University of Pennsylvania, ESE 6190

Model Predictive Control

Chapter 12: Hybrid MPC

Prof. Manfred Morari

Spring 2023

Coauthors: Prof. Francesco Borrelli, UC Berkeley
Prof. Colin Jones, EPFL

F. Borrelli, A. Bemporad, and M. Morari, Predictive Control for Linear and Hybrid Systems, Cambridge University Press, 2017. [Ch. 16, 17].

Outline

1. Modeling of Hybrid Systems

2. Optimal Control of Hybrid Systems

3. Model Predictive Control of Hybrid Systems

4. Explicit MPC of Hybrid Systems

Outline

1. Modeling of Hybrid Systems

2. Optimal Control of Hybrid Systems

3. Model Predictive Control of Hybrid Systems

4. Explicit MPC of Hybrid Systems

Outline

1. Modeling of Hybrid Systems

    Introduction

    Examples of Hybrid Systems

    Piecewise Affine (PWA) Systems

    Mixed Logical Dynamical (MLD) Hybrid Model

Introduction

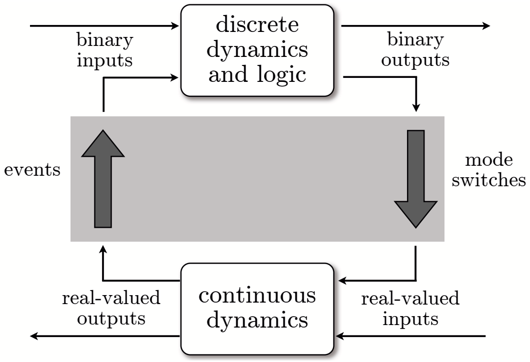
Up to this point: Discrete-time linear systems with linear constraints.

We now consider MPC for systems with

1. **Continuous dynamics**: described by one or more difference (or differential) equations; states are continuous-valued.
2. **Discrete events**: state variables assume **discrete** values, e.g.
    - binary digits $\{0, 1\}$,
    - $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}, \ldots$
    - finite set of symbols

Hybrid systems: Dynamical systems whose state evolution depends on an interaction between continuous dynamics and discrete events.

Introduction



Hybrid systems: Logic-based discrete dynamics and continuous dynamics interact through events and mode switches

Outline

1. Modeling of Hybrid Systems

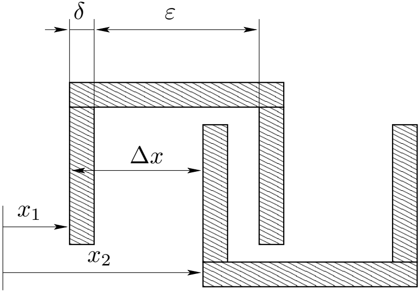
    Introduction

    Examples of Hybrid Systems

    Piecewise Affine (PWA) Systems

    Mixed Logical Dynamical (MLD) Hybrid Model

Mechanical System with Backlash

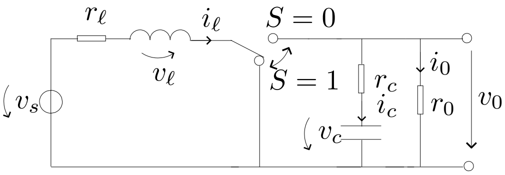


- **Continuous dynamics**: states x_1, x_2, $\dot{x}_1$, $\dot{x}_2$.
- **Discrete events**:
  a) *"contact mode"* $\Rightarrow$ mechanical parts are in contact and the force is transmitted. Condition:

$$[(\Delta x = \delta) \wedge (\dot{x}_1 > \dot{x}_2)] \ \bigvee \ [(\Delta x = \varepsilon) \wedge (\dot{x}_2 > \dot{x}_1)]$$

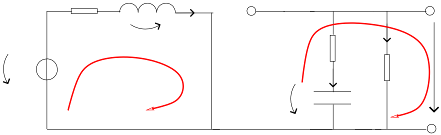
  b) *"backlash mode"* $\Rightarrow$ mechanical parts are not in contact

DCDC Converter

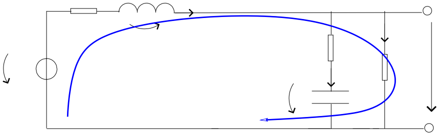


- **Continuous dynamics**: states v_ℓ, i_ℓ, v_c, i_c, v_0, i_0
- **Discrete events**: $S = 0$, $S = 1$

Mode 1 ($S = 1$)



Mode 2 ($S = 0$)



Outline

1. Modeling of Hybrid Systems

    Introduction

    Examples of Hybrid Systems

    Piecewise Affine (PWA) Systems

    Mixed Logical Dynamical (MLD) Hybrid Model

Piecewise Affine (PWA) Systems

PWA systems are defined by:

- **affine dynamics and output** in each region:

$$\left\{\begin{array}{rcl} x(t+1) & = & A_i x(t) + B_i u(t) + f_i \\ y(t) & = & C_i x(t) + D_i u(t) + g_i \end{array}\right\} \text{ if } (x(t), u(t)) \in \mathcal{X}_{i(t)}$$

- **polyhedral partition** of the (x, u)-space:

$$\{\mathcal{X}_i\}_{i=1}^{s} := \{x, u \mid H_i x + J_i u \leq K_i\}$$

with $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$

Physical constraints on $x(t)$ and $u(t)$ are defined by polyhedra $\mathcal{X}_i$

Piecewise Affine (PWA) Systems

Examples:

- *linearization* of a non-linear system at different operating point $\Rightarrow$ useful as an approximation tool
- *closed-loop MPC system* for linear constrained systems
- When the mode i is an exogenous variable, the partition disappears and we refer to the system as a **Switched Affine System** (SAS)

Definition: Well-Posedness

Let P be a PWA system and let $\mathcal{X} = \cup_{i=1}^{s} \mathcal{X}_i \subseteq \mathbb{R}^{n+m}$ be the polyhedral partition associated with it. System P is called **well-posed** if for all pairs $(x(t), u(t)) \in \mathcal{X}$ *there exists only one* index $i(t)$ satisfying the membership condition.

Binary States, Inputs, and Outputs

Remark: In the previous example, the PWA system has only continuous states and inputs.

We will formulate PWA systems including binary state and inputs by treating 0–1 binary variables as:

- **Numbers**, over which arithmetic operations are defined,
- **Boolean variables**, over which Boolean functions are defined.

We will use the notation $x = \left[\begin{smallmatrix} x_c \\ x_\ell \end{smallmatrix}\right] \in \mathbb{R}^{n_c} \times \{0,1\}^{n_\ell}$, $n := n_c + n_\ell$; $y \in \mathbb{R}^{p_c} \times \{0,1\}^{p_\ell}$, $p := p_c + p_\ell$; $u \in \mathbb{R}^{m_c} \times \{0,1\}^{m_\ell}$, $m := m_c + m_\ell$.

Boolean Algebra: Basic Definitions and Notation

- **Boolean variable:** A variable δ is a Boolean variable if $\delta \in \{0, 1\}$, where "$\delta = 0$" means "false", "$\delta = 1$" means "true".

- **A Boolean expression** is obtained by combining Boolean variables through the logic operators $\neg$ (not), $\vee$ (or), $\wedge$ (and), $\leftarrow$ (implied by), $\rightarrow$ (implies), and $\leftrightarrow$ (iff).

- **A Boolean function** $f : \{0, 1\}^{n-1} \mapsto \{0, 1\}$ is used to define a Boolean variable δ_n as a logic function of other variables $\delta_1, \dots, \delta_{n-1}$:

$$\delta_n = f(\delta_1, \delta_2, \dots, \delta_{n-1}).$$

Example

$$
\begin{aligned}
x_c(t+1) &= 2x_c(t) + u_c(t) - 3u_\ell(t) \\
x_\ell(t+1) &= x_\ell(t) \wedge u_\ell(t)
\end{aligned}
$$

can be represented in the PWA form

$$
\begin{bmatrix} x_c(t+1) \\ x_\ell(t+1) \end{bmatrix} = \begin{cases}
\begin{bmatrix} 2x_c(t) + u_c(t) \\ 0 \end{bmatrix} & \text{if} \quad x_\ell \leq \frac{1}{2}, u_\ell \leq \frac{1}{2} \\[2ex]
\begin{bmatrix} 2x_c(t) + u_c(t) - 3 \\ 0 \end{bmatrix} & \text{if} \quad x_\ell \leq \frac{1}{2}, u_\ell \geq \frac{1}{2} + \epsilon \\[2ex]
\begin{bmatrix} 2x_c(t) + u_c(t) \\ 0 \end{bmatrix} & \text{if} \quad x_\ell \geq \frac{1}{2} + \epsilon, u_\ell \leq \frac{1}{2} \\[2ex]
\begin{bmatrix} 2x_c(t) + u_c(t) - 3 \\ 1 \end{bmatrix} & \text{if} \quad x_\ell \geq \frac{1}{2} + \epsilon, u_\ell \geq \frac{1}{2} + \epsilon.
\end{cases}
$$

by associating $x_\ell = 0$ with $x_\ell \leq \frac{1}{2}$ and $x_\ell = 1$ with $x_\ell \geq \frac{1}{2} + \epsilon$ for any $0 < \epsilon \leq \frac{1}{2}$.

Outline

1. Modeling of Hybrid Systems

   Introduction

   Examples of Hybrid Systems

   Piecewise Affine (PWA) Systems

   Mixed Logical Dynamical (MLD) Hybrid Model

Mixed Logical Dynamical Systems

Goal: Describe hybrid system in form compatible with optimization software:

- continuous and Boolean variables
- linear equalities and inequalities

Idea: associate to each Boolean variable p_i a binary integer variable δ_i:

$$p_i \Leftrightarrow \{\delta_i = 1\}, \quad \neg p_i \Leftrightarrow \{\delta_i = 0\}$$

and embed them into a set of constraints as **linear integer inequalities**.

Two main steps:

1. Translation of Logic Rules into Linear Integer Inequalities
2. Translation continuous and logical components into Linear Mixed-Integer Relations

Final result: a compact model with linear equalities and inequalities involving real and binary variables

Boolean formulas as Linear Integer Inequalities

Goal

Given a Boolean formula $F(p_1, p_2, \dots, p_n)$ define a polyhedral set P such that a set of binary values $\{\delta_1, \delta_2, \dots, \delta_n\}$ satisfies the Boolean formula F in P

$$F(p_1, p_2, \dots, p_n) \text{ ``TRUE''} \Leftrightarrow A\delta \le B, \quad \delta \in \{0, 1\}^n$$

where: $\{\delta_i = 1\} \Leftrightarrow p_i = \text{TRUE}$.

Analytic Approach

1. Transform $F(p_1, p_2, \ldots, p_n)$ into a **Conjunctive Normal Form (CNF)**:

$$F(p_1, p_2, \ldots, p_n) = \bigwedge_j \left[\bigvee_i p_i \right]$$

2. Translation of a **CNF** into **algebraic inequalities:**

relation	Boolean	linear constraints
AND	$\delta_1 \wedge \delta_2$	$\delta_1 \geq 1$, $\delta_2 \geq 1$ or $\delta_1 + \delta_2 \geq 2$
OR	$\delta_1 \vee \delta_2$	$\delta_1 + \delta_2 \geq 1$
NOT	$\neg\delta_1$	$(1 - \delta_1) \geq 1$ or $\delta_1 = 0$
XOR	$\delta_1 \oplus \delta_2$	$\delta_1 + \delta_2 = 1$
IMPLY	$\delta_1 \rightarrow \delta_2$	$\delta_1 - \delta_2 \leq 0$
IFF	$\delta_1 \leftrightarrow \delta_2$	$\delta_1 - \delta_2 = 0$
ASSIGNMENT $\delta_3 = \delta_1 \wedge \delta_2$	$\delta_3 \leftrightarrow \delta_1 \wedge \delta_2$	$\delta_1 + (1 - \delta_3) \geq 1$ $\delta_2 + (1 - \delta_3) \geq 1$ $(1 - \delta_1) + (1 - \delta_2) + \delta_3 \geq 1$

Analytic Approach. Example

Given

$$F(p_1, p_2, p_3, p_4) \triangleq [(p_1 \wedge p_2) \Rightarrow (p_3 \wedge p_4)]$$

find the equivalent set of linear integer inequalities.

1. remove implication:

$$F(p_1, p_2, p_3, p_4) = \neg(p_1 \wedge p_2) \vee (p_3 \wedge p_4)$$

2. using DeMorgan's theorem, obtain CNF:

$$F(p_1, p_2, p_3, p_4) = (\neg p_1 \vee \neg p_2 \vee p_3) \wedge (\neg p_1 \vee \neg p_2 \vee p_4)$$

3. introduce $[\delta_i = 1] \Leftrightarrow p_i = \text{TRUE}$ and write the inequalities:

$$F(p_1, p_2, p_3, p_4) = \text{TRUE} \Leftrightarrow \left\{ \begin{array}{lcl} \delta_1 + \delta_2 - \delta_3 & \leq & 1 \\ \delta_1 + \delta_2 - \delta_4 & \leq & 1 \\ \delta_{1,2,3,4} \in \{0, 1\} & & \end{array} \right.$$

Linear Inequality As Logic Condition

Definition: Event Generator

An **event generator** is defined by function $f_{EG} : \mathcal{X}_c \times \mathcal{U}_c \times \mathbb{N}_0 \to \mathcal{D}$:

$$\delta_e(t) = f_{EG}(x_c(t), u_c(t), t)$$

Consider the Boolean expression consisting of a Boolean variable p and continuous variable $x \in \mathbb{R}^n$:

$$p \Leftrightarrow a^T x \leq b$$

where $a \in \mathbb{R}^n$, $b \in \mathbb{R}$, $x \in \mathcal{X} \subset \mathbb{R}^n$:

$$\mathcal{X} = \{x \mid a^T x - b \in [m, M]\}$$

Translated to linear inequalitites:

$$\begin{aligned} a^T x - b &\leq M(1-\delta) \\ a^T x - b &> m\delta \end{aligned}$$

Switched Affine Dynamics

Rewrite the state update functions of the SAS as

$$
\begin{aligned}
z_1(t) &= \begin{cases} A_1 x_c(t) + B_1 u_c(t) + f_1, & \text{if } (i(t) = 1), \\ 0, & \text{otherwise,} \end{cases} \\
&\;\;\vdots \\
z_s(t) &= \begin{cases} A_s x_c(t) + B_s u_c(t) + f_s, & \text{if } (i(t) = s), \\ 0, & \text{otherwise,} \end{cases} \\
x_c(t+1) &= \sum_{i=1}^{s} z_i(t),
\end{aligned}
$$

In general, use the "IF-THEN-ELSE" relations

$$
\text{IF } \delta \text{ THEN } z = a_1' x + b_1' u + f_1 \text{ ELSE } z = a_2' x + b_2' u + f_2,
$$

"IF-THEN-ELSE" Relations

$$
\begin{array}{l}
\text{IF} \quad p \qquad \text{THEN} \\
\qquad z_t = a_1^T x_t + b_1 \\
\text{ELSE} \\
\qquad z_t = a_2^T x_t + b_2
\end{array}
\quad \Longleftrightarrow \quad
\begin{array}{rcl}
(m_2 - M_1)\delta + z_t & \leq & a_2^T x_t + b_2 \\
(m_1 - M_2)\delta - z_t & \leq & -a_2^T x_t - b_2 \\
(m_1 - M_2)(1-\delta) + z_t & \leq & a_1^T x_t + b_1 \\
(m_2 - M_1)(1-\delta) - z_t & \leq & -a_1^T x_t - b_1
\end{array}
$$

where $x \in \mathcal{X}$, with

$$\sup_{x \in \mathcal{X}} \; a_i^T x + b_i \leq M_i,$$

$$\inf_{x \in \mathcal{X}} \; a_i^T x + b_i \geq m_i,$$

$$m_2 \neq M_1, \;\; m_1 \neq M_2.$$

Hybrid Modelling - An Example

Consider the following system with constraints: $|x| \leq 10$, $|u| \leq 10$

$$x_{t+1} = \begin{cases} 0.8x_t + u_t & \text{if } x_t \geq 0 \\ -0.8x_t + u_t & \text{if } x_t < 0 \end{cases}$$

1. associate $\{\delta_t = 1\} \Leftrightarrow \{x_t \geq 0\}$

$$\begin{aligned} -m\delta_t &\leq x_t - m \\ -(M+\epsilon)\delta_t &\leq -x_t - \epsilon, \end{aligned}$$

   where: $M = -m = 10$, $\epsilon > 0$.

2. state update equation:

$$x_{t+1} = 1.6\delta_t x_t - 0.8x_t + u_t$$

Hybrid Modelling - An Example

3. introduce variable: $z_t = \delta_t x_t$

$$x_{t+1} = 1.6 z_t - 0.8 x_t + u_t$$

4. constraints on z:

$$\begin{aligned} z_t &\leq M\delta_t \\ z_t &\geq m\delta_t \\ z_t &\leq x_t - m(1-\delta_t) \\ z_t &\geq x_t - M(1-\delta_t) \end{aligned}$$

MLD Hybrid Model

A DHA can be converted into the following MLD model

$$\begin{aligned} x_{t+1} &= Ax_t + B_1 u_t + B_2 \delta_t + B_3 z_t \\ y_t &= Cx_t + D_1 u_t + D_2 \delta_t + D_3 z_t \\ E_2 \delta_t + E_3 z_t &\leq E_4 x_t + E_1 u_t + E_5 \end{aligned}$$

where $x \in \mathbb{R}^{n_c} \times \{0,1\}^{n_\ell}$, $u \in \mathbb{R}^{m_c} \times \{0,1\}^{m_\ell}$ $y \in \mathbb{R}^{p_c} \times \{0,1\}^{p_\ell}$, $\delta \in \{0,1\}^{r_\ell}$ and $z \in \mathbb{R}^{r_c}$.

Physical constraints on continuous variables:

$$\mathcal{C} = \left\{ \begin{bmatrix} x_c \\ u_c \end{bmatrix} \in \mathbb{R}^{n_c + m_c} \;\middle|\; Fx_c + Gu_c \leq H \right\}$$

MLD Hybrid Model. Well-Posedness

- **Well-Posedness:**
  for a given $x = \left[\begin{smallmatrix} x_t \\ u_t \end{smallmatrix}\right] \Rightarrow x_{t+1}$ and y_t uniquely determined

- **Complete Well-Posedness:**
  well-posedness + uniquely determined δ_t and z_t, $\forall \left[\begin{smallmatrix} x_t \\ u_t \end{smallmatrix}\right]$

- Well-posedness is sufficient for the computation of the state and output prediction

- Complete well-posedness allows transformation into equivalent hybrid models

HYbrid System DEscription Language

HYSDEL

- based on DHA
- enables description of discrete-time hybrid systems in a compact way:
  - automata and propositional logic
  - continuous dynamics
  - A/D and D/A conversion
  - definition of constraints
- automatically **generates MLD models** for MATLAB
- freely available from:

`http://control.ee.ethz.ch/~hybrid/hysdel/`

Outline

1. Modeling of Hybrid Systems

2. Optimal Control of Hybrid Systems

3. Model Predictive Control of Hybrid Systems

4. Explicit MPC of Hybrid Systems

Optimal Control for Hybrid Systems: General Formulation

Consider the CFTOC problem:

$$J^*(x(t)) = \min_{U_0} \; p(x_N) + \sum_{k=0}^{N-1} q(x_k, u_k, \delta_k, z_k),$$

$$\text{s.t.} \begin{cases} x_{k+1} = Ax_k + B_1 u_k + B_2 \delta_k + B_3 z_k \\[2ex] E_2 \delta_k + E_3 z_k \leq E_4 x_k + E_1 u_k + E_5 \\[2ex] x_N \in \mathcal{X}_f \\ x_0 = x(t) \end{cases}$$

where $x \in \mathbb{R}^{n_c} \times \{0,1\}^{n_b}$, $u \in \mathbb{R}^{m_c} \times \{0,1\}^{m_b}$, $y \in \mathbb{R}^{p_c} \times \{0,1\}^{p_b}$, $\delta \in \{0,1\}^{r_b}$ and $z \in \mathbb{R}^{r_c}$ and

$$U_0 = \{u_0, u_1, \ldots, u_{N-1}\}$$

Mixed Integer Optimization

Mixed Integer Linear Programming

Consider the following MILP:

$$\begin{array}{ll} \inf_{[z_c, z_b]} & c_c' z_c + c_b' z_b + d \\ \text{subj. to} & G_c z_c + G_b z_b \le W \\ & z_c \in \mathbb{R}^{s_c},\ z_b \in \{0,1\}^{s_b} \end{array}$$

where $z_c \in \mathbb{R}^{s_c}$, $z_b \in \{0,1\}^{s_b}$

- MILP are nonconvex, in general.
- For a fixed $\bar{z}_b$ the MILP becomes a linear program:

$$\begin{array}{ll} \inf_{[z_c, z_b]} & c_c' z_c + (c' b \bar{z}_b + d) \\ \text{subj. to} & G_c z_c \le W - G_b \bar{z}_b \\ & z_c \in \mathbb{R}^{s_c} \end{array}$$

- Brute force approach to solution: enumerating the 2^{s_b} integer values of the variable z_b and solve the corresponding LPs. By comparing the 2^{s_b} optimal costs one can find the optimizer and the optimal cost of the MILP.

Mixed Integer Quadratic Programming

Consider the following MIQP:

$$\begin{array}{ll} \inf_{[z_c, z_b]} & \frac{1}{2} z' H z + q' z + r \\ \text{subj. to} & G_c z_c + G_b z_b \leq W \\ & z_c \in \mathbb{R}^{s_c},\ z_b \in \{0,1\}^{s_b} \\ & z = [z_c, z_b], s = s_c + s_d \end{array}$$

where $H \succeq 0$, $z_c \in \mathbb{R}^{s_c}, z_b \in \{0,1\}^{s_b}$.

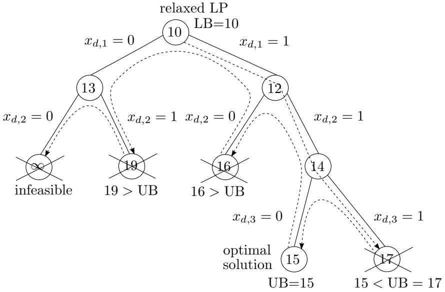
- MIQP are nonconvex, in general.
- For a fixed integer value $\bar{z}_b$ of z_b, the MIQP becomes a quadratic program:

$$\begin{array}{ll} \inf_{[z_c]} & \frac{1}{2} z_c' H_c z_c + q_c' z + k \\ \text{subj. to} & G_c z_c \leq W - G_b \bar{z}_b \\ & z_c \in \mathbb{R}^{s_c} \end{array}$$

- Brute force approach to the solution: enumerating all the 2^{s_b} integer values of the variable z_b and solve the corresponding QPs. By comparing the 2^{s_b} optimal costs one can derive the optimizer and the optimal cost of the MIQP.

Branch and Bound (B&B)

Common solution method for MIPs, based on relaxations of binaries: $\{0, 1\} \rightarrow [0, 1]$.



- Optimal cost of a solution to a modified problem where some binaries are relaxed is a **lower bound** on optimal cost.
- Any feasible solution to original problem is **upper bound** on optimal cost.
- Use bounds to rule out parts of the B&B tree systematically - much more efficient than brute force search.

Outline

1. Modeling of Hybrid Systems

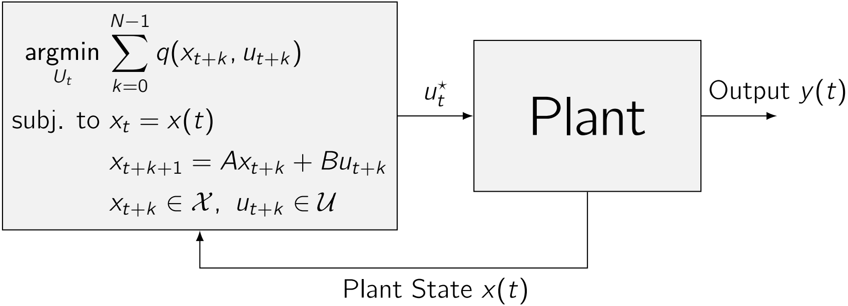
2. Optimal Control of Hybrid Systems

3. Model Predictive Control of Hybrid Systems

4. Explicit MPC of Hybrid Systems

Model Predictive Control of Hybrid Systems

MPC solution: Optimization in the loop



As for linear MPC, at each sample time:

- Measure / estimate current state $x(t)$
- Find the optimal input sequence for the entire planning window N: $U_t^* = \{u_t^*, u_{t+1}^*, \ldots, u_{t+N-1}^*\}$
- Implement only the **first** control action u_t^*
- **Key difference: Requires online solution of an MILP or MIQP**

Outline

1. Modeling of Hybrid Systems

2. Optimal Control of Hybrid Systems

3. Model Predictive Control of Hybrid Systems

4. Explicit MPC of Hybrid Systems

Explicit MPC for Hybrid Systems. Quadratic Case

Theorem

The solution to the CFTOC problem based on an MLD model and with quadratic cost is **time-varying PWA feedback law** of the form:

$$u_t^*(x_t) = F_t^i x_t + G_t^i \quad \text{if } \; x_t \in \mathcal{R}_k^i$$

where $\left\{\mathcal{R}_t^i\right\}_{i=1}^{R_t}$ are regions partitioning the set of feasible states $\mathcal{X}_t^*$ and the closure $\bar{\mathcal{R}}_k^i$ of the sets $\mathcal{R}_k^i$ has the following form:

$$\bar{\mathcal{R}}_k^i := \left\{x : \; x(k)' L(j)_k^i x(k) + M(j)_k^i x(k) \leq N(j)_k^i, \; j = 1, \ldots, n_k^i\right\}, \\ k = 0, \ldots, N-1.$$

(note: quadratic and linear boundaries - not polyhedra!)

Explicit MPC for Hybrid Systems. Quadratic Case

- Denote by $\{v_i\}_{i=1}^{s^N}$ the set of all possible switching sequences over the horizon N
- Fix a certain v_i and constrain the state to switch according to sequence v_i.
- The problem becomes a **CFTOC for a linear time-varying system**. The solution is

$$u^i(x(0)) = \tilde{F}^{i,j} x(0) + \tilde{g}^{i,j}, \quad \forall x(0) \in \mathcal{T}^{i,j}, \quad j = 1, \ldots, N^{r\,i}$$

where $\mathcal{D}^i = \bigcup_{j=1}^{N^{r\,i}} \mathcal{T}^{i,j}$

- The set $\mathcal{X}_0 = \bigcup_{i=1}^{s^N} \mathcal{D}^i$ in general is **not convex**.
- The sets $\mathcal{D}^i$ can, in general, overlap. I.e., **some initial state is feasible for more than one switching sequence**.

Explicit MPC for Hybrid Systems. 1, ∞-norm Case

Theorem

The solution to the CFTOC problem based on an MLD model and with the cost based on norms $\{1, \infty\}$ is a **time-varying PPWA feedback law** of the form:

$$u_t^*(x_t) = F_t^i x_t + G_t^i \quad \text{if } \; x_t \in \mathcal{R}_k^i$$

where $\left\{\mathcal{R}_t^i\right\}_{i=1}^{R_t}$ are polyhedral regions partitioning the set of feasible states $\mathcal{X}_t^*$.

MPC for Hybrid Systems - Complexity

- The complexity strongly depends on the problem structure and the initial setup
- In general:

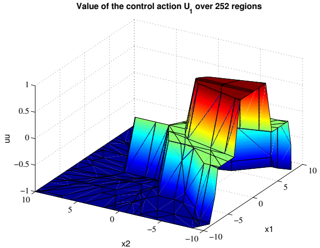
  Mixed-Integer programming is HARD

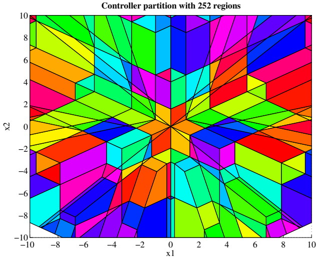
- Efficient general purpose solvers for MILP/MIQP: `CPLEX`, `XPRESS-MP` $\Rightarrow$ based on **Branch-And-Bound**, **Branch-And-Cut** methods + lots of heuristics
- On-line optimization is good for applications allowing large sampling intervals (typically **minutes**), requires expensive hardware and (even more) expensive software
- For very small problems requiring fast sampling rate
  $\Rightarrow$ **explicit solution of the MPC**

Explicit MPC for Hybrid Systems - Example

$$
\begin{cases}
x_{t+1} & = & 0.8 \begin{bmatrix} \cos\alpha_t & -\sin\alpha_t \\ \sin\alpha_t & \cos\alpha_t \end{bmatrix} x_t + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k) \\
\alpha_t & = & \begin{cases} \pi/3 & \text{if } \begin{bmatrix} 1 & 0 \end{bmatrix} x_t \geq 0, \\ -\pi/3 & \text{if } \begin{bmatrix} 1 & 0 \end{bmatrix} x_t < 0 \end{cases} \\
x_t & \in & \begin{bmatrix} -10, & 10 \end{bmatrix} \times \begin{bmatrix} -10, & 10 \end{bmatrix}, \\
u_t & \in & \begin{bmatrix} -1, & 1 \end{bmatrix}
\end{cases}
$$

$$N = 12, P_N = Q_x = I, Q_u = 1, \infty - \text{norm}$$





Summary

- Hybrid systems: mixture of continuous and discrete dynamics
  - Many important systems fall in this class
  - Many tricks involved in modeling - automatic systems available to convert to consistent form

- Optimization problem becomes a mixed-integer linear / quadratic program
  - NP-hard (exponential time to solve)
  - Advanced commercial solvers available

- MPC theory (invariance, stability, etc) applies
  - Computing invariant sets is usually extremely difficult
  - Computing the optimal solution is extremely difficult (sub-optimal ok)