NordDesign 2012
August 22 – 24, 2012
Aalborg, Denmark

Enabling simulation-based mechatronic design by shifting of activities

Fabio Dohr
Saarland University
dohr@lkt.uni-saarland.de

Michael Vielhaber
Saarland University
vielhaber@lkt.uni-saarland.de

Abstract

Simulation has been used in product development for years. Especially in mechatronics simulation is seen as a key factor due to the high complexity of both process and product. To develop a simulation-based design process for mechatronic systems it is essential to know which tools and techniques for modeling and simulation already exist. In this paper an overview of these tools and techniques is provided as well as a rating for their use in early phases. Finally some initial ideas are presented on how to integrate simulation early in the process.

Keywords: Mechatronic Design, Mechatronics, Simulation-based Design, Virtual Prototyping

Introduction

Since Tetsuro Mori created the word "Mechatronics" in 1969 and the emergence of microprocessors in the 1970s, mechatronics has become an indispensable part in engineering and ensures innovative high-tech products since the 1990s. The individual domains involved lead to a high level of complexity both regarding the product and the process – due to the numerous interacting components respectively disciplines. Especially in early phases this poses a challenge to engineers as they have to manage a large solution space as well as to interact with other domains. In addition market and customer situation demands reduced time to market and cost as well as better performing products with at least the same quality.

Virtual design tools have been helping to cope with these challenges for several years. Above all modeling and simulation offer huge potential to face these challenges by analyzing the entire system behavior. Particularly early phases benefit from the early comparison and verification of concepts through the use of system level simulation which does not only save time and costs but also improves product quality and functionality. In addition, more innovative concepts can be developed as engineers are no longer limited to familiar systems and concepts in order to reduce efforts for physical prototypes that are often based on previous products [1].

Although mechatronics is an interdisciplinary field consisting mainly of the domains mechanics, electronics and software engineering, most of the process models are still strongly focused and based on the mechanical domain which does not do justice to the great potential of mechatronic systems. As a consequence a simulation-based mechatronic design process should comprise all the disciplines and their techniques to enable successful mechatronic design. A review of common procedure models in the field of mechatronic design and the individual disciplines and requirements on a simulation-based design process is provided in [2].

It is undisputed that front-loading is a major factor of improvement to mechatronic design. But there are several aspects that distinguish activities in early phases from those in late phases:

- In early phases development is mainly focused on system level without going into detailed domain-specific aspects
- Design takes place on a rather abstract level, often without knowing the concrete implementation of the concepts but rather dealing with physical principles
- Since there is no detailed design yet, design parameters in early phases are very vague – especially for completely new designs

As a consequence it is not possible to simply use established computer-aided tools in early phases. Therefore the current state of the art is analyzed in the following section regarding computer-aided simulation technologies in mechatronic design as well as in the individual domains.

Based on this review an evaluation of those technologies is provided focusing on their applicability in early phases. Finally ideas are given in respect of how the early use of simulation can be achieved and how it will affect the mechatronic design process.

State of the art in modeling and simulation

This section deals with current simulation tools and techniques in the context of mechatronic design. The general steps of a simulation study are illustrated in Figure 1. It indicates that the prerequisite for every simulation is the creation of a valid model of the problem or system. There are numerous different modeling techniques both domain-specific and domain-spanning. Since these are crucial for the simulation the most important modeling techniques suitable for simulation are taken into account in the review as well as.

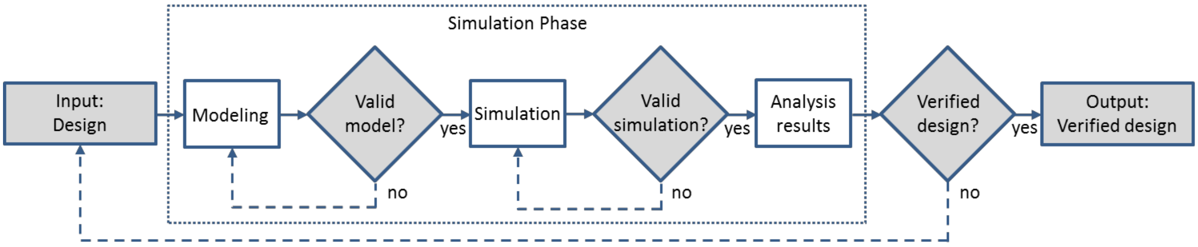

Figure 1 General steps of a simulation phase

Since the main focus of this paper is the use of simulation in early phases and on system level, tools which can only be used in phases where a certain degree of detail is reached – e.g. computer-aided optimization – as well as highly specialized tools are not further considered.

Mechanics

Modeling in Mechanics

In mechanical engineering several modeling techniques have evolved over time. In early phases theses are classical 2D or 3D sketches to visualize ideas. A more abstract way to describe a system in early phases are functional structures [3].

In order to describe the behavior the mechanical system is often idealized by objects like mass and bodies, springs and dampers based on which the mathematical description of the system can be derived. Those are usually called spring-mass-models.

In late phases computer-aided design (CAD) has been widely used for many years now and has accelerated and simplified product development in many ways. Nevertheless the use of CAD tools requires detailed information about the mechanical system and hence the use in early phases is rather difficult. It can even be counter-productive since the typical early phase activities – like concept development – are disregarded [4].

Simulation in Mechanics

There are three main areas of simulation techniques in mechanical engineering. The first one is the structural analysis of mechanical parts which includes the analysis under static as well as dynamic loads. A very mature and sophisticated technique is the finite element method (FEM). Since this numerical technique requires detailed geometric information in order to achieve convincing results, the use in early phases is very limited because of the abstract geometric level. In some cases it might be useful to estimate the structural behavior in early phases. In this case a simplified geometry can be analyzed by the FEM or by simple mathematical descriptions of the mechanics.

Apart from the structural aspects the dynamic behavior of the mechanical systems is essential – especially in mechatronic systems. For its analysis multibody simulation (MBS) has been established. Often the final geometry is simplified for this simulation as it is not as crucial as in structural analysis for example and thereby computational time can be reduced. This aspect can be used in early phases to analyze the basic behavior of mechanical concepts and thus compare them. Since the mechanical system – and thereby the dynamic behavior – usually establishes the foundation for a mechatronic system [5] the use of MBS during conceptual design offers great potential for frontloading.

The third main simulation technique in mechanical engineering is computational fluid dynamics (CFD). It is a numerical tool to study fluid problems and generally depends highly on the geometry – like FEM. Hence application during conceptual design is rather difficult and – if at all – only possible for analyzing phenomena with idealized geometry.

Electronics

Modeling in Electronics

Electronic development is mainly based on the reuse of standard components or circuits. Hence modeling here is different from mechanics. The common way to model an electronic circuit is through the use of a circuit diagram which uses standardized elements like resistors, capacitors or transistors. In digital electronics – which represent the main part of electronics in mechatronic systems – it is common to describe the behavior of a circuit on a higher level of abstraction. For this purpose there are mainly two hardware description languages: Verilog [6] and VHDL [7]. Since those support the conceptual design on an abstract level they are well-suited for early phase design. In order to model analog and mixed signals too, there are extensions of these languages, in fact Verilog-AMS [8] and VHDL-AMS [9].

Today electronic computer-aided design tools (ECAD) support designers in modeling electronics – among many other functionalities. However, only the conceptual aspects of ECAD systems are applicable in early phases.

Simulation in Electronics

As with the modeling techniques there are two main techniques for simulation. The first deals with the simulation of the circuit which includes the components and their behavior. Those circuits easily become very complex leading to large models and high computational efforts. In this case it is preferable to use the the second kind, logic simulation [10]. It operates on a more abstract level and does not consider any components. Hence it is very useful during conceptual design as it does not require any implementation. Based on the hardware description languages Verilog and VHDL there are several simulators existing to directly derive a logic simulation from the model.

Even when not dealing with mechatronics there is always a relation between mechanical and electrical engineering, e.g. concerning the housing. For instance thermal losses play a major role in electronics which leads to heat transfer as a major issue. Thus the mechanical design is usually optimized through the use of CFD. Since this requires geometry details as well as information about the electronic components and their power consumption, those simulations cannot be applied in early phases.

Software

In the context of mechatronics and also of this paper software comprises software in general – e.g. operating systems or graphical user interfaces (GUI) – and in particular the software component of the control system. Hence those two groups of software are examined separately below.

Modeling in Software

For software in general several modeling techniques have been established. To model software behavior Petri nets, finite state machines, state diagrams or flow charts are common examples. In recent years the Unified Modeling Language (UML) [11] has become a widely used language to model the structure and the behavior of software systems. It is mainly based on diagrams, such as those mentioned above. Since UML supports software development from the beginning until code generation it is well-suited for the use in early phases.

In control engineering modeling mainly makes use of block diagrams [12]. Each of these blocks contains the mathematical description of the component it represents. In this way the entire system can automatically be modeled by mathematical relations. Alternatively the system is directly modeled with mathematical description. The problem of the purely mathematical description is that the system structure and behavior is not obvious at first glance. Moreover detailed parameter information is required to use the model. Nevertheless in some cases it might be useful in early phases, e.g. to conceptually analyze a control algorithm.

Simulation in Software

As software – in contrast to mechanics or electronics – is no physical entity it is relatively easy to simulate software code. In general it can be tested by simulating the environment – i.e. the in- and outputs of the software – and tracing the reaction. This procedure is called model (MiL) and software in the loop (SiL) [13].

For simulating the system behavior with implemented control algorithms the simulation is executed based on the generated mathematical model – which is built up either automated from block diagrams or manually. This simulation can be either analytical or numerical depending on the complexity of the system. Usually simulation contains both discrete and continuous aspects.

In most of the cases mechatronic systems have strong requirements on timing. In such cases it is essential to test the software on the final hardware. This procedure is called hardware in the loop (HiL) [14].

Mechatronics

Since Mechatronics is the combination of the above-mentioned individual disciplines, all aspects can be described and analyzed with the domain’s technologies. But synergy only comes into effect when taking into account all domain-specific aspects simultaneously. For that reason there are several attempts integrating the above-mentioned techniques from the individual domains.

Modeling in Mechatronics

As shown above each discipline uses its own modeling language. But during the conceptual design of mechatronic systems it is crucial to have a common understanding to enhance cooperation and communication [15].

Bond graphs constitute one of these modeling languages. The advantage is that they can be applied to all kinds of physical systems and represent the topological and the computational structure of a system at the same time [16]. The disadvantage is that they are difficult to read [16].

Over the last few years the Systems Modeling Language (SysML), a general-purpose modeling language for systems engineering applications [17], has been established in the field of mechatronics. It is based on UML and offers several diagrams for the specification, analysis, design, verification and validation of systems [17]. Since SysML models cannot be simulated directly [18] the behavior of a mechatronic system cannot be directly analyzed.

Two further languages used to model mechatronics systems are VHDL-AMS and Verilog-AMS. Initially they are intended for the use in electronics. But since they can represent discrete-time behavior as well as continuous-time behavior – corresponding to digital respectivley analog electronics – they can be used for modeling of multi-discipline systems [19]. A comparison of VDHL-AMS and Verilog-AMS is provided in [19].

Another language which has become very popular in recent years is the object-oriented modeling language Modelica [20], intended for the modeling of complex physical systems. Modelica allows the description of the system directly by differential, algebraic and discrete equations. Alternativley most of the Modelica environments offer a graphical user interface and model libraries so that the system can be easily composed from its components, represented by those libraries. The system model then is automatically generated.

There are several attempts to combine the modeling languages SysML and Modelica – e.g. in [21] – to take advantage of the specific aspects of each one, for instance the comprehensibility of a SysML model and the feasibility of simulation of a Modelica model [18].

A comparison between Modelica and VHDL-AMS is provided in [22].

Simulation in Mechatronics

As already mentioned in mechatronics the different simulation tools and techniques are combined to analyze the system behavior. Especially on system level it is not beneficial to use the individual domain tools as they interrupt process and data flow [23]. Instead there are software tools which combine the different disciplines' simulation techniques. Hence the system behavior can be analyzed within a single tool and with the relations between the subsystems. The foundation in general is a mechanical system [12] – e.g. a multibody or hydraulic system – which is extended by control systems, sensors, actors etc. Examples for those software tools are Dymola [24] and CAMeL-View [25]. The models of the system which have been generated in those multidiscipline tools or in a multibody tool can be reused in the development process, e.g. for HiL-simulation as the model of the systems environment.

Summary

As shown in the sections above, there are numerous highly sophisticated tools and techniques for modeling and simulation of technical systems. Not all of these are applicable in early phase design. A comprehensible representation of the system is essential to enhance cooperation and communication between domains. Moreover the consideration of system level simulation and the ability to deal with vague information about design parameters are important criteria of early phase modeling and simulation. The results of the review above are summarized in table 1. "+" means that those criteria are fulfilled, "o" means that they are partially fulfilled and “-“ that they are not fulfilled and the tool or technique is not suitable for early phase design.

Table 1 Applicability of modeling and simulation tools in early phase design

Mechanics								Electronics					
Modeling				Simulation				Modeling			Simulation		
Sketches	Functional structures	Body-mass-model	CAD	FEM	MBS	CFD	Solving of Mathematical equations	Circuit diagram	Verilog (-AMS), VHDL (-AMS)	ECAD	Circuit simulation	Logic simulation	Mechanical simulation
+	+	+	-	o	+	o	+	+	+	o	o	+	-

Software							Mechatronics				
Modeling				Simulation			Modeling			Simulation	
Petri net, state diagram, flow chart …	UML	Block diagram	Mathematical description	Software in the loop	Control simulation	Hardware in the loop	Bond graphs	SysML, VHDL-AMS, Verilog-AMS	Modelica	Integrated software systems	Singe software tools: same as in disciplines
+	+	+	o	+	+	-	+	+	+	+	

Shifting of simulation activities to early phases and system level

There are two main benefits of simulation-based design:

- Early acquisition of system knowledge as shown in figure 2 which enables early detection of design faults and problems. Hence the second raise of system knowledge in traditional design – which is caused by late fault detection through physical prototypes – is eliminated.
- Saving of time and money since the number of physical prototypes and tests can be greatly reduced.

Figure 2 clarifies that the change from traditional analysis in late phases to simulation in early phases has to be achieved through a shift of activities. Primarily this means that testing with physical prototypes has to be substituted by simulation, within the limits of feasibility and efficiency. Hence validation and verification are shifted from the traditional prototyping phases to the domain design since simulations are executed in parallel with design activities. Especially in mechatronic design this procedure is necessary but not sufficient. The concept of a mechatronic system has to be developed during system design according to VDI 2206 [26] with all domains involved and communicating. Due to the large solution space for mechatronic systems a sound concept comparison and decision is essential. This can be supported effectively by system simulation. For this purpose a domain-spanning modeling technique as well as a system simulation environment are necessary.

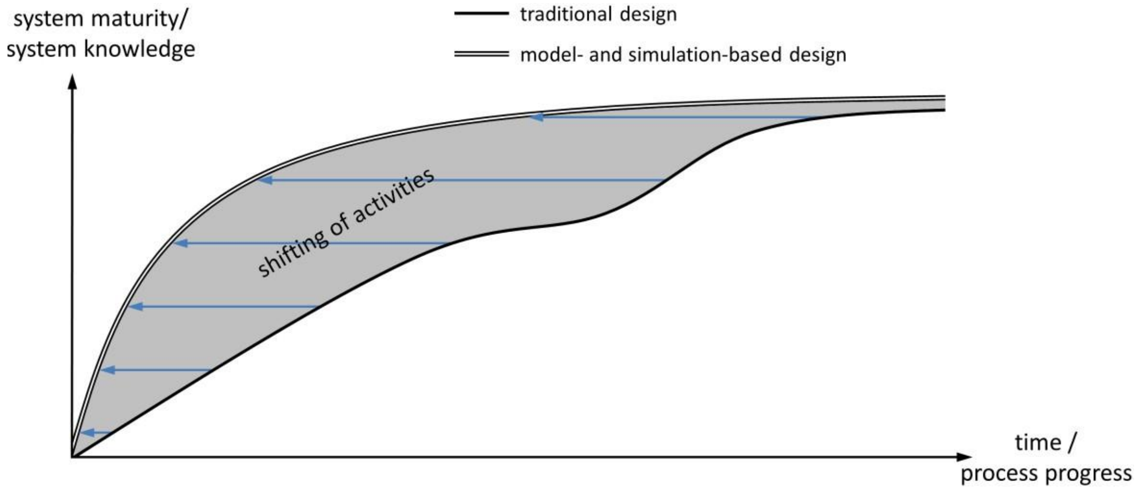

Figure 2 Shifting of activities to enable early system verification

One option is the use of a modeling language like Modelica to generate system models for simulation. Several benefits arise from this:

- The easy-to-handle graphical representation is comprehensible to all domains because the domain-specific aspects are represented in the domain-specific notation.
- The mathematical description is generated by the software in the background but can also be extended or modified should this be necessary.
- The use of component libraries assists the generation of concepts from abstract functional descriptions.
- Systems can directly be simulated in one tool and thereby concepts are easy to compare.

To integrate and align all activities between the individual disciplines the use of a SysML model over the entire process as a system basis is suggested. Thus the transition from system simulation – e.g. with Modelica – to the domain-specific tools can be supported by including results of the system simulation and derived requirements – like interfaces or dimensions – into the SysML model. The domain-specific tools or their users can use the information from the model and also integrate their results into the model. Thereby the interruptions in process and data flow can be bridged.

Conclusion and open issues

Simulation is one of the key factors for the design of mechatronic systems since it reduces development time and costs and improves system knowledge during early phases. Hence there is a large amount of research activity regarding modeling and simulation tools and techniques. The review in this paper shows that a lot of those are already very sophisticated and applicable. What is currently missing is a consistent process to enable simulation-based design. Further work will concentrate on process and method support of the initial ideas presented here for the early use of simulation and the use of a SysML model as a system foundation.

An open issue regarding simulation in early phases is the handling of uncertainty of design parameters. For this purpose the use of statistical methods is conceivable. This allows limiting the parameter space and conducting parameter studies to find the best design parameters.

Another issue is the content and the handling of the SysML system model. Support on which information has to be integrated and how this is handled has still to be provided.

References

[1] Becker, M. C., Salvatore, P., Zirpoli, F., “The impact of virtual simulation tools on problem-solving and new product development organization“, *Research Policy,* Vol. 34, Issue 9, pp. 1305–1321, November 2005

[2] Dohr, F., Vielhaber, M., “Toward Simulation-based Mechatronic Design”, accepted for International Design Conference – DESIGN 2012, Dubrovnik

[3] Pahl, G., Beitz, W., *“Konstruktionslehre“,* Springer, 2007

[4] Marion, T. J., Fixson, S. K., “The Benefits and Pitfalls of Digital Design Tools”, *Proceedings of the 18th International Conference on Engineering Design (ICED 11),* Design Society, 2011

[5] Schiehlen, W., “Multibody System Dynamics: Roots and Perspectives”, *Multibody System Dynamics,* Vol. 1, Issue 2, pp. 149–188, 1997

[6] Thomas, D. E., Moorby, P. R., *“The Verilog hardware description language”,* Springer, 2002

[7] Ashenden, P. J., *“The designer's guide to VHDL”,* Morgan Kaufmann, 2002

[8] Kundert, K. S., Zinke, O., “*The designer's guide to Verilog-AMS”*, Kluwer Academic Publishers, 2004

[9] Ashenden, P. J., Peterson, G. D., Teegarden, D. A., *“The systems designer's guide to VHDL-AMS”*, Morgan Kaufmann, 2003

[10] Pelz, G., *“Mechatronic systems”*, John Wiley and Sons, 2003

[11] http://www.uml.org, Last visit: 02.05.2012

[12] Wallaschek, J., “Modellierung und Simulation als Beitrag zur Verkürzung der Entwicklungszeiten mechatronischer Produkte“, *Simulation in der Praxis,* VDI Bericht 1215, pp. 35-50, 1995

[13] Schäuffele, J., Zurawka, T., *“Automotive Software-Engineering“,* Vieweg, 2006

[14] Abel, D., Bollig, A., “*Rapid Control Prototyping”*, Springer, 2006

[15] Möhringer, S., Gausemeier, J., “An Interface Specification for Principle Solutions Supporting the Cross-Domain Design of Mechatronic Systems”, *Proceedings of the 7th International Design Conference, Design 2002,* Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb, 2002

[16] Cellier, F. E., *“Continuous system modeling“*, Springer, 1991

[17] http://www.sysml.org, Last visit: 02.05.2012

[18] Huang, E., Ramamurthy, R., McGinnis, L. F., “System and simulation modeling using SysML”, Proceedings of the 2007 Winter Simulation Conference, Association for Computing Machinery, 2007

[19] Pecheux, F., Lallement, C., Vachoux, A., “VHDL-AMS and Verilog-AMS as alternative hardware description languages for efficient modeling of multidiscipline systems”, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 24, Issue 2, pp. 204-225, 2005

[20] http://www.modelica.org, Last visit: 07.05.2012

[21] Johnson, T., Kerzhner, A., Paredis, C. J. J., Burkhart, R., “Integrating Models and Simulations of Continuous Dynamics Into SysML”, *Proceedings of the 6th International Modelica Conference*, The Modelica Association, 2008

[22] Schwarz, P., Clauß, C., Haase, J., Schneider A., “VHDL-AMS und Modelica - ein Vergleich zweier Modellierungssprachen“, *Simulationstechnik – 15. Symposium ASIM 2001*, Society for Computer Simulation International

[23] Pahl, G., Beitz, W., *“Konstruktionslehre“*, Springer, 2007

[24] http://www.dymola.com, Last visit: 07.05.2012

[25] http:// www.ixtronics.com, Last visit: 07.05.2012

[26] VDI guideline 2206, *“Design methodology for mechatronic systems”*, VDI, 2004