INTERNATIONAL DESIGN CONFERENCE - DESIGN 2004
Dubrovnik, May 18 - 21, 2004.

DESIGN 2004

VIRTUAL PROTOTYPING OF SELF-OPTIMIZING MECHATRONIC SYSTEMS

J. Gausemeier, W. Müller, V. Paelke, J. Bauch, Q. Shen and R. Radkowski

Keywords: mechatronic systems, self-optimization, virtual prototyping

1. Design challenges of self-optimizing mechatronic systems

The term mechatronics expresses the mergence of technologies from mechanical engineering, electronics and control design in modern industrial products. These innovative products are based on the combined effect of mechatronic system elements (also referred to as "solution elements" [GL00]).
Future systems will comprise configurations of solution elements with an inherent partial intelligence. These components are relying on mathematical models of optimization as well as behavioral optimization, e.g., cognitive capabilities and case-based reasoning. We call this new class of systems “self-optimizing systems” and define the self-optimization of a technical system as an endogenous modification of a goal vector based on changing environmental settings which result from a goal-compliant autonomous adaptation of the structure, the behavior and the parameters of the system.
For this reason, self-optimization reaches far beyond conventional strategies for rules and adaptations. Self-optimization allows for manageable systems with inherent intelligence that are able to react autonomously and are flexible to changing environmental settings. The structure of self-optimizing systems is based on the structure of mechatronic systems, in that self-optimizing information processing is superimposed on the controlling mechatronic information processing. This applies at each level of the hierarchical structure that makes up a complex mechatronic system: e.g., mechatronic function modules (MFM) like an intelligent suspension strut, autonomous mechatronic systems (AMS) like a car, and networked mechatronic systems (VMS) like a vehicle convoy.
The complexity of self-optimizing systems and the necessity to efficiently analyze and explore a large number of potential behavior patterns requires the creation of new development methods and tools.
In this paper we introduce a new concept for the solution element based design and analysis of self-optimizing systems. Our tool uses virtual reality, simulation and visualization techniques to facilitate a more intuitive approach to virtual prototyping in the conceptual design phase. It also supports the communication within interdisciplinary design teams which are typically required in the development of self-optimizing mechatronic systems. For the analysis of self-optimizing systems discreet and continuous simulation models of different levels of abstraction must be assigned to the solution elements.
Our approach builds on the VDI design guideline for mechatronic systems (VDI recommendation 2206 [VDI03]) that defines a procedural model consisting of two main elements: The V-model on the macro level to structure the process, a general problem solution cycle on the micro level, and pre-defined process components for recurring design activities [GM03].

2. Solution-element based virtual prototyping

Modern mechatronic systems are based on the combined effect of solution elements. These elements encapsulate engineering expertise which allows proven solutions to be reused for new design tasks. Solution elements can be mechanical components as well as electronic assemblies and mechatronic

function modules. This concept applies to the customization and variant construction as well as to new designs [Rot00],[KBS+97].
The solution elements are provided to the engineers as part of a library. The necessary information required to develop a self-optimized mechatronic system is attached to the solution elements, e.g. the dimensions, shape and mass of a mechanical body. For actuators and sensors an admittance function that describes its behavior is also provided. We can assume that this information can easily be collected when a new solution element is developed by a component supplier.
Ideally, a new mechatronic system can be assembled from "ready-made" solution elements. However, some components might be inexistent or their attributes might be undetermined in the early design stage. Therefore, they cannot be selected directly from the library. Therefore, we offer two classes of solution elements to the development engineers. Concrete solution elements correspond to fully specified and commercially available elements and can be used directly. In addition, so called abstract solution elements are offered. These abstract solution elements also contain proven knowledge of engineering, but offer it only as a principle of solution without a detailed design, e.g. parametric components or even placeholders. They enable the smooth integration of solution principles established in design theory into the virtual prototypes created with our system.
In practice, procedures like designing the basic mechanical system are typically still done with paper and pencil. It is reasonable to demand, that even these early design steps should be supported by computer based tools so that the acquired results, e.g. physical and simulation models, can be used in later development steps. A solution element based virtual prototyping tool allows to rapidly compose and modify system prototypes and to analyze them through interactive simulation. This simplifies the system design in the early phases of development by enabling developers to rapidly compare and evaluate different approaches. As a first step, our virtual prototyping tool supports the solution element based design of a mechatronic product following the process illustrated in figure 1.

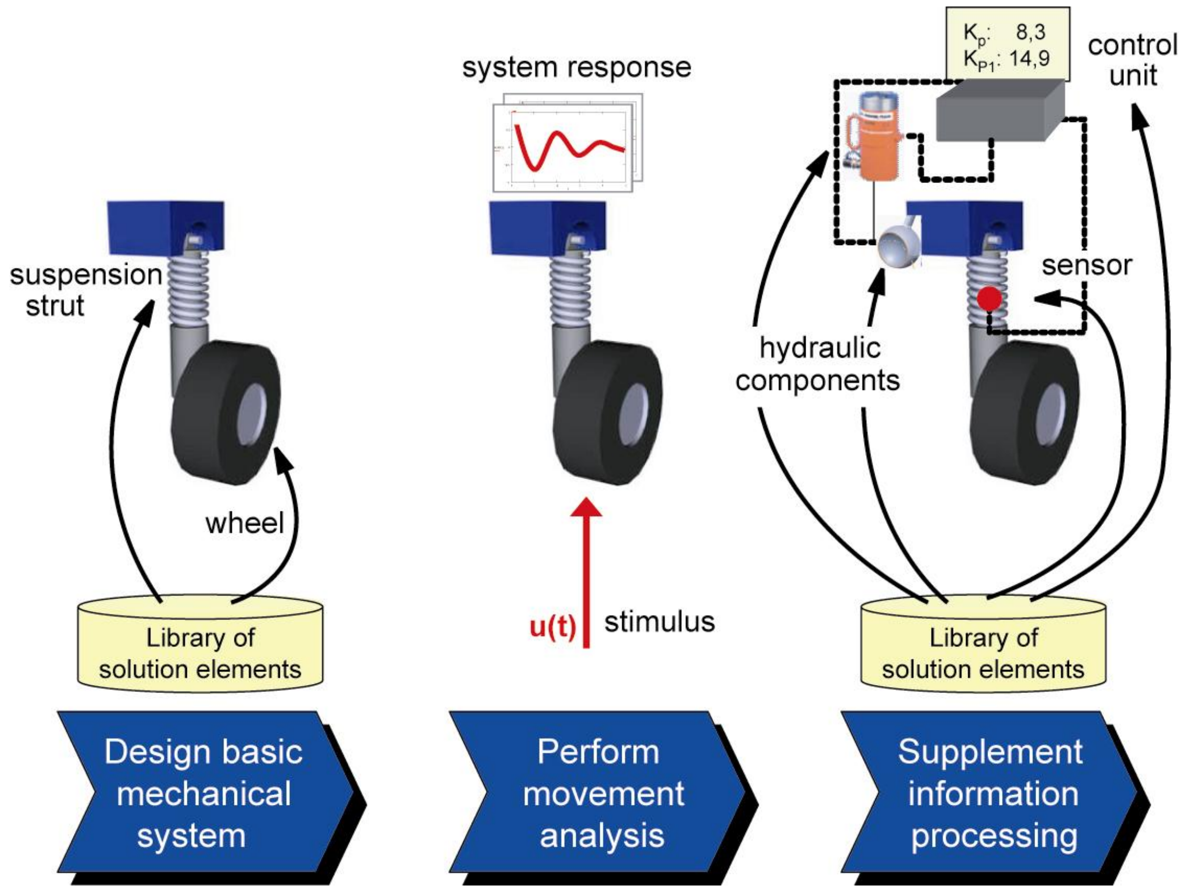

Figure 1. Process steps in solution element based design

In the following we describe the three process steps *Design the basic mechanical system*, *Perform movement analysis* and *Supplement information processing* and how they are supported by our tool.

2.1 Design the basic mechanical system

In this step the mechanical basic system is designed by combining solution elements from the library which are offered to the user in a construction set. Solution elements are represented graphically, so that developers are not required to remember abstract names or codes of the solution elements. Many functions like propulsion or measuring can be realized directly by concrete solution elements. For some Functions there may be no concrete solution element available or there is no exact specification required for the evaluation of the design concept. In these cases, e.g. for a desired mechanical frame or support structure, the missing element can be replaced by abstract parametrical solution elements like cylinders or bars.

The graphical representation of each solution element has defined connection points (CP) of different kinds. Connection points or ports correspond to the possible interconnections between components like a mechanical connection, material flow, energy flow and flow of information. Using these CPs solution elements can be connected graphically. After adjusting the parameters, e.g. of a mechanical joint, the corresponding active structure is updated by the system. In this way the developers can interactively and intuitively compose the system. Since for every solution element the corresponding function, physical model and behavior is specified in the library, we can derive a physical simulation model.

2.2 Perform movement analysis

In the next step the kinematic and dynamic behavior of the construction is tested in an interactive real-time simulation. First, the passive uncontrolled kinematic behavior of the prototype is tested. Ideally, this examination can occur immediately after construction, without the explicit production of a physical model. The system of mathematical equations, necessary for the calculation of a system behavior, was generated automatically during the composition of the mechanical basic system. The necessary information (e.g. the admittance functions) is located in the data model of the solution elements. For the examination of the behavior different test stimuli can be applied. A direct interaction with the developed solution is also possible. It provides immediate feedback on whether the provided kinematics shows the desired movement behavior.

To get the system response of the uncontrolled system we are using the VORTEX library from CMLabs Simulations.

VORTEX is a physics engine for real-time visualization and simulation, providing a set of libraries for robust rigid-body dynamics, collision detection, contact creation, and collision response. Vortex is well suited for 3D interactive simulations requiring stable and accurate physics. It allows programmers to simulate the natural behavior of objects in the physical world. Using the VORTEX libraries, it is possible to create interactive 3D simulations in which objects, particularly jointed objects subject to constraints, exhibit natural movement in all circumstances. [CML02]

2.3 Supplement information processing and design the controller

If a mechanical basic system was sketched which corresponds to the requirements of the designer, the information-processing components can be added. These can be solution elements with real-world geometry (e.g. processors, sensors and actuators) or pure functional components (e.g. block diagrams for control engineering). The controller for the system can be constructed with these block diagrams. Since 3D interaction with 2D diagrams provides no benefit but introduces many interaction problems, we have decided to represent the controller on a second 2D output device. The control engineer can sketch a controller interactively and examine the repercussion, i.e. the system answer of the sketched mechanical basic system, immediately. The change of parameters is reflected directly in a change of system behavior. The graphic representation of the resulting behavior also helps developers from other domains to grasp the implications of changes, facilitating interdisciplinary collaboration.

To design a controller and to simulate the components, e.g. actuators or sensors, we are working with several other simulation programs and libraries. To design and to simulate a continuous controller we are currently using MATLAB/SIMULINK from MathWorks. For discrete simulations we are using SystemC (for further information see [MRR03]).

3. Example

To evaluate our virtual prototyping tool, we have (re-)developed an innovative mechatronic system, a small parallel-robot, the TRIPLANAR (Figure 2). It contains all the elements of a typical mechatronic system. The TRIPLANAR was developed at the Mechatronic Laboratory Paderborn (MLap). The TRIPLANAR was particularly suited for our test, because it consists of a limited number of mechanical components but requires a very elaborate information processing architecture for the regulation of its movement. It consists of a working platform and three legs in which a planar-drive is mounted. Through the precise positioning of the planar-drives the adjustment of the working platform is effected. Sensors are integrated into the drives that deliver exact positioning data. For further reference see [Toe02].

Figure 2. The TRIPLANAR, developed at the MLap, Paderborn [Toe02]

Each solution element required for the TRIPLANAR was integrated into our virtual prototyping tool and is available for further development tasks.
Figure 3 shows how the assembly of the TRIPLANAR works. The connection points are used to create a mechanical connection (described in chapter 2.1). By selecting connection points on two of the elements and by initiating the desired joining operation a physical connection is established. In this manner we assembled the whole mechanical structure of the TRIPLANAR.
In the second step we tested the uncontrolled movement behavior of the resulting system. A prerequisite for this is that a correct simulation model is build up during the assembly of the TRIPLANAR. We are now able to test the behavior by means of direct user interaction. This means we can apply a force by pushing or pulling the graphical representation of the TRIPLANAR. The test shows the potential of our intuitive, interactive approach. In the third step we created a controller for the TRIPLANAR with MATLAB/SIMULINK (Figure 4).

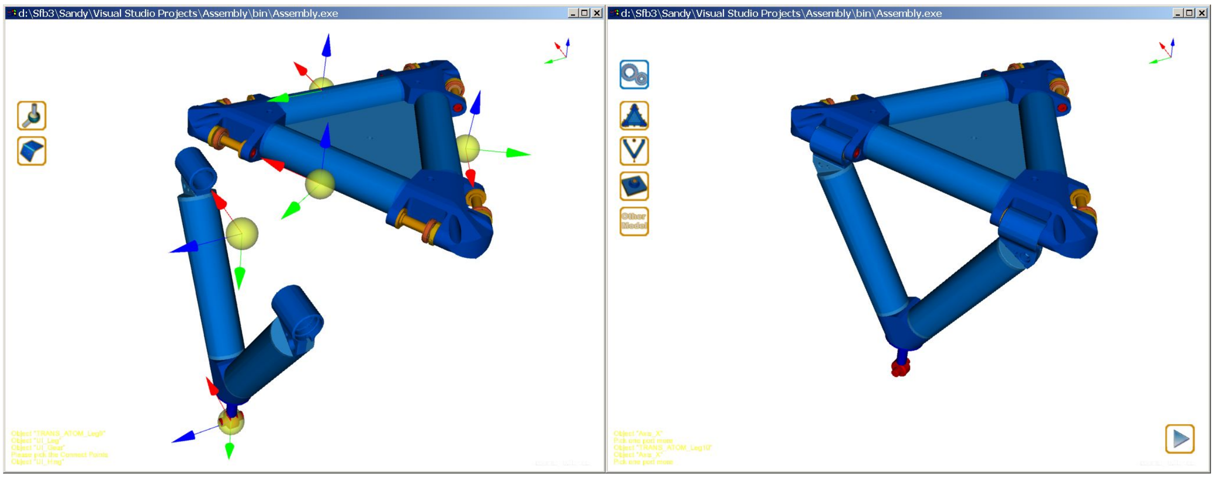

Figure 3. Assembly of the TRIPLANAR in the virtual prototyping environment

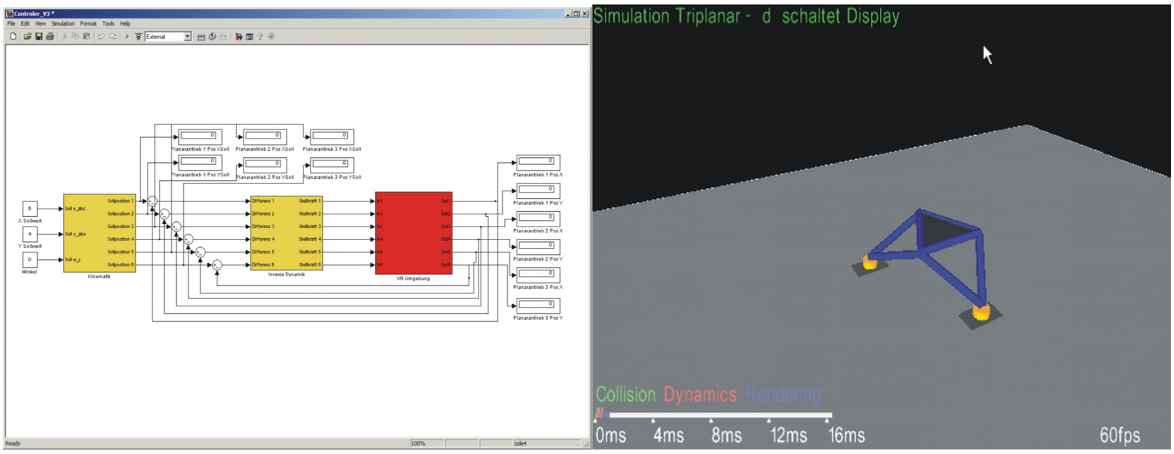

Figure 4. Control design for the TRIPLANAR with MATLAB/SIMULINK

The connection between SIMULINK and the VR-Environment is represented by a custom block diagram. By using the SIMULINK user interface the engineer has online access to all parameters in the controller. He can change the parameters and sees immediately the effect of the changes. In this way the controller can be tested and optimized interactively.

4. Advantages, benefits and outlook

Our virtual prototyping system supports the conceptional design phase of mechatronic systems and thus helps to prevent early and costly errors in the development process. Key benefits are:

- A principal proof of concept can be achieved at an earlier design stage using the simulation based virtual reality environment
- Catalogs of solution elements representing engineering knowledge support reuse of proven solutions
- Simulation based animation of the prototype enables engineers to explore the behavior of complex self-optimizing system in a large number of different operating situations in a rational way

These insights gained during the early analysis phase provide a foundation for a detailed analysis later in the design process and can help to reduce the number of analysis runs with specialized tools later on. In addition the virtual prototyping system also provides an efficient communication tool for interdisciplinary design-teams and could be extended into an integration platform for other analysis and simulation tools in the future. In the next phase we plan to run a detailed comparison of the simulation results from the TRIPLANAR with measured results from the real TRIPLANAR and to conduct studies to improve the usability of the 3D environment.

5. Related Work

In [DPK+98] a software architecture for the component oriented simulation of mechatronic systems is presented. The description of the system is carried out on subsystem level (as a hierarchical function graph) on component level and on process level. [SPK +00] describes how to automatically derive mechanical behavior models starting from shape oriented CAD models for the component oriented simulation. [GGR03] describes the combined use of a large format flat panel display and a projection system for layout planning and visualization of production systems. The "Virtuelle Werkstatt” (virtual workshop) [BJL +02] is a development platform for virtual construction with the help of gestures and speech in a Virtual Reality environment.

Acknowledgment

This work was developed in the course of the Collaborative Research Center 614 – Self-Optimizing Concepts and Structures in Mechanical Engineering – University of Paderborn, and was published on its behalf and funded by the Deutsche Forschungsgemeinschaft.

References

[BJL+02] Biermann, P.; Jung, B.; Latoschik, M.; Wachsmuth, I.: Virtuelle Werkstatt: A Platform for Multimodal Assembly in VR. In Proceedings Fourth Virtual Reality International Conference (VRIC 2002), Laval, France, 19-21 June 2002, 53-62.

[CML02] CMLabs Simulation, Inc: VORTEX Developer Guide, A Manual for the VORTEX Toolkit, V2.0.1, Montreal, 2002

[DPK+98] Diaz-Calderon, A., Paredis, C. J. J., Khosla, P. K.: "A Modular Composable Software Architecture for the Simulation of Mechatronic Systems," in Proceedings of DETC98, Computers in Engineering Conference, paper no. DETC98/CIE-5704, Atlanta, Georgia, September 13-16, 1998.

[GGR03] Gäse, T., Günther, U., Riegel, J: Interaktive Layoutplanung und Visualisierung von Produktionssystemen; Proceedings of Simulation und Visualisierung 2003, Otto-von-Guericke-University, Magdeburg, März 2003

[GL00] Gausemeier, J.; Lückel, J. (Hrsg.): Entwicklungsumgebungen Mechatronik - Methoden und Werkzeuge zur Entwicklung mechatronischer Systeme, HNI-Verlagsschriftenreihe, Bd. 80, Paderborn, 2000.

[GM03] Gausemeier, J.; Möhringer, S.: Ein Vorgehensmodell für den Entwurf mechatronischer Systeme in: Intelligente mechatronische Systeme, Tagungsband zum 1. Paderborner Workshop Intelligente mechatronische Systeme, Paderborn 2003

[KBS+97] Kallenbach, E; Birli, O.; Saffert, E.; Schäffel, C: Zur Gestaltung mechatronischer Produkte; in: Tagung Mechatronik im Maschinen- und Fahrzeugbau, Möhrs, 10. -12.März 1997, VDI-Berichte 1315, VDI-Verlag Düsseldorf, 1997, S.1-14.

[MRR03] Müller, W.; Rosenstiel, W.; Ruf, J.:SystemC, Methodologies and Applications, Kluwer Academic Publishers, Bosten, 2003

[Rot00] Roth, K.: Konstruieren mit Konstruktionskatalogen, 3. Auflage, Sprinter Verlag, Berlin et al, 2000

[SPK+00] Sinha, R.; Paredis, C.J.J.; Khosla, P.K.: "Integration of Mechanical CAD and Behavioral Modeling," in Proceedings of the International Workshop on Behavioral Modeling and Simulation, Orlando, Florida, November 18-21, 2000.

[Toe02] Toepper, S.: Die mechatronische Entwicklung des Parallelroboters TRIPLANAR. Dissertation, Universität Paderborn, Mechatronik Laboratorium Paderborn, 2002

[VDI03] VDI, Verein Deutscher Ingenieure (Hrsg.): Entwicklungsmethodik für mechatronische Systeme, Richtlinie VDI 2206 (Entwurf), Beuth Verlag, Berlin, 2003

Jochen Bauch, Mr
University of Paderborn, Heinz Nixdorf Institute / RIP
D-33102 Paderborn, Fuerstenallee 11, Germany
Telephone: +49 5251 606228, Telefax: +49 5251 606268
E-mail: jochen.bauch@hni.uni-paderborn.de