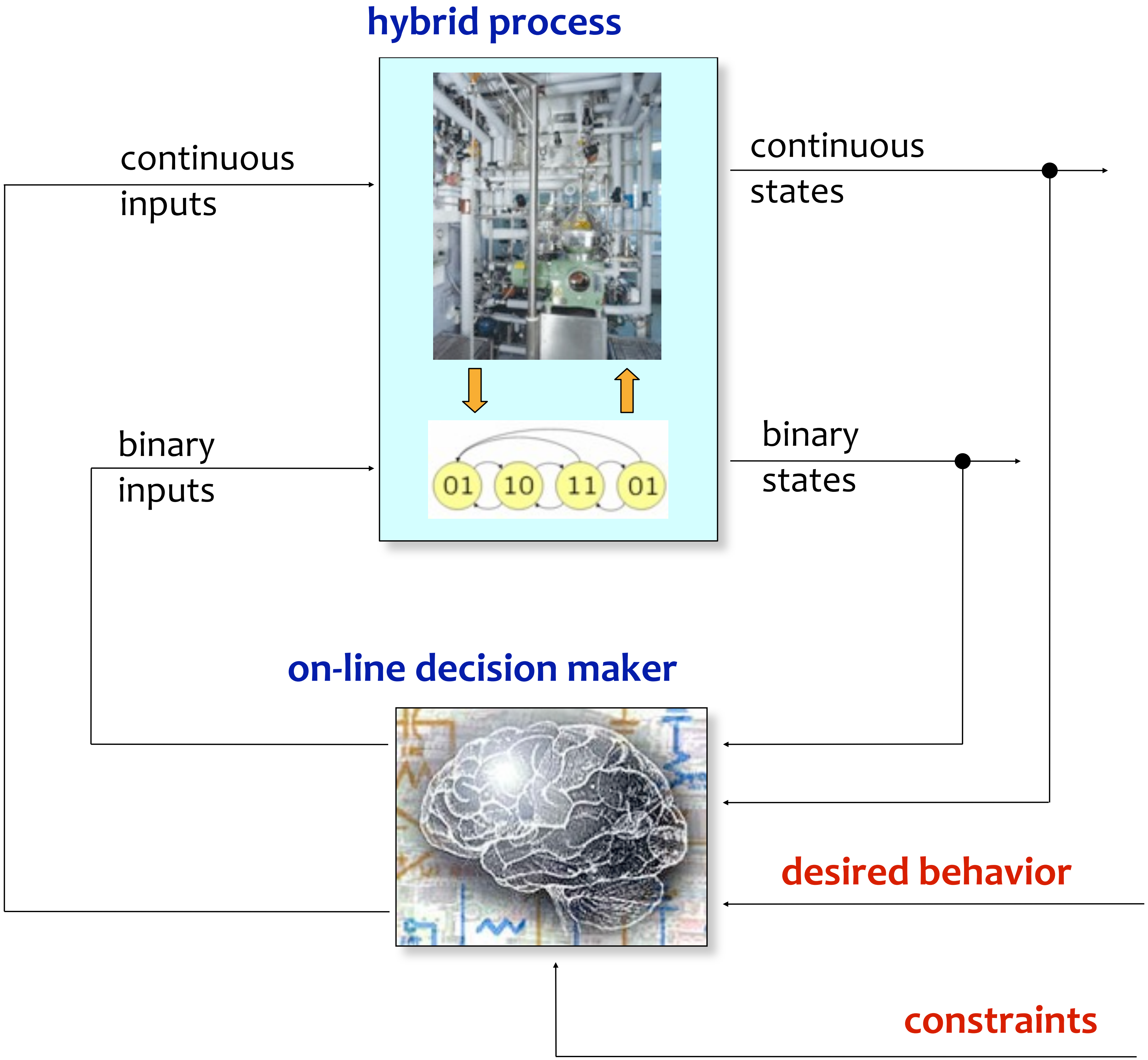
Model Predictive Control of Hybrid Systems

Hybrid Control Problem



hybrid process

continuous inputs

binary inputs

continuous states

binary states

on-line decision maker

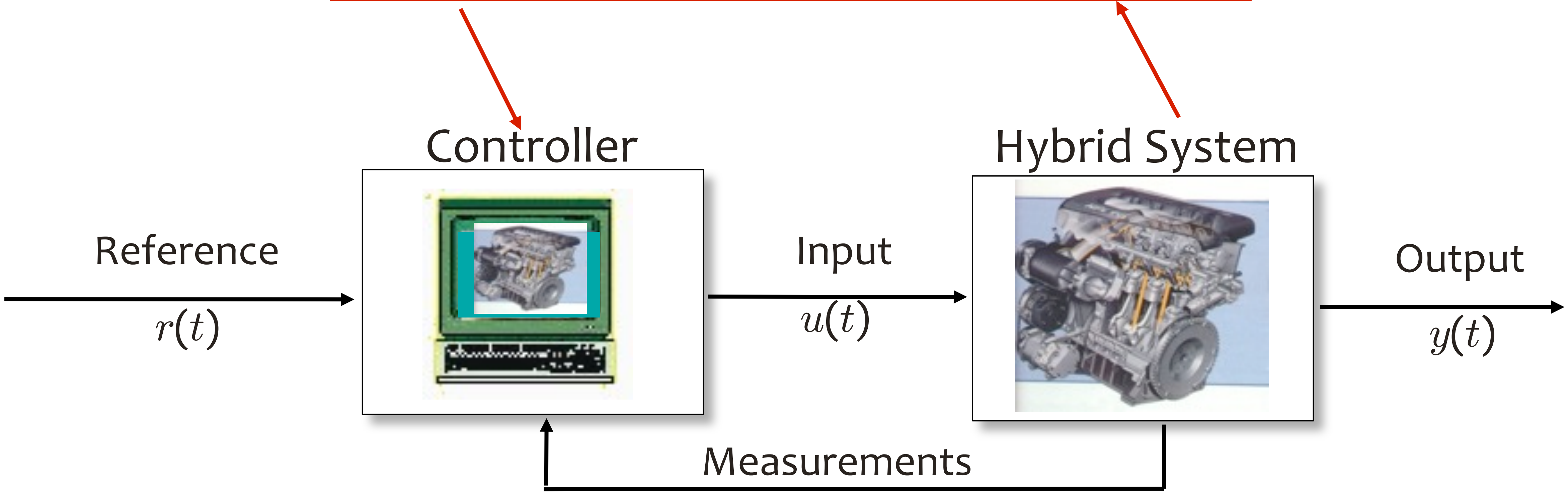
desired behavior

constraints

Model Predictive Control of Hybrid Systems

MLD model

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



Controller

Hybrid System

Reference

$r(t)$

Input

$u(t)$

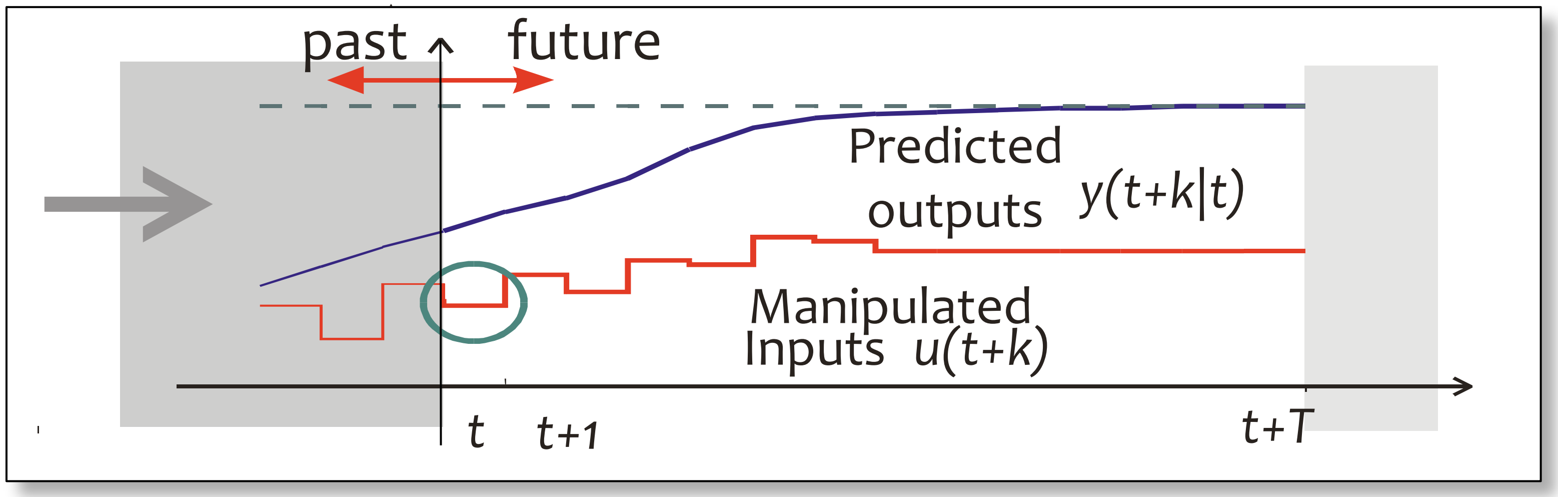
Output

$y(t)$

Measurements

- MODEL: use an MLD (or PWA) model of the plant to predict the future behavior of the hybrid system
- PREDICTIVE: optimization is still based on the predicted future evolution of the hybrid system
- CONTROL: the goal is to control the hybrid system

Hybrid Model Predictive Control



Model
Predictive (MPC)
Control

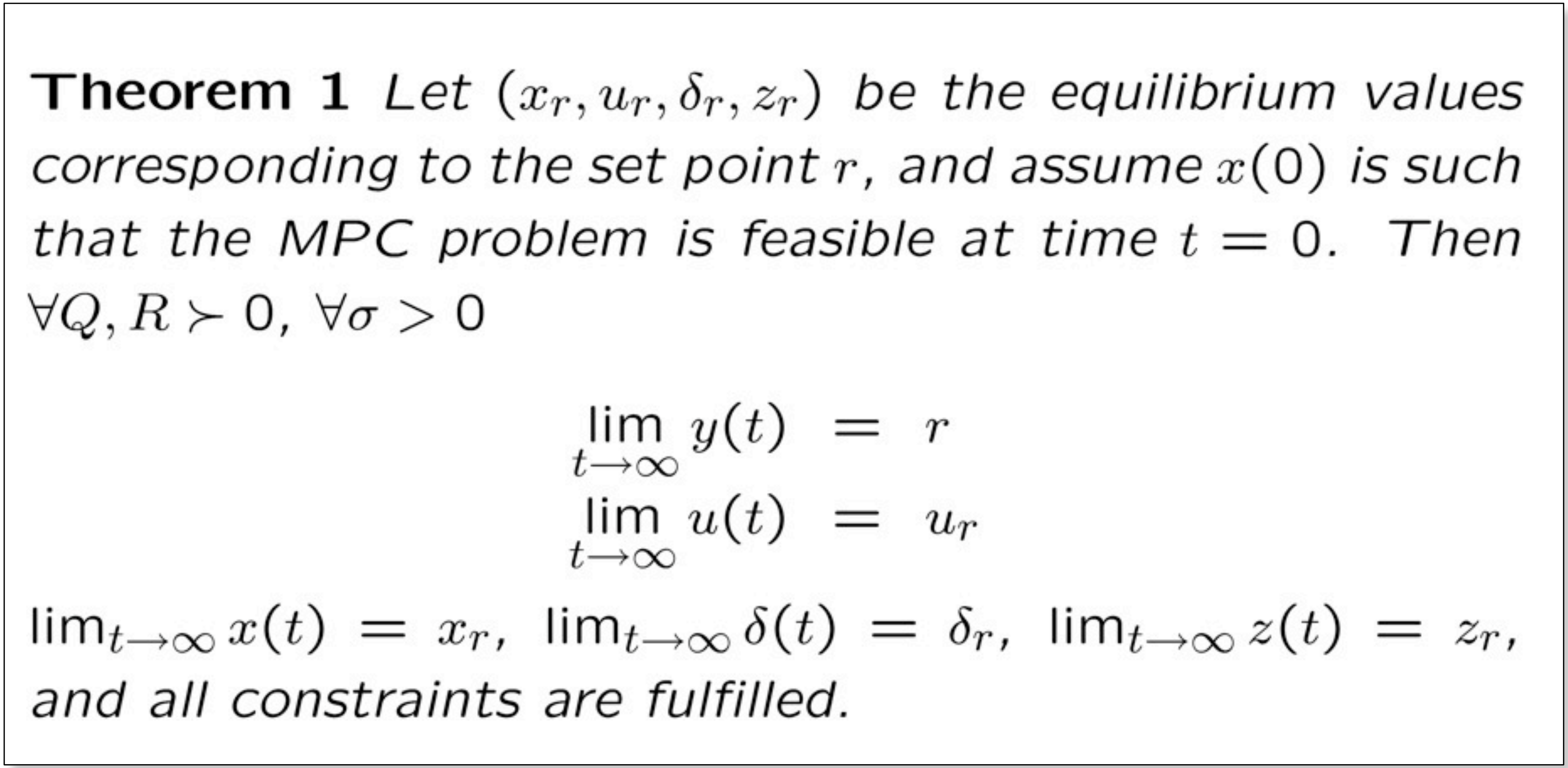
- At time t solve with respect to $U \triangleq \{u(t), u(t+1), \ldots, u(t+T-1)\}$ the finite-horizon open-loop, optimal control problem:

$$\min_{u(t),\ldots,u(t+T-1)} \sum_{k=0}^{T-1} \|y(t+k|t) - r(t)\| + \rho\|u(t+k) - u_r\|$$
$$+\sigma(\|\delta(t+k) - \delta_r\| + \|z(t+k) - z_r\| + \|x(t+k|t) - x_r\|)$$

$$\text{subject to} \quad \text{MLD model}$$
$$x(t|t) = x(t)$$
$$x(t+T|t) = x_r$$

- Apply only $u(t) = u_t^*$ (discard the remaining optimal inputs)
- At time t+1: get new measurements, repeat optimization

Closed-loop Convergence



Theorem 1 *Let $(x_r, u_r, \delta_r, z_r)$ be the equilibrium values corresponding to the set point r, and assume $x(0)$ is such that the MPC problem is feasible at time $t = 0$. Then* $\forall Q, R \succ 0$, $\forall \sigma > 0$

$$\begin{aligned}\lim_{t\to\infty} y(t) &= r\\ \lim_{t\to\infty} u(t) &= u_r\end{aligned}$$

$\lim_{t\to\infty} x(t) = x_r$, $\lim_{t\to\infty} \delta(t) = \delta_r$, $\lim_{t\to\infty} z(t) = z_r$, *and all constraints are fulfilled.*

(Bemporad, Morari 1999)

Proof: Easily follows from standard Lyapunov arguments

More stability results: see (Lazar, Heemels, Weiland, Bemporad, 2006)

Convergence Proof

- Assume we set the terminal constraint $x(t+T|t) = x_r$ in the optimal control problem

- Let $\mathcal{U}_t^*$ denote the optimal control sequence $\{u_t^*(0), \ldots, u_t^*(T-1)\}$

- Let $V(t) \triangleq J(\mathcal{U}_t^*, x(t))$=value function → Lyapunov function

- By construction, $\mathcal{U}_1 = \{u_t^*(1), \ldots, u_t^*(T-1), u_r\}$ is feasible @$t+1$

- Hence,

$$V(t+1) \boxed{\leq} J(\mathcal{U}_1, x(t+1)) = V(t) - \|y(t)-r\|_Q - \|u(t)-u_r\|_R - \sigma(\|\delta(t)-\delta_r\| - \|z(t)-z_r\| - \|x(t)-x_r\|)$$

- Hence $V(t)$ is decreasing and lower-bounded by 0 $\Rightarrow \exists V_\infty = \lim_{t\to\infty} V(t)$ $\Rightarrow V(t+1) - V(t) \to 0$

- Hence, $\|y(t)-r\|_Q \to 0, \|u(t)-u_r\|_R \to 0, \ \ldots, \ \|x(t)-x_r\| \to 0$

Note: Global optimum not needed for convergence !

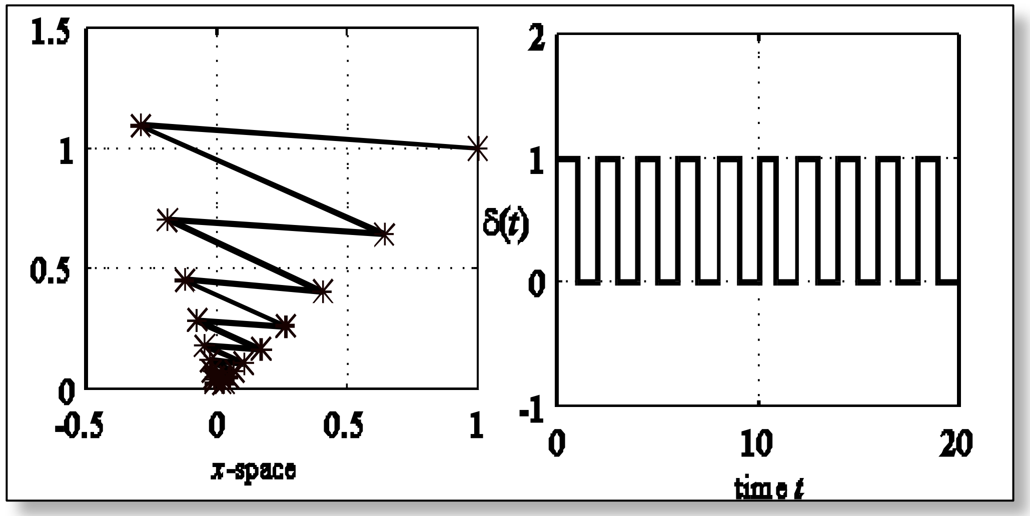
Hybrid MPC - Example

PWA system:

$$
\begin{aligned}
x(t+1) &= 0.8\begin{bmatrix}\cos\alpha(t) & -\sin\alpha(t)\\ \sin\alpha(t) & \cos\alpha(t)\end{bmatrix}x(t)+\begin{bmatrix}0\\1\end{bmatrix}u(t)\\
y(t) &= x_2(t)\\
\alpha(t) &= \begin{cases}\frac{\pi}{3} & \text{if } x_1(t)>0\\ -\frac{\pi}{3} & \text{if } x_1(t)\le 0\end{cases}
\end{aligned}
$$

Constraint: $-1 \le u(t) \le 1$

Open loop behavior



go to demo /**demos/hybrid/bm99sim.m**

Hybrid MPC - Example

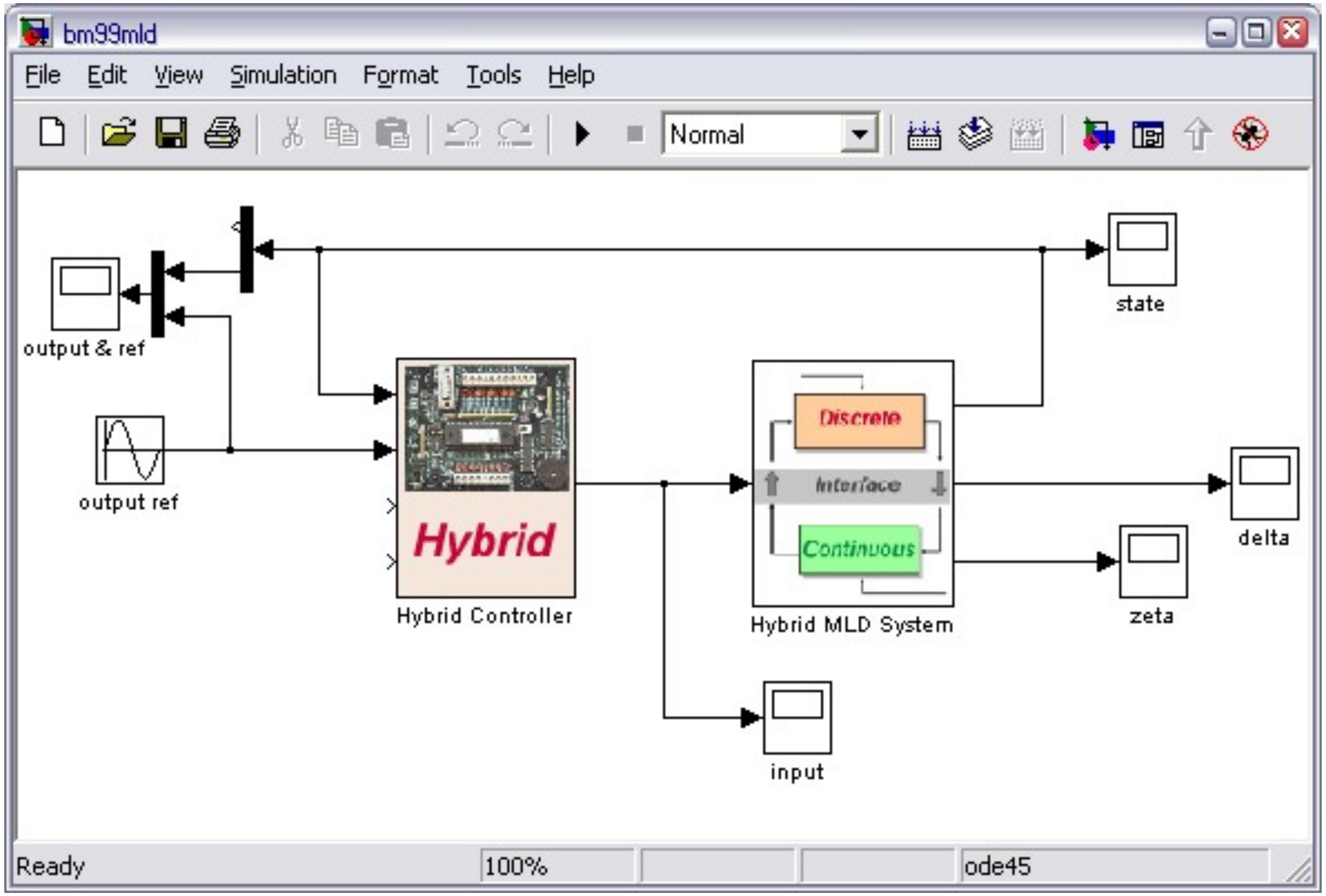
HYSDEL model

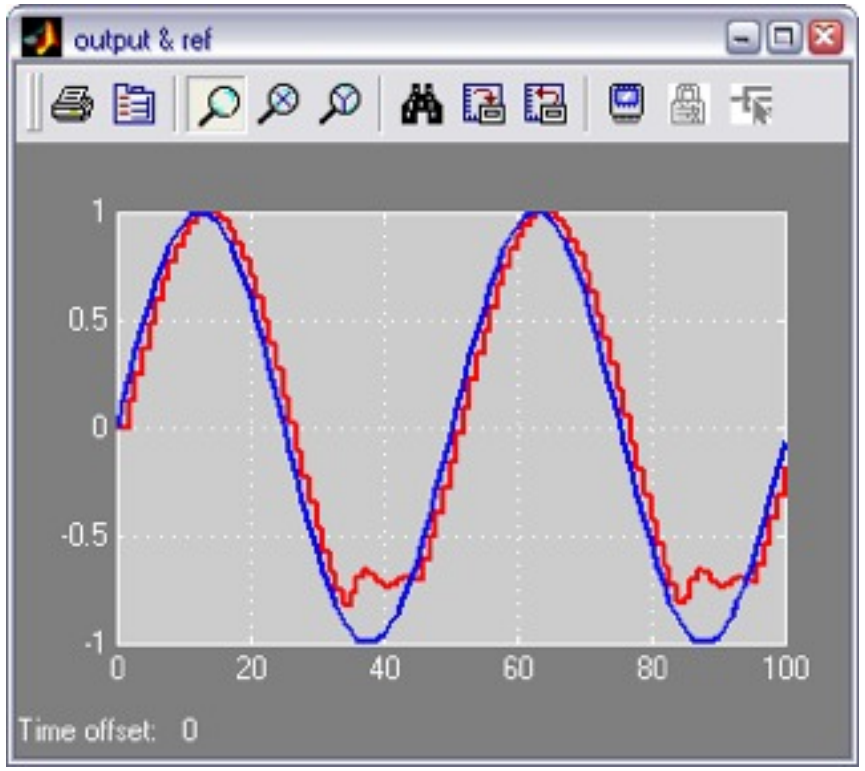
```
/* 2x2 PWA system  - Example from the paper
   A. Bemporad and M. Morari, ``Control of systems integrating logic, dynamics,
   and constraints,'' Automatica, vol. 35, no. 3, pp. 407-427, 1999.
   (C) 2003 by A. Bemporad, 2003 */

SYSTEM pwa {

INTERFACE {
            STATE { REAL x1 [-10,10];
                    REAL x2 [-10,10];}

            INPUT { REAL u [-1.1,1.1];}

            OUTPUT{ REAL y;}

            PARAMETER {
              REAL alpha = 1.0472; /* 60 deg in radiants */
              REAL C = cos(alpha);
              REAL S = sin(alpha);}
           }

IMPLEMENTATION {
            AUX { REAL z1,z2;
                  BOOL sign; }
            AD  { sign = x1<=0; }

            DA  { z1 = {IF sign THEN 0.8*(C*x1+S*x2)
                        ELSE 0.8*(C*x1-S*x2) };
                  z2 = {IF sign THEN 0.8*(-S*x1+C*x2)
                        ELSE 0.8*(S*x1+C*x2) };   }

            CONTINUOUS {x1 = z1;
                        x2 = z2+u; }

            OUTPUT { y = x2;   }
              }
}
```

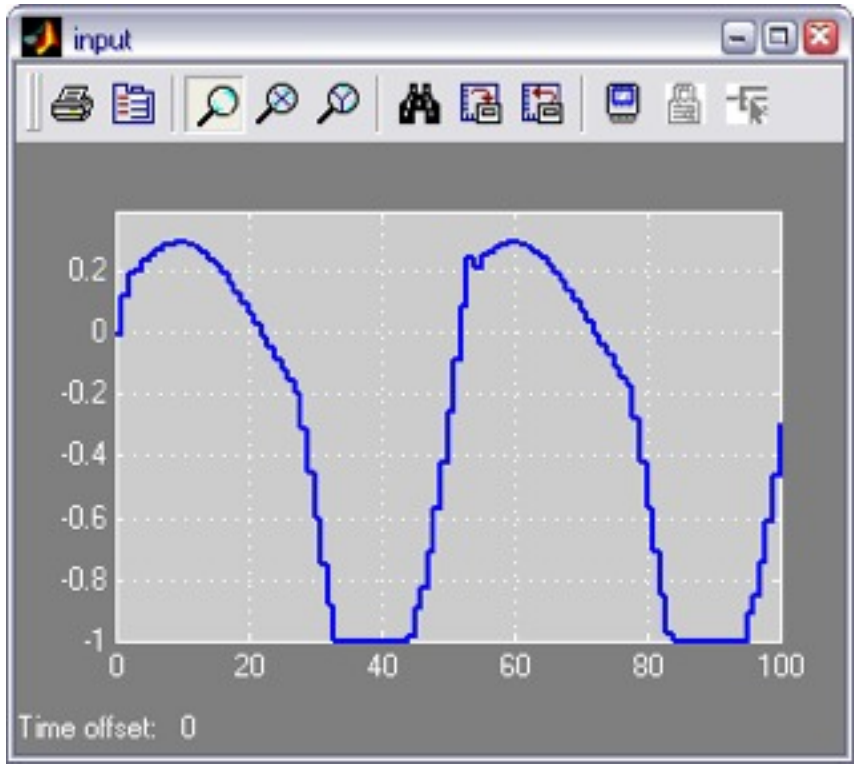
/demos/hybrid/bm99.hys

Hybrid MPC - Example

Closed-loop:







Performance index:

$$\min \sum_{k=1}^{2} |y(t+k|t) - r(t)|$$

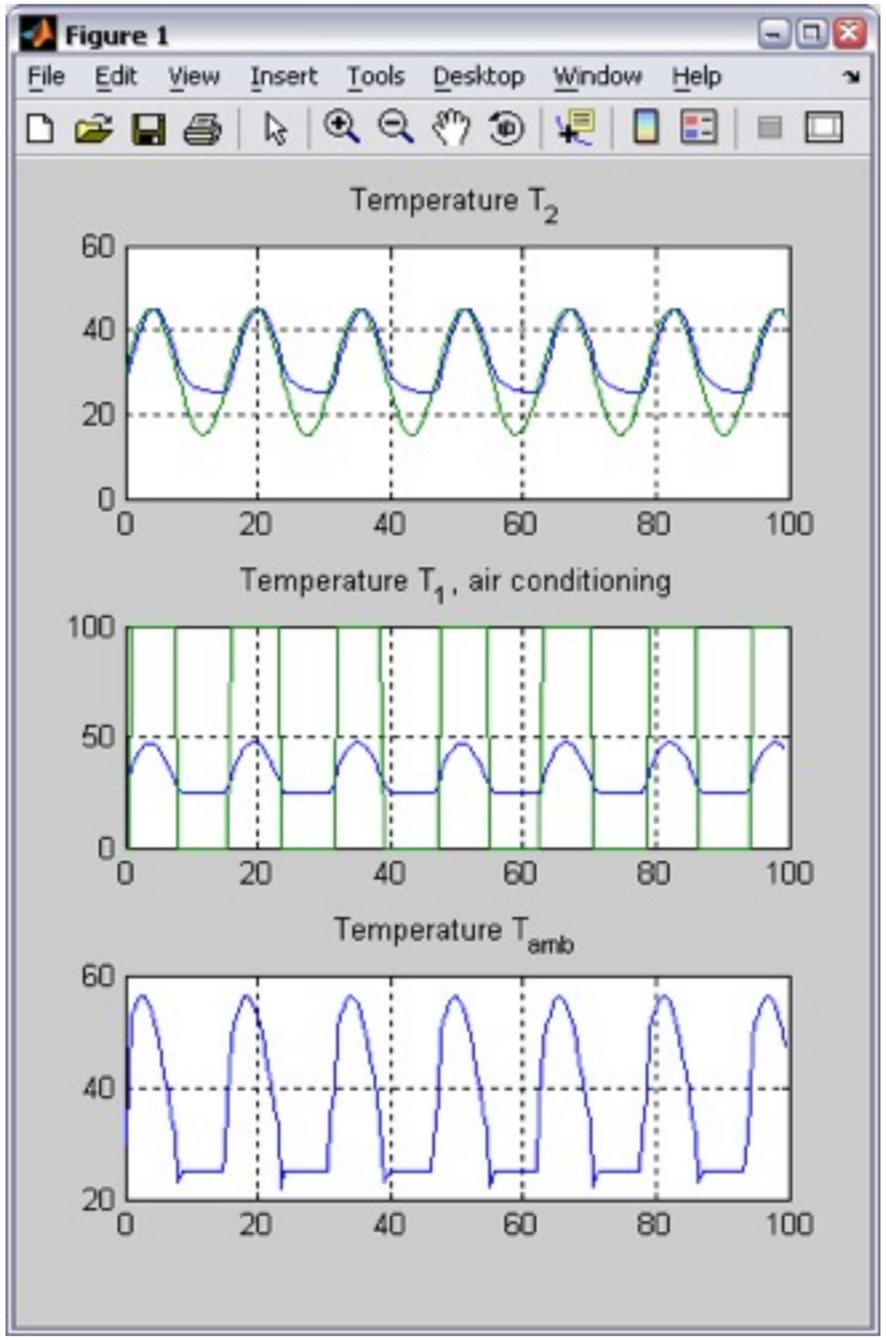
Hybrid MPC – Temperature Control

```
>>refs.x=2;    % just weight state #2
>>Q.x=1;
>>Q.rho=Inf;   % hard constraints
>>Q.norm=2;    % quadratic costs
>>N=2;         % optimization horizon
>>limits.xmin=[25;-Inf];
```

$$\begin{aligned} \min \quad & \sum_{k=1}^{2} (x_2(k) - r)^2 \\ \text{s.t.} \quad & x_1(k) \geq 25 \quad k = 1, 2 \\ & \text{MLD model} \end{aligned}$$

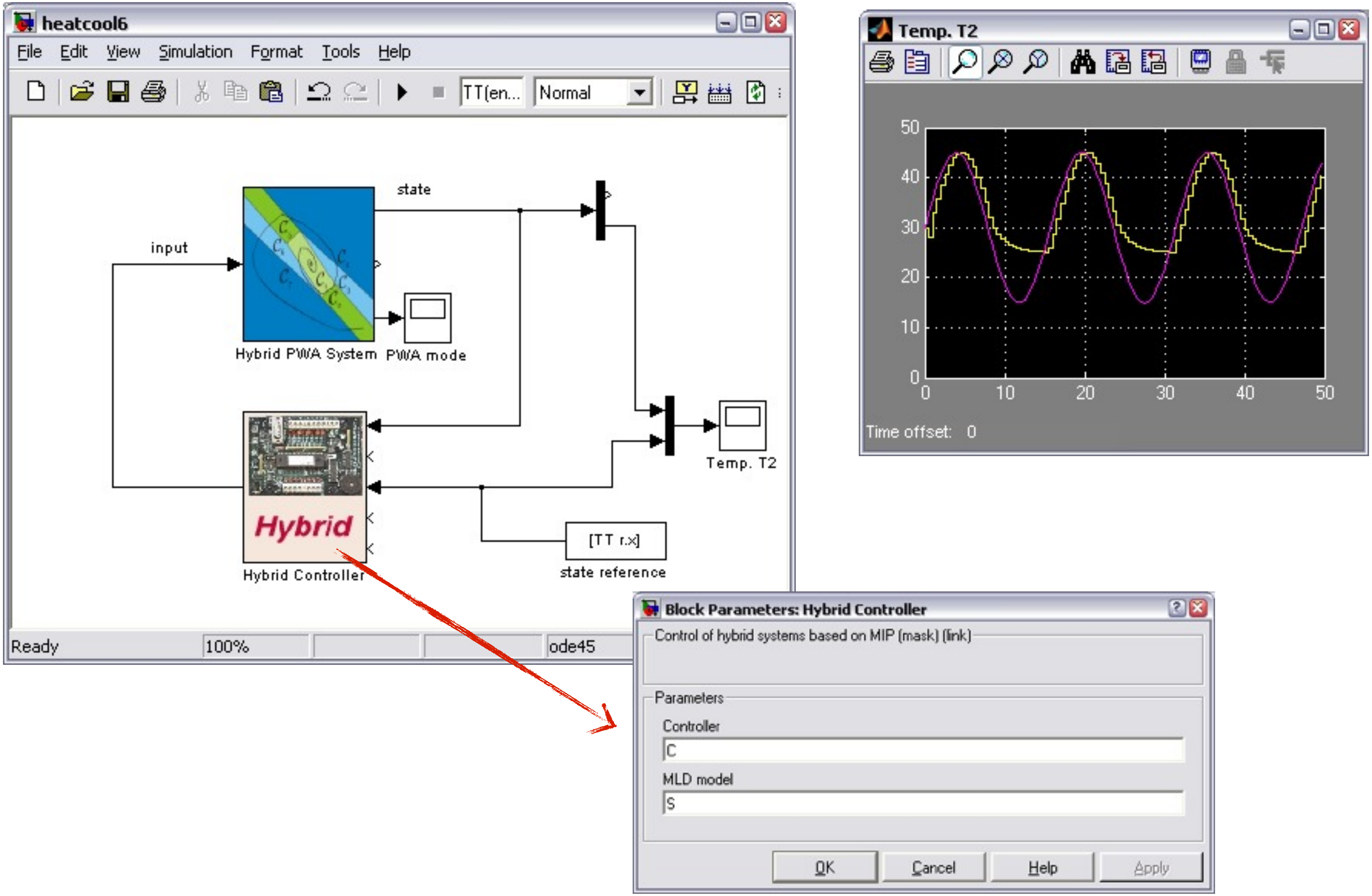
```
>>C=hybcon(S,Q,N,limits,refs);
```

```
>> C

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

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

 20 optimization variable(s) (8 continuous, 12 binary)
 46 mixed-integer linear inequalities
sampling time = 0.5, MILP solver = 'glpk'

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



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

Hybrid MPC – Temperature Control



Block Parameters: Hybrid Controller

Control of hybrid systems based on MIP (mask) (link)

Parameters

Controller: C

MLD model: S

Optimal Control of Hybrid Systems: Computational Aspects

MIQP Formulation of MPC

(Bemporad, Morari, 1999)

$$\min_\xi J(\xi, x(0)) = \sum_{t=0}^{T-1} y'(t)Qy(t) + u'(t)Ru(t)$$

$$\text{subject to} \left\{ \begin{array}{rcl} x(t+1) & = & Ax(t) + B_1u(t) + B_2\delta(t) + B_3z(t) + B_5 \\ y(t) & = & Cx(t) + D_1u(t) + D_2\delta(t) + D_3z(t) + D_5 \\ E_2\delta(t) + E_3z(t) & \leq & E_4x(t) + E_1u(t) + E_5 \end{array} \right.$$

- Optimization vector:

$$\xi = [u(0), \ldots, u(T-1), \delta(0), \ldots, \delta(T-1), z(0), \ldots z(T-1)]'$$

→

$$\begin{array}{l} \min_\xi \frac{1}{2}\xi' H\xi + x(0)'F\xi + \cancel{\frac{1}{2}x'(0)Yx(0)} \\ \text{subj. to} \quad G\xi \leq W + Sx(0) \end{array}$$

Mixed Integer Quadratic Program (MIQP)

$u \in \mathbb{R}^{n_u}, \ \delta \in \{0,1\}^{n_\delta}, \ z \in \mathbb{R}^{n_z}$ → $\xi \in \mathbb{R}^{(n_u+n_z)T} \times \{0,1\}^{n_\delta T}$

ξ has both real and $\{0,1\}$ components

MIQP Formulation of MPC

(Bemporad, Borrelli, Morari, 2000)

$$\begin{array}{ll} \min_{\xi} & J(\xi, x(0)) = \sum_{t=0}^{T-1} \|Qy(t)\|_\infty + \|Ru(t)\|_\infty \\ \text{subject to} & \text{MLD model} \end{array}$$

- Basic trick: introduce slack variables

$$\min_{x} |x| \quad \Longrightarrow \quad \begin{array}{ll} \min_{x,\epsilon} & \epsilon \\ \text{s.t.} & \epsilon \geq x \\ & \epsilon \geq -x \end{array}$$

$$\begin{cases} \epsilon_k^x \geq \|Qy(t+k|t)\|_\infty \\ \epsilon_k^u \geq \|Ru(t+k)\|_\infty \end{cases} \quad \Longrightarrow \quad \begin{cases} \epsilon_k^x \geq Q^i y(t+k|t) & i = 1, \ldots, p \quad k = 0, \ldots, T-1 \\ \epsilon_k^x \geq -Q^i y(t+k|t) & i = 1, \ldots, p \quad k = 0, \ldots, T-1 \\ \epsilon_k^u \geq R^i u(t+k) & i = 1, \ldots, m \quad k = 0, \ldots, T-1 \\ \epsilon_k^u \geq -R^i u(t+k) & i = 1, \ldots, m \quad k = 0, \ldots, T-1 \end{cases}$$

Q^i = ith row of matrix Q

- Optimization vector:

$$\xi = [\epsilon_1^x, \ldots, \epsilon_{T-1}^x, \epsilon_0^u, \ldots, \epsilon_{T-1}^u, u(0), \ldots, u(T-1), \delta(0), \ldots, \delta(T-1), z(0), \ldots z(T-1)]'$$

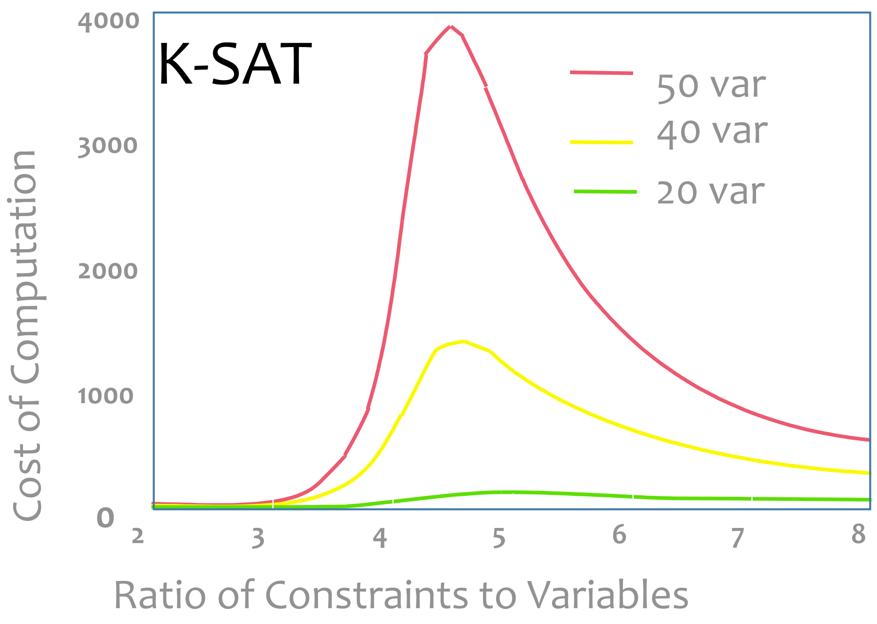
$$\Longrightarrow \quad \begin{array}{ll} \min_{\xi} & J(\xi, x(0)) = \sum_{k=0}^{T-1} \epsilon_k^x + \epsilon_k^u \\ \text{s.t.} & G\xi \leq W + Sx(0) \end{array}$$

Mixed Integer Linear Program (MILP)

ξ has both real and $\{0,1\}$ components

Mixed-Integer Program Solvers

- Mixed-Integer Programming is *NP*-complete



(Monasson et al., *Nature*, 1999)

Phase transitions have been found in computationally hard problems.

BUT

- General purpose Branch & Bound/Branch & Cut solvers available for MILP and MIQP (CPLEX, Xpress-MP, BARON, GLPK, ...)

  More solvers and benchmarks: http://plato.la.asu.edu/bench.html

- No need to reach global optimum (see proof of the theorem), although performance deteriorates

Solving Mixed-Integer Programs

$$\begin{array}{ll} \text{min} & f'x + d'\delta \\ \text{s.t.} & Ax + B\delta \leq c \\ & x \in \mathbb{R}^n, \;\; \delta \in \{0,1\}^m \end{array}$$

$$\begin{array}{ll} \text{min} & \frac{1}{2}\left[\begin{smallmatrix} x \\ \delta \end{smallmatrix}\right]' H \left[\begin{smallmatrix} x \\ \delta \end{smallmatrix}\right] + f'x + d'\delta \\ \text{s.t.} & Ax + B\delta \leq c \\ & x \in \mathbb{R}^n, \;\; \delta \in \{0,1\}^m \end{array}$$

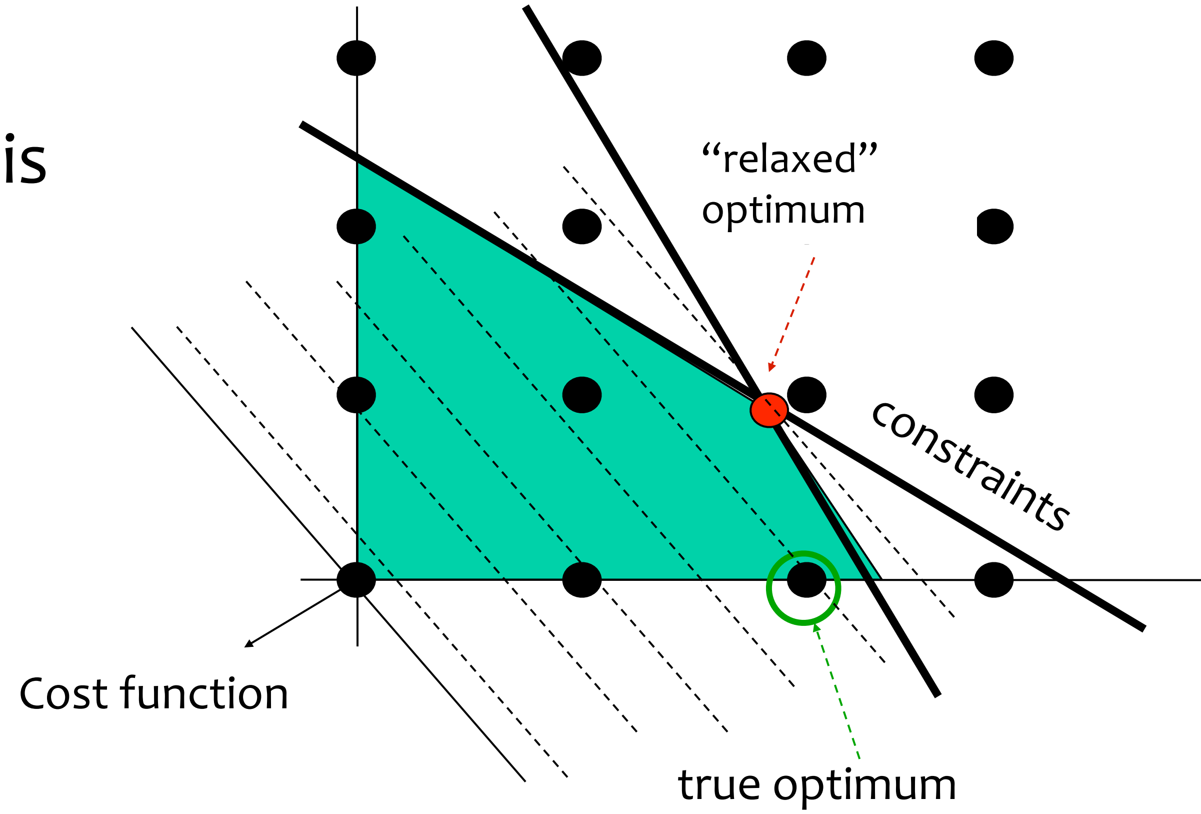
- Some variables are continuous, some are discrete (0/1)
- In general, it is a NP-Hard problem
- Naive solution: enumerate all possible integer solutions and choose the best one.

But m binary variables lead to 2^m solutions, each of which requires a LP

→ Impossible but for extremely small m !

Branch & Bound Algorithm

1. Solve a “relaxed” problem with all binary variables treated as continuous, $0 \leq \delta_i \leq 1$. This gives a (lower) bound on the “best possible” solution. Unfortunately, some δ_i may have fractional parts.

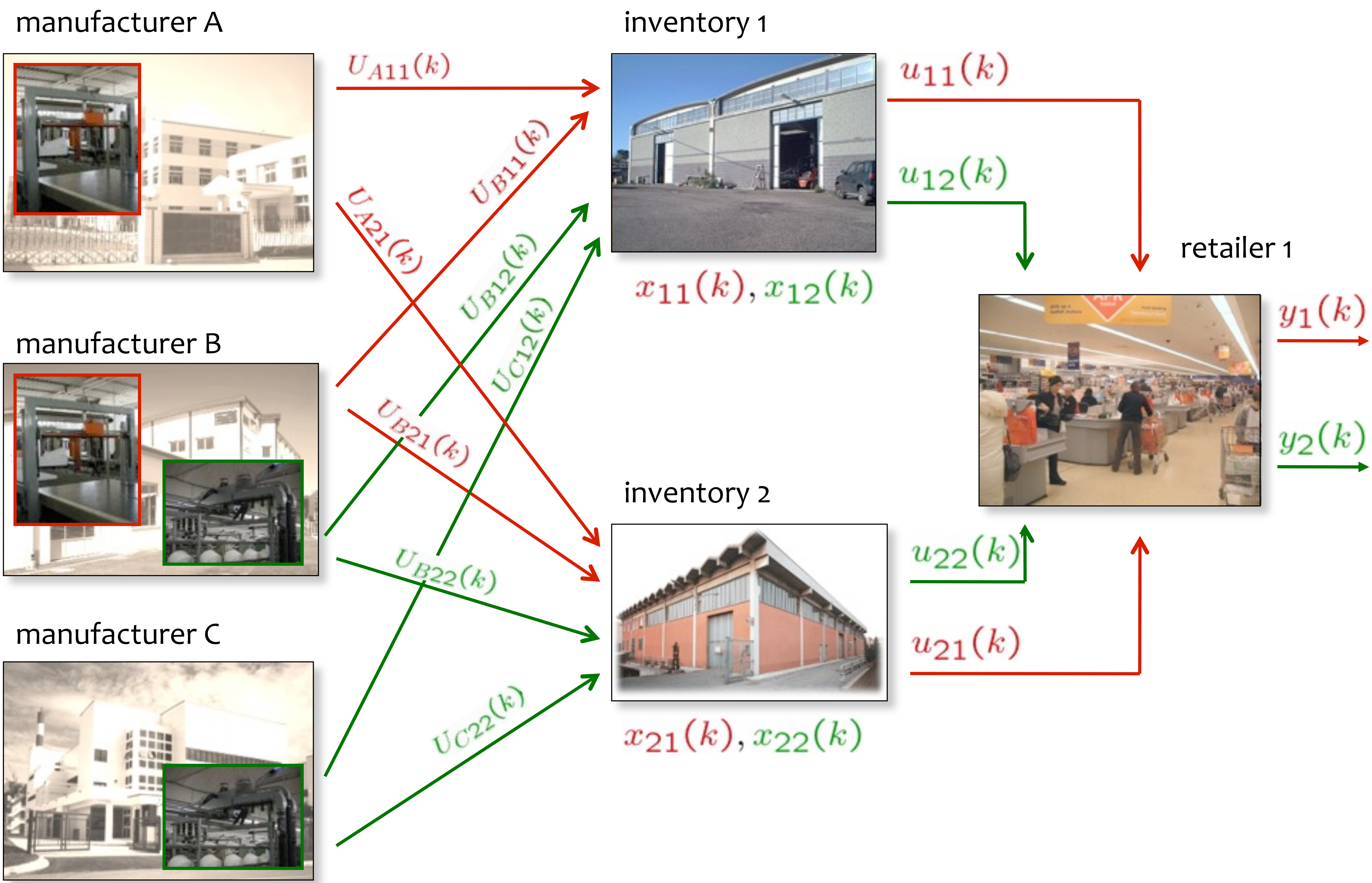


2. Branch on one binary variable: set δ_j=0 and δ_j=1 in two separate solutions, for some j

3. Use a bound on the optimal cost to eliminate in one shot a large number of combinations that are certainly not leading to the optimum (=advantage over full enumeration)

4. Branch again on another variable, and so on, until no further branching is possible.

A Simple Example in Supply Chain Management



manufacturer A

manufacturer B

manufacturer C

inventory 1

inventory 2

retailer 1

$u_{A11}(k)$

$u_{B11}(k)$

$u_{A21}(k)$

$u_{B12}(k)$

$u_{C12}(k)$

$u_{B21}(k)$

$u_{B22}(k)$

$u_{C22}(k)$

$u_{11}(k)$

$u_{12}(k)$

$x_{11}(k), x_{12}(k)$

$u_{22}(k)$

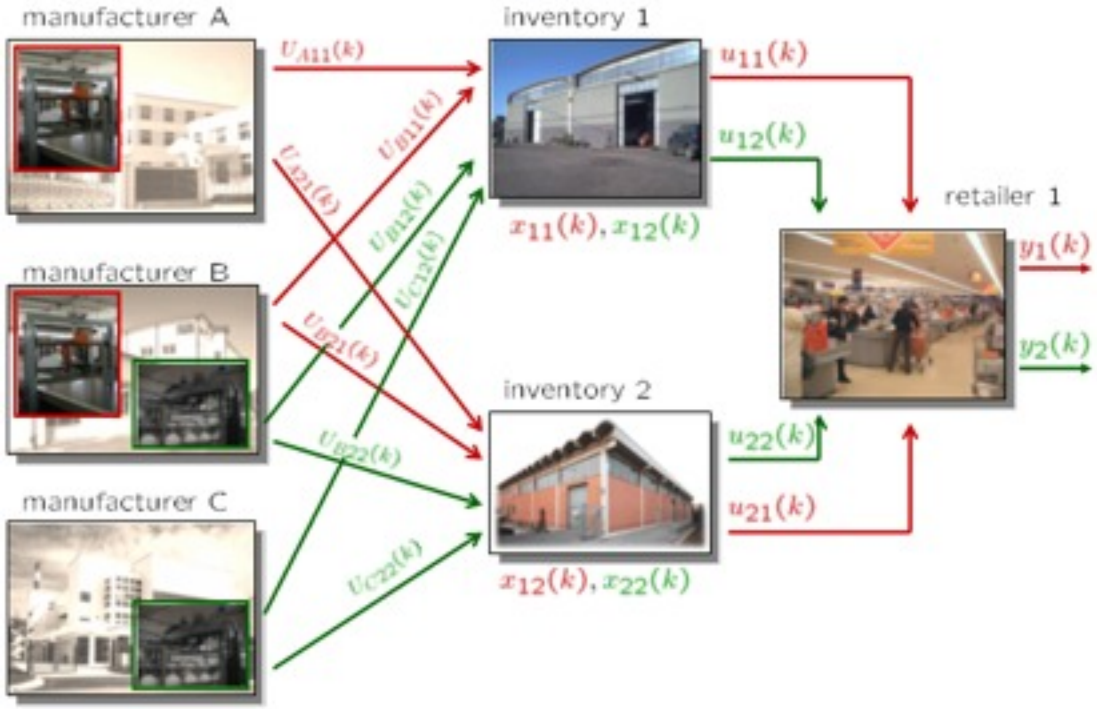
$u_{21}(k)$

$x_{21}(k), x_{22}(k)$

$y_1(k)$

$y_2(k)$

System Variables



- <u>continuous states:</u>

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

- <u>continuous outputs:</u>

$y_j(k)$ = amount of j sold at time k (j=1,2)

- <u>continuous inputs:</u>

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

- <u>binary inputs:</u>

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

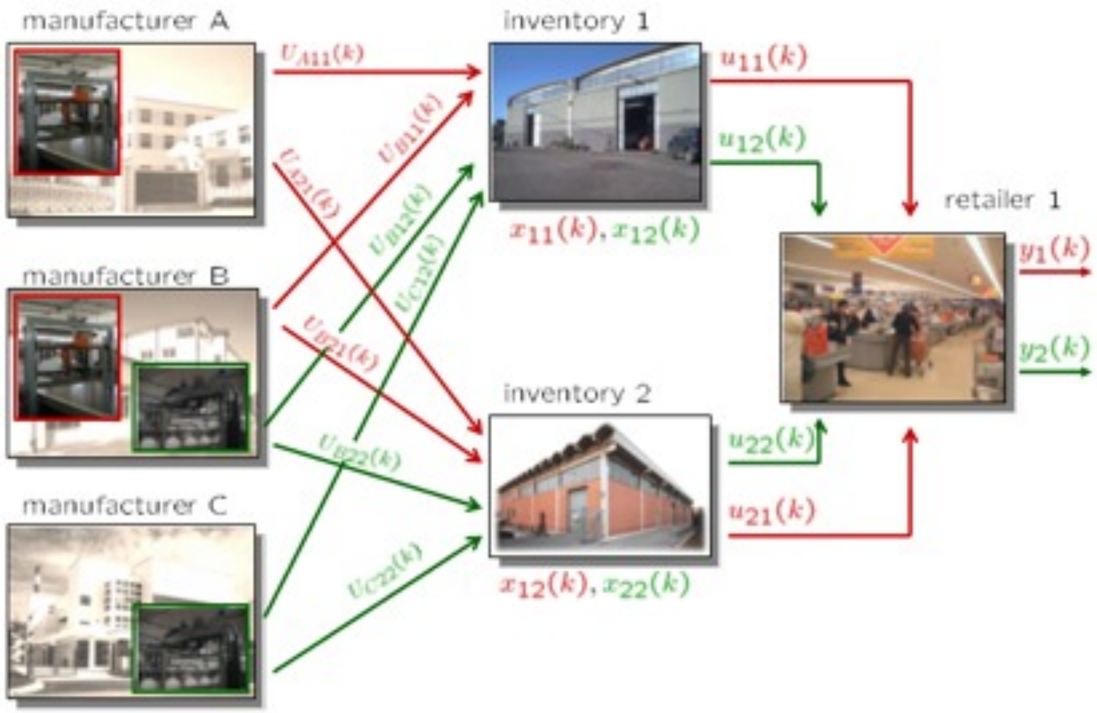
Constraints

- Max capacity of inventory i:

$$0 \leq \sum_j x_{ij}(k) \leq x_{Mi}$$

Numerical values:

$x_{M1}=10$, $x_{M2}=10$



- Max transportation from inventories:

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

- A product can only be sent to one inventory:

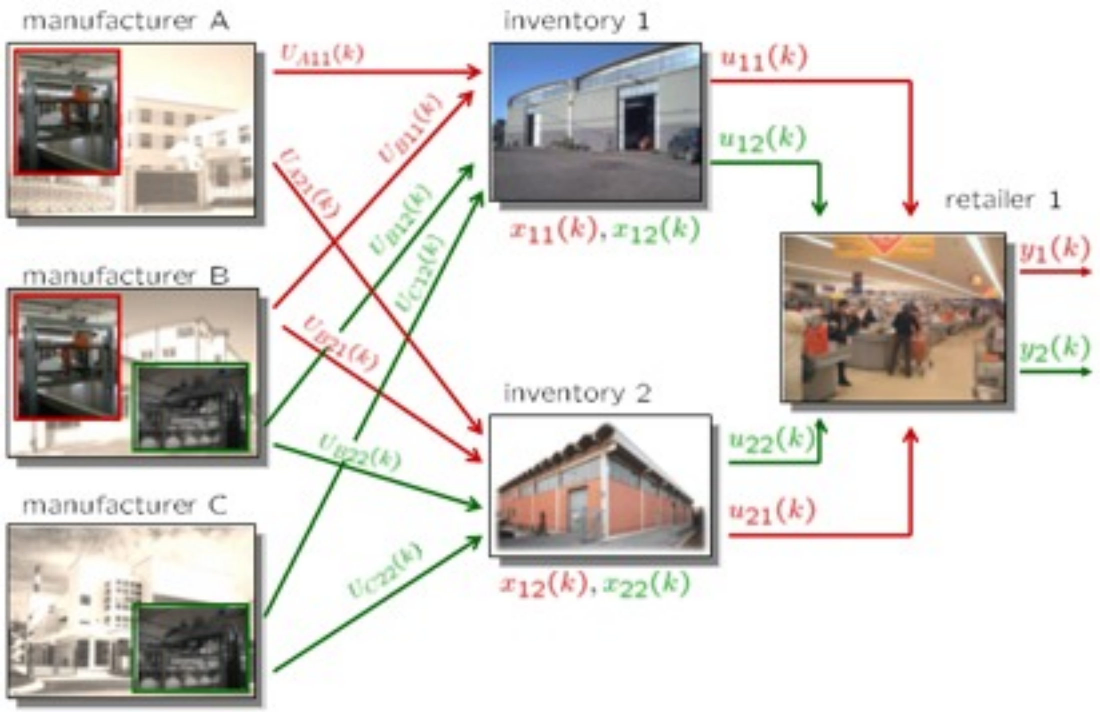
  UA11(k) and UA21(k) cannot be =1 at the same time
  UB11(k) and UB21(k) cannot be =1 at the same time UB12(k) and UB22(k) cannot be =1 at the same time
  UC12(k) and UC22(k) cannot be =1 at the same time

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

  [UB11(k)=1 or UB21(k)=1] and [UB12(k)=1 or UB22(k)=1] cannot be true

Dynamics

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



Numerical values:

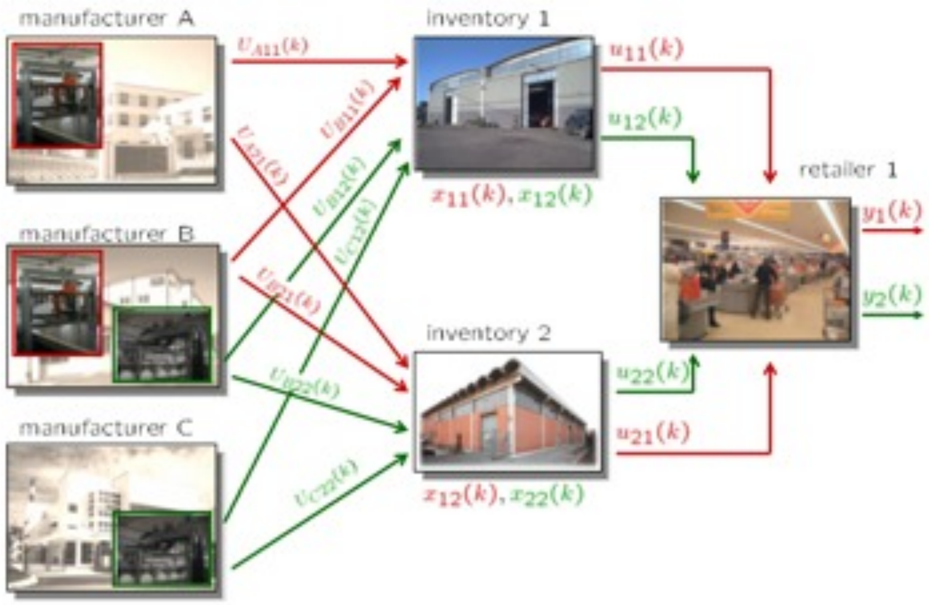
P_{A1}=4, P_{B1}=6, P_{B2}=7, P_{C2}=3

- Level of inventories:

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

Hybrid Dynamical Model

```
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}; }
```

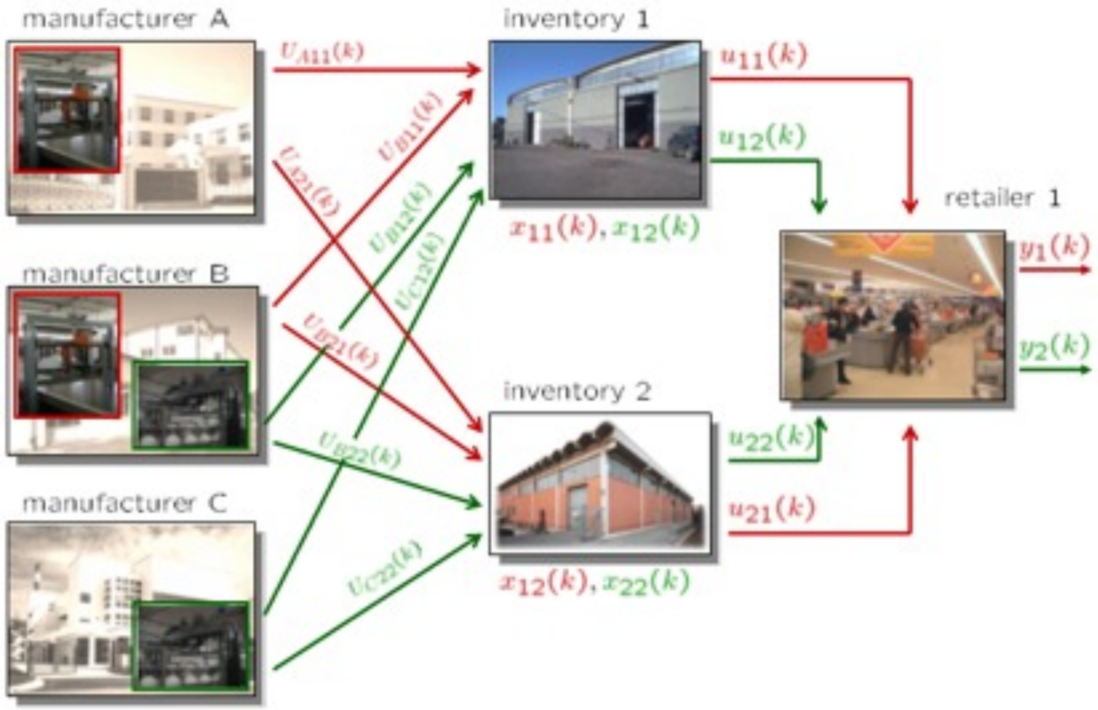


```
            CONTINUOUS {x11 = x11 + zA11 + zB11 - u11;
                        x12 = x12 + zB12 + zC12 - u12;
                        x21 = x21 + zA21 + zB21 - u21;
                        x22 = x22 + zB22 + zC22 - u22;   }

            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; }
} }
```

/demos/hybrid/supply_chain.m

Objectives

- Meet customer demand as much as possible: $y_1 \approx r_1, y_2 \approx r_2$

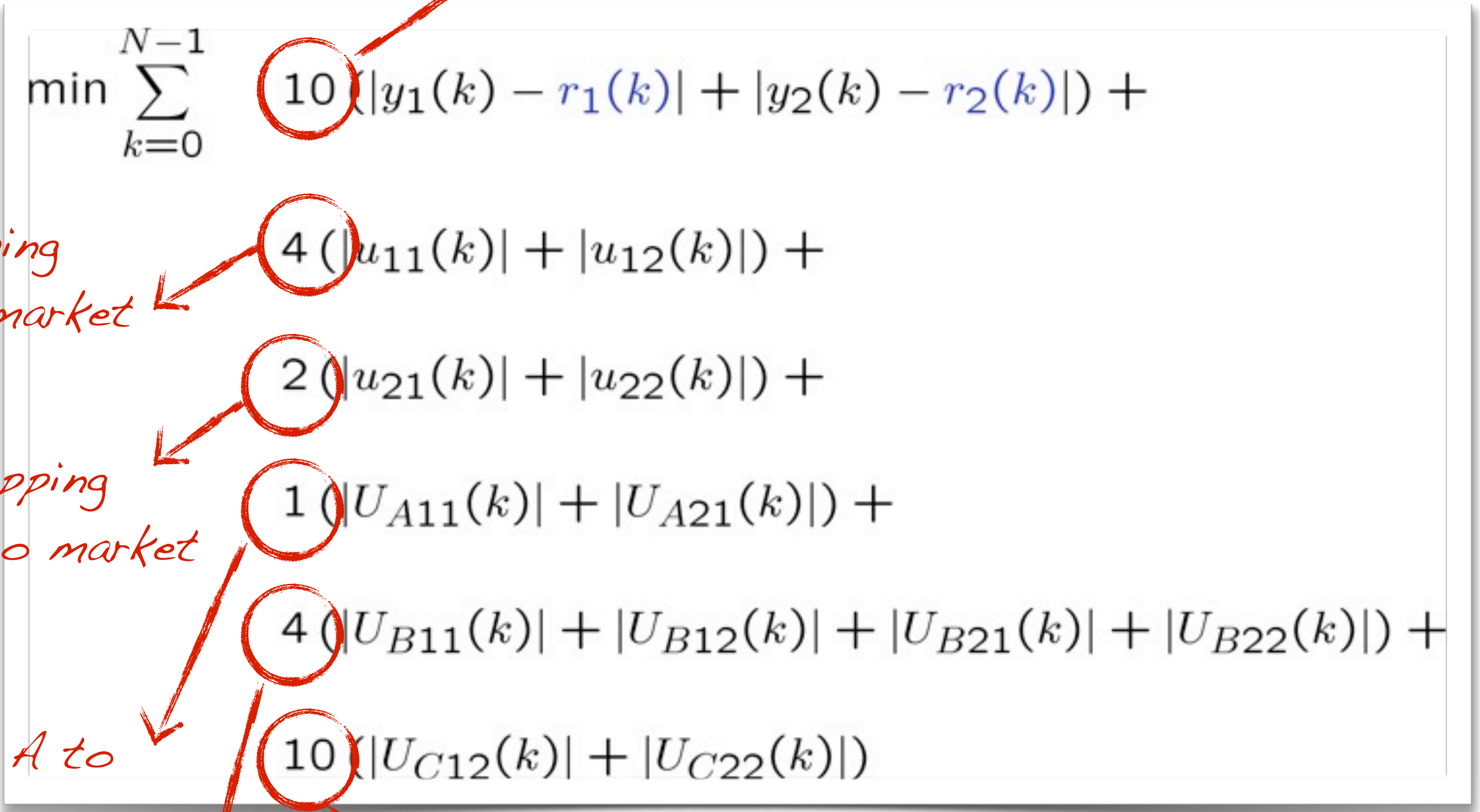


- Minimize transportation costs

- Fulfill all constraints

Performance Specs

penalty on demand tracking error



$$
\begin{aligned}
\min \sum_{k=0}^{N-1} \quad & 10\,(|y_1(k)-r_1(k)|+|y_2(k)-r_2(k)|)+\\
& 4\,(|u_{11}(k)|+|u_{12}(k)|)+\\
& 2\,(|u_{21}(k)|+|u_{22}(k)|)+\\
& 1\,(|U_{A11}(k)|+|U_{A21}(k)|)+\\
& 4\,(|U_{B11}(k)|+|U_{B12}(k)|+|U_{B21}(k)|+|U_{B22}(k)|)+\\
& 10\,(|U_{C12}(k)|+|U_{C22}(k)|)
\end{aligned}
$$

cost for shipping from inv.#1 to market

cost for shipping from inv.#2 to market

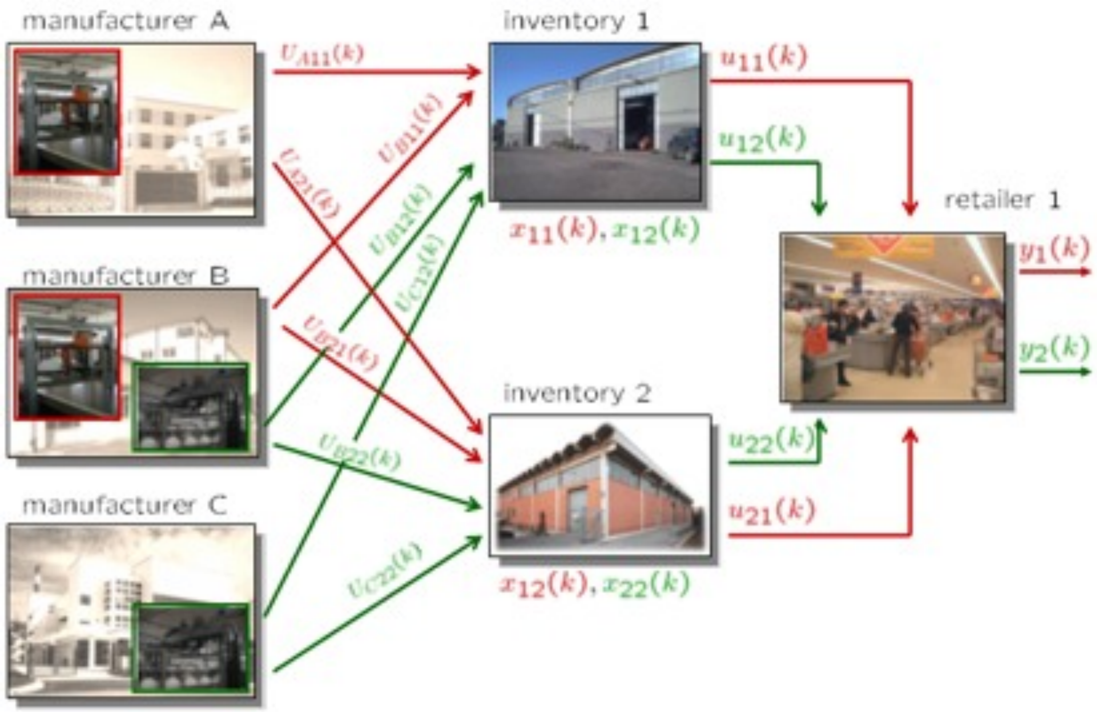
cost from A to inventories

cost from B to inventories

cost from C to inventories

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;
>>limits.xmax=xmax;
```



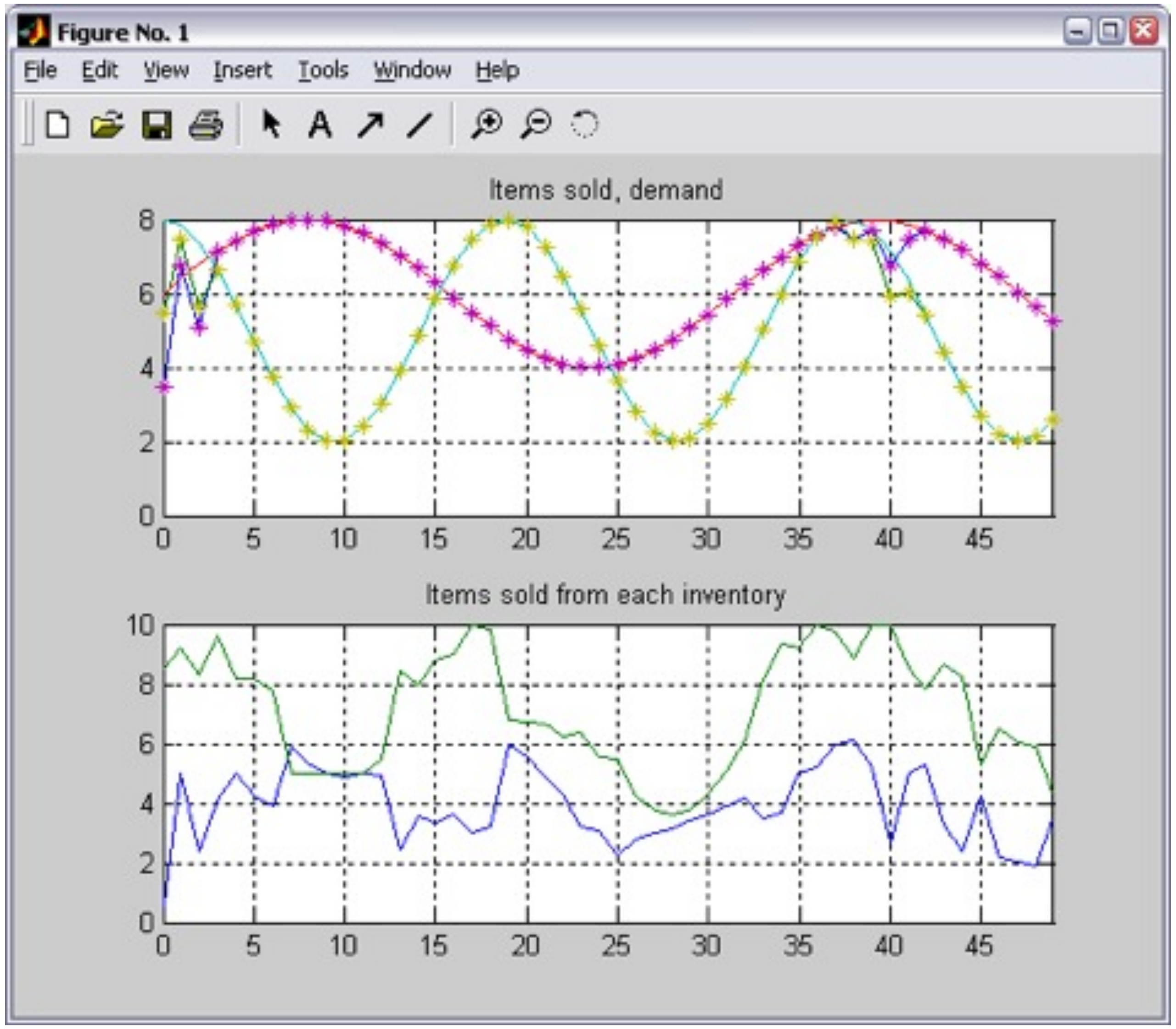
```
>>C=hybcon(S,Q,N,limits,refs);
```

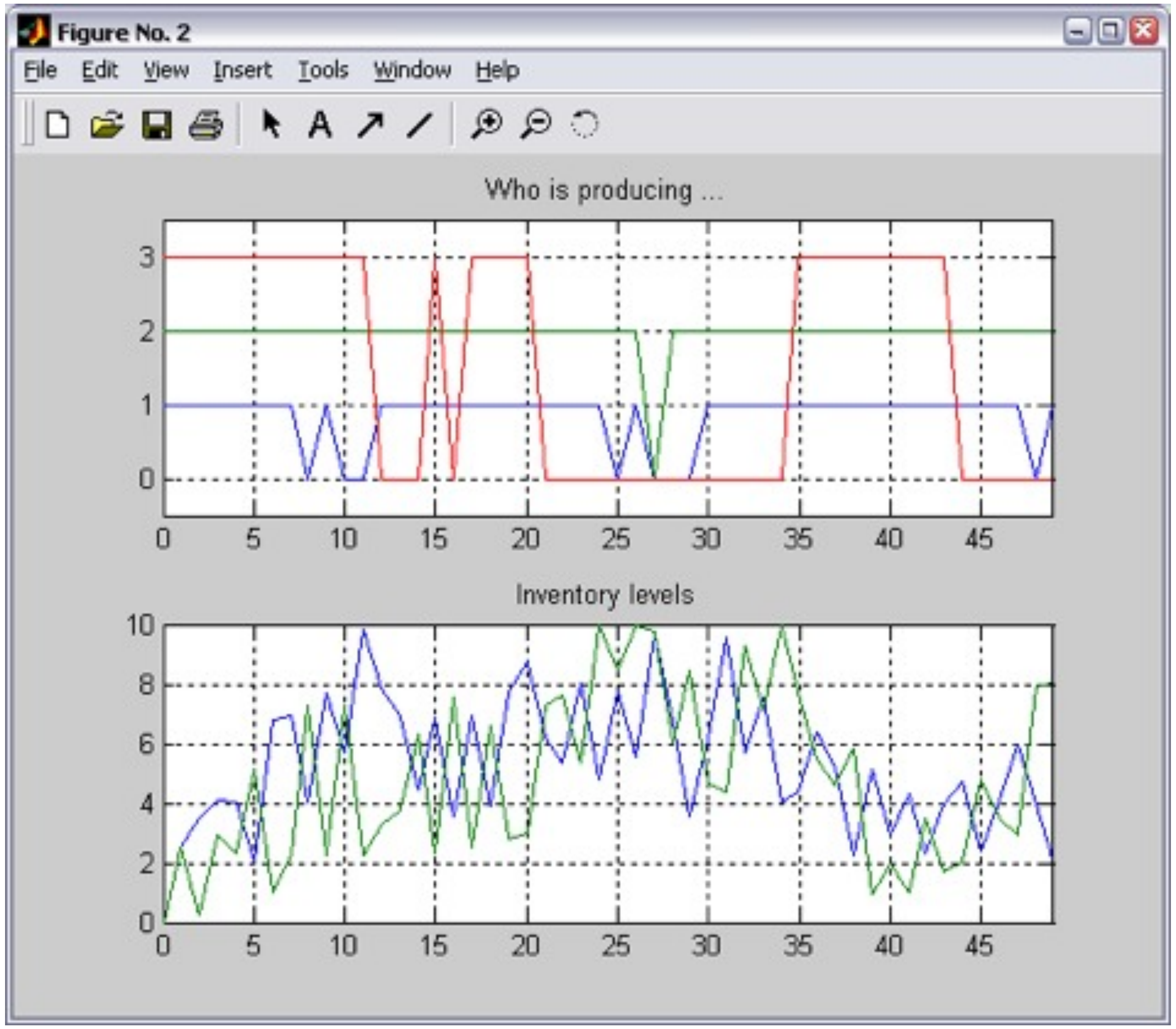
```
>> C

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) (28 continuous, 16 binary)
176 mixed-integer linear inequalities
sampling time = 1, MILP solver = 'glpk'

Type "struct(C)" for more details.

>>
```

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: ≈ 30ms per time step (using GLPK on this machine)

Explicit Hybrid MPC

Explicit Hybrid MPC (MLD)

$$\min_{\xi} J(\xi,\boxed{x(t)},\boxed{r(t)}) = \sum_{k=0}^{T-1} \|Q(y_k - \boxed{r(t)})\|_\infty + \|Ru_k\|_\infty$$

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

- On-line optimization: solve the problem *for each* given $x(t)$

Mixed-Integer Linear Program (MILP)

- Off-line optimization: solve the MILP **for all** $x(t)$ in advance

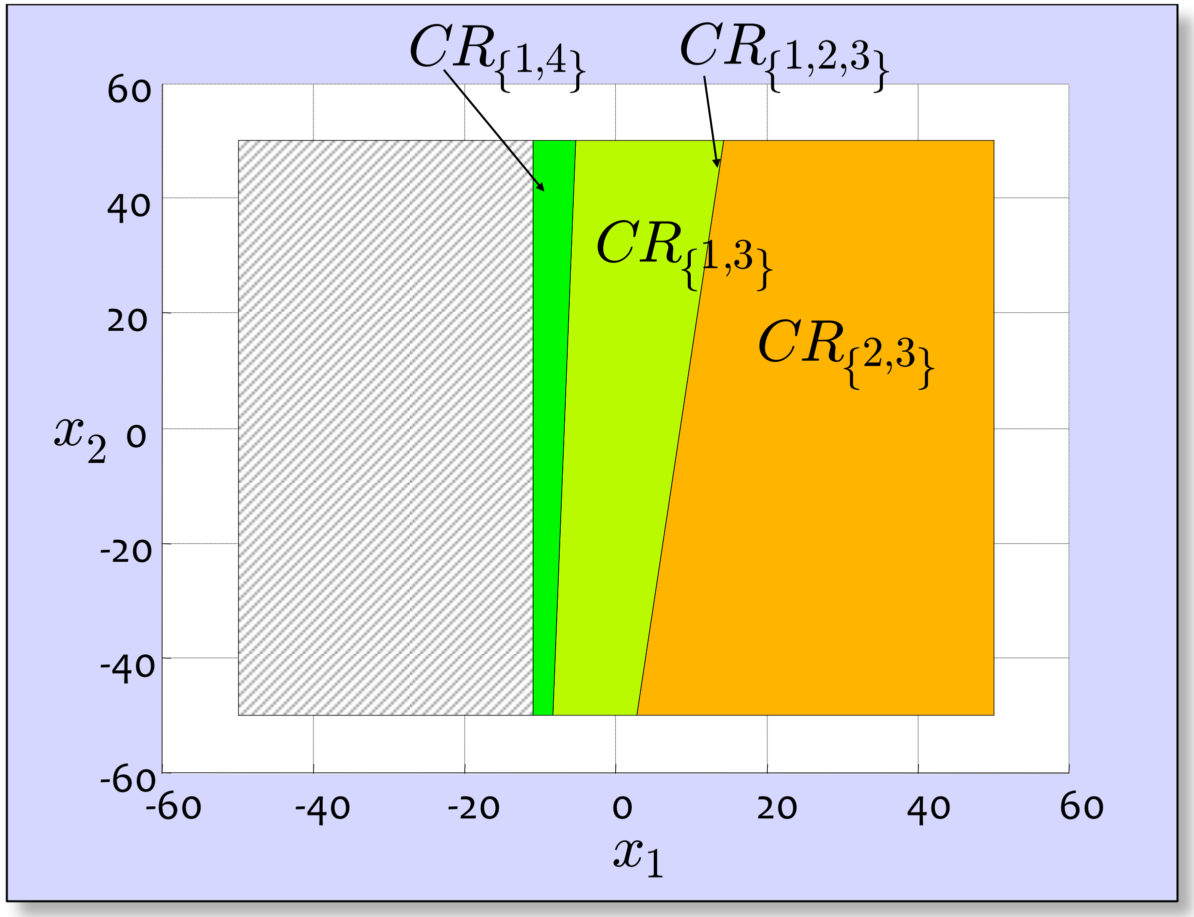
$$\begin{array}{ll} \min_{\xi} & \sum_{k=0}^{T-1} \epsilon_i^x + \epsilon_i^u \\ \text{s.t.} & G\xi \leq W + S \left[\boxed{\begin{matrix} x(t) \\ r(t) \end{matrix}}\right] \end{array}$$

multi-parametric Mixed Integer Linear Program (mp-MILP)

Example of Multiparametric Solution

Multiparametric LP

$$
\begin{array}{ll}
\min\limits_{\xi} & -3\xi_1 - 8\xi_2 \\
\text{s.t.} & \left\{\begin{array}{lcl}
\xi_1 + \xi_2 & \leq & 13 + \enclose{circle}{x_1} \\
5\xi_1 - 4\xi_2 & \leq & 20 \\
-8\xi_1 + 22\xi_2 & \leq & 121 + \enclose{circle}{x_2} \\
-4\xi_1 - \xi_2 & \leq & -8 \\
-\xi_1 & \leq & 0 \\
-\xi_2 & \leq & 0
\end{array}\right.
\end{array}
$$



$$
\xi(x) = \left\{\begin{array}{lll}
\left[\begin{smallmatrix} 0.00 & 0.05 \\ 0 & 0.06 \end{smallmatrix}\right] x + \left[\begin{smallmatrix} 11.85 \\ 9.80 \end{smallmatrix}\right] & \text{if} \begin{bmatrix} 0.02 & 0.00 \\ 0.00 & 0.02 \\ 0.00 & -0.02 \\ -0.12 & 0.01 \end{bmatrix} x \leq \begin{bmatrix} 1.00 \\ 1.00 \\ 1.00 \\ -1.00 \end{bmatrix} & \text{CR}_{\{2,3\}} \\
\left[\begin{smallmatrix} 0.73 & -0.03 \\ 0.27 & 0.03 \end{smallmatrix}\right] x + \left[\begin{smallmatrix} 5.50 \\ 7.50 \end{smallmatrix}\right] & \text{if} \begin{bmatrix} 0.00 & 0.02 \\ 0.00 & -0.02 \\ 0.12 & -0.01 \\ -0.15 & 0.00 \end{bmatrix} x \leq \begin{bmatrix} 1.00 \\ 1.00 \\ 1.00 \\ 1.00 \end{bmatrix} & \text{CR}_{\{1,3\}} \\
\left[\begin{smallmatrix} -0.33 & 0.00 \\ 1.33 & 0 \end{smallmatrix}\right] x + \left[\begin{smallmatrix} -1.67 \\ 14.67 \end{smallmatrix}\right] & \text{if} \begin{bmatrix} 0.00 & 0.02 \\ 0.00 & -0.02 \\ 0.15 & -0.00 \\ -0.09 & 0.00 \end{bmatrix} x \leq \begin{bmatrix} 1.00 \\ 1.00 \\ -1.00 \\ 1.00 \end{bmatrix} & \text{CR}_{\{1,4\}}
\end{array}\right.
$$

Multiparametric MILP

$$\begin{aligned} \min_{\xi=\{\xi_c,\xi_d\}} \quad & f'\xi_c + d'\xi_d \\ \text{s.t.} \quad & G\xi_c + E\xi_d \leq W + Fx \end{aligned}$$

$$\xi_c \in \mathbb{R}^n$$

$$\xi_d \in \{0,1\}^m$$

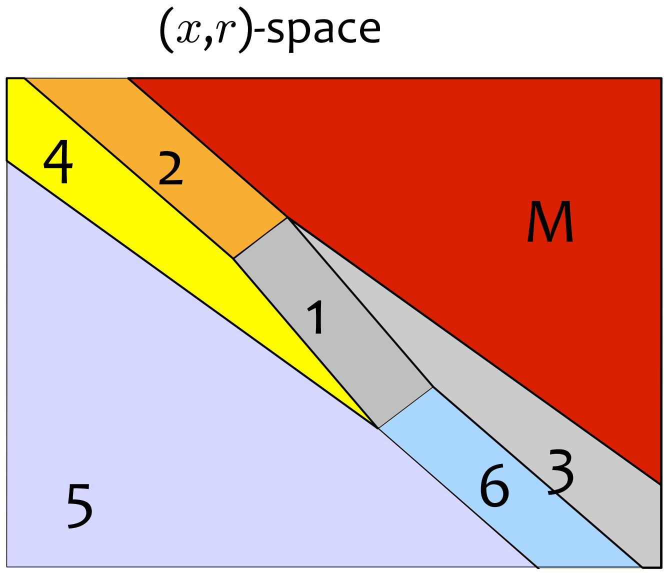
- mp-MILP can be solved (by alternating MILPs and mp-LPs)

(Dua, Pistikopoulos, 1999)

- **Theorem:** The multiparametric solution is piecev$\xi^*(x)$fine

- **The MPC controller is piecewise affine in x,r**

$$u(x,r) = \begin{cases} F_1x + E_1r + g_1 & \text{if} \quad H_1 \left[\begin{smallmatrix} x \\ r \end{smallmatrix}\right] \leq K_1 \\ \vdots & \quad \vdots \\ F_Mx + E_Mr + g_M & \text{if} \quad H_M \left[\begin{smallmatrix} x \\ r \end{smallmatrix}\right] \leq K_M \end{cases}$$



Explicit Hybrid MPC (PWA)

$$\min_U J(U,x,r) = \sum_{k=0}^{T-1} \|R(y(k)-r)\|_p + \|Qu(k)\|_p$$

$$\text{subject to} \begin{cases} \text{PWA model} \\ x(0) = x \end{cases}$$

$p = 1, 2, \infty$

$\|v\|_2 = v'v$

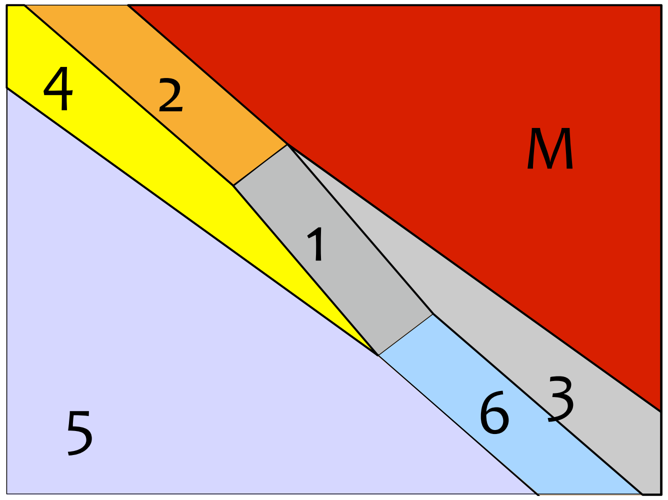
$\|v\|_\infty = \max |v_i|$

$\|v\|_1 = \sum v_i$

- **The MPC controller is piecewise affine in x,r**

$$u(x,r) = \begin{cases} F_1 x + E_1 r + g_1 & \text{if} \quad H_1 \left[\begin{smallmatrix} x \\ r \end{smallmatrix}\right] \leq K_1 \\ \vdots & \vdots \\ F_M x + E_M r + g_M & \text{if} \quad H_M \left[\begin{smallmatrix} x \\ r \end{smallmatrix}\right] \leq K_M \end{cases}$$

(x,r)-space



Note: in the 2-norm case the partition may not be fully polyhedral

Computation of Explicit Hybrid MPC (PWA)

Method A:

(Borrelli, Baotic, Bemporad, Morari, *Automatica*, 2005)

Use a combination of DP (dynamic programming) and mpLP (1-norm, ∞-norm), or mpQP (quadratic forms)

Method B:

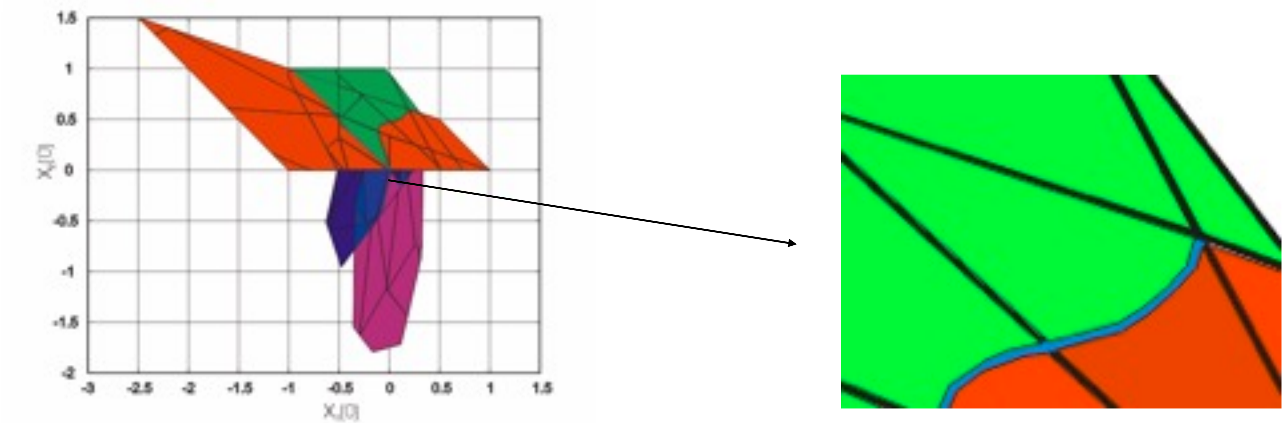
(Bemporad, *Hybrid Toolbox*, 2003) (Alessio, Bemporad, ADHS 2006)(Mayne, ECC 2001)

1 - Use backwards (=DP) reachability analysis for enumerating all feasible mode sequences $I=\{i(0),i(1),\ldots,i(T)\}$;

2 - For each fixed sequence I, solve the explicit finite-time optimal control problem for the corresponding linear time-varying system (mpQP or mpLP);

3 - Case 1/∞-norm: Compare value functions and split regions. Quadratic case: keep overlapping regions (possibly eliminate overlaps that are never optimal) and compare on-line (if needed).

Note: in the 2-norm case, the fully explicit partition may not be polyhedral



Hybrid Control Examples (Revisited)

Hybrid control example

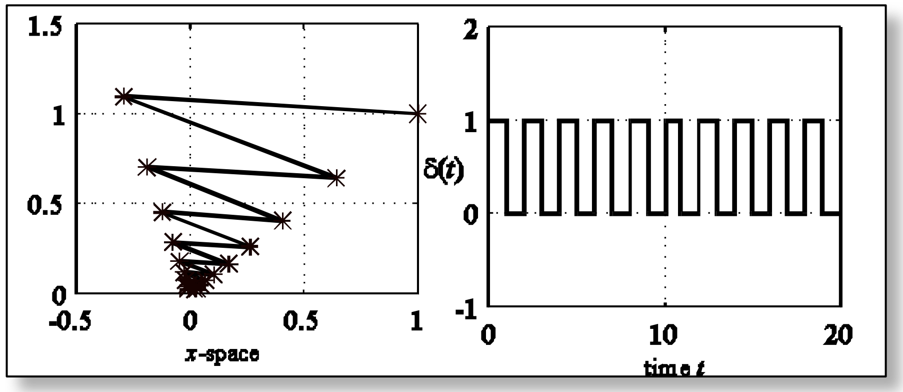
PWA system:

$$
\begin{aligned}
x(t+1) &= 0.8\begin{bmatrix}\cos\alpha(t) & -\sin\alpha(t)\\ \sin\alpha(t) & \cos\alpha(t)\end{bmatrix}x(t)+\begin{bmatrix}0\\1\end{bmatrix}u(t)\\
y(t) &= x_2(t)\\
\alpha(t) &= \begin{cases}\frac{\pi}{3} & \text{if } x_1(t)\geq 0\\ -\frac{\pi}{3} & \text{if } x_1(t)<0\end{cases}
\end{aligned}
$$

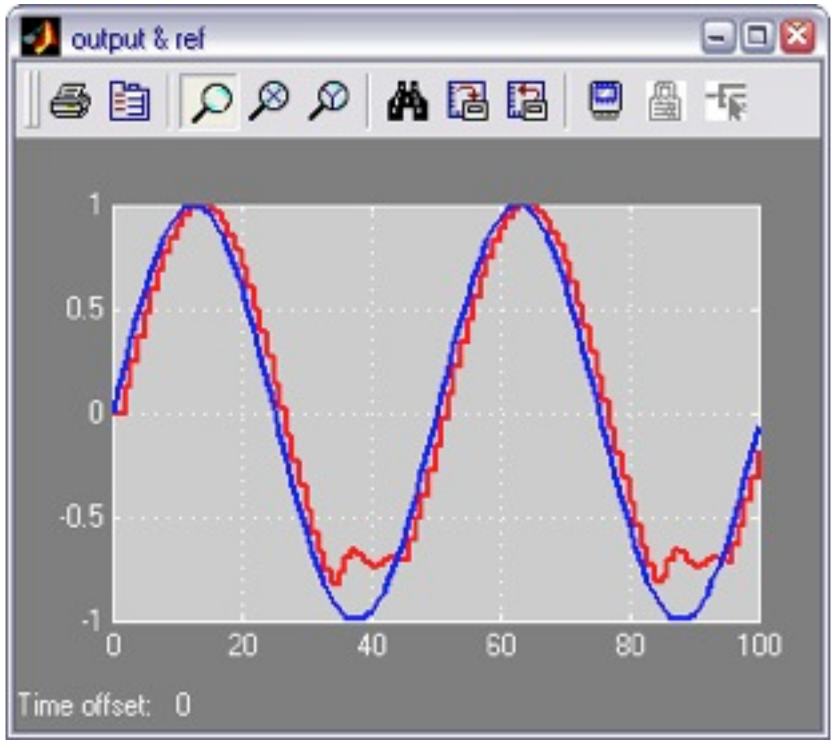
Constraints: $-1 \leq u(t) \leq 1$

Objective: $\min \sum_{k=1}^{2} |y(t+k|t) - r(t)|$

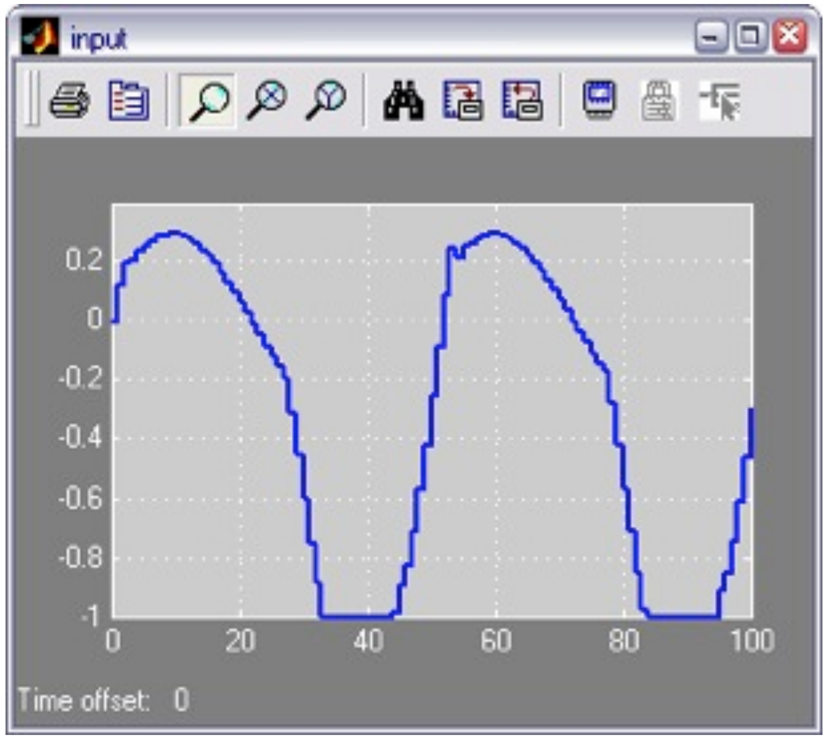
Open loop behavior:



Closed loop:



Time offset: 0

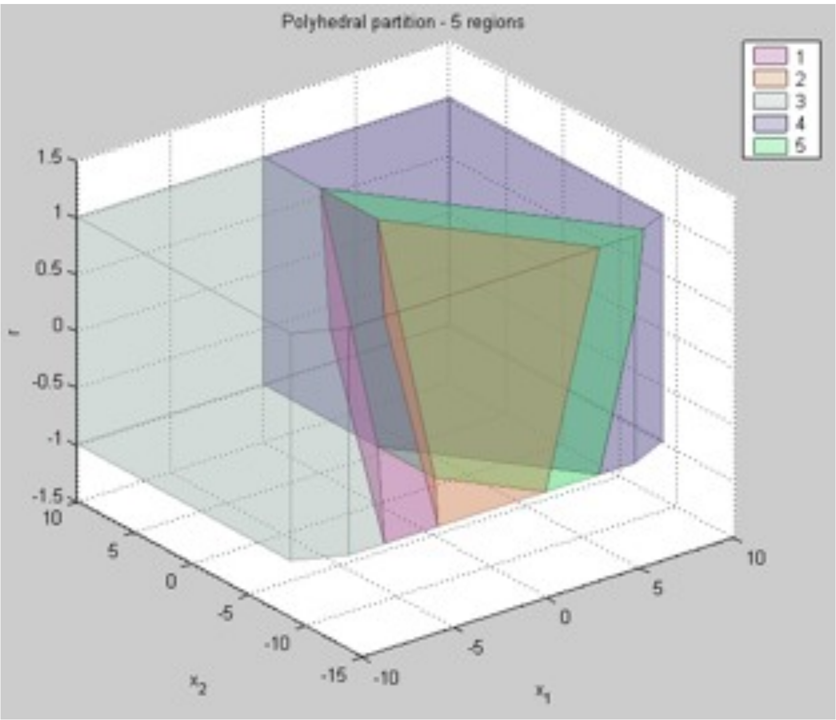


Time offset: 0

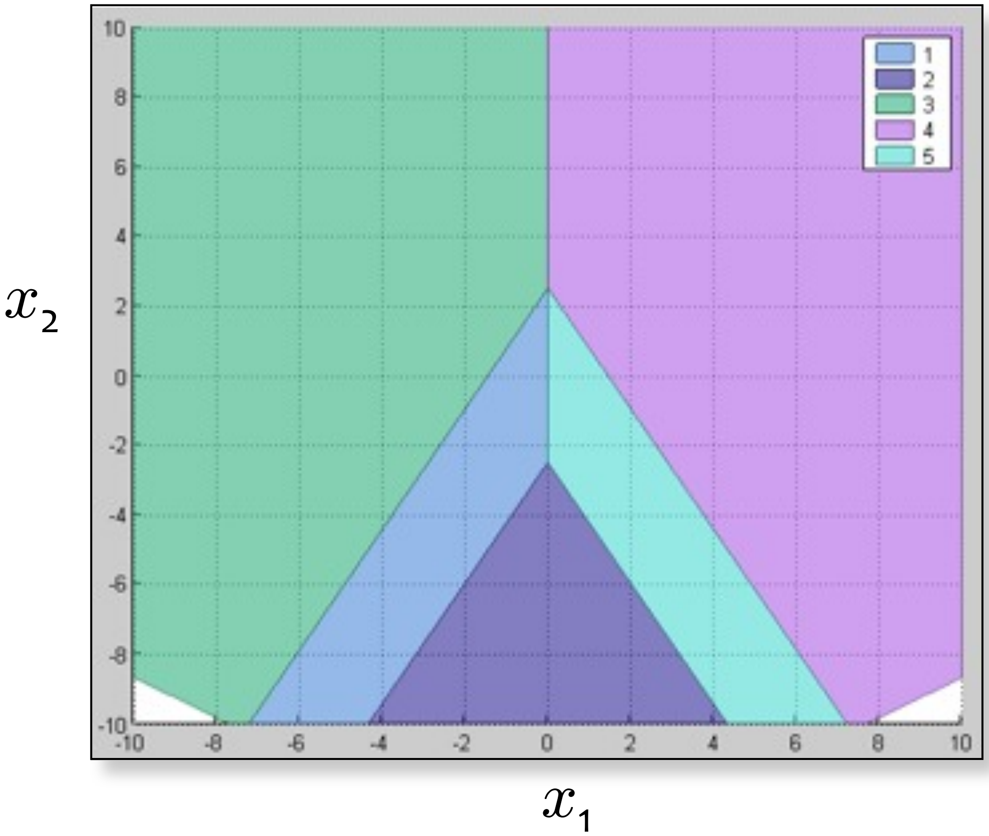
HybTbx: /`demos/hybrid/bm99sim.m`

Explicit PWA Controller

$$u(x,r) = \begin{cases} [\,0.6928\ \ -0.4\ \ 1\,]\left[\begin{smallmatrix}x\\r\end{smallmatrix}\right] & \text{if } \begin{bmatrix} 0.6928 & -0.4 & 1 \\ -0.4 & -0.6928 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \\ -0.6928 & 0.4 & -1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \left[\begin{smallmatrix}x\\r\end{smallmatrix}\right] \le \begin{bmatrix} 1 \\ 10 \\ 10 \\ 1 \\ 1 \\ 1 \\ 1e{-}006 \end{bmatrix} \\ & \text{(Region \#1)} \\ 1 & \text{if } \begin{bmatrix} -0.6928 & 0.4 & -1 \\ 0.6928 & 0.4 & -1 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{bmatrix} \left[\begin{smallmatrix}x\\r\end{smallmatrix}\right] \le \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \\ 10 \end{bmatrix} \\ & \text{(Region \#2)} \\ -1 & \text{if } \begin{bmatrix} -0.4 & -0.6928 & 0 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0.6928 & -0.4 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \\ -1 & 0 & 0 \end{bmatrix} \left[\begin{smallmatrix}x\\r\end{smallmatrix}\right] \le \begin{bmatrix} 10 \\ 10 \\ 1e{-}006 \\ 10 \\ -1 \\ 1 \\ 1 \\ 10 \end{bmatrix} \\ & \text{(Region \#3)} \\ -1 & \text{if } \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0.4 & -0.6928 & 0 \\ 0 & -1 & 0 \\ -0.6928 & -0.4 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix} \left[\begin{smallmatrix}x\\r\end{smallmatrix}\right] \le \begin{bmatrix} 0 \\ 10 \\ 10 \\ 10 \\ -1 \\ 1 \\ 1 \\ 10 \end{bmatrix} \\ & \text{(Region \#4)} \\ [\,-0.6928\ \ -0.4\ \ 1\,]\left[\begin{smallmatrix}x\\r\end{smallmatrix}\right] & \text{if } \begin{bmatrix} -0.6928 & -0.4 & 1 \\ 0.4 & -0.6928 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \\ 0.6928 & 0.4 & -1 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{bmatrix} \left[\begin{smallmatrix}x\\r\end{smallmatrix}\right] \le \begin{bmatrix} 1 \\ 10 \\ 10 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} \\ & \text{(Region \#5)} \end{cases}$$



Section with r=0



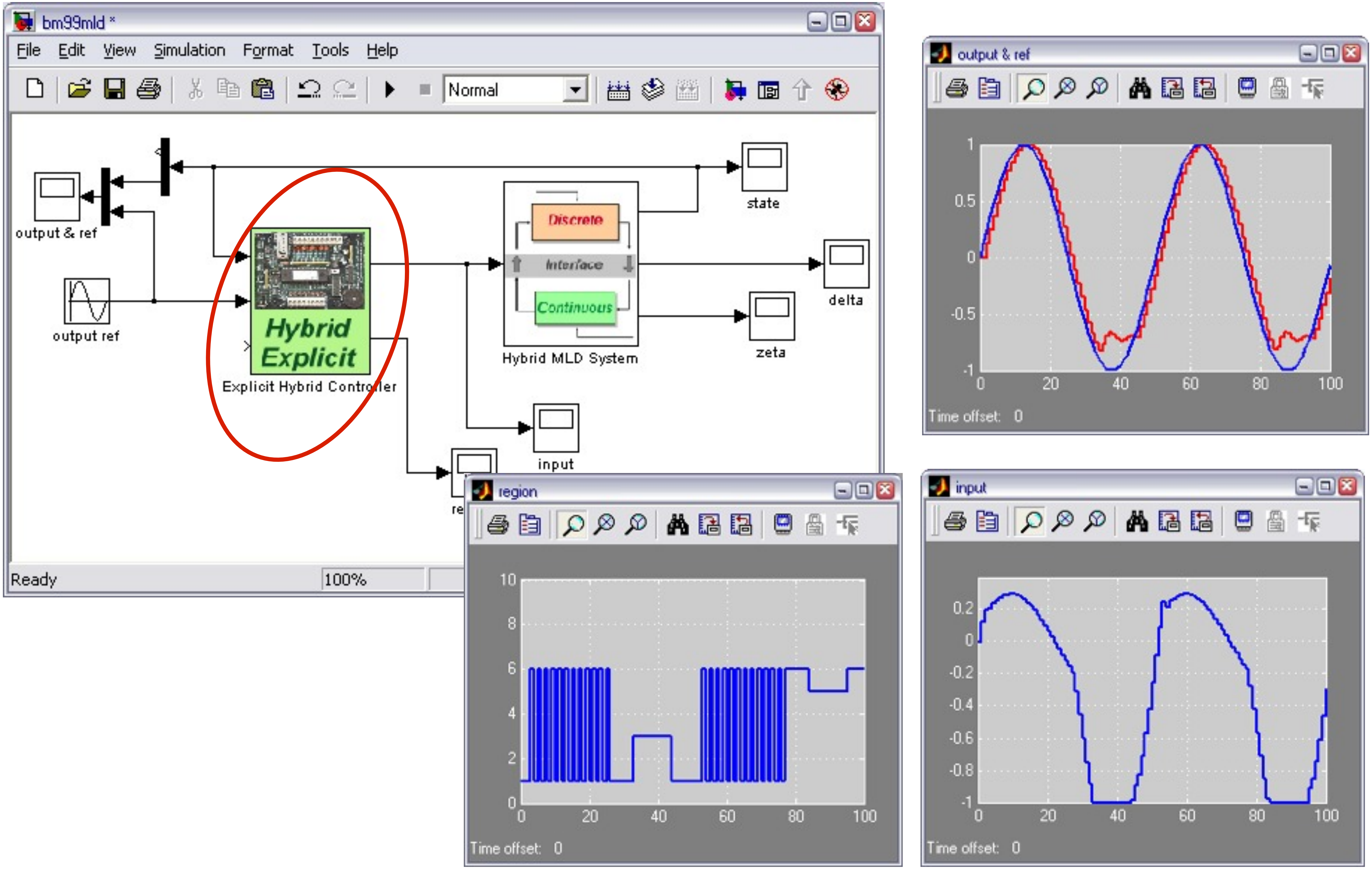
HybTbx: **/demos/hybrid/bm99sim.m**

(CPU time: 1.51 s, Pentium M 1.4GHz)

PWA law $\equiv$ MPC law !

Hybrid control example

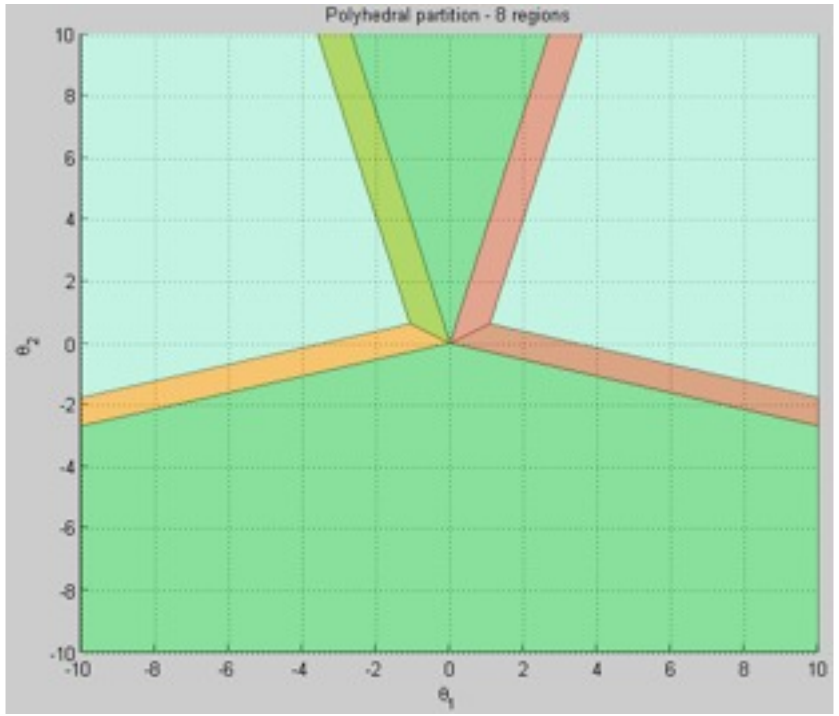
Closed loop:



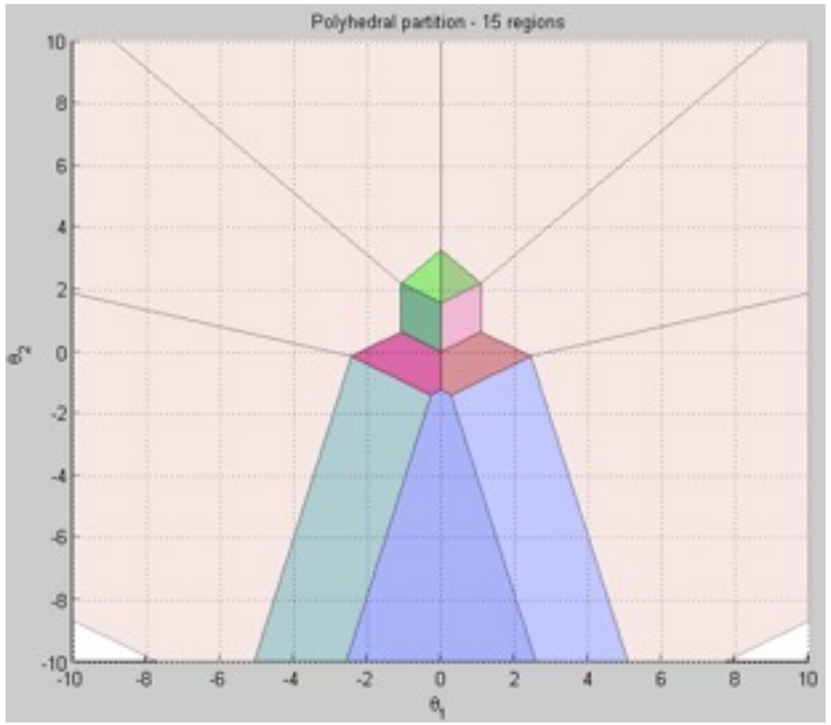
Explicit PWA Regulator

Objective: $$\min \sum_{k=1}^{N} \|x(t+k|t)\|_\infty$$

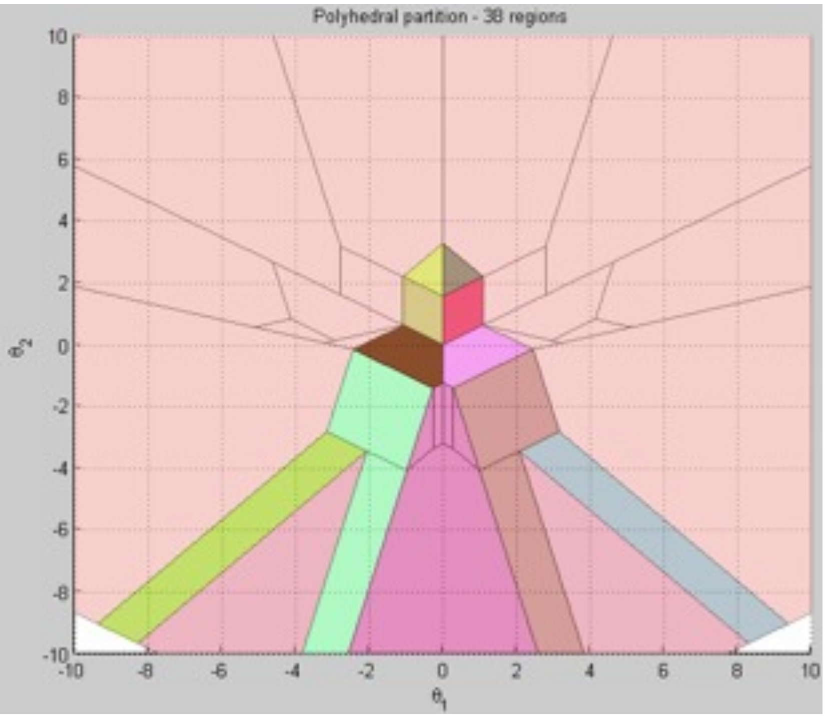
Prediction horizon N=1



Prediction horizon N=2



Prediction horizon N=3

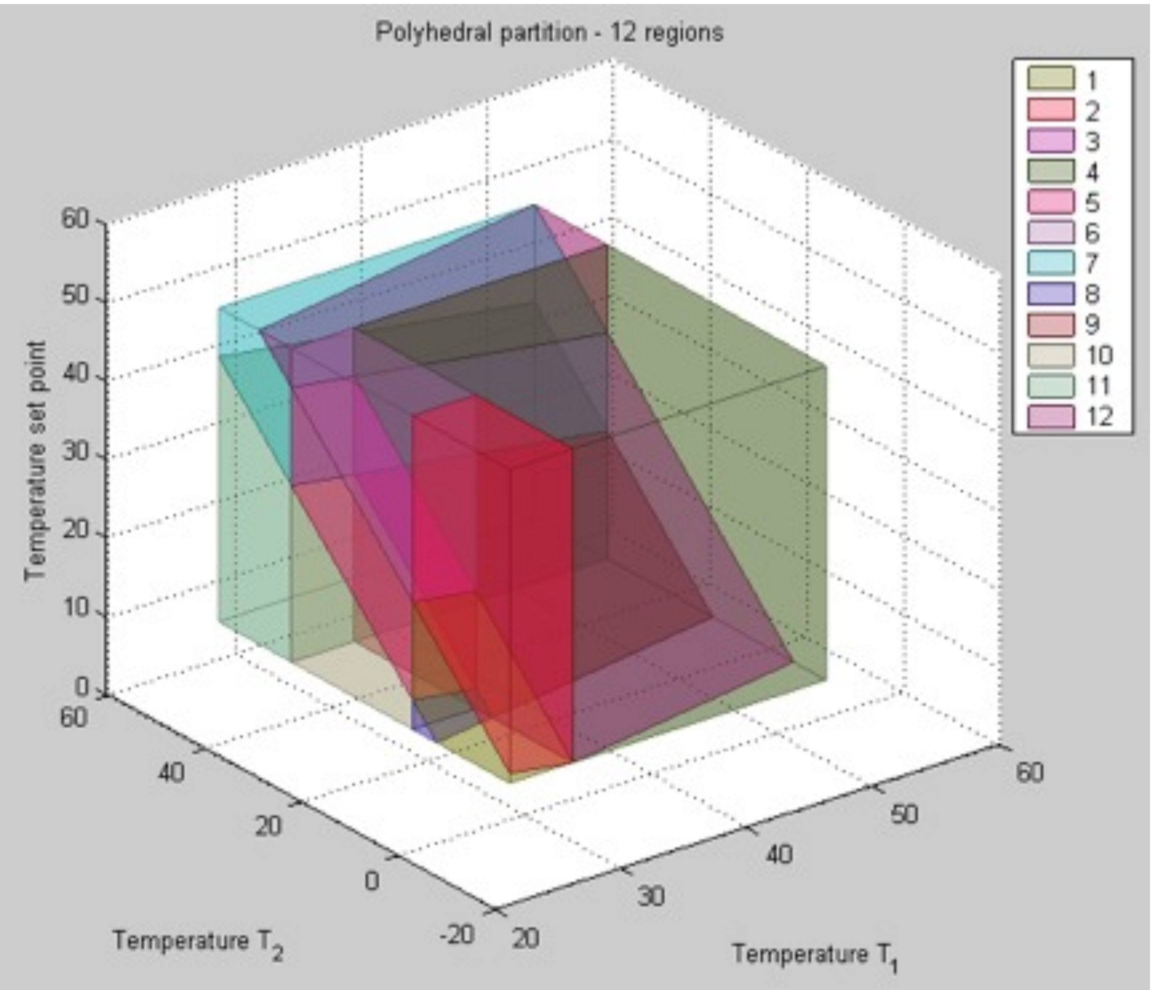


HybTbx: `/demos/hybrid/bm99benchmark.m`

Explicit MPC – Temperature Control

```
>>E=expcon(C,range,options);
```

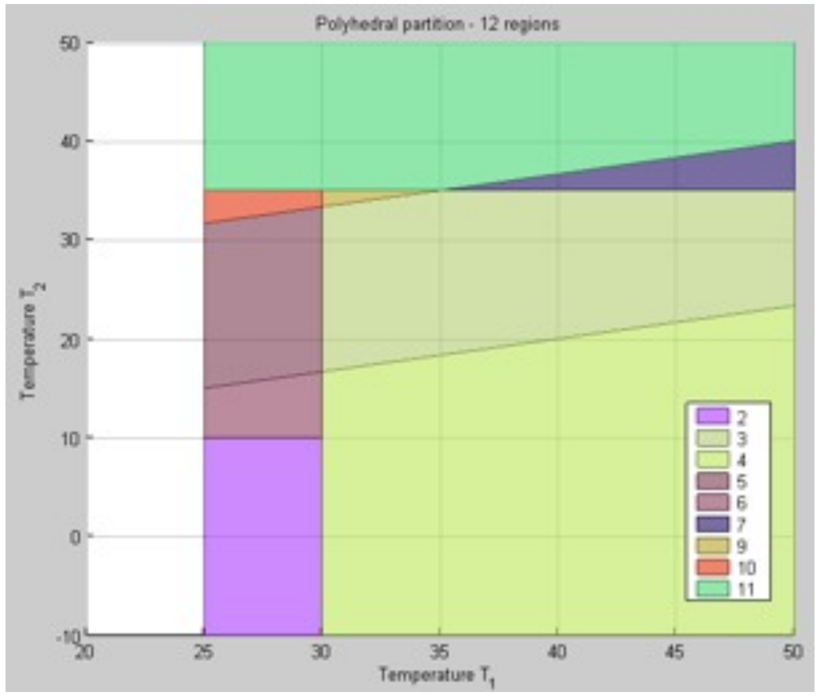
```
>> E

Explicit controller (based on hybrid controller C)
   3 parameter(s)
   1 input(s)
  12 partition(s)
sampling time = 0.5

The controller is for hybrid systems (tracking)
This is a state-feedback controller.

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



$$\min \sum_{k=1}^{2} (x_2(k) - r)^2$$

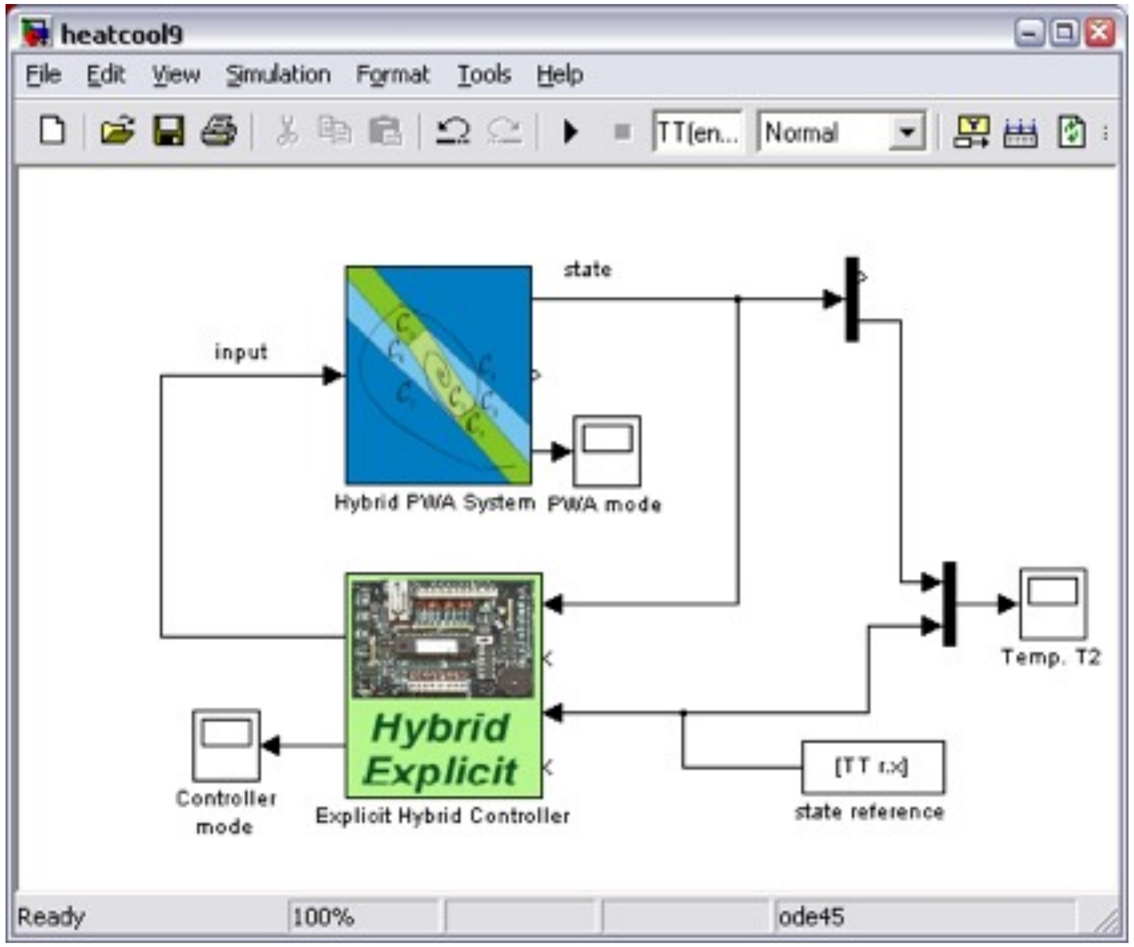
$$\text{s.t.} \quad x_1(k) \geq 25 \;\; k = 1, 2$$

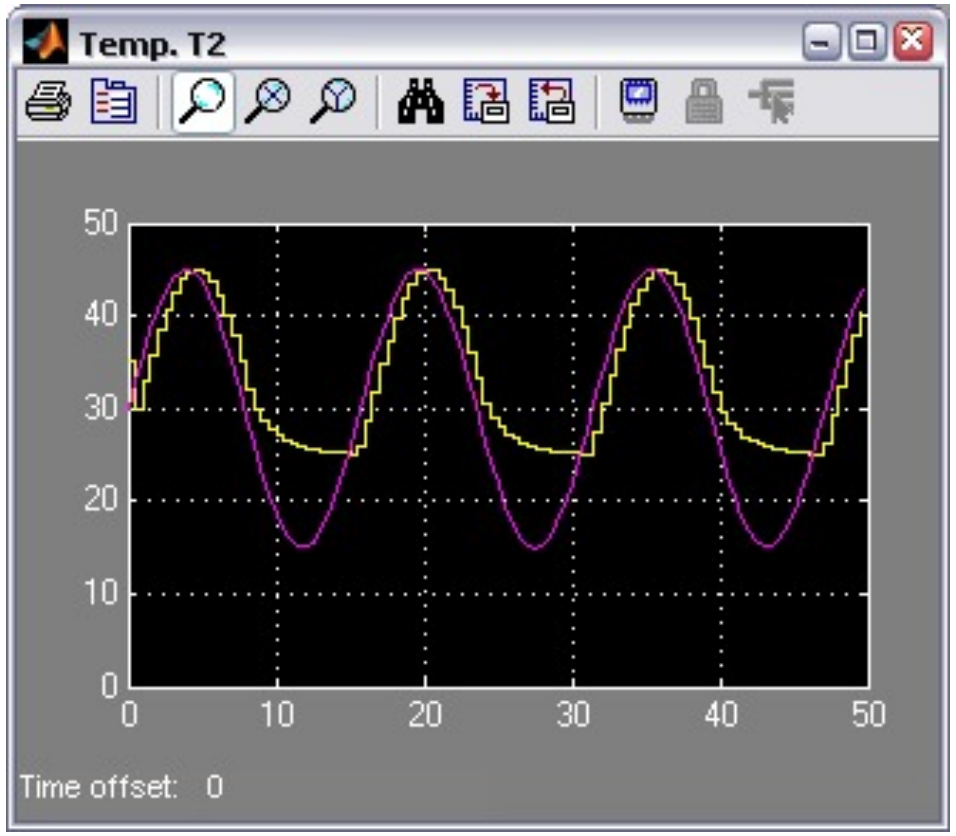
PWA model



Section in the (T$_1$,T$_2$)-space for T$_{ref}$ = 30

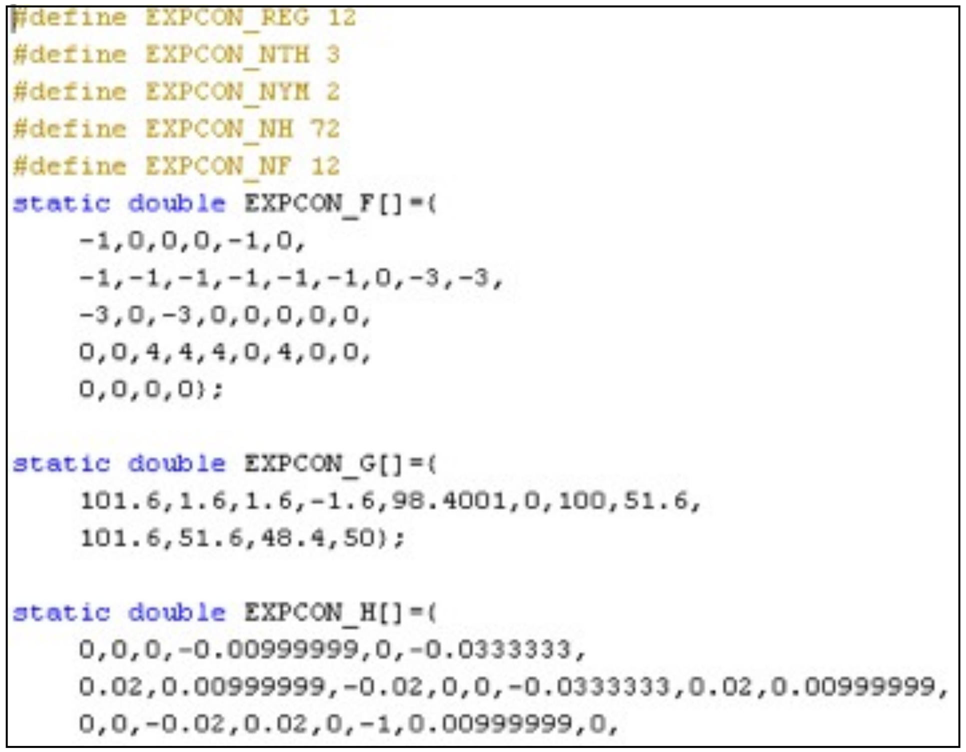
Explicit MPC – Temperature Control



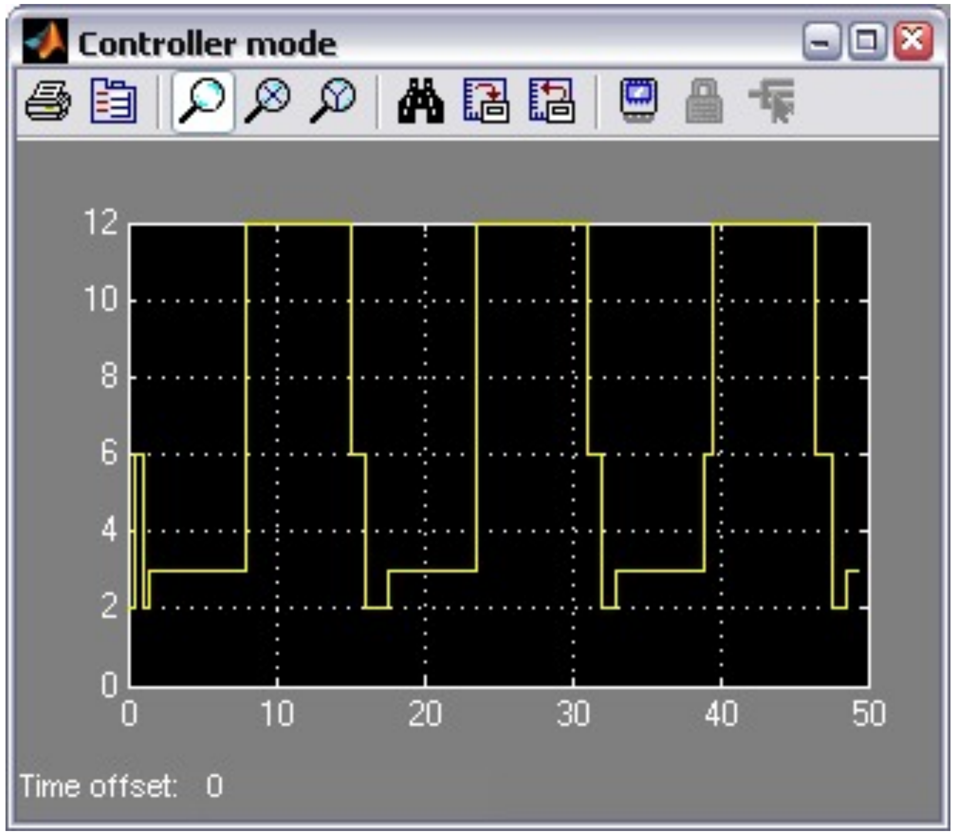


Generated C-code

`utils/expcon.h`



```
#define EXPCON_REG 12
#define EXPCON_NTH 3
#define EXPCON_NYM 2
#define EXPCON_NH 72
#define EXPCON_NF 12
static double EXPCON_F[]={
    -1,0,0,0,-1,0,
    -1,-1,-1,-1,-1,-1,0,-3,-3,
    -3,0,-3,0,0,0,0,0,
    0,0,4,4,4,0,4,0,0,
    0,0,0,0};

static double EXPCON_G[]={
    101.6,1.6,1.6,-1.6,98.4001,0,100,51.6,
    101.6,51.6,48.4,50};

static double EXPCON_H[]={
    0,0,0,-0.00999999,0,-0.0333333,
    0.02,0.00999999,-0.02,0,0,-0.0333333,0.02,0.00999999,
    0,0,-0.02,0.02,0,-1,0.00999999,0,
```



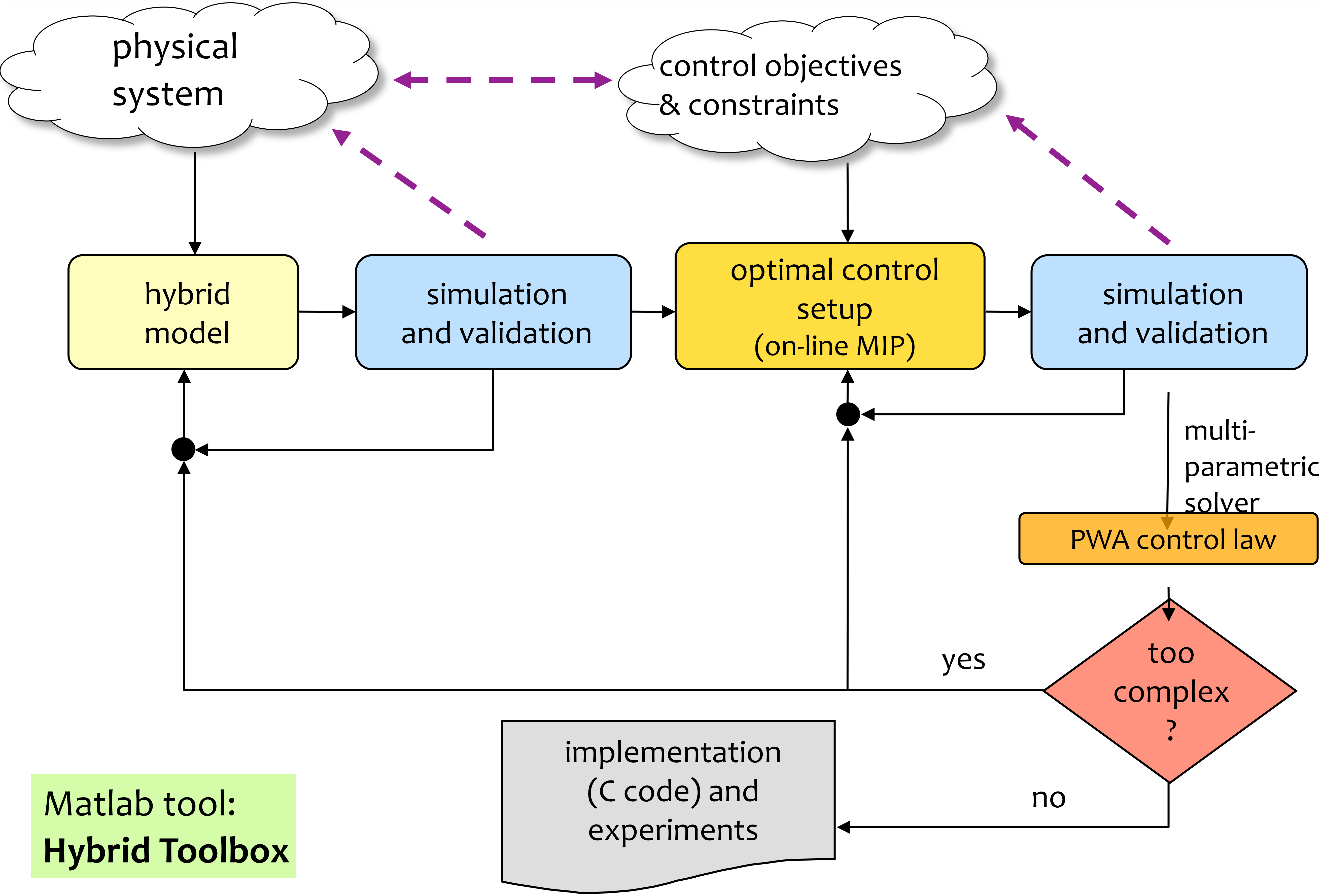
Implementation Aspects of Hybrid MPC

- **Alternatives:** (1) solve MIP on-line
  (2) evaluate a PWA function

- Small problems (short horizon N=1,2, one or two inputs): explicit PWA control law preferable
  - time to evaluate the control law is shorter than MIP
  - control code is simpler (no complex solver must be included in the control software !)
  - more insight in controller's behavior

- Medium/large problems (longer horizon, many inputs and binary variables): MIP preferable

Hybrid control design flow


physical system
control objectives & constraints
hybrid model
simulation and validation
optimal control setup (on-line MIP)
simulation and validation
multi-parametric solver
PWA control law
too complex ?
yes
no
implementation (C code) and experiments

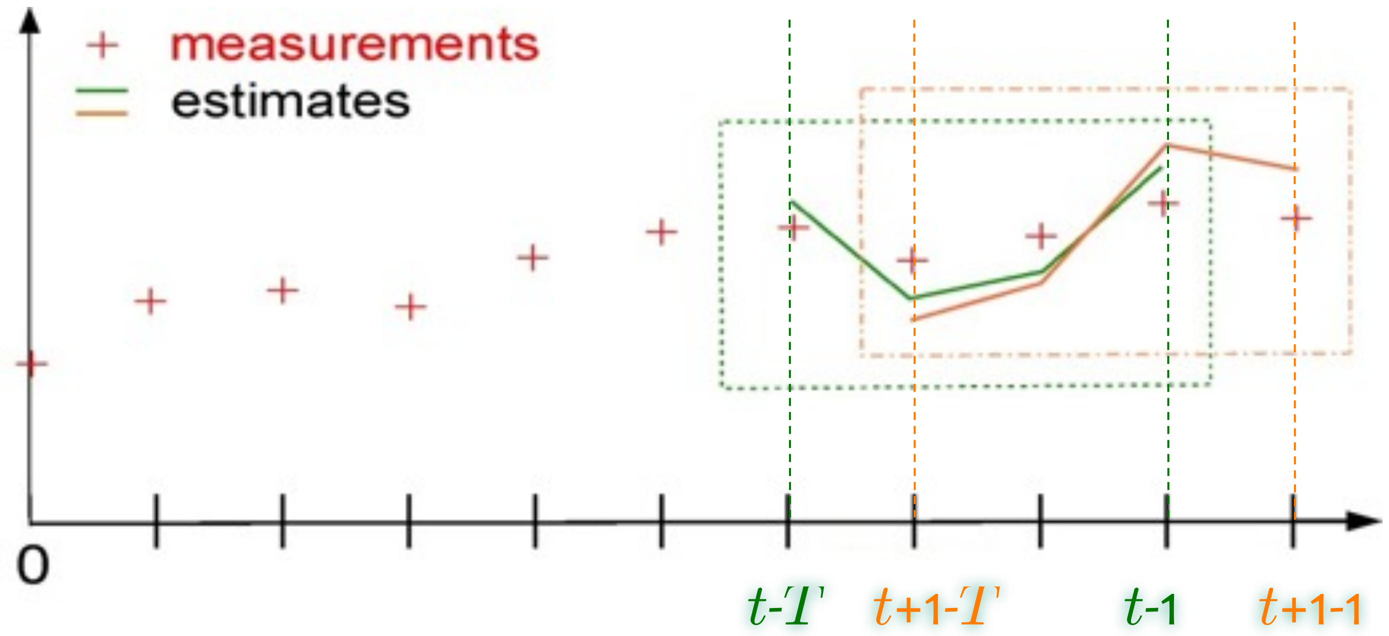
Matlab tool:
Hybrid Toolbox

Moving Horizon Estimation
Fault Detection & Isolation

State Estimation / Fault Detection

- Problem: given past output measurements and inputs, estimate the current states and faults

- Solution: Use Moving Horizon Estimation for MLD systems (dual of MPC)



Augment the MLD model with:

- Input disturbances $\xi \in \mathbb{R}^n$
- Output disturbances $\zeta \in \mathbb{R}^p$

At each time t solve the problem:

$$\boxed{\min \sum_{k=1}^{T} \|\hat{y}(t-k|t) - y(t-k)\|^2 + \ldots}$$

and get estimate $\hat{x}(t)$

→ MHE optimization = MIQP

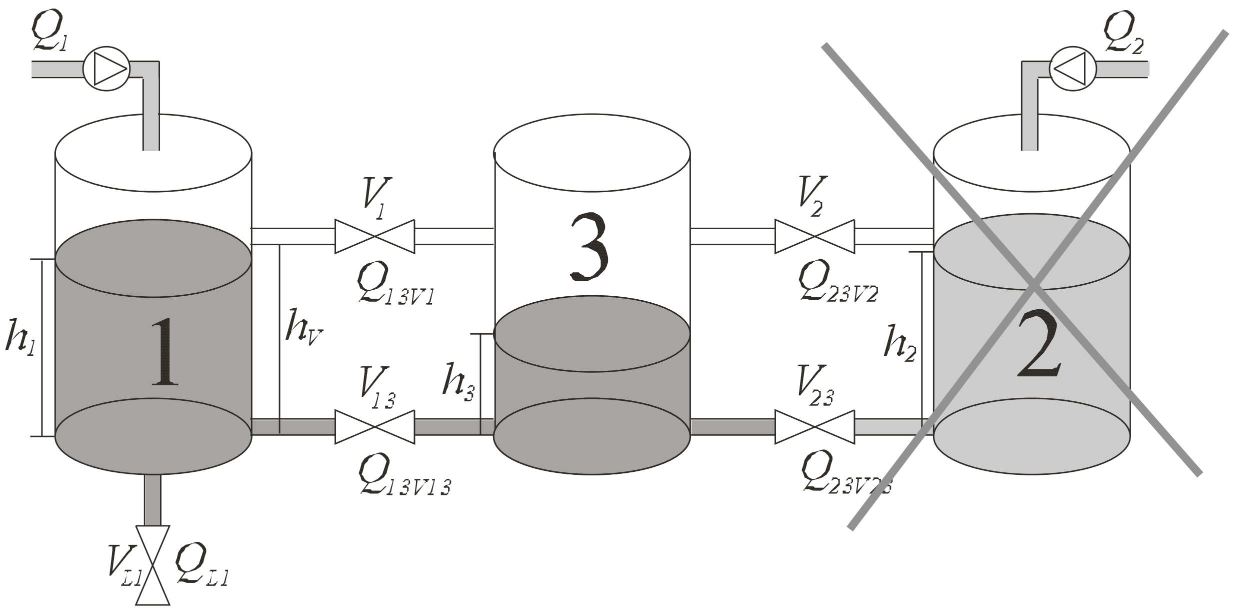
→ Convergence can be guaranteed

(Bemporad, Mignone, Morari, ACC 1999)

(Ferrari-T., Mignone, Morari, 2002)

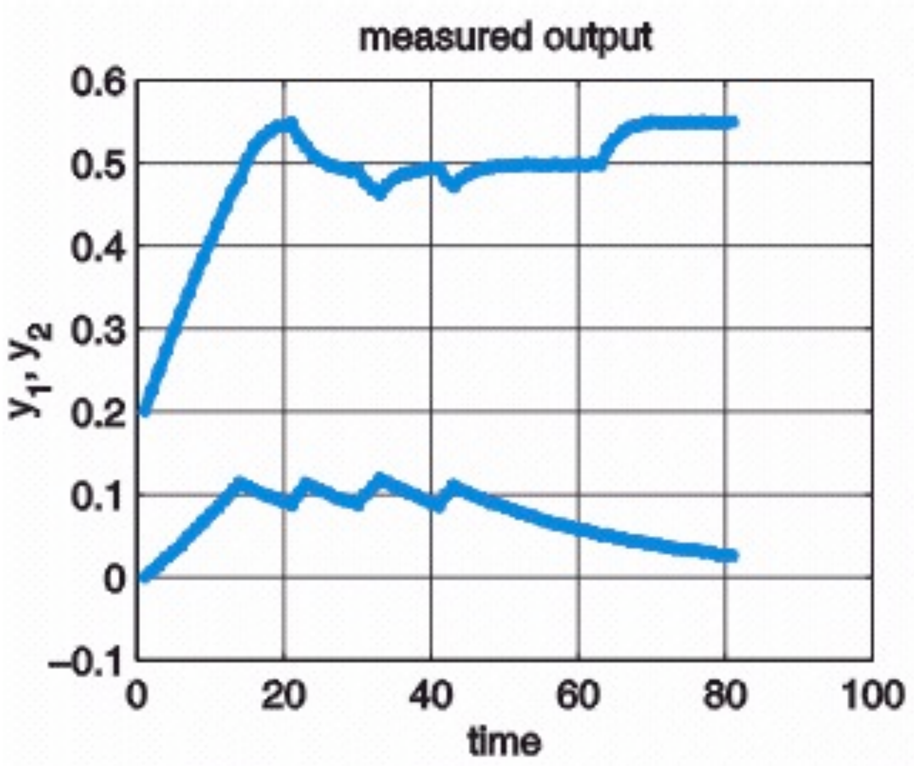
Fault detection: augment MLD with unknown **binary** distubances $\phi \in \{0,1\}^{n_f}$

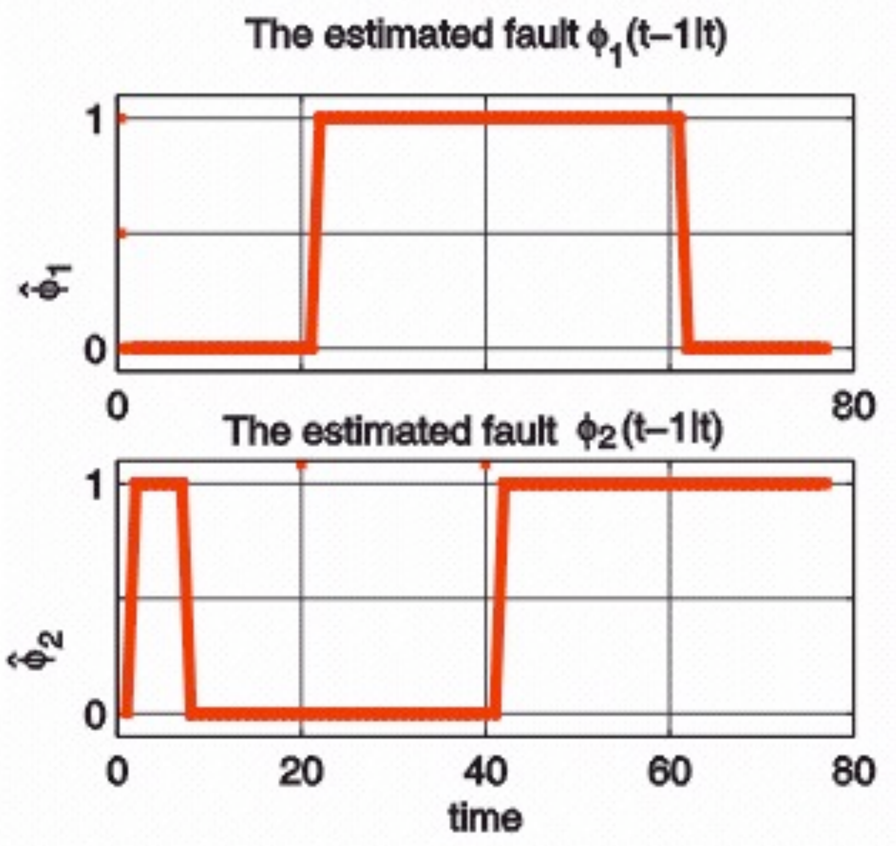
Example: Three Tank System



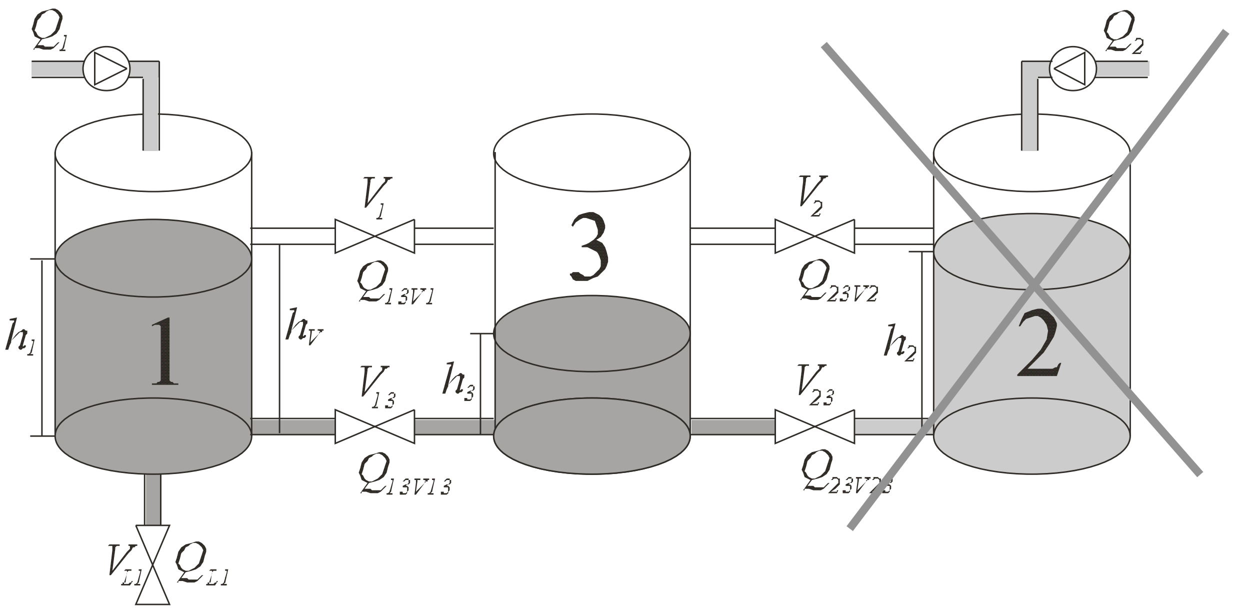
COSY Benchmark problem, ESF

- ϕ_1: leak in tank 1
  for $20\text{s} \le t \le 60\text{s}$
- ϕ_2: valve V_1 blocked
  for $t \ge 40\text{s}$





Example: Three Tank System



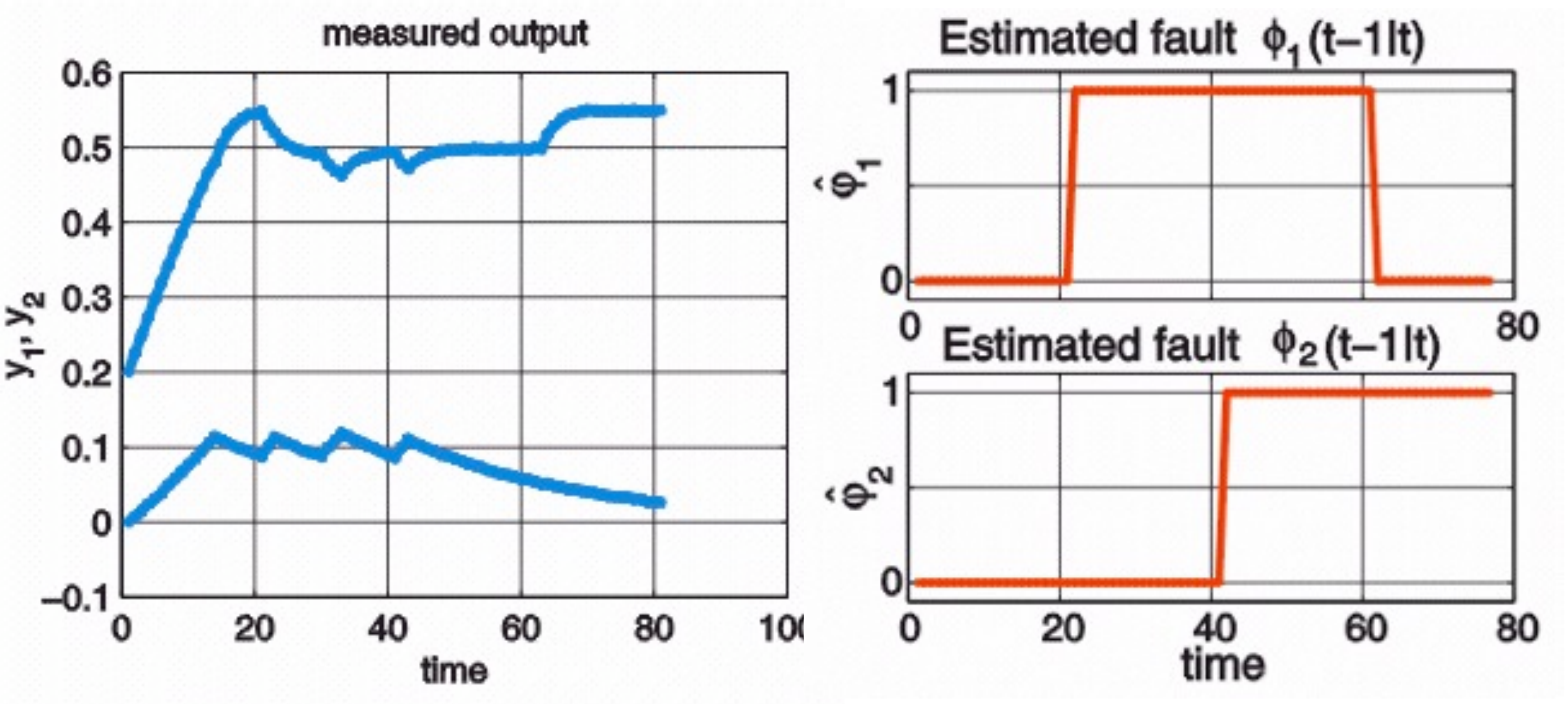
COSY Benchmark problem, ESF

- ϕ_1: leak in tank 1
  for $20\text{s} \leq t \leq 60\text{s}$

- ϕ_2: valve V_1 blocked
  for $t \geq 40\text{s}$

- Add logic constraint

$$[h_1 \leq h_v] \Rightarrow \phi_2 = 0$$



A Few Hybrid MPC Tricks …

Measured Disturbances

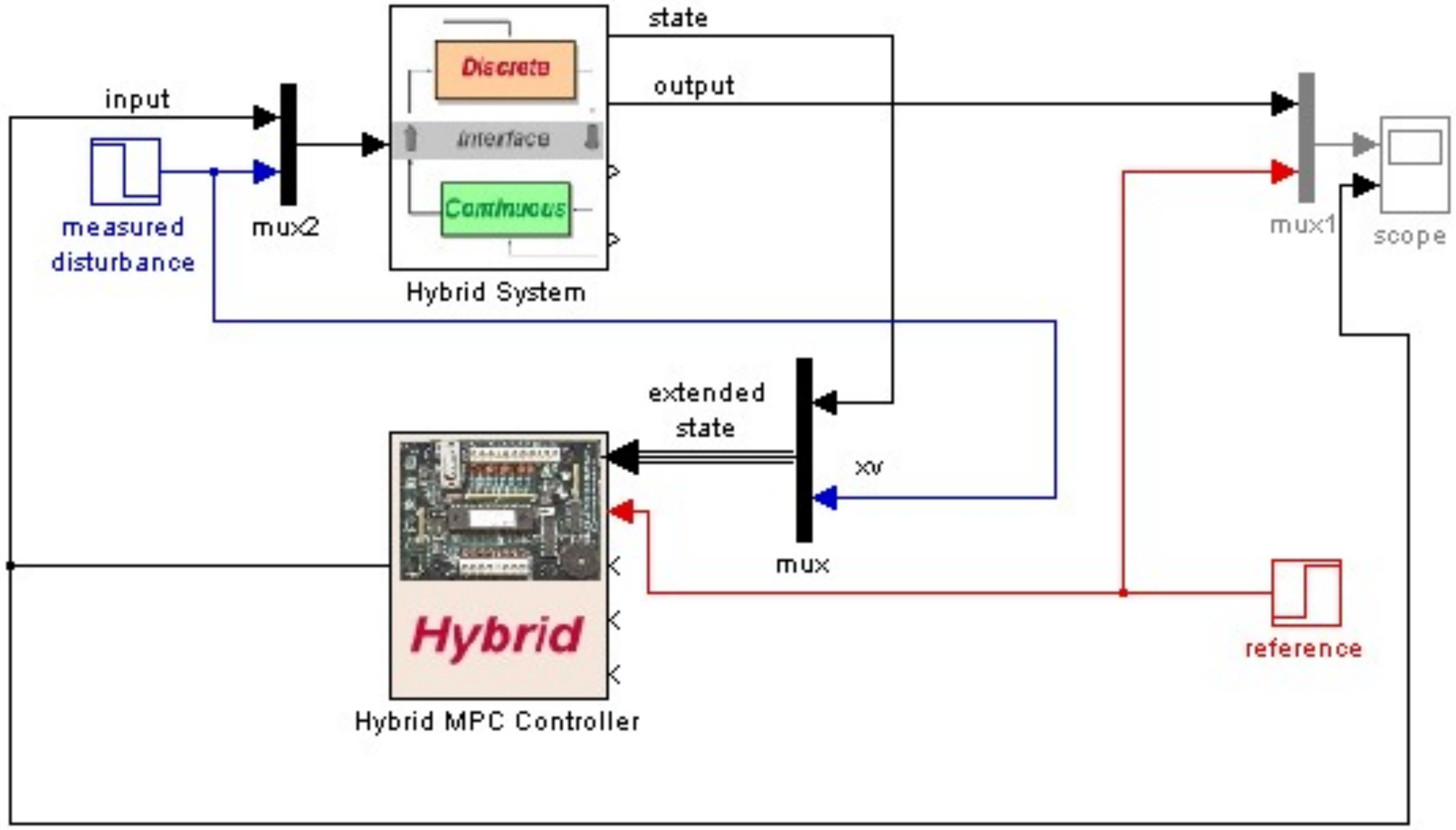
- Disturbance $v(k)$ can be measured at time k

- Augment the hybrid prediction model with a constant state

$$\boxed{x_v(k+1) = x_v(k)}$$

- In Hysdel:

```
INTERFACE{
      STATE{
         REAL x      [-1e3, 1e3];
         REAL xv     [-1e3, 1e3];
      }
      ...
   }
IMPLEMENTATION{
      CONTINUOUS{
         x = A*x + B*u + Bv*xv
         xv= xv;
         ...
      }
}
```



/demos/hybrid/hyb_meas_dist.m

Note: same trick applies to linear MPC

Hybrid MPC - Tracking

- Optimal control problem (quadratic performance index):

$$\min_{\Delta U} \quad \sum_{k=0}^{N-1} \|W^y(y(k+1)-r(t))\|^2 + \|W^{\Delta u}\Delta u(k)\|^2$$

$$[\Delta u(k) \triangleq u(k)-u(k-1)]$$

$$\begin{aligned} \text{subj. to} \quad & u_{\text{min}} \leq u(k) \leq u_{\text{max}}, \ \ k=0,\ldots,N-1 \\ & \Delta u_{\text{min}} \leq \Delta u(k) \leq \Delta u_{\text{max}}, \ \ k=0,\ldots,N-1 \\ & y_{\text{min}} \leq y(k) \leq y_{\text{max}}, \ \ k=1,\ldots,N \end{aligned}$$

- Optimization problem: (MIQP)

$$\min_{\Delta U} \quad J(\Delta U, x(t)) = \tfrac{1}{2}\Delta U' H \Delta U + [x'(t) \ \ r'(t) \ \ u'(t-1)]F\Delta U$$

$$\text{s.t.} \quad G\Delta U \leq W + K \begin{bmatrix} x(t) \\ r(t) \\ u(t-1) \end{bmatrix}$$

Note: same trick as in linear MPC

Integral Action in Hybrid MPC

- Augment the hybrid prediction model with integrators of output errors as additional states:

$$\boxed{\epsilon(k+1) = \epsilon(k) + T_s \cdot (r(k) - y(k))}$$

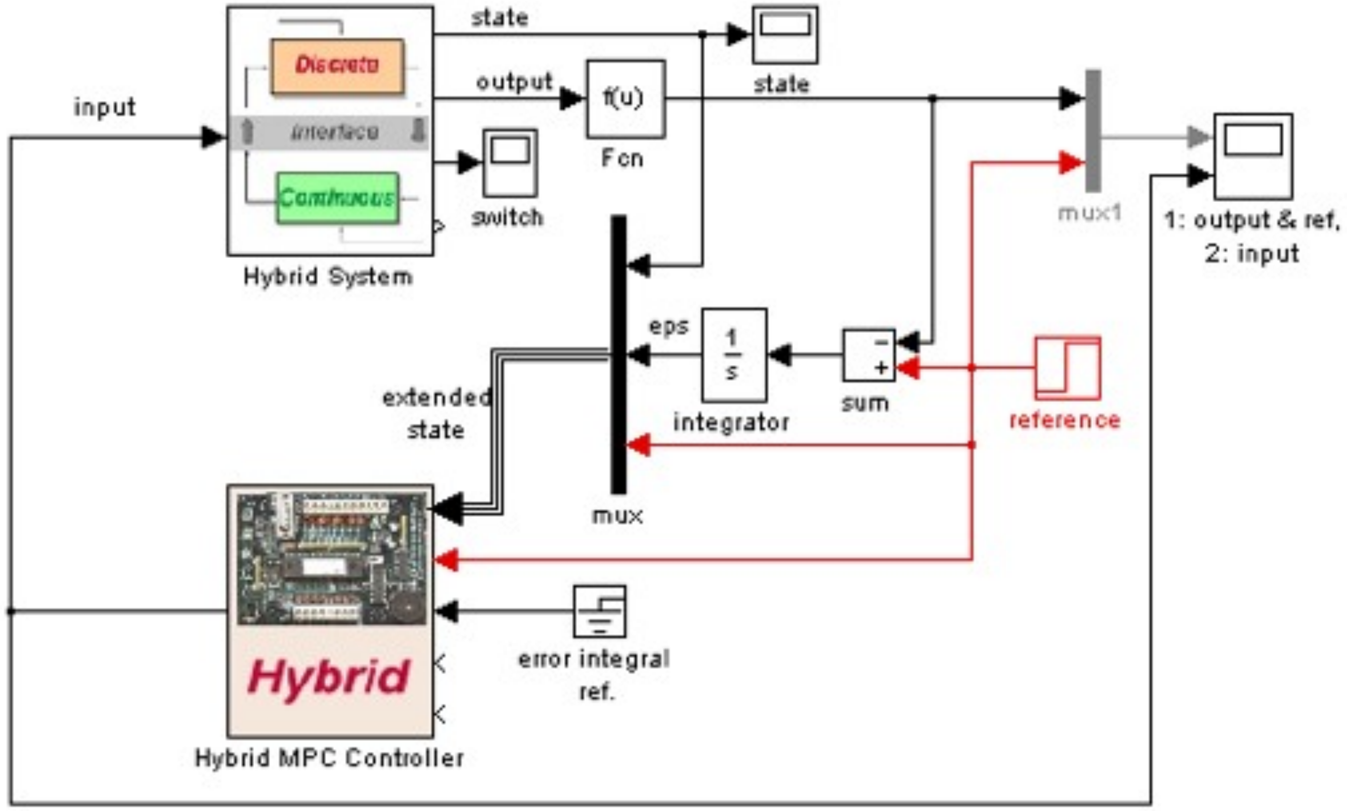
T_s = sampling time

- Treat $r(k)$ as a measured disturbance (=additional constant state)

- Add weight on $\varepsilon(k)$ in cost function to make $\varepsilon(k) \to 0$

- In HYSDEL:

```
INTERFACE{
      STATE{
         REAL x          [-100,100];
         ...
         REAL epsilon    [-1e3, 1e3];
         REAL r          [0,     100]; }
      OUTPUT  {
         REAL y; }
      ... }
IMPLEMENTATION{
      CONTINUOUS{
         epsilon=epsilon+Ts*(r-(c*x));
         r=r;
         ...   }
      OUTPUT{
         y=c*x; }    }
```



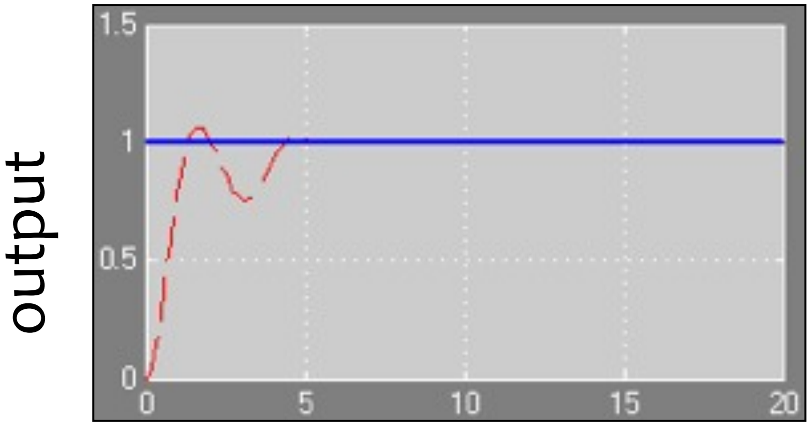
/demos/hybrid/hyb_integral_action.m

Note: same trick applies to linear MPC

Variable Constraints

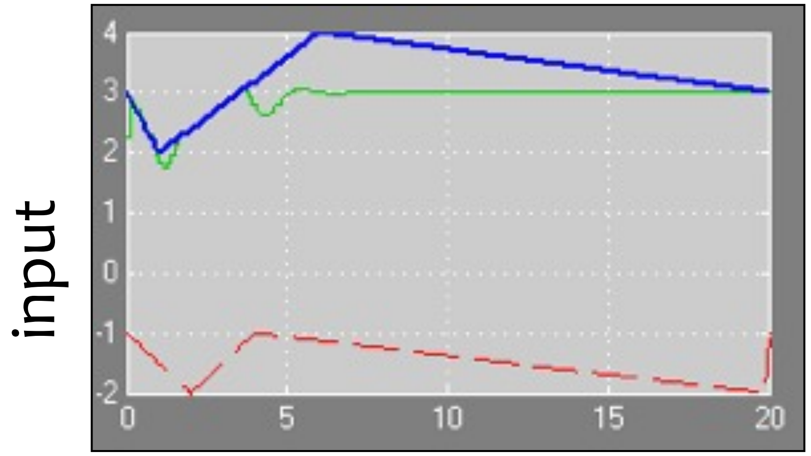
Problem: change upper (and/or lower) bounds on line

$$u(t) \leq u_{\max}(t)$$



1. Add a constant state and a new output in the prediction model:

$$\begin{cases} x_u(k+1) & = & x_u(k) \\ y_u(k) & = & x_u(k) - u(k) \end{cases}$$



2. Add output constraint

$$y_u(k) \geq 0, \;\; k = 0, 1, \ldots, N$$

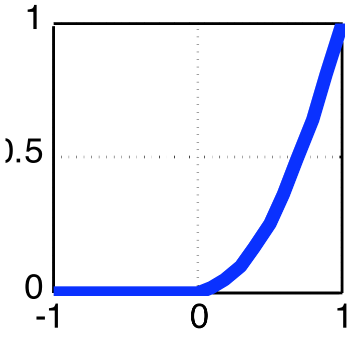
3. On-line implementation: feed the state back to the controller

$$x_u(t) = u_{\max}(t)$$

Note: same trick applies to linear MPC

/demos/linear/varbounds.m

Asymmetric Weights



- Say you want to weight a variable $u(k)$ only if $u(k) \geq 0$

- One way is to introduce a binary variable $[\delta=1] \leftrightarrow [u \geq 0]$, a continuous variable $z_u = u$ if $\delta=1$, $z_u=0$ otherwise, and weight z_u

- Better solution: avoid δ and set:

$$\begin{cases} \min & (\ldots) + \sum z_u^2(k) \\ \text{s.t.} & \epsilon(k) \geq u(k) \\ & \epsilon(k) \geq 0 \end{cases}$$

- In Hysdel:

```
INTERFACE{
      INPUT{
         REAL u      [-100,100];
         REAL zu     [-1, 1e3];
      ... }
IMPLEMENTATION{
      MUST{
         zu >= u;
         zu >= 0;
}    }
```

- When ∞-norms are used, one can do the same trick:

$$\begin{cases} \min & (\ldots) + |z_u(k)| \\ \text{s.t.} & z_u(k) \geq u(k) \\ & z_u(k) \geq 0 \end{cases}$$

(better way: if the MILP problem constructor can be accessed, avoid introducing z_u and just remove the constraint $\epsilon_u(k) \geq -[R]^i u(k)$ used to minimize $|Ru(k)|$, with constraint $\epsilon_u(k) \geq 0$)

Note: same trick applies to linear MPC

General Remarks About MIP Modeling

The complexity of solving a mixed-integer program largely depends on the number of integer (binary) variables involved in the problem.

Henceforth, when creating a hybrid model one has to

Be thrifty with integer variables !

Adding logical constraints usually helps ...

Generally speaking:

Modeling is an art

(a unifying general theory does not exist)