کنترل پیش بین
Model Predictive Control

ارائه کننده: امیرحسین نیکوفرد
مهندسی برق و کامپیوتر دانشگاه خواجه نصیر

Hybrid MPC

- Modeling of Hybrid Systems
  - Introduction
  - Examples of Hybrid Systems
  - Piecewise Affine (PWA) Systems
  - Mixed Logical Dynamical (MLD) Hybrid Model
- Optimal Control of Hybrid Systems
- Model Predictive Control of Hybrid Systems
- MPC of Hybrid Systems Examples

Introduction

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

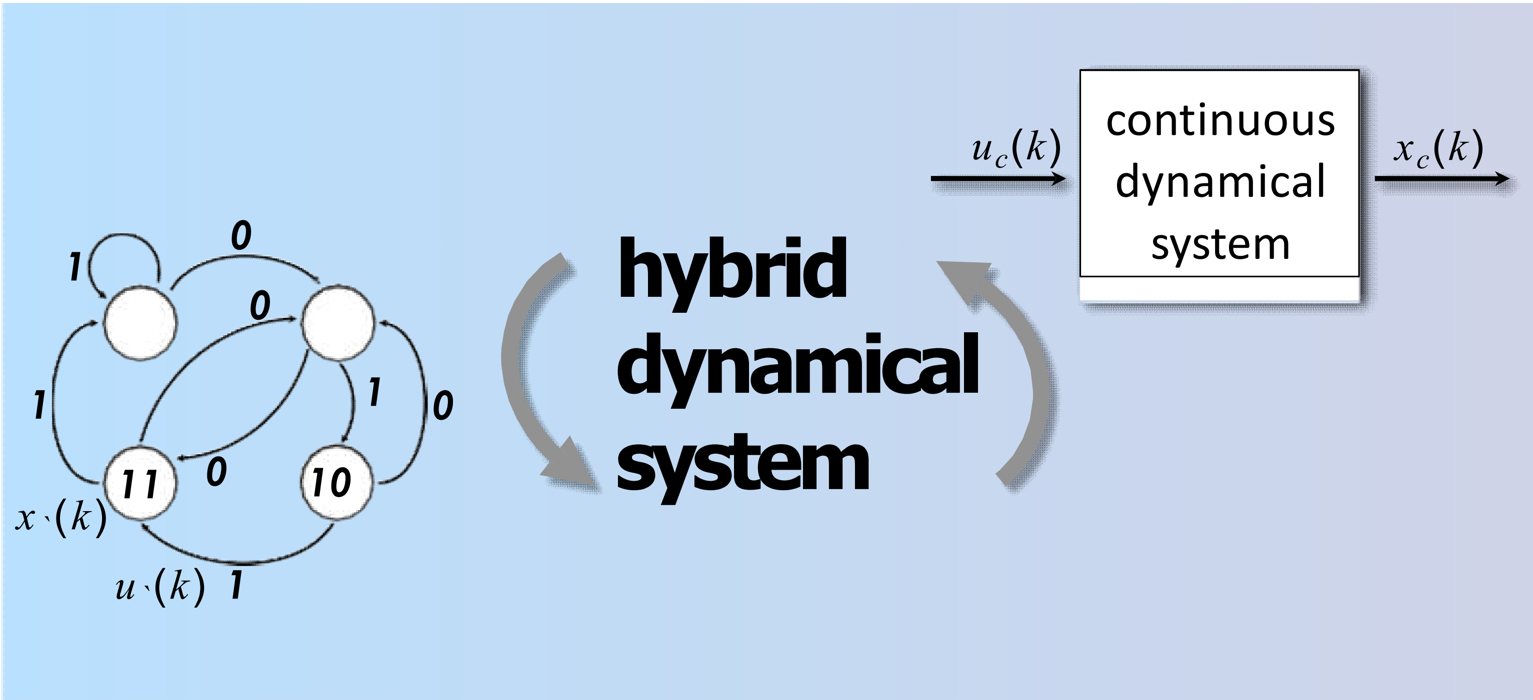
We now consider MPC for systems with

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},
- N, Z, Q, . . .
- finite set of symbols

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

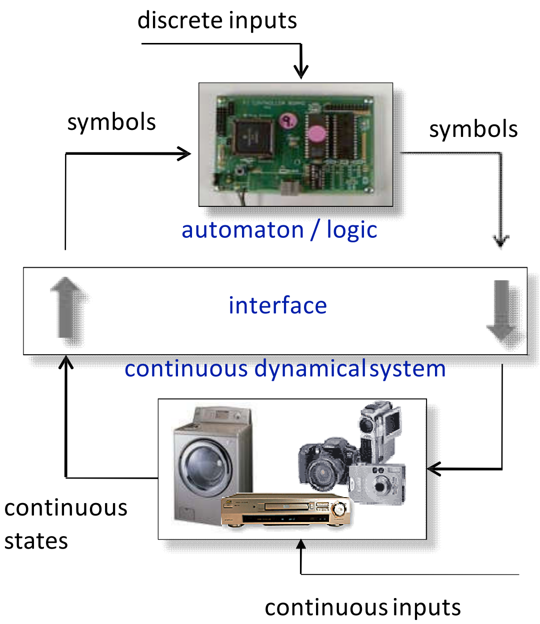
Hybrid dynamical systems



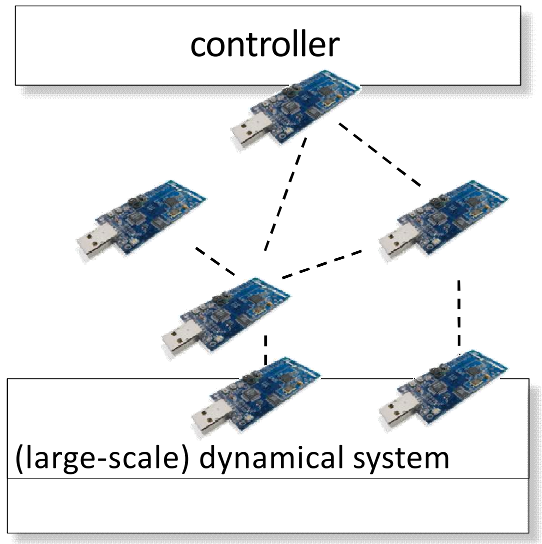
- Variables are **binary-valued**
  $x_\ell \in \{0,\ 1\}^{n_\ell},\ u_\ell \in \{0,\ 1\}^{m_\ell}$
- Dynamics = **finite state machine**
- **Logic constraints**

- Variables are **real-valued**
  $x_c \in \mathbb{R}^{n_c},\ u_c \in \mathbb{R}^{m_c}$
- **Difference/differential equations**
- **Linear inequality** constraints

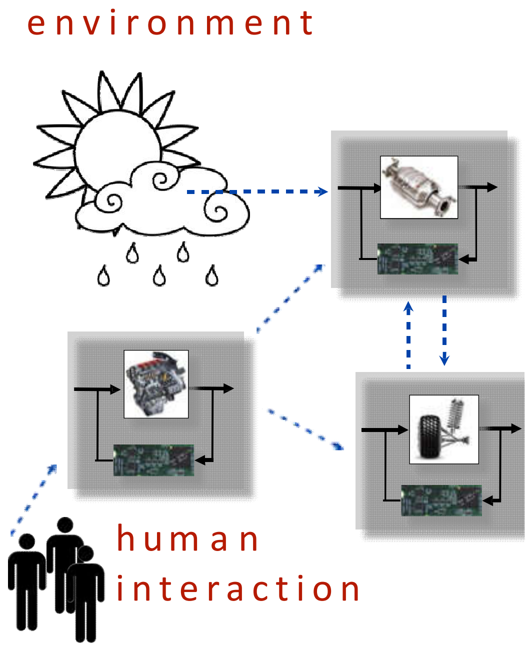
Technological push for studying hybrid systems



embedded systems



networked control systems



cyber-physical systems

>1995 >2005 >2010

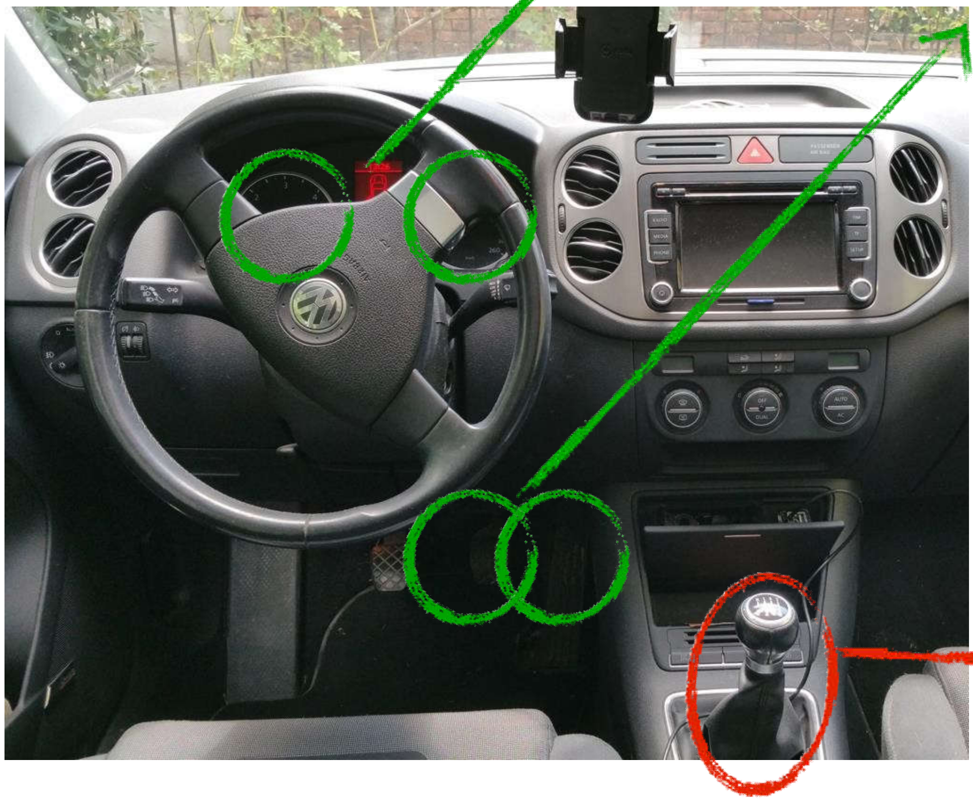
Hybrid MPC

- Modeling of Hybrid Systems
  - Introduction
  - Examples of Hybrid Systems
  - Piecewise Affine (PWA) Systems
  - Mixed Logical Dynamical (MLD) Hybrid Model
- Optimal Control of Hybrid Systems
- Model Predictive Control of Hybrid Systems
- MPC of Hybrid Systems Examples

Examples of Hybrid Systems

- Vehicle

continuous dynamical variables (speed, torque, ...)



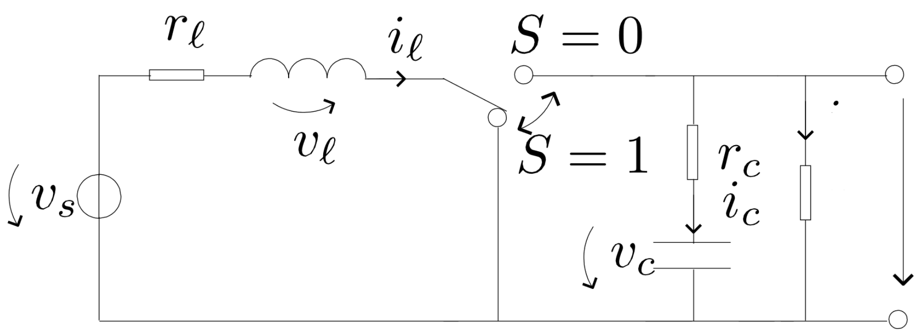
continuous commands (brake & gas pedal)

+

discrete command (R,N,1,2,3,4,5)

Examples of Hybrid Systems

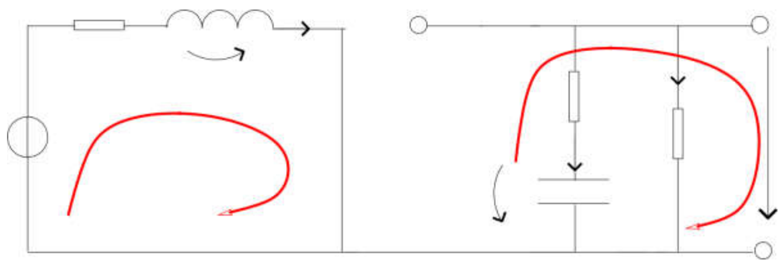
- DC/DC Converter



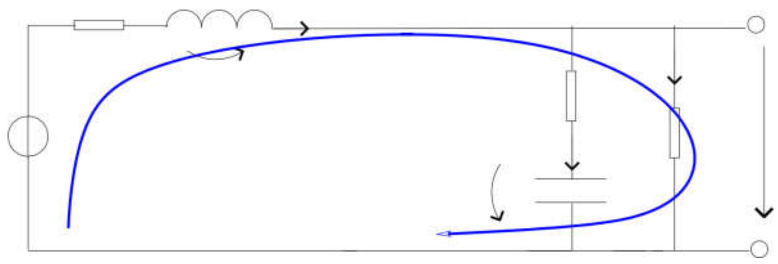
❑ **Continuous dynamics**: states v_l, i_l, v_c, i_c, v_0, i_0

❑ **Discrete events**: $S=0$, $S=1$

Mode 1 (S=1)



Mode 2 (S=0)



Hybrid MPC

- Modeling of Hybrid Systems
  - Introduction
  - Examples of Hybrid Systems
  - Piecewise Affine (PWA) Systems
  - Mixed Logical Dynamical (MLD) Hybrid Model
- Optimal Control of Hybrid Systems
- Model Predictive Control of Hybrid Systems
- MPC of Hybrid Systems Examples

Key requirements for hybrid models

- **Descriptive** enough to capture the behavior of the system
  - **continuous** dynamics (physical systems)
  - **logic** components (switches, automata)
  - **interconnection** between logic and dynamics

- **Simple** enough for solving analysis and synthesis problems

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$

linear hybrid systems

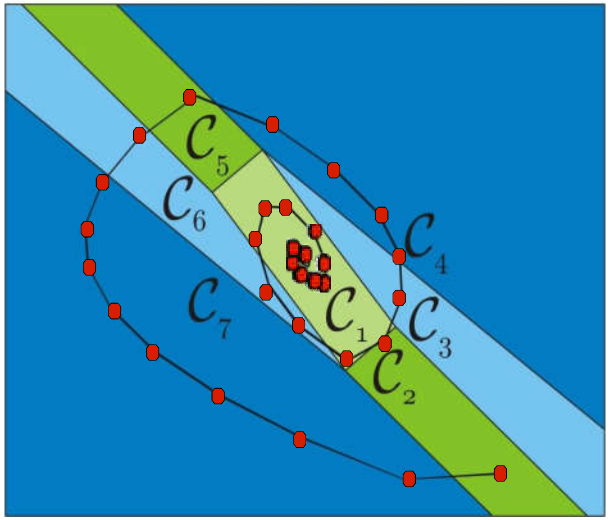
$$\begin{aligned} \dot{x} &= f(x, u, t) \\ y &= g(x, u, t) \end{aligned}$$

"Perfection is achieved not when there is nothing more to add, but when there is nothing left to take away."

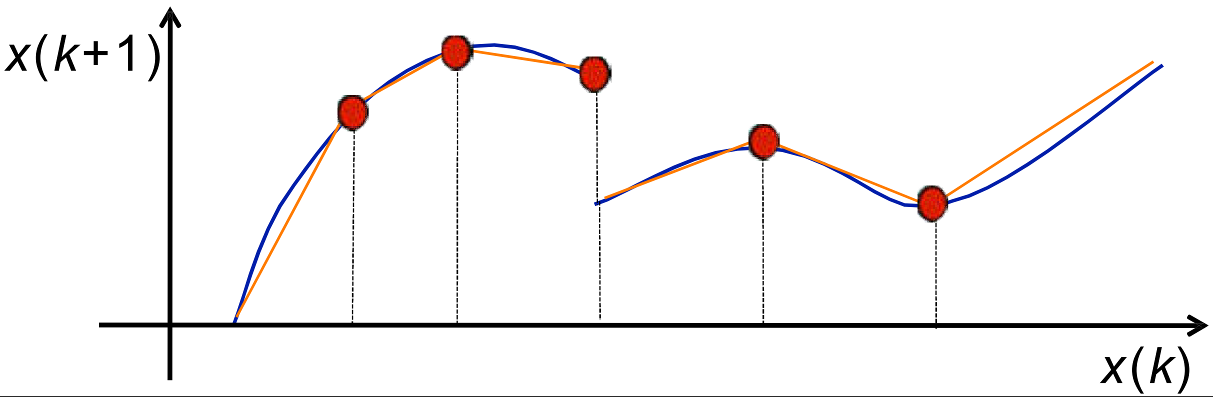
A. de Saint-Exupéry
(1900–1944)

Piecewise affine systems

$$
\begin{aligned}
x(k+1) &= A_{i(k)}x(k) + B_{i(k)}u(k) + f_{i(k)} \\
y(k) &= C_{i(k)}x(k) + D_{i(k)}u(k) + g_{i(k)} \\
i(k) \quad \text{s.t.} \quad & H_{i(k)}x(k) + J_{i(k)}u(k) \leq K_{i(k)}
\end{aligned}
$$



- PWA systems can approximate nonlinear dynamics arbitrarily well (even discontinuous ones)



Piecewise Affine (PWA) Systems

Examples:

- linearization of a non-linear system at different operating point ⇒ 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.

Piecewise Affine (PWA) Systems

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

Boolean Algebra: Basic Definitions and Notation

- **Boolean variable:** A variable 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 $\delta_1, \ldots, \delta_{n-1}$:

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

Piecewise Affine (PWA) Systems

Example:

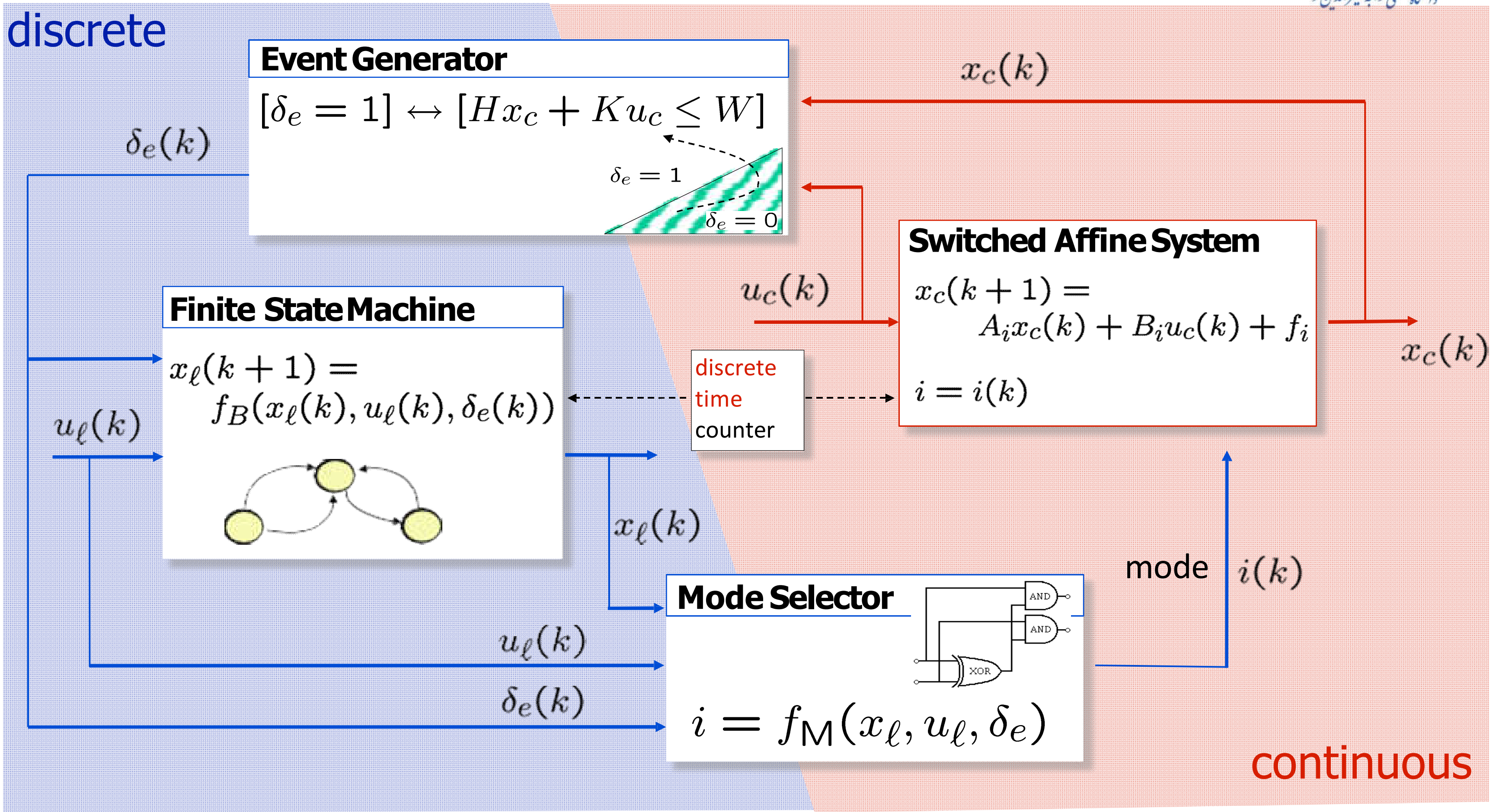
$$\begin{aligned} x_c(t+1) &= 2x_c(t) + u_c(t) - 3u_\ell(t) \\ x_\ell(t+1) &= x_\ell(t) \wedge u_\ell(t) \end{aligned}$$

can be represented in the PWA form

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

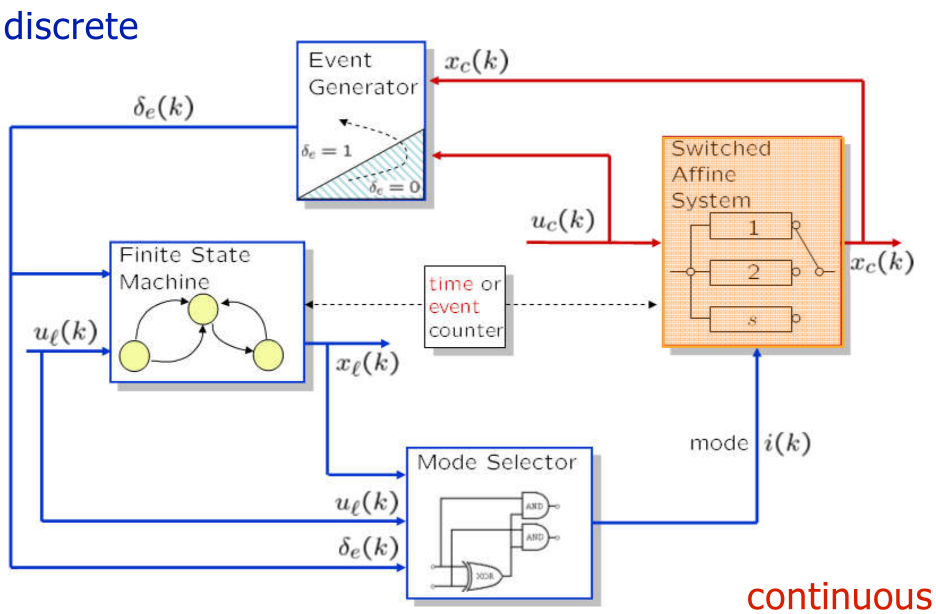
by associating $x_\ell = 0$ with $x_\ell \le \frac{1}{2}$ and $x_\ell = 1$ with $x_\ell \ge \frac{1}{2} + \epsilon$ for any $0 < \epsilon \le \frac{1}{2}$.

Discrete Hybrid Automaton (DHA)



$x_\ell \in \{0, 1\}^{n_\ell}$	=	**binary state**	$x_c \in \mathbb{R}^{n_c}$	=	**real-valued state**
$u_\ell \in \{0, 1\}^{m_\ell}$	=	**binary input**	$u_c \in \mathbb{R}^{m_c}$	=	**real-valued input**
$\delta_e \in \{0, 1\}^{n_e}$	=	**event variable**	$i \in \{1, \ldots, s\}$	=	**current mode**

Switched affine system

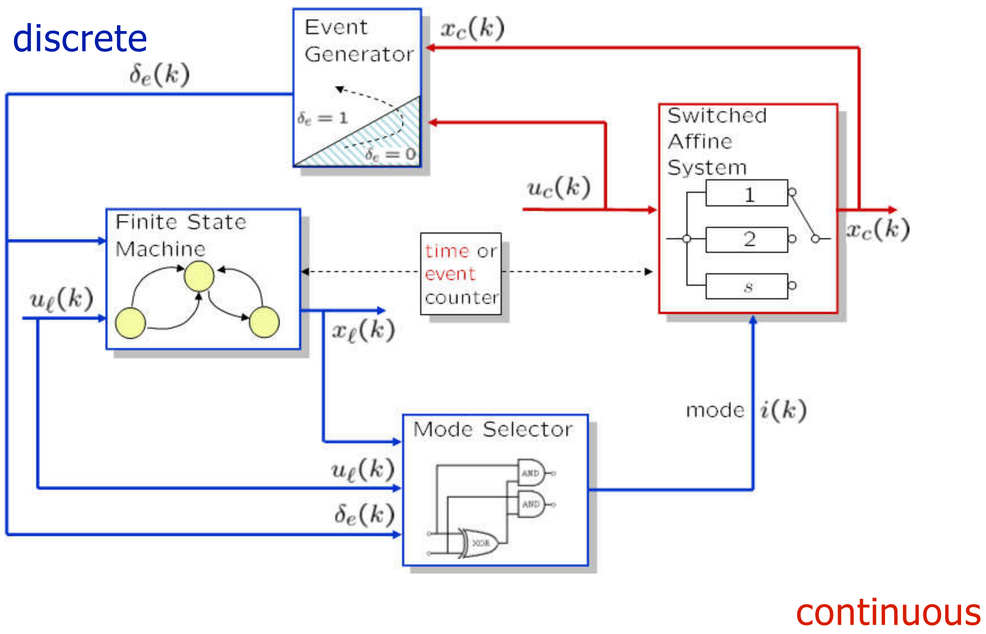


- The **affine dynamics** depend on the current mode $i(k)$:

$$x_c(k+1) = A_{i(k)}x_c(k) + B_{i(k)}u_c(k) + f_{i(k)}$$

$$x_c \in \mathbb{R}^{n_c}, u_c \in \mathbb{R}^{m_c}$$

Event generator



- Event variables are generated by linear threshold conditions over continuous states, continuous inputs, and time:

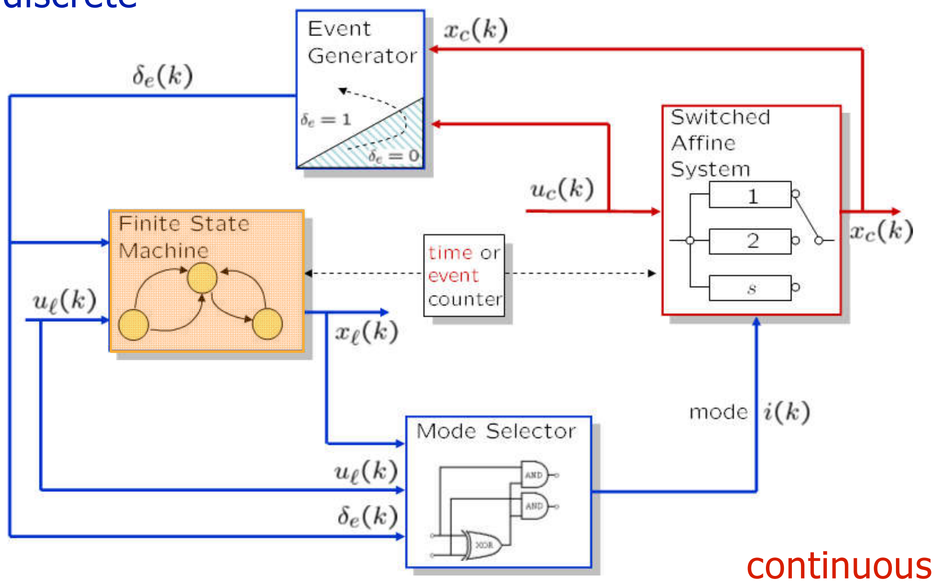
$$[\delta_e^i(k) = 1] \leftrightarrow [H^i x_c(k) + K^i u_c(k) \leq W^i]$$

$$x_c \in \mathbb{R}^{n_c}, \quad u_c \in \mathbb{R}^{m_c}$$
$$\delta_e \in \{0, 1\}^{n_e}$$

- Example: $[\delta_e(k) = 1] \leftrightarrow [x_c(k) \geq 0]$

Finite state machine

discrete



continuous

- The binary state of the **finite state machine** evolves according to a Boolean state update function $f_B : \{0,1\}^{n_\ell+m_\ell+n_e} \to \{0,1\}^{n_\ell}$:

$$x_\ell(k+1) = f_B(x_\ell(k), u_\ell(k), \delta_e(k))$$

$x_\ell \in \{0,1\}^{n_\ell}, \quad u_\ell \in \{0,1\}^{m_\ell}$

$\delta_e \in \{0,1\}^{n_e}$

- Example: $x_\ell(k+1) = \neg\delta_e(k) \vee (x_\ell(k) \wedge u_\ell(k))$

Transformation of a DHA into linear (in)equalities

$X_1 \vee X_2 = \text{TRUE}$ ⟶ $\delta_1 + \delta_2 \geq 1, \quad \delta_1, \delta_2 \in \{0, 1\}$

Any logic statement

$f(X) = \text{TRUE}$

$$\bigwedge_{j=1}^{m} \left(\vee_{i \in P_j} X_i \vee_{i \in N_j} \neg X_i \right) \quad \text{(CNF)}$$

$N_j, P_j \subseteq \{1, \ldots, n\}$

⟶

$$\begin{cases} 1 \leq \sum_{i \in P_1} \delta_i + \sum_{i \in N_1} (1 - \delta_i) \\ \vdots \\ 1 \leq \sum_{i \in P_m} \delta_i + \sum_{i \in N_m} (1 - \delta_i) \end{cases}$$

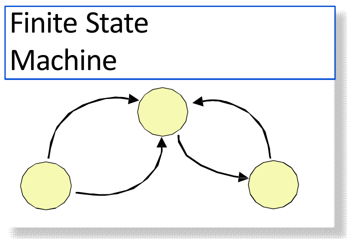
$[\delta_e^i(k) = 1] \leftrightarrow [H^i x_c(k) \leq W^i]$ ⟶

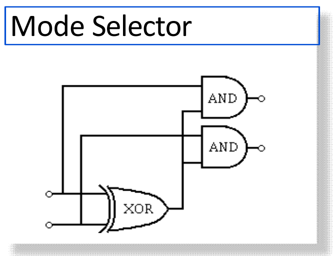
$$\begin{cases} H^i x_c(k) - W^i \leq M^i(1 - \delta_e^i(k)) \\ H^i x_c(k) - W^i > m^i \delta_e^i(k) \end{cases}$$

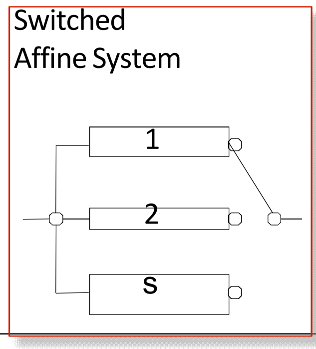
IF $[\delta = 1]$ THEN $z = a_1^T x + b_1^T u + f_1$
ELSE $z = a_2^T x + b_2^T u + f_2$

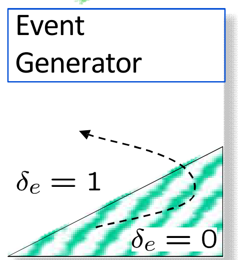
⟶

$$\begin{cases} (m_1 - M_2)(1 - \delta) + z & \leq & a_1 x + b_1 u + f_1 \\ (m_2 - M_1)(1 - \delta) - z & \leq & -a_1 x - b_1 u - f_1 \\ (m_2 - M_1)\delta + z & \leq & a_2 x + b_2 u + f_2 \\ (m_1 - M_2)\delta - z & \leq & -a_2 x - b_2 u - f_2 \end{cases}$$









Hybrid MPC

- Modeling of Hybrid Systems
  - Introduction
  - Examples of Hybrid Systems
  - Piecewise Affine (PWA) Systems
  - Mixed Logical Dynamical (MLD) Hybrid Model
- Optimal Control of Hybrid Systems
- Model Predictive Control of Hybrid Systems
- MPC of Hybrid Systems Examples

Mixed Logical Dynamical (MLD) 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

Mixed Logical Dynamical (MLD) systems

Boolean formulas as Linear Integer Inequalities

Goal

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

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

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

Mixed Logical Dynamical (MLD) systems

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	$\delta_1 \wedge \delta_2$	$\delta_1 \geq 1,\ \delta_2 \geq 1$ also $\delta_1 + \delta_2 \geq 2$
OR	$\delta_1 \vee \delta_2$	$\delta_1 + \delta_2 \geq 1$
NOT	$\neg\delta_1$	$(1 - \delta_1) \geq 1$ also $\delta_1 = 0$
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 $\delta_3 = \delta_1 \wedge \delta_2$	$\delta_3 \leftrightarrow \delta_1 \wedge \delta_2$	$\delta_1 + (1 - \delta_3) \geq 1$ $\delta_2 + (1 - \delta_3) \geq 1$ $(1 - \delta_1) + (1 - \delta_2) + \delta_3 \geq 1$

Mixed Logical Dynamical (MLD) systems

- By converting logic relations into mixed-integer linear inequalities a DHA can be rewritten as the **Mixed Logical Dynamical (MLD)** system

$$\begin{cases} x(k+1) & = & Ax(k) + B_1u(k) + B_2\delta(k) + B_3z(k) + B_5 \\ y(k) & = & Cx(k) + D_1u(k) + D_2\delta(k) + D_3z(k) + D_5 \\ E_2\delta(k) & + & E_3z(k) \leq E_4x(k) + E_1u(k) + E_5 \end{cases}$$

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

- The translation from DHA to MLD can be automatized, see e.g. the language **HYSDEL** (HYbrid Systems DEscription Language) (Torrisi, Bemporad, 2004)

- MLD models allow solving MPC, verification, state estimation, and fault detection problems via **mixed-integer programming**

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/