Hybrid Model Predictive Control

Ricardo G. Sanfelice

Abstract This article collects several model predictive control (MPC) strategies in the literature that have a hybrid flavor, which, due to the diverse use of the term *hybrid*, span a wide range of control settings. These include discrete-time systems with discontinuous right-hand sides, with states that include both continuous-valued and discrete-valued variables. It also includes systems controlled by MPC strategies using memory variables and logic states, continuous-time systems controlled by MPC strategies that update the feedback law periodically as well as those controlled by MPC strategies that employ more than one feedback controller. This article provides a unified presentation of these strategies with the purpose of serving as a self-contained summary of the state of the art in hybrid MPC, as a handbook with precise pointers to the literature to the interested control practitioner, and as a motivator for future research directions on the subject.

1 Summary

The literature features several MPC strategies that are labeled as hybrid, either due to features of the state of the system, its dynamics, or the control algorithm. The term *hybrid* in the context of MPC has been used to refer to systems that are to be controlled (or the control algorithm) with continuous-valued and discrete-valued state components; e.g., in the control of a thermostat system, a continuous-valued state component would represent temperature and a discrete-valued state component would represent whether the heating/cooling device is on or off. The term hybrid has also been used in the literature for systems with dynamics whose right-hand sides depend discontinuously on their state or on their input. In addition, the term hybrid

Ricardo Sanfelice, Associate Professor of Computer Engineering
University of California, Santa Cruz, 1156 High Street MS:SOE3, e-mail: `ricardo@ucsc.edu`

has also been used to emphasize nonsmoothness in the control algorithm, for instance, when the algorithm switches between different control laws or when it is implemented using the sample-data control paradigm.

Due to the need for digitally implementable control algorithms, it is natural to consider dynamical models given in discrete time. In fact, the vast majority of the results in the literature of hybrid MPC fall into such a category. This article presents those strategies first. There are also a number of strategies that follow a sampled-data control approach. Rather than discretizing the system to control, such approaches incorporate into the mathematical models the continuous-time dynamics of the plant as well as the (periodic) discrete events at which computations occur. These type of strategies are presented after the ones for discrete-time systems. Strategies for systems with combined continuous and discrete dynamics in which the state variables may flow and, at times exhibit jumps due to state or input conditions are scarce. As argued in Section 3, new advances in hybrid dynamical systems are needed to develop those strategies.

In light of the outlined state of the art, this chapter covers hybrid MPC results in each of the main three forms seen in the literature and is organized as follows:

1. Discrete-time MPC for systems modeled as discrete-time systems with discontinuous right-hand sides (Section 2.1);
2. Discrete-time MPC for systems modeled as discrete-time systems with a state that contains continuous and discrete-valued states (Section 2.2);
3. Discrete-time MPC for systems modeled as discrete-time systems using memory and logic variables (Section 2.3);
4. Continuous-discrete MPC for systems modeled as continuous-time systems, with piecewise continuous inputs (Section 2.4.1) and piecewise constant inputs (Section 2.4.2);
5. Continuous-discrete MPC for systems modeled as continuous-time systems with local static state-feedback controllers (Section 2.5);
6. Discrete-time MPC for systems modeled as continuous-time linear systems with impulses (Section 2.6).

2 Hybrid Model Predictive Control

The MPC strategies presented in this section perform the following tasks:

- **Measure** the *current state* of the system to control;
- **Predict** for a finite amount of time – the so-called *prediction horizon* – the trajectories of the system to control from the current state and for a family of allowed input signals;

- **Select** an input signal that is a *minimizer* of a given cost functional, which potentially depends on the predicted trajectories and the input, and that satisfies a terminal constraint (if one is given);
- **Apply** the input signal for a finite amount of time – the so-called *control horizon*.

Most MPC algorithms perform these tasks repetitively in the order listed. The following sections provide details on these tasks for each of the strategies listed in Section 1. Regardless of the type of model used, and unless otherwise stated, the state and the input of the system to control are denoted as x and u, while the state and input constraints (if any) are denoted as $\mathcal{X}$ and $\mathcal{U}$, respectively. When the model of the system to control is of discrete-time type, the notation x^+ indicates the value of the state after a discrete-time step. Discrete time is denoted as k, which takes values in $\mathbb{N} := \{0, 1, 2 \ldots\}$. For continuous-time models, the notation $\dot{x}$ denotes the derivative with respect to ordinary time. Ordinary time is denoted as t, and takes values from $\mathbb{R}_{\geq 0} := [0, \infty)$. The MPC strategies require solving an optimization problem using the current state of the system. Since the strategies presented in this article are stated for time-invariant systems, we treat the current state as an initial condition, and denote it as x_0. The prediction horizon in discrete time is denoted $N \in \mathbb{N}_{>0} := \{1, 2, \ldots\}$, while in continuous time is denoted by $T \in \mathbb{R}_{>0} := (0, \infty)$. Similarly, the control horizon in discrete time is denoted $N_c \in \mathbb{N}_{>0} := \{1, 2, \ldots\}$, while in continuous time is denoted by $T_c \in \mathbb{R}_{>0} := (0, \infty)$. We also define $\mathbb{N}_{<N} := \{0, 1, 2, \ldots, N-1\}$ and $\mathbb{N}_{\leq N} := \{0, 1, 2, \ldots, N\}$ for a given $N \in \mathbb{N}_{>0}$. Given a vector x, $|x|$ denotes its Euclidean norm and given $p \in [1, \infty]$, $|x|_p$ denotes its p-norm. Given $n \in \mathbb{N}_{>0}$, $\mathbb{R}^n$ denotes the Euclidean space of dimension n.

2.1 Discrete-time MPC for discrete-time systems with discontinuous right-hand sides

MPC for discrete-time systems that have piecewise-linear but discontinuous right-hand sides is studied in [19]. Under the name Piecewise Affine System (PWA), the systems considered in [19] take the form

$$x^+ = A_i x + B_i u + f_i \tag{1}$$
$$y = C_i x + D_i u \tag{2}$$
$$\text{subject to } x \in \Omega_i, u \in \mathcal{U}_i(x), i \in S \tag{3}$$

where $S := \{1, 2, \ldots, s\}$ with s finite, the sequence of constant matrices $\{(A_i, B_i, f_i, C_i, D_i)\}_{i \in S}$ has elements with appropriate dimensions, $\{\Omega_i\}_{i=1}^s$ is a collection of polyhedra such that

$$\bigcup_{i \in S} \Omega_i = \mathcal{X}$$

where $\mathcal{X}$ is the state space and

$$\text{int}(\Omega_i) \cap \text{int}(\Omega_j) = \emptyset \qquad \forall i \neq j, i, j \in S$$

where, for each $x \in \Omega_i$, $\mathcal{U}_i(x)$ is the set of allowed inputs. The subset of elements i in S for which $0 \in \overline{\Omega_i}$ is denoted as S_0, while all of the other elements in S define S_1. The origin of (1)-(3) is assumed to be an equilibrium state with $u = 0$, and the requirement $f_i = 0$ for all $i \in S_0$ is further imposed. It should be noted that in [19], this class of systems is referred to as hybrid, presumably due to the right-hand side being discontinuous – in fact, in general, the map

$$(x, u) \mapsto \{A_i x + B_i u + f_i \ : \ i \in S, x \in \Omega_i, u \in \mathcal{U}(x) \ \}$$

defined on $\bigcup_{i \in S} \bigcup_{x \in \Omega_i} (\{x\} \times \mathcal{U}(x))$ is discontinuous.

Given the current state x_0, a prediction horizon $N \in \mathbb{N}_{>0}$, a terminal constraint set $\mathcal{X}_f$, a stage cost $\mathcal{L}$, and a terminal cost $\mathcal{F}$, the problem of interest consists of minimizing the cost functional

$$\mathcal{J}(x, i, u) := \mathcal{F}(x(N)) + \sum_{k=0}^{N-1} \mathcal{L}(x(k), i(k), u(k))$$

whose argument is actually $k \mapsto (x(k), i(k), u(k))$, which is subject to the constrained dynamics in (1)-(3). Note that $k \mapsto x(k)$ is uniquely defined by x_0 and $k \mapsto (i(k), u(k))$. The initial state for the x component is such that $x(0) = x_0$ and the final value is restricted to $x(N) \in \mathcal{X}_f$. The argument $k \mapsto (x(k), i(k), u(k))$ of the functional is such that $x(k)$ is uniquely defined for each $k \in \mathbb{N}_{\leq N}$, while $(i(k), u(k))$ is uniquely defined for each $k \in \mathbb{N}_{<N}$.

The problem to solve at each discrete-time instant is as follows:

Problem 1. Given the current state x_0, a prediction horizon $N \in \mathbb{N}_{>0}$, a terminal constraint set $\mathcal{X}_f$, a stage cost $\mathcal{L}$, and a terminal cost $\mathcal{F}$

$$
\begin{aligned}
&\min \mathcal{J}(x,i,u)\\
&\quad \text{subject to}\\
&\qquad x(0) = x_0\\
&\qquad x(N) \in \mathcal{X}_f\\
&\qquad x(k+1) = A_{i(k)}x(k) + B_{i(k)}u(k) + f_{i(k)} \qquad \forall k \in \mathbb{N}_{<N}\\
&\qquad \left.\begin{array}{l} y(k) = C_{i(k)}x(k) + D_{i(k)}u(k) \\ x(k) \in \Omega_{i(k)},\ u(k) \in \mathcal{U}_{i(k)}(x(k)),\ i(k) \in S \end{array}\right\} \quad \forall k \in \mathbb{N}_{\leq N}
\end{aligned}
$$

A minimizer[1] $k \mapsto (x^*(k), i^*(k), u^*(k))$ defines the value of the cost functional $\mathcal{J}^*(x_0) = \mathcal{J}(x^*, i^*, u^*)$.

A typical choice of the functions $\mathcal{L}$ and $\mathcal{F}$ in the cost functional $\mathcal{J}$ is

$$\mathcal{L}(x,i,u) = |Q_i x|_p + |R_i u|_p, \qquad \mathcal{F}(x) = |Px|_p$$

for some $p \in [1,\infty]$, where $\{(Q_i, R_i)\}_{i\in S}$ and P are matrices of appropriate dimensions. When $p = 1$ or $p = \infty$, Problem 1 can be rewritten as a mixed integer linear program (MILP). When the stage and terminal costs are quadratic, Problem 1 can be rewritten as a mixed integer quadratic program (MIQP).

Key properties of Problem 1 were reported in [19], which due to space constraints are not included here. Under suitable assumptions, conditions guaranteeing recursive feasibility and asymptotic stability of the origin are given in [19, Theorem III.2]. Properties of and techniques for the computation of the terminal cost and terminal constraint set are also given; see [19, Section IV and Section V]. The issue of existence of minimizers for Problem 1 requires careful treatment, in particular, due to the partitions of the state space introduced by the sets Ω_i. Furthermore, due to Problem 1 being a nonconvex nonlinear optimization problem, the authors of [19] suggest to use optimization solvers such as *fmincon* and *fminunc* in Matlab.

2.2 Discrete-time MPC for discrete-time systems with mixed states

An MPC formulation for discrete-time systems to handle switching among different linear dynamics, on/off inputs, logic states and their transitions, as well as logic constraints on input and state variables is given in [3, 2, 4, 19,

[1] Or, equivalently, $k \mapsto (i^*(k), u^*(k))$, due to $k \mapsto x^*(k)$ being uniquely defined by x_0 and $k \mapsto (i^*(k), u^*(k))$.

5]. The nominal models considered therein, which are called Mixed Logical Dynamical (MLD) systems, are discrete-time systems involving continuous-valued and discrete-valued states, inputs, and outputs, as well as constraints depending on the states, the inputs, and the outputs. These system models are given as

$$x^+ = Ax + B_1u + B_2\delta + B_3z + B_4 \tag{4}$$
$$y = Cx + D_1u + D_2\delta + D_3z + D_4 \tag{5}$$
$$\text{subject to } E_2\delta + E_3z \leq E_1u + E_4x + E_5 \tag{6}$$

In most MLD models in the literature, the state vector x is partitioned as (x_c, x_ℓ), where $x_c \in \mathbb{R}^{n_c}$ are the continuous-valued components and $x_\ell \in \{0,1\}^{n_\ell}$ are the discrete-valued components of x. Similarly, the input u is partitioned as $(u_c, u_\ell) \in \mathbb{R}^{m_c} \times \{0,1\}^{m_\ell}$ and the output y as $(y_c, y_\ell) \in \mathbb{R}^{p_c} \times \{0,1\}^{p_\ell}$. The continuous-valued auxiliary variables $z \in \mathbb{R}^{r_c}$ and the discrete-valued auxiliary variables $\delta \in \{0,1\}^{r_\ell}$ are added to capture constraints, logic statements, and such. The matrices A, $\{B_i\}_{i=1}^3$, B_4, C, $\{D_i\}_{i=1}^3$, D_4, and $\{E_i\}_{i=1}^5$ have suitable dimensions. Given the current state x_0, a prediction horizon $N \in \mathbb{N}_{>0}$, and a terminal constraint set $\mathcal{X}_f$, the problem of interest consists of minimizing the cost functional

$$\mathcal{J}(x, z, \delta, u)$$

whose argument is actually $k \mapsto (x(k), z(k), \delta(k), u(k))$, and is subject to the constrained dynamics in (4)-(6). The initial state for the x component is such that $x(0) = x_0$ and its final value is restricted to $x(N) \in \mathcal{X}_f$. In the literature, this class of dynamical systems is referred to as hybrid mainly due to having a discontinuous right-hand side and due to the states, inputs, and outputs having continuous-valued and discrete-valued components.

The problem to solve at each discrete-time instant is as follows:

Problem 2. Given the current state x_0, a prediction horizon $N \in \mathbb{N}_{>0}$, a terminal set $\mathcal{X}_f$, and a cost functional $\mathcal{J}$

$$\begin{aligned}
&\min \mathcal{J}(x, z, \delta, u)\\
&\quad \text{subject to}\\
&\qquad x(0) = x_0\\
&\qquad x(N) \in \mathcal{X}_f\\
&\qquad x(k+1) = Ax(k) + B_1u(k) + B_2\delta(k) + B_3z(k) + B_4 \quad \forall k \in \mathbb{N}_{<N}\\
&\qquad \left.\begin{array}{l} y(k) = Cx(k) + D_1u(k) + D_2\delta(k) + D_3z(k) + D_4 \\ E_2\delta(k) + E_3z(k) \leq E_1u(k) + E_4x(k) + E_5 \end{array}\right\} \quad \forall k \in \mathbb{N}_{\leq N}
\end{aligned}$$

A minimizer $k \mapsto (x^*(k), z^*(k), \delta^*(k), u^*(k))$ defines the value of the cost functional $\mathcal{J}^*(x_0) = \mathcal{J}(x^*, z^*, \delta^*, u^*)$.

A particular choice of the cost functional $\mathcal{J}$ made in [3, 2, 4, 19, 5] is

$$\mathcal{J}(x, z, \delta, u) = \sum_{k=0}^{N-1} \left(|Qx(k)|_p + |Ru(k)|_p + |Q_\delta \delta(k)|_p + |Q_z z(k)|_p \right) + |Px(N)|_p$$

for some $p \in [1, \infty]$. The term inside the sum is the stage cost, which, given matrices Q, Q_δ, and Q_z, involves the value of the current and predicted state x, input u, and auxiliary variables (δ, z) for $N-1$ steps in the future. The last term in $\mathcal{J}$ is the terminal cost.

Perhaps the most comprehensive reference about Problem 2 is Chapter 18 of the recent monograph [5]. Therein, the authors consider the same model (but with $B_4 = 0$ and $D_4 = D_5$) in Section 18.1. By picking the cost functional above, Problem 2 is formulated as a MIQP or MILP, according to the choice of p. A complete rewrite of Problem 2 including slack variables is given in (18.28) in the said reference. Mixed-integer optimization methods suitable to solve Problem 2 are also outlined. The chapter concludes with discussions on how to derive state feedback solutions via the batch approach and the recursive approach. This class of systems is referred to as hybrid due to the right-hand side being discontinuous and due to the states, inputs, and outputs having continuous-valued and discrete-valued components.

2.3 Discrete-time MPC for discrete-time systems using memory and logic variables

Variations of the basic MPC formulation, obtained by adding memory and logic states, for discrete-time systems of the following form is proposed in [38]:

$$x^+ = g(x, u) \tag{7}$$

$$\text{subject to } x \in \mathcal{X}, u \in \mathcal{U} \tag{8}$$

The set $\mathcal{X}$ defines the constraint on the state and $\mathcal{U}$ is the set of allowed inputs. Recall from Chapter 1 of this handbook (see also Chapter 2 and Chapter 3) that the basic MPC formulation consists of minimizing the cost functional

$$\mathcal{J}(x, u) := \mathcal{F}(x(N)) + \sum_{k=0}^{N-1} \mathcal{L}(x(k), u(k))$$

where x is the current state, $N \in \mathbb{N}_{>0}$ is the prediction horizon, $\mathcal{L}$ is the stage cost, and $\mathcal{F}$ is the terminal cost. The function $k \mapsto x(k)$ in the cost functional $\mathcal{J}$ is the solution to (7)-(8) at time k, starting from the initial condition x_0 and under the influence of the input sequence $k \mapsto u(k)$. The two variations of this MPC formulation proposed in [38] are described next.

To incorporate memory in the selection of the input, define the buffer gain as $\mu > 1$, the memory horizon as $M \in \mathbb{N}_{>0}, M \leq N$, and the memory state as $\ell = (\ell_1, \ell_2, \dots, \ell_M)$. The optimization problem in [38] involving the memory state ℓ that is to be solved at each discrete-time instant is as follows:

Problem 3. Given the current state x_0, a prediction horizon $N \in \mathbb{N}_{>0}$, a stage cost $\mathcal{L}$, a terminal cost $\mathcal{F}$, a buffer gain $\mu > 1$, a memory horizon $M \in \mathbb{N}_{>0}$ such that $M \leq N$, and the current memory state ℓ, solve the following problems:

Problem 3a:

$$\begin{aligned} &\min \mathcal{J}(x,u) \\ &\quad \text{subject to} \\ &\qquad x(0) = x_0 \\ &\qquad x(k+1) = g(x(k), u(k)) \qquad \forall k \in \mathbb{N}_{<N} \\ &\qquad u(k) \in \mathcal{U} \qquad \forall k \in \mathbb{N}_{\leq N} \end{aligned}$$

Denote the solution to this problem as $k \mapsto (x^*(k), v^*(k))$ and define $V(x_0) = \mathcal{J}(x^*, v^*)$ as the associated value function.[2]

Problem 3b:

$$\begin{aligned} &\min \mathcal{J}(x,u) \\ &\quad \text{subject to} \\ &\qquad x(0) = x_0 \\ &\qquad x(k+1) = g(x(k), u(k)) \qquad \forall k \in \mathbb{N}_{<N} \\ &\qquad u(k-1) = \ell_k \qquad \forall k \in \{1, 2, \dots, M\} \\ &\qquad u(k) \in \mathcal{U} \qquad \forall k \in \mathbb{N}_{\leq N} \end{aligned}$$

Denote the solution to this problem as $k \mapsto (x^*(k), w^*(k))$ and define $W(x_0, \ell) = \mathcal{J}(x^*, w^*)$ as the associated value function.

After solving[3] Problem 3a and Problem 3b, update the memory state according to

$$\ell^+ = \begin{cases} (v^*(1), v^*(2), \dots, v^*(M)) & \text{if } W(x_0, \ell) > \mu V(x_0) \\ (w^*(1), w^*(2), \dots, w^*(M)) & \text{if } W(x_0, \ell) \leq \mu V(x_0) \end{cases}$$

and the minimizing control input $k \mapsto u^*(k)$ is[4]

$$u^* = \begin{cases} v^* & \text{if } W(x_0, \ell) > \mu V(x_0) \\ w^* & \text{if } W(x_0, \ell) \leq \mu V(x_0) \end{cases}$$

Problem 3a provides a solution to the standard MPC problem without memory states. The solution from this problem is used in Problem 3b, which uses the current value of the memory state ℓ as it enforces that the first M entries of u, namely, $(u(0), u(1), \ldots, u(M-1))$, are equal to ℓ. The selection of the control input is such that the improvement provided by the solution to the standard MPC problem is significant when compared to the one with the memory states. The optimal control input u^* is given by v^* in Problem 3a when the improvement provided by the solution to that problem (namely, $k \mapsto (x^*(k), v^*(k))$) is "significantly better" – as characterized by the buffer gain $\mu > 1$ – than the improvement provided by the solution to the problem involving memory states (namely, $k \mapsto (x^*(k), w^*(k))$ in Problem 3b). More precisely, if the value function of the problem that does not use information about the previous solution (i.e., Problem 3a) is a factor $1/\mu \in (0, 1)$ smaller than the value function of the problem solved using previous information (i.e., Problem 3b), namely,

$$V(x_0) < \frac{1}{\mu} W(x_0, \ell) \tag{9}$$

then the optimal solution comes from Problem 3a and the memory state is updated with the input component of the solution to that problem. Note that when $V(x_0) \geq \frac{1}{\mu} W(x_0, \ell)$ and the control horizon is equal to one, the input applied to the system to be controlled would be ℓ_1 and that ℓ is subsequently updated to $(\ell_2, \ell_3, \ldots, \ell_M, w^*(M))$.

To incorporate logic states in the selection of the input, define the buffer gain as $\mu > 1$, the logic state as q taking its value from $Q := \{1, 2, \ldots, \bar{q}\}$ where $\bar{q} \in \mathbb{N}_{>0}$, and, for each $q \in Q$, define the cost functional

$$\mathcal{J}_q(x, u) := \mathcal{F}_q(x(N)) + \sum_{k=0}^{N-1} \mathcal{L}_q(x(k), u(k))$$

[2] Note that the only constraint on $v^*(N)$ is for it to belong to $\mathcal{U}$.

[3] The solution component x^* in Problem 3a and in Problem 3b would be most likely different, but we use the same label due to it not being part of the logic.

[4] The state component $k \mapsto x^*(k)$ associated to $k \mapsto u^*(k)$ is obtained by applying u^* to the system to be controlled.

where $\mathcal{L}_q$ is the stage cost and $\mathcal{F}_q$ the terminal cost associated with q. The proposed optimization problem involving a logic variable q to solve at each discrete-time instant is as follows:

Problem 4. Given the current state x_0, a prediction horizon $N \in \mathbb{N}_{>0}$, stage costs $\{\mathcal{L}_q\}_{q\in Q}$, terminal costs $\{\mathcal{F}_q\}_{q\in Q}$, and a buffer gain $\mu > 1$, solve the following problem for each $q \in Q$:

Problem 4-q:

$$\begin{aligned} &\min \mathcal{J}_q(x,u) \\ &\text{subject to} \\ &x(0) = x_0 \\ &x(k+1) = g(x(k),u(k)) \qquad \forall k \in \mathbb{N}_{<N} \\ &u(k) \in \mathcal{U} \qquad \forall k \in \mathbb{N}_{\leq N} \end{aligned}$$

Denote the solution to this problem as $k \mapsto (x^{q*}(k), v^{q*}(k))$ and define $V_q(x_0) = \mathcal{J}_q(x^{q*}, v^{q*})$ as the associated value function.

After solving Problem 4-q for each $q \in Q$, pick

$$q^* \in \arg\min_{q\in Q} V_q(x_0)$$

update the logic state according to

$$q^+ = \begin{cases} q^* & \text{if } V_q(x_0) > \mu V_{q^*}(x_0) \\ q & \text{if } V_q(x_0) \leq \mu V_{q^*}(x_0) \end{cases}$$

and the minimizing control input $k \mapsto u^*(k)$ is

$$u^* = \begin{cases} v^{q^*} & \text{if } V_q(x_0) > \mu V_{q^*}(x_0) \\ v^q & \text{if } V_q(x_0) \leq \mu V_{q^*}(x_0) \end{cases}$$

The value functions $V_q(x_0)$ associated to each optimal solution obtained in Problem 4-q are compared when determining a new value of the logic variable. Such value of q is denoted as q^*, and is such that $V_{q^*}(x_0)$ is among those minimizers in $\{V_q(x_0)\}_{q\in Q}$ with "enough improvement" – as characterized by μ –when compared to the value function associated to the current value of q. In fact, according to Problem 4, a change on the value of the logic variable occurs when there exists $q^* \in Q$ such that

$$V_{q^*}(x_0) < \frac{1}{\mu} V_q(x_0) \tag{10}$$

Discussions in [38] indicate that the MPC strategies with memory states and logic variables guarantee robustness to small measurement noise. Such robustness is possible due to the hysteresis mechanism incorporated by conditions (9) and (10) in the strategies above. It is also likely that these MPC strategies confer robustness to other classes of perturbations, mainly due to the said hysteresis mechanism they implement, which, in particular, prevents the control law from chattering.

2.4 Periodic Continuous-discrete MPC for Continuous-time Systems

In this section, we present model predictive control strategies for continuous-time systems that periodically recompute an input signal solving the optimization problem and apply it over a bounded horizon. Such MPC strategies appear in the literature under the name *continuous-discrete MPC*.

2.4.1 With piecewise continuous inputs

MPC for continuous-time systems with input constraints is proposed in [7]. The class of systems is given by

$$\dot{x} = f(x, u) \qquad x \in \mathbb{R}^n, \; u \in \mathcal{U} \tag{11}$$

where $\mathcal{U}$ is the input constraint set. The right-hand side f is assumed to be twice continuously differentiable, to satisfy $f(0, 0) = 0$, and such that it leads to unique solutions under the effect of piecewise right-continuous input signals. The input constraint set $\mathcal{U}$ is assumed to be compact, convex, and with the property that 0 belongs to the interior of $\mathcal{U}$.

Given the current state x_0, a prediction horizon $T > 0$, a terminal constraint set $\mathcal{X}_f$, a stage cost $\mathcal{L}$, and a terminal cost $\mathcal{F}$, the problem of interest consists of minimizing the cost functional

$$\mathcal{J}(x, u) := \mathcal{F}(x(T)) + \int_0^T \mathcal{L}(x(\tau), u(\tau))d\tau \tag{12}$$

whose argument is actually $t \mapsto (x(t), u(t))$ which is subject to the constrained dynamics in (11). The initial condition is such that $x(0) = x_0$, and the value of x after T seconds is restricted to $\mathcal{X}_f$. More precisely, the problem to solve every T seconds is as follows:

Problem 5. Given the current state x_0, a prediction horizon $T > 0$, a terminal constraint set $\mathcal{X}_f$, a stage cost $\mathcal{L}$, and a terminal cost $\mathcal{F}$

$$\begin{aligned} &\min \mathcal{J}(x,u) \\ &\quad \text{subject to} \\ &\qquad x(0) = x_0 \\ &\qquad x(T) \in \mathcal{X}_f \\ &\qquad \frac{d}{dt}x(t) = f(x(t),u(t)) \quad \forall t \in (0,T) \\ &\qquad u(t) \in \mathcal{U} \quad \forall t \in [0,T] \end{aligned}$$

A minimizer $t \mapsto (x^*(t), u^*(t))$ defines the value of the cost functional $\mathcal{J}^*(x_0) = \mathcal{J}(x^*, u^*)$.

In [7], the approach to solve this problem consists of picking $\mathcal{X}_f$ to be a neighborhood of the origin that is invariant in forward time for the closed-loop system resulting from using a (local) linear state-feedback law of the form Kx, and by picking $\mathcal{F}$ so that the terminal cost upper bounds the infinite horizon cost from $\mathcal{X}_f$. According to [7], the design of the set $\mathcal{X}_f$, the gain K, and the function $\mathcal{F}$ can be performed offline. Due to the value of the cost functional providing a bound to an infinite horizon cost problem, the authors refer to this strategy as *quasi-infinite horizon nonlinear MPC.*

The application of the stabilizing linear feedback law Kx to the system (11) generates a solution-input pair $t \mapsto (x(t), u(t))$ that satisfies the input and terminal constraints, for any initial condition $x_0 \in \mathcal{X}_f$. Therefore, the feasible set of initial conditions to Problem 5 includes $\mathcal{X}_f$. The actual moving horizon implementation of the MPC strategy in [7] would not use the (local) linear state-feedback law, but rather, guarantee feasibility. The moving horizon implementation would recursively apply the open-loop optimal control solution for $\delta < T$ seconds. The constant δ defines the sampling period for obtaining new measurements of the state of the plant. At such events, the optimal solution to the open-loop problem is recomputed and then the input to the plant is updated.

Note that forward/recursive feasibility of the closed-loop is guaranteed by the terminal constraint and the local feedback law Kx. This is because, as stated in [7], the MPC strategy can be thought of as a receding horizon implementation of the following switching control strategy:

- Over a finite horizon of length T, apply the optimal input obtained by solving Problem 5 so as to drive the state to the terminal set;
- Once the state is in the terminal set, switch the control input to the (local) linear state-feedback law so as to steer the state to the origin.

2.4.2 With piecewise constant inputs

A minimizing input $t \mapsto u^*(t)$ obtained from a solution to Problem 5 is a piecewise-continuous function defined on an interval of length equal to the prediction horizon T. Using a similar continuous-discrete MPC strategy, in [23], the class of inputs allowed is restricted to piecewise-constant functions and the strategy is of sample-and-hold type. More precisely, the input u satisfies the following:

($\star$) The input signal u is a piecewise-constant function with intervals of constant value of length δ seconds, within the control horizon $N_c\delta$, where $N_c \in \mathbb{N}_{>0}$ and $N_c\delta \leq T$.

In such a (zero-order) sample-and-hold approach, the input applied to the plant remains constant in between the sampling events. In [23], this mechanism is modeled by adding an extra state x_u to the system with the following dynamics:

$$\begin{aligned} \dot{x}_u &= 0 && \text{in between sampling events} \\ x_u^+ &= \kappa(x) && \text{at sampling events} \end{aligned}$$

where κ denotes the function assigning the feedback at each event. Furthermore, the setting in [23] allows for state constraints $x \in \mathcal{X}$ in (11), where $\mathcal{X}$ is the state constraint set.

Given the current state x_0, a prediction horizon T, a sampling time $\delta \in (0, T]$, a control horizon $N_c\delta \leq T$, and a terminal constraint set $\mathcal{X}_f$, the problem formulated in [23] is that of minimizing (12) at every sampling time instant, where

$$\mathcal{F}(x) = x^\top P x, \qquad \mathcal{L}(x, u) = x^\top Q x + u^\top R u \tag{13}$$

for given matrices P, Q, and R of appropriate dimensions. The argument of (12) is actually $t \mapsto (x(t), u(t))$ with the input component being a piecewise constant function.

The problem to solve at each periodic sampling event occurring every δ seconds is as follows:

Problem 6. Given the current state x_0, a prediction horizon $T > 0$, a sampling time $\delta \in (0, T]$, a control horizon $N_c\delta \in (0, T]$, a terminal constraint set $\mathcal{X}_f$, and matrices P, Q, and R

$$\begin{aligned}
&\min \mathcal{J}(x,u)\\
&\quad \text{subject to}\\
&\qquad x(0) = x_0\\
&\qquad x(T) \in \mathcal{X}_f\\
&\qquad \frac{d}{dt}x(t) = f(x(t),u(t)) \quad \forall t \in (0,T)\\
&\qquad x(t) \in \mathcal{X}, \quad u(t) \in \mathcal{U} \quad \forall t \in [0,T]\\
&\qquad u \text{ satisfies } (\star)
\end{aligned}$$

A minimizer $t \mapsto (x^*(t), u^*(t))$ defines the value of the cost functional $\mathcal{J}^*(x_0) = \mathcal{J}(x^*, u^*)$.

A somewhat related problem that involves periodic continuous-discrete MPC for continuous-time systems with piecewise constant inputs was studied in [28]. In that reference, MPC is used to solve the problem of finding a sampled version of a continuous-time controller that leads to a trajectory of the resulting sample-data system that is as close as possible to the trajectory of the closed-loop system with the original continuous-time controller. To characterize closeness between them, the stage cost of the MPC problem in [28] penalizes the error between the two trajectories.

2.5 Periodic Continuous-time MPC for continuous-time systems combined with local static state-feedback controllers

A strategy uniting two controllers for the asymptotic stabilization of the origin of continuous-time systems in affine control form is provided in [10]; see also [8, Chapter 5]. The family of continuous-time systems considered in [10] is given by

$$\dot{x} = f_1(x) + f_2(x)u \qquad x \in \mathbb{R}^n,\ u \in \mathcal{U} \tag{14}$$

where $\mathcal{U} = \{u \ : \ |u| \le u_{\max}\}$ for some $u_{\max} \ge 0$ and $f_1(0) = 0$.

One of the controllers in the proposed strategy is a continuous-discrete MPC controller with piecewise-constant inputs and implemented with periodic sampling, similar to the strategy presented in Section 2.4.2. The stage cost used has the same form as in (13). In [10], this particular continuous-discrete MPC algorithm is designed so that, at each periodic sampling event, Problem 6 is solved with control horizon equal to the prediction horizon T and no state constraint set.

The second controller in the strategy in [10] consists of a family of finitely many locally stabilizing static state-feedback controllers $\{\kappa_1, \kappa_2, \ldots, \kappa_r\}$, $r \in \mathbb{N}_{>0}$, that are designed using a family of control Lyapunov functions $\{V_1, V_2, \ldots, V_r\}$, following the universal construction proposed in [20]. These individual control laws can be designed to satisfy the input constraint in (14). When the second controller is the one applied to (14), the particular element in the family that is actually used is such that x belongs to its basin of attraction, which in [10] is defined by a sublevel set of the control Lyapunov function associated with that controller.

The two controllers outlined above are combined via a strategy that uses the static state-feedback controllers as "fall-back" in the event that the continuous-discrete MPC controller is unable to achieve closed-loop stability, which could be the case when Problem 6 does not have a solution or does not terminate before δ seconds. The strategy proposed for combining them is as follows. The control system in (14) is treated as the switching system

$$\dot{x} = f_1(x) + f_2(x)u_\sigma \qquad x \in \mathbb{R}^n,\ u \in \mathcal{U}$$

where $t \mapsto \sigma(t) \in \{1, 2\}$ is a switching signal that determines which controller is being used: $\sigma = 1$ indicates that $u = u_1$ with u_1 assigned by the MPC control law κ, and $\sigma = 2$ that $u = u_2$ with u_2 assigned by an element in the family of static state-feedback laws $\{\kappa_1, \kappa_2, \ldots, \kappa_r\}$. The particular selection of σ in [10] is

$$\sigma(t) = \begin{cases} 1 & \text{if } t \in [0, \bar{T}) \\ 2 & \text{if } t \in [\bar{T}, \infty) \end{cases} \tag{15}$$

where $\bar{T}$ is the smallest time such that

$$L_{f_1} V_i(x(\bar{T})) + L_{f_2} V_i(x(\bar{T}))\kappa(\bar{T}) \geq 0 \tag{16}$$

or the MPC algorithm fails to provide an output value, where $i \in K := \{1, 2, \ldots, r\}$ is such that $x(0)$ belongs to the basin of attraction induced by the static state-feedback controller κ_i. The idea behind the state-based triggering condition (16) is that since $x(0)$ is in the basin of attraction of a controller in the family $\{\kappa_1, \kappa_2, \ldots, \kappa_r\}$, then a solution guarantees a strict decrease of the control Lyapunov function associated with that controller.

The work in [10] also includes a switching strategy that is designed to enhance closed-loop performance. Also, an extension to these strategies for the case when the right-hand side of (14) includes additive uncertainties is proposed in [25]. See also [26].

2.6 Periodic Discrete-time MPC for Continuous-time Linear Systems with Impulses

MPC for linear time-invariant systems with impulses in the state, and with state and input constraints is proposed in [33]. The set of times at which impulses occur are predetermined and given by the sequence of times

$$\{t_k\}_{k\in\mathbb{N}}, \qquad t_k = k\delta$$

where $\delta > 0$ is the sampling (or, as defined in [33], the impulsive) period. The class of impulsive systems is given by

$$\dot{x}(t) = Ax(t) \qquad \forall t \in \mathbb{R}_{\geq 0},\ t \neq k\delta \tag{17}$$

$$x(t^+) = x(t) + Bu_k \qquad \forall t = k\delta \tag{18}$$

for each $k \in \mathbb{N}$, where $t \mapsto x(t)$ is a solution associated to $\{u_k\}_{k\in\mathbb{N}}$ and such that

$$x(t) \in \mathcal{X} \qquad \forall t \in \mathbb{R}_{\geq 0}, \qquad u_k \in \mathcal{U} \qquad \forall k \in \mathbb{N}$$

and $x(t^+)$ is the right limit of $x(t)$ at $t = k\delta$.

The MPC problem in [33] employs over approximation techniques to reduce the infinite number of constraints arising from the dynamics of (17)-(18) to a finite set of inequalities. For a given $\delta > 0$, the collection $\{A_i\}_{i=1}^K$ is introduced to define a polytopic over approximation for the flows of (17)-(18), namely, choose $\{A_i\}_{i=1}^K$ such that

$$\{\exp(At)\ :\ t \in [0,\delta]\ \} \subset \mathrm{co}\{A_i\}_{i=1}^K$$

To determine the stage cost $\mathcal{L}$, define the polytope

$$S(x,u) = \mathrm{co}\{A_i\}_{i=1}^K\,(x + Bu)$$

and, given a set $\mathcal{Z}$ and a terminal constraint set $\mathcal{X}_f \subset \mathcal{Z}$ that, for some feedback, is invariant for (17)-(18), define the input constraint

$$\mathcal{U}_f(x) = \{u \in \mathcal{U}\ :\ \exp(A\delta)(x + Bu) \in \mathcal{X}_f, S(x,u) \subset \mathcal{Z}\ \}$$

With these definitions, the stage cost $\mathcal{L}$ is given by the distance to the set

$$D = \{(x,u)\ :\ x \in \mathcal{X}_f, u \in \mathcal{U}_f(x)\ \}$$

which is the graph of $\mathcal{U}_f$ on $\mathcal{X}_f$.

Within the above setting, given the current state x_0, a prediction horizon $N \in \mathbb{N}_{>0}$, a terminal constraint set $\mathcal{X}_f$, and a set $\mathcal{Z}$, the problem formulated in [33] consists of minimizing the cost functional

$$\mathcal{J}(x,u) = \sum_{k=0}^{N-1} \mathcal{L}(x(\tau_k), u(\tau_k))$$

whose argument is $k \mapsto (x(\tau_k), u(\tau_k))$, where $x(\tau_k)$ is the evaluation at the N future impulse times τ_k of the solution to (17)-(18) from x_0 resulting from applying $u(\tau_k)$ at the impulse times, where for some $k_0 \in \mathbb{N}$,

$$\tau_k = t_{k+k_0}$$

and

$$u(\tau_k) = u_{k+k_0}$$

for each $k \in \{0, 1, \ldots, N-1\}$. The constraints associated to the minimization problem are: (i) the polytope S remains within the state constraint set $\mathcal{X}$, and (ii) the value of the resulting solution reaches the terminal constraint set $\mathcal{X}_f$ at the end of the prediction horizon N. More precisely, the problem to solve at each periodic impulsive event is as follows:

Problem 7. Given the current state x_0, a prediction horizon $N \in \mathbb{N}_{>0}$, a terminal constraint set $\mathcal{X}_f$, a set $\mathcal{Z}$, and a stage cost $\mathcal{L}$

$$\begin{aligned}
&\min \mathcal{J}(x,u)\\
&\quad \text{subject to}\\
&\qquad x(0) = x_0\\
&\qquad x(\tau_N) \in \mathcal{X}_f\\
&\qquad \left.\begin{array}{ll}
\dot{x}(t) = Ax(t) & \forall t \in (0, N\delta), t \neq \tau_k\\
x(t^+) = x(t) + Bu(t) & \forall t = \tau_k\\
u(\tau_k) \in \mathcal{U} & \\
S(x(\tau_k), u(\tau_k)) \subset \mathcal{X} &
\end{array}\right\} \forall k \in \{0, 1, \ldots, N-1\}
\end{aligned}$$

A minimizer $k \mapsto (x^*(k), u^*(k))$ defines the value of the cost functional $\mathcal{J}^*(x_0) = \mathcal{J}(x^*, u^*)$.

In [33], instead of imposing the conditions involving the impulsive system in Problem 7 that are in the first two lines of the expressions within the brace, conditions on the solution x evaluated at each τ_k are imposed. Such a difference is possible due to the impulses occurring periodically and the continuous-time dynamics being linear. In fact, the values of the solution at the instants τ_k are given by the solution to the discrete-time system

$$x^+ = \exp(A\delta)(x + Bu)$$

from $x(0) = x_0$ and under the effect of the input equal to $u(\tau_k)$. The stability notion used therein only requires closeness and convergence of the values of the solution at the instants τ_k, which the authors refer to as a weak property. Following such a discretization approach, it is shown in [33] that Problem 7 can be formulated as a convex quadratic program (when $\mathcal{L}$ is convex). The MPC strategy in [33] combines features of impulsive systems and of sample-data systems, and is one of the MPC approaches found in the literature that is closest to hybrid dynamical systems, as introduced in the next section.

3 Towards MPC for Hybrid Dynamical Systems

Hybrid dynamical systems are systems with states that can evolve continuously (or flow) and, at times, have abrupt changes (or jump). Such systems may have state components that are continuous valued as well as components that are discrete valued, similar to the discrete-time systems described in Section 2.2. The conditions allowing continuous or discrete changes typically depend on the values of the state, the inputs, and outputs. The development of MPC strategies for such systems is in its infancy, possibly the most related strategy being the one described in Section 2.6 (even though it essentially replaces the flows by $\exp(A\delta)$ due to assuming periodic impulses occurring every δ seconds). On the other hand, research on methods to solve optimal control problems for hybrid dynamical systems has been quite active over the past few decades, and such developments could be exploited to develop MPC strategies for such systems. In particular, maximum principles of optimal control following Pontryagin's maximum principle [29] have been generated for systems with discontinuous right-hand side [34] and for certain classes of hybrid systems [35, 12, 32]. Shown to be useful in several applications [35, 9], these principles establish necessary conditions for optimality in terms of an adjoint function and a Hamiltonian satisfying the "classical" conditions along flow, in addition to matching conditions at jumps.

Numerous frameworks for modeling and analysis of hybrid systems have appeared in the literature. These include the work of Tavernini [36], Michel and Hu [27], Lygeros et al. [22], Aubin et al. [1], and van der Schaft and Schumacher [39], among others. In the framework of [14, 13] the continuous dynamics (or flows) of a hybrid dynamical system are modeled using differential inclusions while the discrete dynamics (or jumps) are captured by difference inclusions. Trajectories to a hybrid dynamical system conveniently use two parameters: an ordinary time parameter $t \in \mathbb{R}_{\geq 0}$, which is incremented continuously as flows occur, and a discrete time parameter $j \in \mathbb{N}$, which is incremented at unitary steps when jumps occur. The conditions determining whether a trajectory to a hybrid system should flow or jump are captured by subsets of the state space and input space. In simple terms, given an input $(t, j) \mapsto u(t, j)$, a trajectory $(t, j) \mapsto x(t, j)$ to a hybrid system satisfies, over intervals of flow,

$$\frac{d}{dt}x(t,j) \in F(x(t,j),u(t,j))$$

when

$$(x(t,j),u(t,j)) \in C$$

and, at jump times,

$$x(t,j+1) \in G(x(t,j),u(t,j))$$

when

$$(x(t,j),u(t,j)) \in D$$

The domain of a trajectory x is denoted $\operatorname{dom} x$, which is a *hybrid time domain* [13]. The above definition of trajectory (or solution) implicitly assumes that $\operatorname{dom} x = \operatorname{dom} u = \operatorname{dom}(x,u)$.

In this way, a hybrid dynamical system is defined by a set C, called the *flow set*, a set-valued map F, called the *flow map*, a set D, called the *jump set*, and a set-valued map G, called the *jump map*. Then, a hybrid system with state x and input u can be written in the compact form

$$\mathcal{H} \ : \ \begin{cases} \dot{x} \ \in F(x,u) & (x,u) \in C \\ x^+ \in G(x,u) & (x,u) \in D \end{cases} \tag{19}$$

The objects defining the data of the hybrid system $\mathcal{H}$ are specified as $\mathcal{H} = (C,F,D,G)$. The state space for x is given by the Euclidean space $\mathbb{R}^n$ while the space for inputs u is given by the set $\mathcal{U}$. The set $C \subset \mathbb{R}^n \times \mathcal{U}$ defines the set of points in $\mathbb{R}^n \times \mathcal{U}$ in which flows are possible according to the differential inclusion defined by the flow map $F : C \rightrightarrows \mathbb{R}^n$. The set $D \subset \mathbb{R}^n \times \mathcal{U}$ defines the set of points in $\mathbb{R}^n \times \mathcal{U}$ from where jumps are possible according to the difference inclusion defined by the set-valued map $G : D \rightrightarrows \mathbb{R}^n$.

Given the current value of the state x_0, and the amount of flow time T and the number of jumps J to predict forward in time, which define a *hybrid prediction horizon* (T,J), an MPC strategy will need to compute trajectories of (19) over the window of hybrid time $[0,T] \times \{0,1,\ldots,J\}$ for all possibly allowed inputs. The fact that different inputs may be applied from the current state x_0 suggests that there may be multiple possible trajectories of (19) from such a point. While this feature is already present in the receding horizon approaches in [16, 30, 5, 24, 6, 23], the hybrid case further adds nonuniqueness due to the potential nonuniqueness of solutions to (19), in particular, due to overlaps between the flow and the jump sets. To deal with nonuniqueness, one would need a set-valued model for prediction that includes all possible predicted hybrid trajectories (and their associated inputs) from x_0 and over $[0,T] \times \{0,1,\ldots,J\}$.

An appropriate cost functional for an MPC strategy for (19), defined over the prediction horizon (T,J), may take the form

$$\mathcal{J}(x,u) := \int_{t:(t,j)\in \mathrm{dom}(x,u), 0\leq t\leq T} \mathcal{L}_c(x(t,j),u(t,j))dt + \sum_{j:(t,j)\in \mathrm{dom}(x,u), 0<j\leq J} \mathcal{L}_d(x(t_j,j),u(t_j,j)) + \mathcal{F}(x(T,J)) \tag{20}$$

where $t_1, t_2, \ldots, t_j, \ldots$ are the jump times of (x,u). The first two arguments of J correspond to a solution to (19) from $x_0 = x(0,0)$. The function $\mathcal{L}_c$ captures the stage cost of flowing and $\mathcal{L}_d$ captures the stage cost of jumping relative to desired subsets of the state space and the input space, respectively. The function $\mathcal{F}$ defines the terminal cost. The key challenge is in establishing conditions such that the value function, which at every point x_0 is given by

$$\mathcal{J}^\star(x_0) := \mathcal{J}(x_\star, u_\star)$$

with $(x_\star, u_\star)$ being minimizers of $\mathcal{J}$ from x_0, certifies the desired asymptotic stability property by guaranteeing that the stage cost approaches zero.

The goal of any MPC strategy for (19) would certainly be to minimize the cost functional $\mathcal{J}$ in (20) over the finite-time hybrid horizon $[0,T] \times \{0,1,\ldots,J\}$ defined by the hybrid prediction horizon (T,J). Given the current value of the state x_0, a potential form of this control law would be

$$\kappa_c(x_0) := u_\star \tag{21}$$

where the choice of the function $u_\star$ is updated when a timer τ_c reaches the hybrid control horizon $N_c + T_c \leq T + J$, and the dynamics of τ_c are as follows:

$$\dot{\tau}_c = 1$$

when $\tau_c \in [0, N_c + T_c]$, and

$$\tau_c^+ = \begin{cases} \tau_c + 1 & \text{when } (x,u) \in D, \tau_c < N_c + T_c \\ 0 & \text{when } (x,u) \notin D, \tau_c \geq N_c + T_c \\ \{\tau_c + 1, 0\} & \text{otherwise} \end{cases}$$

when $(x,u) \notin D$ or $\tau_c \geq N_c + T_c$. These dynamics enforce that the timer increases during flows, so as to count ordinary time, and that at every jump of the hybrid dynamical system (19), the counter is incremented by one (this is in the first entry of difference equation for τ_c), while when the timer has counted at most $N_c + T_c$ seconds of flow and $N_c + T_c$ jumps, is reset to zero (this is the second entry in τ_c^+ – the last entry is when both events can occur). For the current value of the state x_0, the function $u_\star$ used for feedback could be chosen so that

$$u_\star \in \underset{u \,:\, (x,u)\in\mathcal{S}(x_0) \text{ subject to Problem } H}{\arg\min} \mathcal{J}(x,u) \tag{22}$$

which is then applied to the hybrid system over the hybrid horizon with length given by T seconds of flow and J jumps from the current time (t', j'). Above, $\mathcal{S}(x_0)$ denotes the set of state/input pairs (x, u) that satisfy the dynamics of $\mathcal{H}$ and also the conditions in the MPC strategy, which is denoted as Problem H and part of ongoing research efforts is to formally define it.

It should be pointed out that, for purely continuous-time or discrete-time systems, it is not generally known if the controllers designed to satisfy the necessary conditions for optimality imposed by Pontryagin-like maximum principles or Bellman-like approaches confer a margin of robustness to perturbations of the closed loop. In fact, it is well know that discontinuous controllers obtained from solving optimal control laws may not be robust to small perturbations [17]; see also [31]. This difficulty motivates the generation of hybrid control strategies with prediction that guarantee optimality and robustness simultaneously.

For general nonlinear systems, continuity of the state-feedback law plays a key role in the establishment of robustness of the induced asymptotic stability property [21, 37]. Early results establishing that discontinuities in the feedback can lead to a closed-loop system with zero margin of robustness appeared in books by Filippov [11] and Krasovskii [18]; see also [17] for an insightful relationship between solution concepts to nonsmooth systems. Control laws (both open-loop and closed-loop) solving optimal control problems may not be continuous, which may indicate a lack of robustness when applied to the system to control. Such lack of robustness may also be present in receding horizon controllers. In particular, when the associated optimization problem involves state constraints or terminal constraints, and the optimization horizon is small, the asymptotic stability of the closed-loop system may have absolutely no robustness: arbitrarily small disturbances may keep the state away from the desired set [15]. On the bright side, results in [13] indicate that, for the case of no inputs, mild properties of the data of (19) lead to an upper semicontinuous dependence of the solutions with respect to initial conditions, which, in turn, guarantees that asymptotically stable compact sets for $\mathcal{H}$ (without inputs) are robust to small perturbations.

4 Further Reading

- Discrete-time MPC with hybrid flavor: [3, 2, 4, 38, 19];
- Continuous-discrete MPC with hybrid flavor: [7, 23, 10, 25, 28, 33];
- Hybrid dynamical systems: [14, 13].
- Software tools for modeling and some MPC problems with hybrid flavor:
  - Multi-Parametric Toolbox (MPT) 3
    `http://control.ee.ethz.ch/~mpt`

– The Hybrid Toolbox
  `http://cse.lab.imtlucca.it/~bemporad/hybrid/toolbox`

References

1. J.-P. Aubin, J. Lygeros, M. Quincampoix, S. S. Sastry, and N. Seube. Impulse differential inclusions: a viability approach to hybrid systems. *IEEE Transactions on Automatic Control*, 47(1):2–20, 2002.
2. A. Bemporad, F.. Borrelli, and M. Morari. Optimal controllers for hybrid systems: Stability and piecewise linear explicit form. In *Proceedings of the 39th IEEE Conference on Decision and Control, 2000*, volume 2, pages 1810–1815. IEEE, 2000.
3. A. Bemporad, F. Borrelli, and M. Morari. Piecewise linear optimal controllers for hybrid systems. In *Proceedings of the 2000 American Control Conference*, volume 2, pages 1190–1194. IEEE, 2000.
4. A. Bemporad, W. M. H. P. Heemels, and B. De Schutter. On hybrid systems and closed-loop MPC systems. *IEEE Transactions on Automatic Control*, 47(5):863–869, 2002.
5. F. Borrelli, A. Bemporad, and M. Morari. *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.
6. H. Chen and F. Allgöwer. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. In *European Control Conference*, pages 1421–1426. IEEE, 1997.
7. H. Chen and F. Allgöwer. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34(10):1205–1217, 1998.
8. P. D. Christofides and N. El-Farra. *Control of nonlinear and hybrid process systems: Designs for uncertainty, constraints and time-delays*, volume 324. Springer Science & Business Media, 2005.
9. C. D'Apice, M. Garavello, R. Manzo, and B. Piccoli. Hybrid optimal control: case study of a car with gears. *International Journal of Control*, 76:1272–1284, 2003.
10. N. H. El-Farra, P. Mhaskar, and P. D. Christofides. Hybrid predictive control of nonlinear systems: method and applications to chemical processes. *International Journal of Robust and Nonlinear Control*, 14(2):199–225, 2004.
11. A.F. Filippov. *Differential Equations with Discontinuous Right-Hand Sides*. Kluwer, 1988.
12. M. Garavello and B. Piccoli. Hybrid necessary principle. *SIAM J. Control Optim.*, 43(5):1867–1887, 2005.
13. R. Goebel, R. G. Sanfelice, and A. R. Teel. *Hybrid Dynamical Systems: Modeling, Stability, and Robustness*. Princeton University Press, New Jersey, 2012.
14. R. Goebel, R. G. Sanfelice, and A.R. Teel. Hybrid dynamical systems. *IEEE Control Systems Magazine*, 29(2):28–93, April 2009.
15. G. Grimm, M. J. Messina, S. E. Tuna, and A. R. Teel. Examples when nonlinear model predictive control is nonrobust. *Automatica*, 40(10):1729–1738, 2004.
16. G. Grimm, M. J. Messina, S. E. Tuna, and A. R. Teel. Nominally robust model predictive control with state constraints. *IEEE Transactions on Automatic Control*, 52(10):1856–1870, 2007.
17. O. Hàjek. Discontinuous differential equations I. *Journal of Differential Equations*, 32:149–170, 1979.
18. N.N. Krasovskii and A.I. Subbotin. *Game-Theoretical Control Problems*. Springer-Verlag, 1988.
19. M. Lazar, W. M. P. H. Heemels, and A. Bemporad. Stabilizing model predictive control of hybrid systems. *IEEE Transactions on Automatic Control*, 51(11):1813–1818, 2006.

20. Y. Lin and E. D. Sontag. A universal formula for stabilization with bounded controls. *Systems & Control Letters*, 16(6):393–397, 1991.
21. Y. Lin, E. D. Sontag, and Y. Wang. A smooth converse Lyapunov theorem for robust stability. *SIAM J. Control and Optimization*, 34(1):124–160, 1996.
22. J. Lygeros, K.H. Johansson, S.N. Simić, J. Zhang, and S. S. Sastry. Dynamical properties of hybrid automata. *IEEE Transactions on Automatic Control*, 48(1):2–17, 2003.
23. L. Magni and R. Scattolini. Model predictive control of continuous-time nonlinear systems with piecewise constant control. *IEEE Transactions on Automatic Control*, 49(6):900–906, 2004.
24. D. Q. Mayne and H. Michalska. Receding horizon control of nonlinear systems. *IEEE Transactions on Automatic Control*, 35(7):814–824, 1990.
25. P. Mhaskar, N.H. El-Farrah, and P.D. Christofides. Robust hybrid predictive control of nonlinear systems. *Automatica*, 41(2):209–217, February 2005.
26. H. Michalska and D.Q. Mayne. Robust receding horizon control of constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 38(11):1623–1633, November 1993.
27. A.N. Michel and B. Hu. Towards a stability theory of general hybrid dynamical systems. *Automatica*, 35(3):371–384, 1999.
28. D. Nešić and L. Grüne. A receding horizon control approach to sampled-data implementation of continuous-time controllers. *Systems & Control Letters*, 55(8):660–672, 2006.
29. L. S. Pontryagin, V. G. Boltyanskij, R. V. Gamkrelidze, and E. F. Mishchenko. *The mathematical theory of optimal processes.* Wiley, 1962.
30. J. B. Rawlings and D. Q. Mayne. *Model predictive control: Theory and design.* Nob Hill Pub., 2009.
31. R. G. Sanfelice, R. Goebel, and A.R. Teel. Generalized solutions to hybrid dynamical systems. *ESAIM: Control, Optimisation and Calculus of Variations*, 14(4):699–724, 2008.
32. M. S. Shaikh and P. E. Caines. On the hybrid optimal control problem: Theory and algorithms. *IEEE Transactions on Automatic Control*, 52:1587–1603, 2007.
33. P. Sopasakis, P. Patrinos, H. Sarimveis, and A. Bemporad. Model predictive control for linear impulsive systems. *IEEE Transactions on Automatic Control*, 60(8):2277–2282, 2015.
34. H. J. Sussmann. Some recent results on the maximum principle of optimal control theory. *Systems and Control in the Twenty-First Century*, pages 351–372, 1997.
35. H. J. Sussmann. A maximum principle for hybrid optimal control problems. In *Proc. 38th IEEE Conference on Decision and Control*, pages 425–430, 1999.
36. L. Tavernini. Differential automata and their discrete simulators. *Nonlinear Analysis, Theory, Methods & Applications*, 11(6):665–683, 1987.
37. A.R. Teel and L. Praly. A smooth Lyapunov function from a class-$\mathcal{KL}$ estimate involving two positive semidefinite functions. *ESAIM: Control, Optimisation and Calculus of Variations*, 5:313–367, 2000.
38. S. E. Tuna, R. G. Sanfelice, M. J. Messina, and A. R. Teel. Hybrid MPC: Open-minded but not easily swayed. In *Assessment and Future Directions of Nonlinear Model Predictive Control*, volume Lecture Notes in Control and Information Sciences 358, pages 17–34. Springer Berlin / Heidelberg, 2007.
39. A. van der Schaft and H. Schumacher. *An Introduction to Hybrid Dynamical Systems.* Lecture Notes in Control and Information Sciences, Springer, 2000.