

Nonlinear Model Predictive Control of a Thermal Management System for Electrified Vehicles using FMI

Torben Fischer¹ Tom Kraus² Christian Kirches² Frank Gauterin³

¹Fraunhofer Institute for Chemical Technology (ICT), Project Group New Drive Systems, Germany, torben.fischer@ict.fraunhofer.de

²Interdisciplinary Center for Scientific Computing (IWR), Heidelberg University, Germany, {tom.kraus,christian.kirches}@iwr.uni-heidelberg.de

³Institute of Vehicle System Technology, Karlsruhe Institute of Technology (KIT), Germany, frank.gauterin@kit.edu

Abstract

Energy-efficient thermal management systems for E-mobility help to decrease energy consumption and increase range. Due to transient external conditions and the increasing system complexity, optimization-based control approaches are required in order to harness the full potential of such systems. In (Fischer et al., 11th Int. Modelica Conf, 2015), we have presented a model-based development cycle for a thermal management system in E-mobility to this end. In this article, we build upon this work to describe the use of this model within a nonlinear model predictive control (NMPC) approach. The main benefits of using an advanced optimization-based control system in this application are a) the ability to control the battery temperature and the cabin temperature simultaneously, b) the increased energy efficiency achieved by exploiting the predictive character of the optimization-based control approach, c) the possibility to include operational limits as constraints in the optimization problems and d) the fast reaction to disturbances or model parameter changes. We evaluate the merit of the proposed advanced control system by way of a comparison to conventional PID controller.

Keywords: thermal management system, nonlinear model predictive control, Functional Mock-up Interface

1 Introduction

E-mobility is widely considered to be a key concept to achieve ambitious goals set forth in contemporary climate and environmental protection plans. Due to higher costs and lower ranges compared to combustion engine driven vehicles, a breakthrough in the mass market has yet to take place. In this article we propose an optimization-based control for energy-efficient operation of a thermal management system. In (Fischer et al., 2015) we observed a decrease of the energy consumption of up to 30%, depending on ambient conditions. To improve the system further, a nonlinear model predictive control (NMPC) approach is proposed with the aim to harness the full potential of the multiple-input multiple-output system (MIMO).

This article constitutes a follow-up of (Fischer et al.,

2015), where the concept of the thermal management system is presented, including simulation results. The remainder of this article is structured as follows: §1 introduces the subject and describes the related state-of-the-art. §2 recalls the process model of the thermal management system. A short discussion of the NMPC approach resides in §3 covering the formulation as a mathematical optimization problem, the multiple-shooting discretization, a real-time solution algorithm and the employed software interface. In §4 process model modifications are described which were necessary in order to employ derivative-based optimization techniques. §5 contains an "offline" case study to compare different approaches of jacobian matrix generation on the basis of the Karush-Kuhn-Tucker (KKT) violation and an "online" case study to compare NMPC to conventional PI control. Finally, we provide conclusions and an outlook on future topics in §6.

1.1 State of the Art

NMPC is widely used in, e.g., process control and chemical engineering, often with rather slow sampling rates. In the past years, the automotive industry has also shown an increased interest in model predictive control. Applications like adaptive cruise control (Kirches, 2011; Kirches et al., 2013), lateral dynamic stabilization, etc., can be typically controlled by a MPC-controller. Further examples may be found in, e.g., (del Re et al., 2010). In the area of heating, ventilation, and air conditioning (HVAC), conventional control methods like PID-controllers and bang-bang-controllers are still state-of-the-art, mostly due to simplicity of design and implementation. There are, as well, investigations on advanced control systems. For example, (Esen et al., 2014) and (Karnik et al., 2016) use MPC-controllers in the application field of thermal management systems, and (Afram and Janabi-Sharifi, 2014) gives an general overview for HVAC systems. To reduce the computational effort, these approaches however often do not rely on first-principles models, but rather on data models or linearized state-space representations. The first publication of an NMPC-controller based on first-principle models using Modelica is (Franke, 2002). In (Gräber et al., 2012), a functional mock-up unit (FMU)

of a first-principle Modelica model was used for the first time within an fully nonlinear MPC setting. This example is closely related to the present article, as it treats a compression-vapor cycle.

2 Thermal Management System

In this section, we review the layout of a thermal management system as introduced in (Fischer et al., 2015). The thermal management system of a vehicle has multiple tasks. Primarily, for passenger comfort heating or cooling of the passenger cabin is required. Moreover, there are a number of legal requirements to be met including windshield defrosting and defogging. Finally, powertrain components have to be kept within their thermal operational range. In the particular case of an electrified vehicle, increasing demands due to thermally even more sensitive components and significant less amount of waste heat lead to a development of new thermal management systems.

2.1 Concept and Model

The main feature of the system, depicted in Fig. 1, is a reversible heat pump, called thermal module (1), which provides heating and cooling power to keep the components within a thermal operating range. Waste heat emitted by electric components (2) is used to increase the temperature level of the heat source, thereby contributing to a higher efficiency of the system. By way of a flexible interconnection, independent thermal conditioning of the cabin (3) and the battery (4) may be achieved. By using waste heat from an optional energy converter like a fuel cell or an internal combustion engine (5) in a hybrid electric vehicle, the thermal module can be switched off.

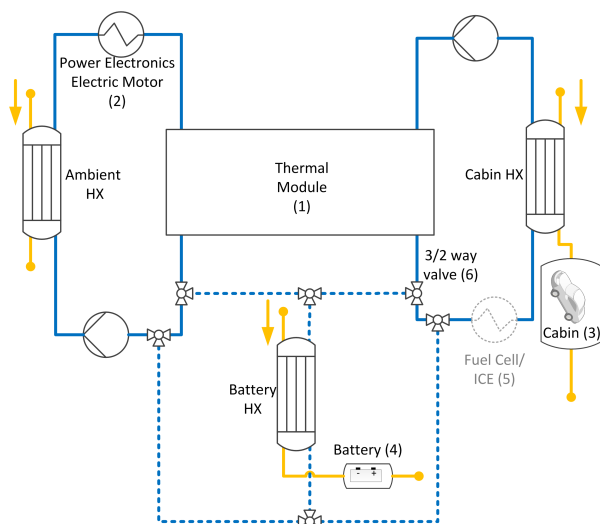


Figure 1. Scheme of the thermal management system

The heat exchangers in the refrigerant cycle are modeled in distinct ways. The condenser is modeled using a moving boundary approach while the evaporator is realized using a finite volume method. Details can be found in §4.2. The heat exchangers modeling the heat flow between

coolant and air are also realized using a finite volume method. The refrigerant accumulator, the coolant reservoirs, and the passenger cabin are modeled using lumped volumes. The fluid data is taken from the TILMedia library, which provides bi-cubic spline interpolants for the used refrigerant. The coolant is modeled as pure water and the ambient air as dry air.

2.2 Controlled Variables

The controlled variables in the system shown in Fig. 1 are the cabin temperature T_{cabin} , the superheating temperature of the heat pump T_{sh} and also the battery temperature T_{battery} . The desired cabin temperature T_{cabin} is assumed to be at 22°C according to passenger preference. To ensure a safe and efficient operating mode, the superheating temperature of the heat pump T_{sh} is desirable to be at 5 K. The admissible thermal operating interval of the battery is from 20°C to 40°C. A temperature above this range can lead to increased aging effects and eventually to degradation and a thermal runaway. The operating temperatures of the electric motor and the power electronics are usually observed but not tracked to a preset value.

2.3 Manipulated Variables

One manipulated variable in the system is the rate of change u_{compr} of the compressor frequency of the heat pump, which in a PID framework would be controlled according to the cabin temperature setpoint. The second input is the rate of change u_{valve} of the expansion valve which in a PID logic would be assigned to the superheating temperature setpoint. As the system contains two thermally conditioned components, the output heat rate is split up by a 3/2-way-valve ((6) in Fig. 1) in the coolant by dividing the mass flow. The valve is a linear control valve and accepts continuous values between 0 (path to battery closed) and 1 (path to battery open).

2.4 Control Approach

In (Fischer et al., 2015), we have described a first control approach, namely a decoupling of the MIMO-system into multiple SISO-systems, which can then be controlled individually and by separate PID controllers. This approach turned out to be problematic, as the battery temperature and thus the cabin comfort was affected by a starting thermal conditioning of the battery. The influences of the PID controllers on each other also led to inefficient overshooting behavior and oscillations. As the main control goal is energy-efficiency in order to allow for maximum electric range of the vehicle, we propose an NMPC approach to control the system.

In a first step, a reduced model of the thermal management system without battery and attached vehicle model is used to compare the control schemes in the computational studies in §5. This is appropriate since it is still convenient to control this reduced system by PI-controllers which serves as good reference for the NMPC-controller.

3 Nonlinear Model Predictive Control

In this section, we briefly review Nonlinear Model Predictive Control (NMPC) as an optimization-based scheme for advanced closed-loop feedback control of dynamic processes.

3.1 NMPC Problem Formulation

A suitable mathematical model description, capable to predict the future behavior of the dynamic process under consideration, must be available for the realization of any NMPC controller. For the system at hand, a nonlinear differential-algebraic system of equations (DAE) is suitable. The objective function to be minimized is typically assumed to be of “tracking type”, i.e. set-points are provided for the controlled variables. In addition, NMPC permits to include nonlinear constraints on process quantities. The optimization problem to be solved in order to find optimal controls may then be formulated as a DAE-constrained optimal control problem and reads

$$\begin{aligned} \min_{x,z,u} & \left\| \int_0^T \ell(x(\tau), z(\tau), u(\tau), p) d\tau \right\|_{2,Q}^2 + \|e(x(T), p)\|_{2,W}^2 \\ \text{s.t.} & \quad \dot{x}(\tau) = f(x(\tau), z(\tau), u(\tau), p) \quad \tau \in [0, T] \quad (1a) \\ & \quad 0 = g(x(\tau), z(\tau), u(\tau), p) \quad \tau \in [0, T] \quad (1b) \\ & \quad x(0) = \hat{x}_0(t) \quad \tau \in [0, T] \quad (1c) \\ & \quad 0 \leq c(x(\tau), z(\tau), u(\tau), p) \quad \tau \in [0, T] \quad (1d) \\ & \quad 0 \leq r_i(x(\tau_i), z(\tau_i), p) \quad \{\tau_i\} \subset [0, T] \quad (1e) \end{aligned}$$

Herein, an objective function of least-squares type on the prediction horizon $[0, T]$ is composed of an integral term ℓ with weight matrix Q and an end-point term e with weight matrix W , and tracks set-points provided for certain controlled variables. The problem is constrained by the DAE model equations (1a, 1b), by inequality path constraints on dynamic states and controls (1d), and by point constraints on a grid $\{\tau_i\} \subset [0, T]$ that may cover, for example, boundary or periodicity conditions.

By way of constraint (1c), the current process state $\hat{x}_0(t)$ at physical time t is embedded into the dynamic optimization problem, and must hence be available as a measurement or be provided by an observer. If the process model is sufficiently accurate and the formulation of the optimization problem is suitable, its solution yields optimal process inputs $u^*(\tau)$ on $\tau \in [0, T]$ where $\tau = 0$ coincides with the physical time t at which the observation was taken.

In practice, however, model predictions must be assumed inaccurate because of inevitable measurement or actuation errors as well as due to systematic model errors. Naturally, this effect becomes more apparent over longer time horizons T . For this reason, u^* , computed from the initial state $\hat{x}_0(t)$, is applied to the process for a “short” time only. The natural choice for the length of this “short” time interval is the system’s sampling time.

The optimization procedure is continuously repeated, each time a new measurement is available. This makes

NMPC a true closed-loop control scheme. Fig. 2 visualizes this control concept.

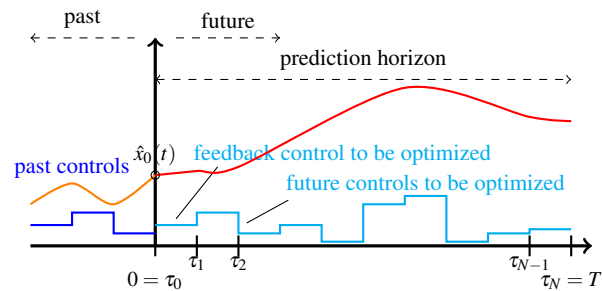


Figure 2. The Nonlinear Model Predictive Control paradigm for a piecewise constant control subject to optimization on the prediction horizon.

3.2 Direct Multiple Shooting

In order to computationally solve problem (1) efficiently, a parameterization of the control u and a discretization of the states x and z in time is necessary in a direct approach to optimal control. With the direct multiple shooting method (Bock and Plitt, 1984), we employ a simultaneous approach.

To this end, the control u is parameterized by piecewise constant control parameters q on a discretization grid $0 = \tau_0 < \tau_1 < \dots < \tau_{N-1} < \tau_N = T$,

$$u(\tau) := q_i \in \mathbb{R}^{n_u}, \quad \tau \in (\tau_i, \tau_{i+1}), \quad 0 \leq i \leq N-1.$$

On each control interval $[\tau_i, \tau_{i+1}]$, a separate DAE initial-value problem (IVP)

$$\dot{x}(\tau) = f(x(\tau), z(\tau), q_i, p), \quad \tau \in [\tau_i, \tau_{i+1}] \quad (2a)$$

$$0 = g(x(\tau), z(\tau), q_i, p) - \theta_i(\tau)g(s_i, z_i, q_i, p) \quad (2b)$$

$$x(\tau_i) = s_i, \quad z(\tau_i) = z_i \quad (2c)$$

is solved, given the initial value $s_i \in \mathbb{R}^{n_x}$.

The DAE condition is relaxed using a function $\theta_i(\tau)$ that is monotonically strictly decreasing on $[\tau_i, \tau_{i+1}]$ and that satisfies $\theta(\tau_i) = 1$ and $\theta(\tau_{i+1}) = 0$. This relieves us of having to find consistent initial values $z_i \in \mathbb{R}^{n_z}$ for solving the IVP. Consistency in the optimal solution will be ensured by requiring

$$0 = g(s_i, z_i, q_i, p), \quad 0 \leq i \leq N.$$

Continuity of the IVP solutions thusly obtained is enforced by additional matching conditions,

$$0 = x(\tau_{i+1}; \tau_i, s_i, z_i, q_i, p) - s_{i+1}, \quad 0 \leq i \leq N-1,$$

wherein $x(\tau_{i+1}; \tau_i, s_i, z_i, q_i, p)$ denotes the solution of the i -th IVP on $[\tau_i, \tau_{i+1}]$ and for initial values s_i and z_i . Finally, path and point constraints (1d, 1e) are enforced in the time grid points τ_i only. The integral least-squares objective function is evaluated along the solution of (2).

The nonlinear programming problem (NLP) resulting from this discretization and parameterization reads

$$\min_{s,z,q} \Phi := \sum_{i=0}^{N-1} \|L_i(s_i, z_i, q_i)\|_{2,Q}^2 + \|e(s_N, z_N)\|_{2,W}^2 \quad (3a)$$

$$\text{s.t. } 0 = x(\tau_{i+1}; \tau_i, s_i, z_i, q_i) - s_{i+1}, \quad 0 \leq i \leq N-1 \quad (3b)$$

$$0 = g(s_i, z_i, q_i) \quad 0 \leq i \leq N \quad (3c)$$

$$0 = s_0 - \hat{x}_0(t) \quad (3d)$$

$$0 \leq r_i(s_i, z_i, q_i) \quad 0 \leq i \leq N. \quad (3e)$$

Herein, L_i is an appropriate quadrature rule for ℓ on $[\tau_i, \tau_{i+1}]$, r_i summarizes the path and point constraints (1d,1e) in τ_i , $q_N := q_{N-1}$, and the dependencies on p have been omitted. For future reference, we denote by $c(\cdot)$ the set of equality constraints (3b, 3c, 3d) and $d(\cdot)$ refers to (3e).

This highly structured nonlinear programming problem is solved by a tailored sequential quadratic programming (SQP) method as described in (Bock and Plitt, 1984; Leineweber et al., 2003). For NMPC, a single iteration k of this method may be divided into three distinct phases according to the real-time iteration scheme first proposed in (Diehl, 2001), as follows:

1. **Prepare:** In the k -th SQP iterate $w^{(k)} = (s^{(k)}, z^{(k)}, q^{(k)})$, compute the gradient $b^{(k)}$ of the objective and the Jacobian $J^{(k)}$ of the least-squares objective residuals of (3), evaluate the (in-)equality constraint residuals $c^{(k)}$, $d^{(k)}$, and compute linearizations $C^{(k)}$, $D^{(k)}$ of the (in-)equality constraints.

2. **Feedback:** Obtain a state measurement or estimate $\hat{x}_0(t)$. Solve the quadratic programming problem

$$\begin{aligned} \min_{\Delta w} \quad & \frac{1}{2} \Delta w^T (J^{(k)T} J^{(k)}) \Delta w + b^{(k)T} \Delta w \\ \text{s.t.} \quad & 0 = C^{(k)} \Delta w + c^{(k)} \\ & 0 \leq D^{(k)} \Delta w + d^{(k)} \end{aligned} \quad (\text{QP})$$

to find $\Delta w = (\Delta s, \Delta z, \Delta q)$ and return $u_0^{(k)} + \Delta u_0^{(k)}$ as the new feedback control.

3. **Transition:** Determine a step length $\alpha^{(k)} \in (0, 1]$ by way of a globalization approach, and let $w^{(k+1)} \leftarrow w^{(k)} + \alpha^{(k)} \Delta w$, $k \leftarrow k+1$.

For online optimal control (NMPC), the three phases are continuously repeated as fast as CPU resources permit and state estimates become available. For offline optimal control, the three phases constitute one iteration of an SQP method for nonlinear programming, cf. (Nocedal and Wright, 2006). These are carried out until the termination criterion

$$\|\nabla \mathcal{L}(w^{(k)})\| + \sum_i \lambda_i |c_i(w^{(k)})| + \sum_j \mu_j [d_j(w^{(k)})]^-,$$

referred to as the KKT violation, falls below a preset threshold. Herein, \mathcal{L} denotes the Lagrangian of (3) and

λ , μ denote the most recent Lagrange multipliers of the equality and inequality constraints of (QP), respectively. In the offline case, the embedding of $\hat{x}_0(t)$ is replaced by a fixed initial value.

3.3 Software Interfaces

A state of the art software package that implements the numerical algorithm just presented is MUSCOD, see (Bock and Plitt, 1984; Leineweber et al., 2003). The DAE initial value problems (2) are solved by DAESOL, cf. (Bauer et al., 1999; Albersmeyer, 2010).

The developed Modelica model has to be interfaced with the DAE solver of MUSCOD. To this end, the Functional Mockup Interface (FMI) (Blochwitz et al., 2011) is one convenient way to do this. Advantages are easy handling, simulation speed (as the model is provided as a dynamic link library), and the small effort required to export existing Modelica models as Functional Mockup Units (FMU). The interfacing between MUSCOD and the FMU is carried out in C++, which has already been described in detail in (Gräber et al., 2012).

Due to limitations in the current version 2.0 of the FMI standard, only an ODE interface can be exposed to MUSCOD. Hence, in place of the DAE IVPs (2), the ODE IVPs

$$\begin{aligned} \dot{x}(\tau) &= f(x(\tau), g^{-1}(x(\tau), q_i, p), q_i, p), \tau \in [\tau_i, \tau_{i+1}], \\ x(\tau_i) &= s_i \end{aligned}$$

are solved and the local inversion of the algebraic constraint g for the unknown $z(t)$ is internally taken care of by the FMU by way of an iterative nonlinear root-finding method. This situation is unfortunate from the point of view of an all-at-once method for dynamic optimization, as these inner iterations could be carried out much cheaper as part of the solution procedure for the nonlinear programming problem (3). Moreover, the possibility of different outcomes of adaptive choices during finite difference approximation of Jacobians of f may introduce unnecessary approximation errors here. Nonetheless, we have not observed numerical instabilities that could be traced back to this issue.

Offline Optimization. "Offline" optimization is a dynamic optimization for a given initial point, time horizon and number of intervals which does not incorporate any feedback from the real-world plant. The output data comprises the state vector and the optimal control vector at each interval, which can be provided in the simulation within a time table in a further step. A stable and robust offline optimization is essential for the online optimization. Fig. 3 shows the scheme of the offline optimization.

For the "online" optimization an integration of the real-world plant model is needed. The optimized manipulated variables, provided by MUSCOD, have to be applied online to the real-world plant model. This requires that an up-to-date measurement of the real-world plant state is available, since it serves as the initial state for predictions

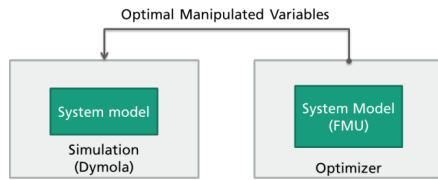


Figure 3. Scheme of the offline optimization

in the control model. Fig. 4 shows the scheme of the online optimization. Basically, there are two paths we follow to interface the model:

Online Optimization with Simulation in Dymola. A Python script is the central control script, which starts the simulation in Dymola, sets the optimal manipulated variable and gets the updated measurement data via the Direct Data Exchange (DDE) Interface. It is challenging though to filter the needed variables, as there is no function in Dymola to get the state vector of a model. The DDE Server of Dymola has to be executed before and the simulation speed must be reduced to real-time.

Online Optimization with Simulation in MUSCOD. In this approach no simulation environment is needed anymore. A Python script starts the optimization, and starts a sequential simulation after each optimization interval. At the end of each simulation part, the states can be extracted out of the FMU and passed to MUSCOD as new initial point. This leads to a very fast result since the simulation is a lot faster than real time. Furthermore, an ideal NMPC can be simulated, where no time is needed to calculate the optimal manipulated variables.

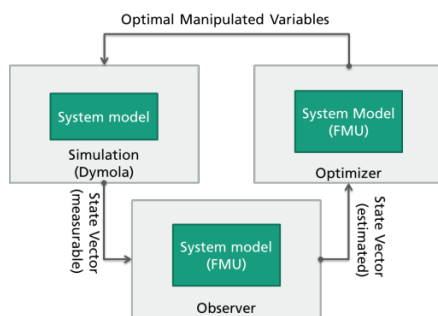


Figure 4. Scheme of the online optimization.

4 Model Adaptations for Optimization

In this section, we report on lessons learned while developing and carrying out adaptation procedures necessary in order to make the existing thermodynamical model fit for optimization based control using gradient-based methods.

4.1 Continuous Differentiable Model

This choice of numerical optimization algorithm and initial value problem solver implies certain smoothness and

regularity assumptions for (1). In particular, ℓ , e , f , g , c , and r_i in (1) need to be twice continuously differentiable w.r.t. all arguments. Furthermore, the algebraic constraint function g needs to be invertible w.r.t. the algebraic state $z(t)$, i.e., the Jacobian $\partial g(x, z, q, p) / \partial z \in \mathbb{R}^{n_z \times n_z}$ has full rank for all applicable values of its arguments.

A first requirement hence is to adapt the existing model to a model that conforms to the requirements set forth. The model must not contain any discontinuities; conditional statements, $\min()$, $\max()$, $\text{abs}()$ -functions and limiters have to be avoided; the use of the `actualStream()`-operator is no longer possible. Occurrences must be replaced by continuous and twice differentiable statements. Fig. 5 shows a discontinuous function, typically used in hybrid simulations, which in this case is replaced by the logistic function with $k = 10$ and $x_0 = 3$,

$$f(x) = (1 + e^{-k(x-x_0)})^{-1}.$$

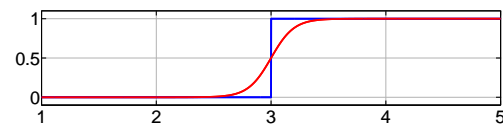


Figure 5. Discontinuous transitions have to be replaced by numerical smooth functions.

4.2 Phase Change in Condenser

The rise of the density when crossing the boiling point curve of the refrigerant in the condenser is discontinuous and leads to problems in the optimizing process. As stated in section 2.1, the heat exchangers are modeled by a finite-volume-method. This means, that the flow path is discretized into N cells. Each cell consists of mass flow and energy conservation. Depending on the operating point or in transient conditions, the crossing of the boiling point curve can occur in different cells, and also within a cell. The discontinuous rise of the density affects the state variable enthalpy, which eventually leads to problems for the optimizer finding a proper gradient.

To solve this problem, the modeling approach of a moving boundary heat exchanger, cf. (Jensen and Tummescheit, 2002) is used. In this case, the flow path is not discretized into N cells, but always into three cells according to the fluid phases: subcooled, two-phase and superheated. Thus, the rise of the density within a cell can be avoided, as one cell is always considered as a homogeneous phase.

However, using a moving boundary heat exchanger leads to another major problem. The model is only valid, if the condenser still contains the three zones superheating, two-phase and subcooled. It is therefore essential to keep the heat pump in an operating state, where all three zones exist. This is achieved by introducing soft constraints punishing operating points of the heat pump which should be avoided. The limits are defined as follows: the length of

the subcooled condenser cell l_{sc} should be greater than 5% of the total length, and the superheating temperature T_{sh} should be greater than 4 K. In case of violating these soft constraints, a high weighted penalty is added to the objective variable, indicating the optimizer to avoid this state. There are some approaches to model a switching moving boundary model, cf. (Bonilla et al., 2015). Since the switching algorithm always relies on boolean logic, it cannot be used with a gradient-based SQP algorithm.

4.3 FMU Status Report

The status of the FMU is returned by each function after calling to indicate the success of the function call. In the case of `fmi2SetContinuousStates` the status can be `fmi2Discard`, indicating that it is recommended to discard the last solution and to evaluate the model equations again with a smaller time step. This information has to be directed to the ODE solver. Ignoring this information can obviously lead to much higher computation time or even to non-convergence.

5 Computational Results

In this section, we report on computational findings for the adapted thermodynamical model both for offline optimal control and in the NMPC context. The results presented in this section focus on the thermal management system without battery and attached vehicle model.

5.1 Problem Setup

The manipulated variables u_{compr} and u_{valve} of the reduced model are the rates of change of the compressor frequency and the area of the expansion valve. The controlled variables are the cabin temperature T_{cabin} and the superheating value of the heat pump T_{sh} . All results are generated with the same basic system model, counting a total number of 43 differential states, and with a time horizon of $h = 20$ s discretized by $N = 20$ shooting intervals.

The sole difference between the model used in the offline and the NMPC study is the incorporation of the ambient temperature as an additional pseudo-dynamic state in the NMPC controller's model ($\dot{T}_{amb} = 0$) while in the offline controller's model the temperature is merely a constant. An incorporation of this easily measureable quantity obviously is the more suited choice since it yields better predictions. However, since required derivative functions for the ambient temperature were not available this was omitted in the offline study for sake of comparability (see also §5.2).

The objective function, defined in Eq. (4), penalizes the differences between controlled variables and their set-points, as well as input changes. Moreover, a soft constraint formulation is chosen to avoid that the superheating temperature T_{sh} drops below the lower operational bound of $T_{sh,LB} = 4$ and that l_{sc} drops below $l_{sc,LB} = 0.05$:

$$\begin{aligned} \max(0, T_{sh,LB} - T_{sh}) &= 0.5 \cdot (T_{sh,LB} - T_{sh} + |T_{sh,LB} - T_{sh}|) \\ \max(0, l_{sc,LB} - l_{sc}) &= 0.5 \cdot (l_{sc,LB} - l_{sc} + |l_{sc,LB} - l_{sc}|) \end{aligned}$$

The final objective function including the two soft constraints reads as follows:

$$\ell(x(t), z(t), u(t), p) = \left(w_i^{-\frac{1}{2}} \Theta_i(\xi_i(t) - \bar{\xi}_i) \right)_{i=1, \dots, 6} \quad (4)$$

wherein $\xi^T = (T_{cabin}, T_{sh}, u_{compr}, u_{valve}, -T_{sh}, -l_{sc})$ and with weights

$$w^T = \left(\frac{10^4}{292}, \frac{10}{5}, \frac{10^{-5}}{72.5}, \frac{10^{-5}}{4.5 \cdot 10^{-7}}, \frac{10^4}{8}, \frac{10^4}{0.1} \right)^T$$

chosen such that denominators normalize quantities to 1 and numerators indicate relative weights. The set-point is

$$\bar{\xi}^T = (295.15, 5, 0, 0, -4, -0.05)^T.$$

The functions Θ_i are $\Theta_i(x) = \text{Id}$ (identity) for $i = 1, \dots, 4$ and $\Theta_i(x) = x \cdot H(x)$ (Heaviside integral) for $i = 5, 6$. To guarantee smoothness assumptions, the Heaviside integral function $x \cdot H(x)$ is exponentially smoothed in computational practice. The end-point term is

$$e(x(T), z(T), p) = \left(w_i^{-\frac{1}{2}} (\xi_i(T) - \bar{\xi}_i) \right)_{i=1, \dots, 2}$$

for $\xi^T = (T_{cabin}, T_{sh})^T$, identical set point, and weights

$$w^T = \left(\frac{10^6}{292}, \frac{50}{5} \right)^T.$$

5.2 Performance Comparison of Different Jacobian Methods

The performance of the online optimizing controller critically depends on the choice of the method to generate Jacobians. Fundamentally, derivatives can be computed by automatic differentiation (AD) or numerically by a finite difference scheme (ND). The FMUs generated by Dymola use numerical Jacobians by default. By setting the flag *Advanced.GenerateAnalyticJacobian*, Dymola can be configured to generate analytic Jacobians and include them in the FMU. For this to be effective, it is necessary that every function used in the Modelica model also declares a corresponding derivative function. For the given system, this required the use of a tailored version of the fluid database TILMedia supplying derivative functions for a wide range of material-dependent functions. MUSCOD can be configured to use Jacobians, which are provided from "outside", e.g. the FMU (numerical or analytical), or to approximate Jacobian matrices numerically by its built-in finite difference scheme.

This section's numerical study compares these three generation methods for Jacobians on the basis of respective offline optimization runs. In the scenario, an instantaneous step change of the ambient temperature of 5 K is applied to a stationary system state (compare also the "online" study at $t = 600$ s). To investigate effects on precision the integration tolerance was set to 10^{-9} .

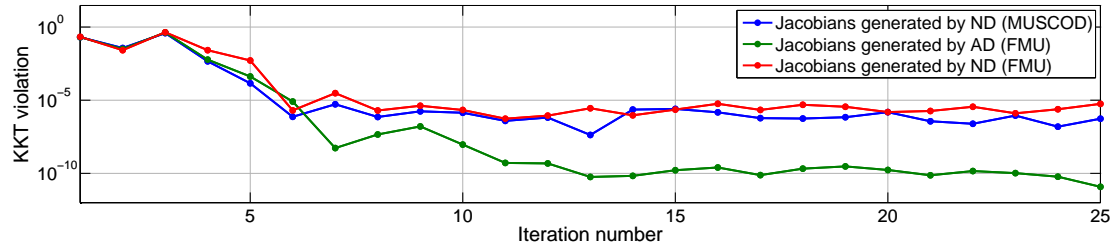


Figure 6. After 25 SQP iterations, the remaining KKT violation during offline optimization is smallest (best) when using AD Jacobians. ND Jacobians provided by MUSCOD are runner-up. ND Jacobians provided by the Modelica FMU perform worst.

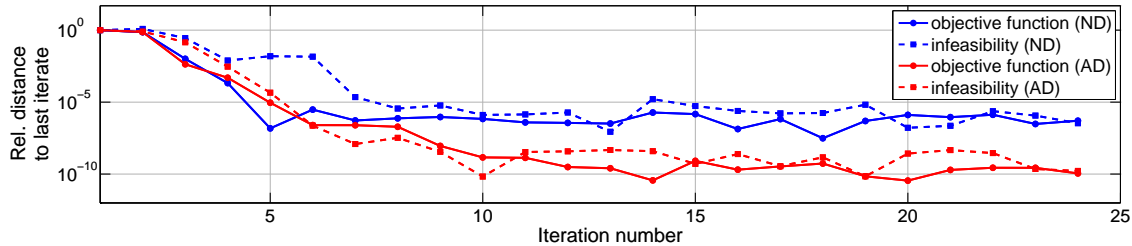


Figure 7. A self-convergence plot of objective function values and infeasibility measures reveals convergence after 10 SQP iterations regardless of the choice of method for generating Jacobians.

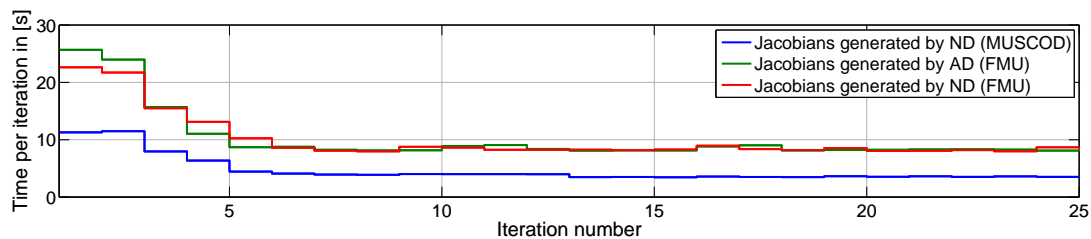


Figure 8. AD Jacobians from the Modelica FMU are computationally more expensive than MUSCOD ND Jacobians. Measured time refers to an integration tolerance of 10^{-9} .

Fig. 6 assesses the impact of the Jacobian generation method on the convergence behavior of MUSCOD in terms of the remaining KKT violation. As expected, AD yields a solution with the highest precision. Fig. 7 shows a self-convergence plot for the objective function value and the infeasibility measure, i.e., we show for iterations $k = 0, \dots, N - 1$ the fractions

$$|\chi_m^{(k)} - \chi_m^{(N)}| / (\chi_m^{(0)} - \chi_m^{(N)})$$

of objective function ($m = 0$) and infeasibility ($m = 1$):

$$\chi_0^{(k)} = \Phi(s^{(k)}, q^{(k)})$$

$$\chi_1^{(k)} = \sum_i [c_i(s^{(k)}, q^{(k)})] + \sum_j [d_j(s^{(k)}, q^{(k)})]^{-}$$

As can be seen, after six SQP iterations convergence has essentially been achieved regardless of the chosen method. As Fig. 8 reveals the MUSCOD internal finite difference scheme is, to our surprise, faster than both of the FMU Jacobian generation methods. Thus, all following numerical results on NMPC were obtained using this Jacobian generation scheme.

5.3 Comparison of PI Control and NMPC

In this section two tuned PI-controllers are compared to the developed NMPC controller. The parameters for the PI-controller were determined using a step response of the system and were manually tuned to a normal and more aggressive behavior. The scenario used for this purpose consists of a transient heat-up, starting from a steady state, at an ambient temperature of 5°C and of a following abrupt ambient temperature change of $+5^\circ\text{C}$, which is applied at $t = 600\text{ s}$ after reaching steady state again, see Fig. 9. The resulting controlled variables are shown in Fig. 10 (cabin temperature) and in Fig. 12 (superheating value). The corresponding manipulated variables are plotted in Fig. 11 (compressor frequency) and in Fig. 13 (expansion valve area). In contrast to §5.2 the integration tolerance is now set to 10^{-5} . Based on experience this is a sufficient value for this application.

Transient Heat-Up of Passenger Cabin We observe, that the NMPC curve rises and settles significantly faster than both PI-controllers and without any temperature overshoot indicating a very efficient control system (cf.

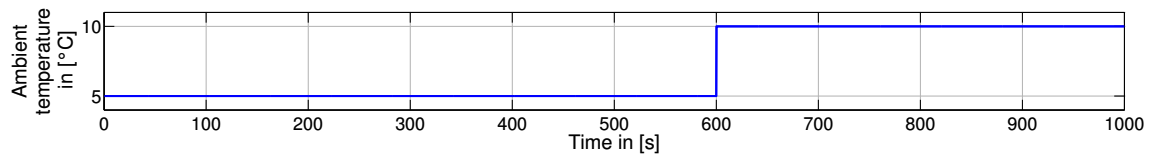


Figure 9. Ambient temperature step of +5K.

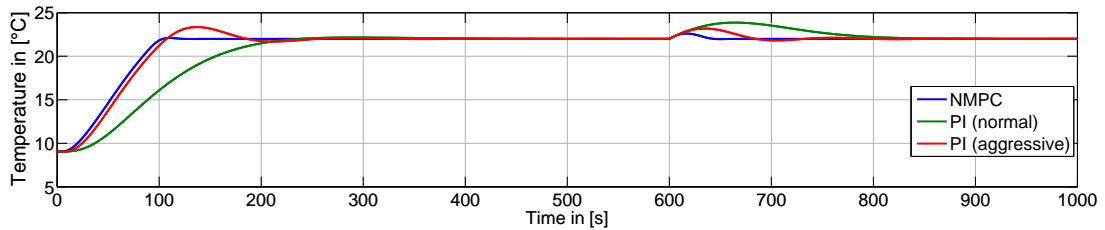


Figure 10. Passenger cabin temperature during scenario (transient heat-up and temperature step) with different controllers.

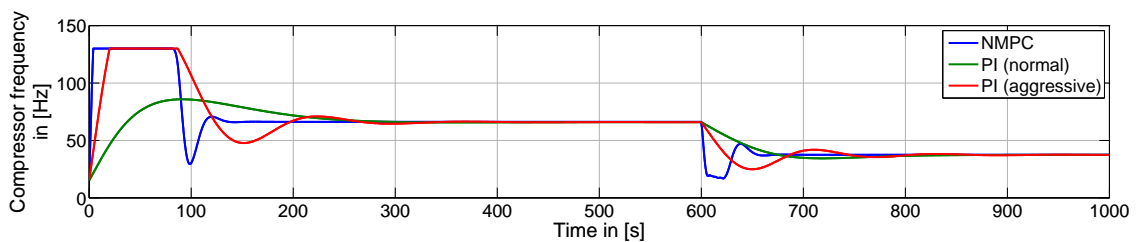


Figure 11. Frequency of the compressor during scenario (transient heat-up and temperature step) with different controllers.

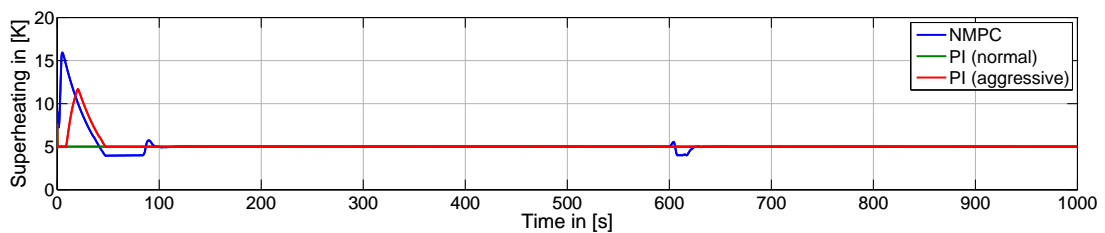


Figure 12. Superheating value during scenario (transient heat-up and temperature step) with different controllers.

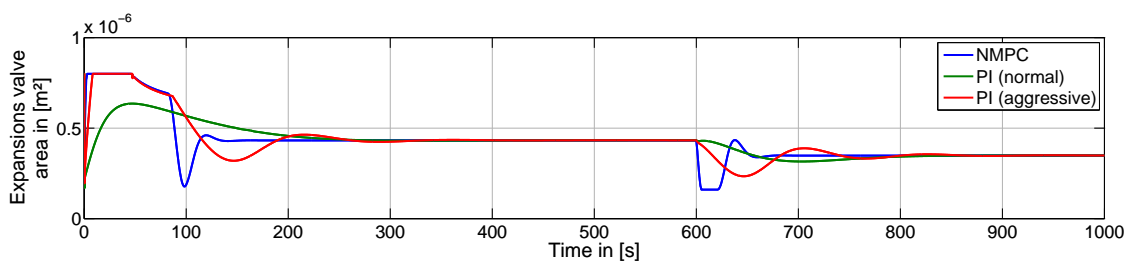


Figure 13. Expansion valve area during scenario (transient heat-up and temperature step) with different controllers.

Fig. 10). The compressor frequency and expansion valve reach their upper bounds faster and, subsequently, stabilize the system to a steady-state far more rapidly with NMPC (cf. Fig. 11 and Fig. 13). The predictive character of NMPC taking into account the system’s thermal inertia can be identified on the basis of the compressor frequency already being decreased at a temperature well below 20°C. Fig. 12 depicts the effect of the rather weak ”tracking” weighting for the superheating temperature in the NMPC

objective function and the corresponding soft-constraint to ensure a lower operational bound of 4 K (cf. §5.1). Superheating value fluctuations are acceptable unlike violations of the operational limits, which are prevented here by the NMPC approach. On the basis of Tab. 1, showing criteria for controller performance as overshoot, settling time and rising time, we can eventually state, that NMPC combines the fastest and most efficient way to heat up the cabin temperature to its set value.

	NMPC	PI (norm.)	PI (aggr.)
Rising time [s]	78.2	162.7	80.4
Settling time [s]	95	194.6	173.1
Overshoot [%]	0.4	1.25	10.50

Table 1. The NMPC controller’s characteristics during transient heat-up significantly outperform PI control.

Reaction to Disturbances At $t = 600$ s the controllers can be compared when dealing with a disturbance, here, an ambient temperature step of $+5^\circ\text{C}$. On the basis of Fig. 10, we conclude that the NMPC shows the best behavior, with a very small amplitude and a very short time interval before the steady state is reached again. This is not due to the predictive character of NMPC, as the temperature change is not known in advance. The NMPC controller gets the information about a temperature change along with the measured state vector at $t = 600$ s. Again, the weak weighting of the superheating value can be observed in Fig. 12, as the superheating value is affected by the disturbance in the case of the NMPC-controller.

CPU Time The computations were performed on a workstation using a single core of an Intel Xeon CPU at 3.5 GHz. To guarantee real-time feasibility for future application in a vehicle, it is necessary that the duration of feedback and prepare phase is shorter than the chosen sampling time (1 s here). If this can be ensured, the feedback phase duration is the time delay between the measurement of the system state and the availability of new values for the manipulated variables. Naturally, a short duration is essential to make sure the applied feedback relies on up-to-date system state information.

Fig. 14 shows a graph of the CPU time consumed by both phases. The real-time feasibility limit is indicated by a dotted red line and mostly not exceeded in the scenario. However, three CPU time peaks within the transient heat-up phase and the abrupt temperature change phase still violate the limit. The peak at the beginning is due to a cold start of the NMPC controller. Since we start stationarily this peak could be avoided by the execution of sufficiently many SQP iterations before a respective warm-start of the NMPC controller. An investigation of the remaining peaking behavior must yet be carried out, i.e. whether it originates from the particular model implementation or is a general property of the system. In future work, the latter could be addressed algorithmically by introducing adaptive relinearization into the NMPC schemes using, e.g., multi-level schemes (Bock et al., 2005; Kirches et al., 2012) or mixed-level schemes (Frasch et al., 2012). Fig. 15 shows a graph of the CPU time consumed by the feedback phase only. It is in the order of 0.5 – 2 milliseconds, which is a near instantaneous response on the measured system state relative to the system dynamics time scale. The feedback phase CPU time rises only very mildly during transient phases, and remains satisfyingly low throughout.

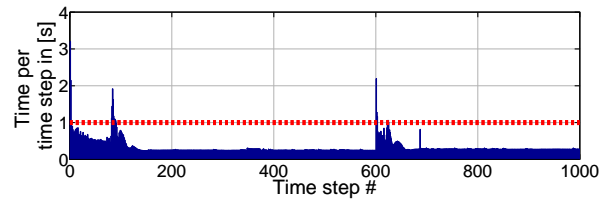


Figure 14. CPU time in seconds per NMPC iteration, consumed by all three phases and including FMU evaluation calls during the preparation phase. Measured time refers to an integration tolerance of 10^{-5} .

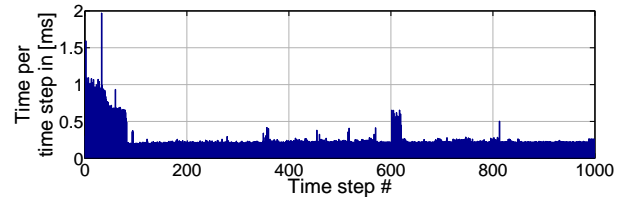


Figure 15. CPU Time in milliseconds per NMPC iteration consumed by the feedback phase only.

6 Conclusion & Outlook

The article discusses the development of an NMPC setup for a thermal management system of electrified vehicles. Compared to conventional PI control, several advantages concerning the transient heat-up and in reaction to disturbances were noted. The NMPC reaches the set-point value and settles considerably faster, nearly without any overshoot. This indicates an overall high degree of energy efficiency. Also, NMPC reacted faster on external disturbances that were not known in advance. A further benefit is the safe operating mode, as each state variable of the system can be constrained and constraints remained satisfied throughout all experiments. In the context of the heat pump application, the superheating value could be kept at a safe distance from the dew line in every operating point. The only observed drawback was the comparably high development effort that was necessary for developing the model and deploying NMPC for the system at hand.

The Functional Mockup Interface turned out to be a convenient way to export a previously developed Modelica model and to use it within the optimizer. To use the developed Modelica simulation model also for optimal control, though, several adaptations concerning smoothness assumptions were necessary. Although the optimizer could not be given access to the whole differential-algebraic equation (DAE) system due to intrinsic limits of the current version 2.0 of the FMI standard, the derivative-based optimization was found to work satisfactorily for the model at hand. A direct implementation of the DAE system in MUSCOD still promises a significant future increase in performance and numerical stability. Finally, we expected a higher impact of using analytic Jacobians provided to the optimizer by the FMU. Without insight into the auto-generated source code from which the FMU was

compiled, we could not rigorously answer the question of why the analytic Jacobian provided by the FMU is about twice as slow as the comparably simple one-sided finite difference approximation method we used in conjunction with MUSCOD.

Outlook. This article focused on controlling the cabin temperature through the use of NMPC. In a further step, the temperatures of cabin and battery will be tracked in parallel and the temperature of the electric components of the powertrain will be restricted to a realistic thermal range. The evaluation of the complete system can then be carried out on the basis of driving cycles.

Right now the NMPC controller still receives the entire measured state vector from the simulation. This is not realistic after deploying the controller to the final hardware application, as only a subset of the system state can be measured in reality. Thus, an observer will be employed to estimate the states that are not physically measurable.

In a second step, the hardware application will be targeted, where the whole developed thermal management system in an electric vehicle is controlled by the NMPC. The optimization must prove real-time feasibility to guarantee a solution within the defined time interval under all circumstances. The direct implementation of the DAE system in MUSCOD might prove to be essential to this end.

Acknowledgements. The authors acknowledge support by DFG Graduate School 220 and the Institutional Strategy of Heidelberg University funded by the German Excellence Initiative, and by the German Federal Ministry of Education and Research program “Mathematics for Innovations in Industry and Service 2013–2016”, grant n° 05M2013-GOSSIP. This publication was also written in the framework of the Profilregion Mobilitätssysteme Karlsruhe, which is funded by the Ministry of Science, Research and the Arts and the Ministry of Economic Affairs, Labour and Housing in Baden-Württemberg and as a national High Performance Center by the Fraunhofer-Gesellschaft. The authors are grateful to the TILMedia team at TLK-Thermo GmbH in Braunschweig for kindly providing access to a specialized version that supplies derivatives.

References

- A. Afram and F. Janabi-Sharifi. Theory and applications of HVAC control systems: A review of model predictive control (MPC). *Building and Environment*, 72:343–355, 2014.
- J. Albersmeyer. *Adjoint based algorithms and numerical methods for sensitivity generation and optimization of large scale dynamic systems*. PhD thesis, Heidelberg University, 2010.
- I. Bauer, H.G. Bock, and J.P. Schlöder. DAESOL – a BDF-code for the numerical solution of differential algebraic equations. Internal report, IWR, SFB 359, Heidelberg University, 1999.
- T. Blochwitz, M. Otter, M. Arnold, C. Bausch, C. Clauss, H. Elmqvist, A. Junghanns, J. Mauss, M. Monteiro, T. Neidhold1, D. Neumerkel, H. Olsson, J.-V. Peetz, and S. Wolf. The functional mockup interface for tool independent exchange of simulation models. *8th Int. Modelica Conf.*, 2011.
- H.G. Bock and K.J. Plitt. A Multiple Shooting algorithm for direct solution of optimal control problems. In *Proceedings of the 9th IFAC World Congress*, pages 242–247, Budapest, 1984. Pergamon Press.
- H.G. Bock, M. Diehl, P. Kühn, E. Kostina, J.P. Schlöder, and L. Wirsching. Numerical Methods for Efficient and Fast Nonlinear Model Predictive Control. In R. Findeisen, F. Allgöwer, and L. T. Biegler, editors, *Assessment and future directions of Nonlinear Model Predictive Control*, volume 358 of *LNCIS*, pages 163–179. Springer, 2005.
- J. Bonilla, S. Dormido, and F. E. Cellier. Switching moving boundary models for two-phase flow evaporators and condensers. *Communications in Nonlinear Science and Numerical Simulation*, 20:743–768, 2015.
- L. del Re, F. Allgöwer, L. Glielmo, C. Guardiola, and I. Kolmanovsky. *Automotive Model Predictive Control*. Springer, 2010.
- M. Diehl. *Real-Time Optimization for Large Scale Nonlinear Processes*. PhD thesis, Universität Heidelberg, 2001.
- H. Esen, T. Tashiro, D. Bernardini, and A. Bemporad. Cabin heat thermal management in hybrid vehicles using model predictive control. *22nd Med. Conf. Contr. Autom. (MED)*, 2014.
- T. Fischer, F. Götz, L. Berg, H.-P. Kollmeier, and F. Gauterin. Model-based development of a holistic thermal management system for an electric car with a high temperature fuel cell range extender. *11th Int. Modelica Conference*, 2015.
- R. Franke. Formulation of dynamic optimization problems using modelica and their efficient solution. *2nd International Modelica Conference*, pages 315–323, 2002.
- J.V. Frasch, L. Wirsching, S. Sager, and H.G. Bock. Mixed-Level Iteration Schemes for Nonlinear Model Predictive Control. In *Proc. IFAC Conf. on NMPC*, 2012.
- M. Gräber, C. Kirches, D. Scharff, and W. Tegethoff. Using functional mock-up units for nonlinear model predictive control. *9th International Modelica Conference*, 2012.
- J.M. Jensen and H. Tummescheit. Moving boundary models for dynamic simulations of two-phase flows. *2nd International Modelica Conference*, pages 235–244, 2002.
- A.Y. Karnik, A. Fuxman, P. Bonkoski, M. Jankovic, and J. Pekar. Vehicle powertrain thermal management system using model predictive control. *SAE International*, 2016.
- C. Kirches. Fast Numerical Methods for Mixed-Integer Nonlinear Model-Predictive Control. In H.G. Bock, W. Hackbusch, M. Luskin, and R. Rannacher, editors, *Advances in Numerical Mathematics*. Springer Vieweg, Wiesbaden, July 2011.
- C. Kirches, L. Wirsching, H.G. Bock, and J.P. Schlöder. Efficient Direct Multiple Shooting for Nonlinear Model Predictive Control on Long Horizons. *J. Proc. Contr.*, 22(3):540–550, 2012.
- C. Kirches, H.G. Bock, J.P. Schlöder, and S. Sager. Mixed-integer NMPC for predictive cruise control of heavy-duty trucks. In *European Control Conference*, pages 4118–4123, Zurich, Switzerland, July 17-19 2013.
- D.B. Leineweber, I. Bauer, A.A.S. Schäfer, H.G. Bock, and J.P. Schlöder. An Efficient Multiple Shooting Based Reduced SQP Strategy for Large-Scale Dynamic Process Optimization (Parts I and II). *Comp. Chem. Eng.*, 27:157–174, 2003.
- J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer Verlag, Berlin Heidelberg New York, second edition, 2006.