Dirk Hartmann, Matthias Herz, Meinhard Paffrath, Joost Rommes, Tommaso Tamarozzi, Herman Van der Auweraer, and Utz Weber

12 Model order reduction and digital twins

Abstract: We are currently facing a substantial transformation of our industrial world and the way our economics are organized. This transformation, known as digitalization, is driven by the systemic integration of information technology in all kinds of devices, machines, and factories such that new smart networks are formed and new smart products have the ability to monitor, to forecast, and to control their behavior. One of the fundamental pillars of digitalization is simulation technology, since it enables the new intelligence layer in the form of digital twins which mirror the physical systems into the digital world – also named by Gartner Inc. as a top technology trend for 2017 and 2018. Creating such intelligence layers over several domains and life cycle phases requires, among other challenges, technologies for transforming and reducing complex simulation models. Exactly for this task a key technology is model order reduction (MOR). However, MOR is not only a key technology within emerging digital twins but also helps to reduce simulation times in the existing everyday business of simulation engineers. This is especially important when for a simulation model a large number of evaluations are needed. Within this chapter we present use cases where MOR is a key enabler for the realization of digital services and the reduction of simulation times. Furthermore we outline the potential of MOR in the context of realizing the digital twin vision.

Keywords: digital twin, virtual sensors, control, predictive maintenance, circuit simulation

MSC 2010: 35B30, 37M99, 41A05, 65K99, 93A15, 93C05

12.1 Introduction

This chapter provides an overview of several projects which we worked on throughout the last years. These projects were initiated from different directions and perspectives since the authors of this chapter work in multiple departments across Siemens. Nevertheless, all of our projects were either part of concrete business opportunities

Acknowledgement: The digital twin was implemented by Christoph Ludwig; see again [13], [64], and [63].

Dirk Hartmann, Meinhard Paffrath, Utz Weber, Siemens AG, Munich, Germany

Matthias Herz, Siemens AG, Erlangen, Germany

Joost Rommes, Mentor, a Siemens Business, Wilsonville, USA

Tommaso Tamarozzi, Herman Van der Auweraer, Siemens Industry Software NV, Leuven, Belgium

Open Access. © 2021 Dirk Hartmann et al., published by De Gruyter. This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

<https://doi.org/10.1515/9783110499001-012>

or of predevelopment activities to evaluate new business opportunities. This means that the goal was always to improve products, to develop new products, or to evaluate the potential lying in innovative business ideas. In this environment the application of model order reduction (MOR) was not a goal in its own. Instead the application of MOR was always triggered by the requirements coming from the project goals. In particular for predevelopment projects such a goal is typically to evaluate the commercial benefit lying in new technologies, which in our case was MOR.

In this chapter we start with outlining the underlying business visions of digitalization and digital twins and the role of MOR within this vision. This part is followed by a report of our experience with productizing MOR algorithms. Finally, we report the content, the challenges, and the results of some of our projects.

Throughout this chapter we try to give an insight into our work between the poles of business models and technological challenges, which is sometimes even the greatest challenge.

12.2 Digitalization and digital twins

Complexity in today's industry is exploding. New production methods, miniaturization of electronics, novel sensor technologies, and last but not least the Internet of things have led to many disruptive developments implying more and more complex products. On the one hand, this offers unique opportunities, e. g., in terms of efficiency or autonomy of components, products, and complex systems. On the other hand, it challenges today's design, engineering, operation, and service paradigms mostly focusing on manual expert interaction, which can hardly, if at all, handle this enormous complexity.

Digitalization changes everything everywhere. With the rise of new technology trends, such as AI foundations, intelligent things, cloud to edge, or immersive experiences [76], many of today's paradigms can be expected to be disrupted. Not only in the consumer market, as we can clearly observe today, but also in the industrial and medical sectors we see disruptions as proven by first early adopters.

Digital twins will be one key answer to these challenges; see, e. g., [24, 35, 81] for a broad overview from an engineering perspective. They are the next wave in simulation technologies (Figure 12.1). Digital twins integrate all (electronic) information and knowledge generated during the lifetime of a product, from the product definition and ideation to the end of its life. Examples of these data range from the initial requirements which have led to the design of the product, the design and engineering data, which have been generated during virtual design, to operation data such as sensor values collected during operation. The data themselves are only a central asset, if it can be used to make relevant predictions providing the right level of information at

Figure 12.1: Simulation is evolving from a troubleshooting tool to a key business driver in the form of digital twins.

the right time. Ultimately, digital twins mirror products and systems from the real into the digital world and vice versa.

From a high-level point of view, information included in digital twins can be split in to two categories: (i) pure data values with only little additional structure and knowledge associated, such as data gathered from sensors, and (ii) structured executable model-based data, in particular simulation models. Thus from this point of view digital twins bring together classical data-based schemes with model-based approaches such as simulation and optimization (Figure 12.2).

Today, most model-based approaches, and in particular simulation, are domain-specific and mostly used during design and engineering. The core concept of the dig-

Figure 12.2: Digital twins integrate model- and simulation-based approaches with data-based approaches such as artificial intelligence.

ital twin is to extend their usage along the complete life cycle and to deliver new services providing the right information at the right place in an efficient way, for example, digital twins supporting early system configuration during the sales process or optimization of operation and service concepts. This broad usage implies a number of requirements to modeling and simulation which diverge from its classical use in design and engineering:

- **Interactivity** – Speed and accuracy define the value of simulation and digital twins. Being very accurate, today's model and simulation approaches are extremely time-consuming. Speeding them up, while retaining the right level of accuracy, is crucial for extending the use of digital twins.
- **Reliability** – Users of digital twins cannot be expected to be sophisticated experts, like it can be expected during the use in design and engineering. Thus any prediction by the digital twin must be fail-safe and/or provided along with confidence intervals such that no expertise is required to interpret the results or can be used autonomously, e. g., by controls.
- **Usability** – Model-based and simulation tools are expert-centric today. Their resources are limited and thus the use of corresponding tools today is limited by the availability. Therefore, any digital twin solution must be accessible also for nonexperts from a usability perspective.
- **Security** – Many business models based on the digital twin will require to exchange digital twins between different parties. Reverse engineering must be prevented, such that no intellectual property is lost.
- **Deployability** – Digital twins will be used differently from the place where they have been created, e. g., on customer premises, in the cloud, on controls. Thus deployment must be easy to reduce barriers and efforts.

The digital twin concept has been originally introduced in 2003 by Michael Grieves [41] and first put to public by NASA in 2012 [38]. Digital twins are considered so important to business, that they were named one of Gartner's Top 10 Strategic Technology Trends for 2017 [76]. They are becoming a business imperative, covering the entire life cycle of an asset or process and forming the foundation for connected products and services. Companies that fail to respond will be left behind. For example, it is predicted that companies who invest in digital twin technology will see a 30 % improvement in cycle times of critical processes [77]. A potential market of 90 billion US dollar per year associated to corresponding offerings is predicted [28].

To realize the vision of digital twins, MOR is a key technology. Other key technologies cover novel user interaction paradigms and devices (such as virtual, augmented, or mixed reality), technologies for merging data and model-based approaches, or semantic technologies to easier built-up systems of digital twins.

12.3 Model order reduction in the context of digitalization and digital twins

The digital twin vision extends the expert-centric focus of modeling, simulation, and optimization technologies towards a digital assistance for everyone in day-to-day decisions. This is supported by a double exponential growth of capability in simulation technology. On the one hand, computational hardware is developing exponentially according to Moore's law [94]. On the other hand, efficiency of simulation algorithms is subject to exponential growth as well [91]. With this growing capability, computer-aided paradigms have become so powerful that they can provide novel simulation-based assistance in many fields, for example, digital twins providing new services for predicting failures, increasing operational efficiency, or for service planning [76].

However, compared to computer-aided tools in engineering, computer-aided assistance by means of digital twins is a niche application. The manual setup of corresponding models is a tedious task requiring simulation experts. This limits the use of model-corresponding concepts since corresponding efforts and costs are major roadblockers for increased use [60]. Furthermore, the lack of rigorous concepts for quantifying errors often implies very conservative safety margins, so that the full potential often cannot be exploited. Missing protection of intellectual property of models and the lack of standards (the functional mock-up interface [FMI] is only adopted slowly [10]) are hindering further. Thus today, digital twin-based approaches are only adopted in applications of high value, e. g., heavy-duty vehicles [44]. MOR [5, 4, 95] is a key technology to solve these challenges in the context of digital twins. By splitting computations in an offline and an online phase, computational effort is shifted to an offline phase allowing interactive simulation during the online phase. However, not only does this imply a speedup of calculations, but due to their reduced information set, reduced-order models (ROMs) protect intellectual property efficiently. While geometries can be recovered from the meshes of three-dimensional simulations, this is not the case for ROMs, in particular since the output generally focuses on the quantity of interest, i. e., a temperature at a single location rather than a complete temperature field. This furthermore increases the usability, since only relevant information is accessible. In addition, ROMs can be efficiently containerized using available standards such as FMI [10], thus increasing usability. In particular in view of the challenges laid down in Section 12.2, MOR is a key technology for digital twins.

A variety of concepts and approaches have been introduced in the last decades mostly using projection-based approaches such as proper orthogonal decomposition (e. g., [111]), balanced truncation (e. g., [42]), the reduced basis method (e. g., [80]), or Krylov subspace methods (e. g., [7]). The key idea of most approaches is to reduce the space of considered functions by means of an appropriate low-dimensional basis. For (close-to-)linear models, MOR is state-of-the-art in computational engineering and science. For nonlinear models it is a highly active field of research (e. g., [8]).

In addition to classical MOR methods, machine learning offers an alternative approach. Many successful applications, such as the efficient operation of wind parks [62], have been realized during the last years. Compared to model-based approaches, machine learning concepts require comparably little manual efforts to be set up. However, being data-centric, machine learning is not applicable where only few data are available. This is often the case in industrial applications, where relevant data cannot be measured, cannot be shared (e. g., due to IP concerns), or is simply not available (e. g., failure data for small lot products). On the one hand machine learning could be used to speed up simulation models by means of learning the underlying simulation data (e. g., [43]), but a combined approach with the ROM as the foundation and machine learning closing the accuracy gap seems to be a more promising approach [58]. However, such combined approaches have rarely been considered in the past and we believe that it has a strong future potential.

Within the following sections we review the application of MOR in projects which tackled concrete aspects of the digital twin vision described above. However, before describing these projects we generally review the process of productizing algorithms since the overall goal of every industrial R&D activity is to improve or deliver new products or services.

12.4 Model order reduction – from algorithms to products

12.4.1 Introduction

The process of making an algorithm suitable for use in commercial (CAE) software, also referred to as productizing, can be long and difficult to plan. Even if algorithms are known in the literature to be generally robust, the applicability to commercial software implementation is not always straightforward. In particular, it is challenging to foresee the user's needs and desired application of a method so that the method's assumptions do not lose validity. Moreover methods and software developers often face strict boundary conditions regarding implementation variants that are dictated by, e. g., the structure of the underlying physics engine or solver in which novel methods are implemented. Furthermore, while algorithms are usually designed by experts, the actual end-users are typically *not* experts in using those algorithms – they are experts in their own domain. Hence successful *productizing* requires not only that algorithms are robust with respect to applications, but also that their parameters can be (re)set in an *automatic* and *dynamic* way: *automatic* to reduce the need for users to set parameters and *dynamic* because parameters may need to be adjusted not only at the start of but also during the simulation. In this way the numerical methods become *transparent* to the user while the freedom of the user to interact with the algorithm is

somehow restricted. A good balance between *transparency* and *user freedom* has to be found. The situation becomes even more complicated if a working algorithm is not available or if the problem at hand is not yet fully understood and analyzed.

In this section we describe the various phases from algorithms to products. We assume that the problem to be solved is sufficiently well-defined (and constrained) and end-user requirements are known, and hence we focus on the process of solving the problem. As a concrete example, one could consider the typical MOR problem: given a dynamical system, find a reduced dynamical system that approximates the original system with a controllable trade-off between error and speed, and preservation of key properties like stability. We identify the following phases that will be discussed in more detail in the next subsections:

- research: literature study and investigation of novel approaches;
- prototyping: implementation of stand-alone or integrated software to allow feasibility studies;
- productizing: implementation in, or as, a product;
- customer feedback: closing the loop with new results and new requirements from end-users.

These phases may overlap in practice and moreover the process might become iterative: After customer feedback, but also during productizing, often new insights are obtained which require further research and prototyping.

12.4.2 Research

During the research phase, traditionally two activities are dominant: literature study and design of novel approaches. Depending on the complexity and confidentiality of the problem, these activities are carried out by one or more researchers, e. g., a technical leader, a (team of) researcher(s), and a MSc/PhD student, or even outsourced to an external party. For literature study, it is not only important to have the problem at hand well-defined, one must also know *which* literature to study. In some cases the right sources are naturally available because the researcher has experience on the topic. In other cases the topic may be less or even not covered in existing literature, or not in the context of the application at hand. Communication with colleagues (potentially in different divisions) and external parties like universities is then required to at least find a starting point. In several cases such contacts, for instance made during conferences or European networks like EU-MORNET [30], may develop to long-lasting collaborations with rewards such as scientific and commercial breakthroughs and staffing opportunities. The circuit simulation-related MOR work described in Section 12.10, for example, has been performed in collaboration with the TU Eindhoven, in the European project ASIVA14 [21], while the drivetrain dynamics simulation tools described

in Section 12.8 have been developed in cooperation with the KU Leuven and the University of Calabria within several years of research interactions and projects such as the Marie Curie H2020 project DEMETRA [57].

Often the problem is not sufficiently covered in the literature: The context or application may be different, the boundary conditions imposed by the main CAE solvers could be a limiting factor to the implementation of original algorithms, or the problem itself may simply be new. Even if the problem is well covered, one usually has to adapt and tune the proposed methods to the problem at hand. This stage, which may vary from simple changes of existing strategies to the design of novel approaches, typically involves prototyping, which we discuss in more detail in the next subsection.

12.4.3 Prototyping

When a set of methods is defined to achieve a specific target it is time to develop the first prototype code in order to test if the assumptions made during the research stage are valid and if the knowledge gained has application potential. In the prototyping phase, usually, one or more method developers and/or software engineers start to define preliminary software architectures and begin the implementation of a prototype code. Common choices for development environments are MATLAB [66] and Python [79]. As a good practice, the developed mock-up code should be easy to extend, it should be tested in a similar environment as compared to the target solver in which the final implementation is foreseen, and it should be flexible enough to be tested in multiple scenarios and maintain a satisfactory level of user-friendliness. In this way new extensions of the methods can be easily tested on multiple scenarios, the code can be shared with colleague researchers and consultants for usage in bilateral projects, and the risk of failure during the prototyping-to-product transition is reduced. Once the set of algorithms is mature enough, it is important to perform stress tests in the largest possible range of applications. Automatic testing is not mandatory but is surely an added value.

Using the specific case of MOR-related algorithms, it can happen that a large amount of parameters must be set by the user and that these are of difficult physical and mathematical interpretation to nonexpert users. Moreover, automatic parameter tuning algorithms are rarely available in the literature for the specific application foreseen for the implemented method. For this reason a big effort during the prototyping phase is generally spent in making the numerical methods *robust* and the automatic parameter setting *transparent* while still allowing advanced users to retain the desired level of control on the numerical method. During the prototyping phase of the method described in Section 12.8 the original number of parameters linked to the underlying MOR strategy was drastically reduced thanks to automatic parameter setting and the remaining parameters have been readapted to represent physical quantities that are easy to understand from a user point of view. Similarly, for the MOR approach

described in Section 12.10, most of the low-level parameters have been combined into macro-options that give the user (and developer) easy control over performance and accuracy.

If this target is achieved, the prototype should be tested on real engineering cases during, e. g., bilateral services projects and/or funded research projects. This step is useful to confirm the potential of the method, find out unforeseen usages, and detect potential limitations.

Often, at the end of the prototyping stage, a preliminary user interface is created to explore the usability of complex numerical solutions.

12.4.4 Productizing

Once the set of algorithms has reached a satisfactory level of robustness and usability the prototyping phase can be sided by the productizing phase. First the developed methods should be assessed for their market value, general applicability, and strategic importance. This stage is fundamental in order to assign a well-balanced amount of development resources. After this assessment the correct number of resources – generally one or more developers and/or software engineers – is assigned the task of implementation into the target commercial CAE solver. The goal is to translate customer specifications, design requirements, and prototype code into a professional and consistent implementation. Especially during the implementation of novel methods, it is of paramount importance that researchers and developers communicate on a regular basis. In practice, the specific research knowledge and the application-oriented character of many methods makes it hard to make consistent and complete code design specifications. In this case, developers may face the challenge of interpreting prototype code and might implement unintended behavior. It is advisable to initially allow researchers and developers to spend time together and even promote pair-coding activities. The more the algorithms are complex and have a dual theoretical-applied character, the more this practice should be promoted. During this period and in parallel with the method implementation into commercial solvers, a team of developers might also start to implement a user-friendly user interface. The more the numerical method has been refined and made robust, the less the user interface creation process is challenging. During the creation of the MOR method applied to drivetrains described in Section 12.8 a prototype user interface was also created in parallel with the research and method prototyping. This and the strong cooperation between the research and development units of Siemens allowed for a smooth transition of the prototype code and prototype user interface into a commercially available solution for MOR applied to drivetrain problems. One of the main challenges in the productizing of the methods in Section 12.10 was the choice on which parameters to make available to the user. This has been an iterative process itself, where researchers, developers, and application engineers were involved.

12.4.5 Customer feedback

No matter how sound the underlying theory is and no matter how many tests have been done, the most useful feedback on the quality (performance, accuracy) of the product is end-user feedback. The difficulty, as mentioned before, is that the test cases used by development teams typically do not cover completely the real cases used by customers. Hence, there is always a risk involved with releasing improved or new functionalities. The key is again communication to manage expectations, not only internally with sales and product engineering teams, but also with the customer (either directly or via customer-based application engineers): Roughly speaking, one of the first things to do when a customer request (bug report or enhancement request) is filed is to analyze whether there is a real bug in the theory and/or implementation, or whether the result is within accuracy tolerances but outside customer expectations. Ideally this first analysis is done by application or test engineers, but depending on the complexity, development teams may need to be involved as well. When the issue is identified as bug, apart from implementation errors, regularly one will have to go back to the underlying theory, for instance to adjust initially made assumptions or estimates, hence reiterating the phases described in the previous subsections.

When the result is within accuracy tolerances but outside customer expectations, the situation can become more complicated. Not only one has to be sure that the result is indeed within tolerances, but one also has to explain this to the customer: Particular care has to be taken here to avoid breaking long-standing trust relations. Furthermore, it might also be an indication that certain settings and options in the software are not clear for users, which may require software and/or documentation to be improved.

During the circuit simulation-related MOR work described in Section 12.10 all of the above-mentioned scenarios have happened. For example, a bug reporting a too large difference in signal delay was initially identified as a side effect of the way the delay was computed during postprocessing of simulation data. A deeper analysis, however, showed that while the actual delays were still within (user-settable) simulation tolerances, the used error estimations in the code were in fact too optimistic, and hence all phases above had to be reiterated in order to fix the issue. After the release of the first version of the drivetrain simulation tools described in Section 12.8, a user signaled an extension request to improve the usability of the tool for large system-level models that include multiple drivetrains. The user was contacted and asked for feedback about the urgency of the required extension. It was then decided in agreement within the party to take the time to develop a proper interface for the requested extension and release it together with the official product release a few months after.

12.4.6 Concluding remarks

We conclude by repeating what was mentioned in the introduction: The phases described in this section are typically visited in an iterative way. Moreover, they may in fact be visited in any order, for instance when through the acquisition of software (or company) one starts with an actual product that has to be integrated in a larger environment.

12.5 Use case – virtual sensors

12.5.1 Vision

The use of models to enhance or extend test-based engineering processes is one of the key application fields of model-based system testing [27]. Test data exploitation can be greatly enhanced by complementing sparse physical sensor measurements with model-based virtual sensor data [107, 20]. Control system efficiency can be increased by providing optimal control inputs using quantities which cannot be measured directly and operating system performance can be tracked through monitoring internal system states. Traditionally such control inputs or internal states of devices are measured during operation by hardware sensors [53]. However, due to cost restrictions or extreme physical conditions it is not possible to place hardware sensors at any desired position in any device. The goal of virtual sensors in all these applications is to provide online information about internal conditions or system performance based on simulation models instead of hardware sensors. These system models can be used offline to expand data sets or may be running parallel to operation, permanently synchronized with the current operation state, and report the desired internal states at the usual rate of the hardware sensors. From a business perspective such virtual sensor software modules may not only add value to the engineering process but can enable new simulation-based products such as advanced condition monitoring for improved availability or reduced downtimes. Furthermore, when virtual sensor algorithms and existing controllers are integrated into one software architecture, novel model-based controllers can be realized.

However, the systematic application of embedded simulation models for extended data analysis or parallel to operation is still a young field of activity. On the other hand, driven by the need to reduce development cycle times, simulation has become a frequently used tool during the development of products [17]. To draw reliable conclusions during the development process detailed three-dimensional simulation models are needed and the evaluation of these simulation models typically involves significant simulation times. This makes their reuse inside virtual sensor software and related state estimation a challenge.

In fact one of the central requirements for simulation models inside virtual sensors is the capability for fast estimation or even real-time capability when the results should be updated within the usual update frequency of hardware sensors. For this reason, MOR [8, 5] is applied to, e. g., detailed three-dimensional simulation models developed for design engineering purposes. This ensures reuse of the already available information and it allows to obtain fast or even real-time capable surrogate models which nevertheless operate within an acceptable accuracy.

12.5.2 Technological challenges

In this section the required steps for a virtual temperature sensor are described.

For a virtual temperature sensor the starting point is the thermal energy equation which reads for heat conduction with Fourier's law $\mathbf{q} = -\kappa \nabla T$ [67, 55, 105] for a computational domain Ω as

$$\begin{aligned} \partial_t(C_p T) + \nabla \cdot (-\kappa \nabla T) &= h && \text{in } \Omega, \\ \mathbf{q} \cdot \mathbf{n} &= h_f && \text{on } \Gamma_N, \\ \mathbf{q} \cdot \mathbf{n} &= \alpha(T - T_{\text{amb}}) && \text{on } \Gamma_R. \end{aligned} \quad (12.1)$$

Here, T is the temperature field, T_{amb} is the ambient temperature, C_p is the specific heat capacity, κ is the heat conductivity, and α is the convection coefficient [61, 56]. In a typical industrial setup, Dirichlet boundary conditions are not used. Instead, the thermal losses are captured by the volume heat load h or the heat fluxes h_f at the boundary. The most important boundary condition is the Robin boundary condition, which is also known as Newton's law of cooling [56]. This boundary condition models the thermal communication with the environment. Especially when a thermal model contains only solid bodies which are surrounded by a coolant, the convective heat transfer coming from the coolant flow can be modeled by a given distribution of convection coefficients. For example, this applies to thermal models of electric motors which contain the solid parts of the stator, rotor, and housing, but not the flow domain of the cooling air flow.¹

To start the MOR procedure, the thermal energy equation has to be written as a state-space system in the form

$$\begin{aligned} E \frac{d}{dt} \mathbf{x} &= \mathbf{A} \mathbf{x} + \mathbf{B} \mathbf{u}, \\ \mathbf{y} &= \mathbf{C} \mathbf{x}. \end{aligned} \quad (12.2)$$

¹ A full conjugate heat transfer model would lead to a dramatic increase in complexity and computational time, since the turbulent and thermal air flow in the rotor-stator gap and around the stator cooling fins needs to be resolved [56].

Here $x \in \mathbb{R}^n$ is the system state, $u \in \mathbb{R}^m$ is the input which drives the system, and $y \in \mathbb{R}^p$ is the measurable respectively observable system output. Furthermore, the system matrices are of dimensions $E, A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, and $C \in \mathbb{R}^{p \times n}$.

To obtain the thermal energy equation (12.1) as system (12.2), the following steps need to be performed.

- The heat load h and heat flux h_f are assumed to consist additively of contributions which only vary in time, i. e., $h = h_1(t) + \dots + h_l(t)$ and $h_f = h_{f,1}(t) + \dots + h_{f,k}(t)$. This assumption is fulfilled for a typical thermal simulation model in the industry since the usual procedure in commercial three-dimensional simulation software is (a) to mark the relevant model components on which the heat loads and the heat fluxes are applied and (b) to specify the total thermal losses which are produced by these model components. In a subsequent step the commercial software distributes the total thermal losses spatially homogeneous over the marked model component [2]. This leads to $m = l + k$ inputs.
- A finite element method or finite volume discretization approach in space brings the thermal energy equation almost into the desired state-space formulation. Some minor changes are necessary since during the finite element method or finite volume assembly procedure a constant vector b_0 occurs at the right-hand side due to the Robin boundary condition [59]. This part is added to the input terms by extending the input matrix B as $B = (B, b_0)$ and the input vector u as $u = (u, 1)$. This leads in total to $m = l + k + 1$ inputs.
- Additionally the output matrix C has to build up according to the desired location of the virtual sensors. This is done by marking for each virtual sensor its relevant nodes or elements in the computational mesh. This determines for each virtual sensor its corresponding row in the output matrix C .

The major technological challenge in this process is to access the assembled system matrices from commercial CAE software. For Simcenter Thermal Flow [2, 99] this was solved with a special subroutine and for NX Nastran [2, 71] this was solved with DMAP [2, 25]. However, there are commercial CAE software packages which do not provide any customization possibility to access the system matrices or some explicit solvers even do not assemble global system matrices.

Once the state-space system corresponding to the thermal simulation model is obtained, any MOR method can be applied which works on state-space systems of type (12.2) [8, 5]. For thermal simulation models the matrices are huge in size but sparse [59]. In our experience, typical industrial small-sized thermal models contain up to 10^6 degrees of freedom and typical medium-sized thermal models contain up to 10^8 degrees of freedom. For this reason the Krylov subspace MOR methods are a good choice since Krylov subspaces have a long history in connection with linear iterative solvers for especially huge and sparse linear equation systems [39]. A detailed review of Krylov subspace MOR methods can be found, e. g., in [5, 8, 7, 72]. Instead of giving yet another introduction into Krylov subspace MOR methods we concentrate

on what is necessary to realize virtual sensors with these methods in an industrial environment.

However, classical Krylov subspaces MOR methods such as [72] are feasible only for linear and time-invariant systems (12.2). For products with temperature-dependent material properties, the heat equation (12.1) becomes nonlinear due to $\kappa = \kappa(T)$. In this case nonlinear algorithms (e. g., [8, 114, 69]) need to be applied.

12.5.3 Project description

The first goal was to establish a user-friendly work flow for generating ROMs from existing three-dimensional thermal simulation models in an industrial environment. For this goal the determining factors are that (a) the simulation models are constructed in a commercial CAE software and (b) the simulation engineers have profound knowledge in their physical domain and the used CAE software but in general they are not experts in MOR nor they are programmers; see Section 12.4 for more details. More precisely, since all commercial CAE software packages are used through graphical user interfaces, simulation engineers are generally not used to run algorithms in command line tools or software development environments.² This starting position requires (a) to interact with the commercial CAE software and (b) to hide the details of the MOR algorithms from the user (Section 12.2). For this reason a MOR plug-in for Simcenter [26], the flagship product of Siemens in the CAE market, was developed. This MOR plug-in adds a ribbon to the Simcenter Graphical User Interface (GUI) which guides the user with buttons and following pop-up windows through the process of generating, applying, and exporting ROMs (Figure 12.3). This MOR plug-in was developed as Siemens internal engineering tool and is in productive usage within different projects and departments. To ensure that the resulting GUI matches the user expectations, several

Figure 12.3: Model order reduction plug-in.

² The main task of simulation engineers is to support or enable the product development process based on simulative information. To accomplish this, simulation models with the relevant physical information are built from CAD models. From the obtained simulation results conclusions are then drawn, e. g., about the product design or the reliability, and this information is fed back in the development process. This means that simulation engineers are focusing on product development and not on algorithm development.

in-house simulation engineers were included in the process of designing the GUI and the work flow of the plug-in (Section 12.4.5). With this plug-in the step to easily generate ROMs from existing three-dimensional thermal simulation models was solved.

For realizing virtual temperature sensors, the next step is to wrap the obtained ROM inside a virtual sensor software module which is runnable on the target hardware and software architecture. For the communication with the surrounding software architecture, the virtual sensor software module must receive the current operating conditions, transform these conditions into the required input for the ROM, call the ROM, transform the ROM results into the required format, and feed the properly formatted ROM results back into the surrounding software architecture. Furthermore, one communication cycle of that kind must be done within an expected frequency.

A crucial point is the available information during operation. Typically the available information is not identical with the required input for the ROM. For example, for electric motors the current is known during operation and can be fed into the virtual sensor software module. However, the ROM obtained from the thermal simulation model requires heat loads as inputs. Thus it must be part of the offline phase, i. e., the creation phase of an ROM, to provide the required information for mapping the available inputs (e. g., current) to the required ones (e. g., heat loads). This task involves detailed product-specific knowledge and is a central key for a vital and accurate virtual sensor software module. In our projects this task was solved with detailed look-up tables which were provided by the respective engineering departments.

Another important ingredient of a virtual sensor software module is to ensure that the ROM is permanently synchronized with the actual operation condition of the product. This requires that the virtual sensor software module receives and adequately processes the relevant information about the current operation state to keep its internal ROM synchronized. In our projects we solved this task with online filtering algorithms [101, 100, 46, 52], such as Kalman filters, where the filtering was done based on the available temperature hardware sensors and the corresponding temperatures coming from the ROM for these locations.

The last step in the development procedure is to run system tests to improve the solutions based on this feedback. Some of our projects have currently reached this stage, whereas in our in-house hardware lab virtual temperature sensor software modules are already running and tested.

12.5.4 Results and summary

The main challenge of our projects was that virtual temperature sensors based on ROMs were realized for the first time for the considered products. This means that not only the software modules had to be developed but we also had to establish a work flow of how to realize these virtual sensor software modules. While customized one-time solutions are sufficient for research projects and first prototypes, they are not a

proper solution for new services or products. New products or services require a sustainable work flow which is integrated into the existing development ecosystem of the involved engineers (Section 12.4). The approach we put into practice started from existing three-dimensional thermal simulation models. These models were compressed with MOR and the resulting ROMs were small and fast enough to be executed within the usual hardware sensor update frequency, either in an embedded environment or in a cloud environment which is connected to the product. In order to integrate this task in the existing development ecosystem of simulation engineers, we developed a plugin for Simcenter, which is the standard CAE software within Siemens for simulation-based engineering steps.

The following task of integrating the ROM into the target hardware and software system was still realized as customized and manual solution for each product. A potential future integration of this step into the existing development ecosystem of automation engineers are new state-observer blocks within the Totally Integrated Automation portal, which is the engineering platform from Siemens for all kinds of automation tasks [104]. During our projects prototypical blocks for such state-observers based on ROMs were developed but a fully integrated solution is still pending. Nevertheless, exactly the integration into existing automation engineering software tools is the second important step in realizing virtual sensors in a standard way. Overall, in a typical industrial development ecosystem, the simulation engineers create the ROMs for virtual sensors and the automation engineers integrate the virtual sensors into the software architecture of the products. Thus, to establish virtual sensors based on ROMs there must be a fully integrated solution for both, the simulation and automation engineering ecosystems (Section 12.2).

12.6 Use case – predictive maintenance

Data-driven operation support has been a topic for about 10 years. The efficiency of methods such as condition-based monitoring or sensor-based fault detection depends on the amount and the placement of sensors.

12.6.1 Motivation for model-based predictive maintenance

Very new is the demand of simulative operation support [29]. It allows monitoring every position and physical size of a system at any time point. Due to this knowledge, the system state may be predicted at any time. A simulation-based software program runs in parallel to the operation and is synchronized by sensor values at every time point. In many reports this is called the digital twin. The benefits are summarized in Figure 12.4. Among the most important benefits are inspection and service planning,

Figure 12.4: Benefits of simulation-based operation support.

Figure 12.5: Advantages for offering a digital twin additional to the hardware.

lifetime prediction, advanced fault detection, and control and optimization during operation. Selling not only the hardware of the system but also additional services can be a huge advantage in countries with high salaries. There are existing first clients of Siemens who demand this kind of operation support. In a first view some services such as giving an availability guarantee may sound risky for a company. On the other hand this is a unique selling point and selling the risk may bring good profit; see any insurance company. A more accurate analysis leads to the conclusion that all participants may benefit from an availability guarantee (Figure 12.5). Giving an availability guarantee for products cannot mean that there are no downtimes due to faults or inspections. Instead, the downtimes, especially the unexpected downtimes, should be reduced. The main task is to detect faults at a very early stage and predict their degradation. Thus, immediate downtimes are transferred to predictive downtimes. The base for giving an availability guarantee is the early detection of faults. If a fault is detected, then its degree of degradation is predicted. Depending on this prediction an inspec-

tion may be scheduled and/or the performance of the system is reduced in order to achieve the inspection time. Often, the plant is located in very isolated regions. Thus, the execution of maintenance and spare part supply must be planned very carefully. The early knowledge of the cause of failure is of tremendous interest.

12.6.2 Oscillatory mechanical systems

We consider a solid body $\Omega \subset \mathbb{R}^3$ with boundary $\partial\Omega = \Gamma_D \cup \Gamma_N$, composed of a material with Young's modulus $E \geq 0$ and Poisson ratio $-1 \leq \nu \leq 0.5$. The body is subject to volume forces $\mathbf{f} : \Omega \rightarrow \mathbb{R}^3$ and surface forces $\mathbf{g} : \partial\Omega \rightarrow \mathbb{R}^3$. Displacements $\mathbf{d} : \Omega \rightarrow \mathbb{R}^3$ from some appropriate function space $\mathcal{E}(\Omega, \mathbb{R}^3)$ are determined by the equations of linear elasticity (see, e. g., [49]):

$$\begin{aligned} -\operatorname{div}(\mathbf{A}\mathbf{e}(\mathbf{d})) &= \mathbf{f} && \text{in } \Omega, \\ (\mathbf{A}\mathbf{e}(\mathbf{d})) \cdot \mathbf{n} &= \mathbf{g} && \text{on } \Gamma_N, \\ \mathbf{d} &= 0 && \text{on } \Gamma_D, \end{aligned} \quad (12.3)$$

where the strain $\mathbf{e}(\mathbf{d})$ is given by the symmetrized gradient of displacements,

$$\mathbf{e}(\mathbf{d}) = \frac{1}{2}(\nabla \mathbf{d} + \nabla \mathbf{d}^T) \in \mathbb{R}^{3 \times 3}, \quad (12.4)$$

and the stress $\mathbf{A}\mathbf{e}(\mathbf{d})$ is given by

$$\begin{aligned} \mathbf{A}\mathbf{e}(\mathbf{d}) &= 2\mu \mathbf{e}(\mathbf{d}) + \lambda \operatorname{trace}(\mathbf{e}(\mathbf{d}))\mathbf{I} \\ &= 2\mu \mathbf{e}(\mathbf{d}) + \lambda \operatorname{div}(\mathbf{d})\mathbf{I}. \end{aligned} \quad (12.5)$$

Here, $\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)}$ and $\mu = \frac{E}{2(1+\nu)}$ are the Lamé constants and $\mathbf{I}$ is the identity matrix.

Equation (12.3) is the strong formulation for linear static elasticity. A Galerkin discretization of the weak formulation (typically by finite elements) yields a linear system

$$\mathbf{K}\mathbf{d} = \mathbf{f}, \quad \mathbf{K} \in \mathbb{R}^{n \times n}, \quad \mathbf{d}, \mathbf{f} \in \mathbb{R}^n, \quad (12.6)$$

where n is the dimension of the ansatz space, $\mathbf{K}$ is the stiffness matrix, and, by abuse of notation, $\mathbf{d}$ is the vector of displacements and $\mathbf{f}$ the vector of acting forces.

In the dynamic case, i. e., when $\mathbf{d}$ and $\mathbf{f}$ are time-dependent, equation (12.6) is extended to [113]

$$\mathbf{M}\ddot{\mathbf{d}} + \mathbf{D}(t)\dot{\mathbf{d}} + \mathbf{K}(t)\mathbf{d} = \mathbf{f}(t), \quad (12.7)$$

where $\mathbf{M} \in \mathbb{R}^{n \times n}$ is the mass matrix and $\mathbf{D} \in \mathbb{R}^{n \times n}$ is the damping matrix. Note that both $\mathbf{D}$ and $\mathbf{K}$ may be time-dependent. An important special case of this is the rotor dynamic equation

$$\mathbf{M}\ddot{\mathbf{d}} + (\mathbf{D}(\omega) + \omega \mathbf{G})\dot{\mathbf{d}} + \mathbf{K}(\omega)\mathbf{d} = \mathbf{f}(t, \omega), \quad (12.8)$$

where ω denotes the angular velocity of the rotor and $\mathbf{G}$ is the so-called gyroscopic matrix [36].

12.6.3 Model order reduction

In many real-world applications, the number n of degrees of freedom of the discretized system (12.7) or (12.8) is large and its numerical integration is not possible in real-time. MOR strategies introduce a reduced state $\mathbf{q} \in \mathbb{R}^r$ with $r \ll n$, via $\mathbf{d} = \Psi \mathbf{q}$, $\Psi \in \mathbb{R}^{n \times r}$.

One way to obtain the reduction matrix Ψ for system (12.7) is to use modal reduction. Setting up the eigenvalue problem of equation (12.7)

$$\omega^2 \mathbf{M} \boldsymbol{\theta} = \mathbf{K} \boldsymbol{\theta} \quad (12.9)$$

and taking the first r eigenvectors, the matrix Ψ may be defined by

$$\Psi = \{\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_r\}. \quad (12.10)$$

A preferable technique may be the Krylov subspace methods [7, 93]. The subspace is defined by

$$\Psi = \{\mathbf{K}_\omega^{-1} \mathbf{f}, \mathbf{K}_\omega^{-1} \mathbf{M} \mathbf{K}_\omega^{-1} \mathbf{f}, \dots, (\mathbf{K}_\omega^{-1} \mathbf{M})^{r-1} \mathbf{K}_\omega^{-1} \mathbf{f}\}. \quad (12.11)$$

The Krylov basis may be computed by the Arnoldi algorithm, which delivers an orthonormal basis of the subset.

Inserting this into (12.7) and multiplying by Ψ^T , one obtains the *reduced equation*

$$\hat{\mathbf{M}} \ddot{\mathbf{q}} + \hat{\mathbf{D}}(t) \dot{\mathbf{q}} + \hat{\mathbf{K}}(t) \mathbf{q} = \Psi^T \mathbf{f}(t), \quad (12.12)$$

where

$$\hat{\mathbf{M}} = \Psi^T \mathbf{M} \Psi, \quad (12.13)$$

$$\hat{\mathbf{D}} = \Psi^T \mathbf{D} \Psi, \quad (12.14)$$

$$\hat{\mathbf{K}} = \Psi^T \mathbf{K} \Psi \in \mathbb{R}^{r \times r} \quad (12.15)$$

are the reduced matrices. In the case of rotor dynamics, $\mathbf{D}$ and $\mathbf{K}$ depend on ω . In the ramp-up phase of an electric engine, where the rotation frequency increases, the reduction operations (12.14) and (12.15) have to be performed in each time step. Interpolation schemes may reduce the computational effort. In the constant phase of the engine, also the reduced matrices remain constant.

As filtering methods are generally applied to first-order equations, we define as usual

$$\mathbf{x} = \begin{pmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{pmatrix}, \quad \mathbf{u}(t) = \begin{pmatrix} \mathbf{0} \\ \mathbf{f}(t) \end{pmatrix}, \quad (12.16)$$

and, assuming that $\hat{\mathbf{M}}$ is invertible,

$$\mathbf{A}(t) = \begin{pmatrix} \mathbf{0} & \mathbf{I} \\ \hat{\mathbf{M}}^{-1} \hat{\mathbf{K}}(t) & \hat{\mathbf{M}}^{-1} \hat{\mathbf{D}}(t) \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} \mathbf{0} \\ \Psi^T \end{pmatrix}, \quad (12.17)$$

so that we obtain the equivalent first-order system in state-space form

$$\dot{\mathbf{x}} = \mathbf{A}(t) \mathbf{x} + \mathbf{B} \mathbf{u}(t). \quad (12.18)$$

Figure 12.7: Model of the electric engine/generator configuration.

Figure 12.8: Unbalance detection of the rotor during operation.

12.6.5 Summary

By combining MOR techniques and nonlinear identification methods, a digital twin for detecting and localizing faults (in terms of unbalances) has been developed for rotating systems. Further efforts are made in order to predict the increase of vibration during operation. The time a critical vibration is achieved defines the moment for scheduling an inspection. Knowing this time at an early stage, an inspection and spare part supply can be prepared.

12.7 Use case – operation control

12.7.1 Engineering controlled systems

The product race has become an innovation race, reconciling challenges of branding, performance, time-to-market, and competitive pricing while complying with ecological, safety, and legislation constraints. The answer lies in “smart” products of high

complexity, relying on heterogeneous technologies and involving active components. The corresponding design and engineering process hence must take the integration of control functions in the product explicitly into account. This adds an important additional complexity to the design engineering process where the interaction between the control and the system requires these should be optimized concurrently. The current industrial practice however still treats passive system design and controller design as different and separate design loops with their own models and their own validation and verification strategies. Suboptimal designs and unexpected integration problems are the result. Not reusing the wealth of engineering models available from earlier detailed system design stages furthermore leads to inconsistency problems and ineffective engineering processes. Closing this gap offers a significant potential for optimized designs, better product performance, and fewer and shorter design iteration cycles [1, 106].

12.7.2 Technological challenges

Two classes of challenges can be distinguished in relation to the integration of control functions in the product. The first one targets the optimization of the controller architectures, strategies, and settings for a controlled product, hereby using a system or “plant” model in a virtual controller optimization process. The second challenge targets the design of an optimal control solution by including a model of the controlled system into the controller itself, for example in a model predictive control (MPC) approach [82]. For both cases, the used system models are typically developed dedicated for the control application taking into account feasible complexities and system simplifications. One objective to do so is to allow fast virtual testing and optimization cycles and to enable hardware validations in physical control environments. The detailed (for example multiphysics) design engineering models from the system design departments are typically not reused. The main reason for such a suboptimal approach is that either detailed system models from the system engineering design departments are not available in the control department or are of too high complexity to be used together with control simulation.

Focusing on the first challenge of optimizing the controller design, one can distinguish three phases:

Phase 1: The combination of the multiphysics simulation model with that of the controller enables the design of the control logic and the performance engineering of the intelligent system. This is referred to as “model-in-the-loop” (MIL). The simulation is offline, i. e., there is no requirement for real-time performance of the simulation. Basically, two interconnecting objectives can be distinguished: One is to perform systems engineering based on the multiphysics “plant” model, including the representation of (often simplified or idealized) control in the multiphysics model; the other is to perform control engineering, using a model of the system to be controlled (“plant

model”). The first objective, for example, serves the purpose of configuration design (how many actuators and sensors, where to place them, etc.) or concept evaluation studies or the optimization of the mechanical system design taking into account the presence of control and certain control laws. The second objective is oriented towards the development of the optimal control logic and the development and verification of control hardware, control libraries, and embedded software up to the validation and calibration of the control system on the electronic control unit (ECU) (Figure 12.9) [106].

Figure 12.9: Associative plant models for control engineering.

To couple the models, different approaches exist. One may embed state equations with a description of the plant system (e.g., multibody simulation models or one-dimensional ordinary differential equation-based system simulation models) into these of the control (or vice versa) to enable the use of one solver, or adopt a true co-simulation approach where each system part runs its own solver [106, 40].

Alternatively, or in combination with the above approaches, a reduction of the plant model (e.g., a finite element or complex, even nonlinear multibody simulation model) into a description compatible with the controller model (e.g., state-space formulation) can be used. The model reduction step mostly achieves its goals at the expense of the full observability and/or controllability of the physical phenomena, leading to a macroscopic “equivalence” but losing direct insight in the microscopic observation domain. The challenge is to develop model compression methodologies that allow maintaining a relation with the physical meaning of model parameters. Such

co-simulation and model reduction approaches are used both for MIL applications for systems engineering and for control logic engineering.

Phase 2: The next step is the development and optimization of the “embedded” control software. This needs also to be done in the context of the functioning of the multiphysics system to be controlled. This is referred to as “software-in-the-loop.” While some of this can be done in offline simulation (provided software libraries of the controller are available), the final optimization needs to take into account the working of the software in real-time, requiring real-time capable multiphysics simulation models.

Phase 3: The final testing and calibration of the controller software and hardware requires the controller to be connected to a multiphysics simulation model of the components, subsystems or system, in a dedicated computing environment that is referred to as “hardware-in-the-loop” [6]; of course, this requires real-time capable simulation models.

The use of MOR is hence a key factor for enabling a true model-based engineering approach where consistent engineering models can be used throughout the various design phases. The applied MOR methods may depend on the reduction purpose and the nature of the master models. For example, in mechatronics systems, these can be finite element, multibody, or multiphysics models which can be linear or weakly or strongly nonlinear. Two application cases will be briefly discussed.

12.7.3 Application to the design of an active sound quality control system

Active noise reduction (and sound shaping) is a widely studied research topic with many potential industrial applications. Structural-acoustic solutions using smart materials as sensor and/or actuators are explored, enabling intelligent structures. Such solutions are however typically developed as add-on systems which prevents optimizing their potential impact as part of the overall system. A model-based mechatronic engineering approach was developed to enable an integrated solution [22, 23]. It was applied to the active sound control of vehicle engine noise to the car interior. The challenge consisted of relating the large three-dimensional, frequency-domain (finite element- and boundary element-based) vibro-acoustic, and structural models for the vehicle structure and structural components, interior vehicle cavities, and exterior propagation field, with models of smart material sensors and actuators and a time-domain control model. A simplified vehicle structure was developed to allow experimental validation. It consisted of a concrete car body, an engine compartment with an artificial source, and a flexible firewall panel on which the structural-acoustic control was to be applied with piezo-elements. The rigid walls of the concrete car structure can be easily modeled and also treated with a dedicated damping surface (Figure 12.10).

Figure 12.10: Simplified car structure.

MOR was a key element in the modeling approach, allowing to incorporate the reduced model as a plant model in the controller simulation. The component mode synthesis (CMS) approach was used. Very large reduction factors were used, reducing the large structural/vibro-acoustic finite element model (25,000 acoustic degrees of freedom but which can overall easily reach hundreds of thousands of degrees of freedom when multiple flexible panels are included) to a time-domain state-space model of realistic size (200 degrees of freedom). The sensors and actuators were represented by one-dimensional models for their functional performance, while their added mass and stiffness are accounted for in the three-dimensional finite element models. The acoustic propagation was related to the structural outputs by means of an “acoustic transfer vector” approach. The modeling approach included the following steps using multiple software tools:

- generate structural mesh and apply material properties (finite element preprocessor);
- add actuator and sensor mechanical models (finite element preprocessor);
- run a modal analysis (finite element analysis);
- build the acoustic finite element model and perform modal analysis (finite element analysis);
- import the structural model and couple it with the acoustic one (finite element analysis);
- calculate actuator and sensor electromechanical coupling (extended finite element analysis);
- reduce and convert the finite element model into a state-space model (MATLAB);
- implement and optimize the controller with the coupled state-space model (MATLAB/Simulink).

The coupling between acoustic and structural models is shown in Figure 12.11.

After performing a coupled modal analysis, the desired degrees of freedom are taken to derive the state-space model. In this case, the state-space model features two inputs (one actuator on the firewall and a sound source in the engine compartment) and four outputs (three pressures in the passenger compartment and one velocity on the firewall). The state-space model derived from this coupled approach allows the implementation of any controller involving the predefined degrees of freedom, and

Figure 12.11: Coupled structural acoustic models to be reduced to a modal basis.

Figure 12.12: Multiattribute cost function for combined mechanical and control parameters.

if the finite element approach involves the systematic representation of the sensors and actuators, the resultant state-space model is, in fact, a representation of the fully coupled electro-vibro-acoustic system, with any possible input-output relationships allowed by the chosen degrees of freedom.

Using this model, an optimization procedure is performed. The cost function takes into account the sound pressure level at the drivers head, the actuator input energy, and the weight of the solution. The firewall thickness and the velocity feedback controller gain are the variables. The position of the collocated sensor/actuator pair (SAP) can be considered fixed or included in the optimization loop. Figures 12.12 shows the cost function for each thickness as a function of the feedback gain, on the best SAP position for each case.

The best SAP position and optimal feedback gain depend on the thickness, which indicates that the global optimum can only be achieved in such a concurrent design between the active and passive system characteristics, proving the effectiveness of an integrated mechatronics simulation approach. The same model can be used to evaluate different controller strategies such as combined feedforward/feedback, filtered-X LMS, and NEX-LMS, and for different performance targets (noise level optimization and/or sound quality control). A more extensive discussion of the various modeling aspects and the detailed optimization procedures can be found in [22, 23].

12.7.4 Application to control development using neural network-based model reduction

The development of the control of mechatronic systems becomes more complex when multiple actuators and sensors are interfacing with a highly dynamic multiphysical system, often not having the sensors available to develop an optimal control. In the automotive industry, controls have mostly been developed using rule-based methods, first by directly writing code, later using a model-based approach. In the automotive industry, the complex balancing of multiple performances of the combustion engine such as emissions, fuel economy, and acceleration performance have increased the complexity of the engine actuators. To develop the controls for such complex mechatronic systems, new methods are required. Optimal control such as MPC in combination with methods to predict virtual controllable quantities using state estimation technologies in combination with Kalman filtering are examples of such new technologies that start to find their entry in the automotive world. MPC and Kalman filtering-based state estimation require, however, models that run fast while keeping a certain level of accuracy. Often, ad hoc simplified models are (re)developed, neglecting the availability of detailed engineering models. Reusing such models would not only save modeling time but would also allow better consistency of the various design engineering models over the different attributes and product versions and variants.

The simulation models for designing mechatronic systems are often created based on a combination of detailed three-dimensional models and test data and have a high level of accuracy but have a too slow calculation time to be used in MPC or state estimation methodologies in real-time on an ECU. To be able to convert such system models into the context of optimal control in combination with virtual sensing, neural networks can be an ideal methodology to develop a control model directly from the detailed plant model to be controlled [50, 65]. Figure 12.13 explains the different steps in the process, showing the reduction of the detailed engine model to a neural network-based model for the virtual sensor as well as the optimal controller. The neural network ROM allows real-time execution of the model for both the sensing and the control action. Initial results using the controls model in closed loop with the detailed plant model for tracking purposes indicate that the performance, for scenarios

Figure 12.13: Advanced combustion engine (upper left), one-dimensional system model (upper right), and reduction by neural networks in the real-time virtual sensor and MPC controller models.

for which the neural network is not trained, remains within good accuracy as long as the important states of the model are kept observable. Further results in this field will bring more clarity in how broadly this technology can be used for engine controls or other advanced vehicle controllers.

By applying multiple load cycles covering the full operating space of the system, the neural networks can be trained to represent the relevant system behavior even in the case of strongly nonlinear system characteristics.

12.7.5 Summary

Model-based approaches find increasingly their way into the design of (optimal) control systems. In the majority of cases, however, the applied system models (“plant models”) are ad hoc developed low-complexity models that are not correlated to the design engineering models developed in the mechanical design stages. This not only leads to inefficient processes redoing efforts that could be recovered, but also gives rise to major issues related to consistency and traceability when design improvements,

versions, or variants are to be processed. MOR can offer an answer for both the control design and control implementation and opens up new opportunities for concurrent design of the mechanical and the control system. Major challenges are still presented by the very large reductions factors to allow fast control optimization or even real-time usage inside state estimators or MPC controllers. The use of a neural network-based approach to reduce complex nonlinear models subject to an envelope of operating conditions offers significant potential that is to be further investigated.

12.8 Use case – drivetrain analysis

12.8.1 MOR for contact mechanics problems

The *simulation of dynamical systems involving contacts between elastic bodies* [112] is a challenging and active research field on its own. In particular high-frequency phenomena, numerical stiffness, high degree of nonlinearity, high dimensionality, and multidisciplinary nature (mechanics, acoustics, fluid dynamics, tribology) are among the major challenges that researchers and software developers must address in order to efficiently solve these types of problems [11]. Despite its seemingly niche description, there is a wide range of applications in which these problems are found and need to be solved. In particular, the simulation of geared transmissions or drivetrains is practically ubiquitous if one has to deal with simulations of electromechanical machines. Drivetrains contain a multitude of components, including bearings, gears, clutches, and spline connections that are known to behave nonlinearly and contain multiple contacts between flexible objects. While several MOR methods have been applied largely and successfully [32] in the field of flexible multibody simulations [97] in both academic and industrial settings with a large growing body of literature, MOR methods dedicated to the field of contact mechanics have been only recently explored [12, 103]. Moreover the developed methods often target high-dynamic contact mechanics simulations with fully flexible bodies and dynamic interactions with flexible eigenmodes of the structure [12]. These problems would indeed remain practically intractable without the usage of MOR and/or computer clusters. On the other hand, a large set of system-level related problems (such as large drivetrains or more complex machines containing several drivetrains) might not need the level of fidelity and the still relatively large computational times necessary to solve a fully nonlinear dynamic problem. They might instead still benefit from MOR solutions that speed up simulation times and decrease memory usage and disk storage, while still retaining the required level of fidelity on both system and component levels (Section 12.2).

The method discussed in this chapter exploits both an advanced MOR strategy and physics-based considerations coming from the targeted application domain of drivetrain simulation and combines them. The result is a numerical strategy that is

efficient and computes accurately most of the drivetrain dynamics-related scenarios that are industrially relevant. The focus is on three-dimensional system-level simulations including lightweight and internal gears and noise-vibration and harshness (NVH) problems in a multibody simulation environment.

12.8.2 Technological challenges

While the application challenge is relatively straightforward to summarize – *efficiently and accurately solve three-dimensional multibody problems involving multiple gear contacts for system-level and NVH purposes* – the technical challenges connected to it are multiple and have their root in the mathematical description of the equations of motion of a multibody system. The following set of equations is an index 3 differential algebraic equation describing the dynamic motion of a flexible multibody problem:

$$\mathbf{M}(\mathbf{x})\ddot{\mathbf{x}} + \mathbf{K}\mathbf{x} + \mathbf{G}^T(\mathbf{x})\boldsymbol{\lambda} = \mathbf{f}_{\text{ext}} + \mathbf{f}_v, \quad (12.19)$$

where $\mathbf{M}(\mathbf{x})$ is the nonlinear mass matrix, $\mathbf{K}$ is the linear stiffness matrix, $\mathbf{x}$ is the vector of generalized coordinates, $\mathbf{G}$ is the Jacobian of the constraints, $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers, and $\mathbf{f}_{\text{ext}}$ and $\mathbf{f}_v$ are the vectors of external forces and the quadratic velocity terms. Without entering in more details (which can be found in, e. g., [11]) we mention that despite being a fully nonlinear problem, the equations describing the deformation of the flexible bodies present in the system can be reduced by using linear MOR methods for second-order systems in the following form:

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{D}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f}, \quad (12.20)$$

where $\mathbf{M}$, $\mathbf{D}$, $\mathbf{K}$ are the linear mass, damping, and stiffness matrix of the underlying finite element model, $\mathbf{u}$ is the vector of linear nodal deformations, and $\mathbf{f}$ is the vector of nodal forces. This system can be reduced thanks to Petrov–Galerkin projection methods:

$$\mathbf{W}^T \mathbf{M} \mathbf{V} \ddot{\mathbf{u}} + \mathbf{W}^T \mathbf{D} \mathbf{V} \dot{\mathbf{u}} + \mathbf{W}^T \mathbf{K} \mathbf{V} \mathbf{u} = \mathbf{W}^T \mathbf{f}, \quad (12.21)$$

where $\mathbf{W}$ and $\mathbf{V}$ are the left and right subspaces used to obtain the reduced system. Galerkin methods can be used in flexible multibody problems within the assumption of large gross motion but small deformations within each of the body frames. This assumption is amply satisfied in problems involving gear contacts.

While flexible bodies that are not involved in contact interactions can be reduced with very efficient techniques, such as balanced truncation [83, 42], Krylov [32], CMS [19], etc., flexible bodies that include contact interactions suffer from the so-called *interface problem* [102]. In practice, a very large and inefficient reduction space needs to be used for an accurate reduction. The size of the reduction space becomes proportional to the amount of degrees of freedom potentially involved in the contact interactions. This causes problems such as high memory usage, large precomputation time,

large storage requirements, and increased numerical stiffness. Finally, the contact detection phase is also computationally very costly and scales with the (large) number of degrees of freedom that can be involved during contact.

In recent years several methods have been presented to maintain a high level of accuracy – similar to nonlinear finite element full-order dynamic computations – but drastically limit the impact of the above-mentioned issues. In particular, the following works [11, 12] obtain very good results in terms of speedup and memory usage while losing only a fraction of the accuracy obtained with nonlinear finite element problems. The field of hyperreduction [18, 31] is also exploited to tackle the contact detection problem with very promising results. However, this methodology is relatively complex to include in proprietary multipurpose multibody solvers and, moreover, the simulation time and the user expertise needed to use these techniques do not match the requirements of system-level three-dimensional multibody software. In order to develop a novel method for gear contact simulation to be included in a commercial multibody solver, the following decisions have been taken based on the available state of the art:

- **Contact detection:** Hyperreduction for contact detection is still in its infancy and needs further development. For this reason the computational performances are improved by using geometrical considerations that are related to the specific application field of drivetrain dynamics.
- **Dynamic flexibility:** The majority of the applications that involve dynamic simulations must include the modal behavior of the full drivetrain but the eigenfrequencies related to gears bodies and teeth themselves are outside of the frequency range of interest for many applications. From this point of view, it was decided to concentrate on a correct representation of the contact stiffness to properly represent the quasi-static behavior of the gear contact and the overall three-dimensional system-level dynamics. The accurate evaluation of the contact stiffness is of paramount importance in the definition of the dynamic modes of the full drivetrain.
- **Contact stiffness formulation:** The contact interactions of finite element meshes require a very fine spatial discretization to properly capture the correct Hertzian nonlinear behavior during contact. For this reason it was decided to focus the attention on the development of a method that combines the advantages of both general finite element formulations but exploits analytical formulas near the contact regions.

While the first point is important but out of the scope of this chapter, the second and third bullets highlight how available techniques that can be found in the literature [3] have been enhanced thanks to a cooperation between Siemens PLM Software and the Mechanical Engineering Department of KU Leuven (Section 12.4.2). These developments lead to a method based on MOR [102, 12, 16] that allows to describe in an accurate way the gear contact stiffness, including effects such as gear body deforma-

Figure 12.14: Finite element mesh of a spur gear.

tion, Hertzian nonlinear stiffness, and teeth convective couplings, while remaining extremely efficient. The reasons for the efficiency and accuracy of the method are:

- **Efficiency:** The problem is treated quasi-statically; thanks to the MOR technique, very few degrees of freedom are retained to describe the teeth deformation. When possible, potential symmetries in the gears geometry are also exploited for efficiency purposes.
- **Accuracy:** Despite the application of MOR, the contact interaction is statically *quasi-exact* with respect to the full-order finite element model. *Convective* deformation terms that couple the deformation of different teeth are accurately retained. Local dynamic effects such as teeth dynamic vibrations that are less often of relevance during standard operations are instead discarded. While these dynamic effects might be relevant for problems such as dynamic ring gears excitations and high-speed applications, the method is implemented in a modular way so that future extensions are efficient to implement.

12.8.3 User-related challenges

The developed technology based on MOR achieves the objectives targeted at the beginning of the development but a key component still needs to be addressed to allow a smooth user experience and limit the amount of expertise needed to use the method: *usability* (Section 12.4). For this reason the parameters that control the level of *static completeness* of the reduction space can be adjusted with a single parameter that ranges between *zero* and *one* where the limit value of *one* represents exact static completeness at the expense of some longer preprocessing time and slightly slower simulations while lower values allow to obtain a trade-off between accuracy and speed. Moreover the user is provided with a parametric mesher that automatically creates the gears finite element meshes based on a few parameters (Figure 12.14) that is used for the automatic generation of the reduction space while further minimizing the user intervention.

The interfaces between the MOR method proposed and the multibody solver are integrated into a user-friendly *application-driven* user interface – the *Simcenter 3D*

Transmission Builder (Figure 12.15) – that proposes also a simplified work flow for the creation of complex drivetrains. Practically, thanks to the dedicated *Simcenter 3D Transmission Builder* interface and application-specific choices related to the MOR technique it was possible to obtain a seamless and user-friendly usage of advanced MOR numerical techniques available to nonexpert users.

Figure 12.15: Simcenter 3D Motion – Transmission Builder.

12.8.4 Validation of MOR for drivetrains against experimental results

The described numerical method based on MOR is released as a product in Simcenter 3D Motion under the name of *Advanced-FE preprocessor* gear contact. Before the product release, given both the complexity of the method and the number of assumptions made during development, the methodology has been validated using multiple numerical and experimental results. In this chapter we present a subset of the experimental validation results to show the accuracy of the proposed method. For a larger set of examples we refer to [85]. The validation process has been carried out thanks to the usage of an in-house precision gear test rig [74] jointly developed by Siemens PLM Software, KU Leuven, and the University of Calabria. The test rig has been designed and manufactured to assess typical gear-related physical quantities in static and dynamic conditions, under imposed conditions of misalignment and shaft compliances. Particular attention is given to the measurement of gear pair transmission error (TE) [75], which is a typical key performance indicator used for the assessment of drivetrain NVH performances.

As an illustrative example we present validation results related to the complex case of gear contact between two spur gears, including large friction, microgeometry modifications, and different loading conditions. The measured TE is shown in Figure 12.16. It can be seen that despite the wide range of torques applied, the proposed method is able to match the TE with a high degree of accuracy. This is particularly striking since the effects of microgeometry, friction (as noticeable in the discontinuous jumps in the TE), teeth flexibility, and local contact nonlinearities are highly interacting with each other. The experimental results and multiple numerical validations carried out confirmed the good performances of the method in terms of both accuracy and speed.

Figure 12.16: Experimental–numerical comparison of the proposed approach for TE evaluation.

12.9 Use case – lifetime analysis

In this section, we consider lifetime analysis for railway axles as an example for a digital twin in the design phase. The engineer would like to know already in the design

phase how lifetime depends on the inspection scheme, the inspection interval, the type of load, and the fatigue crack growth. Stochastic MOR serves as a key element for the realization of this digital twin.

12.9.1 Efficient computation of failure probabilities

An important part of lifetime analysis is the computation of small failure probabilities, which is a challenge for practical problems with CPU time-intensive function calls. As already pointed out in Chapter 10 of this volume of *Model order reduction*, Section 10.2.3, failure probabilities are described by multidimensional probabilistic integrals which may be discretized by a multivariate quadrature rule. In the case of failure probabilities, the integrand of the probabilistic integral is discontinuous such that common quadrature rules will not provide sufficient accuracy. Here the key idea is to reformulate the integral into an integral with a smooth integrand and then to apply the unscented Kalman filter (UKF) [51] for evaluating the integral. The reliability of a product, system, or process is often indicated by a failure function:

$$g(\mathbf{x}) = \begin{cases} \leq 0 & \text{unsafe,} \\ > 0 & \text{safe,} \end{cases} \quad (12.22)$$

where $\mathbf{x}$ is the vector of stochastic variables. The failure function $g(\mathbf{x})$ describes a damage mechanism, e. g.,

1. the mechanical stress or the temperature exceeds a given threshold (finite element analysis, computational fluid dynamics);
2. the amplitude of oscillations exceeds a given threshold (linear and nonlinear modal frequency analysis);
3. changes of the microstructure of the material as a prestage of cracks differs from a given pattern (stochastic Voronoi techniques, finite element method);
4. the crack size exceeds a given length (finite element method + crack size analysis);
5. a chemical species exceeds a given concentration (computational fluid dynamics, ChemKin).

Using this failure function, the failure probability reads

$$P(g(\mathbf{x}) \leq 0) = \int_{g(\mathbf{x}) \leq 0} \rho_{\mathbf{x}}(\mathbf{x}) d\mathbf{x}, \quad (12.23)$$

where $\rho_{\mathbf{x}}(\mathbf{x})$ is the stochastic density of $\mathbf{x}$. We restrict our presentation to the case of independent standard normally distributed variables $\mathbf{x} = (x_1, \dots, x_n)^T$. In the general case of a failure function depending on nonnormally distributed variables $\tilde{x}_i$, e. g., the Rosenblatt transformation may be applied [90], mapping $\tilde{x}_i$ to x_i for $i = 1, \dots, n$. In

order to obtain an integral over $\mathbb{R}^n$, an indicator function is introduced,

$$P(g(\mathbf{x}) \leq 0) = \int_{\mathbb{R}^n} \Gamma_g(\mathbf{x}) \rho_{\mathbf{x}}(\mathbf{x}) d\mathbf{x}, \quad (12.24)$$

where

$$\Gamma_g(\mathbf{x}) = \begin{cases} 0 & g(\mathbf{x}) > 0, \\ 1 & g(\mathbf{x}) \leq 0. \end{cases} \quad (12.25)$$

For practical applications, the Monte Carlo method is too expensive to evaluate the integral in (12.24) because of the high computing time. So a stochastic MOR method is required. A standard method for approximation of the integral in (12.24) is the first-order reliability method (FORM) [48, 47]. This method first computes the so-called beta point (the point of highest failure probability) and then constructs a linear approximation of the failure function in the beta point. For highly nonlinear failure functions g , the failure probability of the FORM will not be accurate. An extension of the FORM, the second-order reliability method [14], is more accurate but requires second-order derivatives, which are often not available in practical applications. Because of the discontinuous integrand in (12.24), standard quadrature formulas will in general lead to bad approximation properties. Our stochastic model order method now consists of a reformulation of (12.24) in an integral with a continuous integrand and subsequent application of a nonlinear filter method. The reformulation is possible if the failure function $g(\mathbf{x})$ is continuously differentiable in its coordinates and strictly monotone in at least one coordinate (backmapping approach); see [110]. Without loss of generality, let $g(\mathbf{x})$ be monotone in x_n . It follows that the critical $\bar{x}_n$ defining the limit state can be expressed as a function of $x_1, \dots, x_{n-1}$:

$$\begin{aligned} \bar{x}_n &= \zeta(x_1, \dots, x_{n-1}), \\ 0 &= g(x_1, \dots, x_{n-1}, \bar{x}_n). \end{aligned} \quad (12.26)$$

Then the failure integral (12.24) reads

$$\begin{aligned} P(g(\mathbf{x}) \leq 0) &= \int_{\mathbb{R}^{n-1}} \rho_1(x_1) \cdots \rho_{n-1}(x_{n-1}) \int_{\bar{x}}^{\infty} \rho_n(x_n) dx_n \cdots dx_n \\ &= \int_{\mathbb{R}^{n-1}} \rho_1(x_1) \cdots \rho_{n-1}(x_{n-1}) h(x_1, \dots, x_{n-1}) dx_1 \cdots dx_{n-1}, \end{aligned} \quad (12.27)$$

with

$$h(x_1, \dots, x_{n-1}) = \frac{1}{2} \operatorname{erfc} \left(\frac{\zeta(x_1, \dots, x_{n-1})}{\sqrt{2}} \right).$$

The smoothness of h follows from the implicit function theorem. The failure probability $P(g(\mathbf{x}) \leq 0)$ can thus be interpreted as the mean of the smooth function h . For evaluation of this mean a nonlinear filter method, called UKF [51], is applied. This filter can be used to estimate the mean and covariance of a nonlinear stochastic process $f(\mathbf{w})$, where $\mathbf{w} \in \mathbb{R}^{n_w}$ is a normally distributed random vector with mean $E(\mathbf{w})$ and covariance $P^{ww} \in \mathbb{R}^{n_w \times n_w}$. So-called sigma points $\mathcal{X}^{(i)}$, together with weights W_i^{mean} and W_i^{cov} , are constructed and mapped to $\mathcal{Z}^{(i)} = f(\mathcal{X}^{(i)})$ for $i = 0, \dots, p$. The unscented filter then yields an approximation of the mean μ and covariance P^{zz} of the nonlinear function by

$$E(z) \approx \sum_{i=0}^p W_i^{\text{mean}} \mathcal{Z}^{(i)},$$

$$P^{zz} \approx \sum_{i=0}^p W_i^{\text{cov}} (\mathcal{Z}^{(i)} - \mathbf{y})(\mathcal{Z}^{(i)} - E(z))^T.$$

By Taylor expansion one can show second-order accuracy of mean and covariance [51].

12.9.2 Stochastic crack growth

In stochastic crack growth, the crack depth a is a function of the stochastic parameter vector $\mathbf{x}$ and load cycle N ,

$$a = a(\mathbf{x}, N), \quad (12.28)$$

and the failure function (12.22) is given by

$$g(\mathbf{x}, N) = a_{\text{crit}} - a(\mathbf{x}, N), \quad (12.29)$$

where a_{crit} denotes a critical crack depth indicating failure of the component. We consider elliptical surface cracks given by crack depth a and crack form $b = a/c$. Both the initial crack depth $a = a_0$ and the initial crack form b_0 are stochastic. Further stochastic parameters are given by the so-called POD curve and the probability of crack initiation. In contrast to most other places in this handbook, here “POD” does not mean “proper orthogonal decomposition,” but “*probability of detection*.” The POD curve gives the probability of crack detection during inspection and so characterizes the inspection scheme. The final crack depth monotonically depends on the initial crack size a_0 such that the reformulation of the failure integral of Section 12.9.1 can be applied with $x_n = a_0$ in (12.26). The goal is to compute the cumulative failure probability for a given number of equidistant inspection intervals, under consideration of:

- the replacement of a component if a crack is detected during inspection;
- the probability of crack initiation (input).

We use the following notation:

T_i	i -th inspection time
$[a_{\min}, a_{\max}]$	domain of definition of crack depth a
α	initial crack depth
β	minimum detectable crack size at time T_k (according to POD curve)
$[\beta_{\min}, \beta_{\max}]$	domain of definition of β
$\hat{x}$	vector with realizations of all stochastic variables except for α and β
$\bar{\alpha}_n$	critical initial crack size leading to failure at time T_n , $a(\hat{x}, \bar{\alpha}_n, T_n) = a_{\text{crit}}$
P_f^n	probability that the crack is not detected during inspections and reaches the critical crack depth at time T_n
P_d^n	probability that the crack does not exceed the critical depth and is detected at time T_n
P_c^n	cumulative failure probability at time T_n under consideration of failure of replaced components
c_n	probability of crack initiation in interval $[T_{n-1}, T_n]$
$\rho_{\hat{x}}, \rho_{\alpha}, \rho_{\beta}$	stochastic densities of $\hat{x}, \alpha, \beta$

The probability of detection of a crack with depth a is given by

$$I_{\beta}(a) = \int_{\beta_{\min}}^a \rho_{\beta} d\beta.$$

The probability of detection of a crack at time T_n is

$$I_{\beta}^n = I_{\beta}^n(\hat{x}, \alpha) = I_{\beta}(a(\hat{x}, \alpha, T_n)).$$

Probabilities P_d^n and P_f^n are given by

$$P_d^n = \int_{\Omega} I_d^n(\hat{x}) \rho_{\hat{x}} d\hat{x}, \quad P_f^n = \int_{\Omega} I_f^n(\hat{x}) \rho_{\hat{x}} d\hat{x}, \quad (12.30)$$

with

$$I_d^n(\hat{x}) = \begin{cases} \int_{a_{\min}}^{\bar{\alpha}_1} I_{\beta}^1 \rho_{\alpha} d\alpha & \text{for } n = 1, \\ \int_{a_{\min}}^{\bar{\alpha}_n} (1 - I_{\beta}^1) \cdots (1 - I_{\beta}^{n-1}) I_{\beta}^n \rho_{\alpha} d\alpha & \text{for } n > 1 \end{cases}$$

and

$$I_f^n(\hat{x}) = \begin{cases} \int_{\bar{\alpha}_1}^{a_{\max}} \rho_{\alpha} d\alpha & \text{for } n = 1, \\ \int_{\bar{\alpha}_n}^{\bar{\alpha}_{n-1}} (1 - I_{\beta}^1) \cdots (1 - I_{\beta}^{n-1}) \rho_{\alpha} d\alpha & \text{for } n > 1. \end{cases}$$

The cumulative failure probability can then be computed by the following recursive scheme:

$$P_c^1 = c_1 P_f^1, \quad (12.31)$$

$$\begin{aligned} P_c^{n+1} = & P_f^1(c_1 + \dots + c_{n+1}) + P_f^2(c_1 + \dots + c_n) + \dots + P_f^{n+1}c_1 \\ & + (c_1 P_d^1) P_c^n \\ & + (c_1 P_d^2 + c_2 P_d^1) P_c^{n-1} \\ & + \dots + \\ & + (c_1 P_d^n + c_2 P_d^{n-1} + \dots + c_n P_d^1) P_c^1. \end{aligned} \quad (12.32)$$

The integrals in (12.30) are the mean values of $I_d^n(\hat{x})$ and $I_f^n(\hat{x})$ and are evaluated by UKF as previously described. The critical initial crack depths $\bar{a}_{n-1}$, $\bar{a}_n$ appearing as integral limits in the definition of $I_d^n(\hat{x})$ and $I_f^n(\hat{x})$ are computed by a bisection algorithm. This procedure of evaluating the failure integrals has been validated by Monte Carlo for a model problem in [73].

12.9.3 Lifetime of railway axles

For lifetime investigation of railway axles, we use the failure function in (12.29) with $a_{\text{crit}} = 10$ mm. The stochastic distributions of the initial depth a_0 and initial form b_0 of the elliptical surface crack are given in Table 12.1. In this study two inspection schemes are considered:

- ultrasound far end scan;
- ultrasound mechanized.

In the first case the axle is scanned by sound from one end of the shaft to the other in the longitudinal direction, in the second case in the radial direction. The POD curves of these schemes are shown in Figure 12.17. The cumulative density function of crack initiation is given later together with the results of the results of lifetime analysis. Fracture mechanics for railway axles are subject of current research [54, 108]. Here the fracture mechanical simulations are accomplished by the simulation program ERWIN from the

Table 12.1: Stochastic parameters of crack growth.

	Type of distribution	Meaning
b_0	Uniform distribution on $[0.6, 1]$	Crack form: quotient of crack depth and length ($b_0 = a/c$)
a_0	Shifted exponential distribution with $\lambda = 100$ and shift 0.012	Initial crack depth

Figure 12.17: POD curves.

Fraunhofer Institute IWM in Freiburg, Germany [34]. It predicts the propagation of elliptical surface cracks, for different types of bendings. The stress intensity factors are represented by so-called polynomial influence factors [15]. Inputs of the crack simulation are internal and external stress profiles (due to the external load spectrum), da/dN -curves, and the initial crack depth and form.

It should be noted that we call the crack simulation a black box. Inputs are the initial crack depth and length, and output is the final crack depth, which is subsequently used for evaluation of (12.29).

Table 12.2: Test cases.

Test cases	Type of load	Inspection scheme	Inspection interval
Case 1	Partial load	UT far end scan	100,000 km
Case 2	Full load	UT far end scan	100,000 km
Case 3	Full load	UT far end scan	50,000 km
Case 4	Full load	UT mechanized	100,000 km

Four test cases are considered with different loads, inspection schemes, and inspection intervals (Table 12.2). The resulting lifetimes are shown in Figure 12.18. Figure 12.18 also shows the probability of crack initiation, which is input to the lifetime calculations. Lifetime is represented as a function of the deferred distance in kilometers, where one kilometer corresponds to 354 load cycles. The results show how lifetime depends on the probability of crack initiation, the type of load, and the inspection scheme. As expected, the full load case with the worst inspection scheme and inspection interval 100,000 km (case 2) has the shortest lifetime. The second worst is the partial load case (case 1) with the same inspection scheme and interval. Lifetime of the full load case can be improved by either switching to shorter inspection intervals

Figure 12.18: Lifetime for different scenarios, with specified probability of crack initiation (black dotted curve).

(case 3) or to a better inspection scheme (case 4). Both cases lead to longer lifetimes than cases 1 and 2, where shorter inspection intervals are more effective than a better inspection scheme. One call of the crack simulation program takes approximately 15 CPU seconds on a 3.2 GHz processor. For each curve shown, 3,000 to 4,000 simulation calls are required.

12.9.4 Conclusion

As an example of a digital twin in the design phase, lifetime analysis for railway axles has been presented, for different inspection and load case scenarios. By using adapted stochastic MOR methods, stochastic crack growth under consideration of inspections can be computed in reasonable computational time, which would not be possible with pure Monte Carlo methods. Key elements are the reformulation of the failure integrals as mean values of continuous integrands so that a nonlinear filter like the UKF can be applied with sufficient accuracy.

12.10 Use case – circuit simulation

12.10.1 MOR in the electronics industry

MOR has been part of the standard techniques used in circuit simulation for a long time, with publications dating back to at least 1990 [78]. The relation is bidirectional, with the circuit simulation and semi-conductor industry providing several benchmark cases [92, 88]. It is not only Moore's law [70], which states that circuit design complexity roughly doubles every 2 years, that drives this relationship; it is also the growing

need of circuit designers to include more physical details in their simulations and at the same time to have control over accuracy and performance. From a technological point of view, MOR methods will have to be developed that can deal with the growing complexity: The number of unknowns typically increases with the size of the design and the advance of the technology node (decrease of transistor dimensions), while the (Jacobian) density of the problem typically increases with the amount of detail included (wire resistance, capacitive coupling, inductive coupling). The MOR challenge lies hence not only in the problem's dimension, but also in the problem's complexity in terms of coupling and detail. From a business point of view, it is clear that scalability of simulation software is key (Section 12.2). What is less clear, however, is in which part of the flow the scalability should apply. For instance, before actually simulating a circuit, the differential algebraic equations describing the behavior of the circuit first need to be constructed. This process is called *extraction* and the resulting description of the circuit that can be translated into a system of differential algebraic equations is called *netlist*. Whether to apply MOR during this extraction phase and/or the simulation phase is not always clear, not only for reasons of robustness and reliability, but also for commercial reasons. In the remainder of this section we will focus mainly on the technical challenges.

12.10.2 Technological challenges

At first sight, MOR problems arising in circuit simulation may seem easy as they fall into the most elementary class of linear time-invariant dynamical systems. Electrical circuits that include nonlinear elements such as CMOS transistors are described by systems of differential algebraic equations of the form

$$\mathbf{j}(\mathbf{x}) + \frac{d\mathbf{q}(\mathbf{x})}{dt} = \mathbf{s}(t),$$

with node voltages and currents $\mathbf{x}(t) \in \mathbb{R}^n$, (non)linear vector-valued $\mathbf{q}(t, \mathbf{x})$, $\mathbf{j}(t, \mathbf{x}) \in \mathbb{R}^n$ with the electrical branch contributions, and sources $\mathbf{s}(t) \in \mathbb{R}^n$. Typically, only a linear subsystem is considered for reduction. This linear system models the behavior of linear resistors (R) and capacitors (C) and is usually considered in the frequency domain:³

$$G\mathbf{v} + sC\mathbf{v} = B\mathbf{u},$$

with node voltages $\mathbf{v} \in \mathbb{R}^n$, inputs $\mathbf{u} \in \mathbb{R}^k$, Laplace variable s , conductance and capacitance matrices $G, C \in \mathbb{R}^{n \times n}$, and input mapping $B \in \mathbb{R}^{n \times k}$. One distinguishes between internal nodes and terminals (or ports): Internal nodes only have connections to other

³ For simplicity we do not include inductors (L).

nodes via RLC elements, while terminals have also connections to non-RLC elements like transistors. This means that internal nodes are candidates for elimination while terminals need to be preserved, in general, because they connect the linear subsystem to the rest of the system. In the circuit simulation community, MOR is also known as netlist reduction or parasitic reduction, with the adjective parasitic referring to the nonintentional nature of the RLC elements that model the wire resistance and capacitive and inductive coupling.

Methods from several well-known categories are used for reduction:

- Krylov subspace projection, among the first of MOR methods to be applied to electrical circuits [33, 72];
- balanced truncation, with a priori error bounds [84, 9];
- modal truncation, used for the construction of behavioral models [87];
- nodal elimination methods, which have as advantages the existence of error bounds and ease of implementation [98, 96].

An advantage of nodal elimination methods (and to a lesser extent modal truncation), especially in the context of circuit simulation software, is that the ROMs can naturally be translated into a reduced circuit with meaningful RLC elements. For Krylov subspace and balanced truncation methods, the ROMs are typically dense with nonphysical (negative) RLC elements, and integration requires interfaces to deal with matrix-based circuit descriptions.

Despite the developments in the MOR domain, even the problem of reduction of linear circuits is still not considered as solved. The following key challenges can be identified for linear circuits:

- Linear solve costs: For subcircuits that contain only resistors or capacitors, projection- and elimination-based MOR procedures are error-free [89, 98], but the question of how to minimize the fill-in created by node elimination for the full design system matrix factors is still open (and becomes more difficult for mixed RLC circuits).
- Coupled problems: With decreasing feature sizes and increasing frequencies, capacitive and inductive coupling becomes stronger and denser. As a result, the original system matrices become denser, the reduction procedure becomes more expensive, and the ROM may become even denser, rendering MOR less effective.
- Precise accuracy performance tuning: For users it is important to be able to trade off between accuracy and performance. For instance, for top-level verification, one can (and often has to) accept less accuracy in order to improve simulation speed or to make simulation possible at all. The challenge is here twofold: (1) how to estimate the effect on accuracy when integrating the reduced circuit into the full design and (2) how to estimate the effect on the overall simulation time.
- Which method to apply when: There is not a single best method for MOR that fits all problems. Hence there is a need to be able to select automatically and dynamically, based on certain characteristics, which method to use.

- Variability-aware analysis: Uncertainty quantification of the impact of process variability on design robustness requires ROMs that are valid for ranges of design parameters (with as additional complication that there can be many parameters).
- With designs having a growing number of nonlinear devices like CMOS transistors, also the need for robust and efficient MOR methods for nonlinear systems increases.

12.10.3 SRAM memories: critical path simulations

SRAM memory designs [109] are of interest for MOR methods for several reasons. SRAM designs typically have a relatively large memory block of 6- or 8-transistor bitcells (depending on the memory size) and some smaller control blocks. Although designers often replace the large bitcell matrix by a much smaller model (manually), for top-level simulations an automatic, accuracy-preserving method is required. Furthermore, the extraction (modeling) of the many and long wordlines and bitlines may result in netlists with not only many resistors but also many coupling capacitors. Especially for so-called critical path simulations, where one wants to ensure that the delay for read and write operations is within specifications, reduction must be done with care to guarantee that the delay error is within picoseconds or even less. Additionally, to assess robustness of the design against process variations, one needs to run many simulations and hence simulation time needs to be minimized. In short, for memory designs, MOR has to deal with all the challenges mentioned in the previous section.

In Figure 12.19 we show the results for time-domain simulations with reduction settings varying from conservative to aggressive. The main impact on accuracy (error in delay) and performance (simulation time) is caused by how coupling capacitors are reduced. It depends on the type of verification how much error is acceptable: This can vary from tens of picoseconds to less than one picosecond. The results are produced using the circuit simulator Eldo Premier [68].

12.10.4 Concluding remarks

Driven by rapidly increasing design sizes and complexity, MOR, also known as parasitic or netlist reduction, has become a standard and indispensable option in modern circuit simulators. Although current MOR methods are suitable for robust accuracy performance control of simulation of advanced CMOS designs, more advanced CMOS nodes and verification requirements will require the development of new methods and approaches. Not only accuracy and performance remain key priority, also methods that can be used in the context of other applications, such as variability-aware design, are required. In particular parameterized MOR methods for linear and nonlinear systems will need to be further developed in order to make them suitable for use in future industrial software (Section 12.2).

Figure 12.19: Voltage on vertical axis between -100 mV and 800 mV. Time on the horizontal axis between 150.125 ns and 150.180 ns. Time-domain simulation results of Eldo Premier [68] for conservative (orange dashed) and aggressive (blue dashed-dotted) reduction settings, compared to the golden reference (green solid). The simulation with aggressive reduction settings is two times faster than the simulation with conservative settings, but one has to pay with less accurate results (which however may still be acceptable depending on the type of verification): The delay error in the signal increases from less than 1 ps to almost 3 ps.

12.11 Conclusions and outlook

Within this contribution, we have reviewed a number of industrial success stories of MOR in the context of digital twins (Sections 12.5–12.10). Through MOR the corresponding simulations could be accelerated and reduced in their memory footprint. This enabled novel applications which would not have been possible without these improvements. Therefore, MOR is a key enabler for a new generation of digital twins (Sections 12.2 and 12.3). With respect to sustainable industrial applications and commercial software packages containing MOR engines it is crucial to close the gap from algorithms to products. Here, professional software development plays a crucial role, which we have addressed in Section 12.4.

MOR allows to reduce computational execution time of models while controlling accuracy. Application- and purpose-specific models with different requirements in terms of speed and accuracy can be realized. At the same time, MOR liberates simulation models from their execution engines, i. e., their specific simulation tools and numerical solvers. This allows a separation of the creator of a digital twin – typically a simulation engineer – and the consumer – anyone downstream including machines themselves – through appropriate application interfaces. Furthermore, this allows not only to reuse the models during operation as highlighted by some of the use cases, such as virtual sensors (Section 12.5), but also novel licenses and business models [37], such as pay-per-execution time. In particular, realizing a pay-per-execution busi-

ness model during operation allows to scale with the number of products sold by a company assuming that each product contains a digital twin. Typical business models in the context of simulation tools only scale with the number of engineers working in a company, assuming that each engineer uses corresponding tools. Therefore, the impact does not only lead to novel application areas but also to the way how industrial value streams are organized.

Summarizing the current rapid advancements in MOR, a novel generation of digital twins, so-called executable digital twins [45], is likely to emerge in the near future. An executable digital twin is a specific encapsulated realization of a digital twin with its execution engines. As such they enable the reuse of simulation models outside R&D departments. In order to do so, the executable digital twin needs to be prepared suitably for a specific application out of existing data and models. In particular, it must have the right accuracy and speed. The executable digital twin can be instantiated on edge devices, on premise servers, or in cloud environments and used autonomously by a nonexpert or a machine through a limited set of specific application programming interfaces (APIs).

In order to realize this vision, several key challenges remain open though many of them are subject to active research efforts:

- **How to prevent virtual reverse engineering?** Thanks to fast execution times, digital twins could be executed many times allowing to reverse engineer specific features, e. g., optimal control logics.
- **How to leverage MOR with black-box solvers?** Many commercial simulation tools do not provide APIs for systematic interaction with their kernels. However, this is a central requirement for integrating novel MOR tools. At the same time the development of such APIs will take significant time due to the existing development processes. That is, first, such APIs must be ranked high enough in feature backlogs for next software releases, and second, these features need to be validated and verified before they are available.
- **How to provide certifiable accuracy bounds for ROMs?** The usage of MOR to enhance operations, e. g., in the context of model predictive control, requires certifiable models, e. g., ensuring conservation of important quantities.
- **How to combine/integrate machine learning and MOR technologies better?** Machine learning technologies, e. g., neural networks, are rapidly expanding in industrial applications addressing similar aspects as MOR. However, combined concepts are still missing.
- **How to package ROMs appropriately?** Even though containerization technologies, e. g., Docker, have matured over the last years, it is not clear how to leverage them in the context of MOR, e. g., appropriate interfaces and standards are missing.

Mathematical research in MOR as well as close collaboration with industrial software providers and users will be key to address these challenges and ultimately realize the vision of executable digital twins.

Acknowledgment

The digital twin concerning the predictive maintenance was implemented by Christoph Ludwig; see again [13], [64], and [63].

Bibliography

- [1] A. A. Alvarez Cabrera, K. Woestenenk, and T. Tomiyama, An architecture model to support cooperative design for mechatronic products: A control design case, *Mechatronics*, **21** (3) (2011), 534–547.
- [2] R. Anderl and P. Binde, *Simulations with NX / Simcenter 3D: Kinematics, FEA, CFD, EM and Data Management*, Carl Hanser Verlag GmbH & Company KG, 2018.
- [3] A. Andersson and L. Vedmar, A dynamic model to determine vibrations in involute helical gears, *J. Sound Vib.*, **260** (2) (2003), 195–212.
- [4] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems*, vol. 6, Siam, 2005.
- [5] A. C. Antoulas, D. C. Sorensen, and S. Gugercin, A survey of model reduction methods for large-scale systems, *Contemp. Math.*, **280** (2001), 193–220.
- [6] M. Bacic, On hardware-in-the-loop simulation, in *Decision and Control 2005 and 2005 European Control Conference, CDC-ECC'05, 44th IEEE Conference on*, pp. 3194–3198, IEEE, 2005.
- [7] Z. Bai, Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems, *Appl. Numer. Math.*, **43** (1–2) (2002), 9–44.
- [8] U. Baur, P. Benner, and L. Feng, Model order reduction for linear and nonlinear systems: a system-theoretic perspective, *Arch. Comput. Methods Eng.*, **21** (4) (2014), 331–358.
- [9] P. Benner, Advances in balancing-related model reduction for circuit simulation, in J. Roos and L. R. J. Costa (eds.) *Scientific Computing in Electrical Engineering SCEE 2008*, pp. 469–482, Springer Berlin Heidelberg, 2010.
- [10] T. Blochwitz, M. Otter, M. Arnold, C. Bausch, H. Elmqvist, A. Junghanns, J. Mauß, M. Monteiro, T. Neidhold, and 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 University; Dresden; Germany, number 063*, pp. 105–114, Linköping University Electronic Press, 2011.
- [11] B. Blockmans, *Model Reduction of Contact Problems in Flexible Multibody Dynamics with Emphasis on Dynamic Gear Contact Problems*, PhD thesis, KULeuven, 2018.
- [12] B. Blockmans, T. Tamarozzi, F. Naets, and W. Desmet, A nonlinear parametric model reduction method for efficient gear contact simulations, *Int. J. Numer. Methods Eng.*, **102** (5) (2015), 1162–1191.
- [13] H. Brandtstaedter, L. Huebner, A. Jungiewicz, C. Ludwig, E. Tsouchnika, and U. Wever, Digital twins for large electric power trains, in *15th Petroleum and Chemical Industry Conference Europe*, pp. 24–28, 2018.
- [14] K. W. Breitung, *Asymptotic Approximations for Probability Integrals*, Lecture Notes in Mathematics, vol. 1592. Springer, 1994.

- [15] M. Busch, M. Petersilge, and I. Varfolomeev, KI Faktoren und Polynomiale Einflussfunktionen für axiale und Oberflächenrisse in Zylindern, Bericht T 18/94, IWM Halle, 1994.
- [16] N. Cappellini, T. Tamarozzi, B. Blockmans, J. Fiszer, F. Cosco, and W. Desmet, Semi-analytic contact technique in a non-linear parametric model order reduction method for gear simulations, *Meccanica*, **53** (1–2) (2018), 49–75.
- [17] K.-H. Chang, *Design Theory and Methods Using CAD/CAE: The Computer Aided Engineering Design Series*, Academic Press, 1st edition, 2014.
- [18] S. Chaturantabut and D. C. Sorensen, Nonlinear model reduction via discrete empirical interpolation, *SIAM J. Sci. Comput.*, **32** (5) (2010), 2737–2764.
- [19] R. Craig and M. Bampton, Coupling of substructures for dynamic analyses, *AIAA J.*, **6** (7) (1968), 1313–1319.
- [20] R. Cumbo, T. Tamarozzi, K. Janssens, and W. Desmet, Kalman-based load identification and full-field estimation analysis on industrial test case, *Mech. Syst. Signal Process.*, **117** (2019), 771–785.
- [21] G. De Luca et al., ASIVA14, <http://www.itn-asiva14.eu/> (2018-06-13).
- [22] L. P. R. De Oliveira, M. M. Da Silva, P. Sas, H. Van Brussel, and W. Desmet, Concurrent mechatronic design approach for active control of cavity noise, *J. Sound Vib.*, **314** (3–5) (2008), 507–525.
- [23] L. P. R. de Oliveira, K. Janssens, P. Gajdatsy, H. Van der Auweraer, P. S. Varoto, P. Sas, and W. Desmet, Active sound quality control of engine induced cavity noise, *Mech. Syst. Signal Process.*, **23** (2) (2009), 476–488.
- [24] Digital twins – believe the hype? <https://www.nafems.org/publications/benchmark/archive/april-2018/>, April 2018.
- [25] DMAP, https://docs.plm.automation.siemens.com/data_services/resources/nxnastran/10/help/en_US/tdocExt/pdf/dmap.pdf (2018-10-02).
- [26] DMAP, <https://www.plm.automation.siemens.com/global/en/products/simcenter/simcenter-cae-simulation.html> (2018-10-02).
- [27] F. L. M. dos Santos, R. Pastorino, B. Peeters, C. Faria, W. Desmet, L. C. Sandoval Góes, and H. Van der Auweraer, Model based system testing: bringing testing and simulation close together, in *Structural Health Monitoring, Damage Detection & Mechatronics*, vol. 7, pp. 91–97, Springer, 2016.
- [28] Double vision: Using digital twins to pair virtual and physical worlds, <https://youtu.be/AtYEpnEpp0>, 2018.
- [29] M. Eigner, T. Dickopf, H. Apostolov, P. Schaefer, K. G. Faisst, and A. Kessler, System lifecycle management: Initial approach for a sustainable product development process based on methods of model based systems engineering, in *PLM 14*, 2014.
- [30] EU-MORNET, <http://www.eu-mor.net/> (2018-06-13).
- [31] C. Farhat, T. Chapman, and P. Avery, Structure-preserving, stability, and accuracy properties of the energy-conserving sampling and weighting method for the hyper reduction of nonlinear finite element dynamic models, *Int. J. Numer. Methods Eng.*, **102** (5) (2015), 1077–1110.
- [32] J. Fehr and P. Eberhard, Simulation process of flexible multibody systems with non-modal model order reduction techniques, *Multibody Syst. Dyn.*, **25** (3) (2011), 313–334.
- [33] P. Feldmann and R. W. Freund, Efficient linear circuit analysis by Padé approximation via the Lanczos process, *IEEE Trans. Comput.-Aided Des.*, **14** (1995), 639–649.
- [34] I. W. M. Fraunhofer, Fracture mechanics and structural integrity, https://www.iwm.fraunhofer.de/en/services/component-safety-lightweight-construction/fracture_mechanics_structural_integrity.html.

- [35] A. Fuller, Z. Fan, and C. Day, Digital twin: Enabling technology, challenges and open research, arXiv preprint arXiv:1911.01276, 2019.
- [36] R. Gasch and H. Pfützner, *Rotordynamik*, Springer, 1975.
- [37] O. Gassmann, K. Frankenberger, and M. Csik, *The st. Gallen business model navigator*, 2013.
- [38] E. Glaessgen and D. Stargel, The digital twin paradigm for future NASA and US air force vehicles, in *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA*, p. 1818, 2012.
- [39] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, 2013.
- [40] F. González, M. Á. Naya, A. Luaces, and M. González, On the effect of multirate co-simulation techniques in the efficiency and accuracy of multibody system dynamics, *Multibody Syst. Dyn.*, **25** (4) (2011), 461–483.
- [41] M. Grieves, Digital twin: Manufacturing excellence through virtual factory replication, *White paper*, 2014.
- [42] S. Gugercin and A. C. Antoulas, A survey of model reduction by balanced truncation and some new results, *Int. J. Control*, **77** (8) (2004), 748–766.
- [43] X. Guo, W. Li, and F. Iorio, Convolutional neural networks for steady flow approximation, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 481–490, ACM, 2016.
- [44] B. Haas, Predictive control systems in heavy-duty commercial vehicles, in *Proc. Automotive Powertrain Control Systems*, 2012.
- [45] D. Hartmann and H. Van der Auweraer, Digital twins, arXiv preprint arXiv:2001.09747, 2020.
- [46] C. Heij, A. C. M. Ran, and F. van Schagen, *Introduction to Mathematical Systems Theory: Linear Systems, Identification and Control*, Birkhäuser, 2006.
- [47] M. Hohenbichler, S. Gollwitzer, W. Kruse, and R. Rackwitz, New light on first- and second-order reliability methods, *Struct. Saf.*, **4** (4) (1987), 267–284.
- [48] M. Hohenbichler and R. Rackwitz, First-order concepts in system reliability, *Struct. Saf.*, **1** (3) (1983), 177–188.
- [49] T. J. Hughes, *Linear Static and Dynamic Finite Element Analysis*, Dover Publication INC., 2000.
- [50] Institute for Mathematics and its Applications, Minneapolis (MN, USA). Integrating machine learning and predictive simulation: From uncertainty quantification to digital twins, 2018.
- [51] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, A new method for the nonlinear transformation of means and covariances in filters and estimators, *IEEE Trans. Autom. Control*, **AC-45** (3) (2000), 477–482.
- [52] R. E. Kalman, A new approach to linear filtering and prediction problems, *J. Basic Eng.*, **82** (1) (1960), 35–45.
- [53] T. Kenny, Sensor fundamentals – chapter 1, in J. S. Wilson (ed.) *Sensor Technology Handbook*, pp. 1–20, Newnes, Burlington, 2005.
- [54] M. Koch, A. Deisl, H.-P. Gaensler, and S. Jenne, Internationales Forschungsprojekt Eisenbahnfahrwerke 3, *ZEVRail*, **138** (2014), 93–97.
- [55] D. Kondeputi and I. Prigogine, *Modern Thermodynamics*, Wiley, 1998.
- [56] F. Kreith, R. M. Manglik, and M. S. Bohn, *Principles of Heat Transfer*, Cengage Learning, 7th edition, 2010.
- [57] KU Leuven University of Calabria and Siemens Industry Software NV, “*demetra*” (*design of mechanical transmissions: Efficiency, noise and durability optimization*), ec fp7 marie curie project nr. 324336.

- [58] R. R. Lam, L. Horesh, H. Avron, and K. E. Willcox, Should you derive, or let the data drive? an optimization framework for hybrid first-principles data-driven modeling, arXiv preprint arXiv:1711.04374, 2017.
- [59] M. G. Larson and F. Bengzon, *The Finite Element Method: Theory, Implementation, and Applications*, Texts in Computational Science and Engineering, Springer, 2013.
- [60] J. H. Lee, Model predictive control: Review of the three decades of development, *Int. J. Control. Autom. Syst.*, **9** (3) (2011), 415.
- [61] E. Lifshitz and L. Landau, *The Classical Theory of Fields*, Course of Theoretical Physics, vol. 2, Elsevier, 4th edition, 1975.
- [62] Z. Liu, W. Gao, W. Yih-Huei, and E. Muljadi, Wind power plant prediction by using neural networks, in *Energy Conversion Congress and Exposition (ECCE), IEEE 2012*, pp. 3154–3160, IEEE, 2012.
- [63] C. Ludwig, O. Junge, and U. Wever, Online parameter identification methods for oscillatory systems: Estimation of changes in stiffness properties, *J. Vib. Control*, 10.1177/1077546318810265, 2018.
- [64] C. Ludwig and U. Wever, Online fault identification for rotating machinery, in *ISMA2018 Conference on Noise and Vibration Engineering*, Leuven, Belgium, 2018.
- [65] P. Mas, S. B. Maddina, F. L. M. Dos Santos, C. Sobie, and H. Van der Auweraer, The application of artificial neural networks in mechatronics system development, in *ISMA2018 Conference on Noise and Vibration Engineering*, Leuven, Belgium, 2018.
- [66] MathWorks, <https://mathworks.com/products/matlab.html>.
- [67] P. Mazur and S. R. de Groot, *Non-Equilibrium Thermodynamics*, North-Holland Publishing, 1969.
- [68] Mentor Graphics, a Siemens Business, Eldo Premier 2018.1, https://www.mentor.com/products/ic_nanometer_design/analog-mixed-signal-verification/eldo-platform.
- [69] S. S. Mohseni, M. J. Yazdanpanah, and N. Abolfazl Ranjbar, New strategies in model order reduction of trajectory piecewise-linear models, *Int. J. Numer. Model.*, **29** (4) (2016), 707–725.
- [70] G. E. Moore, Cramming more components onto integrated circuits, *Electronics*, (April 1965), 114–117.
- [71] N. X. Nastran, <https://www.plm.automation.siemens.com/global/en/products/simcenter/nx-nastran.html> (2018-10-02).
- [72] A. Odabasioglu and M. Celik, PRIMA: Passive Reduced-order Interconnect Macromodeling Algorithm, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, **17** (8) (1998), 645–654.
- [73] M. Paffrath and U. Wever, Stochastic integration methods: Comparison and application to reliability analysis, in *Proceedings of ASME Turbo Expo 2012*, Copenhagen, Denmark, June 11–15, 2012.
- [74] A. Palermo, J. Anthonis, D. Mundo, and W. Desmet, A novel gear test rig with adjustable shaft compliance and misalignments part I: design, in *Advances in Condition Monitoring of Machinery in Non-Stationary Operations*, pp. 497–506, Springer, 2014.
- [75] A. Palermo, L. Britte, K. Janssens, D. Mundo, and W. Desmet, The measurement of gear transmission error as an NVH indicator: Theoretical discussion and industrial application via low-cost digital encoders to an all-electric vehicle gearbox, *Mech. Syst. Signal Process.*, **110** (2018), 368–389.
- [76] K. Panetta, Top 10 strategic technology trends for 2018, <https://www.gartner.com/smarterwithgartner/gartner-top-10-strategic-technology-trends-for-2018/>, 2017.
- [77] C. Pettey, Prepare for the impact of digital twins, <https://www.gartner.com/smarterwithgartner/prepare-for-the-impact-of-digital-twins/>, 2017.
- [78] L. T. Pillage and R. A. Rohrer, Asymptotic Waveform Evaluation for timing analysis, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, **9** (4) (1990), 352–366.

- [79] Python.org, <https://www.python.org/>.
- [80] A. Quarteroni, G. Rozza, and A. Manzoni, Certified reduced basis approximation for parametrized partial differential equations and applications, *J. Math. Ind.*, **1** (1) (2011), 3.
- [81] A. Rasheed, O. San, and T. Kvamsdal, Digital twin: Values, challenges and enablers, arXiv preprint arXiv:1910.01719, 2019.
- [82] J. B. Rawlings and D. Q. Mayne, *Model predictive control: Theory and design*, 2009.
- [83] T. Reis and T. Stykel, Balanced truncation model reduction of second-order systems, *Math. Comput. Model. Dyn. Syst.*, **14** (5) (2008), 391–406.
- [84] T. Reis and T. Stykel, PABTEC: Passivity-preserving balanced truncation for electrical circuits, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, **29** (9) (2010), 1354–1367.
- [85] A. Rezayat, S. Shweiki, M. Park, D. Vivet, S. Donders, S. Flock, P. Jiranek, and T. Tamarozzi, A novel efficient high fidelity approach to gear contact simulation in multibody systems, in *Proceedings of the 6th European Conference on Computational Mechanics (ECCM 6) 7th European Conference on Computational Fluid Dynamics (ECFD 7)*, 2018.
- [86] D. J. Rixen, Dual Craig–Bampton method for dynamic substructuring, *J. Comput. Appl. Math.*, **168** (2004), 383–391.
- [87] J. Rommes, *Methods for Eigenvalue Problems with Applications in Model Order Reduction*, PhD thesis, Utrecht University, 2007.
- [88] J. Rommes, <http://sites.google.com/site/rommes> (2018-06-13).
- [89] J. Rommes and W. H. A. Schilders, Efficient methods for large resistor networks, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, **29** (1) (2010), 28–39.
- [90] M. Rosenblatt, Remarks on a multivariate transformation, *Ann. Math. Stat.*, **23** (1952), 470–472.
- [91] U. Rüde, K. Willcox, L. Curfman McInnes, H. De Sterck, G. Biros, H. Bungartz, J. Coronas, E. Cramer, J. Crowley, and O. Ghattas et al., Research and education in computational science and engineering, arXiv preprint arXiv:1610.02608, 2016.
- [92] J. Saak et al., <https://morwiki.mpi-magdeburg.mpg.de/morwiki> (2018-06-13).
- [93] B. Salimbahrami and B. Lohmann, Order reduction of large scale second-order systems using Krylov subspace methods, *Linear Algebra Appl.*, **415** (2005), 385–405.
- [94] R. R. Schaller, Moore’s law: past, present and future, *IEEE Spectr.*, **34** (6) (1997), 52–59.
- [95] W. H. A. Schilders, H. A. Van der Vorst, and J. Rommes, *Model Order Reduction: Theory, Research Aspects and Applications*, vol. 13, Springer, 2008.
- [96] E. Schrik and N. P. van der Meijs, Comparing two $y - \delta$ based methodologies for realizable model reduction, in *ProRISC IEEE 14th Annual Workshop on Circuits, Systems and Signal Processing*, pp. 148–152, 2003.
- [97] A. A. Shabana, *Dynamics of Multibody Systems*, Cambridge University Press, 2013.
- [98] B. N. Sheehan, Realizable reduction of RC networks, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, **26** (8) (2007), 1393–1407.
- [99] F. Simcenter Thermal, <https://www.plm.automation.siemens.com/global/en/products/simulation-test/thermal-analysis.html> (2018-10-02).
- [100] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*, Wiley, 2006.
- [101] T. S. Söderström and P. G. Stoica, *System Identification*, Prentice Hall International Series in Systems and Control Engineering, Prentice Hall, 1989.
- [102] T. Tamarozzi, G. H. K. Heirman, and W. Desmet, An on-line time dependent parametric model order reduction scheme with focus on dynamic stress recovery, *Comput. Methods Appl. Mech. Eng.*, **268** (2014), 336–358.

- [103] T. Tamarozzi, P. Jiranek, A. Rezayat, and S. Shweiki, An efficient hybrid approach to gear contact simulation in multibody systems leveraging reduced order models, in *International Gear Conference*, Lyon, France, 2018.
- [104] TIA, <https://www.siemens.com/global/en/home/products/automation/industry-software/automation-software/tia-portal.html> (2018-10-02).
- [105] R. A. Toupin and C. Truesdell, Principles of classical mechanics and field theory, in S. Flügge (ed.) *Encyclopedia of Physics*, vol. III/1, Springer, 1960.
- [106] H. Van der Auweraer, J. Anthonis, S. De Bruyne, and J. Leuridan, Virtual engineering at work: the challenges for designing mechatronic products, *Eng. Comput.*, **29** (3) (2013), 389–408.
- [107] H. Van der Auweraer, S. Gillijns, S. Donders, J. Croes, F. Naets, and W. Desmet, State estimation: A model-based approach to extend test data exploitation, in *Special Topics in Structural Dynamics*, vol. 6, pp. 119–128, Springer, 2016.
- [108] I. Varfolomeev and M. Luke, Consideration of fatigue crack growth aspects in the design and assessment of railway axles, in G. Hutter and L. Zybelle (eds.) *Recent Trends in Fracture and Damage Mechanics*, pp. 103–124, Springer, Cham, Heidelberg, New York, 2016.
- [109] H. Veendrick, *Deep-Submicron CMOS ICs: From Basics to ASICs*, second, Kluwer Academic Publishers, 1999.
- [110] M. Wendt, P. Li, and G. Wozny, Nonlinear chance-constrained process optimization under uncertainty, *Ind. Eng. Chem. Res.*, **41** (15) (2002), 3621–3629.
- [111] K. Willcox and J. Peraire, Balanced model reduction via the proper orthogonal decomposition, *AIAA J.*, **40** (11) (2002), 2323–2330.
- [112] P. Wriggers and G. Zavarise, Computational contact mechanics, in *Encyclopedia of Computational Mechanics*, 2004.
- [113] S. Y. Yoon, Z. Lin, and P. E. Allaire, *Control of Surge in Centrifugal Compressors by Active Magnetic Bearings: Theory and Implementation*, chapter 2, Springer, 2013.
- [114] A. Yousefi, B. Lohmann, J. Lienemann, and J. Korvink, *Nonlinear heat transfer modelling and reduction*, 06 2004.