Optimal Trajectory Planning of Hybrid Systems by Efficient MIQP Encoding

Zonglin Liu¹ and Olaf Stursberg¹

Abstract—This paper proposes an efficient encoding for Mixed Integer Quadratic Programming (MIQP) problems in optimally controlling constrained hybrid systems from an initial state into a target region over a finite time horizon. The set of admissible trajectories given the system semantics is formulated by tailored constraints involving binary variables to encode the transition dynamics. A subset of these constraints establish a phase sequencing, which reduces the number of possible value combinations for the binary variables significantly, resulting in lower computation times for solving the MIQP problem. An illustrating example demonstrates the low computation times.

I. INTRODUCTION

In comparison to optimal control for continuous systems, the optimization of hybrid dynamic systems includes the additional challenge of considering the transition dynamics of the discrete state, often paired with instantaneous resets of the continuous state. In this work, we focus on hybrid systems formulated in discrete time and with transitions that are coupled to conditions specified for the continuous states as well as to discrete control inputs. The task of determining the optimal hybrid state trajectory from an initial state into a goal set thus involves the computation of trajectories for mixed inputs. The transition dynamics introduces conditional constraints on the continuous states, leading to integer variables in the problem formulation. The sequence of transitions as well as the discrete control inputs represent two sources of combinatorial complexity for the optimization, that typically lead to large numbers of value combinations for the integer variables, and thus large computation times [13].

Schemes to transform the hybrid dynamics into linear inequalities for integer and continuous variables within the context of optimal and predictive control have been proposed in the past, e.g. with respect to mixed logical dynamic systems [1], or for hybrid automata with linear continuous dynamics [12]. The obtained reformulated problems can then be solved by tools for mixed-integer programming. A common objective for such reformulations is certainly to keep the number of binary variables small. An issue which has not been addressed and solved satisfactory up to-date is how the number of value combinations of the necessary integer variables can be limited to the extent which refers to the set of admissible executions of the hybrid systems – this is the objective of the present paper.

It should be mentioned, of course, that a larger variety of direct and indirect methods exist to solve hybrid optimal control problems without the use of mixed-integer programming, e.g. [2]-[4], [8], [10], [11], [15]. More important for this investigation are those approaches, however, that aim at finding sequences of discrete states (and binary variables) which encode a particular goal-attaining temporal execution of the hybrid system. The work in [6], [7], [14] use, e.g. linear temporal logic [9] as a task-specification tool to force the obtained state trajectory satisfying the desired property of the task. However, the encoding of the LTL formula is often elaborate. As indicated in [14], the number of the binary variables used for encoding a single Until operator is quadratic in the time horizon. This work also aims at determining task specifications by encoding each attained discrete state as well as the guard set with binary variables. But instead of directly encoding the LTL formula as mixedinteger linear constraints, a matrix of binary variables is determined to formulate the trajectories leading from the initial to the goal state. Constraints are formulated for this matrix to impose a particular structure, leaving only value combinations of the binary variables that correspond to admissible executions. In comparison to the LTL encoding, the number of the binary variables one has to use is linear in the time horizon, and obtaining the globally optimal solution can still be guaranteed.

This paper is organized as follows: in Sec. II, the class of systems under consideration and the problem to be solved are specified. Section III explains how the transition dynamics can be cast into algebraic constraints for binary variables. Subsequently, Sec. IV specifies a reformulated optimization problem with reduced search space. A numerical example is presented in Sec. V, followed by conclusions.

II. PROBLEM FORMULATION

Let a hybrid system with mixed inputs be defined by HA = (T, U, X, Z, I, T, G, V, r, f), consisting of:

- the discrete time-domain $T = \{t_k \mid k \in \mathbb{N} \cup \{0\}, \Delta \in \mathbb{R}^{>0} : t_k := k \cdot \Delta\}$, where k is used in the following to refer to t_k ;
- the continuous input space U ⊆ ℝ^{nu} with the continuous input u ∈ U;
- the continuous state space X ⊆ ℝ^{nx} on which the state vector x is defined;
- the finite set of discrete states $Z = \{1, \dots, n_z\};$
- a set $I = \{I_1, \ldots, I_{n_z}\}$ of invariants where the invariant of any discrete state *i* is a polytope $I_i = \{x \mid n_{p_i} \in \mathbb{N}, C_i \in \mathbb{R}^{n_{p_i} \times n_x}, d_i \in \mathbb{R}^{n_{p_i}}, x \in X : C_i \cdot x \leq d_i\};$
- the finite set of transitions T ⊆ Z × Z, in which a transition from i ∈ Z to j ∈ Z is denoted by the ordered

^{*}This work was in part funded by the European Union through the H2020-project UNCOVERCPS under grant No. 643921.

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

pair $(i, j) \in \mathcal{T}$;

- the set G of guard sets contains one polytopic set $G_{(i,j)} = \{x \mid C_{(i,j)} \in \mathbb{R}^{n_{G_{(i,j)}} \times n_x}, d_{(i,j)} \in \mathbb{R}^{n_{G_{(i,j)}}}, x \in I_i : C_{(i,j)} \cdot x \leq d_{(i,j)}\}$ for any transition $(i,j) \in \mathcal{T}$; let for any pair of the outgoing transitions from the discrete state *i* the corresponding guard sets be disjoint, i.e. $G_{(i,j)} \cap G_{(i,l)} = \emptyset, \forall j \neq l;$
- the finite set V of discrete input variables, where any element $v_{(i,j)}$ in V refers to one transition $(i, j) \in \mathcal{T}$; the variable $v_{(i,j)}$ is a binary one, and for $v_{(i,j)} = 1$ the transition (i, j) is triggered if $x \in G_{(i,j)}$ applies; if $v_{(i,j)} = 0$ or $x \notin G_{(i,j)}$, the transition cannot occur;
- a reset function r: T × X → X which updates the state vector x upon a transition (i, j) ∈ T according to x' = E_(i,j) · x + e_(i,j);
- and the function f : X × U × Z → X defining the discrete-time continuous dynamics according to x_{k+1} = A_i · x_k + B_i · u_k with x_{k+1} := x(t_{k+1}), i ∈ Z, x_k ∈ I_i.

The execution of HA is defined as follows: assume a finite time set $T_N = \{0, 1, \ldots, N\} \subset T$, and let the initial states (x_0, z_0) satisfy $x_0 \in I_{z_0}$ and $x_0 \notin G_{(z_0,j)}$ for each $(z_0, j) \in \mathcal{T}$ for $j \in Z$. For given input sequences $\phi_u = \{u_0, u_1, \ldots, u_{N-1}\}$ and $\phi_v = \{v_0, v_1, \ldots, v_{N-1}\}$, the pair of state sequences $\phi_x = \{x_0, x_1, \cdots, x_N\}$ and $\phi_z = \{z_0, z_1, \cdots, z_N\}$ is *admissible*, if and only if for any $k \in \{0, \ldots, N\}$ the pair (x_{k+1}, z_{k+1}) follows from (x_k, z_k) according to the following steps:

1.) $x' := A_i \cdot x_k + B_i \cdot u_k \in I_i,$

2.) if
$$x' \in G_{(i,j)}$$
 and $v_k = 1$, then $x_{k+1} := r((i,j), x') \in I_j$ and $z_{k+1} := j$, else $x_{k+1} := x', z_{k+1} := i$.

The second step makes obvious that a transition is bound to the condition that a discrete control decision is imposed in addition to the fact that the intermediate state x' is contained in a guard set.

In order to introduce the control task, assume now that a set of hybrid goal states (X_g, z_g) is defined by one $z_g \in Z$ and $X_g = \{x \mid n_{p_g} \in \mathbb{N}, C_g \in \mathbb{R}^{n_{p_g} \times n_x}, d_g \in \mathbb{R}^{n_{p_g}}, x \in I_g : C_{X_g} \cdot x \leq d_{X_g}\}$. Furthermore, let a state $x_c \in X_g$ be specified (e.g. the volumetric center of X_g) to later define a distance to the goal region in a computationally easy way.

If (x_0, z_0) , (X_g, z_g) , and T_N are specified, the control objective is to find admissible state sequences ϕ_x and ϕ_z , or corresponding input sequences ϕ_u and ϕ_v respectively, which minimize an appropriate cost functional. Hereto, we define:

$$\mathcal{J}(x_0, x_f, N) = \sum_{k=1}^{N} \{ (x_k - x_{c,k}^{i,j})^{\mathrm{T}} Q(x_k - x_{c,k}^{i,j}) + (u_{k-1} - u_g)^{\mathrm{T}} R(u_{k-1} - u_g) \} + q_g \cdot N_g$$
(1)

where Q and R are positive-definite weighting matrices, and $q_g \in \mathbb{R}^{\geq 0}$. The variable $N_g := \min\{k \in \{1, \ldots, N\} \mid x_k \in X_g, z_k = z_g\}$ encodes the first point of time at which the continuous state has reached the goal set. We assume that (u_k, v_k) exists for $k \in \{N_g, \ldots, N\}$ to hold the system in the goal set. For any $k \in \{1, \ldots, N\}$ with $z_k \neq z_g$, the state $x_c^{i,j}$ encodes the center of the guard set $G_{(i,j)}$, if $x_k \in I_i$ and

if z_k is left through the transition $(i, j) \in \mathcal{T}$. Thus, any term of the sum in (1) encodes the weighted distance to the guard set, which can be seen as a *temporary goal set* while HA is in the discrete state z_k . For $z_k = z_g$, we require $x_c^{i,j} = x_c$.

The overall control problem can then be defined as:

Problem 1: For *HA* initialized to (x_0, z_0) , let a time set T_N and a goal (X_g, z_g) be given. Then, determine input sequences ϕ_u^* and ϕ_v^* as the solution of:

$$\min_{\phi_u,\phi_v} \mathcal{J}(x_0, x_f, N)$$
(2)
s.t.: ϕ_u with $u_k \in U, k \in \{0, \dots, N-1\}$
 ϕ_v with $v_k \in \{0, 1\}, k \in \{0, \dots, N-1\}$
 ϕ_x, ϕ_z admissible for $HA, x_N \in X_g, z_N = z_g$.

The solution of this problem is difficult for large values of N (a parameter for which a sufficiently high value to reach (X_q, z_q) is not known a-priori), due to the combinatorics in ϕ_z and ϕ_v . Note that expressing the conditions for x_k being contained in invariants and guard sets for certain discrete states or transitions implies to use binary variables, when converting Problem 1 into a form that can be processed by available solvers. In addition, solving the problem also includes to decide (by the discrete inputs v_k) whether taking a transition upon reaching a guard set is better in terms of feasibility and costs than continuing to stay in the current discrete state. This is different from most other settings in existing literature on optimal control of hybrid systems. The following sections propose a new approach to approximate the optimal solution to the MIQP problem formulated by Prob. 1 efficiently in many cases.

III. REPRESENTATION OF ADMISSIBLE TRAJECTORIES BY ALGEBRAIC PROGRAMS

This section introduces a particular format to encode Prob. 1 as algebraic program with a number of binary variables that is relatively small compared to other formulations. It is well-known that implications like $(x_k \in I_i) \Leftrightarrow$ (b = 1) for mapping invariant set containment of x_k into a binary variable b can be accomplished by rules as those explained in [5] (often referred to the Big-M-approach). Such mechanisms have been re-used in different work on hybrid system optimization, e.g. [1] and [12], but the particular challenge is to use an as small as possible number of binary variables and constraints on these variables for low computational times. This issue is addressed in the following for Prob. 1. To facilitate the description and understanding of the procedure, we first refer to the simplified case that a phase sequence is known: let the order of the discrete states Z by which HA passes through be known, but the times in T_N at which the discrete states are left or are reached still have to be determined. Hence, the remaining task is to determine the transition times as well as ϕ_u and ϕ_v such that ϕ_x is led (if possible) through the appropriate series of invariants and guards. Formally, a phase sequence is denoted by $\phi_p = \{p_0, \dots, p_L\}$, where p_l with $l \in \{0, \dots, L\}$ is set to the index of the discrete state which is invariant in the l-th phase (i.e. $\phi_p \subset \phi_z$ is obtained from eliminating consecutive equal elements in ϕ_z).

The phases are now important to identify the number of binary variables required to encode the execution of *HA* within the optimization problem: consider a phase p_i , as shown in Fig. 1, from a hybrid state (x_k, z_k) with $z_k = i$ (reached by a preceding transition) up to the state (x_{k+5}, z_{k+5}) with $z_{k+5} = j$, reached through the transition (i, j). Note that $x' \in G_{(i,j)}$ is an intermediate state, which is immediately transferred into $x_{k+5} := r((i, j), x') \in I_j$ by the transition with reset upon $v_{k+4} = 1$, according to the definition of an admissible run above. Two points are obvious from this figure: (1.) for any of the states $\{x_k, \ldots, x_{k+4}, x'\}$ the same invariant constraint (element of I_i) applies, i.e. one binary variable per phase is sufficient to express this fact; (2.) the state x' must be associated with an additional binary variable to encode $x' \in G_{(i,j)}$ for p_i . Since x' must be treated separately, we use an extended index set for the states to be considered: $k \in \{0, N + L\}$. Within this set, the following assignments correspond to an admissible run of HA:

- k = 0 refers to x_0 ;
- L values indicate intermediate states x', and thus an exit from a discrete state;
- L values belong to the entry into a newly reached discrete state;
- and one value encodes the entry into X_g .

Next, the constraints on the continuous states $x_{\tilde{k}}$ have to be formulated suitably. Recall that all invariants, guard sets, and X_g are given as polytopic sets. Exemplarily for an invariant set I_i , the efficient algebraic encoding is explained: using the principles proposed in [5], the constraint $C_i \cdot x_{\tilde{k}} \leq d_i$ can be modeled equivalently by:

$$C_i \cdot x_{\tilde{k}} \le d_i + b_{i\,\tilde{k}} \cdot M_i \tag{3}$$

if $M_i \in \mathbb{R}^{n_{p_i} \times 1}$ is a vector of large constants, and $b_{i,\tilde{k}} \in \{0,1\}$ one binary variable. If $b_{i,\tilde{k}} = 0$, the invariant constraint is enforced, while $b_{i,\tilde{k}} = 1$ relaxes the constraint. Likewise, a guard constraint $x_{\tilde{k}} \in G_{(i,j)}$ results in:

$$C_{(i,j)} \cdot x_{\tilde{k}} \le d_{(i,j)} + b_{(i,j),\tilde{k}} \cdot M_{(i,j)}.$$
(4)

Consider that two binary variables are required per phase (one for the invariant conditions, and one for the guard condition (or the terminal set, respectively)), we introduce a vector of $2 \cdot (L+1)$ binary variables:

$$\mathbf{b}_{\tilde{k}} = [b_{0,\tilde{k}}, b_{(0,1),\tilde{k}}, b_{1,\tilde{k}}, \dots, b_{L,\tilde{k}}, b_{(L,X_g),\tilde{k}}]^{\mathrm{T}}$$
(5)



Fig. 1. Execution of HA within one phase.

for each $\tilde{k} \in \{0, N + L\}$. The last entry represents containment in the goal set X_g . For $\tilde{k} = 0$, the numeric values of this vector are, $\mathbf{b}_0 = [0, 1, \dots, 1]^{\mathrm{T}}$, and for the transition from phase *i* to i + 1 we have: (a) $\mathbf{b}_{\tilde{k}} = [1, \dots, 1, \underbrace{0}_{2i+1}, \underbrace{0}_{2i+2}, 0, 1, \dots, 1]^{\mathrm{T}}$ corresponding to the intermediate state x', and (b) $\mathbf{b}_{\tilde{k}} = [1, \dots, 1, \underbrace{0}_{2i+3}, 1, \dots, 1]^{\mathrm{T}}$ for

the entry in the next invariant. For $\tilde{k} = N + L$, the vector is: $\mathbf{b}_{N+L} = [1, \dots, 1, 0, 0]^{\mathrm{T}}$, and all of these vectors are collected in a matrix:

The last line refers to the time indexing, where $\tilde{k} = \tilde{k}_0^{out}$ refers to the instance in which the first invariant I_0 is left, and $\tilde{k} = \tilde{k}_1^{in}$ to the instance in which the second invariant of ϕ_z is reached. The following holds by construction:

Lemma 1: If ϕ_x and ϕ_z determine an admissible run of HA with $z_N = z_g$ and $x_N \in X_g$, then a matrix $\mathcal{B}_m \in \{0,1\}^{(2L+2)\times(N+L+1)}$ exists according to the rules (3) to (6), and each column in \mathcal{B}_m uniquely determines which constraints apply to $x_{\tilde{k}}$ for $\tilde{k} \in \{0, N+L\}$.

Let all constraints of the form (3) and (4) be collected in the order of the indexing of $x_{\tilde{k}}$ in:

$$\mathcal{C} \cdot x_{\tilde{k}} \le \mathcal{D} + diag(\mathcal{B}_m(:, \tilde{k} + 1)) \cdot \mathcal{M}.$$
(7)

The search for an admissible run ϕ_x and ϕ_z thus means to satisfy (7) for all $\tilde{k} \in \{0, \dots, N+L\}$. While $(2L+2) \times$ (N+L) binary variables $(b_{0,\tilde{k}}$ is known) encode in principle $2^{(2L+2)\times(N+L)}$ combinations (prohibitively many for larger N and L), the particular structure of \mathcal{B}_m reduces the number of possible combinations (and thus of ϕ_z) considerably. The following section proposes a scheme to efficiently exploit this structure in searching for an optimal ϕ_x and ϕ_z .

IV. FORMULATION OF THE OPTIMIZATION PROBLEM

In order to explain how \mathcal{B}_m enables to search only over those value combinations of binary variables that represent admissible runs of HA, we first focus on the first two rows of the matrix. They represent the values of the binary variables $b_{0,\tilde{k}}, b_{(0,1),\tilde{k}}$ over $\tilde{k} \in \{0, N+L\}$, and these variables model that $x_{\tilde{k}}$ is contained in the invariant of the first discrete state (value 0), and respectively, that the first transition is triggered (again value 0):

$$\begin{bmatrix} \mathcal{B}_m(1,:) \\ \mathcal{B}_m(2,:) \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 & 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 & 0 & 1 & 1 & \cdots & 1 \end{bmatrix}.$$
 (8)

Note that the column in which $\mathcal{B}_m(1,:)$ changes from 0 to 1 is not yet determined. Let the value of $\mathcal{B}_m(1, \tilde{k} + 1)$ depend on an auxiliary vector $\mathbf{d}_{1,\tilde{k}+1}^{\mathrm{T}} = [\mathcal{B}_m(1,\tilde{k}), \mathcal{B}_m(2,\tilde{k})]$ according to:

$$\mathcal{B}_m(1,\tilde{k}+1) = \begin{cases} 0\\1 \end{cases} \text{ if } \mathbf{d}_{1,\tilde{k}+1}^{\mathrm{T}} = \begin{cases} [0,1]\\[0,0] \text{ or } [1,1] \end{cases}.$$
(9)

Now, define two parameter/vectors $\alpha_1 \in \mathbb{R}^{3 \times 1}$ and $\beta_1 \in \mathbb{R}^{3 \times 1}$ satisfying the following conditions:

$$\begin{bmatrix} -\infty \\ 0 \\ 0 \end{bmatrix} < \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} \cdot \alpha_1 (1:2) + \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \cdot \alpha_1 (3) < \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \\\begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} < \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} \cdot \beta_1 (1:2) + \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \cdot \beta_1 (3) < \begin{bmatrix} 1 \\ \infty \\ \infty \end{bmatrix},$$
(10)

where the matrices in front of the vectors $\alpha_1(1:2)$ and $\beta_1(1:2)$ encode the possible values of $\mathbf{d}_{1,\tilde{k}+1}^{\mathrm{T}}$ in (9). Then the relation (9) can be algebraically and equivalently formulated as:

$$\mathcal{B}_{m}(1, \tilde{k} + 1) \geq \alpha_{1}^{\mathrm{T}}(1:2) \cdot \mathbf{d}_{1,\tilde{k}+1} + \alpha_{1}(3), \mathcal{B}_{m}(1, \tilde{k} + 1) \leq \beta_{1}^{\mathrm{T}}(1:2) \cdot \mathbf{d}_{1,\tilde{k}+1} + \beta_{1}(3).$$
(11)

While this encoding relates to the first phase, the principle can be transferred to the subsequent phases. For a phase with index $l \in \{1, \dots, L-1\}$, the (2l+1)st row of \mathcal{B}_m is relevant. It refers to the binary variable $b_{l,\tilde{k}}$, and the value of $\mathcal{B}_m(2l+1, \tilde{k}+1)$ is written depending on an auxiliary vector $\mathbf{d}_{2l+1, \tilde{k}+1}^{\mathrm{T}} = [\mathcal{B}_m(2l, \tilde{k}), \mathcal{B}_m(2l+1, \tilde{k}), \mathcal{B}_m(2l+2, \tilde{k})]$:

$$\mathcal{B}_{m}(2l+1,\tilde{k}+1) = \begin{cases} 0\\1 \end{cases}$$

if $\mathbf{d}_{2l+1,\tilde{k}+1}^{\mathrm{T}} = \begin{cases} [0,1,1] \text{ or } [1,0,1]\\[1,1,1] \text{ or } [1,0,0] \end{cases}$. (12)

If parameter vectors $\alpha_l \in \mathbb{R}^{4 \times 1}$ and $\beta_l \in \mathbb{R}^{4 \times 1}$ are defined similarly to (10), the assignment (12) can be equivalently formulated as:

$$\mathcal{B}_{m}(2l+1,\tilde{k}+1) \geq \alpha_{l}^{\mathrm{T}}(1:3) \cdot \mathbf{d}_{2l+1,\tilde{k}+1} + \alpha_{l}(4), \mathcal{B}_{m}(2l+1,\tilde{k}+1) \leq \beta_{l}^{\mathrm{T}}(1:3) \cdot \mathbf{d}_{2l+1,\tilde{k}+1} + \beta_{l}(4).$$
(13)

With respect to the penultimate row of \mathcal{B}_m , which refers to $b_{L,\tilde{k}}$, the value of $\mathcal{B}_m(2L+1,\tilde{k}+1)$ depends likewise on an auxiliary vector $\mathbf{d}_{2L+1,\tilde{k}+1}^{\mathrm{T}} = [\mathcal{B}_m(2L,\tilde{k}), \mathcal{B}_m(2L+1,\tilde{k})]$ with:

$$\mathcal{B}_{m}(2L+1,\tilde{k}+1) = \begin{cases} 0\\1 \end{cases} \text{ if } \mathbf{d}_{2L+1,\tilde{k}+1}^{\mathrm{T}} = \begin{cases} [0,1] \text{ or } [1,0]\\[1,1] \end{cases}.$$
(14)

Using parameter vectors $\alpha_g \in \mathbb{R}^{3 \times 1}$ and $\beta_g \in \mathbb{R}^{3 \times 1}$, (14) is translated into:

$$\mathcal{B}_{m}(2L+1, \tilde{k}+1) \geq \alpha_{g}^{\mathrm{T}}(1:2) \cdot \mathbf{d}_{2L+1, \tilde{k}+1} + \alpha_{g}(3),$$

$$\mathcal{B}_{m}(2L+1, \tilde{k}+1) \leq \beta_{g}^{\mathrm{T}}(1:2) \cdot \mathbf{d}_{2L+1, \tilde{k}+1} + \beta_{g}(3).$$

(15)

For any 2*l*-th row of \mathcal{B}_m (with $l \in \{1, \dots, L\}$), which refers to $b_{(l-1,l),\tilde{k}}$, only one entry equals 0 (indicating that the reset is only triggered once), what can be enforced by:

$$\sum_{\tilde{k}=0}^{N+L} \mathcal{B}_m(2l, \tilde{k}+1) = N+L, \quad \forall l \in \{1, \cdots, L\}.$$
(16)

Finally, for the last row, referring to $b_{(L,X_g),\tilde{k}}$, only the last entry $\mathcal{B}_m(2L+2, N+L+1)$ is forced to 0, modeling $x_N \in X_q$. This is translated into:

$$\mathcal{B}_m(2L+2, N+L+1) = 0.$$
(17)

The condition that $x_{\tilde{k}} \in X_g$ if $x_{\hat{k}} \in X_g$ for $\tilde{k} \ge \hat{k}$ is modeled by:

$$\mathcal{B}_m(2L+2,\tilde{k}) \ge \mathcal{B}_m(2L+2,\tilde{k}+1).$$
(18)

Note that the options considered for $\mathbf{d}_{1,\tilde{k}+1}^{\mathrm{T}}$ in (9), for $\mathbf{d}_{2l+1,\tilde{k}+1}^{\mathrm{T}}$ in (12), and for $\mathbf{d}_{2L+1,\tilde{k}+1}^{\mathrm{T}}$ in (14) are sufficient to encode the part of \mathcal{B}_m which corresponds to the change of phases. Using this fact, and the constructive rules provided above to determine the linear inequalities formulated for elements of \mathcal{B}_m , the following fact can be established:

Lemma 2: If a binary matrix $\mathcal{B}_m \in \{0,1\}^{(2L+2)\times(N+L+1)}$ with first column $\mathcal{B}_m(:,1) = \mathbf{b}_0$ and last column $\mathcal{B}_m(:,N+L+1) = \mathbf{b}_{N+L}$ satisfies the constraints (11), (13), and (15) to (18), then it has the same structure as in (6).

Lemma 1 and 2 together also imply that these constraints encode the set of admissible trajectories of HA. All constraints introduced for the matrix \mathcal{B}_m can be collected in the set of linear constraints:

$$\mathcal{Q} \cdot \mathcal{B}_m \le \mathcal{W} + \mathcal{N},\tag{19}$$

where the matrices Q, W, and N depend on the various parameter vectors α and β . The constraints in (19) reduce the value combinations of the respective binary variables in \mathcal{B}_m from $2^{(2L+2)\times(N+L)}$ to $\binom{N+L}{2L}$.

The search for an admissible run ϕ_x and ϕ_z of *HA* now means to let \mathcal{B}_m satisfy (7) and (19), i.e. the transformed problem is:

Problem 2: For a given phase sequence ϕ_p , determine input sequences ϕ_u^* and a matrix \mathcal{B}_m^* as solution to:

$$\min_{\phi_u, \mathcal{B}_m} \sum_{\tilde{k}=0}^{N+L} \{ (\overline{x}_{\tilde{k}+1} - \overline{x}_{c,\tilde{k}+1})^{\mathrm{T}} Q (\overline{x}_{\tilde{k}+1} - \overline{x}_{c,\tilde{k}+1})$$
(20a)

+
$$(u_{\tilde{k}} - u_g)^{\mathrm{T}} R(u_{\tilde{k}} - u_g)$$
} + $q_g \cdot \sum_{\tilde{k}=0}^{N+L} \mathcal{B}_m(2L+2, \tilde{k}+1)$

s.t.:
$$Q \cdot \mathcal{B}_m \le \mathcal{W} + \mathcal{N};$$
 (20b)
for $\tilde{k} \in \{1, \dots, N+L\}:$

$$\overline{x}_{\tilde{k}} \le x_{\tilde{k}} + \lambda_x \cdot (L - \sum_{i=1}^{L} \mathcal{B}_m(2i, \tilde{k} + 1)),$$
(20c)

$$\overline{x}_{\tilde{k}} \ge x_{\tilde{k}} - \lambda_x \cdot (L - \sum_{i=1}^{L} \mathcal{B}_m(2i, \tilde{k} + 1)),$$
(20d)

$$\overline{x}_{\tilde{k}} \le \lambda_x \cdot (\sum_{i=1}^L \mathcal{B}_m(2i, \tilde{k}+1) + 1 - L), \qquad (20e)$$

$$\overline{x}_{\tilde{k}} \ge -\lambda_x \cdot (\sum_{i=1}^{L} \mathcal{B}_m(2i, \tilde{k}+1) + 1 - L);$$
(20f)

$$\overline{x}_{c,\tilde{k}} = \sum_{i=1}^{L} (1 - \mathcal{B}_m(2i, \tilde{k})) \cdot x_c^{i-1,i};$$
(20g)

$$x_{\tilde{k}} = \sum_{i=0}^{L} [A_i \cdot \xi_{\tilde{k},i} + B_i \cdot \pi_{\tilde{k},i}] + \sum_{i=0}^{L-1} \xi_{\tilde{k},(i,i+1)}; \qquad (20h)$$

$$\mathcal{C} \cdot x_{\tilde{k}} \leq \mathcal{D} + diag(\mathcal{B}_m(:, \tilde{k} + 1)) \cdot \mathcal{M}, \ u_{\tilde{k} - 1} \in U;$$
 (20i)

for
$$i \in \{0, \dots, L-1\}$$
:
 $\xi_{\tilde{k}, i} \leq \Theta_i^+ \cdot (\mathcal{B}_m(2i+2, \tilde{k}) - \mathcal{B}_m(2i+1, \tilde{k})),$ (20j)

$$\xi_{\tilde{k},i} \ge \Theta_i^- \cdot (\mathcal{B}_m(2i+2,\tilde{k}) - \mathcal{B}_m(2i+1,\tilde{k})),$$
(20k)

$$\xi_{\tilde{k},i} \le x_{\tilde{k}-1} + \lambda_x \cdot (1 - \mathcal{B}_m(2i+2,\tilde{k}) + \mathcal{B}_m(2i+1,\tilde{k})),$$
(201)

$$\xi_{\tilde{k},i} \ge x_{\tilde{k}-1} - \lambda_x \cdot (1 - \mathcal{B}_m(2i+2,\tilde{k}) + \mathcal{B}_m(2i+1,\tilde{k})),$$
(20m)

$$\pi_{\tilde{k},i} \le \Theta_u^+ \cdot (\mathcal{B}_m(2i+2,\tilde{k}) - \mathcal{B}_m(2i+1,\tilde{k})), \tag{20n}$$

$$\pi_{\tilde{k}_i} \ge \Theta_u^- \cdot (\mathcal{B}_m(2i+2,\tilde{k}) - \mathcal{B}_m(2i+1,\tilde{k})), \tag{200}$$

$$\pi_{\tilde{k},i} \le u_{\tilde{k}-1} + \lambda_u \cdot (1 - \mathcal{B}_m(2i+2,\tilde{k}) + \mathcal{B}_m(2i+1,\tilde{k})),$$
(20p)

$$\pi_{\tilde{k},i} \ge u_{\tilde{k}-1} - \lambda_u \cdot (1 - \mathcal{B}_m(2i+2,\tilde{k}) + \mathcal{B}_m(2i+1,\tilde{k})),$$
(20q)

$$\xi_{\tilde{k},(i,i+1)} \le \Theta_{i+1}^+ \cdot (1 - \mathcal{B}_m(2i+2,\tilde{k})), \tag{20r}$$

$$\xi_{\tilde{k},(i,i+1)} \ge \Theta_{i+1}^{-} \cdot (1 - \mathcal{B}_m(2i+2,\tilde{k})), \tag{20s}$$

$$\xi_{\tilde{k},(i,i+1)} \le E_{(i,i+1)} \cdot x_{\tilde{k}-1} + e_{(i,i+1)} + \lambda_x \cdot \mathcal{B}_m(2i+2,\tilde{k}),$$
(20t)

$$\xi_{\tilde{k},(i,i+1)} \ge E_{(i,i+1)} \cdot x_{\tilde{k}-1} + e_{(i,i+1)} - \lambda_x \cdot \mathcal{B}_m(2i+2,\tilde{k})$$
(20u)

The cost function (20a) is an equivalent reformulation of the one in Prob. 1, where $\overline{x}_{c,\tilde{k}}$ depends on the guard set relevant for \tilde{k} , according to (20g). The sum in the last term of (20a) counts the total number of steps in which $x_{\tilde{k}}$ is not in X_q .

The constraints (20c) to (20f) ensure that the costs induced by the intermediate states x' are not recorded in the cost function. The conditions (20b) and (20i) force the resulting trajectory ϕ_x to comply to ϕ_p . The equations and inequalities (20h) and (20j) to (20u) refer to standard reformulations of the hybrid dynamics by introducing auxiliary variables $\xi_{\tilde{k},i}$, $\pi_{\tilde{k},i}$, and $\xi_{\tilde{k},(i,i+1)}$. Details of such reformulations can be found in [12]. In addition, the following parameters have to be determined:

$$\Theta_i^+ = \begin{bmatrix} \max_{x \in I_i} x_1 & \cdots & \max_{x \in I_i} x_{n_x} \end{bmatrix}^{\mathrm{T}}, \qquad (21)$$
$$\Theta_u^+ = \begin{bmatrix} \max_{u \in U} u_1 & \cdots & \max_{u \in U} u_{n_u} \end{bmatrix}^{\mathrm{T}},$$

and likewise for minimal values in Θ_i^- and Θ_u^- . The relaxation vectors $\lambda_x \in \mathbb{R}^{n_x}$, $\lambda_u \in \mathbb{R}^{n_u}$ are selected, to have for all $x \in X$ and $u \in U$:

$$\begin{aligned} x + \lambda_x &\gg 0^{n_x \times 1}, \quad x - \lambda_x \ll 0^{n_x \times 1}, \\ u + \lambda_u &\gg 0^{n_u \times 1}, \quad u - \lambda_u \ll 0^{n_u \times 1}. \end{aligned}$$
(22)

Since all constraints in Prob. 2 are linear, the optimization represents an MIQP problem, which can be solved by existing solvers. The constraints (20b) reduce the possible combinations of values for the binary variables significantly. The obtained \mathcal{B}_m^* determines ϕ_v^* straightforwardly.

Furthermore, since (20b) admits all possible values of \mathcal{B}_m corresponding to the structure in (6) and since no approximation is involved, the following applies:

Corollary 1: If no feasible solution exists to Problem 2, then there exists no *admissible* trajectory corresponding to the given phase sequence ϕ_p .

Thus, Prob. 2 can be used to verify the existence of an *admissible* trajectory satisfying Prob. 1 for the considered ϕ_p .

Theorem. 1: If the solution of Problem 2 returns a feasible solution ϕ_u^* and \mathcal{B}_m^* , then it represents the optimal solution of Problem 1 for the given phase sequence ϕ_p . \Box

This result follows from the relation between Prob. 2 and Prob. 1 for the given ϕ_p as established by Lemma 1 and 2, and from the fact that solvers for MIQP problem terminate with the optimal solution if the search tree is fully explored.

If now Prob. 1 is addressed without restriction to certain single ϕ_p , the solution is obtained by solving one instance of Prob. 2 for any possible phase sequence connecting z_0 with z_g . If the number of possible phase sequences connecting the initial discrete state z_0 and the target state z_g is not very high¹, the search can be carried out by enumeration.

V. NUMERIC EXAMPLES

To illustrate the procedure, we consider the example of an HA with $x \in \mathbb{R}^3$ and 5 discrete states Z = $\{z_0, z_1, z_2, z_3, z_g\}$. The invariant sets of these states are marked by yellow regions, and the guard sets by orange regions in the following figures. The continuous dynamics, reset functions, and input constraints are parametrized suitably (but not shown here for brevity), and the set of transitions follows from the adjacency of the invariant sets. The initial state is $x_0 = [12, -7, 0]^T \in I_0$, and the terminal state is set to $x_g = [-2, -12, -2]^T \in I_g$. The terminal region X_g is marked as a green region in the figures, and N is first selected to be 15, which leads to a number of 102 binary variables to be employed in Problem 2. Three different phase sequences are possible, and the respective trajectories are shown in Figs. 2 and 3. Only for $\phi_p = \{z_0, z_2, z_g\}$ and $\phi_p = \{z_0, z_3, z_q\}$ optimal admissible trajectories are found with N = 15, leading to costs of 3135.18 and 3429.26, and requiring computation times of 0.080sec and 0.096sec on a 3.4GHz processor using Matlab 2015a and the solver CPLEX. Through constraint (19), the relevant combinations of the binary variables are reduced from 2^{102} to $\binom{17}{4} = 2380$, and the time to verify the infeasibility of $\phi_p = \{z_0, z_1, z_g\}$ for N = 15 is about 0.01sec. If, for the latter ϕ_p , the

¹As applies not seldomly for hybrid models

time horizon is increased to N = 25, then the admissible trajectory shown in Fig. 3 is obtained with optimal cost of 6160.51 computed in 0.717*sec*. A further test with a longer ϕ_p using L = 3 and a horizon N = 24 is illustrated in Fig. 4, obtained in 1.06*sec*.

VI. CONCLUSION

In this paper, a new method for trajectory optimization of hybrid systems has been proposed. The key aspect of the method is to cast the semantics of admissible trajectories into a tailored set of linear constraints which reduce the value combinations of binary variables required to formulate the transition dynamics. The significant reduction of the number of value combinations, also reduces the search space of the underlying MIP, and thus increases the computational efficiency. The procedure does not involve approximation and thus ensures that the globally optimal solution is found.

Future work will explore the use of tailored binary matrices and constraints without pre-specifying the phase sequence, and for specifications provided in temporal logic.

REFERENCES

- [1] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, 1999.
- [2] S. Branicky, S. Borkar, and K. Mitter. A unified framework for hybrid control: Model and optimal control theory. *IEEE transactions on automatic control*, 43(1):31–45, 1998.
- [3] M. Egerstedt, Y. Wardi, and F. Delmotte. Optimal control of switching times in switched dynamical systems. In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, volume 3, pages 2138–2143. IEEE, 2003.
- [4] S. Hedlund and A. Rantzer. Optimal control of hybrid systems. In Conference on Decision and Control, pages 3972–3977, 1999.
- [5] H.P.Williams. *Model Building in Mathematical Programming*. Wiley, 1st edition, 1978.
- [6] S. Karaman, G. Sanfelice, and E. Frazzoli. Optimal control of mixed logical dynamical systems with linear temporal logic specifications. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 2117–2122. IEEE, 2008.
- [7] V. Nenchev, C. Belta, and J. Raisch. Optimal motion planning with temporal logic and switching constraints. In *Control Conference* (ECC), 2015 European, pages 1141–1146. IEEE, 2015.



Fig. 2. Optimal trajectory for $\phi_p = \{z_0, z_2, z_g\}$ and $\phi_p = \{z_0, z_3, z_g\}$.



Fig. 3. Optimal trajectory for $\phi_p = \{z_0, z_1, z_g\}$ with N = 25.



Fig. 4. Optimal trajectory for $\phi_p = \{z_0, z_1, z_2, z_g\}$ with N = 24.

- [8] B. Passenberg, M. Kröninger, G. Schnattinger, M. Leibold, O. Stursberg, and M. Buss. Initialization concepts for optimal control of hybrid systems. *IFAC Proceedings Volumes*, 44(1):10274–10280, 2011.
- [9] A. Pnueli. The temporal logic of programs. In Foundations of Computer Science, 1977., 18th Annual Symposium on, pages 46–57. IEEE, 1977.
- [10] P. Riedinger, F. Kratz, C. Iung, and C. Zanne. Linear quadratic optimization for hybrid systems. In *Conference on Decision and Control*, pages 3059–3064, 1999.
- [11] M. S. Shaikh and P. E. Caines. On the hybrid optimal control problem: Theory and algorithms. *IEEE Transaction on Automatic Control*, 52:1587–1603, 2007, Corrigenda, ibid. 54:1440-1440, 2009.
- [12] O. Stursberg and S. Panek. Control of switched hybrid systems based on disjunctive formulations. *Hybrid systems: Computation and control*, pages 820–871, 2002.
- [13] J. Till, S. Engell, S. Panek, and O. Stursberg. Applied Hybrid System Optimization - An Empirical Investigation of Complexity. *Control Engineering Practice*, 12(10):1269–1278, 2004.
- [14] E. Wolff, U. Topcu, and R. Murray. Optimization-based trajectory generation with linear temporal logic specifications. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 5319–5325. IEEE, 2014.
- [15] X. Xu and P. J. Antsaklis. Optimal control of switched systems based on parameterization of the switching instants. *IEEE Transactions on Automatic Control*, 49:2–16, Jan. 2004.