

FEM models in System Simulations using Model Order Reduction and Functional Mockup Interface

Andreas Gödecke*, Monika Mühlbauer†, Jörg Nieveler, Iason Vittorias‡, Thomas Vontz
Siemens AG, Corporate Technology
Otto-Hahn-Ring 6
81739 Munich, Germany

Abstract

The integration of a three-dimensional FEM model (ANSYS) in a dynamic, component-based system simulation tool (CoSMOS) is described. In order to avoid high simulation times of a direct co-simulation while maintaining the relevant details of the FEM submodel at the same time, model order reduction is applied to the FEM model. The reduced submodel is encapsulated in an FMU and finally imported in a system simulation. An example use case is presented to demonstrate the workflow.

Keywords: FMI, model exchange, model order reduction, CoSMOS, system simulations

1 Introduction

The need to describe complex dynamic systems, which involve several physical disciplines of potentially different timescales and levels of detail, increases continuously - e.g. almost any mechatronic design or power generation process falls into this category. Due to the interdependence of subsystems, a sequential development process is very time and cost consuming and requires many iterations. To make it more efficient, system simulations are used but their realization is hindered by a large variety of tools and models applied for and best suited to the different subsystems. An expensive, specific coupling for every pair of tools can be avoided by a standardized interface, like the Functional Mockup Interface [1], [5]. It has been introduced at the Siemens AG to combine several component-based system submodels, see [10]. In this contribution, we show a workflow that uses FMI for model exchange (version 1.0) to transport relevant

information from very detailed, three-dimensional FEM models. These are often performed in the detailed design phase of subsystems but bear a level of detail too high for system simulations, which is connected to long simulation times and large data amounts. To overcome these issues, we apply model order reduction to an FEM model, encapsulate it in an FMU and import this in a system simulation, see Fig. 1 for illustration. The latter is done exemplarily in a Siemens-internal, flexible simulation platform called CoSMOS.

In Section 2, we present the workflow in detail, starting from an FEM simulation in ANSYS, describing step-by-step the model order reduction and FMI encapsulation and leading to the CoSMOS simulation. Section 3 gives an example of the approach and Section 4 concludes the paper.

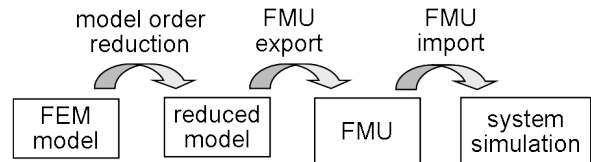


Figure 1: Overview of the workflow proposed.

2 Workflow

2.1 FEM simulation and Model Order Reduction

Model Order Reduction (MOR) techniques can drastically reduce the order and / or dimension of a large dynamical system without considerably sacrificing accuracy. The goal of MOR is to find a projection matrix \mathbf{V} so that the state-space $\mathbf{x} \in \mathfrak{X}^n$ of a full FEM model is projected onto a state-space $\mathbf{x}_r \in \mathfrak{X}^r$, where the dimension r is much smaller, i.e. $r \ll n$, with

$$\mathbf{x} = \mathbf{V} \cdot \mathbf{x}_r + \boldsymbol{\varepsilon} . \quad (1)$$

*andreas.goedecke@siemens.com

†monika.muehlbauer@siemens.com

‡iason.vittorias@siemens.com

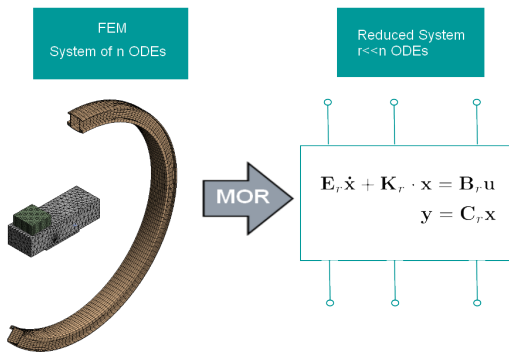


Figure 2: The discretized system, consisting of thousands of ODEs, is projected onto a lower dimension space.

The error difference ε in (1) should be minimal according to a norm specified.

Given the full system matrices from ANSYS and the required new lower dimension, the software MOR for ANSYS [9] generates the reduced order system matrices. Fig. 2 illustrates this workflow. It uses a Krylov-subspace method to find the projection matrix, refer to [8, 6] for more details. For a documentation on how to export the system matrices from ANSYS and how to apply MOR for ANSYS, see [2].

The reduced order matrices can then be imported in Matlab (or other preferred software) where the system can be transformed from the implicit form of Fig. 2 to an explicit one by multiplying with the matrix \mathbf{E}_r^{-1} , given that \mathbf{E}_r is not singular. The final reduced system then has the following form

$$\begin{aligned} \dot{\mathbf{x}}_r &= \mathbf{A}_r \mathbf{x}_r + \mathbf{B}_r \mathbf{u} \\ \mathbf{y} &= \mathbf{C}_r \mathbf{x}_r . \end{aligned} \quad (2)$$

It is important to note here that the matrices of the state-space (2) are independent of the input \mathbf{u} . This means that the reduced system, once created, can be used for many different inputs in transient or harmonic simulations without requiring any repetition of this process.

The method of order reduction applies to various model domains including electrical, thermal, or even fluid flow domain. Although it can in principle be used for second order systems as well, e.g. most mechanical systems, we focus on first order systems in this first approach, which are additionally linear. Note, that extensions for nonlinear systems exist by either using linearization techniques (or even splitting linear and nonlinear parts), or direct application of nonlinear methodology for model order reduction, e.g. proper orthogonal decomposition [7]. The main disadvantage

of the latter, compared to the linear systems case, is that a full simulation of the system is always required, affecting thereby the required development time. To the best of our knowledge, nonlinear equations are not supported by the software MOR for ANSYS and are not discussed in this paper.

2.2 Exporting a MOR model as FMU

The Functional Mockup Interface (FMI) [1] defines an open interface between numerical simulation tools. A zip-file, called Functional Mockup Unit (FMU) is distributed. It encapsulates a model description and static information in an xml-file as well as the model access in an exchangeable, simulator-independent binary file or source code in C. The present workflow uses FMI for model exchange (version 1.0) [4], which exports the differential equations of the model, relying on the importing simulator to perform the integration.

The FMU Software Development Kit (SDK) by QTronic GmbH was utilized to provide the core FMU functionality. Only a thin wrapper was implemented using Matlab, which performed the task of importing the MOR for ANSYS [9] matrices and converting them into explicit form (2), as discussed in the previous section. Moreover, the wrapper finally creates the C and XML files containing the model equations in the format required by the FMU SDK.

2.3 Import of an FMU in a system simulation tool

CoSMOS (Complex Systems Modeling, Optimization & Simulation) is an in-house simulation platform for dynamic system simulations that has been developed at Corporate Technology, Siemens AG, since 2000, cf. [11]. It is written in C# and C++ and is based on a client-server concept to allow easy tool coupling by using an open and modular architecture. Matlab, Ansys, WinCC, Excel and several optimizers have been considered so far among others.

One default client, called the simulation client, comprises component libraries for one-dimensional simulations in various domains as well as a selection of solvers for systems of differential algebraic equations. Continuous process variables, discrete signals and events can be handled. Results are written to .dcc or SQLite files and can be inspected by a graphical user interface. Dynamic simulations of fresh water and sewage flows, of conveying systems and power plant processes, of electric and traffic networks have

been performed.

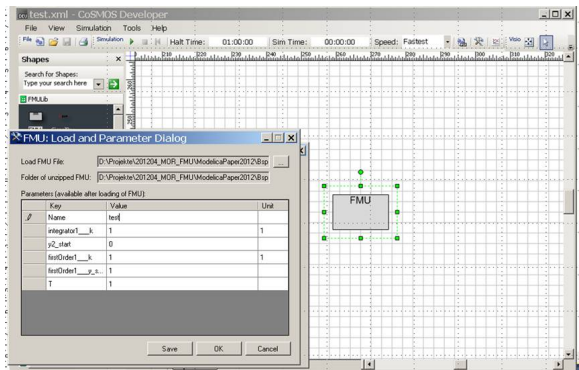


Figure 3: Generic FMU component with parameter dialog upon loading.

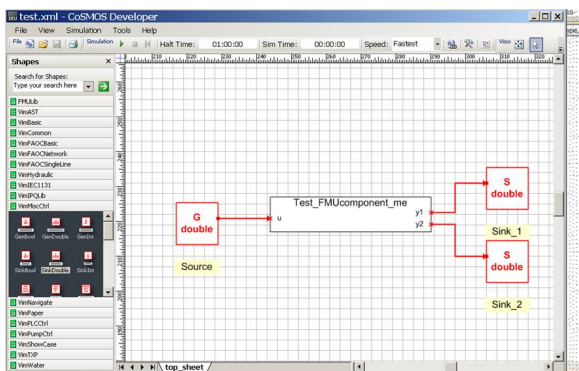


Figure 4: Specified FMU component with known parameters and ports.

Recently, the import of FMUs for model exchange has been implemented. Fig. 3 and Fig. 4 give an overview of the workflow with respect to the graphical user interface. A gray box component that represents a generic FMU is dragged and dropped from the library. After the user has specified the FMU file in the parameter dialog, the box component derives its inputs, outputs and parameters from the FMU xml file containing the model description [4]. This dynamic creation of a component distinguishes an FMU from all other components and its implementation means a considerable effort with respect to the graphical representation. The states of the FMU are added to the degrees of freedom of the overall system, they are updated using the dll-functions provided in the FMU. As FMI version 1.0 does not include Jacobian information yet, the latter is obtained numerically.

3 Example use case: Temperature control of a C-arm device

3.1 Description

In the following, we illustrate our proposed workflow on a thermal model of a C-arm device which is considered in a simple control environment. The C-arm device consists of two main thermal components: An X-ray source, which represents a heat source, and a closed water circuit with the water flowing from the X-ray source to a water tank and forth and back in two channels of the C-arm itself, see Fig. 5. By convection, the heat is transferred to the ambient air. The detailed thermal model of the C-arm device is to be considered in a system simulation which is used to test a very simple two-level control model. The latter observes the material temperature at three positions, where temperature sensors are to be positioned in the real device. If required, an additional fan can be turned on which leads to a decrease of the ambient temperature around the C-arm. The example is kept very simple for demonstration purposes.

3.2 FEM Model

The CAD model of the C-arm is imported in ANSYS (*DesignModeler*), see Fig. 5. The heat source and the tank body are thermally coupled to the C-arm at two areas, the water inlet and outlet. The power profile of

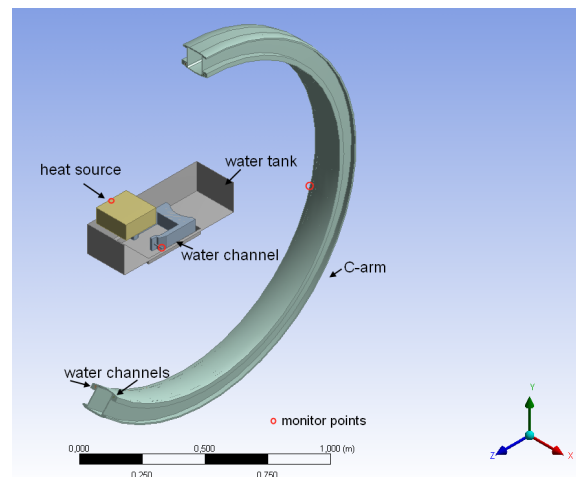


Figure 5: The design of the C-arm device. The two faces of the water tank are hidden to enable the visibility inside it. The red circles indicate the points where the material temperature is observed.

the X-ray source is the input to our model and deter-

mines the amount of heat generated in the heat source. The heat is transferred by conduction through the aluminum wall to the rest of the heat source body and then mainly by convection to the water circuit which has a fixed flow rate of 0.3 kg/s . The colder walls in water tank and C-arm take the heat from the water and discharge it finally to the ambient air by a second convection-dominated process. The convection coefficients are parameters defined as $14 \frac{\text{W}}{\text{m}^2} \text{ } ^\circ\text{C}$ for the C-arm and $0.001 \frac{\text{W}}{\text{m}^2} \text{ } ^\circ\text{C}$ for the water tank. Heat can hence mainly be disposed of in the C-arm itself.

The given geometry and physics data are imported in ANSYS Multiphysics. Accordingly, the load (heat generation) and constraint data (convection coefficients, flow properties) are provided either by using the graphical user interface or by the scripting language APDL. To properly represent the heat transfer caused by the water flow in the ANSYS Multiphysics model, *FLUID116* and *SURF152* elements are used [3].

The full system matrices are extracted using available ANSYS functionality.

3.3 Validation of the reduced model

Model order reduction is applied to the full system matrices using MOR for ANSYS reducing thereby the order of the model from $n = 59990$ to $r = 30$, i.e. $r \ll n$. For validation a comparison of the results was done between a simulation of the order reduced model in Matlab (set up directly and not integrated via an FMU) and an FEM simulation in ANSYS, which is considered to be our baseline. Matlab was chosen, because it was also used to transform the reduced model from implicit to explicit form and to finally build the FMU wrapper.

Three different material temperatures at potential sensor positions of the real device are selected for observation and are indicated by the red circles in Fig. 5: One is located on the heat source, one on the C-arm and one at the inlet of the water tank. The selection of the output nodes affects the structure of the C_r matrix in (2), and naturally, it adjusts the dimensionality of the output vector \mathbf{y} to 3 in this case.

Fig. 6 illustrates the comparison between the FEM simulation and the direct simulation of the reduced model. A huge saving in simulation time is achieved: While the FEM simulation required approximately 6 hours, the reduced system could be simulated in less than a minute. In terms of accuracy, the root mean square error (RMSE) remained very low for all three temperatures with values of 0.19°C , 0.19°C and 0.20°C , respectively. The exchange of the FEM model

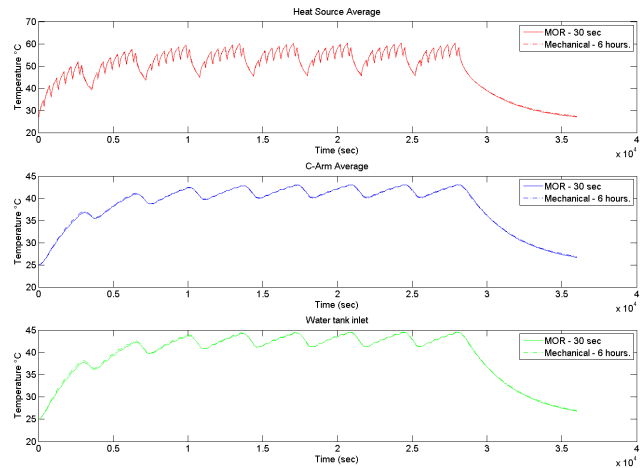


Figure 6: A comparison of a FEM transient simulation in ANSYS with the transient simulation of the reduced order system in Matlab. The power profile is given. Small errors are observed, whereas the reduction in simulation time is drastic, from 6 hours to less than a minute.

with the reduced model is hence justified in a system simulation which considers these three temperatures.

3.4 FMU and overall system behavior

As described in section 2.2, the reduced, thermal model is translated to an FMU which is imported in CoSMOS. The FMU has three inputs: $u1$ represents the power input to the X-ray source, $u2$ and $u3$ the ambient temperatures around the C-arm and the unit of water tank and heat source respectively. The three outputs of the FMU describe the material temperatures at the three intended sensor positions, already discussed in the previous section: $y1$ is located at the X-ray source, $y2$ at the C-arm and $y3$ at the inlet of the water tank.

A very simple two-level controller on the temperature of the heat source is built using the CoSMOS simulation client and its control library, cf. Fig. 7. Its function is tested in the following for a power input of $u1 = 600 \text{ W}$ to the X-ray source. If the temperature measured at the heat source (output $y1$ of the FMU) exceeds 50°C , an additional fan is switched on to full power and reduces the ambient temperature around the C-arm to 0°C (input $u2$ of the FMU). It is assumed that the ambient temperature around the unit of water tank and heat source is not affected (input $u3$). If the material temperature at the heat source falls below 40°C , the fan is turned off.

A modified CHORAL solver (default in CoSMOS) is used to simulate the process where the FMU contributes 90 states. Fig. 8 reveals the behavior of the outputs of the FMU upon simulation. It can be clearly seen that the fan is regularly switched on and off as expected. If the controls are disabled, the temperature at the heat source exceeds 70°C .

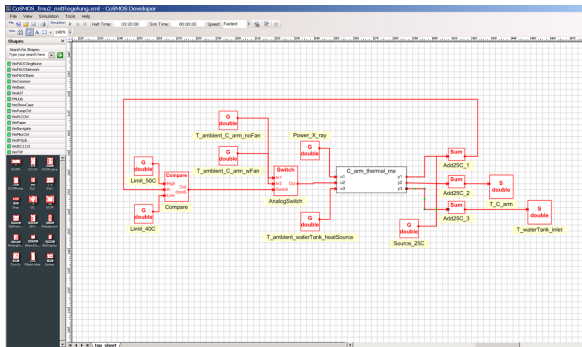


Figure 7: System model in CoSMOS. The thermal model is hidden in the FMU. The controls are built with the CoSMOS control library.

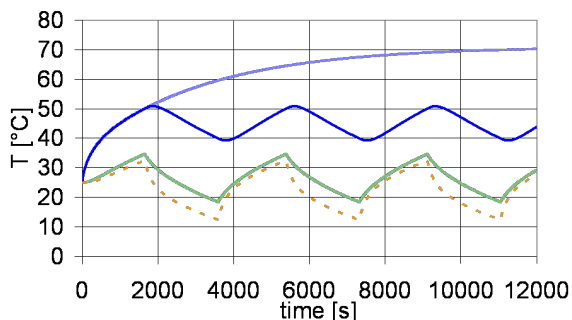


Figure 8: Behavior of the FMU outputs from bottom to top: Temperature at inlet of water tank, at C-arm and at heat source w/ controls and w/ disabled controls

4 Conclusion

In this contribution, we presented and validated a workflow to integrate highly accurate FEM simulations in dynamic system simulations while avoiding long simulation times common to FEM simulations. Model order reduction techniques enabled the creation of state-space representations of the FEM model, and the Functional Mockup Interface was used to transfer the reduced model to the system simulation that was performed with the Siemens internal tool CoSMOS. As an example, the integration of a detailed thermal

model of a C-arm device has been considered in a system simulation which tested a simple control model. The workflow presented applies to any scenario where the accuracy of a lumped model is insufficient for a system simulation, or where overall simulation time is too limited for a direct co-simulation between FEM and system model.

Future work will account for a comparison between various model order reduction algorithms as well as a consideration of stability criteria. The workflow presented up to the FMU generation will be automatized further and the limits of the overall workflow in terms of model complexity and size will be explored.

Acknowledgements

The authors thank Stefan Boschert for providing data to the use case of the C-arm medical device.

References

- [1] <http://www.functional-mockup-interface.org>.
- [2] <http://modelreduction.com/mor4ansys/#Documentat>.
- [3] http://www.cadfem.de/fileadmin/cfappdb/files/Consulting_Flyer_CADFEM_Thermalanalyse_IGBT_Module.pdf.
- [4] http://www.modelisar.com/specifications/FMI_for_ModelExchange_v1.0.pdf.
- [5] T. Blochwitz and M. Otter. The functional mockup interface for tool independent exchange of simulation models. In *Proceedings of 8th Modelica Conference*, Dresden, Germany, 2011.
- [6] R. W. Freund. Krylov-subspace methods for reduced order modeling in circuit simulation. *Journal of Computational and Applied Mathematics*, 123:395–421, 2000.
- [7] P. Holmes and G. Lumley, J. L. and Berkooz. Turbulence, coherent structures, dynamical systems, and symmetry. *Cambridge monographs on mechanics*, Cambridge, New York: Cambridge University Press, 1996.
- [8] E. B. Rudnyi and J. G. Korvink. Review: Automatic model reduction for transient simulation of mems-based devices. *Sensors Update*, 11(1):3–33, 2002.

- [9] E. B. Rudnyi and J. G. Korvink. Model order reduction for large scale engineering models developed in ansys. In *Lecture Notes in Computer Science* v.3732, pages 349–356. Springer, 2006.
- [10] Y. Sun, S. Vogel, and H. Steuer. Combining advantages of specialized simulation tools and modelica models using functional mockup interface (fmi). In *Proceedings of 8th Modelica Conference*, Dresden, Germany, 2011.
- [11] K. Wöllhaf and R. Rosen. A component-oriented simulation approach for industrial plant models: The plantsim simulation tool. *Siemens AG, Corporate Technology, Internal Report*, 2001.