Model Predictive Control of Hybrid Systems

Alberto Bemporad

Dept. Information Engineering - University of Siena, Italy

http://www.dii.unisi.it/~bemporad

University of Siena
(founded in 1240)

European Embedded Control Institute

2nd HYCON PhD School on Hybrid Systems (Siena, 18/7/2007)

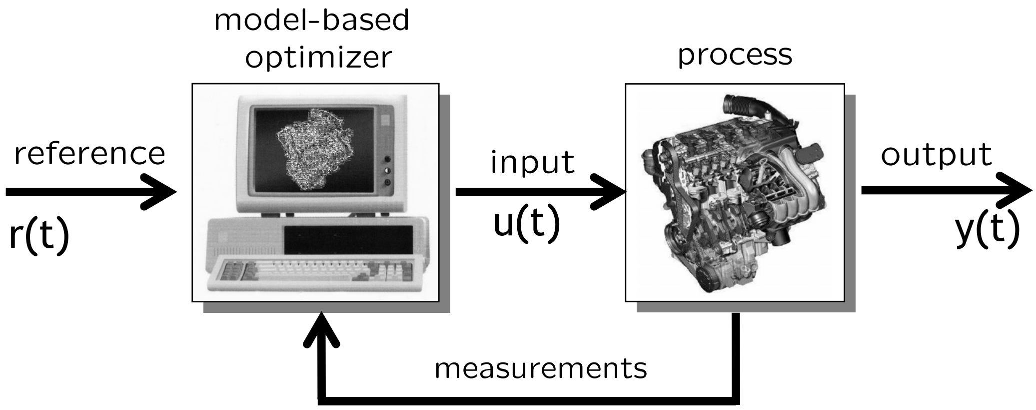
Outline

- Basics of Model Predictive Control (MPC)
- Hybrid models for MPC
- MPC of hybrid systems
- Explicit MPC (multiparametric programming)
- Optimization-based reachability analysis
- Examples

Model Predictive Control

- MPC concepts
- Linear MPC
- Matlab tools for linear MPC

Model Predictive Control



- MODEL: a model of the plant is needed to predict the future behavior of the plant
- PREDICTIVE: optimization is based on the predicted future evolution of the plant
- CONTROL: control complex constrained multivariable plants

Receding Horizon Philosophy

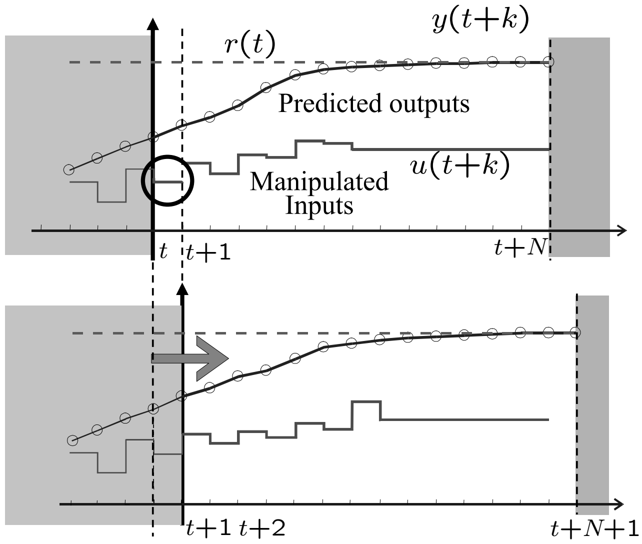
- At time t:

  Solve an optimal control problem over a finite future horizon N

  - minimize $f(|y-r|, |u|)$

  - subject to constraints

  $$u_{\text{min}} \leq \quad u \quad \leq u_{\text{max}}$$
  $$y_{\text{min}} \leq \quad y \quad \leq y_{\text{max}}$$



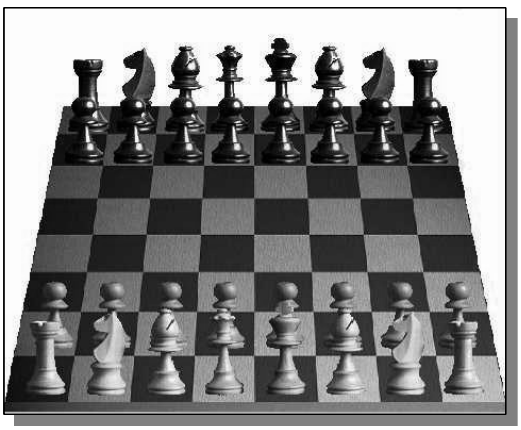
- Only apply the first optimal move $u^*(t)$

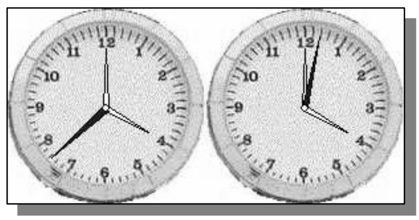
- Get new measurements, and repeat the optimization at time $t+1$

Advantage of on-line optimization: **FEEDBACK!**

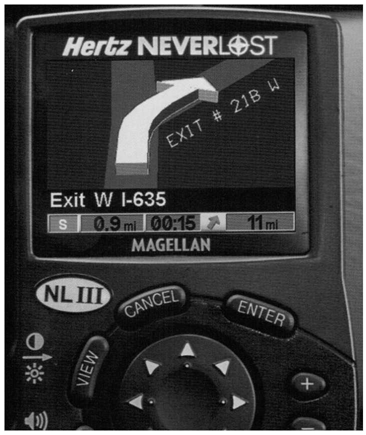
Receding Horizon - Example

- MPC is like playing chess !

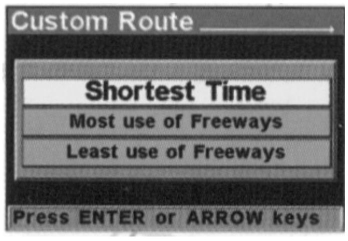




Receding Horizon – GPS Navigation



- **prediction model**
  how vehicle moves on the map
- **constraints**
  drive on roads, respect one-way roads, etc.
- **disturbances**
  road works, driver's inattention, etc.
- **set point**
  desired location
- **cost function:**
  Ex: minimum time
  Ex: penalty on highways
- **receding horizon mechanism**
  event-based (optimal route re-planned when path is lost)



It's a **feedback** strategy !

Constrained Optimal Control

- Linear Model: $\begin{cases} x(t+1) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \end{cases}$ $\quad x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $y \in \mathbb{R}^p$

- Constraints: $\begin{cases} u_{\min} \leq u(t) \leq u_{\max} \\ y_{\min} \leq y(t) \leq y_{\max} \end{cases}$

- Constrained optimal control problem (quadratic performance index):

$$\min_{u(0),\ldots,u(N-1)} \sum_{k=0}^{N-1} \left[x'(k)Qx(k) + u'(k)Ru(k)\right] + x'(N)Px(N)$$

$$\text{s.t.} \quad u_{\min} \leq u(k) \leq u_{\max},\ k = 0, \ldots, N-1$$
$$y_{\min} \leq y(k) \leq y_{\max},\ k = 1, \ldots, N$$

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

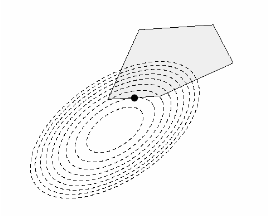
Constrained Optimal Control

- Optimization problem:

$$V(x(0)) = \tfrac{1}{2}x'(0)Yx(0) + \min_U \; \tfrac{1}{2}U'HU + x'(0)FU \quad \text{(quadratic)}$$

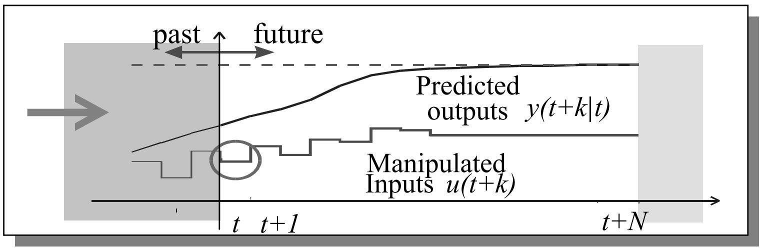
$$\text{s.t.} \quad GU \leq W + Sx(0) \quad \text{(linear)}$$

Convex QUADRATIC PROGRAM (QP)



- $U \triangleq [u'(0) \;\ldots\; u'(N-1)]' \in \mathbb{R}^s$, $s \triangleq Nm$, is the optimization vector
- $H = H' \succ 0$, and H, F, Y, G, W, S depend on weights Q, R, P, upper and lower bounds $u_{\min}$, $u_{\max}$, $y_{\min}$, $y_{\max}$, and model matrices A, B, C

MPC of Linear Systems



At time t:

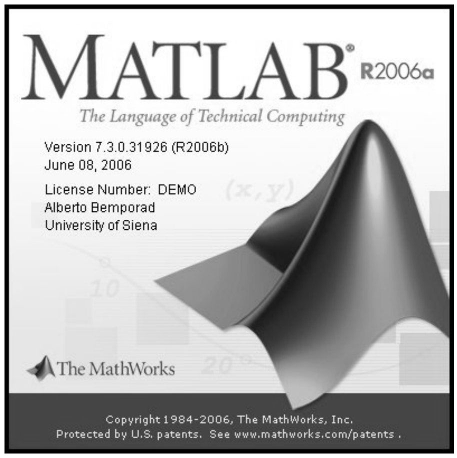
- Get/estimate the current state $x(t)$
- Solve the QP problem

$$\min_U \; \tfrac{1}{2}U'HU + x'(t)FU$$
$$\text{s.t.} \quad GU \leq W + Sx(t)$$

  and let U={$u^*(0)$,...,$u^*(N$-1)} be the solution (=finite-horizon constrained open-loop optimal control)
- Apply only $u(t) = u^*(0)$ and discard the remaining optimal inputs
- Go to time t+1

Model Predictive Control Toolbox

- **MPC Toolbox 2.0** (Bemporad, Ricker, Morari, 1998-2007):
  - Object-oriented implementation (MPC object)
  - MPC Simulink Library
  - MPC Graphical User Interface
  - RTW extension (code generation)
  - Linked to OPC Toolbox v2.0.1



Only linear models are handled

http://www.mathworks.com/products/mpc/

Example: AFTI-16

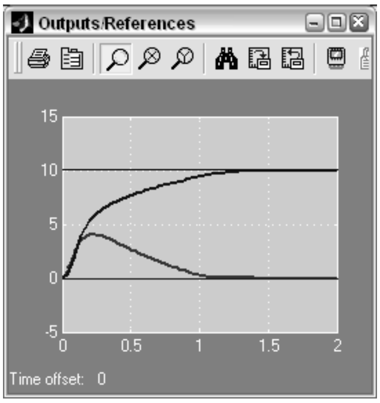
- Linearized model:

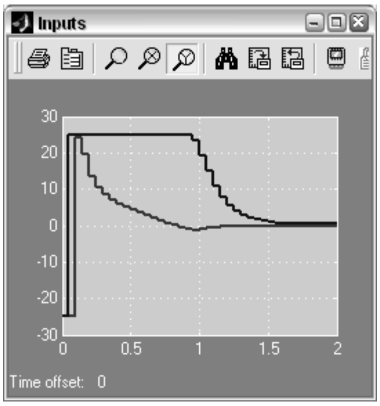
$$\begin{cases} \dot{x} = \begin{bmatrix} -.0151 & -60.5651 & 0 & -32.174 \\ -.0001 & -1.3411 & .9929 & 0 \\ .00018 & 43.2541 & -.86939 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x + \begin{bmatrix} -2.516 & -13.136 \\ -.1689 & -.2514 \\ -17.251 & -1.5766 \\ 0 & 0 \end{bmatrix} u \\ y = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x, \end{cases}$$

- Inputs: elevator and flaperon angle
- Outputs: attack and pitch angle
- Sampling time: $T_s = .05$ s (+ zero-order hold)
- Constraints: max 25° on both angles
- Open-loop response: unstable
  (open-loop poles: $-7.6636, -0.0075 \pm 0.0556j, 5.4530$)

go to demo **afti16.m** (MPC-Tbx)

Example: AFTI-16

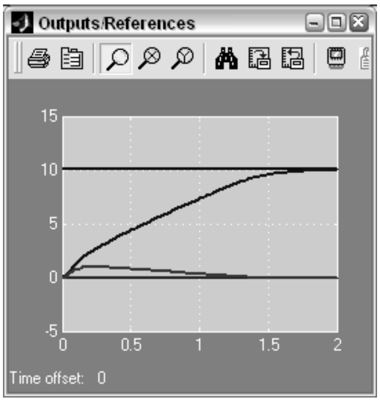


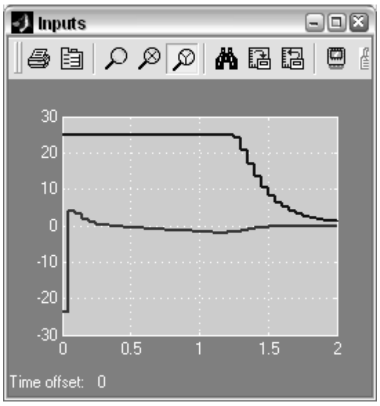


$N_y = 10$, $N_u = 3$,

$w_y = \{10, 10\}$, $w_{\delta u} = \{.01, .01\}$,

$u_{\text{min}} = -25^{\circ}$, $u_{\text{max}} = 25^{\circ}$





$N_y = 10$, $N_u = 3$,

$w_y = \{100, 10\}$, $w_{\delta u} = \{.01, .01\}$,

$u_{\text{min}} = -25^{\circ}$, $u_{\text{max}} = 25^{\circ}$

Convergence Result

Theorem 1 *Consider the linear system*

$$\begin{cases} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{cases}$$

and the MPC control law based on

$$\begin{array}{ll} \min_U J(U, x(t)) = & \sum_{k=0}^{N-1} \left\{ y'(t+k|t) Q y(t+k|t) + u'(t+k) R u(t+k) \right\} \\ \text{subj. to} & y_{min} \leq y(t+k) \leq y_{max} \\ & u_{min} \leq u(t+k) \leq u_{max} \end{array}$$

Assume that the optimization problem is feasible at time $t = 0$. Then, for either $N \to \infty$ or with the extra constraint $x(t+N|t) = 0$, for all $R > 0$, $Q \geq 0$,

$$\lim_{t\to\infty} y(t) = 0, \quad \lim_{t\to\infty} u(t) = 0,$$

while fulfilling the constraints. Moreover, provided that $(Q^{\frac{1}{2}}C, A)$ is a detectable pair, $\lim_{t\to\infty} x(t) = 0$.

(Keerthi and Gilbert, 1988)(Bemporad *et al.*, 1994)

Proof: Use value function as Lyapunov function

Convergence Proof

- Assume we set the terminal constraint $x(t+N|t) = 0$
- Let $\mathcal{U}_t^*$ denote the optimal control sequence @t $\{u_t^*(0), \ldots, u_t^*(N-1)\}$
- Let $V(t) \triangleq J(\mathcal{U}_t^*, x(t))$=value function ➡ Lyapunov function
- By construction, $\mathcal{U}_1 = \{u_t^*(1), \ldots, u_t^*(N-1), 0\}$ is feasible @$t+1$, and hence

$$V(t+1) = J(\mathcal{U}^*{}_{t+1}, x(t+1)) \leq J(\mathcal{U}_1, x(t+1)) = \\ = V(t) - y'(t)Qy(t) - u'(t)Ru(t)$$

- $V(t)$ is decreasing and lower-bounded by 0 $\Rightarrow \exists V_\infty = \lim_{t\to\infty} V(t) \Rightarrow V(t+1) - V(t) \to 0$, which implies $y'(t)Qy(t), u'(t)Ru(t) \to 0$
- Since $R > 0$, $u(t) \to 0$
- Assume for simplicity that $C = I$ (i.e., $y(t) = x(t)$) and $Q > 0$. Then also $x(t) \to 0$.

Global optimum is not needed to prove convergence !

Convergence Proof

If $Q \geq 0$ and/or $C \neq I$:

- For all $\forall k = 0, \ldots, n-1$, we have

$$\lim_{t\to\infty} y'(t+k)Qy(t+k) = \lim_{t\to\infty} \|Q^{\frac{1}{2}}C(A^k x(t) + \sum_{j=0}^{k-1} A^j Bu(t+k-1-j))\|^2 = 0$$

- As $u(t) \to 0$, also $Q^{\frac{1}{2}}CA^k x(t) \to 0$, and hence $\Theta x(t) \to 0$, where Θ is the observability matrix of $(Q^{\frac{1}{2}}C, A)$.
- If $(Q^{\frac{1}{2}}C, A)$ is observable, this also implies $x(t) \to 0$.
- If $(Q^{\frac{1}{2}}C, A)$ is only detectable, through a canonical decomposition one can observe that, as $u(t) \to 0$, unobservable modes go to zero spontaneously.

Convergence Proof

- Similar argument for infinite prediction horizon $N = \infty$:
  - Let $\mathcal{U}_t^*$ denote the infinite optimal control sequence @t $\{u_t^*(0), u_t^*(1) \ldots\}$
  - Let $V(t) \triangleq J(\mathcal{U}_t^*, x(t))$=value function ($\Rightarrow$ Lyapunov function)
  - Because constraints were checked up to $t+k = \infty$, $\mathcal{U}_1 = \{u_t^*(1), u_t^*(2), \ldots\}$ is feasible @$t+1$ by construction.
  - Hence

$$V(t+1) = J({\mathcal{U}^*}_{t+1}, x(t+1)) \leq J(\mathcal{U}_1, x(t+1)) = \\ = V(t) - y'(t)Qy(t) - u'(t)Ru(t)$$

  - Repeat same arguments as before

MPC and LQR

Jacopo Francesco Riccati (1676 - 1754)

- Consider the MPC control law:

$$\begin{array}{ll} \min_U J(U,t) = & x'(t+T|t)Px(t+T|t) + \\ & \sum_{k=0}^{N-1} \left\{ x'(t+k|t)Qx(t+k|t) + u'(t+k)Ru(t+k) \right\} \\ \text{subj. to} & y_{min} \leq y(t+k|t) \leq y_{max},\ k = 1, \ldots, N \\ & u_{min} \leq u(t+k) \leq u_{max},\ k = 0, \ldots, N-1 \\ & u(t+k) = Kx(t+k|t),\ k = N_u, \ldots, N-1 \end{array}$$

$R = R' > 0$, $Q = Q' \geq 0$, and P, K satisfy the Riccati equation

$$\begin{array}{ll} K & = -(R + B'PB)^{-1}B'PA \\ P & = (A+BK)'P(A+BK) + K'RK + Q \end{array}$$

- In a polyhedral region around the origin the MPC control law is equivalent to the constrained LQR controller with weights Q,R. (Chmielewski, Manousiouthakis, 1996)

(Scokaert and Rawlings, 1998)

MPC = constrained LQR

- The larger the horizon, the larger the region where MPC=LQR

Outline

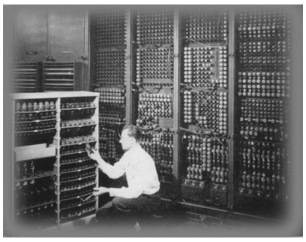
✓ Basics of Model Predictive Control (MPC)

- Hybrid models for MPC
- MPC of hybrid systems
- Explicit MPC (multiparametric programming)
- Optimization-based reachability analysis
- Examples

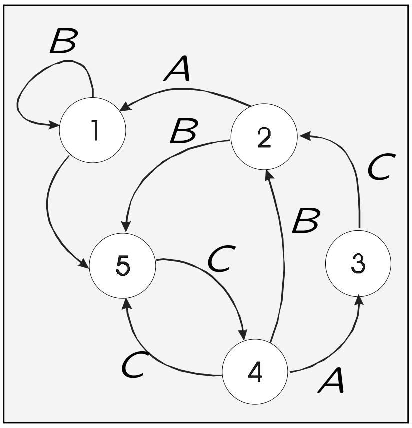
Hybrid Models for MPC

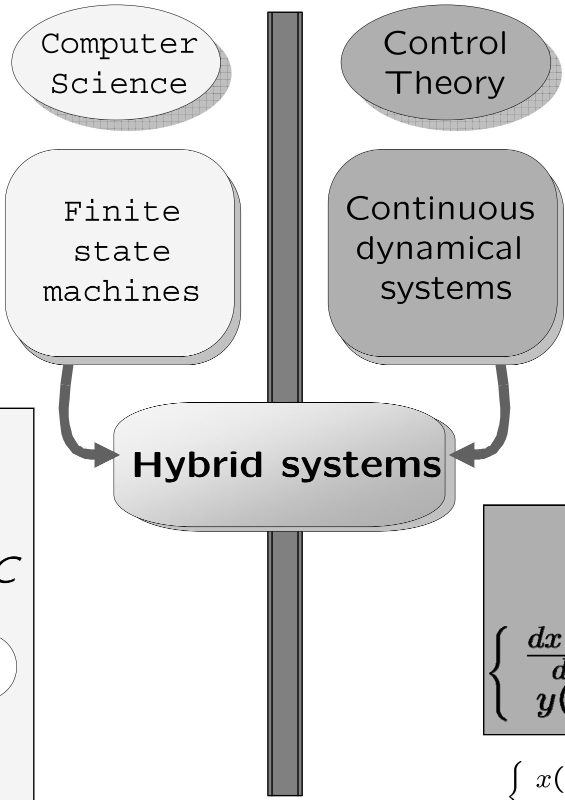
- Discrete Hybrid Automata (DHA)
- Mixed Logical Dynamical (MLD) Systems
- Piecewise Affine (PWA) Systems

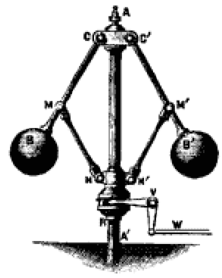
Hybrid Systems



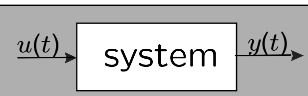
$\mathbf{x} \in \{1,2,3,4,5\}$
$\mathbf{u} \in \{A,B,C\}$







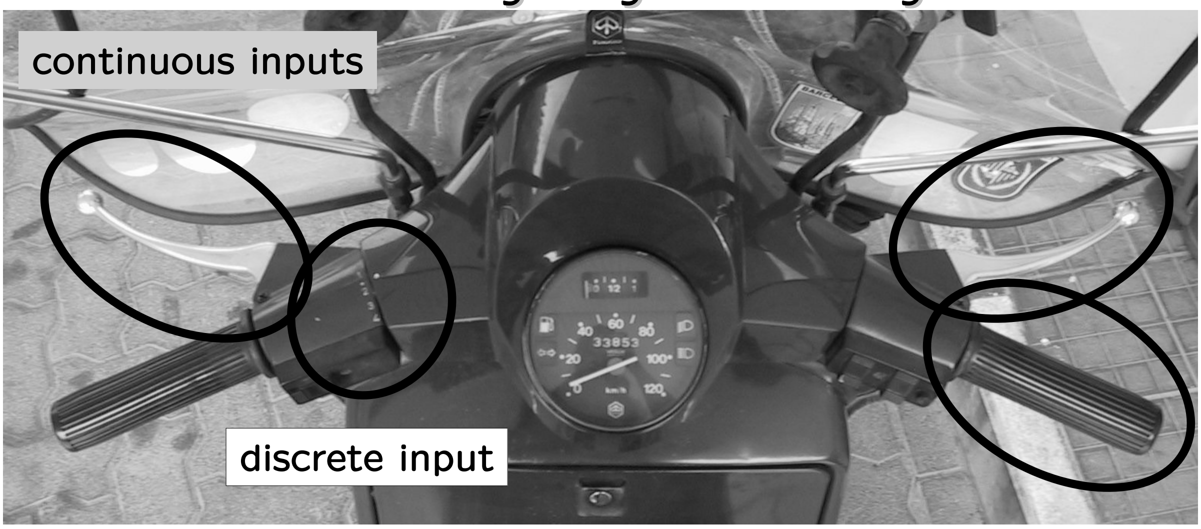
$x \in \mathbb{R}^n$
$u \in \mathbb{R}^m$
$y \in \mathbb{R}^p$



$$\begin{cases} \frac{dx(t)}{dt} &= f(x(t),u(t)) \\ y(t) &= g(x(t),u(t)) \end{cases}$$

$$\begin{cases} x(k+1) &= f(x(k),u(k)) \\ y(k) &= g(x(k),u(k)) \end{cases}$$

"Intrinsically Hybrid" Systems



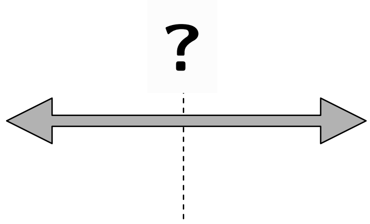
Discrete input (gear 1,2,3,4,N) + Continuous inputs (brakes, gas, clutch) + Continuous dynamical states (velocities, torques, air-flows, fuel level)

Key Requirements for Hybrid Models

- **Descriptive** enough to capture the behavior of the system
  - continuous dynamics (physical laws)
  - logic components (switches, automata, software code)
  - interconnection between logic and dynamics
- **Simple** enough for solving *analysis* and *synthesis* problems

$$\begin{cases} x' &= Ax + Bu \\ y &= Cx + Du \end{cases}$$

linear systems

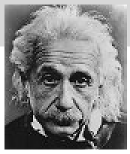


$$\begin{cases} x' &= f(x, u, t) \\ y &= g(x, u, t) \end{cases}$$

nonlinear systems

linear hybrid systems

"Make everything as simple as possible, but not simpler."
— Albert Einstein

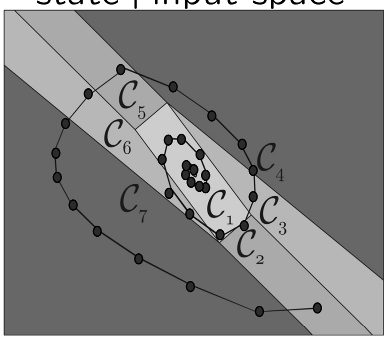


Piecewise Affine Systems

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

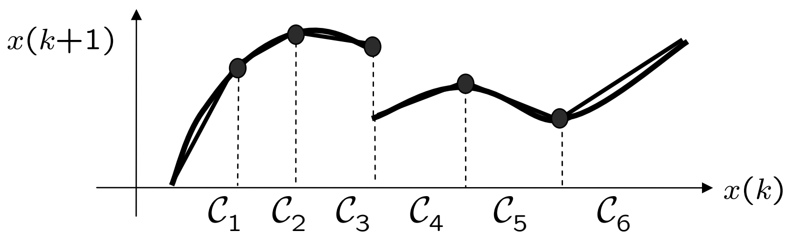
$x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $y \in \mathbb{R}^p$
$i(k) \in \{1, \ldots, s\}$

state+input space



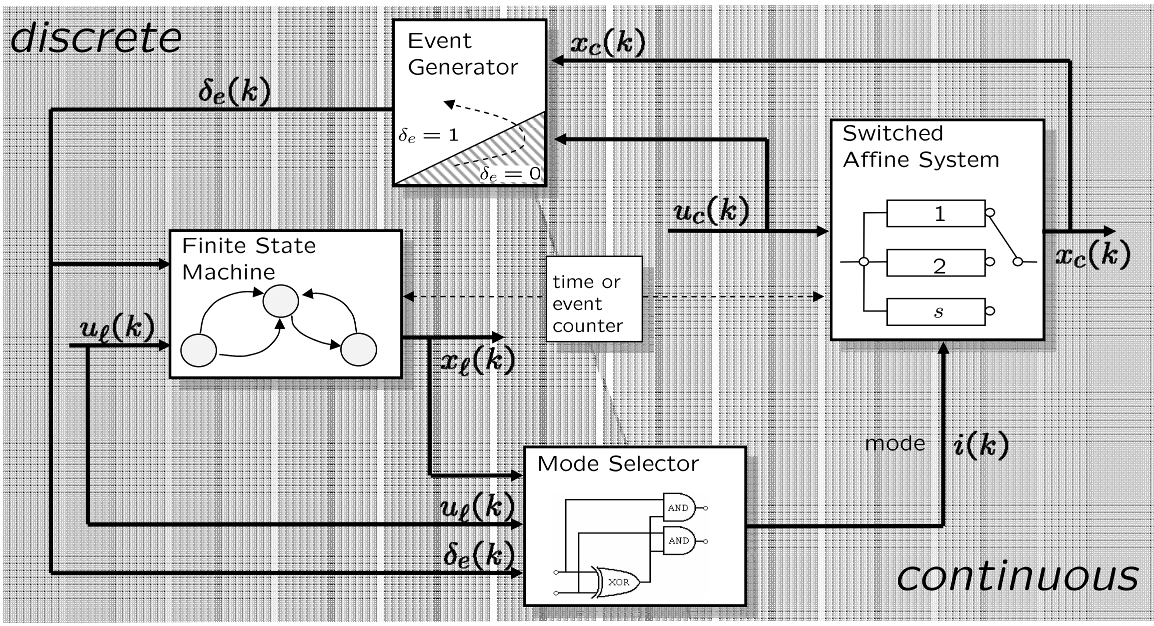
(Sontag 1981)

Can approximate nonlinear and/or discontinuous dynamics arbitrarily well



Discrete Hybrid Automaton

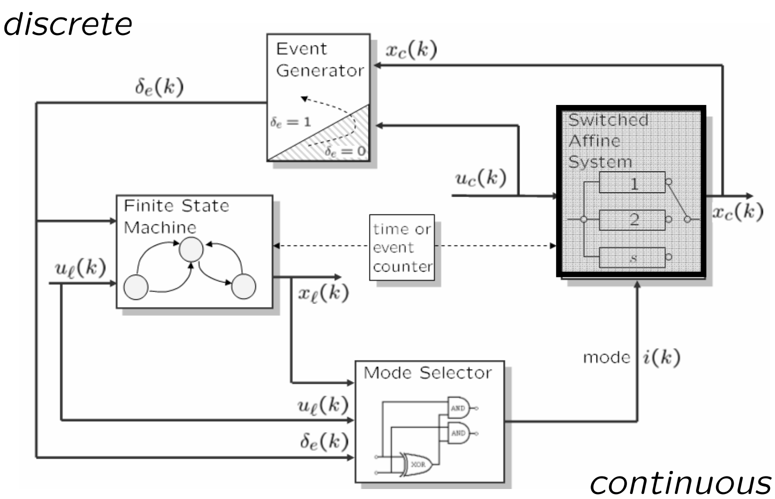
(Torrisi, Bemporad, 2004)



$x_\ell \in \{0,1\}^{n_b}$ = binary states
$u_\ell \in \{0,1\}^{m_b}$ = binary inputs
$\delta_e \in \{0,1\}^{n_e}$ = event variables

$x_c \in \mathbb{R}^{n_c}$ = continuous states
$u_c \in \mathbb{R}^{m_c}$ = continuous inputs
$i \in \{1,2,\ldots,s\}$ = current mode

Switched Affine System

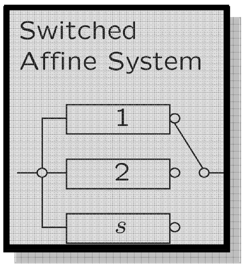


The affine dynamics depend on the current mode $i(k)$:

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

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

Switched Affine System



The state-update equation can be rewritten as a difference equation + *if-then-else* conditions:

$$z_1(t) = \begin{cases} A_1 x_c(t) + B_1 u_c(t) + f_1, & \text{if } (i(t) = 1) \\ 0, & \text{otherwise,} \end{cases}$$

$$\vdots$$

$$z_s(t) = \begin{cases} A_s x_c(t) + B_s u_c(t) + f_s, & \text{if } (i(t) = s), \\ 0, & \text{otherwise,} \end{cases}$$

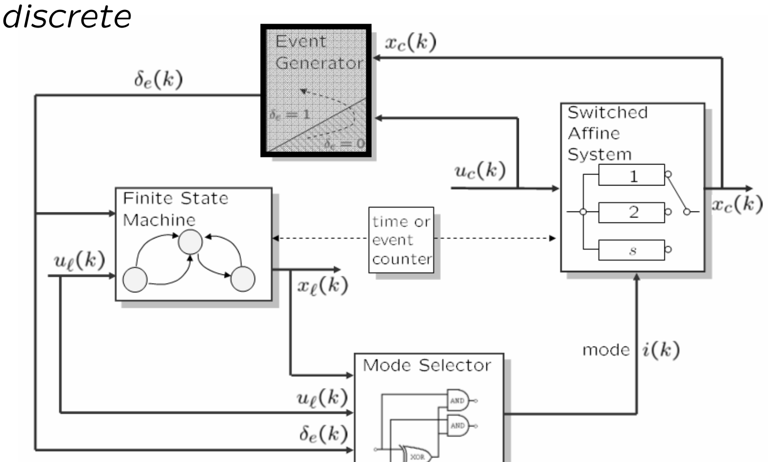
$$x_c(t+1) = \sum_{i=1}^{s} z_i(t)$$

where $z_i(t) \in \mathbb{R}^{n_c}, i = 1, \ldots, s$.

Output equations $y_c(t) = C_i x_c(t) + D_i u_c(t) + g_i$ admit a similar transformation.

Event Generator

discrete



continuous

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

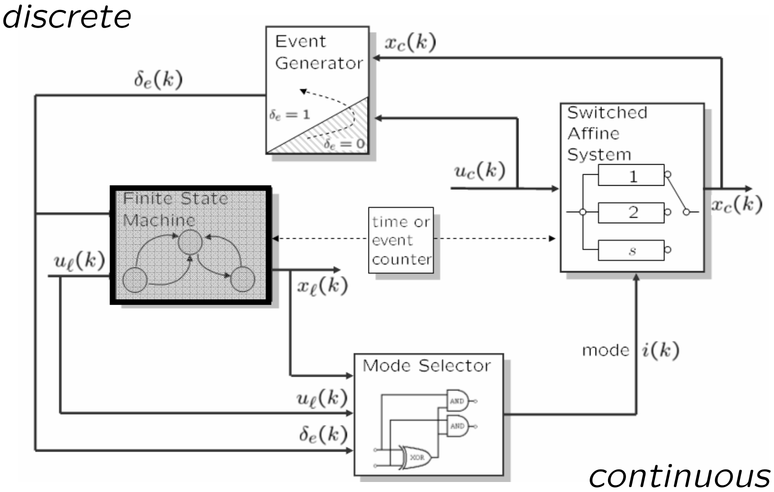
$$[\delta_e^i(k) = 1] \leftrightarrow [H^i x_c(k) + K^i u_c(k) \leq W^i]$$

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

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

Finite State Machine

discrete



continuous

The binary state of the finite state machine evolves according to a Boolean state update function:

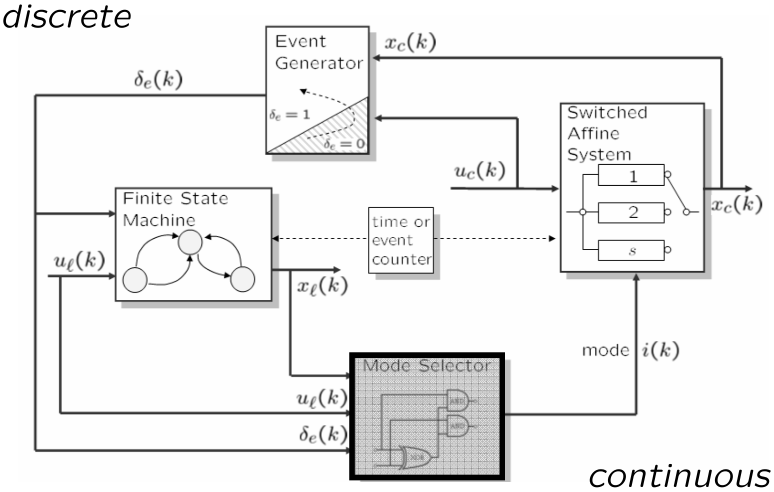
$$x_\ell(k+1) = f_B(x_\ell(k), u_\ell(k), \delta_e(k))$$

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

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

Mode Selector

discrete



continuous

The mode selector can be seen as the output function of the discrete dynamics

The active mode $i(k)$ is selected by a Boolean function of the current binary states, binary inputs, and event variables:

$$i(k) = f_M(x_\ell(k), u_\ell(k), \delta_e(k))$$

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

Example:

$$i(k) = \begin{bmatrix} \neg u_\ell(k) \vee x_\ell(k) \\ u_\ell(k) \wedge x_\ell(k) \end{bmatrix}$$

u_ℓ / x_ℓ	0	1
0	$i = \left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]$	$i = \left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]$
1	$i = \left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$	$i = \left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]$

the system has 3 modes

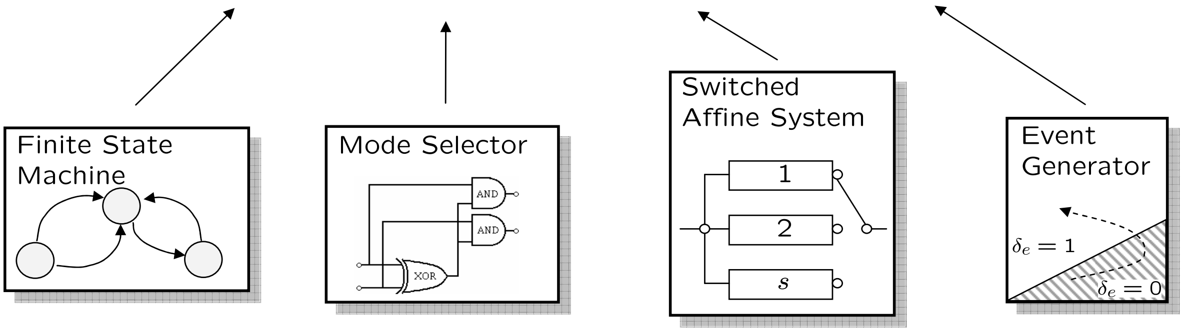
Logic and Inequalities

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

(Glover 1975, Williams 1977, Hooker 2000)

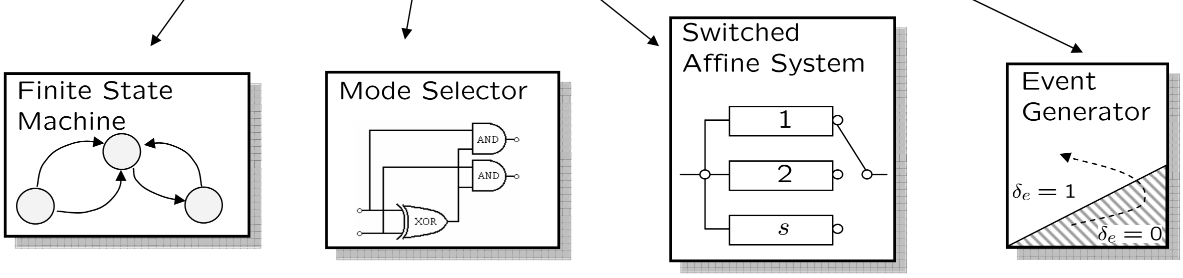
Any logic formula involving **Boolean variables** and **linear combinations of continuous variables** can be translated into a set of **(mixed-)integer linear (in)equalities**

All the DHA blocks can be translated into a set of (mixed-)integer linear equalities and inequalities



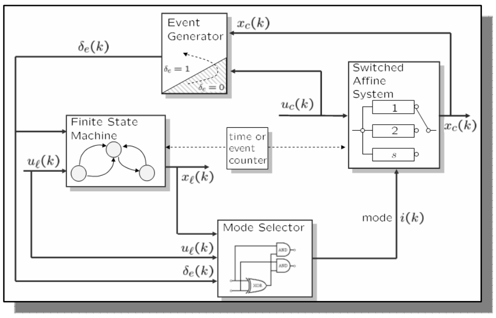
Logic and Inequalities

Any logic statement $f(X) =$TRUE $\bigwedge_{j=1}^{m} \left(\vee_{i \in P_j} X_i \vee_{i \in N_j} \neg X_i \right)$ (CNF) $N_j, P_j \subseteq \{1, \dots, n\}$	$\begin{cases} 1 \leq \sum_{i \in P_1} \delta_i + \sum_{i \in N_1} (1 - \delta_i) \\ \vdots \\ 1 \leq \sum_{i \in P_m} \delta_i + \sum_{i \in N_m} (1 - \delta_i) \end{cases}$
$[\delta_e^i(k) = 1] \leftrightarrow [H^i x_c(k) \leq W^i]$	$\begin{cases} H^i x_c(k) - W^i \leq M^i(1 - \delta_e^i) \\ H^i x_c(k) - W^i > m^i \delta_e^i \end{cases}$
IF $[\delta = 1]$ THEN $z = a_1^T x + b_1^T u + f_1$ ELSE $z = a_2^T x + b_2^T u + f_2$	$\begin{cases} (m_2 - M_1)\delta + z \leq a_2 x + b_2 u + f_2 \\ (m_1 - M_2)\delta - z \leq -a_2 x - b_2 u - f_2 \\ (m_1 - M_2)(1 - \delta) + z \leq a_1 x + b_1 u + f_1 \\ (m_2 - M_1)(1 - \delta) - z \leq -a_1 x - b_1 u - f_1 \end{cases}$



Mixed Logical Dynamical Systems

Discrete Hybrid Automaton



HYSDEL

(Torrisi, Bemporad, 2004)

Mixed Logical Dynamical (MLD) Systems (Bemporad, Morari 1999)

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

Continuous and binary variables $x \in \mathbb{R}^{n_r} \times \{0,1\}^{n_b}$, $u \in \mathbb{R}^{m_r} \times \{0,1\}^{m_b}$, $y \in \mathbb{R}^{p_r} \times \{0,1\}^{p_b}$, $\delta \in \{0,1\}^{r_b}$, $z \in \mathbb{R}^{r_r}$

- Computationally oriented (mixed-integer programming)
- Suitable for controller synthesis, verification, ...

Mixed-Integer Models in OR

Translation of logical relations into linear inequalities is heavily used in operations research (OR) for solving complex decision problems by using mixed-integer programming (MIP)

Example: Optimal investments for quality of supply improvement in electrical energy distribution networks (Bemporad, Muñoz, Piazzesi, 2006)

Example: Timetable generation (for demanding professors ...)

	9	11	14	15	16
lun	Sistemi Operativi (*18)		Misure per la Automazione (*7)		Ingegneria del Software (*18)
			Basi di Dati (*18)		
mar	Basi di Dati (*3)	Sistemi Operativi (*3)	Robotica ed Automazione di Processo (*18)		
mer	Robotica ed Automazione di Processo (*8)	Misure per la Automazione (*7)		Laboratorio di Robotica e Realtà Virtuale (*15)	
	Ingegneria del Software (*18)				
gio	Basi di Dati (*3)		Sistemi Operativi (*5)		
	Laboratorio di Robotica e Realtà Virtuale (*15)				
ven	Robotica ed Automazione di Processo (*8)		Misure per la Automazione (*7)		
	Ingegneria del Software (*18)				
sab					

CPU time: 0.2 s

A Simple Example

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

$$-10 \leq x(k) \leq 10$$

- Associate $[\delta(k) = 1] \leftrightarrow [x(k) \geq 0]$ and transform

$\longrightarrow$
$$\begin{aligned} x(k) &\geq m(1 - \delta(k)) \\ x(k) &\leq -\epsilon + (M + \epsilon)\delta(k) \end{aligned}$$
$M = -m = 10$
$\epsilon > 0$ "small"

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

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

$z(k) = \delta(k)x(k) \longrightarrow$
$$\begin{aligned} z(k) &\leq M\delta(k) \\ z(k) &\geq m\delta(k) \\ z(k) &\leq x(k) - m(1 - \delta(k)) \\ z(k) &\geq x(k) - M(1 - \delta(k)) \end{aligned}$$

- Rewrite as a linear equation

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

(Note: nonlinear system $x(k+1) = 0.8|x(k)| + u(k)$)

Hybrid Toolbox for Matlab

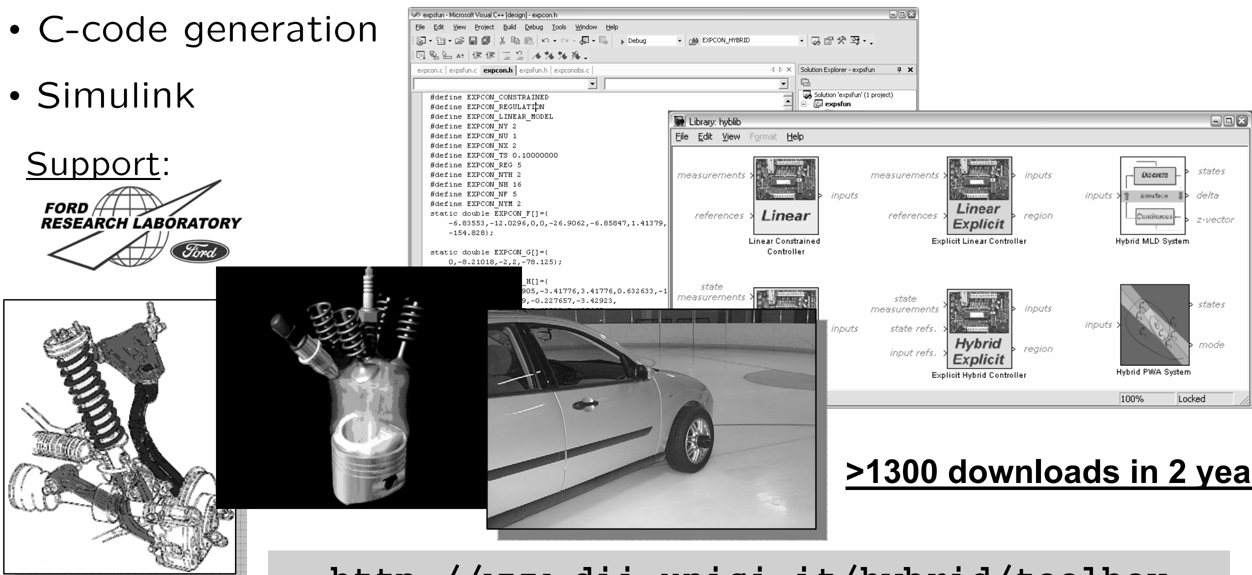
(Bemporad, 2003-today)

Features:

- Hybrid model (MLD and PWA) design, simulation, verification
- Control design for linear systems w/ constraints and hybrid systems (on-line optimization via QP/MILP/MIQP)
- Explicit control (via multiparametric programming)
- C-code generation
- Simulink

Support:

FORD RESEARCH LABORATORY



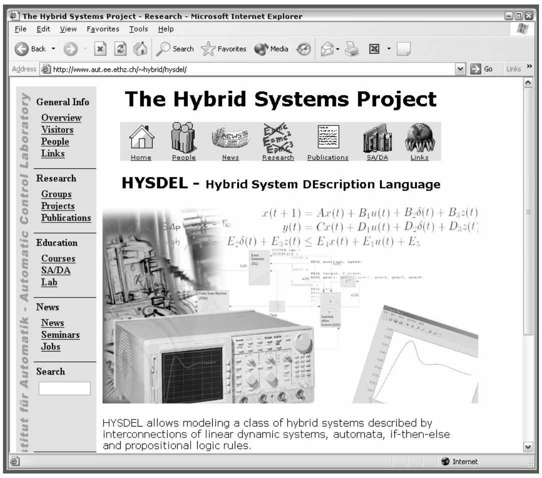
>1300 downloads in 2 years

http://www.dii.unisi.it/hybrid/toolbox

HYSDEL
(HYbrid Systems DEscription Language)

- Describe *hybrid systems:*

  – Automata
  – Logic
  – Lin. Dynamics
  – Interfaces
  – Constraints



(Torrisi, Bemporad, 2004)

- Automatically generate MLD models in Matlab

Download: **http://www.dii.unisi.it/hybrid/toolbox**

Reference: **http://control.ethz.ch/~hybrid/hysdel**

The Multi-Parametric Toolbox (MPT)

(Kvasnica, Grieder, Baotic)

Model ⇨ Control ⇨ Analysis ⇨ Deployment

- MPT is a repository of hybrid systems design tools from international experts utilizing state-of-the-art optimization packages

- Free, open-source, GNU licensed toolbox

- Special emphasis:
  – Low controller complexity
  – Numerical robustness (e.g. handling of degenerate problems)
  – Generation of real-time executable code

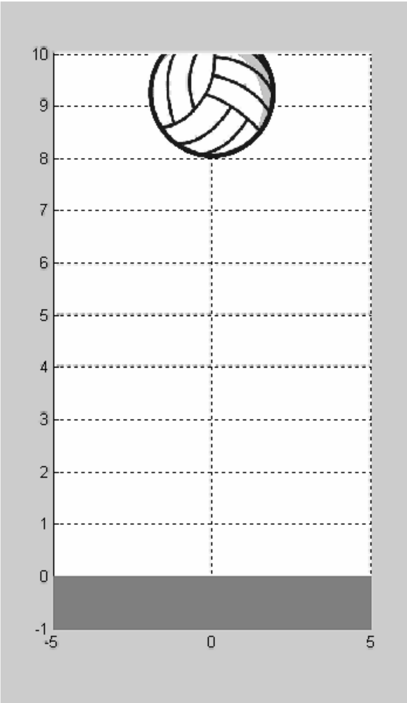
MPT, HYSDEL
Hybrid ID Tbx
YALMIP, CDD
Ellipsoidal Tbx
Proj. algorithms
SeDuMi

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

ABB Honeywell ANALOGIC TECH NASA arcelor Steel solutions for a better world

http://control.ee.ethz.ch/~mpt/

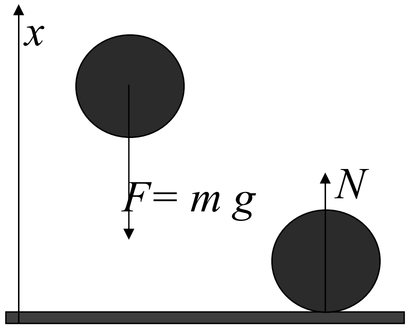
Example: Bouncing Ball



$$\ddot{x} = -g$$

$$x \le 0 \Rightarrow \dot{x}(t^+) = -(1-\alpha)\dot{x}(t^-)$$

$\alpha \in [0, 1]$

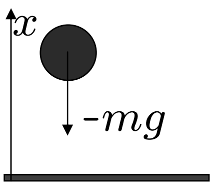


How to model the bouncing ball as a discrete-time hybrid system ?

Bouncing Ball – Time Discretization

$x(k) > 0$:

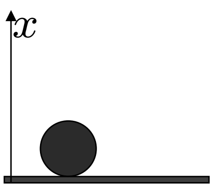
$$v(k) \approx \frac{x(k) - x(k-1)}{T_s} \qquad -g = \ddot{x}(k) \approx \frac{v(k) - v(k-1)}{T_s}$$



$$\begin{cases} v(k+1) &= v(k) - T_s g \\ x(k+1) &= x(k) + T_s v(k+1) \\ &= x(k) + T_s v(k) - T_s^2 g \end{cases}$$

$x(k) \le 0$:

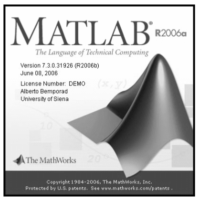
$$v(k) = -(1-\alpha)v(k-1) \qquad \begin{aligned} x(k+1) &= x(k-1) \\ &= x(k) - T_s v(k) \end{aligned}$$



$$\begin{cases} v(k+1) &= -(1-\alpha)v(k) \\ x(k+1) &= x(k) - T_s v(k) \end{cases}$$

HYSDEL - Bouncing Ball

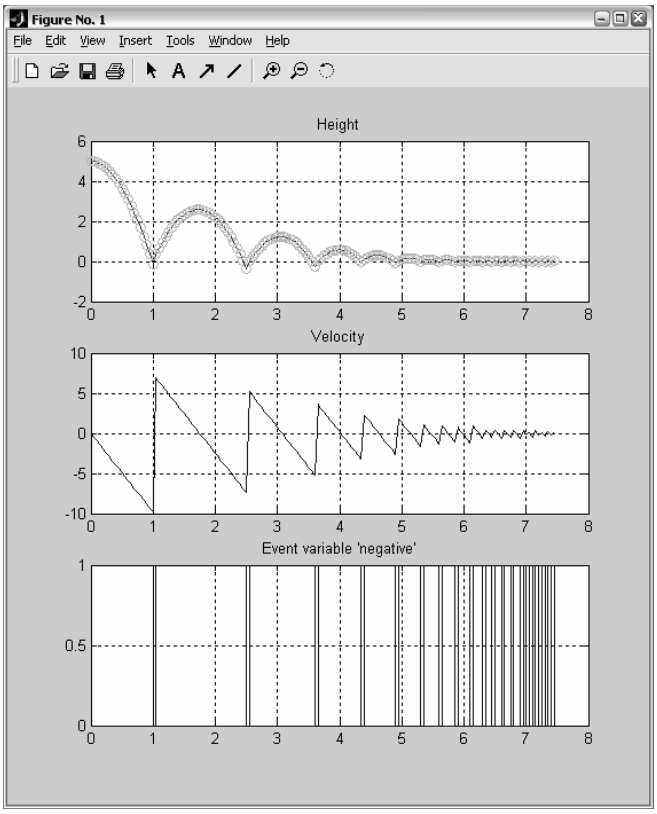
```
SYSTEM bouncing_ball {
INTERFACE {
/* Description of variables and constants */
        STATE { REAL height [-10,10];
                REAL velocity [-100,100];    }

        PARAMETER {
                REAL g;
                REAL alpha;  /* 0=elastic, 1=completely anelastic */
                REAL Ts; }
}
IMPLEMENTATION {
        AUX {   REAL z1;
                REAL z2;
                BOOL negative;    }

        AD {    negative = height <= 0; }

        DA {    z1 = {  IF negative THEN height-Ts*velocity
                        ELSE height+Ts*velocity-Ts*Ts*g};
                z2 = {  IF negative THEN -(1-alpha)*velocity
                        ELSE velocity-Ts*g};  }
        CONTINUOUS {
                height = z1;
                velocity=z2;}
}}
```



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

Bouncing Ball

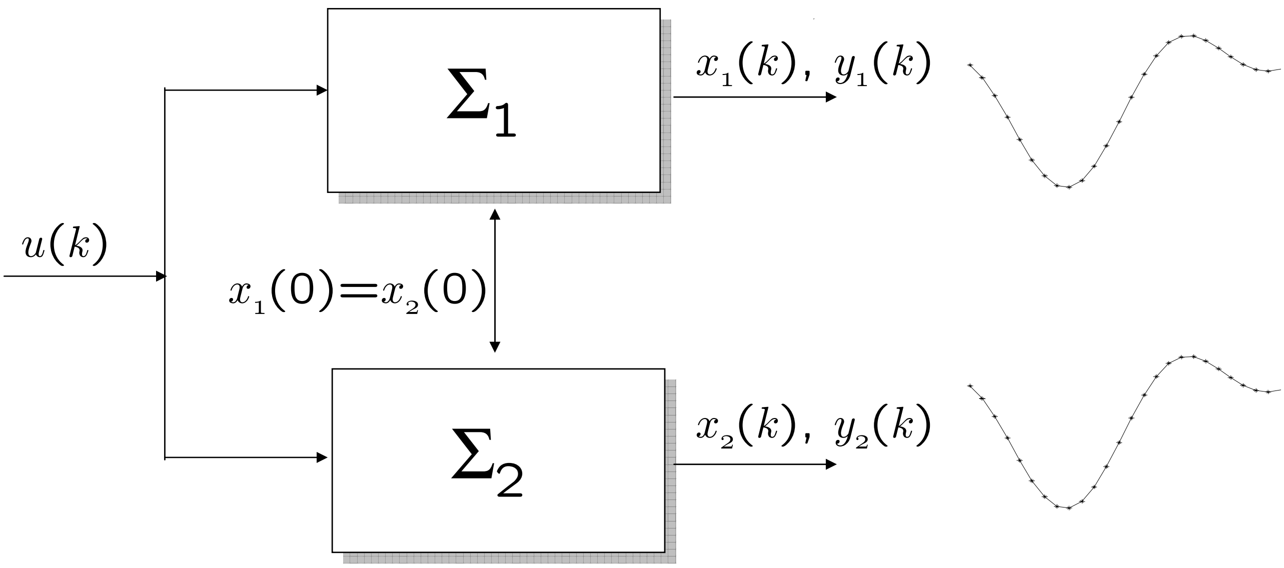
```
>>Ts=0.05;
>>g=9.8;
>>alpha=0.3;

>>S=mld('bouncing_ball',Ts);

>>N=150;
>>U=zeros(N,0);
>>x0=[5 0]';

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



Note: no Zeno effects in discrete time !

Equivalences of Hybrid Models

Definition 1 *Two hybrid systems* Σ_1, Σ_2 *are* equivalent *if for all initial conditions* $x_1(0) = x_2(0)$ *and input* $\{u_1(k)\}_{k\in\mathbb{Z}_+} = \{u_2(k)\}_{k\in\mathbb{Z}_+}$ *then* $x_1(k) = x_2(k)$ *and* $y_1(k) = y_2(k)$, *for all* $k \in \mathbb{Z}_+$.



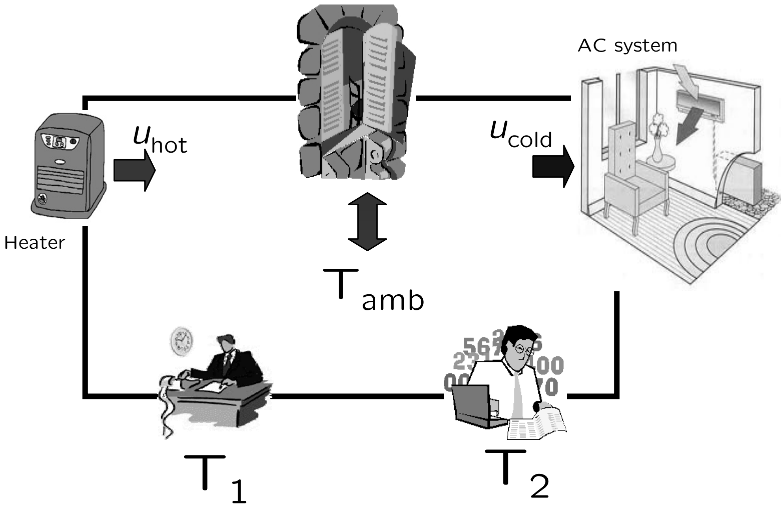
MLD and PWA Systems

Theorem MLD systems and PWA systems are equivalent

(Bemporad, Ferrari-Trecate, Morari, *IEEE TAC,2000*)

- Proof is constructive: given an MLD system it returns its equivalent PWA form
- Drawback: it needs the enumeration of all possible combinations of binary states, binary inputs, and δ variables
- Most of such combinations lead to empty regions
- Efficient algorithms are available for converting MLD models into PWA models avoiding such an enumeration:
  - A. Bemporad, "*Efficient Algorithms for Converting Mixed Logical Dynamical Systems into an Equivalent Piecewise Affine Form*", IEEE Trans. Autom. Contr., 2004.
  - T. Geyer, F.D. Torrisi and M. Morari, "*Efficient Mode Enumeration of Compositional Hybrid Models*", HSCC'03

Example: Room Temperature



Hybrid dynamics

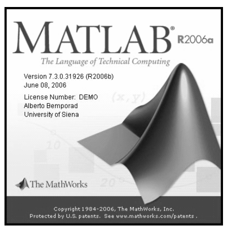
- #1 turns the heater (air conditioning) on whenever he is cold (hot)
- If #2 is cold he turns the heater on, unless #1 is hot
- If #2 is hot he turns the air conditioning on, unless #2 is cold
- Otherwise, heater and air conditioning are off

- $\dot{T}_1 = -\alpha_1(T_1 - T_{\text{amb}}) + k_1(u_{\text{hot}} - u_{\text{cold}})$ (body temperature dynamics of #1)
- $\dot{T}_2 = -\alpha_2(T_2 - T_{\text{amb}}) + k_2(u_{\text{hot}} - u_{\text{cold}})$ (body temperature dynamics of #2)

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

HYSDEL Model

```
SYSTEM heatcool {

INTERFACE {
    STATE { REAL T1 [-10,50];
            REAL T2 [-10,50];
        }
    INPUT { REAL Tamb [-10,50];
        }
    PARAMETER {
        REAL Ts, alpha1, alpha2, k1, k2;
        REAL Thot1, Tcold1, Thot2, Tcold2, Uc, Uh;
        }
}
IMPLEMENTATION {
        AUX { REAL uhot, ucold;
              BOOL hot1, hot2, cold1, cold2;
        }
        AD  { hot1 = T1>=Thot1;
              hot2 = T2>=Thot2;
              cold1 = T1<=Tcold1;
              cold2 = T2<=Tcold2;
        }
        DA  { uhot = {IF cold1 | (cold2 & ~hot1) THEN Uh ELSE 0};
              ucold = {IF hot1 | (hot2 & ~cold1) THEN Uc ELSE 0};
        }
        CONTINUOUS { T1 = T1+Ts*(-alpha1*(T1-Tamb)+k1*(uhot-ucold));
                     T2 = T2+Ts*(-alpha2*(T2-Tamb)+k2*(uhot-ucold));
        }
    }
}
```



Hybrid Toolbox for Matlab

http://www.dii.unisi.it/hybrid/toolbox

`>>S=mld('heatcoolmodel',Ts)` get the MLD model in Matlab

`>>[XX,TT]=sim(S,x0,U);` simulate the MLD model

Hybrid MLD Model

- MLD model

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

- 2 continuous states: (temperatures T_1,T_2)
- 1 continuous input: (room temperature T_{amb})
- 2 auxiliary continuous vars: (power flows u_{hot}, u_{cold})
- 6 auxiliary binary vars: (4 thresholds + 2 for OR condition)
- 20 mixed-integer inequalities

Possible combination of integer variables: $2^6 = 64$

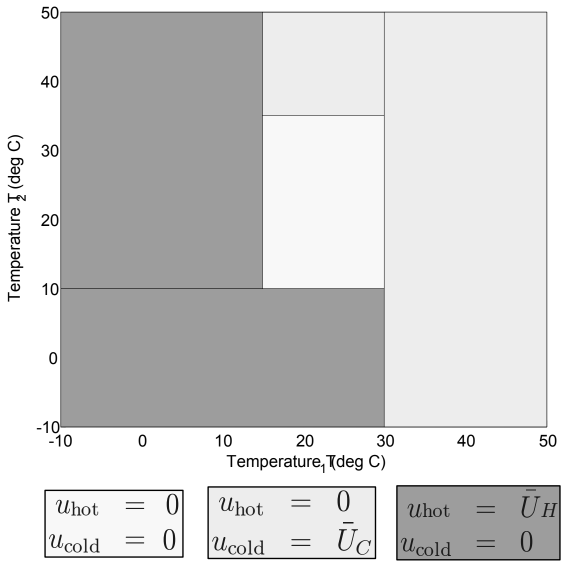
Hybrid PWA Model

- PWA model

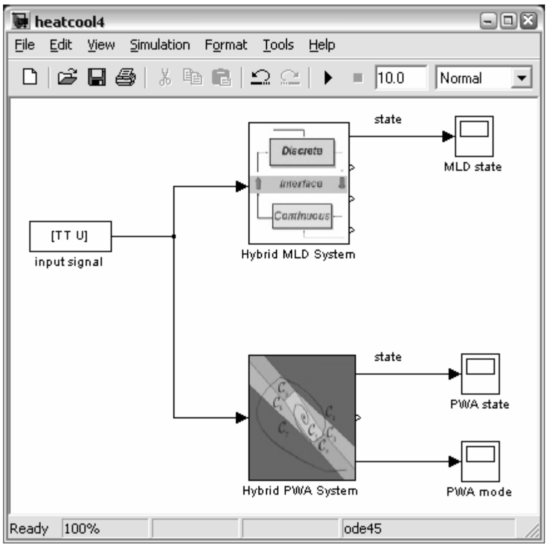
$$
\begin{aligned}
x(k+1) &= A_{i(k)}x(k) + B_{i(k)}u(k) + f_{i(k)}\\
y(k) &= C_{i(k)}x(k) + D_{i(k)}u(k) + g_{i(k)}\\
\\
i(k) \text{ s.t. } & H_{i(k)}x(k) + J_{i(k)}u(k) \le K_{i(k)}
\end{aligned}
$$

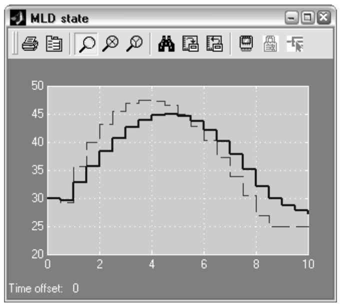
- 2 continuous states:
  (temperatures T_1,T_2)
- 1 continuous input:
  (room temperature T_{amb})
- 5 polyhedral regions
  (partition does not depend on input)

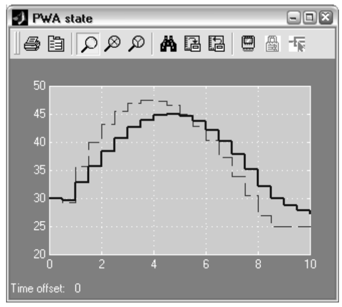
```
>>P=pwa(S);
```

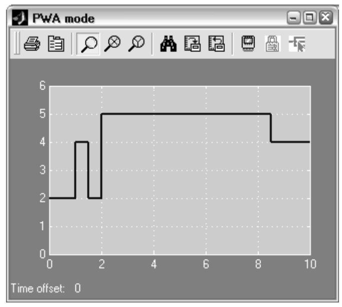


Simulation in Simulink









MLD and PWA models are equivalent

Other Existing Hybrid Models

- Linear complementarity (LC) systems (Heemels, 1999)

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

Ex: mechanical systems
circuits with diodes etc.

- Extended linear complementarity (ELC) systems
  Generalization of LC systems (De Schutter, De Moor, 2000)

- Min-max-plus-scaling (MMPS) systems (De Schutter, Van den Boom, 2000)

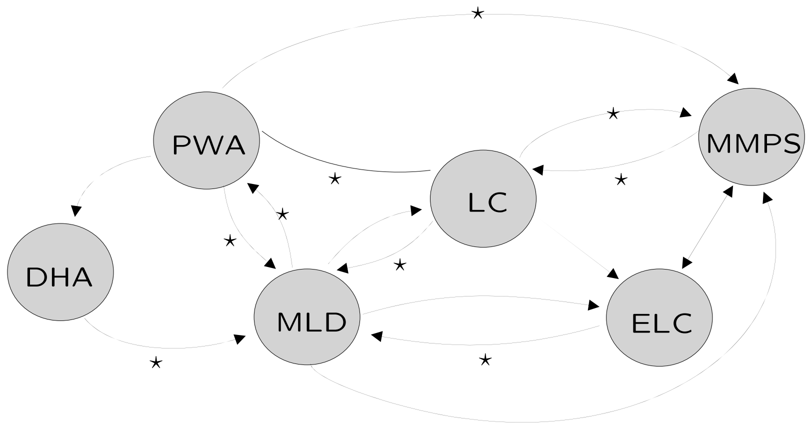
$$\begin{aligned} x(t+1) &= M_x(x(t), u(t), w(t)) \\ y(t) &= M_x(x(t), u(t), w(t)) \\ 0 &\geq M_c(x(t), u(t), w(t)) \end{aligned}$$

MMPS function: defined by the grammar
$M := x_i | \alpha | \max(M_1, M_2) | \min(M_1, M_2) | M_1 + M_2 | \beta M_1$

Example: $x(t+1) = 2\max(x(t), 0) + \min(-\frac{1}{2}u(t), 1)$

Used for modeling discrete-event systems (t=event counter)

Equivalence Results



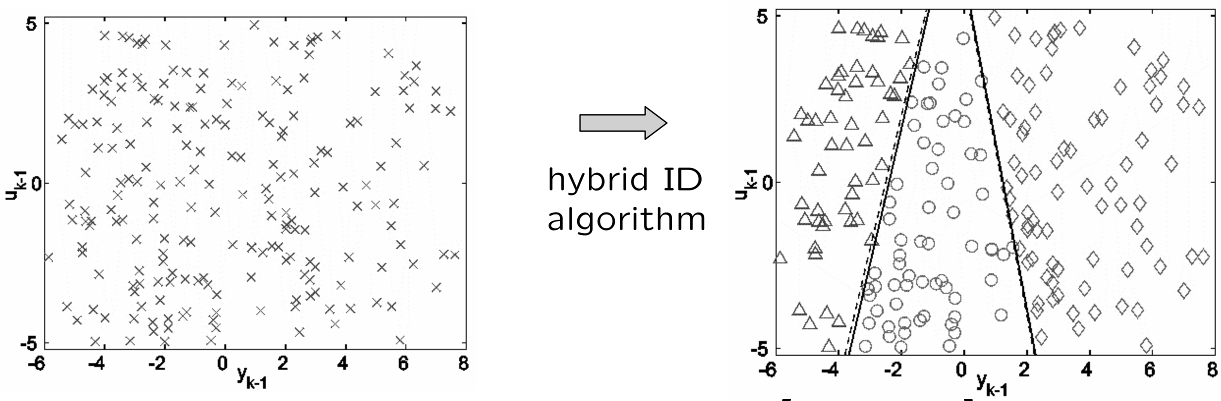
Theorem *All the above six classes of discrete-time hybrid models are equivalent (possibly under some additional assumptions, such as boundedness of input and state variables)*

(Heemels, De Schutter, Bemporad, *Automatica,* 2001)
(Torrisi, Bemporad, *IEEE CST,* 2003)
(Bemporad and Morari, *Automatica*, 1999)
(Bemporad, Ferrari-T.,Morari, *IEEETAC*, 2000)

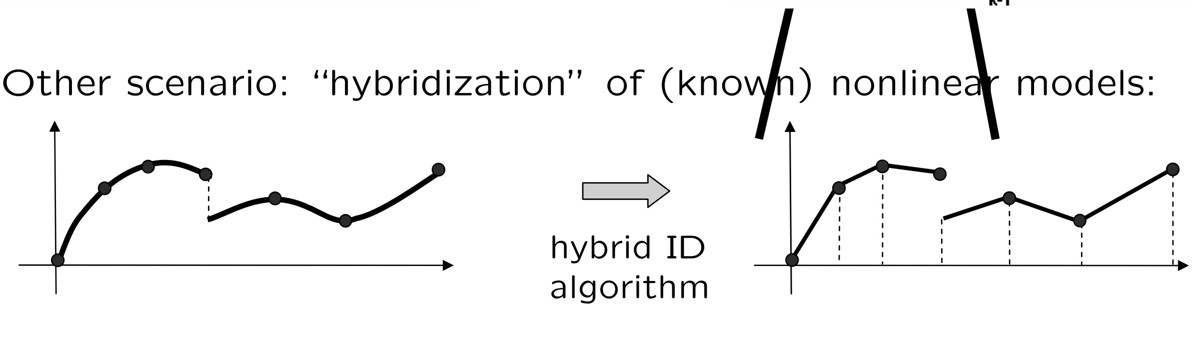
Theoretical properties and analysis/synthesis tools can be transferred from one class to another

Hybrid Systems Identification

Given I/O data, estimate the parameters of the affine submodels **and** the partition of the PWA map



Other scenario: "hybridization" of (known) nonlinear models:



PWA Identification Problem

A. Known Guardlines (partition known, parameters unknown):
least-squares problem **EASY PROBLEM**
(Ljung's ID TBX)

B. Unknown Guardlines (partition *and* parameters unknown):
Generally non-convex, local minima **HARD PROBLEM!**

<u>Some recent approaches to Hybrid ID:</u>

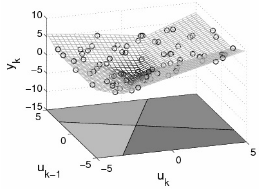
- K-means clustering in a feature space (Ferrari-Trecate, Muselli, Liberati, Morari, 2003)
- Bayesian approach (Juloski, Heemels, Weiland, 2004)
- Mixed-integer programming (Roll, Bemporad, Ljung, 2004)
- Bounded error (partition of infeasible set of inequalities) (Bemporad, Garulli, Paoletti, Vicino, 2003)
- Algebraic geometric approach (Vidal, Soatto, Sastry, 2003)
- Hyperplane clustering in data space (Münz, Krebs, 2002)

Hybrid Identification Toolboxes

- PWAID – Piecewise Affine Identification Toolbox
  (Paoletti, Roll, 2007)

  http://www.control.isy.liu.se/~roll/PWAID/

  - Mixed-integer approach (hinging-hyperplane models)
  - Bounded error approach



- HIT – Hybrid Identification Toolbox
  (Ferrari-Trecate, 2006)

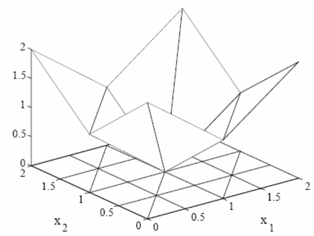
  http://www-rocq.inria.fr/who/Giancarlo.Ferrari-Trecate/HIT_toolbox.html

  - K-means clustering in a feature space

- PWL Toolbox
  (Julian, 2000)

  http://www.pedrojulian.com

  - High level canonical PWL representations



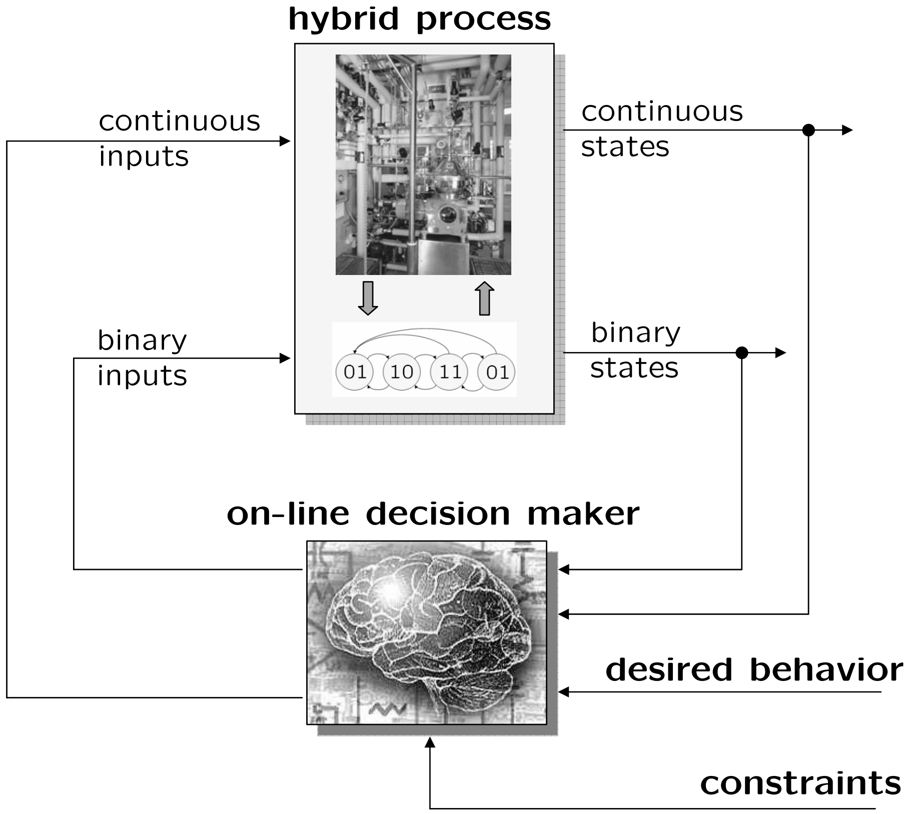
Outline

- ✓ Basics of Model Predictive Control (MPC)
- ✓ Hybrid models for MPC
- MPC of hybrid systems
- Explicit MPC (multiparametric programming)
- Optimization-based reachability analysis
- Examples

MPC of Hybrid Systems

- Problem setup
- Convergence properties
- Computational aspects

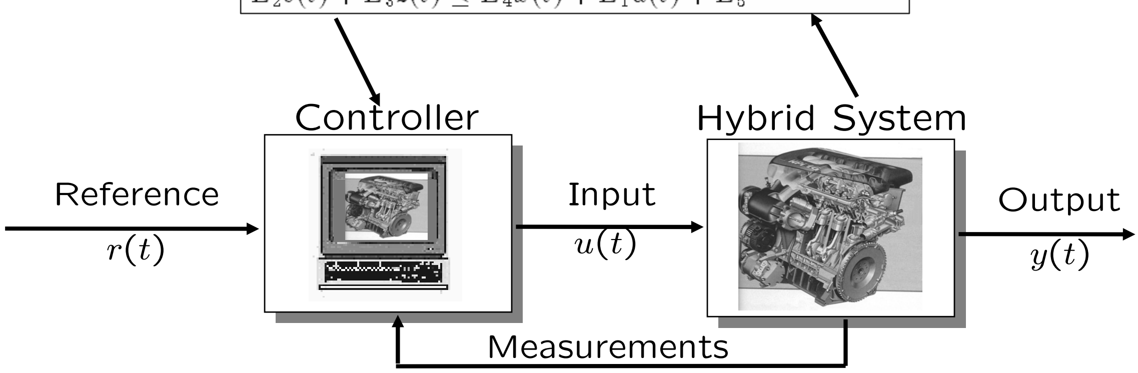
Hybrid Control Problem



Model Predictive Control of Hybrid Systems

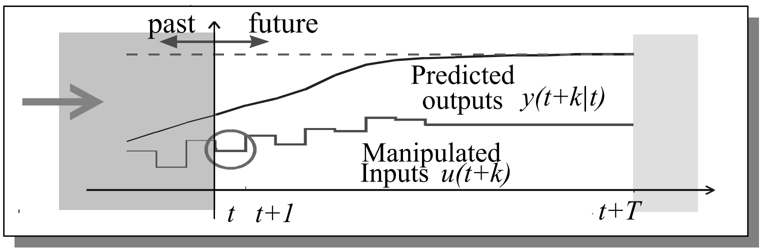
MLD model

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



- 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

MPC for Hybrid Systems



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:

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

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

- 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 $\forall Q, R \succ 0$, $\forall \sigma > 0$*

$$\lim_{t\to\infty} y(t) = r$$
$$\lim_{t\to\infty} u(t) = u_r$$

$\lim_{t\to\infty} x(t) = x_r$, $\lim_{t\to\infty} \delta(t) = \delta_r$, $\lim_{t\to\infty} z(t) = z_r$, *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,

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

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

Note: Global optimum not needed for convergence !

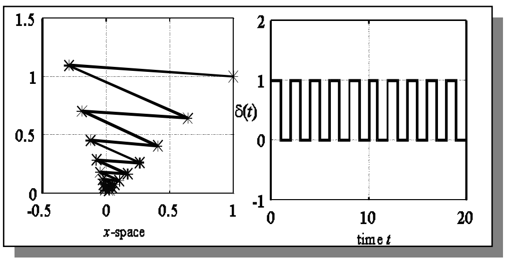
Hybrid MPC - Example

PWA system:

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

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

Open loop behavior



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

Hybrid MPC - Example

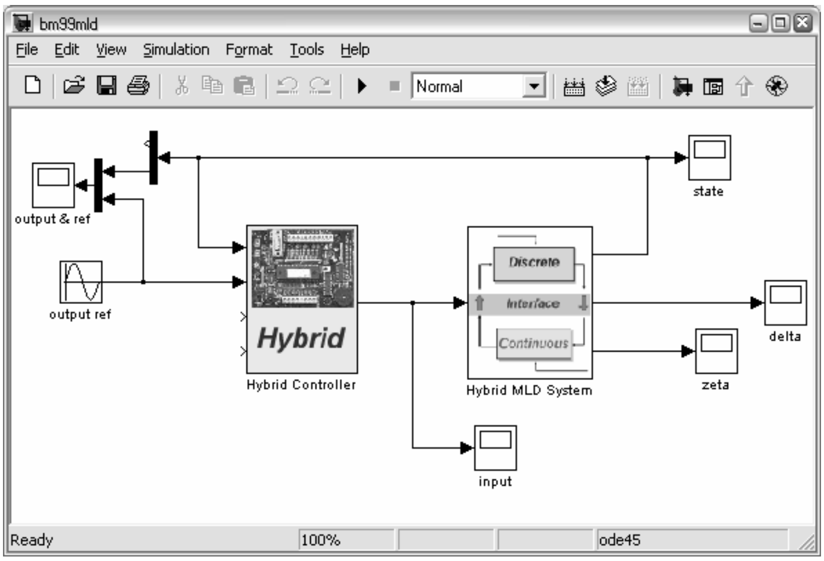
HYSDEL model

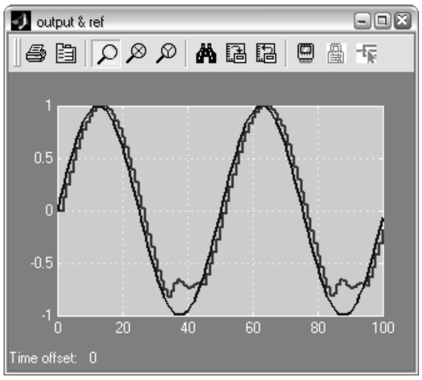
```
/* 2x2 PWA system  - Example from the paper
   A. Bemporad and M. Morari, ``Control of systems integrating logic, dynamics,
   and constraints,'' Automatica, vol. 35, no. 3, pp. 407-427, 1999.
   (C) 2003 by A. Bemporad, 2003 */

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;   }
                }
}
```

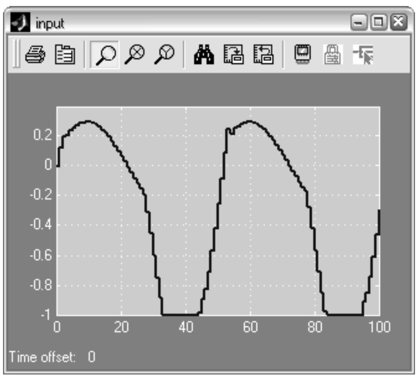
/demos/hybrid/bm99.hys

Hybrid MPC - Example

Closed loop:







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

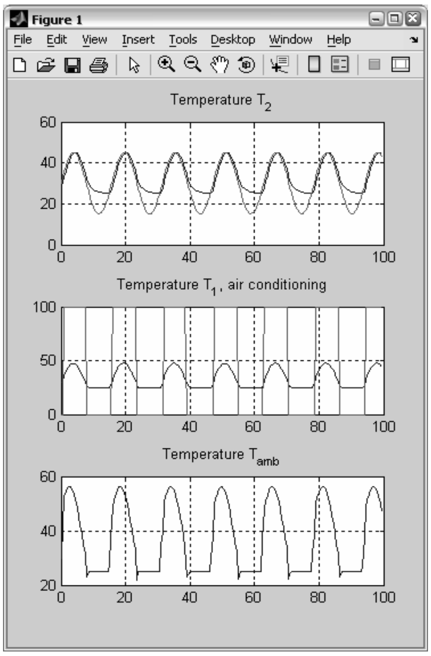
Hybrid MPC – Temperature Control

```
>>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];
```

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

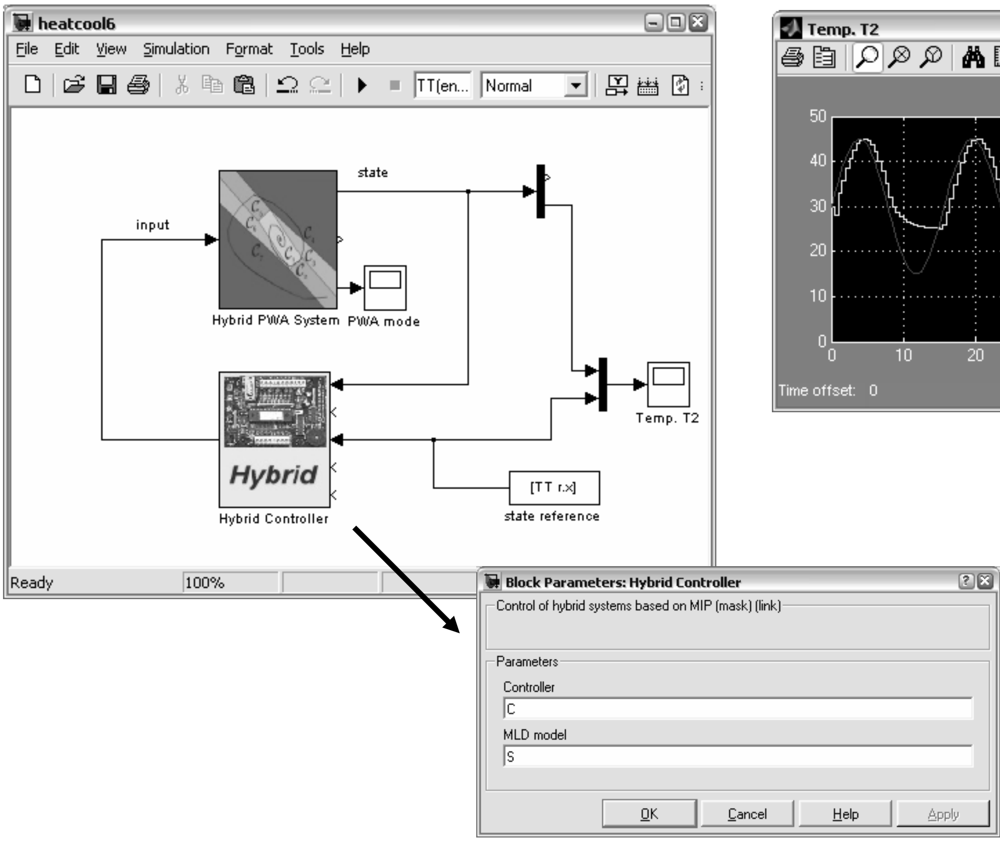
```
>>C=hybcon(S,Q,N,limits,refs);
```

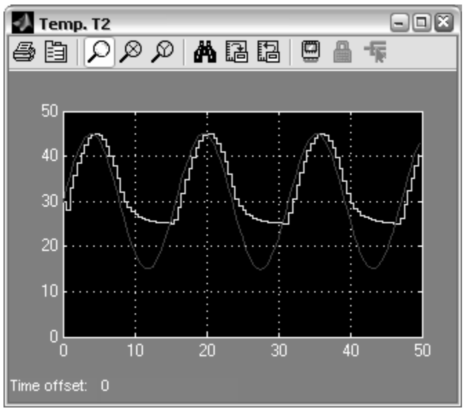
```
>> C

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





Optimal Control of Hybrid Systems: Computational Aspects

MIQP Formulation of MPC

(Bemporad, Morari, 1999)

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

$$\text{subject to} \begin{cases} x(t+1) = Ax(t) + B_1u(t) + B_2\delta(t) + B_3z(t) + B_5 \\ y(t) = Cx(t) + D_1u(t) + D_2\delta(t) + D_3z(t) + D_5 \\ E_2\delta(t) + E_3z(t) \le E_4x(t) + E_1u(t) + E_5 \end{cases}$$

- Optimization vector:

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

$$\Rightarrow \quad \min_{\xi} \frac{1}{2}\xi'H\xi + x(0)'F\xi + \frac{1}{2}x'(0)Yx(0)$$
$$\text{subj. to} \quad G\xi \le W + Sx(t)$$

Mixed Integer Quadratic Program (MIQP)

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

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

MILP Formulation of MPC

(Bemporad, Borrelli, Morari, 2000)

$$\min_{\xi} \quad J(\xi, x(0)) = \sum_{t=0}^{T-1} \|Qy(t)\|_\infty + \|Ru(t)\|_\infty$$

subject to MLD model

- Basic trick: introduce slack variables

$$\min_x |x| \quad \Rightarrow \quad \min_{x,\epsilon} \ \epsilon \quad \text{s.t.} \ \epsilon \geq x, \ \epsilon \geq -x$$

$$\begin{cases} \epsilon_k^x \geq \|Qy(t+k|t)\|_\infty \\ \epsilon_k^u \geq \|Ru(t+k)\|_\infty \end{cases} \Rightarrow \begin{cases} \epsilon_k^x \geq [Qy(t+k|t)]_i & i=1,\dots,p \quad k=1,\dots,T-1 \\ \epsilon_k^x \geq -[Qy(t+k|t)]_i & i=1,\dots,p \quad k=1,\dots,T-1 \\ \epsilon_k^u \geq [Ru(t+k)]_i & i=1,\dots,m \quad k=0,\dots,T-1 \\ \epsilon_k^u \geq -[Ru(t+k)]_i & i=1,\dots,m \quad k=0,\dots,T-1 \end{cases}$$

- Optimization vector:

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

$$\Rightarrow \quad \min_{\xi} \quad J(\xi, x(0)) = \sum_{k=0}^{T-1} \epsilon_k^x + \epsilon_k^u \qquad \text{s.t.} \quad G\xi \leq W + Sx(0)$$

Mixed Integer Linear Program (MILP)

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

Mixed-Integer Program Solvers

- Mixed-integer programming is *NP*-hard

BUT

- Extremely rich literature in operations research (still very active)

MILP/MIQP is nowadays a technology (CPLEX, Xpress-MP, BARON, GLPK, see e.g. http://plato.la.asu.edu/bench.html for a comparison)

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

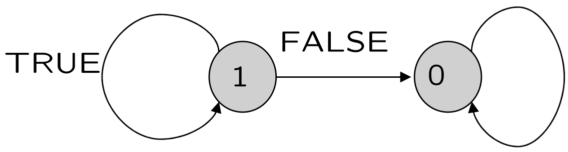
(Ingimundarson, Ocampo-Martinez, Bemporad, 2007)

"Hybrid" Solvers

Main drawback of Mixed-Integer Programming for solving hybrid MPC problems

Loss of the original Boolean structure

$$\begin{cases} x_\ell(k+1) + (1 - x_\ell(k)) + (1 - u_\ell(k)) \geq 1 \\ x_\ell(k) + (1 - x_\ell(k+1)) \geq 1 \\ u_\ell(k) + (1 - x_\ell(k+1)) \geq 1 \end{cases}$$

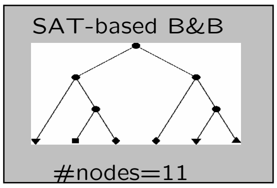


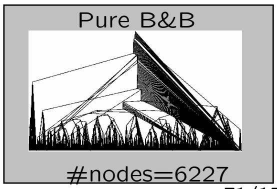
$$x_\ell(k+1) = x_\ell(k) \wedge u_\ell(k)$$

- Possibility of combining symbolic + numerical solvers
  Example: SAT + linear programming

		Sat instances		Unsat instances	
N. Vars	N. Cons	zCHAFF	CPLEX	zCHAFF	CPLEX
20	91	0	0.036	-	-
50	218	0	0.343	0	0.453
75	325	0	0.203	0	3.671
100	430	0	23.328	0	33.921
125	538	0.016	15.171	0.031	209.766
150	645	0.031	20.625	0.281	4949.58
175	753	0.031	> 1500	0.891	> 5000

(Bemporad, Giorgetti, IEEE TAC 2006)





Main Drawbacks of On-line Combinatorial Optimization

- Computation time may be too long: good for large sampling times (>1s), but not for fast sampling applications (e.g. 1ms)
- Requires relatively expensive hardware (e.g. a PC, not suitable on inexpensive μ-controllers)
- Software complexity: MIP solver code difficult to certify (not good for safety critical applications)
- Worst-case computation time is often hard to estimate (over-estimation is usually exaggerate)

Any way to get rid of on-line solvers ?

Outline

- ✓ Basics of Model Predictive Control (MPC)
- ✓ Hybrid models for MPC
- ✓ MPC of hybrid systems
- Explicit MPC (multiparametric programming)
- Optimization-based reachability analysis
- Examples

Explicit Model Predictive Control

- Multiparametric quadratic programming
- Explicit linear MPC
- Explicit hybrid MPC

On-Line vs. Off-Line Optimization

$$\begin{array}{ll} \min_{U} & \frac{1}{2}U'HU + x'(t)F'U + \frac{1}{2}x'(t)Yx(t) \\ \text{subj. to} & GU \leq W + Sx(t), \end{array}$$

- **On-line** optimization: given $x(t)$ solve the problem at each time step t (the control law $u{=}u(x)$ is **implicitly** defined by the QP solver)

➡ Quadratic Program (QP)

- **Off-line** optimization: solve the QP **for all** $x(t)$ to find the control law $u{=}u(x)$ **explicitly**

➡ multi-parametric Quadratic Program (mp-QP)

Multiparametric Quadratic Programming

(Bemporad et al., 2002)

$$\begin{array}{ll} \min_{U} & \frac{1}{2}U'HU + x'F'U + \cancel{\frac{1}{2}xYx} \\ \text{subj. to} & GU \leq W + Sx, \end{array}$$

$U \triangleq [u'_0 \ \dots \ u'_{N-1}]'$

$U \in \mathbb{R}^r$, $r \triangleq mN_u$

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

- Objective: solve the QP **for all** $x \in X \subseteq \mathbb{R}^n$

- Assumptions: $\begin{bmatrix} H & F \\ F' & Y \end{bmatrix} \succeq 0$ (always satisfied if QP problem originates from optimal control problem)

  $H \succ 0$ (can be easily satisfied, e.g. by choosing input weight matrix >0)

Linearity of the Solution

$x_0 \in X$ ⇨ solve QP to find $U^*(x_0)$, $\lambda^*(x_0)$

⇨ identify active constraints at $U^*(x_0)$

⇨ form matrices $\tilde{G}$, $\tilde{W}$, $\tilde{S}$ by collecting active constraints: $\tilde{G}U^*(x_0) - \tilde{W} - \tilde{S}x_0 = 0$

KKT optimality conditions:

(1) $HU + Fx + G'\lambda = 0$, (2) $\tilde{G}U - \tilde{W} - \tilde{S}x = 0$
(3) $\lambda_i(G^iU - W^i - S^ix) = 0$, (4) $\hat{G}U \leq \hat{W} + \hat{S}x$
(5) $\tilde{\lambda}_i \geq 0$, $\hat{\lambda}_i = 0$

$\hat{G}$=rows of G not in $\tilde{G}$ (inactive constraints)

From (1) : $U = -H^{-1}(Fx + \tilde{G}'\tilde{\lambda})$

From (2) :

$$\tilde{\lambda}(x) = -(\tilde{G}H^{-1}\tilde{G}')^{-1}(\tilde{W} + (\tilde{S} + \tilde{G}H^{-1}F)x).$$
$$U(x) = H^{-1}[\tilde{G}'(\tilde{G}H^{-1}\tilde{G}')^{-1}(\tilde{W}+(\tilde{S}+\tilde{G}H^{-1}F)x)-Fx]$$

➡ In some neighborhood of x_0, λ and U are explicit affine functions of x !

Determining a Critical Region

- Impose primal and dual feasibility:

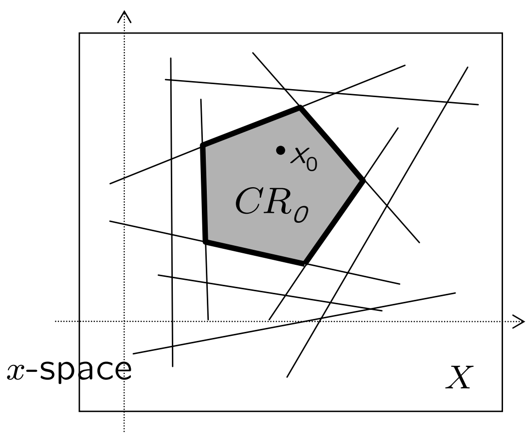
$$\hat{G}U(x) \leq \hat{W} + \hat{S}x$$
$$\tilde{\lambda}(x) \geq 0$$

➡ linear inequalities in x !

- Remove redundant constraints: (this requires solving LP's)

➡ critical region CR_0

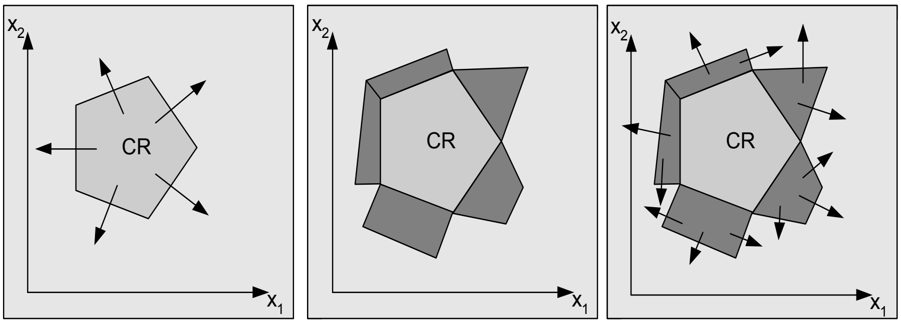
$$CR_0 = \{x \in X : \mathcal{A}x \leq \mathcal{B}\}$$



- CR_0 is the set of all and only parameters x for which $\tilde{G}$, $\tilde{W}$, $\tilde{S}$ is the optimal combination of active constraints at the optimizer

Getting All Critical Regions

(Tøndel, Johansen, Bemporad, 2003)



The active set of a neighboring region is found by using the active set of the current region + knowledge of the type of hyperplane we are crossing:

$\widehat{G}^i U(x) \leq \widehat{W}^i + \widehat{S}^i x \Rightarrow$ The corresponding constraint is **added** to the active set

$\tilde{\lambda}_j(x) \geq 0 \Rightarrow$ The corresponding constraint is **withdrawn** from the active set

Properties of multiparametric-QP

Theorem 1 *Consider a multi-parametric quadratic program with* $H \succ 0$, $\left[\begin{smallmatrix} H & F \\ F' & Y \end{smallmatrix}\right] \succeq 0$. *The set* X^* *of parameters* x *for which the problem is feasible is a polyhedral set, the value function* $V^* : X^* \mapsto \mathbb{R}$ *is piecewise quadratic, convex and continuous and the optimizer* $U^* : X^* \mapsto \mathbb{R}^r$ *is piecewise affine and continuous.*

$$U^*(x) = \arg\min_U \ \tfrac{1}{2}U'HU + x'F'U \quad \text{subj. to } GU \leq W + Sx$$

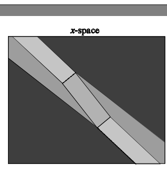
continuous, piecewise affine

$$V^*(x) = \tfrac{1}{2}x'Yx + \min_U \ \tfrac{1}{2}U'HU + x'F'U \quad \text{subj. to } GU \leq W + Sx$$

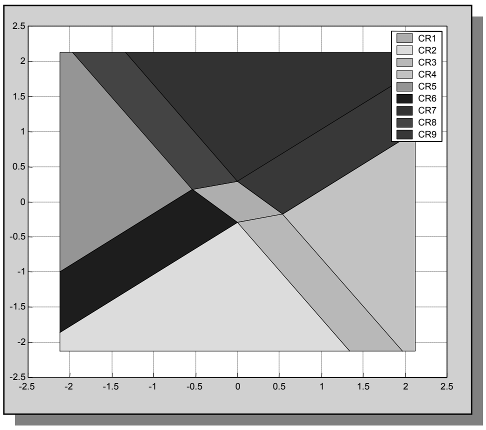
convex, continuous, piecewise quadratic, C^1 (if no degeneracy)

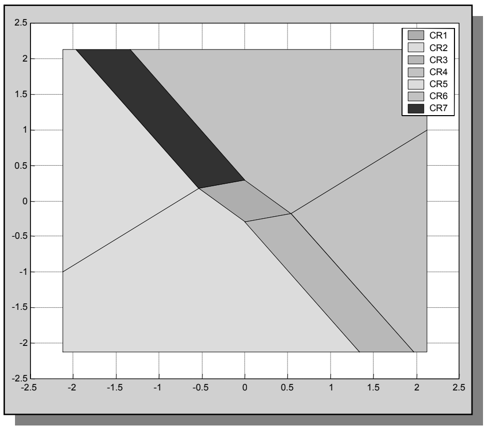
Corollary: The linear MPC controller is a continuous piecewise affine function of the state

$$u(x) = \begin{cases} F_1 x + g_1 & \text{if } H_1 x \leq K_1 \\ \vdots & \vdots \\ F_M x + g_M & \text{if } H_M x \leq K_M \end{cases}$$



Complexity Reduction





$$U(x) \triangleq [u_0'(x)\ u_1'(x)\ \dots\ u_{N-1}'(x)]'$$

Regions where the first component of the solution is the same can be joined (when their union is convex).

(Bemporad, Fukuda, Torrisi, *Computational Geometry*, 2001)

Double Integrator Example

- System: $y(t) = \frac{1}{s^2}u(t)$ sampling + ZOH T_s=1 s

$$\begin{aligned} x(t+1) &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) \\ y(t) &= \begin{bmatrix} 1 & 0 \end{bmatrix} x(t) \end{aligned}$$

- Constraints: $-1 \le u(t) \le 1$
- Control objective: minimize $\sum_{t=0}^{\infty} y^2(t) + \frac{1}{100}u^2(t)$

$$u(t+k) = K_{LQ}x(t+k|t),\ \forall k \ge N_u$$

- Optimization problem: for N_u=2

$$H = \begin{bmatrix} 0.8365 & 0.3603 \\ 0.3603 & 0.2059 \end{bmatrix},\ F = \begin{bmatrix} 0.4624 & 1.2852 \\ 0.1682 & 0.5285 \end{bmatrix}$$

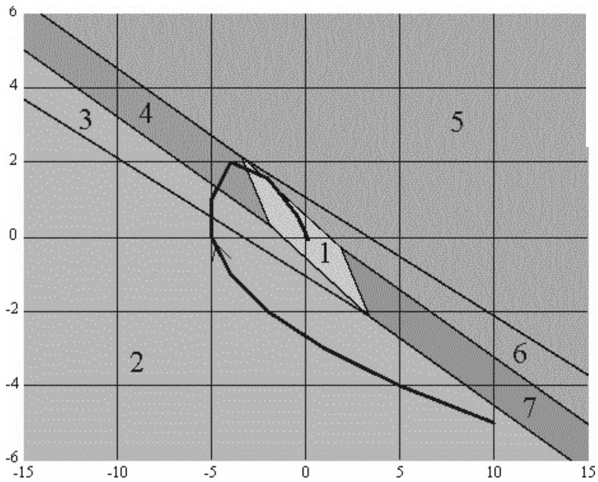
(cost function is normalized by max svd(H))

$$G = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix},\ W = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix},\ S = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

mp-QP solution

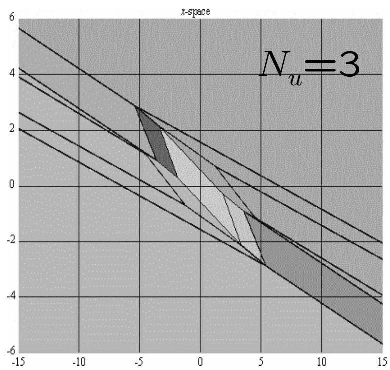
N_u=2

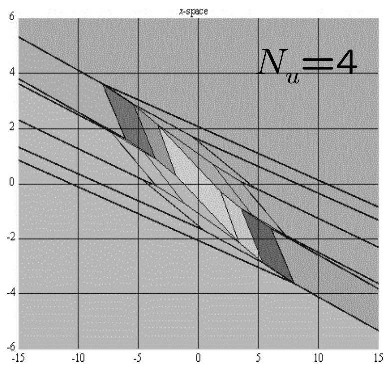
$$u(x) = \begin{cases} \begin{bmatrix} -0.8166 & -1.7499 \end{bmatrix} x & \text{if } \begin{bmatrix} -0.8166 & -1.7499 \\ 0.8166 & 1.7499 \\ 0.6124 & 0.4957 \\ -0.6124 & -0.4957 \end{bmatrix} x \leq \begin{bmatrix} 1.0000 \\ 1.0000 \\ 1.0000 \\ 1.0000 \end{bmatrix} & \text{(Region \#1)} \\ 1.0000 & \text{if } \begin{bmatrix} 0.3864 & 1.0738 \\ 0.2970 & 0.9333 \end{bmatrix} x \leq \begin{bmatrix} -1.0000 \\ -1.0000 \end{bmatrix} & \text{(Region \#2)} \\ 1.0000 & \text{if } \begin{bmatrix} 0.9712 & 2.6991 \\ -0.2970 & -0.9333 \\ 0.8166 & 1.7499 \end{bmatrix} x \leq \begin{bmatrix} -1.0000 \\ 1.0000 \\ -1.0000 \end{bmatrix} & \text{(Region \#3)} \\ \begin{bmatrix} -0.5528 & -1.5364 \end{bmatrix} x + 0.4308 & \text{if } \begin{bmatrix} -0.9712 & -2.6991 \\ 0.3864 & 1.0738 \\ 0.6124 & 0.4957 \end{bmatrix} x \leq \begin{bmatrix} 1.0000 \\ 1.0000 \\ -1.0000 \end{bmatrix} & \text{(Region \#4)} \\ -1.0000 & \text{if } \begin{bmatrix} -0.3864 & -1.0738 \\ -0.2970 & -0.9333 \end{bmatrix} x \leq \begin{bmatrix} -1.0000 \\ -1.0000 \end{bmatrix} & \text{(Region \#5)} \\ -1.0000 & \text{if } \begin{bmatrix} -0.9712 & -2.6991 \\ 0.2970 & 0.9333 \\ -0.8166 & -1.7499 \end{bmatrix} x \leq \begin{bmatrix} -1.0000 \\ 1.0000 \\ -1.0000 \end{bmatrix} & \text{(Region \#6)} \\ \begin{bmatrix} -0.5528 & -1.5364 \end{bmatrix} x - 0.4308 & \text{if } \begin{bmatrix} -0.3864 & -1.0738 \\ 0.9712 & 2.6991 \\ -0.6124 & -0.4957 \end{bmatrix} x \leq \begin{bmatrix} 1.0000 \\ 1.0000 \\ -1.0000 \end{bmatrix} & \text{(Region \#7)} \end{cases}$$

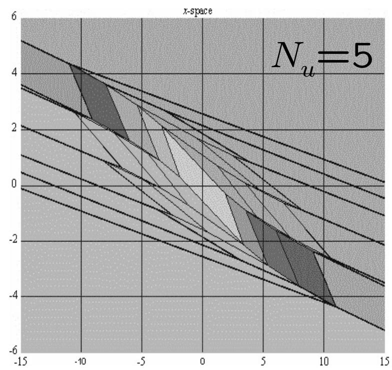


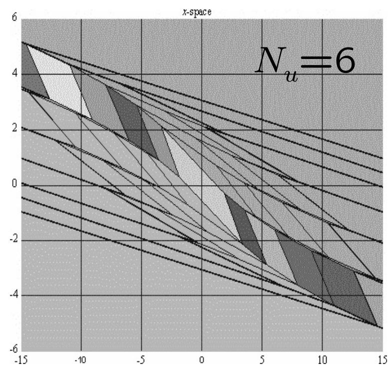
go to demo **/demos/linear/doubleintexp.m** (Hyb-Tbx)

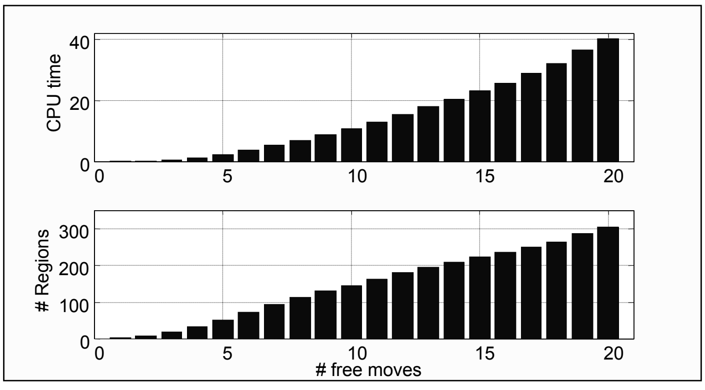
Complexity











(is the number of regions finite for $N_u \to \infty$?)

Complexity

- Worst-case complexity analysis:

$$M \triangleq \sum_{\ell=0}^{q} \binom{q}{\ell} = 2^q \quad \text{combinations of active constraints}$$

- Usually the number of regions is much smaller, as many combinations of active constraints are never feasible and optimal at any parameter vector x
- Strongest dependence on the number q of constraints
- Strong dependence on the number N_u of free moves
- Weak dependence on the number n of parameters x
- Example:

states\horizon	$N=1$	$N=2$	$N=3$	$N=4$	$N=5$
n=2	3	6.7	13.5	21.4	19.3
n=3	3	6.9	17	37.3	77
n=4	3	7	21.65	56	114.2
n=5	3	7	22	61.5	132.7
n=6	3	7	23.1	71.2	196.3
n=7	3	6.95	23.2	71.4	182.3
n=8	3	7	23	70.2	207.9

Data averaged over 20 randomly generated single-input single-output systems subject to input saturation (q=2N)

Reference Tracking, MIMO System

QP-based vs. Explicit MPC:

$2N$	QP (ms) average	worst	explicit (ms) average	worst	regions	[storage kb]
4	1.1	1.5	0.005	0.1	25	16
8	1.3	1.9	0.023	1.1	175	78
20	2.5	2.6	0.038	3.3	1767	811
30	5.3	7.2	0.069	4.4	5162	2465
40	10.9	13.0	0.239	15.6	11519	5598

(Intel Centrino 1.4 GHz)

Average time on 100 random [states,references, previous inputs]

Worst-case time on 100 random [states,references, previous inputs]

Organization of explicit solution on a binary tree can improve CPU time (worst & average) from n_r to about $\log_2 n_r$ (n_r=number of regions)

(Tøndel, Johansen, Bemporad, *Automatica*, 2003)

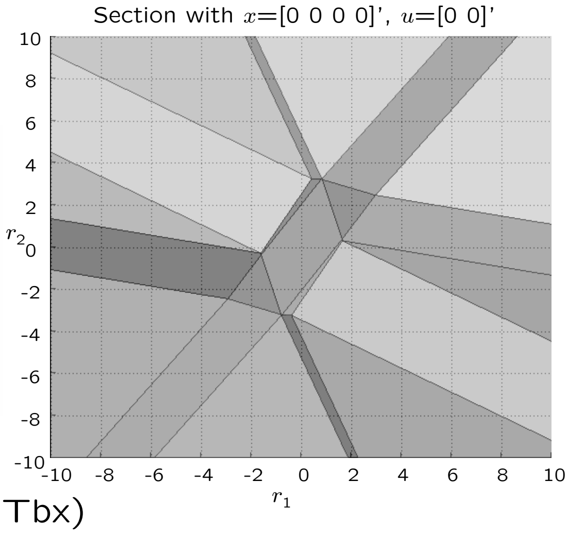
Example: AFTI-16

- Linearized model:

$$\begin{cases} \dot{x} = \begin{bmatrix} -.0151 & -60.5651 & 0 & -32.174 \\ -.0001 & -1.3411 & .9929 & 0 \\ .00018 & 43.2541 & -.86939 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x + \\ \begin{bmatrix} -2.516 & -13.136 \\ -.1689 & -.2514 \\ -17.251 & -1.5766 \\ 0 & 0 \end{bmatrix} u \\ y = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x, \end{cases}$$

- Inputs: elevator and flaperon angle
- Outputs: attack and pitch angle
- Sampling time: $T_s = .05$ s (+ zero-order hold)
- Constraints: max 25° on both angles
- Open-loop response: unstable
  (open-loop poles: $-7.6636, -0.0075 \pm 0.0556j, 5.4530$)

Explicit controller: 8 parameters, 51 regions



go to demo **`linear/afti16.m`** (Hyb-Tbx)

Explicit Hybrid MPC (MLD)

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

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

- 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

$$\min_\xi \sum_{k=0}^{T-1} \epsilon_i^x + \epsilon_i^u$$
$$\text{s.t.} \quad G\xi \le W + S \begin{bmatrix} x(t) \\ r(t) \end{bmatrix}$$

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

Multiparametric MILP

$$\begin{aligned} \min_{\xi=\{\xi_c,\xi_d\}} \quad & f'\xi_c + d'\xi_d \\ \text{s.t.} \quad & G\xi_c + E\xi_d \le W + Fx \end{aligned} \qquad \begin{aligned} &\xi_c \in \mathbb{R}^n \\ &\xi_d \in \{0,1\}^m \end{aligned}$$

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

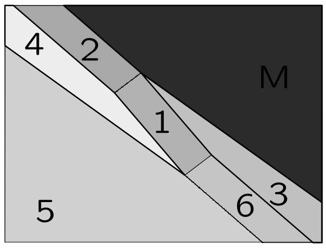
(Dua, Pistikopoulos, 1999)

- **Theorem:** The multiparametric solution $\xi^*(x)$ is piecewise affine

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

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

(x,r)-space



Explicit Hybrid MPC (PWA)

$$\begin{aligned} \min_U J(U,x,r) = & \sum_{k=0}^{T-1} \|R(y(k)-r)\|_p + \|Qu(k)\|_p \\ \text{subject to} & \begin{cases} \text{PWA model} \\ x(0) = x \end{cases} \end{aligned}$$

$p = 1, 2, \infty$

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

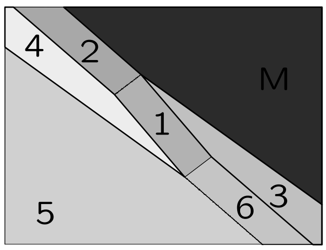
$\|v\|_\infty = \max |v_i|$

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

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

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

(x,r)-space



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

Computation of Explicit Hybrid MPC (PWA)

<u>Method A</u>: (Borrelli, Baotic, Bemporad, Morari, *Automatica*, 2005)

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

<u>Method B</u>: (Bemporad, *Hybrid Toolbox*, 2003) (Alessio, Bemporad, ADHS 2006)(Mayne, ECC 2001)

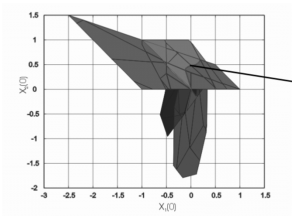
1 - Use backwards (=DP) reachability analysis for enumerating all feasible mode sequences $I=\{i(0),i(1),\ldots,i(T\text{-}1)\}$;

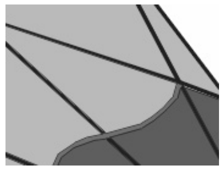
2 - For each fixed sequence I, solve the explicit finite-time optimal control problem for the corresponding linear time-varying system (mpQP or mpLP);

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).

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





Hybrid Control Examples (Revisited)

Hybrid Control Example

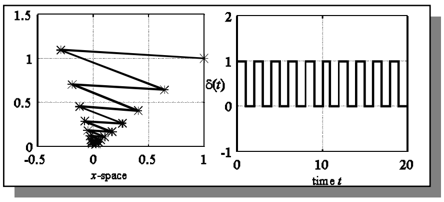
PWA system:

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

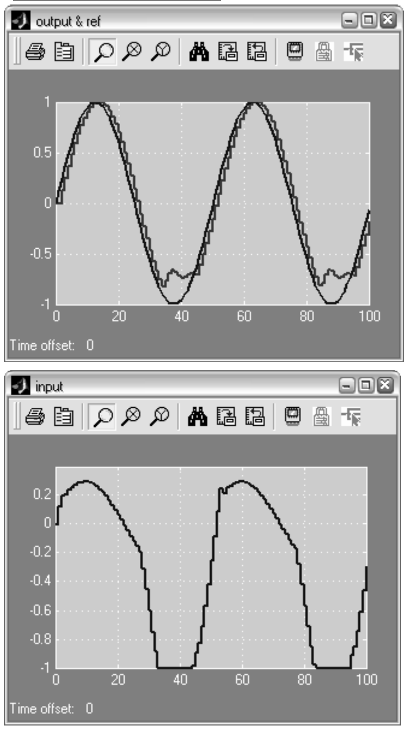
Constraints: $-1 \leq u(t) \leq 1$

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

Open loop behavior:



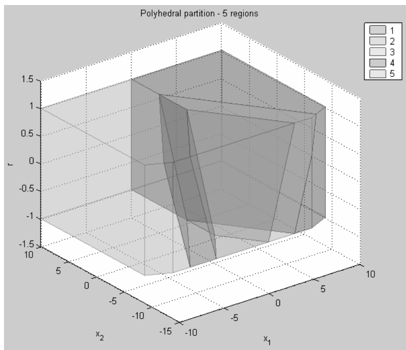
Closed loop:



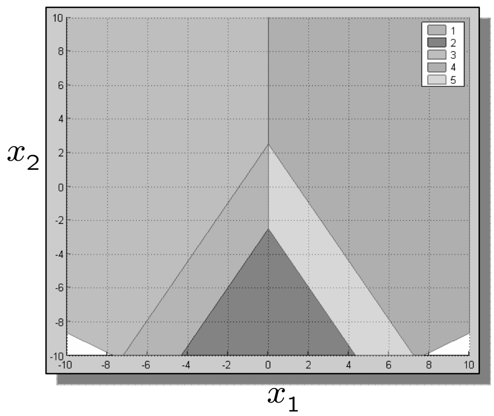
HybTbx: **/demos/hybrid/bm99sim.m**

Explicit PWA Controller

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



Section with $r=0$



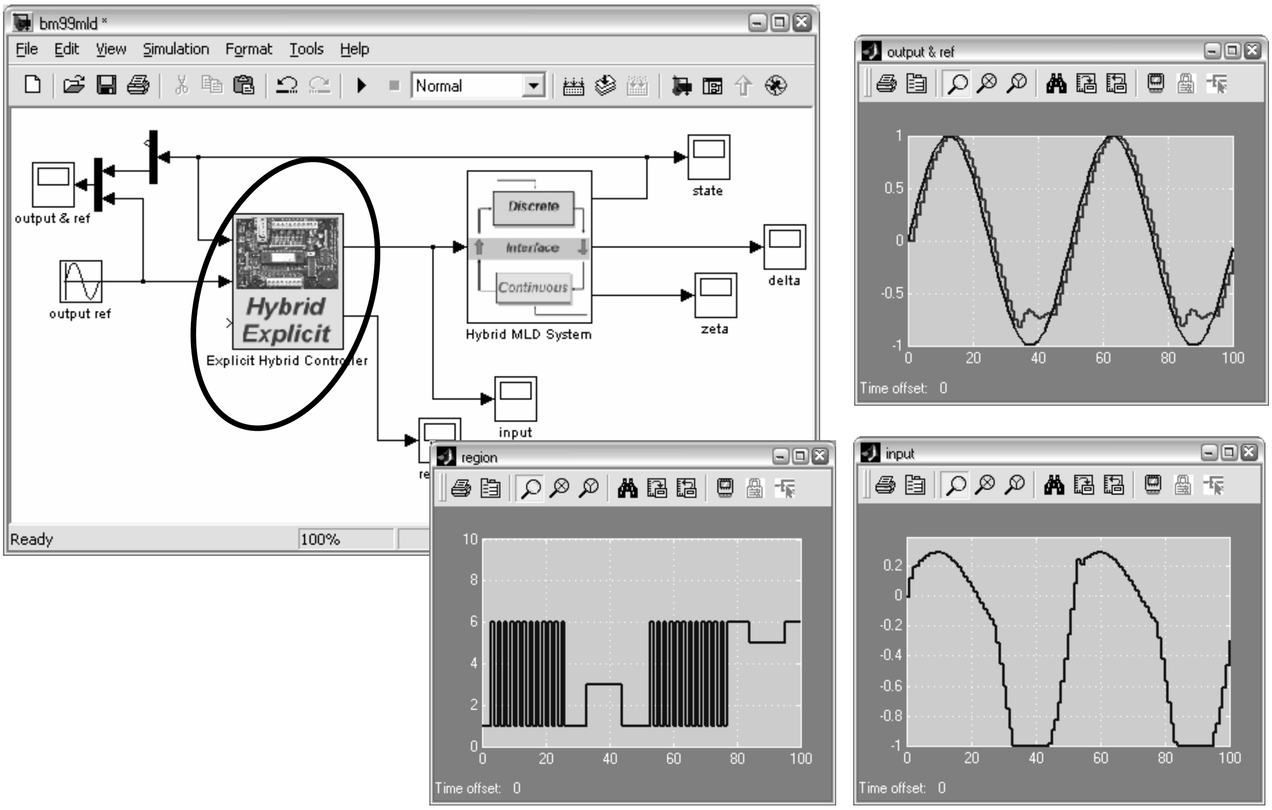
HybTbx: **/demos/hybrid/bm99sim.m**

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

PWA law $\equiv$ MPC law !

Hybrid Control Example

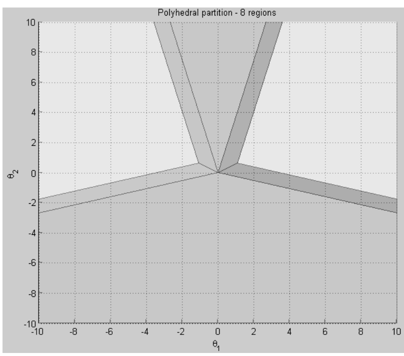
Closed loop:



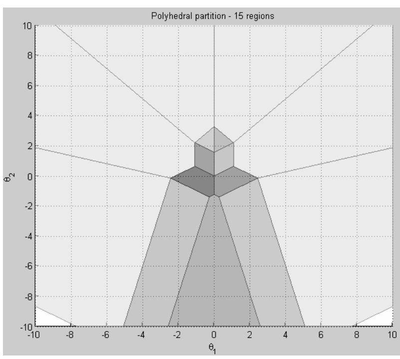
Explicit PWA Regulator

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

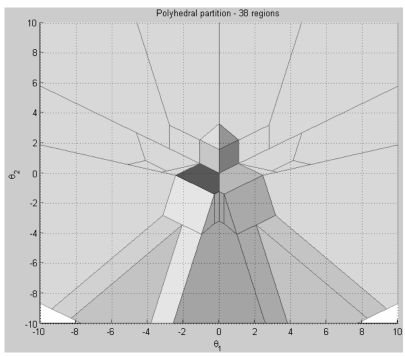
Prediction horizon N=1



Prediction horizon N=2



Prediction horizon N=3

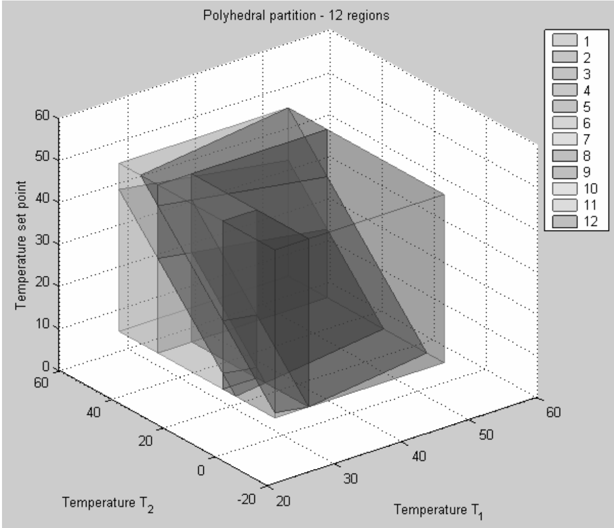


HybTbx: **`/demos/hybrid/bm99benchmark.m`**

Explicit MPC – Temperature Control

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

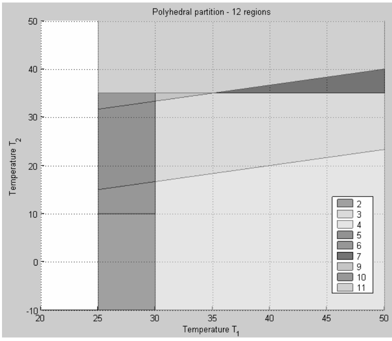
```
>> E

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.
>>
```



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

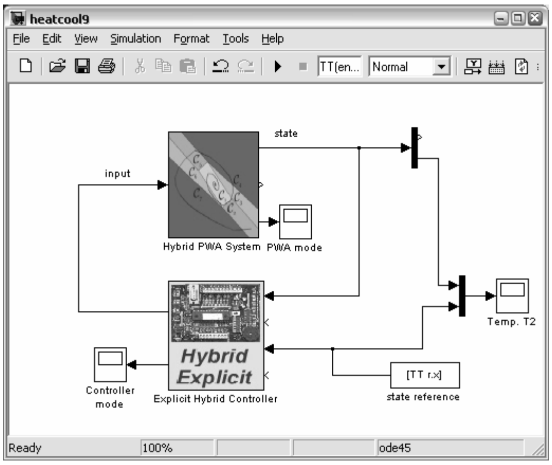
s.t. $x_1(k) \geq 25 \;\; k = 1, 2$

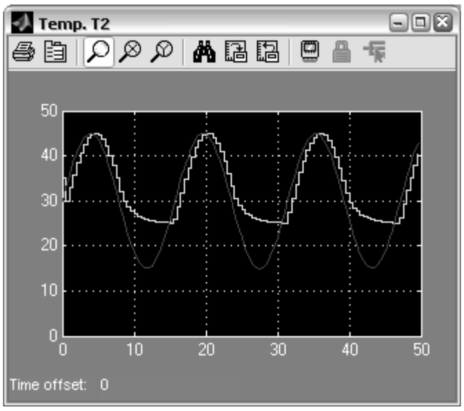
PWA model

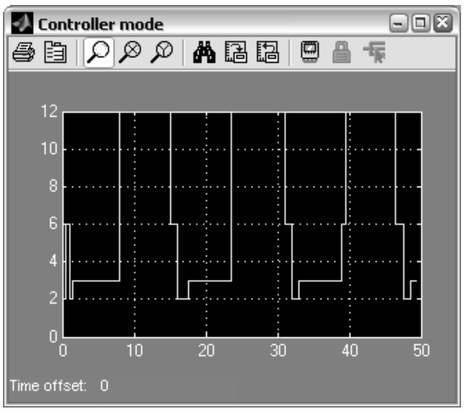


Section in the (T_1,T_2)-space for $T_{ref} = 30$

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,
```

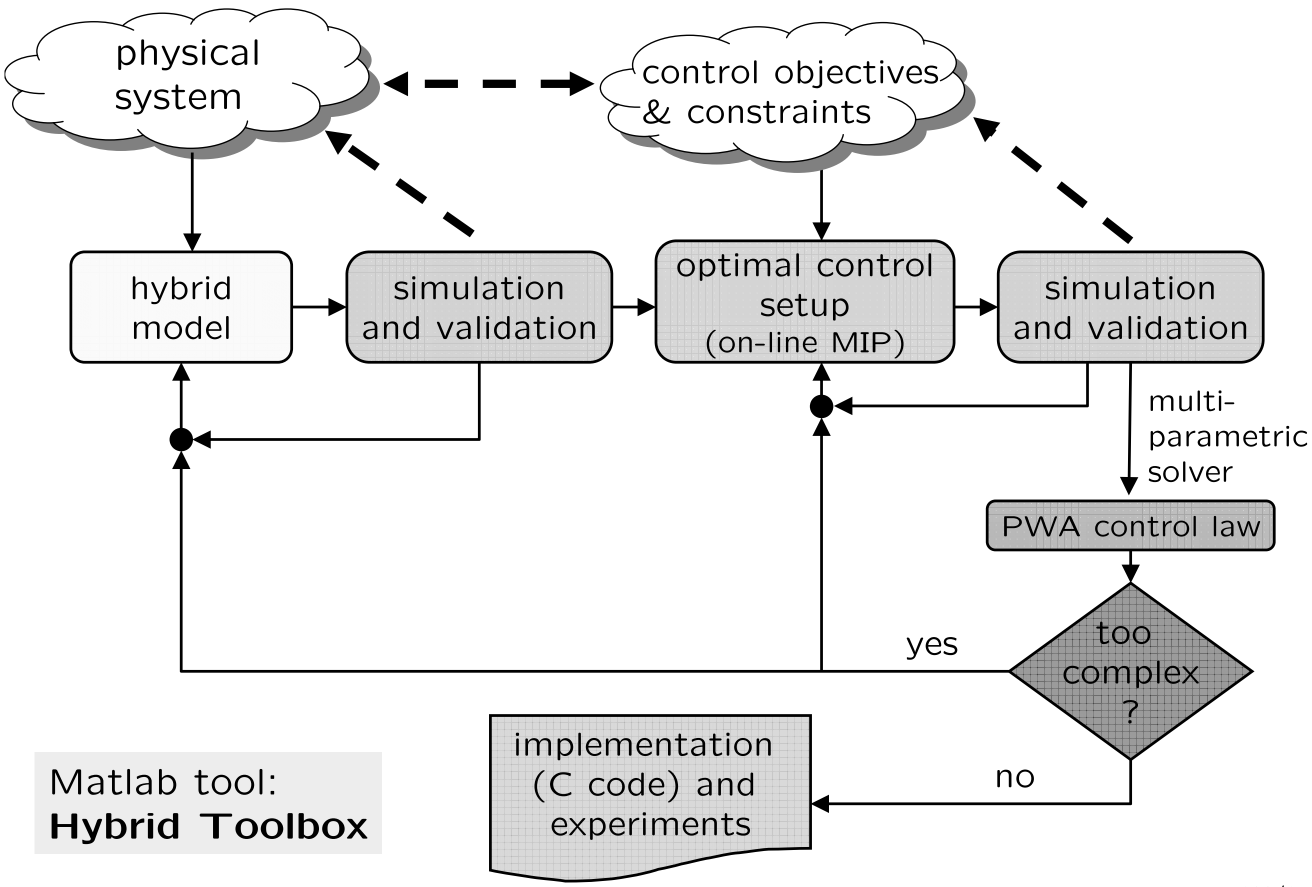
Implementation Aspects of Hybrid MPC

- **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



Matlab tool:
Hybrid Toolbox

A Few Hybrid MPC Tricks ...

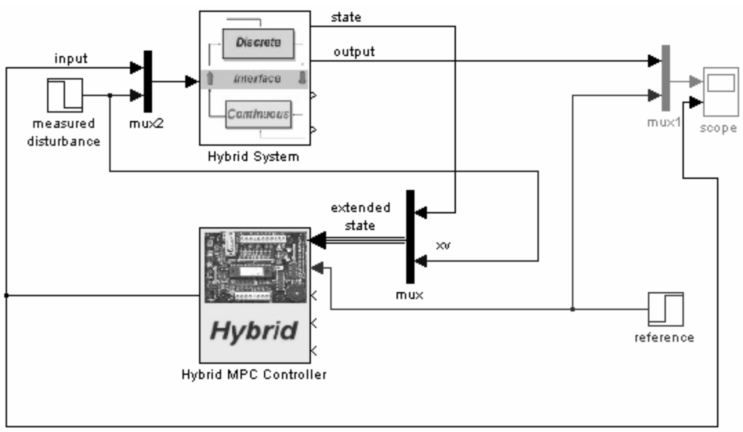
Measured Disturbances

- Disturbance $v(k)$ can be measured at time k
- Augment the hybrid prediction model with a constant state

$$x_v(k+1) = x_v(k)$$

- 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):

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

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

$$\text{subj. to} \quad u_{\text{min}} \leq u(k) \leq u_{\text{max}}, \ k = 0,\ldots,N-1$$

$$\Delta u_{\text{min}} \leq \Delta u(k) \leq \Delta u_{\text{max}}, \ k = 0,\ldots,N-1$$

$$y_{\text{min}} \leq y(k) \leq y_{\text{max}}, \ k = 1,\ldots,N$$

- Optimization problem: (MIQP)

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

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

Note: same trick as in linear MPC

Integral Action in Hybrid MPC

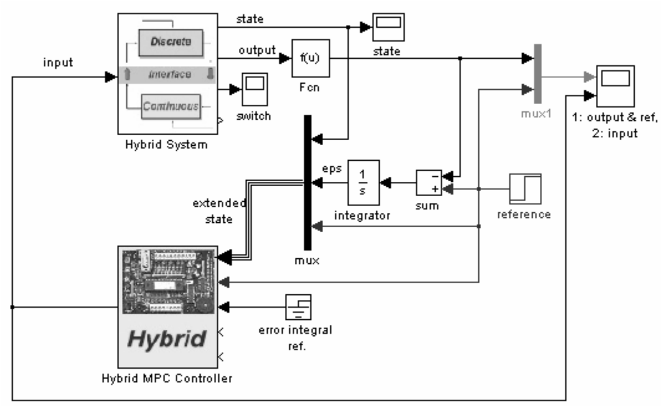
- Augment the hybrid prediction model with integrators of output errors as additional states:

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

$T_s =$ 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

Reference/Disturbance Preview

Measured disturbance $v(t)$ is known M steps in advance

$$\begin{cases} x_{v,M-1}(k+1) &= x_{v,M-2}(k) \\ x_{v,M-2}(k+1) &= x_{v,M-3}(k) \\ &\vdots \\ x_{v,1}(k+1) &= x_{v,0}(k) \\ x_{v,0}(k+1) &= x_{v,0}(k) \\ v(k) &= x_{v,M-1}(k),\ k=0,\dots,N-1 \end{cases}$$

initial condition

Note: same trick applies to linear MPC

$$\begin{cases} x_{v,M-1}(0) &= v(t) \\ x_{v,M-2}(0) &= v(t+1) \\ \vdots &= \vdots \\ x_{v,1}(0) &= v(t+M-2) \\ x_{v,0}(0) &= v(t+M-1) \end{cases}$$

produces v=$\{v(t),v(t+1),\dots,v(t+M\text{-}1),\ v(t+M\text{-}1),\dots\}$

Preview of reference $r(t)$: similar.

Delays – Method 1

- Hybrid model w/ delays:

$$\begin{aligned} x(t+1) &= Ax(t)+B_1u(t-\tau)+B_2\delta(t)+B_3z(t) \\ E_2\delta(t) &+ E_3z(t) \le E_1u(t-\tau)+E_4x(t)+E_5 \end{aligned}$$

- Map delays to poles in z=0:

$$x_k(t) \triangleq u(t-k) \quad \Rightarrow \quad x_k(t+1) = x_{k-1}(t) \qquad k=1,\dots,\tau$$

- Extend the state space of the MLD model:

$$\begin{bmatrix} x \\ x_\tau \\ x_{\tau-1} \\ \vdots \\ x_1 \end{bmatrix}(t+1) = \begin{bmatrix} A & B_1 & 0 & 0 & \dots & 0 \\ 0 & 0 & I_m & 0 & \dots & 0 \\ 0 & 0 & 0 & I_m & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} x \\ x_\tau \\ x_{\tau-1} \\ \vdots \\ x_1 \end{bmatrix}(t) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ I_m \end{bmatrix} u(t) + \begin{bmatrix} B_2 \\ 0 \\ 0 \\ \vdots \\ I_m \end{bmatrix} \delta(t) + \begin{bmatrix} B_3 \\ 0 \\ 0 \\ \vdots \\ I_m \end{bmatrix} z(t)$$

- Apply MPC to the extended MLD system

Note: same trick as in linear MPC

Delays – Method 2

- Delay-free MLD model:
$$\begin{aligned}\bar{x}(t+1) &= A\bar{x}(t) + B_1u(t) + B_2\bar{\delta}(t) + B_3\bar{z}(t)\\ E_2\bar{\delta}(t) &+ E_3\bar{z}(t) \leq E_1u(t) + E_4\bar{x}(t) + E_5\end{aligned}$$

  $\bar{x}(t) \triangleq x(t+\tau)$, $\bar{\delta}(t) \triangleq \delta(t+\tau)$, $\bar{z}(t) \triangleq z(t+\tau)$

- Design MPC for delay-free model: $u(t) = f_{\text{MPC}}(\bar{x}(t))$

- Compute the predicted state:
$$\bar{x}(t) = A^\tau x(t) + \sum_{j=0}^{\tau-1} A^{\tau-1-j}\left(B_1u(t-1-j) + B_2\bar{\delta}(t+j) + B_3\bar{z}(t+j)\right)$$
  where $\bar{\delta}(t+j), \bar{z}(t+j)$ are obtained from MLD ineq. (or HYSDEL model)

- Compute MPC action:
$$u(t) = f_{\text{MPC}}(\bar{x}(t))$$

For better closed-loop performance the model used for predicting the future hybrid state $x(t+\tau)$ may be more accurate than MLD model !

Prioritized Constraints

In optimization problems constraints may be classified in a hierarchy of priorities, where constraints at higher priority must be satisfied before those at lower priority.

Given the optimization problem:

$$\begin{array}{ll}\min_x & f(x)\\ \text{s.t.} & g(x) \leq 0\end{array} \qquad \begin{array}{l} f: \mathbb{R}^{d_1} \times \{0,1\}^{d_2} \to \mathbb{R}\\ g: \mathbb{R}^{d_1} \times \{0,1\}^{d_2} \to \mathbb{R}^c\end{array}$$

1. Define the set of indices of the constraints:
$$\mathcal{C} = \{1, 2, \ldots, c\}$$

2. There are r priority levels (i^{th} priority level > $(i+1)^{\text{th}}$ priority level)

   Let $\mathcal{P}_i \subseteq \mathcal{C}$ be the set of constraints at priority i.

   with $\mathcal{P}_i \cap \mathcal{P}_j = \emptyset, \quad i \neq j$ i.e. a constraint cannot be associated with more than one priority level.

Prioritized Constraints

3. Associate slack variables to each priority level:

$$g_k(x) \leq \epsilon_i, \quad \forall k \in \mathcal{P}_i \qquad \epsilon_i \geq 0, \quad \forall i = 1, \ldots, r$$

4. Associate binary variables to each priority level:

Each binary variable represents the possible violation of one or more constraints on the same level.

$$[\delta_i = 0] \rightarrow [\epsilon_i = 0] \quad \Longleftrightarrow \quad 0 \leq \epsilon_i \leq M_i \delta_i$$

and add:

$$[\delta_j = 0] \rightarrow [\delta_i = 0], \quad \forall j > i \quad \Longleftrightarrow \quad \delta_i \leq \delta_j, \quad i < j$$

i.e. if cons at lower priorities are not violated then cons at higher priorities cannot be violated

5. Add constraints in 3 and 4 to the optimization problem and minimize the following cost function:

$$\min f(x) + W'\delta$$

$W \in \mathbb{R}^r$ "suitably" large

Optimal Control Example

Consider the following optimal control problem:

$$\begin{array}{ll} \min & \sum_{k=0}^{N-1} ||y(k) - r_y||_\infty \\ \text{s.t.} & \text{MLD dynamics} \\ & x_1(k) \geq 1 \\ & x_2(k) \leq 2 \\ & u(k) \geq -1 \end{array}$$

with:

Higher priority ($x_1(k) \geq 1$, $x_2(k) \leq 2$)

Lower priority ($u(k) \geq -1$)

Problem can be reformulated as follows:

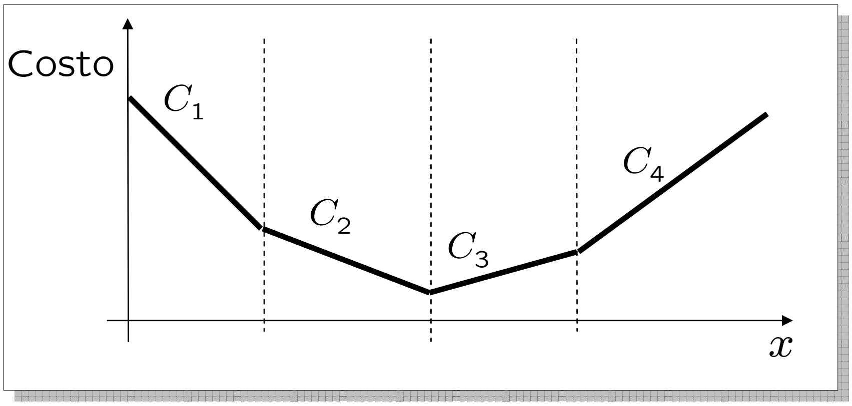
$$\begin{array}{ll} \min & \sum_{k=0}^{N-1} ||y(k) - r_y||_\infty + W_1\delta_1(k) + W_2\delta_2(k) \\ \text{s.t.} & \text{MLD dynamics} \\ & 1 - x_1(k) \leq \epsilon_1(k) \\ & x_2(k) - 2 \leq \epsilon_1(k) \\ & -u(k) - 1 \leq \epsilon_2(k) \\ & 0 \leq \epsilon_i(k) \leq M_i\delta_i(k), \quad i = 1, 2 \\ & \delta_1 \leq \delta_2 \end{array}$$

W_1,W_2 are suitable cost weights

```
…
IMPLEMENTATION{
AUX{
   REAL eps1[0, M1], eps2[0,M2];
   BOOL d1,d2;}
…
MUST{
   1-x1 <= eps1;
   x2-2 <= eps1;
   -u-1 <= eps2;
   eps1 >= 0; eps1 <= M1*d1;
   eps2 >=0;  eps2 <= M2*d2;
   d1 <= d2;}
```

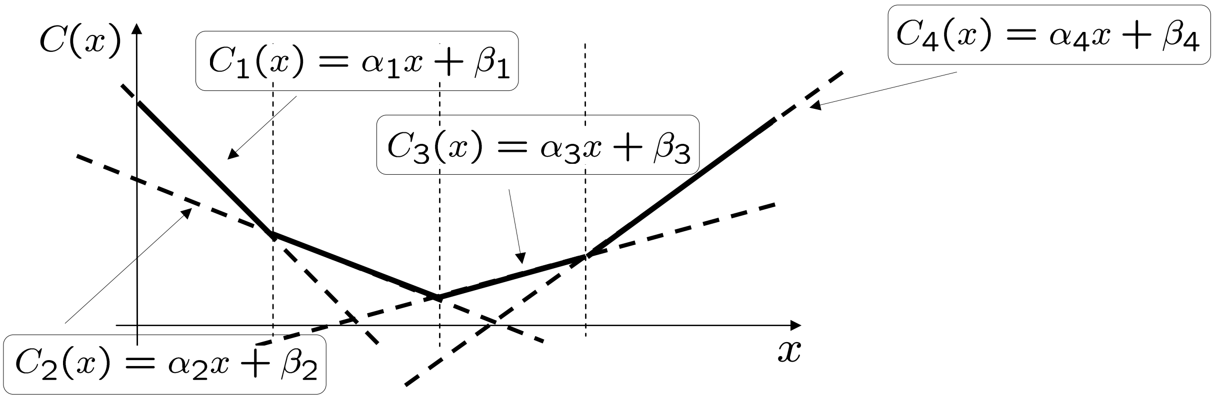
In HYSDEL

Piecewise Affine Cost Functions



Convex piecewise affine cost functions can be represented **without** introducing binary variables

Piecewise Affine Cost Functions

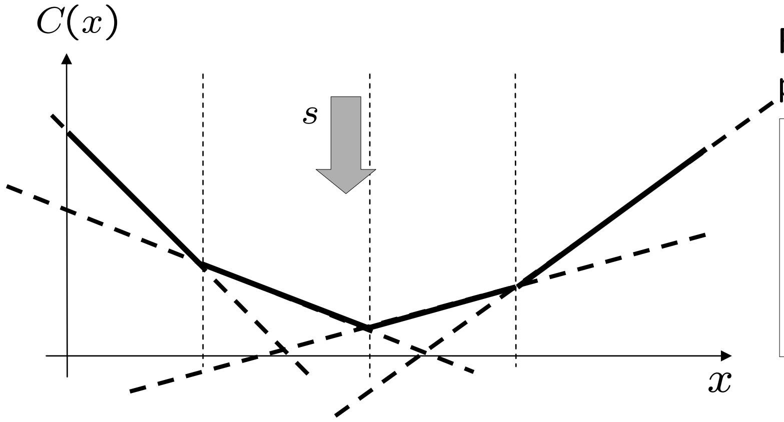


It is easy to see that:

$$C(x) = \max\{\alpha_1 x + \beta_1, \alpha_2 x + \beta_2, \alpha_3 x + \beta_3, \alpha_4 x + \beta_4\}$$

In general: Every convex piecewise affine function can be represented as the max of affine functions, and vice versa (Schechter, 1987)

Piecewise Affine Cost Functions



Reformulation as a linear program:

$$\min \; s \quad \text{subj.to} \; \begin{cases} s \geq \alpha_1 x + \beta_1 \\ s \geq \alpha_2 x + \beta_2 \\ s \geq \alpha_3 x + \beta_3 \\ s \geq \alpha_4 x + \beta_4 \end{cases}$$

Variabile s is an upper-bound on the max

$$s \geq \max\{\alpha_1 x + \beta_1, \alpha_2 x + \beta_2, \alpha_3 x + \beta_3, \alpha_4 x + \beta_4\}$$

It is easy to show (by contradiction) that at optimality we have:

$$s = \max\{\alpha_1 x + \beta_1, \alpha_2 x + \beta_2, \alpha_3 x + \beta_3, \alpha_4 x + \beta_4\}$$

Piecewise affine convex constraints can be dealt with similarly

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 art

(a unifying general theory does not exist)

Outline

✓ Basics of Model Predictive Control (MPC)

✓ Hybrid models for MPC

✓ MPC of hybrid systems

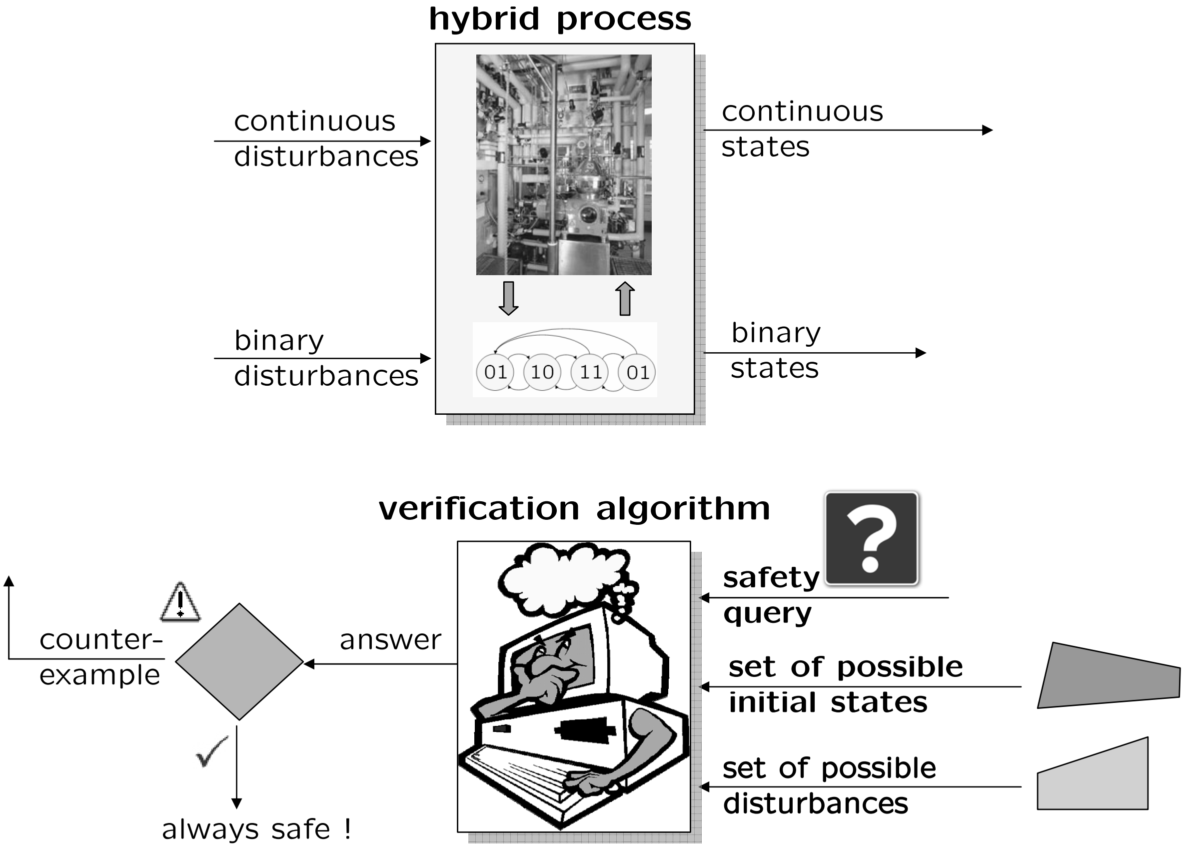
✓ Explicit MPC (multiparametric programming)

- Optimization-based reachability analysis
- Examples

Optimization-based Reachability Analysis

(Verification of Safety Properties via Mixed-integer Programming)

Hybrid Verification Problem



Verification Algorithm

- **QUERY**: Is the target set X_f reachable after N steps from some initial state $x_0 \in X_0$ for some input profile $u \in U$?

- Computation: Solve the mixed-integer linear program (MILP)

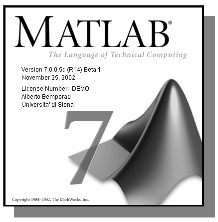
$$
\begin{array}{ll}
\min & 0 \\
\text{s.t.} & \left\{
\begin{array}{rcl}
x(k+1) & = & Ax(k) + B_1u(k) + B_2\delta(k) + B_3z(k) \\
E_2\delta(k) & + & E_3z(k) \leq E_1u(k) + E_4x(k) + E_5 \\
S_uu(k) & \leq & T_u \quad (u(k) \in U) \\
 & & k = 0, 1, \ldots, N-1 \\
 & & \\
S_0x(0) & \leq & T_0 \quad (x(0) \in X_0) \\
S_fx(N) & \leq & T_f \quad (x(N) \in X_f)
\end{array}
\right.
\end{array}
$$

with respect to $u(0),\delta(0),z(0),\ldots, u(N\text{-}1),\delta(N\text{-}1),z(N\text{-}1),x(0)$

- **Alternative solutions**:
  - Exploit the special structure of the problem and use polyhedral computation. (Torrisi, 2003)
  - Use abstractions (LPs) + SAT solvers (Giorgetti, Pappas, Bemporad, 2005)

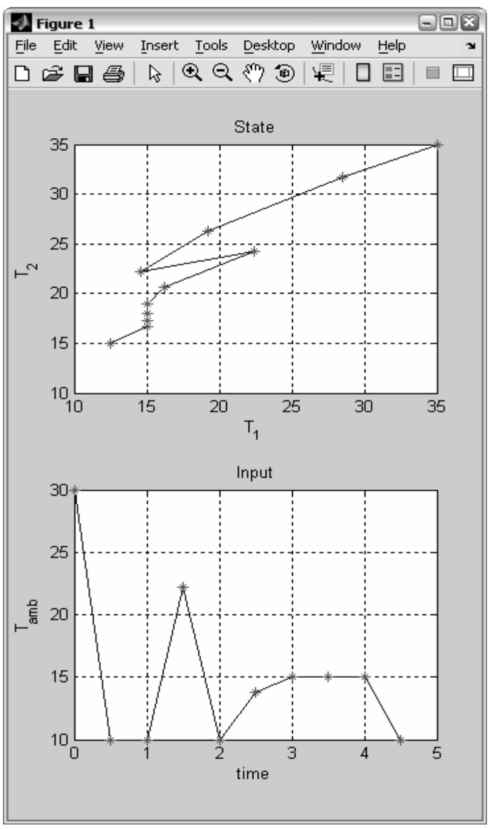
Verification Example 1

- MLD model: room temperature system
- $X_f = \{ \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} : 10 \leq T_1, T_2 \leq 15\}$ (set of unsafe states)
- $X_0 = \{ \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} : 35 \leq T_1, T_2 \leq 40\}$ (set of initial states)
- $U = \{T_{\text{amb}} : 10 \leq T_{\text{amb}} \leq 30\}$ (set of possible inputs)
- N=10 (time horizon)

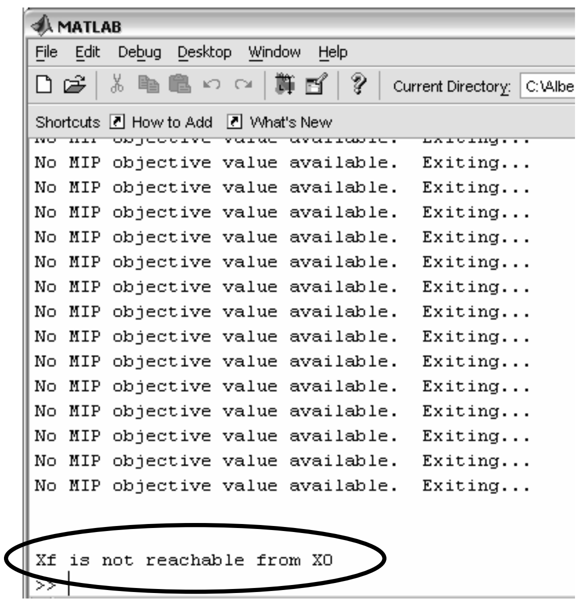



```
>>[flag,x0,U]=reach(S,N,Xf,X0,umin,umax);
```


Verification Example 1



$U = \{T_{\text{amb}} : 10 \leq T_{\text{amb}} \leq 30\}$



$U = \{T_{\text{amb}} : 20 \leq T_{\text{amb}} \leq 30\}$

Verification Algorithm (2)

- **QUERY**: Is the target set X_f reachable **within** N steps from some initial state $x_0 \in X_0$ for some input profile $u \in U$?

- IDEA: Augment the MLD system to register the entrance of the target (unsafe) set $X_f = \{x\colon A_f x \leq b_f\}$:

  - Add a new variable $\delta_f(k)$, where $[\delta_f(k)=1] \rightarrow [x(k+1) \in X_f]$

    big-M $\Rightarrow$ $A_f(Ax(k)+B_1u(k)+B_2\delta(k)+B_3z(k)) \leq b_f+M(1-\delta_f(k))$

  - Add the constraint $\sum_{k=0}^{N-1} \delta_f(k) \geq 1$ ($x \in X_f$ for at least one k)

  - Solve MILP feasibility test

Verification Example 3

- States/inputs: $x_1, x_2, x_3 \in \mathbb{R},\ x_4, x_5 \in \{0,1\}$

  $u_1, u_2 \in \mathbb{R},\ u_3 \in \{0,1\}$

- Events:

$$[\delta_1 = 1] \leftrightarrow [x_1 \leq 0]$$
$$[\delta_2 = 1] \leftrightarrow [x_2 \geq 1]$$
$$[\delta_3 = 1] \leftrightarrow [x_3 - x_2 \leq 1]$$

- Switched dynamics:

$$x_1(k+1) = \begin{cases} 0.1x_1 + 0.5x_2 & \text{if } \delta_1 \& \delta_2 | x_4 \\ -0.3x_3 - x_1 + u_1 & \text{otherwise} \end{cases}$$
$$x_2(k+1) = \begin{cases} -0.8x_1 + 0.7x_3 - u_1 - u_2 & \text{if } \delta_3 | x_5 \\ -0.3x_1 - 2x_2 & \text{otherwise} \end{cases}$$
$$x_3(k+1) = \begin{cases} -0.1x_3 + u_2 & \text{if } \delta_3 \& x_5 | x_4 \& \delta_1 \\ x_3 - 0.5x_1 - 2u_1 & \text{otherwise} \end{cases}$$

- Automaton:

$$x_4(k+1) = x_4 \& \delta_1$$
$$x_5(k+1) = ((x_4 | x_5) \& (\delta_1 | \delta_2)) | \delta_3 \& u_3$$

Query: Starting from X_0, is it possible that $x_i(k) \in X_f$ at some $k \leq N$, under the restriction that $x_3(k)+x_2(k) \leq 0$, $\delta_1(k) | \delta_2(k) | x_5(k)$ is true, and $\sim x_4(k) | x_5(k)$ is also true $\forall k \leq N$?

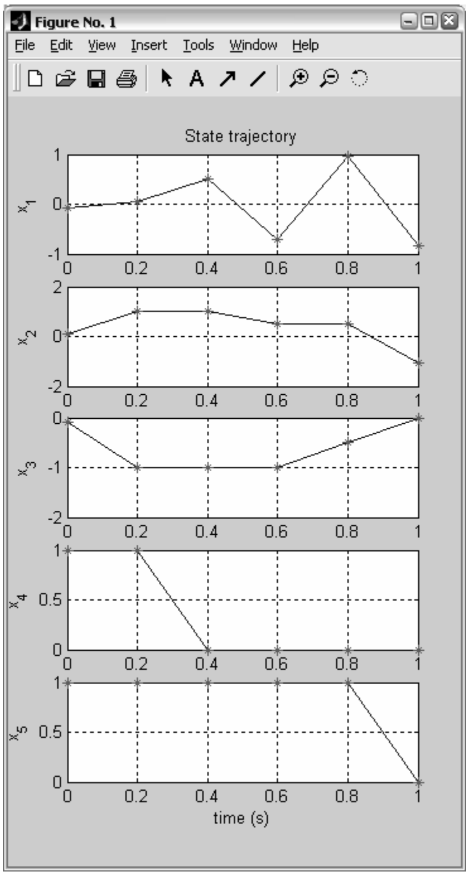
go to demo **/demos/hybrid/reachtest.m**

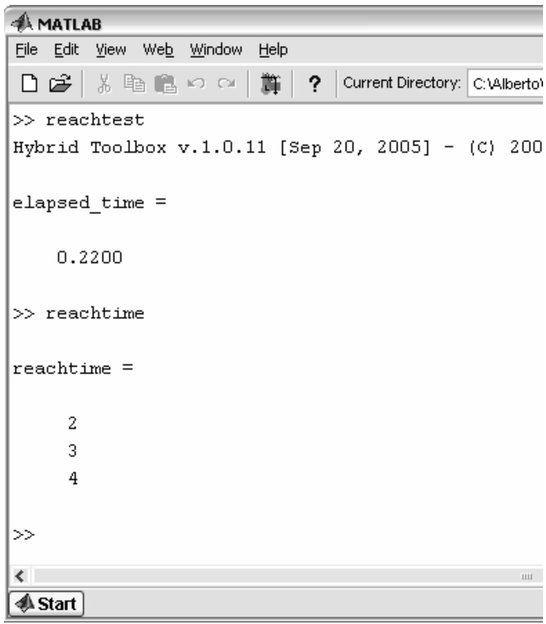
Verification Example 3

- $X_f = \{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} : -1 \leq x_1, x_3 \leq 1,\ 0.5 \leq x_2 \leq 1,\ x_4, x_5 \in \{0,1\}\}$ (set of unsafe states)
- $X_0 = \{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} : -0.1 \leq x_1, x_3 \leq 0.1,\ x_2 = 0.1,\ x_4, x_5 \in \{0,1\}\}$ (set of initial states)
- $U = \{\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} : -1 \leq u_1 \leq 1,\ -2 \leq u_2 \leq 2, u_3 \in \{0,1\}\}$ (set of possible inputs)
- N=5 (time horizon)

```
>>[flag,x0,U,xf,X,T,D,Z,Y,reachtime]=reach(S,[1 N],Xf,X0);
```


Verification Example 3

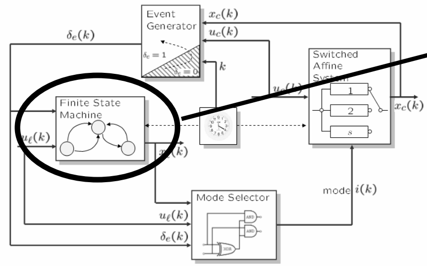




The set X_f is reached by $x(k)$ at times k=2,3,4

DHA - Extensions

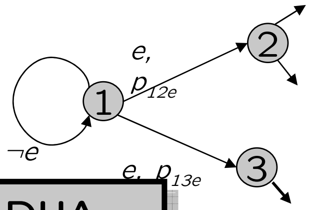
Discrete-time Hybrid Stochastic Automaton (DHSA)



Stochastic Finite State Machine (sFSM)

$$P[x_b(k+1)=1] = f_{\mathsf{sFSM}}(x_b(k), u_b(k), \delta_e(k))$$

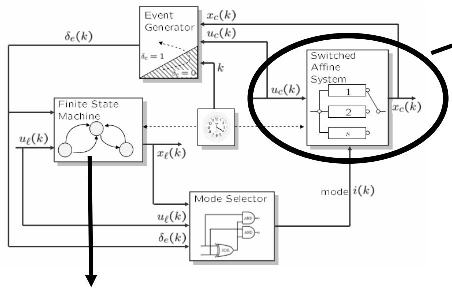
k= discrete-time counter



(Bemporad, Di Cairano, IEEE TAC, submitted)

All control/verification techniques developed for DHA can be extended to DHSA and icHA !

Event-based Continuous-time Hybrid Automaton (icHA)



Asynchronous FSM

Switched integral dynamics

$$\frac{dx_c(t)}{dt} = B_{i(t)}u_c(t) + f_{i(t)}$$

k= **event counter**

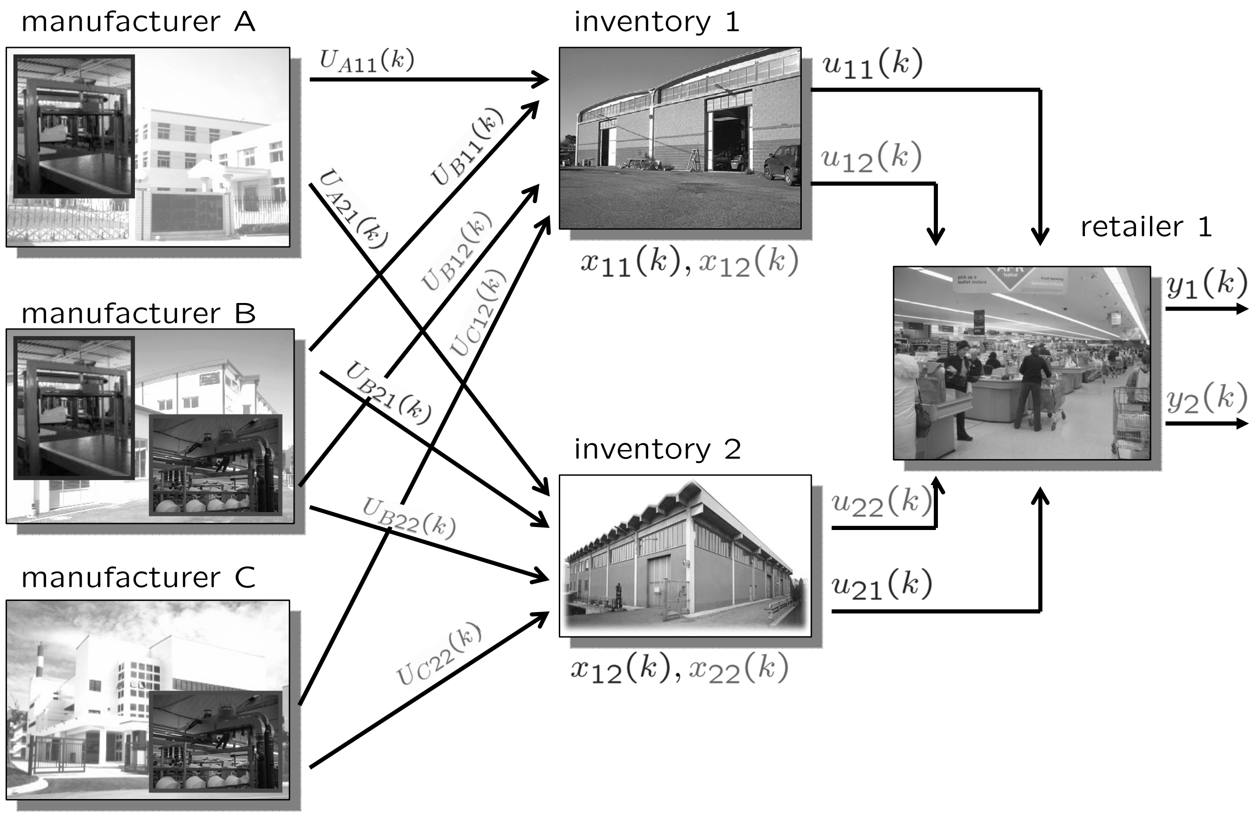
u_c constant between events k and k+1

(Bemporad, Di Cairano, Julvez, CDC05 &HSCC-06)

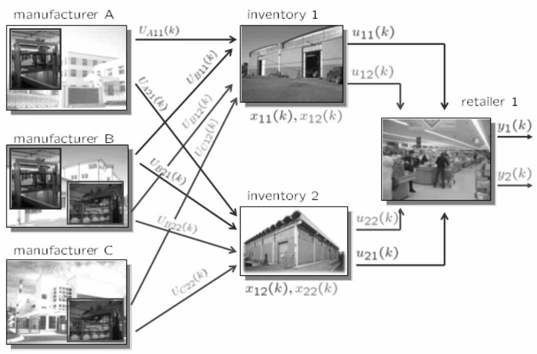
Outline

- ✓ Basics of Model Predictive Control (MPC)
- ✓ Hybrid models for MPC
- ✓ MPC of hybrid systems
- ✓ Explicit MPC (multiparametric programming)
- ✓ Optimization-based reachability analysis
- Examples

A Simple Example in Supply Chain Management



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 (i=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>

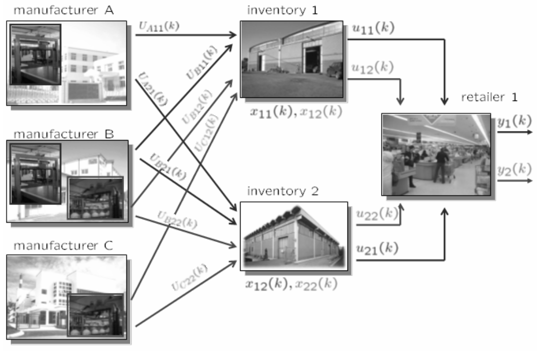
$U_{Xij}(k)$ = 1 if manufacturer X produces and send j to inventory i at time k

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

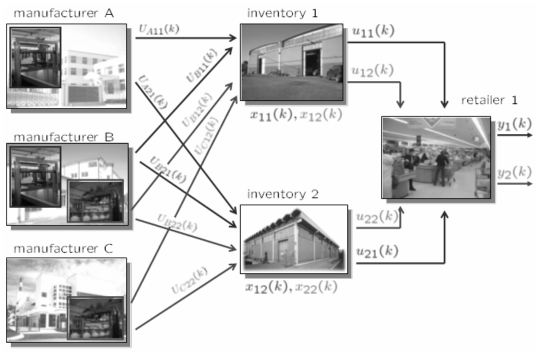
- A manufacturer can only produce one type of product at one time:

  [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

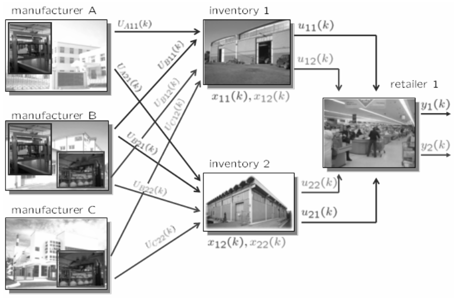


- Level of inventories:

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

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}; }
```



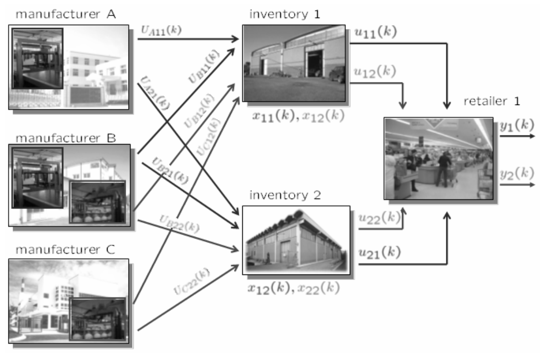

```
        CONTINUOUS {x11 = x11 + zA11 + zB11 - u11;
                    x12 = x12 + zB12 + zC12 - u12;
                    x21 = x21 + zA21 + zB21 - u21;
                    x22 = x22 + zB22 + zC22 - u22;   }

        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

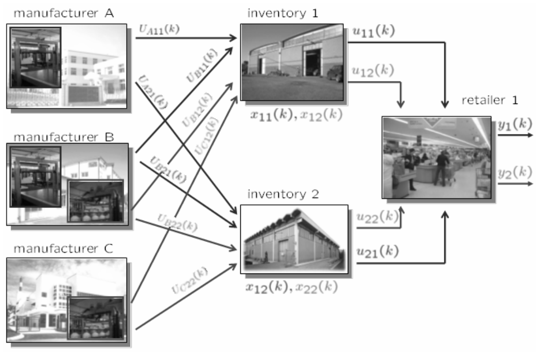
Performance Specs

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

- 10: penalty on demand tracking error
- 4: cost for shipping from inv.#1 to market
- 2: cost for shipping from inv.#2 to market
- 1: cost from A to inventories
- 4: cost from B to inventories
- 10: cost from C to inventories

Hybrid MPC - Example

```
>>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;
```



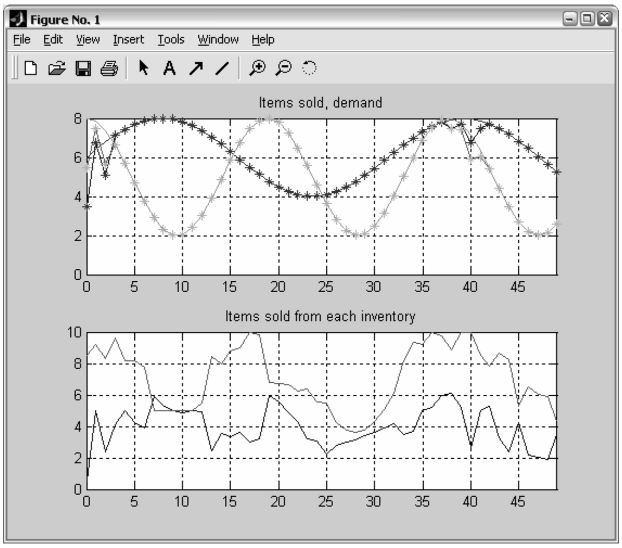

```
>>C=hybcon(S,Q,N,limits,refs);
```

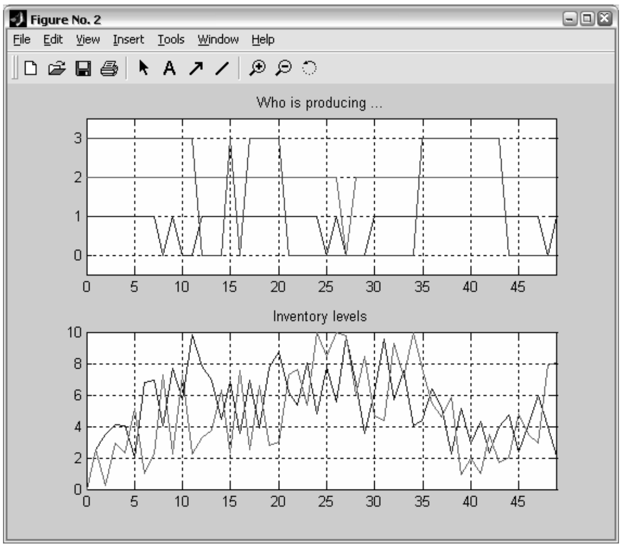
```
>> C

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.

>>
```

Hybrid MPC - Example

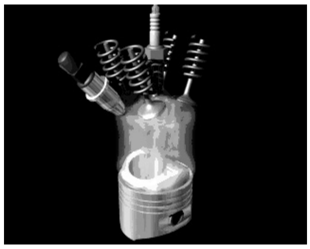
```
>>x0=[0;0;0;0];                         % Initial condition
>>r.y=[6+2*sin((0:Tstop-1)'/5)          % Reference trajectories
       5+3*cos((0:Tstop-1)'/3)];
```

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





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



Hybrid Control of a DISC Engine

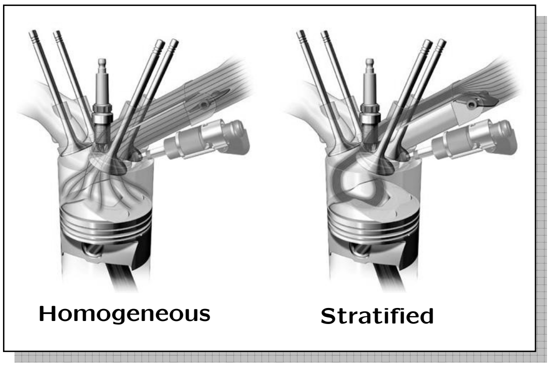
(Giorgetti, Ripaccioli, Bemporad, Kolmanovsky, Hrovat, IEEE Tr. Mechatronics, 2006)

FORD RESEARCH LABORATORY

DISC Engine

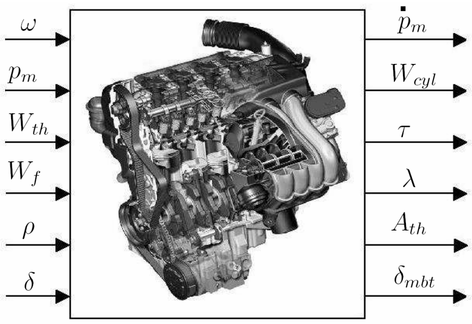
Two distinct regimes:

Regime	fuel injection	air-to-fuel ratio
Homogeneous combustion	intake stroke	$\lambda = 14.64$
Stratified combustion	compression stroke	$\lambda > 14.64$



Pro: reduce consumption up to 15%;
Con: complex treatment of exhaust gas

- **States**: intake manifold pressure (p_m)
- **Outputs:** Air-to-fuel ratio (λ), torque (τ), max-brake-torque spark timing (δ_{mbt})
- **Inputs:** spark advance (δ), air flow (W_{th}), fuel flow (W_f), combustion regime (ρ);
- **Disturbance:** engine speed (ω) [measured]



DISC Engine – Control Objective

Objective: Design a controller for the engine that

- Automatically chooses operating **mode** (homogeneous/stratified)
- Can cope with **nonlinear** dynamics
- Handles **constraints** (on A/F ratio, air-flow, spark)
- Achieves **optimal** performance (tracking of desired torque and A/F ratio)

DISC Engine - HYSDEL List

```
SYSTEM hysdisc{
  INTERFACE{
  STATE{
    REAL pm       [1, 101.325];
    REAL xtau     [-1e3, 1e3];
    REAL xlam     [-1e3, 1e3];
    REAL taud     [0,    100];
    REAL lamd     [10,    60];
     }
  OUTPUT{
    REAL lambda, tau, ddelta;
         }
  INPUT{
    REAL Wth      [0,38.5218];
    REAL Wf       [0,      2];
    REAL delta    [0,     40];
    BOOL rho;
     }
  PARAMETER{
    REAL Ts, pm1, pm2;
    …
     }
  }

  IMPLEMENTATION{
  AUX{
    REAL lam,taul,dmbtl,lmin,lmax;
     }
  DA{
  lam={IF rho THEN l11*pm+l12*Wth...
              +l13*Wf+l14*delta+l1c
       ELSE   l01*pm+l02*Wth+l03*Wf...
              +l04*delta+l0c      };

    taul={IF rho THEN tau11*pm+...
      tau12*Wth+tau13*Wf+tau14*delta+tau1c
          ELSE   tau01*pm+tau02*Wth...
           +tau03*Wf+tau04*delta+tau0c };

    dmbtl ={IF rho THEN dmbt11*pm+dmbt12*Wth...
            +dmbt13*Wf+dmbt14*delta+dmbt1c+7
            ELSE dmbt01*pm+dmbt02*Wth...
            +dmbt03*Wf+dmbt04*delta+dmbt0c-1};

    lmin ={IF rho THEN 13 ELSE 19};
    lmax ={IF rho THEN 21 ELSE 38};
         }
    CONTINUOUS{
        pm=pm1*pm+pm2*Wth;
        xtau=xtau+Ts*(taud-taul);
        xlam=xlam+Ts*(lamd-lam);
        taud=taud; lamd=lamd;
            }
    OUTPUT{
       lambda=lam-lamd;
       tau=taul-taud;
       ddelta=dmbtl-delta;
       }
    MUST{
       lmin-lam     <=0;
       lam-lmax     <=0;
       delta-dmbtl  <=0;
       }
    }
  }
```


MPC of DISC Engine

$$\min \sum_{k=0}^{N-1} (u_k - u_r)'R(u_k - u_r) + (y_k - y_r)'Q(y_k - y_r) + (x_{k+1} - x_r)'S(x_{k+1} - x_r)$$

$$\text{subj. to } \begin{cases} x_0 = x(t), \\ \text{hybrid model} \end{cases}$$

Solve **MIQP problem** (mixed-integer quadratic program) to compute $u(t)$

$u(t) = [W_{th}(t),\ W_f(t),\ \delta(t),\ \rho(t)]$

<u>Weights:</u>

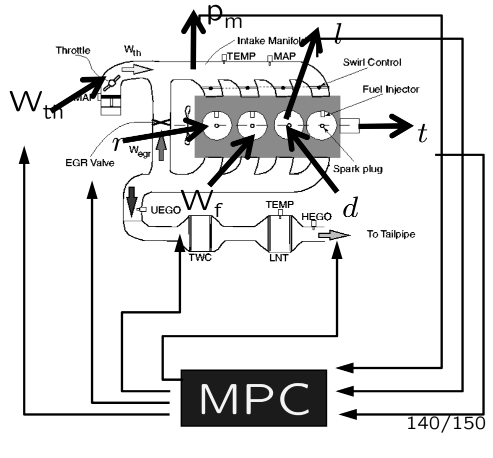
$$R = \begin{pmatrix} 0.01 & 0 & 0 & 0 \\ 0 & 0.001 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

r_ρ (prevents unneeded chattering)

$$Q = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0.001 & 0 \\ 0 & 0 & 0.01 \end{pmatrix} \quad S = \begin{pmatrix} 0.04 & 0 & 0 \\ 0 & 1500 & 0 \\ 0 & 0 & 0.01 \end{pmatrix}$$

q_τ q_λ s_{ε_τ} s_{ε_λ}

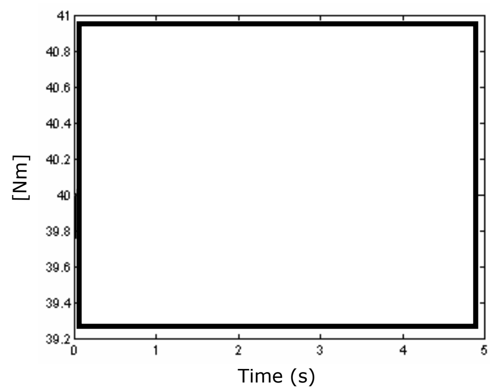
main emphasis on torque



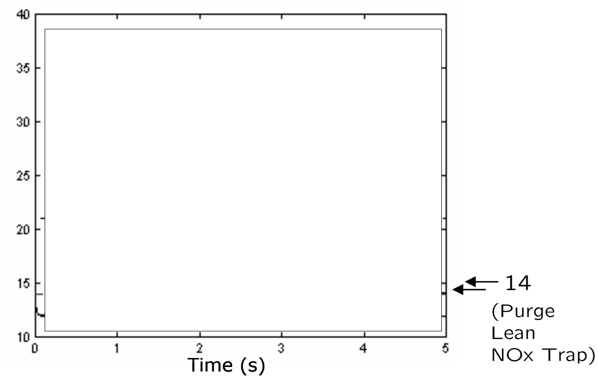
Simulation Results (nominal engine speed)

$\omega = 2000$ rpm

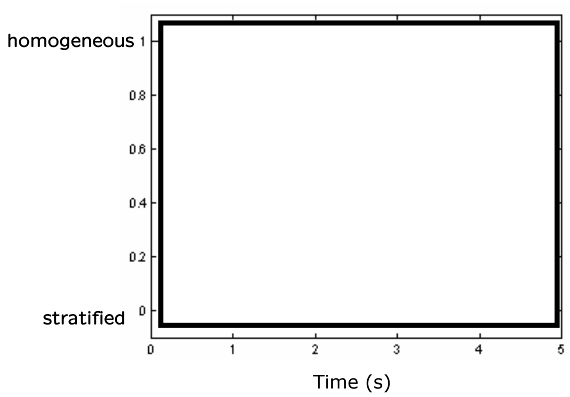
Engine Brake Torque



Air-to-Fuel Ratio



Combustion mode

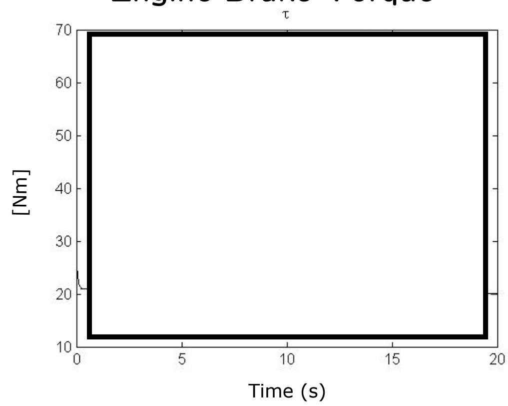


- Control horizon **N=1**;
- Sampling time **Ts=10 ms**;
- PC Xeon 2.8 GHz + Cplex 9.1

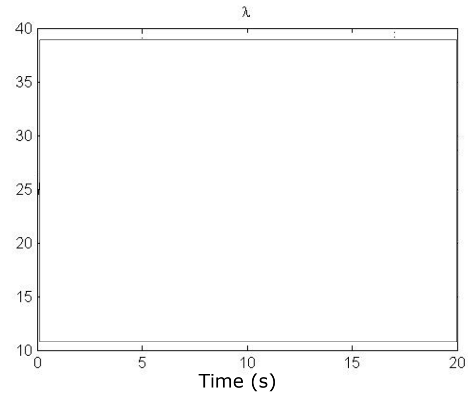
$\simeq$3 ms per time step

Simulation Results (varying engine speed)

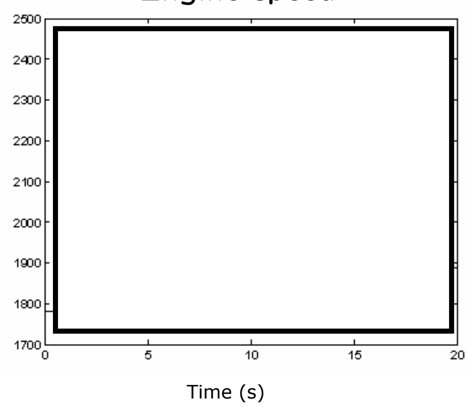
Engine Brake Torque



Air-to-Fuel Ratio



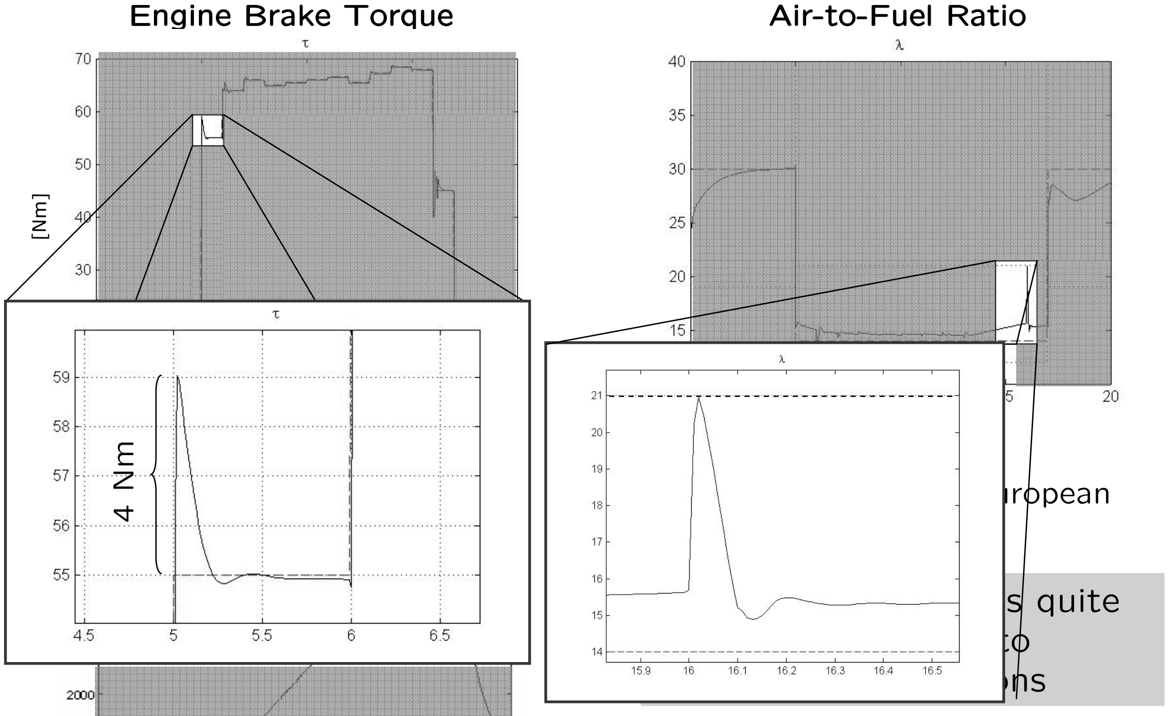
Engine speed

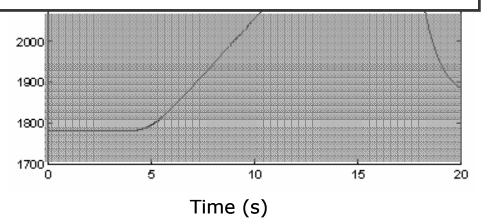


20 s segment of the European drive cycle (NEDC)

Hybrid MPC design is quite robust with respect to engine speed variations

Simulation Results (varying engine speed)





Control code too complex (MILP) ⇒ not implementable !

Explicit MPC Controller

Explicit control law: $u(t) = f(\theta(t))$

where: $u = [W_{th}\ W_f\ \delta\ \rho]'$

$$\theta = [p_m\ \epsilon_\tau\ \epsilon_\lambda\ \tau_{\text{ref}}\ \lambda_{\text{ref}}\ p_{m,\text{ref}}\ W_{th,\text{ref}}\ W_{f,\text{ref}}\ \delta_{\text{ref}}]'$$

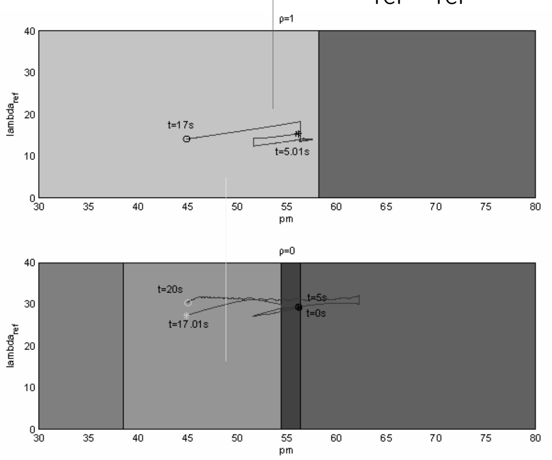
N=1 (control horizon)

42 partitions

- Time to compute explicit MPC: ≈ 3s;
- Sampling time Ts=10 ms;
- PC Xeon 2.8 GHz + Cplex 9.1

≃**8 µs** per time step

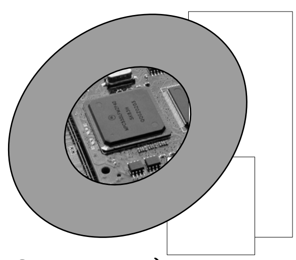
Cross-section by the τ_{ref}-λ_{ref} plane



ρ=0

ρ=1

≈ 3ms on µcontroller Motorola MPC 555 43kb RAM (custom made for Ford)

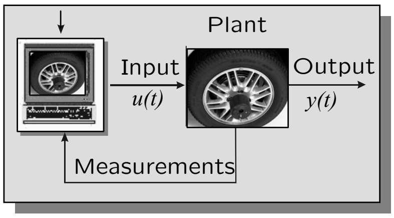


Conclusions

- **Hybrid systems** as a framework for new applications, where both logic and continuous dynamics are relevant

- **Supervisory MPC controllers** schemes can be synthesized via on-line mixed-integer programming (MILP/MIQP)

- **Piecewise Linear MPC Controllers** can be synthesized off-line via multiparametric programming for fast-sampling applications



Hybrid modeling and MPC design

Multiparametric programming

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

C-code download & testing

Conclusions

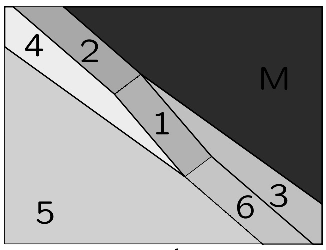
- **Hybrid systems** as a framework for new applications, where both logic and continuous dynamics are relevant

- **Supervisory MPC controllers** schemes can be synthesized via on-line mixed-integer programming (MILP/MIQP)

- **Piecewise Linear MPC Controllers** can be synthesized off-line via multiparametric programming for fast-sampling applications

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

(x,r)-space



- **Matlab tools** available to assist the whole design process (models, simulation, MPC design, code generation):
  ⇨ MPC Toolbox (linear), Hybrid Toolbox (hybrid, explicit), Multi-Parametric Toolbox (PWA, explicit)

References

General References

[1] J.M. Maciejowski, *Predictive Control with Constraints*, Prentice Hall, Harlow, UK, 2002.

[2] E.F. Camacho and C. Bordons, *Model Predictive Control*, Advanced Textbooks in Control and Signal Processing. Springer-Verlag, London, 2nd edition, 2004.

[3] A. Bemporad, M. Morari, and N. L. Ricker, *Model Predictive Control Toolbox for Matlab – User's Guide*, The Mathworks, Inc., 2004, http://www.mathworks.com/access/helpdesk/help/toolbox/mpc/.

[4] A. Bemporad, *Hybrid Toolbox – User's Guide*, Jan. 2004, http://www.dii.unisi.it/hybrid/toolbox.

[5] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789-814, June 2000.

[6] J.B. Rawlings, "Tutorial overview of model predictive control," *IEEE Control Systems Magazine*, pp. 38-52, June 2000.

[7] A. Bemporad, "Model-based predictive control design: New trends and tools," in *Proc. 45th IEEE Conf. on Decision and Control*, San Diego, CA, 2006.

References

MLD and HYSDEL Modeling

[1] A. Bemporad, "Hybrid Toolbox – User's Guide,'' Dec. 2003, http://www.dii.unisi.it/hybrid/toolbox

[2] F.D. Torrisi and A. Bemporad, "HYSDEL - A tool for generating computational hybrid models," *IEEE Transactions on Control Systems Technology*, vol. 12, no. 2, pp. 235-249, Mar. 2004

[3] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407-427, Mar. 1999.

[4] A. Bemporad, "Efficient conversion of mixed logical dynamical systems into an equivalent piecewise affine form," *IEEE Trans. Automatic Control*, vol. 49, no. 5, pp. 832-838, 2004.

[5] A. Bemporad, G. Ferrari-Trecate, and M. Morari, "Observability and controllability of piecewise affine and hybrid systems," *IEEE Trans. Automatic Control*, vol. 45, no. 10, pp. 1864-1876, 2000.

[6] W.P.H.M Heemels, B. de Schutter, and A. Bemporad, "Equivalence of hybrid dynamical models," *Automatica*, vol. 37, no. 7, pp. 1085-1091, July 2001

[7] A. Bemporad, W.P.M.H. Heemels, and B. De Schutter, "On hybrid systems and closed-loop MPC systems," *IEEE Trans. Automatic Control*, vol. 47, no. 5, pp. 863-869, May 2002.

References

Identification

[8] A. Bemporad, A. Garulli, S. Paoletti, and A. Vicino, “A bounded-error approach to piecewise affine system identification,” *IEEE Trans. Automatic Control*, vol. 50, no. 10, pp. 1567-1580, Oct. 2005 .

[9] J. Roll, A. Bemporad, and L. Ljung, “Identification of piecewise affine systems via mixed-integer programming,” *Automatica*, vol. 40, no. 1, pp. 37-50, 2004

[10] G. Ferrari-Trecate, M. Muselli, D. Liberati, and M. Morari, “A clustering technique for the identification of piecewise affine systems,” *Automatica*, vol. 39, no. 2, pp. 205-217, Feb. 2003.

Model predictive control

[11] F. Borrelli, M. Baotic, A. Bemporad, and M. Morari, “Dynamic programming for constrained optimal control of discrete-time linear hybrid systems,” *Automatica*, vol. 41, no. 10, Oct. 2005

[12] A. Bemporad, M. Morari, V. Dua, and E.N. Pistikopoulos, “The explicit linear quadratic regulator for constrained systems,” *Automatica*, vol. 38, no. 1, pp. 3-20, 2002.

[13] A. Bemporad, F. Borrelli, and M. Morari, “Piecewise linear optimal controllers for hybrid systems,” In *Proc. American Control Conference*, 2000, pp. 1190-1194.

[14] A. Bemporad and N. Giorgetti, “Logic-based methods for optimal control of hybrid systems,” *IEEE Trans. Autom. Control*, vol. 51, no. 6, pp. 963–976, 2006.

[15] M. Lazar, M. Heemels, S. Weiland, A. Bemporad, “Stability of Hybrid Model Predictive Control,” *IEEE Trans. Automatic Control*, vol. 51, no. 11, pp. 1813–1818, 2006.

References

Reachability and observability

[16] A. Bemporad, G. Ferrari-Trecate, and M. Morari, “Observability and controllability of piecewise affine and hybrid systems,” *IEEE TAC*, vol. 45, no. 10, pp. 1864-1876, 2000.

[17] A. Bemporad, D. Mignone, and M. Morari, “Moving horizon estimation for hybrid systems and fault detection,” in *Proc. American Control Conf.*, 1999, Chicago, IL, pp. 2471-2475.

[18] G. Ferrari-Trecate, D. Mignone, and M. Morari, “Moving horizon estimation for hybrid systems,” *IEEE TAC*, vol. 47, no. 10, pp. 1663-1676, 2002.

Applications

[19] F. Borrelli, A. Bemporad, M. Fodor, and D. Hrovat, “An MPC/Hybrid System Approach to Traction Control,” *IEEE Control Syst. Tech.*, vol. 14, n. 3, pp. 541-552, 2006.

[20] N. Giorgetti, G. Ripaccioli, A. Bemporad, I.V. Kolmanovsky, and D. Hrovat, “Hybrid model predictive control of direct injection stratified charge engines,” *IEEE/ASME Transactions on Mechatronics*, vol. 11, no. 5, pp. 499–506, Aug. 2006.

[21] N. Giorgetti, A. Bemporad, H. E. Tseng, and D. Hrovat, “Hybrid model predictive control application towards optimal semi-active suspension,” *International Journal of Control*, vol. 79, no. 5, pp. 521–533, 2006.

[22] A. Bemporad, P. Borodani, and M. Mannelli, “Hybrid control of an automotive robotized gearbox for reduction of consumptions and emissions,” LNCS 2623, pp. 81-96, Springer-Verlag, 2003.

more on **`http://www.dii.unisi.it/~bemporad/publications`**