2nd International Conference on System-Integrated Intelligence: Challenges for Product and Production Engineering

Methodology for the identification of potentials for the integration of self-optimization in mechatronic systems

Harald Anacker*a, Roman Dumitrescu*, Jürgen Gausemeierb, Peter Iwanekb, Thomas Schierbaumb

a Project Group Mechatronic Systems Design, Fraunhofer Institute for Production Technology, Zukunftsmeile 1, Paderborn, Germany
b Heinz Nixdorf Institute, University of Paderborn, Fuerstenallee 11, Paderborn, Germany

Abstract

Mechatronics is the close interaction of mechanics, electronics, control engineering and software engineering in order to achieve improved systems behavior. Due to the advancement of information and communication technologies, more and more software solutions will enable more functionality and new customer benefits. The term self-optimization characterizes this perspective: the endogenous adaptation of the system’s objectives due to changing operational conditions. Thus, a major challenge is to consider the possibility of the integration of self-optimization in mechatronic systems and to show a way of this integration, especially for businesses from machinery and plant engineering. Hence, we developed a methodology for the identification of potentials for the integration of self-optimization in mechatronic systems, which support engineers in a systematic way.

© 2014 The Authors. Published by Elsevier Ltd.

Keywords: Mechatronics; model-based systems engineering, self-optimization

1. Introduction

The increasing integration of information and communication technology in the field of engineering implies a large potential for technical innovation. The term “mechatronics” expresses this progress. This term refers to the
symbiotic cooperation of mechanics, electronics, control and software engineering in order to improve the behavior of a technical system. The design and identification of potentials to optimize such systems is an interdisciplinary and complex task. Therefore, effective and continuous cooperation and communication between developers from different disciplines during the whole development process are required [1]. The perspectives for mechatronic systems go far beyond current standards [1]. Keywords such as “Things that Think”, “Cyber-Physical Systems”, “Industry 4.0” or “Self-Optimization” express this perspective of intelligent technical systems [2,3,4]. The integration of methods and technologies of non-technical disciplines, e.g. cognitive science or neurobiology characterizing such systems [5,6]. Based on the implementation of Self-X properties or intelligence for monitoring and controlling processes, these systems are adaptive, robust, foresighted and user-friendly.

Even now, mechanical engineering and related industrial sectors are on the way to intelligent technical systems. In order to provide businesses (particularly small and medium-sized businesses in the context of machinery and plant engineering) from different branches a guide for identifying potentials for self-optimization and developing self-optimizing systems, we have developed and validated a four-stage approach. This paper shows the individual phases of the methodology and the activities, which the developers have to perform. This paper is organized as follows. In Sect. 2 a brief overview of self-optimization and the design of self-optimizing systems is given. Section 3 introduces the challenges for the realization of self-optimizing systems. Section 4 introduces the methodology for the identification of self-optimizing potentials for mechatronic systems. Additionally the paper shows in Sect. 5 the usage of the methodology in an industrial application example. We will conclude the paper in Sect. 6.

2. Self-Optimization

2.1. Introduction of Self-Optimization

Future mechatronic systems are systems with inherent partial intelligence. These systems are also called self-optimizing systems. Self-optimization describes the ability of a technical system to adapt its objectives endogenously (e.g. “max. productivity”, “max. reliability”, “max. energy consumption”) regarding changing influences and thus adapt the system’s behavior [1]. Thus, self-optimization goes beyond the familiar rule-based and adaptive control strategies.

Generally, the self-optimization takes place as a process that consists of the three following actions, called the self-optimization process. Figure 1 shows the self-optimization process and some relevant key topics in the context of self-optimization. The self-optimization process consists of the three action: analyzing the current situation (Which influences from the environment, the user or the system itself occur?), determining the system’s objectives (Are the current objectives still relevant for the new situation?) and adapting the system behavior (Which parameters in the system should be changed in accordance to the changed objectives?). The self-optimization process takes place by performing the three actions repeatedly by the system [1,7,8].

Fig. 1. Self-Optimization Process and some Key Topics (in accordance to [1].
The presented key topics form the basis for the realization of self-optimizing systems. For example, machine-learning methods can be used to analyze patterns in the influences and thus improve the behavior or methods for the multi-objective optimization can be used to determine optimal compromises between concurrent objectives, like “max. productivity” and “min. energy consumption”. These key aspects built the basis for the development of solutions to realize self-optimizing systems. To identify potentials for self-optimization in mechatronic systems the developer has to work out the demand for these specific solutions in the context of self-optimization.

2.2. Conceptual Design of Self-Optimizing Systems

The development of self-optimizing systems is challenging due to the integration of partial intelligence and the involvement of different domains such as mechanical, electrical, software, control engineering, mathematics and artificial intelligence. Therefore a discipline-spanning description of the system is needed which supports the common understanding of the development task and the system between of the developers of the different disciplines involved [1]. For the discipline-spanning description of the system, the specification technique CONSENS is used, which has been developed within the CRC 614 [10]. The principle solution describes the basic structure (e.g. components of the system and interactions between them), operational mode of the self-optimizing system, and its desired behavior. The principle solution forms the basis for the communication and cooperation of the disciplines involved (e.g. mechanical and software engineering) in the course of the further discipline-specific design and development. Additionally the models can increase the understanding in the communication of OEMs and suppliers [11]. The description of the principle solution of self-optimizing systems consists of eight interrelated aspects. These aspects are requirements, environment, system of objectives, application scenarios, functions, active structure, shape and behavior [1,12]. The aspects are computer-externally represented as partial models. For further informations about the specification technique, we refer to CONSENS [1].

3. Challenges for the Realization of Self-Optimizing Systems in Machinery and Plant Engineering

The realization of self-optimization is challenging, due to the involvement of different disciplines and the increasing system complexity. Especially businesses in the context of machinery and plant engineering need an approach, which supports the identification of potentials for self-optimization in their familiar mechatronic systems. Therefore, the businesses need a methodology to identify the demand to improve the current systems and to show solutions, how to satisfy the identified demand. In the following, we will show the state of the art in three different areas, which are relevant for such a methodology. Firstly, we will show some guidelines, about the development of systems. Afterwards the next paragraph show some model-based approaches for the analysis of systems, to identify in general the demand for changes in systems. Subsequently we conclude some approaches for the identification of measures in the proper sense of solution patterns to improve systems.

Guidelines for the Development of Technical Systems: For the systematic design of intelligent technical systems, the developers need a guideline to select the suitable procedures, methods and tools for the individual design task. There are many design guidelines, which exist within the different disciplines involved in the design of intelligent technical systems. These are for example the VDI 2422 for the development of electrics and electronics [13], the VDI 2221, which is essential for the development of mechanics [14], for e.g. the V-Modell XT, which is significant for the development of software [15] or the reference process for the development of self-controlled logistic processes (ALEM-P) [16]. For the development of multidisciplinary mechatronic systems, multidisciplinary guidelines were needed. Therefore, the VDI 2206 were developed [17], but also other design guidelines like the V-Modell by Bender [18], the W-Model of Anderl et al. [19] or the extended V-Modell for Model-Based Systems Engineering [20]. To address the specific requirements for the development of self-optimizing systems, the researchers within the CRC 614 developed a new approach. The result is a reference process, which qualifies developers to realize self-optimizing systems independently [1]. This reference process focuses strongly on the development of new systems; the process does not address an integration of self-optimization into existing systems explicitly. Approaches exist, which suggest a systematic integration based on the identification of self-optimizing potentials, as described in [21] and [22]. In the industrial practice we noticed, that the approaches do not match very well, especially for the use in businesses in the context of machinery and plant engineering. The developers in the
mentioned businesses for usual do not have the time, to perform those “academic” approaches on their own. Particularly, the integration of the total self-optimization process does not seem to be optimal for all regarded systems. Thus, we identified a need for the further development of those methods to identify potentials for self-optimization.

**Model-Based Analysis:** For the systematic identification of potentials for self-optimization, it is very helpful to work with different models of the systems to identify the demand to change or for further development the current system. Based on models a better as-is analysis can be performed because all involved experts have the same understanding of the system of interest [1]. To get many potentials for the optimization it is useful to involve experts from the different business divisions, like the service, the mechanic engineering or the automation. Thus, there is a high need to use models or descriptions of the system, which are usually discipline-spanning. The discipline-spanning description of the whole system (Model-Based Systems Engineering) can be performed by using modelling languages e.g. like SysML (Systems Modelling Language) [23], UML (Unified Modelling Language) [23], the Autonomous Logistics Engineering Methodology-Notation (ALEM-N) [16] or CONSENS [23]. This description of the system, allows further analysis of different aspects, like a dependability analysis, an analysis of the economic efficiency or the analysis of dynamic behavior [1]. A first step to identify potentials and thus the demand for the integration of self-optimization into technical systems is to identify the current weak points and disturbances of the current system. Based on the identified weak points, it is possible to analyze capabilities of the self-optimization to detect those disturbances during the operation of the system, to handle with those or generally to avoid them. Dumitrescu [6] and Pook [21] describe possibilities for the system analysis, in which concurrent objectives are identified for e.g. as basis for the further integration of cognitive functions to realize self-optimizing systems. By performing these methods, the developers mainly just identify potentials for the second phase of the self-optimization process (“determining system’s objectives”). Many approaches exist, which already focus on the identification of faults and disturbances for the analysis of the safety and reliability of the system, like the FMEA (Failure Mode and Effects Analysis) or FTA (Fault Tree Analysis). The developers even have the possibility to perform both analysis, based on models like CONSENS [1], SysML [24], Modelica [25] or UML [26]. However, the analysis are often just used for an as-is analysis of the system. The possibilities of the FMEA or FTA to identify new potentials and the demand for the integration of self-optimization are not considered.

**Solutions Patterns to Improve the System:** Already Hubka described the advantages and effects, which results for developers by using so called solution libraries [27]. Developer can use these libraries to search and find solutions to solve challenges, which occur during the development process. These advantages are for example: an accelerated development process, the avoidance of extra work by reusing known solutions or the improved communication in businesses based on the solutions [27]. For the systematic description of solutions, Alexander developed a definition of so-called solution patterns [28]. Generally, a pattern describes a recurring problem and the core of a solution to that problem [28]. For product development, there are solution patterns, which rely on physical effects (e.g. by Pahl et al. [29]), patterns, which serve information processing (e.g. by Gamma et al. [30]) or patterns, which focus improvements by inspection of possible solutions of suppliers [31,32]. There are just few approaches (e.g. by Schmidt [33] and Dumitrescu [6]) of solution patterns, which consider the paradigm of self-optimization, although the research of Dumitrescu can be described as further development of the approach by Schmidt. However, these approaches are not sufficient for the industrial use, because they require good knowledge about their usage and the system analysis mostly focus on the identification of potentials for optimization in the second phase of the self-optimization process.

The analysis of the state of the art shows the need for action to develop a methodology for the identification of potentials for the integration of self-optimization. There already are approaches to analyze the system in a systematic way, like the model-based FMEA or FTA, and thus find the demand to improve the system for e.g. by the integration of self-optimization, but they are not integrated in a holistic methodology. In addition, the existing research is well, but especially for the machinery and plant engineering industry, they are too complex and not suitable; an adaption has to take place. To achieve the acceptance in this context, it is promising to combine the

* Statement of some developers of businesses within the Leading-Edge-Cluster it’s OWL.
methodology (within the meaning of innovation management) with topics like e.g. the quality (FMEA etc.) and knowledge management (solution patterns etc.) [34,35]. Thus, the combination can generate synergies, and the extra work for the identification of potentials for self-optimization are held minimal and suitable.

4. Methodology

The result of the development is to achieve a system without weak points. In order to reach this goal, the developer must identify and eliminate the contradictions and weak points of the system. The paradigm of self-optimization is a possibility to dissolve contradictions and eliminate weak points. However, it is important to examine whether self-optimization is suitable for each specific case. The steps synthesis, analysis and evaluation of the system are continually executed during the development of the principle solution. In order to provide businesses from different branches (focus on machinery and plant engineering) a guide for identifying and developing optimization potential by using self-optimization, we have developed and tested a four-stage approach (see Fig. 2).

The methodology consists of a procedure model, methods for the identification of demands to change the current system (e.g. methods in the context of quality management like the FMEA based on CONSENS) and an approach in the context of knowledge management (solution patterns for self-optimization). The procedure is divided into the system modelling phase, the system faults analysis, the technical identification and the adaptation definition.

<table>
<thead>
<tr>
<th>phases</th>
<th>activities</th>
<th>results</th>
</tr>
</thead>
<tbody>
<tr>
<td>system modelling</td>
<td>• use cases • environment model • active structure • activities – process sequences</td>
<td>multidisciplinary system description</td>
</tr>
<tr>
<td>system faults analysis</td>
<td>• system faults • causes of the system faults • analyzing of system faults propagation</td>
<td>system faults strings</td>
</tr>
<tr>
<td>technical identification</td>
<td>• sensors to detect the causes of the system faults • identification of physical relationships in case of using secondary sources</td>
<td>technical discovery opportunities</td>
</tr>
<tr>
<td>adaptation definition</td>
<td>• analyzing and defining opportunities to minimize disruption (by using self-x-algorithm) • assessment</td>
<td>self-optimizing potentials</td>
</tr>
</tbody>
</table>

![Fig. 2. Procedure model for the identification of self-optimizing potentials.](image)

**Phase 1 System modelling:** The goal of system modelling is to map the basic structure and basic function of a system. For this purpose, the developers create the system description, by using the specifications technique CONSENS. It is a mechatronic system description in order to identify and depict the interactions between the environment and the system as well as the interactions within the system being considered [1,12]. The representation is created using so-called system elements. These represent modules, components, or software components. Material flows, energy flows, and information flows as well logical relations describe the interactions between the system elements. The partial model application scenario, environment model in combination with the active structure as well as the process sequence work iteratively. The mechatronic view on the system unites the design-related components with sensory, actuating elements, and the information processors in a diagram. It is ultimately the basis for the other steps for the integration of potentials for self-optimization.
Phase 2 System faults analysis: In this phase, the developers perform a system fault analysis (e.g. by using a combination of the FMEA (Failure Mode and Effects Analysis) and FTA (Fault Tree Analysis) [1]), based on the mechatronic system description. Possible faults, their causes and relationships in the form of failure chains can be shown in a diagram and thus systematically documentation is performed. Then the developer will evaluate and classify the detected faults according to the severity. The result is a weighted enumeration and representation in a diagram, which describe the demand to change the current system, for example by the integration of self-optimization.

Phase 3 Technical identification: For this phase, the developer needs a detailed description of faults and their origins. Based on this, the developer examine technical scenarios in order to detect the faults. These may be inferences from disruptions in the form of physical connections, an evaluation of the sensors, or the potential for additional sensors etc. The identified technical discovery opportunities built the potentials to improve the first phase of the self-optimization process: the analysis of the current situation. Therefore, the developer can choose known sensor systems or solution patterns in the context of self-optimization e.g. “Data-Mining”. Hereto firstly, the developer should define the potential system-upgrade in an abstract way (referring to EhrLenspiel [36] and Doerner [37]). Thus, the developer will define the function e.g. “detect occurring disturbance” and afterwards search for possibilities to realize this function. In accordance to the circumstances a suitable solution pattern will be selected (Is there the possibility to integrate sensor systems? Is there maybe data, which allow an extraction of informations about the disturbances?).

Phase 4 Adaptation definition: Building up from the former base, the developer perform the phase adaptation definition to minimize disruptions and define the system optimization. The adaptations provide information where self-optimization is possible and potentially useful in the system. The procedure is close to the phase 2: the concrete problem (e.g. missing control strategies for specific task) will be defined in an abstract way, by defining functions and afterwards solution patterns in the context of self-optimization will be searched (e.g. “Multi-Objective Optimization” to realize the function “situation-specific prioritization of objectives”). Based on this, the developer can perform an analysis and an evaluation on all formerly defined adaptations.

5. Application Example: Separator

In order to explain our results in the following, we use an existing separator and its behavior by processing one specific medium. The task of a separator is the mechanical separation of a raw product. The working principle is based on the centrifugal force and the different mass inertia (two phases) within the raw product (see Fig. 3). The section where the different phases collide is called the separation-zone. A large number of inserts (plate package) increase the forming of the different phases on the contact surface. Therefore, the separation process becomes more efficient. The separator eject the solid load laterally in an inconsistent manner. The separated liquid phase flows to the midpoint of the rotation. The liquid phase can be dispersed externally by the help of a paring disc.

Phase 1 System modelling: In this paper, we will explain the most representative activities of the first phase: the environment model and the active structure. Integrated environment and active structure: The specification technique CONSENS offers an aspect for the integrated view of the environment and active structure of the considered system. For example, there are many interrelations between the system and its environment. Therefore, it is important to analyze the environment of the system to ensure that the final system will work properly in it, without any restrictions caused by interactions not considered. In particular, the developer can describe other elements of the environment (e.g. the user, other technical systems or the underground) and their interrelation with Fig. 3. Working principle of a separator; for more details see [38].
the system. The part of the active structure defines the internal structure and the operational mode of the system. It
describes system elements (e.g. chosen solutions), their attributes as well as the relationships between system
elements. Depending on the level of concretization, system elements may be described abstract (e.g. temperature
sensor) or specific (e.g. resistance thermometer). If necessary, it is also possible to model elements of the
environment (e.g. user) and their interaction with elements of the system (e.g. interaction of the user with the
human-machine-interface of the system). Figure 4 visualizes an excerpt of the environment and active structure of
the separator.

In particular, it shows that the separator feedings are compounds, raw product and water. The interaction with the
operator is realized via a separate control unit. There are also interactions between the separator and the ground:
radial force as well as the disturbing oscillations, which are transmitted. The active structure consists of system
elements such as the plate package, the piston valve and paring disc. To show system elements of the dome at the
same hierarchy level as other modules, the developer can use logical groups. The dome consists of different fishing
tools for flushing water, control water and solid load. The raw product reaches the drum over the feed. In the drum,
the separator separates the raw product into a liquid phase (product) and solid load. The separator transports the
liquid phase through the plate package and reaches the discharge channel over the allocator gorge. The fishing tool
captures the solid load. An external control unit controls the separator itself. It is part of the production control
system.

![Fig. 4. Integrated Environment and Active Structure of the separator (excerpt).](image-url)
**Phase 2 System faults analysis:** As already explained in Sect. 3, one uses a modified version of the failure mode and effects analysis (FMEA) to identify potentials for self-optimization in a systematic manner. Figure 5 shows an excerpt of the FMEA. We subdivided the FMEA into the following columns: position in an active structure (direct allocation of entries into the active structure is done for a better overview), system faults, system faults propagation, causes of the system faults, preventive measures, reactive measures and technical discovery opportunities. In order to manage the complexity of the system, further sub-division takes place during the course of the program in the controller. Figure 5 presents an excerpt from the process sequence “start-up phase”. Furthermore, preconditions are met which are not considered in the framework of the implemented FMEA as disturbances. Some preconditions are (successfully) checked: electrical power supply, construction, valves for cooling water, air, etc. In addition, faults via physical damage by the identification of potentials for self-optimization cannot be accounted for using the self-optimization algorithms. These are examined in a separate component of FMEA. For example, the undertaken analysis has shown that the separation performance in the considered system significantly depends on the raw-product intake in the separator. A common operational error is the improper method of feeding the raw-product into the allocator. This occurs, when there is no fluid level below the feed pipe where the raw-product flows into the separator. The main reason for this is the superior pipe system. Normally a separator is part of the overall production process, which may lead to inconsistencies (volumetric flow and mass flow at entry). In particular, during an impulsive intake which may lead to degradation, inefficiency and loss of product. Therefore, the goal of optimization is to recognize the changing conditions in the process and to react autonomously to ensure a continuous feed to the separator.

**Fig. 5. Modified Failure Mode and Effects Analysis with corresponding phases (excerpt).**

**Phase 3 Technical identification:** The prerequisite for an autonomous reaction to the changing environment conditions are suitable sensors that detects the relevant process parameters (first phase of the self-optimization process). The relevant parameters in our application example are usually located in the drum unit of a separator, which lead to the enormous challenge of developing and integrating the sensors. The reasons for this are the process related properties of the separator such as the high speed of the drum unit which may process food or pharmaceutical products, the limited space inside the drum unit and the durability of the materials used in all the moving components, etc. However firstly, the developer should define this as a function: “detect discontinuous fluid flow”. Practical experiences shows that a more abstract description is often better to find self-optimizing-specific patterns, like “detect parameters”. Afterwards the developer would search for solution patterns to realize the function, e.g. by using conventional solutions patterns like “Pressure Sensor” or self-optimizing-specific patterns like “Learned Models”. By choosing the solution pattern “Learned Models” (can be later concretized e.g. by the
extreme learning machines), the developer can proof the ability of learning coherences of parameters to extract informations about the discontinuous feed of the raw product [39]. To find this, the developer needs a practical description of patterns. For example, they can characterize by their inputs (e.g. for using the pattern “Learned Models” a prototype of the system is needed to acquire the relevant parameters) and outputs (e.g. evaluated model, which describes the coherencies between parameters) as well as by a number of criteria, such as the assignment to the self-optimization process (e.g. pattern for the analysis of the current situation) or functions, which could be realized by the consideration of the pattern (e.g. “detect parameters”). In addition, useful documents (e.g. procedure “to integrate” pattern into the system), concrete methods within this pattern (e.g. extreme learning machines) or literature are described as well. In the chosen fault in Fig. 5, the analysis of the continuous feed by using the pattern “Learned Models” is shown (generated in creativity workshops by considering the patterns), because an integration of sensors into the separator is more challenging.

Phase 4 Adaptation definition: After the identification of the opportunities to detect, the faults or disturbances of the current system, in this phase the developer will define the preventive and reactive measures. For example, we spotted that in consistent feeding of the raw product can cause a non-ideal transition. Therefore, it is helpful to integrate a detection of the continuous feed into the separator (by learned models). This integration would allow an analysis of the current situation (first phase of self-optimization process). The next step is to define how an adaption of the process can be performed to avoid a non-ideal transition of the raw product. For example, it is possible to adjust the demanding incoming flow of the raw product (“adjust the system’s behavior”: e.g. patterns in the conventional control strategies). Additionally, a step could be the consideration of a situation-specific objectives determination (e.g. “Multi-Objective Optimization” based on learned model). For example, if an inconsistent feeding of the raw product is detected and raw product is delicate, the objective “max. reliability” has be prioritized higher. On the other side, if the raw product feed is regular and continuous the objective “max. productivity” can be prioritized more.

6. Conclusion

The design of self-optimization features in mechatronic systems is still a challenge due to the various disciplines that are involved. We have presented the main characteristics of self-optimizing systems and a discipline-spanning specification technique for a holistic system design. We focused primarily on the identification of potentials for the self-optimization in conventional mechatronic systems. Thus, we explained our developed methodology in a general manner. The methodology supports the identification of solution pattern for the different phases of the self-optimization process. The developer demanded that, because the existing approaches mainly focuses on the realization of the second phase of the self-optimization “determining system’s objectives”, which often is not needed to reach improved system behavior. Additionally, the methodology generates synergies in quality and knowledge management and thus improve the development of innovative systems in a pragmatic manner. To conclude we presented the evaluation of our methodology exemplified by a separator. In our future work, we are going to use the methodology by further application examples and will detail the methodology based on the new experience.

Acknowledgements

This research and development project is funded by the German Federal Ministry of Education and Research (BMBF) within the Leading-Edge Cluster “Intelligent Technical Systems OstWestfalenLippe” (it's OWL) and managed by the Project Management Agency Karlsruhe (PTKA). The author is responsible for the contents of this publication.

References


