Optim Eng (2009) 10: 439–463
DOI 10.1007/s11081-008-9047-1

Generating locally optimal trajectories for an automatically driven car

Matthias Gerdts · Simon Karrenberg · Bernhard Müller-Beßler · Gregor Stock

Received: 19 February 2007 / Accepted: 14 April 2008 / Published online: 26 April 2008

Abstract The test-drive of an automobile along a given test-course can be modeled by formulating a suitable optimal control problem. However, if the length of the course is very long or if it has a very complicated structure, the numerical solution of the optimal control problem becomes very difficult. Therefore a moving horizon technique is employed, which splits the optimal control problem into a sequence of local optimal control problems that are combined by suitable continuity conditions. This approach yields a reference trajectory. A controller and differential GPS are integrated in a real-world car and allows a reference trajectory to be followed in real-time. A benefit of this approach is the very high accuracy obtained in reproducing the reference trajectory. Hence, it can be used for testing different setups of cars under the same conditions while excluding the comparatively large influence of a real-world driver. In this article, we will focus on a method for generating the reference trajectory and report our experiences with this algorithm. The method allows an locally optimal solution to be computed for various handling courses in a robust way.

Keywords Optimal control · Automatic test-driving · Direct discretization method · Moving horizon

1 Introduction

A moving horizon approach was implemented at Volkswagen AG, Germany, for the computation of reference trajectories for an automatically driven prototype car,

M. Gerdts (✉)
School of Mathematics, University of Birmingham, Edgbaston, Birmingham B15 2TT, UK
e-mail: gerdtsm@maths.bham.ac.uk
url: web.mat.bham.ac.uk/M.Gerdts

S. Karrenberg · B. Müller-Beßler · G. Stock
Volkswagen AG, Konzernforschung, 38436 Wolfsburg, Germany

Fig. 1 Test-drive of a car along a test-course close to the dynamical limit

Fig. 2 Automatically driven car: the human driver has no influence

cf. Fig. 1. Experiments indicate that even professional drivers produce comparatively large ranges of dispersion when driving several times close to the dynamical limit through a given test-course. This unpredictable influence of the human driver is particularly disadvantageous during an incremental search for an optimal setup of the car. The methods in this article allow us to fix the influence of the driver, who is replaced by a closed-loop controller, in order to provide a reproducible testing environment for the tuning and calibration of automobiles. The vehicle moves without a human driver close to its dynamical limit, cf. Fig. 2. Notice that the driver in Fig. 2 does not touch the steering wheel.

The overall procedure for the automatically driven car along a test-course consists of the following steps:

- exploration of the unknown test-course;
- determination of a locally optimal trajectory through the test-course;
- driving through the test-course tracking the previously determined trajectory using a closed-loop controller and differential GPS.

The first step aims at the exploration of the test-course which is assumed to be unknown. As the course is not known, this first step will be performed carefully at low speed. Once the boundaries of the test-course are determined using an infrared device, the second phase starts. The second phase aims at driving, e.g., as fast as possible through the test-course. More precisely, an optimal reference trajectory, e.g. in view of lap time, is sought. This is a key issue in automatic test-driving as there is no driver in the car. Consequently, the driver has to be substituted by an algorithm that provides steering and braking inputs to the car. These inputs can be obtained by solving appropriate control and state constrained optimal control problems in combination with a moving horizon approach. The numerical methods used for the solution of the highly nonlinear optimal control problems have to be very robust since it is essential that the methods are capable of providing results automatically for many different handling courses. For this reason, we employ a direct discretization method for numerical solution. The direct discretization method has proved its ability in numerous practical applications originating from different disciplines and turns out to be robust and easy to use.

Unfortunately, an optimal trajectory currently cannot be provided in real-time owing to the comparatively high computational costs for solving the optimal control problems. Therefore, a controller is used to provide the control inputs (steering, braking) to the car in real-time while tracking the previously computed optimal trajectory. Details of the controller design can be found in Müller-Beßler et al. (2006).

In this article we will focus mainly on the second step, i.e. the determination of an optimal trajectory resp. a locally optimal trajectory. The organization of the article is as follows. In Sect. 2 the mathematical models of the car and the test-course are discussed. Section 3 addresses the generation of a locally optimal reference trajectory for the automatically driven car along the test-course. Numerical results for some test-courses are presented in Sect. 4.

2 Modeling issues

2.1 Car model

In this article the single-track car model is used. It is a simplified car model, which is commonly used in the automobile industry for basic investigations of the dynamical behavior of cars, see for example Mayr (1991), Neculau (1992), Moder (1994).

A simplifying assumption made to derive the single-track car model is that the rolling and pitching behavior of the car body can be neglected, that is, the roll angle and the pitch angle are small. These assumptions justify the replacement of the two wheels on the front and rear axle by a virtual wheel located in the center of the respective axle. Furthermore, due to the simplifying assumptions it can be presumed that the car's center of gravity is located on the roadway and therefore, it is sufficient to consider the motion of the car solely in the horizontal plane. The car model includes two control variables for the driver: the steering angle velocity $|w_\delta| \leq 0.5$ [rad/s] and a function F_B with values in $[F_{Bmin}, F_{Bmax}]$, $F_{Bmin} = -5000$ [N], $F_{Bmax} = 15000$ [N],

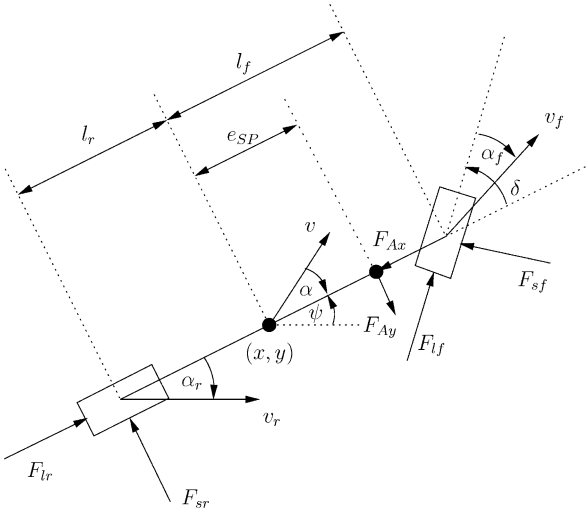

Fig. 3 Geometrical description of the single-track car model

which models a combined brake (if $F_B > 0$) and acceleration (if $F_B < 0$) assembly. The configuration of the car is depicted in Fig. 3.

Herein, (x, y) denotes the center of gravity in a reference coordinate system, v, v_f, v_r denote the velocity of the car and the velocities of the front and rear wheel, respectively, δ, α, ψ denote the steering angle, the side slip angle, and the yaw angle, respectively, α_f, α_r denote the slip angles at front and rear wheel, respectively, F_{sf}, F_{sr} denote the lateral tire forces (side forces) on front and rear wheel, respectively, F_{lf}, F_{lr} denote the longitudinal tire forces on front and rear wheel, respectively, l_f, l_r, e_{SP} denote the distances from the center of gravity to the front and rear wheel, and to the drag mount point, respectively, F_{Ax}, F_{Ay} denote the drag force on the car due to air resistance and side wind, respectively, and m denotes the mass of the car.

The equations of motion are given by the following system of ordinary differential equations

$$x' = v_x, \tag{1}$$

$$y' = v_y, \tag{2}$$

$$\psi' = w_\psi, \tag{3}$$

$$v_x' = \frac{1}{m}\left[F_x \cos\psi - F_y \sin\psi\right], \tag{4}$$

$$v_y' = \frac{1}{m}\left[F_x \sin\psi + F_y \cos\psi\right], \tag{5}$$

$$w_\psi' = \frac{1}{I_{zz}}\left[F_{sf} \cdot l_v \cdot \cos\delta - F_{sr} \cdot l_h - F_{Ay} \cdot e_{SP} + F_{lf} \cdot l_v \cdot \sin\delta\right], \tag{6}$$

$$\delta' = w_\delta, \tag{7}$$

where

$$F_x = F_{lr} - F_{Ax} + F_{lf}\cos\delta - F_{sf}\sin\delta,$$
$$F_y = F_{sr} - F_{Ay} + F_{lf}\sin\delta + F_{sf}\cos\delta.$$

The side slip angle α and the absolute velocity v are given by

$$\alpha = \psi - \arctan\left(\frac{y'}{x'}\right), \qquad v = \sqrt{(x')^2 + (y')^2}.$$

The lateral tire forces are functions of the respective slip angles (and the tire loads, which are constant in our model). A famous model for the lateral tire forces is the 'magic formula' of Pacejka and Bakker (1993):

$$F_{sf}(\alpha_f) = D_f \sin\big(C_f \arctan\big(B_f\alpha_f - E_f\left(B_f\alpha_f - \arctan(B_f\alpha_f)\right)\big)\big),$$
$$F_{sr}(\alpha_r) = D_r \sin\big(C_r \arctan\big(B_r\alpha_r - E_r\left(B_r\alpha_r - \arctan(B_r\alpha_r)\right)\big)\big),$$

compare Fig. 4. Herein, $B_f, B_r, C_f, C_r, D_f, D_r, E_f, E_r$ are constants depending on the tire.

According to Mitschke (1990), p. 23, (notice the opposite sign in the definition of α) the slip angles are given by

$$\alpha_f = \delta - \arctan\left(\frac{l_f\psi' - v\sin\alpha}{v\cos\alpha}\right), \qquad \alpha_r = \arctan\left(\frac{l_r\psi' + v\sin\alpha}{v\cos\alpha}\right).$$

The drag due to air resistance is modeled by

$$F_{Ax} = \frac{1}{2}\cdot c_w \cdot \rho \cdot A \cdot v^2,$$

where c_w is the air drag coefficient, ρ is the air density, and A is the effective flow surface. In this article, it is assumed that there is no side wind, i.e. $F_{Ay} = 0$.

The longitudinal tire forces on front and rear wheel, respectively, are given by

$$F_{lf} = -F_{Bf} - F_{Rf}, \qquad F_{lr} = -F_{Br} - F_{Rr},$$

where F_{Bf} and F_{Br} are the braking forces and F_{Rf} and F_{Rr} denote the rolling resistances on the front wheel and the rear wheel, respectively.

In the sequel, we assume that the car has front wheel drive. The force F_B controlled by the driver is distributed on the front and rear wheels according to

$$F_{Bf} = \begin{cases} \frac{2}{3}F_B, & \text{if } F_B > \Delta, \\ \frac{5}{6}F_B - \frac{1}{4\Delta}F_B^2 + \frac{1}{12\Delta^3}F_B^4, & \text{if } |F_B| \le \Delta, \\ F_B, & \text{otherwise,} \end{cases}$$

$$F_{Br} = \begin{cases} \frac{1}{3}F_B, & \text{if } F_B > \Delta, \\ \frac{2}{3\Delta}F_B^2 - \frac{1}{3\Delta^2}F_B^3, & \text{if } 0 < F_B \le \Delta, \\ 0, & \text{otherwise,} \end{cases}$$

where $\Delta = 0.01$. Recall that $F_B > 0$ corresponds to a braking force whereas $F_B < 0$

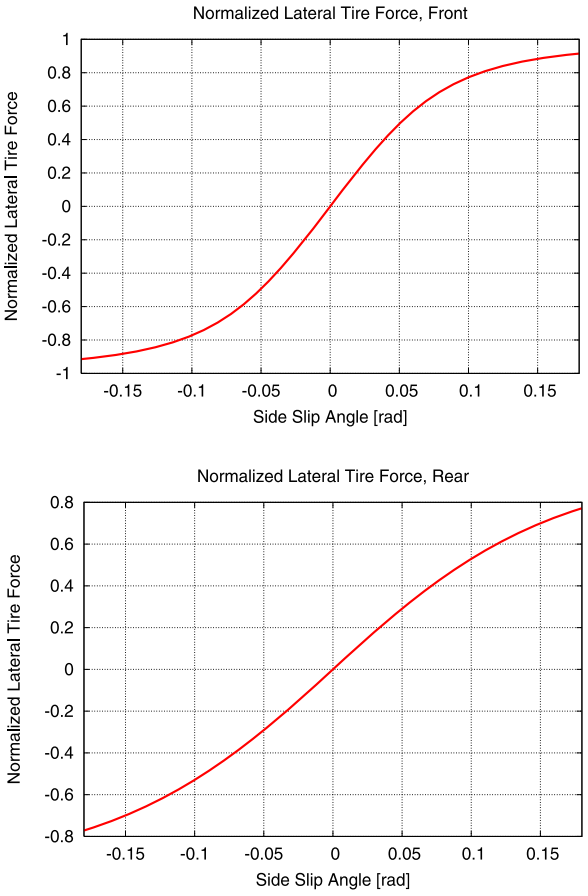

Fig. 4 Shape of the lateral tire forces on front (*top*) and rear (*bottom*) axis as functions of the slip angle computed by the 'magic formula' of Pacejka and Bakker (1993). Notice, that the tire forces are normalized w.r.t. the maximum tire force. The actual curves of the tire forces without normalization are subject to proprietary rights of Volkswagen

is an accelerating force. Finally, the rolling resistance forces are given by

$$F_{Rf} = f_R(v) \cdot F_{zf}, \qquad F_{Rr} = f_R(v) \cdot F_{zr},$$

where

$$f_R(v) = f_{R0} + f_{R1}\frac{v}{100} + f_{R4}\left(\frac{v}{100}\right)^4 \quad (v \text{ in [km/h]}),$$

is the friction coefficient and

$$F_{zf} = \frac{m \cdot l_r \cdot g}{l_f + l_r}, \qquad F_{zr} = \frac{m \cdot l_f \cdot g}{l_f + l_r}$$

denote the static tire loads on the front and rear wheel, respectively, cf. Risse (1991).

For the following numerical computations we used realistic data for the various parameters involved in this model. Unfortunately, these parameter values are proprietary and may not be published. For a different parameter set which is quite realistic please refer to Gerdts (2005).

2.2 Test-course

The shape of the test-course is defined by an a-priori unknown number of gates. It is assumed that the starting point and the end point of the test-course are the same, i.e. the test-course is periodic. Each gate consists of two pylons, one marking the left boundary and the other marking the right boundary. In a first phase the automatically driven car explores the unknown boundaries (i.e. the gates) of the test-course by itself using an infrared device. The slowly driving car starts its drive without any knowledge of the track. While driving, the infrared device detects the positions in the (x, y)-plane of those gates that are within reaching distance of the infrared device. Having detected the next gate, the car determines its way through this gate. The local path through the next gate is defined by an interpolating spline curve of the next gate and some previous gates. Herein, the exact coordinates (within a tolerance of 2 [cm]) of the gates are computed by differential GPS. Having completed the first phase successfully, the positions of all gates defining the shape of the overall test-course are known. Let these positions be denoted

$$(x_0^\ell, y_0^\ell), \quad \ldots, \quad (x_{N_\ell}^\ell, y_{N_\ell}^\ell)$$

for the pylons on the left and

$$(x_0^r, y_0^r), \quad \ldots, \quad (x_{N_r}^r, y_{N_r}^r)$$

for the pylons on the right. Notice that the case $N_\ell \neq N_r$ is not excluded although the interpretation of gates is not valid anymore in this case. In the sequel, it is just important that the points defining the boundaries are classified such that they belong either to the left boundary or to the right boundary. Furthermore, we assume the periodic conditions

$$(x_0^\ell, y_0^\ell) = (x_{N_\ell}^\ell, y_{N_\ell}^\ell), \qquad (x_0^r, y_0^r) = (x_{N_r}^r, y_{N_r}^r)$$

leading to a closed circuit. The left and the right boundaries of the test-course are modeled by interpolating periodic spline curves

$$\gamma^\ell = (x^\ell, y^\ell) : [0, s_{N_\ell}^\ell] \to \mathbb{R}^2, \ s \mapsto \gamma^\ell(s) = (x^\ell(s), y^\ell(s)),$$
$$\gamma^r = (x^r, y^r) : [0, s_{N_r}^r] \to \mathbb{R}^2, \ s \mapsto \gamma^r(s) = (x^r(s), y^r(s)),$$

which satisfy the interpolating conditions

$$\gamma^\ell(s_i^\ell) = (x^\ell(s_i^\ell), y^\ell(s_i^\ell)) = (x_i^\ell, y_i^\ell), \quad i = 0, \ldots, N_\ell,$$
$$\gamma^r(s_i^r) = (x^r(s_i^r), y^r(s_i^r)) = (x_i^r, y_i^r), \quad i = 0, \ldots, N_r,$$

where

$$s_0^\ell = s_0^r = 0,$$
$$s_i^\ell = s_{i-1}^\ell + \sqrt{(x_i^\ell - x_{i-1}^\ell)^2 + (y_i^\ell - y_{i-1}^\ell)^2}, \quad i = 1, \ldots, N_\ell,$$
$$s_i^r = s_{i-1}^r + \sqrt{(x_i^r - x_{i-1}^r)^2 + (y_i^r - y_{i-1}^r)^2}, \quad i = 1, \ldots, N_r.$$

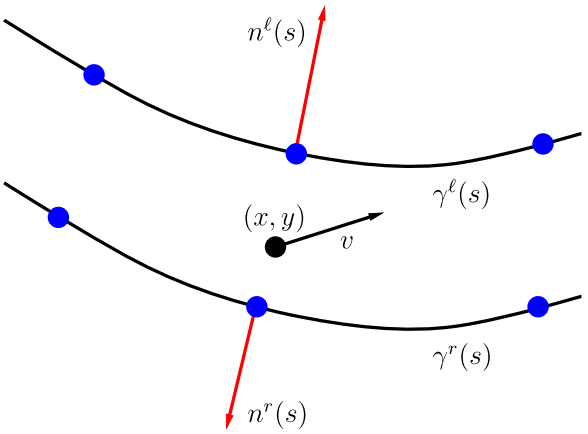

Fig. 5 (Color online) Gates of the test-course (*blue*), spline approximation of the boundaries of the track (*black*), outer normals (*red*), and center of gravity (x, y) and velocity vector v of the car

The curves $\gamma^\ell(\cdot)$ and $\gamma^r(\cdot)$ are twice continuously differentiable by construction. The outer normals to the right and left boundaries, respectively, at curve parameter s are given by

$$n^\ell(s) = \begin{pmatrix} -(y^\ell)'(s) \\ (x^\ell)'(s) \end{pmatrix}, \qquad n^r(s) = \begin{pmatrix} (y^r)'(s) \\ -(x^r)'(s) \end{pmatrix}.$$

In view of test-driving it is important to decide whether a given point $(x(t), y(t))^\top$, e.g. the center of gravity of the car at time t, is on the track or off the track, i.e. beyond the boundaries, cf. Fig. 5.

Let $\gamma^r(s^r) = (x^r(s^r), y^r(s^r))^\top$ resp. $\gamma^\ell(s^\ell) = (x^\ell(s^\ell), y^\ell(s^\ell))^\top$ denote those points on the curves γ^r and γ^ℓ, respectively, with minimal distance to a given point $(x(t), y(t))^\top$ at time t. Then, $(x(t), y(t))^\top$ is on the track if the following state constraints are satisfied:

$$\big(x(t) - x^r(s^r), y(t) - y^r(s^r)\big)\frac{n^r(s^r)}{\|n^r(s^r)\|} \le 0, \tag{8}$$

$$\big(x(t) - x^\ell(s^\ell), y(t) - y^\ell(s^\ell)\big)\frac{n^\ell(s^\ell)}{\|n^\ell(s^\ell)\|} \le 0. \tag{9}$$

It remains to compute s^r resp. s^ℓ i.e. those curve parameters that yield the respective points of minimal distance. We restrict the discussion to the computation of s^ℓ.

Consider the squared minimal distance to the curve γ^ℓ:

$$d^\ell(t) := \min_{0 \le s \le s^\ell_{N_\ell}} \left\{ (x(t) - x^\ell(s))^2 + (y(t) - y^\ell(s))^2 \right\}.$$

The constraint $0 \le s \le s^\ell_{N_\ell}$ may be neglected as the spline is periodic. Define

$$s^\ell(t) := \arg\min_s \left\{ (x(t) - x^\ell(s))^2 + (y(t) - y^\ell(s))^2 \right\}.$$

Necessarily, it holds for all t:

$$0 = 2(x(t) - x^\ell(s^\ell(t)))(-(x^\ell)'(s^\ell(t))) + 2(y(t) - y^\ell(s^\ell(t)))(-(y^\ell)'(s^\ell(t))).$$

Differentiation of this identity in t w.r.t. t yields a differential equation for $s^\ell(t)$ (dependence on t is suppressed for brevity):

$$(s^\ell)' = \frac{x'(x^\ell)'(s^\ell) + y'(y^\ell)'(s^\ell)}{(x^\ell)'(s^\ell)^2 + (y^\ell)'(s^\ell)^2 - (x - x^\ell(s^\ell))(x^\ell)''(s^\ell) - (y - y^\ell(s^\ell))(y^\ell)''(s^\ell)}. \tag{10}$$

Initially, at $t = 0$ the curve parameter $s^\ell(0)$ has to be computed. This is particularly easy for test-courses starting with a straight segment. In this case $s^\ell(0) = 0$ is suitable. If the test-course does not start with a straight segment, the computation of $s^\ell(0)$ is more involved as a nonlinear program has to be solved in order to determine $d^\ell(0)$.

Analogously, we obtain a differential equation for s^r:

$$(s^r)' = \frac{x'(x^r)'(s^r) + y'(y^r)'(s^r)}{(x^r)'(s^r)^2 + (y^r)'(s^r)^2 - (x - x^r(s^r))(x^r)''(s^r) - (y - y^r(s^r))(y^r)''(s^r)}. \tag{11}$$

The following modifications are suggested, if the width b of the car has to be taken into account.

- The left and the right side of the car body (on the line passing through the center of gravity and being perpendicular to the longitudinal axis of the car) are given by the points

$$z^\ell = \begin{pmatrix} x \\ y \end{pmatrix} + \frac{b}{2}\begin{pmatrix} -\sin\psi \\ \cos\psi \end{pmatrix}, \qquad z^r = \begin{pmatrix} x \\ y \end{pmatrix} + \frac{b}{2}\begin{pmatrix} \sin\psi \\ -\cos\psi \end{pmatrix},$$

  where ψ denotes the yaw angle of the car and $(x, y)^\top$ its center of gravity. The above formulae for $(x, y)^\top$ have to be derived likewise for both, z^ℓ and z^r. In particular, the minimal distances to the curves γ^ℓ and γ^r have to be computed for both, z^ℓ and z^r. Hence, the effort to determine whether both, z^ℓ and z^r, are on the track is twice as large as for $(x, y)^\top$.
- We neglect the yaw angle in the previous considerations in order to reduce the computational effort and consider the points

$$z^\ell = \begin{pmatrix} x \\ y \end{pmatrix} + \frac{b}{2}\frac{n^\ell}{\|n^\ell\|}, \qquad z^r = \begin{pmatrix} x \\ y \end{pmatrix} + \frac{b}{2}\frac{n^r}{\|n^r\|}.$$

  The points z^ℓ and z^r have to satisfy the state constraints

$$\left(z_1^r - x^r, z_2^r - y^r\right)\frac{n^r(s^r)}{\|n^r(s^r)\|} \le 0, \tag{12}$$

$$\left(z_1^\ell - x^\ell, z_2^\ell - y^\ell\right)\frac{n^\ell(s^\ell)}{\|n^\ell(s^\ell)\|} \le 0. \tag{13}$$

  These formulae depend on the minimal distance to the boundaries of the center of gravity $(x, y)^\top$ only and we may use the corresponding formulae from above.

3 Optimal trajectory generation

As we aim at driving as fast as possible through the test-course, we need an optimal driver. Such a driver is modeled by a suitable control and state constrained optimal control problem. A natural objective functional in this problem would be to minimize the time t_f needed to complete one round of the test-course. However, this minimum time optimal control problem will cause difficulties in the moving horizon algorithm below as a free final time cannot be handled properly and stopping criterions are difficult to satisfy. For these reasons, instead of the minimum time optimal control problem we investigate an alternative (yet not equivalent) objective function on a fixed time interval $[0, t_f]$ with sufficiently large and fixed final time t_f. The alternative objective function aims at maximizing within a fixed time the distance covered by the car as it proceeds through the test-course. This progression on the test-course is measured by adding the respective progression on the right boundary $s^r(t_f) - s^r(0)$ and the left boundary $s^\ell(t_f) - s^\ell(0)$. Often, we will add a regularization term to the objective. Thus, we have to solve the following optimal control problem with weighting parameters $c_1, c_2 \geq 0$:

Problem 1 *Minimize*

$$-c_2\left(s^r(t_f) - s^r(0) + s^\ell(t_f) - s^\ell(0)\right) + c_1 \int_0^{t_f} w_\delta(t)^2 dt$$

w.r.t. $z = (x, y, \psi, v_x, v_y, w_\psi, \delta, s^r, s^\ell)^\top \in W^{1,\infty}([0, t_f], \mathbb{R}^9)$, $u = (w_\delta, F_B)^\top \in L^\infty([0, t_f], \mathbb{R}^2)$ *with* t_f *fixed and sufficiently large subject to the differential equations* (1)–(7), (10)–(11), *the initial condition* $z(0) = Z_0$, *the state constraints* (8)–(9) (*resp.* (12)–(13)), *the control constraints* $|w_\delta| \leq 0.5$, $-5000 \leq F_B \leq 15000$, *and the boundary conditions* $s^r(t_f) \geq s^r_{N_r}$ *and* $s^\ell(t_f) \geq s^\ell_{N_\ell}$.

In the above optimal control problem, the final time t_f has to be chosen large enough in order to allow for a completion of at least one lap of the course. Notice that the actual value of t_f in our application from test-driving is not important.

We briefly discuss the direct solution approach for general optimal control problems. Notice that Problem 1 can be transformed by standard techniques to the following general form where φ, f, c, r are sufficiently smooth functions of appropriate dimension.

Problem 2 *Minimize*

$$\varphi(z(t_0), z(t_f))$$

w.r.t. $z \in W^{1,\infty}([t_0, t_f], \mathbb{R}^{n_z})$ *and* $u \in L^\infty([t_0, t_f], \mathbb{R}^{n_u})$ *subject to the differential equation*

$$z'(t) = f(t, z(t), u(t)), \quad \text{a.e. in } [t_0, t_f]$$

the control-state constraints

$$c(t, z(t), u(t)) \leq 0, \quad \text{a.e. in } [t_0, t_f],$$

and the boundary conditions

$$r(z(t_0), z(t_f)) = 0.$$

Problem 2 will be solved numerically by a direct discretization approach as in Gerdts (2003a). We outline the basic ideas very briefly. Related approaches can be found in, e.g., Bock and Plitt (1984), Goh and Teo (1988), Hager (1990), Teo et al. (1991, 1999), Polak et al. (1993), von Stryk (1993), Martin and Teo (1994), Büskens (1998), Betts (2001), Jennings et al. (2004), Wu and Teo (2006). Please refer to chapter one of Grötschel et al. (2001) and the literature cited therein for an overview on direct discretization methods.

The control u is discretized on the grid

$$\pi_N := \{t_0 + ih \mid i = 0, 1, \ldots, N\}, \quad N \in \mathbb{N}, \tag{14}$$

with step-size $h = (t_f - t_0)/N$ by the piecewise constant function

$$u_h(\tau) = u_i \quad \text{for } \tau \in [t_i, t_{i+1}),\ i = 0, \ldots, N-1.$$

The differential equation is discretized on the grid π_N by an s-staged Runge-Kutta method with coefficients b_i, c_i, a_{ij}, $1 \le i, j \le s$, cf. Hairer et al. (1993):

$$z_{i+1} = z_i + h \sum_{j=1}^{s} b_j k_j, \quad i = 0, \ldots, N-1, \tag{15}$$

$$k_j = f\left(t_i + c_j h, z_i + h \sum_{\ell=1}^{s} a_{j\ell} k_\ell, u_i\right). \tag{16}$$

Piecewise linear interpolation of the values z_i, $i = 0, \ldots, N$, yields a continuous state approximation $z_h : [t_0, t_f] \to \mathbb{R}^{n_x}$. A discretization of Problem 2 is thus given by the subsequent finite dimensional nonlinear programming problem:

Problem 3 (Discretized Optimal Control Problem (DOCP)) *Minimize*

$$\varphi(z_0, z_N)$$

w.r.t. $(z_0, \ldots, z_N, u_0, \ldots, u_{N-1})^\top \in \mathbb{R}^{(N+1)n_x + Nn_u}$ *subject to* (15)–(16) *and*

$$c(t_i, z_i, u_i) \le 0, \quad i = 0, \ldots, N,$$
$$r(z_0, z_N) = 0.$$

DOCP can be solved numerically by the sequential quadratic programming (SQP) method, cf. Gill et al. (1981) and Schittkowski (1983). Basically there are two philosophies in doing this. The full discretization approach includes the Runge-Kutta constraints (15)–(16) explicitly as equality constraints in DOCP. This leads to a large but sparse nonlinear program which can be solved by a large scale version of SQP,

cf. Gill et al. (2002). The advantage is that Jacobians of the constraints are easy to compute.

The reduced approach does not include (15)–(16) as explicit constraints in DOCP but solves these equation step by step starting at t_0. This leads to a small but dense nonlinear program since the variables z_i with exception of z_0, do not appear anymore. Gradients needed for the SQP method may be obtained by a sensitivity analysis of the Runge-Kutta equations, cf. Gerdts (2003a), or simply by finite difference approximations. The sensitivity analysis is known as IND (internal numerical differentiation), cf. Bock (1981).

For the upcoming numerical calculations we used the reduced approach, which is implemented in the software package OC-ODE, cf. Gerdts (2006). Actually, OC-ODE allows the use of different discretization methods. For the calculations we used the classical Runge-Kutta discretization scheme of order 4 given by the Butcher-table

$$\begin{array}{c|cccc} 0 & & & & \\ 1/2 & 1/2 & & & \\ 1/2 & 0 & 1/2 & & \\ 1 & 0 & 0 & 1 & \\ \hline & 1/6 & 1/3 & 1/3 & 1/6 \end{array}$$

Convergence results for the discretized problem (DOCP) can be found in Malanowski et al. (1997), Dontchev et al. (2000a, 2000b), and Hager (2000). We refer the reader to these references as it is not within the scope of this article to discuss the various assumptions needed for convergence.

3.1 A moving horizon technique

With increasing length and more complicated structures of the test-course the computational difficulties in the numerical solution of the optimal control problem grow rapidly. Firstly, for a long-distance test-course with complicated structure it is very difficult and time consuming to construct a sufficiently good initial guess needed by the SQP method in order to converge. Moreover, severe stability problems may occur within the iterative solution of DOCP owing to the fact that the state may be very sensitive w.r.t. changes of the control especially at the beginning of the test-course. Secondly, the number of grid points N has to be adapted to the length of the test-course for a sufficiently accurate numerical solution and thus the resulting nonlinear program becomes large scale for long test-courses. These arguments suggest not to solve the optimal control problem for the complete test-course. Instead, we will employ a moving horizon approach as in Gerdts (2003b). The idea is to decompose the global optimal control problem into a sequence of local optimal control problems which are comparatively easy to solve. The local solutions are combined by appropriate continuity conditions, such that a trajectory along the complete test-course is obtained. The structure of the local optimal control problems is similar to that of the global problem except, that only a comparatively short time horizon is considered. Hence, only parts of the overall test-course will be optimized at any one time. This situation reflects reality quite well as a human driver in general will only know the test-course within his personal range of vision.

The moving horizon algorithm depends on the following parameters, cf. Fig. 6:

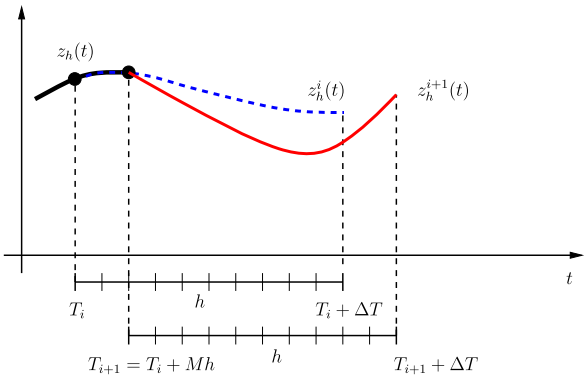

Fig. 6 Principle of the moving horizon approach: the process is considered in a moving time slot of length ΔT with discretization parameter N and step size $h = \Delta T/N$. By accepting the interval $[T_i, T_i + Mh]$ of the local trajectory $z_h^i(\cdot)$ the transition to the next time slot $[T_{i+1}, T_{i+1} + \Delta T]$ is performed

- length of the local time horizon $\Delta T > 0$ (range of vision)
- number of (equidistant) time intervals $N \geq 1$ used for discretization of the local optimal control problems (discretization parameter)
- number of (equidistant) time intervals $0 < M \leq N$ to be accepted from the current local solution (shifting parameter)

For simplicity, in our application we have chosen the same parameters for all local optimal control problems. Of course, these parameters may be chosen individually for each local optimal control problem according to some adaptive algorithm.

Algorithm 1 (Moving Horizon Algorithm)

(i) *Let the range of vision* $\Delta T > 0$, *the discretization parameter* $N \geq 1$, *the shifting parameter* $0 < M \leq N$, *the initial position* Z_0, *and weighting parameters* c_1, c_2 *be given. Let* $T_0 := t_0$, $i := 0$.

(ii) *Solve the local optimal control problem* $OCP(T_i, \Delta T, Z_i)$ *by the direct discretization method on an equidistant grid with step size* $h = \Delta T/N$. *Let* (z_h^i, u_h^i) *denote the numerical solution.*

(iii) *If the test-course is completed, i.e. if* $T_i + \Delta T \geq t_f$, *define*

$$(z_h(t), u_h(t)) := \begin{cases} (z_h^j(t), u_h^j(t)) & \text{for } t \in [T_j, T_j + Mh) \text{ and } 0 \leq j \leq i-1, \\ (z_h^i(t), u_h^i(t)) & \text{for } t \in [T_i, t_f] \end{cases} \tag{17}$$

and stop.

(iv) *Let* $T_{i+1} := T_i + Mh$, $Z_{i+1} := z_h^i(T_i + Mh)$, $i := i + 1$, *and go to* (ii).

The local optimal control problems $OCP(T_i, \Delta T, Z_i)$ to be solved in step (ii) of the moving horizon algorithm are given as follows.

Problem 4 ($OCP(T, \Delta T, Z)$) *Minimize*

$$-c_2\left(s^r(T+\Delta T) - s^r(T) + s^\ell(T+\Delta T) - s^\ell(T)\right) + c_1 \int_T^{T+\Delta T} w_\delta(t)^2 dt$$

w.r.t. $z = (x, y, \psi, v_x, v_y, w_\psi, \delta, s^r, s^\ell)^\top \in W^{1,\infty}([T, T + \Delta T], \mathbb{R}^9)$ *and* $u = (w_\delta, F_B)^\top \in L^\infty([T, T + \Delta T], \mathbb{R}^2)$ *subject to the differential equations* (1)–(7), (10)–(11), *the initial condition* $z(T) = Z$, *the state constraints* (8)–(9) (*resp.* (12)–(13)), *and the control constraints* $|w_\delta| \le 0.5$, $-5000 \le F_B \le 15000$.

Notice, that in (17) only the first part $[T_j, T_j + Mh]$ is accepted from each local solution (z_h^j, u_h^j) while the remaining trajectory in $(T_j + Mh, T_j + \Delta T]$ is neglected. This is done for the following reasons. Firstly, if the interval $(T_j + Mh, T_j + \Delta T]$ is chosen large enough, it can be guaranteed that the subsequent problem $OCP(T_{j+1}, \Delta T, Z_{j+1})$ possesses a solution (at least for our problem from test-driving). Otherwise it may happen that no solution exists because the set of feasible trajectories of $OCP(T_{j+1}, \Delta T, Z_{j+1})$ emanating from Z_{j+1} is empty. To maximize efficiency, it would be desirable to have $Mh = \Delta T$, i.e. the whole local trajectory is accepted, but this requires 'drive-on-conditions' to guarantee that the above mentioned set of feasible trajectories is non-empty. For general nonlinear test-courses such conditions are impossible to formulate. Secondly, as the local optimal control problems $OCP(T_j, \Delta T, Z_j)$ and $OCP(T_{j+1}, \Delta T, Z_{j+1})$ overlap each other on $[T_{j+1}, T_j + \Delta T]$, the solution (z_h^j, u_h^j) of $OCP(T_j, \Delta T, Z_j)$ may serve as an initial estimate for the solution (z_h^{j+1}, u_h^{j+1}) of $OCP(T_{j+1}, \Delta T, Z_{j+1})$.

Remark 5

- In general, the moving horizon algorithm as presented in this article cannot handle general boundary conditions like periodic conditions.
- As the solution (z_h, u_h) resulting from the moving horizon algorithm is defined via solutions of local optimal control problems, it may differ substantially from the solution of the global optimal control problem. However, it can be conjectured that (z_h, u_h) converges towards the global solution with increasing range of vision ΔT.
- The choice of the range of vision ΔT and the shifting parameter M is a compromise in the following sense. On the one hand, the larger ΔT and Mh are, the less is the number of local optimal control problems to be solved until completion of the circuit. On the other hand, with increasing ΔT the numerical difficulties within the solution of $OCP(T_i, \Delta T, Z_i)$ increase for the same reasons as described at the beginning of this section. In addition, if Mh is chosen too large, it may happen that the subsequent problem has no feasible solution.

4 Numerical results

All numerical results were computed on a personal computer with a 2.66 GHz processor and 512 MB RAM. The following parameters of the moving horizon approach were used for all computations: discretization parameter $N = 50$, shifting parameter $M = 2$ and weighting parameters $c_1 = 1, c_2 = 0.01$.

The SQP method used to solve the discretized optimal control problem uses an augmented Lagrangian merit function as in Schittkowski (1983) in combination with a non-monotone Armijo-type linesearch algorithm and the modified BFGS-update

of Powell (1978). The quadratic programming subproblems are solved by a primal active-set strategy. Inconsistent linearized constraints of the quadratic subproblem are treated as in Stoer (1985). The SQP method stops successfully if the first order necessary KKT conditions are satisfied within some prescribed optimality tolerance and the constraints are feasible within some prescribed feasibility tolerance. For each optimal control Problem 4 being solved in the moving horizon algorithm, the optimality tolerance and the feasibility tolerance of the underlying SQP method used to solve the discretizations of Problem 4 are set to 10^{-7}.

Numerical results are presented for a total of five different test-courses depicted in Figs. 7–11. The thick lines in the respective (x, y)-plots show the trajectory of the car's center of gravity. The respective bottom right plot shows the number of SQP iterations for each step of the moving horizon algorithm. A starter's flag indicates the starting position on the test-courses and points in the starting direction. All numerically computed reference trajectories (and many more) were integrated in the existing automatically driven prototype car and have been tested in reality. As expected, the automatically driven car reproduces the reference trajectories reliably. Experiments show that the lap time can be reproduced with a deviation of only 0.1 [s] per 1000 [m] of track length.

The stopping criterion for the moving horizon algorithm has to be specified. A natural stopping criterion would be to stop the moving horizon algorithm immediately as soon as exactly one lap is completed. This stopping criterion is difficult to realize because we are working with fixed time intervals in the moving horizon algorithm. Consequently, in the present form we are not able to stop exactly at the end of the test-course. It would be possible to adapt the algorithm to reach this goal, e.g. by formulating a different optimal control problem with free final time and terminal boundary conditions for the very final part of the test-course. However, fomulating terminal constraints is always crucial in the context of moving horizon algorithms as there is no guarantee that they can be fulfilled due to the local character of the algorithm. We didn't follow this approach because in our application it is not necessary to meet the finish line exactly. Moreover, this approach usually leads to a solution which touches the boundary of the track and does not allow to continue the drive beyond the final position without violating the constraints. So, instead we continued the moving horizon algorithm into the second lap until $s^r(T) > s^r_{N_r} + 100$ or $s^\ell(T) > s^\ell_{N_\ell} + 100$ hold. This explains the overlap of trajectories in the figures at the beginning of the test-courses.

The first course is the racing-course of Oschersleben in Sachsen-Anhalt, Germany. For this course having an approximate length of 3650 [m], the state constraints (12)–(13) are used as the width of the track is large. We used a range of vision of $\Delta T = 10$ [s] for this track.

The remaining test-courses are comparatively short. Their lengths range from approximately 350 [m] to 1300 [m]. For these courses, the width of the track is already reduced by the width of the car and hence, we used the state constraints (8)–(9) and a range of vision of $\Delta T = 5$ [s]. While test-course 1 is comparatively wide, the test-courses 2 to 4 are very narrow courses. Notice that especially the test-courses 2 and 3 are very narrow and the two pylons of each gate even seem to coincide in Figs. 10 and 11, which is due to the scaling of the figures. For the latter courses, an inexperience human driver will have difficulties when driving through these courses for the

first couple of times. With some training however, the human driver will improve and very good drivers may even outperform the automatically driven car. Nevertheless, the algorithm has no problems right from the beginning and performs consistently well.

It is interesting to note, that the control constraints for the steering angle velocity w_δ never become active in all subsequent results and that the steering effort for the driver is comparatively small. Likewise, the absolute value of the side-slip angle remains below 0.04 [rad] in all results, which shows that a dramatic sliding of the car cannot be observed. Nevertheless, a very high lateral acceleration up to approximately 12 [m/s^2] occurs in all subsequent results. This value shows that the car is actually close to the dynamical limit, but it moves rather smoothly without any agitated behavior. The longitudinal acceleration chart shows repeated periods of full braking, each period being comparatively short. There seems to be a correlation of SQP iterations and the longitudinal acceleration in Figs. 7–11. A full brake application leads to a drastic increase in the SQP iterations. This behavior is reasonable as a full brake application leads to a different structure of the optimal solution when it is compared to the previous solution without full braking which served as an initial guess.

We are now interested in the question of how the trajectories develop when the moving horizon algorithm is continued over several laps. For the ease of demonstration we focus on test-course 1 only and computed 8 consecutive laps. Figure 12 numerically demonstrates that all consecutive laps have the same structure with exception of the initial and final lap. The structural deviation of the initial and final lap from the pattern of the intermediate laps is not surprising because the initial lap uses a prescribed initial state (with small velocity) and the final lap is being stopped according to the stopping criterion described at the beginning of this section. It is interesting to see that already the second lap follows the same pattern as the consecutive laps. At least numerically, the results suggest that a periodic solution occurs, although there are small deviations in the control w_δ.

Finally, Fig. 13 demonstrates for comparatively simple test-course 3 how the result of the moving horizon algorithm relates to the optimal solution. Therefore, Problem 4 was solved for $T = 0$ and $\Delta T = 37$ with a discretization parameter $N = 400$. We used a homotopy to obtain the optimal solution. Herein, we initially computed an optimal solution for a time interval of 8 [s]. Using the previous optimal solution as an initial guess, the length of the time interval was increased successively by one second until a final length of 37 [s] was reached. The CPU times for the solution of the optimal control problems increased with increasing final time. It took approximately 2 CPU hours to solve the final optimal control problem. For this particular test-course, Fig. 13 demonstrates that the moving horizon algorithm produces a very good approximation of the optimal solution with only minor deviations. The qualitative structure of both solutions is the same. This coincides with our experiences for other (simple) test-courses. Of course, the quality of the moving horizon result depends on the choice of the parameters of the algorithm. For instance, if ΔT is chosen too small, larger deviations between optimal solution and moving horizon approximation occur as only a comparatively small fraction of the test-course is considered in the moving horizon algorithm. Another aspect from which the moving horizon algorithm benefits in this case is the narrow track of test-course 3. This does not allow

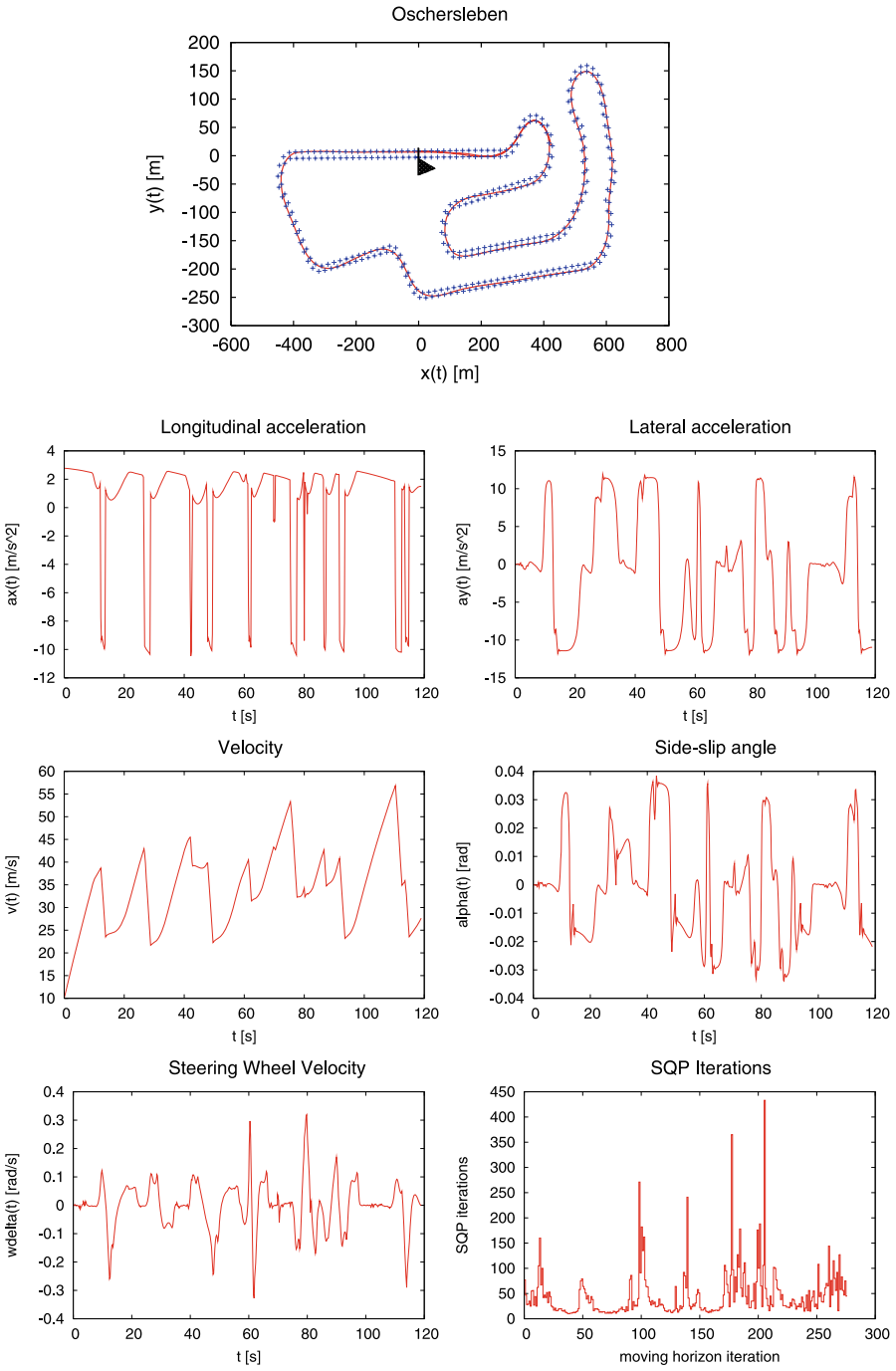

Fig. 7 Test-course of Oschersleben: $N_r = 181$, $N_\ell = 184$, $s^r_{N_r} = 3623$ [m], $s^\ell_{N_\ell} = 3687$ [m], CPU time 11 min 51.509 s

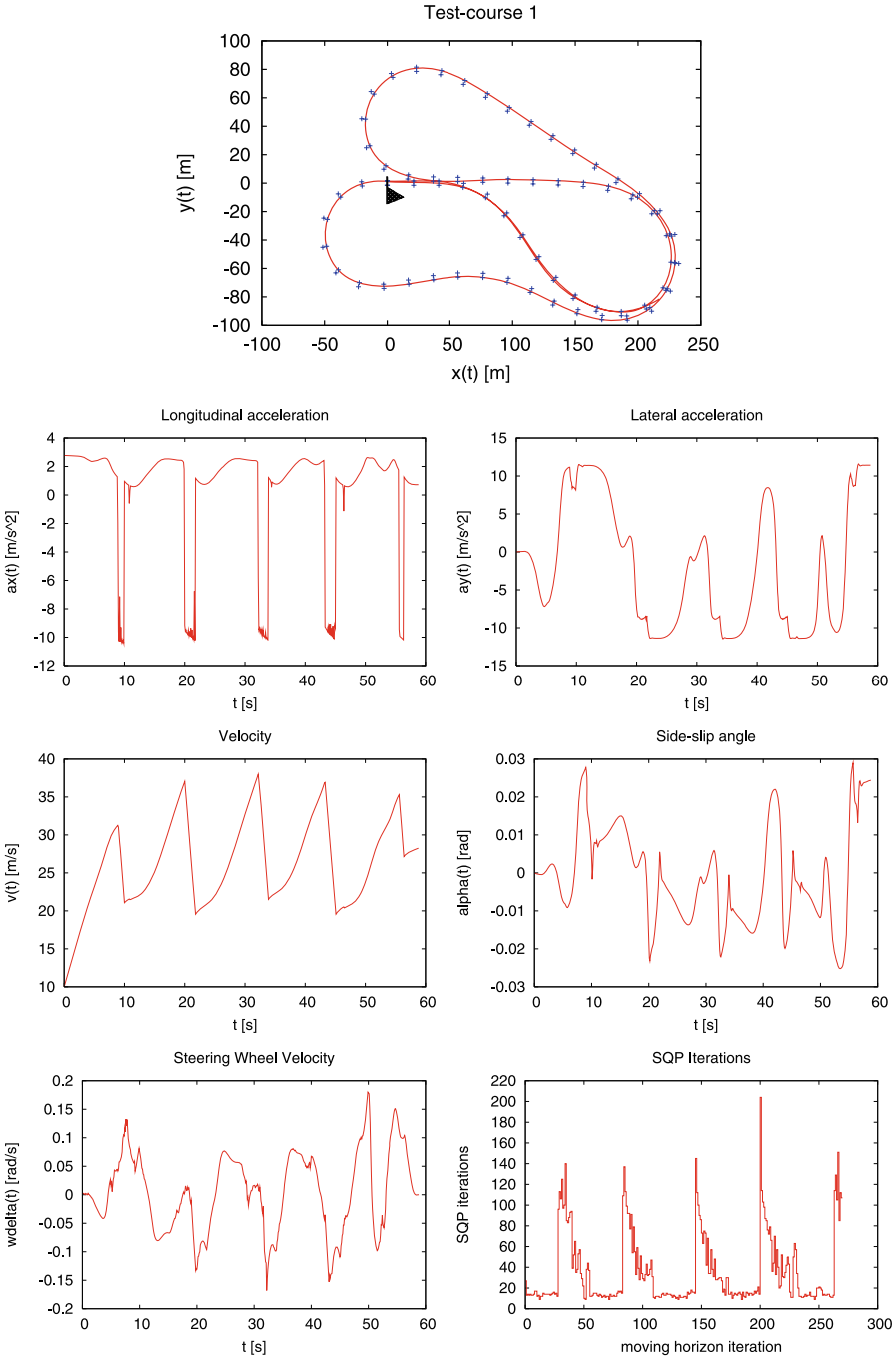

Fig. 8 Test-course 1: $N_r = 64$, $N_\ell = 64$, $s^r_{N_r} = 1286$ [m], $s^\ell_{N_\ell} = 1305$ [m], CPU time 5 min 20.890 s

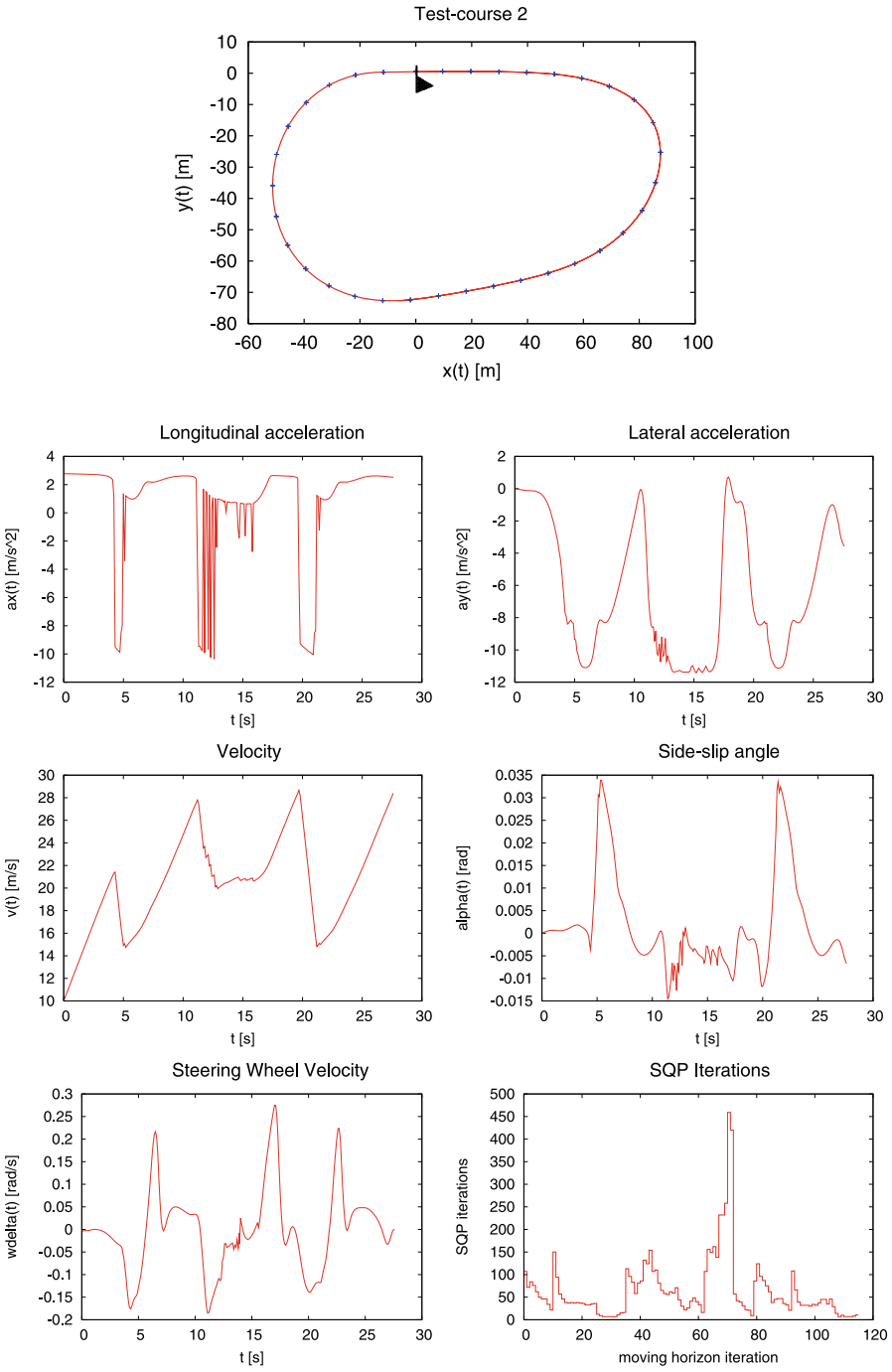

Fig. 9 Test-course 2: $N_r = 34$, $N_\ell = 34$, $s^r_{N_r} = 350.2$ [m], $s^\ell_{N_\ell} = 350.8$ [m], CPU time 5 min 24.780 s

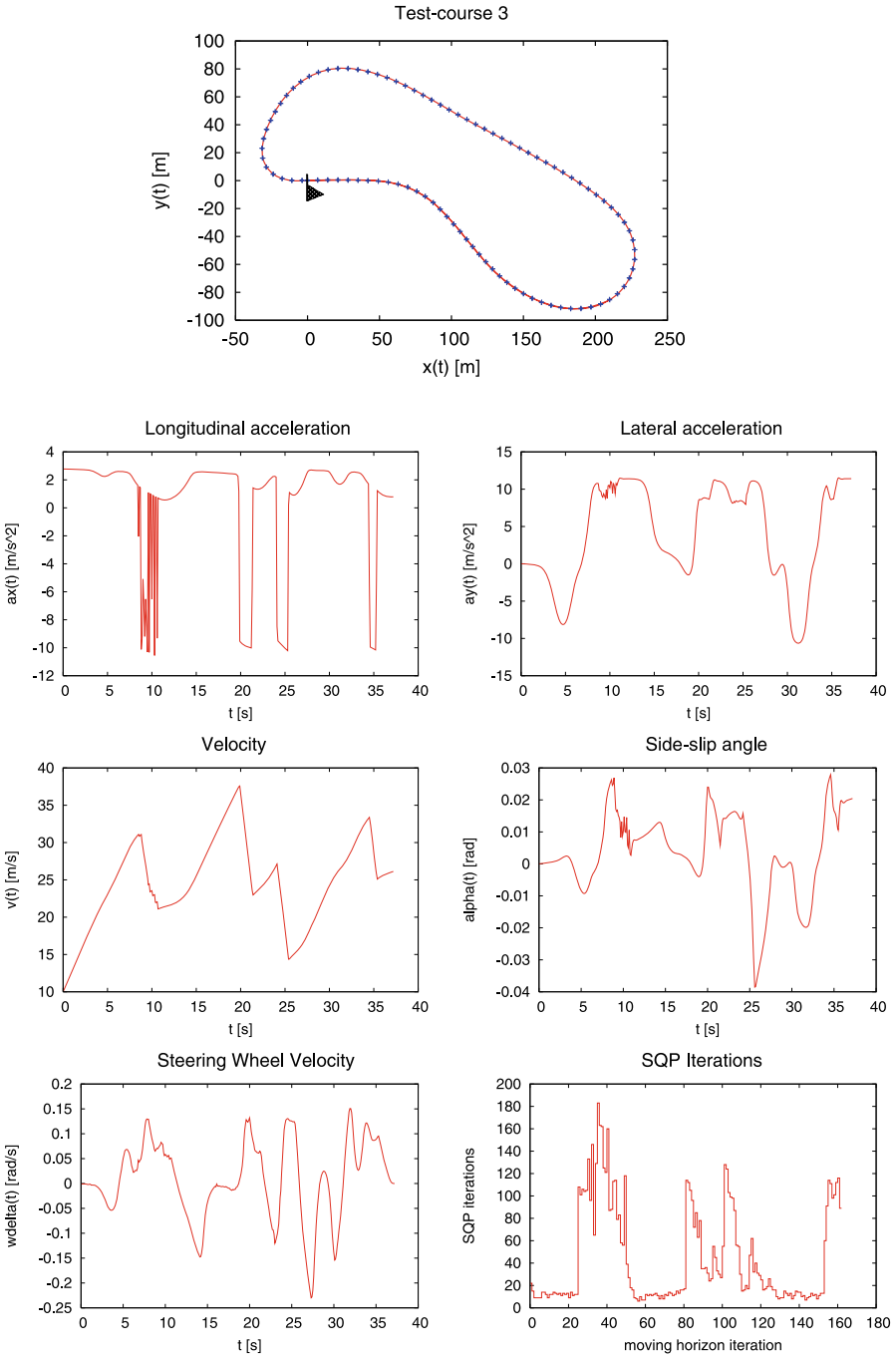

Fig. 10 Test-course 3: $N_r = 95$, $N_\ell = 95$, $s^r_{N_r} = 670$ [m], $s^\ell_{N_\ell} = 666.8$ [m], CPU time 4 min 12.632 s

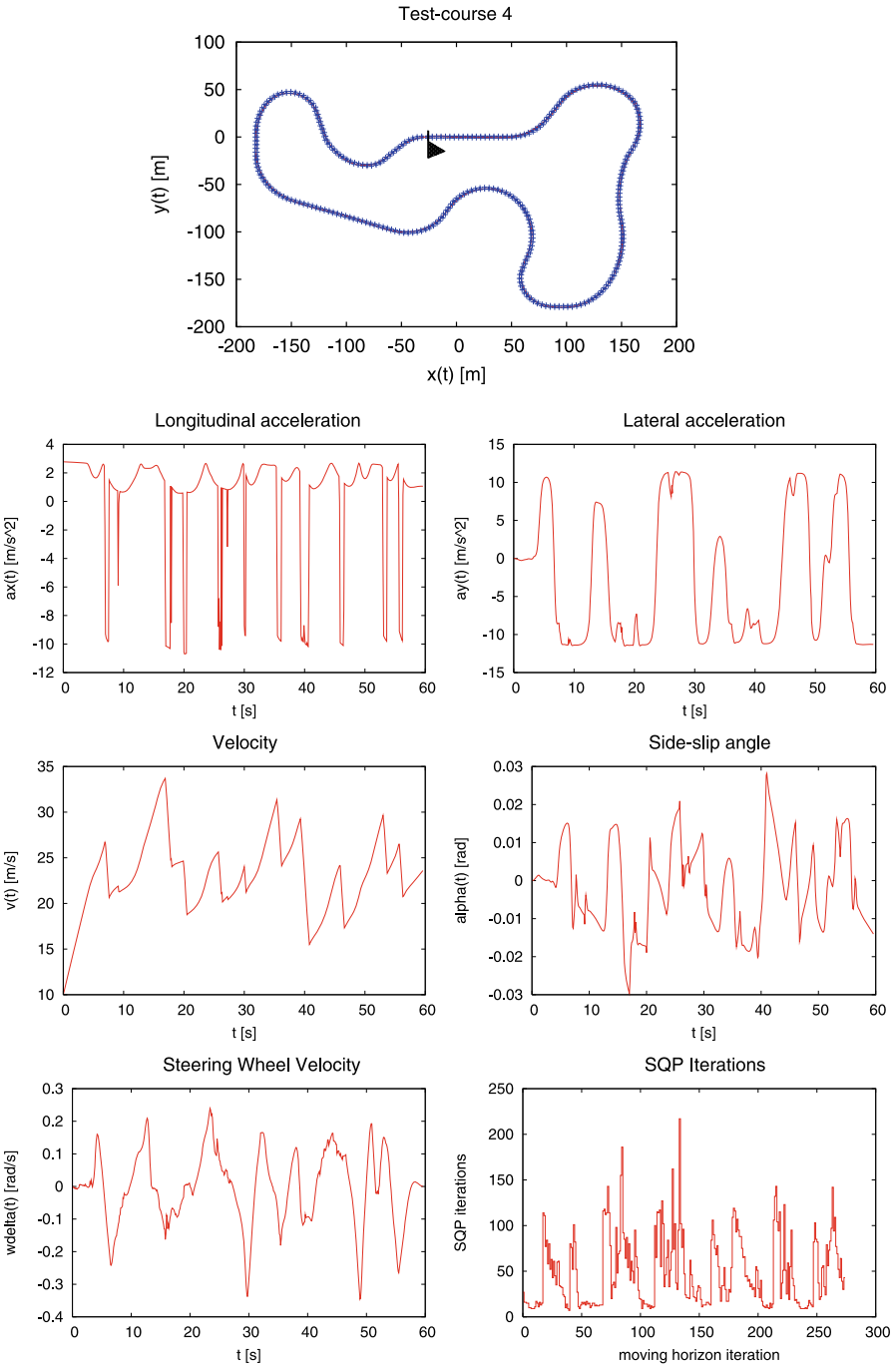

Fig. 11 Test-course 4: $N_r = 289$, $N_\ell = 289$, $s^r_{N_r} = 1154$ [m], $s^\ell_{N_\ell} = 1167$ [m], CPU time 8 min 43.943 s

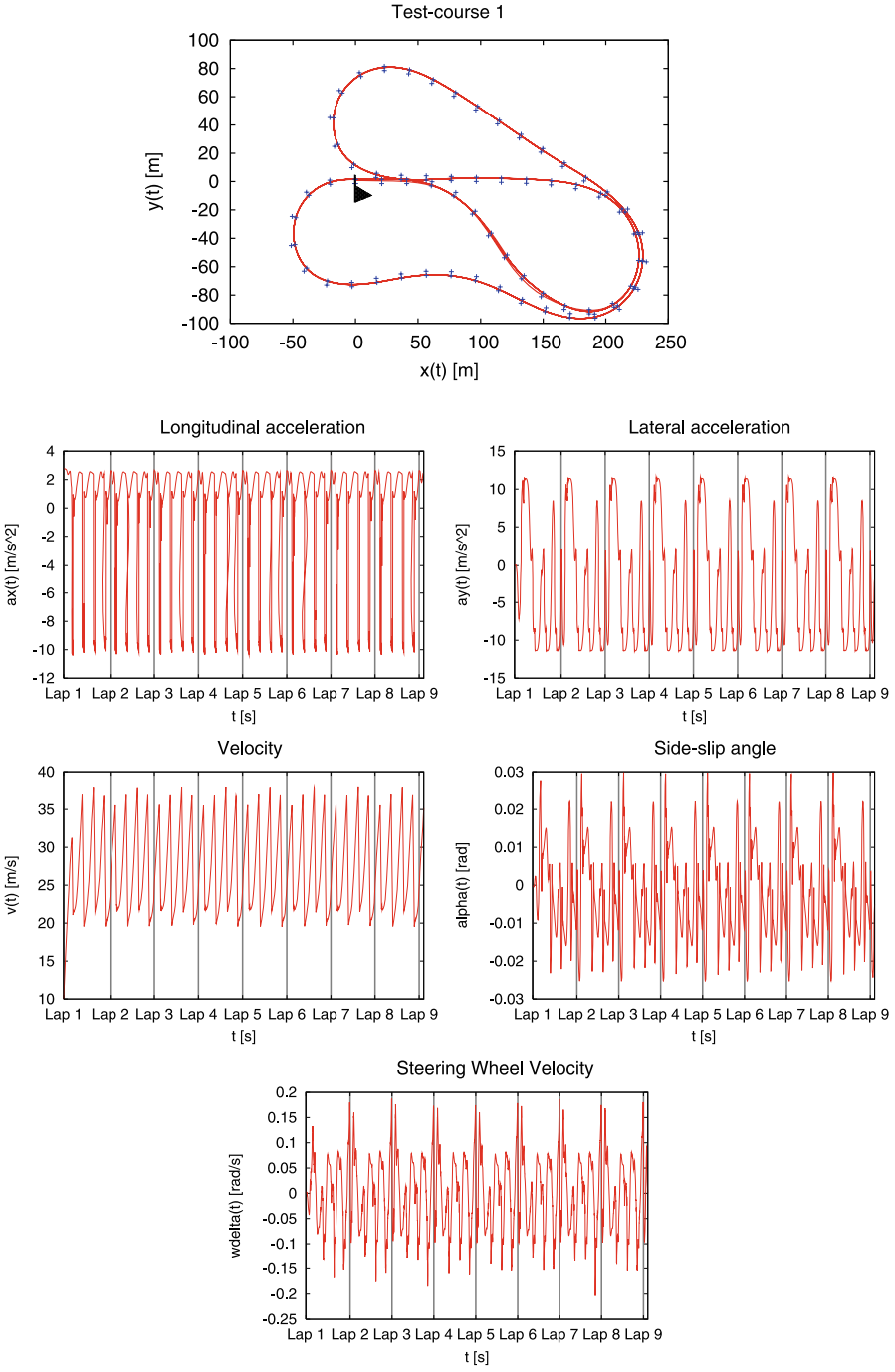

Fig. 12 Test-course 1: Results of the moving horizon algorithm for 8 consecutive laps. All laps except the first lap and the final lap show the same structure. The *vertical lines* indicate the starting time points of a new lap

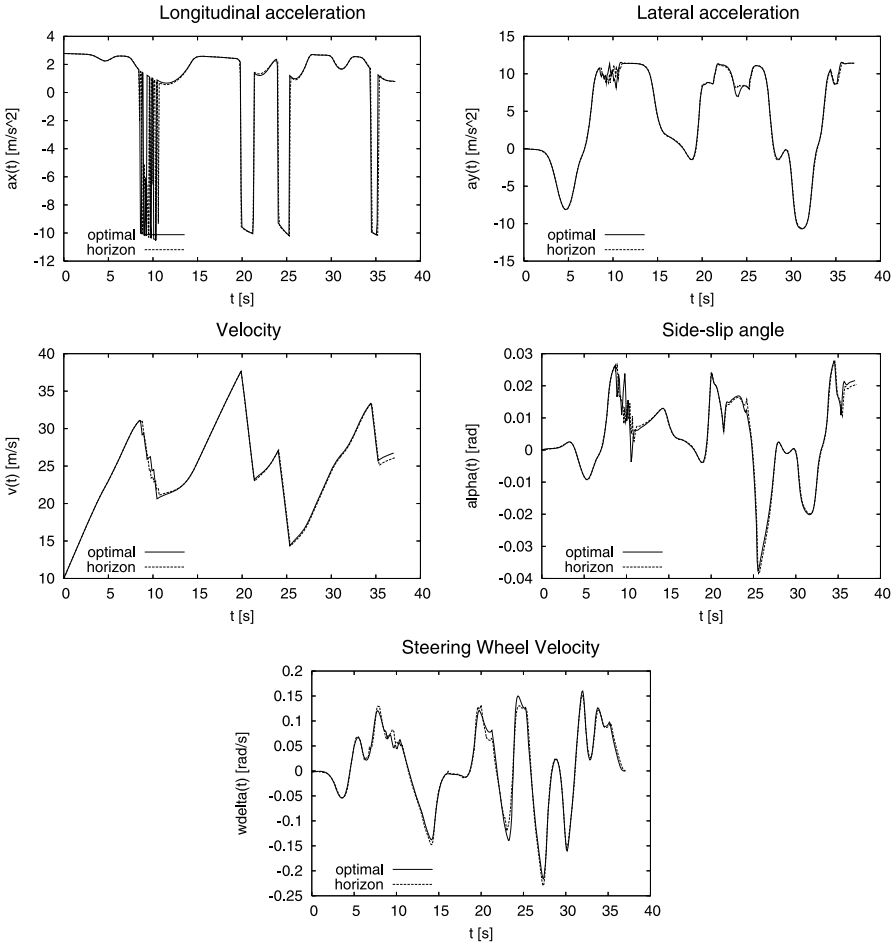

Fig. 13 Test-course 3: Comparison of the optimal solution for the whole test-course with the result of the moving horizon algorithm for $N = 400$

large variations in the lateral dynamics. But it still allows variations in the longitudinal dynamics. As a summary, we conclude that the moving horizon algorithm has the potential of providing a sufficiently good approximation of the optimal solution.

5 Conclusions

The present paper discusses a method which allows to compute locally optimal reference trajectories for an automatically driven car. In this paper we focused particularly on trajectories close to the dynamical limit of the car. These were obtained by maximising the distance covered in a fixed time. Of course, reference trajectories for other objectives, e.g. comfort, could be computed easily by changing the objective function of the optimal control problems. However, trajectories close to the dynamical limit

are particularly useful for the setup of cars because (small) changes in the setup will have a comparatively large impact on the motion. In order to study these effects, it is important that always the same reference trajectory is used without too much influence by the driver. Of course, one could compute an optimal solution for every setup, but optimal solutions tend to hide weaknesses in the setup. So, it is reasonable to use always the same trajectory even for a modified car.

In order to produce even more realistic results, especially close to the dynamical limit, the simple single track car model could be replaced by a more sophisticated car model which takes into account the rolling and pitching behavior of the car. Professional or optimal drivers certainly exploit this behavior. However, such car models usually lead to large systems of ordinary differential equations or even differential-algebraic equations with very many degrees of freedom. Our own experiences with such models show that the computational effort will increase significantly, the robustness of the method will decrease, and the overall computation time for one test-course will not be acceptable anymore for a daily use in a company. Moreover, the controller used for tracking is based on a single track model as well. Summarizing, a very sophisticated car model may lead to improved results, but the drawbacks relativize the benefit.

References

Betts JT (2001) Practical methods for optimal control using nonlinear programming. Advances in design and control, vol 3. SIAM, Philadelphia

Bock HG (1981) Numerical treatment of inverse problems in chemical reaction kinetics. In: Ebert KH, Deuflhard P, Jäger W (eds) Modelling of chemical reaction systems. Springer series in chemical physics, vol 18. Springer, Berlin

Bock HG, Plitt KJ (1984) A multiple shooting algorithm for direct solution of optimal control problems. In: Proceedings of the 9th IFAC worldcongress, Budapest, Hungary

Büskens C (1998) Optimierungsmethoden und sensitivitätsanalyse für optimale steuerprozesse mit steuer- und zustandsbeschränkungen. PhD thesis, Fachbereich Mathematik, Westfälische Wilhems-Universität Münster

Dontchev AL, Hager WW, Malanowski K (2000a) Error bounds for Euler approximation of a state and control constrained optimal control problem. Numer Funct Anal Optim 21(5 & 6):653–682

Dontchev AL, Hager WW, Veliov VM (2000b) Second-order RUnge-KUtta approximations in control constrained optimal control. SIAM J Numer Anal 38(1):202–226

Gerdts M (2003a) Direct shooting method for the numerical solution of higher index dae optimal control problems. J Optim Theory Appl 117(2):267–294

Gerdts M (2003b) A moving horizon technique for the simulation of automobile test-drives. ZAMM 83(3):147–162

Gerdts M (2005) Solving mixed-integer optimal control problems by branch & bound: a case study from automobile test-driving with gear shift. Optim Control Appl Methods 26(1):1–18

Gerdts M (2006) OC-ODE—optimal control of ordinary differential equations: User's guide. Technical report, Department Mathematik, Universität Hamburg

Gill PE, Murray W, Saunders MA (2002) Snopt: an SQP algorithm for large-scale constrained optimization. SIAM J Optim 12(4):979–1006

Gill PE, Murray W, Wright MH (1981) Practical optimization. Academic Press, London

Goh CJ, Teo KL (1988) Control parametrization: a unified approach to optimal control problems with general constraints. Automatica 24:3–18

Grötschel M, Krumke SO, Rambau J (2001) Online optimization of large scale systems. Springer, Berlin

Hager W (1990) Multiplier methods for nonlinear optimal control. SIAM J Numer Anal 27:1061–1080

Hager WW (2000) Runge-Kutta methods in optimal control and the transformed adjoint system. Numer Math 87(2):247–282

Hairer E, Norsett SP, Wanner G (1993) Solving ordinary differential equations I: Nonstiff problems. Springer series in computational mathematics, vol 8, 2nd edn. Springer, Berlin

Jennings LS, Fisher ME, Teo KL, Goh CJ (2004) MISER3 optimal control software version 3: Theory and user manual. Department of Mathematics, The University of Western Australia, Nedlands, Australia. http://www.cado.uwa.edu.au/miser

Malanowski K, Büskens C, Maurer H (1997) Convergence of approximations to nonlinear optimal control problems. In: Fiacco A (ed) Mathematical programming with data perturbations. Lecture notes in pure and applied mathematics, vol 195. Dekker, New York, pp 253–284

Martin R, Teo K (1994) Optimal control of drug administration in cancer chemotherapy. World Scientific, Singapore. xiii, 187 p

Mayr R (1991) Verfahren zur Bahnfolgeregelung für ein automatisch geführtes Fahrzeug. PhD thesis, Fakultät für Elektrotechnik, Universität Dortmund

Mitschke M (1990) Dynamik der Kraftfahrzeuge, Band c: Fahrverhalten, 2nd edn. Springer, Berlin

Moder T (1994) Optimale Steuerung eines KFZ im fahrdynamischen Grenzbereich. Master's thesis, Mathematisches Institut, Technische Universität München

Müller-Beßler B, Stock G, Hoffmann J (2006) Reproducible driving near the stability limit. Technical report, presented at race.tech 2006 in Munich, Volkswagen AG, Wolfsburg, Germany

Neculau M (1992) Modellierung des Fahrverhaltens: Informationsaufnahme, Regel- und Steuerstrategien in Experiment und Simulation. PhD thesis, Fachbereich 12: Verkehrswesen, Technische Universität Berlin

Pacejka H, Bakker E (1993) The magic formula tyre model. Suppl Veh Syst Dyn 21:1–18

Polak E, Yang T, Mayne D (1993) A method of centers based on barrier function methods for solving optimal control problems with continuum state and control constraints. SIAM J Control Optim 31: 159–179

Powell MJD (1978) A fast algorithm for nonlinearily constrained optimization calculation. In: Watson G (ed) Numerical analysis. Lecture notes in mathematics, vol 630. Springer, Berlin

Risse H-J (1991) Das Fahrverhalten bei normaler Fahrzeugführung. VDI Fortschrittberichte Reihe 12: Verkehrstechnik/Fahrzeugtechnik, vol 160. Springer, Verlag

Schittkowski K (1983) On the convergence of a sequential quadratic programming method with an augmented Lagrangean line search function. Math Operationsforsch Stat Ser Optim 14(2):197–216

Stoer J (1985) Principles of sequential quadratic programming methods for solving nonlinear programs. In: Schittkowski K (ed) Computational mathematical programming. NATO ASI series, vol F15. Springer, Berlin, pp 165–207

Teo K, Goh C, Wong K (1991) In: Pitman monographs and surveys in pure and applied mathematics. A unified computational approach to optimal control problems, vol 55. Wiley, New York. ix, 329 p.

Teo KL, Jennings LS, Lee HWJ, Rehbock V (1999) The control parametrization enhancing transform for constrained optimal control problems. J Aust Math Soc Ser B 40:314–335

von Stryk O (1993) Numerical solution of optimal control problems by direct collocation. In: Optimal control. International series of numerical mathematics, vol 111. Birkhäuser, Basel, pp 129–143

Wu CZ, Teo KL (2006) Global impulsive optimal control computation. J Ind Manag Optim 2(4):435–450