Modeling & Control of Hybrid Systems

Chapter 2 — Modeling frameworks

- Many modeling frameworks for hybrid systems
  $\Rightarrow$ trade-off: modeling power $\leftrightarrow$ decision power, tractability
- Hybrid automata:
  - very general, high modeling power, but low decision power
  - analysis and control $\longrightarrow$ computationally hard
    (NP-hard, undecidable problems)

- Computer simulation and verification tools: Modelica, HyTech, KRONOS, Chi, 20-sim, UPPAAL, . . .
  - \+ simulation models can represent plant with high degree of detail (high modeling power)
  - \- computationally very demanding for large systems
  - \- difficult to understand from simulation how behavior depends on model parameters
- In this chapter: special classes of hybrid systems for which *tractable* analysis and control design techniques are available (cf. next chapters)

Overview

1. Piecewise affine systems (PWA)
2. Mixed Logical Dynamical systems (MLD)
3. Linear Complementarity systems (LC)
4. Extended Linear Complementarity systems (ELC)
5. Max-Min-Plus-Scaling systems (MMPS)
6. Equivalence of MLD, LC, ELC, PWA and MMPS systems
7. Timed automata
8. Timed Petri nets

1. Piecewise affine (PWA) systems

- PWA systems are described by

$$\begin{aligned} x(k+1) &= A_i x(k) + B_i u(k) + f_i \\ y(k) &= C_i x(k) + D_i u(k) + g_i \end{aligned} \quad \text{for} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \in \Omega_i, \; i = 1, \ldots, N$$

- $\Omega_1, \ldots, \Omega_N$: convex polyhedra (i.e., given by finite number of linear inequalities) in input/state space, non-overlapping interiors
- PWA can be used as approximation of nonlinear model

$$\begin{aligned} x(k+1) &= \mathcal{N}_x(x(k), u(k)) \\ y(k) &= \mathcal{N}_y(x(k), u(k)) \end{aligned}$$

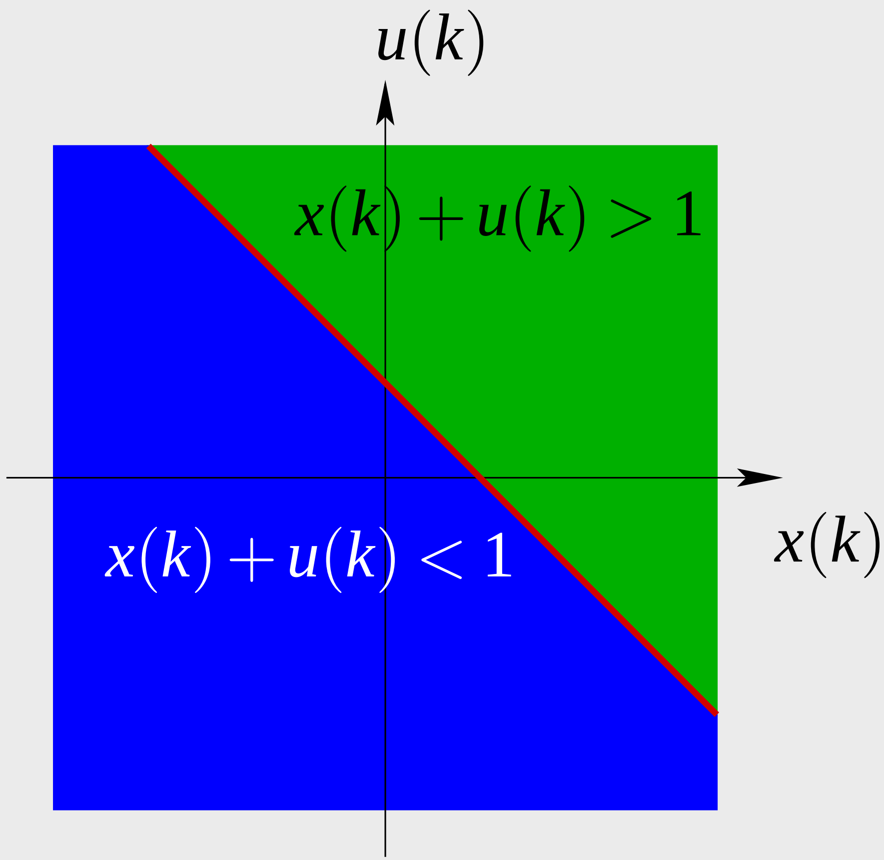
$\rightarrow$ "simplest" extension of linear systems that can still model non-linear & non-smooth processes with arbitrary accuracy + are capable of handling hybrid phenomena

Example of PWA model

Integrator with upper saturation:

$$x(k+1) = \begin{cases} x(k)+u(k) & \text{if } x(k)+u(k) \leqslant 1 \\ 1 & \text{if } x(k)+u(k) \geqslant 1 \end{cases}$$

$$y(k) = x(k)$$



2. Mixed Logical Dynamical (MLD) systems

2.1 Preliminaries

- Boolean operators:
  $\wedge$ (and), $\vee$ (or), $\sim$ (not), $\Rightarrow$ (implies), $\Leftrightarrow$ (iff), $\oplus$ (xor)

X_1	X_2	$X_1 \wedge X_2$	$X_1 \vee X_2$	$\sim X_1$	$X_1 \Rightarrow X_2$	$X_1 \Leftrightarrow X_2$	$X_1 \oplus X_2$
T	T	T	T	F	T	T	F
T	F	F	T	F	F	F	T
F	T	F	T	T	T	F	T
F	F	F	F	T	T	T	F

- Properties:
  - $X_1 \Rightarrow X_2$ is same as $\sim X_1 \vee X_2$
  - $X_1 \Rightarrow X_2$ is same as $\sim X_2 \Rightarrow \sim X_1$
  - $X_1 \Leftrightarrow X_2$ is same as $(X_1 \Rightarrow X_2) \wedge (X_2 \Rightarrow X_1)$

- Associate with literal X_i logical variable $\delta_i \in \{0,1\}$:
  $\delta_i = 1$ iff $X_i = \text{T}$, $\delta_i = 0$ iff $X_i = \text{F}$
  $\longrightarrow$ compound statement can be transformed into
  linear integer program
- Examples:
  * $X_1 \wedge X_2$ equivalent to $\delta_1 = \delta_2 = 1$
  * $X_1 \vee X_2$ equivalent to $\delta_1 + \delta_2 \geqslant 1$
  * $\sim X_1$ equivalent to $\delta_1 = 0$
  * $X_1 \Rightarrow X_2$ equivalent to $\delta_1 - \delta_2 \leqslant 0$
  * $X_1 \Leftrightarrow X_2$ equivalent to $\delta_1 - \delta_2 = 0$
  * $X_1 \oplus X_2$ equivalent to $\delta_1 + \delta_2 = 1$
- For $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ and $x \in \mathscr{X}$ with $\mathscr{X}$ bounded, define

$$M \stackrel{\text{def}}{=} \max_{x \in \mathscr{X}} f(x) \qquad m \stackrel{\text{def}}{=} \min_{x \in \mathscr{X}} f(x)$$

- Equivalences:
  - * $[f(x) \leqslant 0] \wedge [\delta = 1]$ true iff $f(x) - \delta \leqslant -1 + m(1-\delta)$
  - * $[f(x) \leqslant 0] \vee [\delta = 1]$ true iff $f(x) \leqslant M\delta$
  - * $\sim[f(x) \leqslant 0]$ true iff $f(x) \geqslant \varepsilon$ (with ε machine precision)
  - * $[f(x) \leqslant 0] \Rightarrow [\delta = 1]$ true iff $f(x) \geqslant \varepsilon + (m - \varepsilon)\delta$
  - * $[f(x) \leqslant 0] \Leftrightarrow [\delta = 1]$ true iff $\begin{cases} f(x) \leqslant M(1-\delta) \\ f(x) \geqslant \varepsilon + (m-\varepsilon)\delta \end{cases}$
- Product $\delta_1\delta_2$ can be replaced by auxiliary variable $\delta_3 = \delta_1\delta_2$
  Since $[\delta_3 = 1] \Leftrightarrow [\delta_1 = 1] \wedge [\delta_2 = 1]$,

$$\delta_3 = \delta_1 \delta_2 \quad \text{is equivalent to} \quad \begin{cases} -\delta_1 + \delta_3 \leqslant 0 \\ -\delta_2 + \delta_3 \leqslant 0 \\ \delta_1 + \delta_2 - \delta_3 \leqslant 1 \end{cases}$$

- $\delta f(x)$ can be replaced by auxiliary real variable $y = \delta f(x)$ with $[\delta = 0] \Rightarrow [y = 0]$, $[\delta = 1] \Rightarrow [y = f(x)]$, or equivalently

$$\begin{cases} y \leqslant M\delta \\ y \geqslant m\delta \\ y \leqslant f(x) - m(1-\delta) \\ y \geqslant f(x) - M(1-\delta) \end{cases}$$

2.2 Mixed logical dynamical (MLD) systems

- $$x(k+1) = Ax(k) + B_1 u(k) + B_2 \delta(k) + B_3 z(k)$$
  $$y(k) = Cx(k) + D_1 u(k) + D_2 \delta(k) + D_3 z(k)$$
  $$E_1 x(k) + E_2 u(k) + E_3 \delta(k) + E_4 z(k) \leqslant g_5,$$

- $x(k) = \left[x_{\mathrm{r}}^{\mathsf{T}}(k) \; x_{\mathrm{b}}^{\mathsf{T}}(k) \right]^{\mathsf{T}}$ with $x_{\mathrm{r}}(k)$ real-valued, $x_{\mathrm{b}}(k)$ boolean
  $z(k)$: real-valued auxiliary variables
  $\delta(k)$: boolean auxiliary variables

- Applications: PWA systems, systems with discrete inputs, qualitative inputs, bilinear systems, finite state machines

- Reference: A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, March 1999.

2.3 Example

- Consider PWA system:

$$x(k+1) = \begin{cases} 0.8x(k) + u(k) & \text{if } x(k) \geqslant 0 \\ -0.8x(k) + u(k) & \text{if } x(k) < 0 \end{cases}$$

  where $x(k) \in [-10, 10]$ and $u(k) \in [-1, 1]$

- Associate binary variable $\delta(k)$ to condition $x(k) \geqslant 0$ such that $[\delta(k) = 1] \Leftrightarrow [x(k) \geqslant 0]$ or

$$-m\delta(k) \leqslant x(k) - m$$
$$-(M + \varepsilon)\delta(k) \leqslant -x(k) - \varepsilon$$

  where $M = -m = 10$, and ε is machine precision

- PWA system can be rewritten as

$$x(k+1) = 1.6\,\delta(k)x(k) - 0.8x(k) + u(k)$$

- $x(k+1) = 1.6\,\delta(k)x(k) - 0.8\,x(k) + u(k)$
- Define new variable $z(k) = \delta(k)\,x(k)$ or

$$
\begin{aligned}
z(k) &\leqslant M\delta(k) \\
z(k) &\geqslant m\delta(k) \\
z(k) &\leqslant x(k) - m(1-\delta(k)) \\
z(k) &\geqslant x(k) - M(1-\delta(k))
\end{aligned}
$$

- PWA system now becomes

$$x(k+1) = 1.6z(k) - 0.8x(k) + u(k)$$

subject to linear constraints above $\rightarrow$ MLD

3. Linear Complementarity (LC) systems

- LC systems:

$$\begin{aligned} x(k+1) &= Ax(k) + B_1u(k) + B_2w(k) \\ y(k) &= Cx(k) + D_1u(k) + D_2w(k) \\ v(k) &= E_1x(k) + E_2u(k) + E_3w(k) + e_4 \\ 0 \leqslant v(k) &\perp w(k) \geqslant 0 \end{aligned}$$

- $v(k)$, $w(k)$: “complementarity variables” (real-valued)
- Applications: constrained mechanical systems, electrical networks with ideal diodes, dynamical systems with PWA relations, variable-structure systems, projected dynamical systems
- Examples: two-cars system, boost converter (*continuous*-time LC systems)

4. Extended Linear Complementarity (ELC) systems

- ELC systems:

$$x(k+1) = Ax(k) + B_1u(k) + B_2d(k) \quad (1)$$
$$y(k) = Cx(k) + D_1u(k) + D_2d(k) \quad (2)$$
$$E_1x(k) + E_2u(k) + E_3d(k) \leqslant e_4 \quad (3)$$
$$\sum_{i=1}^{p} \prod_{j\in\phi_i} \left(e_4 - E_1x(k) - E_2u(k) - E_3d(k)\right)_j = 0 \quad (4)$$

- $d(k)$: real-valued auxiliary variable
- Condition (4) is equivalent to

$$\prod_{j\in\phi_i} \left(e_4 - E_1x(k) - E_2u(k) - E_3d(k)\right)_j = 0 \quad \text{for each } i \in \{1,\ldots,p\}$$

$\longrightarrow$ system of linear inequalities with p groups, in each group at least one inequality should hold with equality

5. Max-Min-Plus-Scaling (MMPS) systems

- Max-min-plus-scaling expression:

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

  with α, $\beta \in \mathbb{R}$ and f_k, f_l again MMPS expressions.

- Example: $5x_1 - 3x_2 + 7 + \max(\min(2x_1, -8x_2), x_2 - 3x_3)$
- MMPS systems:

$$\begin{aligned} x(k+1) &= \mathscr{M}_x(x(k), u(k), d(k)) \\ y(k) &= \mathscr{M}_y(x(k), u(k), d(k)) \\ \mathscr{M}_\text{c}(x(k), &u(k), d(k)) \leqslant c \end{aligned}$$

  with $\mathscr{M}_x$, $\mathscr{M}_y$, $\mathscr{M}_\text{c}$ MMPS expressions

- $d(k)$: real-valued auxiliary variables

5. Max-Min-Plus-Scaling (MMPS) systems (continued)

- Applications:
  - discrete-event systems (also max-plus)
  - traffic-signal controlled intersection
  - railway networks
  - manufacturing systems
  - systems with soft & hard synchronization constraints
  - logistic systems

Example of MMPS system

- Integrator with upper saturation:

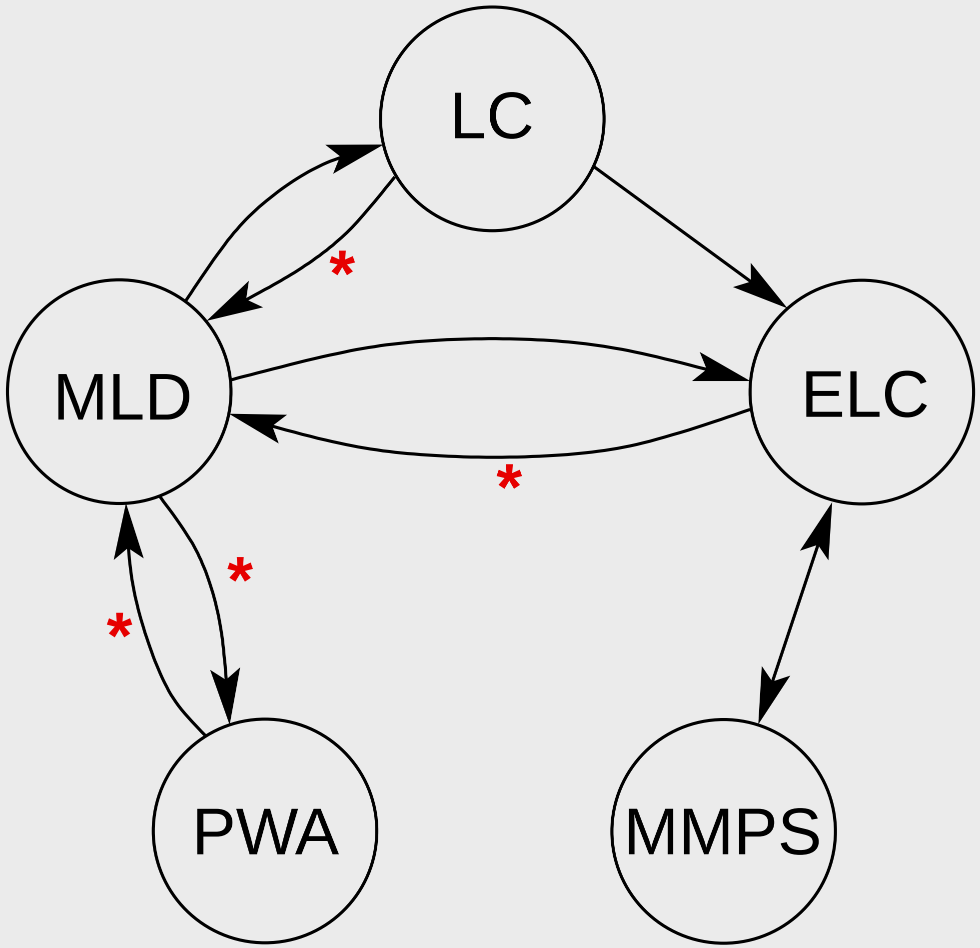
$$
x(k+1) = \begin{cases} x(k)+u(k) & \text{if } x(k)+u(k) \leqslant 1 \\ 1 & \text{if } x(k)+u(k) \geqslant 1 \end{cases}
$$
$$
y(k) = x(k)
$$

can be recast as

$$
x(k+1) = \min(x(k)+u(k), 1)
$$
$$
y(k) = x(k)
$$

6. Equivalence of MLD, LC, ELC, PWA and MMPS systems

Equivalence between model classes $\mathscr{A}$ and $\mathscr{B}$:
for each model $\in \mathscr{A}$ there exists model $\in \mathscr{B}$ with same input/output behavior (+ vice versa)



Equivalence of MLD, LC, ELC, PWA and MMPS systems

- Each subclass has own advantages:
  - stability criteria for PWA
  - control and verification techniques for MLD
  - control techniques for MMPS
  - conditions of existence and uniqueness of solutions for LC

  $\longrightarrow$ transfer techniques from one class to other
- It depends on the application which class is best suited

6.1 MLD and LC systems

Proposition *Every MLD system can be written as LC system.*

- $\delta_i(k) \in \{0,1\}$ is equivalent to $0 \leqslant \delta_i(k) \perp 1 - \delta_i(k) \geqslant 0$
  $\rightarrow$ introduce auxiliary variable $p(k) = [1 \;\; 1 \; \ldots \; 1]^{\mathsf{T}} - \delta(k)$ with

$$0 \leqslant \delta(k) \perp p(k) \geqslant 0$$

- For constraint $E_1 x(k) + E_2 u(k) + E_3 \delta(k) + E_4 z(k) \leqslant g_5$, introduce auxiliary variables $q(k) = g_5 - E_1 x(k) - E_2 u(k) - E_3 \delta(k) - E_4 z(k) \geqslant 0$ and $r(k) = 0$ with

$$0 \leqslant q(k) \perp r(k) \geqslant 0$$

- For LC: all variables $\geqslant 0$
  $\rightarrow$ split real-valued variable $z(k)$ in “positive” and “negative part”:
  $z(k) = z^+(k) - z^-(k)$ with $z^+(k) = \max(0, z(k))$, $z^-(k) = \max(0, -z(k))$
  or $0 \leqslant z^+(k) \perp z^-(k) \geqslant 0$
- Results in LC system:

$$x(k+1) = Ax(k) + B_1 u(k) + [B_2 \;\; 0 \;\; B_3 \;\; -B_3] w(k)$$

$$y(k) = Cx(k) + D_1 u(k) + [D_2 \;\; 0 \;\; D_3 \;\; -D_3] w(k)$$

$$\underbrace{\begin{pmatrix} p(k) \\ q(k) \\ s(k) \\ t(k) \end{pmatrix}}_{=:v(k)} = \begin{pmatrix} e \\ g_5 - E_1 x(k) - E_2 u(k) \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} -I & 0 & 0 & 0 \\ -E_3 & 0 & -E_4 & E_4 \\ 0 & 0 & 0 & I \\ 0 & 0 & I & 0 \end{pmatrix} \underbrace{\begin{pmatrix} \delta(k) \\ r(k) \\ z^+(k) \\ z^-(k) \end{pmatrix}}_{=:w(k)}$$

$$0 \leqslant v(k) \perp w(k) \geqslant 0$$

Proposition *Every LC system can be written as MLD provided that* $w(k)$ *and* $v(k)$ *are bounded.*

- LC complementarity condition $0 \leqslant v(k) \perp w(k) \geqslant 0$ implies that for each i we have $v_i(k) = 0,\ w_i(k) \geqslant 0$ or $v_i(k) \geqslant 0,\ w_i(k) = 0$
- Introduce boolean vector $\delta(k)$ such that

$$
\begin{aligned}
v_i(k) = 0,\ w_i(k) \geqslant 0 \quad &\leftrightarrow \quad \delta_i(k) = 1 \\
v_i(k) \geqslant 0,\ w_i(k) = 0 \quad &\leftrightarrow \quad \delta_i(k) = 0
\end{aligned}
$$

- Can be achieved by introducing constraints

$$
\begin{aligned}
& w(k) \leqslant M_w \delta(k) \\
& v(k) \leqslant M_v([1\,1\ \ldots\ 1]^{\mathsf{T}} - \delta(k)) \\
& w(k), v(k) \geqslant 0
\end{aligned}
$$

with M_w, M_v diagonal matrices containing upper bounds on $w(k), v(k)$

- Note: Upper bounds usually known in practice due to physical reasons/insight.
- Finally results in MLD model

$$x(k+1) = Ax(k) + B_1u(k) + B_2z(k)$$
$$y(k) = Cx(k) + D_1u(k) + D_2z(k)$$
$$\begin{bmatrix} 0 \\ E_1 \\ 0 \\ -E_1 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ E_2 \\ 0 \\ -E_2 \end{bmatrix} u(k) + \begin{bmatrix} -M_w \\ M_v \\ 0 \\ 0 \end{bmatrix} \delta(k) + \begin{bmatrix} I \\ E_3 \\ -I \\ -E_3 \end{bmatrix} z(k) \leqslant \begin{bmatrix} 0 \\ M_v e - e_4 \\ 0 \\ e_4 \end{bmatrix}$$

6.2 LC and ELC systems

Proposition *Every LC system can be written as ELC system.*

- $v(k) \perp w(k)$ is equivalent to $\sum_i v_i(k) w_i(k) = 0$

6.3 PWA and MLD systems

Proposition *Well-posed PWA system can be rewritten as MLD system assuming that set of feasible states and inputs is bounded.*

- Cf. examples.

Proposition *Completely well-posed MLD can be rewritten as PWA.*

- If $\delta(k) \in \{0,1\}^s \rightarrow 2^s$ possible combinations
- For each combination MLD constraint

$$E_1 x(k) + E_2 u(k) + E_3 \delta(k) + E_4 z(k) \leqslant g_5$$

  defines polyhedral region in $x/u/z$ space
- For each combination, $z(k)$ is linear function of $u(k)$ and $x(k)$ due to well-posedness + linearity of all constraints
- Results in linear state space model for each polyhedral region

6.4 MMPS and ELC systems

Proposition *The classes of MMPS and ELC systems coincide.*

MMPS $\subseteq$ ELC

- Basic constructors for MMPS expressions fit ELC framework:
  - Expressions of form $f = x_i$, $f = \alpha$, $f = f_k + f_l$, $f = \beta f_k$ result in linear equations
  - $f = \max(f_k, f_l) = -\min(-f_k, -f_l)$ can be rewritten as

$$f - f_k \geqslant 0, \quad f - f_l \geqslant 0, \quad (f - f_k)(f - f_l) = 0$$

    $\rightarrow$ is ELC expression
- Two or more ELC systems can be combined into one large ELC

6.4 MMPS and ELC systems (continued)

ELC $\subseteq$ MMPS

- Linear equations are MMPS expressions (albeit without $\max$ or $\min$)
- Complementarity condition can be rewritten as

$$\forall i, \exists j \in \phi_i \text{ such that } \underbrace{\big(e_4 - E_1 x(k) - E_2 u(k) - E_3 d(k)\big)_j}_{\geqslant 0} = 0$$

So

$$\min_{j \in \phi_i} \big(e_4 - E_1 x(k) - E_2 u(k) - E_3 d(k)\big)_j = 0 \quad \text{for each } i$$

6.5 MLD and ELC systems

Proposition *Every MLD system can be rewritten as ELC system.*

- Condition $\delta_i(k) \in \{0,1\}$ is equivalent to ELC conditions

$$-\delta_i(k) \leqslant 0$$
$$\delta_i(k) \leqslant 1$$
$$\delta_i(k)(1-\delta_i(k)) = 0$$

- Note: condition $\delta_i(k) \in \{0,1\}$ also equivalent to MMPS constraints

$$\max(-\delta_i(k), \delta_i(k) - 1) = 0$$

or

$$\min(\delta_i(k), 1-\delta_i(k)) = 0$$

Proposition *Every ELC system can be written as MLD system, provided that* $e_4 - E_1 x(k) - E_2 u(k) - E_3 d(k)$ *is bounded.*

- Introduce conditions

$$(e_4)_j - (E_1 x(k) + E_2 u(k) + E_3 d(k))_j \leqslant M_j \delta_j(k) \quad \text{for each } j \in \phi_i$$

$$\sum_{j \in \phi_i} \delta_j(k) \leqslant \#\phi_i - 1$$

  with $\delta_j(k) \in \{0, 1\}$ auxiliary variables,
  and M_j upper bound for $(e_4 - E_1 x(k) - E_2 u(k) - E_3 d(k))_j$

- By last condition at least one $\delta_h(k)$ is zero for some $h \in \phi_i$
  $\rightarrow$ 1st inequality and ELC inequality $(e_4)_j - (E_1 x(k) + E_2 u(k)$ $+ E_3 d(k))_j \geqslant 0$ degenerate to equality condition for $j = h$

- Hence, (nonlinear) ELC complementarity condition can be replaced by above (linear) equations $\rightarrow$ MLD system

6.6 Example

- Consider

$$x(k+1) = \begin{cases} 0.8x(k) + u(k) & \text{if } x(k) \geqslant 0 \\ -0.8x(k) + u(k) & \text{if } x(k) < 0 \end{cases}$$

  with $m \leqslant x(k) \leqslant M$

- MLD:

$$\begin{aligned} x(k+1) &= -0.8x(k) + u(k) + 1.6z(k) \\ -m\delta(k) &\leqslant x(k) - m & x(k) &\leqslant (M+\varepsilon)\delta(k) - \varepsilon \\ z(k) &\leqslant M\delta(k) & z(k) &\geqslant m\delta(k) \\ z(k) &\leqslant x(k) - m(1-\delta(k)) & z(k) &\geqslant x(k) - M(1-\delta(k)) \end{aligned}$$

  with $\delta(k) \in \{0,1\}$

- MMPS:

$$x(k+1) = -0.8x(k) + 1.6\max(0, x(k)) + u(k)$$

6.6 Example (continued)

- Consider

$$x(k+1) = \begin{cases} 0.8x(k) + u(k) & \text{if } x(k) \geqslant 0 \\ -0.8x(k) + u(k) & \text{if } x(k) < 0 \end{cases}$$

- LC:

$$x(k+1) = -0.8x(k) + u(k) + 1.6z(k)$$
$$0 \leqslant w(k) = -x(k) + z(k) \perp z(k) \geqslant 0$$

- ELC:

$$x(k+1) = -0.8x(k) + u(k) + 1.6d(k)$$
$$-d(k) \leqslant 0, \qquad x(k) - d(k) \leqslant 0, \qquad \big(x(k) - d(k)\big)\big(-d(k)\big) = 0$$

7. Timed automata

- Timed automata involve simple continuous dynamics:
  - all differential equations of form $\dot{x} = 1$
  - all invariants, guards, etc. involve comparison of real-valued states with constants (e.g., $x = 1$, $x < 2$, $x \geqslant 0$, etc.)
- Timed automata are limited for modeling physical systems
- However, very well suited for encoding timing constraints such as "event A must take place at least 2 seconds after event B and not more than 5 seconds before event C"
- Applications: multimedia, Internet, audio protocol verification

7.1 Rectangular sets

- Subset of $\mathbb{R}^n$ set is called rectangular if it can be written as finite boolean combination of constraints of form

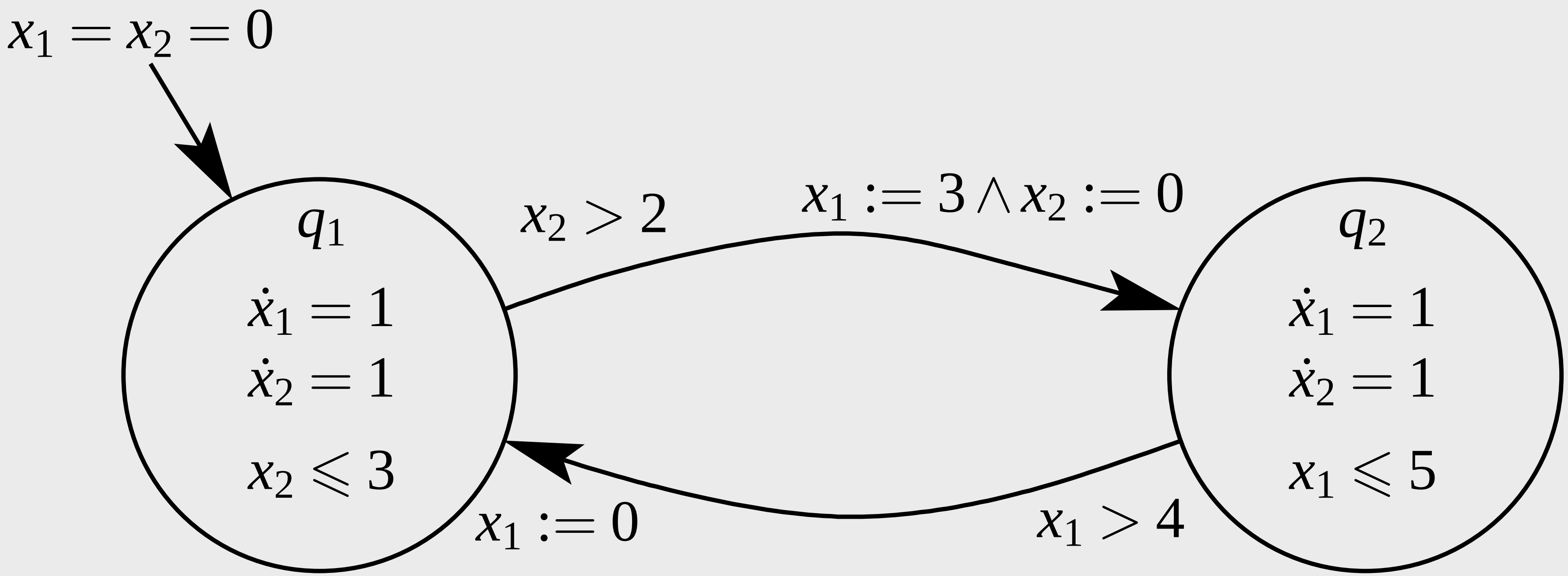
$$x_i \leqslant a, \quad x_i < b, \quad x_i = c, \quad x_i \geqslant d, \quad x_i > e$$

- Rectangular sets are "rectangles" or "boxes" in $\mathbb{R}^n$ whose sides are aligned with the axes, or unions of such rectangles/boxes
- Examples:
  - $\{(x_1,x_2) \mid (x_1 \geqslant 0) \wedge (x_1 \leqslant 2) \wedge (x_2 \geqslant 1) \wedge (x_2 \leqslant 2)\}$
  - $\{(x_1,x_2) \mid ((x_1 \geqslant 0) \wedge (x_2 = 0)) \vee ((x_1 = 0) \wedge (x_2 \geqslant 0))\}$
  - empty set (e.g., $\varnothing = \{(x_1,x_2) \mid (x_1 > 1) \wedge (x_1 \leqslant 0)\}$)
- However, set $\{(x_1,x_2) \mid x_1 = 2x_2\}$ is not rectangular

7.2 Timed automaton

- Timed automaton is hybrid automaton with following characteristics:
  - automaton involves differential equations of form $\dot{x}_i = 1$ continuous variables governed by this differential equation are called “clocks” or “timers”
  - sets involved in definition of initial states, guards, and invariants are rectangular sets
  - reset maps involve either rectangular set, or may leave certain states unchanged

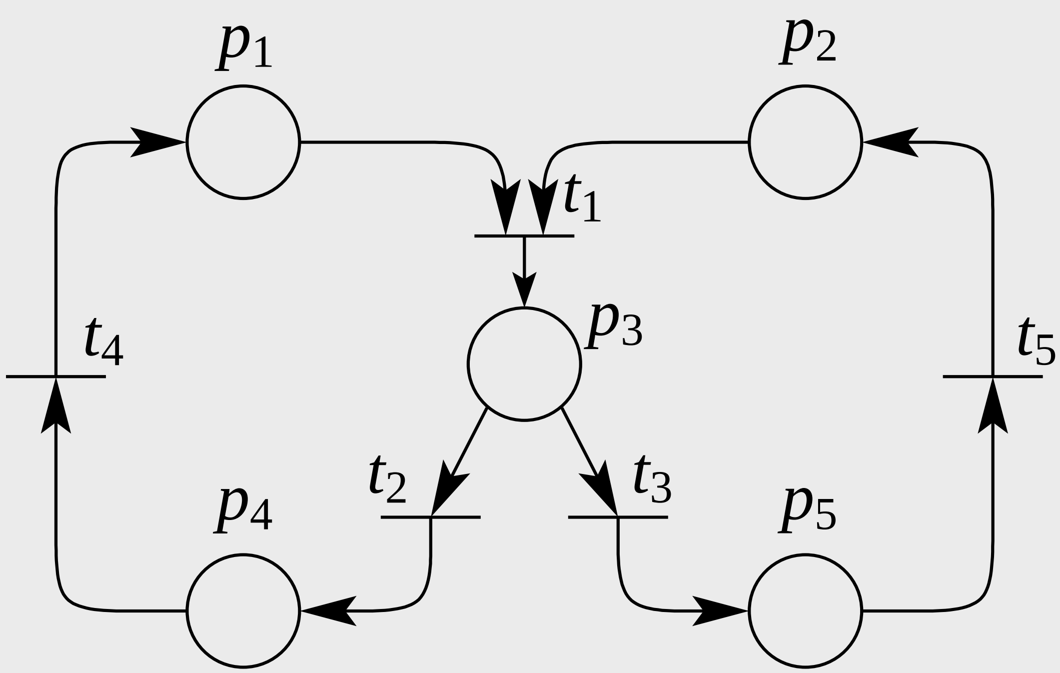
7.3 Example of timed automaton



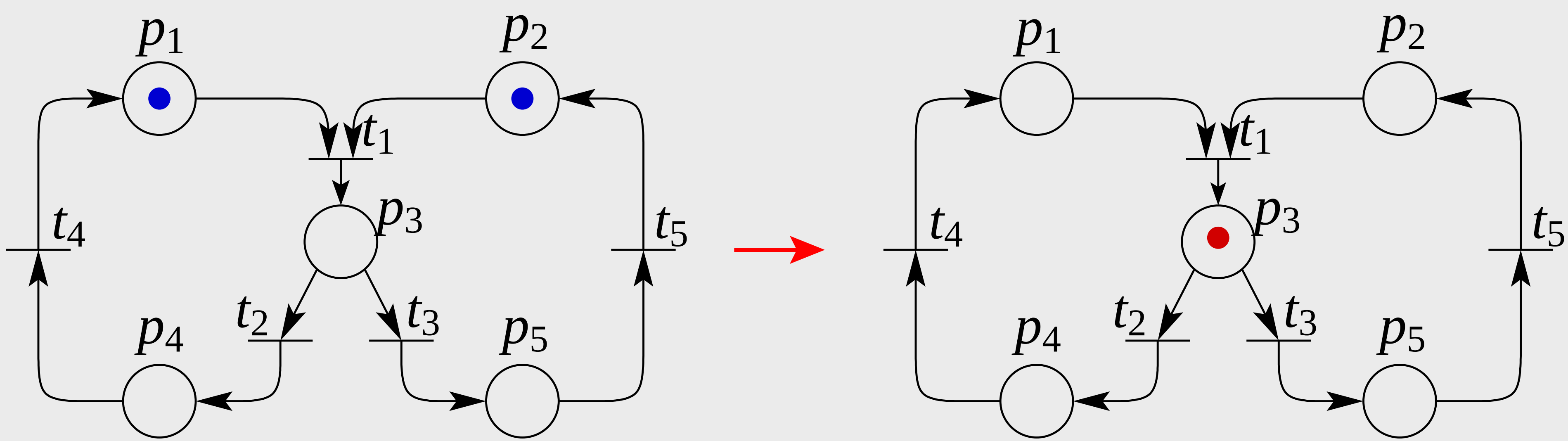
8. Timed Petri nets

8.1 Petri nets

- Graphical representation: bipartite directed graph
  - places (circles) $\rightarrow$ activities
  - transitions (bars) $\rightarrow$ events, actions



- marking $\longrightarrow$ tokens are assigned to places
- execution of Petri net:
  - transition enabled if all input places ($^\bullet t$) contain at least 1 token
  - enabled transition can fire:
    - one token is removed from each input place ($^\bullet t$)
    - one token is deposited in each output place ($t^\bullet$)



- synchronization & choice

8.2 Timed Petri nets

- Untimed Petri net describes order in which events can occur, but no timing
- Timed Petri $\rightarrow$ timing, transition should be executed within certain time interval after it becomes enabled
  - discrete state variables (markings, $m_\theta(p)$)
  - continuous state variables (arrival times, $M_\theta(p)$)
- $M_\theta(p) := \{\theta_1, \ldots, \theta_{m_\theta(p)}\}$ with arrival times $\theta_1 \leqslant \theta_2 \leqslant \ldots \leqslant \theta_{m_\theta(p)}$ of $m_\theta(p)$ tokens in place p
- For each transition t we define interval $[L(t), U(t)]$

8.2 Timed Petri nets (continued)

- Transition t becomes enabled at

$$\max_{p \in {}^{\bullet}t} \min M_\theta(p)$$

- Then transition t may fire at some time

$$\theta \in [\max_{p \in {}^{\bullet}t} \min M_\theta(p) + L(t), \max_{p \in {}^{\bullet}t} \min M_\theta(p) + U(t)]$$

  provided t is enabled during whole interval

- If enabling condition is still valid at final time of firing interval, then transition is forced to fire
- Many techniques for untimed Petri nets can be extended to timed Petri nets
- However, many problems are undecidable or NP-hard