DISTRIBUTED OPTIMIZATION OF REFERENCE TRAJECTORIES FOR ACTIVE SUSPENSION WITH MULTI-AGENT SYSTEMS

Eckehard Münch
Oliver Oberschelp
Thorsten Hestermeyer
Mechatronics Laboratory Paderborn -
University of Paderborn
Pohlweg 98, D - 33098 Paderborn, Germany
E-Mail: Oliver.Oberschelp|Thorsten.Hestermeyer@mlap.de
Eckehard.Muench@mlap.de

Peter Scheideler
Andreas Schmidt
Heinz Nixdorf Institute -
University of Paderborn
Fürstenallee 11, D - 33102 Paderborn, Germany
E-mail: Peter.Scheideler|Andreas.Schmidt@hni.upb.de

KEYWORDS

multi-agent systems, mechatronics, distributed optimization, railway systems

ABSTRACT

Self-optimization in mechatronic systems is not restricted to inner processes within a technical system. By communication between different technical entities of the same or similar kind using appropriate communication structures, a technical system can use the experiences of other entities to optimize its own behavior. Multi-agent systems together with agent-based controllers are an excellent means to model the software of such collaborating systems. This paper suggests a multi-agent system for the self-optimization of the active suspension system of small railway vehicles. It also presents an architecture for the distributed optimization of this complex mechatronic system. The shuttles use their experience to find an optimal trajectory for the active suspension along the track, thus using the ability of the active suspension to deviate a small distance from the actual track trajectory.

INTRODUCTION

Mechatronic systems are complex technical systems whose dynamics are controlled using actuators, sensors and modern information processing. The continuous improvement of microprocessors and control units allows a more and more complex information processing, which is used to further improve the dynamic behavior of mechatronic systems. In the end, systems are supposed to adapt to changing environmental and internal conditions on their own, which leads to self-optimizing systems (SFB614 2004; SFB376 2004). In order to achieve this goal it is necessary to combine modern control theory with methods of artificial intelligence. The optimization can be done locally within the control system, where each controller is managed, monitored and optimized by its own agent (Oberschelp et al. 2002). These systems will be called agent-based controllers from here on.

It is quite obvious to interconnect agent-based controllers, in order to exchange experiences between systems of the same kind or to reformulate the objectives. While doing so it is of utter importance to consider as many influences as possible. The collaboration of agent-systems leads us to multi-agent-systems (Ferber 1999). The following pages present a concept that shows self-optimization by multi-agent-systems. As application example the authors choose the *Railcab*-system (Railcab 2004) - a novel railway-system based on driverless autonomous shuttles driven by linear motors, which transport passengers and goods directly to their respective final destinations. The focus lies not on the logistics but rather on the organization of knowledge within a complex, distributed system and how to use this information to optimize the dynamic behavior of the system.

The system behavior can be improved in many ways. The physical plant, consisting of actuators, sensors and supporting mechanical structure, can be improved, but usually not during operation. The main handle for optimization is thus the controller. Here, usually the controller parameters are changed. The overall behavior of a system depends however not only on internal processes, but also on external reference values. Many control systems have to move from one state to another by moving along a given trajectory (Föllinger 1994). If this trajectory can be chosen freely, as with industrial robots, the overall behavior of the system can be changed by optimizing the trajectory. If the quality of the trajectory can be assessed by given accessible objectives, optimization can be done with a model or directly with the real system.

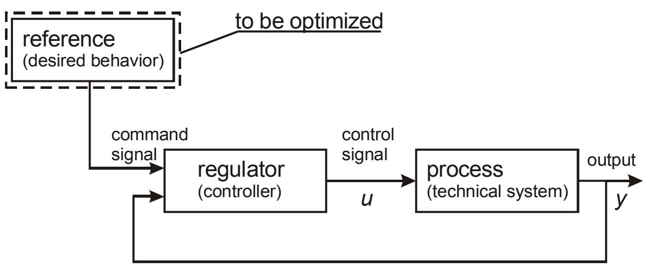

Figure 1. Common Control Structure

Fig. 1 shows the general structure of a control loop according to (Aström et al. 1989). The procedure described in this paper optimizes the desired behavior by means of the modification of trajectory defaults.

Optimization of a process can be performed online only if

Proceedings 18th European Simulation Multiconference
Graham Horton (c) SCS Europe, 2004
ISBN 3-936150-35-4 (book) / ISBN 3-936150-36-2 (CD)

the process can be repeated often enough. The recorded values of computed objectives for each repetition can be called *experience* of the technical system, that can be weighted and used to optimize the system behavior. In the example here a shuttle shall run along a fixed track sector with optimum comfort, limited only by the technical constraints of the suspension. As all shuttles are considered here to be basically the same, the experience of the shuttles can be used for optimization. Each shuttle is thus a probe for comfort. In order to use the shuttle experience it is necessary to communicate its knowledge and optimize in a distributed way.

In conventional systems the reference trajectory is computed offline based on static data (Wahl and Gilles 2003). In addition to this it is even not usual to control behavior of an active suspension via given trajectory (Streiter 1996). The usage of a static approach is only feasible if the environment is static and easy to measure. A centralized trajectory optimization on a global model is also possible but due the fact that the optimization problem results into many local optimization problems a decentralized optimization is more sufficient.

This paper presents a multi-agent-system for distributed optimization of trajectories, which practically corresponds to learning through individuals.

The remainder of the paper is organized as follows:

First we give an overview of the railway system and classify its relations to multi-agent systems. Then we present the active suspension system in detail. After this we discuss the use of the multi-agent approach to optimize the course planning. At last we give a short conclusion and an outlook to our future work.

STRUCTURE OF RAILWAY SYSTEMS

Modern railway systems have to compete with individual transport in respect of comfort, flexibility and cost. A research project which accepts the challenge is the Neue Bahntechnik Paderborn. The project has been initiated and worked upon by several departments of the University of Paderborn and the Heinz Nixdorf Institute. In the project, a modular rail system is being developed; it is to combine modern chassis technology with a new linear motor similar to that of the Transrapid and the use of existing rail tracks. The interaction between information technology and sensor/actuator technology paves the way for an entirely new type of mechatronic rail system. The vehicles designed apply the linear drive technology used in the Transrapid, but travel on existing rail tracks. The use of existing rail tracks will eliminate an essential barrier to the proliferation of new rail bound transport systems (Hestermeyer 2003).

Distributed Information Processing. While classic railway-systems use centralistic approaches to control the trains on the railway network, this is virtually impossible with the high number of vehicles in the NBP system which move without time table on routes according to their demands. Thus, several decentralized approaches are being tested for realization of the logistic system, which manage shuttles and track using varying structures. However, all have in common, that the overall functionality is structured in a decentralized and function-oriented way. This requires the task distribution to different autonomous components, which have to exchange information, execute plans and perform actions on their own initiative. An approach that can fulfil these requirements on the information processing is the multi-agent-technology. It is therefore obvious to model the whole information processing on the planning level according to this scheme (Epple 2000).

The example in this paper uses the following hierarchical structure: A regional track network forms a *region* on the uppermost level. This region comprises a multitude of stations, junctions and tracks, which are again assigned to a *local region*. The local region is arranged in stations, switches and *track segments*. The latter are formed by *track sections* between stations and switches. This example concentrates only on the level track segments and track sections.

The overall system is administered by a multi-agent system, which performs partly hierarchically, partly cooperatively functions of the administration, as track allocation, energy management or shuttle localization. As a consequence, the resulting agent structure corresponds to the hierarchical structure of the railway system. The shuttles themselves form autonomous elements within the system. The liberty is limited, however, as each shuttle has to submit to a local management and cannot act in a completely free way.

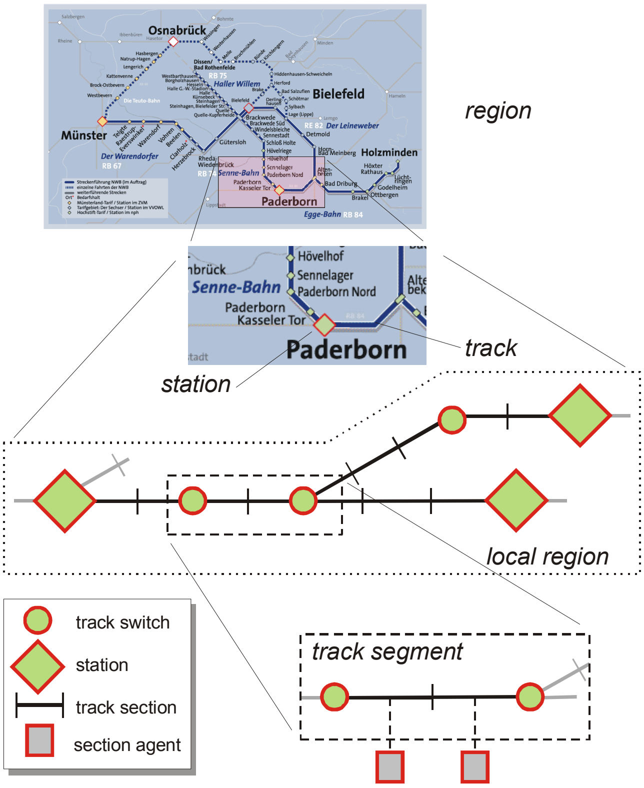

Figure 2. Hierarchical organization of the rail system

This information structure - set-up according to the functional and technical structure - now allows a clever distributed optimization using learning through individuals. The described approach assumes that a large number of vehicles can move nearly autonomously (not freely!) on a complex track network. During its journeys a vehicle collects data, which can be used as experience. This can be - like in the example here - measured data about the dynamic behavior during the transit of a certain track. This data is passed on to other vehicles. Separate measurement runs are not necessary any more.

At first look it seems a good idea to have each shuttle with all the others in order to exchange their experience. However, finding a shuttle with a certain feature, like one that has measured data about a certain visited track section is costly. In addition a shuttle has to save data about a given track section even though this data might be already obsolete because of experience made by other shuttles - an adjustment of the data of all shuttles would be necessary. It is therefore preferable to collect data at a track section and to have it administered by an agent that is assigned to a fixed local track section. Each passing shuttle communicates its experience in form of measurement data to this local knot and requests already processed experience of other shuttles.

The shuttles and track sections therefore form a multi-agent-system, that can be used for the management of the experience made by different shuttles. The connection of the agent technology to the technical processes in a shuttle requires a closer look of the technological and information structure of the vehicle and the integration in the overall system.

Structure of the Multi-Agent-System. Basically, the shuttles have a cognitive behavior to individually process the knowledge coming from other entities. However in this example an essential feature of a cognitive agent, its autonomy, is replaced with a Master-Slave-Cooperation (Nwana et al. 1996). For two agents connected by this relationship, the master sends its commands or arrangements to the slave and the slave will act as the master instructs, providing the feedbacks and results to the master. This means that the planning of the trajectory is done by the track sections (Masters). They send the trajectory parameters to the shuttles (Slaves) which have to execute them (see Fig. 3).

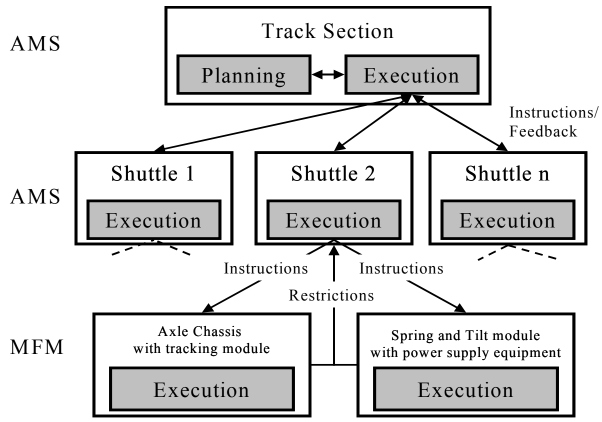

Figure 3. Master-Slave-Cooperation

Agent Micro-Achitecture. The software realization of the described structure requires a modular approach that fulfills the requirements of control theory (real time requirements, stability) while offering a link to multi-agent systems. In addition, single modules shall also be equipped with the ability for self-optimization. An approach to fulfill this requirement is the operator-controller-module. It is a structure-concept for the development of agent-based controllers (Oberschelp et al. 2002).

The basic idea is the separation of the control code into three different levels, which have different requirements. The lower level motor level implements the basic functions of the control which have direct impact on the technical process. This level is managed by the reflective operator, which is located on the next level. Control of reconfiguration within the controller, which includes emergency routines, but also the realization of parameter changes are some of the tasks of this operator.

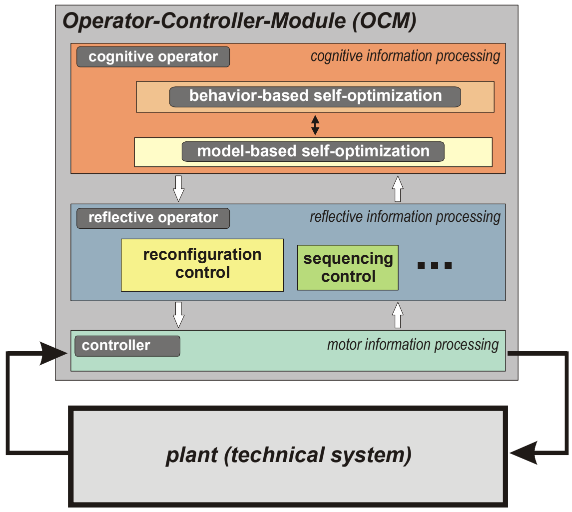

Figure 4. Scheme of the Operator-Controller-Module

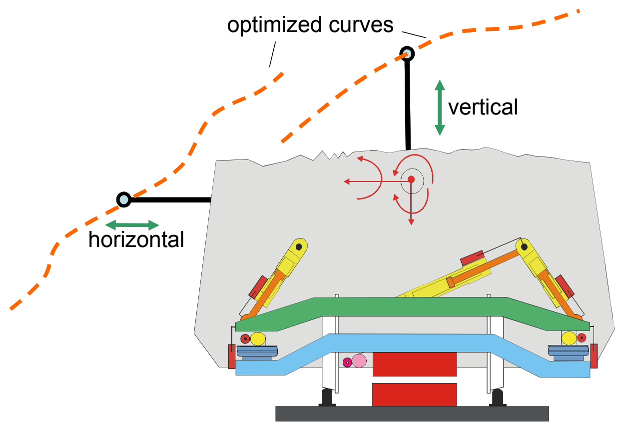

Figure 5. Trajectory tracking for the body movement

On the top level we have the cognitive operator. This element is of utmost importance, since it is here, where the actual agent functions are located. The cognitive operator optimizes the system and also includes the interface to other agents. While controller and reflective operator are subject to hard real-time conditions, the cognitive operator can work asynchronously to real time. The reflective operator manages the change suggestions of the optimization in the cognitive operator and realizes them when suitable conditions (e.g. a certain controller state) arise. This ensures a save controller operation (Oberschelp et al. 2004).

ACTIVE SUSPENSION WITH TRAJECTORY TRACKING

Trajectory Tracking. Trajectory Control systems do not only serve to make a dynamic system insensitive to disturbances and to keep it within a well-defined state, but also to control a dynamic process. An example for this are machine tools: The tool must run along a given programmed trajectory despite all disturbing external forces; the controller provides a decoupling of the trajectory from the dynamics of the machine tool. In many technical applications, however, the trajectory is linked to the dynamics of the machine. Manipulators position a part at a given position. The trajectory can be chosen freely except for its endpoints, the work space of the robot and constraints like collision avoidance. In order to determine an appropriate trajectory, dynamic properties of the manipulator are considered in the optimization. With the dynamic properties, the quickest trajectory or the trajectory with the lowest energy consumption can be determined.

The approach can also be applied to active suspensions (see Fig. 5).

Active Suspension. Today, active suspensions can be found in upper-class cars and in some trains. In cars they are used to increase comfort while increasing safety at the same time (Streiter 1996). In trains active suspensions are before all used for tilt in order to be able to reach higher curve velocities. The NBP suspension system integrates both properties in an overall concept. The active suspension developed there allows the modification of lateral and vertical dynamics and also active tilt (RailCab 2004). It is based on air springs, which can be damped actively by base displacement of the springs. It dispenses completely with passive dampers parallel to the springs. Thus the car body is linked to the undercarriage and the train exclusively by air springs, resulting in an excellent decoupling of the car body from high-frequent excitation of drive or rails. The active control of the spring base displacement can be restricted to relatively low frequencies around the natural frequency of the car body. All in all this system can improve the ride comfort noticeably especially with higher frequencies.

The employed air spring bellows have as well horizontal as vertical stiffness so that excitations from all spatial directions can be cushioned. The active spring base displacement is done by hydraulic cylinders. Three hydraulic cylinders, arranged in a plane, move the bases of the air springs via an intermediate frame, the “suspension frame”. This arrangement allows damping forces in lateral and vertical directions (see Fig. 6). In addition it is also possible to regulate the level of the car and add active tilting of the car body.

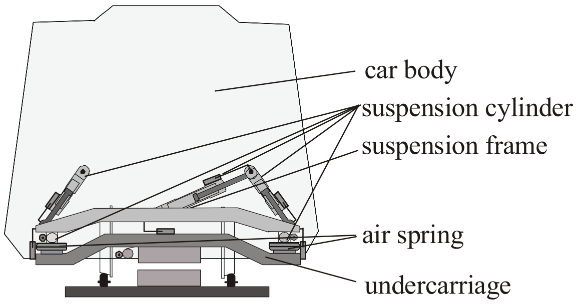

Figure 6. Active suspension system

Suspension Control. The classical control algorithm of an active suspension system uses acceleration measurements to determine the absolute velocity of the car body and displacement measurements to determine the distance between track and car body (relative position of the car body). The relative position of the car body and its first time derivative can be used to create pseudo spring and damper forces by creating additional airspring displacements proportional to the desired forces. The absolute car body velocity forms a virtual damper attached to a so called "sky hook", which suppresses the movement of the car body. For comfort reasons the sky-hook damping should be as large as possible. However, as the sky-hook damper suppresses the car body movement, a vehicle with large sky-hook-damping is not able to move up a slope or to go around a curve - after all, this requires vertical or lateral movement, respectively. If the sky-hook damper is linked to a trajectory along the optimal way across a slope or around a curve, only movement apart from the necessary movement for the desired motion would be suppressed. This technique makes it possible to have large sky-hook damping.

AGENT-BASED TRAJECTORY PLANNING

This section demonstrates how a self optimized trajectory planning can be modelled in a Multi-Agent-System.

The optimization process. The optimization is based on the measured data of the actual trajectory and the actual acceleration of the body. The aim is to maximize the comfort, which is often defined via the dying out behavior. A simple evaluation function can be obtained from the quadratic defective area against the rest position in the z-direction against the location x.

$$\varepsilon^2{}_{G_K} = \int_{(x=0)}^{(x=x)} \{(g_1 z_A)^2 + (g_2 \ddot{z}_A)^2\} dx \rightarrow min \quad (1)$$

$\varepsilon^2{}_G$ is the measure of comfort, z_A is the body position and $\ddot{z}_A$ is the body acceleration. g_1 and g_2 are weighting factors.

The gain of comfort can most simply described as the reduction of the body acceleration if the relative amplitude does not have to be considered ($g_1 = 0$). What we need, for the iterative improvements, is a correlation between the trajectory and the vertical acceleration to vary the trajectory. The improvements of the trajectory can be measured through the comparison of the actual acceleration data versus the previous ones. Therefore we divide the sections, as shown in Fig. 7, in intervals (n) and calculate the quadratic deviation between the previous acceleration curve and the actual acceleration curve. The result is the measure of improvement or worsening and can be directly used for the calculation of the new interpolation points for the trajectory..

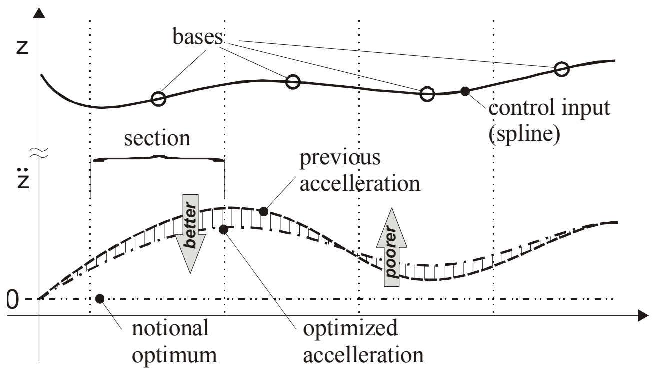

Figure 7. To optimized trajectory

The evaluation function of the variance of acceleration is:

$$\Delta\varepsilon^2{}_{G_K} = \int_{x=x_n}^{x=x_{n+1}} \ddot{z}_{A,P}{}^2 dx - \int_{x=x_n}^{x=x_{n+1}} \ddot{z}_{A,N}{}^2 dx \quad (2)$$

The signed result is a criterion if the last step was successful or not. A positive algebraic sign means success, a negative one failure. The amplitude of the change can be used for the calculation of the next increment. Here the step Δz is a function of the defective area and the previous increment Δz_p.

$$\Delta z_n = \mathrm{sgn}(\varepsilon_{G_K}) \cdot q \cdot \varepsilon^2{}_{G_K} \cdot \Delta z_p \quad (3)$$

q is simply a weighting factor which can be varied against the number of successful steps. $\mathrm{sgn}(\varepsilon_{G_K})$ defines the change of the direction and $\varepsilon^2{}_{G_K}$ the change of increment.

The search algorithm. The here used method is a modified Hill-Climbing-Search Algorithm (Russel and Norvig 2003). We don't need to start with a random trajectory but with a mathematically optimized one calculated by the track section, if there are no informations about the track characteristic available, the algorithm starts with zero-trajectory. The shuttle follows this trajectory and sends the experienced vertical acceleration during the ride back to the track section (see Fig. 8). Based on an assessment of the actual acceleration versus the previous acceleration, the track section calculates a new trajectory for the next shuttle. The problem that the Hill-Climbing-Search Algorithm can stay in a local maximum/minimum is ignored since our only intention is to improve the offline mathematically optimized trajectory (even though we know that we could overcome this obstacle by using Simulated Annealing which combines the efficiency of Hill-Climbing with the completeness of a random walk). However we have to consider two important aspects in our scenario. The first is that our algorithm has to remember the history to decide the intensity and the direction of the search. In case it has experienced a worse state the algorithm jumps back to the last best state and adjusts its search parameters. This process is continued until a defined stop criterion is met. The second aspect is the occurrence of dependent states (interpolation points). The change of one state (interpolation point) can have significant negative and positive effects on the other ones.

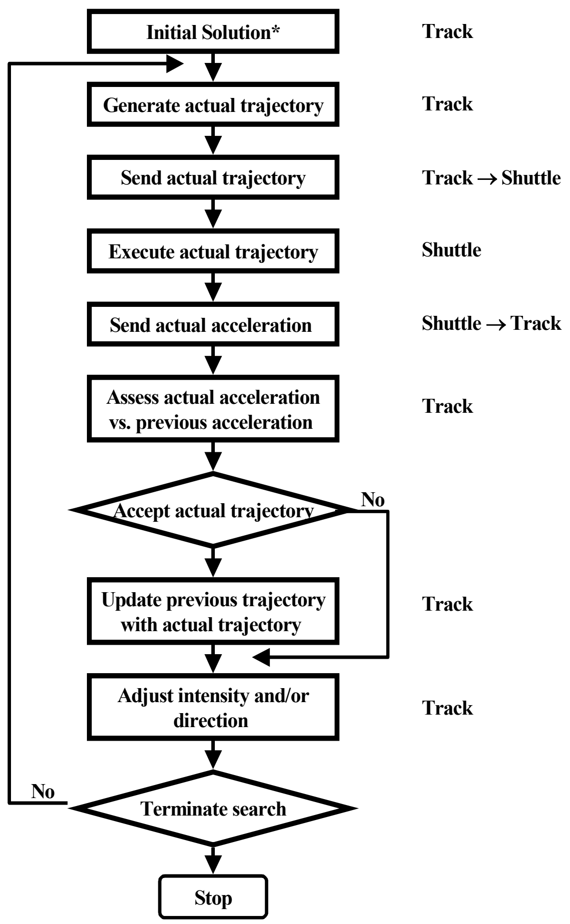

Figure 8. Structure of our algorithm

The optimization process. The trajectory parameters are the interpolation points of a spline-curve. A vehicle that approaches a track section receives the interpolation points for the default trajectory from the section administration. During the passage over the track section data about the vertical acceleration of the body is measured and sent to the respective section administration.

Since for the default trajectory only continuous functions are permitted, a compact description by spline-curves is sufficient for this scenario. Here one date per interpolation point is sufficient if only equidistant interpolation points are permitted.

SIMULATION ENVIRONMENT

In the scenario described above we have to combine discrete and continuous simulation. We model the technical system by means of ordinary differential equations (ODE). The Computer-Aided Engineering (CAE) tool CAMeLView (iXtronics 2003) automatically derives differential equations for the mechatronic part which is described by multibody system models .

For this purpose, nonlinear differential vector state equations are used as follows :

$$\dot{\underline{x}} = \underline{f}(\underline{x}, \underline{u}, \underline{p}, t) \tag{4}$$

$$\underline{y} = \underline{g}(\underline{x}, \underline{u}, \underline{p}, t) \tag{5}$$

with $\dot{\underline{x}}$ the state vector, $\underline{y}$ the output vector, $\underline{u}$ the input vector, and t the time.

We use ODE solvers which are a part of the CAE environment for the computation of the model. Since the used solver fulfills real-time requirements we are able to use the same model in offline as in online simulation (Hardware-in-the-loop).

In addition we choose the IPANEMA library as the simulation platform. IPANEMA allows the distributed real-time simulation of ODE-based models as well as finite-state machines. In addition it is possible to include external C-code (Gambuzza and Oberschelp 2003). To fulfill the requirements of an agent-based environment IPANEMA is extended by the PUB library (Paderborn University BSP-library) which allows an automatic distribution to massively parallel clusters for a realistic parallel simulation.

The PUB library simplifies the implementation of massively parallel programs according to the BSP model (bulk-synchronous parallel model). The BSP model subdivides a program sequence into so-called supersteps. At the end of a superstep a synchronisation between single parallel processes take place. During these synchronizations, messages from the previous superstep are received. For this purpose the process will be suspended. The next superstep is executed after all messages are received (Bonorden 1999).

For the simulation of mechatronic systems, modelled as time-based ODE models, the BSP model is very suitable. To each IPANEMA model a single process is allocated. The synchronisation in time is realized by means of the synchronization mechanism of the PUB library. During a single superstep IPANEMA models are evaluated for the time interval ΔT. During synchronization messages between agents can be exchanged.

SIMULATION RESULTS

For an example, the optimization of a trajectory with track disturbance (track irregularities) is shown. Under the assumption of an ideal reference reaction of the control loop, it is sufficient to add the second derivative of the track disturbance to the acceleration of the body mass, which corresponds to the second derivative of the default curve. With the chosen distance of the spline bases, the reference reaction can be assumed to be ideal. With shorter distances a stronger curvature of the curve is possible, which requires a higher bandwidth of the actuator. The result of the addition is an incompletely corrigible disturbance of the comfort which, however, can be compensated for by the optimization.

Due to the chosen boundary conditions an analytical optimization of the curve can also be performed. In our case this is desired, since now the quality of the result can be examined more easily. In reality the disturbance is essentially unknown and the reference reaction of the body mass is not ideal.

In the following example the optimization of a section consisting of ten segments was accomplished. Fig. 9 shows the trajectory before and after optimization. The original curve displays relatively strong curvatures. By the optimization this curve can be smoothed to a high degree in approximately 65 steps:

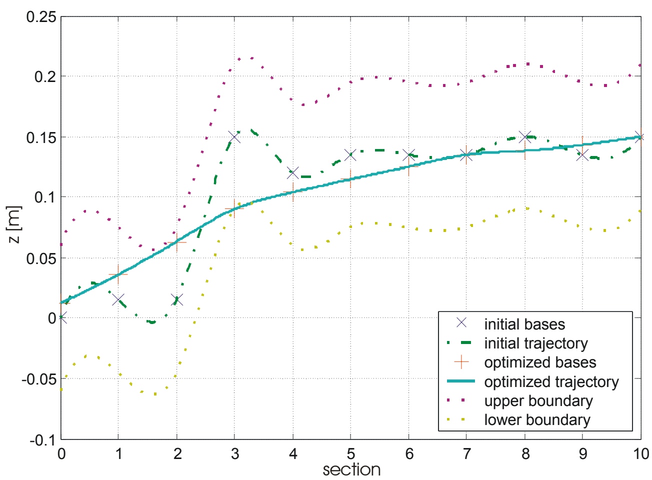

Figure 9. Initial and optimized trajectory

Fig. 10 represents the acceleration of the body mass. As can be seen, the accelerations are very strongly reduced. Since an ideal reference reaction was assumed, the acceleration of the body mass coincides precisely with the second derivative of the trajectory:

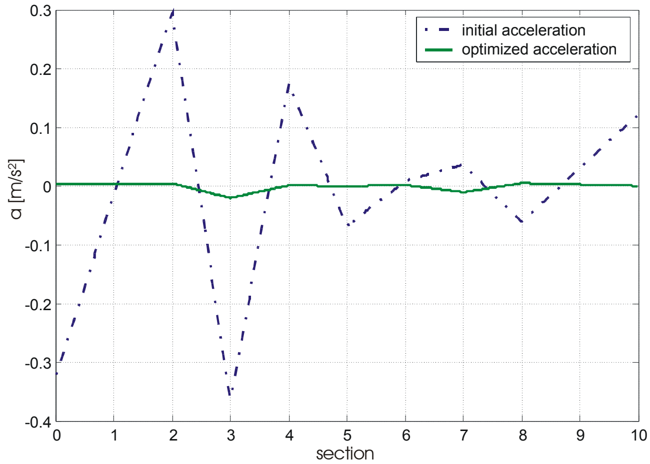

Figure 10. Initial and optimized acceleration

EXCURSUS

The active adjustable stroke of the body kinematics is appr. 12 cm; if one considers the accuracy of the track and acceleration sensors a resolution of 16 bit is more than sufficient. In the example the distance between the interpolation points is 6 metres. Therefore, at a total length of 600 metres, 100 parameters per section are needed; this corresponds to a data volume of 400 bytes per section. In our example the shuttle has a maximum speed of 200 km/h. Thus the shuttle needs 11 sec. to pass the whole length which demands a transmitting of at least 37 bytes per second. Afterwards the shuttle transmits the gauged data of the actual acceleration back to the track section. In this scenario the entire net data flow from the shuttle and back adds up to 111 bytes per second.

CONCLUSION

The paper shows how distributed learning with multi-agent systems can be used for the optimization of dynamic processes in complex technical systems. The distributed optimization is especially suitable for systems, where experience of many individual systems can be used and is thus applicable for other similar applications.

ACKNOWLEDGEMENTS

This work was developed in the course of the "Collaborative Research Center 376 - Massively Parallel Computing - Algorithms, Design Methods, Applications" - University of Paderborn, and was published on its behalf and funded by the Deutsche Forschungsgemeinschaft.

REFERENCES

Bonorden, O.; B. Juurlink; I. von Otte and I. Rieping. 1999. "The Paderborn University BSP (PUB) Library - Design, Implementation and Performance". In *Proceedings of the International Parallel Processing Symposium* IPPS, San Juan, Puerto Rico (Apr).

Epple, U. 2000. "Agentensysteme in der Leittechnik", *Automatisierungstechnische Praxis (atp)* 42(2000) 8, Aachen.

Ferber, J. 1999, *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*, Addison-Wesley.

Föllinger, O. 1994. *Regelungstechnik*, Hüthig, Heidelberg.

Gambuzza, A. and O. Oberschelp. 2003. "Distributed Modular Simulation of Mechatronic Systems". In *Proceedings of the European and Modelling Conference*, Naples.

Lückel, J.; T. Hestermeyer and X. Liu-Henke. 2001. "Generalization of the Cascade Principle in View of a Structured Form of Mechatronic Systems". In *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics AIM 2001*, Como.

Wahl, A. and E. D. Gilles. 2003. "Optimale Regelverfahren zur automatischen Bahnführung von Binnenschiffen", *Automatisierungstechnik (at)* 51(2003) 6, pp. 255-264.

Nwana, H.; L. Lyndon and N. Jennings. 1996. "Co-ordination in software agent systems. *The British Telecom Technical Journal* 14(4), pp. 79-88.

Oberschelp, O.; H. Hestermeyer; B. Kleinjohann and E. Kleinjohann. 2002. "Design of Self-Optimizing Agent-Based Controllers" In *Proceedings of the CfP Workshop 2002 - Agent-Based Simulation 3*, Passau.

Oberschelp, O.; H. Hestermeyer and H. Giese. 2004. "Strukturierte Informationsverarbeitung für selbstoptimierende mechatronische Systeme" In *Proceedings of the Intelligente Mechatronische Systeme 2*, Paderborn.

RailCab. 2004. *RailCab Project Homepage*, http://nbp-www.upb.de/en/concept/challenge.php, Paderborn.

Schüssler, L. 1985. "Aktive Federungssysteme, Regelungskonzept zur Ansteuerung der aktiven Kraftstellsysteme (AKTAKON)". Technical Report Nr. 85/008, Daimler-Benz AG, R/D, Stuttgart.

SFB376. 2004. *Collaborative Research Centre 376 Homepage,* http://www.uni-paderborn.de/sfb376, Paderborn.

SFB614. 2004. *Collaborative Research Centre 614 Homepage,* http://www.sfb614.de/eng, Paderborn.

Streiter, R. H. 1996. "Entwicklung und Realisierung eines analytischen Regelkonzeptes für eine aktive Federung". PhD thesis., Technical University Berlin

Russel, S. J. and P. Norvig. 2003. *Artificial Intelligence*, Prentice Hall International.