# Model Predictive Control for Linear and Hybrid Systems.
# Hybrid Systems

Francesco Borrelli

Department of Mechanical Engineering,
University of California at Berkeley,
USA

fborrelli@berkeley.edu

April 25, 2011

# Introduction

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

**A step further:** Systems consisting of

1. continuous dynamics : described by one or more difference (or differential) equations, states belong to a continuum
2. discrete events : state variables assume discrete values from a *countable set*, e.g.
   - binary digits $\{0, 1\}$,
   - $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, $\dots$
   - finite set of symbols

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

# Introduction

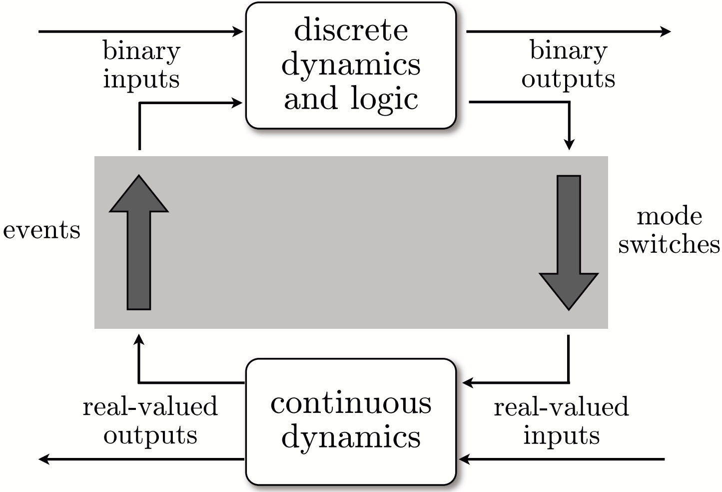
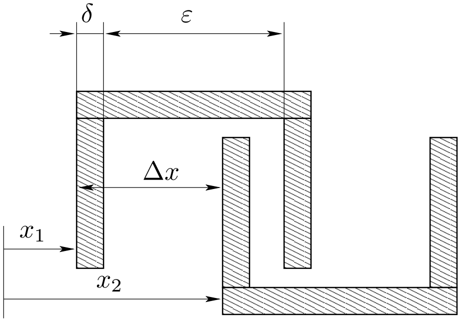



Figure: Hybrid systems. Logic-based discrete dynamics and continuous dynamics interact through events and mode switches

# Hybrid Systems: Examples (I)

**Mechanical system with backlash**



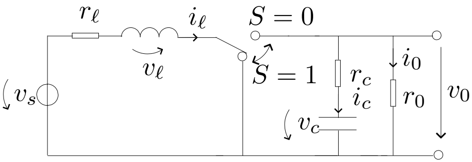

- Continuous dynamics : states $x_1$, $x_2$, $\dot{x}_1$, $\dot{x}_2$.
- Discrete events :
  a) "contact mode" $\Rightarrow$ mechanical parts are in contact and the force is transmitted. Condition:

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

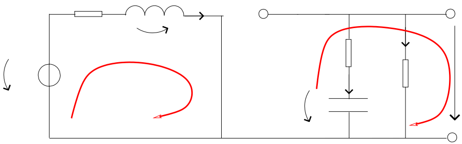
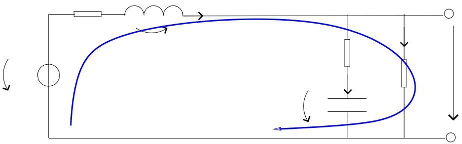
  b) "backlash mode" $\Rightarrow$ mechanical parts are not in contact

# Hybrid Systems: Examples (II)

## DC2DC Converter




- Continuous dynamics : states $v_\ell$, $i_\ell$, $v_c$, $i_c$, $v_0$, $i_0$
- Discrete events : $S = 0$, $S = 1$

| **MODE 1** ($S = 1$) | **MODE 2** ($S = 0$) |
|---|---|
|  |  |

# Hybrid Systems: Examples (III)

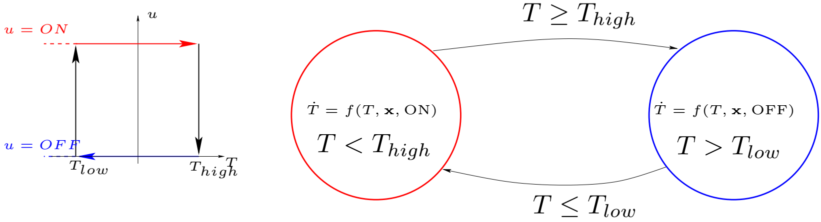
**Temperature Control System**

Evolution of the temperature in a room:

$$\dot{T} = f(T, \mathbf{x}, u)$$

- Continuous states : temperature $T$, other states $\mathbf{x}$
- Discrete input : heating $u \in \{\text{ON}, \text{OFF}\}$

Control system with hysteresis:



# Piecewise Affine (PWA) Systems

PWA are defined by:

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

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

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

$$\boxed{\{\mathcal{X}_i\}_{i=1}^s := \{\left[\begin{smallmatrix} x \\ u \end{smallmatrix}\right] \mid H_i x + J_i u \leq K_i\}}$$

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

**Remark:** 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

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.

# Modeling Discontinuities

Discontinuous systems can be approximated by ***disconnecting the domain***. Example:

$$x(t+1) = \begin{cases} \frac{1}{2}x(t) & \text{if} \quad x(t) \leq 0 \\ 0 & \text{if} \quad x(t) > 0 \end{cases}$$

can be approximated by

$$x(t+1) = \begin{cases} \frac{1}{2}x(t) & \text{if} \quad x(t) \leq 0 \\ 0 & \text{if} \quad x(t) \geq \epsilon \end{cases}$$

where $\epsilon > 0$ is an arbitrarily small number, for instance the machine precision.

**We prefer to assume that in the definition of the PWA dynamics the polyhedral cells $\mathcal{X}_{i(t)}$ are closed sets.**

# Example

$$\begin{cases} x(t+1) &= 0.8 \begin{bmatrix} \cos\alpha(t) & -\sin\alpha(t) \\ \sin\alpha(t) & \cos\alpha(t) \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) \\ y(t) &= \begin{bmatrix} 0 & 1 \end{bmatrix} x(t) \\ \alpha(t) &= \begin{cases} \frac{\pi}{3} & \text{if } [\,1\ 0\,]\,x(t) \geq 0 \\ -\frac{\pi}{3} & \text{if } [\,1\ 0\,]\,x(t) < 0 \end{cases} \end{cases}$$

can be described in PWA form as

$$\begin{cases} x(t+1) &= \begin{cases} 0.4 \begin{bmatrix} 1 & -\sqrt{3} \\ \sqrt{3} & 1 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) \\ \qquad \text{if } \begin{bmatrix} 1 & 0 \end{bmatrix} x(t) \geq 0 \\ \\ 0.4 \begin{bmatrix} 1 & \sqrt{3} \\ -\sqrt{3} & 1 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) \\ \qquad \text{if } \begin{bmatrix} 1 & 0 \end{bmatrix} x(t) \leq -\epsilon \end{cases} \\ y(t) &= \begin{bmatrix} 0 & 1 \end{bmatrix} x(t) \end{cases}$$

for all $x_1 \in (-\infty, -\epsilon] \cup [0, +\infty)$, $x_2 \in \mathbb{R}$, $u \in \mathbb{R}$, and $\epsilon > 0$.

# 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.

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

# Boolean Algebra. Basic Definitions and Notation

- ***Boolean variable***. A variable $\delta$ 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 $\delta_n$ as a logic function of other variables $\delta_1, \dots, \delta_{n-1}$:

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

# Boolean Algebra. Basic Definitions and Notation

- ***Boolean formula***. Given $n$ Boolean variables $\delta_1, \ldots, \delta_n$, a Boolean formula $F$ defines a relation

$$F(\delta_1, \ldots, \delta_n)$$

  that must hold true.

- ***Conjunctive Normal Form (CNF)***. Every Boolean formula $F(\delta_1, \delta_2, \ldots, \delta_n)$ can be rewritten in the CNF

$$\text{(CNF)} \qquad \bigwedge_{j=1}^{m} \left( \bigvee_{i \in P_j} \delta_i \bigvee_{i \in N_j} \sim \delta_i \right)$$
$$N_j, P_j \subseteq \{1, \ldots, n\}, \ \forall j = 1, \ldots, m.$$

# Example

Consider the system

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

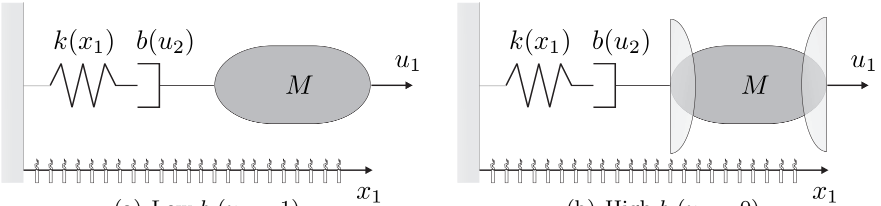
can be represented in the PWA form

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

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 $0 < \epsilon \leq \frac{1}{2}$.

# Example

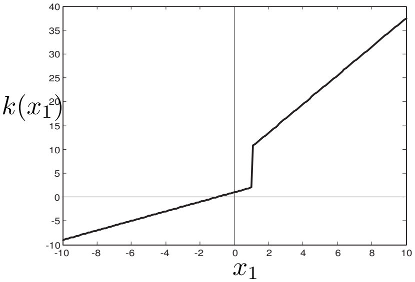
Consider the spring-mass system




(a) Low $b$ ($u_2 = 1$)

(b) High $b$ ($u_2 = 0$)

With the following spring nonlinear characteristic



# Example

The system dynamics can be described in continuous-time as:

$$M\dot{x}_2 = u_1 - k(x_1) - b(u_2)x_2$$

The spring coefficient is

$$k(x_1) = \begin{cases} k_1 x_1 + d_1 \text{ if } x_1 \leq x_m \\ k_2 x_1 + d_2 \text{ if } x_1 > x_m, \end{cases}$$

and the viscous friction coefficient is

$$b(u_2) = \begin{cases} b_1 \text{ if } u_2 = 1 \\ b_2 \text{ if } u_2 = 0. \end{cases}$$

Assume the system description is valid for $-5 \leq x_1, x_2 \leq 5$, and for $-10 \leq u_2 \leq 10$.

# Example

The system has four modes, depending on the binary input and the region of linearity.

$$
x(t+1) = \begin{cases}
\begin{bmatrix} 0.90 & 0.02 \\ -0.02 & -0.00 \end{bmatrix} x(t) + \begin{bmatrix} 0.10 \\ 0.02 \end{bmatrix} u_1(t) + \begin{bmatrix} -0.01 \\ -0.02 \end{bmatrix} \\
\qquad \text{if } x_1(t) \leq 1, u_2(t) \leq 0.5 \\
\begin{bmatrix} 0.90 & 0.02 \\ -0.06 & -0.00 \end{bmatrix} x(t) + \begin{bmatrix} 0.10 \\ 0.02 \end{bmatrix} u_1(t) + \begin{bmatrix} -0.07 \\ -0.15 \end{bmatrix} \\
\qquad \text{if } x_1(t) \geq 1 + \epsilon, u_2(t) \leq 0.5 \\
\begin{bmatrix} 0.90 & 0.38 \\ -0.38 & 0.52 \end{bmatrix} x(t) + \begin{bmatrix} 0.10 \\ 0.38 \end{bmatrix} u_1(t) + \begin{bmatrix} -0.10 \\ -0.38 \end{bmatrix} \\
\qquad \text{if } x_1(t) \leq 1, u_2(t) \geq 0.5 \\
\begin{bmatrix} 0.90 & 0.35 \\ -1.04 & 0.35 \end{bmatrix} x(t) + \begin{bmatrix} 0.10 \\ 0.35 \end{bmatrix} u_1(t) + \begin{bmatrix} -0.75 \\ -2.60 \end{bmatrix} \\
\qquad \text{if } x(t) \geq 1 + \epsilon, u_2(t) \geq 0.5
\end{cases}
$$

for $x_1(t) \in [-5, 1] \cup [1 + \epsilon, 5]$, $x_2(t) \in [-5, 5]$, $u(t) \in [-10, 10]$, and for any arbitrary small $\epsilon > 0$.

# Example

Consider the following SISO system:

$$x_1(t+1) = ax_1(t) + bu(t).$$

A logic state $x_2 \in [0, 1]$ stores the information whether the state of system has ever gone below a certain lower bound $x_{lb}$ or not:

$$x_2(t+1) = x_2(t) \bigvee [x_1(t) \leq x_{lb}],$$

Assume that the input coefficient is a function of the logic state:

$$b = \begin{cases} b_1 \text{ if } x_2 = 0 \\ b_2 \text{ if } x_2 = 1. \end{cases}$$

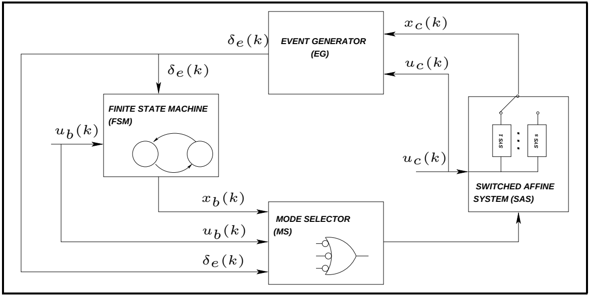
# Example

The system can be described by the PWA model:

$$x(t+1) = \begin{cases} \begin{bmatrix} a & 0 \\ 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} b_2 \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ \qquad \text{if } \begin{bmatrix} 1 & 0 \end{bmatrix} x(t) \leq x_{lb} \\ \\ \begin{bmatrix} a & 0 \\ 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} b_1 \\ 0 \end{bmatrix} u(t) \\ \qquad \text{if } \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} x(t) \geq \begin{bmatrix} x_{lb} + \epsilon \\ -0.5 \end{bmatrix} \\ \\ \begin{bmatrix} a & 0 \\ 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} b_2 \\ 0 \end{bmatrix} u(t) \\ \qquad \text{if } x(t) \geq \begin{bmatrix} x_{lb} + \epsilon \\ 0.5 \end{bmatrix} \end{cases}$$

for $u(t) \in \mathbb{R}$, $x_1(t) \in (-\infty, x_{lb}] \cup [x_{lb} + \epsilon, +\infty)$, $x_2 \in \{0, 1\}$, and for any $\epsilon > 0$.

# Discrete Hybrid Automata (DHA)




Interconnection between:

- switched affine system (SAS) $\Rightarrow$ continuous dynamics
- finite state machine (FSM) $\Rightarrow$ discrete events

Interconnection based on:

- Event Generator (EG)
  - logic signals from the constraints on continuous states and time
  - triggers *mode switching* of the FSM
- Mode Selector (MS) $\Rightarrow$ *selection of an affine subsystem*

# DHA. Switched Affine System

State update equation:

$$\boxed{\begin{array}{rcl} x_c(t+1) & = & A_{i(t)}x_c(t) + B_{i(t)}u_c(t) + f_{i(t)}, \\ y_c(t) & = & C_{i(t)}x_c(t) + D_{i(t)}u_c(t) + g_{i(t)}, \\ & & \quad\quad\quad \text{if } \; i(k) = j \end{array}}$$

- $x_c \in \mathcal{X}_c \subseteq \mathbb{R}^{n_c}$, $u_c \in \mathcal{U}_c \subseteq \mathbb{R}^{m_c}$
- $i(t) \Rightarrow$ dynamic mode of SAS
- $i(t),\ j \in \mathcal{I} = \{1, 2, \ldots, s\}$

# DHA. Mode Selector

***The mode*** $i(t)$ of the SAS is generated by a ***Mode Selector*** function that depends on

1. the FSM state
2. *Discrete events* generated by the continuous variables of the SAS
3. Exogenous discrete inputs
4. Time events

Boolean function $f_{\mathrm{M}} : \{0,1\}^{n_b} \times \{0,1\}^{m_b} \times \{0,1\}^{n_e} \to \mathcal{I}$:

$$\boxed{i(t) = f_M(x_b(t), u_b(t), \delta_e(t))}$$

- $i(k) \in \mathcal{I} \subset \mathbb{N} \Rightarrow$ active mode
- selection of a dynamic mode of the SAS

# DHA. Event Generator

Defined by function $f_{EG} : \mathcal{X}_c \times \mathcal{U}_c \times \mathbb{N}_0 \to \mathcal{D}$:

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

Generates a binary vector $\delta_e(t) \in \{0, 1\}^{n_e}$ of *event conditions* according to the satisfaction of a linear (or affine) condition.

- time events: $\{\delta_e^i = 1\} \Leftrightarrow \{t > t^*\}$
- threshold events: $\{\delta_e^i = 1\} \Leftrightarrow \{a^T x_c(t) + b^T u_c(t) \leq c\}$

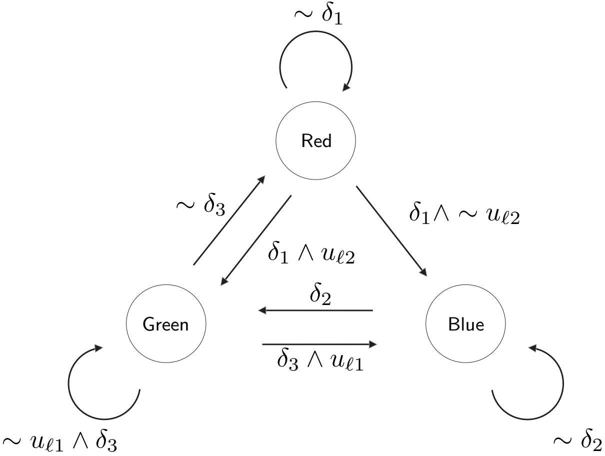
# DHA. Finite State Machine

Discrete time dynamic process:

$$\boxed{x_b(k+1) = f_{FSM}(x_b(k), u_b(k), \delta_e(k))}$$

- $f_{FSM} : \mathcal{X}_b \times \mathcal{U}_b \times \mathcal{D} \to \mathcal{X}_b$ is deterministic logic function .
- $\delta_e \in \mathcal{D} \subseteq \{0, 1\}^{n_e}$ (Boolean event from EG),
- $x_b \in \mathcal{X}_b \subseteq \{0, 1\}^{n_b}$ (Boolean state),
- $u_b \in \mathcal{U}_b \subseteq \{0, 1\}^{m_b}$ (Boolean input),

# Finite State Machine. Example




$u_\ell = [u_{\ell 1} \ u_{\ell 2}]^T$ is the input vector, and $\delta = [\delta_1 \dots \delta_3]^T$ is a vector of signals coming from the event generator.

# Finite State Machine. Example

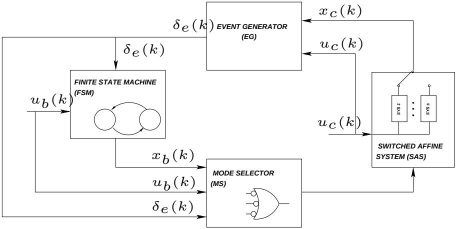
The Boolean state update function (also called *state transition function*) is:

$$x_\ell(t+1) = \begin{cases} \mathsf{Red} \text{ if } ((x_\ell(t) = \mathsf{Green}) \wedge \sim \delta_3) \vee \\ \quad ((x_\ell(t) = \mathsf{Red}) \wedge \sim \delta_1), \\ \mathsf{Green} \text{ if } ((x_\ell(t) = \mathsf{Red}) \wedge \delta_1 \wedge u_{\ell 2}) \vee \\ \quad ((x_\ell(t) = \mathsf{Blue}) \wedge \delta_2) \vee \\ \quad ((x_\ell(t) = \mathsf{Green}) \wedge \sim u_{\ell 1} \wedge \delta_3), \\ \mathsf{Blue} \text{ if } ((x_\ell(t) = \mathsf{Red}) \wedge \delta_1 \wedge \sim u_{\ell 2}) \vee \\ \quad ((x_\ell(t) = \mathsf{Green}) \wedge (\delta_3 \wedge u_{\ell 1})) \vee \\ \quad ((x_\ell(t) = \mathsf{Blue}) \wedge \sim \delta_2)). \end{cases}$$

Code the four states as $\mathsf{Red} = \left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$, $\mathsf{Green} = \left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]$, and $\mathsf{Blue} = \left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]$:

$$\begin{aligned} x_{\ell 1}(t+1) &= (\sim x_{\ell 1} \wedge \sim x_{\ell 2} \wedge \delta_1 \wedge \sim u_{\ell 2}) \vee \\ &\quad (x_{\ell 1} \wedge \sim \delta_2) \vee (x_{\ell 2} \wedge \delta_3 \wedge u_{\ell 1}), \\ x_{\ell 2}(t+1) &= (\sim x_{\ell 1} \wedge \sim x_{\ell 2} \wedge \delta_1 \wedge u_{\ell 2}) \vee \\ &\quad (x_{\ell 1} \wedge \delta_2) \vee (x_{\ell 2} \wedge \delta_3 \wedge \sim u_{\ell 1}), \end{aligned}$$

# Summary




$$
\begin{aligned}
\delta_e(t) &= h(x_c(t), u_c(t), t) \\
i(t) &= \mu(x_\ell(t), u_\ell(t), \delta_e(t)) \\
y_c(t) &= C_{i(t)} x_c(t) + D_{i(t)} u_c(t) + g_{i(t)} \\
y_\ell(t) &= g_\ell(x_\ell(t), u_\ell(t), \delta_e(t)) \\
x_c(t+1) &= A_{i(t)} x_c(t) + B_{i(t)} u_c(t) + f_{i(t)} \\
x_\ell(t+1) &= f_\ell(x_\ell(t), u_\ell(t), \delta_e(t))
\end{aligned}
$$

# Summary

Needed: a mathematical model

- descriptive: captures the behavior of the hybrid system
- simple (enough) for analysis and the prediction of the states

***PWA:***

- mode enumeration explodes

***DHA:***

- good framework for the description of a hybrid systems
- not convenient for the MPC formulation

**Basic problem:** How to describe the interaction between continuous dynamics and propositional logic rules?

# Mixed Logical Dynamical Systems

***Idea:*** associate to each Boolean variable $p_i$ a binary integer variable $\delta_i$:

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

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 DHA 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

## *Aim*

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

$$\boxed{F(p_1, p_2, \ldots, p_n) \;\; \text{“TRUE”} \Leftrightarrow A\delta \leq B, \quad \delta \in \{0,1\}^n}$$

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

# Analytic Approach

1. Transform $F(p_1, p_2, \ldots, p_n)$ into a Conjuctive 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** | $\delta_1 \wedge \delta_2$ | $\delta_1 = 1,\ \delta_2 = 1$ **or** $\delta_1 + \delta_2 \geq 2$ |
| **OR** | $\delta_1 \vee X_2$ | $\delta_1 + \delta_2 \geq 1$ |
| **NOT** | $\sim \delta_1$ | $\delta_1 = 0$ |
| **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**<br>$\delta_3 = \delta_1 \wedge \delta_2$ | $\delta_3 \leftrightarrow \delta_1 \wedge \delta_2$ | $\delta_1 + (1 - \delta_3) \geq 1$<br>$\delta_2 + (1 - \delta_3) \geq 1$<br>$(1 - \delta_1) + (1 - \delta_2) + \delta_3 \geq 1$ |

# 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 $[\delta_i = 1] \Leftrightarrow p_i =$ "TRUE" and write the inequalities:

$$F(p_1, p_2, p_3, p_4) = \text{“TRUE”} \Leftrightarrow \begin{cases} \delta_1 + \delta_2 - \delta_3 & \leq \quad 1 \\ \delta_1 + \delta_2 - \delta_4 & \leq \quad 1 \\ \delta_{1,2,3,4} \in \{0, 1\} & \end{cases}$$

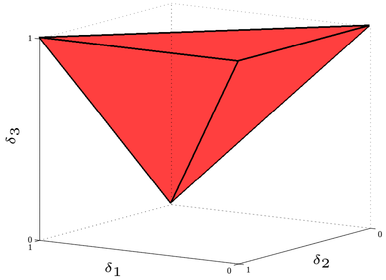
# Geometric Approach

***Key idea***

The polytope $\mathcal{P} = \{\delta \in \{0,1\}^n \mid A\delta \leq B\}$ is the convex hull of the rows of the truth table defining a logic proposition $\Omega(p_i)$.

***Example***. Given $\Omega(p_1, p_2) \triangleq [p_1 \Rightarrow p_2]$. Build the truth table:

| $\delta_1$ | $\delta_2$ | $\delta_3$ |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |




$$hull \left\{ \begin{bmatrix}0\\0\\1\end{bmatrix}, \begin{bmatrix}0\\1\\1\end{bmatrix}, \begin{bmatrix}1\\0\\0\end{bmatrix}, \begin{bmatrix}1\\1\\1\end{bmatrix} \right\} = \left\{ \begin{array}{rcr} \delta_2 - \delta_3 & \leq & 0 \\ \delta_3 & \leq & 1 \\ \delta_1 - \delta_2 + \delta_3 & \leq & 1 \\ -\delta_1 - \delta_3 & \leq & -1 \end{array} \right.$$

# Linear Inequality As Logic Condition

***Event Generator***

Defined by function $f_{EG} : \mathcal{X}_c \times \mathcal{U}_c \times \mathbb{N}_0 \rightarrow \mathcal{D}$:

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

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:

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

# 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

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

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

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

$$\inf_{x \in \mathcal{X}} a_i^T x - b_i \geq m_i.$$

# 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 *disaggregated variable*: $z_t = \delta_t x_t$

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

4. constraints on $z$:

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

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

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 $\begin{bmatrix} x_t^T & u_t^T \end{bmatrix}^T$
  $\Rightarrow x_{t+1}$ and $y_t$ uniquely determined

- **Complete Well-Posedness:**
  well-posedness + uniquely determined $\delta_t$ and $z_t$ $\forall \begin{bmatrix} x_t^T & u_t^T \end{bmatrix}^T$

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

Complete well-posedness : transformation into equivalent hybrid models

# Linear Complementary (LC) Systems

$$
\boxed{
\begin{array}{rcl}
x_{t+1} & = & Ax_t + B_1u_t + B_2w_t, \\
y_t & = & Cx_t + D_1u_t + D_2w_t, \\
v_t & = & E_1x_t + E_2u_t + E_3w_t + g_4 \\
0 \leq v_t & \perp & w_t \geq 0
\end{array}
}
$$

where: $v_t, w_t \in \mathbb{R}^s$ and "$\perp$" denotes elementwise complementarity: $v_i w_i = 0$.

Examples : mechanical systems, electrical circuit with ideal diodes

# Max-Min-Plus-Scaling (MMPS) Systems

MMPS expressions defined by the *grammar*:

$$\boxed{f := x_i \ | \ \alpha \ | \ \max(f_k, f_l) \ | \ \min(f_k, f_l) \ | \ f_k + f_l \ | \ \beta f_k}$$

where $\alpha, \beta \in \mathbb{R}$, $i = \{1, 2, \ldots, n\}$ and $f_k, f_l$ are MMPS expressions.
Example:

$$f = 7x_1 + 0.5x_2 + \max(\min(3x_1 - 2x_2, x3))$$

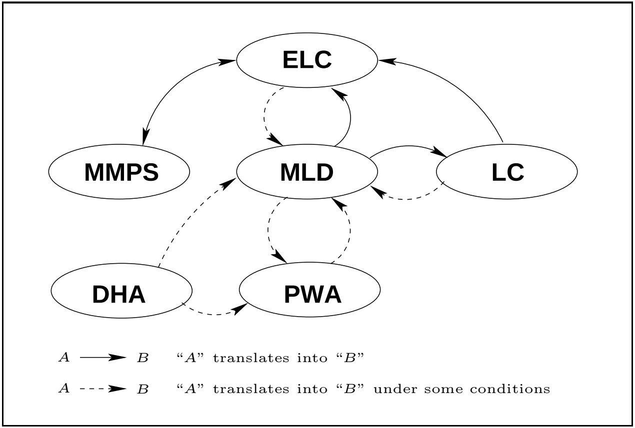
MMPS system:

$$\boxed{\begin{array}{rcl} x_{t+1} & = & \mathcal{M}_x(x_t, u_t, d_t), \\ y_t & = & \mathcal{M}_y(x_t, u_t, d_t), \\ \mathcal{M}_c(x_t, u_t, d_t) & \leq & c \end{array}}$$

where $\mathcal{M}_{x,y,c}(\cdot)$ are MMPS expressions in components of $x_t$, $u_t$ and $d_t$.

# Equivalence of Hybrid Models

"The big picture":




**Modelling & Simulation:** DHA
**Control synthesis:** MLD & PWA

# 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/`

# Mixed Integer Linear Programming

Consider the following MILP

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

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:

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

- 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 Linear Programming

Denote by $J^*$ the optimal value and by $Z^*$ the set of optimizers.

**Case 1.** The MILP solution is unbounded, i.e., $J^* = -\infty$.

**Case 2.** The MILP solution is bounded, i.e., $J^* > -\infty$ and the optimizer is unique. $Z^*$ is a singleton.

**Case 3.** The MILP solution is bounded and there are infinitely many optima corresponding to the same integer value.

**Case 4.** The MILP solution is bounded and there are finitely many optima corresponding to different integer values.

**Case 5.** The union of Case 3 and Case 4.

# 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.

# Mixed Integer Quadratic Programming

Denote by $J^*$ the optimal value and by $Z^*$ the set of optimizers of problem

**Case 1.** The MIQP solution is unbounded, i.e., $J^* = -\infty$. This cannot happen if $H \succ 0$.

**Case 2.** The MIQP solution is bounded, i.e., $J^* > -\infty$ and the optimizer is unique. $Z^*$ is a singleton.

**Case 3.** The MIQP solution is bounded and there are infinitely many optima corresponding to the same integer value. subset of $\mathbb{R}^s$ and an integer number $z_b^*$. This cannot happen if $H \succ 0$.

**Case 4.** The MIQP solution is bounded and there are finitely many optima corresponding to different integer values.

**Case 5.** The union of Case 3 and Case 4.

# MPC for Hybrid Systems. General Formulation

Consider the CFTOC problem:

$$J^*(x_{t|t}) = \min_{U} \ \ell_N(x_{t+N|t}) + \sum_{k=0}^{N-1} \ell(x_{t+k|t}, u_{t+k|t}, \delta_{t+k|t}, z_{t+k|t}),$$

$$\text{s.t.} \begin{cases} x_{t+k+1|t} = Ax_{t+k|t} + B_1 u_{t+k|t} + B_2 \delta_{t+k|t} + B_3 z_{t+k|t} \\ E_2 \delta_{t+k|t} + E_3 z_{t+k|t} \leq E_4 x_{t+k|t} + E_1 u_{t+k|t} + E_5 \\ x_{t+N|t} \in \mathcal{X}_f \end{cases}$$

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

$$U = \begin{pmatrix} u_{t|t}^T & u_{t+1|t}^T & \dots & u_{t+N-1|t}^T \end{pmatrix}^T$$

# MPC for Hybrid Systems. 2-norm Case

- The final and stage costs become

$$
\begin{aligned}
\ell_N(x) &= \|Px\|_2 \\
\ell(x,u,\delta,z) &= \|Q_x x\|_2 + \|Q_u u\|_2 + \|Q_\delta \delta\|_2 + \|Q_z z\|_2
\end{aligned}
$$

- Use the substitution:

$$
\begin{aligned}
x_{t+k|t} = A^k x_{t|t} + \sum_{j=0}^{k-1} \big\{ A^j (B_1 u_{t+k-j-1|t} + \\
+ B_2 \delta_{t+k-j-1|t} + B_3 z_{t+k-j-1|t}) \big\}
\end{aligned}
$$

- Introduce the optimization vector:

$$
\xi = \left[ U^T, \delta_{t|t}, \ldots, \delta_{t+N-1|t}, z_{t|t}, \ldots, z_{t+N-1|t} \right]^T
$$

# MPC for Hybrid Systems. 2-norm Case

- Written in a more compact form, the problem becomes:

$$\begin{aligned} \min_{\xi} \quad & \|H_1\xi\|_2 + \xi^T H_2 x_{t|t} + \|H_3 x_{t|t}\|_2 + c_1^T \xi + c_2 x_{t|t}, \\ \text{s. t.} \quad & G\xi \leq W + S x_{t|t} \end{aligned}$$

- **On-line optimization problem:**
  ***Mixed-Integer Quadratic Program (MIQP)***

# MPC for Hybrid Systems. 2 Norm Case

## Theorem

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

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

*where* $\{\mathcal{R}_t^i\}_{i=1}^{R_t}$ *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:*

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

# MPC for Hybrid Systems. 2 Norm 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 the 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 switching sequences***.

# MPC for Hybrid Systems. 2 Norm Case

- If $\mathcal{T}^{i,j} \cap \mathcal{T}^{l,m} = \emptyset$ for all $l \neq i$, $l = 1, \ldots, s^N$, $m = 1, \ldots, N^{rl}$, then $v_i$ is the only feasible sequence and

$$u^*(x(0)) = \tilde{F}^{i,j} x(0) + \tilde{g}^{i,j}, \quad \forall x \in \mathcal{T}^{i,j}.$$

- If $\mathcal{T}^{i,j}$ intersects one or more polyhedra $\mathcal{T}^{l_1,m_1}$,$\mathcal{T}^{l_2,m_2}$ the states therein are feasible also for the sequences $v_i$, $v_{l_1}$, $v_{l_2}$.
- Compare the value functions $J^*_{v_i}(x(0))$, $J^*_{v_{l_1}}(x(0))$,$J^*_{v_{l_2}}(x(0))$.

$$u^*(x(0)) = \begin{cases} \tilde{F}^{i,j} x(0) + \tilde{g}^{i,j}, \\ \qquad \forall x(0) \in \mathcal{T}^{(i,j),(l,m)} : J^*_{v_i}(x(0)) < J^*_{v_l}(x(0)) \\ \\ \tilde{F}^{l,m} x(0) + \tilde{g}^{l,m}, \\ \qquad \forall x(0) \in \mathcal{T}^{(i,j),(l,m)} : J^*_{v_i}(x(0)) > J^*_{v_l}(x(0)) \\ \\ \begin{cases} \tilde{F}^{i,j} x(0) + \tilde{g}^{i,j} \text{ or} \\ \tilde{F}^{l,m} x(0) + \tilde{g}^{l,m} \end{cases} \\ \qquad \forall x(0) \in \mathcal{T}^{(i,j),(l,m)} : J^*_{v_i}(x(0)) = J^*_{v_l}(x(0)) \end{cases}$$

# MPC for Hybrid Systems. 1, $\infty$-norm Case

- The final and stage costs become

$$\begin{array}{rcl} \ell_N(x) & = & \|Px\|_{1,\infty} \\ \ell(x,u,\delta,z) & = & \|Q_x x\|_{1,\infty} + \|Q_u u\|_{1,\infty} + \|Q_\delta \delta\|_{1,\infty} + \|Q_z z\|_{1,\infty} \end{array}$$

- Introduce decision variables to model $\infty$-norm:

$$\begin{array}{rcl} -\mathbf{1}_n \varepsilon^x_{t+k|t} & \leq & \pm Q_x x_{t+k|t} \\ -\mathbf{1}_m \varepsilon^u_{t+k|t} & \leq & \pm Q_u u_{t+k|t} \\ -\mathbf{1}_{r_b} \varepsilon^\delta_{t+k|t} & \leq & \pm Q_\delta \delta_{t+k|t} \\ -\mathbf{1}_{r_c} \varepsilon^z_{t+k|t} & \leq & \pm Q_z z_{t+k|t} \\ -\mathbf{1}_n \varepsilon^x_{t+N|t} & \leq & \pm P x_{t+N|t}, \end{array}$$

  where $k = 0, \ldots, N-1$, $n = n_c + n_b$, $m = m_c + m_b$.
- Introduce the optimization vector:

$$\xi = \Big[ U^T, \varepsilon^x_{t|t}, \ldots, \varepsilon^x_{t+N|t}, \varepsilon^u_{t|t}, \ldots, \varepsilon^u_{t+N-1|t}, \\ \varepsilon^\delta_{t|t}, \ldots, \varepsilon^\delta_{t+N-1|t}, \varepsilon^z_{t|t}, \ldots, \varepsilon^z_{t+N-1|t} \Big]^T$$

# MPC for Hybrid Systems. 1, $\infty$-norm Case

- Written in a more compact form, the problem becomes:

$$\min_{\xi} \quad \varepsilon^x_{t+N|t} + \sum_{k=0}^{N-1} (\varepsilon^x_{t+N|t} + \varepsilon^\delta_{t+N|t} + \varepsilon^z_{t+k|t} + \varepsilon^u_{t+k|t})$$

$$\text{s. t.} \quad G\xi \leq W + Sx_{t|t}$$

- **On-line optimization problem:**
  ***Mixed-Integer Linear Program (MILP)***

**Receding Horizon Policy**

- given $x_{t|t}$ solve MILP/MIQP and obtain the optimizer vector $\xi^* \Rightarrow$ extract $u^*_{t|t}$ and apply it to the plant
- stability $\Rightarrow$ general stability theory for MPC with receding horizon
  - terminal state constraint $x_{N|t} = 0$
  - $x_{N|t}$ inside an invariant set around the origin

# MPC for Hybrid Systems. 1, $\infty$-norm Case

## Theorem

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

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

*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 + a lot of heuristics
- on-line optimization is good for applications allowing large sampling intervals (tipically minutes ), requires expensive hardware and (even more) expensive software
- for problems requiring fast sampling rate $\Rightarrow$ explicit solution of the MPC

# MPC for Hybrid Systems - Example

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

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