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

Optimal Control for Hybrid Systems: General Formulation

Consider the CFTOC problem:

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

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

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:

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

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 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 & bound method for MIQP

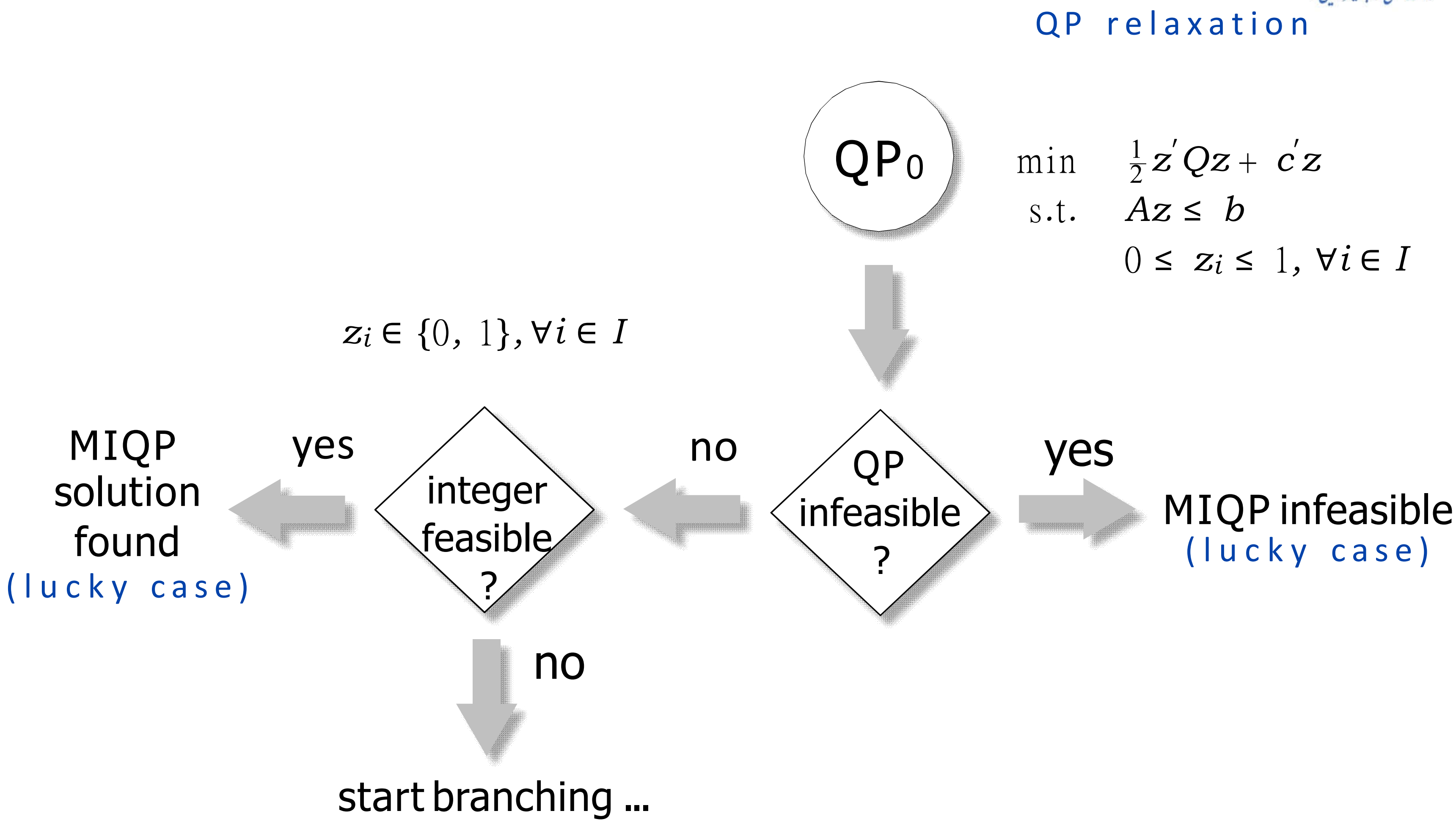
(Dakin, 1965)

- We want to solve the following MIQP

$$\begin{array}{lll} \min & V(z) \triangleq \frac{1}{2}z'Qz + c'z & z \in \mathbb{R}^n \\ \text{s.t.} & Az \leq b & Q = Q' \succeq 0 \\ & z_i \in \{0,1\},\ \forall i \in I & I \subseteq \{1,\ldots,n\} \end{array}$$

- **Branch & Bound (B&B)** is the simplest (and most popular) approach to solve the problem to optimality
- **Key idea:**
  - for each binary variable $z_i, i \in I$, either set $z_i = 0$, or $z_i = 1$, or $z_i \in [0,1]$
  - solve the corresponding **QP relaxation** of the MIQP problem
  - use QP result to decide the next combination of fixed/relaxed variables

Branch & bound method for MIQP

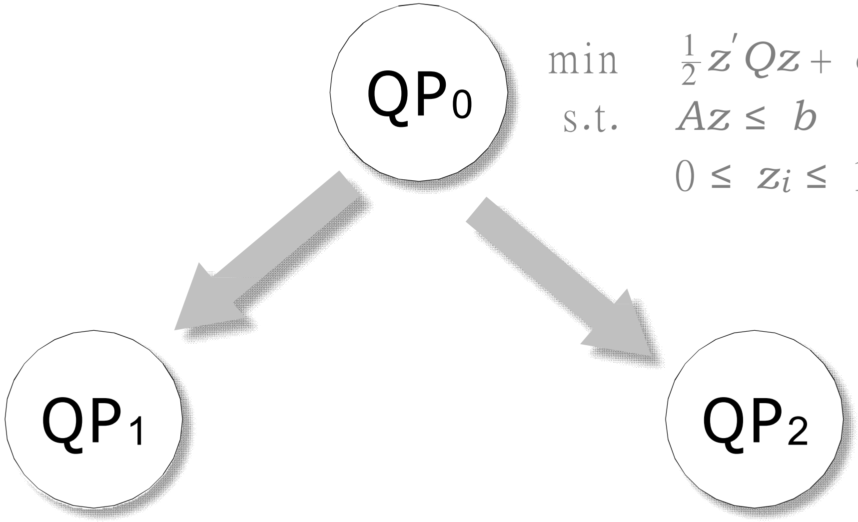


Branch & bound method for MIQP

- **Branching rule**: pick the index i such that z_i is closest to $\frac{1}{2}$ (max fractional part)

  (Breu, Burdet, 1974)

- Solve two new QP relaxations



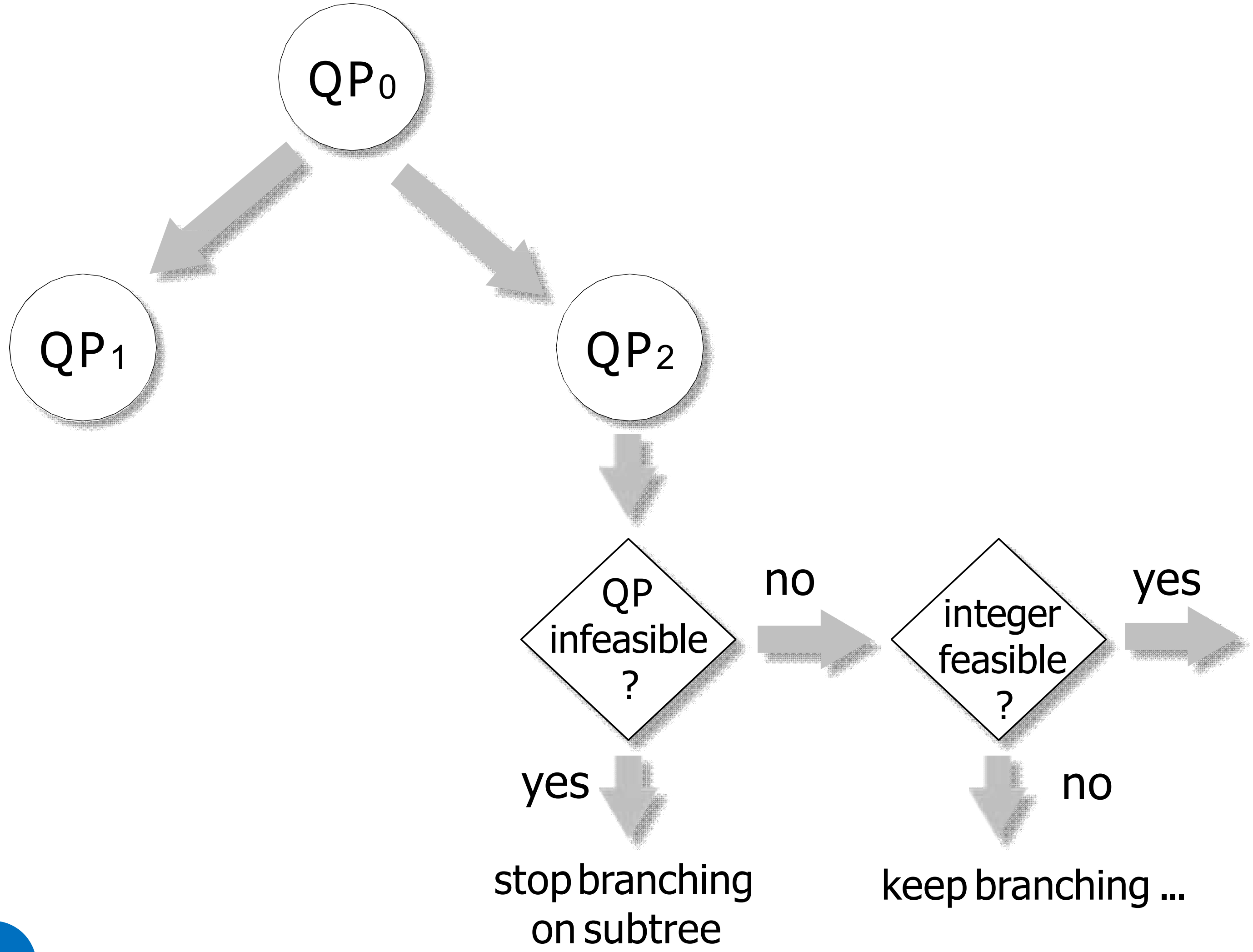
$$\begin{aligned} \min \quad & \tfrac{1}{2} z' Q z + c' z \\ \text{s.t.} \quad & A z \leq b \\ & 0 \leq z_i \leq 1, \ \forall i \in I \end{aligned}$$

$$\begin{aligned} \min \quad & \tfrac{1}{2} z' Q z + c' z \\ \text{s.t.} \quad & A z \leq b \\ & z_i = 0 \\ & 0 \leq z_j \leq 1, \ \forall j \in I, \ j \neq i \end{aligned}$$

$$\begin{aligned} \min \quad & \tfrac{1}{2} z' Q z + c' z \\ \text{s.t.} \quad & A z \leq b \\ & z_i = 1 \\ & 0 \leq z_j \leq 1, \ \forall j \in I, \ j \neq i \end{aligned}$$

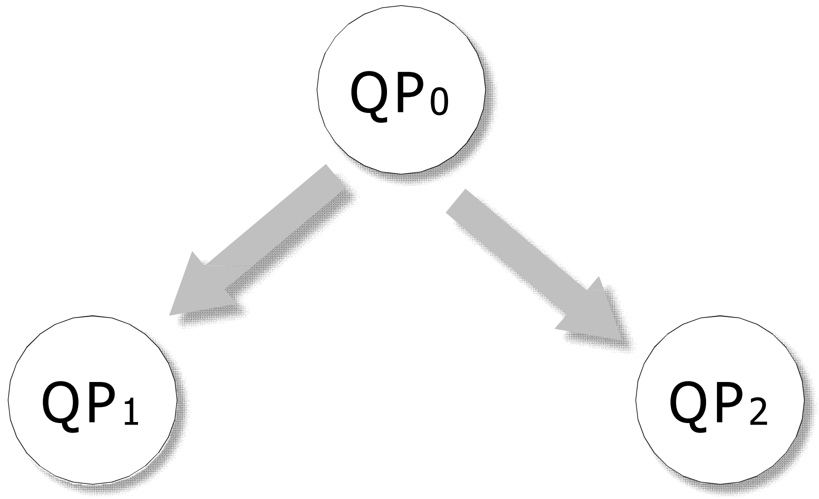
- Possibly exploit **warm starting** from QP_0 when solving new relaxations QP_1 and QP_2

Branch & bound method for MIQP

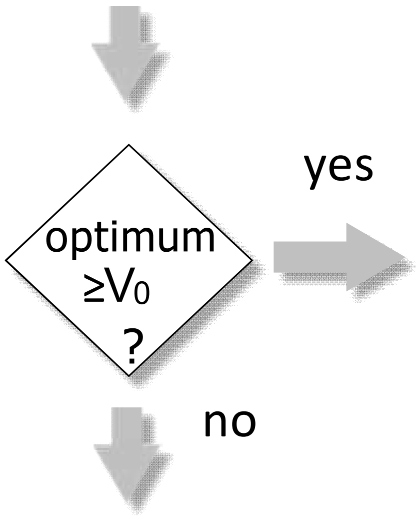


update upper bound $V_0 \geq V^*$ on MIQP solution

Branch & bound method for MIQP



The cost V_0 of the best integer-feasible solution found so fare gives an upper bound $V_0 \geq V^*$ on MIQP solution



stop branching
(adding further equality constraints can only increase theoptimal cost)

keep branching ...

Branch & bound method for MIQP

- While solving the QP relaxation, if the **dual cost** is available it gives a **lower bound** to the solution of the relaxed problem

- The QP solver can be stopped whenever the dual cost ≥ V_0 !

  This may save a lot of computations

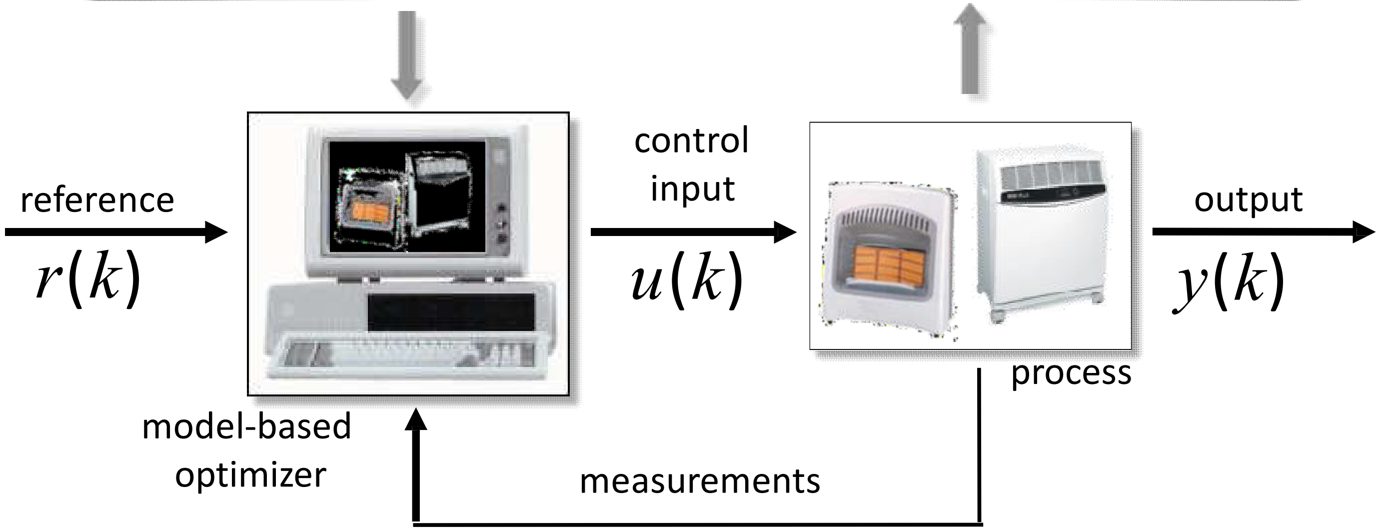
- When no further branching is possible, either the MIQP problem is recognized infeasible or the optimal solution z^* has been found

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

Hybrid Model Predictive Control

$$
\begin{aligned}
x(k+1) &= Ax(k) + B_1 u(k) + B_2 \delta(k) + B_3 z(k) + B_5 \\
y(k) &= Cx(k) + D_1 u(k) + D_2 \delta(k) + D_3 z(k) + D_5 \\
E_2 \delta(k) &+ E_3 z(k) \leq E_4 x(k) + E_1 u(k) + E_5
\end{aligned}
$$



Use a hybrid dynamical model of the process to predict its future evolution and choose the "best" control action

MIQP formulation of Hybrid MPC

(Bemporad, Morari, 1999)

- Finite-horizon optimal control problem (regulation)

$$\min \sum_{k=0}^{N-1} y_k' Q y_k + u_k' R u_k$$

$$\text{s.t.} \begin{cases} x_{k+1} &= Ax_k + B_1 u_k + B_2 \delta_k + B_3 z_k + B_5 \\ y_k &= Cx_k + D_1 u_k + D_2 \delta_k + D_3 z_k + D_5 \\ E_2 \delta_k &+ E_3 z_k \leq E_4 x_k + E_1 u_k + E_5 \\ x_0 &= x(t) \end{cases}$$

$Q = Q' \succ 0, R = R' \succ 0$

- Treat u_k, δ_k, z_k as free decision variables, $k = 0, \ldots, N-1$

- Predictions can be constructed **exactly as in the linear case**

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

MIQP formulation of Hybrid MPC

(Bemporad, Morari, 1999)

- After substituting x_k, y_k the resulting optimization problem becomes the following Mixed-Integer Quadratic Programming (MIQP) problem

$$\begin{aligned} \min_{\xi} \quad & \frac{1}{2}\xi' H \xi + x'(t) F' \xi + \frac{1}{2} x'(t) Y x(t) \\ \text{s.t.} \quad & G\xi \leq W + Sx(t) \end{aligned}$$

- The optimization vector $\xi = [u_0, \ldots, u_{N-1}, \delta_0, \ldots, \delta_{N-1}, z_0, \ldots, z_{N-1}]$ has mixed real and binary components

$$\begin{aligned} & u_k \in \mathbb{R}^{m_c} \times \{0,1\}^{m_b} \\ & \delta_k \in \{0,1\}^{r_b} \\ & z_k \in \mathbb{R}^{r_c} \end{aligned} \quad \Longrightarrow \quad \xi \in \mathbb{R}^{N(m_c+r_c)} \times \{0,1\}^{N(m_b+r_b)}$$

Hybrid MPC for reference tracking

- Consider the more general set-point tracking problem

$$\begin{aligned} \min_\xi \quad & \sum_{k=0}^{N-1} \|y_k - r\|_Q^2 + \|u_k - u_r\|_R^2 \\ & +\sigma \left(\|x_k - x_r\|_2^2 + \|\delta_k - \delta_r\|_2^2 + \|z_k - z_r\|_2^2\right) \\ \text{s.t.} \quad & \text{MLD model equations} \\ & x_0 = x(t) \\ & x_N = x_r \end{aligned}$$

with $\sigma > 0$ and $\|v\|_Q^2 = v'Qv$

- The equilibrium $(x_r, u_r, \delta_r, z_r)$ corresponding to r can be obtained by solving the following mixed-integer feasibility problem

$$\begin{aligned} x_r &= Ax_r + B_1u_r + B_2\delta_r + B_3z_r + B_5 \\ r &= Cx_r + D_1u_r + D_2\delta_r + D_3z_r + D_5 \\ E_2\delta_r &+ E_3z_r \leq E_4x_r + E_1u_r + E_5 \end{aligned}$$

Closed-loop convergence

(Bemporad, Morari, 1999)

- **Theorem.** Let $(x_r, u_r, \delta_r, z_r)$ be the equilibrium corresponding to r. Assume $x(0)$ such that the MIQP problem **is feasible at time** $t = 0$. Then $\forall Q, R \succ 0, \sigma > 0$ the hybrid MPC closed-loop **converges asymptotically**

$$\lim_{t \to \infty} y(t) = r \qquad \lim_{t \to \infty} x(t) = x_r$$

$$\lim_{t \to \infty} \delta(t) = \delta_r$$

$$\lim_{t \to \infty} u(t) = u_r \qquad \lim_{t \to \infty} z(t) = z_r$$

and **all constraints are fulfilled** at each time $t \geq 0$.

- The proof easily follows from standard Lyapunov arguments (see next slide)
- **Lyapunov asymptotic stability** and **exponential stability** follows if proper terminal cost and constraints are imposed (Lazar, Heemels, Weiland, Bemporad, 2006)

Closed-loop convergence proof

- **Main idea**: Use the value function $V^*(x(t))$ as a Lyapunov function
- Let $\xi_t = [u_0^t, \ldots, u_{N-1}^t, \delta_0^t, \ldots, \delta_{N-1}^t, z_0^t, \ldots, z_{N-1}^t]$ be the optimal sequence @t
- By construction @t+1 $\bar{\xi} = [u_1^t, \ldots, u_{N-1}^t, u_r, \delta_1^t, \ldots, \delta_{N-1}^t, \delta_r, z_0^t, \ldots, z_{N-1}^t, z_r]$ is feasible, as it satisfies all MLD constraints + terminal constraint $x_N = x_r$
- The cost of $\bar{\xi}$ is $V^*(x(t)) - \|y(t) - r\|_Q^2 - \|u(t) - u_r\|_R^2$

$$-\sigma\left(\|\delta(t) - \delta_r\|_2^2 + \|z(t) - z_r\|_2^2 + \|x(t) - x_r\|_2^2\right) \geq V^*(x(t+1))$$

- $V^*(x(t))$ is monotonically decreasing and ≥ 0, so $\exists \lim_{t \to \infty} V^*(x(t)) \in \mathbb{R}$
- Hence $\|y(t) - r\|_Q^2, \|u(t) - u_r\|_R^2, \|\delta(t) - \delta_r\|_2^2, \|z(t) - z_r\|_2^2, \|x(t) - x_r\|_2^2 \to 0$
- Since $R, Q \succ 0$, $\lim_{t \to \infty} y(t) = r$ and all other variables converge. $\square$

Global optimum is not needed to prove convergence !

MILP formulation of Hybrid MPC

- Finite-horizon optimal control problem using infinity norms

$$\min_{\xi} \sum_{k=0}^{N-1} \|Qy_k\|_\infty + \|Ru_k\|_\infty$$

$$\text{s.t.} \begin{cases} x_{k+1} &=& Ax_k + B_1u_k + B_2\delta_k + B_3z_k + B_5 \\ y_k &=& Cx_k + D_1u_k + D_2\delta_k + D_3z_k + D_5 \\ E_2\delta_k &+& E_3z_k \leq E_4x_k + E_1u_k + E_5 \\ x_0 &=& x(t) \end{cases}$$

$Q \in \mathbb{R}^{m_y \times n_y}$
$R \in \mathbb{R}^{m_u \times n_u}$

- Introduce additional variables $\epsilon_k^y, \epsilon_k^u, k = 0, \ldots, N-1$

$$\begin{cases} \epsilon_k^y & \geq & \|Qy_k\|_\infty \\ \epsilon_k^u & \geq & \|Ru_k\|_\infty \end{cases} \Rightarrow \begin{cases} \epsilon_k^y & \geq & \pm Q^i y_k \\ \epsilon_k^u & \geq & \pm R^i u_k \end{cases}$$

Q^i = ith row of Q

MILP formulation of Hybrid MPC

- Finite-horizon optimal control problem using infinity norms

$$\min_{\xi} \quad \sum_{k=0}^{N-1} \|Qy_k\|_\infty + \|Ru_k\|_\infty$$

$$\text{s.t.} \quad \begin{cases} x_{k+1} &= Ax_k + B_1u_k + B_2\delta_k + B_3z_k + B_5 \\ y_k &= Cx_k + D_1u_k + D_2\delta_k + D_3z_k + D_5 \\ E_2\delta_k &+ E_3z_k \leq E_4x_k + E_1u_k + E_5 \\ x_0 &= x(t) \end{cases}$$

$Q \in \mathbb{R}^{m_y \times n_y}$

$R \in \mathbb{R}^{m_u \times n_u}$

- Introduce additional variables $\epsilon_k^y, \epsilon_k^u, k = 0, \ldots, N-1$

$$\begin{cases} \epsilon_k^y &\geq \|Qy_k\|_\infty \\ \epsilon_k^u &\geq \|Ru_k\|_\infty \end{cases} \Longrightarrow \begin{cases} \epsilon_k^y &\geq \pm Q^i y_k \\ \epsilon_k^u &\geq \pm R^i u_k \end{cases}$$

Q^i = ith row of Q

MILP formulation of Hybrid MPC

(Bemporad, Borrelli, Morari, 2000)

- After substituting x_k, y_k the resulting optimization problem becomes the following **Mixed-Integer Linear Programming (MILP)** problem

$$\begin{array}{ll} \min_{\xi} & \sum_{k=0}^{N-1} \epsilon_k^y + \epsilon_k^u \\ \text{s.t.} & G\xi \leq W + Sx(t) \end{array}$$

- $\xi = [u_0, \ldots, u_{N-1}, \delta_0, \ldots, \delta_{N-1}, z_0, \ldots, z_{N-1}, \epsilon_0^y, \epsilon_0^u, \ldots, \epsilon_{N-1}^y, \epsilon_{N-1}^u]$ is the optimization vector, with **mixed real and binary** components

$$\begin{array}{l} u_k \in \mathbb{R}^{m_c} \times \{0,1\}^{m_b} \\ \delta_k \in \{0,1\}^{r_b} \\ z_k \in \mathbb{R}^{r_c} \\ \epsilon_k^y, \epsilon_k^u \in \mathbb{R} \end{array} \quad \Rightarrow \quad \xi \in \mathbb{R}^{N(m_c+r_c+2)} \times \{0,1\}^{N(m_b+r_b)}$$

- Same approach applies to any **convex piecewise affine** stage cost

Mixed-Integer Programming solvers

- Binary constraints make Mixed-Integer Programming (MIP) a hard problem (NP-complete)
- However, excellent general purpose branch & bound / branch & cut solvers available for MILP and MIQP (Gurobi, CPLEX, FICO Xpress, GLPK, CBC, ...)
- MIQP approaches tailored to embedded hybrid MPC applications:
  - B&B + (dual) active set methods for QP

    (Leyffer, Fletcher, 1998) (Axehill, Hansson, 2006) (Bemporad, 2015) (Bemporad, Naik, 2018)
  - B&B + interior point methods: (Frick, Domahidi, Morari, 2015)
  - B&B + fast gradient projection: (Naik, Bemporad, 2017)
  - B&B + ADMM: (Stellato, Naik, Bemporad, Goulart, Boyd, 2018)

MPC for Hybrid Systems - Complexity

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

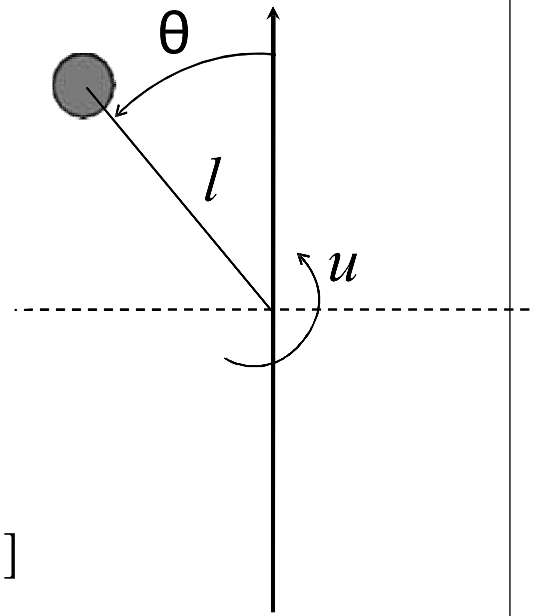
  Mixed-Integer programming is HARD

- Efficient general purpose solvers for MILP/MIQP: CPLEX, XPRESS-MP ⇒ 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—> **explicit solution of the MPC**

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

Hybrid MPC of an inverted pendulum



- **Goal**: swing the pendulum up

- **Non-convex** input constraint

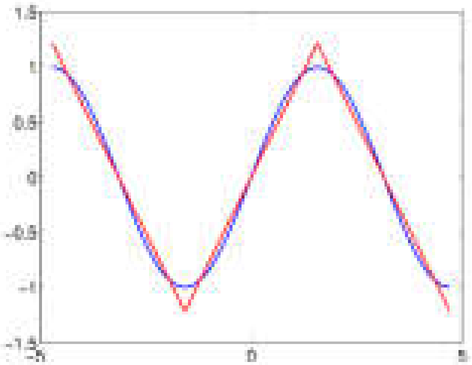
$$u \in [-\tau_{max}, -\tau_{min}] \cup \{0\} \cup [\tau_{min}, \tau_{max}]$$

- **Nonlinear** dynamical model

$$l^2 M \ddot{\theta} = Mgl \sin\theta - \beta \dot{\theta} + u$$

Hybrid MPC of an inverted pendulum

دانشگاه صنعتی خواجه نصیرالدین طوسی



- Approximate $\sin(\theta)$ as the piecewise linear function

$$\sin\theta \approx s \triangleq \begin{cases} -\alpha\theta - \gamma & \text{if} \quad \theta \le -\frac{\pi}{2} \\ \alpha\theta & \text{if} \quad |\theta| \le \frac{\pi}{2} \\ -\alpha\theta + \gamma & \text{if} \quad \theta \ge \frac{\pi}{2} \end{cases}$$

- Get optimal values for α and γ by minimizing fit error

$$\min_{\alpha} \quad \int_0^{\frac{\pi}{2}} (\alpha\theta - \sin(\theta))^2 d\theta$$

$$= \quad \frac{\theta}{2} - \frac{1}{2}\cos\theta\sin\theta - 2\alpha\sin\theta + \frac{1}{3}\alpha^2\theta^3 + 2\alpha\theta\cos\theta \Big|_0^{\frac{\pi}{2}} = \frac{1}{24}\pi^3\alpha^2 - 2\alpha + \frac{\pi}{4}$$

- Zeroing the derivative with respect to α gives $\alpha = \frac{24}{\pi^3}$

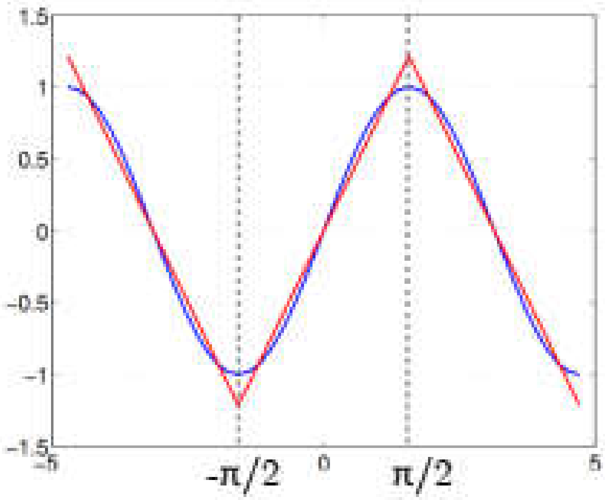
- Requiring $s = 0$ for $\theta = \pi$ gives $\gamma = \frac{24}{\pi^2}$

Hybrid MPC of an inverted pendulum

- Introduce the event variables

$$[\delta_3 = 1] \quad \leftrightarrow \quad [\theta \leq -\tfrac{\pi}{2}]$$

$$[\delta_4 = 1] \quad \leftrightarrow \quad [\theta \geq \tfrac{\pi}{2}]$$



along with the logic constraint

$$[\delta_4 = 1] \rightarrow [\delta_3 = 0]$$

- Set $s = \alpha\theta + s_3 + s_4$ with

$$s_3 = \begin{cases} -2\alpha\theta - \gamma & \text{if } \delta_3 = 1 \\ 0 & \text{otherwise} \end{cases}$$

$$s_4 = \begin{cases} -2\alpha\theta + \gamma & \text{if } \delta_4 = 1 \\ 0 & \text{otherwise} \end{cases}$$

Hybrid MPC of an inverted pendulum

- To model the constraint $u \in [-\tau_{\max}, -\tau_{\min}] \cup \{0\} \cup [\tau_{\min}, \tau_{\max}]$ introduce the auxiliary variable

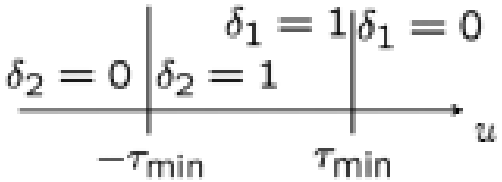
$$\tau_A = \begin{cases} u & \text{if } -\tau_{\min} \le u \le \tau_{\min} \\ 0 & \text{otherwise} \end{cases}$$

and let $u - \tau_A$ be the torque acting on the pendulum, with

$$u \in [-\tau_{\max}, \tau_{\max}]$$

- The input u has no effect on the dynamics for $u \in [-\tau_{\min}, \tau_{\min}]$. Hence, the solver will not choose values in that range if u is penalized in the MPC cost

Hybrid MPC of an inverted pendulum



- Introduce new event variables

$$[\delta_1 = 1] \quad \leftrightarrow \quad [u \leq \tau_{\min}]$$

$$[\delta_2 = 1] \quad \leftrightarrow \quad [u \geq -\tau_{\min}]$$

along with the logic constraint $[\delta_1 = 0] \rightarrow [\delta_2 = 1]$ and set

$$\tau_A = \begin{cases} u & \text{if } [\delta_1 = 1] \wedge [\delta_2 = 1] \\ 0 & \text{otherwise} \end{cases}$$

so that $u - \tau_A$ is zero in for $u \in [-\tau_{\min}, \tau_{\min}]$

Hybrid MPC of an inverted pendulum

- Set $x \triangleq \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}$, $y \triangleq \theta$ and transform into linear model

$$\begin{aligned} \frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ \frac{g}{l}\alpha & -\frac{\beta}{l^2 M} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{g}{l} & \frac{1}{l^2 M} \end{bmatrix}\begin{bmatrix} s_3 + s_4 \\ u - \tau_A \end{bmatrix} \\ y &= \begin{bmatrix} 1 & 0 \end{bmatrix} x \end{aligned}$$

- Discretize in time with sample time $T_s = 50$ ms

$$\begin{aligned} \begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} &= A\begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + B\begin{bmatrix} s_3(k) + s_4(k) \\ u(k) - \tau_A(k) \end{bmatrix} \\ y(k) &= \begin{bmatrix} 1 & 0 \end{bmatrix} x(k) \end{aligned}$$

$$A \triangleq e^{T_s A_c},\ B \triangleq \int_0^{T_s} e^{tA_c} B_c dt$$

Hybrid MPC of an inverted pendulum

```
/* Hybrid model of a pendulum

   (C) 2012 by A. Bemporad, April 2012 */

SYSTEM hyb_pendulum {

INTERFACE {
  STATE {
    REAL th    [-2*pi,2*pi];
    REAL thdot [-20,20];
          }
  INPUT {
    REAL u [-11,11];
          }
  OUTPUT{
    REAL y;
          }
  PARAMETER {
    REAL tau_min,alpha,gamma;
    REAL a11,a12,a21,a22,b11,b12,b21,b22;
  }
}

IMPLEMENTATION {
   AUX {
     REAL tauA,x3,x4;
     BOOL d1,d2,d3,d4;
   }
   AD {
     d1 = u<=tau_min;
     d2 = u>=-tau_min;
     d3 = th <= -0.5*pi;
     d4 = th >= 0.5*pi;
   }

  DA  {
    tauA = {IF d1 & d2 THEN u ELSE 0};
    x3 = {IF d3 THEN -2*alpha*th-gamma ELSE 0};
    x4 = {IF d4 THEN -2*alpha*th+gamma ELSE 0};
  }

  CONTINUOUS {
    th    = a11*th+a12*thdot+b11*(x3+x4)+b12*(u-tauA);
    thdot = a21*th+a22*thdot+b21*(x3+x4)+b22*(u-tauA);
  }

  OUTPUT {
    y = th;
  }

  MUST {
    d4->~d3;
    ~d1->d2;
  }
}
}
```

```
>> S=mld('pendulum',Ts);
```

go to demo `demos/hybrid/pendulum_init.m`

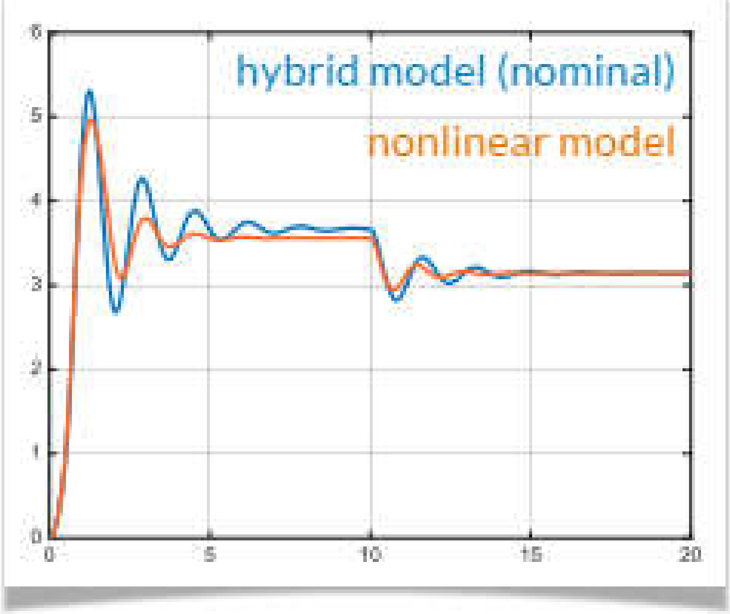
Hybrid MPC of an inverted pendulum

- Open-loop simulation from initial condition $\theta(0) = 0, \dot{\theta}(0) = 0$
- Input torque excitation

$$u(t) = \begin{cases} 2\,\mathrm{Nm} & \text{if } 0 \le t \le 10\,\mathrm{s} \\ 0 & \text{otherwise} \end{cases}$$

```
>> u0=2;
>> U=[2*ones(200,1);zeros(200,1)];
>> x0=[0;0];
```

```
>> [X,T,D,Z,Y]=sim(S,x0,U);
```



Hybrid MPC of an inverted pendulum

دانشگاه صنعتی خواجه نصیرالدین طوسی

```
>> refs.y=1;
>> refs.u=1;
>> Q.y=1;
>> Q.y=0.01;
>> Q.rho=Inf;
>> Q.norm=Inf;
>> N=5;
>> limits.umin=-10;
>> limits.umax=10;
```

- MPC cost function

$$\sum_{k=0}^{4} |y_k - r(t))| + |0.01 u_k|$$

- MPC constraints $u \in [-\tau_{\max}, \tau_{\max}]$

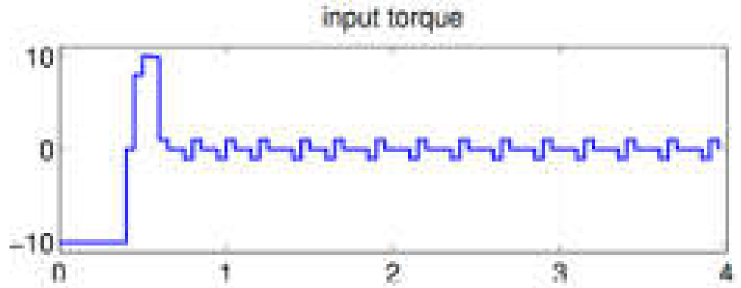
```
>> C=hybcon(S,Q,N,limits,refs);
```

```
>> C

Hybrid controller based on MLD model S <pendulum.hys> [Inf-norm]

2 state measurement(s)
1 output reference(s)
1 input reference(s)
0 state reference(s)
0 reference(s) on auxiliary continuous z-variables

55 optimization variable(s) (30 continuous, 25 binary)
155 mixed-integer linear inequalities
sampling time = 0.05, MILP solver = 'gurobi'

Type "struct(C)" for more details.
>>
```

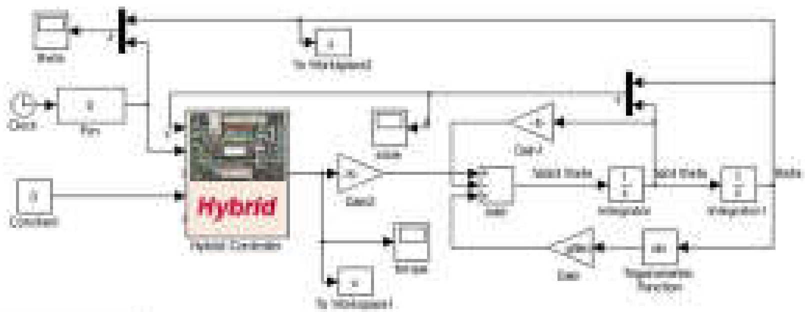
Hybrid MPC of an inverted pendulum

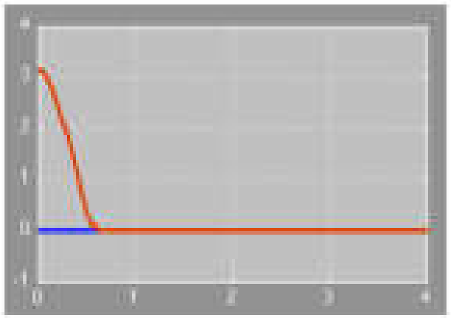
- Nominal simulation

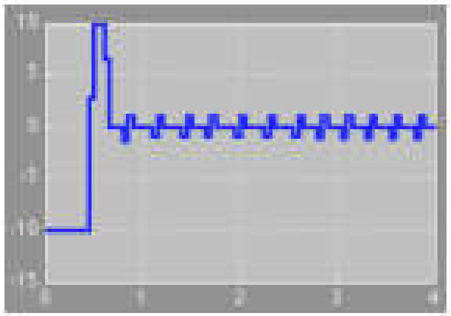
```
>> [X,U,D,Z,T,Y]=sim(C,S,r,x0,4);
```



- Nonlinear simulation

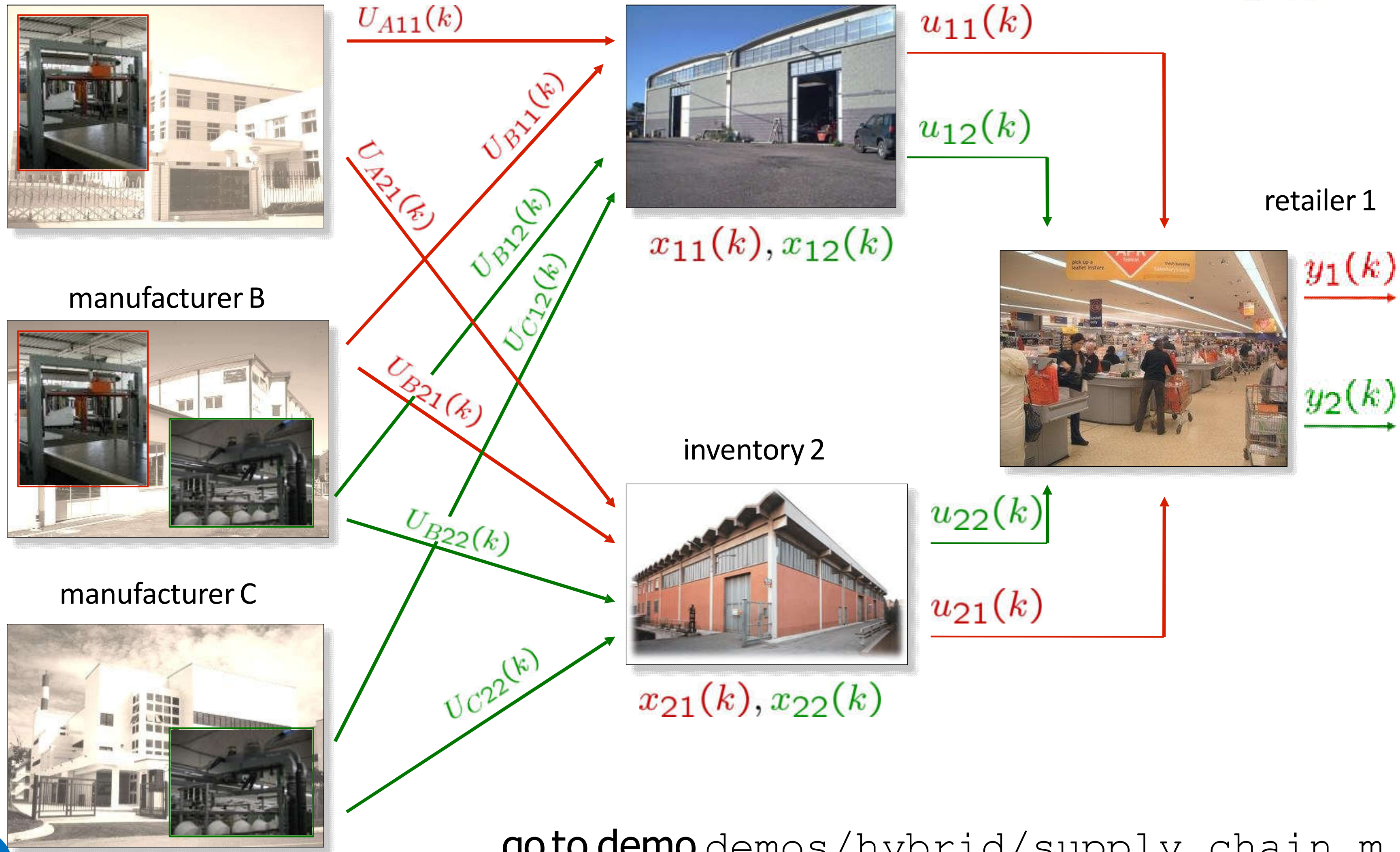






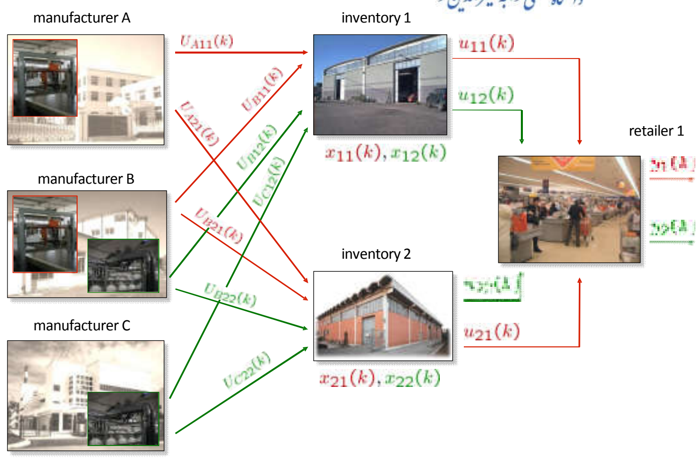
CPU time:
51 ms per time step (GLPK)
22 ms per time step (CPLEX)
25 ms (GUROBI)
(Macbook Pro 3GHz Intel Core i7)

Example in supply chain management



go to demo `demos/hybrid/supply_chain.m`

Example in supply chain management



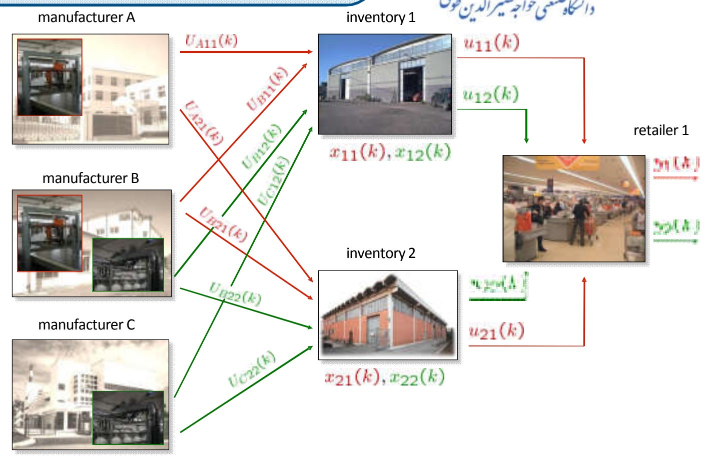
- **Continuous states**:
  $x_{ij}(k)$ =amount of j hold in inventory i at time k ($i = 1, 2, j = 1, 2$)

- **Continuous outputs**:
  $y_j(k)$ =amount of j sold at time k ($j = 1, 2$)

- **Continuous inputs**:
  $u_{ij}(k)$ =amount of j taken from inventory i at time k ($i = 1, 2, j = 1, 2$)

- **Binary inputs**:
  $U_{Xij}(k) = 1$ if manufacturer X produces and send j to inventory i at time k

Example in supply chain management



- Max capacity of inventory i:

$$0 \leq \sum_{j=1}^{2} x_{ij} \leq x_{Mi}$$

- Max transportation from inventories:

$$0 \leq u_{ij}(k) \leq u_M$$

- A product can only be sent to one inventory:

$U_{A11}(k)$ and $U_{A21}(k)$ cannot be both $=1$

$U_{B11}(k)$ and $U_{B21}(k)$ cannot be both $=1$

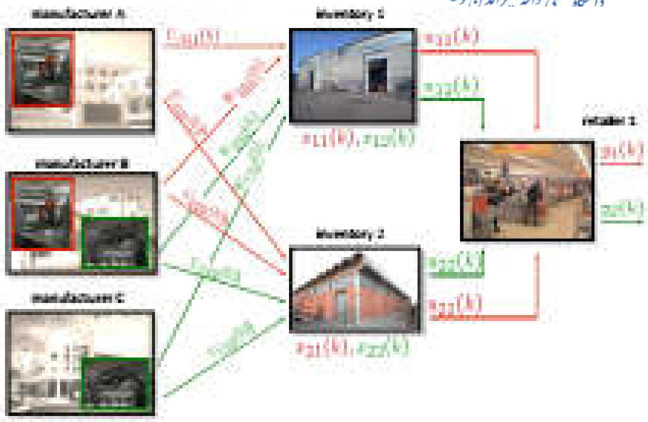
$U_{B12}(k)$ and $U_{B22}(k)$ cannot be both $=1$

$U_{C12}(k)$ and $U_{C22}(k)$ cannot be both $=1$

- A manufacturer can only produce one type of product at one time:

$[U_{B11}(k)$ or $U_{B21}(k) = 1]$, $[U_{B12}(k)$ or $U_{B22}(k) = 1]$ cannot be both true

Supply chain management - Dynamics



- Let $P_{A1}, P_{B1}, P_{B2}, P_{C2}$ = amount of product of type 1 (2) produced by A (B, C) in one time interval

- Level of inventories

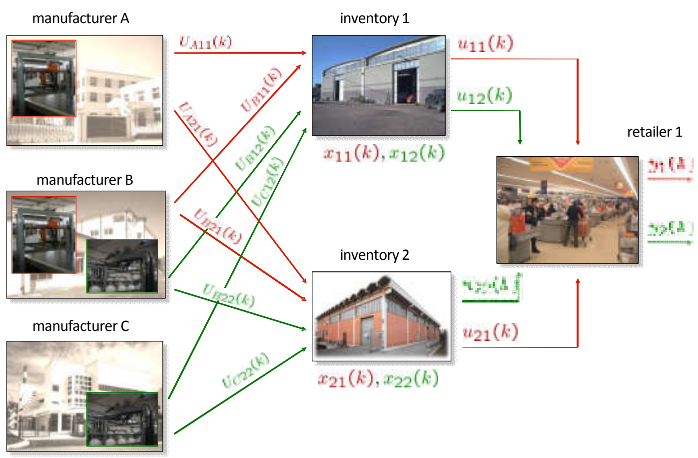
$$\begin{cases} x_{11}(k+1) &= x_{11}(k) + P_{A1}U_{A11}(k) + P_{B1}U_{B11}(k) - u_{11}(k) \\ x_{12}(k+1) &= x_{12}(k) + P_{B2}U_{B12}(k) + P_{C2}U_{C12}(k) - u_{12}(k) \\ x_{21}(k+1) &= x_{21}(k) + P_{A1}U_{A21}(k) + P_{B1}U_{B21}(k) - u_{21}(k) \\ x_{22}(k+1) &= x_{22}(k) + P_{B2}U_{B22}(k) + P_{C2}U_{C22}(k) - u_{22}(k) \end{cases}$$

- Retailer: all items requested from inventories are sold

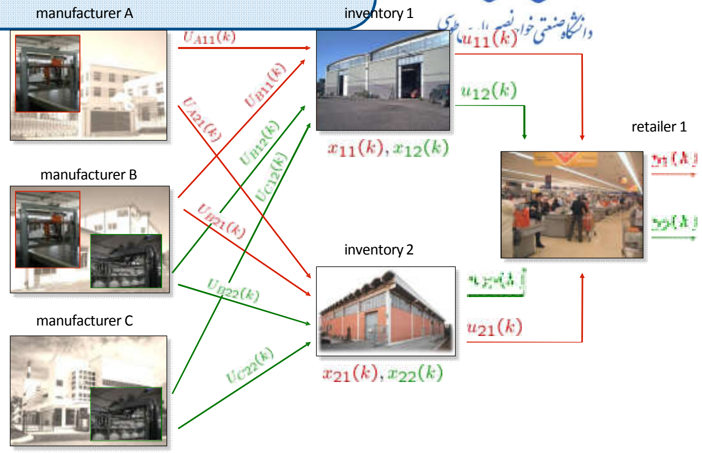
$$\begin{cases} y_1 &= u_{11} + u_{21} \\ y_2 &= u_{12} + u_{22} \end{cases}$$

Supply chain management - HYSDEL code

```
SYSTEM supply_chain{
INTERFACE {
        STATE { REAL x11   [0,10];
                REAL x12   [0,10];
                REAL x21 [0,10];
                REAL x22 [0,10]; }

        INPUT { REAL u11 [0,10];
         REAL u12 [0,10];
         REAL u21 [0,10];
         REAL u22 [0,10];
         BOOL UA11,UA21,UB11,UB12,UB21,UB22,UC12,UC22;  }

        OUTPUT {REAL y1,y2;}

        PARAMETER { REAL PA1,PB1,PB2,PC2,xM1,xM2;}
}
IMPLEMENTATION {

        AUX { REAL zA11, zB11, zB12, zC12, zA21, zB21, zB22, zC22;}


        DA {     zA11 = {IF UA11 THEN PA1 ELSE 0};
                 zB11 = {IF UB11 THEN PB1 ELSE 0};
                 zB12 = {IF UB12 THEN PB2 ELSE 0};
                 zC12 = {IF UC12 THEN PC2 ELSE 0};
                 zA21 = {IF UA21 THEN PA1 ELSE 0};
                 zB21 = {IF UB21 THEN PB1 ELSE 0};
                 zB22 = {IF UB22 THEN PB2 ELSE 0};
                 zC22 = {IF UC22 THEN PC2 ELSE 0}; }

        CONTINUOUS {x11 = x11 + zA11 + zB11 - u11;
                    x12 = x12 + zB12 + zC12 - u12;
                    x21 = x21 + zA21 + zB21 - u21;
                    x22 = x22 + zB22 + zC22 - u22;  }

        OUTPUT {    y1 = u11 + u21;
                    y2 = u12 + u22; }

         MUST {  ~(UA11 & UA21);
                  ~(UC12 & UC22);
                 ~((UB11 | UB21) & (UB12 | UB22));
                 ~(UB11 & UB21);
                 ~(UB12 & UB22);
                 x11+x12 <= xM1;
                 x11+x12 >=0;
                 x21+x22 <= xM2;
                 x21+x22 >=0; }
} }
```



Supply chain management - Objectives



- Meet customer demand as much as possible:

$$y_1 \approx r_1, \quad y_2 \approx r_2$$

- Minimize transportation costs

- Fulfill all constraints

Supply chain management - Performance index

$$\min \sum_{k=0}^{N-1} \overbrace{10(|y_{1,k} - r_1(t)| + |y_{2,k} - r_2(t)|}^{\text{penalty on demand tracking error}} +$$

$$\overbrace{4(|u_{11,k}| + |u_{12,k}|)}^{\text{shipping cost from inv. 1 to market}} \quad +$$

$$\overbrace{2(|u_{21,k}| + |u_{22,k}|)}^{\text{shipping cost from inv. 2 to market}} \quad +$$

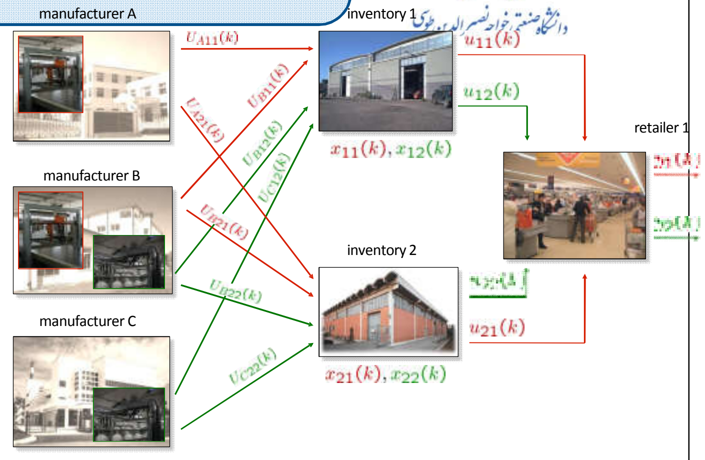
$$\overbrace{1(|U_{A11,k}| + |U_{A21,k}|)}^{\text{cost from A to inventories}} +$$

$$\overbrace{4(|U_{B11,k}| + |U_{B12,k}| + U_{B21,k}| + |U_{B22,k}|)}^{\text{cost from B to inventories}} +$$

$$\overbrace{10(|U_{C12,k}| + |U_{C22,k}|)}^{\text{cost from C to inventories}}$$

Supply chain management - Simulation setup

```
>> refs.y=[1 2];                          % weights output2 #1, #2
>> Q.y=diag([10 10]);                          % output weights
…
>> Q.norm=Inf;                                 % infinity norms
>> N=2;                                 % optimization horizon
>> limits.umin=umin;                              % constraints
>> limits.umax=umax;
>> limits.xmin=xmin;                                % xij(k)>=0
>> limits.xmax=xmax;                    % xij(k)<=xMi (redundant)
```



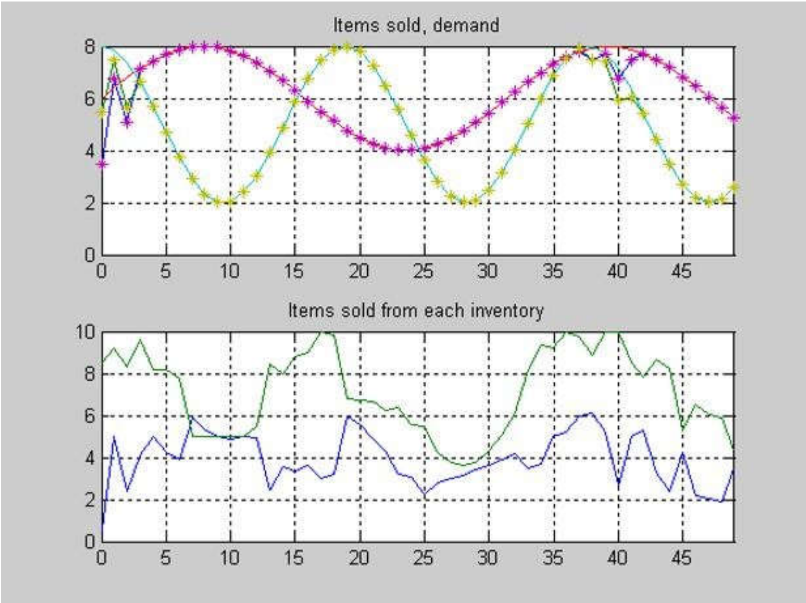

```
>> C=hybcon(S,Q,N,limits,refs);
```

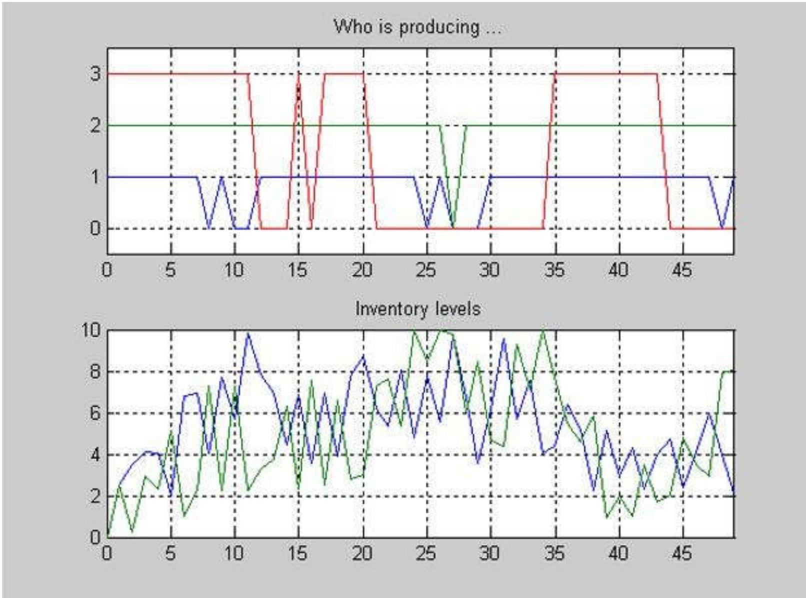
```
>> C

Hybrid controller based on MLD model S <supply_chain.hys>


[Inf-norm]

4 state measurement(s)
2 output reference(s)
12 input reference(s)
0 state reference(s)
0 reference(s) on auxiliary continuous z-variables

44 optimization variable(s) (8 continuous, 12 binary)
176 mixed-integer linear inequalities
sampling time = 1, MILP solver = 'glpk'

Type "struct(C)" for more details.
>>
```

Supply chain management - Simulation results

```
>> x0=[0;0;0;0];                                     % Initial condition
>> r.y=[6+2*sin((0:Tstop-1)'/5)                % Reference trajectories
     5+3*cos((0:Tstop-1)'/3)];
```

```
>> [XX,UU,DD,ZZ,TT]=sim(C,S,r,x0,Tstop);
```





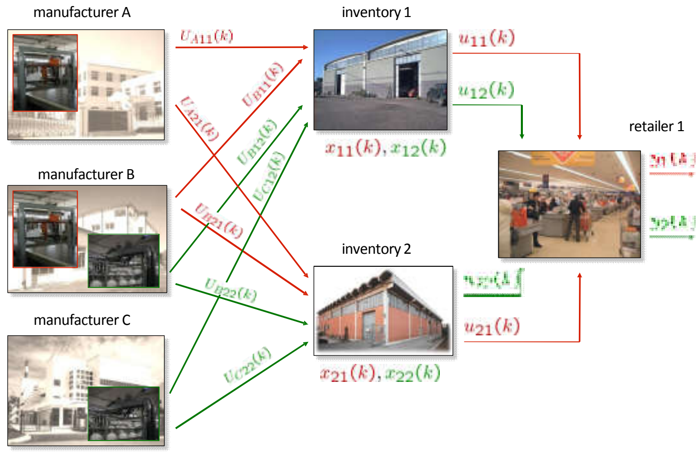
CPU time: ≈ 13 ms/sample (GLPK) or 9 ms (CPLEX) on Macbook Pro 3GHz Intel Core i7

setup

```
>> refs.y=[1 2];                    % weights output2 #1, #2
>> Q.y=diag([10 10]);               % output weights
…
>> Q.norm=Inf;                      % infinity norms
>> N=2;                             % optimization horizon
>> limits.umin=umin;                % constraints
>> limits.umax=umax;
>> limits.xmin=xmin;                % xij(k)>=0
>> limits.xmax=xmax;                % xij(k)<=xMi (redundant)
```

```
>> C=hybcon(S,Q,N,limits,refs);
```

```
 >> C

Hybrid controller based on MLD model S <supply_chain.hys>

[Inf-norm]

4 state measurement(s)
2 output reference(s)
12 input reference(s)
0 state reference(s)
0 reference(s) on auxiliary continuous z-variables

44 optimization variable(s) (8 continuous, 12 binary)
176 mixed-integer linear inequalities
sampling time = 1, MILP solver = 'glpk'

Type "struct(C)" for more details.
>>
```



Hybrid MPC: 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