

Time-optimal control of automobile test drives with gear shifts

Christian Kirches^{1,*}, Sebastian Sager^{1,2}, Hans Georg Bock¹ and Johannes P. Schlöder¹

¹*Interdisciplinary Center for Scientific Computing (IWR), University of Heidelberg, D-69120 Heidelberg, Germany*

²*IMDEA-Mathematics, Universidad Autónoma de Madrid, E-28049 Madrid, Spain*

SUMMARY

We present a numerical method and results for a recently published benchmark problem (*Optim. Contr. Appl. Met.* 2005; **26**:1–18; *Optim. Contr. Appl. Met.* 2006; **27**(3):169–182) in mixed-integer optimal control. The problem has its origin in automobile test-driving and involves discrete controls for the choice of gears. Our approach is based on a convexification and relaxation of the integer controls constraint. Using the direct multiple shooting method we solve the reformulated benchmark problem for two cases: (a) As proposed in (*Optim. Contr. Appl. Met.* 2005; **26**:1–18), for a fixed, equidistant control discretization grid and (b) As formulated in (*Optim. Contr. Appl. Met.* 2006; **27**(3):169–182), taking into account free switching times. For the first case, we reproduce the results obtained in (*Optim. Contr. Appl. Met.* 2005; **26**:1–18) with a speed-up of several orders of magnitude compared with the Branch&Bound approach applied there (taking into account precision and the different computing environments). For the second case we optimize the switching times and propose to use an initialization based on the solution of (a). Compared with (*Optim. Contr. Appl. Met.* 2006; **27**(3):169–182) we were able to reduce the overall computing time considerably, applying our algorithm. We give theoretical evidence on why our convex reformulation is highly beneficial in the case of time-optimal mixed-integer control problems as the chosen benchmark problem basically is (neglecting a small regularization term). Copyright © 2009 John Wiley & Sons, Ltd.

Received 20 December 2007; Revised 28 November 2008; Accepted 8 April 2009

KEY WORDS: optimal control; direct multiple shooting; mixed-integer nonlinear optimization; mixed-integer optimal control; double-lane change manoeuvre; gear choice

1. INTRODUCTION

Mixed-integer optimal control problems (MIOCPs) in ordinary differential equations (ODEs) have gained increasing interest over the last years, see references [1–16]. This is probably due to the fact that the underlying processes have a high potential for optimization.

Typical examples are the choice of gears in transport, [1, 2, 6, 15] or processes in chemical engineering involving on–off valves, [5, 17].

Although the first MIOCPs, namely the optimization of subway trains that are equipped with discrete acceleration stages, were already solved in the early 80s for the city of New York, [18], the so-called *indirect methods* used there do not seem appropriate for generic large-scale optimal control problems with underlying nonlinear differential algebraic equation systems. Instead *direct methods*, in particular *all-at-once approaches*, [19–21], have become the methods of choice for most practical problems. See [22] for an overview.

*Correspondence to: Christian Kirches, Interdisciplinary Center for Scientific Computing (IWR), University of Heidelberg, Im Neuenheimer Feld 368, D-69120 Heidelberg, Germany.

†E-mail: christian.kirches@iwr.uni-heidelberg.de

Contract/grant sponsor: SIMUMAT Project; contract/grant number: S-050/ESP-0158

In direct methods infinite-dimensional control functions are discretized by basis functions and corresponding finite-dimensional parameters that enter into the optimization problem. The drawback of direct methods with binary control functions obviously is that they lead to high-dimensional vectors of binary variables. For many practical applications a fine control discretization is required, however. Therefore, techniques from mixed-integer nonlinear programming like Branch&Bound or Outer Approximation will work only on limited and small time horizons because of the exponentially growing complexity of the problem, [23].

We propose to use a convexification with respect to the binary controls. The reformulated control problem has two main advantages compared with the standard formulations or convexifications. First, especially for time-optimal control problems, the optimal solution of the relaxed[‡] problem will exhibit a bang–bang structure, and is thus already integer feasible. Second, theoretical results have recently been found, [3, 6], that show that even for path-constrained and sensitivity-seeking arcs the optimal solution of the relaxed problem yields the exact lower bound on the minimum of the integer problem. This allows to calculate the loss of performance, if a coarser control discretization grid, a simplified switching structure for the optimization of switching times or heuristics is used.

This paper is structured as follows. In Section 2 we present the single-track car model and the test course as used by the references [1, 2]. In addition the resulting MIOCP is formulated. In Section 3 we describe the direct multiple shooting method used for the efficient solution of the underlying continuous optimal control problem. Subsequently, recently developed extensions for the treatment of integer controls are discussed, with a special focus on a convexification with respect to the integer controls. Section 4 presents numerical results and compares them to the reference solutions [1, 2]. Section 5 concludes this paper.

[‡]In the following we will use the expression *relaxed* whenever a control constraint $w(t) \in \{0, 1\}$ is relaxed to $w(t) \in [0, 1]$.

2. PROBLEM FORMULATION

In this section we will give a short overview of car model and test course used for the computational study, summing up the description given in [1].

2.1. Car model

We consider a single-track model, derived under the simplifying assumption that rolling and pitching of the car body can be neglected. Consequentially, only a single front and rear wheel is modeled, located in the virtual center of the original two wheels. Motion of the car body is considered on the horizontal plane only.

Four controls represent the driver's choice on steering and velocity, and are listed in Table I. We denote with w_δ the steering wheel's angular velocity. The force F_B controls the total braking force, while the accelerator pedal position ϕ is translated into an accelerating force according to the torque model presented in (Equation (13)). Finally, the selected gear μ influences the effective engine torque's transmission ratio.

The single-track dynamics are described by a system of ODEs. The individual system states are listed in Table II. Figure 1 visualizes the choice of coordinates, angles, and forces.

The center of gravity is denoted by the coordinate pair (c_x, c_y) , which is obtained by integration over the directional velocity,

$$\dot{c}_x(t) = v(t) \cos(\psi(t) - \beta(t)) \quad (1)$$

$$\dot{c}_y(t) = v(t) \sin(\psi(t) - \beta(t)) \quad (2)$$

Acceleration is obtained from the sum of forces attacking the car's mass m in the direction of driving,

$$\begin{aligned} \dot{v}(t) = & \frac{1}{m} ((F_{lr}^\mu - F_{Ax}) \cos \beta(t) \\ & + F_{lf} \cos(\delta(t) + \beta(t)) \\ & - (F_{sr} - F_{Ay}) \sin \beta(t) \\ & - F_{sf} \sin(\delta(t) + \beta(t))) \end{aligned} \quad (3)$$

Table I. Controls used in the car model.

Control	Range	Unit	Description
w_δ	$[-0.5, 0.5]$	$\frac{\text{rad}}{\text{s}}$	Steering wheel angular velocity
F_B	$[0, 1.5 \times 10^4]$	N	Total braking force
ϕ	$[0, 1]$	—	Accelerator pedal position
μ	$\{1, \dots, 5\}$	—	Selected gear

Table II. Coordinates and states used in the car model.

State	Unit	Description
c_x	m	Horizontal position of the car
c_y	m	Vertical position of the car
v	$\frac{\text{m}}{\text{s}}$	Magnitude of directional velocity of the car
δ	rad	Steering wheel angle
β	rad	Side slip angle
ψ	rad	Yaw angle
w_z	$\frac{\text{rad}}{\text{s}}$	Yaw angle velocity

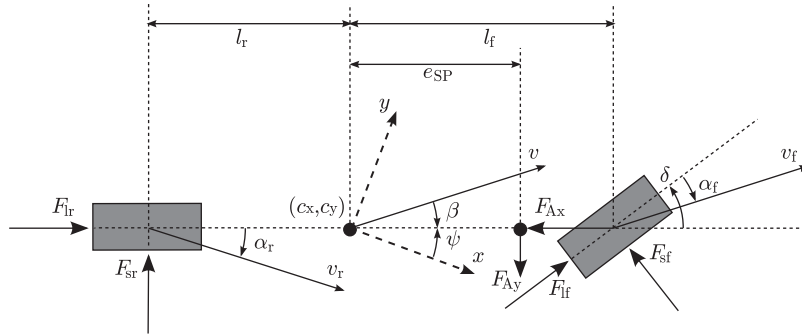


Figure 1. Coordinates and forces in the single-track car model. The figure aligns with the vehicle's local coordinate system while dashed vectors denote the earth-fixed coordinate system chosen for computations.

The steering wheel's angle is obtained from the corresponding controlled angular velocity,

$$\dot{\delta}(t) = w_\delta \quad (4)$$

The slip angle's change is controlled by the steering wheel and counteracted by the sum of forces attacking perpendicular to the car's direction of driving. The forces' definitions are given in (Equation (8)).

$$\dot{\beta}(t) = w_z(t) - \frac{1}{mv(t)} ((F_{lr} - F_{Ax}) \sin \beta(t) + F_{lr} \sin(\delta(t) + \beta(t)))$$

$$\begin{aligned} &+ (F_{sr} - F_{Ay}) \cos \beta(t) \\ &+ F_{sf} \cos(\delta(t) + \beta(t)) \end{aligned} \quad (5)$$

The yaw angle is obtained by integrating over its change w_z ,

$$\dot{\psi}(t) = w_z(t) \quad (6)$$

which in turn is the integral over the sum of forces attacking the front wheel in direction perpendicular to

the car's longitudinal axis of orientation,

$$\dot{w}_z(t) = \frac{1}{I_{zz}}(F_{sf} l_f \cos \delta(t) - F_{sr} l_{sr} - F_{Ay} e_{SP} + F_{lf} l_f \sin \delta(t)) \quad (7)$$

We now list and explain the individual forces used in this ODE system. We first discuss lateral and longitudinal forces attacking at the front and rear wheels. In view of the convex reformulation we will undertake later; we consider the gear μ to be fixed and denote dependencies on the selected gear by a superscript μ like, e.g., in w_{mot}^μ .

The side (lateral) forces on the front and rear wheels as functions of the slip angles α_f and α_r according to the so-called 'magic formula' due to [24] are

$$F_{sf, sr}(\alpha_{f,r}) := D_{f,r} \sin(C_{f,r} \arctan(B_{f,r} \alpha_{f,r} - E_{f,r}(B_{f,r} \alpha_{f,r} - \arctan(B_{f,r} \alpha_{f,r})))) \quad (8)$$

The front slip angle itself is obtained from

$$\alpha_f := \delta(t) - \arctan\left(\frac{l_f \dot{\psi}(t) - v(t) \sin \beta(t)}{v(t) \cos \beta(t)}\right) \quad (9)$$

while the rear slip angle is

$$\alpha_r := \arctan\left(\frac{l_r \dot{\psi}(t) + v(t) \sin \beta(t)}{v(t) \cos \beta(t)}\right) \quad (10)$$

The longitudinal force at the front wheel is composed from braking force F_{Bf} and resistance due to rolling friction F_{Rf}

$$F_{lf} := -F_{Bf} - F_{Rf} \quad (11)$$

Assuming a rear wheel drive, the longitudinal force at the rear wheel is given by the transmitted engine torque M_{wheel} and reduced by braking force F_{Br} and rolling friction F_{Rr} . The effective engine torque M_{mot}^μ is transmitted twice. We denote by i_g^μ the gearbox transmission ratio corresponding to the selected gear μ , and by i_t the axle drive's fixed transmission ratio. R is the rear wheel radius.

$$F_{lr}^\mu := \frac{i_g^\mu i_t}{R} M_{\text{mot}}^\mu(\phi) - F_{Br} - F_{Rr} \quad (12)$$

The engine's torque, depending on the acceleration pedal's position ϕ , is modeled as follows:

$$M_{\text{mot}}^\mu(\phi) := f_1(\phi) f_2(w_{\text{mot}}^\mu) + (1 - f_1(\phi)) f_3(w_{\text{mot}}^\mu) \quad (13)$$

$$f_1(\phi) := 1 - \exp(-3\phi) \quad (14)$$

$$f_2(w_{\text{mot}}) := -37.8 + 1.54w_{\text{mot}} - 0.0019w_{\text{mot}}^2 \quad (15)$$

$$f_3(w_{\text{mot}}) := -34.9 - 0.04775w_{\text{mot}} \quad (16)$$

Here, w_{mot}^μ is the engine's rotary frequency in Hertz. For a given gear μ it is computed from

$$w_{\text{mot}}^\mu := \frac{i_g^\mu i_t}{R} v(t) \quad (17)$$

The total braking force F_B is controlled by the driver. The distribution to front and rear wheels is

$$F_{Bf} := \frac{2}{3} F_B, \quad F_{Br} := \frac{1}{3} F_B \quad (18)$$

The braking forces F_{Rf} and F_{Rr} due to rolling resistance are obtained from

$$F_{Rf}(v) := f_R(v) \frac{m l_f g}{l_f + l_r}, \quad F_{Rr}(v) := f_R(v) \frac{m l_f g}{l_f + l_r} \quad (19)$$

where the velocity-dependent amount of friction is modeled by

$$f_R(v) := 9 \times 10^{-3} + 7.2 \times 10^{-5} v + 5.038848 \times 10^{-10} v^4 \quad (20)$$

Finally, drag force due to air resistance is given by F_{Ax} , while we assume that no sideward drag forces (e.g. side wind) are present.

$$F_{Ax} := \frac{1}{2} c_w \rho A v^2(t), \quad F_{Ay} := 0 \quad (21)$$

The model parameters $m, g, l_f, l_r, e_{SP}, R, I_{zz}, c_w, \rho, A, i_t$, and i_g^μ and the Pacejka coefficients $B_{f,r}, C_{f,r}, D_{f,r}, E_{f,r}$ can be found in [1, 2] and on the website [25].

2.2. Test course

The double-lane change manoeuvre presented in [1] is realized by constraining the car's position onto a prescribed track at any time $t \in [t_0, t_f]$, see Figure 2. Starting in the left position with an initial prescribed velocity, the driver is asked to manage a change of lanes

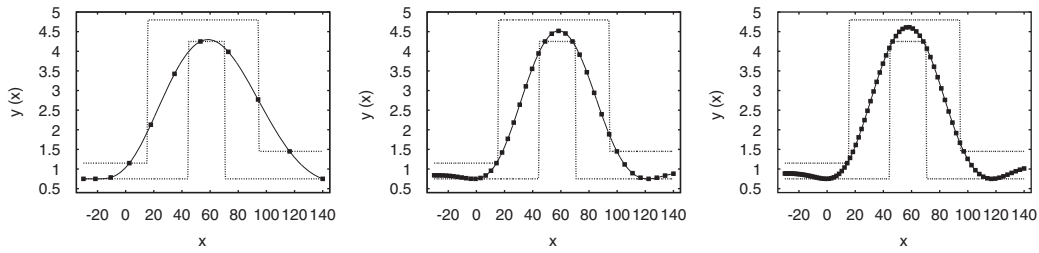


Figure 2. Track with path constraints (22d) on the vertical position for $N_{\text{shoot}} = 10, 40, 80$, from left to right. Squares show evaluation points equidistant in time.

modeled by an offset of 3.5 m in the track. Afterwards he is asked to return to the starting lane. This manoeuvre can be regarded as an overtaking move or as an evasive action taken to avoid hitting an obstacle suddenly appearing on the starting lane. A description of the course can be found in [1, 2, 25].

2.3. Optimal control problem

We denote with x the state vector of the ODE system and by f the corresponding right-hand side function. The vector u shall be the vector of continuous controls, whereas the integer control $\mu(\cdot)$ will be written in a separate vector,

$$x := (c_x, c_y, v, \delta, \beta, \psi, w_z)^\top, \quad u := (w_\delta, F_B, \phi)^\top$$

With this notation, the resulting MIOCP reads as

$$\min_{t_f, x(\cdot), u(\cdot), \mu(\cdot)} \quad t_f + \int_0^{t_f} w_\delta^2(t) dt \quad (22a)$$

$$\text{s.t.} \quad \dot{x}(t) = f(t, x(t), u(t), \mu(t)) \quad (22b)$$

$$\forall t \in [t_0, t_f]$$

$$c_y(t) \in \left[P_l(c_x(t)) + \frac{B}{2}, P_u(c_x(t)) - \frac{B}{2} \right] \quad (22c)$$

$$\forall t \in [t_0, t_f]$$

$$w_\delta(t) \in [-0.5, 0.5] \quad \forall t \in [t_0, t_f] \quad (22d)$$

$$F_B(t) \in [0, 1.5 \times 10^4] \quad \forall t \in [t_0, t_f] \quad (22e)$$

$$\phi(t) \in [0, 1] \quad \forall t \in [t_0, t_f] \quad (22f)$$

$$\mu(t) \in \{1, \dots, 5\} \quad \forall t \in [t_0, t_f] \quad (22g)$$

$$x(t_0) = (-30, \text{free}, 10, 0, 0, 0, 0)^\top \quad (22h)$$

$$c_x(t_f) = 140 \quad (22i)$$

$$\psi(t_f) = 0 \quad (22j)$$

By employing the objective function (Equation (22a)) we strive to minimize the total time t_f required to traverse the test course, and to do so with minimal steering effort $w_\delta(t)$. At any time, the car must be positioned within the test course's boundaries; this requirement is formulated by the double inequality path constraint (Equation (22c)). The system's initial values are fixed in (Equation (22h)) with the exception of the car's initial vertical position on the track, which remains a free variable only constrained by the track's boundary. Finally, constraints (Equation (22i), (22j)) guarantee that the car actually arrives at the end of the test course driving straight ahead.

3. GENERAL PROBLEM CLASS AND ALGORITHM

In this section we will abstract the control problem to a more general class and propose algorithms for the solution. In Section 3.2 we sum up the direct multiple shooting method that we use to solve optimal control problems without integer constraints. In Section 3.3 we present a reformulation strategy that will partly convexify the MIOCP. In Section 3.4 we discuss methods to come up with integer solutions. In Section 3.5 we present the switching time optimization approach.

3.1. General problem class

The MIOCP formulated in Section 2.3 belongs to a broader class of equality- and inequality-constrained optimal control problems on dynamic processes modeled by ODE systems. We consider the following class of optimal control problems:

$$\min_{\substack{t_f, p, \\ x(\cdot), u(\cdot), \mu(\cdot)}} M(t_f, x(t_f), p) \quad (23a)$$

$$\text{s.t. } \dot{x}(t) = f(t, x(t), u(t), \mu(t), p) \quad \forall t \in \mathcal{T} \quad (23b)$$

$$0 \leq c(t, x(t), u(t), p) \quad \forall t \in \mathcal{T} \quad (23c)$$

$$0 \leq r^{\text{in}}(x(t_1^{\text{in}}), \dots, x(t_{N_{\text{in}}}^{\text{in}}), p) \quad (23d)$$

$$0 = r^{\text{eq}}(x(t_1^{\text{eq}}), \dots, x(t_{N_{\text{eq}}}^{\text{eq}}), p) \quad (23e)$$

$$\mu(t) \in \Omega \quad \forall t \in \mathcal{T} \quad (23f)$$

Herein, let $t \in [t_0, t_f] =: \mathcal{T} \subset \mathbb{R}$ be a fixed time horizon, and let $x(t) \in \mathbb{R}^{n_x}$ describe the state vector of the dynamic process at any time $t \in \mathcal{T}$. Further, let $u(t) \in \mathbb{R}^{n_u}$ be the vector of continuous controls influencing the dynamic process, and let $\mu(t) \in \mathbb{R}^{n_\mu}$ be a vector of integer control functions, constrained to values from a discrete set Ω . Finally we denote by $p \in \mathbb{R}^{n_p}$ a vector of time-independent model parameters. Point inequalities and equalities are defined on suited time grids $\{t_i^{\text{in}}\}$ and $\{t_i^{\text{eq}}\}$.

We require the objective function $M: \mathcal{T} \times \mathbb{R}^{n_x} \rightarrow \mathbb{R}$, the path constraint function $c: \mathcal{T} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_p} \rightarrow \mathbb{R}^{n_c}$, and the equality as well as the inequality point constraint functions $r^{\text{eq}}: (\mathcal{T} \times \mathbb{R}^{n_x})^{N_{\text{eq}}} \times \mathbb{R}^{n_p} \rightarrow \mathbb{R}^{n_{\text{eq}}}$ and $r^{\text{in}}: (\mathcal{T} \times \mathbb{R}^{n_x})^{N_{\text{in}}} \times \mathbb{R}^{n_p} \rightarrow \mathbb{R}^{n_{\text{in}}}$ to be sufficiently often continuously differentiable with respect to all arguments. The same shall hold true for the ODE system's right-hand side function $f: \mathcal{T} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_\mu} \times \mathbb{R}^{n_p} \rightarrow \mathbb{R}^{n_x}$.

3.2. The direct multiple shooting method

This section briefly sketches the direct multiple shooting method, first described by Plitt [19] and Bock and Plitt [20] and extended in a series of subsequent works (see, e.g. [26]). With the optimal control software

package MUSCOD-II [27], an efficient implementation of this method is available.

The purpose of this method is to transform the infinite-dimensional OCP presented in Section 3.1 (neglecting the integer variables) into a finite-dimensional nonlinear program (NLP) by discretization of the control functions on a time grid $t_0 < t_1 < \dots < t_{N_{\text{shoot}}} = t_f$. For this, let $b_{ij}: \mathcal{T} \rightarrow \mathbb{R}^{n_u}$, $1 \leq j \leq n_{q_i}$ be a set of sufficiently often continuously differentiable base function of the control discretization for the shooting interval $[t_i, t_{i+1}] \subset \mathcal{T}$. Further, let $q_i \in \mathbb{R}^{n_{q_i}}$ be the corresponding set of control parameters, and define

$$\hat{u}_i(t, q_i) := \sum_{j=1}^{n_{q_i}} q_{ij} b_{ij}(t), \quad t \in [t_i, t_{i+1}] \quad (24)$$

$$0 \leq i < N_{\text{shoot}}$$

The control space is hence reduced to functions that can be written as in (24), depending on finitely many parameters q_i . The right-hand side function f and the constraint functions c, r^{eq} , and r^{in} are assumed to be adapted accordingly. In the simplest case we have $n_{q_i} = 1$ and $b_{i1}(t) = q_{i1}$, thereby realizing a piecewise constant control discretization. More elaborate discretizations such as linear or cubic base functions are easily found. Multiple shooting variables s_i are introduced on the time grid to parameterize the differential states. The node values serve as initial values for an ODE solver computing the state trajectories independently on the shooting intervals.

$$\dot{x}_i(t) = f(t, x_i(t), \hat{u}_i(t, q_i), p) \quad \forall t \in [t_i, t_{i+1}] \quad (25a)$$

$$0 \leq i < N_{\text{shoot}}$$

$$x_i(t_i) = s_i, \quad 0 \leq i \leq N_{\text{shoot}} \quad (25b)$$

One advantage of the multiple shooting approach is the ability to use state-of-the-art adaptive integrator methods, see, e.g. [28–30]. Obviously we obtain from the above IVPs N_{shoot} trajectories, which in general will not combine to a single continuous trajectory. Thus, continuity across shooting intervals needs to be ensured by additional matching conditions entering the NLP as equality constraints,

$$s_{i+1} = x_i(t_{i+1}; s_i, q_i, p), \quad 0 \leq i < N_{\text{shoot}} \quad (26)$$

Here we denote by $x_i(t_{i+1}; s_i, q_i, p)$ the solution of the IVP on shooting interval i , evaluated in t_{i+1} , and depending on the initial values s_i , control parameters q_i , and model parameters p .

The path constraints $c(\cdot)$ are discretized on an appropriately chosen grid. This discretization may have a strong effect on the solution of the problem, as is the case for the problem at hand. We will come back to this point in Section 4.2. To ease the notation, we assume in the following that all constraint grids match the shooting grid.

From this discretization and parameterization results a highly structured NLP of the form

$$\min_{\xi} M(s_{N_{\text{shoot}}}, p) \quad (27a)$$

$$\text{s.t.} \quad 0 = s_{i+1} - x_i(t_{i+1}; s_i, q_i, p) \\ 0 \leq i < N_{\text{shoot}} \quad (27b)$$

$$0 \leq c(t_i, s_i, \hat{u}_i(t_i, q_i), p) \\ 0 \leq i \leq N_{\text{shoot}} \quad (27c)$$

$$0 \leq r^{\text{in}}(s_0, s_1, \dots, s_{N_{\text{shoot}}}, p) \quad (27d)$$

$$0 = r^{\text{eq}}(s_0, s_1, \dots, s_{N_{\text{shoot}}}, p) \quad (27e)$$

where the vector ξ shall contain all unknowns of the problem

$$\xi = (s_0, \dots, s_{N_{\text{shoot}}}, q_0, \dots, q_{N_{\text{shoot}}-1}, p)$$

For the ease of notation in (27c), we write $\hat{u}_{N_{\text{shoot}}}(t_{N_{\text{shoot}}}, q_{N_{\text{shoot}}}) := \hat{u}_{N_{\text{shoot}}-1}(t_{N_{\text{shoot}}}, q_{N_{\text{shoot}}-1})$. We solve this large-scale, but structured NLP by a tailored sequential quadratic programming (SQP) method. This includes an extensive exploitation of the arising structures, in particular using block-wise high-rank updates and condensing for a reduction of the size of the quadratic problems to that of a single-shooting method. For more details see [20, 26].

3.3. Convex relaxation of integer controls

We convexify problem (26) with respect to the integer control functions $\mu(\cdot)$ as first suggested in [3]. We assign one control function $w_i(\cdot)$ to every possible control $\mu^i \in \Omega$. This corresponds to $n_w = |\Omega|$ controls, which may be a large number. In practice, however,

there often is a small set of admissible choices leading to logical exclusion of most of the elements of Ω . Here n_w would correspond to the number of remaining feasible choices. Examples are the selection of a distillation column tray [3], an inlet stream port [5], or a gear in the presented case. In all examples n_w is linear in the number of choices. Furthermore, in most practical applications the binary control functions enter linearly (such as valves that indicate whether a certain term is present or not). Therefore, the drawback of an increased number of control functions is outweighed by the advantages concerning the avoidance of integer variables associated with the discretization in time for most applications we know of. By convexifying (26) with respect to $\mu(\cdot)$, we obtain the following optimal control problem:

$$\min_{t_f, p, x(\cdot), u(\cdot), w(\cdot)} M(t_f, x(t_f), p) \quad (28a)$$

$$\text{s.t.} \quad \dot{x}(t) = \sum_{i=1}^{n_w} f(x(t), \mu^i, u(t), p) w_i(t) \\ \forall t \in \mathcal{T} \quad (28b)$$

$$0 \leq c(t, x(t), u(t), p) \quad \forall t \in \mathcal{T} \quad (28c)$$

$$0 \leq r^{\text{in}}(x(t_1^{\text{in}}), \dots, x(t_{N_{\text{in}}}^{\text{in}}), p) \quad (28d)$$

$$0 = r^{\text{eq}}(x(t_1^{\text{eq}}), \dots, x(t_{N_{\text{eq}}}^{\text{eq}}), p) \quad (28e)$$

$$w(t) \in \{0, 1\}^{n_w} \quad \forall t \in \mathcal{T} \quad (28f)$$

$$1 = \sum_{i=1}^{n_w} w_i(t) \quad \forall t \in \mathcal{T} \quad (28g)$$

There obviously is a bijection $\mu(t) = \mu^i \leftrightarrow w_i(t) = 1$ between the solutions of problems (26) and (32), compare [3]. The relaxation of problem (32) consists in replacing constraint (28f) by

$$w(t) \in [0, 1]^{n_w} \quad \forall t \in \mathcal{T} \quad (29)$$

This formulation has two main advantages. First, for many optimal control problems the optimal solution will have a bang–bang character; therefore, the solution of the relaxed problem will yield the optimal integer solution. Second, for problems that fit into the class (32)

a theory has been developed that allows to deduce information on the optimal integer solution from the optimal value of the relaxed problem, even if this solution is not bang–bang, but path-constrained or sensitivity-seeking. See [3, 5] for theory and applications.

3.4. Calculation of integer solutions

Different methods for the calculation of integer solutions for MIOCPs, based on a direct approach, have been described and compared in [3]. Among them one finds Branch&Bound, Outer Approximation, penalization heuristics, and rounding strategies. All methods that suffer from a combinatorial explosion when the number of discretized binary control variables increases have a very limited applicability, though.

It can often be observed that the solution of the relaxed, purely continuous problem already yields an integer solution for almost all control discretizations. In addition, simple rounding strategies, taking the special ordered set constraint (28g) into account, often result in integer solutions without affecting the objective function value.

For cases in which path constraints play a role or a different objective function leads to sensitivity-seeking arcs; we recommend to use a sum up rounding strategy as developed in [3, 6] in combination with a switching time optimization approach to be discussed in Section 3.5. Sum up rounding yields integer solutions arbitrarily close to the optimal integer solution, if a *sufficiently fine* time discretization is used. If guaranteed global solutions are an issue, this approach can be readily combined with methods in global optimization, of course.

3.5. Switching time optimization

A different approach to solve problem (26) is motivated by the idea to optimize the switching times and to take the values of the integer controls $\mu(\cdot)$ fixed on given intervals, as is done, e.g. for bang–bang arcs in indirect methods. Let us consider the one-dimensional binary case; that is, we have a binary control function $w(\cdot) \in \{0, 1\}$. Instead of the control $w(\cdot) : [t_0, t_f] \mapsto \{0, 1\}$ we do get N_{switch} fixed constant control functions

$$w_k : [\tilde{t}_k, \tilde{t}_{k+1}] \mapsto \{0, 1\}$$

defined w.l.o.g. by

$$w_k(t) = \begin{cases} 0 & \text{if } k \text{ even,} \\ 1 & \text{if } k \text{ odd,} \end{cases} \quad t \in [\tilde{t}_k, \tilde{t}_{k+1}] \quad (30)$$

with $0 \leq k \leq N_{\text{switch}} - 1$ and $t_0 = \tilde{t}_0 \leq \tilde{t}_1 \leq \dots \leq \tilde{t}_{N_{\text{switch}}} = t_f$.

If we assume that an optimal binary control function $w(\cdot)$ switches only finitely often, then the original problem is equivalent to optimizing N_{switch} and the time vector \tilde{t} in a multistage formulation with all $w_k(t)$ is fixed to either 0 or 1 and with the additional constraint

$$\sum_{k=0}^{N_{\text{switch}}-1} h_k = t_f - t_0 \quad (31)$$

for positive $h_k \geq 0$. In practice we do not optimize the switching points, but the scaled vector h of model stage lengths $h_k := \tilde{t}_{k+1} - \tilde{t}_k$, see, e.g., [2, 31]. This approach is visualized in Figure 3 for $N_{\text{switch}} = 5$.

For fixed values of N_{switch} we have an optimal control problem in which the stage lengths h_k take the role of time-independent variables that have to be determined. The approach can be extended in a straightforward way to an n_w -dimensional binary control function $w(\cdot)$. Instead of (30) one defines w_k as

$$w_k(t) = w^i \quad \text{if } k = j 2^{n_w} + i - 1, \quad t \in [\tilde{t}_k, \tilde{t}_{k+1}] \quad (32)$$

for some $j \geq 0$ and some $1 \leq i \leq 2^{n_w}$. The values w^i enumerate all 2^{n_w} possible assignments of $w(\cdot) \in \{0, 1\}^{n_w}$, compare Section 3.3.

A closer look at (32) shows some intrinsic problems of the switching time approach. First, the number of model stages grows exponentially not only in the number of control functions, but also grows linearly in the number of expected switches of the binary control functions. Starting from a given number of stages, allowing a small change in one of the control functions requires an additional 2^{n_w} stages. If it is indeed exactly one function $w_i(\cdot)$ that changes while all others stay fixed, $2^{n_w} - 1$ of the newly introduced stages will have length 0.

This leads to a second drawback, namely a nonregular situation that may occur when stage lengths are reduced to zero. Assume the length of an intermediate stage, say h_2 , has been reduced to zero by the optimizer. Then the sensitivity of the optimal control problem with

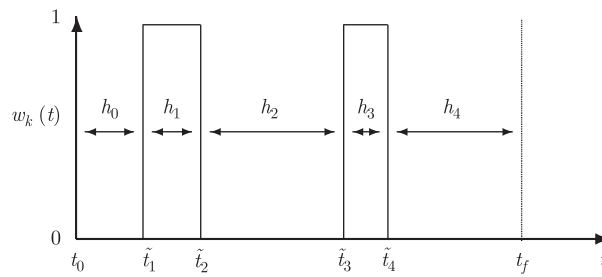


Figure 3. Switching time optimization, one-dimensional example with $N_{\text{switch}} = 5$.

respect to h_1 and h_3 is given by the value of their sum $h_1 + h_3$ only. Thus special care has to be taken to treat the case when stages vanish during the optimization procedure. In [32–34] an algorithm to eliminate such stages is proposed. Although this is possible, still the stage cannot be reinserted, as the time when to reinsert it is undetermined.

The third drawback is that the number of switches is typically not known, let alone the precise switching structure. Some authors propose to iterate on N_{switch} until there is no further decrease in the objective function of the corresponding optimal solution, cf. [32–34]. But it should be stressed that this can only be applied to more complex systems if initial values for the location of the switching points close to the optimum are available, as they are essential for the convergence behavior of the underlying method.

This is closely connected to the fourth and most important drawback of the switching time approach. The reformulation yields additional nonconvexities in the optimization space. Even if the optimization problem is convex in the optimization variables resulting from a constant discretization of the control function $w(\cdot)$, the reformulated problem may be nonconvex.

The mentioned drawbacks of the switching time optimization approach can be overcome, though, if it is combined with an assortment of other concepts, compare [3]. This includes rigorous lower and upper bounds, a strategy to deal with diminishing stage lengths, and a direct all-at-once approach like direct multiple shooting that helps when dealing with nonconvexities as discussed in [3]. In a general setting, good

initial guesses for all optimization variables are of utmost importance, as the reformulation is known to be nonconvex. In this paper we will apply the switching time optimization approach, starting from a near-optimal initialization for all optimization variables, in particular for the stage lengths, that are deduced from the optimal solution obtained for a fixed control discretization grid, using the techniques described above.

3.6. Algorithmic aspects of the benchmark problem

Considering the convex relaxation for the test drive benchmark problem, we will replace the question which gear $\mu(t)$ to choose at every instant t in time by the five questions whether or not to choose gear number i . Corresponding relaxed binary controls $w_i(t) \in [0, 1]$ are introduced. It turns out that the two potential main advantages of this approach mentioned earlier in this paper can be fully exploited for the case of the benchmark problem at hand, as will be shown in Section 4.

Concerning rounding strategies for the reconstruction of integer solutions from the relaxed ones, it turned out that the solution of the relaxed, purely continuous problem already yields an integer solution for almost all control discretizations. In the cases when it does not, a simple rounding strategy (taking the special ordered set constraint (28g) into account) resulted in the same objective function value as the noninteger solution.

Note that problem (32) may still be a nonconvex problem, as the convexification is only done with respect to the integer controls, but not to other variables. In our calculations we use a local, gradient-based optimization technique for the relaxed problem. We assume that we do thus find the global optimum. Indeed,

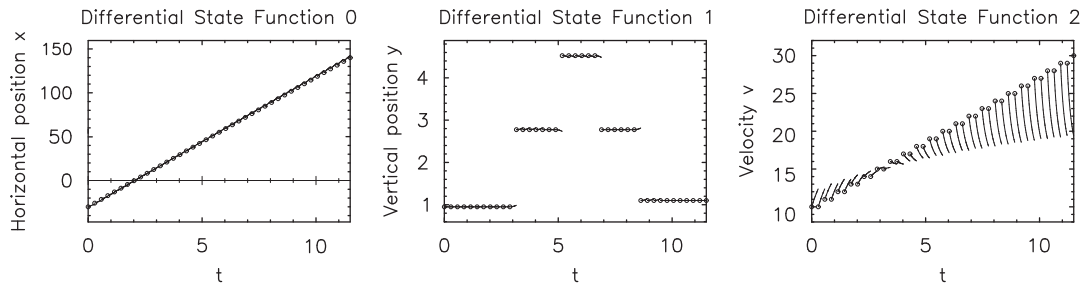


Figure 4. Initialization of the shooting variables chosen to start the solution process.

different start values for both controls and differential states lead to the same results, which indicates that the presented solutions are indeed global ones. As investigated in [3], the multiple shooting approach with a reasonable initialization of the state variables may help to overcome problems of local minima. Note, however, that the same assumption on global optimality is made in the reference paper for the optimal control problems solved in the tree nodes of the Branch&Bound approach.

4. NUMERICAL RESULTS

In this section we will present numerical results for the application of the presented algorithm to the test drive problem. We start with a short summary of the initialization of the optimization variables, an important issue if run times are compared. In Section 4.2 we present the obtained solutions on different fixed grids. In Section 4.3 we allow for an optimization of the switching times for a prescribed switching structure. In Section 4.4 we compare our results and computing times to those given for this benchmark problem in [1] resp. [2], and in Section 4.5 we analyze why our convexification-based algorithm is so much faster.

4.1. Variable initialization

The variables ξ introduced in Section 3.2 have to be initialized within the iterative optimization procedure. The direct multiple shooting method allows to supply additional information on the state vector by way of the

multiple shooting variables, which is typically much better known than the controls that cause this dynamic behavior. Figure 4 shows the initial shooting node values of $(c_x, c_y, v)(t)$ for $N_{\text{shoot}} = 40$ that were chosen to start the multiple shooting method. Clearly, this choice is very easy to come up with. The initial guess for the end time is made as $t_f = 11.5$ s, far away from the actual solution. We initialize $w_\delta \equiv 0$, $F_B \equiv 0$, $\phi \equiv 1$, and $\mu \equiv 1$, as done in the reference papers.

If the shooting node values are determined by integration with given values for the controls $u(\cdot)$ and $w(\cdot)$, the number of iterations needed to obtain convergence typically *increases* slightly as compared with the presented initialization. For unphysical initializations of the control variables this procedure may fail, however.

4.2. Solutions

In the following sections we present numerical calculations that have been computed with the optimal control software package MUSCOD-II [27]. All computations were performed on an AMD Athlon XP 3000+ with 2.166 GHz and 1024 MB of RAM, running SuSE Linux 10.2.

For the solution of the ODE system, an adaptive fourth/fifth-order Runge–Kutta–Fehlberg method equipped with internal numerical differentiation (IND), as described in [28, 29], was used. An integration tolerance of 10^{-9} was used. All subproblems were solved to a KKT tolerance of 10^{-8} for the SQP algorithm described in Section 3.

We convexify condition (22h) as described in Section 3.3, and relax the decisions $w_i(t) \in \{0, 1\}$ to

Table III. Gear choice depending on N_{shoot} .

N_{shoot}	$\mu=1$	$\mu=2$	$\mu=3$	$\mu=4$	$\mu=5$	t_f
10	0.0	0.679839	2.719356	—	—	6.798389
20	0.0	0.338952	2.711614	6.440083	—	6.779035
40	0.0	0.509004	2.714692	6.617062	—	6.786730
80	0.0	0.424345	2.800674	6.619775	—	6.789513

Time points when gear becomes active. No dropping of gears occurs.

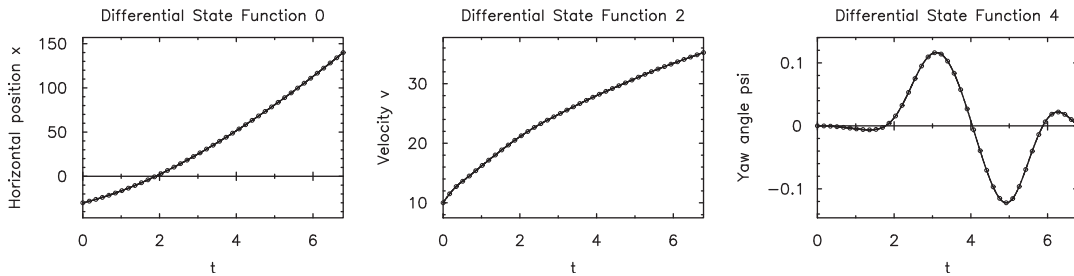


Figure 5. States for $N_{\text{shoot}}=40$: horizontal position, velocity, and raw angle ψ . For vertical position see Figure 2.

Table IV. Solutions computed by our convex relaxation approach.

N_{shoot}	N_{var}	N_{eq}	N_{in}	Iterations	CPU time	Relaxed: t_f	Rounded: t_f	N_{vI}
10	155	89	365	65	00:00:07	6.798389	6.798389	0
20	316	169	695	93	00:00:24	6.779035	6.779035	0
40	616	329	1355	91	00:00:46	6.786730	6.786730	2
80	1216	649	2765	129	00:04:19	6.789513	6.789513	2

$w_i(t) \in [0, 1], i = 1 \dots 5$. If the so-obtained trajectory is not integer, we round the solution. Doing this for different values of N_{shoot} , we obtain the values for $\mu(\cdot)$ given in Table III. Furthermore, in all cases optimization results in $F_B \equiv 0$ and $\phi \equiv 1$, i.e. no braking and full acceleration. Parts of the resulting differential states are shown in Figure 5.

In Table IV we summarize properties of the problem and numerical results. Columns one to four give the problem dimensions. Here we list N_{shoot} , the number of multiple shooting intervals used for the state, control, and constraint discretization. The dimensions of the underlying NLP, namely the number of variables (unknowns) N_{var} , the number of equality constraints

N_{eq} , and the number of inequality constraints N_{in} are also given. The next column lists the number of SQP iterations that were required to satisfy the KKT tolerance of 10^{-8} . The CPU time in hh:mm:ss format spent on these iterations is given in the next column. The final time t_f is listed once for the result of the relaxed, continuous control problem and once for the solution that was obtained by rounding the gear choice decisions. The latter solution can be obtained easily from the relaxed one, a procedure that requires less than half a second computation time. The value N_{vI} in the last column indicates how many discrete decisions were not in $\{0,1\}$ in the relaxed solution. As can be observed, the relaxed solution is in all the cases almost

integer feasible. The difference occurs on the last two intervals, as these do not have much influence on the outcome any more. This is also the reason why the objective function value of the rounded solution is identical to six digits. We will investigate and explain the astonishing observation that the relaxed solutions are already integer in Section 4.5.

Note that the objective function values are not monotonically decreasing, as one might expect considering that additional degrees of freedom are available as N_{shoot} increases and the control discretization becomes finer. In [2] it is claimed that this is due to the fact that the application of his Runge–Kutta method on different grids was a nonlinear operation. Although this is true, we doubt that this can explain the observed behavior, especially considering our error-controlled integration with a fine tolerance. Instead we see two reasons. First, the objective term contains also the steering velocity, compare (22a), which has a minor influence. Second, and more important, the evaluation of the path constraints is coupled to the multiple shooting discretization in our implementation. Using more shooting nodes leads to more inequalities constraining the problem, thus potentially increasing the minimum attained objective. Figure 2 shows this graphically. This argument is supported by the observation that, in the presented computations, the number of iterations tends to increase with the number of shooting nodes, while we usually find it being rather insensitive. We suppose that this coupling of constraints is also the case for the calculations presented in [1, 2], as the objective function values coincide. If the accurate compliance with the path constraint is an issue, we recommend to include safety distances, to evaluate at more points, or to apply techniques as proposed in [35] in order to track violations.

A more detailed distribution of the computing times is given in Table V. As can be clearly seen, for increasing values of N_{shoot} condensing and solution of the condensed quadratic programs (QPs) become predominant.

4.3. Switching time optimization

If we use the results from Section 4.2 to initialize the optimization of the switching points, compare

Section 3.5, we obtain the solutions to the control problem given in Table VI.

Note how the switching points are moved by the optimizer from the initialization given in Table III to values that are very much alike, although the continuous controls and path constraints are still discretized on different grids.

Table VII gives details about the computational effort. This effort is an additional cost and has to be added to the initialization phase, which corresponds to Table IV. Note again how the final time is growing with N_{shoot} due to a finer discretization of the path constraint.

4.4. Comparison

We compare our results to the reference solutions published in the OCAM papers [1, 2]. A Branch&Bound method was used in [1] for the solution of the MINLPs arising from the discretization of the presented optimal control problem. In [2] a time-transformation method was applied, which is basically similar to the switching time optimization approach presented in Section 3.5. The NLP subproblem generated per node of the branching tree was solved using the SODAS software package [36], employing a classical fourth-order Runge–Kutta scheme with fixed discretization grid for the solution of the ODE system. Both reference papers as well as our implementation computed derivatives via finite differences.

In Table VIII, the first column gives the number N of discretization points. This number corresponds to the number N_{shoot} in our multiple shooting approach. In the second column the number N_{nodes} of Branch&Bound nodes examined (NLP subproblems solved) in [1] is given. The third column lists the final time t_f , which is the global optimum determined by the Branch&Bound algorithm. The fourth column gives the computing time. Columns five and six give the respective results presented in [2]. According to [1], computation times refer to a Pentium III machine with 750 MHz, in [2] to a Pentium mobile processor with 1.6 GHz processing speed.

As can be seen comparing Tables IV and VIII, our approach on a fixed grid yields better results than the Branch&Bound applied in [1]. We applied

Table V. Detailed CPU times for most important calculations.

	$N_{\text{shoot}} = 10$	$N_{\text{shoot}} = 20$	$N_{\text{shoot}} = 40$	$N_{\text{shoot}} = 80$
Sensitivity generation (IND)	00:00:06	00:00:16	00:00:19	00:00:24
State integration	00:00:01	00:00:01	00:00:01	00:00:01
Condensing	00:00:00	00:00:01	00:00:05	00:00:44
Solution of condensed QPs	00:00:00	00:00:05	00:00:21	00:03:10

Table VI. Gear choice depending on N_{shoot} .

N_{shoot}	$\mu = 1$	$\mu = 2$	$\mu = 3$	$\mu = 4$	$\mu = 5$	t_f
10	0.0	0.435956	2.733326	—	—	6.764174
20	0.0	0.435903	2.657446	6.467723	—	6.772046
40	0.0	0.436108	2.586225	6.684504	—	6.782052
80	0.0	0.435796	2.748930	6.658175	—	6.787284

Time points when gear becomes active.

Table VII. Solutions obtained by switching time optimization.

N_{shoot}	Iterations	CPU time	Initial: t_f	Optimal: t_f
10	13	00:00:01	6.798389	6.764174
20	38	00:00:05	6.779035	6.772046
40	49	00:00:11	6.786730	6.782052
80	43	00:00:56	6.789513	6.787284

Table VIII. Selection of reference solutions presented in [1, 2].

N	N_{nodes}	t_f in [1]	CPU time [1]	t_f in [2]	CPU time [2]
20	1119	6.779751	00:23:52	6.772516	00:02:01
40	146 941	6.786781	232:25:31	6.783380	00:09:40
80	—	—	—	6.789325	01:05:03

All CPU times given in hh:mm:ss.

error-controlled integration (tolerance 10^{-9}) and optimized to a KKT tolerance of 10^{-8} , thus being more accurate than the reference. Still, for the case $N = 40$ only 46 s instead of almost 10 days are needed (although on a faster machine), being faster by about four orders of magnitude (18 190). As comparing computing times on different computing architectures is always an issue, the number of 146 941 nonlinear control problems that needed to be solved vs only 1 in our approach is

more representative for what is happening. For higher values of N the combinatorial explosion will have even larger effects in favor of our method, as the effort grows exponentially in the Branch&Bound approach, but polynomially for our approach.

For the case that the switching time points are not fixed and may vary, we propose to use the precalculated solution as initialization. This helps to overcome some of the intrinsic problems of the switching time

approach, in particular the occurrence of many local minima as pointed out in [3]. Another advantage is that no *a priori* knowledge on the switching structure has to be fed to the optimizer. Comparing Table VIII with Table VII we see that, including the computation time we required to compute the initialization using the first presented approach, for $N=80$ we obtain a speed-up from 01:05:03 to 00:04:19 + 00:00:56; roughly a factor of one order of magnitude despite the higher precision (on 2.166 GHz vs 1.6 GHz, though). The main point, however, is not the speed-up for this particular case, but the fact that the outer convexification and the initialization of switching times have very positive effects on computational performance.

For the sake of comparability we used the same grids as in [1, 2]. Applying an adaptive refinement of the grid only in regions where the computed control is not integer or a switch is close by, compare [3], the algorithm will be even more effective and scale way better.

4.5. Discussion

The question arises, of course, why our convex reformulation of the MIOCP yields already an integer feasible solution for the relaxed problem, which would correspond to the root node of a possible Branch&Bound tree. Although this has been observed in other applications as well [5], this property is not true for general problems.

Assume we have a solution (x^*, w^*, u^*, p^*) of a convexified, relaxed (and unconstrained for the sake of simplicity) MIOCP that is optimal. For this solution, the maximum principle must hold, see, e.g. [37] or [38]. Thus, we have the condition on the controls w^* almost everywhere in $[t_0, t_f]$ that

$$w^*(t) = \underset{w}{\operatorname{argmin}} \mathcal{H}(x^*(t), w, u^*(t), p^*, \lambda^*(t)) \quad (33)$$

As the Hamiltonian $\mathcal{H}(\cdot)$ of the convexified system reads as

$$\begin{aligned} \mathcal{H}(x^*, w, u^*, p^*, \lambda^*) \\ = \lambda^{*T} \left(\sum_{i=1}^{n_w} f(x^*, \mu^i, u^*, p^*) w_i \right) \end{aligned}$$

$$\begin{aligned} &= \sum_{i=1}^{n_w} \underbrace{\lambda^{*T} f(x^*, \mu^i, u^*, p^*)}_{\alpha_i :=} w_i \\ &\geq \sum_{i=1}^{n_w} \min_j \{\alpha_j\} w_i = \min_j \{\alpha_j\} \sum_{i=1}^{n_w} w_i = \min_j \{\alpha_j\} \end{aligned}$$

it follows

$$\min_w \mathcal{H}(x^*, w, u^*, p^*, \lambda^*) = \min_j \{\alpha_j\}$$

and the minimum of α_j with $1 \leq j \leq n_w$ determines pointwise the vector $w(t)$. If

$$k = \operatorname{argmin} \{\alpha_j, 1 \leq j \leq n_w\}$$

is unique, then the pointwise minimization of the Hamiltonian requires

$$w_i = \begin{cases} 1, & i = k \\ 0, & i \neq k \end{cases}$$

and the optimal solution is purely bang–bang. There are examples where this minimum is not unique, as, e.g. in [4]. For the case at hand, however, we deal with a special situation, as it is basically a time-optimal control problem. The target, to reach a certain goal in minimum time, can thus be reached best by a maximum acceleration if no constraints are active. Remember that the velocity v depends mainly on F_{lr}^μ given by Equation (12), which is a function of gas pedal position ϕ , the braking force F_B , the current gear μ , and velocity v itself. If we fix ϕ to 1, full acceleration, and F_B to zero, no braking, we can thus calculate the maximum value of F_{lr}^μ for all v in a certain range.

In Figure 6 we see that for every velocity v always one gear gives the maximum value, as any convex combination will be in between, and thus below the maximum. Exemplarily, a convex combination between the second and the third gear is shown in circle-line. Thus, any time-optimal solution in our convexified formulation will have bang–bang structure and be therefore integer. This is also true for the discretized problem, except on intervals on which the velocity v changes such that $\operatorname{argmax}_\mu F_{lr}^\mu(v)$ changes as well. Note that for path-constrained arcs things will be

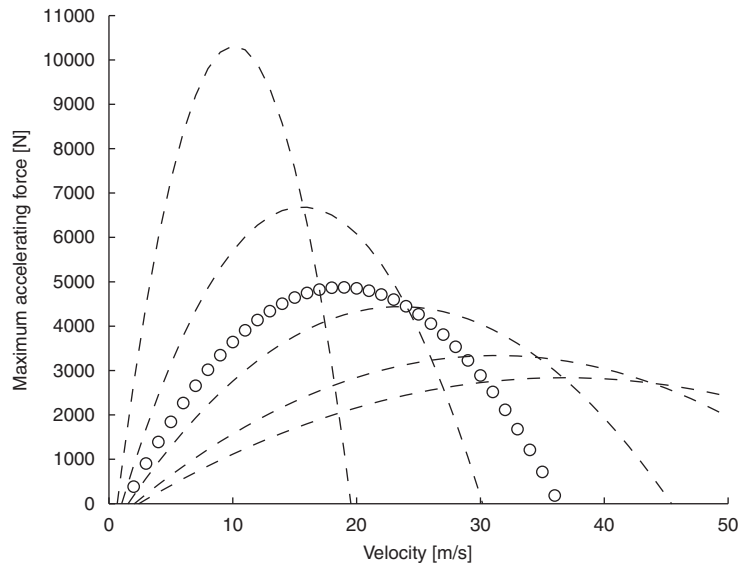


Figure 6. The longitudinal forces at the rear wheel F_{lr}^μ depending on velocity v , for $\mu = 1, \dots, 5$ in dotted lines. Depicted in circles is a noninteger solution, the convex combination $0.5F_{lr}^2 + 0.5F_{lr}^3$, which will result in less acceleration than the one obtained from an optimal gear choice, for all velocities v .

different and noninteger solutions may turn out to be optimal.

The outer convexification is only performed with respect to the integer variables. Hence, the obtained solutions are, like the reference solutions in [1, 2], local optima. All approaches could, if guaranteed global optima were of relevance from a practical point of view, be combined with established techniques from global optimization, e.g. convex underestimators and a spatial Branch&Bound.

5. CONCLUSIONS

We presented a reformulation of a recently published benchmark problem in mixed-integer optimal control, based on a convexification and a relaxation of the integer constraint. We solved the reformulated problem for two cases: (a) for a fixed, equidistant control discretization grid and (b) taking into account free switching times. For the first case, we reproduced the results obtained in [1] with a higher precision with

a speed-up of more than three orders of magnitude (taking into account the different computing environments) compared with a Branch&Bound approach. Our approach also gives the possibility to discretize even finer, as it does not suffer from a combinatorial explosion. For the second case we propose to use an initialization based on the solution of (a). We could reduce the computing time from over 1 h to 5 min (although on a slightly faster machine), applying our algorithm. We explained, why our reformulation is highly beneficial in the general case of time-optimal mixed-integer control problems.

The tremendous speed-up compared with previous approaches allows for an extension of the problem under investigation to longer horizons, using the same discretization precision, and also to investigate more complicated test tracks that result in nonintuitive switching structures. Future research will also concentrate on path-constrained solutions. Here we will apply theoretical results that guarantee an exact lower bound on the integer solution and apply novel rounding techniques to get initializations for the switching time optimization approach, as proposed in [6].

ACKNOWLEDGEMENTS

Financial help for one of the authors within the SIMUMAT Project S-050/ESP-0158 in the PRICIT program of the CAM in Spain is gratefully acknowledged.

REFERENCES

- Gerdtz M. Solving mixed-integer optimal control problems by Branch&Bound: a case study from automobile test-driving with gear shift. *Optimal Control Applications and Methods* 2005; **26**:1–18.
- Gerdtz M. A variable time transformation method for mixed-integer optimal control problems. *Optimal Control Applications and Methods* 2006; **27**(3):169–182.
- Sager S. *Numerical Methods for Mixed-integer Optimal Control Problems*. Der andere Verlag: Tönning, Lübeck, Marburg, 2005. URL: <http://sager1.de/sebastian/downloads/Sager2005.pdf>, ISBN: 3-89959-416-9, available at: <http://sager1.de/sebastian/downloads/Sager2005.pdf>.
- Sager S, Bock H, Diehl M, Reinelt G, Schlöder J. Numerical methods for optimal control with binary control functions applied to a Lotka–Volterra type fishing problem. In *Recent Advances in Optimization: Proceedings of the 12th French–German–Spanish Conference on Optimization*, Seeger A (ed.). Lectures Notes in Economics and Mathematical Systems, vol. 563. Springer: Heidelberg, 2006; 269–289.
- Sager S, Diehl M, Singh G, Küpper A, Engell S. Determining SMB superstructures by mixed-integer control. In *Proceedings OR2006*, Karlsruhe, Waldmann KH, Stocker U (eds). Springer: Berlin, 2007; 37–44.
- Sager S, Reinelt G, Bock H. Direct methods with maximal lower bound for mixed-integer optimal control problems. *Mathematical Programming*, 2008. Published online at: <http://dx.doi.org/10.1007/s10107-007-0185-6> on 14 August 2007, URL <http://dx.doi.org/10.1007/s10107-007-0185-6>.
- Antsaklis P, Koutsoukos X. On hybrid control of complex systems: a survey. *Third International Conference ADMP'98, Automation of Mixed Processes: Dynamic Hybrid Systems*, Reims, France, March 1998, 1998; 1–8.
- Oldenburg J, Marquardt W, Heinz D, Leineweber D. Mixed logic dynamic optimization applied to batch distillation process design. *AIChE Journal* 2003; **49**(11):2900–2917.
- Oldenburg J. *Logic-based Modeling and Optimization of Discrete-continuous Dynamic Systems, Fortschritt-Berichte VDI Reihe 3, Verfahrenstechnik*, vol. 830. VDI Verlag: Düsseldorf, 2005.
- Sussmann H. A maximum principle for hybrid optimal control problems. *Conference Proceedings of the 38th IEEE Conference on Decision and Control*, Phoenix, 1999.
- Shaikh M. Optimal control of hybrid systems: theory and algorithms. *Ph.D. Thesis*, Department of Electrical and Computer Engineering, McGill University, Montreal, Canada 2004. URL: <http://www.cim.mcgill.ca/~mshaikh/>.
- Shaikh M, Caines P. On the hybrid optimal control problem: theory and algorithms. *IEEE Transactions on Automatic Control* 2007; **52**:1587–1603.
- Attia S, Alamir M, Canudas de Wit C. Sub optimal control of switched nonlinear systems under location and switching constraints. *IFAC World Congress*, Prague, Czech Republic, 2005.
- Alamir M, Attia SA. On solving optimal control problems for switched hybrid nonlinear systems by strong variations algorithms. *Sixth IFAC Symposium, NOLCOS*, Stuttgart, Germany, 2004.
- Terwen S, Back M, Krebs V. Predictive powertrain control for heavy duty trucks. *Proceedings of IFAC Symposium in Advances in Automotive Control*, Salerno, Italy, 2004; 451–457.
- Chachuat B, Singer A, Barton P. Global methods for dynamic optimization and mixed-integer dynamic optimization. *Industrial and Engineering Chemistry Research* 2006; **45**(25):8373–8392.
- Kawajiri Y, Biegler L. Large-scale optimization strategies for zone configuration of simulated moving beds. *Sixteenth European Symposium on Computer Aided Process Engineering and Ninth International Symposium on Process Systems Engineering*. Elsevier: Amsterdam, 2006; 131–136.
- Bock H, Longman R. Computation of optimal controls on disjoint control sets for minimum energy subway operation. *Proceedings of the American Astronomical Society. Symposium on Engineering Science and Mechanics*, Taiwan, 1982.
- Plitt K. Ein superlinear konvergentes Mehrzielverfahren zur direkten Berechnung beschränkter optimaler Steuerungen. *Master's Thesis*, Universität Bonn, 1981.
- Bock H, Plitt K. A multiple shooting algorithm for direct solution of optimal control problems. *Proceedings 9th IFAC World Congress*, Budapest. Pergamon Press: Oxford, 1984; 243–247. URL: <http://www.iwr.uni-heidelberg.de/groups/agbock/FILES/Bock1984.pdf>.
- Biegler L. Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation. *Computers and Chemical Engineering* 1984; **8**:243–248.
- Binder T, Blank L, Bock H, Bulirsch R, Dahmen W, Diehl M, Kronseder T, Marquardt W, Schlöder J, Stryk O. Introduction to model based optimization of chemical processes on moving horizons. In *Online Optimization of Large Scale Systems: State of the Art*, Grötschel M, Krumke S, Rambau J (eds). Springer: Berlin, 2001; 295–340. URL: <http://www.zib.de/dfg-echtzeit/Publikationen/Preprints/Preprint-01-15.html>.
- Till J, Engell S, Panek S, Stursberg O. Applied hybrid system optimization: an empirical investigation of complexity. *Control Engineering* 2004; **12**:1291–1303.
- Pacejka H, Bakker E. The magic formula tyre model. *Vehicle System Dynamics* 1993; **21**:1–18.
- Sager S. MIOCP benchmark site. Available from: <http://mintoc.de>.
- Leineweber D, Bauer I, Schäfer A, Bock H, Schlöder J. An efficient multiple shooting based reduced SQP strategy for

- large-scale dynamic process optimization (Parts I and II). *Computers and Chemical Engineering* 2003; **27**:157–174.
27. Diehl M, Leineweber D, Schäfer A. MUSCOD-II Users' Manual. *IWR-Preprint 2001-25*, Universität Heidelberg, 2001.
 28. Bauer I, Bock H, Schlöder J. DAESOL—a BDF-code for the numerical solution of differential algebraic equations. *Internal Report, IWR, SFB 359*, Universität Heidelberg, 1999.
 29. Albersmeyer J, Bock H. Sensitivity generation in an adaptive BDF-method. In *Modeling, Simulation and Optimization of Complex Processes: Proceedings of the International Conference on High Performance Scientific Computing*, Hanoi, Vietnam, 6–10 March 2006, Bock H, Kostina E, Phu X, Rannacher R (eds). Springer: Berlin, 2008; 15–24.
 30. Petzold L, Li S, Cao Y, Serban R. Sensitivity analysis of differential-algebraic equations and partial differential equations. *Computers and Chemical Engineering* 2006; **30**:1553–1559.
 31. Leineweber D. Efficient reduced SQP methods for the optimization of chemical processes described by large sparse DAE models. *Fortschritt-Berichte VDI Reihe 3. Verfahrenstechnik*, vol. 613. VDI Verlag: Düsseldorf, 1999.
 32. Kaya C, Noakes J. Computations and time-optimal controls. *Optimal Control Applications and Methods* 1996; **17**:171–185.
 33. Kaya C, Noakes J. A computational method for time-optimal control. *Journal of Optimization Theory and Applications* 2003; **117**:69–92.
 34. Maurer H, Büskens C, Kim J, Kaya Y. Optimization methods for the verification of second-order sufficient conditions for bang-bang controls. *Optimal Control Methods and Applications* 2005; **26**:129–156.
 35. Potschka A. Handling path constraints in a direct multiple shooting method for optimal control problems. *Diplomarbeit*, Universität Heidelberg, 2006. URL: <http://apotschka.googlepages.com/APotschka2006.pdf>.
 36. Gerdt M. Direct shooting method for the numerical solution of higher index DAE optimal control problems. *Journal of Optimization Theory and Applications* 2003; **117**(2):267–294.
 37. Bryson A, Ho YC. *Applied Optimal Control*. Wiley: New York, 1975.
 38. Pontryagin L, Boltyanski V, Gamkrelidze R, Mischenko E. *The Mathematical Theory of Optimal Processes*. Wiley: Chichester, 1962.