Dynamic Programming Approach for Discrete-Valued Time Discrete Optimal Control Problems with Dwell Time Constraints

Michael Burger[1], Matthias Gerdts[2(✉)], Simone Göttlich[3], and Michael Herty[4]

[1] Abteilung Mathematische Methoden in Dynamik und Festigkeit MDF,
Fraunhofer-Institut für Techno- und Wirtschaftsmathematik ITWM,
Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany

[2] Institut für Mathematik und Rechneranwendung (LRT), Universität der
Bundeswehr München, Werner-Heisenberg-Weg 39, 85577 Neubiberg, Germany
matthias.gerdts@unibw.de

[3] Department of Mathematics, University of Mannheim, 68131 Mannheim, Germany

[4] Department of Mathematics, RWTH Aachen,
Templergraben 55, 52056 Aachen, Germany

Abstract. The article discusses a numerical approach to solve optimal control problems in discrete time that involve continuous and discrete controls. Special attention is drawn to the modeling and treatment of dwell time constraints. For the solution of the optimal control problem in discrete time, a dynamic programming approach is employed. A numerical example is included that illustrates the impact of dwell time constraints in mixed integer optimal control.

Keywords: Dynamic programming · Mixed-integer optimization · Dwell time constraints

1 Introduction

Mixed-integer optimal control is a field of increasing importance as practical applications often include discrete decisions in addition to continuous-valued control variables. Examples of such problems can be found, e.g., in [1,2,4,5,8,14,15,18]. One way to approach mixed-integer optimal control problems is by solving necessary optimality conditions provided by the well-known maximum principle. These necessary conditions are valid even for discrete control sets. In [6] a graph-based solution method exploiting the maximum principle was developed, which is however limited to single-state problems. The drawback of maximum principle based methods is that a very good initial guess of the switching structure is needed which is often not available for practical applications. Direct discretization methods based on variable time transformations as in [1,2,9–11,18,19], or direct discretization methods based on relaxations and sum-up-rounding strategies as in [13,16,17] have shown their ability to solve difficult

L. Bociu et al. (Eds.): CSMO 2015, IFIP AICT 494, pp. 159–168, 2016.
DOI: 10.1007/978-3-319-55795-3_14

real-world examples. Extensions towards mixed control-state constraints, which depend on the discrete variable and lead to vanishing constraints, can be found in [7, Chap. 5] and [12]. The dynamic programming approach in [3, Chap. 7] allows to consider switching costs to avoid frequent switches of the discrete-valued controls. This paper particularly addresses so-called dwell time constraints in mixed-integer optimal control problems in discrete time. Dwell times are an important aspect in many engineering applications, especially in the context of switched systems. After a switch has occured, the physical process typically requires a certain time period to recover and to return to normal operation. During this dwell time the process is limited in its operation and follows specified dynamics. An example for such a behavior is a truck with gear shifts. As the clutch is engaged, the motor torque does not arrive at the wheels for a short time period. Only after the clutch is released again, the motor torque is distributed at the wheels. Throughout the paper, we focus on optimal control problems in discrete time, which exist in their own rights but often they are obtained as discretizations of optimal control problems in continuous time. The purpose of the paper is twofold. Firstly, a model taking into account dwell time is suggested. Herein, we will use delays in the dynamics. Secondly, it is shown that a dynamic programming principle applies and can be used for numerical computations, if the state dimension is low. The paper is organized as follows. Section 2 defines the mixed-integer optimal control problem in discrete time with dwell time constraints. Section 3 discusses a dynamic programming approach to solve the problems. Section 4 discusses an illustrative numerical example.

2 Modeling Dwell Time Constraints in Mixed-Integer Optimal Control in Discrete Time

Let a fixed grid $\mathbb{G}_N = \{t_0 < t_1 < \ldots < t_N\}$ with $N \in \mathbb{N}$ be given. For $i = 0, \ldots, N$ and $x \in \mathbb{R}^{n_x}$ let $\emptyset \neq \mathcal{X}(t_i) \subseteq \mathbb{R}^{n_x}$ and $\emptyset \neq \mathcal{U}(t_i, x) \subseteq \mathbb{R}^{n_u}$ be closed connected sets and

$$\mathcal{V} = \{v^1, \ldots, v^M\}$$

a discrete set of vectors $v^j \in \mathbb{R}^{n_v}$, $j = 1, \ldots, M$. On the grid $\mathbb{G}_N$, grid functions are introduced. The grid function $x : \mathbb{G}_N \longrightarrow \mathbb{R}^{n_x}$ is called state and it is restricted by the state constraints $x(t_i) \in \mathcal{X}(t_i)$, $i = 0, \ldots, N$. The grid function $u : \mathbb{G}_N \longrightarrow \mathbb{R}^{n_u}$ is called real-valued control and it is restricted by the control constraints $u(t_i) \in \mathcal{U}$, $i = 0, \ldots, N$. The grid function $v : \mathbb{G}_N \to \mathbb{R}^{n_v}$ is called discrete-valued control and it is restricted by the constraints $v(t_i) \in \mathcal{V}$, $i = 0, \ldots, N$. We say the discrete-valued control switches at time point $t_i \in \mathbb{G}_N$ with $i \in \{0, \ldots, N-1\}$, if $v(t_i) \neq v(t_{i+1})$. Switches in the grid function v can be measured with the help of the discrete variation $dv : \mathbb{G}_N \to \mathbb{R}^{n_v}$ defined by

$$dv(t_i) = \begin{cases} v(t_{i+1}) - v(t_i), & \text{for } i = 0, \ldots, N-1, \\ v(t_N) - v(t_{N-1}), & \text{for } i = N. \end{cases}$$

A jump of v at t_i occurs if and only if $dv(t_i) \neq 0$. In order to define dwell time constraints let $L \in \mathbb{N}$ denote a number of steps and $\bar{u} \in \mathcal{U}$ a fixed control vector.

Definition 1. *A dwell time constraint of time horizon length $L \in \mathbb{N}$ and control $\bar{u}$ applies, if and only if the following conditions are met:*

(a) v switches at t_i for some $i \in \{0, \dots, N-1\}$ and
(b) $u(t_{i+\ell}) = \bar{u}$ for $\ell = 1, \dots, L$.

In other words, a dwell time constraint is a logical implication of type

$$v(t_i) \neq v(t_{i+1}) \quad \Longrightarrow \quad u(t_{i+\ell}) = \bar{u} \text{ for } \ell = 1, \dots, L,$$

i.e. if the discrete control switches, then control u is fixed to $\bar{u}$ for a defined future time horizon of length L.

Remark 1. For notational simplicity we fixed the full control vector u in Definition 1 (b) to a specified value $\bar{u}$ for all time points $t_{i+\ell}$, $\ell = 1, \dots, L$. This can be generalized such that only some components of u are fixed to specified values which may even vary with the index ℓ. This more general setting would cause a more technical notation, but otherwise can be treated with the same techniques as below.

Let the state evolve according to the discrete dynamics

$$x(t_{i+1}) = f(t_i, x(t_i), u(t_i), v(t_i)), \qquad i = 0, 1, \dots, N-1.$$

Imposing a dwell time constraint of length L and value $\bar{u}$ implies that the future state depends not just on the current control value but on the history of the control v, i.e.

$$\begin{aligned} u(t_i) \; &\in \bar{\mathcal{U}}(v(t_i), v(t_{i-1}), \dots, v(t_{i-L})) \\ &:= \begin{cases} \mathcal{U}, & \text{if } v(t_{i-\ell}) = v(t_{i-\ell-1}) \text{ for } \ell = 0, 1, \dots, L-1, \\ \{\bar{u}\}, & \text{otherwise.} \end{cases} \end{aligned}$$

Herein and throughout the paper we use the convention

$$v(t_{-\ell}) := v(t_0) \qquad \text{for } \ell = 1, \dots, L. \tag{1}$$

We arrived at the following mixed-integer optimal control problem in discrete time with a dwell time constraint of length L with value $\bar{u}$:

Probelm 1 (DMIOCP).
Minimize

$$F(x, u, v) := \varphi(x(t_N)) + \sum_{i=0}^{N-1} f_0(t_i, x(t_i), u(t_i), v(t_i))$$

subject to the constraints

$$\begin{aligned} x(t_{i+1}) &= f(t_i, x(t_i), u(t_i), v(t_i)), && i = 0, 1, \dots, N-1, \\ x(t_0) &= x_0, \\ x(t_i) &\in \mathcal{X}(t_i), && i = 0, 1, \dots, N, \\ u(t_i) &\in \bar{\mathcal{U}}(v(t_i), v(t_{i-1}), \dots, v(t_{i-L})), && i = 0, 1, \dots, N-1, \\ v(t_i) &\in \mathcal{V}, && i = 0, 1, \dots, N-1. \end{aligned}$$

Note that the control $v(t_N)$ does not have any influence in the problem. Consequently, only switches at the time points $t_0, \ldots, t_{N-1}$ are taken into account. In the sequel we will make use of the convention $v_N = v_{N-1}$ whenever useful.

3 A Dynamic Programming Approach

DMIOCP is embedded into a family of perturbed problems as follows. To this end we use the notation $v_{(k)}^L := (v_{k-1}, \ldots, v_{k-L})$ for $L \in \mathbb{N}$ and some index k with values $v_{k-\ell} \in \mathcal{V}$ for $\ell = 1, \ldots, L$. We denote with $\mathcal{V}^L$ the cartesian product $\mathcal{V} \times \cdots \times \mathcal{V}$ (L times).

Probelm 2 (DMIOCP($t_k, x_k, v_{(k)}^L$)).
For a given index $t_k \in \mathbb{G}_N$, $x_k \in \mathbb{R}^{n_x}$, and $v_{(k)}^L \in \mathcal{V}^L$, minimize

$$F_k(x, u, v) := \varphi(x(t_N)) + \sum_{i=k}^{N-1} f_0(t_i, x(t_i), u(t_i), v(t_i))$$

subject to

$$\left.\begin{aligned} x(t_{i+1}) &= f(t_i, x(t_i), u(t_i), v(t_i)), && i = k, \ldots, N-1, \\ x(t_k) &= x_k, && \\ x(t_i) &\in \mathcal{X}(t_i), && i = k, \ldots, N, \\ u(t_i) &\in \bar{\mathcal{U}}(v(t_i), v(t_{i-1}), \ldots, v(t_{i-L})), && i = k, \ldots, N-1, \\ v(t_i) &\in \mathcal{V}, && i = k, \ldots, N-1, \\ v(t_{k-\ell}) &= v_{k-\ell}, && \ell = 1, \ldots, L. \end{aligned}\right\} \tag{2}$$

Definition 2. *The function $W : \mathbb{G}_N \times \mathbb{R}^{n_x} \times \mathcal{V}^L \longrightarrow \mathbb{R}$ that assigns to every $(t_k, x_k, v_{(k)}^L) \in \mathbb{G}_N \times \mathbb{R}^{n_x} \times \mathcal{V}^L$ the optimal value of DMIOCP($t_k, x_k, v_{(k)}^L$) is called* optimal value function of DMIOCP, *i.e.*

$$W(t_k, x_k, v_{(k)}^L) := \begin{cases} \inf\limits_{x,u,v \text{ with } (2)} F_k(x, u, v), & \text{if DMIOCP}(t_k, x_k, v_{(k)}^L) \text{ is feasible}, \\ \infty, & \text{otherwise.} \end{cases}$$

The next theorem establishes Bellman's optimality principle.

Theorem 1. *Let $(\hat{x}, \hat{u}, \hat{v})$ be optimal grid functions for DMIOCP and let $\mathbb{G}_N^k := \mathbb{G}_N \setminus \{t_0, \ldots, t_{k-1}\}$. Then, the restrictions on $\mathbb{G}_N^k$ given by $\hat{x}|_{\mathbb{G}_N^k}$, $\hat{u}|_{\mathbb{G}_N^k}$, and $\hat{v}|_{\mathbb{G}_N^k}$ are optimal for DMIOCP($t_k, \hat{x}(t_k), \hat{v}_{(k)}^L$) with $\hat{v}_{(k)}^L = (\hat{v}(t_{k-1}), \ldots, \hat{v}(t_{k-L}))$ for all $k = 0, \ldots, N$ subject to the convention (1).*

Proof. Assume that the restrictions $\hat{x}|_{\mathbb{G}_N^k}$, $\hat{u}|_{\mathbb{G}_N^k}$, and $\hat{v}|_{\mathbb{G}_N^k}$ are not optimal for DMIOCP$(t_k, \hat{x}(t_k), \hat{v}_{(k)}^L)$ for some $k \in \{0, \ldots, N-1\}$. Then there exist feasible grid functions $\tilde{x} : \mathbb{G}_N^k \longrightarrow \mathbb{R}^{n_x}$, $\tilde{u} : \mathbb{G}_N^k \longrightarrow \mathbb{R}^{n_u}$, and $\tilde{v} : \mathbb{G}_N^k \longrightarrow \mathcal{V}$ for DMIOCP$(t_k, \hat{x}(t_k), \hat{v}_{(k)}^L)$ with

$$F_k(\tilde{x}, \tilde{u}, \tilde{v}) < F_k(\hat{x}, \hat{u}, \hat{v}), \tag{3}$$

$\tilde{x}(t_k) = \hat{x}(t_k)$, and $\tilde{v}(t_{k-\ell}) = \hat{v}(t_{k-\ell})$ for $\ell = 1, \ldots, L$. Hence, the trajectories $x : \mathbb{G}_N \longrightarrow \mathbb{R}^{n_x}$, $u : \mathbb{G}_N \longrightarrow \mathbb{R}^{n_u}$, and $v : \mathbb{G}_N \longrightarrow \mathcal{V}$ with

$$(x(t_i), u(t_i), v(t_i)) := \begin{cases} (\hat{x}(t_i), \hat{u}(t_i), \hat{v}(t_i)) \text{ for } i = 0, 1, \ldots, k-1, \\ (\tilde{x}(t_i), \tilde{u}(t_i), \tilde{v}(t_i)) \text{ for } i = k, k+1, \ldots, N, \end{cases}$$

are feasible for DMIOCP and satisfy

$$\begin{aligned} F(x,u,v) &= \varphi(x(t_N)) + \sum_{i=0}^{N-1} f_0(t_i, x(t_i), u(t_i), v(t_i)) \\ &= \varphi(\tilde{x}(t_N)) + \sum_{i=0}^{k-1} f_0(t_i, \hat{x}(t_i), \hat{u}(t_i), \hat{v}(t_i)) + \sum_{i=k}^{N-1} f_0(t_i, \tilde{x}(t_i), \tilde{u}(t_i), \tilde{v}(t_i)) \\ &< \varphi(\hat{x}(t_N)) + \sum_{i=0}^{k-1} f_0(t_i, \hat{x}(t_i), \hat{u}(t_i), \hat{v}(t_i)) + \sum_{i=k}^{N-1} f_0(t_i, \hat{x}(t_i), \hat{u}(t_i), \hat{v}(t_i)) \\ &= \varphi(\hat{x}(t_N)) + \sum_{i=0}^{N-1} f_0(t_i, \hat{x}(t_i), \hat{u}(t_i), \hat{v}(t_i)) \\ &= F(\hat{x}, \hat{u}, \hat{v}), \end{aligned}$$

where (3) is exploited. This contradicts the optimality of $\hat{x}(\cdot)$, $\hat{u}(\cdot)$, and $\hat{v}(\cdot)$. □

From the definition of the optimal value function one immediately obtains

$$W(t_N, x_N, v_{(N)}^L) = \begin{cases} \varphi(x_N), \text{ if } x_N \in \mathcal{X}(t_N), \\ \infty, \quad \text{otherwise.} \end{cases} \tag{4}$$

Bellman's optimality principle allows to derive the following result.

Theorem 2. *For all $(t_k, x_k, v_{(k)}^L) \in \mathbb{G}_N \times \mathcal{X}(t_k) \times \mathcal{V}^L$ and $k = N-1, \ldots, 0$, the optimal value function in Definition 2 satisfies the recursion*

$$\begin{aligned} W(t_k, x_k, v_{(k)}^L) = \inf_{u \in \bar{\mathcal{U}}(v, v_{k-1}, \ldots, v_{k-L}), v \in \mathcal{V}} \Big\{ & f_0(t_k, x_k, u, v) \\ & + W(t_{k+1}, f(t_k, x_k, u, v), v_{(k+1)}^L) \Big\} \end{aligned}$$

where $v_{(k)}^L = (v_{k-1}, \ldots, v_{k-L})$ and $v_{(k+1)}^L = (v, v_{k-1}, \ldots, v_{k-L+1})$ and convention (1) is used. Herein, W at $t = t_N$ is given by (4) and the convention $W(t_k, x_k, v_{(k)}^L) = \infty$ is used whenever $(x_k, v_{(k)}^L) \notin \mathcal{X}(t_k) \times \mathcal{V}^L$.

Proof. Let $(t_k, x_k, v_{(k)}^L) \in \mathbb{G}_N \times \mathcal{X}(t_k) \times \mathcal{V}^L$ and $k \in \{0, \ldots, N-1\}$ be given. If $(x_k, v_{(k)}^L) \notin \mathcal{X}(t_k) \times \mathcal{V}^L$, then $W(t_k, x_k, v_{(k)}^L) = \infty$ by definition.

If $f(t_k, x_k, u, v) \notin \mathcal{X}(t_{k+1})$ for all $u \in \bar{\mathcal{U}}(v, v_{k-1}, \ldots, v_{k-L})$ and all $v \in \mathcal{V}$, then DMIOCP$(t_k, x_k, v_{(k)}^L)$ and DMIOCP$(t_{k+1}, f(t_k, x_k, u, v), v_{(k+1)}^L)$ are infeasible for every $u \in \bar{\mathcal{U}}(v, v_{k-1}, \ldots, v_{k-L})$ and every $v \in \mathcal{V}$ and hence

$W(t_k, x_k, v_{(k)}^L) = \infty$. For arbitrary $u \in \bar{\mathcal{U}}(v, v_{k-1}, \dots, v_{k-L})$ and $v \in \mathcal{V}$ with $f(t_k, x_k, u, v) \in \mathcal{X}(t_{k+1})$ the definition of the optimal value function yields

$$W(t_k, x_k, v_{(k)}^L) \le f_0(t_k, x_k, u, v) + W(t_{k+1}, f(t_k, x_k, u, v), v_{(k+1)}^L).$$

Taking the infimum over all $u \in \bar{\mathcal{U}}(v, v_{k-1}, \dots, v_{k-L})$ and $v \in \mathcal{V}$ yields the first part of the assertion. Now let $\varepsilon > 0$ and feasible grid functions $\tilde{x}, \tilde{u}, \tilde{v}$ with $\tilde{x}(t_k) = x_k$ and

$$F_k(\tilde{x}, \tilde{u}, \tilde{v}) \le W(t_k, x_k, \tilde{v}_{(k)}^L) + \varepsilon$$

with $\tilde{v}_{(k)}^L = (\tilde{v}(t_{k-1}), \dots, \tilde{v}(t_{k-L}))$ be given. Then,

$$\begin{aligned} W(t_k, x_k, \tilde{v}_{(k)}^L) &\ge F_k(\tilde{x}, \tilde{u}, \tilde{v}) - \varepsilon \\ &\ge f_0(t_k, x_k, \tilde{u}(t_k), \tilde{v}(t_k)) \\ &\quad + W(t_{k+1}, f(t_k, x_k, \tilde{u}(t_k), \tilde{v}(t_k)), \tilde{v}_{(k+1)}^L) - \varepsilon \\ &\ge \inf_{u \in \bar{\mathcal{U}}(v, \tilde{v}(t_{k-1}), \dots, \tilde{v}(t_{k-L})), v \in \mathcal{V}} \Big\{ f_0(t_k, x_k, u, v) \\ &\quad + W(t_{k+1}, f(t_k, x_k, u, v), v_{(k+1)}^L) \Big\} - \varepsilon, \end{aligned}$$

where $\tilde{v}_{(k+1)}^L = (\tilde{v}(t_k), \dots, \tilde{v}(t_{k-L+1}))$ and $v_{(k+1)}^L = (v, \tilde{v}(t_{k-1}), \dots, \tilde{v}(t_{k-L+1}))$. As $\varepsilon > 0$ was arbitrary, the assertion follows. □

Theorem 2 allows to deduce the following dynamic programming algorithm. Note that owing to the presence of $v_{(k)}^L$ in DMIOCP, the recursive formula for the optimal value function is not of standard type. The vector $v_{(k)}^L$ contains the history of the discrete-valued control, which influences the decision at stage k. The vector $v_{(k)}^L$ can be interpreted as additional states in the argument of W and leads to an additional runtime factor of M^L when compared to standard dynamic programming. In general this is a prohibitively large factor, but often M and L are moderate numbers. For instance, a binary control ($M = 2$) and $L = 3$ leads to a factor of 8.

Algorithm: Dynamic Programming

Init: Let $\mathbb{G}_N = \{t_0 < t_1 < \dots < t_N\}$ be given. Set

$$W(t_N, x_N, v_{(N)}^L) = \begin{cases} \varphi(x_N), \text{ if } x_N \in \mathcal{X}(t_N), \\ \infty, \text{ otherwise} \end{cases}$$

for all $x_N \in \mathbb{R}^{n_x}$ and all $v_{(N)}^L = (v_{N-1}, \dots, v_{N-L}) \in \mathcal{V}^L$.

Phase 1 (Backward solution): For $k = N-1, \dots, 0$ compute

$$\begin{aligned} W(t_k, x_k, v_{(k)}^L) = \inf_{u \in \bar{\mathcal{U}}(v, v_{k-1}, \dots, v_{k-L}), v \in \mathcal{V}} &\Big\{ f_0(t_k, x_k, u, v) \\ &+ W(t_{k+1}, f(t_k, x_k, u, v), v_{(k+1)}^L) \Big\} \end{aligned} \tag{5}$$

with $v^L_{(k+1)} = (v, v_{k-1}, \ldots, v_{k-L+1})$ for all $x_k \in \mathbb{R}^{n_x}$ and all $v^L_{(k)} \in \mathcal{V}^L$.

Phase 2 (Forward solution): Set $\hat{x}(t_0) = x_0$. For $k = 0, \ldots, N-1$ find

$$\begin{aligned}(\hat{u}(t_k), \hat{v}(t_k)) = \arg \min_{u \in \bar{\mathcal{U}}(v, \hat{v}(t_{k-1}), \ldots, \hat{v}(t_{k-L})), v \in \mathcal{V}} \Big\{ & f_0(t_k, \hat{x}(t_k), u, v) \\ & + W(t_{k+1}, f(t_k, \hat{x}(t_k), u, v), \hat{v}^L_{(k+1)}) \Big\} \end{aligned} \qquad (6)$$

with $\hat{v}^L_{(k+1)} = (v, \hat{v}(t_{k-1}), \ldots, \hat{v}(t_{k-L+1}))$ and set

$$\hat{x}(t_{k+1}) = f(t_k, \hat{x}(t_k), \hat{u}(t_k), \hat{v}(t_k)).$$

3.1 Practical Issues in the Dynamic Programming Algorithm

The implementation of the dynamic programming approach works on a compact state space $\Omega = \{x \in \mathbb{R}^{n_x} \mid x_\ell \leq x \leq x_u\} \subset \mathbb{R}^{n_x}$ with lower and upper bounds $x_\ell, x_u \in \mathbb{R}^{n_x}$ and $-\infty < x_\ell < x_u < \infty$. The bounds should be chosen such that all realistically relevant trajectories are contained in this set. The state space Ω is discretized with an equidistant partition

$$\Omega_{N_x} = \left\{ (x_1, \ldots, x_{n_x})^\top \in \Omega \;\middle|\; \begin{array}{l} x_j = x_{\ell,j} + i \frac{x_{u,j} - x_{\ell,j}}{N_x}, \\ j = 1, \ldots, n_x, \\ i \in \{0, \ldots, N_x\} \end{array} \right\}$$

with $N_x \in \mathbb{N}$. During the backward solution and forward solution phases of the dynamic programming algorithm values of the value function at points $x = f(t_k, x_k, u, v)$ not in Ω_{N_x} are approximated by polynomial interpolation of the value function on a cell of Ω_{N_x} that contains x.

The minimization in (5) and (6) is usually done by a sufficiently dense discretization of the control set $\bar{\mathcal{U}}$ and complete enumeration of all possible values. The v-component is not crucial as $\mathcal{V}$ is supposed to be a finite discrete set. For a fixed v the minimization w.r.t. the component u could be carried out using methods from nonsmooth optimization, e.g. Bundle methods or subgradient methods. However, such methods may result in a local minimum and therefore the previously mentioned enumeration technique is preferred. The general drawback of the dynamic programming approach is the curse of dimensions. Since the value function W depends on the state x and the control history $v^L_{(k)}$ of length L, the approach is computationally feasible merely for low state dimensions and moderate values of L and M, where M is the number of elements in $\mathcal{V}$.

Remark 2. Please note that a more efficient coding of the switching history is possible. To this end, it would be sufficient to replace $v^L_{(k)}$ in W by v_{k-1} and an additional state with values in $\{0, \ldots, L\}$, which encodes how long the last switch is ago (with 0 meaning longer than L).

4 Numerical Results

We consider a car of mass m driving on a road with a given slope profile $\gamma(\cdot)$ and aim to optimize the gear shift control on a given time horizon $[0, T]$. The car model is adapted from [1] with the state (x, v) (position and velocity) and controls $\phi \in \mathcal{U} := [0, 1]$ (gas pedal position) and $\mu \in \mathcal{V} := \{1, \dots, 5\}$ (gear shift). The optimal control problem in discrete time with step-size $h = T/N$, $N \in \mathbb{N}$, reads as follows:

Minimize

$$-\alpha x(t_N) + h \sum_{k=0}^{N-1} f_0(v(t_k), \phi(t_k), \mu(t_k))$$

subject to the constraints $x(0) = 1470$, $v(0) = 23$, *and for* $k = 0, 1, \dots, N-1$,

$$\begin{aligned} x(t_{k+1}) &= x(t_k) + hv(t_k), \\ v(t_{k+1}) &= v(t_k) + \frac{h}{m} \left(\frac{i_g(\mu(t_k)) i_t M_{mot}(\phi(t_k), v(t_k), \mu(t_k))}{R} \right. \\ &\qquad \left. - F_R(v(t_k)) \cos(\gamma(x(t_k))) - F_L(v(t_k)) - mg \sin(\gamma(x(t_k))) \right), \\ &\mu(t_k) \in \mathcal{V}, \quad \phi(t_k) \in \mathcal{U}, \quad v(t_k) \in [22, 30]. \end{aligned}$$

Herein, $\alpha \geq 0$ is a weight factor, i_g is the gear transmission coefficient, i_t the motor torque transmission, R the wheel radius, F_R the friction force of the rolling wheel, $F_L(v) = \frac{1}{2} c_w \rho A v^2$ the drag, and

$$f_0(v, \phi, \mu) = \beta_1 \omega_{mot}(v, \mu) + \beta_2 \left| M_{mot}(\phi, v, \mu) \right| + \beta_3 \omega_{mot}(v, \mu) \left| M_{mot}(\phi, v, \mu) \right|$$

models the fuel costs, where ω_{mot} denotes the rotary frequency of the motor and M_{mot} the motor torque, for details on ω_{mot}, M_{mot}, and F_R please refer to [1].

Figure 1 shows the results of the dynamic programming algorithm for both, the problem with dwell time constraint (with $L = 3$ and $\bar{u} = \bar{\phi} = 0$, i.e. ϕ is set to zero, if μ switches) and the problem without dwell time constraint. The parameters $\alpha = 1/10$, $\beta_1 = 1/120$, $\beta_2 = 10/258$, $\beta_3 = 1/10320$, $c_w = 0.3$, $\rho = 1.249512$, $A = 2$, $R = 0.302$, $i_t = 3.91$, $g = 9.81$, $m = 2000$, $T = 10$, $N = 25$, $N_x = (100, 45)$, $N_u = 100$, $i_g(1) = 3.91$, $i_g(2) = 2.002$, $i_g(3) = 1.33$, $i_g(4) = 1$, $i_g(5) = 0.805$ were used. The optimal value for the problem with dwell time constraint is -79.7656, while the optimal value without dwell time constraint is -88.0430. In the presence of the dwell time constraint it can be nicely seen that ϕ is set to zero after μ switches at approximately $t = 4$. The CPU time on a PC with 2.3 GHz is 1 min 47.146 s without dwell time constraint and 14 min 38.9 s with dwell time constraint.

5 Conclusions

The paper discusses a dynamic programming principle that allows to consider dwell times in a time discrete optimal control problem with discrete-valued controls. Dwell times have to be taken into account in many applications where

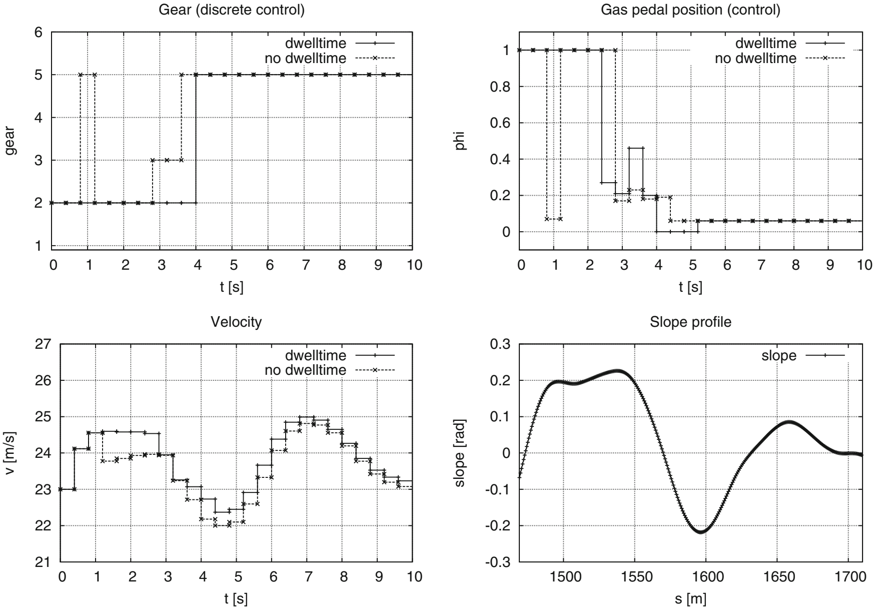

Fig. 1. Optimal gear sequence (top left), gas pedal position (top right), velocity (bottom left), and slope profile (bottom right).

a switch in a discrete-valued control fixes the dynamics of the process for a specified time period before the process can be controlled again. A particular example is the optimization of gear shifts in a car, where a shift in gears implies that the motor torque is reduced for a given time period. A dynamic programming approach is suggested to solve such problems and a numerical example shows the applicability of the method. However, the curse of dimension associated to dynamic programming approaches limits the method to low dimensional systems with few states. Hence, it is desirable to develop further methods for this problem class and extend it to optimal control problems in continuous time. To this end, a formulation of the dwell time constraint in a continuous time setting becomes necessary, which is subject to future research.

References

1. 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), 1–18 (2005)
2. Gerdts, M.: A variable time transformation method for mixed-integer optimal control problems. Optim. Control Appl. Methods **27**(3), 169–182 (2006)
3. Gerdts, M.: Optimal Control of ODEs and DAEs. DeGruyter, Berlin (2011)
4. Göttlich, S., Herty, M., Ziegler, U.: Modeling and optimizing traffic light settings on road networks. Comput. Oper. Res. **55**, 36–51 (2015)

5. Howlett, P.: Optimal strategies for the control of a train. Automatica **32**(4), 519–532 (1996)
6. Khmelnitsky, E.: A combinatorial, graph-based solution method for a class of continuous-time optimal control problems. Math. Oper. Res. **27**(2), 312–325 (2002)
7. Kirches, C.: Fast numerical methods for mixed-integer nonlinear model-predictive control. Ph.D. thesis, Naturwissenschaftlich-Mathematische Gesamtfakultät, Universität Heidelberg, Heidelberg, Germany (2010)
8. 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). doi:10.1002/oca.892
9. Lee, H.W.J., Teo, K.L., Cai, X.Q.: An optimal control approach to nonlinear mixed integer programming problems. Comput. Math. Appl. **36**(3), 87–105 (1998)
10. Lee, H.W.J., Teo, K.L., Rehbock, V., Jennings, L.S.: Control parameterization enhancing technique for time optimal control problems. Dyn. Syst. Appl. **6**(2), 243–262 (1997)
11. Lee, H.W.J., Teo, K.L., Rehbock, V., Jennings, L.S.: Control parametrization enhancing technique for optimal discrete-valued control problems. Automatica **35**(8), 1401–1407 (1999)
12. Palagachev, K., Gerdts, M.: Mathematical programs with blocks of vanishing constraints arising in discretized mixed-integer optimal control problems. Set-Valued Var. Anal. **23**(1), 149–167 (2015)
13. Sager, S.: Reformulations and algorithms for the optimization of switching decisions in nonlinear optimal control. J. Process Control **19**(8), 1238–1247 (2009)
14. Sager, S., Bock, H.G., Diehl, M., Reinelt, G., Schlöder, J.P.: Numerical methods for optimal control with binary control functions applied to a Lotka-Volterra type fishing problem. In: Seeger, A. (ed.) Recent Advances in Optimization (Proceedings of the 12th French-German-Spanish Conference on Optimization). Lectures Notes in Economics and Mathematical Systems, vol. 563, pp. 269–289. Springer, Heidelberg (2006)
15. Sager, S., Kirches, C., Bock, H.G.: Fast solution of periodic optimal control problems in automobile test-driving with gear shifts. In: Proceedings of the 47th IEEE Conference on Decision and Control (CDC 2008), Cancun, Mexico, pp. 1563–1568 (2008). ISBN: 978-1-4244-3124-3
16. Sager, S.: Numerical methods for mixed-integer optimal control problems. Ph.D. thesis, Heidelberg: Univ. Heidelberg, Naturwissenschaftlich-Mathematische Gesamtfakultät (Diss.). viii, 219 p. (2006)
17. Sager, S., Bock, H.G., Reinelt, G.: Direct methods with maximal lower bound for mixed-integer optimal control problems. Math. Program. (A) **118**(1), 109–149 (2009)
18. Siburian, A.: Numerical Methods for Robust, Singular and Discrete Valued Optimal Control Problems. Ph.D. thesis, Curtin University of Technology, Perth, Australia (2004)
19. Teo, K.L., Jennings, L.S., Lee, V., Rehbock, H.W.J.: The control parameterization enhancing transform for constrained optimal control problems. J. Aust. Math. Soc. **40**(3), 314–335 (1999)