A Model Predictive Control Framework for Hybrid Dynamical Systems [⋆]

Berk Altın [*] **Pegah Ojaghi** [*] **Ricardo G. Sanfelice** [*]

[*] *Department of Computer Engineering, University of California, Santa Cruz, CA 95064, USA (e-mail: berkaltin, pojaghi, ricardo@ucsc.edu)*

Abstract: This paper presents a model predictive control (MPC) algorithm that asymptotically stabilizes a compact set of interest for a given hybrid dynamical system. The considered class of systems are described by a general model, which identifies the dynamics by the combination of constrained differential and difference equations. The model allows for trajectories that exhibit multiple jumps at the same time instant, or portray Zeno behavior. At every optimization time, the proposed algorithm minimizes a cost functional weighting the state and the input during both the continuous and discrete phases, and at the terminal time via a terminal cost, without discretizing the continuous dynamics. To account for the structure of time domains defining solution pairs, the minimization is performed in a manner akin to free end-time optimal control. When the terminal cost is a control Lyapunov function on the terminal constraint set, recursive feasibility and asymptotic stability of the proposed algorithm can be guaranteed. A sample-and-hold control system and a bouncing ball model are two examples reported to demonstrate the applicability and effectiveness of the proposed approach.

Keywords: hybrid systems, predictive control, optimal control, asymptotic stability, nonlinear control systems.

1. INTRODUCTION

The term "hybrid" has generally been used in the model predictive control (MPC) literature to refer to systems possessing both continuous-valued and discrete-valued states and inputs, as well as discontinuities in the control algorithm or right-hand side of the dynamic equations (Mayne, 2014; Camacho et al., 2010; Borrelli et al., 2017). Of note, works falling into the former category often partition the state such that continuous states evolve according to a difference equation derived from the discretization of the differential dynamics, while the discrete states correspond to logical variables (Mayne, 2014, Section 2.2.5). On the other hand, development of MPC strategies for systems with continuously evolving states which are subject to discrete transitions (*jumps*) at times has been fairly limited, with the most relevant publications being the impulsive and measure-driven frameworks in Sopasakis et al. (2015); Pereira et al. (2015). See Sanfelice (2019) for a recent survey.

This paper presents a stabilizing MPC strategy for hybrid dynamical systems (referred to as the hybrid MPC algorithm, or in short, *hybrid MPC*) characterized by the interaction of continuous and discrete dynamics. The modeling framework that we rely on (Section 2), identifies a hybrid system by a combination of constrained differential and difference equations (Goebel et al., 2012), thereby encapsulating numerous models of a hybrid nature, as highlighted in (Mayne, 2014, Section 2.2.5). These include sample-and-hold control (Magni and Scattolini, 2004), event/self-triggered control, and switching systems (Cassandras and Mookherjee, 2003), in addition to systems where continuous variables can undergo jumps, such as hybrid automata and the aforementioned impulsive systems—see Goebel et al. (2009, 2012) for further details. The primary objective of this work is to introduce the MPC community to the recent developments in predictive control of such systems. Rather than a full technical treatment and a thorough discussion of numerics, emphasis is placed on the illustration of the key ideas of the hybrid MPC algorithm through various examples.

As an example to a hybrid system in the constrained differential/difference equations framework, throughout the paper, we will consider the vertical motion of a ball bouncing on a horizontal flat surface, modeled as a point-mass with height x_1 and velocity x_2. When $x_1 \geq 0$, the state $x = (x_1, x_2)$ evolves according to the differential equation

$$\dot{x}_1 = x_2, \dot{x}_2 = -\gamma, \tag{1}$$

where $\gamma > 0$ is the gravitational constant. Impacts occur when the ball reaches the surface with nonpositive velocity; i.e., when $x_1 = 0$ and $x_2 \leq 0$. At this point, the state is reset according to the difference equation

$$x_1^+ = 0, x_2^+ = -\lambda x_2 + u, \tag{2}$$

where $\lambda \geq 0$ is the coefficient of restitution and u is a scalar control input affecting the post-impact velocity. The superscripts $+$ in (2) indicate that x_1 and x_2 change

[⋆] This research has been partially supported by the National Science Foundation under CAREER Grant no. ECS-1450484, Grant no. ECS-1710621, and Grant no. CNS-1544396, by the Air Force Office of Scientific Research under Grant no. FA9550-16-1-0015, by the Air Force Research Laboratory under Grant no. FA9453-16-1-0053, and by CITRIS and the Banatao Institute at the University of California.

instantaneously. Aside from the presence of constraints describing when the state x evolves according to (1) or (2), an interesting feature of this prototypical system is the existence of *Zeno* trajectories (Goebel et al., 2009).

We reiterate that unlike the majority of the existing work, the hybrid MPC algorithm *does not rely on the discretization of the differential equation* associated with the hybrid system. Aside from laying the theoretical foundations of a generalized hybrid MPC framework, one reason for this approach is due to the nontriviality of discretization for certain hybrid systems because of constraints, which can lead to nonperiodic jumps. This issue is especially problematic for systems in which infinitely many jumps can accumulate over a bounded time horizon, as in the bouncing ball model in (1) and (2). In the nonactuated case with inelastic collisions (i.e., when $u = 0$ and $\lambda < 1$), the time between consecutive impacts tends to zero as the number of jumps tends to infinity, since the speed $|x_2|$ of the ball decreases exponentially with each impact due to the equation $|x_2^+| = |-\lambda x_2| < |x_2|$ when x_2 is nonzero. For this reason, the hybrid MPC algorithm presented in Section 3 relies on the minimization of a cost functional weighting the state during both the continuous and discrete phases, and imposes constraints on the terminal state and time. Building from MPC results in the literature, in Section 4, it is discussed how recursive feasibility and asymptotic stability can be achieved if the terminal cost is a control Lyapunov function (CLF) on the terminal constraint set. The hybrid MPC algorithm is illustrated in Section 5 through the bouncing ball system, and concluding remarks are given in Section 6.

2. PRELIMINARIES

We denote by $\mathbb{R}$ the real numbers and $\mathbb{R}_{\geq 0}$ its nonnegative subset, and by $\mathbb{N}$ the set of nonnegative integers. The distance of a vector $x \in \mathbb{R}^n$ to a nonempty set $\mathcal{A} \subset \mathbb{R}^n$ is expressed as $|x|_{\mathcal{A}} := \inf_{a \in \mathcal{A}} |x - a|$, where $|.|$ indicates the 2 norm. A continuous function $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ belongs to class-$\mathcal{K}_\infty$ if it is strictly increasing and $\alpha(0) = 0$.

2.1 Hybrid Control Systems

We consider hybrid control systems of the form

$$\mathcal{H} \begin{cases} \dot{x} = f(x,u) & (x,u) \in C' \times U_C =: C \\ x^+ = g(x,u) & (x,u) \in D' \times U_D =: D, \end{cases} \tag{3}$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ denote the state and input of the system, respectively. The *flow set* C (respectively, the *jump set* D) is a closed constraint set where flows (respectively, jumps) are allowed. The continuous mapping $f : C \to \mathbb{R}^n$ is called the *flow map*. Similarly, the continuous mapping $g : D \to \mathbb{R}^n$ is called the *jump map*. We assume that the set U_C is compact. In general, the sets C' and D' do not need to be disjoint, as is the case with the bouncing ball model in Section 1.

Example 2.1. (Bouncing Ball). Consider the bouncing ball with actuated jumps evolving according to (1) and (2). The dynamics of the bouncing ball in Section 1 can be represented in the form of (3) by incorporating the constraints therein. The jump map of this representation is given by $g(x,u) = (0, -\lambda x_2 + u)$ on the jump set D, which is defined according to $D' = \{x \in \mathbb{R}^2 : x_1 = 0, x_2 \leq 0\}$ and $U_D = \mathbb{R}$. The flow map is given by $f(x,u) = (x_2, -\gamma)$ on the flow set. For the flow set C, the state constraints are defined according to $C' = \{x \in \mathbb{R}^2 : x_1 \geq 0\}$, while the input constraints can be defined by any compact $U_C \subset \mathbb{R}^m$, as f does not depend on u. Note that in this model, the sets C' and D' overlap. In particular, $C' \cap D' = D'$.

A solution pair (x, u) of $\mathcal{H}$ is given by the state trajectory $x : \operatorname{dom} x \to \mathbb{R}^n$ and input $u : \operatorname{dom} u \to \mathbb{R}^m$ representing the input. The functions x and u are parametrized by $(t, j) \in \mathbb{R}_{\geq 0} \times \mathbb{N}$, where t indicates the ordinary time elapsed during flows and j indicates the number of jumps that have occurred. Both x and u are defined on a common domain, in other words, $\operatorname{dom} x = \operatorname{dom} u =: \operatorname{dom}(x,u)$. The set $\operatorname{dom}(x,u) \subset \mathbb{R}_{\geq 0} \times \mathbb{N}$ has a *hybrid time domain* structure, in the sense that there exist a real nondecreasing sequence $\{t_j\}_{j=0}^J$ with $t_0 = 0$, and when J is finite, $t_{J+1} \in [t_J, \infty]$ such that $\operatorname{dom}(x,u) = \cup_{j=0}^J I_j \times \{j\}$. Here, $I_j = [t_j, t_{j+1}]$ for all $j < J$. If J is finite, I_{J+1} can take the form $[t_J, t_{J+1}]$ (when $t_{J+1} < \infty$), or possibly $[t_J, t_{J+1})$. Examples of hybrid time domains can be seen in Figure 1. For the domain indicated in dashed red, $t_0 = t_1 = t_2 = 0$ and $t_3 = t_4 = t_5 = 2$. On each interval I_j, x is absolutely continuous, and u is Lebesgue measurable and locally essentially bounded.

A solution pair (x, u) has to satisfy the initial condition constraint $(x(0,0), u(0,0)) \in C \cup D$, along with the dynamics of $\mathcal{H}$. Namely, for each j,

$$\dot{x}(t,j) = f(x(t,j), u(t,j))$$

for almost all $t \in (t_j, t_{j+1})$,

$$(x(t,j), u(t,j)) \in C \quad \forall t \in (t_j, t_{j+1}),$$

and

$$x(t_{j+1}, j+1) = g(x(t_{j+1}, j), u(t_{j+1}, j)),$$
$$(x(t_{j+1}, j), u(t_{j+1}, j)) \in D.$$

In other words, the ordinary times $\{t_j\}_{j=1}^J$ correspond to when jumps occur. Note that $u(t_{j+1}, j)$ affects only the jumps dynamics. Hence, even if u is continuous on $[t_j, t_{j+1})$, it might have a discontinuity at t_{j+1}.

In addition, it should be noted that two solution pairs to the same system may not have the same hybrid time domain. The following example illustrates this.

Example 2.2. (Sample-and-Hold Control). The control of a linear system via a sample-and-hold mechanism can be described by a hybrid system with state $x = (z, \ell, \tau)$ and dynamics

$$\begin{cases} \dot{x} = (Az + B\ell, 0, 1) & (x,u) \in \{x : \tau \in [0, \bar{\tau}]\} \times \{0\} \\ x^+ = (z, u, 0) & (x,u) \in \{x : \tau = \bar{\tau}\} \times \mathbb{R}, \end{cases}$$

During flows, while ℓ stays constant, the plant state z evolves according to $\dot{z} = Az + B\ell$. The state component ℓ represents a zero-order hold mechanism on the input u and gets updated when jumps occur. Jumps of this system correspond to sampling events and are determined strictly by the timer variable τ, which increases with rate 1 during flows and resets to 0 every $\bar{\tau}$ seconds. Hence, jump times of solution pairs should be such that $t_{j+1} - t_j = \bar{\tau}$ when $j \geq 1$ and $t_1 - t_0 = t_1 \leq \bar{\tau}$. More specifically, the domain of the solution pair with initial condition $(z(0,0), \ell(0,0), \tau(0,0))$ is such that $t_j = j\bar{\tau} - \tau(0,0)$ for $j \geq 1$, save for possibly the last t_j when the domain is bounded. Due to this dependency of the jump times on $\tau(0,0)$, solution pairs of the system do not necessarily have the same domain. Note also that by definition of solution pairs of (3), for any (x,u), we have $u(t,j) = 0$ if $t \neq t_j$, while $u(t_j,j)$ determines the updated values of the zero-order hold state for $j \geq 1$, save for possibly the last t_j.

For the system $\mathcal{H}$ in (3), we assume uniqueness of trajectories. For hybrid systems *without inputs*, uniqueness is guaranteed when the continuous dynamics generate unique trajectories, and flows are not possible on the intersection of the flow and jump sets (Goebel et al., 2012, Proposition 2.11). A slightly different result holds in the case of hybrid systems with inputs: uniqueness of trajectories of $\mathcal{H}$ is equivalent to the uniqueness of trajectories of the differential equation $\dot{x} = f(x,u)$ with the constraint $(x,u) \in C$. This is due the fact that given any two solution pairs (x_1,u) and (x_2,u) with the initial conditions $x_1(0,0) = x_2(0,0)$, the domain of u of determines when jumps occur since $\operatorname{dom} x_1 = \operatorname{dom} x_2 = \operatorname{dom} u$ by definition, and uniqueness of trajectories due to the discrete dynamics follows from g being single valued.

2.2 Hybrid Control Systems under Static State-Feedback

Given a pair of continuous functions $\kappa := (\kappa_C, \kappa_D)$, called a *feedback pair*, where $\kappa_C : C' \to U_C$ and $\kappa_D : D' \to U_D$, we will also consider the autonomous hybrid system

$$\mathcal{H}_\kappa \begin{cases} \dot{x} = f_\kappa(x) := f(x,\kappa_C(x)) & x \in C' \\ x^+ = g_\kappa(x) := g(x,\kappa_D(x)) & x \in D'. \end{cases} \tag{4}$$

Under the effect of κ, a solution pair (x,u) can be generated as demonstrated in the next example.

Example 2.3. (Bouncing Ball with Feedback). Given the model in Example 2.1, consider the feedback pair κ with $\kappa_C(x) = 0$ and $\kappa_D(x) = \lambda x_2 + \sqrt{2\gamma h}$, where $h \geq 0$. This leads to an autonomous hybrid system of the form (4) with $f_\kappa(x) = f(x)$ and $g_\kappa(x) = (0, \sqrt{2\gamma h})$, whose trajectories are periodic after the first jump. For instance, the initial condition $(0, \sqrt{2\gamma h})$ leads to the trajectory

$$x(t,j) = (t\sqrt{2\gamma h} - (t-t_j)^2\gamma/2, \sqrt{2\gamma h} - (t-t_j)\gamma)$$

with jump times $t_j = 2j\sqrt{2h/\gamma}$. The corresponding input is such that $u(t_{j+1},j) = (1-\lambda)\sqrt{2\gamma h}$, and $u(t,j) = 0$ if $t \neq t_j$.

3. HYBRID MODEL PREDICTIVE CONTROL

As in conventional continuous/discrete-time MPC, the hybrid MPC algorithm relies on the solution of a finite horizon hybrid optimal control problem at specific (hybrid) time instants, called *optimization times*. The optimal control problem involves a cost functional $\mathcal{J}$, defined as [1]

$$\mathcal{J}(x,u) := \left(\sum_{j=0}^{J} \int_{t_j}^{t_{j+1}} L_C(x(t,j),u(t,j))\, dt \right) + \left(\sum_{j=0}^{J-1} L_D(x(t_{j+1},j),u(t_{j+1},j)) \right) + V(x(T,J)) \tag{5}$$

[1] The second sum is to be interpreted as zero if $J = 0$.

for every solution pair (x,u) of $\mathcal{H}$ with compact hybrid time domain (i.e., $\operatorname{dom}(x,u)$ is a compact set), not necessarily maximally defined. Above, (T,J) is the *terminal (hybrid) time* of the solution pair $(x,u)$and $\{t_j\}_{j=0}^{J+1}$ is the sequence defining $\operatorname{dom}(x,u)$, with $t_{J+1} = T$. The *flow cost* L_C is a continuous function defined on the flow set C, the *jump cost* L_D is a continuous function defined on the jump set D, and the *terminal cost* V is a continuous function defined on the closed *terminal constraint set* $X \subset C' \cup D'$.

Example 3.1. Consider the sampled-data control system in Example 2.2 with state $x = (z,\ell,\tau)$. Since the timer variable is uncontrollable and of no interest for the purposes of stabilizing the plant state z, the costs can be chosen to be independent of τ. For the flow cost, a quadratic function of the form $L_C(x,u) = z^\top Qz + \ell^\top R\ell$ weighting the plant state z and zero-order hold state ℓ can be selected as in Magni and Scattolini (2004). Quadratic functions can similarly be chosen for L_D and V.

3.1 Prediction Horizon

In comparison to finite-horizon optimal control problems arising in continuous/discrete-time MPC, insisting on a fixed end-time optimal control problem is restrictive for hybrid dynamical systems. Indeed, for the case of the bouncing ball in Example 2.1, any feedback controller or open-loop input which steers the state x to the origin necessarily leads to trajectories with the time between jumps converging to zero, i.e., $\lim_{j\to\infty} t_{j+1} - t_j = 0$ (Goebel et al., 2012, Example 2.12). To address this issue, similar to free end-time optimal control, we allow the terminal times of feasible solution pairs to belong to a set $\mathcal{T} \subset \mathbb{R}_{\geq 0} \times \mathbb{N}$, called the *prediction horizon*. For simplicity, we assume

$$\mathcal{T} = \{(t,j) \in \mathbb{R}_{\geq 0} \times \mathbb{N} : \max\{t/\delta, j\} = \tau\} \tag{6}$$

for some $\tau \in \{1,2,\dots\}$ and $\delta > 0$, where δ resembles a sampling time parameter, so that the terminal time (T,J) of any feasible solution pair satisfies $\max\{T/\delta, J\} = \tau$. Figure 1 shows the hybrid time domains associated with two generic prediction scenarios for the case of $\tau = 4$ and $\delta = 1$.

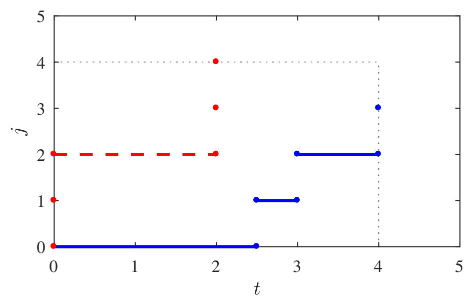


Fig. 1. Hybrid time domains of two solution pairs with terminal times in $\mathcal{T}$ for the case of $\tau = 4$ and $\delta = 1$. The terminal time of the solid blue domain is $(4,3)$, while the terminal time of the dashed red domain is $(2,4)$.

3.2 Hybrid Optimal Control Problem

Our hybrid MPC algorithm, introduced in the next subsection, solves the following minimization problem over solution pairs with compact domains at every optimization time. In addition to the explicit constraints described by $\mathcal{T}$ and X, which dictate that feasible trajectories have their terminal times in $\mathcal{T}$ and their terminal point in X, the data of $\mathcal{H}$ defines implicit state-input constraints since solution pairs are allowed to flow only on C and to jump only on D.

Problem ($\star$). Given $x_0 \in \mathbb{R}^n$,

$$\begin{aligned} \text{minimize} \quad & \mathcal{J}(x,u) \\ \text{subject to} \quad & (x,u) \in \widehat{\mathcal{S}}_{\mathcal{H}}(x_0) \\ & x(T,J) \in X \\ & (T,J) \in \mathcal{T}, \end{aligned}$$

where $\widehat{\mathcal{S}}_{\mathcal{H}}(x_0)$ is the set of solution pairs (x,u) of $\mathcal{H}$ satisfying $x(0,0) = x_0$, and (T,J) denotes the terminal time of (x,u).

Existence of solutions to Problem ($\star$) is easily addressed using techniques similar to Theorem 5.2 of Goebel (2017). However, computation of such solutions is a nontrivial task. The development of numerical methods to solve Problem ($\star$) is the current object of research, and can take the form of Hamilton-Jacobi-Bellman equations or the maximum principle Sussmann (1999). See, for example, Pakniyat and Caines (2017) and the references therein.

3.3 Hybrid MPC Algorithm

Using the definitions above, we are ready to introduce our hybrid MPC algorithm, which features a *control horizon* to regulate the optimization times. The control horizon has the same structure as the prediction horizon $\mathcal{T}$ in (6), parametrized by $\tau_c \in \{1, 2, \dots, \tau\}$ and $\delta_c \in (0, \delta]$. Using this parameter, the moving horizon implementation is summarized by the inner while loop of the hybrid MPC algorithm, demonstrated later in Figures 2 and 3.

Algorithm Hybrid MPC

```
1:  Set $i = 0$.
2:  Set initial prediction time $(T_0, J_0) = (0, 0)$.
3:  Set $x_0 = x(0, 0)$.
4:  while true do
5:      Solve Problem ($\star$) to obtain the optimal solution
        pair $(x_i^\star, u_i^\star)$.
6:      while $\max\{(t - T_i)/\delta_c, j - J_i\} \le \tau_c$ do
7:          Apply $u_i^\star$ to $\mathcal{H}$ to generate the trajectory $x$.
8:      end while
9:      $i = i + 1$.
10:     $(T_i, J_i) = (t, j)$.
11:     $x_0 = x(T_i, J_i)$.
12: end while
```

In the algorithm, the trajectory x results from the application of a sequence of optimal control inputs $\{u_i^\star\}_{i=0}^\infty$ to $\mathcal{H}$. The portion of the state trajectory x from hybrid time (T_i, J_i) to (T_{i+1}, J_{i+1}) corresponds to the optimal state trajectory $x_i^\star$ associated with $u_i^\star$ (Line 5), where the optimization times $\{(T_i, J_i)\}_{i=0}^\infty \in \operatorname{dom} x$ are regulated online. The first optimization occurs at $(T_0, J_0) = (0, 0)$ with optimal control parameter $x_0 = x(0, 0)$ (Lines 1-3) and leads to the the optimal control input $u_0^\star$ (Line 5). The input $u_0^\star$ (Line 5) is applied until either τ_c/δ_c units of ordinary time elapses, or τ_c jumps occur, whichever comes first (Lines 6-8). More formally, $u_0^\star$ is applied until a hybrid time $(t, j) \in \operatorname{dom} x$ of the generated trajectory satisfies $\max\{t/\delta_c, j\} = \tau_c$. At this point, (t, j) is recorded in (T_1, J_1), and the parameter x_0 of the optimal control problem is changed to $x(T_1, J_1)$ (Lines 9-11). Then, Problem ($\star$) is re-solved with $x_0 = x(T_1, J_1)$ to find the optimal control $u_1^\star$, which is again applied until either τ_c/δ_c units of ordinary time elapses after hybrid time (T_1, J_1), or τ_c jumps occur after hybrid time (T_1, J_1), whichever comes first (Line 6). Note that due to the condition in Line 6, $(T_{i+1} - T_i, J_{i+1} - J_i)$ is not necessarily constant. However, when the jump set D is empty, the algorithm simplifies to the typical periodic implementation of continuous-time MPC. Similar observations can be made when C is empty, with $\tau_c = 1$ recovering the standard "one-step ahead" implementation in discrete-time. We emphasize that the optimal control inputs $u_k^\star$ are defined over hybrid time domains and therefore are not discretized during flows.

The hybrid MPC algorithm is demonstrated in Figure 2 for the case of $\tau = 4$, $\delta = 1$ $\tau_c = 2$ and $\delta_c = 1$. The optimal control $u_0^\star$, whose domain is shown by the dash-dotted purple line, is applied until $(T_1, J_1) = (2, 1)$, 2 ordinary time units after $(T_0, J_0) = (0, 0)$. Problem ($\star$) is then re-solved to find the new optimal control $u_1^\star$. The domain of $u_1^\star$, shifted forward in-time by (T_1, J_1), is indicated by solid blue. The next optimization occurs at $(T_2, J_2) = (2, 3)$, 2 jumps after the prior optimization time (T_1, J_1). This leads to the optimal controls $u_2^\star$ to be applied until $(T_3, J_3) = (4, 4)$, exactly 2 ordinary time units after (T_2, J_2); the domain of $u_2^\star$ shifted by (T_2, J_2) is depicted by the dashed red line. The domain of the resulting trajectory until (T_3, J_3) is shown in Figure 3.

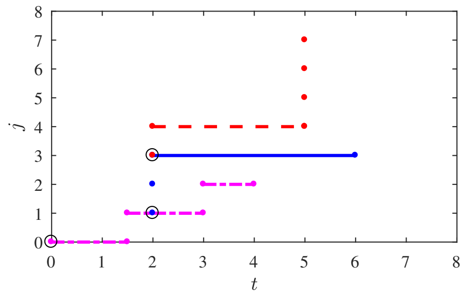


Fig. 2. Illustration of the hybrid MPC algorithm for the case of $\tau = 4$, $\delta = 1$, $\tau_c = 2$, and $\delta_c = 1$.

Remark 3.2. In Pereira et al. (2015), the optimization times are governed in real time in a similar fashion. However, while the framework there forces an optimization after every impulsive control input, the optimization times of the hybrid MPC algorithm can be after jumps of the optimal solution pairs. Furthermore, the algorithm guarantees $\lim_{i \to \infty} T_i + J_i = \infty$.

Remark 3.3. As in conventional MPC, when optimal controls are nonunique, the hybrid MPC algorithm can generate nonunique state trajectories from the same initial condition, which may have different hybrid time domains, and consequently, different optimization times.

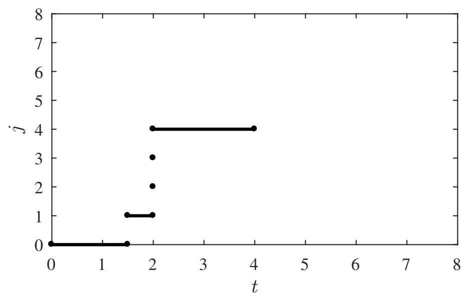


Fig. 3. Hybrid time domain of the generated trajectory corresponding to Figure 2.

4. ASYMPTOTIC STABILITY OF HYBRID MPC

The proposed MPC algorithm guarantees an asymptotic stability property when the cost functions, the terminal constraint, and the point (or set) to stabilize satisfy conditions that are similar to those encountered in the continuous/discrete-time MPC literature. Next, we outline the basic conditions required in the hybrid dynamical systems setting; in particular, the need for set stabilization, and the properties of the cost functions and terminal constraint relative to this set.

In general, the aforementioned stability properties are only meaningful with respect to compact sets, denoted by $\mathcal{A}$. This generalization of stability notions from equilibrium points to compact sets, which are in the usual Lyapunov sense (see (Goebel et al., 2012, Definition 7.1)), is due to the fact that hybrid systems can have state variables that do not necessarily converge to an equilibrium point (e.g., timers and logic variables). For the sample-and-hold control system described in Example 2.2, since the timer increases with rate 1 in the intersample period and gets reset periodically, any set $\mathcal{A}$ to be stabilized for this system must be such that its projection onto $[0, \bar{\tau}]$ contains $[0, \bar{\tau}]$. In particular, if the control objective is to stabilize the origin for the plant state z, and $z \in \mathbb{R}^p$, the set $\mathcal{A}$ can be taken as $\{0_p\} \times \{0\} \times [0, \bar{\tau}]$, where $0_p \subset \mathbb{R}^p$ is the zero vector.

In MPC regulation problems with no state constraints, the terminal constraint set X is designed to contain the origin in its interior. In our hybrid setting, this can be restrictive when $C' \cup D' \neq \mathbb{R}^n$. Instead, we assume that $\mathcal{A}$ is contained in the relative interior of X, in the sense that there exists an open set $S \supset \mathcal{A}$ such that $S \cap (C' \cup D') \subset X$. Indeed, for the bouncing ball in Example 2.1, no $X \subset C' \cup D' = C'$ contains the origin in its interior as it resides on the boundary of C'; however, X can be chosen as the closure of $S \cap C'$, for a neighborhood S of the origin.

As in conventional MPC, the cost functions defining the functional $\mathcal{J}$ in (5) should be chosen to have basic positive definiteness properties, with respect to the set $\mathcal{A}$ of interest. In particular, L_C and L_D should be designed such that

$$\begin{aligned} L_C(x,u) &\geq \alpha_C(|x|_{\mathcal{A}}) \quad \forall (x,u) \in C, \\ L_D(x,u) &\geq \alpha_D(|x|_{\mathcal{A}}) \quad \forall (x,u) \in D, \end{aligned}$$

for some class-$\mathcal{K}_\infty$ functions α_C and α_D,

$$\begin{aligned} L_C(x,u) &= 0 \quad \forall (x,u) \in (\mathcal{A} \cap C') \times U_C, \\ L_D(x,u) &= 0 \quad \forall (x,u) \in (\mathcal{A} \cap D') \times U_D, \end{aligned}$$

and V should be designed such that $V(x) = 0$ for all $x \in \mathcal{A}$ and $V(x) > 0$ for all $x \in X \backslash \mathcal{A}$.

The basic stabilizing ingredient we impose on the cost functions and terminal constraint extends the familiar CLF-like assumption in the MPC literature (Mayne, 2014, Section 2.2.1). Essentially, it requires V to be a CLF on the terminal constraint set X with respect to $\mathcal{A}$, and the existence of a feedback pair κ (as in (4)) rendering X forward invariant and $\mathcal{A}$ asymptotically stable. Specifically, it requires that V is differentiable on a neighborhood of $X \cap C'$,

$$\begin{aligned} \langle \nabla V(x), f_\kappa(x) \rangle &\leq -L_C(x, \kappa_C(x)) \quad \forall x \in X \cap C', \\ V(g_\kappa(x)) - V(x) &\leq -L_D(x, \kappa_D(x)) \quad \forall x \in X \cap D', \end{aligned} \tag{7}$$

and solution pairs generated by the feedback κ from the set X should be such that the state trajectory remains in X. These conditions are "necessary" in the context of stability of autonomous hybrid systems, in the sense that under mild conditions, asymptotic stability implies the existence of a Lyapunov function with exponential decrease during flows and jumps (Goebel et al., 2012, Theorem 7.31). Similarly, also under mild conditions, existence of a stabilizing feedback κ and a CLF V satisfying (7) is equivalent (Sanfelice, 2013). When solution pairs of $\mathcal{H}$ have persistent jumps (respectively, flows), the right-hand side of the second (respectively, first) inequality in (7) can be replaced with zero to establish local asymptotic stability under κ, as shown in Proposition 3.24 (respectively, 3.27) of Goebel et al. (2012), which could be extended for our hybrid MPC algorithm.

It can be shown that under the conditions we impose, the proposed hybrid MPC is asymptotically stabilizing. Asymptotic stability of $\mathcal{A}$ is certified by the *value function*, which is continuous on $\mathcal{A}$, has a continuous positive definite (with respect to the distance to $\mathcal{A}$) lower bound, and decreases along trajectories due to (7). Furthermore, these conditions ensure that Problem ($\star$) does not lose feasibility along feasible trajectories. The formal statements and proofs corresponding to these properties will be published elsewhere.

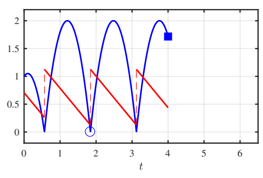
5. CASE STUDY: CONTROL OF THE BOUNCING BALL

The aim of this section is to demonstrate the hybrid MPC algorithm with the bouncing ball in Example 2.1. Consider the total energy $W(x) := \gamma x_1 + x_2^2/2$ of the ball when $x \in C' \cup D' = C'$. The control objective is to stabilize the limit cycle of the system originating from $(h, 0)$ under the feedback κ in Example 2.3, equivalently represented by the compact set $\mathcal{A} = \{x \in C' : W(x) = \gamma h\}$, where h is the desired height. To achieve this objective, we assume the terminal constraint $X = C'$, the prediction horizon parameters $\tau = 4$ and $\delta = 1$, and the following cost functions:

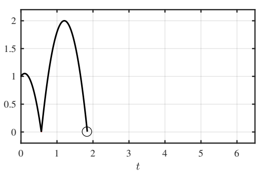
$$\begin{aligned} L_C(x,u) &= \gamma(W(x) - \gamma h)^2/(1 + 2W(x)) \quad && \forall (x,u) \in C, \\ L_D(x,u) &= \gamma h(x_2 - \sqrt{2\gamma h})^2/2 && \forall (x,u) \in D, \end{aligned}$$

$$V(x) = (3 + \arctan x_2)(W(x) - \gamma h)^2 \quad \forall x \in X,$$

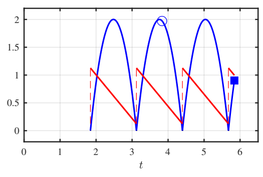
It can be verified that the cost functions and the terminal constraint satisfy the stabilizing conditions in Section 4.



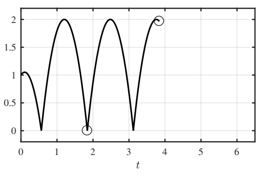
(a) Position trajectory and input predicted at (T_0, J_0)



(b) Generated position trajectory up to (T_1, J_1)



(c) Position trajectory and input predicted at (T_1, J_1)



(d) Generated position trajectory up to (T_2, J_2)

Fig. 4. Simulation of the bouncing ball under hybrid MPC.

Simulation results[2] of the bouncing ball under hybrid MPC with $\gamma = 9.81$, $\lambda = 0.8$, $h = 2$, $\tau_c = 2$, $\delta_c = 1$, and the data above are presented in Figure 4. Figures 4a and 4c show the optimal position trajectories (blue) and inputs (red) associated with the optimization times (T_0, J_0) and (T_1, J_1), respectively, projected onto t. The projection of the resulting position trajectory onto t is depicted in Figures 4b and 4d. The optimization times are indicated by the circles, and the terminal times are indicated by the squares. Here, we note that optimal state trajectories are none other than state trajectories of the system in Example 2.3, and as such, the simulations are performed with this autonomous model, using the Hybrid Equations Toolbox (Sanfelice et al., 2013): since there are no input constraints during jumps ($U_D = \mathbb{R}$) and the jump cost L_D does not penalize u, the flow map f does not depend on u and the set $\mathcal{A}$ is forward invariant under the flow map f_κ, and $g_\kappa(D') \subset \mathcal{A}$, optimal solution pairs must be generated by the feedback κ.

6. CLOSING REMARKS

In this paper, a new formulation of MPC for hybrid dynamical systems based on finite-horizon hybrid optimal control is proposed. The optimal control problem corresponding to the hybrid MPC algorithm is formulated with a novel set-based notion of a terminal time to account for hybrid time domains. Asymptotic stability of compact sets under the proposed approach can be guaranteed by designing the terminal cost to be a CLF on the terminal constraint set. Simulation result for the bouncing ball with actuated jumps show the effectiveness of the proposed MPC approach. Potential applications of the algorithm include bipedal robotic walking and power systems.

There are many open problems relevant to the work we have presented here. On the near horizon, numerical solutions to the optimal control problem, in addition to extension of the proposed approach to the case where flows are discretized is the focus of our current work. Another problem of interest is the relaxation of the Lyapunov conditions and structure of the prediction horizon for the systems with persistent flows or jumps. Indeed, due to persistent jumps, stability of the origin for the bouncing ball can be verified by using the total energy function as a Lyapunov certificate, which is constant during flows since the flow map is not dissipative. Finally, although robustness is a challenging problem, existing theory for autonomous hybrid systems shows a promising way to tackle this issue for certain classes of hybrid systems.

[2] Files for this simulation can be found at the following adress: https://github.com/HybridSystemsLab/HybridMPCBouncingBall

REFERENCES

Borrelli, F., Bemporad, A., and Morari, M. (2017). *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press, New York, NY.

Camacho, E., Ramirez, D., Limon, D., Muñoz de la Peña, D., and Alamo, T. (2010). Model predictive control techniques for hybrid systems. *Annual Reviews in Control*, 34(1), 21 – 31.

Cassandras, C.G. and Mookherjee, R. (2003). Receding horizon control for a class of hybrid systems with event uncertainties. In *Proceedings of the 2003 American Control Conference, 2003.*, volume 6, 5197–5202 vol.6.

Goebel, R., Sanfelice, R.G., and Teel, A.R. (2009). Hybrid dynamical systems. *IEEE Control Systems*, 29(2), 28–93. doi:10.1109/MCS.2008.931718.

Goebel, R. (2017). Optimal control for pointwise asymptotic stability in a hybrid control system. *Automatica*, 81, 397 – 402.

Goebel, R., Sanfelice, R.G., and Teel, A.R. (2012). *Hybrid Dynamical Systems: Modeling, Stability, and Robustness*. Princeton University Press, Princeton, NJ.

Magni, L. and Scattolini, R. (2004). Model predictive control of continuous-time nonlinear systems with piecewise constant control. *IEEE Transactions on Automatic Control*, 49(6), 900–906.

Mayne, D.Q. (2014). Model predictive control: Recent developments and future promise. *Automatica*, 50(12), 2967 – 2986.

Pakniyat, A. and Caines, P.E. (2017). On the relation between the minimum principle and dynamic programming for classical and hybrid control systems. *IEEE Transactions on Automatic Control*, 62(9), 4347–4362.

Pereira, F.L., Fontes, F.A.C.C., Aguiar, A.P., and de Sousa, J.B. (2015). *An Optimization-Based Framework for Impulsive Control Systems*, 277–300. Springer International Publishing, Cham.

Sanfelice, R.G. (2013). On the existence of control lyapunov functions and state-feedback laws for hybrid systems. *IEEE Transactions on Automatic Control*, 58(12), 3242–3248.

Sanfelice, R., Copp, D., and Nanez, P. (2013). A toolbox for simulation of hybrid systems in matlab/simulink: Hybrid equations (HyEQ) toolbox. In *Proceedings of the 16th International Conference on Hybrid Systems: Computation and Control*, HSCC '13, 101–106. ACM, New York, NY, USA.

Sanfelice, R.G. (2019). *Hybrid Model Predictive Control*, to appear. Birkhäuser, Basel.

Sopasakis, P., Patrinos, P., Sarimveis, H., and Bemporad, A. (2015). Model predictive control for linear impulsive systems. *IEEE Transactions on Automatic Control*, 60(8), 2277–2282.

Sussmann, H.J. (1999). *A nonsmooth hybrid maximum principle*, 325–354. Springer London, London.