Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Multiobjective Optimization of Control Trajectories for the Guidance of a Rail-bound Vehicle

M.Sc. Jens Geisler * **Dipl.-Math. Katrin Witting** **
Prof. Dr.-Ing. Ansgar Trächtler * **Prof. Dr. Michael Dellnitz** **

* *Control Engineering and Mechatronics (RtM), University of Paderborn, Germany (e-mail: {jgeisler|ansgar.traechtler}@rtm.upb.de)*
** *Faculty of Computer Science, Electrical Engineering and Mathematics, University of Paderborn, Germany (e-mail: {dellnitz|witting}@math.upb.de)*

Abstract: Self-optimization refers to the ability of a mechatronic system to autonomously adapt the way it performs its functions to changing environmental and operational conditions or user demands. In this work we propose to use multiobjective optimal control to enable the self-optimization of the guidance of a rail-bound vehicle. We consider different strategies to reduce the computational cost of the optimization. Most importantly, a two-degree-of-freedom controller is used to separate optimal trajectory generation from disturbance compensation. Also, in order to solve the multiobjective optimization problem, an approximation of the entire set of optimal compromises of the objectives, the so-called Pareto set, is computed offline at design time. From this, we can derive a collection of weighting vectors that capture the best trade-off between the objectives for different situations. Given this set of preselected weights, for the online optimization, the objective function can be taken to be a weighted sum that best matches the situation at hand. For the guidance system we consider three objectives. Preliminary offline simulation results are presented.

Keywords: Mechatronic systems, Guidance and control, Model predictive and optimization-based control, Trajectory generation, Self-optimization, Multiobjective optimization

1. INTRODUCTION

Modern mechatronic systems benefit from the rapidly advancing capacities of information processing and microcontrollers. Mechatronic systems of tomorrow will have the inherent ability to adapt their structures, objectives, behaviors, and parameters to changing environmental and operational conditions at runtime. These attributes are summarized in the term "self-optimization" (s.o.). In order to fully exploit the emerging possibilities, new approaches and new methods are necessary that go far beyond the currently familiar design methodologies. The collaborative research center 614 - "Self-optimizing concepts and structures in mechanical engineering" - was set up to create tools that support the development of such systems (cf. Frank et al. (2004)).

In this work we present a new design pattern for s.o. systems. The idea is to extend receding-horizon optimal control to the explicit multiobjective case.

In contrast to classical (scalar) optimization where the global minimizer of one single function $f : \mathbb{R}^n \to \mathbb{R}$ is to be computed, several objective functions are taken into account in *multiobjective optimization* (cf. Miettinen (1999), Ehrgott (2000)). More precisely, a multiobjective optimization problem is given by

$$\min\{\underline{F}(\underline{x}) : \underline{x} \in S \subseteq \mathbb{R}^n\}, \qquad \text{(MOP)}$$

where $\underline{F}$ is defined as the vector of objective functions, $\underline{F}(\underline{x}) = (f_1(\underline{x}), \ldots, f_k(\underline{x}))$. S denotes the feasible region for $\underline{x}$. Here, the meaning of 'min' is based on the partial ordering $\leq_p$. A vector $u \in \mathbb{R}^k$ is defined to be $\leq_p$ a vector $v \in \mathbb{R}^k$, if $u_i \leq v_i \forall i = 1, \ldots, k$. A point $\underline{x}^\star \in S$ is called *(globally) Pareto optimal* for (MOP) if there is no $\underline{x} \in S$ with $\underline{F}(\underline{x}) \leq_p \underline{F}(\underline{x}^\star)$ and $f_j(\underline{x}) < f_j(\underline{x}^\star)$ for at least one $j \in \{1, \ldots, k\}$.

In case of our application, in a first step the entire Pareto set of solutions to the multiobjective optimal control problem is computed. In an a priori decision making, we choose several solutions within this set which suit different situations, such as dry or wet rails, different kinds of cross wind, etc. From the Pareto points chosen we can derive a weighted sum of the objectives whose optimization yields the same solution. The sets of weights are stored in the controller and then the resulting single-objective optimization problem can be efficiently solved online. Adapting the weighting of multiple objective functions is a powerful and intuitive method to manipulate the system behavior in the presence of changing conditions.

Along with the extension to several objective functions we also pay special attention to the fast system dynamics that is commonly found in mechatronic systems. The feedback

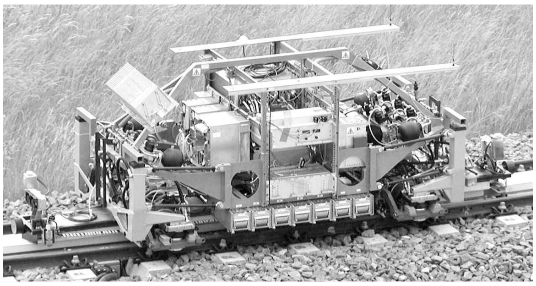

Fig. 1. Photograph of one RailCab vehicle without cover

control as well as the optimization are based on a linear single-track model.

Our test application is the guidance system of an innovative rail-bound vehicle, called RailCab (cf. Trächtler (2006) and `www.railcab.de`). RailCabs are small autonomous rail-bound vehicles that are propelled by a linear motor. They feature an active guidance system that enables them to determine the path through a new kind of passive switch at their own discretion. Furthermore, it is a major component for an increase in passenger comfort and a reduction of the wear on wheels and rails (cf. Ettingshausen et al. (2003)). An active suspension and tilting system provides for increased comfort even on badly maintained tracks and allows faster traveling speed in curves. A test track including two RailCabs on a scale of 1:2.5 was built at the University of Paderborn (see Fig. 1). The vehicles are designed to provide the amenities of passenger cars on the rails. As opposed to trains that run on a fixed schedule the RailCabs will operate on demand and provide an uninterrupted point-to-point journey without changeovers. At present, the RailCabs are used extensively as a testbed for different control strategies in the context of the CRC614 (e.g., in Trächtler et al. (2006)).

The paper is structured as follows: Section 2 describes the example application and the associated control problem. Then, the structure of the proposed self-optimizing controller is introduced in Section 3. The plant for the example control problem is the lateral vehicle dynamics detailed in Section 4. Section 5 deals with the objective functions and the calculation of the Pareto set as well as resulting control trajectories. Finally, some possible improvements are discussed.

2. PROBLEM SETUP

The purpose of the guidance module in the RailCab vehicles is to actively control the lateral displacement in the rails. Laterally a rail-bound vehicle can move within the clearance between the flanges and the rail-heads (Fig. 2). In traditional railway systems the coupled wheel-set with conic treads keeps the vehicle in an uncontrolled side-to-side motion that leads to the flanges striking periodically against the rail-heads. This causes noise as well as wear on the wheels and rails. Also, high lateral accelerations may occur which deteriorate driving comfort. To overcome these negative effects, the active guidance system avoids flange strikes and abrupt lateral accelerations by steering the wheels accordingly.

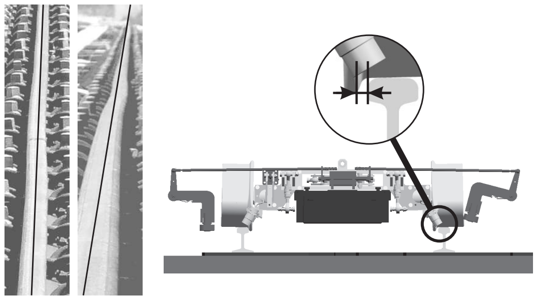

Fig. 2. Illustration of clearance between flange and rail-head and the bent/buckled rails

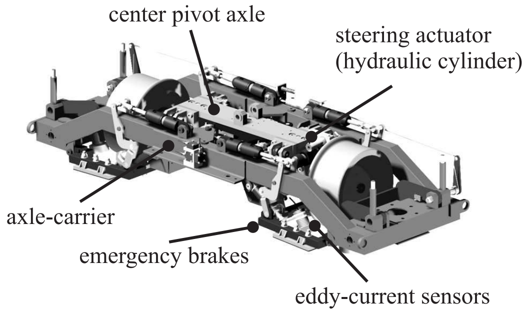

Fig. 3. CAD model of a single axle

2.1 The Guidance System for a Rail Vehicle

The guidance system is made up of two identical axle modules. Each module comprises a center-pivot axle with cylindrical loose wheels, an actuator, and several sensors (Fig. 3). The axle is pivoted by a servo-hydraulic cylinder with integrated displacement sensor. This allows direct control of the steering angle. Four eddy-current sensors per axle are used to measure the distance between flange and rail-head on either side. Further, the guidance system is equipped with a yaw velocity sensor and an accelerometer for the lateral acceleration of the center of gravity of the vehicle.

2.2 Multiobjective Control Problem

Real rails are not as ideally straight or curved as intended by their architect; instead they will bend and buckle (Fig. 2). Thus, following the course of the rails implies a constantly changing lateral acceleration of the vehicle which constitutes an unwanted disturbance to the passengers' comfort. But within the clearance we are free to choose an arbitrary reference trajectory which can be optimized with respect to several objectives. In this work we study the simultaneous optimization of safety, passenger comfort, and actuation energy.

3. OPTIMAL CONTROL FOR SELF-OPTIMIZING SYSTEMS

Self-optimization is the ability of a mechatronic system to variably adapt itself to different situations, environmental conditions, and user demands. One strategy for self-optimization is to adjust the priority of the control objectives to the situation at hand. The general idea of model-based optimal control is to use a plant model for predicting the performance of potential future control inputs usually over a finite horizon. In receding-horizon control, optimization over an infinite horizon is avoided by concatenating short sequences of optimal control inputs as time progresses. Each control subsequence is the first part of the solution to a finite-horizon open-loop optimal control problem. The rest of the solution is discarded. For each subproblem the current (estimated) system state is taken as the initial condition.

Usually all optimal control problems are formulated using multiple objective functions, which are often recast into a single, scalar objective function by using a weighted sum with fixed weights (e.g., in the LQR design). Since the optimization is solved online there is no objection to altering the weights on the fly as the circumstances change.

What we propose here is a receding-horizon control scheme with a superordinated decision heuristics to adjust the weighting of the objectives according to the current conditions and user demands. The details of the decision process shall not be of further interest in this discourse.

A similar systematic approach was proposed by Kerrigan and Maciejowski (2002). They use a priority-based rather than a weighting-based multiobjective optimization for resolving conflicting control objectives. In this context prioritizing means observing objective "B" only after objective "A" is at its best, rather than a mere trade-off between the two. To achieve this behavior a prioritized queue of single-objective functions is sequentially optimized with each set of solutions adding further constraints to the solution of the subsequent objectives.

This approach has certainly some very intriguing aspects as thinking in priorities often seems more natural than trading-off. But the sequential optimization of all objectives with a growing number of constraints comes at an increased computational expense. We expect that via an adequate design of the decision heuristics a similar behavior can be realized with variably weighted objective functions. If the Pareto set is known, arbitrary priority constraints can be imposed on the choice of the concrete solution.

3.1 The Two-degree-of-freedom Approach to Systems with Fast Dynamics

The most important drawback of many optimal control schemes – such as model predictive control (MPC) – is their computational expense which is due to the numerical optimization that has to be performed at every sampling instant (cf. Mayne et al. (2000)). For mechatronic systems with fast mechanical and electrical dynamics, computation time is an important issue.

Various propositions have been made on how to make MPC fit for the real-time control of systems with fast dynamics. But most publications focus on streamlining the optimization algorithms (see e.g. Wright (1997)). In many cases the gain in computational efficiency comes at the cost of further restrictions on the structure of the problem. This leads to many highly specialized algorithms for a very limited number of applications. Another approach is to avoid online optimization altogether by precomputing the so-called piecewise affine explicit form of MPC solutions

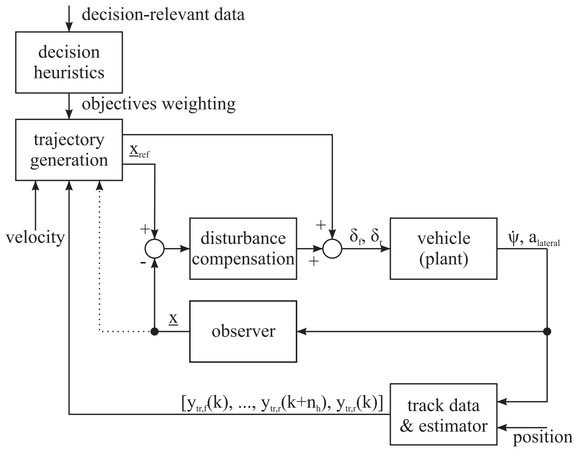

Fig. 4. Diagram of two-degree-of-freedom controller

resulting in a gain-scheduling control law (cf. Cairano et al. (2006)) so that the ability to tune the controller online is completely lost.

To avoid the necessity for online optimizations at the rate of the fast plant dynamics, Ronco et al. (2001) propose a combination of predictive control and linear state feedback in a so-called two-degree-of-freedom controller design. An example of a successful application of the two-degree-of-freedom control design can be found in Kehl et al. (2007) where this approach is used to realize an automatic steering control of a passenger car for reproducible path-following in aggressive test scenarios.

The structure of the controller we propose here is illustrated in Fig. 4. Disturbance rejection and regulation of the prediction error are handled by a fast gain-scheduled state feedback law that guarantees stability while demanding only little computational resources. Gain-scheduling is necessary to allow varying system dynamics at different vehicle speeds (see Sec. 4.1). Further details of the feedback controller will not be discussed here. For now we assume the closed-loop behavior to be ideal.

The role of the optimal predictive controller becomes that of a trajectory generator which provides feasible reference trajectories that satisfy the dynamic equations, system and actuator constraints, and optimally follow the command input. The trade-off between the different objectives is determined by the decision heuristics in response to the current situation.

The function of the observer is to reconstruct the complete system-state from noisy measurements. The "track data" supplies the trajectory generator with information about the course of the rails which can be iteratively estimated (cf. Münch et al. (2005)).

3.2 Fast Generation of Optimal Trajectories

Another drawback of traditional dynamic optimization is the explicit use of the states $\underline{x}$ and inputs $\underline{u}$ in the objective function which makes the computations very time-consuming. One very elegant way to reduce the number of optimization variables is to parametrize both states and inputs in terms of a differentially flat system output $\underline{y}$ and its derivatives up to a certain order q.

A system $\underline{\dot{x}} = f\left(\underline{x}, \underline{u}\right)$ with $\underline{x} \in \mathbb{R}^n$, $\underline{u} \in \mathbb{R}^m$ is said to be *differentially flat* if there exists a (fictitious) output $\underline{y} \in \mathbb{R}^m$ with the following properties (cf. Fliess et al. (1999)):

$$\begin{aligned} \underline{y} &= h\left(\underline{x}, \underline{u}, \underline{\dot{u}}, \underline{\ddot{u}}, \ldots, \underline{u}^{(p)}\right) \\ \underline{x} &= \alpha\left(\underline{y}, \underline{\dot{y}}, \underline{\ddot{y}}, \ldots, \underline{y}^{(q)}\right) \\ \underline{u} &= \beta\left(\underline{y}, \underline{\dot{y}}, \underline{\ddot{y}}, \ldots, \underline{y}^{(q+1)}\right) \end{aligned} \tag{1}$$

Thus, flatness is exploited by substituting all occurrences of states and inputs in the objectives and constraints by α resp. β. This eliminates the differential constraints from the optimization problem while the model equations do not need to be discretized or integrated either. The optimization problem is solved in the lower-dimensional space and then the optimal states and inputs are recovered from the inverse mapping. Chaplais and Petit (2003) state that this inversion greatly simplifies the optimization procedure. In Milam et al. (2000) this method is used in combination with a two-degree-of-freedom control design; its superior performance in comparison to a traditional MPC approach is demonstrated.

4. LATERAL DYNAMICS OF THE RAILCAB VEHICLE

The plant model is the central element of model-based control. In our case of a rail-bound vehicle with two steerable axles, a linear model of 4th order is used.

4.1 The Augmented Single-track Model

Lateral vehicle dynamics have been of interest for a long time and are accordingly well studied. A standard is the well-known single-track model (cf. Mitschke (1990)). In order to reflect the particular features of our vehicle, the standard model equations had to be modified:

- The front and rear steerable axles were accounted for by a second input.
- Due to the perfectly symmetric design of the vehicle, the center of mass lies exactly halfway between the axles, a feature which allows some simplifications.
- Because the lateral positions of the axle centers are the variables to be controlled, two extra states had to be introduced which are also the system outputs.

As the resulting linear model is only valid for small angles, the lateral dynamics is formulated in the shape of deviations from the global course of the *ideal track* (Fig. 5). The course of the *ideal track* as it was planned by its architect is characterized by a slowly varying orientation and curvature κ. The curvature is defined as the change of orientation with respect to the arc-length $\kappa = |\,\mathrm{d}\varphi/\mathrm{d}s|$ and its rate of change is assumed to be slow with respect to the system dynamics ($\dot{\kappa} \approx 0$). The quasi-static nature of the vehicle traveling along the *ideal track* makes it possible to treat the dynamic equations of the deviations (relative dynamics (2) with (3)) independently of the *ideal track* dynamics. For the sake of brevity, the common Δ for denoting relative values is omitted in this paper. Instead, the actual values, as measured on board the vehicle, are indicated by a star ($\star$).

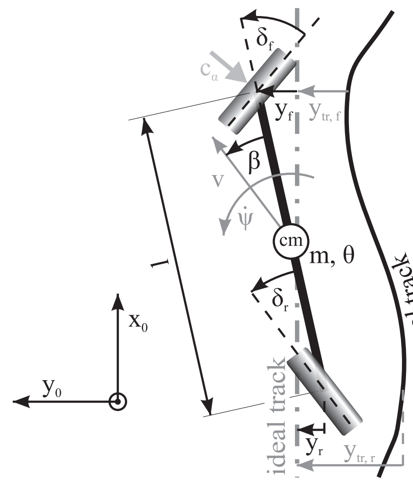

Fig. 5. Variables of the single-track model

$$\begin{aligned} \underline{\dot{x}} &= \underline{A} \cdot \underline{x} + \underline{B} \cdot \underline{u} \\ \underline{y} &= \underline{C} \cdot \underline{x} + \underline{D} \cdot \underline{u} \end{aligned} \tag{2}$$

$$\underline{x} = \left[\, y_f \;\, y_r \;\, \dot{\psi} \;\, \beta \,\right]^T, \; \underline{u} = \left[\, \delta_f \;\, \delta_r \,\right]^{\mathrm{T}}$$

$$\underline{A} = \begin{bmatrix} \frac{v}{l} & -\frac{v}{l} & \frac{l}{2} & v \\ \frac{v}{l} & -\frac{v}{l} & -\frac{l}{2} & v \\ 0 & 0 & -\frac{l^2\, c_\alpha}{2\, v\, \theta} & 0 \\ 0 & 0 & -1 & -\frac{2\, c_\alpha}{m\, v} \end{bmatrix}, \; \underline{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{l\, c_\alpha}{2\, \theta} & -\frac{l\, c_\alpha}{2\, \theta} \\ \frac{c_\alpha}{m\, v} & \frac{c_\alpha}{m\, v} \end{bmatrix} \tag{3}$$

$$\underline{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \; \underline{D} = \underline{0}$$

The state vector is made up of the lateral distance between the front and rear axle centers and the *ideal track* y_f resp. y_r, the relative yaw rate $\dot{\psi} = \dot{\psi}^\star - v\,\kappa$, and the relative side slip angle $\beta = \beta^\star$; the control inputs are the relative steering angles $\delta_f = \delta_f^\star - \kappa\, l/2 - \kappa\, m\, v^2/(2\, c_\alpha)$, $\delta_r = \delta_r^\star + \kappa\, l/2 - \kappa\, m\, v^2/(2\, c_\alpha)$ of the front and rear axles, respectively (cf. Fig. 5). The matrices depend on the physical parameters of the vehicle: axle base l, longitudinal velocity v, combined mass of chassis and body m, moment of inertia for yawing motions θ, and cornering stiffness c_α.

While the curvature of the *ideal track* is assumed to be a priori known, local deviations have to be gauged or estimated. We use a special estimation technique that constantly adapts estimates of the track at discrete sampling points. This algorithm will not be discussed here, instead we take the "map of local deviations" as given.

4.2 Derivation of States and Inputs from Flat Outputs

As in our example system the inputs act only on the last two lines of the state equation (3) while the outputs are the first two states, inversion is effected by a straightforward rearrangement and (sub)matrix-divisions. Formulation of the system states and inputs in terms of the flat outputs is given in (4):

$$\dot{\psi} = \frac{\dot{y}_f}{l} - \frac{\dot{y}_r}{l}, \; \beta = \frac{y_r}{l} - \frac{y_f}{l} + \frac{\dot{y}_r}{2\,v} + \frac{\dot{y}_f}{2\,v} \tag{4}$$
$$\delta_f = \frac{y_r - y_f}{l} + \frac{\dot{y}_f}{v} + \frac{\ddot{y}_r\,\left(\frac{m}{4} - \frac{\theta}{l^2}\right) + \ddot{y}_f\,\left(\frac{m}{4} + \frac{\theta}{l^2}\right)}{c_\alpha}$$
$$\delta_r = \frac{y_r - y_f}{l} + \frac{\dot{y}_r}{v} + \frac{\ddot{y}_r\,\left(\frac{m}{4} + \frac{\theta}{l^2}\right) + \ddot{y}_f\,\left(\frac{m}{4} - \frac{\theta}{l^2}\right)}{c_\alpha}$$
$$a_{lat,f} = \ddot{y}_f, \; a_{lat,r} = \ddot{y}_r$$

Here a_{lat} is the relative lateral acceleration that is used in the objective function f_2.

5. MULTIOBJECTIVE OPTIMIZATION OF THE TRAJECTORY

The optimal reference trajectories y for the front and rear axles are parametrized by n_h spacial discretization points $\underline{y} = y(s_i = i\Delta s)$; $i = 0 \dots n_h - 1$ with a trajectory length of $s_h = (n_h - 1)\Delta s$. This has the advantage that track data, such as the deviation from the ideal track and the clearance, can be stored in the same grid. It allows direct evaluation at the discretization points with no need for interpolations (as used e.g. in (5) and (9)). Also, since the spacial grid is velocity-independent, the trajectories can still be used when the vehicle accelerates. The derivatives of the trajectory that are needed for the inverse model are approximated only at the discretization points by central differences: $\dot{y}_i \approx 0.5v(y_{i+1} - y_{i-1})/\Delta s$ and $\ddot{y}_i$ and $y_i^{(3)}$, accordingly.

5.1 The Objectives

In this work we consider three important objectives that every machine has to fulfill: safety, comfort, and efficiency. They are mathematically expressed as:

$$f_1 = \|\underline{y}_{tr,f} - \underline{y}_f\|_2 + \|\underline{y}_{tr,r} - \underline{y}_r\|_2 \tag{5}$$
$$f_2 = \|\underline{a}_{lat,f}\|_2 + \|\underline{a}_{lat,r}\|_2 \tag{6}$$
$$f_3 = \|\dot{\underline{\delta}}_f\|_1 + \|\dot{\underline{\delta}}_r\|_1 \tag{7}$$

All three objectives are computed for the prediction horizon s_h in the norms of vectors with length n_h. The objective function for safety (f_1) is defined as the 2-norm of the deviation of the vehicle from the track centerline ($y - y_{tr}$). On the centerline the chances of the flange striking the railhead are minimal in the case of unbiased disturbances. The 2-norm was chosen to reflect the probabilistic nature of the uncertainties. Passenger comfort is measured in terms of the forces, and hence the accelerations, that act on the body. Thus, the comfort objective (f_2) is expressed as the 2-norm of lateral accelerations a_{lat}. The 2-norm is a common choice for assessing comfort in the presence of stochastic excitations. Finally, efficiency of a hydraulic actuator is directly related to the velocity of displacement. Therefore, the average energy consumption of the actuator is represented by the 1-norm of the rate of change of the steering angle $\dot{\delta}$ in f_3.

To guarantee continuity of the first and second derivatives at the beginning of the trajectory, an initial cost is imposed on the first two discretization points (8) that rates the deviation from the preceding trajectory. Here, k is the currently computed trajectory, s_0 being its start position.

$$f_{ini} = \|y_{f,k}(s_0, s_1) - y_{f,k-1}(s_0, s_1)\|_2 + \|y_{r,k}(s_0, s_1) - y_{r,k-1}(s_0, s_1)\|_2 \tag{8}$$

The only hard constraint we consider in this work is the clearance r_{clr} that limits the maximum deviation of the trajectory points from the track centerline:

$$|y - y_{tr}| < r_{clr} \tag{9}$$

5.2 Calculation of the Pareto Set

In order to calculate the entire Pareto set of the three objectives (5)-(7), we use the software package GAIO [1] which contains set-oriented methods for, amongst others, (continuous) multiobjective optimization. The methods can be divided into two main classes: the *subdivision techniques* (see Dellnitz et al. (2005), Schütze (2003), Schütze et al. (2003)) and the *recovering techniques* (see Schütze (2004), Schütze et al. (2005), Dell'Aere (2006)). The subdivision techniques are of global nature and suitable for derivative-free optimization, but restricted to moderate dimensions. The recovering techniques are of local nature, but applicable in higher dimensions both in parameter space and in image space (typically $1 \leq k \leq 5$). Both types of algorithms come up with a fine covering of the (global) Pareto set in a comparatively short time.

As in case of our application the objective functions are convex and the problem rapidly becomes high-dimensional (we have twice as many optimization variables as discretization points on each trajectory, $\underline{x} = [\underline{y}_f, \underline{y}_r] \in \mathbb{R}^{2n_h}$ with $n_h \geq 25$), the so-called algorithm for 'Recovering in image space' (see Dell'Aere (2006)) fits best. This algorithm belongs to the above-mentioned recovering techniques and is a specific set-oriented continuation method that – based on an initial solution (e.g., given by the vector of function values for a single optimum of one of the objectives) around which a box is created – inserts step by step all those neighboring boxes that contain images of Pareto points. The insertion of neighboring boxes in image space is based on the idea of choosing proper targets $\underline{T} \in \mathbb{R}^k$ within these boxes and solving the distance minimization problem

$$\min_x ||\underline{F}(\underline{x}) - \underline{T}||.$$

The algorithm finally gives a covering of the entire Pareto set as on the one hand it has been shown that every Pareto set locally is part of a $k - 1$ -dimensional manifold (Hillermeier (2001)) and on the other hand is connected as our objectives are convex (Miettinen (1999)).

The determination of the entire Pareto set is computationally costly and therefore cannot be computed online. But the big advantage of a precomputation of the entire Pareto set to the problem is that good trajectories can be determined much easier than by testing the optimization of several weighted sums. Computing the weights for the objectives, one can design an online MPC that switches situationally between different weighted sums of the objectives.

[1] Global Analysis of Invariant Objects, http://math-www.uni-paderborn.de/˜agdellnitz

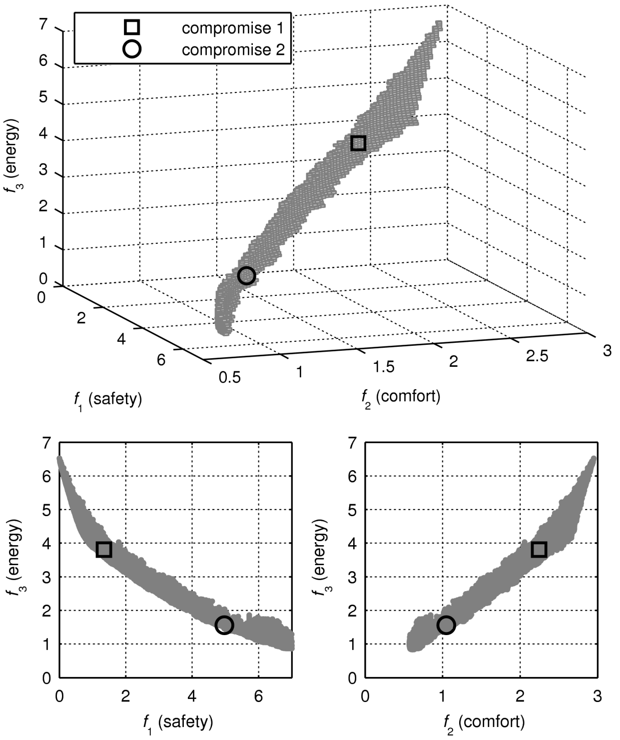

Fig. 6. Pareto front for the three objectives

5.3 Optimization Results

The following examples were computed with $n_h = 25$ and $\Delta s \approx 0.25$ m yielding an optimization horizon of $s_h = 6$ m. Fig. 6 shows the Pareto set in image space (f_1 vs. f_2 vs. f_3), the so-called Pareto front. The Pareto front is a narrow, ribbon-like surface that exhibits a nearly linear characteristics in the f_2-f_3-projection while the other two projections display the shape of a banana. Along this ribbon, an improvement of safety causes a deterioration in both comfort and energy which means that the objectives for comfort and energy have a similar impact on the solution. This is due to the fact that a smooth trajectory also requires less vigorous steering action. In analogy, the curves 2) and 3) are close together in preimage space (Fig. 7) where some trajectories for the front axle are plotted. The solid gray lines represent the clearance bounding all possible solutions. The trajectories 1) through 3) are the single optima for f_1 through f_3 respectively, while 4) and 5) depict two examples of Pareto-optimal compromises. As one can see in Fig. 6, the corresponding Pareto front images do lie in the middle section of the surface. They were chosen to be safe resp. comfortable/energy- efficient but not to the extreme. Choosing Pareto points too close to the extrema makes no sense because safety will only slightly improve while comfort and energy deteriorate significantly, and vice versa. Finding such Pareto-optimal compromises is very intuitive and at the same time mathematically funded if the entire Pareto front is available.

6. OUTLOOK

In this work we have discussed the way a modified MPC strategy can be used to design a self-optimizing guidance

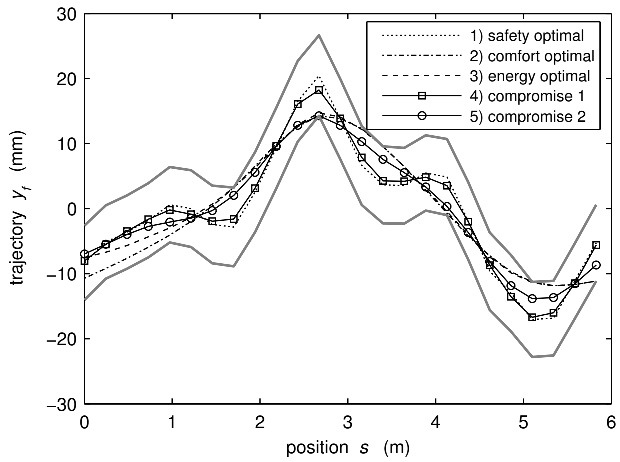

Fig. 7. Different reference trajectories for the front axle

system. The method proposed is a basic framework allowing a number of enhancements and improvements. Since central differences are not very accurate for approximating the derivatives, it is planned to parametrize the trajectories by splines. Also, further elaboration of the online optimization algorithm and of the decision-making process are necessary. Additional technical constraints may be considered, such as a limited steering angle or maximum possible slip angle. Finally, the control strategies shall be applied to the real test vehicle.

ACKNOWLEDGEMENTS

This work was mainly developed in the course of the Collaborative Research Center 614 – University of Paderborn, and was published on its behalf and funded by the Deutsche Forschungsgemeinschaft.

REFERENCES

S. Di Cairano, A. Bemporad, I. Kolmanosky, and D. Hrovat. Model predictive control of nonlinear mechatronic systems: An application to a magnetically actuated mass spring damper. In *2nd IFAC Conference on Analysis and Design of Hybrid Systems*, Alghero, Italy, 2006.

F. Chaplais and N. Petit. Inversion in indirect optimal control. In *Proc. of 7th European Control Conference*, Cambridge, UK, 2003.

A. Dell'Aere. Multi-objective optimization in self-optimizing systems. In *Proceedings of the IEEE 32nd Annual Conference on Industrial Electronics (IECON)*, pages 4755–4760, 2006.

M. Dellnitz, O. Schütze, and T. Hestermeyer. Covering Pareto sets by multilevel subdivision techniques. *Journal of Optimization Theory and Applications*, 124:113–155, 2005.

M. Ehrgott. *Multicriteria Optimization*, volume 491 of *Lecture Notes in Economics and Mathematical Systems*. Springer Verlag, Berlin, 2000.

C. Ettingshausen, T. Hestermeyer, and S. Otto. Aktive Spurführung und Lenkung von Schienenfahrzeugen. In *6. Magdeburger Maschinenbautage*, Magdeburg, 2003.

M. Fliess, J. Levine, P. Martin, and P. Rouchon. A Lie-Backlund approach to equivalence and flatness of nonlinear systems. In *IEEE Transactions on Automatic Control*, volume 44, pages 922–937, May 1999.

U. Frank, J. Gausemeier, H. Giese, F. Klein, O. Oberschellp, A. Schmidt, B. Schulz, H. Vöcking, and K. Witting. *Selbstoptimierende Systeme des Maschinenbaus - Definitionen und Konzepte*. HNI-Schriftenreihe 155, Bonifatius GmbH, Paderborn, 2004.

C. Hillermeier. *Nonlinear Multiobjective Optimization - A Generalized Homotopy Approach*. Birkhäuser, Berlin, 2001.

S. Kehl, W.-D. Pölsler, and M. Zeitz. Querregelung eines versuchsfahrzeugs entlang vorgegebener bahnen. *at - Automatisierungstechnik*, 55(6), 2007.

E. C. Kerrigan and J. M. Maciejowski. Designing model predictive controllers with prioritised constraints and objectives. In *Proceedings of IEEE International Symposium on Computer Aided Control System Design*, pages 33–38, Glasgow, 2002.

D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

K. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, 1999.

M. Milam, K. Mushambi, and R. Murray. A new computational approach to real-time trajectory generation for constrained mechanical systems. In *Proceedings of the 39th IEEE Conference on Decision and Control*, volume 1, pages 845–551, 2000.

M. Mitschke. *Dynamik der Kraftfahrzeuge Band C: Fahrverhalten*. Springer-Verlag, Berlin, 1990.

E. Münch, H. Vöcking, and T. Hestermeyer. Self-learning disturbance compensation for active suspension systems. In *International Conference on Informatics in Control, Automation and Robotics*, Barcelona, Spain, 2005.

E. Ronco, B. Srinivasan, J. Y. Favez, and D. Bonvin. Predictive Control with Added Feedback for Fast Nonlinear Systems. In *European Control Conference*, pages 3167–3172, 2001.

O. Schütze. *A new data structure for the nondominance problem in multi-objective optimization*, volume 2 of *Lecture Notes in Computer Science*. Springer, Berlin, 2003.

O. Schütze. *Set Oriented Methods for Global Optimization*. PhD Thesis, University of Paderborn, 2004.

O. Schütze, S. Mostaghim, M. Dellnitz, and J. Teich. Covering Pareto sets by multilevel evolutionary subdivision techniques. In C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, and L. Thiele, editors, *Evolutionary Multi-Criterion Optimization*, Lecture Notes in Computer Science, Berlin, 2003. Springer.

O. Schütze, A. Dell'Aere, and M. Dellnitz. On continuation methods for the numerical treatment of multi-objective optimization problems. In J. Branke, K. Deb, K. Miettinen, and R. E. Steuer, editors, *Practical Approaches to Multi-Objective Optimization*, number 04461 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2005. Internationales Begegnungs- und Forschungszentrum (IBFI).

A. Trächtler. Railcab - mit innovativer Mechatronik zum Schienenverkehrssystem der Zukunft. In *VDE Kongress 2006; Aachen*, 2006.

A. Trächtler, E. Münch, and H. Vöcking. Iterative Learning and Self-Optimization Techniques for the Innovative Railcab-System. In *IECON 2006 - 32nd Annual Conference of the IEEE Industrial Electronics Society*, pages 4683–4688, Paris, 2006.

S. Wright. Applying new optimization algorithms to model predictive control. In *Proceedings of the Fifth International Conference on Chemical Process Control*, volume 93, pages 147–155, Tahoe City, California, 1997.