Paper:

Probabilistic Planning for Predictive Condition Monitoring and Adaptation Within the Self-Optimizing Energy Management of an Autonomous Railway Vehicle

Benjamin Klöpper*, **Christoph Sondermann-Wölke****, **and Christoph Romaus****

*National Institute of Informatics
Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan
E-mail: benjamin.kloepper@googlemail.com
**University of Paderborn, Germany
[Received January 2, 2011; accepted March 22, 2011]

Self-optimizing mechatronic systems are a new class of technical systems. On the one hand, new challenges regarding dependability arise from their additional complexity and adaptivity. On the other hand, their abilities enable new concepts and methods to improve the dependability of mechatronic systems. This paper introduces a multi-level dependability concept for self-optimizing mechatronic systems and shows how probabilistic planning can be used to improve the availability and reliability of systems in the operating phase. The general idea to improve the availability of autonomous systems by applying probabilistic planning methods to avoid energy shortages is exemplified on the example of an innovative railway vehicle.

Keywords: self-optimizing systems, dependability, probabilistic planning, energy management

1. Self-Optimizing Systems and Dependability

Technical systems and machines are designed to fulfill tasks for humans. The spectrum of tasks and the quality of task fulfillment is continuously improved by technical progress. The quality of task fulfillment can be measured in various dimensions, depending on the current area of applications. Examples for such dimensions are: timeliness, resource consumption, processing accuracy (e.g., in case of machining tools), or comfort and driving pleasure (in case of vehicles). The close integration of electromechanical systems, electronic and information technology in mechatronic systems [1] opens new paths to further improvement: the availability of information and communication technology enables systems to adapt their behavior to changing environmental settings and user preferences. We use the term self-optimization to characterize such systems.

Self-optimizing systems are able to adapt their objectives autonomously [2]. This includes modifying the relative weighting or ranking of the objectives. Adapting the objectives must result in an adaptation of the system behavior. For this purpose, the identified objectives are transformed into a corresponding optimization problems. The solutions of the optimization problems indicate the suitable behavior adaptations. This is realized by adapting parameters (e.g., changing a control parameter) or the structure of the system (e.g., replacing the current controller). Hence, a self-optimization process is defined as an iterative sequence of three actions.

Situation Analysis: Situations include the system's state and all observations about its environment.

Determination of Objectives: The relevant objectives are ranked and weighted. Objectives can be also transformed into side conditions.

Behavior Adaptation: The determination of objectives results in a formulation of optimization problems which determine appropriate adaptation of the system behavior. This adaptation is implemented by changing control parameters or replacing controller variants.

According to Laprie [3] dependability encompasses four attributes: safety, reliability, availability and confidentiality. These attributes are integrated in the system of objectives and may increase the dependability under consideration of the application and current situation. Regarding dependability, self-optimizing mechatronic systems comprise both: on the one hand, the risk of unforeseen failures, due to their complexity and inherent non-deterministic behavior; on the other hand, the chance of developing new dependability concepts by using the paradigm of self-optimization.

An important factor in the dependability of many mechatronic systems is the assurance of sufficient energy supply. This paper introduces a multi-level dependability concept for self-optimizing mechatronic systems applied during the operating phase and shows how planning can be used to implement a pro-active and risk-avoiding behavior. This method is especially promising to ensure sufficient energy supply and thus improve the availability of mechatronic systems. A first version of the approach has been introduced in [4].

The next section introduces the multi-level dependability concept for self-optimizing mechatronic systems. Subsequently, the importance of the energy storage for the dependability of mechatronic systems is motivated. The fourth section introduces our application example, the autonomous railway vehicle RailCab. In Section 5 a planning concept for mechatronic systems is explained and that is extended to a probabilistic planning procedure that is integrated in the dependability concept (Section 6). Simulation results are presented in Section 7. The last section presents the conclusion.

2. Related Work

2.1. Planning for Mechatronic System

Although planning is a promising method, it is hardly used in mechatronic systems. One reason is a significant difference in the system models. Most planning approaches model system activities as discrete sequence of states and activities [5]. In mechatronic systems, the continuous trajectories of system activities are important. One possibility is the usage of planning models based on hybrid automata [6], which integrate discrete change between modes and continuous evolution of state variables. Maier and Sachenbacher introduce an application to mechatronic systems in manufacturing [7]. Considering the long planning horizon required in the RailCab system, the information about the future environment is not precise enough to derive the differential equations required for the definition of hybrid automata. Hence, a hybrid planning architecture was developed [8, 9]. The hybrid planning integrates planning and simulation in order to react to unavoidable plan deviations. The just-in-case planning introduced in this paper is an important building block within this architecture.

2.2. Probabilistic Planning

There are various planning algorithms which consider random variables in the state description and action effects. Examples of such planning algorithms are Paragraph [10], Probapop [11], and Weaver [12]. These planning systems do not support the consideration of an objective function, which is required to implement the self-optimization process. Weaver has the most advanced representation of environmental influences. It includes exogenous events and a probabilistic model of their influence towards the action results and dynamically constructs a Bayes Network to calculate the joint probability distribution of the plan. Nevertheless, Weaver does not include external knowledge about the environment and is restricted to a priori knowledge included in the action definition.

Thus, all these planning approaches clearly lack the ability to adapt themselves to changing environmental circumstances. The approach presented in this contribution uses a distributed system of expert agents and provides an integration of up-to-date information about the environment. Furthermore, the basic concept of the presented approach to apply a probabilistic plan analysis to determine threshold values enables the integration of arbitrary deterministic planning concept (e.g., state space search like in the example or meta-heuristics) to provide the original and alternative plans.

2.3. Stochastic Dynamic Programming

Stochastic Dynamic Programming (SDP) based on Markov Chains (MC) can be considered as an approach for planning under uncertainty. SDP for instance is applied in optimal control problems for hybrid vehicles [13] and electric vehicles with hybrid energy storages [14]. Besides a position independent determination of the control strategy, Johanneson et al. [13] suggest to consider a position dependent Markov Chain, similar to the suggested agent based approach. Nevertheless, there are some major difference regarding the presented approach and SDP. Result of an SDP is an (optimal) policy, which defines for every possible state from the state space a corresponding action which is to be chosen for that case. The policy selects actions in such a way, that the expected costs (e.g., fuel consumption [13] or energy losses [14]) of all actions in a finite or infinite horizon are minimized. Here a problem arises regarding the integration with the dependability concept introduced in the next section: it is not possible to specify a threshold that defines the acceptable probability of a failure. Thus, to integrate a SDP into the dependability concept a cost model has to be chosen that results in a corresponding failure probability of the optimal policy. The selection of such a cost model is not a trivial problem. Furthermore, the simulation results in Section 8 show that the planning procedure introduced in this contribution reduces the failure probability already if a small number of alternative plans are added. This feature enables the implementation as an anytime algorithm and the reliability of the system can be improved with limited exploration of the state space and thus in limited calculation time. In contrast, SDP usually analyze the complete state space to generate a policy, thus it is not possible to use intermediate results. Online planning approaches with partial exploration of the state space exists for similar problem models such as partially observable Markov decision processes [15]. Nevertheless, the consideration of a threshold probability for failure is even more difficult to consider in these approaches.

3. Predictive Condition Monitoring

To reduce the risks and exploit the potentials of self-optimization a predictive condition monitoring policy was developed. The policy comprises the design phase, the operating phase and the maintenance phase (cf. [16]). The main element of the policy in the operating phase is the Multi-Level Dependability Concept (MLDC) depicted in **Fig. 1**. The MLDC monitors the system state and

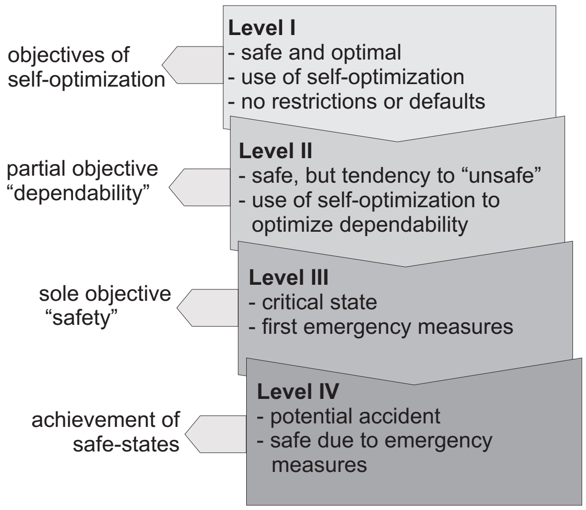


Fig. 1. Multi-level dependability concept.

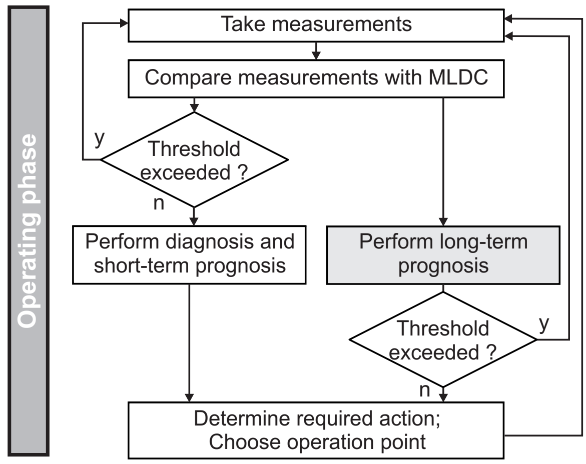


Fig. 2. Condition monitoring.

classifies occurred errors or predicted perilous states into four different levels depending on the current hazard status (cf. [17]). The difference of the MLDC to other condition monitoring and assessment concepts is the direct influence on the self-optimization process. Every level induces a certain impact on the system of objectives. The levels of the MLDC and the impact on the system of objectives are explained in detail in the following.

- First level: The system operates in a dependable way. Dependability is one objective among others.
- Second level: A perilous state is detected. Self-optimization is used to return to the first level. Therefore, the priority of the affected attribute of dependability is increased.
- Third level: An error has occurred. First emergency mechanisms are triggered to reach a safer state. In the system of objectives, safety is the sole objective to avoid the failure of the whole system and the consequences involved. The other attributes of dependability may occur as sub-objectives of safety.
- Fourth level: The control over the system is lost. Mechanisms like emergency routines are executed to reach a fail-safe state.

An important feature of any implementation of the MLDC is the transition from the first level to the second level. The system monitoring must be enabled to identify perilous states, which possibly result in error states. Furthermore, the self-optimization process must be able to identify suitable modification of the system objectives to avoid the occurrence of an error. Therefore, the MLDC is embedded into a predictive condition monitoring policy, which is based on the common ISO 17359 policy [18]. The operating phase of the policy is illustrated in **Fig. 2**.

The first step in the operating phase is taking measurements of the monitored parameters and calculate the resulting dependability quantities. This information is transmitted on the one hand to a hard real time information processing (left branch in **Fig. 2**). The quantities are compared immediately with the threshold values of the MLDC. If the state of the system does not exceed the defined threshold of Level I, the system is in a regular state. Otherwise, the system performs application specific diagnosis and short-term prediction methods to determine the required action. This process step works in hard real time, which is necessary to initiate emergency routines. On the other hand, the quantities are passed to the predictive part of the policy, which works in soft real time (right branch in **Fig. 2**). On the basis of further knowledge the long-term progression of the system is analyzed and the dependability quantities are updated for a point in time in the future. For the long-term prognosis a probabilistic planning procedure is used. Afterward, the retrieved quantities are compared with the MLDC and checked if the system will still be in Level I for a finite horizon of time. If the threshold of Level I is exceeded a suitable proactive measure has to be determined. In the case of the predictive condition monitoring a different suitable operation point has to be chosen.

Section 7 will introduce a probabilistic planning procedure that identifies conditions which indicate a transition from the first to the second level and identifies appropriate modification of the system objectives.

4. Energy Storage and Dependability

The dependability of systems can be affected by miscellaneous factors. One particular important factor is energy, as most systems rely on a sufficient supply of energy for operation. Especially for autonomous systems such as vehicles or isolated systems, where steady external energy supply is not possible, energy management is a critical issue. In these applications often an electrical energy storage is installed to store energy for continuous supply or to compensate for power peaks.

For the availability and thus the dependability of the system it is indispensable to operate the energy storage only in a valid range, especially observing its admissible State Of Charge (*SOC*) as well as its maximum power ratings. This is done by an energy management which controls the power demands of the system. It is a serious issue that the state of charge of the energy storage is dependent on previous decisions of the system, namely the previous energy flows of the system components. Inappropriately chosen operating conditions in the past thus can later lead to a highly suboptimal operation or even failure of the system caused by unavailability of energy.

An energy management that secures the availability of the energy storage and thus the functionality of the system can be realized by planning the energy demand of the system. The idea is to avoid the heavy constriction by only slightly restricting the operation of the system at earlier points in time.

As the power supply of the system is such a significant factor, it is particularly considered in the dependability concept. Probabilistic planning of the future power demands of the system offers an intelligent solution at the first and second level and presents a method to design an energy management for the system. It maintains full operability of the system if sufficient energy is available (first level). In risk of shortage of energy (second level), it takes actions to save energy and thus leads the system back to the first level.

Of course, planning cannot eliminate all possible risks. In case of incorrect planning or unforeseeable energy demand a quick reaction is necessary to prevent a failure. This can be achieved by a precalculated static priority list which defines which modules of the system have to be switched off in which order and which other measures, e.g., higher energy transfer, have to be applied to lead the system into the safe range (level three). At this level, no self-optimization is applied.

If this priority list also fails, the system has to be led to a fail-safe state, e.g., an emergency stop or fail-safe minimal operation (fourth level).

5. Application Example: The RailCab System

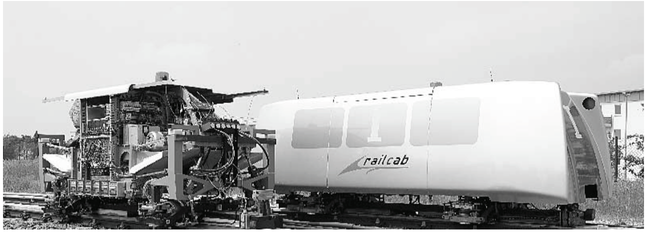


Fig. 3. Two RailCab prototypes on the test track.

The RailCab system (**Fig. 3**) consists of small autonomously driven rail-bound vehicles [19].[1] The vehicles encompass several innovative subsystems that are designed to perform specific tasks. This section focuses on three subsystems: the propulsion system, the hybrid energy storage system, and the active suspension module.

The *propulsion system* of the RailCab vehicles is realized by a doubly fed linear drive [20, 21]. The linear drive is comparable to a sliced asynchronous three-phase induction motor. The primary motor part called stator is installed between the rails. The secondary motor part is mounted to the vehicle. The main idea is to enable the operation of RailCabs with different velocities on the same stator section. Furthermore, the drive is able to transfer power from the stator into the vehicle, which leads to the omission of overhead contact lines or conductor rails.

The primary *power supply* of the on-board electrical system of the RailCab is the power transfer via the doubly fed linear motor. However, this power transfer heavily depends on the operating conditions, thus it may be limited and to some extend not sufficient. To offer a continuous power supply of the function modules of the RailCab, a *Hybrid Energy Storage system* (HES) is installed on the vehicle. It consists of a combination of nickel metal hydride batteries and double layer capacitors. Its main task is to compensate for the difference between the transferred power of the motor and the power demand of the modules. In addition to the long-term energy management, a storage management determines the power distribution to both storage types. Probabilistic deviations in the continuous power flow within a short-term horizon (track section) can be taken into account [14, 22].

The *active suspension module* increases the comfort for passengers. The basic idea of this suspension system is to omit passive dampers as they would transmit high-frequency disturbances from the rail-track to the coach body. Hence, the body is connected to the carriage only via springs. The necessary forces to damp the coach body movement are generated by displacing the spring bases via hydraulic cylinders depending on the current movement of the body. A feed-forward disturbance compensation offers further improvement of the comfort, if data about the track is known in advance. This data is accessible from track agents and gained from previous runs of other vehicles on the regarded track section. Self-optimization within the active suspension module introduces a degree of freedom, that is used as running example in this paper. In [23] a multiobjective optimization regarding the trade-off between provided comfort and consumed energy is introduced. The multiobjective optimization considers two objective function $f_1(t)$ and $f_2(t)$:

$$f_1(t) = \frac{1}{T} \cdot \int_{\tau=t-T}^{T} \sum_i |W_i(a_i(\tau))| dt \quad . \ . \ . \ . \quad (1)$$

$$f_2(t) = \frac{1}{T} \cdot \int_{\tau=t-T}^{T} \sum_j P_{hydr,j}(\tau) dt. \quad . \ . \ . \ . \quad (2)$$

The objective function in Eq. (1) is to be minimized in order to optimize the traveling comfort. It represents the weighted average body acceleration in vertical ($i = 1$) and lateral ($i = 2$) direction and the weighted angular acceleration in rolling direction ($i = 3$). The objective of minimizing the energy consumption is expressed in Eq. (2).

1. RailCab Project, Paderborn University. Website: www.railcab.de.

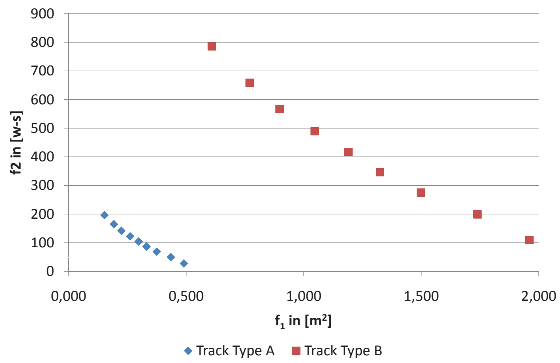


Fig. 4. Results of the offline multi-objective optimization (f_1 and f_2 from Eqs. (1) and (2), respectively).

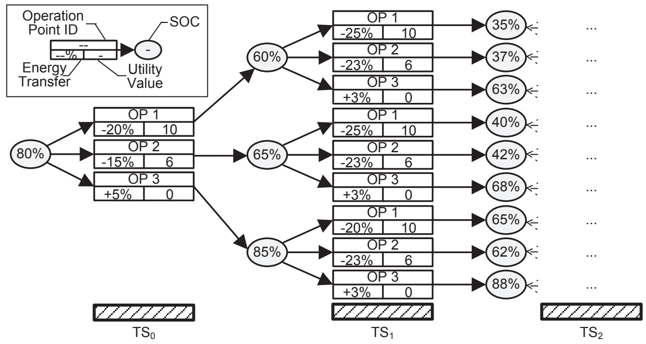


Fig. 5. Sketch of a search space with 3 operation points.

It describes the average hydraulic power of all cylinders of the suspension module and can be mapped on a corresponding electrical energy consumption. **Fig. 4** shows two examples of Pareto-sets in the objective space for typical track sections: track type A is a smooth track while track B represents a rough track with considerable amount of disturbances arising from the track. It can be seen that the trade-off regarding energy consumption and comfort depends on the track type. Considering this difference, a mechanism is required to make efficient use of the limited energy resources. These points in the Pareto-space correspond to configuration of the active suspension module including parameters such as compensation rate or working pressure of the hydraulic system.

6. Planning in the RailCab System

Planning refers to a process that determines the future state of something and, more specifically, is about identification of a future course of actions to accomplish a goal [24]. In case of the RailCab system, planning is related to the task of traveling from a location A to an location B. The most appropriate route (e.g., the shortest path) in the railway network is provided by a logistics planning and scheduling systems. Hence, we consider the RailCab traveling along a number of consecutive *track sections*. A track section is as a cut-out from the network, which has homogeneous features (e.g., track excitation, slope, etc.) and does not contain a switch (cf. [8]). The planning procedure selects for each track section on a given route and each sub-module within the RailCab an appropriate behavior for each sub-module. The behavior is described by alternative operation points, which can be derived from offline (prior-to operation) Pareto-optimization. Hence, an operation point defines a specific trade-off between several objective functions and the consumption of resources, e.g., energy. For instance, the Pareto-optimal settings of the active suspension in **Fig. 4** can be interpreted as operation points of the sub-module active suspension. **Fig. 5** shows an example state space defined by three operation points: for each track section an operation point has to be selected. With the expected energy transfer, the subsequent state of charge SOC of the energy storage module before entering the next track section can be calculated. During planning, both $SOC \in [0 \dots 100]$ and energy transfer $\Delta^i_{SOC} \in [-100 \dots 100]$ from or to the storage are modeled as discrete variables. The discrete values refer to a percentage of the entire capacity of the energy storage, such that $SOC_i = 80$ refers to a situation in which the energy storage holds 80% of its capacity and a transfer of $\Delta^i_{SOC} = -10$ means that additional 10% of the capacity are transfered from the storage to the energy consuming modules, resulting in a new state of charge $SOC_{i+1} = 70$. The planning procedure has to select the operation points in such a way that the plan is feasible (e.g., energy storage is never empty) and an accumulated utility value (e.g., derived from Eq. (1)) of all operation points is maximized.

Since operation points are selected solutions from a Pareto-set, they correspond to a specific weighting of objective values. The actual possible values such as energy consumption and comfort value depend on the specific environmental settings and track properties. To limit the amount of data to define an operation point and collect enough empirical data, track sections are not considered individually, but grouped in classes according to topographical properties and previously experienced objective values and energy consumptions. A joint classification of track sections enables faster exploration and more recent information since track sections are more frequently visited. To determine a specific control regime for a specific track section, the relative weighting defined by the Pareto-set regarding a typical track section is used to derive a single objective optimization problem using online information about the track section. Hence, the planning procedure uses only *expected values* and the actual encountered values will differ during the execution. If the deviations grow large, it is possible that failure and energy shortages occur.

7. Just-In-Case-Planning Combined with the MLDC

The term just-in-case planning refers to the idea to provide proactively alternative plans for the case that specific

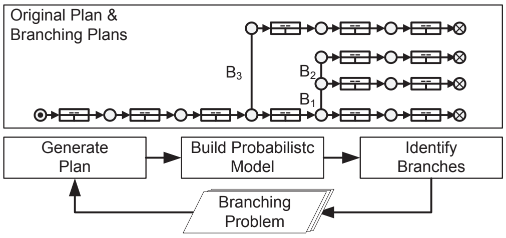


Fig. 6. Just-in-case-procedures and initial plan with branching plans.

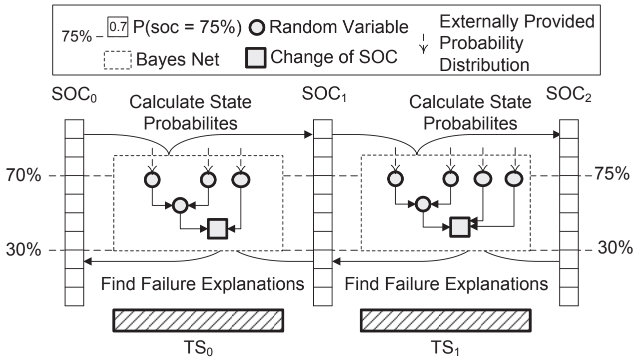


Fig. 7. SOC and Bayes Networks in the probabilistic analysis.

deviations occur during the execution of the plan. Thus, just-in-case planning corresponds to conditional planning as originally introduced by Warren [25] and enriched by probabilistic information by various works (for instance cf. [26, 27]). It basically constructs a plan with branches, where each branch is annotated with a condition (**Fig. 6**). If the condition becomes true during execution, the system executes the corresponding branch.

Figure 7 gives an overview of the planning procedure and its output. Just-in-case planning encompasses three main procedures which are executed in a closed loop. In the the *first step*, the initial plan is generated. The planning procedure (e.g., search based or any heuristic or meta heuristic such as Genetic Algorithm or Simulated Annealing) explores the state space introduced in the previous section and returns a linear sequence of operation points and resulting states, linking the first and the last track section of a route. In order to construct this linear sequence, the state changes induced by operation points are considered to be deterministic. In the *second step*, this deterministic model is transformed into a probabilistic model, considering the energy transfer for each operation point and the state of charge of the storage system as random variable. In the *third step*, the probabilistic model is exploited to validate the plan against a threshold probability p of failure ($SOC = 0$) and to identify specific SOC_i values that (1) occur with a probability above the threshold and (2) increase the probability of a failure. These SOC_i define a state and a condition in the plan, from which an alternative plan branches from the original plan. Those alternative plans are proactively generated and recursively hedged against the risk of failure.

The just-in-case planning contributes to the multi-level dependability concept (Section 3) by identifying conditions under which the RailCab should adjust its behavior: the conditions define transitions from the first to the second level by prioritizing the dependability related objective *safe energy* in the alternative plans.

The planning process covers all three steps of the self-optimization process and defines a system of objectives for each track section.

Situation Analysis (1): Collect information about the route sections.

Objective Determination: Identify alternative plans (with branching conditions) with respect to possible discrete state trajectories.

Situation Analysis (2): Monitor the current execution trajectory.

Behavior Adaption: If the monitoring perceives a branching condition, the RailCab switches to the corresponding alternative plan.

7.1. Building the Probabilistic Model

The generation of the deterministic plan is done by a modified search procedure that explores the state space shown in **Fig. 5**. Deviations from the deterministic plan are caused by not exactly known or unknown environmental influences. Bayes Networks are a feasible approach to build a probabilistic model of environmental influences and the energy transfer resulting from the operation points. A Bayes Network is a graphical representation of a joint probability distribution over several random variables represented by the node set N and directed edges $\{n_1, n_2\}$ denote that the probability distribution of n_2 depends on the actual value of n_1 [28].

Each pair of operation point and track type in the deterministic planning model corresponds to one Bayes Network describing the conditional probability distribution $P(\Delta^i_{SOC}|\varepsilon_i)$ with Δ^i_{SOC} denoting the change in the state of charge (transfer from or to the storage) and ε_i the relevant environmental influences towards the RailCab. Thus, the Bayes Networks of operation points and track types enable the inference of the probability distribution in the change of state of charge (Δ_{SOC}) from the probability distribution of the environmental influences. Determining an updated probability distribution of Δ_{SOC} given some evidence about the current or expected environmental influences on track section is standard inference procedure of Bayes Network. In our work, we used Pearl's message passing algorithm [29]. **Fig. 7** gives an overview of the probabilistic plan model. States are modeled as discrete random variables $SOC_i \in [0 \ldots 100]$ and the Bayes Network of the corresponding pair of operation points and track type links SOC_i to SOC_{i+1}.

Given a discrete probability distribution of SOC_i at the start of start section i the probability that SOC_{i+1} equals

Table 1. Simplified example of determining the probabilistic state variable $SOC_i + 1$ from the previous probabilistic state SOC_i and the probability distribution of the energy consumption Δ^i_{SOC}.

	$\Delta^i_{SOC} = -10$			$\Delta^i_{SOC} = +10$		
	Event	Probability	SOC_{i+1}	Event	Probability	SOC_{i+1}
$SOC_i = 50$	A	$0.3 \cdot 0.6 = 0.18$	$50 - 10 = 40$	C	$0.3 \cdot 0.4 = 0.12$	$50 + 10 = 60$
$SOC_i = 30$	B	$0.7 \cdot 0.6 = 0.42$	$30 - 10 = 20$	D	$0.7 \cdot 0.4 = 0.28$	$50 + 10 = 40$

	Supporting Events	Resulting Probability
$SOC_{i+1} = 20$	B	0.42
$SOC_{i+1} = 40$	A, D	$0.18 + 0.28 = 0.46$
$SOC_{i+1} = 60$	C	0.12

any integer value $x \in [0 \cdots 100]$ can be calculated by:

$$P(SOC_{i+1} = x) = \sum_{(j=-100)}^{100} P(\Delta^i_{SOC} = j|\varepsilon_i) \cdot P(SOC_i = x - j) \quad . \quad (3)$$

The probability of the random event $SOC_{i+1} = x$ can be calculated by summing up all elementary events regarding the energy consumption Δ^i_{SOC} and the previous SOC, SOC_i that result in the value x. By multiplying the probability of a specific energy consumption $\Delta^i_{SOC} = j$ with the corresponding $SOC_i = x - j$, the probability of all such elementary events resulting in $SOC_{i+1} = x$ can be calculated. Obviously, Eq. (3) assumes the independence of the two events $\Delta^i_{SOC} = j$ and $P(SOC_i = x - j)$. This assumption is appropriate since $P(SOC_i)$ is determined by Δ^{i-1}_{SOC} and any dependability must be caused by similar environmental influences (by proximity in both place and time) and hence Δ^{i-1}_{SOC} and Δ^i_{SOC} are conditionally independent given these similar environmental influences.

A small and simplified example gives a better impression of the probabilistic plan structure. We assume only two possible values SOC_i with $P(SOC_i = 50) = 0.3$ and $P(SOC_i = 30) = 0.7$ and consequently $P(SOC_i = x) = 0.0$ for all other values x. Furthermore, we assume only two possible values of Δ^i_{SOC} with $P(\Delta^i_{SOC} = -10) = 0.6$ and $P(\Delta^i_{SOC} = +10) = 0.4$ and $P(\Delta^i_{SOC} = x) = 0.0$ for all other values y. **Table 1** shows how according to Eq. (3) the probability of elementary events is calculated and how the probability of possible values of SOC_{i+1} is calculated by summing up the probabilities of the corresponding elementary events. Obviously, in a real planning run there will be usually much more possible values for both the state variable SOC_i and the energy transfer Δ^i_{SOC} and hence more calculations necessary.

The probability distribution of SOC_i can be calculated for each track section i by starting from a predetermined probability distribution of the first state of the plan.

7.2. Identification of Branching States

A branching condition is defined by a threshold value of SOC_i. If SOC_i is below or equal to this threshold value, the RailCab should choose a specific alternative plan for the remaining route. The combination of a branching condition and an alternative plan is referred to as a branch in the conditional plan. To determine the branching condition, it is necessary to determine the most likely explanation for a state of charge below a value y ($SOC_{i+1} \leq y$) in terms of the previous SOC_i. For instance, the probability of an empty energy storage $SOC_f \leq 0$ after the final track section is checked. The value of SOC_{f-1} which is the *most likely explanation* for the empty storage is than determined for $y = 0$ by:

$$\max_{x \in [1 \cdots 100]} : P(SOC_i \leq x) \cdot P(\Delta^i_{SOC} \geq (x - y)) \quad . \quad (4)$$

To obtain the *most likely explanation* for $SOC_{i+1} \leq y$ it is again necessary to analyses the combinations of the previous state of charge SOC_i and the energy consumption Δ^i_{SOC}: a specific value $SOC_i = x$ results in a state new state of charge $SOC_{i+1} \leq y$ if the energy consumption Δ^i_{SOC} is greater than or equal to the difference between the two values $SOC_i = x$ and $SOC_{i+1} = y$. But of course, all other possible $SOC_i < x$ will also result in a new state of charge $SOC_i + 1 \leq y$ below y given these energy consumptions. Hence, the probabilities $P(SOC_i \leq x)$ and $P(\Delta^i_{SOC} \geq x - y)$ are combined in order to get the *most likely explanation* for a new state of charge $SOC_{i+1} \leq y$. The process of branching condition identification starts in the final state of a plan. An empty energy storage is assumed (assumed evidence $P(SOC_f = 0) = 1$). The Bayes Network of the corresponding operation point and track type is used to identify the most likely explanation for an empty energy storage referring to SOC_{f-1} at the start of the previous track section (according to Eq. (4)). Moving backwards in the plan, the most likely explanation of all identified branching conditions is determined. If the probability of the branching condition is beyond a threshold probability p, the condition is added to the list of open branching conditions.

7.3. Resulting Plan Structure

While planning time is available (the RailCab travel did not start), the planning procedure continuously determines alternative plans for identified branching points. By recursively applying the analysis to alternative plans, a plan structure as shown in **Fig. 8** is generated. Several alternative plans (A, B, C) starting from different track sections are generated. For each step in a plan correspond-

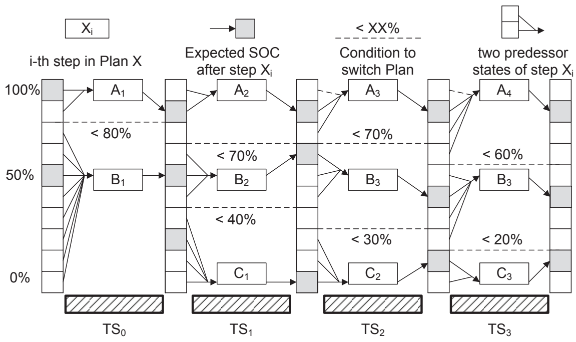


Fig. 8. Conditional plan structure.

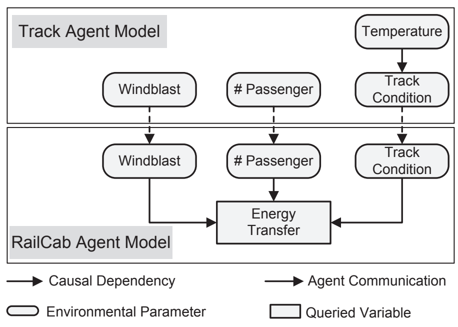


Fig. 9. Structure of the probabilistic model for experiments.

ing to a track section i ($i \in \{0,\ldots,3\}$ in **Fig. 8**), these plans define a number of alternative operation points with corresponding conditional values. Hence, for each conditional value the corresponding behavior adaption (changing operation points) is proactively determined. In accordance with the condition monitoring concept of the MLDC (**Fig. 2**), the conditional values are monitored during operation: if the currently measured SOC_i is below any conditional value, the RailCab changes from the first level of the dependability concept to second level and selects the plan attached to the smallest condition value equal to or above the actual value of SOC_i. Consequently, each available alternative operation point has a number of possible preceding state, defined by the conditional value attached to the plan and the next smaller conditional value of another branch. Thus, the operation point for the current track section i is selected in such a way that the failure situation of an empty energy storage is avoided (below the threshold probability). The same selection process is repeated on the next track sections.

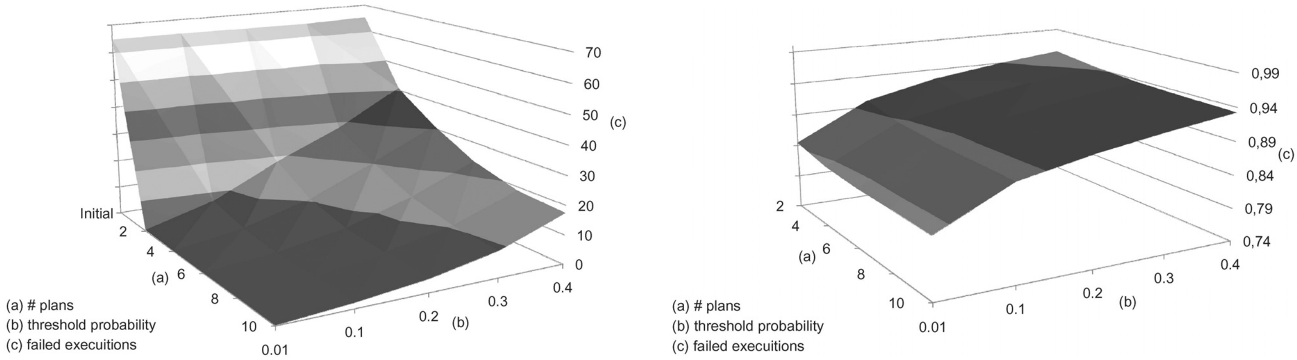
This structure enables the RailCab also to return to the first level of the dependability concept and to use plans with higher energy consumption (and hence higher objective value), if meanwhile less energy was consumed than assumed and hence the subsequent $SOC_j, j > i$ is not below any conditional value. Thus, the quality of system behavior is degraded gracefully and always reflects the current risk of an empty energy storage.

8. Simulation Results

For simulation experiments, rather small planning problems with just 9 operations points, 10 different track types and 100 track sections were considered. The operation points are derived from the offline multiobjective optimization of the active suspension module (cf. **Fig. 4**, complete data given in Appendix A). The behavior of the propulsion module is assumed to be fixed. Hence, each operation point corresponds to a trade-off regarding the comfort (Eq. (1)) and change in the state of charge SOC_i resulting from the energy consumption of the active suspension (Eq. (2)) and fixed consumption or surplus of the propulsion module. The planning procedure maximizes the comfort value while maintaining the state of charge feasible ($100 > SOC_i > 0$). The limited problem size enabled the optimal solution of the problems and hence the definition of a definite upper-bound for the results. The initial planning procedure is a modified breadth-first-search that is able to find the optimal solution of a deterministic planning problems.

Figure 9 shows the structure of the Bayes Networks used in the experiments. The probabilistic model is split into two parts: environmental parameters and the internal model of the RailCab, representing the energy consumption or production of each operation mode. The environmental model contains probability distribution regarding the weather conditions (wind blasts, temperature), derived track condition, and a forecast of the number of passengers. This splitting of the probabilistic model enables the application of a system of distributed expert agents providing up-to-date probability distribution for each track section (for details cf. [30]). Basically, the up-to-date probability distributions are sent via message communication by the expert agents and replace their counterparts in the vehicle model. The simulated probabilistic deflections of energy transfer are rather high and the assumed initial *SOC* was between 5% and 13% above the minimal required energy to travel route. These configurations assured to obtain a significant number of threatening plan executions. Branching conditions were added to the list if their probability was above a threshold value. A heuristic procedure generated the alternative plans, replacing operation points with operation points lower consumption, until the failure probability was below the threshold value. Each generated plan was tested in 100 Monte-Carlo simulation runs.

Figure 10 shows the impact of two planning parameters: the threshold probability and the number of considered branches. If the risk of an empty storage is below the threshold probability, a plan is accepted without adding new branching conditions and plans. Hence, a lower threshold probability corresponds to more risk-avers plans. The number of branches reflects the impact of limited planning time. Less planning time results in fewer branching plans. The left diagram in **Fig. 10** shows the failure probability depending on the threshold value and the number of branches in the plan repository.



(i) Failed plan executions (%) (c), depending on threshold value (a), and number of branches (b)

(ii) Deviation from optimal result (c), depending on threshold probability (a), and number of branches (b)

Fig. 10. Experimental results just-in-case-planning.

Both variables show a significant impact on the number of failed plan executions: the number of failed executions quickly decreases with higher number of branches and lower threshold probabilities. The second diagram in **Fig. 10** shows the impact of the variables on the plan quality. To measure the plan quality, each simulation run was transformed into a deterministic planning problem with the experienced energy transfers and then solved optimally. This optimal result under perfect knowledge defines a theoretical upper bound for the problem under uncertainty. It can be seen that the correlation between the two parameters and the plan quality are rather weak. Only extreme values of threshold and number of branches reduce the plan quality significantly. The reason is that the RailCab will often change to very risk-averse plans which consequently have poor utility values. Hence, there is a fundamental trade-off between reliability and plan quality.

Regarding run-times, the initialization of Bayes Networks (including reading files) took in average 5.60 ms for 100 tracks (3.39 ms for 25 tracks, 4.52 ms for 50 tracks) and the maximal measured value was 78 ms.[2] The subsequent analysis took in average 8.37 ms for 100 track sections (3.45 ms for 25 tracks, 3.56 ms for 50 tracks) and the worst value was 31 ms. Since the probabilistic analysis is always applied to complete plans defining one operation point for each track section, the run-time of the analysis is independent from the number of alternative operations modes, a clear benefit of the separation of planning and probabilistic analysis. The generation of an alternative plan took between 1.63 ms (25 tracks and threshold value 0.4) and 250 ms (100 tracks and threshold value 0.0001).

2. All time measurements on this paper were made on a desktop PC with an Intel Core 2 Quad CPU (2.99 GHz), 8 GB RAM, Windows 7 64 Bit, and JVM 1.6 using a single thread implementation.

9. Conclusion

This paper introduced a new class of technical systems: self-optimizing mechatronics. The complexity and the inherent non-determinism of the systems call for new dependability concepts. Thus, a multi-level dependability concept for self-optimizing mechatronic systems has been defined. An important element of the multi-level concept is the prioritization of dependability related objectives within the process of self-optimization. Especially for systems with autonomous energy supply, the early detection of perilous states is crucial to maintain the availability and reliability of the systems.

The autonomous railway vehicle RailCab was introduced as demonstration system. A planning concept for the RailCab system was briefly explained and extended to a probabilistic planning, which is able to detect possible flaws in deterministic plans and creates plan branches and conditional values. The conditional values indicate the necessity to adapt the current objectives of the mechatronic systems and the plan branches define appropriate modification of the system of objectives. Compared to conventional probabilistic planners, the introduced algorithm is able to include up-to-date information provided by expert agents. Simulation results show that the introduced algorithm can improve the dependability and in particular the availability of autonomous systems.

Acknowledgements

This contribution was developed in the course of the Collaborative Research Centre 614 "Self-Optimizing Concepts and Structures in Mechanical Engineering" funded by the German Research Foundation (DFG). Benjamin Klöpper is a visiting researcher at the National Institute of Informatics and scholarship holder of the German Academic Exchange Service (DAAD).

References:

[1] R. Isermann, "Mechatronic Systems: Fundamentals," Springer-Verlag, London, 2005.

[2] J. Gausemeier, S. Kahl, and S. Pook, "From Mechatronics to Self-Optimizing Systems," In Self-optimizing Mechatronic Systems: Design the Future, 7th Int. Heinz Nixdorf Symposium, 2008.

[3] J. C. Laprie (Ed.), "Dependability: Basic Concepts and Terminology in English, French, German, Italian, and Japanese," Springer-Verlag, Wien, 1992.

[4] B. Klöpper, C. Sondermann-Wölke, C. Romaus, and H. Voecking, "Probabilistic Planning Integrated in a Multi-level Dependability Concept for Mechatronic Systems," In 2009 IEEE Symposium on Computational Intelligence in Control and Automation, pp. 104-111, 2009.

[5] S. Russell and P. Norvig, "Artificial Intelligence: A Modern Approach," Prentice Hall, 3 edition, 2009.

[6] G. Della Penna, D. Magazzeni, F. Mercorio, and B. Intrigila, "UPMurphi: A Tool for Universal Planning on PDDL+ Problems," In Proc. of 19th Int. Conf. on Automated Planning and Scheduling, pp. 106-113, 2009.

[7] P. Maier and M. Sachenbacher, "Self-Monitoring and Control for Embedded Systems using Hybrid Constraint Automata," In Self-X in Engineering, pp. 8-23. MV-Verlag, 2009.

[8] P. Adelt, N. Esau, and A. Schmidt, "Hybrid Planning for an Air Gap Adjustment System Using Fuzzy Models," J. of Robotics and Mechatronics, Vol.21, No.5, pp. 647-655, 2009.

[9] P. Adelt and B. Klöpper, "Buildings Blocks and Prototypical Implementation of a Hybrid Planning Architecture," In Self-x in Engineering, pp. 55-67, 2009.

[10] I. Little and S. Thiebaux, "Concurrent Probabilistic Planning in the Graphplan Framework," In Proc. of the Int. Conf. on Automated Planning & Scheduling, June 6-10, 2006, Cumbria, pp. 263-273, 2006.

[11] N. Onder, G. C. Whelan, and L. Li, "Engineering a Conformant Probabilistic Planner," J. of Artificial Intelligence Research, Vol.25, pp. 1-15, 2006.

[12] J. Blythe, "Planning Under Uncertainty in Dynamic Domains," Ph.D. thesis, Carnegie Mellon University Computer Science Department, 1998.

[13] L. Johannesson, M. Asbogard, and B. Egardt, "Assessing the Potential of Predictive Control for Hybrid Vehicle Powertrains Using Stochastic Dynamic Programming," Intelligent Transportation Systems, IEEE Trans. on, Vol.8, No.1, pp. 71-83, 2007.

[14] C. Romaus, K. Gathmann, and J. Böcker, "Optimal Energy Management for a Hybrid Energy Storage System for Electric Vehicles Based on Stochastic Dynamic Programming," In Vehicle Power and Propulsion Conf. (VPPC 2010), IEEE, 2010.

[15] S. Ross, J. Pineau, S. Paquet, and B. Chaib-draa, "Online Planning Algorithms for POMDPs," J. of Artificial Intelligence Research, Vol.32, pp. 663-704, 2008.

[16] C. Sondermann-Wölke and W. Sextro, "Integration of Condition Monitoring in Self-Optimizing Function Modules Applied to the Active Railway Guidance Module," Int. J. on Advances in Intelligent Systems, Vol.3, No.1-12, pp. 65-74, 2010.

[17] C. Sondermann-Wölke, J. Geisler, T. Müller, A. Trächtler, and J. Böcker, "The active guidance module of a rail-bound vehicle as an application for the dependability oriented design in self-optimizing systems," In ASME 2008 Int. Design Engineering Technical Conf. & Computers and Information in Engineering Conf., 2008.

[18] "ISO 17359, Condition monitoring and diagnostic of machines – general guidelines," 2003.

[19] C. Henke, N. Fröhleke, and J. Böcker, "Advanced Convoy Control Strategy for Autonomously Driven Railway Vehicles," IEEE Conf. on Intelligent Transportation Systems, 2006.

[20] H. Grotstollen, "Design of Long-Stator Linear Motor Drives for RailCab TestTrack," J. of Power Electronics (JPE), KIPE, Vol.5, pp. 166-172, 2005.

[21] T. Schneider, B. Schulz, C. Henke, K. Witting, D. Steenken, and J. Böcker, "Energy Transfer via Linear Doubly-Fed Motor in Different Operating Modes," In Int. Electric Machines and Drives Conf. (IEMDC), 2009.

[22] C. Romaus, J. Böcker, K. Witting, A. Seifried, and O. Znamenshchykov, "Optimal Energy Management for a Hybrid Energy Storage System Combining Batteries and Double Layer Capacitors," In Energy Conversion Congress and Exposition (ECCE), 2009.

[23] H. Vöcking and A. Trächtler, "Self-optimization of an Active Suspension System Regarding Energy Requirements," In Int. Conf. on Control, Automation and Systems 2008, 2008.

[24] W. v. Wezel and R. Jorne, "Paradoxes in Planning," Engineering Applications of Artificial Intelligence, Vol.14, pp. 269-286, 2001.

[25] D. Warren, "Generating Conditional Plans and Programms," In Proc. of the summer Conf. on AI and Simulation, 1977.

Table 2. Operation modes derived from multiobjective optimization of the active suspension module: f_1 in m/s^2 an f_2 in $w-s$.

OM	Track Type I		II		III		IV		V		VI		VII		VII		IX		X	
	f_1	f_2	f_1	f_2	f_1	f_2	f_1	f_2	f_1	f_2	f_1	f_2	f_1	f_2	f_1	f_2	f_1	f_2	f_1	f_2
a	0.117	196	0.233	393	0.350	589	0.466	786	0.583	982	0.699	1179	0.816	1375	0.932	1572	1.049	1768	1.166	1965
b	0.152	165	0.304	329	0.457	494	0.609	659	0.761	823	0.913	988	1.066	1153	1.218	1317	1.370	1482	1.522	1647
c	0.192	142	0.385	283	0.577	425	0.770	567	0.962	709	1.155	850	1.347	992	1.540	1134	1.732	1275	1.925	1417
d	0.224	122	0.449	245	0.673	367	0.897	489	1.122	612	1.346	734	1.570	856	1.794	979	2.019	1101	2.243	1224
e	0.262	104	0.523	208	0.785	313	1.047	417	1.308	521	1.570	625	1.832	730	2.093	834	2.355	938	2.617	1042
f	0.298	87	0.595	173	0.893	260	1.191	346	1.488	433	1.786	520	2.084	606	2.381	693	2.679	779	2.977	866
g	0.331	69	0.662	138	0.994	206	1.325	275	1.656	344	1.987	413	2.318	482	2.649	550	2.981	619	3.312	688
h	0.375	50	0.749	99	1.124	149	1.499	199	1.873	248	2.248	298	2.623	348	2.997	398	3.372	447	3.747	497
i	0.435	27	0.870	55	1.305	82	1.739	110	2.174	137	2.609	164	3.044	192	3.479	219	3.914	247	4.349	274

[26] J. Blythe, "Artificial intelligence today: recent trends and developments," chapter An overview of planning under uncertainty, pp. 85-110, Lecture Notes in Computer Science, Springer-Verlag, Berlin, Heidelberg, 1999.

[27] R. Dearden, "Planning and learning in hybrid discrete domains – continuous models," In 20th Int. Joint Conf. on Artificial Intelligence, 2005.

[28] I. Ben-Gal, "Encyclopedia of Statistics in Quality & Reliability," chapter Bayesian Networks, Wiley & Sons, 2007.

[29] J. Pearl, "Probabilistic Reasoning in Intelligent Systems – Networks of Plausible Inference," Morgan Kaufmann, 1988.

[30] W. Dangelmaier, B. Klöpper, and A. Blecken, "An Agent Based Modeling Approach for Stochastic Planning Parameters," In HoloMAS 2007, Vol.4659 of Lecture Notes in Artificial Intelligence, pp. 225-236, 2007.

Appendix A. Operation Modes and Track Sections for Experiments

Table 2 shows the operation modes used in the experiments. They were derived from a multi objective optimization for the active suspension module. Different track types were generated by varying the amplitude of the track excitation profile. During experiments, it is assumed that all track section have the same length.

Name:
Benjamin Klöpper

Affiliation:
National Institute of Informatics

Address:
Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan

Brief Biographical History:
2005-2009 Sub-Project Coordinator "Behavior-based Self-Optimization" within Collaborative Research Center 614
2009 Received Ph.D. for "Planning for intelligent Mechatronic Systems"
2009-2011 Visiting Researcher, the Honiden Laboratory, National Institute of Informatics

Main Works:
- planning for self-optimizing mechatronic systems
- self-healing systems
- scheduling for self-optimizing manufacturing systems

Membership in Academic Societies:
- The Institute of Electrical and Electronics Engineers (IEEE)

Name:
Christoph Sondermann-Wölke

Affiliation:
Mechatronics and Dynamics, University of Paderborn

Address:
Fürstenallee 11, Paderborn 33103, Germany

Brief Biographical History:
2007 Received Dipl-Ing. degree in mechanical engineering, University of Paderborn, Germany
2007- Scientific Assistant & Ph.D. Student, Mechatronics and Dynamics, University of Paderborn, Germany

Main Works:
- dependability of self-optimizing systems
- condition monitoring and reliability assessment

Membership in Academic Societies:
- The Association of German Engineers (VDI)

Name:
Christoph Romaus

Affiliation:
Power Electronics and Electrical Drives, University of Paderborn

Address:
Warburger Straße 100, Paderborn 33098, Germany

Brief Biographical History:
2004 Received Dipl-Ing. degree in Electrical Engineering, RWTH Aachen University, Germany
2005-2010 Scientific Assistant & Ph.D. Student, Power Electronics and Electrical Drives, University of Paderborn, Germany
2008-2010 Sub-Project Coordinator "Interconnected Self-Optimizing Modules and Systems" and coordinator of several workgroups, Collaborative Research Center 614
2010- Senior Engineer, Power Electronics and Electrical Drives, University of Paderborn, Germany

Main Works:
- self-optimization control strategies
- energy management of a hybrid energy storage

Membership in Academic Societies:
- The Institute of Electrical and Electronics Engineers (IEEE)
- German Association for Electrical, Electronic & Information Technologies (VDE)
- Power Engineering Society (ETG)