MECH 6v29.002 – Model Predictive Control

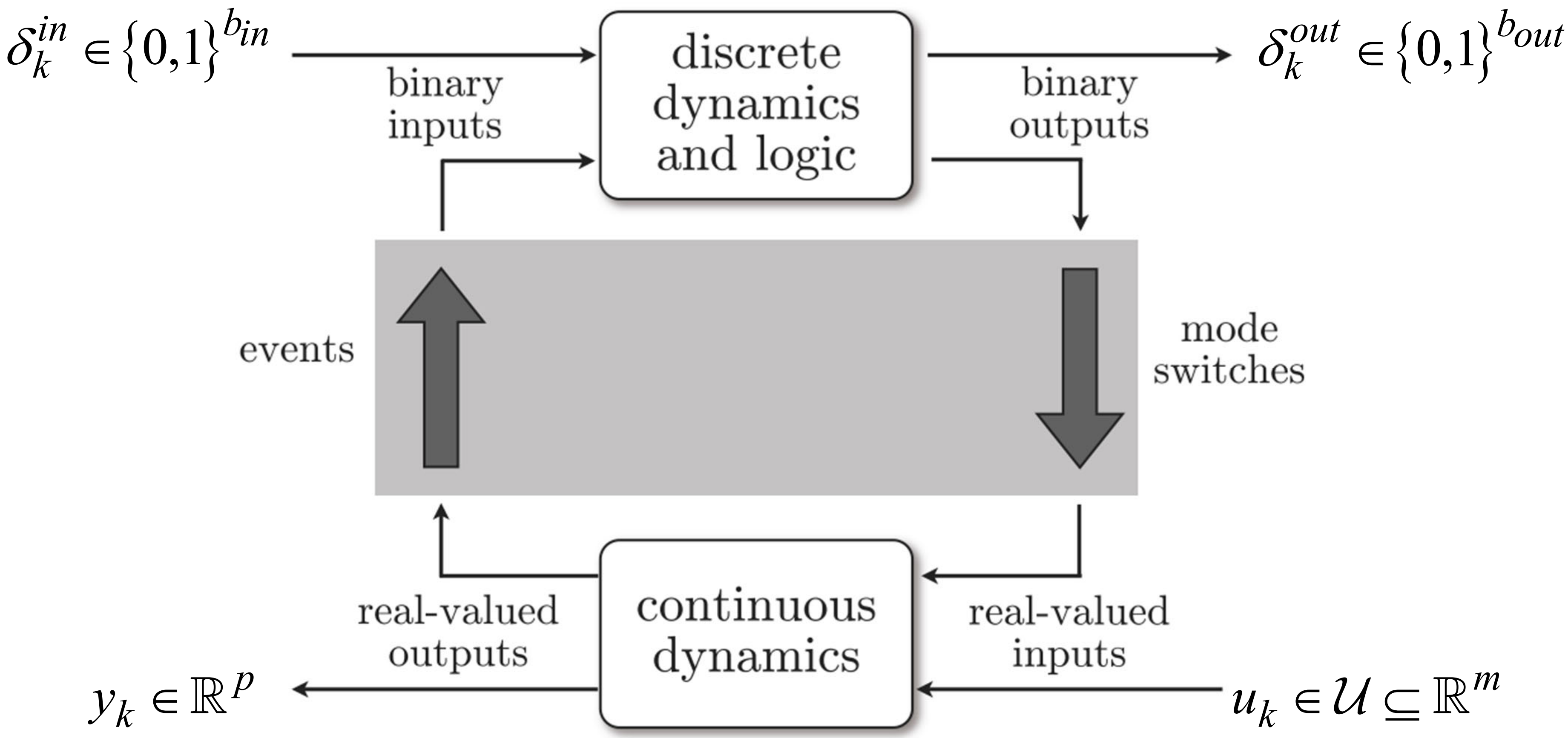
L23 – Hybrid MPC

Outline

- Hybrid Systems
- Models
  - Piecewise Affine systems (PWA)
  - Discrete Hybrid Automata (DHA)
  - Mixed Logical Dynamical (MLD)
- Logic into Mixed-Integer Inequalities
- Nonconvex Sets and Obstacle Avoidance
- Hybrid MPC using MLDs
- Example

Hybrid Systems[1]

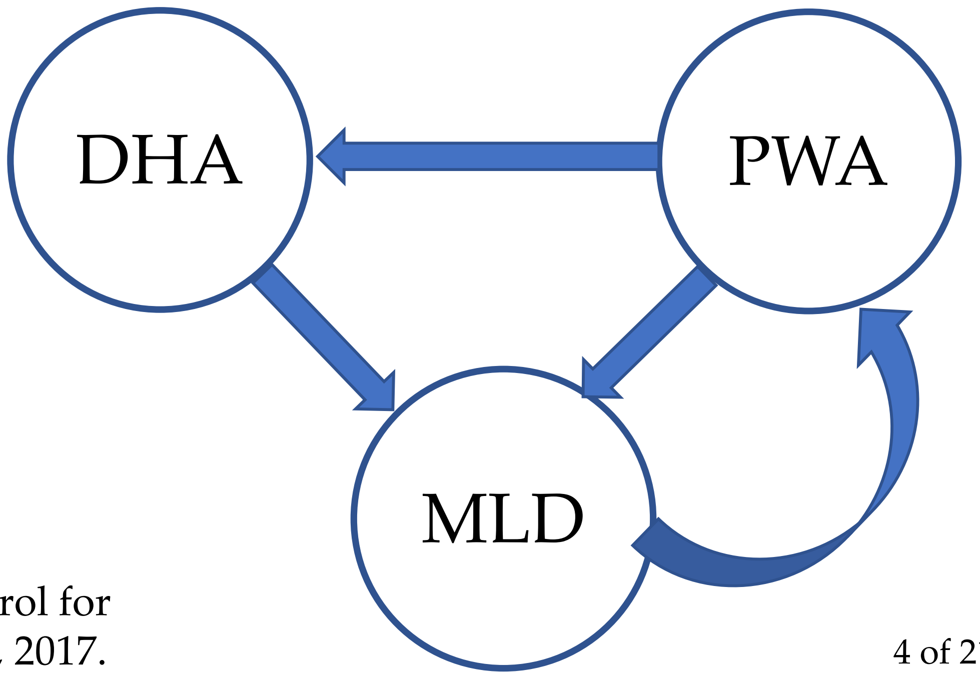
- Hybrid systems are a combination of continuous dynamics (which we have assumed up to now) and logic-based discrete dynamics
  - Not to be confused with continuous-time vs discrete-time dynamics



[1] Borrelli, F; Bemporad, A; Morari, M. Predictive Control for linear and hybrid systems, Cambridge University Press, 2017.

Hybrid System Models[1]

- We will look at 3 ways of modeling hybrid systems
  - PWA – Piecewise Affine systems
  - DHA – Discrete Hybrid Automata
  - MLD – Mixed Logical Dynamical
- Each form has its advantages
  - DHA – most natural in the modeling phase
  - MLD – easiest to formulate MPC optimization problem
  - PWA – easiest to explicitly solve finite-time optimization problem
- Model Equivalence
  - With some assumptions we can convert between different representations



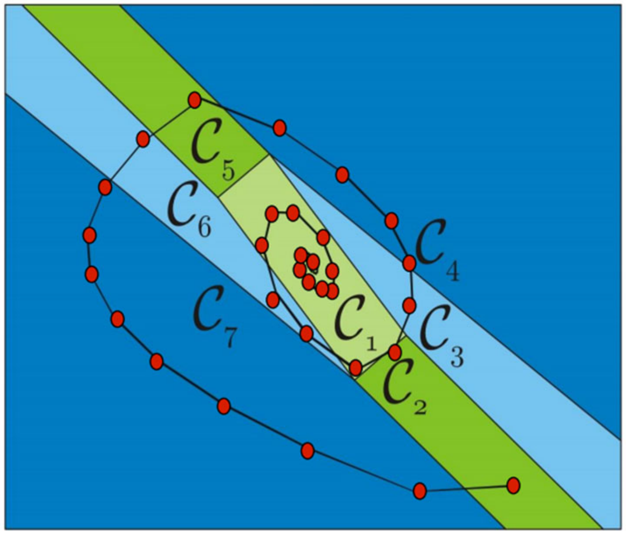
[1] Borrelli, F; Bemporad, A; Morari, M. Predictive Control for linear and hybrid systems, Cambridge University Press, 2017.

PWA – Piecewise Affine Systems

- Defined by partitioning the space of states and inputs into polyhedral regions, where each region has different linear state and output equations
- Let there be s different (non-overlapping) regions, where each region is defined as
$$C_i = \left\{ \begin{bmatrix} x \\ u \end{bmatrix} \in \mathbb{R}^{n+m} \mid H_i x + J_i u \le K_i \right\}$$
- Let the mode of operation i_k correspond to the region to the system operates in at time step k
$$i_k \in I = \{1, 2, \ldots, s\}$$
- Then the PWA model of the system is

$$x_{k+1} = A_{i_k} x_k + B_{i_k} u_k + f_{i_k}$$
$$y_k = C_{i_k} x_k + D_{i_k} u_k + g_{i_k}$$
$$i_k \text{ such that } H_{i_k} x_k + J_{i_k} u_k \le K_{i_k}$$

[1]

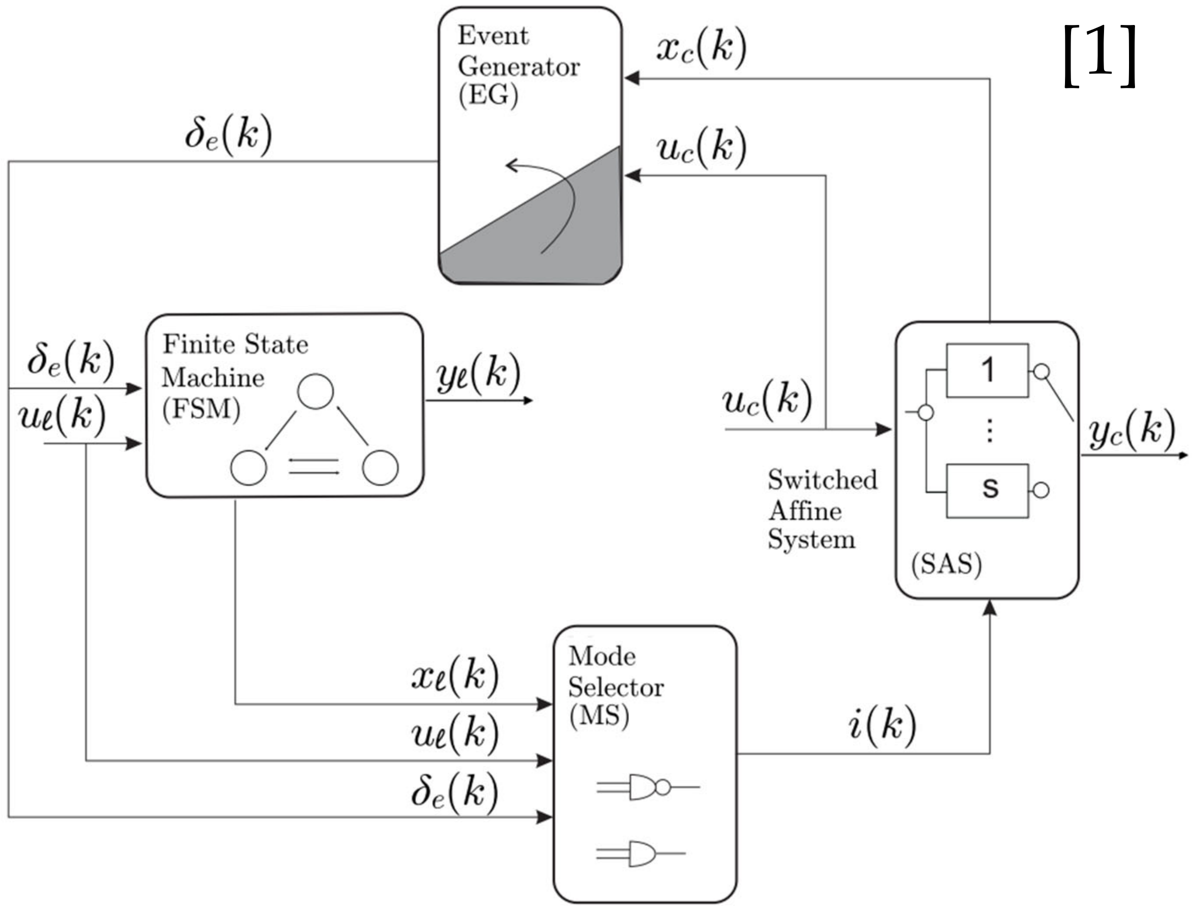


Great way to represent closed-loop system under explicit MPC

[1] Lecture notes by Alberto Bemporad
http://cse.lab.imtlucca.it/~bemporad/teaching/mpc/imt/4-hybrid_models.pdf

DHA – Discrete Hybrid Automata

- Continuous dynamics represented by a Switched Affine System (SAS)
- Discrete dynamics represented by a Finite State Machine (FSM)
- Connected by a Mode Selector and Event Generator



[1]

[1] Borrelli, F; Bemporad, A; Morari, M. Predictive Control for linear and hybrid systems, Cambridge University Press, 2017.

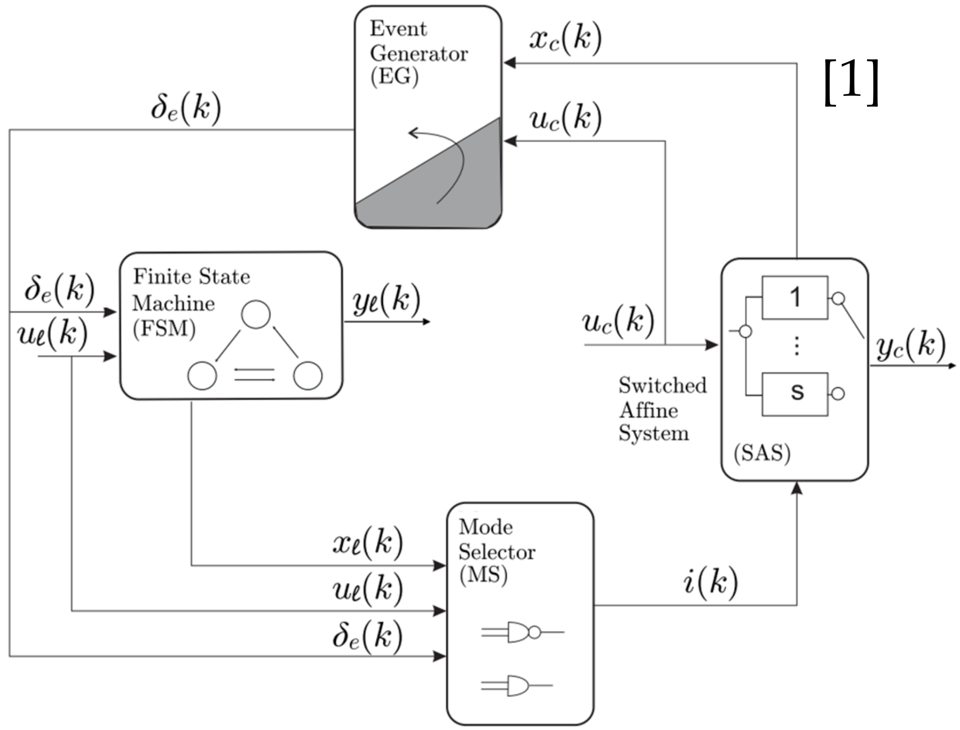
DHA – Discrete Hybrid Automata (cont.)

- Switched Affine System (SAS)
  - Similar to PWA

$$x_{k+1} = A_{i_k} x_k + B_{i_k} u_k + f_{i_k}$$

$$y_k = C_{i_k} x_k + D_{i_k} u_k + g_{i_k}$$

  - But now the mode i_k is an input to the system from the Mode Selector



[1]

- Event Generator
  - Triggers events (represented by a binary vector $\delta_e(k) \in \{0,1\}^{n_e}$) based on linear inequalities on states and inputs of SAS
- Finite State Machine
  - Discrete dynamic process with logic determining Boolean state update
- Mode Selector – Determines active mode for SAS at each time step

[1] Borrelli, F; Bemporad, A; Morari, M. Predictive Control for linear and hybrid systems, Cambridge University Press, 2017.

MLD – Mixed Logical Dynamical

- We will see that the logical relationships of a DHA can be converted into mixed-integer linear inequalities, resulting in an MLD system

$$x_{k+1} = Ax_k + B_1 u_k + B_2 \delta_k + B_3 z_k + B_5$$

$$y_k = Cx_k + D_1 u_k + D_2 \delta_k + D_3 z_k + D_5$$

$$E_2 \delta_k + E_3 z_k \leq E_1 u_k + E_4 x_k + E_5$$

$x_k \in \mathbb{R}^{n_c} \times \{0,1\}^{n_l}$ - combination of continuous and binary states

$u_k \in \mathbb{R}^{m_c} \times \{0,1\}^{m_l}$ - combination of continuous and binary inputs

$y_k \in \mathbb{R}^{p_c} \times \{0,1\}^{p_l}$ - combination of continuous and binary outputs

$\delta_k \in \{0,1\}^{r_l}$ - auxiliary binary variables

$z_k \in \{0,1\}^{r_c}$ - auxiliary continuous variables

- Extremely general and expressive

Example – PWA to MLD[1]

- Consider the system

$$x_{k+1} = \begin{cases} 0.8x_k + u_k & if \;\; x(k) \geq 0 \\ -0.8x_k + u_k & if \;\; x(k) < 0 \end{cases}$$

- Note that this is the same as the nonlinear system $x_{k+1} = 0.8|x_k| + u_k$
- Strict inequalities can be handled by introducing a small number (e.g. the machine precision, eps in MATLAB)

$$x_{k+1} = \begin{cases} 0.8x_k + u_k & if \;\; x(k) \geq 0 \\ -0.8x_k + u_k & if \;\; x(k) \leq -\varepsilon \end{cases} \quad \begin{matrix} \text{Mode 1} \\ \text{Mode 2} \end{matrix}$$

- Introduce an event (or mode) variable

$$\delta_k \in \{0,1\} \qquad \begin{matrix} \delta_k = 1 \Leftrightarrow x_k \geq 0 & \text{(Mode 1)} \\ \delta_k = 0 \Leftrightarrow x_k \leq -\varepsilon & \text{(Mode 2)} \end{matrix}$$

[1] Borrelli, F; Bemporad, A; Morari, M. Predictive Control for linear and hybrid systems, Cambridge University Press, 2017.

Example – PWA to MLD[1] (cont.)

$$x_{k+1} = \begin{cases} 0.8x_k + u_k & \text{if } x(k) \geq 0 \\ -0.8x_k + u_k & \text{if } x(k) \leq -\varepsilon \end{cases}$$

- Use big-M technique

$$\delta_k = 1 \Leftrightarrow x_k \geq 0 \quad \text{(Mode 1)}$$

$$\delta_k = 0 \Leftrightarrow x_k \leq -\varepsilon \quad \text{(Mode 2)}$$

  - Introduce large number M (large depends on the problem)

$$x_k \geq -M(1-\delta_k) \qquad \delta_k = 1 \Rightarrow x_k \geq -M(1-1) = 0$$

$$x_k \leq -\varepsilon + M\delta_k \qquad x_k \geq 0 \Rightarrow 0 \leq -\varepsilon + M\delta_k \Rightarrow \delta_k = 1$$

  - M is used to relax (remove) inequality constraint corresponding to inactive mode

[1] Borrelli, F; Bemporad, A; Morari, M. Predictive Control for linear and hybrid systems, Cambridge University Press, 2017.

Example – PWA to MLD[1] (cont.)

- Now look at the dynamics

$$x_{k+1} = \begin{cases} 0.8x_k + u_k & if \;\; x(k) \geq 0 \\ -0.8x_k + u_k & if \;\; x(k) \leq -\varepsilon \end{cases}$$

$$x_{k+1} = \underbrace{(0.8x_k + u_k)\delta_k}_{\text{Mode 1}} + \underbrace{(-0.8x_k + u_k)(1-\delta_k)}_{\text{Mode 2}}$$

$$\delta_k = 1 \Leftrightarrow x_k \geq 0 \quad \text{(Mode 1)}$$

$$\delta_k = 0 \Leftrightarrow x_k \leq -\varepsilon \quad \text{(Mode 2)}$$

$$x_{k+1} = 0.8\delta_k x_k - 0.8x_k + 0.8\delta_k x_k + \delta_k u_k + u_k - \delta_k u_k$$

$$x_{k+1} = 1.6\delta_k x_k - 0.8x_k + u_k$$

$$x_k \geq -M(1-\delta_k)$$

$$x_k \leq -\varepsilon + M\delta_k$$

- Introduce auxiliary variable $z_k = \delta_k x_k$

$$x_{k+1} = 1.6z_k - 0.8x_k + u_k$$ Dynamics are linear now!

- Can write $z_k = \delta_k x_k$ in terms of linear inequalities

$z_k = \delta_k x_k$

$if \;\; \delta_k = 0, then \;\; z_k = 0 \;\Rightarrow\;$ $z_k \leq M\delta_k$, $z_k \geq -M\delta_k$ — Constraints "go away" if $\delta_k = 1$

$if \;\; \delta_k = 1, then \;\; z_k = x_k \;\Rightarrow\;$ $z_k \leq x_k + M(1-\delta_k)$, $z_k \geq x_k - M(1-\delta_k)$ — Constraints "go away" if $\delta_k = 0$

[1] Borrelli, F; Bemporad, A; Morari, M. Predictive Control for linear and hybrid systems, Cambridge University Press, 2017.

Example – PWA to MLD[1] (cont.)

- Summary

PWA

$$x_{k+1} = \begin{cases} 0.8x_k + u_k & if \;\; x(k) \geq 0 \\ -0.8x_k + u_k & if \;\; x(k) \leq -\varepsilon \end{cases}$$

$$x_{k+1} = 1.6z_k - 0.8x_k + u_k$$

$$x_k \geq -M\left(1 - \delta_k\right)$$

$$x_k \leq -\varepsilon + M\delta_k$$

$$z_k \leq M\delta_k$$

$$z_k \geq -M\delta_k$$

$$z_k \leq x_k + M\left(1 - \delta_k\right)$$

$$z_k \geq x_k - M\left(1 - \delta_k\right)$$

MLD

$$x_{k+1} = Ax_k + B_1u_k + B_2\delta_k + B_3z_k + B_5$$

$$y_k = Cx_k + D_1u_k + D_2\delta_k + D_3z_k + D_5$$

$$E_2\delta_k + E_3z_k \leq E_1u_k + E_4x_k + E_5$$

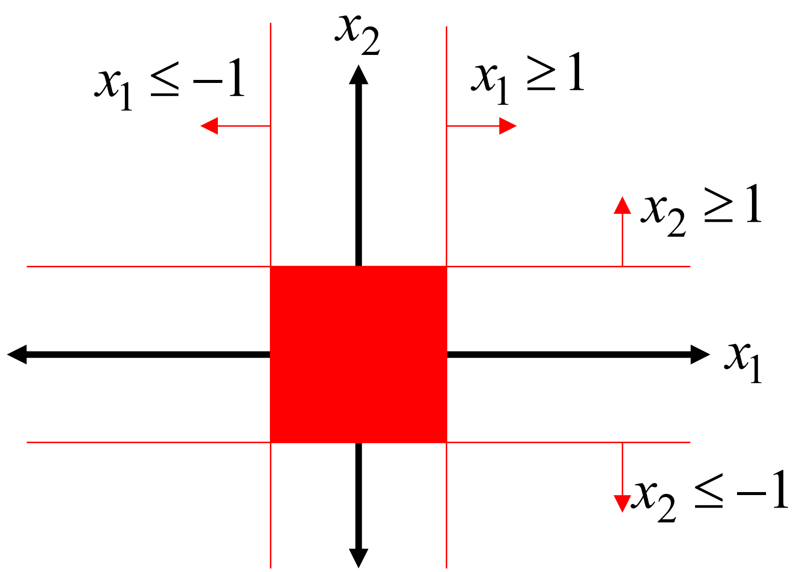
[1] Borrelli, F; Bemporad, A; Morari, M. Predictive Control for linear and hybrid systems, Cambridge University Press, 2017.

Logic into Mixed-Integer Inequalities

- As we have seen, we will often need to convert logic operations into the equivalent mixed-integer inequalities
- Define the binary variables $\delta_1, \delta_2 \in \{0,1\}$
- Recast the Boolean operators into the equivalent linear constraints
  - AND $\quad \delta_1 \wedge \delta_2 \iff \delta_1 = 1, \delta_2 = 1 \quad$ or $\quad \delta_1 + \delta_2 \geq 2$
  - OR $\quad \delta_1 \vee \delta_2 \iff \delta_1 + \delta_2 \geq 1$
  - NOT $\quad \sim \delta_1 \iff \delta_1 = 0$
  - XOR $\quad \delta_1 \oplus \delta_2 \iff \delta_1 + \delta_2 = 1$
  - IMPLY $\quad \delta_1 \rightarrow \delta_2 \iff \delta_1 \leq \delta_2$
  - IFF $\quad \delta_1 \leftrightarrow \delta_2 \iff \delta_1 = \delta_2$

Nonconvex Sets and Obstacle Avoidance

- For robot control applications, it is common to ask the robot to avoid obstacles and other robots.
- In an MPC framework, the set of feasible solutions is no longer convex
  - There is a hole in it corresponding to the obstacle
- Binary variables can be used to represent this nonconvex set using linear inequality constraints



$$\delta_1 = 1 \rightarrow x_1 \ge 1 \quad \Rightarrow \quad x_1 \ge 1 - M(1-\delta_1)$$

$$\delta_2 = 1 \rightarrow x_1 \le -1 \quad \Rightarrow \quad x_1 \le -1 + M(1-\delta_2)$$

$$\delta_3 = 1 \rightarrow x_2 \ge 1 \quad \Rightarrow \quad x_2 \ge 1 - M(1-\delta_3)$$

$$\delta_4 = 1 \rightarrow x_2 \le -1 \quad \Rightarrow \quad x_2 \le -1 + M(1-\delta_4)$$

- Need at least one to be true $\quad \delta_1 + \delta_2 + \delta_3 + \delta_4 \ge 1$
- Could do this with only 2 binary variables [1]

[1] S. C. Pinto and R. J.M. Afonso. Risk Constrained Navigation Using MILP-MPC Formulation, IFAC-PapersOnLine, 2017.

Hybrid MPC using MLDs

- Our standard finite-time optimal control problem can be written using an MLD model

$$J_0^*(x_0) = \min_{U_0,\Delta_0,Z_0} \sum_{k=0}^{N-1} x_k^T Q x_k + u_k^T R u_k + x_N^T P x_N$$

$$s.t. \quad \forall k \in \{0,1,\ldots,N-1\}$$

$$x_{k+1} = Ax_k + B_1 u_k + B_2 \delta_k + B_3 z_k + B_5,$$

$$y_k = Cx_k + D_1 u_k + D_2 \delta_k + D_3 z_k + D_5,$$

$$E_2 \delta_k + E_3 z_k \le E_1 u_k + E_4 x_k + E_5,$$

$$x_0 = x(0)$$

$$U_0 = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}, \quad \Delta_0 = \begin{bmatrix} \delta_0 \\ \delta_1 \\ \vdots \\ \delta_{N-1} \end{bmatrix}, \quad Z_0 = \begin{bmatrix} z_0 \\ z_1 \\ \vdots \\ z_{N-1} \end{bmatrix}$$

Hybrid MPC using MLDs (cont.)

$$
\begin{aligned}
J_0^*(x_0) = \min_{U_0,\Delta_0,Z_0} & \sum_{k=0}^{N-1} x_k^T Q x_k + u_k^T R u_k + x_N^T P x_N \\
s.t. \quad & \forall k \in \{0,1,\ldots,N-1\} \\
& x_{k+1} = A x_k + B_1 u_k + B_2 \delta_k + B_3 z_k + B_5, \\
& y_k = C x_k + D_1 u_k + D_2 \delta_k + D_3 z_k + D_5, \\
& E_2 \delta_k + E_3 z_k \le E_1 u_k + E_4 x_k + E_5, \\
& x_0 = x(0)
\end{aligned}
$$

- Note that since our state update equation is linear, we can rewrite future states just like we did for the linear system case

$$
x_k = A^k x_0 + \sum_{j=0}^{k-1} A^j \left(B_1 u_{k-1-j} + B_2 \delta_{k-1-j} + B_3 z_{k-1-j} + B_5 \right)
$$

- Therefore, we can cast it into the more general Mixed-Integer Quadratic Programming (MIQP) problem

$$
\begin{aligned}
J_0^*(x_0) = \min_{\xi} \; & \frac{1}{2}\xi^T H \xi + x(0)^T F^T \xi + x(0)^T Y x(0) \\
s.t. \quad & G\xi \le W + Sx(0)
\end{aligned}
$$

- With a mixture of continuous and binary variables $\xi = \begin{bmatrix} U_0 \\ \Delta_0 \\ Z_0 \end{bmatrix}$

Example – Double Integrator Vehicle

- Consider the 2 dimensional double-integrator vehicle with the following states and inputs

x_1 = position in x direction
x_2 = velocity in x direction
x_3 = position in y direction
x_4 = velocity in y direction

u_1 = force in x direction
u_2 = force in y direction

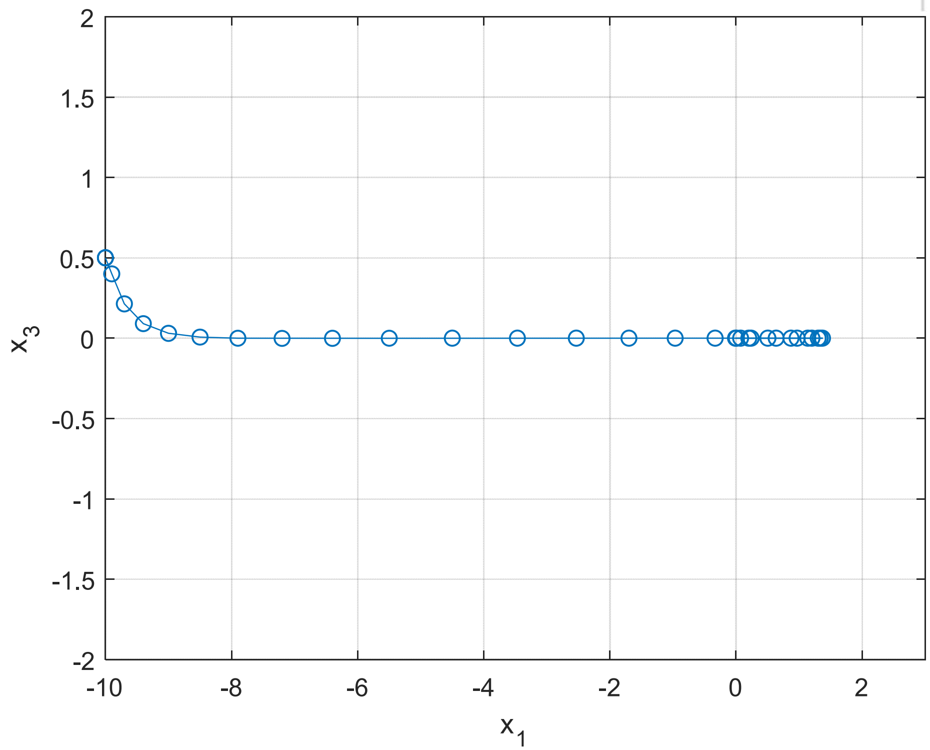
$$x_{k+1} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} u_k \qquad -0.1 \le u_k \le 0.1$$

```
% Double Integrator in 2D
n = 4;
m = 2;
A = [1 1 0 0; 0 1 0 0; 0 0 1 1; 0 0 0 1];
B = [0 0; 1 0; 0 0; 0 1];
x0 = [-10; 0; 0.5; 0];
Q = eye(n);
R = eye(m);
```

Example – Double Integrator Vehicle (cont.)

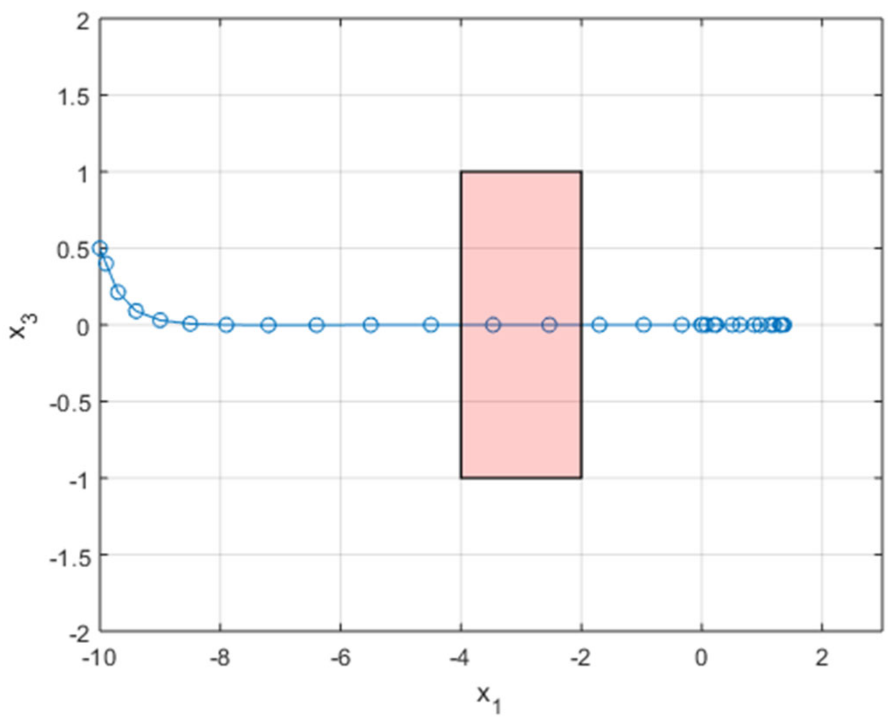
- Standard QP MPC formulation

```
%% Controller formulation
N = 10;
u_ = sdpvar(repmat(m,1,N),repmat(1,1,N));
x_ = sdpvar(repmat(n,1,N+1),repmat(1,1,N+1));

constraints = [];
objective = 0;
for k = 1:N
    objective = objective + x_{k}'*Q*x_{k} + u_{k}'*R*u_{k};
    constraints = [constraints, x_{k+1} == A*x_{k} + B*u_{k}];
    constraints = [constraints, -0.1 <= u_{k} <= 0.1];
end

controller = optimizer(constraints,objective,sdpsettings('solver','gurobi'),x_{1},u_{1});
```



Example – Double Integrator Vehicle (cont.)

- Add an obstacle



$$\delta_1 = 1 \rightarrow x_1 \geq -2 \quad \Rightarrow \quad x_1 \geq -2 - M\left(1-\delta_1\right)$$

$$\delta_2 = 1 \rightarrow x_1 \leq -4 \quad \Rightarrow \quad x_1 \leq -4 + M\left(1-\delta_2\right)$$

$$\delta_3 = 1 \rightarrow x_3 \geq 1 \quad \Rightarrow \quad x_3 \geq 1 - M\left(1-\delta_3\right)$$

$$\delta_4 = 1 \rightarrow x_3 \leq -1 \quad \Rightarrow \quad x_3 \leq -1 + M\left(1-\delta_4\right)$$

$$\delta_1 + \delta_2 + \delta_3 + \delta_4 \geq 1$$

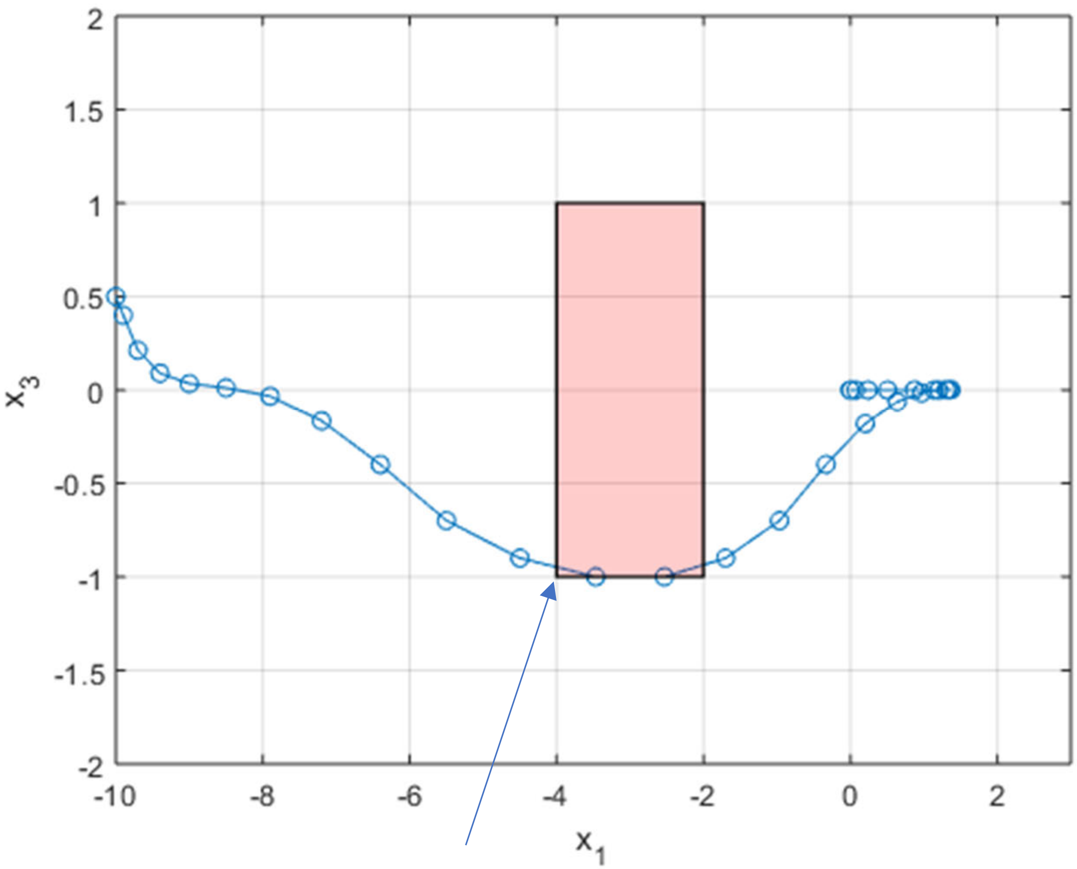
Example – Double Integrator Vehicle (cont.)

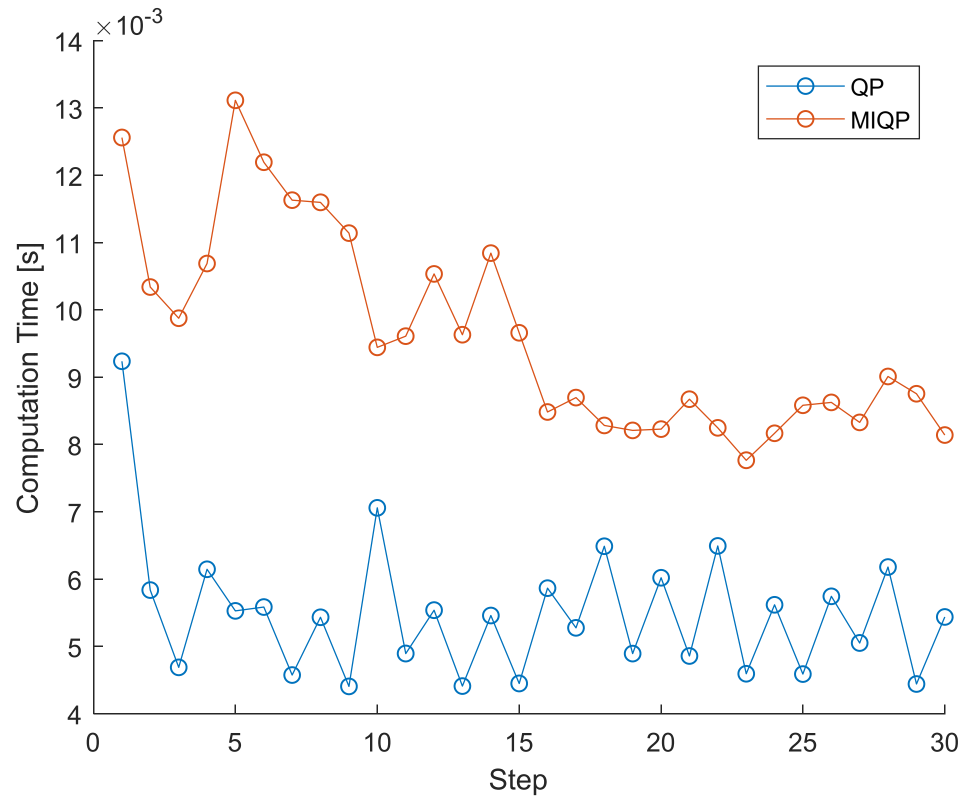
- MIQP MPC formulation

```
%% Controller formulation
N = 10;
u_ = sdpvar(repmat(m,1,N),repmat(1,1,N));
x_ = sdpvar(repmat(n,1,N+1),repmat(1,1,N+1));
b_ = binvar(repmat(4,1,N),repmat(1,1,N));
bigM = 20;

constraints = [];
objective = 0;
for k = 1:N
    objective = objective + x_{k}'*Q*x_{k} + u_{k}'*R*u_{k};
    constraints = [constraints, x_{k+1} == A*x_{k} + B*u_{k}];
    constraints = [constraints, -0.1 <= u_{k} <= 0.1];
    constraints = [constraints, x_{k}(1) >= -2 - bigM*(1-b_{k}(1))];
    constraints = [constraints, x_{k}(1) <= -4 + bigM*(1-b_{k}(2))];
    constraints = [constraints, x_{k}(3) >=  1 - bigM*(1-b_{k}(3))];
    constraints = [constraints, x_{k}(3) <= -1 + bigM*(1-b_{k}(4))];
    constraints = [constraints, sum(b_{k}) >= 1];
end

controller = optimizer(constraints,objective,sdpsettings('solver','gurobi'),x_{1},u_{1});
```

Example – Double Integrator Vehicle (cont.)

- MIQP MPC formulation





Can fix this "corner cutting" or "step over" by enlarging obstacle [1]

4x10 – states
2x10 – inputs
4x10 – binary variables

[1] Y. Kuwata, Real-time Trajectory Design for Unmanned Aerial Vehicles using Receding Horizon Control, M.S. Thesis, MIT, 2003.