	Density of the particle [kg/m ³]
ρ_{p0}	Density at t=0 [kg/m ³]
x	Radial coordinate of the particle [m]
ξ	Conversion [-]
γ	Geometric factor [-]
H_r	Heat of reaction [J/kg]
c_{pp}	Heat capacity [J/kg K]
k_p	Conductive transfer coefficient [W/m ² K]
T_p	Temperature of particle [K]
T_s	Temperature of surroundings [K]
h	Convective transfer coefficient [W/m ² K]
ϵ	Emissivity [W/m ² K ⁴]

Assuming the volume to be constant x^γ , and only the density decreasing as the reactions occur within the particle, e.g. particle getting more porous as the reaction takes place through the constant volume, the x^γ can be taken out of the derivative and canceled against $1/x^\gamma$.

To simplify:

$$\frac{k_p}{\rho_p c_{pp}} = r \quad (12)$$

And

$$\rho_p = \rho_0(1 - \xi) \quad (14)$$

Assuming Arrhenius kinetics, the mass balance can then be formulated as:

$$\frac{\partial \xi}{\partial t} = A_0 * e^{\frac{-Ea}{RT_p}} * (1 - \xi) \quad (15)$$

With the following boundary and initial conditions:

$$t = 0 \forall x = T_p = T_0 \cap \xi = 0 \quad (16)$$

$$x = 0 \forall t \frac{\partial T_p}{\partial x} = 0 \quad (17)$$

$$x = r_p \forall t \frac{\partial T_p}{\partial x} = \frac{h}{k_p} (T_s - T_{p,x=rp}) + \epsilon \sigma (T_s^4 - T_{p,x=rp}^4) \quad (18)$$

After combining, the equation is then discretized in time and space:

$$T_{p,n}^{i+1} = bT_{n+1}^i - (1 - 2b)T_{p,n}^i + bT_{p,n-1}^i - \frac{\rho_{p0} r \Delta H_r}{k_p} (A_0 * e^{\frac{-Ea}{T_{p,n}^i}} * (1 - \xi^i)) \quad (19)$$

The initial condition is

$$T(0,x) = T_0$$

The Neumann boundary conditions are:

$$x = r_p \forall t \frac{\partial T_p}{\partial x} = \frac{h}{k_p} (T_s - T_{p,x=rp}) + \epsilon \sigma (T_s^4 - T_{p,x=rp}^4) \quad (20)$$

$$T_{n+1}^i = -2\Delta x \frac{h}{k_p} (T_s - T_{p,n}^i) + \epsilon \sigma (T_s^4 - T_{p,n}^i) + T_{n-1}^i \quad (21)$$

3 Results and Discussions

3.1 Kinetic Models

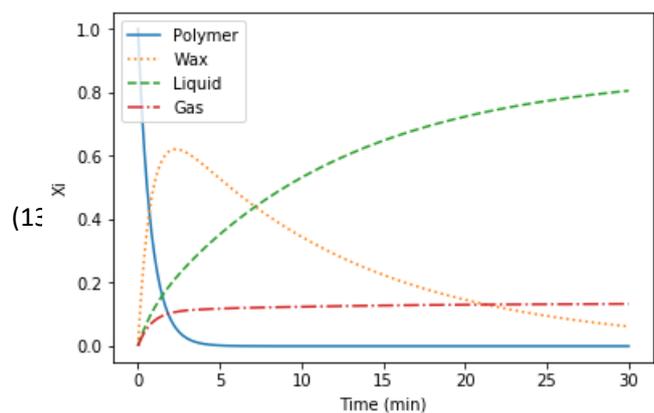


Figure 3 - Results of kinetic models for PP at 400 C.

As can be seen in Figure 3 the model gives results in the range of what can be expected from pyrolysis at these temperatures. These plots, being a function of both the primary and secondary reactions is difficult to validate without suitable experimental data tailored to fit the model. This is especially due to the fact that two

similar phases at high temperatures are present (wax and liquid), and because the exact composition and physical properties were poorly defined in the literature where the kinetic data was sourced from.

In order to validate the model, TGA (ThermoGravimetric Analysis) data were used. The data was sourced from the thesis of (A. Azuibuke, 2017). Thermogravimetric analysis works by measuring the mass of a sample while the temperature is ramped.

Hence, if the rate of decomposition of the polymer can be validated against the TGA-data, this should at least validate that the total magnitude of the primary reactions is correct.

Part of the difference between the model predicted TGA and the experimental results, can be attributed to the difference in conditions in a TGA meter and a pyrolysis reactor/autoclave which the kinetic data has been sourced from.

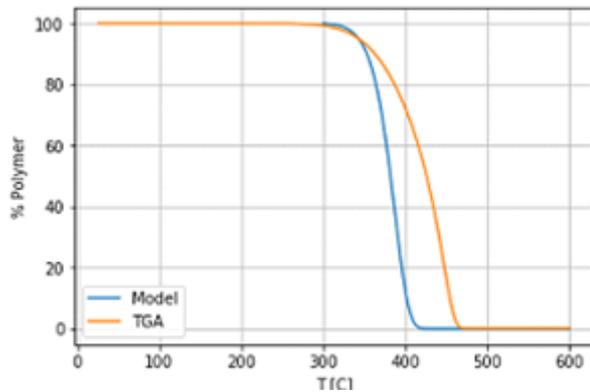


Figure 4 - Comparison between PP model and TGA. 10°C/min rate.

The comparison of experimentally obtained TGA-data with the TGA modeling results from the 3-step model for PP are shown in Figure 6

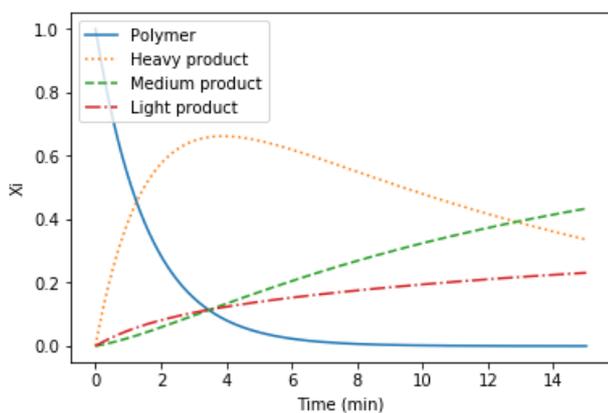


Figure 5 - Results of kinetic models for HDPE at 450 C.

Results of the HDPE-model can be seen in Figure 5. As can be seen, it decomposes somewhat slower, even though the temperature is higher than for the PP-simulation. This is can be explained due to PP having a

more branched structure than PE, making it more prone to breaking apart.

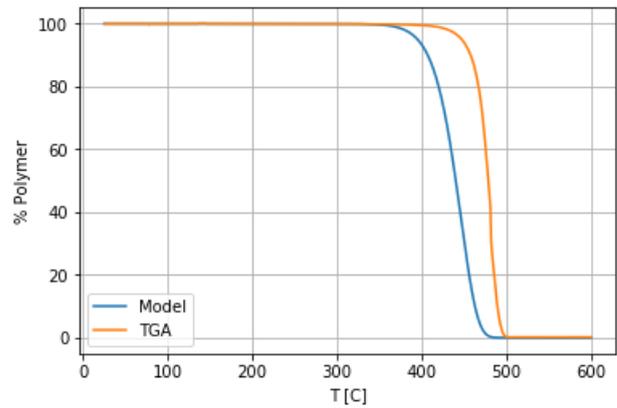


Figure 6 - Comparison between HDPE model and TGA. 10°C/min rate.

The HDPE-model was also validated against TGA-data sourced from (A. Azuibuke, 2017). As can be seen in Figure 6 it shows a similar magnitude of accuracy as the PP-model

3.2 Thermal model

As with the pure kinetic model the thermal model were also compared with TGA-data from (A. Azuibuke, 2017). The polymer pellets were almost spherical and the average diameter (1-3mm) as used as a parameter in the model. In Figure 7 the results of a comparison between the model and TGA results can be seen for HDPE.

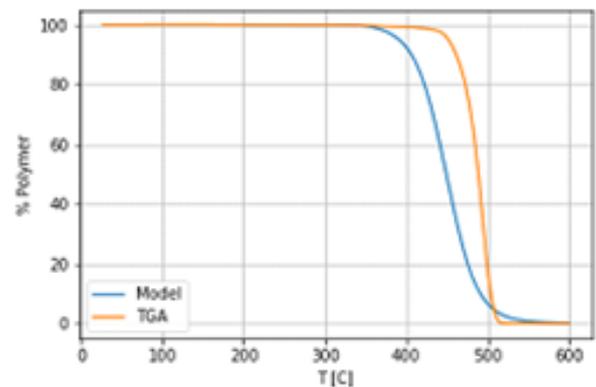


Figure 7 - Comparison between HDPE model and TGA. 10°C/min rate.

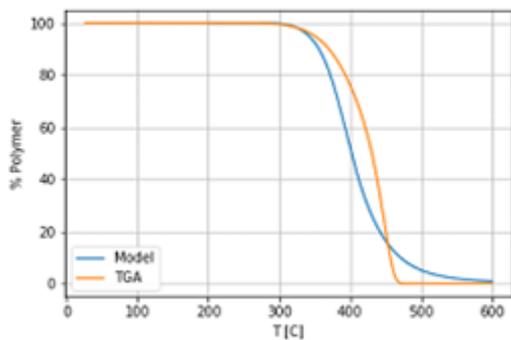


Figure 8 - Comparison between PP model 1 and TGA. 10°C/min rate.

In Figure 8 the results of the comparison of between the model and TGA-results can be seen for PP. Since both the pure kinetic model and this model is using the same kinetic parameters this implies that it is important to take heat transfer resistance in to account when modeling polymer pyrolysis. It is probably also possible that heat transfer resistance will affect the composition of the products as many of the species will be prone to decompose by means of the same kind of mechanisms as the polymer itself.

In Table 2 the % deviation between the thermal models and the TGA-results can be seen.

Table 2 - % Deviation of 90% conversion temperature between models and TGA for two heating rates.

Polymer	Rate [°C/min]	
	10	20
PP - Model	1.74	6.20
HDPE - Model	2.37	6.64

4 Conclusions

It can be concluded that the decomposition of pure polymers by pyrolysis can be modeled with a good (~90-95%) accuracy using models can be found in open literature. With larger particles(1mm+) and/or high temperature it is also important to model heat transfer resistance to get accurate results. If more accurate results are sought more complex models will be needed to account for factors such as variable reaction order, feed composition and structure, and in general the complexity of the multiple radical type of reactions that makes up the pathway between reactant and product.

The models presented in this report could be used together with the energy balance, operation cost estimations and products values, to find the optimal residence time that maximizes the profit for the pyrolysis process. However, to estimate this further work is recommended as the product fractions would probably have to be more precisely defined in terms of

fuel-value, and the models' robustness towards non-idealities such as impurities commonly found in plastic waste should be evaluated.

Bibliographic References

- M. Andersen, Process simulation of plastic waste to environmental friendly fuel, Master Thesis, University College of Southeast Norway, 2017
- A. Azubuike, Experimental study of catalysis in plastic pyrolysis, Master Thesis, University College of Southeast Norway, 2017.
- W. Beili, M. Fukushima, K. Wakai, H. Ibe, S. Ito "Current issues of oil reclamation system for recycling municipal waste plastic containing PVC and PET in Japan." The 5th symposium on feedstock and mechanical recycling of polymeric materials (ISFR2009). 2009.
- B. Csukas, M. Varga, N. Miskolczi, S. Balogh, A. Angyal, and L. Bartha, "Simplified dynamic simulation model of plastic waste pyrolysis in laboratory and pilot scale tubular reactor," *Fuel processing technology*, vol. 106, pp. 186-200, 2013.
- A. Demirbaş, "Recovery of Chemicals and Gasoline-Range Fuels from Plastic Wastes via Pyrolysis", *Energy Sources* 27:14, pp. 1313-1319, 2005.
- F. Ding, L. Xiong, C. Luo, H. Zhang, and X. Chen, "Kinetic study of low-temperature conversion of plastic mixtures to value added products," *Journal of Analytical and Applied Pyrolysis*, vol. 94, pp. 83-90, 2012.
- F. Pinto, P. Costa, I. Gulyurtlu, I Cabrita "Pyrolysis of plastic wastes. 1. Effect of plastic waste composition on product yield", *Journal of Analytical and Applied Pyrolysis*, vol 51, pp. 39-55, 1999.
- A. Urionabarrenechea, "Chemical recycling of municipal packaging waste by pyrolysis," Phd, University of the Basque Country, 2011.
- R. Westerhout, R. Balk, R. Meijer, J. Kuipers, and W. Van Swaaij, "Examination and evaluation of the use of screen heaters for the measurement of the high temperature pyrolysis kinetics of polyethylene and polypropene," *Industrial & engineering chemistry research*, vol. 36, pp. 3360-3368, 1997.
- R. Westerhout, J. Waanders, J. Kuipers, and W. Van Swaaij, "Kinetics of the low-temperature pyrolysis of polyethylene, polypropene, and polystyrene modeling, experimental determination, and comparison with literature models and data," *Industrial & Engineering Chemistry Research*, vol. 36, pp. 1955-1964, 1997.
- H. Zhang, F. Ding, C. Luo, and X. Chen, "Kinetics of the Low Temperature Conversion of Polypropylene to Polypropylene Wax," *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects*, vol. 37, pp. 1612-1619, 2015.

Intelligent Multimodel Simulation in Decomposed Systems

Esko K. Juuso

Control Engineering, Faculty of Technology, University of Oulu, Finland, esko.juuso@oulu.fi

Abstract

Intelligent methodologies provide a good basis for multimodel simulation. Small, specialised systems have a large number of feasible solutions, but developing truly adaptive, and still understandable, systems for highly complex systems require domain expertise and more compact approaches at the basic level. The nonlinear scaling approach extends the application areas of linear methodologies to nonlinear modelling and reduces the need for decomposition with local models. Fuzzy set systems provide a good basis for understandable models for decomposed systems. Data-based methodologies are suitable for developing these adaptive applications via the following steps: variable analysis, linear models and intelligent extensions. Complex problems are solved level by level to keep the domain expertise as an essential part of the solution.

Keywords: nonlinear systems, intelligent methods, composite local modelling, linguistic equations, fuzzy logic

1 Introduction

Models are understood as relationships between variables and used to predict properties or behaviour of the system. Variable interactions and nonlinearities are important in extending the operation areas (Juuso, 2004a). Phenomenological models based on physics, chemistry and mathematics require domain expertise (Figure 1). Linear methodologies extended with principal components (Jolliffe, 2002; Gerlach et al., 1979) and semi-physical models (Ljung, 1999) provide a feasible solution for many applications. Nonlinearities have been handled commonly with interaction and quadratic terms (Box and Wilson, 1951). Linear parameter varying (LPV) extend these solutions to decomposed systems (Hjartarson et al., 2015; Theis et al., 2018).

Artificial neural networks (ANNs) starting from (Rumelhart et al., 1986) continue this by using more complex architectures with deep learning for complicated interactions within different sources of varying data (Schmidhuber, 2015). Big data analytics and deep learning are in high focus in data science (Najafabadi et al., 2015).

Knowledge-based information can be handled with fuzzy set systems introduced by Zadeh (1965): numerous methodologies have been developed, see (Takagi and Sugeno, 1985; Driankov et al., 1993; Dubois et al., 1999), and combined with neural networks (Fullér, 2000). Different fuzzy approaches can be efficiently combined (Juuso, 2014).

First order ordinary differential equations are solved by numerical integration and special solutions have been developed for identification (Ljung, 1999). These approaches, which are also used in fuzzy set systems (Babuška and Verbruggen, 2003) and low complexity neural networks (Sahoo et al., 2013), define structures for hybrid dynamic models (Figure 1). Local models need to be combined in complex systems (Sontag, 1981; Ljung, 2008; Jardine et al., 2006).

The linguistic equation (LE) approach originates from fuzzy set systems (Juuso and Leiviskä, 1992): rule sets are replaced with equations, and meanings of the variables are handled with scaling functions which have close connections to membership functions (Juuso, 1999a). The nonlinear scaling technique is needed in constructing nonlinear models with linear equations (Juuso, 2004a). Constraints handling (Juuso, 2009a) and data-based analysis (Juuso and Lahdelma, 2010), improve possibilities to update the scaling functions recursively (Juuso, 2011). The LE approach together with knowledge-based systems, neural networks and evolutionary computation form the computational intelligence part (Figure 1).

Three levels of smart adaptive systems (SAS) are identified in (Anguita, 2001): (1) adaptation to a changing environment, (2) adaptation to a similar setting without explicitly being ported to it, and (3) adaptation to a new or unknown application. The smart use of intelligence by integrating specific intelligent systems is essential in the development of complex adaptive applications. Implementation of smart adaptive systems on silicon has been proposed to adapt perceptual and cognitive tasks autonomously to the changing environment (Valle, 2004).

Technically, an automatic black box modelling could be possible in various big data problems by using combinations of these methodologies. The domain expertise is an essential part in integrated solutions to understand and assess the applicability. This paper classifies modelling methodologies and focuses on the nonlinear scaling and integrates the LE approach into the modelling applications for complex systems. Various applications are shortly discussed.

This paper focuses on the LE modelling approach enhanced with statistical and knowledge-based methodologies within decomposed systems (Section 2). Different methodologies are combined in the multimodel LE simulation (Section 3). Various applications summarised in Section 4 are discussed in Section 5. Conclusions and future research presented in Section 6.

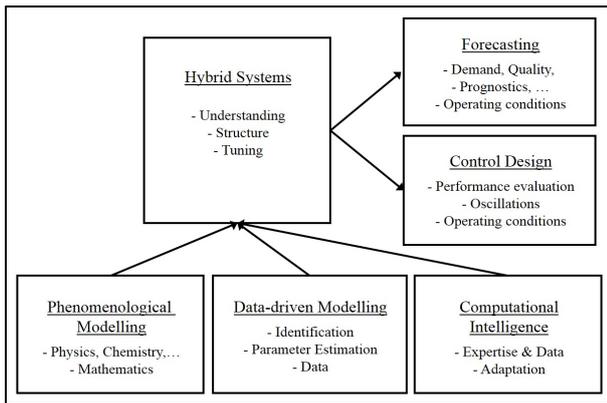


Figure 1. Methodologies and application types of modelling and simulation, modified from (Juuso, 2004b)

2 Modelling methodologies

The modelling methodologies include statistical analysis, knowledge-based methodologies and decomposition solutions.

2.1 Statistical analysis

Nonlinear scaling and steady-state statistical modelling with linear methodologies are the basis of the LE modelling. Dynamic modelling introduces additional model structures.

2.1.1 Nonlinear scaling

The nonlinearities of the process are handled by the nonlinear scaling of the variables. The scaling functions are monotonously increasing functions $x_j = f(X_j)$ where x_j is the variable and X_j the corresponding scaled variable. The function $f()$ consist of two second order polynomials, one for the negative values of X_j and one for the positive values, respectively. The corresponding inverse functions $x_j = f^{-1}(X_j)$ based on square root functions are used for scaling to the range $[-2, 2]$, denoted linguistification. In LE models, the results are scaled to the real values by using the function $f()$. (Juuso, 2004a)

The support area is defined by the minimum and maximum values of the variable, i.e. the support area is $[\min(x_j), \max(x_j)]$ for each variable $j, j = 1, \dots, m$. The central tendency value, c_j , divides the support area into two parts, and the core area is defined by the central tendency values of the lower and the upper part, $(c_l)_j$ and $(c_h)_j$, correspondingly. This means that the core area of the variable j defined by $[(c_l)_j, (c_h)_j]$ is within the support area. The parameters of the functions are extracted from measurements by using generalised norms and moments (Juuso and Lahdelma, 2010).

2.1.2 Steady-state modelling

The steady-state simulation models are linear *multiple input, multiple output (MIMO)* models $\vec{y} = F(\vec{x})$, where the output vector $\vec{y} = (y_1, y_2, \dots, y_n)$ is calculated by a lin-

ear function F from the input vector $\vec{x} = (x_1, x_2, \dots, x_m)$. Statistical modelling in its basic form uses linear regression for solving coefficients for a linear function. Linear methodologies are suitable for large multivariable systems. Quadratic and interactive terms are not used here. Principal components compress the data by reducing the number of dimensions with a minor loss of information (Jolliffe, 2002). Partial least squares regression (PLS) is an extension of these ideas (Gerlach et al., 1979). Known semi-physical models of inputs are important in linear modelling, see (Ljung, 1999).

2.1.3 Dynamic modelling

Data-driven modelling with parametric models, also known as identification (Ljung, 1999), is the key methodology in the dynamic modelling (Figure 1). Nonlinear scaling reduces the number of input and output signals needed for the modelling of nonlinear systems. For the default LE model, all the degrees of the polynomials become very low:

$$Y(t) + a_1 Y(t-1) = b_1 U(t-n_k) + e(t) \quad (1)$$

for the scaled variables Y and U . Phenomenological models can be integrated with these solutions.

2.2 Knowledge-based methodologies

Knowledge-based information can be introduced by *fuzzy logic* which emerged from approximate reasoning: the connection of fuzzy rule-based systems and *artificial intelligence (AI)* is clear, e.g. the vocabulary of AI is kept in fuzzy logic (Dubois et al., 1999). *Fuzzy set theory* first presented by Zadeh (1965) form a conceptual framework for linguistically represented knowledge.

Domain expertise can be combined with statistical models with following fuzzy methodologies:

- *Linguistic fuzzy models* (Driankov et al., 1993), where both, the antecedent and consequent are fuzzy propositions, suit well for natural language, heuristics and common sense knowledge.
- *Takagi-Sugeno (TS) fuzzy models* (Takagi and Sugeno, 1985), where each consequent $y_i, i = 1, \dots, n$, is a crisp function of the antecedent variables \vec{x} , can integrate local linear models. A smoothing technique is needed for drastically different local models (Babuška, 1998).
- *Singleton models* can be regarded as special cases of both the linguistic fuzzy models and the TS fuzzy models.

The *extension principle* is the basic generalisation of the arithmetic operations if the *inductive mapping* $F(x_j)$ is a monotonously increasing function of the input. The interval arithmetic presented by Moore (1966) is used together with the extension principle on several membership

α -cuts of the fuzzy number x_j for evaluating fuzzy expressions (Buckley and Qu, 1990; Buckley and Hayashi, 1999; Buckley and Feuring, 2000). The fuzzy sets can be modified by intensifying or weakening modifiers (De Cock and Kerre, 2004; Le and Tran, 2018)

Type-2 fuzzy models introduced by Zadeh (1975) take into account uncertainty about the membership function (Mendel, 2007; Sadeghian et al., 2013).

Dynamic fuzzy models use the same parametric structures as the statistical models (Babuška and Verbruggen, 2003; Sahoo et al., 2013).

2.3 Decomposition

Decomposition is needed to extend the solutions to different subprocesses, process phases, phenomena and multiple operating conditions.

2.3.1 Subsystems

Modelling problems are divided into smaller parts for developing separate models for subprocesses or different stages in the process operation interconnected with process streams. In addition to spatial or logical blocks, the decomposed modelling can be based on different frequency ranges. Cluster analysis provides hundreds of algorithms for the data-driven analysis (Xu and Tian, 2015). The mixed systems may also include models based on the first principles. Clustering is used in the data analytics to find feasible areas for local models.

2.3.2 Composite local models

The composite local model approach constructs a global model from local models, which usually are linear approximations of the nonlinear system in different neighbourhoods. If the partitioning is based on a measured regime variable, the partitioning can be used in weighting the local models. In linear parameter varying (LPV) models, the matrices of the state-space models depend on an exogenous variable measured during the operation (Hjartarson et al., 2015; Theis et al., 2018). Piecewise affine (PWA) systems extend the local linear models to a polyhedral partition where the models can be state-space or parametric models (Christoffersen, 2007). The model switches between different modes as the state variable varies over the partition. However, a high number of local models brings an overfitting risk.

2.3.3 Intelligent systems

Composite local models enhanced with fuzzy set systems form feasible solutions to handle partially overlapping models. Fuzzy models combine local modelling approaches and facilitate gradual changes. The smoothing problem around the submodel borders of Takagi-Sugeno (TS) fuzzy models needs special techniques, e.g. smoothing maximum, or by making the area overlap very strong. The ANFIS method (Jang, 1993) is widely used in the tuning of TS fuzzy systems, but it increases the overlap of clusters and destroys the meanings of the individual linear models, e.g. the role of some submodels may transform

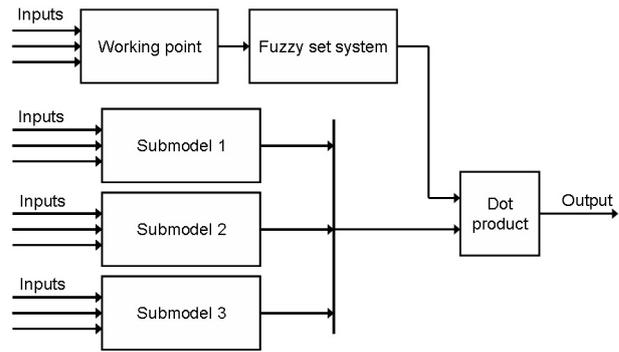


Figure 2. Multimodel LE system with a fuzzy decision module (Juuso, 2009b).

into a part of a smoothing algorithm. A Kalman-based learning algorithm has been proposed for online TS identification (Vafamand et al., 2018).

In multiple neural network systems, the task decomposition and an ensemble of redundant networks improve generalisation. Ensemble averaging is the process of creating multiple models and combining them to produce a desired output. Multimodel ensemble methods are important in deep neural networks (Xiao et al., 2018).

3 Multimodel LE simulation

A multimodel approach has been developed for combining specialised linguistic equation (LE) submodels where the nonlinear scaling need to be done by a set of functions due to very strong nonlinearities. Additional properties are achieved because equations and delays are allowed to vary between different submodels. In this multimodel approach, the working area is defined by a separate working point model. The submodels are developed using the case-based modelling approach.

3.1 Fuzzy LE models

The multimodel system contains several submodels and a fuzzy decision system for selecting and weighting suitable models for each situation using several working point variables. If several inputs are combined into a single working point index, the fuzzy set system is reduced to a fuzzification block.

Linguistic Takagi-Sugeno fuzzy models (LTS) belong to this class of models with one limitation: the fuzzy partition is defined with the same variables as the models. As LE models are nonlinear, the local models are also nonlinear. LTS models can be developed and tuned with the same methods as the normal TS models with one difference: the variable values are scaled with the nonlinear scaling functions. Correspondingly, the LTS models extend the normal LE model by handling the equation part with a fuzzy set system. (Juuso, 2009b)

Table 1. Steady-state LE model applications.

<i>Case</i>	<i>Application area</i>	<i>Modelling</i>
Electric furnace	DSS for process design	Nonlinear models transformed to LE models Interactions
Lime kiln	Feedforward control	Fuel feed in changing capacity conditions
Solar collector field	Control adaptation	Steady-state working point model: Irradiation, temperature difference, special cases with fuzzy set systems
Continuous cooking	Quality control	Quality forecasting
Fatigue	Stress contributions	LE based stress-cycle curve 2nd order and logarithmic scaling
Water treatment	Feedforward control	Turbidity for control Forecasting residual aluminium
Wastewater treatment	Diagnostics	Operating conditions

Table 2. Dynamic LE model applications.

<i>Case</i>	<i>Application area</i>	<i>Modelling</i>
Gas furnace	Modelling	Tuning: training, validation, testing
Solar collector field	Controller tuning	Time varying transport delay Cloudy periods
Fatigue	Forecasting fatigue risk	Cumulative sum of stress contributions Rolling mill, LHD machines
Water treatment	Controller tuning	Water quality indicator Water circulation, Drinking water
Condition monitoring	Prognostics	Recursive tuning

3.2 Nonlinear parameter (NPV) models

The LE models are defined by the parameters of the scaling functions and the coefficients of the interaction models. The idea of the exogenous variables can be used for these parameters, which opens a set of new modelling approaches for the nonlinear parameter (NPV) varying models. There are three levels of complexity: (1) individual scaling functions are compressed or expanded, (2) the shape of the functions is modified, and (3) also the coefficients of the equations are modified.

Clustering methodologies are used for finding areas for the submodels. The clustering variables define the operating conditions are not necessarily included in the submodels.

3.3 Genetic tuning

Evolutionary computing is widely used to tune intelligent systems which incorporate expert knowledge with data. Genetic algorithms are well suited for LE models based on nonlinear scaling and linear interactions. The scaling functions handle efficiently the parameter constraints of the monotonously increasing second order polynomials and the whole system is configured with a set of parameters. (Juuso, 2009a)

4 Applications

Nonlinear scaling forms the basis for the LE modelling: an important benefit of the linear approach is that the models can be inverted, technically to any direction. The compact basic solution makes extensions to dynamic and case-based systems possible. Complex models for steady-state and dynamic systems can be built with the cascade and interactive structures.

4.1 Steady-state LE models

Steady-state LE models are mainly used in adaptation and feedforward control (Table 1). In most cases, the models include only a single linear equation. The first LE model developed for designing submerged arc furnaces was an exception which used well known relations represented by five equations (Juuso and Leiviskä, 1992). A steady-state LE model was developed in an early control application from the process measurements of a lime kiln (Juuso et al., 1997). For continuous cooking, a LE model has been developed for predicting the Kappa number, which is widely used quality variable (Leiviskä et al., 2001).

The working point model presented in (Juuso et al., 1998) is still an essential part of the model-based LE control of a solar power plant (Juuso and Yebra, 2013). Stress-cycle (S-N) curves, also known as Wöhler curves, are rep-

Table 3. Decomposed LE model applications.

<i>Case</i>	<i>Application area</i>	<i>Modelling</i>
Lime kiln	Fuel quality Adaptive control	Controller tuning by using multiple models
Solar collector field	Controller tuning for oil flow	Models for different operating conditions Distributed parameter models
Batch cooking	On-line forecasting	Three interactive models: alkali, lignin and dissolved solids
Fluidised bed granulation	Forecasting	Three interactive models: temperature, humidity and granular size
Fed-batch fermentation	On-line forecasting	Submodels of three growth phases, each including three interactive models, totally nine interactive models
Wastewater treatment	Detection of operating conditions	Three submodels: load, treatment and settling Trend analysis

resented by a linguistic equation (Juuso and Ruusunen, 2013).

In drinking water applications, models have been developed for forecasting and control (Tomperi et al., 2013). Operating conditions are detected in diagnosing the wastewater treatment process (Juuso and Laakso, 2013).

4.2 Dynamic LE models

The basic dynamic LE model is represented by the parametric model (1) with an appropriate number of variables. The approach was first tested in a gas furnace data provided by (Box and Jenkins, 1970). The dynamic models of the solar plant are based on test campaigns, which cannot be planned in detail because of changing weather conditions (Juuso, 2003a). The basic dynamic flotation model is the core of the quality indicator in water treatment (Ainali et al., 2002; Joensuu et al., 2005). A dynamic LE model has been used for the fatigue prediction in (Juuso and Ruusunen, 2013). In all these models, only one equation is needed. The applications are indirect measurements and controller tuning (Table 2). Drinking water applications focus on model-based control and forecasting (Tomperi et al., 2016). Trend analysis is important in the wastewater treatment (Tomperi et al., 2017).

4.3 Decomposition in LE models

The multimodel LE system can include several submodels and complex interactions (Table 3). All basic models are represented by the model (1) with an appropriate number of variables.

The model with a fuzzy decision module was first used for a lime kiln (Juuso, 1999b) and then for a solar thermal power plant (Juuso, 2003a). The lime kiln model had six operating areas defined by the production level and the trend of the fuel feed (increasing, decreasing). The model of the collector field includes four operating areas: start-up, low, normal and high operation. For handling special

situations in the solar plant, additional fuzzy models have been developed by using the Fuzzy-ROSA method (Juuso et al., 2000). Interactive dynamic models were needed in several cases: batch cooking (Juuso, 2003b), fluidised bed granulator (Mäki et al., 2004), industrial fed-batch fermenter (Saarela et al., 2003) and wastewater treatment (Juuso et al., 2009). Linguistic equations, neural networks and fuzzy modelling with several variants have been compared by using the process data obtained from the fed-batch fermenter.

4.4 Distributed parameter LE models

In the distributed parameter models, the solar collector field is divided into modules, where the dynamic LE models are applied in a distributed way (Juuso, 2004b). The same single equation model (1) with an appropriate number of variables is used in all modules. Element locations for partial differential equations (PDEs) are defined by the flow rate. In cloudy conditions, the heating effect can be strongly uneven.

5 Discussion

The nonlinear scaling methodology is the key in the extensions of the linear methodologies. Steady-state models form the basis and extended to dynamic applications with additional structures. Several steady-state and dynamic models are combined with fuzzy set systems. The ensemble averaging could be used in the neural computing with linear networks. Distributed parameter systems can use the same algorithms. All the applications discussed in Section 4 use the scaling functions explained in Section 2.1.1 have been developed before the current data-based analysis. Clustering is used in finding the areas of the submodels.

All parameters of the multimodel LE systems can be tuned with genetic algorithms. Constraints are taken into account in the coding which means that penalties are not needed in the optimisation.

The variable specific recursive analysis of the parameters of the scaling functions is feasible for the machine learning phase. The multimodel structure facilitates deep learning extensions.

6 Conclusions

The nonlinear scaling approach extends the application areas of linear methodologies to nonlinear modelling: the meanings of variables and interactions are analysed sequentially. Local nonlinear models reduce the need for decomposition with local models. The close connection to the fuzzy set systems provides a good basis for understandable models. Data-based methodologies are suitable for developing models for decomposed systems. Big data problems are solved level by level to keep the domain expertise as an essential part of the solution. The basic models are compact and additional properties, including dynamics, uncertainty and decomposition are included if needed.

Acknowledgment

The combined approach has been developed within the research program *Measurement, Monitoring and Environmental Efficiency Assessment (MMEA)* on the basis of several earlier and parallel projects.

References

- I. Ainali, M. Piironen, and E. Juuso. Intelligent water quality indicator for chemical water treatment unit. In *Proceedings of SIMS 2002 - the 43rd Scandinavian Conference on Simulation and Modelling, September 26-27, 2002, Oulu, Finland*, pages 247–252. Oulu, 2002.
- D. Anguita. Smart adaptive systems - state of the art and future directions for research. In *Proceedings of Eunit 2001 - European Symposium on Intelligent Technologies, Hybrid Systems and their implementation on Smart Adaptive Systems, July 13-14, 2001, Tenerife, Spain*, pages 1–4. Wissenschaftsverlag Mainz, Aachen, 2001.
- R. Babuška. *Fuzzy Modeling and Identification*. Kluwer Academic Publisher, Boston, 1998.
- R. Babuška and H. Verbruggen. Neuro-fuzzy methods for nonlinear system identification. *Annual Reviews in Control*, 27(1):73–85, 2003.
- G. E. P. Box and G. M. Jenkins. *Time Series Analysis, Forecasting and Control*. Holden Day, San Francisco, 1970.
- G. E. P. Box and K. B. Wilson. On the experimental attainment of optimum conditions. *Journal of the Royal Statistical Society. Series B*, 13(1):1–45, 1951.
- J. J. Buckley and T. Feuring. Universal approximators for fuzzy functions. *Fuzzy Sets and Systems*, 113:411–415, 2000.
- J. J. Buckley and Y. Hayashi. Can neural nets be universal approximators for fuzzy functions? *Fuzzy Sets and Systems*, 101:323–330, 1999.
- J. J. Buckley and Y. Qu. On using α -cuts to evaluate fuzzy equations. *Fuzzy Sets and Systems*, 38(3):309–312, 1990.
- F. J. Christophersen. Piecewise Affine Systems. In M. M. Gupta and T. Yamakawa, editors, *Optimal Control of Constrained Piecewise Affine Systems, Lecture Notes in Control and Information Sciences*, volume 359, pages 39–42. Springer, Berlin, Heidelberg, 2007. doi:10.1007/978-3-540-72701-9_4.
- M. De Cock and E. E. Kerre. Fuzzy modifiers based on fuzzy relations. *Information Sciences*, 160(1-4):173–199, 2004.
- D. Driankov, H. Hellendoorn, and M. Reinfrank. *An Introduction to Fuzzy Control*. Springer, Berlin, Germany, 1993.
- D. Dubois, H. Prade, and L. Ughetto. Fuzzy logic, control engineering and artificial intelligence. In H. B. Verbruggen, H.-J. Zimmermann, and R. Babuska, editors, *Fuzzy Algorithms for Control, International Series in Intelligent Technologies*, pages 17–57. Kluwer, Boston, 1999.
- R. Fullér. *Introduction to Neuro-Fuzzy Systems*. Advances in Soft Computing. Springer, 2000. 289 pp.
- R. W. Gerlach, B. R. Kowalski, and H. O. A. Wold. Partial least squares modelling with latent variables. *Anal. Chim. Acta*, 112(4):417–421, 1979.
- A. Hjartarson, P. Seiler, and A. Packard. LPVTools: A toolbox for modeling, analysis, and synthesis of parameter varying control systems. *IFAC-PapersOnLine*, 48(26):139–145, 2015. URL www.scopus.com.
- J.-S. R. Jang. ANFIS: Adaptive-Network-based Fuzzy Inference Systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(3):665–685, 1993.
- A. K. S. Jardine, D. Lin, and D. Banjevic. A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, 20(7):1483–1510, 2006.
- I. Joensuu, M. Piironen, and E. Juuso. Dynamic simulator for dosing of water treatment chemicals. *Computer-aided chemical engineering, European Symposium on Computer Aided Process Engineering-15 (Escape-15), Barcelona, Spain, May 29 - June 1, 2005*, 20A:301–306, 2005.
- I. T. Jolliffe. *Principal Component Analysis*. Springer, New York, 2 edition, 2002. 487 pp.
- E. Juuso and I. Laakso. Hybrid le systems for simulation of an activated sludge process. In *Proceedings - 8th EUROSIM Congress on Modelling and Simulation, EUROSIM 2013, 10-13 September, 2013, Cardiff, UK*, pages 124–129, 2013. doi:10.1109/EUROSIM.2013.32. URL www.scopus.com.
- E. Juuso and S. Lahdelma. Intelligent scaling of features in fault diagnosis. In *7th International Conference on Condition Monitoring and Machinery Failure Prevention Technologies, CM 2010 - MFPT 2010, 22-24 June 2010, Stratford-upon-Avon, UK*, volume 2, pages 1358–1372, 2010. URL www.scopus.com.

- E. Juuso and M. Ruusunen. Fatigue prediction with intelligent stress indices based on torque measurements in a rolling mill. In *10th International Conference on Condition Monitoring and Machinery Failure Prevention Technologies, CM 2013 - MFPT 2013, 18-20 June 2013, Krakow, Poland*, volume 1, pages 460–471, 2013. URL www.scopus.com.
- E. Juuso, T. Latvala, and I. Laakso. Intelligent analysers and dynamic simulation in a biological water treatment process. In I. Troch and F. Breitenecker, editors, *6th Vienna Conference on Mathematical Modelling - MATHMOD 2009, February 11-13, 2009, Argesim Report no. 35*, pages 999–1007. Argesim, 2009. ISBN 978-3-901608-35-3.
- E. K. Juuso. Fuzzy control in process industry: The linguistic equation approach. In H. B. Verbruggen, H.-J. Zimmermann, and R. Babuška, editors, *Fuzzy Algorithms for Control, International Series in Intelligent Technologies*, volume 14 of *International Series in Intelligent Technologies*, pages 243–300. Kluwer, Boston, 1999a. doi:10.1007/978-94-011-4405-6_10.
- E. K. Juuso. Intelligent dynamic simulation of a lime kiln with linguistic equations. In H. Szczerbicka, editor, *ESM'99: Modelling and Simulation: A tool for the Next Millennium, 13th European Simulation Multiconference, Warsaw, Poland, June 1-4, 1999*, pages 395–00, Delft, The Netherlands, 1999b. SCS.
- E. K. Juuso. Intelligent dynamic simulation of a solar collector field. In A. Verbraeck and V. Hlupic, editors, *Simulation in Industry, 15th European Simulation Symposium ESS 2003*, pages 443–449. SCS, Gruner Druck, Erlangen, Germany, 2003a.
- E. K. Juuso. Intelligent dynamic simulation of batch cooking. In *44th Scandinavian Conference on Simulation and Modeling SIMS 2003, September 18 -19, 2003*, pages 163–169. Malardalen University, Västerås, Sweden, 2003b.
- E. K. Juuso. Integration of intelligent systems in development of smart adaptive systems. *International Journal of Approximate Reasoning*, 35(3):307–337, 2004a. doi:10.1016/j.ijar.2003.08.008.
- E. K. Juuso. Dynamic simulation of a solar collector field with intelligent distributed parameter models. In B. Elmegaard, Jon Sparring, Kenny Erleben, and Kim Soerensen, editors, *Proceedings of SIMS 2004 - the 45th Scandinavian Conference on Simulation and Modelling, September 23-24, Copenhagen, Denmark*, pages 141–153. DTU, Lungby, Denmark, 2004b.
- E. K. Juuso. Tuning of large-scale linguistic equation (LE) models with genetic algorithms. In M. Kolehmainen, editor, *Revised selected papers of the International Conference on Adaptive and Natural Computing Algorithms - ICANNGA 2009, Kuopio, Finland, Lecture Notes in Computer Science*, volume LNCS 5495, pages 161–170. Springer-Verlag, Heidelberg, 2009a. doi:10.1007/978-3-642-04921-7_17.
- E. K. Juuso. Development of multiple linguistic equation models with Takagi-Sugeno type fuzzy models. In J. P. Carvalho, D. Dubois, U. Kaymak, and J. C. M. Sousa, editors, *Proceedings of 2009 IFSA World Congress / 2009 EUSFLAT Conference, July 20-24, 2009, Lisboa, Portugal*, pages 1779–1784. IFSA-EUSFLAT, 2009b. http://www.eusflat.org/publications_proceedings_IFSA-EUSFLAT_2009.php.
- E. K. Juuso. Recursive tuning of intelligent controllers of solar collector fields in changing operating conditions. In S. Bittani, A. Cenedese, and S. Zampieri, editors, *Proceedings of the 18th World Congress The International Federation of Automatic Control, Milano (Italy) August 28 - September 2, 2011*, pages 12282–12288. IFAC, 2011. doi:10.3182/20110828-6-IT-1002.03621.
- E. K. Juuso. Intelligent methods in modelling and simulation of complex systems. *Simulation Notes Europe SNE*, 24(1):1–10, 2014. doi:10.11128/sne.24.on.102221.
- E. K. Juuso and K. Leiviskä. Adaptive expert systems for metallurgical processes. In S.-L. Jämsä-Jounela and A. J. Niemi, editors, *Expert Systems in Mineral and Metal Processing, IFAC Workshop, Espoo, Finland, August 26-28, 1991, IFAC Workshop Series, 1992, Number 2*, pages 119–124, Oxford, UK, 1992. Pergamon.
- E. K. Juuso and L. J. Yebra. Model-based intelligent control of a solar energy collector field. In *Proceedings - 8th EUROSIM Congress on Modelling and Simulation, EUROSIM 2013, 10-13 September, 2013, Cardiff, UK*, pages 513–518, 2013. doi:10.1109/EUROSIM.2013.92. URL www.scopus.com.
- E. K. Juuso, T. Ahola, and K. Leiviskä. Fuzzy modelling of a rotary lime kiln using linguistic equations and neuro-fuzzy methods. In L. Foulloy, editor, *Proceedings of the 3rd IFAC Symposium on Intelligent Components and Instruments for Control Applications - SICICA'97, Annecy, France, June 9-11, 1997*, pages 579–584. Pergamon, 1997. ISBN 9780080426082.
- E. K. Juuso, P. Balsa, and L. Valenzuela. Multilevel linguistic equation controller applied to a 1 MW solar power plant. In *Proceedings of the ACC'98, Philadelphia, PA, June 24–26, 1998*, volume 6, pages 3891–3895. ACC, 1998.
- E. K. Juuso, D. Schauten, T. Slawinski, and H. Kiendl. Combination of linguistic equations and the fuzzy-ROSA method in dynamic simulation of a solar collector field. In L. Yliniemi and E. Juuso, editors, *Proceedings of TOOLMET 2000 Symposium - Tool Environments and Development Methods for Intelligent Systems, Oulu, April 13-14, 2000*, pages 63–77, Oulu, 2000. Oulun yliopistopaino.
- V.-H. Le and D.-K. Tran. Extending fuzzy logics with many hedges. *Fuzzy Sets and Systems*, 345:126 – 138, 2018. ISSN 0165-0114. doi:10.1016/j.fss.2018.01.014. Theme : Logic.
- K. Leiviskä, E. Juuso, and A. Isokangas. Intelligent modelling of continuous pulp cooking. In K. Leiviskä, editor, *Industrial Applications of Soft Computing, Studies in Fuzziness and Soft Computing*, pages 147–158. Springer, Heidelberg, 2001. doi:10.1007/978-3-7908-1822-2_10.
- L. Ljung. *System Identification - Theory for the User*. Prentice Hall, Upper Saddle River, N.J., 2nd edition, 1999.

- L. Ljung. Perspectives on system identification. In M. J. Chung and P. Misra, editors, *Plenary papers, milestone reports & selected survey papers, 17th IFAC World Congress, Seoul, Korea, July 6-11, 2008*, pages 47–59. IFAC, 2008. URL <http://www.ifac-papersonline.net/>.
- T. Mäki, E. Juuso, and K. Leiviskä. Fuzzy modelling and dynamic simulation of a fluidised bed granulator. In K. Leiviskä, editor, *Proceedings of AFNC'04 - the 2nd IFAC Workshop on Advanced Fuzzy/Neural, September 16-17, 2004, Oulu, Finland*, pages 133–138. Finnish Automation Society, Helsinki, 2004.
- J. M. Mendel. Advances in type-2 fuzzy sets and systems. *Information Sciences*, 177(1):84–110, 2007.
- R. E. Moore. *Interval Analysis*. Prentice Hall, Englewood Cliffs, NJ, 1966.
- M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic. Deep learning applications and challenges in big data analytics. *Journal of Big Data*, 2(1):1–21, Feb 2015. doi:10.1186/s40537-014-0007-7.
- D. E. Rummelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In D. E. Rummelhart and J. McClelland, editors, *Parallel Data Processing*, pages 318–362. M.I.T. Press, Cambridge, MA, 1986.
- U. Saarela, K. Leiviskä, E. Juuso, and A. Kosola. Modelling of a fed-batch enzyme fermentation process. In *IFAC International Conference on Intelligent Control Systems and Signal Processing, Faro, Portugal, April 8-11, 2003*. IFAC, 2003.
- A. Sadeghian, J.M. Mendel, and H. Tahayori. *Advances in Type-2 Fuzzy Sets and Systems: Theory and Applications*. Studies in Fuzziness and Soft Computing. Springer New York, 2013. ISBN 9781461466666. URL <https://books.google.fi/books?id=1sO7BQAAQBAJ>.
- H.K. Sahoo, P.K. Dash, and N.P. Rath. Narx model based nonlinear dynamic system identification using low complexity neural networks and robust H_∞ filter. *Applied Soft Computing*, 13(7):3324 – 3334, 2013. ISSN 1568-4946. doi:10.1016/j.asoc.2013.02.007.
- J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61 (Supplement C):85 – 117, 2015. ISSN 0893-6080. doi:<https://doi.org/10.1016/j.neunet.2014.09.003>. URL <http://www.sciencedirect.com/science/article/pii/S0893608014002135>.
- E. Sontag. Nonlinear regulation: The piecewise linear approach. *IEEE Transactions Automatic Control*, 26(2):346–358, 1981.
- T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, 15(1):116–132, 1985.
- J. Theis, P. Seiler, and H. Werner. LPV model order reduction by parameter-varying oblique projection. *IEEE Transactions on Control Systems Technology*, 26(3):773–784, 2018. URL www.scopus.com.
- J. Tomperi, E. Juuso, and K. Leiviskä. Hybrid le systems for simulation of an activated sludge process. In *Proceedings - 8th EUROSIM Congress on Modelling and Simulation, EUROSIM 2013, 10-13 September, 2013, Cardiff, UK*, pages 118–123, 2013. doi:10.1109/EUROSIM.2013.31. URL www.scopus.com.
- J. Tomperi, E. Juuso, and K. Leiviskä. Early warning of changing drinking water quality by trend analysis. *Journal of Water and Health*, 14(3):433–442, 2016. doi:10.2166/wh.2016.330.
- J. Tomperi, E. Juuso, A. Kuokkanen, and K. Leiviskä. Monitoring a municipal wastewater treatment process using a trend analysis. *Environmental Technology (United Kingdom)*, pages 1–10, 2017. doi:10.1080/09593330.2017.1375026.
- N. Vafamand, M. Mehdi Arefi, and A. Khayatian. Nonlinear system identification based on Takagi-Sugeno fuzzy modeling and unscented kalman filter. *ISA Transactions*, 74:134 – 143, 2018. ISSN 0019-0578. doi:10.1016/j.isatra.2018.02.005.
- M. Valle. *Smart Adaptive Systems on Silicon*. Springer US, 1 edition, 2004. doi:10.1007/978-1-4020-2782-6.
- Y.-W. Xiao, J. Wu, Z.-G. Lin, and X.-D. Zhao. A deep learning-based multi-model ensemble method for cancer prediction. *Computer Methods and Programs in Biomedicine*, 153:1 – 9, 2018. ISSN 0169-2607. doi:10.1016/j.cmpb.2017.09.005.
- D.-K. Xu and Y.-J. Tian. A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2(2):165–193, Jun 2015. ISSN 2198-5812. doi:10.1007/s40745-015-0040-1.
- L. A. Zadeh. Fuzzy sets. *Information and Control*, 8(June): 338–353, 1965.
- L. A. Zadeh. The Concept of a Linguistic Variable and Its Application to Approximate Reasoning-1. *Information Sciences*, 8: 199–249, 1975.

Experimental and Computational Study of Oxygen Carriers in Chemical Looping Combustion

Rajan K. Thapa D.I Erandi N. Wijeratne Britt M.E. Moldestad

Department of Process, Energy and Environmental Technology, University College of Southeast Norway

rajan.k.thapa@usn.no erandinw@yahoo.com britt.moldestad@usn.no

Abstract

In this paper, the fluidization properties of possible oxygen carriers for Chemical Looping Combustion (CLC) have been studied. Experiments were performed in a cold fluidized bed to study fluidization properties of particles having similar properties as the oxygen carriers. The particles used were Zirconium oxides (ZrO), glass beads and sand. Minimum fluidization velocity, bed expansion and bubble behaviors of the particles have been studied. The Electrical Capacitance Tomography (ECT) was used to analyze bubbles in different planes in the bed of particles. The oxidation properties of ilmenite particles have been studied using Thermal Gravimetric Analysis (TGA). Computational Fluid Dynamic (CFD) numerical method was chosen to model the particle fluidization and simulations were carried out using the commercial CFD software ANSYS Fluent. 2D models were solved for most of the particles and a 3D model was solved for white ZrO. The simulation results are compared with experimental data. The bubble behavior and bed expansion are similar in the experiments and the simulations. However, the simulated results of other fluidization properties are higher than in the experiments due to a wide size distribution in experimental particles that was not possible to consider in the simulations.

Keywords: CLC, fluidized bed, CO₂ capture, combustion, CFD, TGA, ECT,

1 Introduction

Chemical Looping combustion (CLC) is an energy effective CO₂ capturing technology. The technology is based on fuel combustion using two separate reactors: oxidation and reduction reactors. This allows separating CO₂ from other flue gases already during the combustion process. Therefore, the separation does not require additional energy input making the technology more cost effective. The principle of the CLC process is shown in Figure 1. The two reactors presented in the figure are interconnected fluidized bed reactors. One of them is an air reactor. Metal particles (oxygen carriers) are oxidized in this reactor in the presence of air. The particles are then recirculated to the fuel reactor through

fast fluidization regime. In fuel reactor, the metal oxide particles provide required oxygen for fuel combustion in the bubbling fluidization regime.

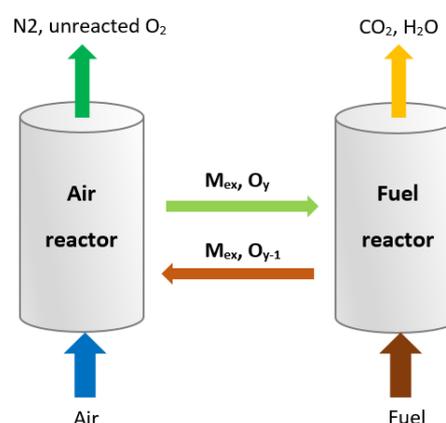


Figure 1: Principles of chemical looping combustion.

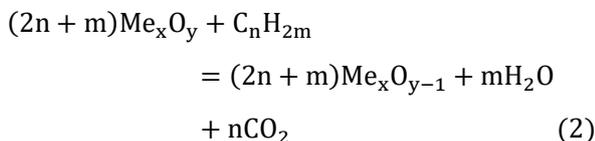
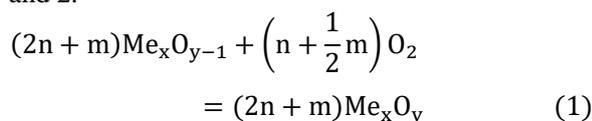
In this way, the flue gas coming out of the fuel reactor is only CO₂. The particles coming out of the bubbling fluidized bed fuel reactor are recirculated back to air reactor for further oxidation. The flue gas from both of the reactors are cooled for heat recovery, and almost pure CO₂ is captured. The two fluidized bed reactors maintain a good contact between the gas and solid phases. In addition, CLC does not provide favourable conditions for the formation of NO_x. Thermal NO_x is generated at high temperatures. The quantity is significant above 1400°C. In CLC, the air reactor has temperatures in the range of 800°C -1200°C and the temperature in the fuel reactor is in the range of 400°C - 1000°C (Lyngfelt et al, 2001).

The CLC process has been successfully demonstrated both in bench scale fixed-bed reactors and in continuously operated prototype reactors based on interconnected fluidized-bed system in the size range of 0.3–140kW(He Fang, 2009).

2 Oxygen Carriers

The oxygen carrier (metal particles) is oxidized in the air reactor and reduced back to its original state in the fuel reactor. Respective reaction equations for the

oxidation and reduction processes are given by Equation 1 and 2.



where, $\text{Me}_x\text{O}_{y-1}$, Me_xO_y and C_nH_{2m} are metal oxide in its reduced form, metal oxide in its oxidized form and hydrocarbon fuel respectively. The chemical reaction given by Equation 2 shows non-diluted CO_2 as one of the products. The net chemical reaction obtained by adding Equation 1 and 2 give the normal combustion reaction between the oxidizer and fuel.

There has been extensive research on CLC during the last decade with respect to oxygen carrier development, reaction kinetics, reactor design, system efficiencies and prototype testing (He Fang, 2009). Development of a suitable oxygen carrier is one of the challenging task, which determines the commercial success of the technology.

Many researchers have focused on metal oxides in the forms of industrial waste, synthetic materials and naturally occurring materials. A good oxygen carrier should exhibit characteristics such as high reaction rate, high conversion, sufficient oxygen capacity, resistance for attrition, fragmentation, carbon deposition and durability. It should have low cost and should be environmentally friendly. Most of the metal oxides are combined with an inert material (not taking part in the chemical reactions) which acts as a porous support providing a higher surface area for the reaction, and as a binder for increasing mechanical strength and attrition resistance (Labiano et al, 2005). Therefore, the average bulk density of the oxygen carriers are usually lower than that of metals and metal oxides. For example, some metal oxides such as NiO , CuO , Mn_3O_4 , Fe_2O_3 have been identified as promising candidates, and when they are supported by an inert binder, the carrier particles have shown increased reactivity (Fossdala, 2010; Johansson, 2007).

Al_2O_3 , TiO_2 , SiO_2 , ZrO are recognized as possible inert binders (Naqvi, 2006). Synthetic oxygen carriers are expensive. The naturally occurring oxides such as ilmenite, iron ore and manganese are more suitable because of their low cost. In this paper, we focus on the investigation of the fluidization properties of the possible oxygen carriers in order to minimize gas and solid circulation. For this purpose, we investigated different particles that resemble the approximate properties of oxygen carriers. The experiments were carried out using pressure sensors, ECT and TGA. The

simulations were run using the CFD software ANSYS Fluent. One of the promising candidates for oxygen carriers is ilmenite particles. The reactivity of the particles were investigated using TGA analysis and the fluidization properties of the particles were investigated using a validated CFD model.

3 Experimental work

In the CLC plant, air and fuel reactors are fluidized bed reactors for circulating oxygen carrier from one reactor to another. Performance of the reactors is effected by fluidization behaviors of the oxygen carrier used in the reactors. The apparent density of most of the possible oxygen carriers are in between 2000 – 9000 kg/m^3 . Therefore, particles with varying density and particle size were used in the experiments to study the fluidization behaviors of the possible oxygen carriers. The particles used in the experimental work are presented in Table 1.

Table 1: Particles used in experimental work.

Particle	Mean particle size [μm]	Density [kg/m^3]
Brown ZrO	709	5850
White ZrO	508	3800
Glass beads	153	2485
Sand	83	2500
Ilmenite	115.83	4580

3.1 Experimental setup and procedure

Experiments were performed in a cold model of a fluidized bed reactor located in the University College of Southeast Norway. The cold model consists of a plastic cylindrical column of 1400 mm height and 84mm inner diameter.

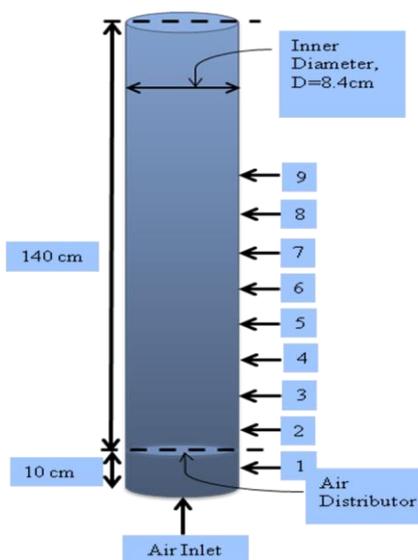


Figure 2: Location of the pressure nodes.

The cylinder has porous metal plate as air distributor at the bottom. A series of pressure nodes are mounted

along the height of the cylinder for pressure measurement as shown in Figure 2. The nodes are connected to pressure sensors as well as to a computer with LabVIEW program for pressure reading and airflow measurement. The distance between two pressure nodes is 10mm.

Thermo-Gravimetric Analyzer (TGA) was used to determine the oxidation characteristics of the ilmenite particles when exposed to air at 950°C. The experimental set up is shown in Figure 3. The set up consist of Perkin Elmer TGA7, Perkin Elmer Thermal Analysis Controller TAC7/DX, Thermal Analysis Gas Station MT-0029 and Pyris software.



Figure 3: Experimental set up of TGA.

The TGA analyzer consists of an oven that can reach a temperature of 1000°C, a thermometer, an accurate balance and a platinum pan where the sample is placed. The sample can be purged with nitrogen to prevent it from reacting with oxygen when drying the sample in the oven. When the temperature is high enough, oxygen can be switched on to oxidize the sample.

The water content, organic and inorganic materials, volatile and non-volatile matters in the particle samples can also be analyzed. Approximately 5g ilmenite particles were used for the sample.

According to several investigations, (Ana Cuadrat; Henrik Leion, 2008; Juan Adnez, 2010) ilmenite has shown good results as an oxygen carrier for solid fuels and it has become attractive due to its low cost and abundant availability compared to other oxygen carrier particles.

4 Electric Capacitance Tomography

Electrical capacitance tomography (ECT) is used to study bubble behavior of possible oxygen carrier particles. ECT is a novel technique that determines the concentration distribution of a two-phase mixture within a closed cylinder. Using this data the number of bubbles inside a plane of the cylinder can be determined. The ECT sensor consists of two planes and there are 12

electrodes per plane. The electrodes are mounted on the outside of the cylindrical rig as shown in Figure 4.

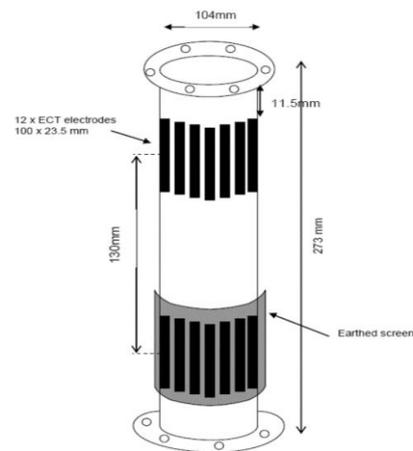


Figure 4: Rig with electrodes for ECT sensors.

5 Computational study

A Computational Fluid Dynamic (CFD) model for the experimental cylinder was prepared for 2D and 3D simulations. The grids for the 2D and 3D simulations are shown in Figure 5. The model was simulated using the CFD simulation software ANSYS Fluent. Simulations were carried out to investigate the minimum fluidization velocities, void fraction variation and bubble behavior. Simulations were run for both a single particle phase and for more than one particle phase. The particle phases were determined by different particle sizes.

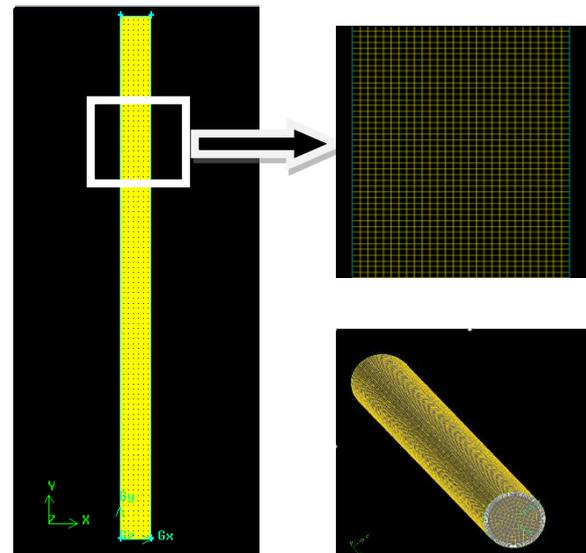


Figure 5: Mesh of experimental rig.

The model parameters used in simulations of the solid phase are shown in Table 2. The drag force between solid and gas phase was modeled using Syamlal & O'Brien's drag model. Five monitors were set in the simulations to record the pressure data. The locations of the monitors in the simulations are similar to the

location of the pressure nodes in the experimental cylinder. The simulations were performed for maximum 8 seconds.

Table 2: Models used to describe the properties of solid.

Property	Model
Granular viscosity	Syamlal-O'Brien
Granular bulk viscosity	Constant
Frictional viscosity	Schaeffer
Frictional pressure	Based-ktgf
Solid pressure	Ma-ahmadi
Radial distribution	Ma-ahmadi

Simulations were run for all the particles used in experiments (see Table 1). For glass beads, simulations with two particle phases were performed using particle size 120 μm as phase 1 and 120 μm as phase 2 with corresponding static solid volume fraction of 0.321 and 0.299. For white ZrO, 3D simulations were performed to investigate the bubble formation in different planes along the height of the bed.

6 Results and discussion

6.1 Experimental

A series of experiments were carried out to determine the minimum fluidization velocities for the particles. Figure 6 shows the average pressure drop as a function of superficial air velocity for the four types of particles under consideration. The air velocities are plotted using logarithmic scale in order to cover the wide range of air velocities for these different types of particles.

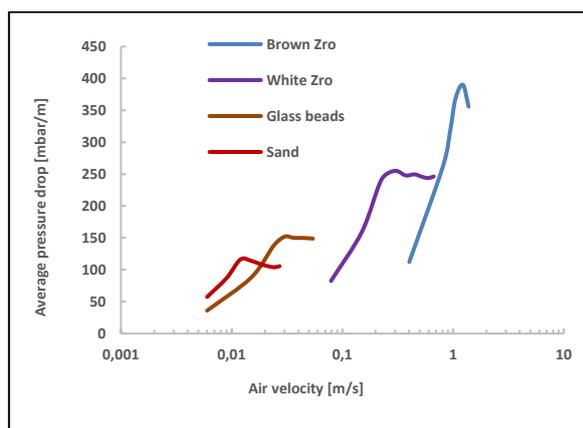


Figure 6: Minimum fluidization velocities.

The fuel reactor in CLC is operated in the bubbling fluidized bed regime while the air reactor operates in fast fluidized regime and should handle about 10 times more air on volume basis (Lyngfelt et al., 2001). Lower minimum fluidization velocity indicates low-pressure drop along the bed height and lower consumption of air

in the bed. This makes lower amount of air handling during the CLC process. Particles such as sand, glass beads having lower densities are more suitable for CLC than the particles with higher bulk densities. The experimental observation also shows that the brown ZrO particles has slugging behavior at higher air velocities which is not a desired phenomenon for gas solid mixing.

During the experiment in TGA analyzer, a particle sample of 5g ilmenite was heated in the oven. Figure 7 shows an enlarged view of the oxidation period of the sample.

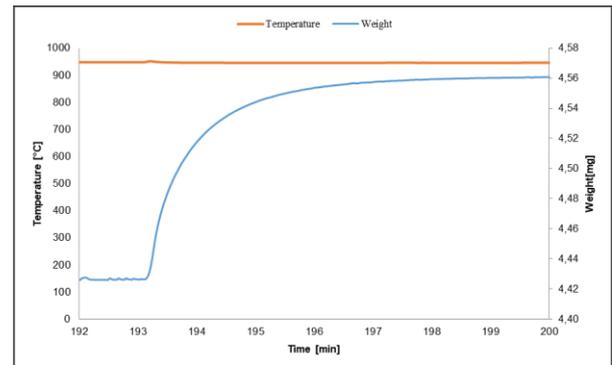


Figure 7: Temperature and weight variation with time in oxidation period for ilmenite.

The sample particles were heated for 30 minutes at the temperature of 25°C and then the temperature was increased from 25°C to 110°C. During the time interval, there was a small weight loss. This is due to volatilization of the moisture. Then the temperature was increased from 110°C to 950°C as shown in Figure 7 and the air flow was turned on. A sharp increase in the weight of ilmenite can be observed due to the oxidation of ilmenite particles in air.

The oxygen transfer capacity of an oxygen carrier determines the number of minimum circulation cycles and the mass of carrier required in the reactor.

$$R_0 = \frac{m_{ox} - m_{red}}{m_{ox}} \quad (3)$$

where R_0 is mass fraction of oxygen in the carrier, m_{red} is the mass in the reduced form and m_{ox} is the mass in the oxidized form. In the calculation, it is assumed that the weight prior the oxidation period is equal to the weight of the sample in reduced form and weight after the oxidation period is equal to the weight of the sample in oxidized form. The experimental data (see Figure 7) is used to calculate the transfer capacity.

$$R_0 = \frac{4.56 - 4.43}{4.56} = 0.029$$

The experimental results show that the ilmenite entering the fuel reactor contains 2.3% of its mass in the form of oxygen. Reduced ilmenite is in the form FeTiO_3 and the oxidized carrier is in the form $\text{Fe}_2\text{TiO}_5 + \text{TiO}_2$ (Henrik Leion, 2008).

Glass beads with size distribution from 100 to 200 μm were used in the experiment with ECT to measure bubble distribution at the two different planes of the cylindrical rig and to determine minimum fluidization velocity. The number of bubbles increases with the increasing air velocity as shown in Figure 8.

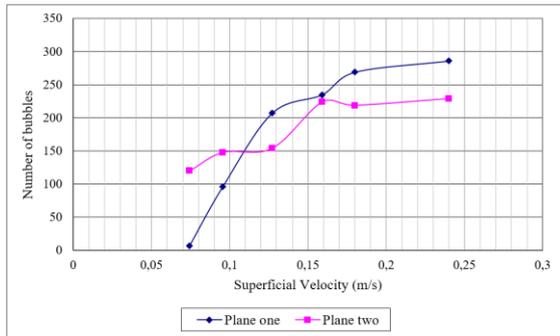


Figure 8: Number of bubbles variation with superficial air velocity.

The number of bubbles are higher in plane one than plane two. The plane one is in the lower part of the bed and the plane two is the upper part of the bed. The results show that the number of bubbles are greater in the lower part of the bed than the upper part.

6.2 Simulations

A series of simulations were carried out to compare and validate the CFD model prediction with experimental results. The simulations are used to study minimum fluidization velocities, void fractions and bubble behaviors in the bed of the particles. There was some difficulties to use ilmenite particles in the experimental fluidized bed rig due to excess moisture content in the sample. Therefore, the validated CFD model has been helpful to determine fluidization properties for ilmenite particles computationally.

The simulation results for average pressure drops as a function of air velocities are compared against experimental results for white ZrO in Figure 9. The packing limit in this case is 0.6.

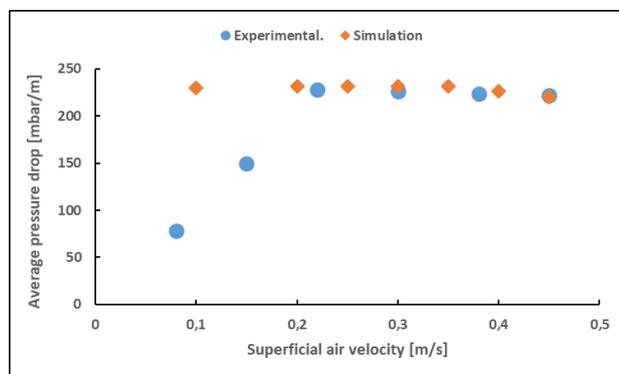


Figure 9: Pressure drop vs. air velocity for white ZrO. Static solid volume fraction 0.6.

The comparison of results show that the experimental and computational minimum fluidization velocities are near to each other, but the computational value is slightly higher than experimental. The results for the other particles show good agreement with experimental results.

3D simulations were carried out to investigate the bubble behavior of the ZrO particles further. Figure 10 shows the bubble in two different planes, located at a distance of 0.2m from each other along the height of the bed. The solid volume fraction at the upper plane (p5) shows that the bed is fluidized with formation of some bubbles at the upper part of the bed. The bubble size is increased with increasing simulation time.

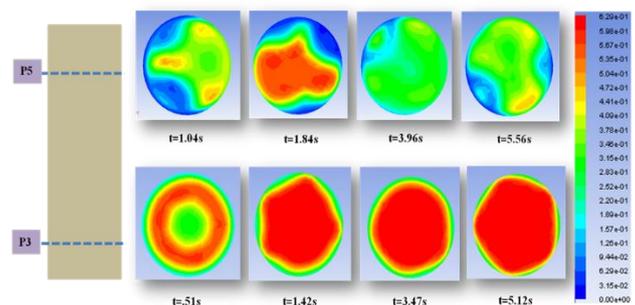


Figure 10: Bubble behavior of white ZrO in 3D simulation at air velocity of 0.4 m/s.

The similarity of minimum fluidization velocity and the bubble behavior between the experimental and the simulation results allows to move further only with simulations to study the fluidization behavior of some oxygen carriers, which could be the good candidates.

The CFD model was used to simulate the fluidization behavior of the ilmenite particles. The computational results of the bubble behavior of the ilmenite particles are shown in Figure 11.

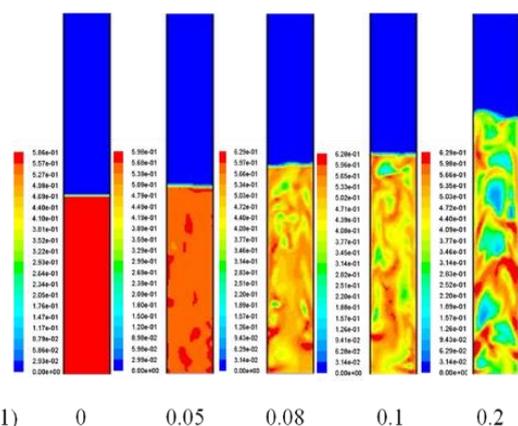


Figure 11: Contours of solid volume fraction for Ilmenite. The minimum fluidization velocity for the particles is less than 0.05 m/s. Bubble formation and increase in bubble size occurs at the air velocity between 0.08 m/s to 0.2 m/s with significant bed expansion. The bubble formation at the velocity of 0.2 m/s shows uniform

bubble distribution along the height of the bed. This indicates that the ilmenite particles can have good gas-solid mixing properties in fluidized bed reactors. This makes ilmenite an even more suitable candidate as oxygen carrier for the chemical looping combustion process.

7 Conclusion

The fluidization properties of four types of particles have been studied using an experimental cold fluidized bed cylinder with pressure sensors. The density and the size of the particles resemble possible oxygen carriers for CLC. The particles studied in fluidized bed cylinder were sand, glass beads, white ZrO and brown ZrO. The minimum fluidization velocity is lowest for sand and highest for brown ZrO.

Tests using Electrical Capacitance Tomography have been performed for glass beads to study the bubble behavior. The number of bubbles in the lower part of the bed is higher than in the upper part of the bed.

Thermo Gravimetric Analysis was performed to study oxidation characteristics of ilmenite. Ilmenite particles could not be used in the fluidized bed study because of the high moisture content in sample. The oxygen carrying capacity of the particles have been calculated using experimental results.

Computational study (2D simulations) were run using the simulation software ANSYS Fluent. 3D simulations were carried out for white ZrO. Simulations with two particle phases were carried out for glass beads.

Minimum fluidization velocities are determined by using both experimental and computational results. In the experiments, the calculation is based on the pressure drop measurements. The computational results for the minimum fluidization velocities are higher than the experimental results due to neglecting the effects of particle size distribution and particle shape in the simulations. Simulations with two particle phases give results closer to the experimental results, but it is more time consuming. The 3D simulations resulted in nearly the same minimum fluidization velocity as the 2D simulations for white ZrO. The reason may be that: although the 3D simulations are more accurate, these simulations do not account for the particle size distribution. The bed expansion are observed both in experiment and simulation and they show closer results according to the Geldart classification for particles.

References

J. Adnez, A. Cuadrat., A. Abad, P. Gayn. Ilmenite activation during Consecutive Redox Cycles in Chemical-Looping Combustion. *Energy & Fuels*.24(2):1402-1413,2010. DOI: 10.1021/ef900856d A. Cuadrat, A. Abad, J. Adánez, L.F. de Diego, F. García-Labiano, P. Gayán. Behaviour of Ilmenite as Oxygen Carrier in Chemical-Looping

Combustion. *Fuel Processing Technology*. 94(1): 101-112, 2012. <https://doi.org/10.1016/j.fuproc.2011.10.020>

- H. Fang, L. Haibin, and Z. Zengli. Advancements in Development of Chemical-Looping Combustion: A Review. *International Journal of Chemical Engineering*, 2009. <http://dx.doi.org/10.1155/2009/710515>
- A. Fossdal, E. Bakken, B.A.Øye, C. Schøninga, I. Kaus, T.Mokkelbost, Y. Larringb. Study of inexpensive oxygen carriers for chemical looping combustion. *International Journal of Greenhouse Gas Control*.5(3) 2011. <https://doi.org/10.1016/j.ijggc.2010.08.001>
- M. Johansson. Screening of oxygen-carrier particles based on iron-, manganese-,copper- and nickel oxides for use in chemical-looping technologies. (Phd thesis), Chalmers University of Technology, 2007.
- F. G. Labiano, L. F. de Diego, J. Adánez, A. Abad, and P.Gayán. Temperature variations in the oxygen carrier particles during their reduction and oxidation in a chemical-looping combustion system. *Chemical Engineering Science*, 60(3):851-862,2005. <https://doi.org/10.1016/j.ces.2004.09.049>
- H. Leion, A. Lyngfelt, M. Johansson, E. Jerndal, T. Mattisson. The use of ilmenite as an oxygen carrier in chemical-looping combustion. *Chemical Engineering Research and Design*.86(9):1017–1026.2008. <https://doi.org/10.1016/j.cherd.2008.03.019>
- A. Lyngfelt, B.Leckner, and T. Mattisson. A fluidized-bed combustion process with inherent CO₂ separation; application of chemical-looping combustion. *Chemical Engineering Science*, 56(10): 3101-3113,2001 doi:[https://doi.org/10.1016/S0009-2509\(01\)00007-0](https://doi.org/10.1016/S0009-2509(01)00007-0)
- R. Naqvi. Overview of CLC Activities Outside ENCAP. Paper presented at the ENCAP Seminar on CLC. 2006.

Optimization of Bioreactor Volumes in Steady State - A Simulation Study

Hanna Molin¹ Jesús Zambrano^{2,*} Bengt Carlsson¹

¹Department of Information Technology, Uppsala University, Uppsala, Sweden.

²School of Business, Society and Engineering, Mälardalen University, Västerås, Sweden.

*Corresponding author e-mail: jesus.zambrano@mdh.se

Abstract

This work deals with the analysis and design of bioreactors formed by a number of completely stirred tank reactors (CSTRs) in series. The model includes three main components: the concentration of one dominant particulate biomass, one soluble substrate component and one particulate inert matter. The model is analyzed in steady-state conditions. Monod kinetics is used for describing the specific growth rate, and the decay rate of the biomass is included. Two main optimization problems were studied: (i) minimize the effluent substrate concentration for a given total volume, and (ii) minimize the total volume for a given effluent substrate concentration. An alternative to optimize a large number of CSTRs is to consider the asymptotic case of one CSTR followed by a plug-flow reactor (PFR). Numerical results show that there is an optimal volume distribution for the two configurations in each optimization problem.

Keywords: Bioreactor optimization, completely stirred tank, effluent minimization, plug-flow reactor.

1 Introduction

Modeling and analysis of bioreactors in wastewater treatment processes have been active research areas in the last 50 years, see Grady Jr. et al. (1999) and the references therein. Special attention has been placed in the steady-state analysis because it can provide guidelines for achieving a proper process operation and optimization. During those decades, the optimization of bioreactors has also been extensively analyzed. Every new contribution aims to design methods for analyzing and optimizing the volume distribution of completely stirred tank reactors (CSTRs) in series.

In general, the aim is to design the process in such a way that the substrate concentration of a certain incoming flow can be minimized or reduced, even by using one or more bioreactors in series (de Gooijer et al., 1996). Hence, the optimization of CSTRs usually involves finding the optimal distribution of volumes for a given effluent substrate concentration, see Abu-Reesh (1996).

Alternatives to solve the optimization of CSTRs includes graphical solutions referred as *nomograms*, which give an approximate graphical computation of mathemat-

ical functions (Braha and Hafner, 1985).

Most of the analysis and optimization of CSTRs assumes up to two main components in the model: dissolved substrate and particulate biomass. Some examples are mentioned as follows. Asymptotic solutions for the effluent concentration leaving a system of up to $N = 4$ CSTRs in series were presented by Nelson and Holder (2009). The analysis assumes equally-sized CSTRs and Contois kinetics (Contois, 1959) for the biomass growth rate. Two scenarios were studied, when a total residence time (τ_r) is given, and when this residence time is slightly larger than the wash-out condition. In the first scenario, results show that the effluent decreases with the form $1/\tau_r^N$. In the second scenario, results show that a large decrease in the effluent concentration can be obtained by a small increase in τ_r .

The optimal design of CSTRs in series has often been compared to the design of a plug-flow reactor (PFR). Another aspect is that, in practice, the behavior of a biological reactor is between a CSTR and a PFR (Diehl et al., 2016, 2017). The work by Luyben and Tramper (1982) presents an analytical expression for the minimum volume to achieve a certain effluent substrate concentration in a process with Monod kinetics. The work shows some numerical results for up to $N = 10$ CSTRs in series considering two main scenarios: optimal volume distribution and equally sized volume. The case of a PFR was used for comparison. Results show that the difference between the first two volumes is larger for low N and decreases as N increases. Results also show that the case of N CSTRs approaches to the case of a PFR as N increases. Hill and Robinson (1989) derived an expression for different growth kinetics that could be used to find the minimum possible total residence time to obtain a certain substrate conversion for N CSTRs. An illustration for Monod kinetics was included. The results were compared with a PFR and showed that for most cases, three optimally designed CSTRs in series give the same total mean residence time as a PFR. Abu-Reesh (1996) calculates the optimal design of N CSTRs to get the minimum overall volume needed for a certain substrate conversion, and compares the results with the behavior of a PFR. The study used a Michaelis-Menten kinetics.

Zambrano et al. (2015) shows an approach for the opti-

mal design of CSTRs in series when the number of CSTRs is large. The process assumes Monod kinetics for the biomass growth, and the model includes two main components: one particulate biomass and one soluble substrate. The biomass decay is negligible. Analytical expressions were derived for solving the following two scenarios: to find the optimal volume distribution for a given total volume so that the effluent substrate concentration can be minimized, and to find the minimum total volume that gives a certain effluent substrate concentration.

Recently, Gómez-Pérez and Espinosa (2017) analyzed the design of continuous bioreactors in series with recirculation. The bioreactors were represented by a system of linear equations. The growth rate was assumed to be constant and the decay of biomass was neglected. Non-trivial solutions to the equation system was found by using Singular Value Decomposition (SVD) as an analysis tool. The SVD analysis makes it possible to characterize the solutions to the system of equations, and thereby improve the design of bioreactors in series.

In the present work, we deal with a simulation analysis of the process presented in Zambrano et al. (2015). In this case, the inert particulate matter is included in the bioreactor model, as well as the decay rate of the biomass. Two optimization problems were investigated: (i) the optimal volume distribution of N CSTRs in series in a given total volume in order to minimize a given effluent substrate concentration, and (ii) the minimum total volume needed for a given effluent substrate concentration. The optimization results were compared with the case of a process formed by a CSTR followed by a PFR.

2 Materials and Methods

2.1 The bioreactor model

Consider a bioreactor of volume V_{tot} formed by N CSTRs in series as shown in Figure 1, where each CSTR is modeled assuming one limiting substrate, one biomass and one inert particulate matter. The time-dependent ordinary

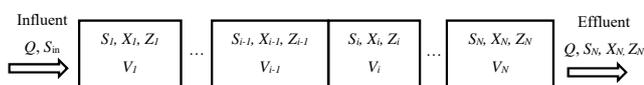


Figure 1. N CSTRs in series.

differential equations that describe the dynamics of these three components in the i -th ($i = 1, 2, \dots, N$) CSTR are

$$\frac{dS_i}{dt} = - \left[\frac{\mu(S_i)}{Y} - (1 - f_p)b \right] X_i + \frac{Q}{V_i} (S_{i-1} - S_i), \quad (1)$$

$$\frac{dX_i}{dt} = [\mu(S_i) - b] X_i + \frac{Q}{V_i} (X_{i-1} - X_i), \quad (2)$$

$$\frac{dZ_i}{dt} = f_p b X_i + \frac{Q}{V_i} (Z_{i-1} - Z_i), \quad (3)$$

where the i -th CSTR has volume V_i , substrate concentration S_i , biomass concentration X_i and inert concentration

Nomenclature

A	area [m^2]
b	biomass decay rate [$1/\text{d}$]
f_p	fraction of decay of biomass $[-]$
h	position in plug-flow reactor [m]
K_S	half-saturation constant for S [kg/m^3]
N	number of bioreactors in series $[-]$
Q	influent volumetric flow rate [m^3/d]
S	dissolved substrate concentration [kg/m^3]
t	time [d]
V	volume [m^3]
X	particulate biomass concentration [kg/m^3]
Y	yield constant $[-]$
Z	particulate inert concentration [kg/m^3]

Greek letters

μ	Monod function [$1/\text{d}$]
μ_{max}	maximum specific growth rate [$1/\text{d}$]

Subscripts

\square_e	effluent
\square_i	component in i^{th} bioreactor
\square_{in}	influent
\square_{tot}	total

Superscripts

\square^*	optimal
\square^{min}	minimum

Z_i . Y is the yield factor, b is the decay rate of the biomass and f_p is the fraction of decay of biomass ($0 \leq f_p \leq 1$). The growth of the biomass is assumed to follow a single growth rate equation described by the Monod kinetics (Monod, 1949) given by

$$\mu(S_i) = \mu_{\text{max}} \frac{S_i}{K_S + S_i}, \quad (4)$$

where μ_{max} is the maximum specific growth rate, and K_S is a half-saturation constant. The influent and effluent volumetric flow rate are equal to Q .

The influent to the process is assumed to be formed only by a substrate concentration S_{in} , no biomass and inert concentration are present. Hence, from (1)–(3) and Figure 1 we have $S_0 = S_{\text{in}}$, $X_0 = X_{\text{in}} = 0$ and $Z_0 = Z_{\text{in}} = 0$.

See in (1)–(3) that a fraction f_p of the decay of biomass concentration is assumed to be inert concentration, whereas the remaining fraction $(1 - f_p)$ is used as substrate.

2.2 The steady-state expressions for CSTRs

By assuming steady-state conditions in (1)–(3), an expression for S_i can be obtained

$$S_i = S_{i-1} - \frac{1}{Y} (X_i - X_{i-1}) - \frac{b}{QY} \left(1 - (1 - f_p)Y \right) V_i X_i. \quad (5)$$

For N CSTRs in series, the recursive expression (5) gives

$$S_N = S_{\text{in}} - \frac{1}{Y} X_N - \frac{b}{QY} \left(1 - (1 - f_p)Y \right) \sum_{n=1}^N V_n X_n. \quad (6)$$

For a single CSTR, the solution for the substrate, biomass and inert concentration, and the minimum volume (V_1^{\min}) to avoid wash-out are given by the following expressions

$$\mu(S_1) = \frac{Q}{V_1} + b, \quad (7)$$

$$X_1 = \frac{YQ(S_{in} - S_1)}{Q + V_1b(1 - Y(1 - f_p))}, \quad (8)$$

$$Z_1 = \frac{f_p b V_1 X_1}{Q}, \quad (9)$$

$$V_1 > V_1^{\min} = \frac{Q}{\mu(S_{in}) - b}. \quad (10)$$

2.3 The steady-state expression for a PFR

When the number of N CSTRs in series becomes large, the resulting process can be seen as a CSTR followed by a PFR. The CSTR is assumed to be large enough to avoid wash-out condition (i.e., $V_1 > V_1^{\min}$). The PFR is formed by dividing the remaining volume ($V_{tot} - V_1$) into a large number of volumes ΔV , see Figure 2.

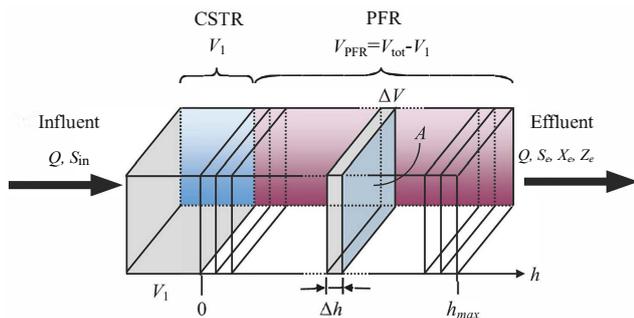


Figure 2. Dividing the remaining volume ($V_{tot} - V_1$) into a large number of volumes ΔV .

It is also assumed that each sliced bioreactor has the same area A , giving $\Delta V = A\Delta h$. For a small interval ($h, h + \Delta h$) the conservation of mass for the dissolved substrate gives

$$\underbrace{\frac{d}{dt} \int_h^{h+\Delta h} AS(x,t)dx}_{\text{mass increase per time unit}} = \underbrace{QS(h,t)}_{\text{flux in}} - \underbrace{QS(h+\Delta h,t)}_{\text{flux out}} - \underbrace{\int_h^{h+\Delta h} A \left[\frac{\mu(S)}{Y} - (1 - f_p)b \right] X dx}_{\text{consumption per time unit}}. \quad (11)$$

Dividing the previous expression by $A\Delta h$ and taking the limit $\Delta h \rightarrow 0$ we get

$$\frac{\partial S}{\partial t} + \frac{Q}{A} \frac{\partial S}{\partial h} = - \left[\frac{\mu(S)}{Y} - (1 - f_p)b \right] X. \quad (12)$$

In the same way, for the biomass and the inert matter

we get

$$\frac{\partial X}{\partial t} + \frac{Q}{A} \frac{\partial X}{\partial h} = [\mu(S) - b]X, \quad (13)$$

$$\frac{\partial Z}{\partial t} + \frac{Q}{A} \frac{\partial Z}{\partial h} = f_p b X. \quad (14)$$

In steady-state conditions, (12)–(14) become

$$\frac{Q}{A} \frac{\partial S}{\partial h} = - \left[\frac{\mu(S)}{Y} - (1 - f_p)b \right] X, \quad (15)$$

$$\frac{Q}{A} \frac{\partial X}{\partial h} = [\mu(S) - b]X, \quad (16)$$

$$\frac{Q}{A} \frac{\partial Z}{\partial h} = f_p b X, \quad (17)$$

where $S = S(h)$, $X = X(h)$ and $Z = Z(h)$ are the steady-state concentrations along the PFR that should satisfy the boundary conditions: $S(0) = S_1$, $X(0) = X_1$ and $Z(0) = Z_1$. The effluent concentrations are $S_e = S(h_{max})$, $X_e = X(h_{max})$ and $Z_e = Z(h_{max})$, where h_{max} refers to the maximum length of the PFR.

See in (12) that when $b = 0$, the partial derivative for S is negative, i.e. the substrate decreases as long as the PFR length increases. When $b > 0$, there is a condition for (12) to be positive, which means that there is a minimum effluent substrate concentration S_e^{\min} achievable in the effluent of the PFR, and is given by

$$S_e^{\min} = \frac{K_S Y b (1 - f_p)}{\mu_{max} - Y b (1 - f_p)}. \quad (18)$$

2.4 Optimization problems

Given N CSTRs in series as shown in Figure 1, assume that the process is in steady-state condition. It is of interest to get the solution of the following optimization problems:

Problem 1N. Given a total volume V_{tot} of N CSTRs in series, find the optimal volume distribution that minimize the effluent substrate concentration S_N expressed in (6), i.e.,

$$\underset{(V_1, \dots, V_N)}{\text{minimize}} \{S_N(V_1, \dots, V_N)\}, \quad (19)$$

subject to

$$V_1 > V_1^{\min}, \{V_i\}_{i=2}^N > 0, \sum_{i=1}^N V_i = V_{tot}. \quad (20)$$

Problem 2N. Given an effluent substrate concentration $S_e < S_{in}$ of N CSTRs in series, find the optimal volume distribution that minimize the total volume V_{tot} , i.e.,

$$\underset{(V_1, \dots, V_N)}{\text{minimize}} \left\{ V_{tot} = \sum_{i=1}^N V_i \right\}, \quad (21)$$

subject to

$$V_1 > V_1^{\min}, \{V_i\}_{i=2}^N > 0, S_N(V_1, \dots, V_N) = S_e. \quad (22)$$

For a large number of N CSTRs in series, an alternative to solve Problem 1N and Problem 2N is by considering the volumes V_2, \dots, V_N as a PFR. This gives two optimization problems to solve:

Problem 1PFR. Given a total volume $V_{\text{tot}} > V_1^{\min}$ of a process formed by a CSTR followed by a PFR, find the optimal volume V_1 of the CSTR that minimizes the effluent substrate concentration S_e of the PFR, i.e.,

$$\underset{(V_1)}{\text{minimize}} \{S_e(V_1)\}, \quad (23)$$

subject to

$$V_1^{\min} < V_1 \leq V_{\text{tot}}. \quad (24)$$

Problem 2PFR. Given an effluent substrate concentration $S_e < S_{\text{in}}$ of a CSTR followed by a PFR, find the optimal volumes V_1 of the CSTR and V_{PFR} of the PFR that minimize the total volume $V_{\text{tot}} = V_1 + V_{\text{PFR}}$, i.e.,

$$\underset{(V_1, V_{\text{PFR}})}{\text{minimize}} \{V_{\text{tot}} = V_1 + V_{\text{PFR}}\}, \quad (25)$$

subject to

$$S_{\text{PFR}} = S_e, \quad (26)$$

where S_{PFR} is the effluent substrate concentration of the PFR.

3 Numerical illustration

This section shows a numerical example of the process described in the previous section. Let assume the following parameter values: $V_{\text{tot}} = 1.1 \text{ m}^3$, $Q = 1 \text{ m}^3/\text{d}$, $A = 0.428 \text{ m}^2$, $\mu_{\text{max}} = 2 \text{ d}^{-1}$, $b = [0, 0.8] \text{ d}^{-1}$, $f_p = [0, 1]$, $Y = 0.8$, $K_S = 1.2 \text{ kg/m}^3$, $S_{\text{in}} = 10 \text{ kg/m}^3$. The simulations were performed using the software Matlab.

3.1 Response of the system for a given V_1

Given a total volume V_{tot} and the volume of the first CSTR, we show the value of the substrate, biomass and inert concentrations for the remaining volume ($V_{\text{tot}} - V_1$) for some values of N CSTRs in series and for the configuration CSTR+PFR. In this case, V_1 should be larger than V_1^{\min} (cf. (10)). See that V_1^{\min} depends on b . As illustration, this example shows results for $b = 0$ and $b = 0.1$. Hence, to use the same V_1 in all the evaluations, we take the most restrictive case, i.e. $b = 0.1$, giving $V_1^{\min} = 0.593$. We select $V_1 = 1.2V_1^{\min} = 0.712$. Results are shown in Figure 3 for three different combinations of b and f_p .

See that, as expected, the biomass concentration decreases and the substrate concentration increases when b increases. Also see that the inert concentration increases as f_p increases.

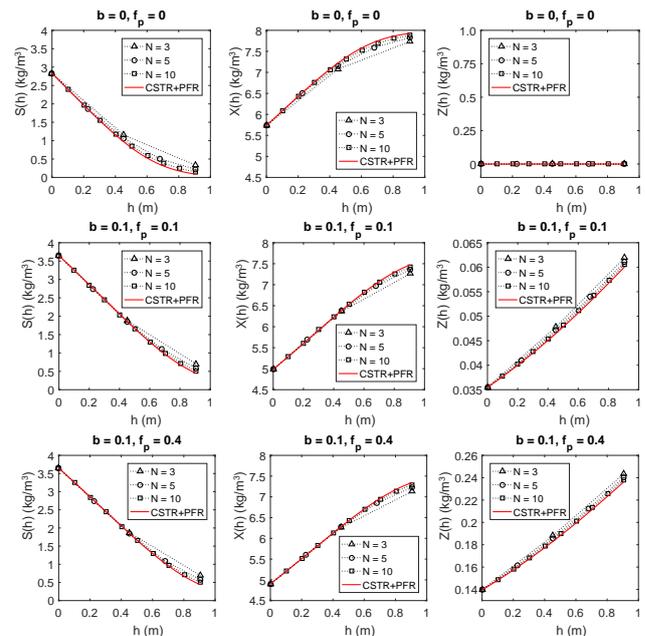


Figure 3. Response of S, X and Z for a given V_1 . Results for N CSTRs and CSTR+PFR. Top: $b = f_p = 0$. Middle: $b = 0.1, f_p = 0.1$. Bottom: $b = 0.1, f_p = 0.4$.

3.2 Optimal design for a given total volume

In this example, the aim is to find the solution of Problem 1N and Problem 1PFR. In order to show the optimal design in an illustrative way, we proceed to get the effluent substrate concentration S_e for different values of V_1 .

For the case of Problem 1N, the volumes V_2 to V_N are optimized using the Matlab function `fmincon` since it can find the minimum value of a multivariate function with linear and nonlinear constraints. See that Problem 1N only has linear constraints.

For the case of Problem 1PFR, S_e is obtained by solving (15)–(17). After the evaluation with the full possible range of V_1 , the optimum V_1 that gives the minimum effluent concentration is obtained. The results are shown in Figure 4 for the case of $N = 2, 3, 5, 10$ and for the case of CSTR+PFR, for some values of b and f_p .

See that the solution given by the N CSTRs converges to the solution of CSTR+PFR as N increases. Note that the steady-state solutions converge to the value $S_e = S_{\text{in}} = 10$ which corresponds to $V_1 = V_1^{\min}$, and to the value $S_e = S_1$ when $V_1 = V_{\text{tot}}$, which is the case of a single CSTR.

3.3 Optimal vs. sub-optimal design for a given total volume

Given a total volume V_{tot} , two volume designs of N CSTRs are compared: (a) suboptimal design, where volume V_1 is taken from the optimal design of CSTR+PFR and the rest of the volumes are equally sized; and (b) optimal design, where all the volumes are optimized. The results are shown in Figure 5 for different values in b and f_p .

As expected, the effluent substrate concentration increases as b increases. Note also that, when b increases,

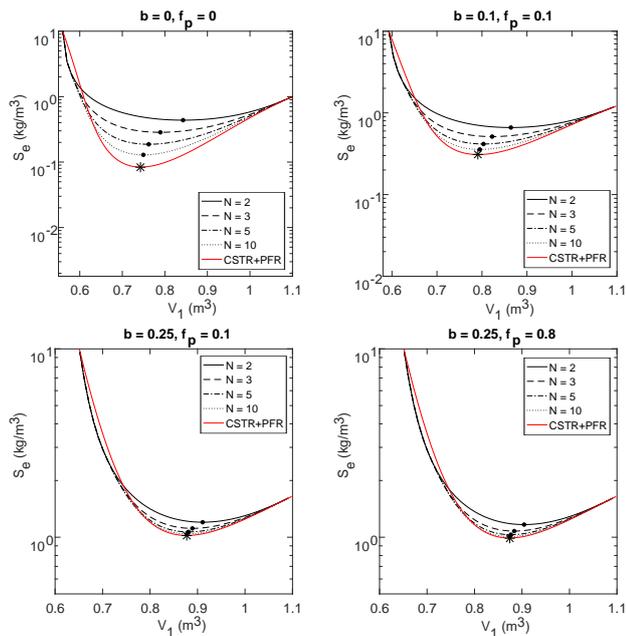


Figure 4. Steady-state solution of the effluent substrate concentration as a function of V_1 . Results shown for N CSTRs and CSTR+PFR, and for different values in b and f_p . The volume V_1 that gives the minimum effluent is marked with black dot (for N CSTRs) and with asterisk (for CSTR+PFR).

not much S_e reduction is obtained when N increases. This is because the minimum volume to avoid wash-out increases as b increases (see (10)), which makes V_1 to be closer to V_{tot} .

3.4 Optimal design for a given S_e

In this example, the aim was to compare the solution of Problem 2N and Problem 2PFR. This is, we compare the minimum total volume required in N CSTRs to achieve a given effluent substrate concentration S_e , with the minimum total volume required in the case of CSTR+PFR. Note that there is no bounds imposed for the total volume required in both problems. Both problems were solved with the Matlab function `fmincon`. Note that these problems involve linear constraints (regarding the volume distribution) and nonlinear constraints (regarding the effluent substrate concentration).

To facilitate the comparison, the ratio $V_{\text{opt}}/V(N)$ with respect to the ratio S_e/S_{in} is plotted, where $V_{\text{opt}} = V_1^* + Ah^*$, and $V(N)$ refers to the minimum total volume required for N CSTRs. We evaluate this example for up to $N = 5$ and for S_e/S_{in} between 10^{-2} and 1. Results are shown in Figure 6 for some values in b and f_p , the value $S_e^{\text{min}}/S_{\text{in}}$ (cf. (18)) is also shown.

See that, for a given effluent requirement, the ratio between V_{opt} and $V(N)$ approaches to one as N increases, i.e. the solution given by Problem 2N converges to the solution given by Problem 2PFR when more CSTRs are involved in the process. See also that when the effluent requirement is less exigent (S_e/S_{in} close to 10^{-1}), the total volume of both problems is very similar (i.e. ratio very

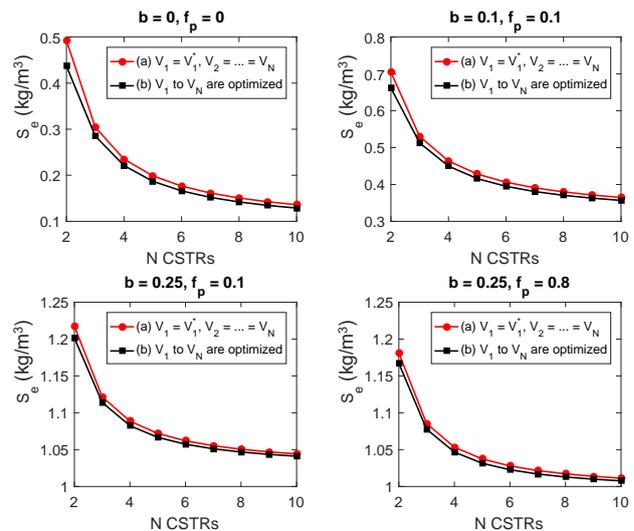


Figure 5. Effluent substrate concentration as a function of N CSTRs for two configurations. V_1^* refers to the optimal volume of the case CSTR+PFR. Results shown for different values in b and f_p .

close to one).

Note that the ratio $V_{\text{opt}}/V(N)$ decreases as b increases, and that this decrease is more evident for low N . Since increasing b promotes that more biomass die (which later will become part as substrate and the remaining as inert), then a larger total volume is needed to reduce the substrate concentration to the required level. Hence, compared to the case CSTR+PFR, a larger total volume is needed for low N CSTRs.

On the other hand, the ratio $V_{\text{opt}}/V(N)$ increases as f_p increases. Since increasing f_p promotes more inert and less substrate concentration, then it has the opposite effect in the ratio of volumes compared to increasing b .

4 Conclusions

A process consisting of N CSTRs in series has been studied in steady-state conditions. A simple model for the biological reactions based on ordinary differential equations has been used, which describes the growth of biomass via consuming dissolved substrate and producing new dissolved substrate and inert particulate matter.

Numerical results suggest that it is possible to find an optimal distribution of CSTR volumes in order to either: (i) minimize the effluent substrate concentration for a given volume, or (ii) minimize the total volume to fulfill an imposed effluent substrate concentration. The configuration of CSTRs in series was compared to a configuration formed by one CSTR followed by a PFR, i.e. CSTR+PFR. In the configuration CSTR+PFR, an optimal distribution of volume was also found. This configuration can be used as an approximation to the optimal design, by finding the optimal design of the first CSTR and then dividing the remaining volume in equally distributed CSTRs.

The decay rate b has a strong influence in the optimiza-

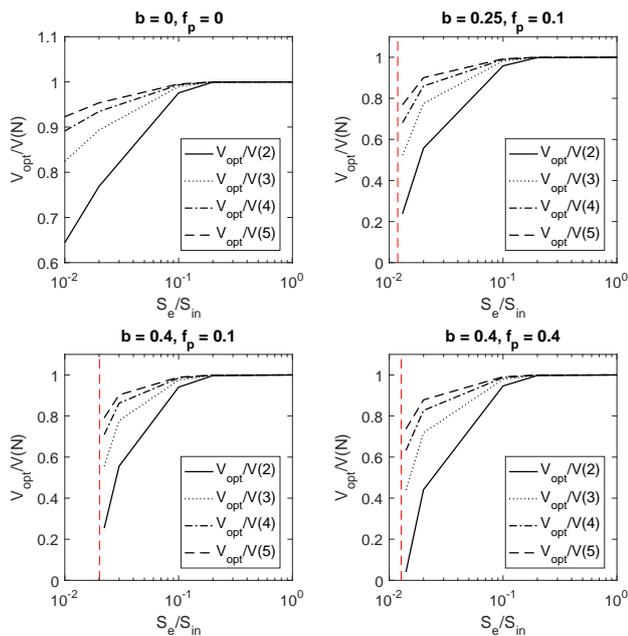


Figure 6. Minimum total volume needed for a given effluent substrate concentration S_e . V_{opt} refers to the solution of Problem 2PFR, and $V(N)$ refers to the solution of Problem 2N. Results shown for different values in b and f_p . $S_e^{\text{min}}/S_{\text{in}}$ is indicated with a vertical red dashed line.

tion results of both the N CSTRs and the CSTR+PFR. The amount of substrate generated by increasing b makes it less determinant to include a PFR in the process.

This study assumed the hydrolysis as an instantaneous step, since part of the product generated by the biomass is considered already dissolved. A possible extension of this work is to include a hydrolysis step, which involves adding a particulate substrate concentration in the process modelling.

References

- I. M. Abu-Reesh. Optimal design for CSTR's in series using reversible Michaelis-Menten reactions. *Bioprocess Engineering*, 15(5):257–264, 1996. doi:10.1007/bf02391587.
- A. Braha and F. Hafner. Use of monod kinetics on multi-stage bioreactors. *Water Research*, 19(10):1217–1227, 1985. doi:10.1016/0043-1354(85)90174-5.
- D. E. Contois. Kinetics of bacterial growth: Relationship between population density and specific growth rate of continuous cultures. *Journal of General Microbiology*, 21(1–3):40–50, 1959.
- C. D. de Gooijer, W. A.M. Bakker, H. H. Beftink, and J. Tramper. Bioreactors in series: An overview of design procedures and practical applications. *Enzyme and Microbial Technology*, 18(3):202–219, 1996. doi:10.1016/0141-0229(95)00090-9.
- S. Diehl, J. Zambrano, and B. Carlsson. Steady-state analysis of activated sludge processes with a settler model including sludge compression. *Water Research*, 88:104–116, 2016. doi:10.1016/j.watres.2015.09.052.
- S. Diehl, J. Zambrano, and B. Carlsson. Steady-state analyses of activated sludge processes with plug-flow reactor. *Journal of Environmental Chemical Engineering*, 5(1):795–809, 2017. doi:10.1016/j.jece.2016.06.038.
- C. A. Gómez-Pérez and J. Espinosa. The design analysis of continuous bioreactors in series with recirculation using Singular Value Decomposition. *Chemical Engineering Research and Design*, 125:108–118, 2017. doi:10.1016/j.cherd.2017.06.030.
- C. P. L. Grady Jr., G. T. Daigger, and H. C. Lim. *Biological Wastewater Treatment*. Marcel Dekker, Inc., New York and Basel, 1999. ISBN 0-8247-8919-9.
- G. A. Hill and C. W. Robinson. Minimum tank volumes for CFST bioreactors in series. *The Canadian Journal of Chemical Engineering*, 67(5):818–824, 1989. doi:10.1002/cjce.5450670513.
- K. Ch. A. M. Luyben and J. Tramper. Optimal design for continuous stirred tank reactors in series using Michaelis-Menten kinetics. *Biotechnology and Bioengineering*, 24(5):1217–1220, 1982. doi:10.1002/bit.260240518.
- J. Monod. The growth of bacterial cultures. *Annual Review of Microbiology*, 3(1):371–394, 1949.
- M. I. Nelson and A. Holder. A fundamental analysis of continuous flow bioreactor models governed by Contois kinetics. II. Reactor cascades. *Chemical Engineering Journal*, 149(1–3):406–416, 2009. doi:10.1016/j.cej.2009.01.028.
- J. Zambrano, B. Carlsson, and S. Diehl. Optimal steady-state design of zone volumes of bioreactors with Monod growth kinetics. *Biochemical Engineering Journal*, 100:59–66, 2015. doi:10.1016/j.bej.2015.04.002.

System Development for On-line Monitoring using Raman Spectroscopy for CO₂ Absorption by MEA

M. H. Wathsala N. Jinadasa, K. Amila Chandra, Maths Halstensen

Faculty of Technology, Natural Sciences and Maritime Sciences – University of South-eastern Norway, Post box 235, N-3603 Kongsberg, Norway, (Eigenvector Research)@usn.no

Abstract

Among various kinds of technologies available, carbon dioxide (CO₂) capture by monoethanolamine (MEA) is considered to be the most technically and scientifically matured technology which can be tested in industrial scale. When CO₂ is absorbed by an MEA, a chemical reaction takes place which results to form different carbon and amine species in the system. In this work, Raman spectroscopy has been used to measure those concentrations in-situ. Since the instrument does not provide direct measurements, multivariate analysis has been used to develop models and predictions are made using these models for future measurements. This study presents the methodology of acquiring measurements by the Raman spectroscopy for MEA-CO₂-H₂O system, transferring the measurement data into Matlab/Labview, converting data into concentration values and presenting the results in a graphical user interface. This software based platform makes the Raman spectroscopy to be accessed as a real-time instrument in CO₂ capture plants.

Keywords: CO₂ absorption by MEA, real-time monitoring, Raman spectroscopy, chemometrics, Labview

1 Introduction

68% of global anthropogenic greenhouse gas emissions comes from energy production and 90% of these greenhouse gases are carbon dioxide (IEA, 2017). Therefore capturing CO₂ emissions from energy production is a mandatory task under the climate change mitigation actions. As stated in (IEA, 2017) from 1870 to 2015, CO₂ emission by fossil fuels dramatically increased from near zero to over 33 Gt CO₂ which resulted fossil fuel power sector to be the largest source of greenhouse gas emissions. Greenhouse gases are responsible for increasing the world atmospheric temperature causing threatening impacts of climate change. While switching towards renewable and non-fossil fuel sources, the immediate action to reduce further CO₂ emission to the atmosphere is CO₂ capturing and storage from current emission sources.

After several studies on CO₂ capturing, the so-called ‘monoethanolamine technology’ or in other words, absorbing CO₂ chemically to an MEA solution is considered to be the most promising technology which

can be tested in industrial level. Some examples for MEA based commercial plants are Boundary Dam CO₂ capture plant and Cansolv CO₂ capture system in Canada. When a CO₂ molecule bounds to an MEA molecule, they are converted chemically into different reaction intermediates. Among them carbonate (CO₃²⁻), bicarbonate (HCO₃⁻), carbamate (MEACOO⁻) and protonated MEA (MEA^{H+}) are prime important. There can also be free MEA and dissolved CO₂ (unreacted) in the system. In an overall perspective, if one wants to get an overview about the chemical concentration of an MEA-CO₂-H₂O system, concentrations of above-mentioned chemical intermediates should be presented. Knowing the concentrations of intermediates gives the benefit to understand reaction path, impact of process parameters to the CO₂ absorption, inputs to thermodynamic modelling and thus help to design and optimize the process.

Raman spectroscopy is a process analytical instrument. Fundamentally, it gives information on molecular vibrations and crystal structures in a chemical system. This information can be mapped with other properties of the system such as concentration of a chemical specie. Multivariate regression models can be developed to estimate those properties using the instrument. In the CO₂ capture field, the use of such types of instruments are gradually becoming popular due to their fast response, ability to locate in-situ and facility to integrate with process automatic control systems. The Applied Chemometrics and Research Group (ACRG) at University of South-Eastern Norway (USN) has developed a method using Raman spectroscopy to determine concentration profiles in an MEA-CO₂-H₂O system based on multivariate analysis (Jinadasa, Jens, Øi, & Halstensen, 2017). The developed method can be used for laboratory experiments, R&D tasks, pilot plant operations and commercial applications of the CO₂ absorption process. It features over the traditional offline analyses due to the fast response. We use two software packages to operate the Raman instrument. The iC Raman 4.1 software which comes together with Raman analyzer is used to configure instrument and acquire measurements while Matlab is used to convert these measurements to concentration values. So far, these steps are carried out manually. Although it takes a couple of minutes to take a measurement from the system, file transferring

between the two software packages and giving commands for data processing and calculations takes time. Therefore the current approach is time consuming.

This paper describes a system development that can join the two software packages into one platform and enables the Raman spectroscopy as a real time analyzer for measuring concentrations of chemical species of the MEA-CO₂-H₂O system. It is also compatible to use by plant operators who have no/less knowledge on working with the data processing and calculations by Matlab.

2 Problem Identification

Several offline analytical instruments and chemical methods are available such as titration, nuclear magnetic resonance (NMR) spectroscopy and chromatography to determine chemical concentration in MEA-CO₂-H₂O system. All these methods are time consuming. Titration by BaCl₂ is common and a well-established method but it needs massive chemical preparation and the analysis takes around 2 hours per sample. NMR spectroscopy and chromatography are mostly limited to laboratory because they do not fit well in plant operations. Spectroscopic methods need prior calibration according to application and are expensive. Some of the spectroscopic instruments cannot be located in an industrial environment.

A fast, online method to monitor CO₂ absorption is important in process control and optimization. Maintaining the chemical concentration of the process streams at required levels is important for getting the target output from the plant. After several feasibility studies, Raman spectroscopy was selected as one of preferable solutions to monitor concentrations in this process. Several studies have been performed using Raman spectroscopy to investigate the concentration profiles in CO₂ capture process by amines. The method developed by USN using Raman spectroscopy was tested and validated at a mini-pilot CO₂ capture plant in Porsgrunn (Jinadasa et al., 2017). In this method, multivariate regression models have been calibrated and validated using test set validation. The raw Raman signals from the instrument was pretreated using techniques available in chemometrics and the regression models are based on partial least squares regression (PLSR) which is a powerful multivariate modelling approach.

However, there is a weak point in this approach as we move between two software platforms to transfer data and calculate the concentration value for each measurement. This operation is carried out manually and prevent the analyzer from being an in-situ monitoring application eventhough the Raman analyzer can take measurements continuously within a few seconds.

2.1 Existing method vs proposed method

When the Raman instrument is used to measure the chemical information in MEA-CO₂-H₂O system, first the Raman probe is fixed into the system. A new file is open in iC Raman 4.1 software and suitable number of scans and laser exposure time is selected. (Kaiser, 2018) A file saving path and a file type (either .spc or .csv) for auto export is configured. When the command to measure the chemical system is given through the software, laser light is sent through the fibre optic cable to the Raman probe to go through the sample. After a scan is finished, a plot of Raman intensity vs Raman wavenumber is shown in the computer connected to the Raman instrument. The software simultaneously saves the data to a data file. To convert such a Raman measurement into a concentration value, the following steps are carried out:

- 1). Taking measurements from the process by running the iC Raman software – the measurements can be batch or continuous-wise
- 2). Saving the data in iC Raman software
- 3). Importing the data file into Matlab
- 4). Start PLS toolbox in Matlab (Eigenvector Research, 2018a)
- 5). Preprocessing the Raman data file using preprocessing methods in PLS toolbox
- 5). Input the data file to a PLS model which has been previously calibrated and validated
- 6). Run the PLS model to get the predicted concentration value
- 7). Showing the resulting concentration value

The disadvantage with this approach is that this system does not give the concentration value at the same time as the measurement is taken. It takes time to import the data from iC Raman software to Matlab and follow the preprocessing step. There should be always an operator to carry out the steps from 3 to 6. It requires knowledge and experience in operation of the mentioned software.

The proposed system aims at removing the barrier of manual file transferring and data processing steps between the two software to save time. In the proposed system there is no need of importing data from iC Raman software to Matlab interface. The user should first open the iC Raman software and Labview interface. After setting the data saving locations, number of scans per measurement and laser exposure time, both software can be started. The readings from the Raman instrument automatically transfers to the labview from iC Raman software. In the Matlab/Labview interface, this data is preprocessed and concentrations are predicted based on the PLS model automatically. Anyone who is not experienced with the software or analysis method can operate the instrument and read the required concentration profiles with time both as numerically and graphically. The predicted concentration data are also saved in an excel file. Figure 1 shows how the software

and hardware link together to get a concentration value using the Raman measurement and the proposed system.

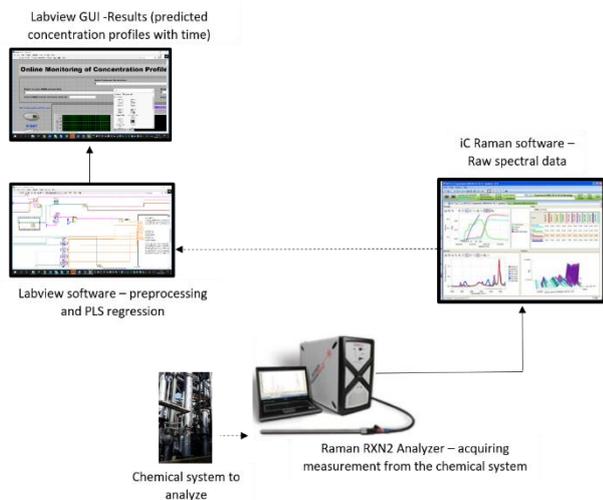


Figure 1. Schematic of Raman spectroscopy measurement showing hardware and software links

3 System development

3.1 Instrumentation

The Raman spectrometer available at USN is the Kaiser RXN2 Analyzer with 785 nm laser wavelength, 400 mW maximum laser power and 100-3425 cm^{-1} spectral range. An immersion optic probe is connected to the RXN2 Analyzer via a fibre optic cable. When the analyzer is switched on to take a measurement, the laser light reaches the molecules in the sample, scattered and the Raman scattered light is filtered by the analyzer. The output display is a plot of Raman intensity (y-axis) which is the Raman scattered radiation and Raman wavenumber (x-axis) which is the frequency difference from the incident radiation (Kaiser, 2018). Peaks and their intensity which appear in this plot carry information about the chemicals present in the system and their composition respectively.

3.2 Process description

The process of CO_2 capture by amine using an absorption column is shown in Figure 2. Flue gas which contains CO_2 is fed upward to the absorber. The CO_2 free flue gas goes out from the top plate of the absorber. The absorbing agent, which is an amine solvent is fed to the top plate of the absorber. This stream is called the 'lean amine' stream. While it flows through the absorber it absorbs CO_2 in the flue gas. The CO_2 rich amine flows out from the absorber bottom plate and this stream is called the 'rich amine' stream. In industry, the difference between CO_2 concentrations in the lean and rich streams is used as an important indicator to know

how much CO_2 is absorbed by the amine. The concentration of other species are also important to yield the maximum process efficiency. The Raman RXN2 analyzer comes with four channels with facility to connect four Raman immersion probes to one analyzer thus providing to measure four streams simultaneously. The aim of this study was to optimize the absorption process in a CO_2 capture process and hence configuration was set for two Raman probes to take measurements from both lean and rich streams.

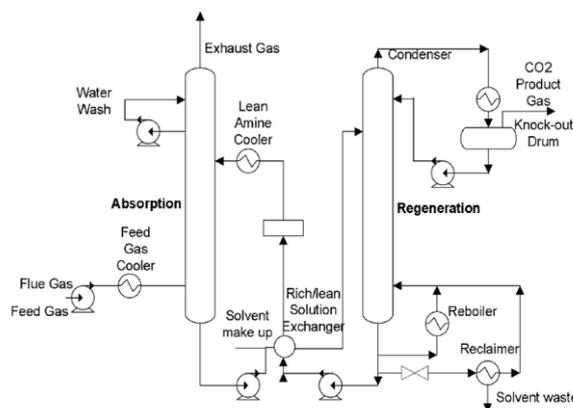


Figure 2. Process flow diagram of CO_2 capture absorption by amine (T. Li & Keener, 2016)

3.3 Data treatment

The outcome of a measurement series using RXN2 Analyzer is a data matrix of $n \times p$ where n is number of objects (eg: it can be different samples or signals with time) and p is 3326 Raman wavenumbers. Since this data matrix contains noise, preprocessing is needed before a model can be calibrated based on the PLSR algorithm. By performing such a data treatment, the noise generated due to interference of other chemical components, laser input variations or instrument noise can be minimized and the model predictability can be improved. There are different preprocessing techniques available in multivariate data analysis. The choice of which pre-processing method depends on many factors such as the instrument type, sample, impurities in the sample, environmental factors and properties of interest to measure. Based on our previous experience in developing PLS models for Raman measurements in MEA- CO_2 - H_2O systems, three combinations of multivariate data processing methods showed lowest prediction error. They were scripted as Matlab codes which were the baseline-whittaker filter, standard normal variate (SNV) and mean centering.

3.4 Estimation of concentrations using the Partial Least Square Regression (PLSR) algorithm

There are different algorithms to develop a PLS model (Andersson, 2009). PLS is available in many commercial software packages, such as The Unscrambler, PLS Toolbox, ProSensus and SIMCA-P. To be independent from using commercial software packages except Matlab/Labview, Matlab codes were used in this study. PLS calibration and validation using the so-called 'test set' method and data pretreatment were coded in Matlab. Procedures mentioned in `pls.m`, `pretreat.m` available at libPLS package was used for PLS regression and mean centering (H.-D. Li, Xu, & Liang, 2018). `snv.m` file and `wlsbaseline.m` file available at PLS toolbox were referred for SNV and whittaker filter coding (Eigenvector Research, 2018b). The values for regression coefficients for each specie was stored in an `m`-file called `regcon.m` file and after data pretreatment, prediction was performed using the equation of regression coefficient values.

Figure 3 shows the flow path when the proposed system is in operation. After launching the iC Raman software the user selects a location to save Raman files in csv format for rich and lean stream. Each measurement is saved according to the format of "name_date_time_sample name" and hence each file is uniquely distinguished. Then the user should move to the Labview GUI and enter the input variables. These variables are the locations of Labview file, location where Raman measurements from lean and rich stream are saved by iC Raman software and location where the predicted concentrations from matlab/labview system for lean and rich stream should be saved as excel files. After entering these inputs, both iC Raman software and labview are started. As shown in Figure 3, after setting the Raman data saving folder path (for lean and rich stream) and locating a location to save predicted concentrations (excel files), the Labview file start searching for a new Raman measurement file. When such a file is found, it is fed as an input to the PLS model. This is the only input variable to the PLS model. Output variable from the PLS model is concentration of specie. For each specie, there is a specific PLS model. Each PLS model gives the output value at the same time.

4 Results and Discussion

4.1 LabVIEW-based software platform

According to Figure 4, the GUI of the Labview based system shows the total CO₂ concentration and five concentrations of chemical species present in both lean and rich amine streams in a CO₂ capture process. The first graph in Figure 4 shows CO₂ concentration in units

of mol per mol MEA. The second graph shows concentration of free (unreacted MEA) and protonated MEA in units of mol per kg of H₂O. The third graph shows concentration of carbonate, bicarbonate and carbamate in units of mols per kg H₂O.

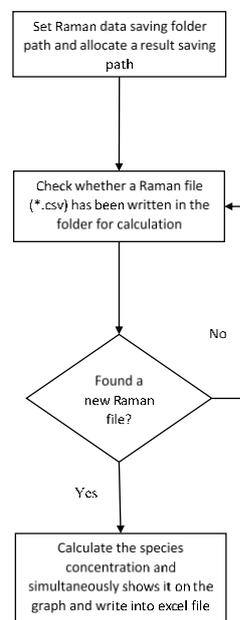


Figure 3. System flow sheet

To show how the online predicted concentration of CO₂ loading matches with offline titration Table 1 provides results of a CO₂ loaded process stream with decreasing CO₂ concentration and prediction error. During this experiment, the Raman spectroscopy was connected to the process stream

Table 1. Validation of the online monitoring system with offline measurements

Time	CO ₂ loading (mol CO ₂ / mols MEA)	
	Raman spectroscopy-predictions from online system ± prediction error	Offline titration results
15:15:56	0.491672 ± 0.0150	0.484216
15:35:41	0.481887 ± 0.0150	0.463091
15:55:26	0.489647 ± 0.0148	0.462196
16:16:21	0.44994 ± 0.0145	0.42835
16.35.08	0.350732 ± 0.0146	0.331385
16.56.04	0.262569 ± 0.0144	0.28351
17.15.53	0.196636 ± 0.0150	0.191744

The concentrations recorded by the Raman system closely match with the offline measurements for CO₂ loading. The average prediction error for Raman measurement is 0.015 mol CO₂ / mol MEA.

Online Monitoring of Concentration Profiles in CO₂-MEA-H₂O System

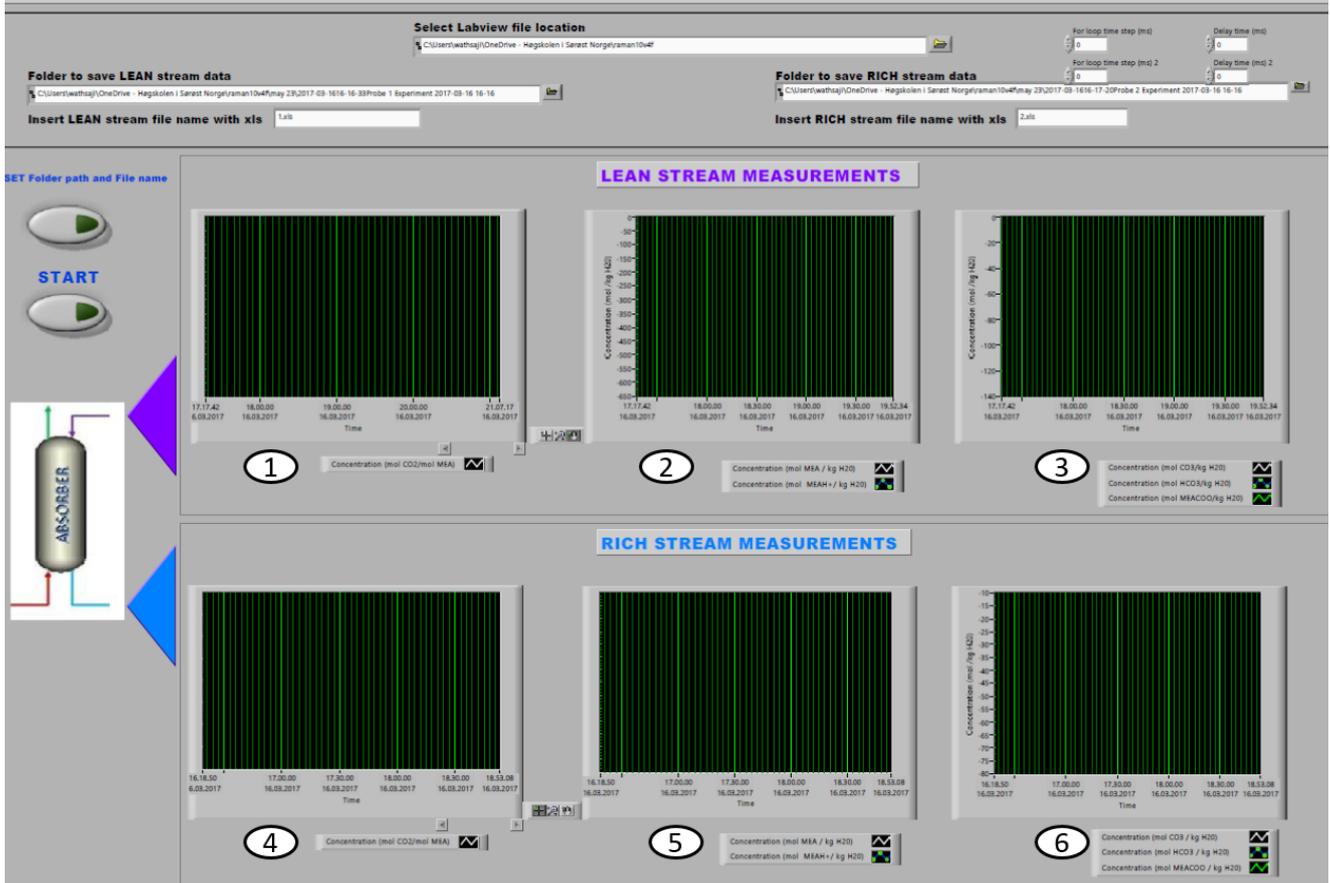


Figure 4. Graphical user interface for concentration profiles; plot 1, plot 2 and plot 3 are CO₂ loading, MEA/MEA⁺ concentrations and CO₃²⁻/HCO₃⁻/MEACOO⁻ for lean stream ; plot 4, plot 5 and plot 6 are CO₂ loading, MEA/MEA⁺ concentrations and CO₃²⁻/HCO₃⁻/MEACOO⁻ for rich stream

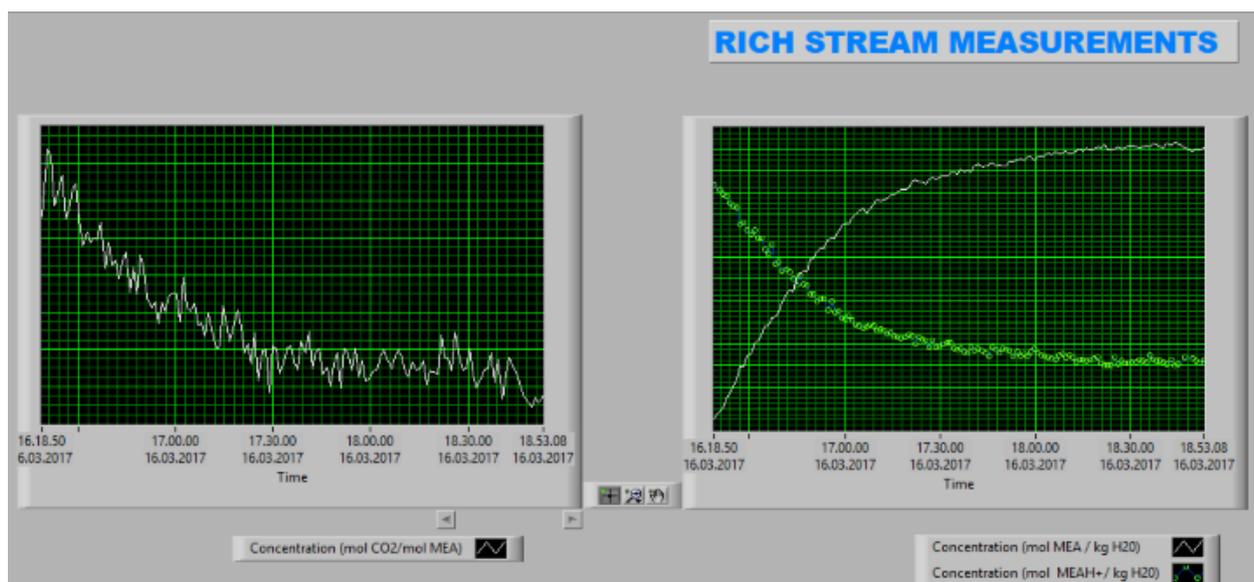


Figure 5. Example for recording real-time measurement for CO₂ loading and MEA/MEA⁺ concentration in rich stream for two and half hour continuous operation of CO₂ desorption process

5 Conclusion and future work

Determining CO₂ absorption in liquid phase of MEA-CO₂-H₂O system by an analytical technique is useful in many aspects. It saves time and gives access to online monitoring of the system. The accuracy of laboratory methods is based on the skill of the analyst, demand time, labour and resource. Offline laboratory methods cannot be used for process automation. Eventhough, these traditional methods are still used to control the process parameters in CO₂ plants due to the non-availability of in-situ analysis methods.

In this paper, we present the development of Labview/Matlab based software platform which is connected to iC Raman software in Raman RXN2 Analyzer. The platform provides concentration profiles of different chemical species present in an MEA-CO₂-H₂O system. These concentration values are calculated indirectly from measurements from Raman analyzer. The calculation is based on partial least square regression method. PLS and data pretreatment algorithms were written as matlab scripts.

If a calibration and validation data set is available, this system can be easily modified to another amine based CO₂ capture system without extensive effort. For instance, there are other amines which have the ability to absorb CO₂ and Raman instrument can be used to determine the total CO₂ absorbed and the concentration of other chemical species. If the user needs to use this GUI for such amines, he can input new calibration and validation data set into PLS script and change the preprocessing script in Matlab accordingly. The developed system can also be used to monitor how the reaction between an amine and CO₂ evolves with time in a batch reaction.

It is also recommended to take the use of the data to perform other chemometric analysis such as principle component analysis, outlier detection and multivariate curve resolution to better understand the chemical system. The plots related to these analysis can be implemented in the developed GUI similar to the concentration plots.

References

- Andersson, M. (2009). A comparison of nine PLS1 algorithms. *Journal of Chemometrics*, 23(10), 518-529. doi:<https://doi.org/10.1002/cem.1248>
- Eigenvector Research, I. (29 March 2013). Advanced preprocessing: Noise, offset, and baseline filtering.
- Eigenvector Research, I. (2018a). PLS toolbox. Retrieved from http://www.eigenvector.com/software/pls_toolbox.htm
- Eigenvector Research, I. (2018b). PLS_Toolbox 8.2.1 software Manson, WA USA 98831. Retrieved from <http://www.eigenvector.com>
- IEA. (2017). CO₂ emissions from fuel combustion: Overview Retrieved from <http://www.iea.org/publications/freepublications/publi>
- [cation/CO2EmissionsFromFuelCombustion2017Overview.pdf](http://www.iea.org/publications/freepublications/publication/CO2EmissionsFromFuelCombustion2017Overview.pdf)
- Jinadasa, M. H. W. N., Jens, K.-J., Øi, L. E., & Halstensen, M. (2017). Raman Spectroscopy as an Online Monitoring Tool for CO₂ Capture Process: Demonstration Using a Laboratory Rig. *Energy Procedia*, 114, 1179-1194. doi:<https://doi.org/10.1016/j.egypro.2017.03.1282>
- Kaiser. (2018). iC Raman™ 4.1 Software. Retrieved from http://www.kosi.com/na_en/products/raman-spectroscopy/raman-software/ic-raman-4.1.php
- Li, H.-D., Xu, Q.-S., & Liang, Y.-Z. (2018). libPLS: An integrated library for partial least squares regression and linear discriminant analysis. *Chemometrics and Intelligent Laboratory Systems*, 176, 34-43. doi:<https://doi.org/10.1016/j.chemolab.2018.03.003>
- Li, T., & Keener, T. (2016). *A review: Desorption of CO₂ from rich solutions in chemical absorption processes* (Vol. 51).

Sensitivity Analysis and Effect of Simulation parameters of CFPD Simulation in Fluidized Beds

Janitha C. Bandara¹ Henrik K. Nielsen² Britt M.E. Moldestad¹ Marianne S. Eikeland¹

¹Department of Process, Energy and Environmental Technology, University College of Southeast Norway, Norway, {Janitha.bandara, britt.moldestad, Marianne.Eikeland}@usn.no

²Department of Engineering Sciences, University of Agder, Norway, {henrik.kofoed.nielsen}@uia.no

Abstract

Fluidized bed technology is broadly applied in industry due to its distinct advantages. CFD simulation of fluidized beds is still challenging compared to single-phase systems and needs extensive validation. Multiphase particle-in-cell is a recently developed lagrangian modeling technique and this work is devoted to analyze the sensitivity of grid size, time step, and model parameters, which are the essences of accurate results. Barracuda VR 17.1.0 commercial CFD package was used in this study.

500 μ m sand particles and air was used as the bed material and fluidization gas respectively. Five different grids, having 27378, 22176, 16819, 9000 and 6656 computational cells were analysed, where five different time steps of 0.05, 0.01, 0.005, 0.001 and 0.0005 were used for each grid. One velocity step was maintained for 8 seconds. The bed pressure drop at packed bed operation was high for simulations with reduced time steps while equal pressure drops were observed during fluidization for all time steps. Time steps of 0.0005s and 0.001s and 0.005s produced equal result of 0.15 m/s for minimum fluidization velocity, irrespective of the grid size. The results from time steps of 0.05 and 0.01 are converged to the results from time steps of 0.005 and 0.001 by increasing simulation time per one velocity step.

Keywords: Fluidized bed, Minimum fluidization velocity, CFD simulations, Multiphase particle-in-cell method, Grid size, Time step

1 Introduction

Gas-solid fluidized bed technology is widely utilized in energy generation, chemical, petrochemical, pharmaceutical, environmental, electronic and metallurgical processing industries due its distinct advantages of high heat and mass transfer, controlled material handling, large thermal inertia of solids and isothermal operating conditions (de Souza Braun et al. 2010)(Vejahati et al. 2009)(Esmaili and Mahinpey 2011). Computational fluid dynamic (CFD) modeling has been identified as an excellent tool to produce

missing information during the scaling up of lab/pilot scale fluidized beds to industrial scale. Further, it is a fast and cost effective method for system optimization.

CFD solves the conservation equations for mass, momentum, energy and species where this technique has been critically validated for the accurate performance in gas or liquid single phase flows. However, there are certain challenges related to interface coupling, solid phase modeling and scale differences in gas-solid multiphase flow systems. Eulerian-eulerian and eulerian-lagrangian are the two basic approaches for CFD modeling of multiphase flows.

Multiphase Particle-In-Cell (MP PIC) modeling is a development of eulerian-lagrangian modeling and aims to reduce the computational cost in discrete modeling of particle phase. Instead of tracking individual particles, it considers the parcels containing a certain number of particles with similar properties. The parcels are modeled in the discrete phase while the particle phase interactions are modeled in Eulerian frame. Therefore particle properties are calculated in both Eulerian and Lagrangian frames, which are correlated via interpolation functions. The successive developments of the MP-PIC method is illustrated in the works of Snider, O'Rourke and Andrew (Andrews and O'Rourke 1996)(D M Snider 2001)(D M Snider, O'Rourke, and Andrews 1998)(Dale M Snider 2007)(O'Rourke and Snider 2012).

Validated CFD models can be used to analyze the bubbling fluidized beds in terms of minimum fluidization velocity, bubble rise velocity, bubble diameter and particle mixing and segregation. The conservation equations of mass, species, momentum and energy are in partial differential form. The particular simulation geometry is divided into small cells, which is referred as the computational grid. The conservation equations are then discretized in space and time to get a set of algebraic equations. Finite difference, finite element and finite volume are the main techniques where the finite volume method is mostly used in mass, momentum and energy related 3D systems.

Errors and uncertainties are integrated from the modeling stage to the final computer simulations. Use of empirical equations and model simplification lead to deviations during the model development. The errors

imposed due to the selection of mesh size, time step and discretization method are referred as numerical errors. Truncation and limiting functions at the discontinuities also cause deviations in the result. Iterative algorithms used in simulations provide certain errors while the round off errors are integrated depending upon computer resource (i.e. 32 bit or 64 bit). Finally, improper coding can also lead to certain errors where these are absorbed as discretization errors. Therefore, it is required to identify the possibilities to reduce the errors in the simulations with minimal computational cost.

As the model equations are concerned, it is possible to check the best functioning empirical models. This includes selecting the best drag model in gas-solid multiphase flow systems. Checking different values for the model constants/coefficients in a meaningful way is another approach. Different schemes such as first order upwind, second order upwind and central differencing etc. can be optimized in terms of computer cost and accuracy required. However, many of the mentioned parameters are optimized for general setting in many of the commercial CFD packages. Hence, the most primary parameters to study in first hand are the grid size and the simulation time step. These two parameters are correlated to form Courant-Friedrichs-Lewy (CFL) conditions, which gives the primary indication of the convergence of the simulation. Apart from the convergence, improper implementation of mesh lead to errors and missing information of the systems.

Mesh sensitivity analysis has used to develop a grid independent model. Many of the related works for the mesh sensitivity were carried out for EE simulations and fixed time steps has been adopted based on convergence criteria. In contrast, as solid phase is modeled as discrete particles in EL modeling, the solid phase interactions are directly calculated. Therefore, the effect of the mesh size is comparatively less. Many authors have used the bed pressure drop and the solid volume fraction as the parameters to check the mesh sensitivity. Even though the MP PIC modeling preserves the discrete nature of the particles, it deviates from the original Lagrangian modeling as selected particle properties are calculated in the Eulerian frame.

Barracuda VR is a tailor-made CFD code for multiphase flow systems, which uses MP PIC modeling. Many of the previous studies on sensitivity analysis have been carried out in steady boundary conditions. Instead, this work is focused on studying both changing boundary conditions of inlet flow velocity during the transition of packed bed to fluidized bed and steady boundary conditions in bubbling fluidizing regime. The minimum fluidization velocity, bed pressure drop and bubble characteristics were compared. Barracuda VR 17.1.0 version was used to compare the minimum fluidization velocity, bed pressure drop and bubble characteristics between different configurations of time

step, grid size along with different models and model parameters.

2 MP PIC Model Description

The gas phase mass and momentum conservation are modeled with continuity and time averaged Navier-Stokes equations:

$$\frac{\partial(\alpha_g \rho_g)}{\partial t} + \nabla \cdot (\alpha_g \rho_g u_g) = 0 \quad (1)$$

$$\frac{\partial(\alpha_g \rho_g u_g)}{\partial t} + \nabla \cdot (\alpha_g \rho_g u_g u_g) = -\nabla P + F + \nabla \cdot (\alpha_g \tau_g) + \alpha_g \rho_g g \quad (2)$$

Where α_g , ρ_g , and u_g are gas phase volume fraction, density and velocity respectively. F is total momentum exchange with particle phase per volume, g is gravitational acceleration, P is pressure and τ_g is the gas phase stress tensor, which is given by:

$$\tau_g = \mu_g \left[(\nabla u_g + \Delta u_g^T) - \frac{2}{3} \nabla \cdot u_g I \right] \quad (3)$$

μ_g refers to the shear viscosity that is the sum of the laminar and turbulent components. The large eddy simulation is used for the large-scale turbulence modeling while the subgrid scale turbulence is captured with Smagorinsky model:

$$\mu_{g,t} = C_s \rho_g \Delta^2 |\nabla u_g + \Delta u_g^T| \quad (4)$$

Where Δ is the subgrid length scale and calculated by equation 05. The default value for the model constant C_s is 0.01.

$$\Delta = (\delta x \delta y \delta z)^{1/3} \quad (5)$$

The interface momentum transfer is calculated through the viscous drag force:

$$F = \iint f \left\{ m_p \left[D_p (u_g - u_p) - \frac{\nabla P}{\rho_p} \right] \right\} dm_p du_p \quad (6)$$

Subscript P refers to the particle phase properties where m and u symbolizes the mass and velocity. D_p is the drag function. The particle phase dynamics are derived using particle distribution function (PDF) calculated from the Liouville equation given as:

$$\frac{\partial f}{\partial t} + \nabla (f u_p) + \nabla u_p (f A_p) = 0 \quad (7)$$

Where A_p , is the particle acceleration and is expressed by:

$$A_p = \frac{\partial(u_p)}{\partial t} = D_p (u_g - u_p) - \frac{\nabla P}{\rho_p} - \frac{\nabla \tau_p}{\rho_p \alpha_p} + g \quad (8)$$

α_p is particle volume fraction and τ_p is particle stress function that is used in formulating interphase interactions of particles.

$$\alpha_p = \iint f \frac{m_p}{\rho_p} dm_p du_p \quad (9)$$

$$\tau_p = \frac{10P_s \alpha_p^\beta}{\max[(\alpha_{cp} - \alpha_p), \varepsilon(1 - \alpha_p)]} \quad (10)$$

P_s is a constant with the units of pressure, α_{cp} is the particle volume fraction at close packing, β is a constant between 2 and 5 where ε is a very small number on the order of 10^{-7} .

3 Methods and Computational Model

The minimum fluidization velocity (MFV) was used as the primary measurement for the mesh and time step sensitivity analysis. The simulations were started at the packed bed conditions and the gas velocity was gradually increased from zero to 0.4 m/s with 0.025 increments. Simulations were carried out for 8 seconds at each velocity step. Average pressure drop gradient across the column was plotted against the gas superficial velocity and the minimum fluidization velocity is read (U_{mf}) as illustrated in Figure 1. Five different grid sizes and five different time steps for each grid were used to compare the MFVs. The simulation time for each velocity step was gradually increased in the following simulations gradually up to 20 seconds in selected grids and the results were compared. As the drag model is a function of particle volume fraction, the MFV was analyzed at varied close pack volume fractions.

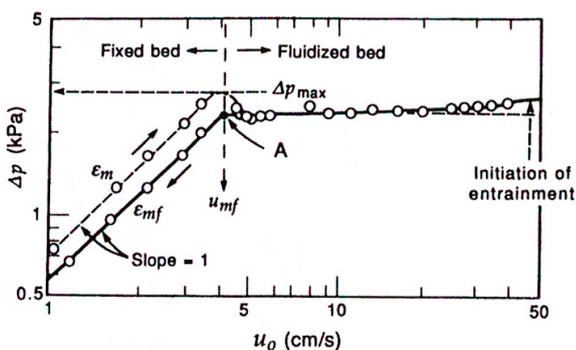


Figure 1. Calculation of minimum fluidization velocity

3.1 Computational model

The dimensions of the geometry were adopted from the experimental rig at the University of Southeast Norway. As shown in Figure 2, a cylindrical column with 84mm in diameter and 1000mm in height with pressure monitoring points in 100mm intervals along the height was created. The gas inlet was set up as a flow boundary while the top gas exit as a pressure boundary at atmospheric pressure with no particle exit. Fluidizing gas was air at 300K with varying superficial velocity.

Further, the velocity inlet was formulated as it homogeneously injects air in axial direction throughout the whole bottom cross section. Each velocity was maintained for 8 seconds. Spherical sand particles with 2200 Kg/m³ in density and 500 micron in diameter was the bed material used. The initial particle bed height was set up to 350mm.

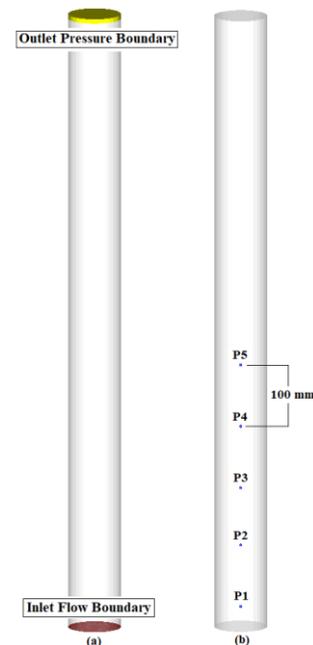


Figure 2. (a) Boundary conditions, (b) Pressure points
Five different meshes with 6656, 9000, 16819, 22176 and 27378 cells were tested and cross sectional views are illustrated in Figure 3.

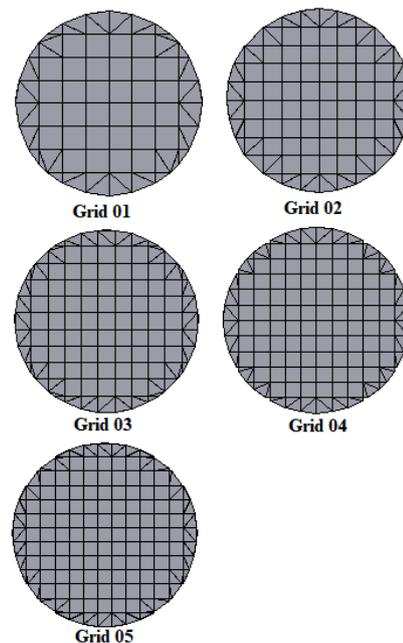


Figure 3. Cross sectional views of different grids
The grid dimensions in x, y and z direction for each mesh are given in Table 1. The normalized grid size in all x, y and z directions were kept below the warning

line in the grid check plot. Grid refinements at the wall was not performed as it was assumed that there was no boundary layer formation with the dense phase particle system. Default grid generator settings were used, which removes the cells having less fraction of volume than 0.04 and greater aspect ratio than 15:1. Four time steps of 0.05, 0.01, 0.005 and 0.001 seconds were checked for each grid.

Table 1. Cell dimensions

No of cells	ΔX (mm)	ΔY (mm)	ΔZ (mm)	Grid No
6656	10.5	10.5	9.6	01
9000	8.40	8.40	11.1	02
16819	7.60	7.60	7.20	03
22176	7.00	7.00	6.5	04
27378	6.46	6.46	6.17	05

Adopting to the previous experience of the author (Bandara, Thapa, Moldestad, & Eikeland, 2016), Wen-Yu-Ergun correlation was used for the initial simulations. It is a combined formulation of Wen-Yu model and Ergun model, which is selected upon the gas volume fraction. When the gas volume fraction is greater than 0.8, Wen-Yu correlation is applied which is given by,

$$K_{\text{WenYu}} = \frac{3 C_d \rho_g \varepsilon_g (1 - \varepsilon_g) (u_s - u_g)}{4 d_p} \varepsilon_g^{-2.65} \quad (11)$$

Where C_d is given by,

$$C_d = \begin{cases} \frac{24}{\varepsilon_g Re_s} \left[1 + 0.15 (\varepsilon_g Re_s)^{0.687} \right], & Re_s \leq 1000 \\ 0.44, & Re_s > 1000 \end{cases} \quad (12)$$

When the gas volume fraction is less than 0.8, Ergun correlation is used which is given by,

$$K_{\text{Ergun}} = 150 \frac{\mu_g (1 - \varepsilon_g)^2}{\varphi^2 d_p^2 \varepsilon_g} + 1.75 \frac{\rho_g (u_g - u_s) (1 - \varepsilon_g)}{\varphi d_p} \quad (13)$$

Where, subscripts g, p and s refer to gas phase, particle and solid phase respectively. K_{sg} is the interface momentum transfer coefficient, U is the velocity, ρ is the density, ε is the volume fractions, φ is the sphericity, μ is the viscosity, Re is the Reynold's number and d is the particle diameter.

The close pack volume fraction, maximum momentum redirection from collisions, normal to wall momentum retention and tangent to wall momentum retention were set to 0.6, 40%, 0.3 and 0.99 respectively. Default values for the parameters in the particle stress model were kept unchanged. Large eddy simulation was enabled for the turbulence modeling and "partial-donor-cell" was used as the numerical scheme.

4 Results and Discussion

Minimum fluidization is a crucial parameter as it represents the minimum gas required to operate the reactor. It is sensitive to particle properties (size, shape, density etc.) and gas properties (density, humidity, viscosity etc.) along with geometry (aspect ratio). Therefore, it is required to know the minimum fluidization velocity at different contexts. A CFD model can be useful in predicting MFV at various process conditions. This work demonstrates the grid size and time step dependency in calculating the minimum fluidization velocity.

Apart from the 20 simulations mentioned under the methods, time step of 0.0005 was used for grids with high resolutions and one other simulation was carried out at a coarse grid. The plots were generated for each grid at different time steps and each time step for different grids.

According to the force balance at the minimum fluidization condition, the bed pressure drop is proportional to the particle weight and can be expressed as,

$$\frac{\Delta P}{H} = (1 - \varepsilon_{mf}) g (\rho_s - \rho_g) \quad (14)$$

Where, ΔP is bed pressure drop, H is bed height and ε_{mf} is the void fraction at the minimum fluidization.

Equation 14 and many correlations for the MFV need the knowledge of void fraction at minimum fluidization, which is difficult to determine. However, Gidaspo (1994) and Das et al have mentioned about the void fraction at minimum fluidization (ε_{mf}), which varies between 0.44 and 0.476. Implementing the value of 0.45 for the ε_{mf} in equation 14, the pressure drop per unit height of the bed at minimum fluidization is 11.87 Pa/mm.

As approximated by Wen and Yu, the minimum fluidization velocity U_{mf} can be expressed as,

$$U_{mf} = \frac{\mu_g}{\rho_g d_p} \left[\sqrt{1135.7 + 0.048 Ar} - 33.7 \right] \quad (15)$$

Where μ_g is gas viscosity, d_p is particle diameter and Ar is the Archimedes number given by,

$$Ar = \frac{d_p^3 (\rho_s - \rho_g) \rho_g g}{\mu^2} \quad (16)$$

Using Equation 15 and 16, the MFV for the simulated system can be calculated as 0.165 m/s.

Air velocity (u_g) vs pressure drop (ΔP) plots for different grids are illustrated in Figure 4 to 8. Each figure contains plots for different time steps used. Each velocity step was maintained for 8 seconds and the pressure drop was taken as the average value of the 8th second of respective velocity. The averaging was performed to minimize the effect of random pressure fluctuations during fluidization on results. The pressure gradient (Pa/mm) along the column height was

calculated based on P1 and P2 data as illustrated in sketch (b) - **Figure 2**. It was assumed that the P1 and P2 would reach the steady values before the rest of the transient data points (P3, P4 and P5) for pressure.

4.1 Bed Pressure drop

Being a fundamental formulation, pressure drop at onset of fluidization calculated from Equation 14, which is 11.87 Pa/mm, was used as the baseline to compare the results from simulations.

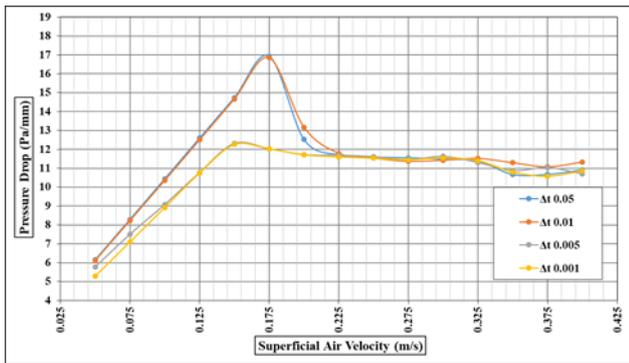


Figure 4. Effect of the time step for MFV at grid 01

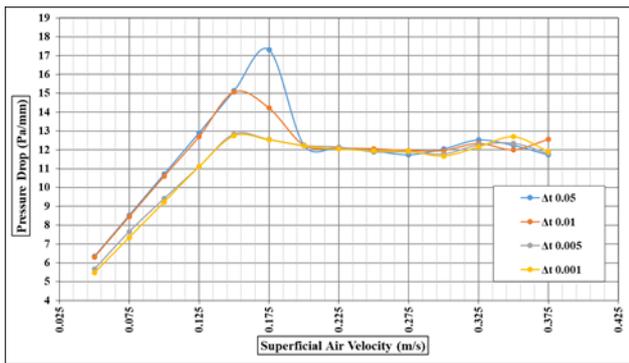


Figure 5. Effect of the time step for MFV at grid 02

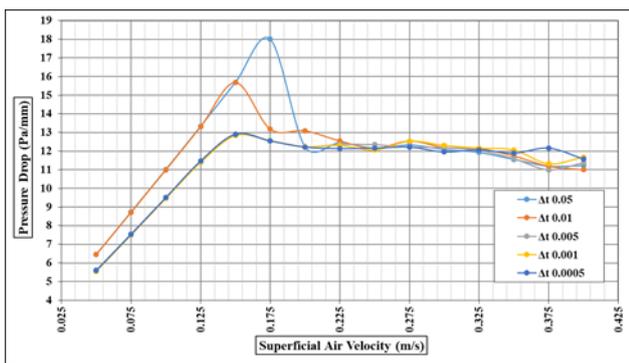


Figure 6. Effect of the time step for MFV at grid 03

4.1.1 Pressure drop at minimum fluidization

The results for the pressure drop at minimum fluidization ($(\Delta P)_{MF}$) using time step 0.05 show the highest variation of 18 Pa/mm in grid 05 and 03. The respective value changes between 17 and 18 Pa/mm at different grids without any distinguishable pattern. At the coarsest grid, grid 01, both time steps of 0.05 and

0.01 give the same of 17 Pa/mm for $(\Delta P)_{MF}$. However, the $(\Delta P)_{MF}$ using time step 0.01 gradually increases from 15 Pa/mm to 16.5 Pa/mm as the grid size is reduced from grid 02 to grid 05. The $(\Delta P)_{MF}$ calculated from time steps of 0.005 and 0.001 are identical for each grid, which gradually increase from 12.5 Pa/mm in grid 01 to 13.5 Pa/mm in grid 05.

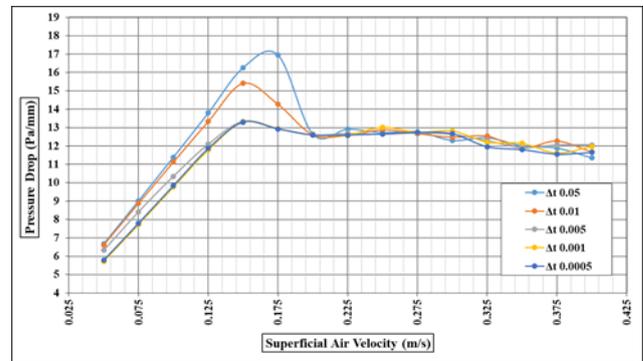


Figure 7. Effect of the time step for MFV at grid 04

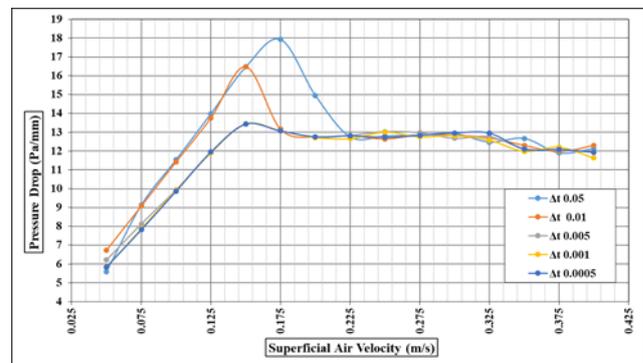


Figure 8. Effect of the time step for MFV at grid 05

4.1.2 Pressure drop during packed bed

Simulation results from time steps of 0.05 and 0.01 behaves almost equally at each grid during packed bed operation. The observed ΔP s are considerably higher compared to time steps of 0.005 and 0.001 at each velocity step. The curves from time steps of 0.001 and 0.0005 are identical throughout the full range of air velocities. The ΔP using time step 0.005 almost follow the time step of 0.001 with slight over prediction in grid 01 and 04. However, the curve converges to that of the time step of 0.001 before the onset of fluidization.

4.1.3 Pressure drop at fluidization regime

The ΔP during fluidization was similar for all the time steps at each grid. However, respective value increases from 11.5 Pa/mm in grid 01 to 13 Pa/mm in grid 02. Almost steady pressure drops can be observed for time steps of 0.0005, 0.001 and 0.005 between 0.2 m/s and 0.325 m/s air velocities. After 0.325 m/s of air velocity, the ΔP starts to fluctuate for all the simulations. The ΔP is dropped down nearly by 1 Pa/mm after 0.325 m/s air velocity except in grid 02, in which the ΔP is slightly increased.

4.2 Minimum Fluidization Velocity

The results for minimum fluidization velocity does not show much variations over the grid sizes. The MFV obtained from time step of 0.05 is 0.175 m/s for all the grids. Time steps of 0.005, 0.001 and 0.0005 produce the same MFV of 0.15 m/s irrespective of the grid size. The time step of 0.01 gives the same MFV velocity of 0.15 m/s for grid 02, 03, 04 and 05 where in grid 01, MFV is increased to 0.175 m/s.

The observed differences in the ΔP and MFV might be related to CFL conditions or not reaching steady state conditions at each velocity steps. The CFL equation is given by:

$$CFL = u \frac{\Delta t}{\Delta x} \quad (17)$$

Where Δt is time step, Δx is cell size (one dimensional modeling) and u is the convective flow velocity.

At lower time steps the air flow is not fully developed. This effect is progressively increased along the column height. Due to that, the air velocity is getting lesser along the height, which force the cells near the inlet flow boundary to store more air according to the step wise increment of air velocity. This leads to increased pressures near the inlet boundary and consequently increased pressure drop gradients. Therefore, the pressure gradients along the height are less linear for higher time steps. Hence, selecting the transient data points of P1, P2...P3 (refer sketch (b) in Figure 2) to calculate the pressure drop gradient was critical for previous simulation results. This variation is clearly illustrated in air velocity vs pressure drop plots in Figure 9. The simulation results from grid 03 was used and the pressure gradients were calculated using different transient data points according to the formulations mentioned at the lower right hand corner of each plot. The time steps of 0.005 and 0.001 produce almost same results irrespective of the transient data points used. Even though the $(\Delta P)_{MF}$ is high with time step of 0.05, all the plots follow a similar trend. In contrast, the curves for time step 0.01 show higher deviations from each other and however, with less $(\Delta P)_{MF}$ compared to time step 0.05. The collective outcome of these results clearly illustrates that the system has not achieved steady state operation completely with the implemented boundary conditions at lower time steps of 0.05 and 0.01.

Therefore, further simulations were carried out for time steps of 0.05 and 0.01 with extended simulation time of 14 seconds and 20 seconds for each velocity step. Grid 03 was used and air velocity vs pressure drop plots are illustrated in Figure 10 and Figure 11 along with the results from 8 seconds simulation time.

When the plots in Figure 11 are compared, results from 14 second and 20 second simulation time are converged to same values in terms of both $(\Delta P)_{MF}$ and

MFV. $(\Delta P)_{MF}$ remains at 13.6 Pa/mm while the MFV is further reduced to 0.13 m/s. This suggests the inability of further improvement of the results merely by increasing the simulation time for time step 0.01. Therefore, it is necessary to carry out additional simulations with increased simulation time for time steps of 0.005 and 0.001 for other grids to see the provisions for the improvements of the results.

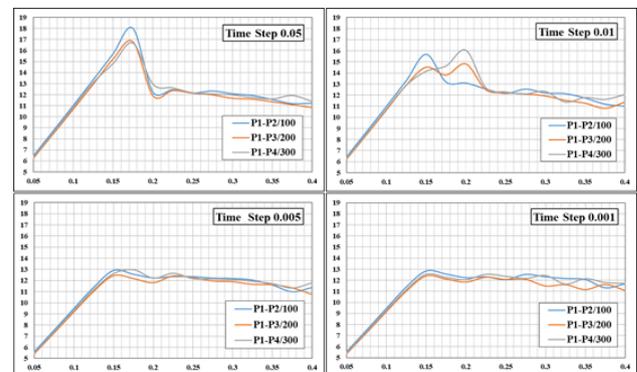


Figure 9. Effect of the time step for MFV at grid 03

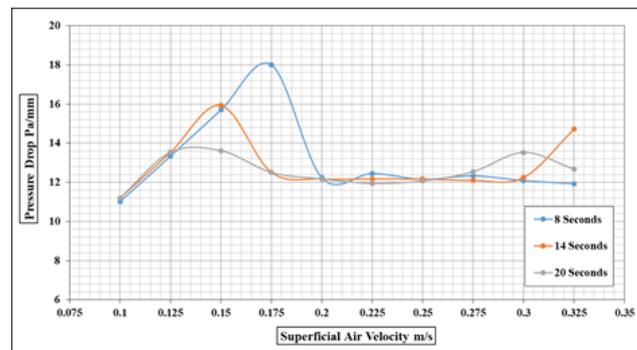


Figure 10. Effect of the simulation time for MFV and pressure drop at time step of 0.05 seconds

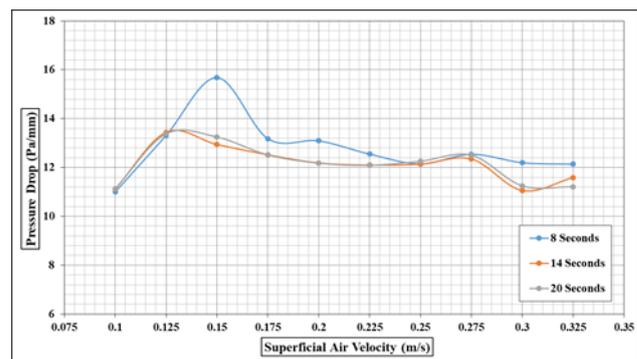


Figure 11. Effect of the simulation time for MFV and pressure drop at time step of 0.01 seconds

4.3 Effect of the close volume fraction for minimum fluidization velocity

Most of the drag models are a function of particle volume fraction (α_p), which is changed depending on particle shape and size distribution. The previously illustrated simulation results were based on close volume fraction of 0.6 and successive simulations are

carried out for 0.55, 0.58 and 0.65. The results are illustrated in Figure 12.

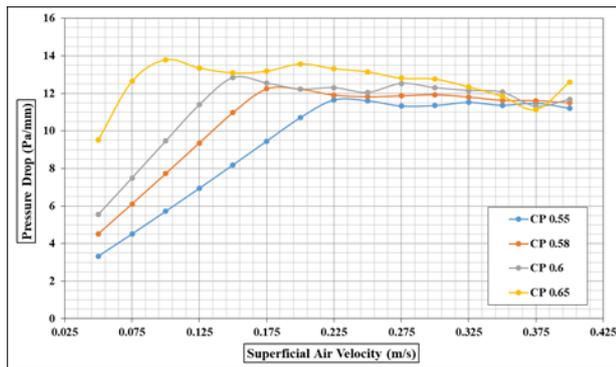


Figure 12. Effect of the close volume fraction for MFV and pressure drop

Grid 03 and time step of 0.001 second were used for the simulations. The pressure drop during packed bed operation is increased with the increased close pack volume fraction. The particles are closely packed that makes it hard for gas to pass through. A slight change in the $(\Delta P)_{MF}$ can also be observed, which is increased proportionally with close volume fraction. There is a significant variation in the MFV, which is reduced down to 0.1 m/s at close volume fraction of 0.65 and as high as 0.225 m/s at 0.55. The drag functions are functions of the particle volume fraction which leads to the difference in MFV. The bed pressure drops are converged together as the air velocity is increased. This is because, the densely packed particles are loosened and attain a more or less common particle volume fraction as the system undergoes rigorous fluidization.

4.4 Effect of the grid size for bubble behavior

The differences in the scales involved is one of the main challenges related to CFD modeling of multiphase systems. Mostly, the particles are in sub-millimeter range while the reactors are in scale of meters. Further, the computational grid can be in the scale of millimeters, centimeters or either in meters depending upon the size of the geometry and computational capacity. Unlike in packed beds, bubbling fluidized beds contain a dense particle phase and a dilute bubble phase. Therefore, the grid should be fine enough to capture the bubble properties as the bubbles play an important role in heat and mass transfer along with particle mixing inside the bed.

Grid 01, 02... 05 and a coarser grid having 2000 cells were simulated for 50 seconds in the bubbling fluidization regime. The time step of 0.001 seconds was used and a constant air velocity of 0,225 m/s was maintained. The behavior of the bubbles in the 40th second of the simulation are illustrated in Figure 13. Smaller and increased number of bubbles appears in the finer grids of grid 05 and 04. The bubble size is becoming larger as the grid size is increased. Finally, the

bubbles are almost disappeared at the coarsest grid with 2000 cells. Therefore, the grid should be fine enough to capture the localized bubble structures. In this case, grid 03 seems to be good enough because, the grid 04 and 05 produce almost the same bubble size.

5 Conclusion

The main objective of the paper was to analyze the effect of time step and grid size for the results in MP PIC modeling. The CPFV commercial package of Barracuda was used in this work. The results give a guidance about the critical parameters to be considered rather than presenting details with model validation.

The minimum fluidization velocity and pressure drop at minimum fluidization were greatly affected by the time step and however, it could be improved by increasing the simulation time. Time steps 0.005s and 0.001s produce the same of minimum fluidization velocity of 0.15 m/s irrespective of the grid resolution. The bed pressure drop at bubbling fluidization regime was not affected considerably by the time step which is 12 pa/mm. However, the minimum fluidization velocity could converge together when the simulation time for a particular air velocity was increased.

The grid size showed a minimal effect on the minimum fluidization velocity. However, the grid size had a great effect on the bubble size and consequently on the bed hydrodynamics. The close volume fraction was also found to be a deciding parameter in simulations for finding the minimum fluidization velocity.

Therefore, the simulation set up should be well optimized depending on the required accuracy of the results and availability of computer power. The physical parameters such as close volume fraction should be accurately measured and implemented in the simulations.

Acknowledgements

The authors like to forward their gratitude to University College of Southeast Norway for providing the Barracuda VR CFD package and computer resources for simulation.

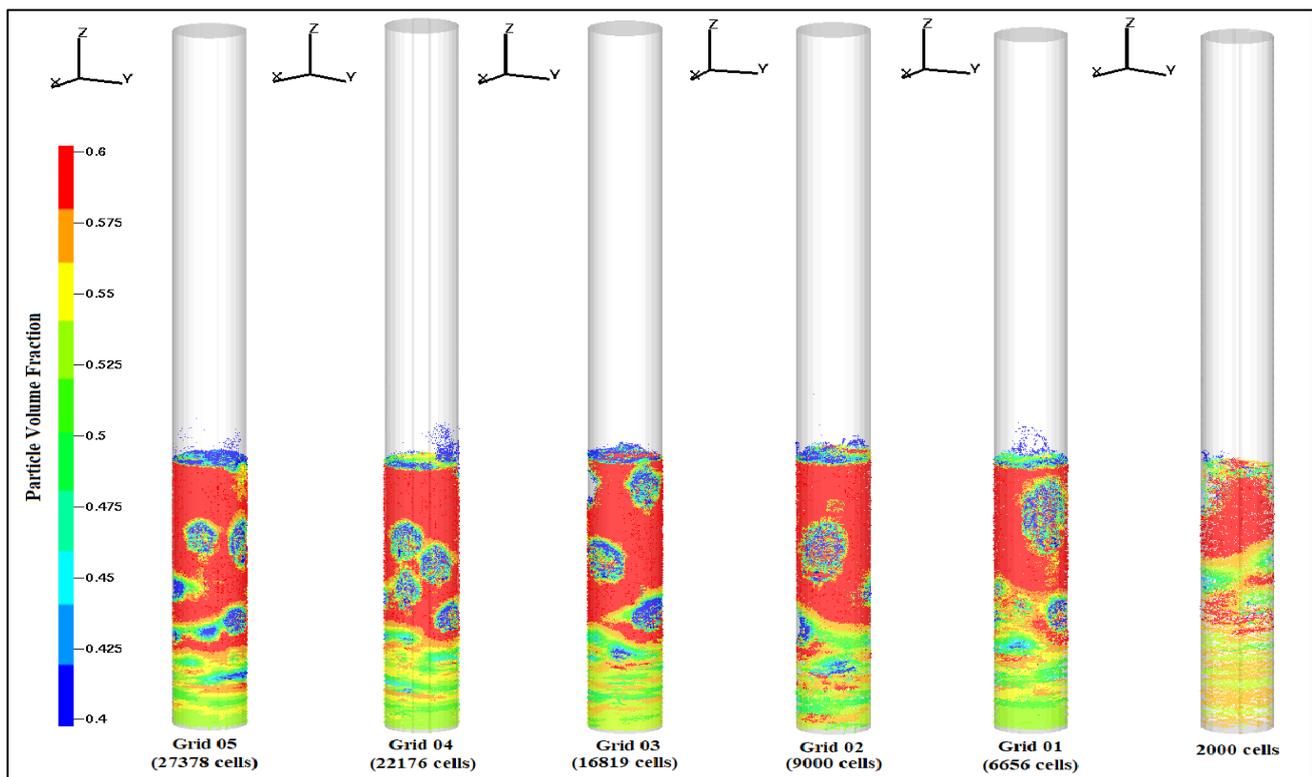


Figure 13. Effect of the computational grid size for the bubble size at 0.225 m/s superficial air velocity

References

- Andrews, M J, and P J O'Rourke. 1996. "The Multiphase Particle-in-Cell (MP-PIC) Method for Dense Particulate Flows." *International Journal of Multiphase Flow* 22 (2): 379–402.
- Gidaspow. 1994, *Multiphase Flow and Fluidization*, Academic press
- Esmaili, Ehsan, and Nader Mahinpey. 2011. "Adjustment of Drag Coefficient Correlations in Three Dimensional CFD Simulation of Gas–solid Bubbling Fluidized Bed." *Advances in Engineering Software* 42 (6): 375–86.
- Geldart, D. 1973. "Types of Gas Fluidization." *Powder Technology* 7 (5): 285–92.
- O'Rourke, Peter J, and Dale M Snider. 2012. "Inclusion of Collisional Return-to-Isotropy in the MP-PIC Method." *Chemical Engineering Science* 80: 39–54.
- Snider, Dale M. 2007. "Three Fundamental Granular Flow Experiments and CPFD Predictions." *Powder Technology* 176 (1): 36–46.
- Snider, D M. 2001. "An Incompressible Three-Dimensional Multiphase Particle-in-Cell Model for Dense Particle Flows." *Journal of Computational Physics* 170 (2): 523–49.
- Snider, D M, P J O'Rourke, and M J Andrews. 1998. "Sediment Flow in Inclined Vessels Calculated Using a Multiphase Particle-in-Cell Model for Dense Particle Flows." *International Journal of Multiphase Flow* 24 (8): 1359–82.
- Souza Braun, Meire Pereira de, Andreza Tangerino Mineto, Hélio Aparecido Navarro, Luben Cabezas-Gómez, and Renato César da Silva. 2010. "The Effect of Numerical Diffusion and the Influence of Computational Grid over Gas–solid Two-Phase Flow in a Bubbling Fluidized Bed." *Mathematical and Computer Modelling* 52 (9): 1390–1402.
- Vejahati, Farshid, Nader Mahinpey, Naoko Ellis, and Mehrdokht B. Nikoo. 2009. "CFD Simulation of Gas-Solid Bubbling Fluidized Bed: A New Method for Adjusting Drag Law." *Canadian Journal of Chemical Engineering* 87 (1): 19–30.
- Bandara, J. C., Thapa, R. K., Moldestad, B. M. E., & Eikeland, M. S. (2016). *Simulation of Particle Segregation in Fluidized Beds* Paper presented at the 9th EUROSIM Congress on Modelling and Simulation, Oulu, Finland.

SIMS2017 The impact of the weather data file on the energy performance certificate, the case of Norway

Alex Gonzalez Caceres¹ Dimitrios G. Zenginis² Tor Arvid Vik¹

¹Department of Civil Engineering and Energy Technology, OsloMet – Oslo Metropolitan University, Norway, alexgc@oslomet.no

²Department of Civil Engineering, Aristotle University of Thessaloniki, Greece

Abstract

Energy Performance Building Directive (EPBD) ask the Member of States to develop a mandatory energy labelling scheme for new and existing buildings, which should include a label rating of the energy efficiency of the building and a list of recommended energy saving measures. The label will provide prospective buyers and tenants of a building with correct information about the energy performance of the building to compare with other options. However, some countries use only one or a limited number of weather zones. In a country like Norway, with significant variations in weather conditions between locations, this is likely to cause a deviation when comparing with the reality. This study aims to present the implication of using only one weather zone in Norway. The method used is based on the comparison of three types of weather files. The first one is used in the labelling system, which is a typical year, while the others are typical reference years from the local site from different providers. The results show significant differences in energy consumption, savings and labelling ratings when applying files with local weather data instead of the official weather data file used in Norway.

Keywords: certificate, energy labelling, IWEC, weather file, reference year, EPC

1 Introduction

The Energy Performance Building Directive (2010/31/EU) requires the Member States to lay down the necessary measures to establish a system of certification of the energy performance of buildings, including a methodology for the calculation of the energy performance of buildings. The building energy certification and rating system is an essential method for improving energy efficiency, minimizing energy consumption and enabling greater transparency regarding energy use in buildings. Basically, in a building energy rating scheme, a standardized method to evaluate and compare energy use, energy efficiency and energy cost in buildings, is provided (Institute for Market Transformation (IMT), 2009). Building rating schemes usually have common purposes. Their

implementation and design can have large differences, though, due to local conditions in each country, such as building stock, climate and Landlord-Tenant legislation (International Energy Agency (IEA), 2010).

The final goal of building energy rating schemes is to reduce energy consumption in the building sector. In order to do so it has to be accompanied by other policy packages, since the labelling does not improve the energy efficiency itself. Instead, the energy labelling allow to assess the efficacy of policies, and to improve and design complementary instruments to achieve energy reduction in buildings (Taskroup, 2014; Hinge, Neely, & Taylor, 2014). In order to obtain positive results through the system of policies for energy reduction in buildings, tools like energy labelling should provide robust data. Stakeholders must be confident that the analysis can predict relevant metrics such as energy use and energy savings accurately (Polly, Kruis, & Roberts, 2011).

This research focus at the accuracy of the energy labelling tools under the Norwegian scheme. The aspects covered in this study are mainly the impact of the weather data from different sources analysing the energy performance, energy rating and renovation strategies under the Energy Performance Certificate (EPC), with special attention to its variations.

1.1 Importance of the weather file

The energy consumption and indoor environment can be determined by considering the impact of different factors acting in the energy and mass transfer. These factors are weather data, occupant behaviour and auxiliary systems (Hensen, 1999). The parameter of the weather is within the most important factors that impact the energy demand of buildings (Fumo, 2014). Even more, the selected weather data should be considered carefully. Due to this, weather data quality will determine the effectiveness of building design strategies and the accuracy of the energy calculation (Hui & Cheung, 1997). Thus, the weather file becomes a major component in order to achieve reliable energy savings from energy management practices and retrofits (Fumo, 2014).

In addition, the weather data is completely out of the control of the person performing an energy labelling calculation (Erba, Causone, & Armani, 2017). Because of this, decisions related to the design of the labelling tools, should exhaustively include the criteria for selection of weather data.

1.2 Type of weather files

Weather data files are used in simulation according to the purposes of the study, and each developer designs their data to meet a particular need (Crawley, 1998). Weather data files are usually generated to represent long-term statistical trends and patterns of the climate for a long period of record.

The weather data file can be generated mainly by two primary sources: measured weather data using physical sensors, and simulated data using mathematical models (Hong, Chang, & Lin, 2013). The preparation of these weather data files includes many variables, which are not necessarily correlated, and thus represent a great challenge. For energy labelling tools, where the aim is to compare different buildings, a typical weather year file will normally be used. When the aim is to obtain retrofit recommendations, average weather data would be most practical (Fumo, 2014). According to Crawley (Crawley, 1998), typical reference year data should be avoided for energy simulations, since they cannot represent typical long-term weather patterns. Several methods have been proposed to generate annual hourly weather files, also known as reference year, which are the most common for energy simulations. Examples are Typical Meteorological Year (TMY), test reference year (TRY), weather year for energy calculation (WYEC), International Weather for Energy Calculations (IWEC), as well as the design reference year (DRY).

Reference year files are a single year composed of 12 months selected as the most representative in a multi-year series. These files are characterized by having a true frequency, true sequences and true correlations (Lund, 1991; Lund & Eidorff, 1981), the last one being the most important (Guan, 2009). This means that a reference year has to be prepared starting from actual weather recordings, selecting the most typical ones without altering the cross-correlations among weather variables (Pernigotto, Prada, Cappelletti, & Gasparella, 2017).

Among the most widespread weather data is the IWEC (Gherri, 2015; Roetzel & Tsangrassoulis, 2011), which were developed under the American Society of Heating, Refrigerating, and Air-Conditioning Engineers (ASHRAE) research project RP-1015 (Thevenard & Brunger, 2002). ASHRAE collected hourly raw data worldwide in attempt to unify weather files in a format similar to TMY3 (Herrera et al., 2017). A comprehensive amount of observations are considered, covering from 12 to 25 years of records, including parameters such as wind speed and direction, sky cover,

visibility, ceiling height, dry-bulb temperature, dew-point temperature, atmospheric pressure, liquid precipitation etc. Hourly data for radiation are estimated empirically based on other parameters. Recently, ASHRAE has updated the format into a IWEC2 (Y. J. Huang, 2014), which contain an improved estimation of the solar radiation. Currently there are available 3012 weather files for locations outside USA and Canada (J. Huang, 2010).

Lack of long-term weather records usually limits the generation of typical annual weather data files in any format (Al-Mofeez, Numan, Alshaibani, & Al-Maziad, 2012). Because of this, it is common for engineers to use synthetic weather data, generated e.g. by the commercial software Meteororm. This tool includes a database for solar energy applications combined with a synthetic weather data generator. It can produce weather files for any site with a time resolution of an hour or even a minute (Remund, Müller, Schilter, & Rihm, 2010). Meteororm integrates a climate database of measurements collected from various sources (8350 weather stations by 2018), which allow to create an average year based on monthly mean values from a long-term dataset.

1.3 Energy labelling system

The certificate is required whenever a building is constructed or for existing buildings, before it is marketed for sale or rent. EPCs are produced using standard methods with standard assumptions about energy usage so that the energy efficiency of one building can easily be compared with another building of the same type (Cappelletti, Dalla Mora, Peron, Romagnoni, & Ruggeri, 2015). EPCs and inspection reports intend to provide accurate and valuable information to building owners and tenants on the energy performance of their buildings. It is also supposed to specify recommendations about how the property can be improved, which will be an important factor to help building owners in a tangible way. According to the IDEAL EPBD project, the trust in the energy performance rating was found to be much higher if specific recommendations on how to improve the energy performance of the building are included in the certificates (Backhaus, Tigchelaar, & de Best-Waldhober, 2011).

Since the EPBD allow some freedom in the implementation of the labelling system, many countries have implemented the EPC system according to the local needs. With this variety of certification schemes, challenges were identified such as how to improve the quality of the results and along with that the recommendations, without increasing the values of the certificates. One of the suggestions is to ensure that recommendations accompanying the EPC relate to actual climate and energy consumption (Geissler & Altmann, 2015). This is relevant particularly for

countries like Norway, where only one weather file for the entire country is used in the energy labelling system.

The reason of harmonizing the climate of Norway by using Oslo data could not be found by the authors. However, a study performed under the TRANSES in 2009 states that it can be estimated that about 76% of dwellings are located in areas with a climate comparable to or milder than the Oslo climate. Hence, the Oslo climate could be adopted as an average climate valid for the entire building stock (Sartori, Wachenfeldt, & Hestnes, 2009). However, Norway can be classified in seven different climate zones based on the Degree-Days method (Enova, 2006), where large cities can be identified in each zone.

1.4 Methodology

The study is presented in three bodies. First, an energy-labelling tool is used to estimate the energy demand of the building. The results are used to calibrate a model in EnergyPlus (e+). Second step is to run the e+ model with the weather data sets for the seven climate zones found for Norway, Alta, Bergen, Fagernes, Mo i Rana, Oslo, Trondheim and Tromso. From this, energy consumption and rating are compared. Third step shows how the weather file influences the energy consumption when upgrading the building to passive house level. Three different types of weather data, IWEC, Meteonorm and the standardized Norwegian weather file are applied and their influence on the results is compared. It is noticed that ASHRAE does not have data for zone 5 (Mo i Rana). Due to this, the file from Meteonorm is used, since a comparison between both sources was done, showing similar results with the other cities (Section 2.1).

1.5 Cases and materials

The case used for the study is Figure 1 shows the building considered in this investigation. It is conceived as a module of a wooden terraced dwelling from 1994 located in Oslo. The building code from 1987 was the current one at that time.

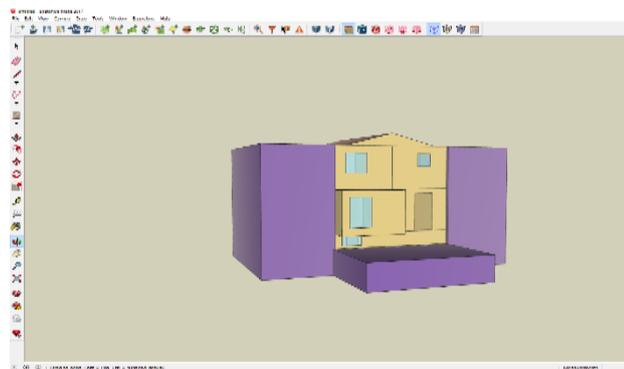


Figure 1. Building model used in the simulations. The model was constructed in Open Studio, and simulated with e+.

The dwelling is 146 m², distributed on three levels. Some key properties of the dwelling are presented in Table 1. The U-values are realistic values, ventilation rate is according to NS 3031, while air leakage is the minimum requirement in the 1987 building code.

Table 1. Some key properties of the dwelling.

Component	Bases level	Passive house level
U-values [W/(m²K)]		
External walls	0.28	0.15
Ceiling	0.32	0.10
Expose floors	0.17	0.10
Basement walls	0.3	0.15
Windows/doors	1.9	0.8
Ventilation [m³/h]	1.2, exhaust	1.2, exhaust
Air leakage at 50 Pa [ACH]	4.0	2.8*

*airtightness value does not follow the passive standard since this is a retrofitting case, the air leakage was set according to literature (Gillott et al., 2016).

A second round of simulation were performed after upgrading the initial case to passive house standard (NS 3700). The input data are also presented in Table 1.

The software used for the study is Simien, which is the most common simulation tool for such calculations in Norway. The tool has the Norwegian energy labelling regulation, which is based on NS 3031, integrated. E+ is used to compare the different weather files. The building model, which is shown in Figure 1, is created in Open Studio. According to IBPSA, e+ is the most common building simulation tool worldwide.

1.6 Weather data analysis

In order to evaluate the pertinence of the climatic zones proposed by ENOVA, the weather data set (IWEC) for each city are compared based on Heating Degree Days (HDD). Table 2 provides the geo-references of each city, while ENOVA weather classification and HDD are presented in Figure 2. The IWEC weather file contains real weather data, while the Meteonorm data are synthetic. Therefore, the IWEC data are used for this task (except for Mo i Rana as explained above). Figure 2 shows an average difference of 667 HDD between the cities. The minimum difference was 299 HDD between Trondheim and Oslo, while the maximum difference was 3999 HDD between Alta and Bergen. The variations in temperature and solar radiation are presented in Figure 3. The assessment of the weather data shows significant variation in HDD and solar radiation between the selected cities. This reflects that the weather data selected for the assessment are pertinent for the purpose of the study. Since they clearly represent climatic singularities.

Table 2. Geo-references of the cities

Climate Zone	City	Latitude	Longitude	Altitude
1	Oslo	59.90	10.62	17
2	Bergen	60.38	5.33	36
3	Fagernes	60.98	9.23	365
4	Trondheim	63.47	10.93	17
5	Mo i Rana	66.31	14.14	26
6	Tromso	69.68	18.92	10
7	Alta	69.98	23.37	3

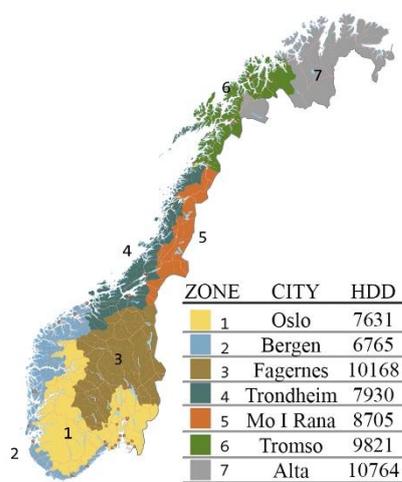


Figure 2. Map of the Climate zones, based on the ENOVA classification, with the HDD from the IWEC file for the selected cities (for Mo i Rana, Meteororm file was used).

2 Results

2.1 Energy consumption comparisons

The energy consumption obtained by using IWEC and Meteororm files is presented in Table 3 and Figure 4. Figure 4 provides the results in case of using the official Norwegian weather data file as well.

Comparing the weather data, an average difference of 3,7% between IWEC and Meteororm is observed, while Meteororm data occurs a lower energy demand for all the cities. Moreover, important differences between the official Norwegian weather file and the other data files, even for zone 1, are also displayed. These differences are attributed to considerable deviations in terms of global solar radiation and dry temperature that exist between the weather files.

Considering IWEC weather data file and taking into account Oslo (zone 1) as a reference, the highest deviation is 60 kWh/m² per year, represents 31% increase and is found in Alta (zone 7). Only zone 2, represented by Bergen, shows a lower energy demand than Oslo; 10.5 kWh/m² less per year, which is a difference of around 7%. However, the step variation between the zones ranges has an average difference of

7.5 (without considering Mo i Rana). This indicates that the distances in terms of energy consumption between the zones may be too close, however other aspects different than heating needs may be the determinant factor for its differentiation, such as thermal oscillation, solar radiation etc. Moreover, the behaviour per season are notoriously different, which indicate that different strategies might be needed particularly in summer.

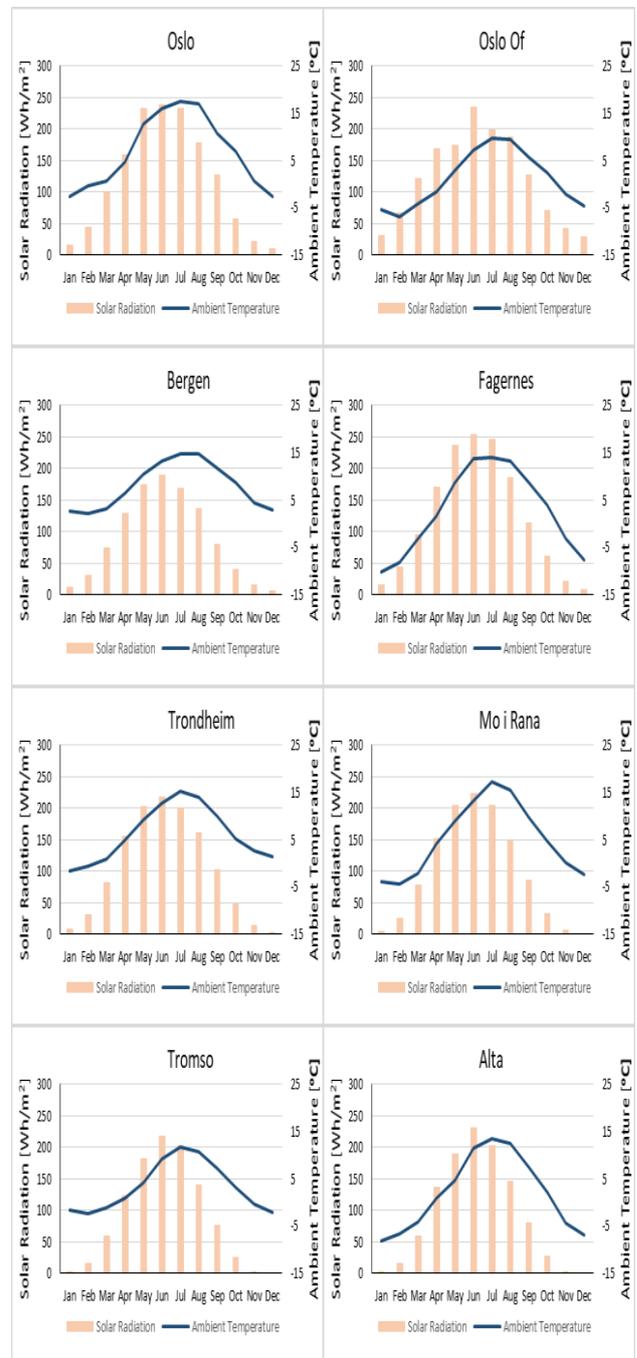


Figure 3. Climate graphs of the cities in the study. The data shows that they are important variations, mainly in temperature and solar radiation, which both have a great impact on the energy demand.

Table 3. Annual energy consumption for the various climate zones with IWEC and Meteororm weather data.

Climate Zone	IWEC	Meteororm	Deviation
	[kWh/m ²]	[kWh/m ²]	[%]
1. Oslo	149.4	136.8	8.4
2. Bergen	138.9	126.5	8.9
3. Fagernes	185.8	180.9	2.6
4. Trondheim	156.5	143.9	8.0
5. Mo i Rana	-	182.8	-
6. Tromso	191.7	187.6	2.1
7. Alta	209.3	200.1	4.4
Average	172.0	165.6	3.7

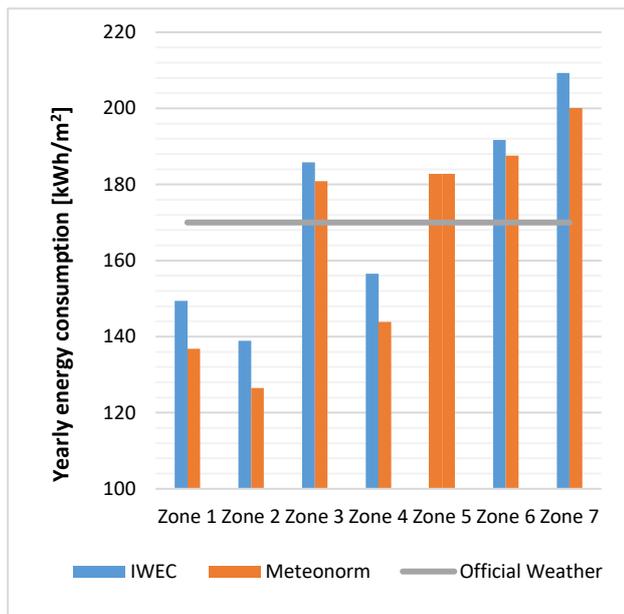


Figure 4. Energy consumption obtained by using the different weather data sources.

2.2 Monthly analysis

The results from the simulation with the IWEC database, showed that in terms of heating demands there is not a harmonized map over the different zones. In order to show these differences more clearly, the surface plot in Figure 5 and 6 was designed. The cities are arranged from the highest to the lowest energy demand on one axis, and the same is done with the months on the second axis. The range was set up based on the minimum average differences between the seasons. Thus, four prominent areas cover most of the surface plot. A fifth area (dark blue) with the highest energy demand can be seen for Alta, Fagernes and Mo i Rana. In spite that Tromso has the second highest annual energy demand, it does not reach the dark blue area. This is because the heating demand during summer is much higher than in the other zones. As a reference of the current labelling system in Norway, Oslo was highlighted with a line in red, to easily compare its energy consumption pattern with the rest of the zones. In winter (yellow and dark blue area) Bergen has the

shortest period of energy demand, followed by Oslo. The rest of the cities have a common pattern, with a winter behavior period of five months, six for Alta. During the warmest months (Jun-Aug-May) most of the zones have similar heating demands (0 to 6 kWh/m² per month, light blue area), except Bergen which has almost five months with low energy demand. Due to low solar radiation, Tromso has no summer period, and Alta has only two months. It could also be noted that Oslo has the most regular distribution of energy demand over all the months.

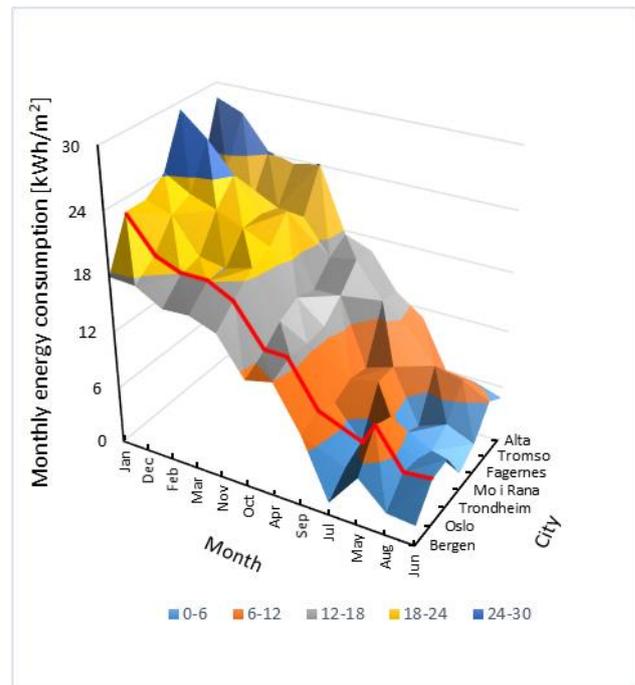


Figure 5. Monthly energy consumption in different Norwegian cities for the base case scenario in descending order.

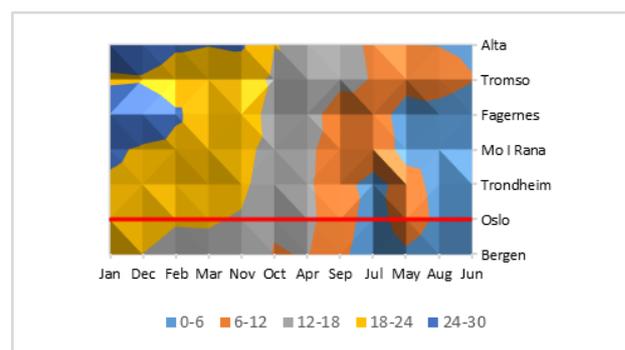


Figure 6. Monthly energy consumption in different Norwegian cities for the base case scenario in descending order-Front view.

The second round of simulation, with the improvements, is shown in Figure 7 and 8. It shows a more regular map, but the same trend as in the first round is visible. Bergen is the most unique zone, with the lowest energy demand, follow by Oslo and Trondheim with similar distribution of energy consumption through the months. Mo i Rana,

Fagernes, Tromso and Alta is a third group, where the length of the heating periods is similar. Fagernes and Alta had the highest peak, incorporating an extra zone during January.

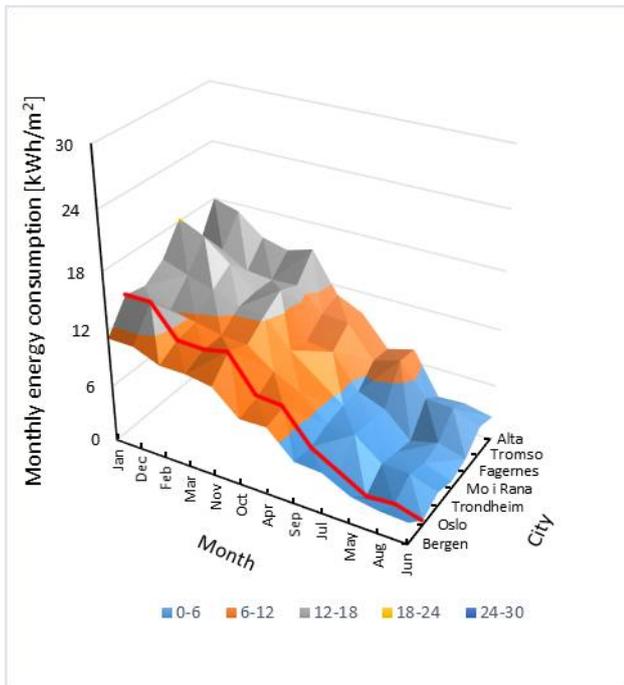


Figure 7. Monthly energy consumption in different Norwegian cities for the Norwegian passive house standard scenario in descending order.

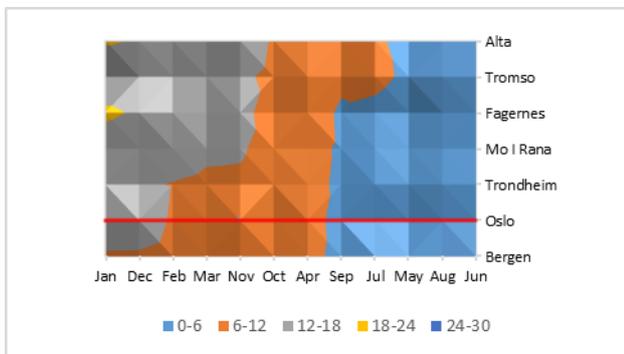


Figure 8. Monthly energy consumption in different Norwegian cities for the Norwegian passive house standard scenario in descending order-Front view.

2.3 Energy labelling assessment

An energy labelling simulation of the base case was performed with the Norwegian tool (and the standardized Norwegian weather file), which gave a D. The base case was also simulated with the IWEC and Meteororm weather files. The results of the energy labelling simulations for each city are presented in Table 4, and shows that the same label is achieved with both IWEC and Meteororm weather data. However, the label for Oslo, Bergen and Trondheim was upgraded to a C, while the rest of the cities maintained the same labelling as achieved with the official Norwegian weather file. However, the steps for each letter from A to D is 38

kWh/m², the cities that had a score of C have an average improvement of 26 kWh/m² from the D level, and these indicates that cities are not near the limit values between ranges, as can be seen in Figure 9.

Table 4. Labelling result for the base case.

Climate Zone	Official Weather	IWEC	Meteororm
1. Oslo	D	C	C
2. Bergen	D	C	C
3. Fagernes	D	D	D
4. Trondheim	D	C	C
5. Mo i Rana	D		D
6. Tromso	D	D	D
7. Alta	D	D	D

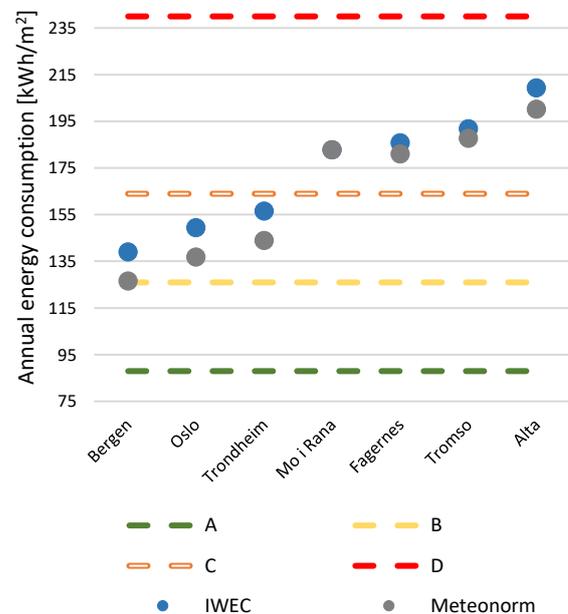


Figure 9. Results from the labelling under the bases cases scenario.

The scenario with the improvements (passive standard level) using the official Norwegian weather data file, gave the dwelling a B level. The dwelling was only 4.32 kWh/m² per year from an A, however, which is relatively close (the range of the B level is 38 kWh/m²). The same case using IWEC weather data gave an A in Bergen and a C in Alta, while the rest of the cities reached a B. Alta was only 1 kWh/m² per year from a B, though. The labels achieved by the Meteororm weather data did not show the same concordance as the ones with IWEC data. This is because most of the results are close to the edge of the range of each letter. In spite of this, it is proved that

by simulating the energy consumption of each city with local climate data they perform differently from Oslo.

Table 5. Labelling results for the passive house level.

Climate Zone	Official Weather	IWEC	Meteonorm
1. Oslo	B	B	A
2. Bergen	B	A	A
3. Fagernes	B	B	B
4. Trondheim	B	B	A
5. Mo i Rana	B	B	B
6. Tromso	B	B	B
7. Alta	B	C	B

2.4 Discussion

Norway has a long tradition with energy performance regulations, and new buildings have already a high standard, one of the strictest in EU with this, housing envelope are prepared to efficiently isolate weather conditions. The good quality of the construction indicate that the main measures to reduce energy are not the properties of the envelope but the mechanical heating and ventilation systems. Nevertheless, the building envelope still play a role to reduce the energy losses. The results from this study clearly shows that the weather data file has an impact on the labelling, even if the building in question is on passive house level. These outcomes should be considered in the purposes of the certification system, since the aim of the labelling tools is to inform stakeholders about the performance of the property and with this enable them to compare different options. According to the presented results, however, this may not be the case under a single weather data file scheme. The study was conducted with a terraced dwelling, attached to adjacent dwellings on two sides. This leaves an exposed facade area of 82 m², which represents 16 % of the total envelope. This makes the energy demand of the dwelling less sensitive to weather than e.g. a detached dwelling. Thus, the study has a limitation in order to be representative for various kinds of dwellings. In this respect, future work should focus on proposing climate zones for labelling purposes in Norway, which can be used for policymaking as well as for improving energy saving estimations. In addition, overheating might be an issue in cold climates where buildings are built according to passive house standards. However, this was not confirmed in this study.

3 Conclusion

The analysis of the IWEC and Meteonorm weather data shows a correlation in energy consumption, which means that both can be used for energy labelling. Since IWEC represents real weather data, further analysis of

the climate was performed. It became clear that the zone designation is somehow appropriate if the HDD are used as basis for the classification. However, if the energy consumption is used for the assessment of the zones, short steps can be seen between some zones, e.g. zone 5 (Mo i Rana) and 3 (Fagernes) with only 1% difference, zone 6 (Tromso) and 3 (Fagernes) with 3% and zone 4 (Trondheim) and 1 (Oslo) with 4,5%. Despite that this results might show that the zones can be merged, there is reason to think that cooling need might also influence, increasing the distance of the simulation results, but under a single zone approach this is hardly evident.

Labelling rating is the most important piece of evidence from the article to state that local weather data should be used, especially if renovation measures in order to target high level rating are considered. As it was mentioned before, the envelope insulation level is not the main source of energy reductions, however it plays an important role after the heating system is improved. The energy rating used in this study is mainly based on the envelope upgrade, following the Norwegian Passive standard. The conclusion from that task shows that in order to achieve a higher level of energy rating, different level of renovations should be considered according to climatic zone.

References

- Al-Mofeez I. A., Numan M. Y., Alshaibani K. A., and Al-Maziad, F. A. Review of typical vs. synthesized energy modeling weather files. *Journal of Renewable and Sustainable energy*, 4(1), 2012.
- Backhaus J., Tigchelaar, C., de Best-Waldhober and M. Key findings & policy recommendations to improve effectiveness of Energy Performance Certificates & the Energy Performance of Buildings Directive. *Retrieved from*, 2011.
- Cappelletti F., Dalla Mora T., Peron F., Romagnoni P., and Ruggeri P. Building renovation: which kind of guidelines could be proposed for policy makers and professional owners? *Energy Procedia*, 78, 2366-2371, 2015.
- Crawley D. B. Which weather data should you use for energy simulations of commercial buildings?/Discussion. *Ashrae Transactions*, 104, 1998.
- Enova. Series of Bygningsnettverkets energistatistikk: 2002, 2003, 2004, 2005, 2006. *Retrieved from Trondheim*, 2006.
- Erba S., Causone F. and Armani R. The effect of weather datasets on building energy simulation outputs. *Energy Procedia*, 134, 545-554, 2017.
- Fumo N. A review on the basics of building energy estimation. *Renewable and Sustainable Energy Reviews*, 31, 53-60, 2014.
- Geissler S. and Altmann N. Certification. Overview and outcomes. In *2016-Implementing the Energy Performance of Buildings Directive (EPBD) – Featuring Country Reports. Concerted action EPBD*, 2015.
- Gherri B. *Assessment of Daylight Performance in Buildings: Methods and Design Strategies*: WIT Press. 2015.
- Gillott M., Loveday D. L., White J., Wood C., Chmutina K. and Vadodaria K. Improving the airtightness in an existing UK dwelling: The challenges, the measures and their effectiveness. *Building and Environment*, 95, 227-239, 2016.

- Guan L. Preparation of future weather data to study the impact of climate change on buildings. *Building and Environment*, 44(4), 793-800, 2009.
- Hensen, J. L. Simulation of building energy and indoor environmental quality-some weather data issues. *Paper presented at the Proc. Int. Workshop on Climate data and their applications in engineering*, 1999.
- Herrera M., Natarajan S., Coley D. A., Kershaw T., Ramallo-González A. P., Eames M. and Wood, M. A review of current and future weather data for building simulation. *Building Services Engineering Research and Technology*, 38(5), 602-627, 2017.
- Hinge A., Neely A. C. B. and Taylor C. Building energy rating schemes around the world: What do we know. *Proceedings of the ACEEE*, 2014.
- Hong T., Chang W.-K. and Lin H.-W. A fresh look at weather impact on peak electricity demand and energy use of buildings using 30-year actual weather data. *Applied Energy*, 111, 333-350, 2013.
- Huang J. ASHRAE Research Project 1477-RP-Development of over 2,500 weather files for international locations. *Final Report. ASHRAE*, 2010.
- Huang Y. J. Development of 3012 IWEC2 weather files for international locations (RP-1477). *Ashrae Transactions*, 120, 2014.
- Hui S. C. and Cheung K. Climatic data for building energy design in Hong Kong and Mainland China. *Proceedings of the CIBSE National Conference 1997*, 1997.
- Institute for Market Transformation (IMT). A roadmap for creating building energy rating systems in central Asia, 2009.
- International Energy Agency (IEA). Energy Performance Certification of Buildings: A policy tool to improve energy efficiency. Retrieved from https://www.iea.org/publications/freepublications/publication/buildings_certification.pdf, 2010.
- Lund H. The design reference year. *3rd International IBPSA Conference*, 1991.
- Lund H. and Eidorff S. Selection Methods for Production of Test Reference Years: Final Report, Short Version: *Commission of the European Communities*, 1981.
- Pernigotto G., Prada A., Cappelletti F. and Gasparella, A. Impact of Reference Years on the Outcome of Multi-Objective Optimization for Building Energy Refurbishment. *Energies*, 10(11), 2017.
- Polly B., Kruis N. and Roberts D. Assessing and improving the accuracy of energy analysis for residential buildings. 2011.
- Remund J., Müller S., Schilter C. and Rihm, B. The use of Meteororm weather generator for climate change studies. *10th European Conference on Applications of Meteorology (ECAM) Abstracts, held Sept. 13-17, 2010 in Zürich, Switzerland*. <http://meetings.copernicus.org/ems2010/>, id. EMS2010-417., 2010.
- Roetzel A. and Tsangrassoulis A. Climate change sensitivity of comfort and energy performance criteria for offices. *ANZAScA 2011: From principles to practice in architectural science: Proceedings of the 45th Annual conference of the Australian and New Zealand Architectural Science Association*, 2011.
- Sartori I., Wachenfeldt B. J. and Hestnes A. G. Energy demand in the Norwegian building stock: Scenarios on potential reduction. *Energy Policy*, 37(5), 1614-1627, 2009.
- Tasktroup I. B. E. E. Building Energy Rating Schemes: Assessing Issues and Impacts. *International Partnership for Energy Efficiency Cooperation*, 2014.
- Thevenard D. J. and Brunger A. P. The development of typical weather years for international locations: part I, algorithms. *Ashrae Transactions*, 108, 2002.