Optimizing Online Control of Constrained Systems with Switched Dynamics

Zonglin Liu[1] and Olaf Stursberg[1]

Abstract— **This paper studies the online control of switched systems over a finite time horizon subject to time-varying constraints on the continuous states and bounded input set. The formulation leads to optimization problems of the class mixed-integer nonlinear programming (MINLP), which is known to be computationally hard. This paper proposes a method to approximate the optimal solution while keeping the computational effort low enough, to enable real-time applicability for many systems. The main idea is to use heuristics based on the value function for relaxed sub-problems to prune the tree encoding the possible sequences of discrete choices (i.e. the selected continuous dynamics) over a prediction horizon. Numeric tests show that the times for computation are drastically below those for standard MINLP solution in the vast majority of cases, while good approximations of the optimal solutions are obtained.**

I. Introduction

Switched systems are an appropriate class of model for all those processes in which discrete controls enable to select the continuous dynamics, while the latter is governed by continuous controls. Automatic gear-shifts, which determine a gear in conjunction to the engine torque, are an example for such processes. While the presence of mixed (continuous-discrete) degrees of freedom may enhance the chance of reaching the control goal, the challenging aspect is to select the combination of controls which maximizes the system performance. If formulated as optimization problem, the class of MINLP problems is encountered, which is known to be NP hard, see e.g. [3], [10]. The use of relaxations (i.e. temporarily treating an integer variable $v \in \{0,1\}$ as a continuous one $v \in [0,1]$) is an established method to generate lower cost bounds when exploring and pruning the tree of integer variables, and it has been used in a number of approaches for solving control problems for switched systems [4], [11], [13], [14]. While for some problem instances relaxations may lead to satisfactory results (in terms of efficiently pruning the search tree), the opposite effect can occur for switched systems in some cases: using $v \in [0,1]$ instead of $v \in \{0,1\}$ may mean to average between two distinct dynamics, leading to system evolution which is not possible for the switched system. A gross under-estimation of a bound may result, i.e. many nodes of the search tree may be explored which later turn out to be infeasible. If the solution is approached by existing solvers for MINLP problems, such as 'GUROBI' [1], 'DICOPT' [9], it can be observed that typically the computation times increase quickly with a growing number of integer variables. (In addition, it is not guaranteed that global optima are determined.) These shortcomings motivate the investigation of techniques that do not not rely on relaxations of binary variables.

This work studies the online control of switched systems by optimizing over a finite horizon, considering state and input constraints, and deriving effective bounds for pruning the search tree from value function. The consideration of constraints is in contrast to the work in [4], [14], which also explores the use of value function for pruning, but for the unconstrained case. In addition, the two approaches involve the identification of a relaxation parameter on which the outcome of the optimization depends with high sensitivity, i.e. an appropriate value is difficult to find. The same problem as in this paper (thus including constraints) has been investigated in [2], but the latter does not focus on computational complexity and thus on applicability in real-time. In contrast, the paper on hand first decomposes the original complex problem into simpler subproblems, each of which can be solved in polynomial time with known solvers, and which provide cost bounds. In a second step, tree search with suitable branch and pruning strategies is employed to determine good choices for the continuous and discrete degrees of freedom.

Section 2 first defines the considered control problem, Sec. 3 proposes methods to derive lower and upper cost bounds, Sec. 4 contains the tree search procedure, while several numerical tests in Sec. 5 demonstrate the effectiveness of the method, before Sec. 6 concludes the paper.

II. Problem Formulation

The class of models studied in this paper are switched systems, where a discrete input switches the continuous dynamics. For $k \in \mathbb{N}^{\geq 0}$, consider the discrete-time model:

$$\begin{aligned} &x_{k+1} = A_{v_k} x_k + B_{v_k} u_k, \\ &x_k \in X_k,\ u_k \in U,\ \ v_k \in V, \end{aligned} \tag{1}$$

where $x_k \in \mathbb{R}^{n_x \times 1}$ is the continuous state bounded to time-varying convex constraints X_k, $u_k \in \mathbb{R}^{n_u \times 1}$ is the continuous input selected from a convex input space U, and v_k is the discrete input which determines the parametrization of the continuous dynamics. The latter is chosen from the set $V = \{v_{[1]}, \dots, v_{[n_v]}\}$, and any $v_k \in V$ corresponds to a pair of matrices (A_{v_k}, B_{v_k}). Note that the selection of this pair by v_k leads to a nonlinear right-hand side of the dynamics equation.

In this paper, an online control scheme is considered: for a finite horizon $N \in \mathbb{N}$, let a state $x_k \in X_k$ measured in the current step k, and a sequence of convex state constraints $(X_k, X_{k+1|k}, \dots, X_{k+N|k})$ be given (through prediction).

[1]Control and System Theory, EECS, University of Kassel, Germany {Z.Liu, stursberg}@uni-kassel.de

The index $k+j|k$ denotes a prediction of a quantity for step $k+j$ made at step k. If sequences of continuous inputs $\phi^u_{k,N} = (u_{k|k}, \ldots, u_{k+N-1|k})$ and discrete inputs $\phi^v_{k,N} = (v_{k|k}, \ldots, v_{k+N-1|k})$ are selected, then a sequence of continuous states $\phi^x_{k,N} = (x_k, x_{k+1|k}, \cdots, x_{k+N|k})$ results satisfying for $j \in \{0, \cdots, N-1\}$ according to [1]:

$$x_{k+j+1|k} = \\ (\prod_{l=0}^{j} A_{v_{k+l|k}}) \cdot x_k + \sum_{l=0}^{j} [(\prod_{t=0}^{j-l-1} A_{v_{k+j-t|k}}) \cdot B_{v_{k+l|k}} \cdot u_{k+l|k}]. \tag{2}$$

The control objective is to drive the state into the region $X_{k+N|k}$ containing the target state $x_f \in X_{k+N|k}$ (with a target input $u_f = 0^{n_u \times 1}$) in N steps, while minimizing the following cost function:

$$\Omega(x_k, x_f, \phi^v_{k,N}, \phi^u_{k,N}) := (x_{k+N|k} - x_f)^{\mathrm{T}} Q_N (x_{k+N|k} - x_f) \\ + \sum_{j=0}^{N-1} \underbrace{(x_{k+j|k} - x_f)^{\mathrm{T}} Q_1 (x_{k+j|k} - x_f) + u^{\mathrm{T}}_{k+j|k} Q_2 u_{k+j|k}}_{\textit{step cost: } \mathcal{L}(x_{k+j|k}, u_{k+j|k})}. \tag{3}$$

Here, $Q_1 = {Q_1}^{\mathrm{T}} \geq 0$, $Q_2 = {Q_2}^{\mathrm{T}} > 0$ and $Q_N = {Q_N}^{\mathrm{T}} > 0$ are the weighting matrices of the different parts. Now, the overall online control problem can be defined as follows:

Problem 1:

$$\min_{\phi^v_{k,N}, \phi^u_{k,N}} \Omega(x_k, x_f, \phi^v_{k,N}, \phi^u_{k,N})$$
$$\text{s.t.:} \quad (2),\ x_{k+j|k} \in X_{k+j|k},\ j \in \{1, \ldots, N\},$$
$$x_f \in X_{k+N|k},$$
$$u_{k+j|k} \in U, v_{k+j|k} \in V,\ j \in \{0, \ldots, N-1\}.$$

Such problems can be solved by available solvers for MINLP problems, but the combinatorics with respect to ϕ^v leads typically to large computation times (or only permits the use of small values of N): Since $v(k+j|k)$ has to be selected for any $j \in \{0, \ldots, N-1\}$, the number of available discrete input sequences $\phi^v_{k,N}$ is n_v^N, i.e. it increases exponentially with N, and the N must be chosen large enough to enable $x_f \in X_{k+N|k}$. This motivates the development of an algorithm in this paper which aims at finding suitable compromises between performance and applicability for large N. This is achieved by a tree search algorithm motivated by the one in [12] together with cost bounds and search heuristics which are tailored to the structure of Problem 1: Let a tree denoted by $\mathcal{T}_k = \{\mathcal{G}_k, \mathcal{E}_k\}$, consisting of a set of nodes $\mathcal{G}_k$ and a set of edges $\mathcal{E}_k$, see also Fig. 1. Each node represents a continuous state $x_{k+j|k}$ reachable by one pair of input sequences $\phi^u_{k,j} = (u_{k|k}, \ldots, u_{k+j-1|k})$ and $\phi^v_{k,j} = (v_{k|k}, \ldots, v_{k+j-1|k})$. Let the set $\mathbb{G}_{k+j|k}$ for $j \in \{1, \ldots, N\}$ denote the set of nodes possibly explored by the algorithm by choosing the possible options of input sequence $\phi^v_{k,j}$ and $\phi^u_{k,j}$. Obviously $\mathbb{G}_{k|k} = \{x_k\}$ can be regarded as the root node of the tree (where we name the nodes identical to the corresponding continuous states for simplification). The symbol $\mathcal{E}_k = \{\mathbb{E}_{k|k}, \cdots, \mathbb{E}_{k+N-1|k}\}$ denotes the set of edges of the tree, where an element of $\mathbb{E}_{k+j|k}$ denotes an edge representing a particular choice for the discrete input $v_{k+j|k} \in V$; here again, the edges are denoted equivalently to the discrete inputs for simplicity. For an efficient algorithm operating on the tree, the obvious goal is to explore as few nodes of the tree as possible to obtain trajectories that are (at least) close to the optimal solution of Problem 1.

[1]Note that $\prod_{t=0}^{p} \cdot = 1$ for $p < 0$, as applies for $j = 0$.

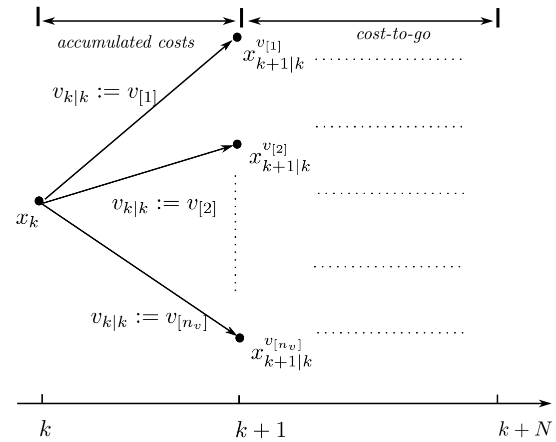

Fig. 1. Structure of the search tree with node sets $\mathbb{G}_{k|k} = \{x_k\}$, $\mathbb{G}_{k+1|k} = \{x^{v_{[1]}}_{k+1|k}, \cdots, x^{v_{[n_v]}}_{k+1|k}\}$ and set of edges: $\mathbb{E}_{k|k} = \{v_{[1]}, \cdots, v_{[n_v]}\}$.

To avoid the exponential growth of the tree with increasing j, two mechanisms are proposed and explored in this paper:

- any node is evaluated by its costs, i.e. by the *accumulated costs* as the sum of the *step-costs* $\mathcal{L}(x, u)$ for the transfer from the root node to the current node, and by the *cost-to-go* as the costs estimated to be acquired for the steps from the current node to the end of the time horizon. Only if these costs compare to specific bounds such that the node may belong to a path of low overall costs of (3), the node is further explored, see Sec. 3;
- if two continuous states (nodes) $x_{k+j|k,1}$ and $x_{k+j|k,2}$ are reached by employing two different pairs of partial strategies $(\phi^u_{k,j}, \phi^v_{k,j})_1$ and $(\phi^u_{k,j}, \phi^v_{k,j})_2$ of same length j, and are close to each other (i.e. $\|x_{k+j|k,1} - x_{k+j|k,2}\|_2$ being small), only the node with smaller *accumulated cost* is further explored, see Sec. 4.

III. Lower and Upper Cost Bounds

In order to prepare the computation of cost bounds, the notion of *cost-to-go* is formalized first, and then schemes for determining costs bounds are proposed, which are used later to reduce the size of the search tree to be explored.

A. Definition of the Cost-To-Go

According to some recent results on optimizing discrete-time switched systems (cf. [4], [15]), the notion of the value function $\mathcal{V}(x_{k+j|k})$ can be used to formulate the minimal *cost-to-go* from state $x_{k+j|k}$ to the end of the prediction horizon. It takes the form:

$$\begin{aligned}\mathcal{V}(x_{k+j|k}) := \min_{\phi^v_{k+j,N},\phi^u_{k+j,N}} \{ &\sum_{i=j}^{N-1} \mathcal{L}(x_{k+i|k}, u_{k+i|k}) \\ &+ (x_{k+N|k} - x_f)^{\mathrm{T}} Q_N (x_{k+N|k} - x_f) \} \\ \text{s. t.: } & v_{k+i|k} \in V,\ u_{k+i|k} \in U,\ \forall i \in \{j, \cdots, N-1\}, \\ & x_{k+i|k} \in X_{k+i|k},\ \forall i \in \{j+1, \cdots, N\}. \end{aligned} \tag{4}$$

In here, $\phi^v_{k+j,N} = (v_{k+j|k}, \ldots, v_{k+N-1|k})$ denotes the discrete input sequence from step $k+j$ to step $k+N-1$ (equivalently for $\phi^u_{k+j,N}$). Note that for $j=0$, the value of $\mathcal{V}(x_k)$ is identical to the solution of Problem 1, and is denoted then by $\Omega^*(x_k, x_f, \phi^{v,*}_{k,N}, \phi^{u,*}_{k,N})$. If state $x_{(k+j+1|k)} \in X_{(k+j+1|k)}$ is reachable from $x_{k+j|k}$ by selecting a feasible $u_{k+j|k} \in U$ and $v_{k+j|k} \in V$, then $\mathcal{V}(x_{k+j|k})$ satisfies:

$$\begin{aligned}&\mathcal{V}(x_{k+j|k}) = \min_{v_{k+j|k}, u_{k+j|k}} \{\mathcal{L}(x_{k+j|k}, u_{k+j|k}) + \mathcal{V}(x_{k+j+1|k})\} \\ &\text{s. t.: } x_{k+j+1|k} = A_{v_{k+j|k}} x_{k+j|k} + B_{v_{k+j|k}} u_{k+j|k}. \end{aligned} \tag{5}$$

B. Lower Bound Identification

A lower bound of $\mathcal{V}(x_{k+j|k})$ (and thus of the *cost-to-go* from $x_{k+j|k}$) is determined as follows: By omitting the constraints for the continuous inputs and states in (4), the following relaxed problem is formulated:

$$\begin{aligned}\mathcal{V}^{un}(x_{k+j|k}) := \min_{\phi^v_{k+j,N},\phi^u_{k+j,N}} \{ &\sum_{i=j}^{N-1} \mathcal{L}(x_{k+i|k}, u_{k+i|k}) \\ &+ (x_{k+N|k} - x_f)^{\mathrm{T}} Q_N (x_{k+N|k} - x_f) \} \\ \text{s. t.: } & v_{k+i|k} \in V,\ \forall i \in \{j, \cdots, N-1\}. \end{aligned} \tag{6}$$

Due to an enlarged feasible space in comparison to problem (4), the optimal cost satisfies: $\mathcal{V}^{un}(x_{k+j|k}) \leq \mathcal{V}(x_{k+j|k})$, i.e. it determines a lower bound of $\mathcal{V}(x_{k+j|k})$. The difference between $\mathcal{V}^{un}(x_{k+j|k})$ and $\mathcal{V}(x_{k+j|k})$ depends on the extent by which input and state constraints are active for the solution of (4). For computing $\mathcal{V}^{un}(x_{k+j|k})$, the use of *difference Riccati equations* according to [8] has been discussed in different approaches (see [4], [11], [15]). This concept is briefly reviewed here:

1) starting from the last step $k+N$, the value of $\mathcal{V}^{un}(x_{k+N|k})$ is determined by the continuous state achievable at step $k+N$:

$$\begin{aligned}&\mathcal{V}^{un}(x_{k+N|k}) := \\ &\min_{x_{k+N|k}} (x_{k+N|k} - x_f)^{\mathrm{T}} Q_N (x_{k+N|k} - x_f) \end{aligned} \tag{7}$$

Meanwhile, a matrix $\mathcal{P}^{un,*}_N := Q_N$ is defined to replace Q_N in (7) for a notation that is consistent with the following;

2) for each available input $v_{[q]} \in V$ at any step $k+i$, $i \in \{j+1, \cdots, N\}$, a matrix $\mathcal{P}^{un,v_{[q]}}_{i-1}$ originated from matrix $\mathcal{P}^{un}_i$ is determined by applying the following *difference Riccati equation* backwards:

$$\begin{aligned}\mathcal{P}^{un,v_{[q]}}_{i-1} &= A^{\mathrm{T}}_{v_{[q]}} \mathcal{P}^{un}_i A_{v_{[q]}} + 2\mathcal{K}^{\mathrm{T}}_{i-1} B^{\mathrm{T}}_{v_{[q]}} \mathcal{P}^{un}_i A_{v_{[q]}} \\ &+ Q_1 + \mathcal{K}^{\mathrm{T}}_{i-1}(B^{\mathrm{T}}_{v_{[q]}} \mathcal{P}^{un}_i B_{v_{[q]}} + Q_2)\mathcal{K}_{i-1}, \\ \mathcal{K}_{i-1} &= -(B^{\mathrm{T}}_{v_{[q]}} \mathcal{P}^{un}_i B_{v_{[q]}} + Q_2)^{-1} B^{\mathrm{T}}_{v_{[q]}} \mathcal{P}^{un}_i A_{v_{[q]}}. \end{aligned} \tag{8}$$

Let the symbol $\mathbb{P}^{un}_{i-1}$ denote the set of matrices $\mathcal{P}^{un,v_{[q]}}_{i-1}$ obtained in step $i-1$ for all $v_{[q]} \in V$;

3) if the computation above is repeated backwards up to step $k+j$, a set of ${n_v}^{N-j}$ different matrices are contained in the set $\mathbb{P}^{un}_j$. Then, the value of $\mathcal{V}^{un}(x_{k+j|k})$ can be determined by:

$$\mathcal{P}^{un,*}_j = \arg\min_{\mathcal{P}^{un}_j \in \mathbb{P}^{un}_j} (x_{k+j|k} - x_f)^{\mathrm{T}} \mathcal{P}^{un}_j (x_{k+j|k} - x_f), \tag{9}$$

$$\mathcal{V}^{un}(x_{k+j|k}) = (x_{k+j|k} - x_f)^{\mathrm{T}} \mathcal{P}^{un,*}_j (x_{k+j|k} - x_f). \tag{10}$$

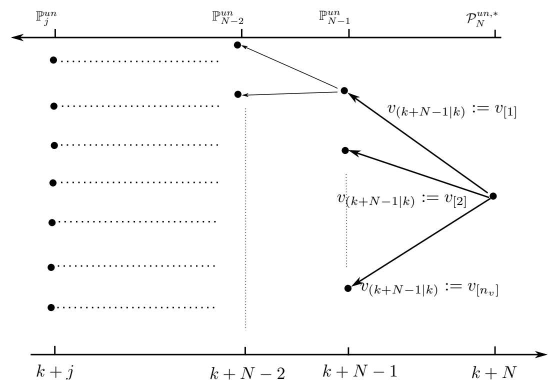

Fig. 2. Backward computation for $\mathcal{V}^{un}(x_{k+j|k})$.

The procedure is illustrated in Fig. 2. Obviously, the explicit enumeration of the discrete choices induces an exponential growing of the computational complexity and lets the lower bound identification become intractable rapidly, since up to $\sum_{i=1}^{N} {n_v}^i$ different matrices have to be evaluated for the determination of $\mathcal{P}^{un,*}_0$ on the root layer.

As a countermeasure, [4] and [11] proposed to multiply the *step-cost* $\mathcal{L}$ by a relaxation factor, leading to an efficient reduction of the size of the set $\mathbb{P}^{un}_j$. However, selecting an appropriate value of the relaxation factor is difficult, and no constructive rule seems to exist. The pruning of the tree appears to be extremely sensitive to the factor, as indicated in [4]: increasing the factor from $\alpha = 1.0$ to $\alpha = 1.0001$ reduces the size of $\mathbb{P}^{un}_0$ from 386 to 22. While the computation of $\mathcal{V}^{un}(x_{k+j|k})$ is very similar to the one in [4], [11], our objective is not quite the same: the references aim at an explicit and efficient approximation of $\mathcal{V}^{un}(x_{k+j|k})$, whereas we go for approximating $\mathcal{V}(x_{k+j|k})$ instead of $\mathcal{V}^{un}(x_{k+j|k})$, where $\mathcal{V}^{un}(x_{k+j|k})$ only provides a lower bound of the original variant. Thus, computing a lower bound of $\mathcal{V}(x_{k+j|k})$ is just an intermediate step for the later comparison of different nodes, i.e. approximating $\mathcal{V}^{un}(x_{k+j|k})$ should not incur significant effort.

To this end, the following method is proposed in order to reduce the complexity of computing $\mathcal{V}^{un}(x_{k+j|k})$: Note first that the definition of the value function $\mathcal{V}(x_{k+j|k})$ in (5) is based on the principle of *Dynamic Programming*, which can be transferred to the computation of $\mathcal{V}^{un}(x_{k+j|k})$, i.e. for any $v_{k+j|k} \in V$ applies:

$$\begin{aligned} &\mathcal{V}^{un}(x_{k+j|k}) := \\ &\min_{v_{k+j|k}, u_{k+j|k}} \{\mathcal{L}(x_{k+j|k}, u_{k+j|k}) + \mathcal{V}^{un}(x_{k+j+1|k})\} \quad (11) \\ &\text{s. t.: } x_{k+j+1|k} = A_{v_{k+j|k}} x_{k+j|k} + B_{v_{k+j|k}} u_{k+j|k}. \end{aligned}$$

This can be rewritten by substituting $\mathcal{V}^{un}(x_{k+j+1|k})$ by $(x_{k+j+1|k} - x_f)^{\mathrm{T}} \mathcal{P}_{j+1}^{un,*}(x_{k+j+1|k} - x_f)$ such that:

$$\begin{aligned} \mathcal{V}^{un}(x_{k+j|k}) := &\min_{v_{k+j|k}, u_{k+j|k}} \{\mathcal{L}(x_{k+j|k}, u_{k+j|k}) \\ &+ (x_{k+j+1|k} - x_f)^{\mathrm{T}} \mathcal{P}_{j+1}^{un,*}(x_{k+j+1|k} - x_f)\} \quad (12) \\ \text{s. t.: } &x_{k+j+1|k} = A_{v_{k+j|k}} x_{k+j|k} + B_{v_{k+j|k}} u_{k+j|k}. \end{aligned}$$

Now for state $x_{k+j|k}$, assuming that the optimal state sequence $\phi_{k+j,N}^{x,*} = (x_{k+j|k}, x_{k+j+1|k}^{*} \cdots, x_{k+N|k}^{*})$ resulting from (6) would be known and that only one pair $(u_{k+j|k}^{*}, v_{k+j|k}^{*})$ can drive the state $x_{k+j|k}$ to $x_{k+j+1|k}^{*}$, then the value of the first term $\mathcal{L}(x_{k+j|k}, u_{k+j|k})$ in (12) can be fixed, and the minimization of the second term $(x_{k+j+1|k} - x_f)^{\mathrm{T}} \mathcal{P}_{j+1}^{un,*}(x_{k+j+1|k} - x_f)$ can be approximated by:

$$\min_{\mathcal{P}_{j+1}^{un} \in \mathbb{P}_{j+1}^{un}} (x_{k+j+1|k}^{*} - x_f)^{\mathrm{T}} \mathcal{P}_{j+1}^{un} (x_{k+j+1|k}^{*} - x_f). \quad (13)$$

As the optimal state $x_{k+j+1|k}^{*}$ is assumed to be known, the value of $\mathcal{V}^{un}(x_{k+j|k})$ solely depends on the selection of the matrix $\mathcal{P}_{j+1}^{un}$. This indicates that, in the backwards computational procedure, one only has to expand the matrix $\mathcal{P}_{j+1}^{un}$ which leads to the minimal value of (13) over the set $\mathbb{P}_{j+1}^{un}$, instead of expanding all of the matrices. However, the optimal state sequence $\phi_{k+j,N}^{x,*}$ cannot be determined before $\mathcal{V}^{un}(x_{k+j|k})$ is identified. But if it is assumed that N is chosen large enough for $x_N = x_f$, the reasoning (used also in [5], [6]) applies that the infinite-horizon costs can be formulated as quadratic function of the initial state. Here, of course, $x_{k+j+1|k}^{*}$ plays the role of the initial state of the consideration, and the matrix for encoding the costs (within the quadratic function) has to account for the v-dependency of the cost terms. If so, we choose to minimize the trace of $\mathcal{P}_{j+1}^{un}$ to replace problem (13) in determining the minimal value of the first step of the *cost-to-go*:

$$\min_{\mathcal{P}_{j+1}^{un} \in \mathbb{P}_{j+1}^{un}} \mathit{trace}(\mathcal{P}_{j+1}^{un}). \quad (14)$$

Then, the traces of the matrices in the set $\mathbb{P}_{j+1}^{un}$ have to be compared, and the one with minimal value is expanded for the next step. Algorithm 1 shows how this concept can be used to determine an approximation of $\mathcal{V}^{un}(x_{k+j|k})$ efficiently.

Starting from the last step $k+N$ of the prediction horizon, the algorithm constitutes a 'best-first' type search method, in which the traces of all $\mathcal{P}_{N-i}^{un} \in \mathbb{P}_{N-i}^{un}$ are compared, and only the one with the minimal trace found so far is kept. Thus, the sets $\mathbb{P}_{N-i}^{un}$ contain for all $i \in \{0, \cdots, N-j\}$ only one element, and the only matrix $\mathcal{P}_{j}^{un,apx}$ contained in $\mathbb{P}_{j}^{un}$ determines an approximation of $\mathcal{P}_{j}^{un,*}$. Thereby, as each matrix with minimal trace refers to a specific discrete input, the sequence $\phi_{k+j,N}^{v,apx} = (v_{k+j|k}^{apx}, \ldots, v_{k+N-1|k}^{apx})$ is obtained, which corresponds to the approximation of the lower cost bound. In contrast to the explicit enumeration of all discrete sequences, the computational complexity is reduced from $\sum_{i=1}^{N-j} {n_v}^{i}$ to $(N-j) \cdot n_v$. Moreover, the tuning of a relaxation factor is not required, and numeric studies for several examples show that the obtained lower bounds for $\mathcal{V}(x_{k+j|k})$ are suitable for pruning the search tree significantly.

Eventually, the approximated lower bound $\underline{\mathcal{V}}(x_{k+j|k})$ of $\mathcal{V}(x_{k+j|k})$ is defined as:

$$\begin{aligned} &\underline{\mathcal{V}}(x_{k+j|k}) := (x_{k+j|k} - x_f)^{\mathrm{T}} \mathcal{P}_{j}^{un,apx}(x_{k+j|k} - x_f) \\ &\approx (x_{k+j|k} - x_f)^{\mathrm{T}} \mathcal{P}_{j}^{un,*}(x_{k+j|k} - x_f) = \mathcal{V}^{un}(x_{k+j|k}) \\ &\leq \mathcal{V}(x_{k+j|k}). \end{aligned} \quad (15)$$

The above relation suggests that once $x_{k+j|k}$ and $\mathcal{P}_{j}^{un,apx}$ are known, the lower bound of the *cost-to-go* from state $x_{k+j|k}$ in Problem 1 can be computed immediately.

C. Upper Bound Determination

In reference to (4), the optimal discrete and continuous input sequences $\phi_{k+j,N}^{v,*}, \phi_{k+j,N}^{u,*}$ should satisfy the following property: for all $\phi_{k+j,N}^{v}$ and $\phi_{k+j,N}^{u}$ satisfying that $x_{k+i|k} \in X_{k+i|k}$ for all $i \in \{j+1, \cdots, N\}$, the corresponding value of the cost function in (4) is not smaller than $\mathcal{V}(x_{k+j|k})$. Thus, when applying $\phi_{k+j,N}^{v,apx} = (v_{k+j|k}^{apx}, \ldots, v_{k+N-1|k}^{apx})$ obtained from Algorithm 1, the evolution of the continuous state in (2) only depends on the continuous input. Correspondingly, the binary variables in the original MINLP problem are

Algorithm 1 The approximation of $\mathcal{V}^{un}(x_{k+j|k})$

Given: Node $x_{k+j|k}$, $\mathbb{P}_{N}^{un} = \mathcal{P}_{N}^{un,*} = Q_N$;
for $i = 1 : N-j$ **do**
  for $m = 1 : |\mathbb{P}_{N-i+1}^{un}|$ **do**
    for $q = 1 : n_v$ **do**
      compute the matrix $\mathcal{P}_{N-i}^{un,v_{[q]}}$ according to (8) and insert it into the set $\mathbb{P}_{N-i}^{un}$
    end for
  end for
  find the $\mathcal{P}_{N-i}^{un} \in \mathbb{P}_{N-i}^{un}$ with the minimal trace and assign $\mathbb{P}_{N-i}^{un} := \{\mathcal{P}_{N-i}^{un}\}$
end for
for the only matrix $\mathcal{P}_{j}^{un,apx}$ in $\mathbb{P}_{j}^{un}$ compute: $\mathcal{V}^{un,apx}(x_{k+j|k}) := (x_{k+j|k} - x_f)^{\mathrm{T}} \mathcal{P}_{j}^{un,apx}(x_{k+j|k} - x_f)$.

fixed with the choice of $\phi^{v,apx}_{k+j,N}$, and a simple Quadratic Programming (QP) problem results, which can be solved efficiently. In general, an upper bound $\overline{\mathcal{V}}(x_{k+j|k})$ of $\mathcal{V}(x_{k+j|k})$ is obtained by solving the following QP problem:

$$\overline{\mathcal{V}}(x_{k+j|k}) = \min_{\phi^u_{k+j,N}} \{ \sum_{i=j}^{N-1} \mathcal{L}(x_{k+i|k}, u_{k+i|k}) + (x_{k+N|k} - x_f)^{\mathrm{T}} Q_N (x_{k+N|k} - x_f) \} \quad (16)$$
$$\text{s.t.: } x_{k+i|k} \in X_{k+i|k},\ i \in \{j, \dots, N\},$$
$$u_{k+i|k} \in U,\ v_{k+i|k} := v^{apx}_{k+i|k},\ i \in \{j, \dots, N-1\}.$$

In case no feasible solution exists to this problem, $\overline{\mathcal{V}}(x_{k+j|k})$ is assigned with a value of ∞. In summary, for any state $x_{k+j|k}$ on any corresponding tree layer with index $k+j$, the *costs-to-go* in Problem 1 is bounded by:

$$\underline{\mathcal{V}}(x_{k+j|k}) \leq \mathcal{V}(x_{k+j|k}) \leq \overline{\mathcal{V}}(x_{k+j|k}). \quad (17)$$

Overall, the computational effort for the bound computations can be estimated to comprise $(N-j) \cdot n_v$ trace computations for the lower bound and the solution of one QP problem for the upper bound.

IV. The Search Procedure

The bounds derived before are now used within a tree search procedure to reduce the number of candidates for the optimal solution of Problem 1. The root node of the tree $\mathcal{T}_k$ to be investigated for the step k represents the state x_k. For any node $x_{k+j|k}$ on the layer referring to time $k+j$, the *accumulated cost* are determined from the partial strategy $(\phi^u_{k,j},\ \phi^v_{k,j})$ that transfers x_k to $x_{k+j|k}$:

$$\Omega(x_k, x_{k+j|k}, \phi^v_{k,j}, \phi^u_{k,j}) = \sum_{i=0}^{j-1} \mathcal{L}(x_{k+i|k}, u_{k+i|k}) \quad (18)$$
$$\text{s.t.: } (2),\ u_{k+i|k} \in \phi^u_{k,j},\ v_{k+i|k} \in \phi^v_{k,j},\ i \in \{0, \dots, j-1\}.$$

In order to determine the best partial strategy from step $k+j$ to $k+j+1$, consider the transition from $x_{k+j|k}$ to $x^{*,v_{[q]}}_{k+j+1|k}$ upon $v_{[q]} \in V$ for this step, leading according to (5) to the node:

$$x^{*,v_{[q]}}_{k+j+1|k} = \arg\min_{u_{k+j|k} \in U} \{ \mathcal{L}(x_{k+j|k}, u_{k+j|k}) + \mathcal{V}(x_{k+j+1|k}) \}$$
$$\text{s.t.: } x_{k+j+1|k} = A_{v_{[q]}} x_{k+j|k} + B_{v_{[q]}} u_{k+j|k} \quad (19)$$
$$x_{k+j+1|k} \in X_{k+j+1|k}.$$

Since $\mathcal{V}(x_{k+j+1|k})$ in here is not yet known, it is substituted by its upper and lower bound:

$$x^{*,v_{[q]}}_{k+j+1|k} := \arg\min_{u_{(k+j|k)} \in U} \{ \underbrace{\mathcal{L} + \underline{\mathcal{V}}(x_{(k+j+1|k)})}_{\textit{Lower Bound}} + \underbrace{\mathcal{L} + \overline{\mathcal{V}}(x_{(k+j+1|k)})}_{\textit{Upper Bound}} \}$$
$$\text{s.t.: } x_{k+j+1|k} = A_{v_{[q]}} x_{k+j|k} + B_{v_{[q]}} u_{k+j|k}, \quad (20)$$
$$\underline{\mathcal{V}}(x_{(k+j+1|k)}) \text{ according to (15)},$$
$$\overline{\mathcal{V}}(x_{(k+j+1|k)}) \text{ according to (16)},$$
$$x_{k+i|k} \in X_{k+i|k},\ i \in \{j+1, \dots, N\},$$
$$u_{k+i|k} \in U,\ v_{k+i|k} := v^{apx}_{k+i|k},\ i \in \{j+1, \dots, N-1\}.$$

In comparison to (19), the new optimization problem minimizes the upper and lower bounds of the costs-to-go of the new node simultaneously. The continuous state constraints in (20) ensures the feasibility of solution $x^{*,v_{[q]}}_{k+j+1|k}$ as well as the corresponding state trajectory in the remaining steps from $k+j+1$ to $k+N$. If no solution is obtained for (20), the corresponding edge $v_{[q]}$ is not a feasible option. If the solution exists, a node corresponding to $x^{*,v_{[q]}}_{k+j+1|k}$ is inserted in the set $\mathbb{G}_{k+j+1|k}$ of the tree. The solution of (20) leads to relatively low effort, since only a QP problem is used.

Now, given the procedure to generate new nodes, criteria for comparing the nodes on one layer (for index $k+j+1$) with respect to their overall costs in (3), and for pruning those nodes which are definitely suboptimal. The following lemma obviously holds:

Lemma. 1: Let two different nodes $x^{v_{[m]}}_{k+j+1|k}$ and $x^{v_{[n]}}_{k+j+1|k}$ be generated on the layer $k+j+1$ as successors of node $x_{k+j|k}$ under two different discrete inputs $v_{(k+j|k)} := v_{[m]}$ and $v_{(k+j|k)} := v_{[n]}$. Then, if $\underline{\mathcal{V}}(x^{v_{[m]}}_{k+j+1|k}) > \overline{\mathcal{V}}(x^{v_{[n]}}_{k+j+1|k})$, the relation $\mathcal{V}(x^{v_{[m]}}_{k+j+1|k}) > \mathcal{V}(x^{v_{[n]}}_{k+j+1|k})$ follows.

Assuming that the continuous inputs to reach both nodes were also known, namely $u_{(k+j|k)} := u_{[m]} \in U$ and $u_{(k+j|k)} := u_{[n]} \in U$, then the overall costs $\Omega(x_k, x_f, \phi^v_{k,N}, \phi^u_{k,N})$ for first reaching $x_{k+j|k}$ in terms of (18), then reaching $x^{v_{[m]}}_{k+j+1|k}$ in step $k+j+1$, can be formulated as follows:

$$\Omega(x_k, x_f, \phi^v_{k,N}, \phi^u_{k,N}) = \Omega(x_k, x_{k+j|k}, \phi^v_{k,j}, \phi^u_{k,j}) + \mathcal{L}(x_{(k+j|k)}, u_{[m]}) + \mathcal{V}(x^{v_{[m]}}_{k+j+1|k}), \quad (21)$$

and likewise for $\Omega(x_k, x_f, \tilde{\phi}^v_{k,N}, \tilde{\phi}^u_{k,N})$ by reaching $x^{v_{[n]}}_{k+j+1|k}$. For the unknown terms $\mathcal{V}(x^{v_{[m]}}_{k+j+1|k})$ and $\mathcal{V}(x^{v_{[n]}}_{k+j+1|k})$, their range of values can be once more determined by (17), leading to ranges for $\Omega(x_k, x_f, \phi^v_{k,N}, \phi^u_{k,N})$ and $\Omega(x_k, x_f, \tilde{\phi}^v_{k,N}, \tilde{\phi}^u_{k,N})$. Now, if the lower bound of $\Omega(x_k, x_f, \phi^v_{k,N}, \phi^u_{k,N})$ is higher than the upper bound of $\Omega(x_k, x_f, \tilde{\phi}^v_{k,N}, \tilde{\phi}^u_{k,N})$, then reaching node $x^{v_{[m]}}_{k+j+1|k}$ will result in a higher overall cost and thus need not to be explored further according to Lemma 1 (and is eliminated from $\mathbb{G}_{k+j+1|k}$).

As a further means to reduce the search graph, the concept of adjacency of states as introduced in [12] can be applied to the remaining nodes in the set $\mathbb{G}_{k+j+1|k}$. Let again two different nodes in $\mathbb{G}_{k+j+1|k}$ be denoted by $x^{v_{[m]}}_{k+j+1|k}$ and $x^{v_{[n]}}_{k+j+1|k}$ (now they may not originate from the same node at last layer). They are said to be *adjacent* if:

$$\| x^{v_{[m]}}_{k+j+1|k} - x^{v_{[n]}}_{k+j+1|k} \|_2 \leq \ \gamma \quad (22)$$

holds for an appropriately chosen small $\gamma > 0$. For any adjacent pair of states, only the one with the smaller *accumulated costs* is kept in the set $\mathbb{G}_{k+j+1|k}$.

The following Algorithm 2 combines the details explained above in order to search the tree of possible discrete input sequences:

Algorithm 2 Search over the tree $\mathcal{T}_k$

Given: $\mathbb{G}_{k|k} = \{x_k\}$, U, V;
for $j = 0 : N-1$ **do**
  for $m = 1 : |\mathbb{G}_{k+j|k}|$ **do**
    for $q = 1 : n_v$ **do**
      compute $x^{*,v_{[q]}}_{(k+j+1|k)}$ for each $v_{[q]} \in V$, $x^m_{k+j|k} \in \mathbb{G}_{k+j|k}$ by using (20) and insert it into set $\mathbb{G}_{k+j+1|k}$. Determine the corresponding range of $\Omega(x_k, x_f, \phi^v_{k,N}, \phi^u_{k,N})$ by using (21)
    end for
  end for
  eliminate the nodes from $\mathbb{G}_{k+j+1|k}$, for which the lower bound of $\Omega(x_k, x_f, \phi^v_{k,N}, \phi^u_{k,N})$ is higher than the upper bound of the nodes in this set
  check the adjacency criterion according to (22) for any pair of nodes in $\mathbb{G}_{k+j+1|k}$ and (if satisfied) eliminate the node with a higher *accumulated costs*
end for

After executing Algorithm 2, the nodes still present in the search tree determine a subset of the discrete input sequences $\phi^v_{k,N}$ which connect the initial node with a node in $\mathbb{G}_{k+N|k}$. For any such $\phi^v_{k,N}$, the Problem 1 is once more solved for the continuous inputs, and the pair $(\phi^{v,*}_{k,N}, \phi^{u,*}_{k,N})$ leading to the minimal value of $\Omega(x_k, x_f, \phi^v_{k,N}, \phi^u_{k,N})$ is determined as the approximation of the solution of the original problem.

V. Numerical Evaluation

A. Test Series

As a first instance of evaluating the proposed scheme, the latter is applied to a set of 30 randomly created systems of the type in (1) with $n_x = 10$, $n_u = 8$, $n_v = 4$, $N = 8$, as well as chosen cost functions, constraints, and an adjacency parameter of $\gamma = 2$. Fig. 3 shows a summary of the results. In this test, the average deviation of using proposed method to the globally optimal solution is 1.16%, and the average computation time was $0.0802s$ (the global optimum is obtained by enumerating all possible ϕ^v with an average computation time of 5 minutes). By increasing the value of adjacency parameter to $\gamma = 5$, the average deviation from the globally optimal solution stays at 1.10%, and the average computation time is slightly reduced to $0.0651s$.

The figures 4 and 5 show the course of the average computation time (in seconds) for using the proposed technique for N increasing from 1 to 20, as well as the average relative deviation (in percent) from the globally optimal solution from $N = 1$ to $N = 10$.

B. Vehicle Platoon

To illustrate the use for an application, the method is now applied to a vehicle platooning example inspired from

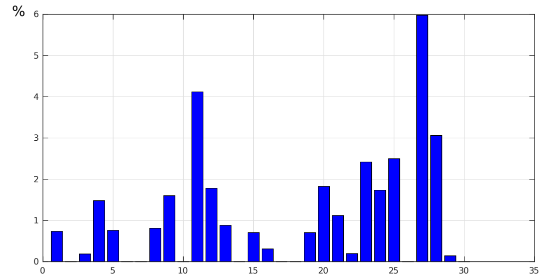

Fig. 3. Comparison of the costs obtained with the proposed method $\Omega^{*,a}$ and the globally optimal solution $\Omega^{*,b}$ for randomly generated switched systems: the ordinate shows the ratio $\frac{\Omega^{*,a} - \Omega^{*,b}}{\Omega^{*,b}}\%$ over the test index.

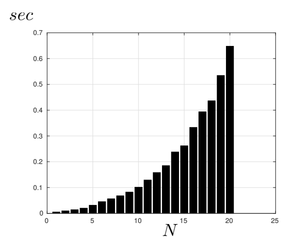

Fig. 4. Average time over N.

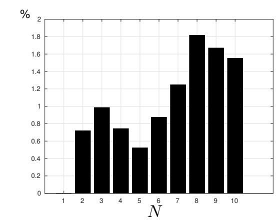

Fig. 5. Average deviation over N.

[7], but extended to constraints, discrete inputs, different vehicle dynamics. The example comprises 4 vehicles on a highway (one lane), which have to reach defined target positions. Each vehicle here is modelled by a switched system, where a discrete input represent a gear by which a mode of acceleration can be selected. The continuous input u corresponds to the gas-level of local vehicle. The vehicle-specific constraints models different vehicle-masses (which results in different accelerations under the same gas-level) and ranges of possible velocities and accelerations. Coupling constraints arise from keeping a desired distances between the vehicles ($d = 25$) in order to avoid collision. In addition, the local cost functions of each vehicle have different parametrization.

The considered horizon N is selected to be 15. Thus, based on the above description, overall a number of 36 modes over all vehicles results for a single time step. For $N = 15$, a number of 540 binary variables would be required for modeling, leading to $36^{15} \approx 2.2107 \times 10^{23}$ different discrete input sequences in each prediction. The initial and target state of each vehicle are given. When using the proposed

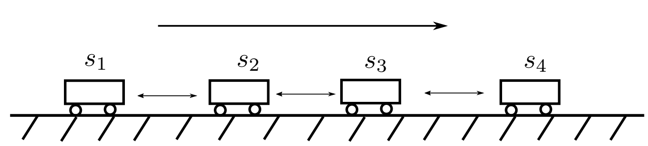

Fig. 6. A platoon of 4 vehicles are driving in the highway, over-take behavior between vehicles is not allowed.

method, the vehicles reach their target state after 19 steps with an average computation time of $0.9557s$ in each step. The following figures show the position, gear, velocity and gas-level of each vehicle (s_4 in black, s_3 in magenta, s_2 in blue, s_1 in red), and the vehicles reach their designated target states while satisfying all given constraints.

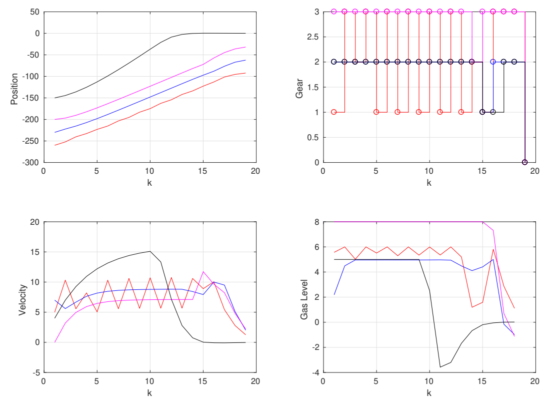

Fig. 7. Position, gear, velocity, and gas-level of each vehicle.

VI. Conclusion

In this paper, an approach for optimal control of discrete-time switched systems with time-varying state constraints and constant input constraints was proposed (the extension to time-varying input constraints is straightforward). The main asset of the proposed technique is that lower and upper cost bounds are derived, which serve to reduce the tree representing the possible discrete input sequences that may be applied. While the bounds depend on the cost function parametrization and the specific dynamics, it does (except of a parameter used in the adjacency criterion) not require a tuning factor to adjust the effectiveness of the bounds – such factors are typically very difficult to select as the pruning effect is very sensitive to them. The proposed scheme for obtaining the bounds turned out to be very effective with respect to reducing the computational time, while the distance to the true optimal solution was observed to be small for a larger set of test runs.

Current work comprises to extend the given scheme also to hybrid systems in which switching between different modes occurs autonomously (not as degree of freedom of the optimization).

Acknowledgement

This work was funded by the European Union through the H2020-project UnCoVerCPS under grant No. 643921.

References

[1] K. Mustafa B. Pierre and L. Jeff. Algorithms and software for convex mixed integer nonlinear programs. *Mixed integer nonlinear programming*, pages 1–39, 2012.

[2] M. Balandat, W. Zhang, and A. Abate. On infinite horizon switched LQR problems with state and control constraints. volume 61, pages 464–471. Elsevier, 2012.

[3] M. Bussieck and A. Pruessner. Mixed-integer nonlinear programming. *SIAG/OPT Newsletter: Views & News*, 14(1):19–22, 2003.

[4] D. Goerges, M. Izák, and S. Liu. Optimal control and scheduling of switched systems. *IEEE Transactions on Automatic Control*, 56(1):135–140, 2011.

[5] D. Gross, M. Jilg, and O. Stursberg. Design of distributed controllers and communication topologies considering link failures. In *Control Conference (ECC), 2013 European*, pages 3288–3292. IEEE, 2013.

[6] D. Gross and O. Stursberg. Optimized distributed control and network topology design for interconnected systems. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 8112–8117. IEEE, 2011.

[7] D. Groß and O. Stursberg. Distributed predictive control for a class of hybrid systems with event-based communication. *IFAC Proceedings Volumes*, 46(27):383–388, 2013.

[8] D. Kleinman. On an iterative technique for Riccati equation computations. *IEEE Trans. AC*, 13(1):114–115, 1968.

[9] G. Kocis and I. Grossmann. Computational experience with DICOPT solving MINLP problems in process systems engineering. *Computers & Chemical Engineering*, 13(3):307–315, 1989.

[10] J. Lee and S. Leyffer. *Mixed integer nonlinear programming*, volume 154. Springer Science & Business Media, 2011.

[11] B. Lincoln and A. Rantzer. Relaxing dynamic programming. *IEEE Transactions on Automatic Control*, 51(8):1249–1260, 2006.

[12] O. Stursberg. A graph search algorithm for optimal control of hybrid systems. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, volume 2, pages 1412–1417. IEEE, 2004.

[13] T. Westenbroek and H. Gonzalez. Optimal control of hybrid systems using a feedback relaxed control formulation. *arXiv preprint arXiv:1510.09127*, 2015.

[14] W. Zhang, A. Abate, J. Hu, and M. Vitus. Exponential stabilization of discrete-time switched linear systems. *Automatica*, 45(11):2526–2536, 2009.

[15] W. Zhang, J. Hu, and A. Abate. On the value functions of the discrete-time switched LQR problem. *IEEE Transactions on Automatic Control*, 54(11):2669–2674, 2009.