Abstract—Model Predictive Control (MPC) has revolutionized the standards for control of smooth dynamical systems, allowing the synthesis of reliable optimal controllers which can naturally cope with constraints, and whose intuitive parameterization greatly simplifies control design. However, the discrete nature of hybrid systems, which lends itself naturally to transcription as a mixed-integer optimization problem, results in programs that are generally considered to be too computationally expensive to be used in real-time control applications. Although the branch-and-bound algorithm has a worst-case cost that is exponential in the number of integer variables, in MPC we expect to solve a nearly identical optimization problem on the very next time step. Surprisingly, the concept of warm-starting the solver (by reusing the computation from the previous time step), that has had great success in optimal control of linear systems, has not yet been used effectively in mixed-integer MPC formulations. In this paper we present a novel warm-start technique for hybrid MPC that uses dual solutions from one time step to imply rigorous lower bounds on the solution tree for the next time step, even in the presence of disturbances, allowing branch-and-bound to prune very quickly. In practice, we find that this method often reduces computation to the combinatoric complexity of only a one-step lookahead problem.

I. INTRODUCTION

In the last two decades, hybrid systems have been a major focus in control, and the community has significantly increased the understanding of the characteristic phenomena of these dynamical systems [1]. However, state-of-the-art hybrid control techniques are not yet nearly as reliable, simple, and broadly applicable as the ones available for smooth systems. Model Predictive Control (MPC) is a numerical method that enables the design of optimal feedback controllers for a wide variety of hybrid systems [2], [3]. The main idea behind it is straightforward: if we are able to solve trajectory optimization problems quickly enough, we can replan the future motion of the system at every sampling time and achieve a reactive behavior. The simplicity and the wide applicability of hybrid MPC has however a cost: the discrete phenomena that hybrid systems can exhibit have to be modeled with integer variables, requiring the online solution of Mixed-Integer Convex Programs (MICPs), which is considered to be unpractical even for systems with “slow” dynamics and of moderate size.

MICPs are NP-complete problems and, as such, no polynomial-time algorithm is known for their solution. Even though heuristic [4] and local [5] algorithms have recently been proved to be very effective, the most robust and widely-used method for the solution of MICPs is Branch and Bound (B&B) [6], [7]. Despite its worst-case performance, this is a very appealing algorithm: whenever a solution exists, B&B is guaranteed to find a globally-optimal one or, otherwise, provide a proof of infeasibility. Several improvements of this algorithm, tailored to the structure of an MPC problem, have been presented. Some of these have focused on accelerating the solution of the convex subproblem solved at each node of the B&B tree [8], [9], [10], others on designing search heuristics able to exploit the temporal structure of the problem [11], [10]. In [12] we showed that solution times for different MICP formulations of the same MPC problem can vary wildly when using B&B. Here we address the warm starting of solutions across multiple time steps of a receding-horizon MPC controller: the algorithm we propose can be applied in synergy with these related ideas.

MPC solves a sequence of optimal control problems which only differ in a shift of the time window, which in turn entails the deletion and the addition of a few variables and constraints. The principle behind warm start is to properly shift the solution of one problem to derive a good starting point for the subsequent solve. This idea has been extremely successful in linear MPC [13], [14], [15], but its application in the hybrid case is more involved and obstructed by the complexity of B&B algorithms. The problem of warm starting (or “reoptimizing”) a Mixed-Integer Linear Program (MILP) has been analyzed before by the operational research community [16]. In [17] the case of a sequence of MILPs with different constraint right-hand sides has been analyzed, and a similar approach has been proposed in [18], [19], where also variations in the cost vector have been discussed. However the MPC setting we consider here is drastically different: two consecutive optimization problems have different cost functions, constraint sets, and even optimization variables.

The warm-start technique we present consists in two main parts. First we propose a simple procedure to process the leaves of a B&B tree, and generate a set of root nodes for the following MPC problem. Then we show how duality theory can be used to associate to each one of these nodes a lower bound for the related convex minimization problem, even in the presence of disturbances. Starting from this refined partition of the search space, the B&B algorithm generally only requires the solution of few subproblems to detect a feasible solution. The synthesized lower bounds are then
used to immediately prune most of the branches of the tree, accelerating the convergence of the B&B, without sacrificing global optimality. By means of a thorough statistical analysis, we show that on average our algorithm strongly outperforms the standard approach of solving each problem from scratch.

**Notation**

We denote with $\mathbb{R}$ the field of real numbers and with, e.g., $\mathbb{R}_{\geq 0}$ nonnegative reals. The same notation is used for integer $\mathbb{Z}$ and natural numbers $\mathbb{N} := \mathbb{Z}_{\geq 0}$. Writing $x \in \mathbb{R}^n$ we tacitly assume $n \in \mathbb{N}$. For two vectors $x, y \in \mathbb{R}^n$, $(x, y) \in \mathbb{R}^{n \times n}$ represents their concatenation. For $x \in \mathbb{R}^n$ and $A \in \mathbb{R}^{n \times n}$, we let $\|x\|_A := x^T A x$. All physical units may be assumed to be expressed in the MKS system.

**II. HYBRID MODEL PREDICTIVE CONTROL**

In this paper we assume the hybrid system under analysis to be described within the framework of Mixed Logical Dynamical (MLD) systems [2]. MLD systems are in fact the intermediate representation in which most of the hybrid systems are cast for numerical optimal control [3], [7], and this choice makes our analysis independent from this recasting process (which itself can have a drastic impact on the efficiency of the optimization problem we formulate [12]). A time-varying MLD system has the following structure

$$x_{\tau+1} = A_{\tau} x_{\tau} + B_{\tau} u_{\tau} + b_{\tau},$$

$$F_{\tau} x_{\tau} + G_{\tau} u_{\tau} \leq g_{\tau},$$

where $x_{\tau} \in \mathbb{R}^{n_{\tau}} \times \{0, 1\}^{m_{\tau}}$ collects the continuous and binary states at discrete time $\tau \in \mathbb{Z}$, $u_{\tau} \in \mathbb{R}^{n_{\tau}} \times \{0, 1\}^{m_{\tau}}$ denotes the inputs, whereas the remaining matrices are real-valued and of appropriate size. Frequently, in the literature, a distinction between the independent and dependent (auxiliary) input variables $u_{\tau}$ is made [2], where the second are assumed to be uniquely determined by the first and the state $x_{\tau}$ through (1b). However the role of these variables is identical from the perspective of numerical optimization, so we do not distinguish between them here. In order to further streamline our analysis, we limit our attention to systems with continuous state ($m_{\tau} = 0$), even if this hypothesis could be dropped at the cost of a more intricate presentation.

At the generic time step $\tau$, given the current state $x_{\tau}$, we formulate the Mixed-Integer Quadratic Program (MIQP)

$$\min_{t=0}^{T} \sum_{t=0}^{T-1} \|z_t^T\|_{Q_t} + \sum_{t=0}^{T-1} \|w_t^T\|_{R_t}$$

subject to

$$x_0^T = x_{\tau},$$

$$x_{t+1}^T = A_{\tau+t} x_t^T + B_{\tau+t} u_t^T + b_{\tau+t}, \quad \forall t \in \mathbb{N} \leq T-1,$$

$$F_{\tau+t} x_t^T + G_{\tau+t} u_t^T \leq g_{\tau+t}, \quad \forall t \in \mathbb{N} \leq T-1,$$

$$z_t^T = C_{\tau+t} x_t^T + c_{\tau+t}, \quad \forall t \in \mathbb{N} \leq T,$$

$$w_t^T = D_{\tau+t} u_t^T + d_{\tau+t}, \quad \forall t \in \mathbb{N} \leq T-1,$$

$$V u_t^T \in \{0, 1\}^{m_{\tau}}, \quad \forall t \in \mathbb{N} \leq T-1,$$

where the optimization variables are $\{x_t^T, z_t^T\}_{t=0}^T$, $\{u_t^T, w_t^T\}_{t=0}^{T-1}$ and, e.g., $x_t^T$ represents the value of the state at time $\tau + t$ within the problem solved at time $\tau$. Additionally, $T \in \mathbb{Z}_{\geq 1}$ denotes the time horizon of the controller, $V \in \mathbb{R}^{m_{\tau} \times (n_{\tau} + m_{\tau})}$ selects the binary elements in $u_{\tau}$ (denoted as $v_{\tau} := V u_{\tau}$), $z_t^T$ and $w_t^T$ are output vectors selected by the real time-varying matrices in (2e) and (2f), finally, $Q_t^T$ and $R_t^T$ are positive-definite time-varying weight matrices. (Note that we do not assume, e.g., $Q_t^T = Q_t^{T-1}$.)

The outcome of (2) is an optimal open-loop control sequence $(\hat{u}_{\tau}^T)_{\tau=0}^{T-1}$ with the related state trajectory $(\hat{x}_{\tau}^T)_{\tau=0}^{T-1}$.

Assuming, without loss of generality, the current time to be $\tau = 0$, in MPC only the first action $u_0 := \hat{u}_{\tau=0}^0$ is applied to the system then, at time step $\tau = 1$, problem (2) is solved again in a moving-horizon fashion with the new measure of the state $x_1 = A_0 x_0 + B_0 u_0 + b_0 + e_0$, where $e_0$ represents the modeling error at time $\tau = 0$. Given the similarity of the problems we solve at time $\tau = 0$ and $\tau = 1$, it is natural to ask whether part of the computations performed at one time step can be exploited to speed up the solution of the next problem. In the next section we will introduce the notions necessary to formalize and discuss this problem.

**III. THE BRANCH AND BOUND ALGORITHM**

Here we concisely review the principles behind B&B considering its application to problem (2); a thorough description of this algorithm can be found, e.g., in [6]. Generally B&B is presented as tree search, where each node is associated with a convex subproblem of the MIQP; however, to analyze how this algorithm can be warm started, a more abstract interpretation of it is required. The following analysis shares multiple common points with the ones in [18] and [19].

Let $P^V$ be the subproblem we obtain replacing (2g) with the convex constraint $(\bar{u}_{\tau}^T)_{\tau=0}^{T-1} \in V$, and let $\theta^V \in \mathbb{R}_{\geq 0} \cup \{\infty\}$ be its optimal value, equal to infinity in case of infeasibility. (For simplicity, in this section, we do not express explicitly the dependence of the variables on the time step $\tau$.) At the generic iteration $i \in \mathbb{N}$ of the algorithm, we are given the following three inputs.

- A collection $\mathcal{T}^i$ of polyhedra in $\mathbb{R}^{T m_{\tau}}$ whose union covers $(\{0, 1\}^{T m_{\tau}}$. These sets can be thought as the leaves of the B&B tree at the current iteration; note also that we left open the possibility to have multiple root nodes, i.e., $|\mathcal{T}^0| \geq 1$.
- A lower bound $\theta^V \in \mathbb{R}_{\geq 0} \cup \{\infty\}$ on $\theta^V$, for all $V \in \mathcal{T}^i$. Except for the case of a root node, this value is equal to the objective of the parent node.
- An upper bound $\bar{\theta} \in \mathbb{R}_{\geq 0} \cup \{\infty\}$ on the optimal value of the MIQP (2), i.e., the cost of the best incumbent solution, if one is available.

Given an optimality tolerance $\varepsilon \in \mathbb{R}_{\geq 0}$, a B&B iteration consists in the selection of a leaf $V^i \in \mathcal{T}^i$ such that

$$\theta^V < \bar{\theta} - \varepsilon,$$

and in the solution of the Quadratic Program (QP) $P^{V^i}$. We then check the following conditions, in the given order.

- **Solution update.** If the solution of $P^{V^i}$ is integer feasible, i.e. verifies (2g), then $\theta^{V^i}$ is an upper bound
for the optimal value of (2). Hence we set $\bar{\theta}^{t+1} \leftarrow \min\{\theta^V, \bar{\theta}^t\}$, $\bar{\theta}^V \leftarrow \theta^V$, and $\gamma^{t+1} \leftarrow \gamma^t$.

- **Pruning.** If $\theta^V \geq \bar{\theta} - \varepsilon$, any integer-feasible solution in $\mathcal{V}^t$ cannot be $\varepsilon$-better than the one we already have.
  Hence we set $\theta^V \leftarrow \theta^V$, $\gamma^{t+1} \leftarrow \gamma^t$, and $\bar{\theta}^{t+1} \leftarrow \bar{\theta}$.

- **Branching.** If none of the previous, we select an element of $\mathcal{V}^t$, for some $t$, whose optimal value is not integer, and we split $\mathcal{V}^t$ into two subsets $\mathcal{V}_0^t$ and $\mathcal{V}_1^t$: one in which this variable is forced to be zero, another in which it must equal one. We then let $\mathcal{V}^{t+1} \leftarrow \{\mathcal{V}_0^t, \mathcal{V}_1^t\} \cup \mathcal{V}^t$, $\theta^V_0 \leftarrow \theta^V$, $\bar{\theta}^V_1 \leftarrow \theta^V$, and $\gamma^{t+1} \leftarrow \gamma^t$.

The algorithm terminates if, at iteration $j$, condition (3) is not met for any $\mathcal{V} \in \mathcal{V}^j$, and returns the leaves $\mathcal{V}^j := \mathcal{V}^j$ and the optimal value $\bar{\theta} := \bar{\theta}^j$. If at termination we have $\bar{\theta} = \infty$, then $\bar{\theta}^V = \infty$ for all $\mathcal{V} \in \mathcal{V}^j$, and we can conclude that (2) is infeasible. Otherwise the solution which led to the best upper bound $\bar{\theta}$ is integer feasible, and the global minimum lies within the interval $[\bar{\theta} - \varepsilon, \bar{\theta}]$. Clearly, B&B is a finite algorithm since, in the worst case, it amounts to the enumeration of all the $2^{|T^{\mathcal{V}}|}$ potential binary assignments.

Central to this work is the choice of the algorithm inputs $\mathcal{V}^0$, $\bar{\theta}^V$, and $\bar{\theta}^0$. Clearly, in case no information about the solution is available, the initialization $\mathcal{V}^0 := \{\mathcal{V}\}$, with $\mathcal{V} := [0,1]^{T^{\mathcal{V}}}$ unit hypercube, $\bar{\theta}^0 := \infty$, and $\bar{\theta}^V := 0$ is always valid. On the other hand, for some problems, we might have a good guess for the optimal solution and be able to exploit it to warm start the B&B search. As we will see in Section V, problem (2) belongs to the latter category.

In the next section we review the basics of duality theory for quadratic programming; this is a crucial building block of most efficient MIQP solvers and will pave the way for the developments presented in this paper.

IV. DUALITY IN BRANCH AND BOUND

Duality is a tool of great use in B&B: for a generic (not necessarily convex) optimization problem, in fact, any feasible solution to the dual problem yields a lower bound on the optimal value of the primal problem and, as we will see shortly, this matches very well the B&B logic.

In the B&B algorithm presented in Section III we assumed the covers $\mathcal{V}^t$ to be composed by polyhedra. However, for practical reasons, it is more convenient to limit our attention to the class of intervals in which some of the binaries are fixed to a specific value, while the others are allowed to range between 0 and 1. Such a set can be represented as

$\{v^T \bar{\tau} \mid 0 \leq v^T \leq \bar{v}^T\}, \quad \text{(4)}$

for some $\bar{v}^T, \bar{v}^T \in [0,1]^{T^{\mathcal{V}}}$ such that $v^T \leq \bar{v}^T$. Another safe hypothesis, which streamlines the analysis, is assuming the collection $\mathcal{V}^t$ to be composed of pairwise-disjoint sets. Note that the simple initialization with the hypercube, together with the branching rule described above, is coherent with these choices and preserves this structure across iterations.

Let $\mathbf{P}_\tau$ be the QP obtained replacing (2g) with the condition that $v^T_0 \bar{\tau} \geq 0$ must lie in the interval defined in (4).

(For simplicity of notation, we drop the explicit dependence of $\mathbf{P}_\tau$ on the set (4).) To derive the dual of this QP, we associate to each constraint a Lagrange multiplier: $\alpha^t_0$ with (2b), $\alpha^t_+1$ with (2c), $\delta^t_0$ with (2d), $\gamma^t_-$ with (2e), and $\delta^t_+$ with (2f). Finally, we let $\pi^t_0$ and $\pi^t_+$ be the multipliers for the constraints $v^T_0 \leq v^T$ and $v^T \leq \bar{v}^T$, respectively. The dual problem $\mathbf{D}_\tau$ of $\mathbf{P}_\tau$, presented in (5), is a QP and involves the maximization of a concave function over a convex set.

By weak duality the cost of any feasible solution of $\mathbf{D}_\tau$ is a lower bound to the optimal value $\theta_\tau$ of $\mathbf{P}_\tau$, and by strong duality the optimal value of $\mathbf{D}_\tau$ is also $\theta_\tau$.

Analyzing the structure of $\mathbf{D}_\tau$ we notice immediately an interesting feature that distinguishes the dual problem from the primal: thanks to the particular structure of (4), the bounds $(\bar{v}^T, \bar{v}^T)$ are now cost coefficients and do not affect the constraints. Therefore, when splitting a QP into two subproblems during the branching phase, it is possible to reuse very efficiently the solution of the parent problem as a feasible starting point for the children. This is also the reason why active set methods (which take great advantage of good starting points) are generally the best choice for the solution of the subproblems [6], [8], [9].

The second affinity between dual problems and B&B is even more evident: Solving $\mathbf{D}_\tau$ involves a sequence of feasible solutions whose cost increases monotonically towards $\theta_\tau$, and the availability in B&B of an cutoff value $\bar{\theta}$ above which a subproblem can be safely pruned, enables the early termination of a significant amount of QP solves.

Another distinctive feature of the dual $\mathbf{D}_\tau$ is feasibility: the choice of setting all the dual variables to zero always complies with the constraints in $\mathbf{D}_\tau$, regardless the input data. This implies that the consequence of weak duality, for which unboundedness of $\mathbf{D}_\tau$ implies infeasibility of $\mathbf{P}_\tau$, also holds in the opposite direction. Finally, we underline the role of $\tau$ in (5): the current state enters in the dual problem as a cost coefficient and, as such, it does not affect dual feasibility. As we will see, this simple observation will allow to extend our warm-start algorithm, without any additional work, to the case in which the system model (1) is incorrect.

V. CONSTRUCTION OF THE WARM START

In Section III we have seen that a warm start for problem (2) should include: an initial cover $\mathcal{V}^0$, a set of lower bounds $\theta^V$ for all $\mathcal{V} \in \mathcal{V}^0$, and an upper bound $\bar{\theta}^0$ on the MIQP objective. Here we leverage on the structure of problem (2) to efficiently construct most of these elements.

A. Construction of an Initial Cover

Consider again $\tau = 0$ to be the current time, we now describe how to assemble an initial cover $\mathcal{V}^0_\tau$ for problem (2) at time $\tau = 1$. Assuming (2) to be feasible for $\tau = 0$, we denote with $\mathcal{V}_0$ the cover of $\{0,1\}^{T^{\mathcal{V}}}$ we obtain form its solution. In light of the discussion from Section IV, we assume the latter to contain pairwise-disjoint intervals of the form (4). Starting from $\mathcal{V}_0$, we construct $\mathcal{V}^0_\tau$ in two steps.

- **We discard from $\mathcal{V}_0$ all the sets for which the condition $v^{\mathcal{V}}_0 \leq v_0 := V u_0 \leq \bar{v}^{\mathcal{V}}_0$ does not hold. This excludes the intervals which do not comply with the action taken on step $\tau = 0$, and which are therefore irrelevant at $\tau = 1$.**
For all the retained sets, we add to $\mathcal{Y}_1^0$ the set 
\[ [(v_0^0, \ldots, v_{T-1}^0), (\bar{v}_0^0, \ldots, \bar{v}_{T-1}^0, 1)]. \]
We now verify that the resulting collection of sets is consistent with our assumptions.

Proposition 1: $\mathcal{Y}_1^0$ covers \{0, 1\}$^{T_m}$ and is composed by pairwise-disjoint intervals.

Proof: Let $(v_i^1)_{i=0}^{T-1}$ be a generic element of \{0, 1\}$^{T_m}$. Since $\mathcal{Y}_0$ covers \{0, 1\}$^{T_m}$, there must be a set in it which contains $(v_0, v_0^1, \ldots, v_{T-2}^1)$. This implies, by construction, the existence of a set in $\mathcal{Y}_1^0$ which contains $(v_i^1)_{i=0}^{T-1}$. Hence $\mathcal{Y}_1^0$ covers \{0, 1\}$^{T_m}$. Now consider $(v_i^1)_{i=0}^{T-1} \in \mathbb{R}^{T_m}$, and assume the existence of two sets in $\mathcal{Y}_1^0$ which contain this point. Then there must also be two sets in $\mathcal{Y}_0$ which contain $(v_0, v_0^1, \ldots, v_{T-2}^1)$, which is a contradiction, and we conclude that the sets in $\mathcal{Y}_1^0$ are pairwise disjoint.

To better understand the logic behind the construction above note that, for an exact model (2c), as the controller horizon $T$ grows, the solution of two consecutive problems tend to “line up”, i.e., $\lim_{T \to \infty} \bar{v}_t^1 = v_{t+1}^0$. Hence, if $\mathcal{Y}_0 \in \mathcal{Y}_1^0$ is the set which contains the optimal binary assignment at time $t = 0$, and if $\mathcal{Y}_1 \in \mathcal{Y}_1^0$ is its descendant through this procedure, we also have $\lim_{T \to \infty} (\bar{v}_t^1)_{t=0}^{T-1} \in \mathcal{Y}_1$.

B. Equipping the Initial Cover with Lower Bounds

We now face the more intricate issue of equipping the sets in the family $\mathcal{Y}_1^0$ with a lower bound for the associated minimization problems. To address this point we will take advantage of duality theory (see Section IV) and we will show how these lower bounds can be generated by constructing a dual-feasible solution in element $\mathcal{Y}_1^0$.

Let, for $t = 0$, $\mathcal{D}_0$ be defined as in Section IV and $\mathcal{D}_1$ be its dual problem (5). Following the construction presented in Subsection V-A, the QP associated with a generic element of $\mathcal{Y}_1^0$ can be obtained from (2), replacing (2g) with $u_t^0 \leq v_{t+1}^0 \leq v_t^1$, for all $t \in \mathbb{N}_{\leq T-2}$, and $0 \leq v_{T-1}^1 \leq 1$. Defining
\[
(y_t^1, \bar{v}_t^1) := \begin{cases} (v_0^0, v_{T-0}^0) & \text{if } t \in \mathbb{N}_{\leq T-2}, \\
(0, 1) & \text{if } t = T - 1, \end{cases}
\]
we get a problem $\mathcal{P}_1$ with the form analyzed in Section IV, whose dual problem $\mathcal{D}_1$ is given by (5) for $t = 1$.

The following lemma relates the duals $\mathcal{D}_0$ and $\mathcal{D}_1$, and sets the basis for the developments we present in this paper.

Lemma 1: If $(\alpha_t^0, \gamma_t^0)_{t=0}^{T}$ and $(\beta_t^0, \delta_t^0, \pi_t^0, \bar{\pi}_t^0)_{t=0}^{T-1}$ is a feasible solution for $\mathcal{D}_0$, then the following is a set of feasible multipliers for $\mathcal{D}_1$:
\[
\begin{align*}
\{ (\alpha_t^1, \gamma_t^1)_{t=0}^{T} \} & := (\alpha_t^0)_{t=0}^{T}, \quad \forall t \in \mathbb{N}_{\leq T-1}, \quad (\alpha_t^1)_{t=0}^{T} := 0; \\
\{ (\beta_t^1, \delta_t^1, \pi_t^1, \bar{\pi}_t^1)_{t=0}^{T-1} \} & := (\beta_t^0, \delta_t^0, \pi_t^0, \bar{\pi}_t^0)_{t=0}^{T-1}, \quad \forall t \in \mathbb{N}_{\leq T-2}, \\
\{ (\beta_t^0, \delta_t^0, \pi_t^0, \bar{\pi}_t^0)_{t=0}^{T-1} \} & := 0.
\end{align*}
\]

Proof: This statement can be easily verified substituting the candidate solution in the constraints of $\mathcal{D}_1$, while taking into account feasibility of the given multipliers for $\mathcal{D}_0$.

Given a set in $\mathcal{Y}_0$ and a feasible dual solution for the associated QP, we can now equip with feasible multipliers, and hence a lower bound, the related set in $\mathcal{Y}_1^0$. Since we just assumed feasibility of the given multipliers, Lemma 1 applies even if, as it is frequently the case, $\mathcal{D}_0$ is not solved to optimality. Similarly, if the bound we generate is tight enough to prevent the solution of $\mathcal{D}_1$ within the B&B at time $t = 1$, the synthesized dual variables can be used themselves to generate a feasible solution for $t = 2$. On the other hand, if solving $\mathcal{D}_1$ is necessary, we can still use the multipliers from Lemma 1 to warm start this QP solve. Finally, we remark that this result holds despite any potential mismatch $e_0$ between our model and the real system.

The following theorem concerns the tightness of the lower bounds we construct through Lemma 1.

Theorem 1: Let $(\alpha_t^0, \gamma_t^0)_{t=0}^{T}$ and $(\beta_t^0, \delta_t^0, \pi_t^0, \bar{\pi}_t^0)_{t=0}^{T-1}$ be a feasible solution for $\mathcal{D}_0$ with cost $\tilde{h}_0$, and define
\[
\begin{align*}
\lambda_1 & := -C_0 x_0 + c_0 \|Q_0\|_2^2 - \|D_0 u_0 + d_0\|_2^2, \\
\lambda_2 & := \sum_{t=1}^{T-1} \left( \|\gamma_t^0/2\|_{Q_0^2}^2 - \|D_0 u_0 + d_0\|_2^2 \right) + \|\delta_t^0\|_{Q_0}^2 - (v_0^0 - V u_0)^\top \pi_0 + (v_0^0 - V u_0)^\top \bar{\pi}_0, \\
\lambda_3 & := -\left( F_0 x_0 + C_0 x_0 - g_0 \right) \top \rho_0 + + \|Q_0^{-1} \gamma_0^0/2 + C_0 x_0 + c_0 \|_{Q_0}^2 + \|D_0 u_0 + d_0\|_2^2, \\
\lambda_5 & := -e_0 \alpha_1^0.
\end{align*}
\]
The following is a lower bound of the optimal value of $P_1$

$$\theta_1 := \theta_0 + \sum_{i=1}^{5} \lambda_i. \quad (7)$$

**Proof:** See the Appendix.

Despite the many terms, Theorem 1 is very informative, and a quick inspection of the expressions in (6) reveals the following.

- **$\lambda_1$:** This negative term represents the MIQP stage cost for $\tau = 0$ and, noticeably, it does not depend on the solution of $D_0$ we employ to calculate the lower bound.

- **$\lambda_2$:** This term is the cost due to the mismatch between the weight matrices $Q^1_t, Q^0_{t+1}$ and $R^1_t, R^0_{t+1}$. Note that a sufficient condition for it to be nonnegative is $Q^1_{t-1} \succeq Q^0_t$ and $R^1_{t-1} \succeq R^0_t$. This holds, for example, in the common case of constant weight matrices, in which case we have $\lambda_2 = 0$.

- **$\lambda_3$:** Because of the feasibility of $u_0$, the condition $v^0_0 \leq \bar{v}^0_0$ imposed in the construction of $\gamma^0_1$, and the nonnegativity of $(\beta^0_t, \bar{s}^0_t, \bar{a}^0_t), \lambda_3$ is always nonnegative.

- **$\lambda_4$:** This term is nonnegative and can be proved to vanish at a stationary point of the Lagrangian of $P_0$, hence it measures the suboptimality of $\gamma^0_0$ and $\bar{e}_0^0$.

- **$\lambda_5$:** This term takes into account the modeling error $e_0$. It is null in case of a perfect model, whereas it can assume both signs in case of disturbances.

As a byproduct of this analysis we have an insightful result: for a perfect model and, e.g., constant weight matrices, the MIQP stage cost for $\tau = 0$ (here $\lambda_1$), a value which is completely independent from the particular pair of subproblems we are working with.

As we will see shortly, Theorem 1 allows also to draw conclusions on the feasibility of $P_1$. To this end, we restate Farkas’ lemma in a form suitable for the QPs under analysis. **Lemma 2:** Either $P_\tau$ is feasible, or there exist feasible multipliers $\{\alpha^0_t, \gamma^0_t = 0\}_{T=0}^T$ and $\{\beta^0_t, \delta^0_t = 0, \bar{s}^0_t, \bar{r}^0_t\}_{T=0}^{T-1}$ whose dual cost $\theta_0$ is strictly positive.

**Proof:** To prove this particular version of Farkas’ lemma, we need to consider the linear program we get setting the objective of $P_\tau$ to zero. The thesis follows from standard strong duality arguments.

We call the set of dual variables defined in Lemma 2 a certificate of infeasibility for $P_\tau$.

Solving $D_0$ with an active set method, unboundedness is detected when, at the current feasible solution, we find a direction in which we can move arbitrarily while increasing the objective and remaining feasible [20]. Since the feasible set of (5) is a cone, it is immediate to verify that such a direction will also be a certificate of infeasibility for $P_0$.

It is hence natural to wonder if the procedure presented in Lemma 1 can, in this case, be used to draw conclusions about the feasibility of $P_1$.

**Theorem 2:** Assume we are given a set of dual variables $\{\alpha^0_t, \gamma^0_t\}_{T=0}^T$ and $\{\beta^0_t, \delta^0_t, \bar{s}^0_t, \bar{r}^0_t\}_{T=0}^{T-1}$ with cost $\theta_0$ that certifies infeasibility of $P_0$. Then, the set of dual variables defined in Lemma 1 is a certificate of infeasibility for $P_1$ as long as $e_0$ lies in the open halfspace

$$E_0 := \{e_0 \mid (\alpha^0_0)^T e_0 < \theta_0 + \lambda_3\}.$$  

Moreover, $E_0$ contains the origin $e_0 = 0$.

**Proof:** We check the conditions from Lemma 2. In Lemma 1 we have shown dual feasibility of these multipliers and, by construction, we have $(\gamma^0_1)^T = 0$ and $(\delta^0_1)^T = 0$. We then derive the set of $e_0$ for which the cost of this dual solution is strictly positive using Theorem 1. In this case, we have $\lambda_2 = \lambda_1 + \lambda_3 = 0$ and the dual objective becomes $\theta_0 + \lambda_3 > 0$, which leads to the definition of $E_0$. Finally, since $\lambda_3 \geq 0$ and $\theta_0 > 0$, we conclude that $e_0 = 0 \in E_0$.

Theorem 2 completes the tools we need to construct and equip with lower bounds the initial cover $\gamma^0_0$. To each set in $\gamma_0$, whose related optimization problem is infeasible, we can now associate a halfspace $E_0$ in the error space inside which the descendant problem $P_1$ will also be infeasible ($\theta_1 = \infty$), with the further guarantee that in the nominal case $e_0 = 0$ this condition will always hold. Moreover, as in the case of Lemma 1, the process can be iterated, and the same certificate propagated across multiple time steps. Finally, if the condition $e_0 \in E_0$ is not verified, we simply set $\theta_1 = 0$.

**C. Upper Bounds and Recursive Feasibility**

Unfortunately, the structure of problem (2) is not suitable for the derivation of an upper bound on the optimal value of the MIQP at time $\tau = 1$, and we can only rely on the observation made in the end of Subsection V-A to guess where the optimal solution is likely to lie.

In a certain sense, we could say that our ability to reuse previous solutions to construct a feasible point for (2), is in contrast with the possibility of generating lower bounds for the subproblems. Classically, in MPC, “recursive feasibility” is in fact obtained through the use of a terminal constraint [7] which, however, would change the structure of the dual problem (5) and would make Lemma 1 untrue. Nevertheless, it must be said that the classical “primal” way to think about the MPC problem does not take model errors into account, making an upper bound for (2), even when available, useless in practice.

**VI. SUMMARY OF THE WARM-START ALGORITHM**

The overall warm-start algorithm can be summarized as follows. Assuming the availability of the solution of problem (2) at time $\tau = 0$, for $\tau = 1, 2, \ldots$, we:

- measure the system state $x_\tau$ and compute the error $e_\tau$;
- evaluate the lower bounds for each set in $\gamma^0_\tau$ for the measured error $e_\tau$, as in Subsection V-B;
- solve problem (2) warm starting the B&B algorithm, inject in the system the control action $u_\tau := \tilde{u}^0_\tau$;
- process the solution of (2), and generate the warm start for time $\tau + 1$.

As discussed in Subsection V-A, for long horizons $T$ and moderate errors $e_\tau$, open- and closed-loop trajectories tend to coincide. When this is the case, the initial cover $\gamma^0_1$ allows
to quickly identify the optimal solution and the lower bounds $\theta^V$ to readily prune the great majority of the leaves and prove optimality right away. In these cases, the number of QP solves is hence independent from the controller horizon, reducing the combinatorial complexity of our problem to the one of a one-step MPC problem. As we will see in the next section, through adequate B&B heuristics, this best-case analysis is often also representative of the average scenario.

VII. NUMERICAL STUDY

We conclude testing the proposed warm-start algorithm on a numerical example. We consider the cart-pole system from Figure 1: the goal is to regulate the cart in the center of the two walls with the pole upright. To perform this task we can apply a force $u_1$ directly on the cart and exploit contact forces $u_2$ and $u_5$ that arise when the tip of the pole collides with the walls. As we will see, the modeling of contact phenomena requires binary indicators (e.g., $u_3$ and $u_4$ for the left wall), making the overall system, after a suitable linearization, an MLD. This regulation problem, or a slight variation of it, is a customary benchmark in control through contacts [21], [22], and its moderate size allows an in-depth statistical analysis of the performances of our algorithm.

A. Mixed Logical Dynamical Model

We let $x_1$ be the position of the cart, $x_2$ the angle of the pole, and we denote with $x_3$ and $x_4$ their time derivatives. The continuous-time equations of motion, linearized around the nominal angle of the pole $x_2 = 0$, are

$$\dot{x}_1 = x_3,$$
$$\dot{x}_2 = x_4,$$
$$\dot{x}_3 = \frac{g m_p}{m_c} x_2 + \frac{1}{m_c} u_1,$$
$$\dot{x}_4 = \frac{g (m_c + m_p)}{l m_c} x_2 + \frac{1}{l m_c} u_1 - \frac{1}{l m_p} u_2 + \frac{1}{l m_p} u_5,$$

where $m_c = m_p = 1$ are the mass of the cart and the pole, respectively. $g = 10$ is the gravity acceleration, $l = 1$ is the length of the pole, and $u_2$, $u_5$ are the contact forces with the left and the right wall, respectively. The dynamics are discretized using the explicit Euler method with a time step $h = 0.05$. The force applied to the cart is limited by the constraints $u_1 \leq u_1 \leq \bar{u}_1$, with $\bar{u}_1 = -u_1 = 2$, and must keep the state inside the box $x \leq x \leq \hat{x}$ at all time steps. Here $\hat{x} = 0$ is the position $x = (d, \pi/8, 2, 1)$ and $d = 0.5$ is half of the distance between the walls (see Figure 1).

Impacts of the pole with the walls are modeled with soft contacts: $\kappa = 100$ is the stiffness and $\nu = 30$ is the damping in the contact model. The relative position of the tip of the pole with respect to the walls (positive in case of penetration), after linearization, is $\delta_2 := -x_1 + l x_2 - d$ for the left wall, and $\delta_5 := x_1 - l x_2 - d$ for the right wall. For $i \in \{2, 5\}$, we then define the contact forces

$$u_i := \begin{cases} \kappa \delta_i + \nu \dot{\delta}_i & \text{if } \delta_i \geq 0 \text{ and } \kappa \delta_i + \nu \dot{\delta}_i \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

These are piecewise-linear functions of the penetration and the relative velocity, they are always nonnegative, and can be nonzero only in case of nonnegative penetration. We then introduce two binary variables per contact

$$u_{i+1} := \begin{cases} 1 & \text{if } \delta_i \geq 0, \\ 0 & \text{otherwise.} \end{cases}, \quad u_{i+2} := \begin{cases} 1 & \text{if } \kappa \delta_i + \nu \dot{\delta}_i \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

By means of the state limits, we can derive explicit bounds $(\bar{\delta}_i, \bar{\delta}_i)$ on the penetration, as well as on its time derivative $(\bar{\delta}_i, \bar{\delta}_i)$, which in turn are used to bound the contact force with $u_i := \kappa \delta_i + \nu \dot{\delta}_i$ and $u_i := \kappa \delta_i + \nu \dot{\delta}_i$. Conditions (9) can be then enforced through the linear inequalities

$$\bar{\delta}_i (1 - u_{i+1}) \leq \delta_i \leq \kappa \delta_i + \nu \dot{\delta}_i u_{i+1},$$
$$u_i (1 - u_{i+2}) \leq \delta_i \leq \kappa \delta_i + \nu \dot{\delta}_i u_{i+2}.$$

With a similar logic, we can express (8) through the conditions:

$$0 \leq u_i \leq \bar{u}_i u_{i+1}, \quad u_i \leq \bar{u}_i u_{i+2}, \quad \nu \dot{\delta}_i (u_{i+1} - 1) \leq u_i - \bar{\delta}_i - \nu \dot{\delta}_i \leq u_i (u_{i+2} - 1).$$

Considering the binary inputs introduced as contact indicators, we have an MLD system with $n_x = 4$ continuous states, $n_u = 3$ continuous inputs, and $m_u = 4$ binary inputs.

B. Branch and Bound Heuristics and Parameters

In the numeric results presented below we consider an optimality tolerance $\varepsilon = 0$ in the solution of problem (2) via B&B. We adopt a best-first search, i.e., among all the sets which verify condition (3), we pick the $V^l \in V^+$ for which $\theta^V$ is minimum. Finally, we perform the branching step in chronological order; each time this subroutine is called, we select the relaxed variables $v^l_i$ for which $t$ is lowest and, among these, we split the one with the smallest index. This frequently-used heuristic, leveraging on the control limits, quickly rules out excessively fast mode transitions [11], [10].

C. Statistical Analysis

To analyze the efficiency of the proposed warm-start algorithm, we consider a “push-recovery” task in presence of modeling errors of increasing magnitude. The initial state of the system is set to $(0, 0, 1, 0)$, simulating the effects of a push towards the right wall. We synthesize an MPC controller with horizon $T = 30$, where we penalize the state
Fig. 2. Optimal open- and closed-loop trajectories for the cart-pole system recovering from a push towards the right wall. Top: input force applied to the cart. Bottom: horizontal position of the tip of the pole. The penetration of the pole in the right wall is allowed by the soft contact model.

$x$ and the force applied to the cart $u_1$ through the constant weight matrices $Q^T = 10I$ and $R^T = 1$.

Assuming a perfect model, Figure 2 depicts the control action and the related trajectory of the tip of the pole for a closed-loop simulation of length 50 steps. As a comparison, we also represent the open-loop trajectory obtained solving problem (2) at time $\tau = 0$. The system exploits the presence of the (soft) right wall to decelerate and come back to the center of the track, whereas the control action is sufficiently distant from the limits and has room to react to potential disturbances. The finite-horizon in (2) is responsible for the difference between open- and closed-loop trajectories.

We now consider the same task in presence of a random mismatch between the model (1) and the real system. At each sampling time $\tau \in \mathbb{Z}_{\geq 1}$ we draw the $i$th component of the error $e_\tau := x_{\tau + 1} - A_\tau x_\tau - B_\tau u_\tau - b_\tau$ from the normal distribution with zero mean and standard deviation $\sigma_i = c \bar{x}_i$, where $\bar{x}_i$ is the upper bound of the $i$th state. For $c \in \{10^{-3}, 3 \cdot 10^{-3}, 10^{-2}\}$, we simulate 100 closed-loop trajectories (for which the error in the dynamics does not drive the system to an infeasible state), and we inspect:

- the number of QPs solved within the B&B algorithm in case of warm and cold start,
- the number of sets in the warm-start cover $\mathcal{Y}_\tau^0$.

The first quantifies the computational savings entailed by the proposed algorithm with respect to the customary approach of solving each problem from scratch. (To give an order of magnitude, specialized solvers, e.g., [9], can solve relatively large QPs in few milliseconds.) The second represents the amount of information our algorithm requires to transfer from one time step to the next. In Figure 3 we report, for each of these three quantities, the median, minimum, and maximum values registered in the 100 trials, together with the results obtained in the nominal case ($e_\tau = 0$ for all $\tau$).

Despite of relatively large plant-model mismatches, the proposed approach requires on average an order of magnitude less QPs to solve problem (2) to global optimality. (Note that $c = 10^{-2}$ leads often to errors, e.g., of centimeters in the position of the cart.) Remarkably, in case of warm start, global optimality is most of the times proved just by solving 10 or 20 QPs, reaching very frequently the lower bound of $2m_u + 1 = 9$ QPs. In these cases the computational burden is hence comparable to that of a one-step MPC problem, validating the observations from Section VI. It is worth to underline that, also in case of cold start, performances often coincide with the best case scenario of $2T m_u + 1 = 241$ QPs, showing the extreme effectiveness of the heuristics described in Subsection VII-B. On the other hand, in the worst case, as the error grows, our method tends to perform as the standard one. Finally, we notice that also the amount of subproblems we need to process for the construction of the warm starts, measured as the cardinality of $\mathcal{Y}_\tau^0$, is very stable both in time $\tau$ and as a function of the error standard deviation $\sigma_i$. 

Fig. 3. Statistical analysis of the performances of the warm-start algorithm in the task of regulating the cart-pole system to the origin, in case of initial conditions $(0, 0, 1, 0)$. Orange (blue) lines: number of QPs needed to solve problem (2) with (without) warm start, as functions of time for different standard deviations of the random perturbations $e_\tau$. Gray lines: amount of information to be processed for the generation of the warm start, represented by the cardinality of the initial cover $\mathcal{Y}_0^0$, as a function of time and the perturbation size. Solid (dashed, dash-dot) lines: median (minimum, maximum) values of the above quantities over 100 feasible trial trajectories.
VIII. CONCLUSION

In this paper we have analyzed the problem of hybrid starting mixed-integer programs arising in receding-horizon optimal control of hybrid systems. We have shown how, through a proper problem formulation, the computations performed at each sampling time can be efficiently reused to accelerate the solution of subsequent problems, even in the presence of modeling errors. We have presented a novel warm-start algorithm and we have shown how it outperforms the common approach of solving each optimization problem from scratch, often reducing the combinatorial complexity of these problems to that of a one-step MPC problem.

APPENDIX

PROOF OF THEOREM 1

Given a feasible solution for $D_0$ we define a set of feasible multipliers for $D_1$ as in Lemma 1. Substituting these in the objective (5a), for $t = 1$, we get the lower bound

$$
\theta_1 := - \sum_{t = 1}^{T} \| \gamma_0 / 2 \|_{Q(t-1)}^{-1} - \sum_{t = 1}^{T-1} \| \delta_t / 2 \|_{A(t-1)}^{-1} + \sum_{t = 1}^{T-1} \left[ \alpha_t \delta_t + g_t \delta_t^0 + c_t \gamma_t + d_t \delta_t^0 \right] + - x_1 \alpha_1 + \sum_{t = 1}^{T-1} \left[ \left( \alpha_t \right)^T \pi^0_t - \left( \delta_t^0 \right)^T \pi^0_t \right].
$$

Taking into account (5a) for $t = 0$, the cost of the candidate solution can be restated as $\theta_1 = \theta_0 + \lambda_2 + \sum_{i = 1}^{3} \omega_i$, where $\lambda_2$ is defined in (6b) and

$$
\omega_1 := \| \gamma_0 / 2 \|_{Q_1^{-1}} + 1 - \| \delta_0 / 2 \|_{R_0^{-1}} + 1,
$$

$$
\omega_2 := (b_0 - x_1)^T \alpha_1 + x_0 \alpha_0,
$$

$$
\omega_3 := (b_0 - x_0)^T \beta_0 + (c_0 - \gamma_0)^T \delta_0 + (d_0 - \delta_0)^T \delta_0 - (\gamma_0)^T \pi^0_0 + (\delta_0)^T \pi^0_0.
$$

Substituting the dynamics $x_1 = A_0 x_0 + B_0 u_0 + b_0 + c_0$ we get $\omega_2 = -(A_0 x_0 + B_0 u_0 + c_0)^T \alpha_1 + x_0 \alpha_0$. Using (5c) and (5d) for $t = 1$, we have

$$
\omega_2 = - x_1^T \left( F_0 \alpha_0 + C_0^T \gamma_0 + u_0 \left( G_0 \alpha_0 + D_0 \delta_0 + V \left( \gamma_0 - \delta_0 \right) \right) \right) + \lambda_5,
$$

where $\lambda_5$ is defined in (6e). Adding $\omega_3$, we obtain

$$
\omega_2 + \omega_3 = (C_0 x_0 + c_0)^T \gamma_0 + (D_0 u_0 + d_0)^T \delta_0 + \lambda_5,
$$

where $\lambda_3$ is defined in (6c). Finally, adding $\omega_1$

$$
\omega_1 + \omega_2 + \omega_3 = \| \gamma_0 / 2 \|_{Q_1^{-1}} + (C_0 x_0 + c_0)^T \gamma_0 + + \| \delta_0 / 2 \|_{R_0^{-1}} + (D_0 u_0 + d_0)^T \delta_0 + + \lambda_3 + \lambda_5.
$$

The first four terms in the right-hand side of the latter equation can be rearranged to get

$$
\| \gamma_0 / 2 \|_{Q_1^{-1}} (C_0 x_0 + c_0) + \| \delta_0 / 2 \|_{R_0} - \| C_0 x_0 + c_0 \|_{Q_0}^2 + + \| \delta_0 / 2 \|_{R_0} (D_0 u_0 + d_0) \|_{R_0}^2 - - \| D_0 u_0 + d_0 \|_{R_0}^2.
$$

Remembering the definitions of $\lambda_1$ in (6a) and $\lambda_2$ in (6d), we have $\sum_{i = 1}^{3} \omega_i = \lambda_1 + \lambda_2 + \lambda_4 + \lambda_5$, and hence the thesis (7).