

An Optimal Control Approach To Real-Time Vehicle Guidance

Martin Vögel¹, Oskar von Stryk¹, Roland Bulirsch¹, Cornelius Chucholowski², and Thieß-Magnus Wolter²

¹ Technische Universität München, Zentrum Mathematik/M2, D-80290 München, Germany, URL: <http://www-m2.ma.tum.de>

² Tesis DYNAware GmbH, Implersstraße 26, D-81371 München, Germany, URL: <http://www.thesis.de>

Abstract. A newly developed two-level driver model is presented. On the anticipation level, optimal control problems for a reduced vehicle dynamics model are solved repeatedly on a moving prediction horizon to yield near optimal setpoint trajectories for the full model. On the stabilization level, a nonlinear position controller is developed to accurately track the setpoint trajectories with a full motor vehicle dynamics model in real-time. The formulation of the optimal control problems on the anticipation level is based on a nonlinear single track model which is extended by a complex tire model and further nonlinear model details such as to match the main properties of the full vehicle dynamics model. The optimal control problems are solved efficiently by a recently developed sparse direct collocation method. Numerical results for various vehicle maneuvers are presented, including a time-optimal double lane change at high speed.

1 Introduction

Driving comfort and safety of modern passenger cars can highly be improved by electronic vehicle control units (ECUs) actively interfering in the vehicle dynamics. Popular examples include anti-lock braking systems (ABS) and electronic stability programs (ESP). However, through the use of ECUs the vehicle design and the dynamical driving properties become even more complex. Therefore, major car manufacturers as well as automotive suppliers make use of suitable software for virtual prototyping in order to cut the product development time and cost, and to improve the design quality.

Specifically, Hardware-in-the-Loop (HIL) experiments provide efficient, cost effective, reproducible, and save tests for ECUs. Here, a test bench is used to link the ECU to the numerical real-time simulation of the full motor vehicle dynamics (Fig. 1). Furthermore, to rate the handling and driveability properties of a virtual prototype, the performance of the closed loop of *driver, vehicle, and environment* must be investigated in real-time [27]. Thus, handling characteristics of the physical prototype such as body roll, ride quality including vibration and bumps, vehicle safety, and performance parameters can be predicted.

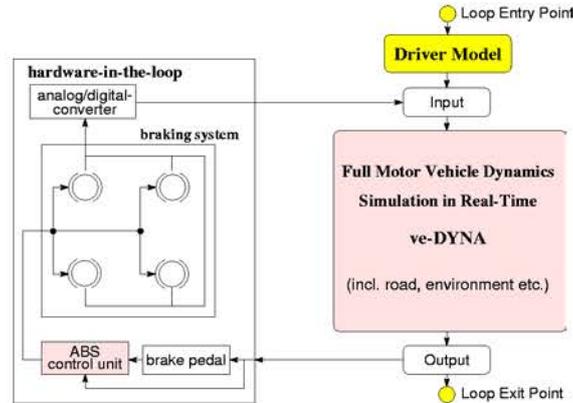


Fig. 1. Hardware-in-the-Loop setup for a braking system with ABS control unit linked to the real-time simulation of full motor vehicle dynamics.

To realistically simulate the full motor vehicle dynamics in real-time tailored models of the vehicle dynamics and the road properties are needed (Sect. 2). A new driver model consisting of two levels is outlined in Sect. 3. On the *anticipation level*, optimal control problems on a moving prediction horizon (Sect. 3.3) are solved for a newly developed extended single-track vehicle model (Sect. 3.2). Thus, optimal open-loop controls and trajectories are obtained, e. g., specifying the location and the velocity of the vehicle’s center of gravity along the road. The latter are used as setpoint trajectories for the nonlinear real-time position control which serves to compensate any disturbances on the *stabilization level* of the driver model (Sect. 3.5). For the efficient numerical solution of the optimal control problems on the anticipation level, a recently developed version of a sparse direct collocation method has been employed (Sect. 3.4).

The numerical results for various virtual test drives demonstrate the efficiency of the approach (Sect. 4).

2 Modeling and Simulation of Full Vehicle Dynamics

To realistically “test drive” entire vehicles in the computer, computer models of the whole car, including the suspensions, the powertrain, the engine, the steering mechanism, and ECUs are required along with models for the road geometries and conditions.

2.1 Numerical Simulation of Full Motor Vehicle Dynamics in Real-Time with *veDYNA*

A detailed and comprehensive vehicle model is needed to allow for the nonlinear kinematics of wheel and axle, and to describe the drive train, the steering

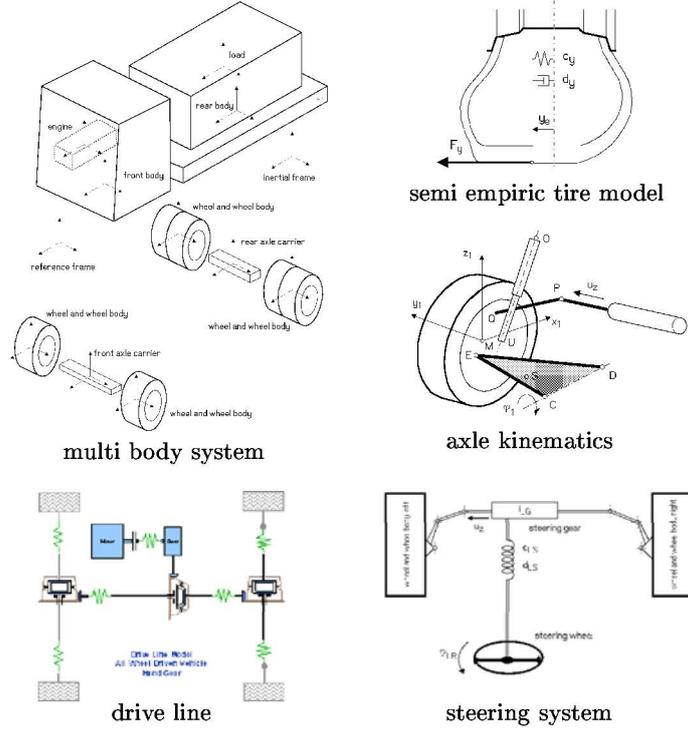


Fig. 2. Submodels of the full motor vehicle dynamics model of *veDYNA*.

mechanism and the tire dynamics. Our vehicle model consists of a suitable multibody system with kinematical connections and force elements which is supplemented by a sophisticated tire model. General purpose methods for modeling multibody systems use the descriptor form of the equations of motion leading to a large-scale system of differential-algebraic equations (DAEs) of index 3 [20]. For DAE systems particular numerical techniques must be applied to prevent a “drift-off” from the algebraic constraints [21]. Here, we make use of an optimally tailored model description which yields a system of ordinary differential equations (ODEs) which is suitable for real-time simulation.

The vehicle model of *veDYNA* consists of a system of nine rigid bodies comprising the vehicle body, the axle suspensions and the wheels. Further submodels are employed to depict the characteristics of the drive train, the steering mechanism, and the tires (Fig. 2). Suitable minimum coordinates and generalized velocities are used to describe the spatial state of the vehicle and its components [19]. The equations of motion are derived from Jourdain’s Principle yielding

$$M_{BV}(y_{BV}) \dot{z}_{BV} = Q_{BV}(y_{BV}, z_{BV}, y_{ST}, z_{ST}, y_{DT}, z_{DT}) \quad (1)$$

$$\dot{y}_{BV} = K_{BV}^{-1}(y_{BV}) z_{BV} \quad (2)$$

$$M_{DT} \dot{z}_{DT} = Q_{DT}(y_{DT}, z_{DT}) \quad (3)$$

$$\dot{y}_{DT} = V_{DT} z_{DT} \quad (4)$$

$$M_{ST}(y_{ST}, y_{BV}) \dot{z}_{ST} = Q_{ST}(y_{ST}, z_{ST}) \quad (5)$$

$$\dot{y}_{ST} = V_{ST} z_{ST} \quad (6)$$

$$D \dot{y}_T = F_{stat} - C y_T. \quad (7)$$

Thus, the vehicle dynamics is fully characterized by the system of 24 first-order ODEs comprising the vehicle body and the axles, (1) and (2). Eight ODEs (7) describe the lateral and longitudinal deviations of the tires by means of spring and damper elements. The vertical deformations of the tires are covered by (1). The dynamic model of the drive train consists of 19 ODEs, (3) and (4), including four equations governing the angular wheel speeds. Five additional ODEs account for the dynamics of the steering system (5) and (6). Couplings between the separate systems occur via the generalized forces and torques Q_{BV} . Wind forces and moments result in additional forces applied to the multibody system of the vehicle [17,19].

The tire forces have a significant impact on the dynamical behavior of a vehicle. The semi-empirical tire model that is used here describes the behavior of a real tire accurately [11,19]. About 80 parameters which can be measured or estimated enter the model for each tire in *veDYNA*. The model covers different driving situations, including effects at the driving limits such as sliding and spinning. The actual tire model is selected online depending on the respective road and weather conditions [1].

Due to the stiffness of the system (1)–(7) its numerical integration is carried out recursively with a semi-implicit one-step Euler scheme using a constant step size [19]. In particular, the integration method makes efficient use of the special block structure of the ODEs which yields

$$z_{ST}^{k+1} = z_{ST}^k + h \left(M_{ST} - h \frac{\partial Q_{ST}}{\partial z_{ST}} - h^2 \frac{\partial Q_{ST}}{\partial y_{ST}} V_{ST} \right)^{-1} Q_{ST}^k \quad (8)$$

$$y_{ST}^{k+1} = y_{ST}^k + h V_{ST} z_{ST}^{k+1} \quad (9)$$

$$z_{DT}^{k+1} = z_{DT}^k + h \left(M_{DT} - h \frac{\partial Q_{DT}}{\partial z_{DT}} - h^2 \frac{\partial Q_{DT}}{\partial y_{DT}} V_{DT} \right)^{-1} Q_{DT}^k \quad (10)$$

$$y_{DT}^{k+1} = y_{DT}^k + h V_{DT} z_{DT}^{k+1} \quad (11)$$

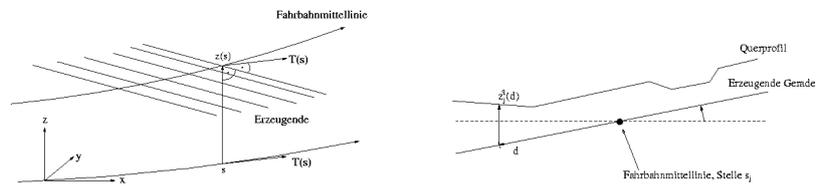
$$z_{BV}^{k+1} = z_{BV}^k + h \left(M_{BV} - h \frac{\partial Q_{BV}}{\partial z_{BV}} - h^2 \frac{\partial Q_{BV}}{\partial y_{BV}} K_{BV}^{-1} \right)^{-1} Q_{BV}^k \quad (12)$$

$$y_{BV}^{k+1} = y_{BV}^k + h K_{BV}^{-1} z_{BV}^{k+1}. \quad (13)$$

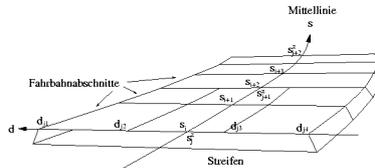
It turns out that a fast and stable solution is possible in real-time on modern PC hardware.

The development of intricate vehicle control devices requires the performance of the virtual car to match the actual vehicle behavior accurately. For calibrating the vehicle model of *veDYNA*, a parameter estimation tool has been developed which relies on observations obtained from physical test drives. The associated nonlinear least-squares problems are solved efficiently by means of mathematical optimization algorithms. Significant speed-ups in the computational time are achieved when employing a low-cost parallel computing platform, such as a heterogeneous cluster of PCs, which is well suited for the needs of the automotive industries and suppliers applying vehicle dynamics simulations [4].

2.2 Parameterized Road Model



(a) The curvature of the road's center line is a piecewise linear function of the arc length. (b) The variable transverse profile enables the modeling of bumps, dips, treads etc.



(c) Road sections with different surface properties (traction values, roughness).

Fig. 3. Components of the parameterized road model *veDYNA* Advanced Road.

To exercise the dynamic vehicle models under various road conditions (test tracks, public streets) in the computer, three-dimensional road geometries such as banked curves, hills, and bumpy roads, as well as variations in the road parameters due to weather and road surface conditions must be considered. These parameters vary widely on real courses, but can be controlled easily for virtual test drives if suitable models are employed.

For this purpose, we developed the tailored and parameterized road model *veDYNA* Advanced Road (Fig. 3) for the application in real-time vehicle dynamics simulations [22]. The course of the road is defined by means of the center line which is a curve in \mathbb{R}^3 . Its vertical projection onto the horizontal plane is defined by a curve in \mathbb{R}^2 whose curvature is a piecewise linear function

of the arc length as it is usual in road construction. The vertical coordinate of the center line can be selected arbitrarily. A variable transverse profile can be added at each point of the center stripe for generating three-dimensional road properties like bumps and dips. Road surface properties such as traction coefficients and the grade of roughness or smoothness are further parameters of the transverse profile, thus, permitting a description of different road and weather conditions.

The road model is flexible and easy-to-use. A realistic road model can be obtained with quite a few parameters, though it is possible to increase the level of detail arbitrarily. Moreover, the road model is well suited for real-time simulation. We refer to [22] for details. The road model has been incorporated into a graphical user interface of the MATLAB/Simulink emulation of *veDYNA* and into the 3D-visualization socket for computed test drives. Thus, the comfortable, modular modeling and visualization of any test course of interest is made possible.

3 Optimal Control Approach to Driver's Anticipation and Response

To enable virtual test drives the control actions of the vehicle's driver, such as steering, braking, accelerating, gear shifting, and operating the clutch, must be simulated by a virtual driver. In this context, we do *not* want to investigate the specific biomechanical, neuromuscular or psychological behavior of a *human* driver. Rather, a *mathematical* driver is needed for investigating the objective handling properties of the virtual prototype vehicle. Therefore, a virtual driver can use information which a human driver usually does not have (and cannot use directly), e. g., the traction values at the tires or the exact side slip angle. However, we require that the driver model is able to guide the virtual car along a test track at the dynamical driving limits in a way which is close to the performance of experienced human test drivers. An optimal control approach is motivated by "the basic assumption ... that the well-motivated, well-trained human operator behaves in a near optimal manner subject to his inherent limitations and constraints, and his control task" [12].

The two-level model of Donges [6] has been a fundamental contribution to modeling the driver's response. The control tasks in vehicle guidance are separated into tasks on the guidance and the stabilization level. Nowadays, ECUs are typically considered to be control systems on the stabilization level to compensate for disturbances and enabling the driver to manage the vehicle's dynamic behavior even in critical situations [7]. By this definition, control systems on the guidance level are systems that assist the driver in his/her steering task by looking ahead of the vehicle (anticipation).

Table 1. Variables of the single-track vehicle model

β	sideslip angle [rad]
ψ	yaw angle [rad]
ω_z	yaw angular velocity [rad/s]
v	velocity of the center of gravity [m/s]
X, Y	position of center of gravity [m]
δ_v	steering angle (at tire) [rad]
α_v, α_h	sideslip angle at front and rear wheel [rad]
S_v, S_h	front and rear lateral forces [N]
H	front and rear longitudinal forces [N]
T	drag force (wind) [N]
l_v, l_h	horizontal distance of front/rear axle to center of gravity [m]
m	mass of the vehicle [kg]
θ	inertia due to z-axis [kg m ²]

assumption that the center of gravity of the vehicle has zero height above a planar road. Thus, the left and right tire loads are equal even in case of large lateral accelerations; the two tires at the front and the rear axle are treated as one single “wide” tire. Moreover, under the given assumptions the tire loads are also constant during acceleration and braking maneuvers, and changes in the roll or pitch angle cannot occur. Therefore, it is not necessary to model the axle kinematics, e. g. by spring and damper elements, and the single-track model behaves as if the tires were rigidly linked to the vehicle body.

The single-track vehicle dynamics depends on the acceleration, the braking and the lateral forces acting on the fictitious front and rear wheels. The acceleration and braking forces V and H at the front and rear wheels vitally determine the vehicle’s velocity v ; the steering angle δ_v is used to guide the vehicle along any curve. Only six vehicle state variables (cf. Table 1) are contained in the equations of motion (14) – (19).

$$\dot{\beta}(t) = \omega_z(t) - \frac{1}{m v(t)} \left[\left(H(t) - T(v(t)) \right) \sin(\beta(t)) + S_v(\alpha_v(t)) \cos(\delta_v(t) + \beta(t)) + S_h(\alpha_h(t)) \cos(\beta(t)) \right] \quad (14)$$

$$\dot{\psi}(t) = \omega_z(t) \quad (15)$$

$$\dot{\omega}_z(t) = \frac{1}{\theta} \left[S_v(\alpha_v(t)) l_v \cos(\delta_v(t)) - S_h(\alpha_h(t)) l_h \right] \quad (16)$$

$$\dot{v}(t) = \frac{1}{m} \left[H(t) \cos(\beta(t)) - T(v(t)) - S_v(\alpha_v(t)) \sin(\delta_v(t) + \beta(t)) + S_h(\alpha_h(t)) \sin(\beta(t)) \right] \quad (17)$$

$$\dot{X}(t) = v(t) \cos(\psi(t) - \beta(t)) \quad (18)$$

$$\dot{Y}(t) = v(t) \sin(\psi(t) - \beta(t)) . \quad (19)$$

Approximations for the side slip angles at the front and the rear wheel can be computed from

$$\alpha_v = \beta - \omega_z \frac{l_v}{v} + \delta_v \quad (20)$$

$$\alpha_h = \beta + \omega_z \frac{l_v}{v}, \quad (21)$$

provided that the yaw angular velocity ω_z is “small”. Although the lateral forces S_v and S_h mainly depend on the respective side slip angle, also the longitudinal force H has to be taken into account. In particular, the vector $(S_v, S_h, H)^T = F(\alpha_v, \alpha_h, H^{des})$ is a highly nonlinear function of the side slip angles and a desired longitudinal force H^{des} . The latter can be interpreted as accelerating or braking pedal position. The function F serves to account for the interdependencies between lateral and longitudinal forces. Under standard driving conditions (i. e., $\alpha_v, \alpha_h, H^{des}$ are small enough) S_v and S_h are almost independent of H^{des} , and H is essentially given by H^{des} . However, if the overall slip – which is calculated internally for each tire when evaluating F – becomes larger, the direction and the magnitude of the overall tire force determines the magnitude of the lateral and the longitudinal forces. The function F also switches between accelerating and braking, and considers different drive train configurations, such as front-, rear- and all-wheel drive.

When combining the single-track model with a suitable tire model, the main dynamical vehicle properties, including skidding effects, can be described properly [14,15,27]. As the most important enhancement we linked the nonlinear single track model with the sophisticated tire model used within the realistic full vehicle dynamics simulation package *veDYNA* (Sect. 2.1). This tire model, which is denoted by F here, covers the nonlinear interdependencies between the lateral and the longitudinal forces. Both *veDYNA* and our extended single-track model use the same subprograms and input data to evaluate the current road contact and to compute the respective tire forces. Thus, the forces H , S_v , and S_h in the single-track vehicle dynamics are computed by means of the tire model of the full motor vehicle dynamics program. Each tire model depends on about 40 parameters, i. e., half of the parameters of the full *veDYNA* tire model. Moreover, the large extension of the nonlinear lateral and longitudinal dynamics and the consideration of wind forces make the extended single-track model suitable for computing optimal setpoint trajectories which are consistent with the full vehicle dynamics simulation. The computational results for the extended single-track model are in good agreement with the full vehicle dynamics simulations, although only six ODEs are needed instead of 56 (cf. Sect. 4).

3.3 Optimal Control Problem on a Moving Horizon

The minimum time control problem for the extended nonlinear single-track model is stated as to

$$\text{minimize } J[\mathbf{u}] = t_f + \rho_0 x_8(t_f) \quad (22)$$

where the $n_x = 8$ state variables and the $n_u = 2$ control variables are given by

$$\mathbf{x} = (\beta, \psi, \omega_z, v, X, Y, \delta_v, x_8)^T, \quad \mathbf{u} = (H^{des}, u_2)^T. \quad (23)$$

The optimal control problem is subject to the equations of motion given by the Eqs. (14) – (19) and the additional equations

$$\dot{x}_7 = \dot{\delta}_v = u_2, \quad \dot{x}_8 = \rho_1 u_1^2 + \rho_2 u_2^2. \quad (24)$$

Here, $\rho_0, \rho_1, \rho_2 \geq 0$ denote suitable non-negative weights. Even if only a small weight ρ_0 for $x_8(t_f) = \int_{t_0}^{t_f} (\rho_1 u_1^2 + \rho_2 u_2^2) dt$ is considered in the objective (22), the Hamiltonian becomes regular and the solution is more smooth.

The maximum vehicle speed is constrained by

$$\dot{x}_4 = v \leq v_{\max}. \quad (25)$$

The non-negativity of two nonlinear state constraints

$$0 \leq g_i(\mathbf{x}(t)), \quad i = 1, 2, \quad (26)$$

ensures that the left and right road limits are obeyed during the maneuver.

By the Principle of Optimality, the optimal control problem over a long course may be decoupled into several problems over smaller sections without loss of optimality, provided that the optimal trajectory \mathbf{x}^* is known for some intermediate values. From the basic dynamical vehicle properties and the geometry of the course, intermediate values $x_{0,i}^*, x_{f,j}^*$ of the optimal trajectory can often be estimated quite well, e. g., by considering the maximum vehicle speed and the maximum lateral acceleration in a curve. Then, the optimal control problem on a prediction horizon $[t_0, t_f]$ with the unknown duration $t_f - t_0$ consists in optimally steering the vehicle from a given initial state to a given final state, i. e.,

$$x_i(t_0) = x_{0,i}, \quad x_j(t_f) = x_{f,j}, \quad i \in I_i, \quad j \in I_j, \quad I_i, I_j \subset \{1, \dots, 8\}. \quad (27)$$

The optimal trajectory $\mathbf{x}^*(t)$, $\mathbf{u}^*(t)$, $t_0 \leq t \leq t_f$, must satisfy various necessary conditions derived from the Euler-Lagrange differential equations (EL-DEQs) and the Maximum Principle, e. g., [16,24]. Some of them can be used for an a-posteriori verification of the consistency of the solution computed by the method described in Sect. 3.4. Let $\lambda : [t_0, t_f] \rightarrow \mathbb{R}^{n_x}$ denote the adjoint or costate variable, η the multiplier function of the state constraints, and $\mathcal{H}(\mathbf{x}, \mathbf{u}, \lambda, \eta) = \sum_{i=1}^{n_x} \lambda_i f_i(\mathbf{x}, \mathbf{u}) + \sum_j \eta_j g_j(\mathbf{x}, \mathbf{u})$ the Hamiltonian. Then:

- The Hamiltonian \mathcal{H} of an autonomous problem must be a (piecewise) constant function of time for the optimal trajectory.
- The final value of $x_8(t_f)$ is free. Thus, the corresponding adjoint variable λ_8 must satisfy

$$\lambda_8^*(t_f) = \frac{\partial J}{\partial x_8(t_f)} = \rho_0 \quad (28)$$

for the Mayer type objective (22).

- The Hamiltonian is a nonlinear function of the control $u_2 = \delta_v$. Thus,

$$\frac{\partial \mathcal{H}}{\partial u_2} = \lambda_7 + \lambda_8 2\rho_2 u_2 \equiv 0 \quad \iff \quad u_2 = -\frac{\lambda_7}{2\rho_2 \lambda_8} \quad (29)$$

must hold for all parts of the optimal trajectory lacking active constraints that affect u_2 .

3.4 Sparse Direct Collocation for Numerical Optimal Control

Consider the general optimal control problem on the horizon $[t_0, t_f]$

$$J[\mathbf{u}] = \varphi(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} L(\mathbf{x}(t), \mathbf{u}(t), t) dt \longrightarrow \min! \quad (30)$$

$$\text{subject to } \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t), \quad (31)$$

$$0 \leq g_i(\mathbf{x}(t), \mathbf{u}(t), t), \quad i = 1, \dots, n_g, \quad (32)$$

$$0 = r_j(\mathbf{x}(t_0), t_0, \mathbf{x}(t_f), t_f), \quad j = 1, \dots, n_r, \quad (33)$$

where $\mathbf{x} : [t_0, t_f] \rightarrow \mathbb{R}^{n_x}$ and $\mathbf{u} : [t_0, t_f] \rightarrow \mathbb{R}^{n_u}$ denote the state and the control variables respectively.

Direct shooting and direct collocation methods both promise high flexibility and robustness when solving optimal control problems numerically with low or moderate accuracies [3,26]. However, in many practical applications the problem functions only have low, local differentiability properties, i. e., discontinuities in the first or second derivatives. Thus, obtaining a useful gradient approximation for shooting-type discretizations takes much more effort, since a numerical sensitivity analysis for initial value problems with switching points must be carried out. On the other hand, for a collocation-type discretization a careful, but much cheaper finite difference approximation may be sufficient, where no special treatment of discontinuities in first or second derivatives by switching functions is required. A further advantage of the direct collocation approach is the potentially faster computation as compared to direct shooting, because the ODE simulation (31) and the control optimization problems (30), (32) are solved simultaneously for collocation, but not iteratively as by shooting methods. To achieve an optimal speed-up for collocation, the NLP sparsity must fully be utilized. Otherwise the large size of the NLP will severely limit the efficiency.

In the sequel, we make use of a discretization for \mathbf{x} by piecewise cubic Hermite polynomials $\tilde{\mathbf{x}}(t) = \sum_k \alpha_k \hat{\mathbf{x}}_k(t)$ and a discretization for \mathbf{u} by piecewise linear functions $\tilde{\mathbf{u}}(t) = \sum_k \beta_k \hat{\mathbf{u}}_k(t)$ on a discretization grid $t_0 = t_1 < t_2 < \dots < t_{n_t} = t_f$. Adherence to the equations of motion is prescribed at the grid points and their respective midpoints (collocation at Lobatto points), thus yielding a set of nonlinear NLP equality constraints $\mathbf{a}(\mathbf{y}) = 0$. Moreover, any inequality constraints on the control or state variables are to be satisfied at the grid points which results in set of nonlinear NLP inequality constraints $\mathbf{b}(\mathbf{y}) \geq 0$. In both cases, $\mathbf{y} = (\alpha_1, \alpha_2, \dots, \beta_1, \beta_2, \dots, t_f)^T$ denotes the n_y parameters of the parameterization. The resulting nonlinearly constrained optimization problem basically reads as

$$\text{NLP: } \min_{\mathbf{y}} \Phi(\mathbf{y}) \quad \text{subject to } \mathbf{a}(\mathbf{y}) = 0, \mathbf{b}(\mathbf{y}) \geq 0, \quad (34)$$

where $\Phi : \mathbb{R}^{n_y} \rightarrow \mathbb{R}$ denotes the parameterized cost index (30) with $y \in \mathbb{R}^{n_y}$, and $\mathbf{a} : \mathbb{R}^{n_y} \rightarrow \mathbb{R}^{n_a}$, $\mathbf{b} : \mathbb{R}^{n_y} \rightarrow \mathbb{R}^{n_b}$ are the nonlinear NLP constraints.

The selected discretizations $\tilde{\mathbf{u}}$, $\tilde{\mathbf{x}}$ must satisfy certain convergence properties. One requirement is that the discretized solution must approximate a solution of the EL-DEQs and the Maximum Principle if the grid becomes arbitrarily fine, i. e., for $n_t \rightarrow \infty$ and $\max\{t_{i+1} - t_i : i = 1, \dots, n_t - 1\} \rightarrow 0$ [24]. A great advantage of the direct collocation approach is that it provides reliable estimates $\tilde{\lambda}$ of the adjoint variable trajectory along the discretization grid. These estimates are derived from the Lagrange multipliers of the NLP [24]. They enable a verification of the optimality conditions for the discretized solution although the EL-DEQs have not been solved explicitly.

Moreover, local optimality error estimates can be derived which enable efficient strategies for successively refining a first solution on a coarse grid [24,25]. Thus, a sequence of related NLPs must be solved whose dimensions increase with the number of grid points. The numerical solution of the NLPs can be done efficiently by sequential quadratic programming (SQP) methods. In each SQP iteration the current guess of the solution \mathbf{y}^* is improved by solving a quadratic subproblem, derived from a quadratic approximation of the Lagrangian of the NLP, subject to the linearized constraints [2,9]. The NLPs which result from a direct collocation discretization have several specific properties:

- The NLPs are large-scale with very many variables and very many constraints.
- Most of the NLP constraints are active at the solution, e. g., the equality constraints from collocation. Thus, the number n_s of free NLP variables is much smaller than the total number of variables n_y .
- The constraints' Jacobians $(\nabla \mathbf{a}(\mathbf{y}), \nabla \mathbf{b}(\mathbf{y}))$ are sparse and structured. Only a small number of the elements will be nonzero, and the percentage decreases as the number of grid points increases.

- The NLP objective $\Phi(\mathbf{y})$ only depends on a few, fixed number of variables regardless of the actual grid size, provided that the objective (30) is of Mayer type, i. e., $L \equiv 0$.

These features are fully utilized by means of the recently developed large-scale SQP method SNOPT [9] which partitions the NLP variables into basic, superbasic and nonbasic variables. The Hessian of the NLP Lagrangian function is approximated by limited-memory quasi-Newton updates, and a reduced Hessian algorithm is used for solving the QP subproblems. The null-space matrix of the working set in each iteration is obtained from a sparse LU factorization.

When compared to standard “dense” SQP methods, the computational speed-up which can be achieved by fully utilizing the NLP structure is more than a factor of one hundred for typical discretized optimal control problems (Sect. 4). The described sparse direct collocation method is implemented in the software DIRCOL [24,25], which is freely available from the authors and has already been distributed upon request to about 40 institutions from research and industry.

3.5 Nonlinear Position Control of Optimal Setpoint Trajectories

On the stabilization level of our driver model, a nonlinear position control of the vehicle’s center of gravity along the road is employed to guide the full vehicle dynamics model [14,22]. We use the computed optimal trajectory $(X^*(t), Y^*(t)) =: (w_x(t), w_y(t))$ of the center of gravity as a setpoint (“target”) for the actual position $(X(t), Y(t))$ of the vehicle. The setpoint trajectories $((w_x(t), w_y(t))$ are repeatedly computed by solving optimal control problems for the reduced vehicle model (cf. Sections 3.2, 3.3 and 3.4). The computed optimal controls \mathbf{u}^* are not directly used here.

The position control algorithm is based on the theory of nonlinear system decoupling and control originally developed for robot control [8]. Several state variables of the full vehicle dynamics model in *veDYNA* enter into the control law, among them the direction of the velocity of the vehicle’s center of gravity $\nu = \psi - \beta$, the side slip angle β , the actual position $(X(t), Y(t))$ and velocity $(\dot{X}(t), \dot{Y}(t))$, as well as the position of the target $((w_x(t), w_y(t))$, its velocity $((\dot{w}_x(t), \dot{w}_y(t))$ and its acceleration $((\ddot{w}_x(t), \ddot{w}_y(t))$. We compute a desired front lateral force S_v^{des} and a desired longitudinal force H^{des} by

$$\begin{aligned}
 S_v^{des}(t) &= -S_h(x(t)) - m \left[\begin{aligned} & \left(\cos(\nu(t)) \beta(t) + \sin(\nu(t)) \right) a_x^{des}(t) + \\ & \left(\sin(\nu(t)) \beta(t) - \cos(\nu(t)) \right) a_y^{des}(t) \end{aligned} \right] \quad (35) \\
 H^{des}(t) &= T(x(t)) + m \left[\cos(\nu(t)) a_x^{des}(t) + \sin(\nu(t)) a_y^{des}(t) \right],
 \end{aligned}$$

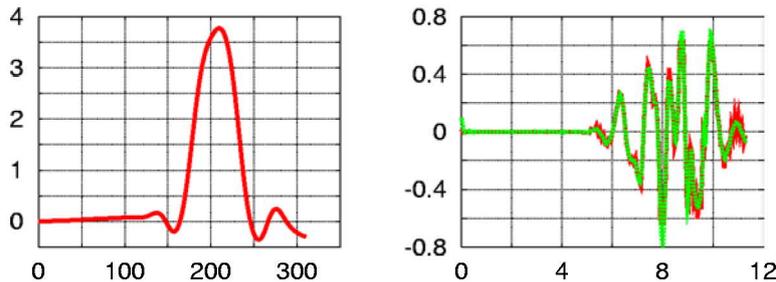


Fig. 5. Numerical solution of the minimum time double lane change maneuver with the extended nonlinear single track model for 161 grid points. Left: trajectory in the (X, Y) -plane. Right: comparison of the computed control $\tilde{u}_2 = \delta_v$ (solid, red line) and the function $-\tilde{\lambda}_7 / (2\rho_2 \tilde{\lambda}_8)$ (dashed, green line) over time (cf. Eq. (29)).

where the desired second derivatives $a^{des} = (a_x^{des}, a_y^{des})^T$ of the coordinates X, Y are given in inertial coordinates and calculated as follows

$$\begin{aligned} a_x^{des}(t) &:= \Lambda \left(w_x^{cor}(t) - X(t) \right) + 2\sqrt{\Lambda} \left(\dot{w}_x - \dot{X}(t) \right) + \ddot{w}_x \\ a_y^{des}(t) &:= \Lambda \left(w_y^{cor}(t) - Y(t) \right) + 2\sqrt{\Lambda} \left(\dot{w}_y - \dot{Y}(t) \right) + \ddot{w}_y. \end{aligned} \quad (36)$$

4 Numerical Results for Virtual Test Drives

4.1 Minimum Time Double Lane Change of a Passenger Car

As an example, we consider the ISO double lane change which shall be performed in minimum time with a standard passenger car, i. e., a BMW E30 with rear wheel drive.

First, on the anticipation level of our driver model, the optimal control problem for the corresponding single-track model (Sects. 3.2, 3.3) is solved by the method of Sect. 3.4. The control constraint during the lane change consists of a maximum speed of $v_{\max} = 33 \text{ [m/s]} \approx 119 \text{ [km/h]}$, and the weights are given by $\rho_0 = 5 \cdot 10^{-6}$, $\rho_1 = 10^{-7}$, $\rho_2 = 10^3$. The problem formulation also includes an optimal acceleration maneuver on a straight road before the actual double lane change is performed. There, the vehicle is accelerated from the initial velocity $v(t_0) = 8 \text{ [m/s]} \approx 29 \text{ [km/h]}$ to its maximum value v_{\max} within 5.1s which is kept until $t_f^* = 11.2958 \text{ s}$.

The numerical results for different improvements V1 – V4 of the sparse direct collocation method DIRCOL are reported in Table 2 (standard SQP: NPSOL-5.0 [10], sparse SQP: SNOPT-5.3-5 [9]). For reasons of demonstration, only equidistant grid points have been used here. Though, usually it is much more efficient to apply grid refinement based on suitable local error monitoring functions [24]. The speed-up factor for the improved collocation methods depends on the grid size; for the most advanced version V4

Table 2. Numerical results for the minimum time double lane change with the extended single-track model for the varying utilization of the NLP structure and sparsity

n_t	n_y	n_a	n_b	n_s	Jacbn		itn.	CPU [s] P III/500 Mhz min:sec	CPU speed-up V0/ V_i	memory [MB]	$J[u]$
21	200	160	42	—	9.3%	V0	32	0:26	1.0	1.4	11.28235
				—		V1	58	0:36	0.7	1.4	11.28233
				24		V2	36	0:05	5.2	1.7	11.28235
				24		V3	30	0:03	8.7	1.3	11.28235
				30		V4	24	0:01	26.0	0.5	11.28235
41	400	320	82	—	4.7%	V0	55	7:17	1.0	5.3	11.29100
				—		V1	86	9:57	0.7	5.3	11.29102
				51		V2	151	2:27	3.0	6.6	11.29142
				54		V3	44	0:20	21.9	4.8	11.29107
				54		V4	23	0:03	145.7	1.2	11.29147
81	800	640	162	—	2.4%	V0	75	73:30	1.0	21	11.29657
				—		V1	110	108:40	0.7	21	11.29658
				109		V2	74	9:27	7.8	25	11.29660
				110		V3	46	2:00	36.8	17	11.29670
				109		V4	65	0:29	152.1	2	11.29660
161	1600	1280	322	—	1.2%	V0	56	486:40	1.0	83	11.29852
				—		V1	39	329:10	1.5	83	11.29853
				217		V2	26	31:40	15.4	94	11.29852
				224		V3	18	7:58	61.1	64	11.29856
				219		V4	24	0:55	530.9	3	11.29852

- V0: standard finite differences + standard SQP
- V1: analyt. deriv. & fin. diffcs. + standard SQP
- V2: analyt. deriv. & fin. diffcs. + sparse SQP – no structure
- V3: analyt. deriv. & fin. diffcs. + sparse SQP + NLP-structure
- V4: analyt. deriv. & fin. diffcs. + sparse SQP + NLP-structure + sparse Jacobian

of DIRCOL, it is more than 500 when using 161 grid points (Table 2). The computed solution for the finest grid is depicted in Fig. 5. It should be noted, however, that for the online computation it is not necessary to solve the problem on the finest grid. A moderate accuracy of the solution is sufficient for which a computational time smaller than the time needed for the maneuver or its simulation can be achieved.

The optimality condition (29) for the computed trajectory is satisfied to reasonable accuracy (Fig. 5, right), although the adjoint differential equations have not been solved explicitly. Furthermore, condition (28) is also satisfied with high accuracy. The computed estimate $\lambda_g^*(t_f) = 4.999325 \cdot 10^{-6}$ agrees with the optimal value $\lambda_g^*(t_f) = \rho_0 = 5 \cdot 10^{-6}$ to four digits.

In the second step, the computed optimal trajectories for the single-track model serve as near optimal setpoint trajectories for the nonlinear real-

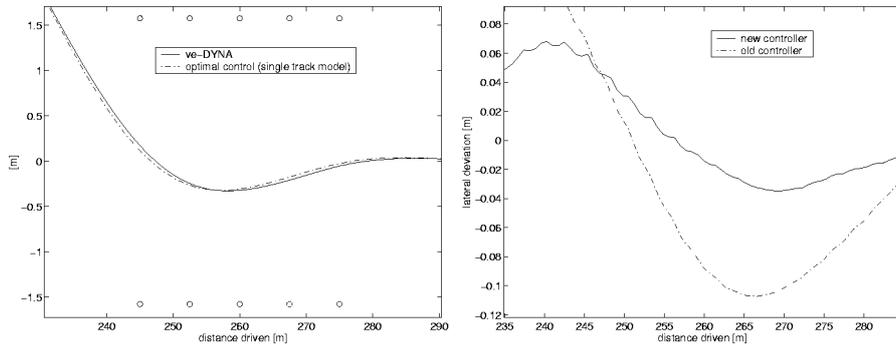


Fig. 6. Zoom into the state space trajectories for the last part of the double lane change maneuver with a BMW E30 at 119 km/h. Left: center of the front axle (—) and its near optimal setpoint trajectory (- - -). Right: deviation from the setpoint trajectory for the recent position controller (—) as compared to an earlier version (- - -).

time position control of the full motor vehicle dynamics model in *veDYNA* (Sect. 2.1). The difference between the state space trajectories (X^*, Y^*) and (X, Y) of the reduced and the full vehicle dynamics model is hardly visible. A zoom into the end of the double lane change (cf. Fig. 6) shows a maximum difference of about 7 cm. The small tracking errors for the double lane change maneuver at high speed is quite remarkable for the only rear wheel driven car, and demonstrates both, the capabilities of the position controller of the driver model, as well as the good prediction of the full vehicle dynamics behavior by the significantly smaller, extended single-track model.

4.2 Further Vehicle Maneuvers

Recently, Lim and Hedrick suggested a longitudinal and lateral vehicle controller for automated vehicle operation of passenger cars [13]. The position controller was applied to a challenging test maneuver with quick changes in road curvature and braking in turns (Fig. 7, left). High lateral accelerations of about 5 m/s occur. We have investigated the same maneuver with the full vehicle dynamics model of a BMW E30 and the position controller from Sect. 3.5 which was used to track the given setpoint trajectory. For our approach (Fig. 7, right), we observed only one tenth of the deviation from the setpoint trajectory that have been reported in [13].

Finally, a fast virtual test drive of the full-scale vehicle dynamics model of a BMW E30 has been performed along the BMW handling course at Aschheim which is about 2 km long (Fig. 8). Animated results for further vehicle maneuvers are also available from <http://www-m2.ma.tum.de/Projekte/kfz>.

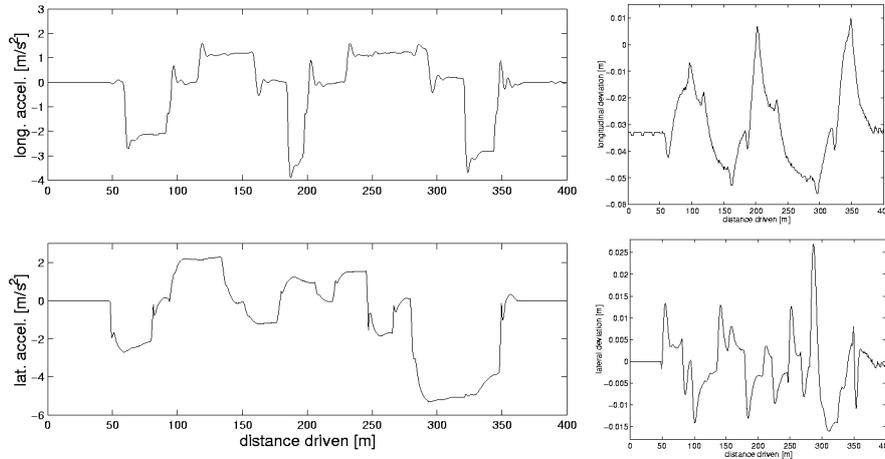


Fig. 7. Results for the driving cycle of [13]. Left: numerical simulation of the longitudinal and lateral acceleration of the vehicle's center of gravity with *veDYNA* and the position controller from Sect. 3.5. Right: longitudinal and lateral deviations of the setpoint trajectory.

Acknowledgement. The authors thank Dr.-Ing. Karl Naab of BMW AG for helpful discussions and support and Dipl.-Math. Torsten Butz for carefully proofreading the manuscript. This research has been supported by the Federal BMBF under grant 03-BU7TM2-8.

References

1. *veDYNA* User's Guide. TESIS DYNAware GmbH, München, Germany (1997)
2. Barclay, A., Gill, P.E., Rosen, J.B.: SQP methods and their application to numerical optimal control. In: W.H. Schmidt et al. (eds.): Variational Calculus, Optimal Control and Application, Int. Series of Numerical Mathematics **124**, Birkhäuser, Basel (1998) 207–222
3. Betts, J.T.: Survey of numerical methods for trajectory optimization. *AIAA J. Guidance, Control, and Dynamics* **21**, 2 (1998) 193–207
4. Butz, T., von Stryk, O., Wolter, T.-M.: A parallel optimization scheme for parameter estimation in motor vehicle dynamics. In: Proc. European Conference on Parallel Computing 2000, München, August 29-31 (to appear)
5. Chucholowski, C., Vögel, M., von Stryk, O., Wolter, T.-M.: Real time simulation and online control for virtual test drives of cars. In: H.-J. Bungartz, F. Durst, C. Zenger (eds.): High Performance Scientific and Engineering Computing, Lecture Notes in Computational Science and Engineering **8**, Springer-Verlag, Berlin (1999) 157–166
6. Donges, E.: Two level model of driver steering behavior. *Human Factors* **20**, 6 (1978) 691–707

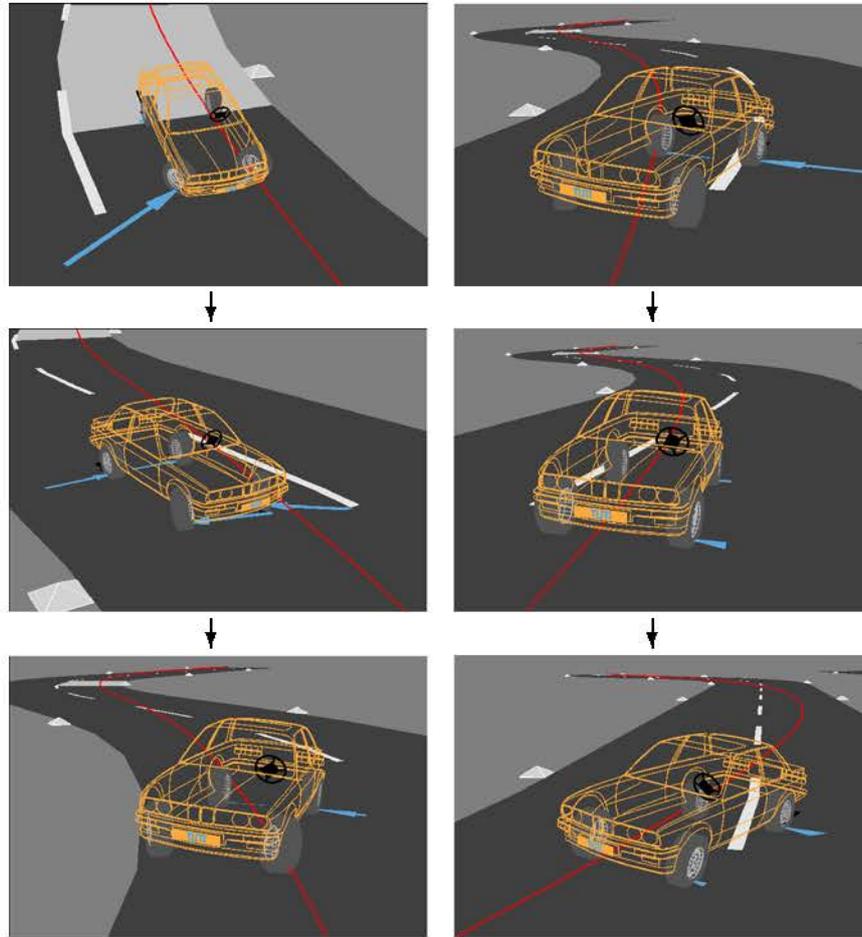


Fig. 8. Virtual test drive along the BMW test track at Aschheim including a short icy section of the road surface. The red line denotes the near optimal setpoint trajectory for the vehicle's center of gravity on a dry road.

7. Donges, E., Naab, K.: Regelsysteme zur Fahrzeugführung und -stabilisierung in der Automobiltechnik. *at – Automatisierungstechnik* 44, 5 (1996) 226–236
8. Freund, E.: Fast nonlinear control with arbitrary pole-placement for industrial robots and manipulators. *Intern. J. Rob. Res.* 1, 1 (1982) 65–78
9. Gill, P.E., Murray, W., Saunders, M.A.: SNOPT: An SQP algorithm for large-scale constrained optimization. Report NA 97-2, Department of Mathematics, University of California, San Diego (1997)
10. Gill, P.E., Murray, W., Saunders, M.A., Wright, M.H.: User's guide for NPSOL (Version 5.0): A Fortran package for nonlinear programming. Numerical Analysis Report 98-2, Department of Mathematics, University of California, San Diego, 1998.

11. Gipser, M.: DNS-Tire – ein dynamisches, räumliches, nichtlineares Reifenmodell. In: *Berechnung im Automobilbau*, VDI-Bericht Nr. 650, Düsseldorf: VDI-Verlag (1987) 115–135
12. Kleinman, D.L.; Baron, S.; Levison, W.H.: An optimal control model of human response, part I: theory and validation. *Automatica* **6,3** (1970) 357–369
13. Lim, E. H. M., Hedrick, J. K.: Lateral and longitudinal vehicle control coupling for automated vehicle operation. *Proc. American Control Conf.*, San Diego, CA (June 1999) 3676–3680
14. Mayr, R.: *Verfahren zur Bahnfolgeregelung für ein automatisch geführtes Fahrzeug*. Dissertation, Universität Dortmund (1991).
15. Mitschke, M.: *Dynamik der Kraftfahrzeuge*. Band C: Fahrverhalten. 3. Aufl., Springer-Verlag (1994)
16. Pesch, H.J.: A practical guide to the solution of real-life optimal control problems. *Control and Cybernetics* **23** (1994) 7–60
17. Popp, K., Schiehlen, W.: *Fahrzeugdynamik*. Stuttgart: Teubner (1993)
18. Rieckert, P.; Schunck, T.E.: Zur Fahrmechanik des gummibereiften Kraftfahrzeugs. *Ing.-Archiv*, Bd. XI (1940) 210–224
19. Rill, G.: *Simulation von Kraftfahrzeugen*. Braunschweig: Vieweg 1994
20. Schiehlen, W.: *Multibody Systems Handbook*. Springer-Verlag, Berlin (1990)
21. Simeon, B.: MBSPACK – Numerical integration software for constrained mechanical motion. *Surv. Math. Ind.* **5** (1995) 169–202
22. Vögel, M.: *Fahrbahnmodellierung und Kursregelung für ein echtzeitfähiges Fahrdynamikprogramm*. Diplomarbeit, Zentrum Mathematik, Technische Universität München (1997)
23. Vögel, M., von Stryk, O., Bulirsch, R., Chucholowski, C.: *Vehicle Dynamics Simulation*. In: H.-C. Hege, K. Polthier (eds.): *VideoMath Festival at ICM '98*, Berlin. A Collection of Mathematical Videos (Springer-Verlag, 1998)
24. von Stryk, O.: *Numerische Lösung optimaler Steuerungsprobleme: Diskretisierung, Parameteroptimierung und Berechnung der adjungierten Variablen*. *Fortschritt-Berichte VDI*, Reihe 8, Nr. 441, VDI-Verlag, Düsseldorf (1995)
25. von Stryk, O.: *User's Guide for DIRCOL Version 2.1: A direct collocation method for the numerical solution of optimal control problems*. Report, Lehrstuhl M2 Höhere Mathematik und Numerische Mathematik, Technische Universität München (November 1999)
26. von Stryk, O., Bulirsch, R.: Direct and indirect methods for trajectory optimization. *Annals of Operations Research* **37** (1992) 357–373
27. Zomotor, A.: *Fahrwerktechnik: Fahrverhalten*. 2. Aufl. Würzburg: Vogel (1991)