



Computational Approaches for Mixed Integer Optimal Control Problems with Indicator Constraints

Michael N. Jung¹ · Christian Kirches²  · Sebastian Sager³ · Susanne Sass⁴

Received: 19 December 2017 / Accepted: 21 August 2018 / Published online: 5 October 2018
© Vietnam Academy of Science and Technology (VAST) and Springer Nature Singapore Pte Ltd. 2018

Abstract

Optimal control problems with mixed integer control functions and logical implications, such as a state-dependent restriction on when a control can be chosen (so-called indicator or vanishing constraints) frequently arise in practice. A prominent example is the optimal cruise control of a truck. As every driver knows, admissible gear choices critically depend on the current velocity. A large variety of approaches has been proposed on how to numerically solve this challenging class of control problems. We present a computational study in which the most relevant of them are compared for a reference model problem, based on the same discretization of the differential equations. This comprehends dynamic programming, implicit formulations of the switching decisions, and a number of explicit reformulations, including mathematical programs with vanishing constraints in function spaces. We survey all of these approaches in a general manner, where several formulations have not been reported in the literature before. We apply them to a benchmark truck cruise control problem and discuss advantages and disadvantages with respect to optimality, feasibility, and stability of the algorithmic procedure, as well as computation time.

Keywords Mixed integer optimal control · Indicator constraints · Vanishing constraints · Switched systems · MINLP · Heavy-duty truck · Cruise control · Dynamic programming · Switching function · Partial outer convexification

Mathematics Subject Classification (2010) 49-04 · 49M37 · 65K05 · 90-08 · 90C30 · 90C33 · 90C39 · 90C59 · 90C90 · 93B40

1 Introduction

Mixed integer optimal control problems (MIOCPs, also known as switched systems) have been gaining significantly increased interest. The underlying processes have a high potential

Dedicated to Hans Georg Bock on the occasion of his 70th birthday.

✉ Christian Kirches
c.kirches@tu-bs.de

Extended author information available on the last page of the article.

for optimization, while at the same time, they are hard to assess manually due to their combinatorial, nonlinear, and dynamic nature. Typical examples are water or gas networks [14, 45], traffic flow [22, 28], supply chain networks [44], distributed autonomous systems [1], processes in chemical engineering that involve valves [35, 65], and the choice of gears in automotive control [13, 25, 42]. See [56] for an open benchmark library for MIOCPs. The truck benchmark problem we discuss in this article is motivated by work of [30, 33, 36–38, 40, 68].

Definition 1 (MIOCP) In this article, we refer to the switched dynamic optimization problem given by

$$\begin{aligned}
 & \min_{\mathbf{x}(\cdot), \mathbf{u}(\cdot), \mathbf{Y}(\cdot)} e(\mathbf{x}(t_f)) \\
 & \text{s.t. } \bigoplus_{1 \leq i \leq n_\omega} \left[\begin{array}{l} Y_i(t) \\ \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{v}^i) \\ \mathbf{0} \leq \mathbf{c}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{v}^i) \end{array} \right] \quad t \in [0, t_f] \text{ a.e.}, \\
 & \mathbf{x}(0) = \mathbf{x}_0, \\
 & \mathbf{0} \leq \mathbf{d}(\mathbf{x}(t), \mathbf{u}(t)) \quad t \in [0, t_f] \text{ a.e.}
 \end{aligned}$$

as a mixed integer optimal control problem (MIOCP). The disjunction \oplus over $1 \leq i \leq n_\omega$ signifies that, at every point on the time horizon $t \in [0, t_f]$ in time, exactly one of the n_ω possible modes is chosen. This choice is represented here by time-dependent logical literals $Y_i(\cdot)$, $1 \leq i \leq n_\omega$. Setting $Y_i(t) = \text{true}$ selects a mode; the other literals then assume the value $Y_j(t) = \text{false}$ for $j \neq i$. The control $\mathbf{u} : [0, t_f] \rightarrow \mathbb{R}^{n_u}$ is assumed to be measurable and of bounded variation. The differential states $\mathbf{x} : [0, t_f] \rightarrow \mathbb{R}^{n_x}$ are assumed to be uniquely determined by \mathbf{f} and \mathbf{x}_0 once a switching regime $\mathbf{Y}(\cdot)$ and a control $\mathbf{u}(\cdot)$ are fixed. The vectors $\mathbf{v}^i \in \mathbb{R}^{n_v}$ comprise constant values specific for the given mode, and we let $\Omega := \{\mathbf{v}^1, \mathbf{v}^2, \dots, \mathbf{v}^{n_\omega}\}$. The objective function $e : \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ of Mayer type and the constraint functions $\mathbf{c} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_v} \rightarrow \mathbb{R}^{n_c}$ and $\mathbf{d} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_d}$ are assumed to be twice continuously differentiable.

A challenging part of solving a MIOCP is to find optimal discrete mode choices $\mathbf{Y}(\cdot)$. We are particularly interested in the indicator constraint $0 \leq \mathbf{c}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{v}^i)$ in Definition 1, which only plays a role if mode i is active (indicated) at time t . In this article, we discuss several approaches to formulating these and to computationally solving the arising optimization problems.

- In *direct methods with explicit switching*, the boolean literals $\mathbf{Y}(\cdot)$ are included as optimization variables, giving rise to nonlinear non-convex mixed integer optimization problems. Typically, continuous relaxations are solved within methods that provide integer solutions, such as Branch & Bound or rounding.
- In *direct methods with implicit switching*, the truth values of the boolean literals $\mathbf{Y}(\cdot)$ are computed from a *switching function* that uniquely determines the current mode and makes sure that the indicator constraints $0 \leq \mathbf{c}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{v}^i)$ are fulfilled.
- *Dynamic programming* provides a global solution on a given discretization and allows an enumerative treatment of integrality and indicator constraints.

There are of course connections between the different formulations; e.g., a transcription method from implicit switching to explicit switching on a time discretization grid

is proposed in [11]. And there are further approaches that we are not addressing in this paper. Indirect methods use necessary optimality conditions for (MIOCP) in function space, compare [26], and solve the resulting boundary value problem. Moment relaxations use polynomial optimization for switched systems [16, 58]. Time transformation approaches transform the problem into one where the stage lengths become continuous optimization variables, [25]. See [50] for an extension to indicator constraints. See [60] for a survey with further references and a discussion of advantages of different formulations for MIOCPs without indicator constraints.

There are several reformulations of logical relationships in the literature. Generalized disjunctive programming results directly from a logical modeling paradigm. It generalizes the disjunctive programming approach of [4]. Logical variables are usually either incorporated by means of big- M constraints or via a convex hull formulation (see [27, 49, 51, 67]). From a different point of view, disjunctive programming formulations can be interpreted as the result of *reformulation-linearization technique* (RLT) steps [64]. For both, the convex hull relaxation uses perspective functions. Based on this, the use of perspective cuts to strengthen convex MINLP relaxations has been proposed in various articles, for example, [15, 21, 29]. MINLP techniques are surveyed in [8].

Complementarity and indicator/vanishing constraints are another way to look at logical implications. The general concept of nonlinear optimization over non-convex structures is discussed in [62, 63]. For the comparatively new problem class of MPVCs, we refer to [2, 31]. Due to the lack of constraint qualification, various approaches for the computational solution of MPCCs and MPVCs have been devised and include regularizations [31, 54, 66], smoothing [17, 31], and combinations thereof (see [23] for an overview). Nonlinear programming for MPCCs is addressed in [3, 19, 46, 47]. Active set methods tailored to the non-convex structure are discussed in [18, 36, 41]. Formulations of MPCCs and MPVCs in optimal control can be found in [5, 36, 43, 52, 53].

In this article, we propose a new implicit approach to solve mixed integer optimal control problems with indicator constraints, and tailor it to the case of truck control. In addition, we provide the first comprehensive numerical assessment of different explicit, implicit, local, and global solution approaches to this problem class. Some of the results concerning explicit approaches have already been published in the book contribution [33] and the PhD thesis [32]. They are extended here by new variants of perspective reformulations, and also by considerations of dynamic programming [13] and implicit switching approaches. Several aspects play a role in this comparison:

- Some approaches provide an integer solution, while others solve a relaxed problem. For integer-feasible solutions, a *small objective function value* is obviously desirable, preferably the global minimum of (MIOCP). For *fractional* solutions of relaxed problems, the situation is different: in addition to local vs. global minimality, the aspect of the tightness of the relaxation plays a role. For relaxed solutions, a *higher objective function value* may indicate that the relaxation is tighter and of better use in a Branch & Bound scheme.
- Due to discretization (e.g., the discretization of the state space in dynamic programming) or rounding solutions may be *infeasible* in a forward simulation.
- The computation of optimal controls for the autonomous vehicle problem has to take place in real-time in order to enable a near instantaneous reaction to speed limits, slope changes, and to unforeseen alterations of traffic conditions. At the same time, all computations need to be performed by an on-board embedded system with limited computational resources.

We compare all approaches with respect to objective function value, fractionality, infeasibility, and CPU time.

In Section 2, we describe a benchmark MIOC problem, the control of a heavy duty truck, including two scenarios. In Section 3, we describe a family of so-called explicit approaches to MIOC, including relaxations from partial inner and outer convexification, and we propose several relaxations based on the perspective. In Section 4, we propose a new family of formulations belonging to the so-called implicit solution approach to MIOC. In Section 5, we describe a dynamic programming approach to MIOC. We describe and discuss the setting and the numerical results of the computational study in Section 6 and conclude with a summary in Section 7.

2 A Heavy-Duty Truck Cruise Control Problem

We present a mathematical model for a truck cruise control problem that is the base of all following comparisons. For more detailed expositions of various related models, we refer the reader to [12, 32, 33, 36, 38].

2.1 Controls and Dynamic System

The independent variable of the model is the traveled distance s in the range $[0, s_f]$ (in m). There are two control functions with continuous domain, the indicated engine torque M_{ind} and the engine brakes torque M_{brk} (both in Nm). The gear choice $\mu : [0, s_f] \mapsto \{1, \dots, n_\mu\}$ is a discrete control. There are two differential states, velocity v (in m/s), and accumulated fuel consumption Q (in l/s). The longitudinal dynamics are given by the ordinary differential equation

$$\begin{aligned} \dot{v} &= \frac{1}{m} v \left(\frac{i_A}{r_{stat}} (i_T(\mu) \eta_T(\mu) M_{ind} - M_{brk} - i_T(\mu) M_{fric}(v)) + M_{ext}(v, \gamma) \right) \\ &=: f_v(v, M_{ind}, M_{brk}, i_T(\mu), \eta_T(\mu), \gamma), \end{aligned} \tag{1}$$

and the accumulated fuel consumption is given by

$$\begin{aligned} \dot{Q} &= \frac{1}{v} \left(c_0 + (c_1 n_{eng}(v, i_T(\mu)) + c_2 M_{ind})^2 \right) \\ &=: f_Q(v, M_{ind}, i_T(\mu)), \end{aligned} \tag{2}$$

with engine speed

$$n_{eng}(v, i_T(\mu)) := v i_A i_T(\mu) 60 / (2\pi r_{stat}), \tag{3}$$

engine friction (in Nm)

$$M_{fric}(v) := \left(c_4 n_{eng}(v, i_T(\mu))^2 + c_5 n_{eng}(v, i_T(\mu)) + c_6 \right), \tag{4}$$

and the influence of air, gravity, and the influence of the road slope γ (in rad) via rolling friction as part of the external engine torque M_{ext} (in Nm)

$$M_{ext}(v, \gamma) := -\frac{1}{2} c_w A \rho v^2 - mg \sin(\gamma) + c_7 \cos(\gamma).$$

The values i_A, r_{stat} (in m), $i_T, \eta_T, c, m, g, c_w, A$ are constant model parameters and are independent of s , while all other quantities depend on the position s .

2.2 Objective Function

The cost to be minimized on the time horizon $[0, s_f]$ comprises three contributing terms, namely a penalty term for the deviation of the velocity from the desired one

$$\Phi_{\text{dev}}(s_f) := \int_0^{s_f} (v(s) - v_{\text{des}}(s))^2 ds, \tag{5a}$$

and the overall fuel consumption

$$\Phi_{\text{fuel}}(s_f) := Q(s_f). \tag{5b}$$

The terms are weighted by weights $\lambda_d, \lambda_f \geq 0$ and summed up. A comfort term that penalizes rapid torque changes

$$\Phi_{\text{comf}}(s_f) := \int_0^{s_f} \dot{M}_{\text{ind}}^2(s) + \dot{M}_{\text{brk}}^2(s) ds \tag{5c}$$

could also be included, but would require \dot{M}_{ind} and \dot{M}_{brk} to be the controls subject to optimization in order to obtain a differentiable objective.

2.3 Constraints

We account for mechanical constraints on the engine speed (3) with

$$n_{\text{eng}}^{\min} \leq n_{\text{eng}}(v(s), i_T(\mu(s))) \leq n_{\text{eng}}^{\max} \tag{6}$$

and on the torques with

$$M_{\text{ind}}(s) \leq M_{\text{ind}}^{\max}(v(s), i_T(\mu(s))), \tag{7a}$$

$$0 \leq M_{\text{ind}}(s), \tag{7b}$$

$$0 \leq M_{\text{brk}}(s) \leq M_{\text{brk}}^{\max}, \tag{7c}$$

for $s \in [0, s_f]$. The maximum engine torque is given by

$$M_{\text{ind}}^{\max}(v, i_T(\mu)) := c_7 - (n_{\text{eng}}(v, i_T(\mu)) - c_8)^2 / c_9.$$

There may be speed limits, e.g., by law,

$$v(s) \leq v_{\text{law}}(s), \quad s \in [0, s_f]. \tag{8}$$

2.4 Problem Formulation

Summarizing, this leads to the following problem formulation for the heavy-duty truck control problem on the time horizon $s \in [0, s_f]$:

$$\begin{aligned} & \min_{\mathbf{x}(\cdot), \mathbf{u}(\cdot), \mu(\cdot)} \lambda_d \Phi_{\text{dev}}(s_f) + \lambda_f \Phi_{\text{fuel}}(s_f) \\ & \text{s.t. } \bigoplus_{1 \leq i \leq n_\mu} \left[\begin{array}{l} \mu(s) = i \\ \text{ODE system (1), (2)} \\ \text{Constraints (6), (7a)} \end{array} \right] \quad s \in [0, s_f] \text{ a.e.}, \\ & \mathbf{x}(0) = \mathbf{x}_0, \\ & \text{Constraints (7b), (7c), (8)} \quad s \in [0, s_f] \text{ a.e.}, \end{aligned} \tag{9}$$

with state vector $\mathbf{x}(s) = (v(s), Q(s))^T$, continuous control vector $\mathbf{u}(s) = (M_{\text{ind}}(s), M_{\text{brk}}(s))^T$, the integer control $\mu(s)$, and fixed initial state \mathbf{x}_0 .

3 Explicit Switching Formulations

We investigate explicit approaches to reformulate problem (MIOCP), with a special focus on problem (10). They all use binary control functions $\mathbf{y} : [0, s_f] \mapsto \{0, 1\}^{n_\mu}$ that indicate whether gear i is chosen at position s or not. These reformulations are by construction equivalent for integer gear choices, but their relaxations to $\mathbf{y} : [0, s_f] \mapsto [0, 1]^{n_\mu}$ may have completely different characteristics. As the relaxations play an important role in mixed integer algorithms (such as Branch & Bound or rounding heuristics), a thorough study of the tightness of the relaxations and of numerical properties is needed.

The character of the exclusive disjunction in (10) is captured via the special ordered set type 1 constraints

$$\sum_{j=1}^{n_\mu} y_j(s) = 1 \quad \forall s \in [0, s_f].$$

For a more compact notation, we leave away the argument (s) for the states v , Q and the controls M_{ind} , M_{brk} , \mathbf{y} in the following.

3.1 Inner Convexification (IC)

For problem (10), it is possible to reformulate the time-dependent disjunctions by means of a function $\mathbf{g} : [1, n_\mu] \rightarrow \mathbb{R}^2$ that can be inserted into the right-hand side function $f(\cdot)$ and into the constraints $c(\cdot)$ and has the property

$$\mathbf{g}(\mu) = (i_T(\mu), \eta_T(\mu))^T$$

for $\mu \in \{1, \dots, n_\mu\}$. One possibility is a convex combination of the tabulated values,

$$\mathbf{g}(\mathbf{y}) = \sum_{j=1}^{n_\mu} y_j (i_T(j), \eta_T(j))^T. \tag{10}$$

Other possibilities are a piecewise linear representation with special ordered set type 2 variables or fitted smooth convex functions $\mathbf{g}(\cdot)$ as suggested in [24].

Applying IC to the ODEs (1), (2) and the constraints (6), (7a) in problem (10), we obtain

$$\dot{v} = f_v(v, M_{\text{ind}}, M_{\text{brk}}, g_1(\mathbf{y}), g_2(\mathbf{y}), \gamma), \tag{11a}$$

$$\dot{Q} = f_Q(v, M_{\text{ind}}, g_1(\mathbf{y})) \tag{11b}$$

and

$$n_{\text{eng}}^{\min} \leq n_{\text{eng}}(v, g_1(\mathbf{y})) \leq n_{\text{eng}}^{\max}, \tag{12a}$$

$$M_{\text{ind}} \leq M_{\text{ind}}^{\max}(v, g_1(\mathbf{y})). \tag{12b}$$

3.2 Outer Convexification (OC)

Partial outer convexification [39, 55, 57, 60] uses a convex combination of all function evaluations on the top (outermost) level. The resulting problem may still be non-convex in the differential states or continuous controls; thus, it may only be a *partial* convexification.

Applying OC to the ODEs (1)–(2) and the constraints (6)–(7a) in problem (10), we obtain the one-row relaxations

$$\dot{v} = \sum_{j=1}^{n_\mu} y_j f_v(v, M_{\text{ind}}, M_{\text{brk}}, i_T(j), \eta_T(j), \gamma), \tag{13a}$$

$$\dot{Q} = \sum_{j=1}^{n_\mu} y_j f_Q(v, M_{\text{ind}}, i_T(j)) \tag{13b}$$

and

$$n_{\text{eng}}^{\min} \leq \sum_{j=1}^{n_\mu} y_j n_{\text{eng}}(v, i_T(j)) \leq n_{\text{eng}}^{\max}, \tag{14a}$$

$$M_{\text{ind}} \leq \sum_{j=1}^{n_\mu} y_j M_{\text{ind}}^{\max}(v, i_T(j)). \tag{14b}$$

3.3 Big M Constraints (bigM)

A classical way to reformulate indicator constraints of the type $y = 1 \Rightarrow c(x) \leq 0$ are Big-M formulations $c(x) \leq M(1 - y)$ with $M \geq \max_x c(x)$. The constraints (6)–(7a) in problem (10) are reformulated as

$$-n_{\text{eng}}(v, i_T(j)) \leq -n_{\text{eng}}^{\min} + M(1 - y_j), \tag{15a}$$

$$n_{\text{eng}}(v, i_T(j)) \leq n_{\text{eng}}^{\max} + M(1 - y_j), \tag{15b}$$

$$M_{\text{ind}} \leq M_{\text{ind}}^{\max}(v, i_T(j)) + M(1 - y_j). \tag{15c}$$

Problem-specific values M can be found in [33, p. 181].

3.4 Relaxed Vanishing Constraints (relVC)

We formulate the constraints (6)–(7a) via a multiplication with the indicating variable y_j and a relaxation by ε , i.e.,

$$y_j n_{\text{eng}}^{\min} - \varepsilon \leq y_j n_{\text{eng}}(v, i_T(j)) \leq y_j n_{\text{eng}}^{\max} + \varepsilon, \tag{16a}$$

$$y_j M_{\text{ind}} - \varepsilon \leq y_j M_{\text{ind}}^{\max}(v, i_T(j)). \tag{16b}$$

3.5 Smoothed Vanishing Constraints (smoVC)

We reformulate (6)–(7a) using a smoothing-relaxation formulation suggested by [31],

$$\phi_\varepsilon^{\text{VC}}(a, b) = \frac{1}{2} \left(ab + \sqrt{a^2 b^2 + \varepsilon^2} + \sqrt{b^2 + \varepsilon^2} - b \right) - \varepsilon. \tag{17}$$

This results in

$$0 \geq \phi_\varepsilon^{\text{VC}} \left(n_{\text{eng}}^{\min} - n_{\text{eng}}(v, i_T(j)), y_j \right), \tag{18a}$$

$$0 \geq \phi_\varepsilon^{\text{VC}} \left(n_{\text{eng}}(v, i_T(j)) - n_{\text{eng}}^{\max}, y_j \right), \tag{18b}$$

$$0 \geq \phi_\varepsilon^{\text{VC}} \left(M_{\text{ind},j} - M_{\text{ind}}^{\max}(v, i_T(j)), y_j \right). \tag{18c}$$

3.6 Perspective and Full Lifting (PFL)

In generalized disjunctive programming (GDP, [27]) two concepts are combined: lifting and convex hull relaxations via the perspectives of constraint functions.

Definition 2 (Perspective function) The *perspective* of a function $f : \mathbb{R}^{n_1} \times \dots \times \mathbb{R}^{n_n} \rightarrow \mathbb{R}$ with respect to the first p arguments $\mathbf{x}_1, \dots, \mathbf{x}_p$ is the function $\hat{f} : \mathbb{R} \times \mathbb{R}^{n_1} \times \dots \times \mathbb{R}^{n_n} \rightarrow \mathbb{R}$ defined by

$$\hat{f}(y; \mathbf{x}_1, \dots, \mathbf{x}_n) := \begin{cases} yf\left(\frac{\mathbf{x}_1}{y}, \dots, \frac{\mathbf{x}_p}{y}, \mathbf{x}_{p+1}, \dots, \mathbf{x}_n\right) & \text{if } y > 0, \\ 0 & \text{if } y = 0, \mathbf{x}_j = \mathbf{0} \text{ for some } j \leq p, \\ \infty & \text{otherwise.} \end{cases}$$

It is well known that, for fixed index p , this function is convex if f is convex. In this case, it yields the convex hull of the feasible disjunctive set if the binary variable $y \in \{0, 1\}$ is relaxed and the variables are linked via

$$\mathbf{x}_j \leq y\mathbf{x}_j^{\max}, \quad 1 \leq j \leq p$$

compare, e.g., [29]. The tightness of perspective-based relaxations is often very good, compare the study of [48] for switched affine control systems. Perspectives are, however, well known to cause numerical issues for small values of y . For computational purposes, we use the smooth perspective, defined as follows.

Definition 3 (Smooth perspective [61]) The function

$$\tilde{f}(y; \mathbf{x}) := \hat{f}((1 - \varepsilon)y + \varepsilon; \mathbf{x}) + \varepsilon(y - 1)f(\underbrace{\mathbf{0}, \dots, \mathbf{0}}_{p \text{ times}}, \mathbf{x}_{p+1}, \dots, \mathbf{x}_n)$$

is called smooth perspective function.

In GDP, auxiliary (lifted) variables are introduced, one copy for each disjunction. It is an open question, though, if really all variables should be lifted. In our *full lifting* formulation, we introduce trajectories $v_j, Q_j, M_{\text{ind},j}, M_{\text{brk},j} : [0, s_f] \mapsto \mathbb{R}$ for all states and continuous controls, and for all $j \in \{1, \dots, n_\mu\}$. We couple them via constraints

$$v = \sum_{j=1}^{n_\mu} v_j, \quad M_{\text{ind}} = \sum_{j=1}^{n_\mu} M_{\text{ind},j}, \tag{19a}$$

$$Q = \sum_{j=1}^{n_\mu} Q_j, \quad M_{\text{brk}} = \sum_{j=1}^{n_\mu} M_{\text{brk},j} \tag{19b}$$

that allow to evaluate the objective function. All constraints have to hold for all n_μ vectors of lifted variables, i.e., (7b), (7c), (8) are replaced by

$$0 \leq M_{\text{ind},j}, \tag{20a}$$

$$0 \leq M_{\text{brk},j} \leq y_j M_{\text{brk}}^{\text{max}}, \tag{20b}$$

$$v_j \leq y_j v_{\text{law}}. \tag{20c}$$

Applying perspectives to (1), (2) we obtain for $y_j > 0$

$$\dot{v}_j = \hat{f}_v(y_j; v_j, M_{\text{ind},j}, M_{\text{brk},j}, i_T(j), \eta_T(j), \gamma) \tag{21a}$$

$$= y_j f_v\left(\frac{v_j}{y_j}, \frac{M_{\text{ind},j}}{y_j}, \frac{M_{\text{brk},j}}{y_j}, i_T(j), \eta_T(j), \gamma\right) \tag{21b}$$

$$\dot{Q}_j = \hat{f}_Q(y_j; v_j, M_{\text{ind},j}, i_T(j)) \tag{21c}$$

$$= y_j f_Q\left(\frac{v_j}{y_j}, \frac{M_{\text{ind},j}}{y_j}, i_T(j)\right), \tag{21d}$$

or the respective smoothed counterparts.

Definition 4 (Smoothed dynamics) After eliminating y_j in the denominators of \hat{f}_v or \hat{f}_Q where possible, any gear indicator variables y_j remaining in the denominators are replaced by the smoothing

$$S\left(\frac{v_j}{y_j}, \varepsilon\right) := \frac{(1 - \varepsilon)v_j}{(1 - \varepsilon)y_j + \varepsilon} + \varepsilon. \tag{22}$$

Applying perspectives to the constraints (6), (7a) yields

$$y_j n_{\text{eng}}^{\text{min}} \leq y_j n_{\text{eng}}\left(\frac{v_j}{y_j}, i_T(j)\right) \leq y_j n_{\text{eng}}^{\text{max}}, \tag{23a}$$

$$M_{\text{ind},j} \leq y_j M_{\text{ind}}^{\text{max}}\left(\frac{v_j}{y_j}, i_T(j)\right). \tag{23b}$$

Because n_{eng} is linear in its first argument, elimination of all denominator occurrences of y_j is possible. Only constraint (23b) needs a smoothing that formally reads

$$M_{\text{ind},j} \leq y_j M_{\text{ind}}^{\text{max}}\left(S\left(\frac{v_j}{y_j}, \varepsilon\right), i_T(j)\right),$$

and wherein the y_j occurrence associated with the linear term of $M_{\text{ind}}^{\text{max}}$ is eliminated before smoothing.

3.7 Perspective and Partial Lifting (PPL)

We further study whether a partial lifting is beneficial. Looking closer at (1), (2) and (21), it appears tempting to only introduce lifted trajectories $v_j, M_{\text{ind},j} : [0, s_F] \mapsto \mathbb{R}$ for all $j \in \{1, \dots, n_\mu\}$, as M_{brk} and Q enter the dynamics and the objective function independently of the disjunction. This leaves only the constraints (19b), (20a), and (20c) to be considered.

Additionally, we split the right-hand side f_v into disjunction-dependent and independent parts. This allows to choose the v_j as algebraic variables instead of differential states,

$$\dot{v} = \frac{1}{mv} \left(M_{\text{ext}}(v, \gamma) - \frac{i_A}{r_{\text{stat}}} M_{\text{brk}} + \frac{i_A}{r_{\text{stat}}} \sum_{j=1}^{n_\mu} y_j \left(i_{\text{T}}(j) \eta_{\text{T}}(j) S \left(\frac{M_{\text{ind},j}}{y_j}, \varepsilon \right) - M_{\text{fric}} \left(S \left(\frac{v_j}{y_j}, \varepsilon \right), i_{\text{T}}(j) \right) \right) \right) \tag{24a}$$

$$\dot{Q} = \sum_{j=1}^{n_\mu} y_j f_Q \left(S \left(\frac{v_j}{y_j}, \varepsilon \right), S \left(\frac{M_{\text{ind},j}}{y_j}, \varepsilon \right), i_{\text{T}}(j) \right) \tag{24b}$$

and indicator constraints (23). After elimination of denominator y_j occurrences where possible, remaining velocities in the denominators of f_v and f_Q are smoothed using (22).

Additionally, not lifting the term $\frac{1}{mv}$ even though v is an aggregate of the lifted velocities v_j is justified by observing that

$$\sum_{j=1}^{n_\mu} (v_j \dot{v}_j) = \left(\sum_{j=1}^{n_\mu} v_j \right) \left(\sum_{j=1}^{n_\mu} \dot{v}_j \right) \quad \text{if } v_i \dot{v}_k = 0, i \neq k.$$

In binary feasible points y_j , the condition $v_i \cdot \dot{v}_k = 0$ for $i \neq k$ is implied by feasibility of the constraint (21a).

3.8 Perspective and Minimal Lifting (PML)

Compared to Section 3.7, we reduce the number of lifted trajectories further using the projections $v_j = y_j v$ and $M_{\text{ind},j} = y_j M_{\text{ind}}$ as first suggested in [32, 33]. The single ODEs can be written as

$$y_j \dot{v} = \dot{v}_j = y_j f_v (y_j; v, M_{\text{ind},j}, M_{\text{brk}}, i_{\text{T}}(j), \eta_{\text{T}}(j), \gamma).$$

Summing up over j we obtain

$$\dot{v} = \sum_{j=1}^{n_\mu} y_j f_v (y_j; v, M_{\text{ind},j}, M_{\text{brk}}, i_{\text{T}}(j), \eta_{\text{T}}(j), \gamma), \tag{25a}$$

$$\dot{Q} = \sum_{j=1}^{n_\mu} y_j f_Q (y_j; v, M_{\text{ind},j}, i_{\text{T}}(j)). \tag{25b}$$

As we do not use lifted velocities v_j anymore, we formulate the indicator constraints using the smoothing-relaxation (17) and obtain the constraints (18). Again, the perspectives and the velocities in the denominators of f_v and f_Q are smoothed according Definition 3 and to (22).

4 Implicit Switching Formulations

In this section, we propose a new family of implicit approaches to MIOCP. The general idea is to reformulate problem (10) by eliminating the integer control through introduction of implicit, i.e., state-dependent switches, before solving it numerically. This way, we obtain a

nonlinear optimal control problem with continuous domain of feasibility, but with implicitly discontinuous system dynamics. We consider the following class of problems:

Definition 5 (Switched system) In this article, we refer to the optimal control problem

$$\begin{aligned}
 & \min_{\mathbf{x}(\cdot), \mathbf{u}(\cdot)} e(\mathbf{x}(t_f)) & (26) \\
 & \text{s.t. } \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), \text{sign } \boldsymbol{\sigma}(\mathbf{x}(t))), \quad t \in [0, t_f] \text{ a.e.,} \\
 & \quad \mathbf{x}(0) = \mathbf{x}_0, \\
 & \quad \mathbf{0} \leq \mathbf{d}(\mathbf{x}(t), \mathbf{u}(t)), \quad t \in [0, t_f] \text{ a.e.}
 \end{aligned}$$

with right-hand side function \mathbf{f} depending on the sign structure of a vector-valued *switching function* $\boldsymbol{\sigma} : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_\sigma}$ that satisfies the *transversality condition*

$$\lim_{t \nearrow t_*} \frac{d\sigma_j}{dt}(\mathbf{x}(t)) \lim_{t \searrow t_*} \frac{d\sigma_j}{dt}(\mathbf{x}(t)) > 0 \tag{27}$$

for all $t_* \in [0, t_f]$ with $\sigma_j(t_*) = 0$ for some index $1 \leq j \leq n_\sigma$, as a *switched system* control problem.

For the particular case of the truck MIOCP, we obtain a switched system by modeling the discrete gear choice using a fixed dependency on a differential state, e.g., the current velocity. Whenever certain *switching velocities* \bar{v} are reached, a gear shift takes place.

Definition 6 (Switching velocity and set) The set $\bar{V} = \{\bar{v}_1, \dots, \bar{v}_{n_\mu+1}\}$ is called switching set, if

$$v_{\min} = \bar{v}_1 < \bar{v}_2 < \dots < \bar{v}_{n_\mu+1} = v_{\max}$$

and the gear $\mu(s)$ is chosen according to

$$\mu(s) = j \iff v(s) \in [\bar{v}_j, \bar{v}_{j+1}) \quad \forall s \in [0, s_f]$$

for a given trajectory $v : [0, s_f] \mapsto \mathbb{R}$ and given v_{\min} and v_{\max} . The elements \bar{v}_μ are called switching velocities.

4.1 Switching Velocities for Torque

We may define switching velocities for problem (10) based on any selected gear dependency. The gear choice enters the dynamics (1) and (2) as well as the constraints on the engine speed n_{eng} (6) and on the indicated engine torque (7a). In fact, the gear choice selects from a number of possible modes, each of which is characterized by a different pair of values (i_T, η_T) and affects the algebraic variables n_{eng} , M_{ind}^{\max} , f_Q and M_{fric} . Via the constraint on the engine speed n_{eng} (6), gear choice and velocity are linked. This results in restrictions on the switching velocities as shown in Fig. 1a.

On the other hand, gear choice and velocity restrict the indicated engine torque M_{ind} via the constraint (7a). The largest feasible domain for M_{ind} is obviously obtained for the

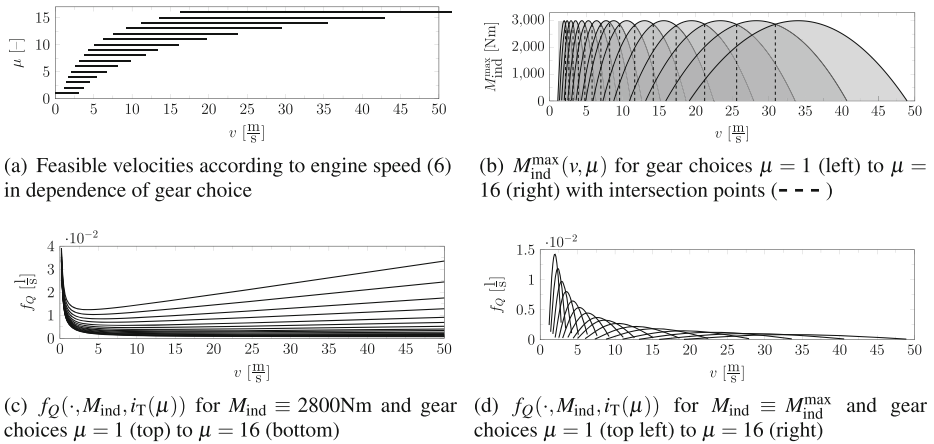


Fig. 1 Details of implicit modeling

switching velocities that result in the maximum upper bound, as illustrated in Fig. 1b. This results in the following switching set $\bar{V}_{M_{ind}^{max}}$:

μ	1	2	3	4	5	6	7	8
\bar{v}_μ	v_{min}	2.1767	2.6363	3.2093	3.9358	4.8226	5.8154	7.0174
μ	9	10	11	12	13	14	15	16
\bar{v}_μ	8.2176	9.5819	11.6467	14.1881	17.3212	21.2210	25.6023	30.8988

4.2 Switching Velocities for Fuel Consumption

Another approach to determine a switching set \bar{V} is to pick switching velocities according to f_Q . Such switching velocities help to minimize the fuel consumption Q that depends on $f_Q = f_Q(v, M_{ind}, i_T(\mu))$, c.f. (2). The behavior of the quotient is depicted in Fig. 1 for two different choices of M_{ind} . In Fig. 1a, M_{ind} is fixed to 2,800 Nm, a value that is often close to optimal in solutions using explicit formulations. In Fig. 1b, the control M_{ind} is fixed to its maximum value M_{ind}^{max} .

Figures 1 shows that the fuel consumption increases at a slower rate if higher gears are chosen. Hence, the fuel consumption can effectively be minimized by shifting to a higher gear at the earliest convenience, i.e., the lowest admissible velocity. This early switching may however also have an adverse affect. Choosing a high gear may prevent acceleration, or make it more costly. This line of thought is confirmed by the numerical results presented in Section 6. Hence, this approach for determining a switching set is discarded and we determine the switching set V_Q according to f_Q by computing the intersection points of the curves in Fig. 1b.

μ	1	2	3	4	5	6	7	8
\bar{v}_μ	v_{min}	2.3095	2.7965	3.4047	4.173	5.1166	6.1695	7.4453
μ	9	10	11	12	13	14	15	16
\bar{v}_μ	8.7239	10.1661	12.3530	15.0533	18.3689	22.5149	27.1612	32.7830

4.3 Further Switching Sets

Investigating M_{fric} for switching points does not yield new insights because the engine friction is minimized, too, if the highest admissible gear is chosen. In order to obtain further switching sets for computational comparison, we define

$$\bar{V}_{\text{ave}} = \left\{ \frac{1}{2} (\bar{v}_{Q,i} + \bar{v}_{\text{Mindmax},i}), 1 \leq i \leq 17 \right\}$$

and via

$$\hat{v}_i := \bar{v}_{Q,i} - \bar{v}_{\text{ave},i} = \frac{1}{2}(\bar{v}_{Q,i} - \bar{v}_{\text{ave},i})$$

the sets

$$\begin{aligned} \bar{V}_{\text{plus}} &= \{\bar{v}_{Q,i} + 1 \cdot \hat{v}_i, 1 \leq i \leq 17\}, \\ \bar{V}_{\text{pp}} &= \{\bar{v}_{Q,i} + 2 \cdot \hat{v}_i, 1 \leq i \leq 17\}, \\ \bar{V}_{\text{ppp}} &= \{\bar{v}_{Q,i} + 4 \cdot \hat{v}_i, 1 \leq i \leq 17\}. \end{aligned}$$

For an overview, all calculated switching velocities are shown in Fig. 2a. Switching velocities are depicted as lower bounds for all gear choices to improve clarity of exposition. The switching velocity of gear j then is the upper bound on the velocity of gear $j - 1$ according to condition (iii) of Definition 6.

4.4 Switching Function

Given a set \bar{V} , we can determine the switching function σ that relates the current velocity $v(s)$ uniquely to the current gear choice $\mu(s)$.

Definition 7 (Switching function) A switching function for problem (5) is a function

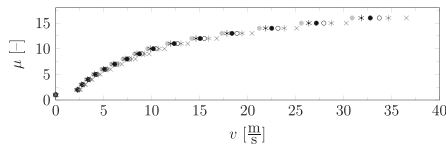
$$\sigma : [v_{\text{min}}, v_{\text{max}}] \rightarrow \mathbb{R}^{n_\mu}$$

that satisfies transversality (27) and

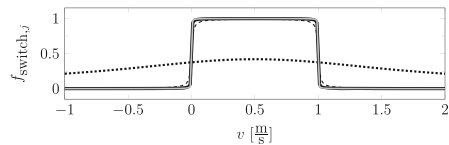
$$\text{sign } \sigma_j(v) = \begin{cases} +1 & \text{if } v \in (\bar{v}_j, \bar{v}_{j+1}) \\ -1 & \text{otherwise.} \end{cases} \stackrel{\text{Definition 6}}{\iff} \mu = j,$$

In addition, we define the indicator functions

$$\tilde{\sigma}_j(v) := \frac{1}{2}(\text{sign } \sigma_j(v) + 1) \in \{0, 1\}.$$



(a) Switching velocities of the different switching sets \bar{V}_{Mindmax} (●), \bar{V}_Q (●), \bar{V}_{ave} (*), \bar{V}_{plus} (○), \bar{V}_{pp} (*) and \bar{V}_{ppp} (×)



(b) Exemplary switching functions (28) with $[\bar{v}_j, \bar{v}_{j+1}] := [0, 1]$, $c_{\text{sw}} = 1$ (.....), $c_{\text{sw}} = 100$ (---), $c_{\text{sw}} = 500$ (—), $c_{\text{sw}} = 1000$ (—).

Fig. 2 Visualization of switching velocity sets and switching functions

This definition implies that $\tilde{\sigma}_j$ is a binary algebraic variable that equals 1 if and only if gear j is chosen. Hence, we have

$$\mu(s) = \sum_{j=0}^{n_\mu} j \cdot \tilde{\sigma}_j(v(s)) \quad \forall s \in [0, s_f].$$

As a consequence, the gear choice μ and all algebraic variables that depend on the gear choice are non-smooth step functions. For use with derivative-based solvers like the interior point code IPOPT [69], the resulting OCP violates the requirement of second order continuous differentiability, and this will in general cause convergence difficulties.

To address this issue, the smooth function

$$\sigma_j(v) := \left(\frac{1}{\pi} \arctan(c_{sw} \cdot (v - \bar{v}_j)) + \frac{1}{2} \right) \cdot \left(\frac{1}{\pi} \arctan(c_{sw} \cdot (\bar{v}_{j+1} - v)) + \frac{1}{2} \right), \quad (28)$$

with switching parameter $c_{sw} > 0$ may be used in place of the switching indicator function $\tilde{\sigma}$. As can be seen in Fig. 2b, this function meets Definition 7 for $c_{sw} \rightarrow \infty$. However, choosing c_{sw} too big results in a large derivative when the current velocity is close to a switching velocity, which may also cause difficulties for derivative based solvers. As a compromise with regard to the equivalence of the implicit approach and the original problem formulation as well as the convergence properties of a derivative solver, we chose $c_{sw} = 500$ for the numerical results of Section 6.

4.5 Reformulation

As in the explicit approaches, the switching function can be used in an Inner Convexification or an Outer Convexification setting. By using

$$g(v) = \sum_{j=1}^{n_\mu} \sigma_j(v(s))(i_T(j), \eta_T(j))^T$$

analogously to (10), the IC formulations (11) and (12) can be used. Substituting y_j by $\sigma_j(v(s))$ in (13), (14) provides OC formulations.

5 Dynamic Programming

All approaches described so far relied on the ability to compute a local minimum of a nonlinear program (NLP). In the course of our later assessment of the merits of these approaches, we are also interested in the quality of the local solutions obtained, i.e., in their distance from global optimality. A well-known path to (approximately) solving optimal control problems to global optimality is dynamic programming, see [6, 7, 9, 10], and [13].

5.1 Dynamic Programming

Dynamic programming computes optimal control trajectories by enumerating all possible control values. Hence, it relies on a discretized approximation of the control space, but cannot get stuck in local optima. Like all enumerative schemes, it suffers from the curse of dimensionality and may require excessive computational effort for larger instances. With some exceptions, e.g. [30], dynamic programming is seldom used for real-time optimization purposes, but has significant merit as a means of obtaining a reference value for comparison.

5.2 Bellman’s Principle of Optimality for MIOCP

DP is based on Bellman’s principle of optimality, c.f. [7], stating that “An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.” To describe this principle with formulas, we define the *cost-to-go function* for a MIOCP:

Definition 8 (Cost-to-go) The optimal cost-to-go function $J : [0, t_f] \times \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ for the MIOCP of Definition 1, a starting time $\bar{t} \in [0, t_f]$, and an initial value $\bar{\mathbf{x}} \in \mathbb{R}^{n_x}$ reads

$$\begin{aligned}
 J(\bar{t}, \bar{\mathbf{x}}) &:= \min_{\mathbf{x}(\cdot), \mathbf{u}(\cdot)} e(\mathbf{x}(t_f)) \\
 \text{s.t. } &\bigoplus_{1 \leq i \leq n_\omega} \left[\begin{array}{c} Y_i(t) \\ \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{v}^i) \\ \mathbf{0} \leq \mathbf{c}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{v}^i) \end{array} \right], \quad t \in [\bar{t}, t_f] \text{ a.e.,} \\
 &\mathbf{0} \leq \mathbf{d}(\mathbf{x}(t), \mathbf{u}(t)), \quad t \in [\bar{t}, t_f] \text{ a.e.,} \\
 &\mathbf{x}(\bar{t}) = \bar{\mathbf{x}},
 \end{aligned}$$

Bellman’s principle of optimality implies that, for the time discretization grid $0 = t_0 < t_1 < \dots < t_N = t_f$, we have the following recurrence relation for the cost-to-go function:

$$J(t_k, \mathbf{x}_k) = J(t_{k+1}, \mathbf{x}_{k+1}) + \min_{\mathbf{x}(\cdot), \mathbf{u}(\cdot)} \Phi(t_{k+1}; t_k, \mathbf{x}_k, \mathbf{u}(\cdot)),$$

where the evaluation of the contribution $\Phi(t_{k+1})$ on $[t_k, t_{k+1}]$ for a given initial value \mathbf{x}_k and control trajectories $\mathbf{u}(\cdot), \mathbf{v}(\cdot)$ requires solving the boundary-value problem

$$\begin{aligned}
 \dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{v}(t)), \quad t \in [t_k, t_{k+1}] \text{ a.e.,} \\
 \mathbf{0} &\leq \mathbf{c}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{v}(t)), \\
 \mathbf{0} &\leq \mathbf{d}(\mathbf{x}(t), \mathbf{u}(t)), \quad t \in [t_k, t_{k+1}] \text{ a.e.,} \\
 \mathbf{x}(t_k) &= \mathbf{x}_k, \\
 \mathbf{x}(t_{k+1}) &= \mathbf{x}_{k+1}.
 \end{aligned}$$

Furthermore, $J(t_N, \mathbf{x}_N) = e(\mathbf{x}_N)$ provides a starting point and minimizing the function $J(t_0, \mathbf{x}_0)$ over all admissible values for \mathbf{x}_0 yields the optimal objective function value and the solution of problem (1).

5.3 Dynamic Programming Algorithm for MIOCP

The main idea of dynamic programming is to compute all values $J(t_k, \mathbf{x}_k)$ in a backward loop $k = N, N - 1, \dots, 0$ that determines the resulting states \mathbf{x}_k and optimal controls $\mathbf{u}(\cdot)$ on $t \in [t_k, t_{k+1}]$. To this end, we introduce appropriately chosen discrete subsets $X_D \subset X, U_D \subset U$ of the bounded admissible state space $X \subset \mathbb{R}^{n_x}$ and control space $U \subset \mathbb{R}^{n_u}$,

$$X_D := \{\mathbf{x}^1, \dots, \mathbf{x}^{N_x}\}, \quad U_D := \{\mathbf{u}^1, \dots, \mathbf{u}^{N_u}\},$$

let $\mathbf{u}(t) := \mathbf{u}_k$ on $t \in [t_k, t_{k+1}]$, and examine the sets X_D, U_D and Ω by exhaustive search as follows:

Algorithm 1 Dynamic programming

1. For $i = 1, \dots, N_x$.
Let $J(t_N, \mathbf{x}^i) = e(\mathbf{x}^i)$.
2. For $k = N - 1, \dots, 0$.
 - (a) For $i = 1, \dots, N_x$.
 - i. For $j = 1, \dots, N_u, \ell = 1, \dots, n_\omega$.
Compute $\Phi(t_{k+1}; t_k, \mathbf{x}^i, \mathbf{u}^j, \mathbf{v}^\ell)$.
 - ii. Store $J(t_k, \mathbf{x}^i)$, minimizer $(\mathbf{u}^*, \mathbf{v}^*)$, and resulting state \mathbf{x}_{k+1} .
3. Let $\mathbf{x}_0^* := \operatorname{argmin} J(t_0, \mathbf{x}_0)$.
4. For $k = 0, \dots, N$:
 - (a) Let $\mathbf{u}_k^* := \mathbf{u}^*(\mathbf{x}_{k-1}^*)$ and look up \mathbf{x}_k^* .

A fixed initial value and/or small admissible state or control domains accelerate the algorithm. It is easy to see that Algorithm 1 has a runtime complexity of $\mathcal{O}(N \cdot N_x \cdot N_u \cdot n_\omega)$ and a storage requirement of $\mathcal{O}(N \cdot N_x)$ memory locations. For a uniform discretization granularity Δ of $X \subset \mathbb{R}^{n_x}$, we observe $N_x \in \mathcal{O}(\Delta^{n_x})$, and the same is true for the control spaces. Hence, both computational effort and memory requirements grow exponentially with the number of states n_x and the numbers of continuous controls n_u . This observation is often referred to as *curse of dimensionality*. The discrete choices, however, are easily included in the enumeration scheme. For more details on dynamic programming, we refer the reader to Bellman [6, 7] and, e.g., Bertsekas [9, 10].

To accelerate the algorithm, it is possible to terminate the for-loop(s) early if the solution clearly becomes infeasible. For example, the ODE system need not be solved if the particular new gear choice is found to violate the engine speed constraint (6) for the given current velocity. Furthermore, monotonicity properties may be used, i.e., if a given value of the engine torque is found to result in a new velocity in violation of the upper bound, even larger values of the engine torque need not be examined. On top of this, results of the ODE solver are stored in look-up tables to avoid repeating multiple identical computations. For more details on the efficient implementation we used, see Buchner [12].

6 Numerical Results and Comparison

6.1 Scenarios, Initial Values, and Parameter Values

All problem parameters and initial values, such as the number of gears $n_\mu = 16$ or $v_0 = v_{\text{des}} = 22.2 \frac{\text{m}}{\text{s}}$, can be found on the web page `mintoc.de`, a benchmark library for MIOCPs [56]. We consider two scenarios, for which the numerical data can also be found on `mintoc.de`.

Scenario 1 is characterized by a linear slope on the first half of the horizon, and a general speed limit of $27.8 \frac{\text{m}}{\text{s}} \approx 100 \frac{\text{km}}{\text{h}}$, cf. the left-hand part of Fig. 3. The weights for the objective are set to $\lambda_d = 1.0$, $\lambda_f = 25.0$, and $\lambda_c = 1.0$, i.e., we primarily minimize fuel consumption. In Scenario 2 the truck has to transit a valley. The speed on the way down is limited to $22.2 \frac{\text{m}}{\text{s}} \approx 80 \frac{\text{km}}{\text{h}}$ as can be seen on the right-hand side of Fig. 3. For this scenario, we set $\lambda_d = 100.0$, $\lambda_f = 10.0$, i.e., the main aspect is to minimize the deviation of velocity from the desired one. The considered horizons have a length of 1000m in both scenarios, which was shown to be a good prediction horizon in a moving horizon context, [12].

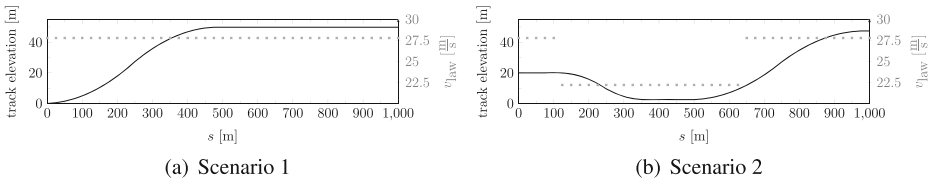


Fig. 3 Track elevations (—) and velocity limits (---) of the considered scenarios

6.2 Implementation Details

To make the numerical results as comparable as possible, we use an identical discretization throughout. All optimal control problems are discretized on the same grids, using 40 equidistant intervals for the piecewise constant controls (i.e., 25m) and 400 equidistant intervals for the states. The dynamics are discretized using an implicit Euler method. Due to the different algorithmic approaches a comparison of solutions has to consider numerical errors, compare Fig. 4.

All results in this article were computed on a single core of a x64-based Intel(R) Xeon(TM) E5-2640 v3 CPU with 2.60GHz and 32GB memory. For the solution of the direct collocation problems of all explicit and implicit formulations, we used IPOPT 3.12.4 [69] with standard solver options, invoked from 64-bit AMPL [20] version 20171002. The Dynamic Programming solution was obtained from a C++ implementation of Algorithm 1.

6.3 Comparison of Solutions

For the assessment of solutions, we use the indicators *fractionality* and *feasibility* that measure the violation of the integrality constraints $y_j \in \{0, 1\}$ and of the gear-dependent model constraints (6) and (7a).

Definition 9 (Fractionality of solutions) The *fractionality* of piecewise constant control functions $y_j : [0, s_f] \mapsto [0, 1]$ on a grid $0 = s_0 < \dots < s_N = s_f$ is given by the value

$$\frac{1}{N} \sum_{i=0}^{N-1} \sum_{j=1}^{n_\mu} (0.5 - |y_j(s_i) - 0.5|).$$

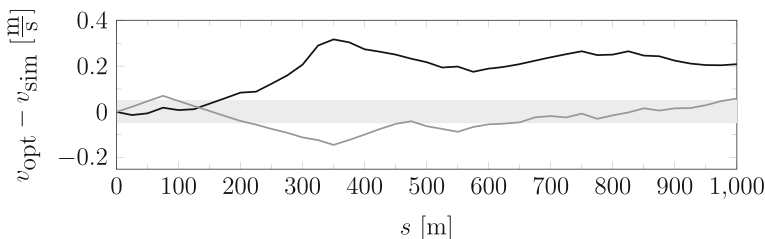


Fig. 4 Impact of rounding to state discretization grid in Dynamic Programming. Shown are the deviations of a forward simulation from the rounded values using $\Delta v = 0.05 \frac{m}{s}$ (■) for Scenario 1 (—) and Scenario 2 (—). This error makes an accurate comparison difficult

Definition 10 (Infeasibility of solutions) The *infeasibility* of states v and piecewise constant controls $M_{\text{ind}}, \mathbf{y}$ on a grid $0 = s_0 < \dots < s_N = s_f$ is given by the value

$$\frac{1}{3(N + 1)n_\mu} \sum_{i=0}^N \sum_{j=1}^{n_\mu} \sum_{k=1}^3 \max\{0, y_j(s_i)\delta_j^k(s_i)\}$$

with the residuals of the indicator constraints

$$\begin{aligned} \delta_j^1(s_i) &= M_{\text{ind}}(s_i) - M_{\text{ind}}^{\max}(v(s_i), i_T(j)), \\ \delta_j^2(s_i) &= n_{\text{eng}}^{\min} - n_{\text{eng}}(v(s_i), i_T(j)), \\ \delta_j^3(s_i) &= n_{\text{eng}}(v(s_i), i_T(j)) - n_{\text{eng}}^{\max}. \end{aligned}$$

Definition 11 (Relaxed solutions) For explicit approaches of Section 3, we call the solution of a control problem *relaxed*, if the constraints $y_j(s) \in \{0, 1\}$ have been relaxed to $y_j(s) \in [0, 1]$ for all $j \in \{1, \dots, n_\mu\}$ and all $s \in [0, s_f]$. For implicit approaches of Section 4 relaxation refers to the use of the smoothed switching functions $\sigma_j(v) \in [0, 1]$. Typically, relaxed solutions have fractionality and infeasibility larger zero.

Definition 12 (Integer solutions) We call a solution *integer solution*, if a relaxed solution is rounded by setting the largest of all n_μ entries of either $\mathbf{y}(s)$ or $\boldsymbol{\sigma}(s)$ to 1, all others to 0. The continuous controls and states are optimized in a second optimization run to obtain feasibility with respect to all constraints in (10). For the implicit approach, the additional constraints

$$\sum_{j=1}^{n_\mu} \sigma_j(s)\bar{v}_j \leq v(s) \leq \sum_{j=1}^{n_\mu} \sigma_j(s)\bar{v}_{j+1}$$

are included. If integer solutions are feasible, they have a fractionality and infeasibility of zero.

Note that different strategies to obtain integer solutions from relaxed solutions for MIOCPs have been proposed in the literature, such as combinatorial integral approximation [34, 59]. We do not apply and compare these strategies here to focus on the main aspect, the tightness of relaxations. We shortly summarize all formulations for the truck problem (10) that are compared in Table 1.

IC Inner Convexification as described in Section 3.1 has differential states v, Q and controls $M_{\text{ind}}, \bar{M}_{\text{ind}}, \mathbf{y}$. The dynamics are specified by (11), the indicator constraints are given by (12).

OC Outer Convexification (Section 3.2) is identical to IC, but with dynamics (13) and indicator constraints (14).

bigM The Big-M approach (Section 3.3) is identical to OC, but with indicator constraints (14).

relVC The relaxed Vanishing Constraint approach (Section 3.4) is identical to OC, but with indicator constraints (16) that are relaxed by ε . The value of ε is reduced by a homotopy from 10^5 to 10^{-4} in multiplicative steps of 0.6, using solutions as initialization for the next problem. For details on the homotopy, see [33].

smoVC The smoothed Vanishing Constraint approach (Section 3.5) is identical to relVC (also with a similar homotopy), but with indicator constraints (16) that are ε -smoothened using an NCP function.

Table 1 Results of all approaches, compare Section 6.3 for explanation. Fractionality and infeasibility of the integer solutions are 0 by construction. Dynamic Programming results directly in integer controls. (i infeasible, m maximum number of iterations exceeded, p presolve)

		Scenario 1						Scenario 2					
relaxed		Integer			runtime			relaxed			Integer		
obj.	frac.	infeas.	obj.	infeas.	runtime	obj.	frac.	infeas.	obj.	frac.	infeas.	obj.	runtime
Explicit Approaches													
IC	50,717	0.172	589	0.172	4 s	243, 109 ⁱ	0.265	803	3,604,660	0.265	803	1, 326, 750 ⁱ	14 s
OC	55,451	0.442	1.96	0.442	17 s	58,133	0.308	6.36	3,797,820	0.308	6.36	3,903,890	37 s
bigM	49,031	0.7	53.5	0.7	13 s	62,762	0.748	187	3,398,250	0.748	187	6, 634, 100 ⁱ	48 s
PFL	115, 541 ^m	0.0796	63.5	0.0796	51.7 min	180, 000 ^p	0.113	64.4	9, 076, 590 ^m	0.113	64.4	180, 000 ^p	50.7 min
PPL	3,273	0.865	3.54e+03	0.865	8.0 min	67, 407 ⁱ	0.709	3.34e+03	163,557	0.709	3.34e+03	879, 767 ⁱ	4.9 min
PML	40, 697 ^m	0.499	3.98e+03	0.499	46.5 min	18,244	0.275	5.59e+03	121,777	0.275	5.59e+03	145,696	11.2 min
reIVC	57,551	0.036	1.5e-05	0.036	1.7 min	57,633	0	1.5e-05	3,838,600	0	1.5e-05	3,838,600	1.5 min
smoVC	57,031	0.033	0	0.033	2.1 min	57,081	0	1e-06	3,813,970	0	1e-06	3,813,970	1.7 min
Implicit Approaches													
\bar{V}_{Mindmax}	63,470	0.0281	2.16	0.0281	55 s	63,487	0.00606	2.47	3,944,800	0.00606	2.47	3,974,060	27 s
\bar{V}_{ave}	72,053	0.0153	2.03	0.0153	41 s	72,236	0.0122	2.51	3,745,560	0.0122	2.51	3,749,310	41 s
\bar{V}_Q	103,840	0.0256	1.94	0.0256	46 s	101,303	0.00415	2.53	3,707,590	0.00415	2.53	3,718,560	16 s
\bar{V}_{plus}	70,980	0.0611	2.04	0.0611	1.1 min	70,997	0.00415	2.55	3,741,790	0.00415	2.55	3,749,540	1.4 min
\bar{V}_{pp}	104, 946 ⁱ	0.0577	22.4	0.0577	55 s	100, 227 ⁱ	0.00377	2.57	3,761,350	0.00377	2.57	3,768,480	1.3 min
\bar{V}_{ppp}	99, 649 ⁱ	0.0191	204	0.0191	33 s	0, 000 ^p	0.0301	8.72	4, 130, 030 ⁱ	0.0301	8.72	4, 277, 100 ⁱ	22 s

Table 1 (continued)

	Scenario 1				Scenario 2				
	relaxed		Integer		relaxed		Integer		
	obj.	frac.	infeas.	runtime	obj.	frac.	infeas.	runtime	
Dynamic Programming									
$\Delta v = 0.1$	$\Delta M = 20$			60,580				8.7 min	4,165,790
$\Delta v = 0.05$	$\Delta M = 25$			60,038				11.1 min	3,947,260
$\Delta v = 0.05$	$\Delta M = 20$			60,026				17.1 min	3,957,580
$\Delta v = 0.05$	$\Delta M = 10$			59,879				67.6 min	3,919,570
$\Delta v = 0.01$	$\Delta M = 20$			60,666				90.2 min	3,907,910

PFL, PPL, PML The Perspective approaches (Sections 3.6, 3.7, and 3.8) are obtained by lifting the Outer Convexification formulation and, for PPL and PML, by subsequent aggregation.

Implicit Approaches All implicit approaches (Section 4) use an OC formulation of the dynamics and of the constraints, substituting y_j by $\sigma_j(v(s))$ in (13) and (14). They differ in the choice of the switching velocity sets \bar{V}_{Mindmax} , \bar{V}_{ave} , \bar{V}_Q , \bar{V}_{plus} , \bar{V}_{pp} , and \bar{V}_{ppp} .

Dynamic Programming Five different discretizations Δv of the state v and ΔM of the engine torque M were applied for an enumeration of the admissible state space $X \subset \mathbb{R}^{n_x}$ and control space $U \subset \mathbb{R}^{n_u}$ (Section 5).

Further combinations like using OC dynamics (13) and IC constraints (12), or using implicit approaches based on IC dynamics and constraints are possible. They are not promising, though, as Fig. 5 shows for the case of implicit switching.

6.4 Discussion

The numerical results for the different approaches and the two scenarios are shown in Table 1 and Figs. 6, 7, 8, and 9. The figures show relaxed gear choices using grey intensities, overlaid with the resulting velocity profiles.

Concerning computational times, IC, OC, and bigM are close to being real-time feasible with a CPU times on the order of a few seconds. The other explicit approaches and implicit approaches show CPU times around one minute. The perspective formulation PFL could not be brought to convergence for both Scenarios evaluated. A possible explanation for this adverse behavior of the nonlinear optimization method may be found in the ill-posedness of the perspective of a nonlinear nonconvex function near the points $y = 0$, which appears to persist even after smoothing. Aggregation of variables helps to ameliorate the situation, as is seen in Scenario 2 where both aggregated variants PPL and PML converge after several minutes. Expectedly, the runtimes of Dynamic Programming increase exponentially due to the curse of dimensionality.

Looking at the relaxed solutions, bigM and IC result in small objective function values at the price of high values for fractionality and infeasibility. Looking at the relaxed gear choices in Figs. 6 and 8, one observes the unphysical combination of low and high gears which result in very poor performance when rounding. This is due to large deviations between relaxed and rounded velocity (compare Figs. 6 and 8). This situation is much improved for OC, where both velocities are similar. This result is also expected from the supporting approximation theorem, c.f. [57].

Slightly higher objective function values for the relaxed problems in smoVC and relVC are justified by very small fractionality and numerically zero infeasibility. The obtained

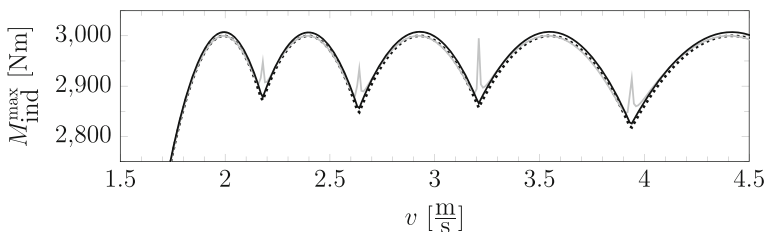


Fig. 5 $M_{\text{ind}}^{\text{max}}$ as a function of velocity with exact switching (.....), IC-like switching (—) and OC-like switching (—), using \bar{V}_{Mindmax} . IC leads to undesirable spikes

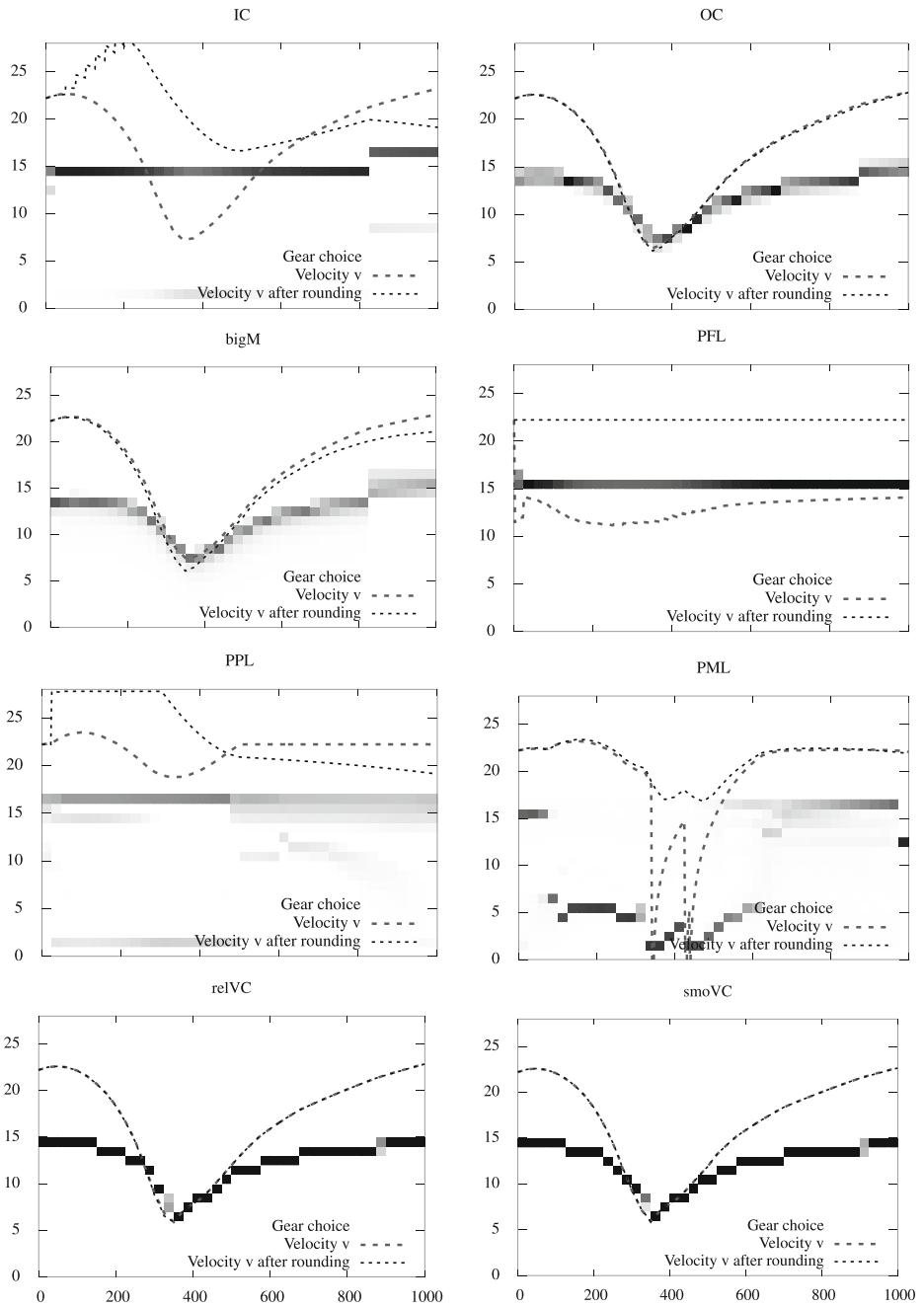


Fig. 6 Velocities and gear choices for scenario 1 and explicit approaches

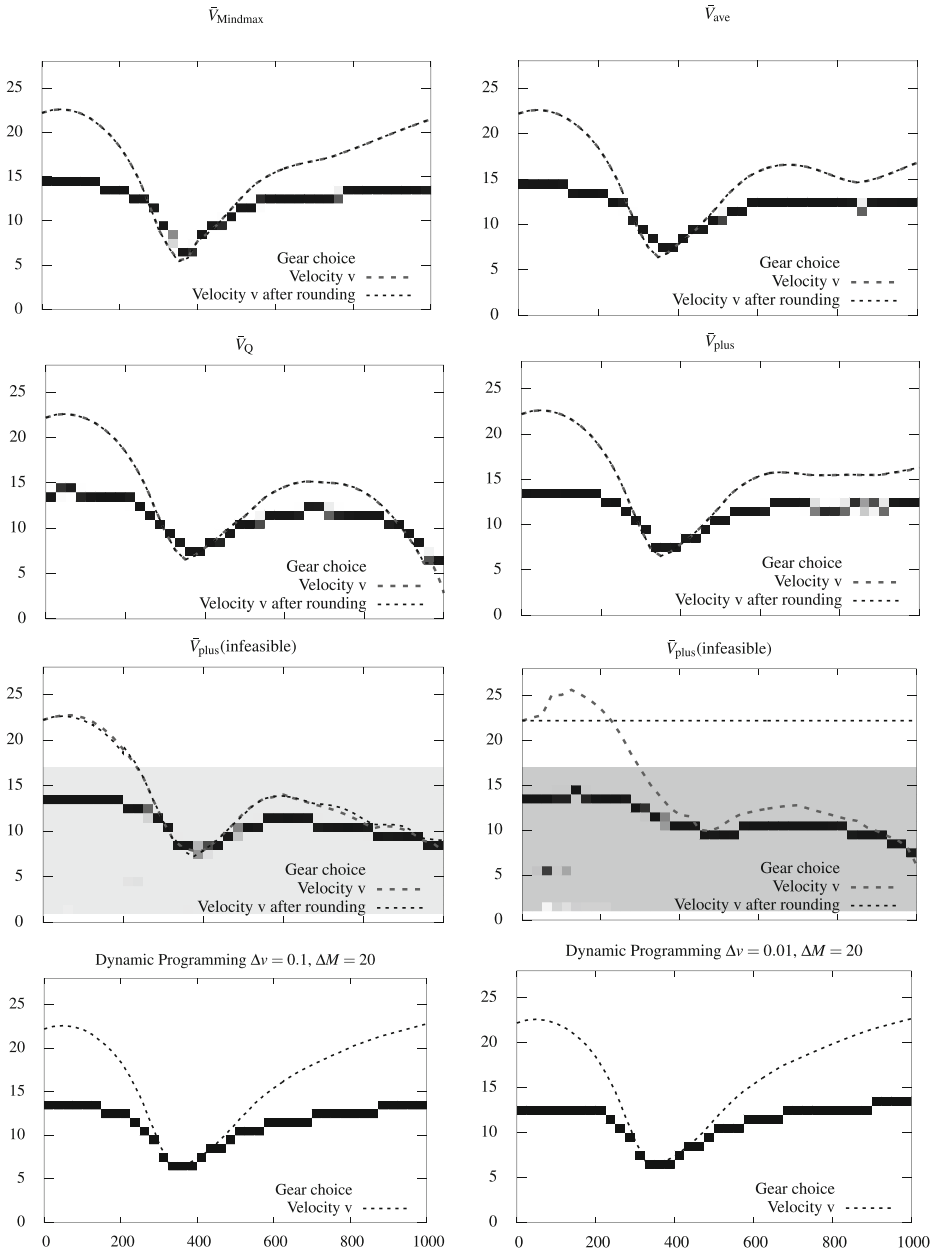


Fig. 7 Velocities and gear choices for scenario 1 and implicit and two DP approaches

solutions come at the expense of slightly increased runtime due to the homotopy approach for smoothing-relaxation. Most importantly, they can be rounded to integer feasible solutions with an almost identical function value. These solution approaches are ranked best among the explicit ones for both Scenarios.

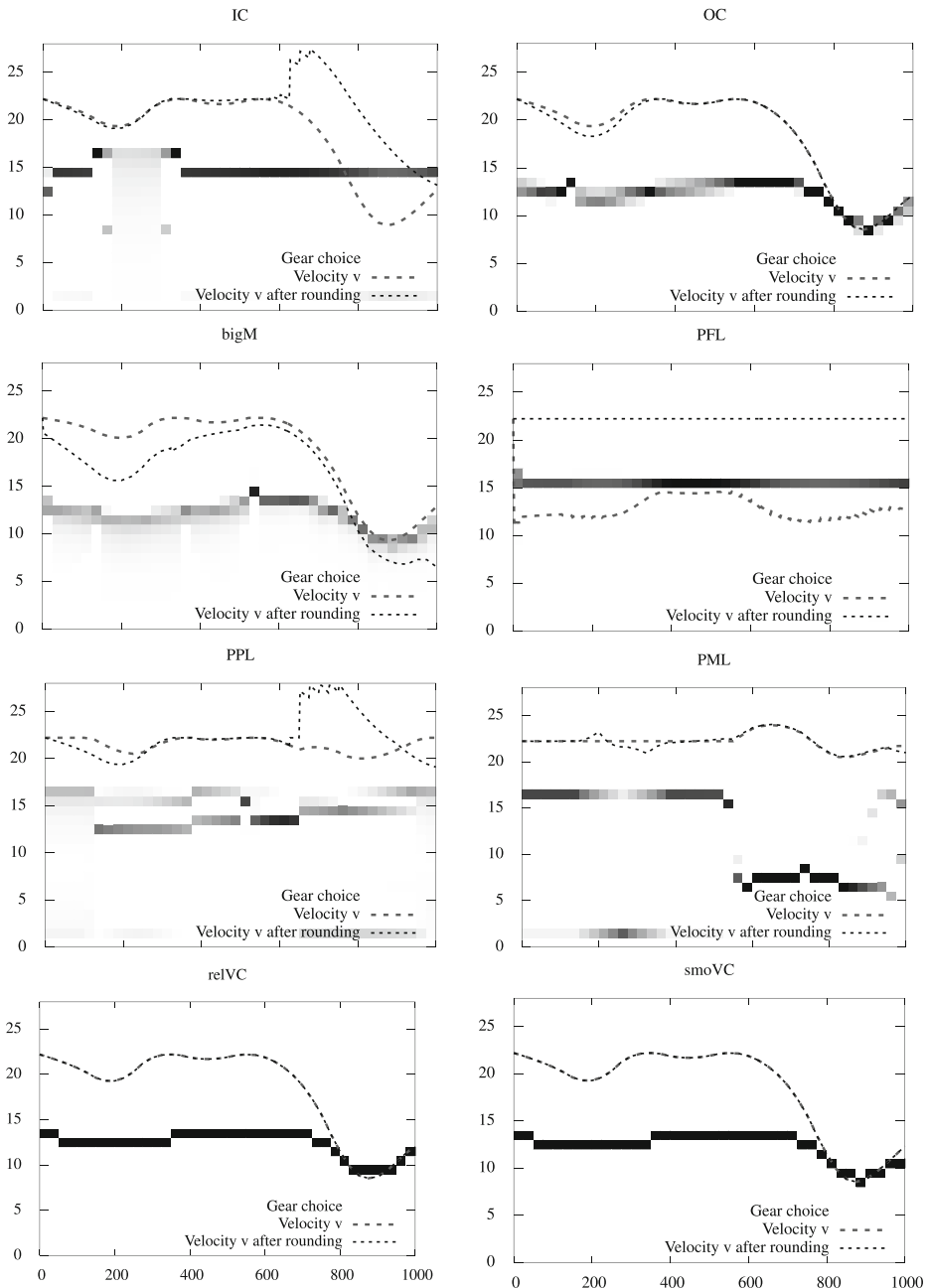


Fig. 8 Velocities and gear choices for scenario 2 and explicit approaches

For the implicit approaches, rounded solutions are similar to the relaxed ones. Concerning performance, however, a more diffuse picture emerges. For scenario 1, two approaches lead to convergence to infeasible points, and none of the objective function values is

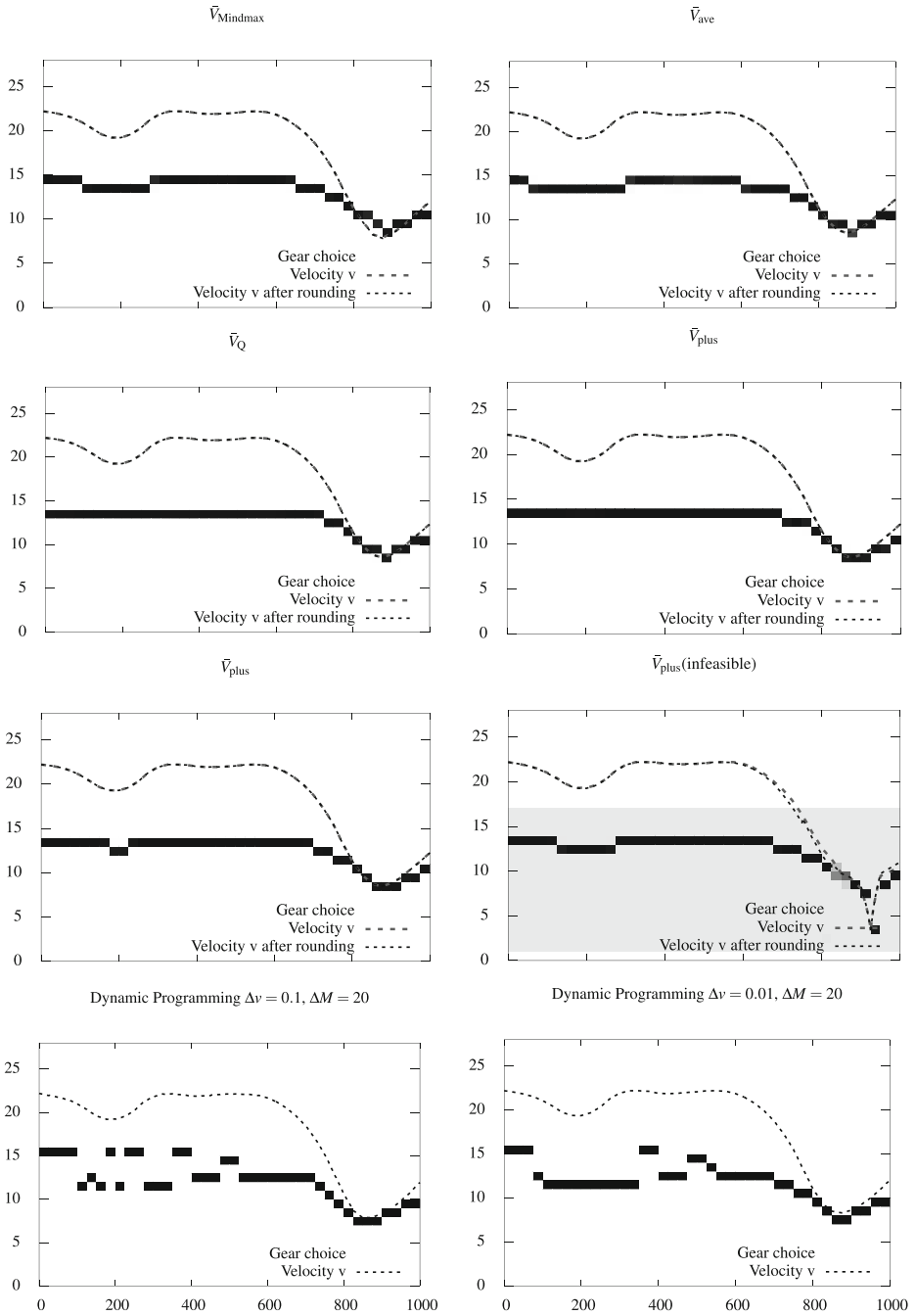


Fig. 9 Velocities and gear choices for scenario 2 and implicit and two DP approaches

competitive. For scenario 2, however, the four lowest objective function values result from implicit approaches. The relaxed solutions are slightly infeasible (infeas. ≈ 2.5), though.

The solution quality of Dynamic Programming results strongly depends on the underlying discretization grid. The enumerative nature of the subproblems and the curse of dimensionality impair using discretizations that are fine enough to compete with the local optimization approaches. In addition, as shown in Fig. 4, the solutions come with an additional mismatch when exact forward simulation is used.

7 Conclusion

An explicit approach using outer convexification for the dynamics and a good way to treat the vanishing constraints outperformed implicit formulations and dynamic programming concerning computational time, performance, and robustness of solution quality. Better algorithms to numerically solve optimization problems with vanishing constraints need to be developed to reliably harness these advantages also in larger scale applications.

Acknowledgements The work reported in this article was conducted when S. Sass was with Institut für Mathematische Optimierung, Otto-von-Guericke-Universität Magdeburg.

Funding information This study received funding from Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 314838170, GRK 2297 MathCoRe and Priority Programme 1962 “Non-smooth and Complementarity-based Distributed Parameter Systems: Simulation and Hierarchical Optimization,” grant no. KI1839/1-1; the German Federal Ministry of Education and Research, program “Mathematics for Innovations in Industry and Service,” grants no. 05M17MBA-MoPhaPro, 05M18MBA-MoRENet; and program “IKT 2020: Software Engineering,” grant no. 61210304-ODINE. Dynamic programming results were obtained using an implementation by Alexander Buchner.

References

1. Abichandani, P., Benson, H.Y., Kam, M.: Multi-vehicle path coordination under communication constraints. In: 2008 American Control Conference, pp. 650–656. IEEE (2008)
2. Achtziger, W., Kanzow, C.: Mathematical programs with vanishing constraints: optimality conditions and constraint qualifications. *Math. Program. Ser. A* **114**, 69–99 (2008)
3. Anitescu, M., Tseng, P., Wright, S.J.: Elastic-mode algorithms for mathematical programs with equilibrium constraints: global convergence and stationarity properties. *Math. Program. Ser. A* **110**, 337–371 (2007)
4. Balas, E.: Disjunctive programming and a Hierarchy of relaxations for discrete optimization problems. *SIAM J. Algebraic Discrete Methods* **6**, 466–486 (1985)
5. Baumrucker, B.T., Biegler, L.T.: MPEC Strategies for optimization of a class of hybrid dynamic systems. *J. Process Control* **19**, 1248–1256 (2009)
6. Bellman, R.: The theory of dynamic programming. *Bull. Am. Math. Soc.* **60**, 503–515 (1954)
7. Bellman, R.E.: *Dynamic Programming*, 6th edn. University Press, Princeton (1957)
8. Belotti, P., Kirches, C., Leyffer, S., Linderoth, J.T., Luedtke, J., Mahajan, A.: Mixed-integer nonlinear optimization. *Acta Numer.* **22**, 1–131 (2013)
9. Bertsekas, D.P.: *Dynamic Programming and Optimal Control*, 3rd edn., vol. 1. Athena Scientific, Belmont (2005)
10. Bertsekas, D.P.: *Dynamic Programming and Optimal Control*, 4th edn., vol. 2. Athena Scientific, Belmont (2012)
11. Bock, H.G., Kirches, C., Meyer, A., Potschka, A.: Numerical solution of optimal control problems with explicit and implicit switches. *Optim. Methods Softw.* **33**, 450–474 (2018)
12. Buchner, A.: *Auf Dynamischer Programmierung Basierende Nichtlineare Modellprädiktive Regelung Für LKW*. Diploma thesis, Ruprecht-Karls-Universität Heidelberg (2010)

13. Burger, M., Gerdts, M., Göttlich, S., Herty, M.: Dynamic programming approach for discrete-valued time discrete optimal control problems with dwell time constraints. In: Bociu, L., Désidéri, J.-A., Habbal, A. (eds.) *System Modeling and Optimization. CSMO: IFIP Conference on System Modeling and Optimization. IFIP Advances in Information and Communication Technology*, vol. 494, pp. 159–168. Springer, Cham (2015)
14. Burgschweiger, J., Gnädig, B., Steinbach, M.C.: Nonlinear programming techniques for operative planning in large drinking water networks. *Open Appl. Math. J.* **3**, 14–28 (2009)
15. Ceria, S., Soares, J.: Convex programming for disjunctive convex optimization. *Math. Program. Ser. A* **86**, 595–614 (1999)
16. Claeys, M., Daafouz, J., Henrion, D.: Modal occupation measures and LMI relaxations for nonlinear switched systems control. *Automatica* **64**, 143–154 (2016)
17. Facchinei, F., Jiang, H., Qi, L.: A smoothing method for mathematical programs with equilibrium constraints. *Math. Program. Ser. A* **85**, 107–134 (1999)
18. Fang, H.-R., Leyffer, S., Munson, T.S.: A pivoting algorithm for linear programming with linear complementarity constraints. *Optim. Methods Softw.* **27**, 89–114 (2012)
19. Fletcher, R., Leyffer, S.: Solving mathematical programs with complementarity constraints as nonlinear programs. *Optim. Methods Softw.* **19**, 15–40 (2004)
20. Fourer, R., Gay, D.M., Kernighan, B.W.: *AMPL: A Modeling Language for Mathematical Programming*. Duxbury Press, USA (2002)
21. Frangioni, A., Gentile, C.: Perspective cuts for a class of convex 0–1 mixed integer programs. *Math. Program. Ser. A* **106**, 225–236 (2006)
22. Fügenschuh, A., Herty, M., Klar, A., Martin, A.: Combinatorial and continuous models for the optimization of traffic flows on networks. *SIAM J. Optim.* **16**, 1155–1176 (2006)
23. Fukushima, M., Qi, L. (eds.): *Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods*. Kluwer Academic, Dordrecht (1999)
24. Gerdts, M.: Solving mixed-integer optimal control problems by branch&bound: a case study from automobile test-driving with gear shift. *Optim. Control Appl. Methods* **26**, 1–18 (2005)
25. Gerdts, M.: A variable time transformation method for mixed-integer optimal control problems. *Optim. Control Appl. Methods* **27**, 169–182 (2006)
26. Gerdts, M., Sager, S.: Mixed-integer DAE optimal control problems: Necessary conditions and bounds. In: Biegler, L., Campbell, S.L., Mehrmann, V. (eds.) *Control and Optimization with Differential-Algebraic Constraints*, Chapter 9, pp. 189–212. SIAM (2012)
27. Grossmann, I.E.: Review of nonlinear mixed-integer and disjunctive programming techniques. *Optim. Eng.* **3**, 227–252 (2002)
28. Gugat, M., Herty, M., Klar, A., Leugering, G.: Optimal control for traffic flow networks. *J. Optim. Theory Appl.* **126**, 589–616 (2005)
29. Günlük, O., Linderoth, J.T.: Perspective reformulations of mixed integer nonlinear programs with indicator variables. *Math. Program. B* **124**, 183–205 (2010)
30. Hellström, E., Ivarsson, M., Aslund, J., Nielsen, L.: Look-ahead control for heavy trucks to minimize trip time and fuel consumption. *Control Eng. Pract.* **17**, 245–254 (2009)
31. Hoheisel, T.: *Mathematical Programs with Vanishing Constraints*. PhD thesis, Julius–Maximilians–Universität Würzburg (2009)
32. Jung, M.: *Relaxations and Approximations for Mixed-Integer Optimal Control*. PhD thesis, University Heidelberg (2013)
33. Jung, M.N., Kirches, C., Sager, S.: On perspective functions and vanishing constraints in mixed-integer nonlinear optimal control. In: Jünger, M., Reinelt, G. (eds.) *Facets of Combinatorial Optimization: Festschrift for Martin Grötschel*, pp. 387–417. Springer, Berlin (2013)
34. Jung, M.N., Reinelt, G., Sager, S.: The Lagrangian relaxation for the combinatorial integral approximation problem. *Optim. Methods Softw.* **30**, 54–80 (2015)
35. Kawajiri, Y., Biegler, L.T.: Nonlinear programming superstructure for optimal dynamic operations of simulated moving bed processes. *Ind. Eng. Chem. Res.* **45**, 8503–8513 (2006)
36. Kirches, C.: *Fast Numerical Methods for Mixed-Integer Nonlinear Model-Predictive Control. Advances in Numerical Mathematics*. Springer Vieweg, Wiesbaden (2011)
37. Kirches, C., Bock, H.G., Leyffer, S.: Modeling mixed-integer constrained optimal control problems in AMPL. In: Breitenacker, F., Troch, I. (eds.) *Proceedings of MATHMOD 2012, Vienna (2012)*. ARGESIM Report No. S38
38. Kirches, C., Bock, H.G., Schlöder, J.P., Sager, S.: Mixed-integer NMPC for predictive cruise control of heavy-duty trucks. In: *2013 European Control Conference (ECC)*, pp. 4118–4123 (2013)
39. Kirches, C., Lenders, F.: Approximation properties and tight bounds for constrained mixed-integer optimal control. *Optimization Online (Technical Report) 5404* (2015)

40. Kirches, C., Leyffer, S.: TACO: A toolkit for AMPL control optimization. *Math. Program. Comput.* **5**, 227–265 (2013)
41. Kirches, C., Potschka, A., Bock, H.G., Sager, S.: A parametric active-set method for QPs with vanishing constraints arising in a robot motion planning problem. *Pac. J. Optim.* **9**, 275–299 (2013)
42. Kirches, C., Sager, S., Bock, H.G., Schlöder, J.P.: Time-optimal control of automobile test drives with gear shifts. *Optim. Control Appl. Methods* **31**, 137–153 (2010)
43. Kirches, C., Wirsching, L., Bock, H.G., Schlöder, J.P.: Efficient direct multiple shooting for nonlinear model predictive control on long horizons. *J. Process Control* **22**, 540–550 (2012)
44. Kirchner, C., Herty, M., Göttlich, S., Klar, A.: Optimal control for continuous supply network models. *Netw. Heterog. Media* **1**, 675–688 (2007)
45. Koch, T., Hiller, B., Pfetsch, M.E., Schewe, L. (eds.): Evaluating Gas Network Capacities. SIAM-MOS series on Optimization. SIAM (2015)
46. Leyffer, S.: Complementarity constraints as nonlinear equations: Theory and numerical experiences. In: Dempe, S., Kalashnikov, V. (eds.) *Optimization with Multivalued Mappings*. Springer Optimization and Its Applications, vol. 2, pp. 169–208. Springer, Boston, MA (2006)
47. Leyffer, S., Munson, T.S.: A globally convergent filter method for MPECs. Preprint ANL/MCS-p1457-0907, Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439 USA (2007)
48. Moehle, N., Boyd, S.: A perspective-based convex relaxation for switched-affine optimal control. *Syst. Control Lett.* **86**, 34–40 (2015)
49. Oldenburg, J., Marquardt, W.: Disjunctive modeling for optimal control of hybrid systems. *Comput. Chem. Eng.* **32**, 2346–2364 (2008)
50. Palagachev, K., Gerdt, M.: Mathematical programs with blocks of vanishing constraints arising in discretized mixed-integer optimal control problems. *Set-valued Var. Anal.* **23**, 149–167 (2015)
51. Prata, A., Oldenburg, J., Kroll, A., Marquardt, W.: Integrated scheduling and dynamic optimization of grade transitions for a continuous polymerization reactor. *Comput. Chem. Eng.* **32**, 463–476 (2008)
52. Raghunathan, A.U., Biegler, L.T.: Mathematical programs with equilibrium constraints (MPECs) in process engineering. *Comput. Chem. Eng.* **27**, 1381–1392 (2003)
53. Raghunathan, A.U., Diaz, M.S., Biegler, L.T.: An MPEC formulation for dynamic optimization of distillation operations. *Comput. Chem. Eng.* **28**, 2037–2052 (2004)
54. Ralph, D., Wright, S.J.: Some properties of regularization and penalization schemes for MPECs. *Optim. Methods Softw.* **19**, 527–556 (2004)
55. Sager, S.: *Numerical Methods for Mixed-Integer Optimal Control Problems*. Der Andere Verlag, Lübeck (2005)
56. Sager, S.: A Benchmark library of mixed-integer optimal control problems. In: Lee, J., Leyffer, S. (eds.) *Mixed Integer Nonlinear Programming*. The IMA Volumes in Mathematics and Its Applications, vol. 154, pp. 169–208. Springer, New York (2012)
57. Sager, S., Bock, H.G., Diehl, M.: The integer approximation error in mixed-integer optimal control. *Math. Program. Ser. A* **133**, 1–23 (2012)
58. Sager, S., Claeys, M., Messine, F.: Efficient upper and lower bounds for global mixed-integer optimal control. *J. Glob. Optim.* **61**, 721–743 (2015)
59. Sager, S., Jung, M., Kirches, C.: Combinatorial integral approximation. *Math. Methods Oper. Res.* **73**, 363–380 (2011)
60. Sager, S., Bock, H.G., Reinelt, G.: Direct methods with maximal lower bound for mixed-integer optimal control problems. *Math. Program.* **118**, 109–149 (2009)
61. Sawaya, N.W., Grossmann, I.E.: Computational implementation of non-linear convex hull reformulation. *Comput. Chem. Eng.* **31**, 856–866 (2007)
62. Scholtes, S.: Convergence properties of a regularization scheme for mathematical programs with complementarity constraints. *SIAM J. Optim.* **11**, 918–936 (2001)
63. Scholtes, S.: Nonconvex structures in nonlinear programming. *Oper. Res.* **52**, 368–383 (2004)
64. Sherali, H.D.: RLT: A unified approach for discrete and continuous nonconvex optimization. *Ann. Oper. Res.* **149**, 185–193 (2007)
65. Sonntag, C., Stursberg, O., Engell, S.: Dynamic optimization of an industrial evaporator using graph search with embedded nonlinear programming. In: Cassandras, C., et al. (eds.) *Analysis and Design of Hybrid System 2006*. IPV-IFAC Proceedings Volume, pp. 211–216. Elsevier (2006)
66. Stein, O., Oldenburg, J., Marquardt, W.: Continuous reformulations of discrete–continuous optimization problems. *Comput. Chem. Eng.* **28**, 1951–1966 (2004)
67. Stubbs, R.A., Mehrotra, S.: Generating convex polynomial inequalities for mixed 0–1 programs. *J. Glob. Optim.* **24**, 311–332 (2002)

68. Terwen, S., Back, M., Krebs, V.: Predictive powertrain control for heavy duty trucks. In: Proceedings of IFAC Symposium in Advances in Automotive Control, pp. 451–457, Salerno, Italy (2004)
69. Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program.* **106**, 25–57 (2006)

Affiliations

Michael N. Jung¹ · Christian Kirches²  · Sebastian Sager³ · Susanne Sass⁴

Sebastian Sager
sager@ovgu.de

Susanne Sass
susanne.scholl@avt.rwth-aachen.de

- ¹ Interdisziplinäres Zentrum für Wissenschaftliches Rechnen (IWR), Ruprecht-Karls-Universität Heidelberg, Im Neuenheimer Feld 205, 69120 Heidelberg, Germany
- ² Institut für Mathematische Optimierung, Technische Universität Carolo-Wilhelmina zu Braunschweig, Universitätsplatz 2, 38106 Braunschweig, Germany
- ³ Institut für Mathematische Optimierung, Otto-von-Guericke-Universität Magdeburg, Universitätsplatz 2, 39106 Magdeburg, Germany
- ⁴ Aachener Verfahrenstechnik, Rheinisch-Westfälische Technische Hochschule Aachen, Forckenbeckstraße 51, 52074 Aachen, Germany