Optimization-based Control of Hybrid Dynamical Systems

Alberto Bemporad

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

University of Siena
(founded in 1240)

Department of
Information Engineering

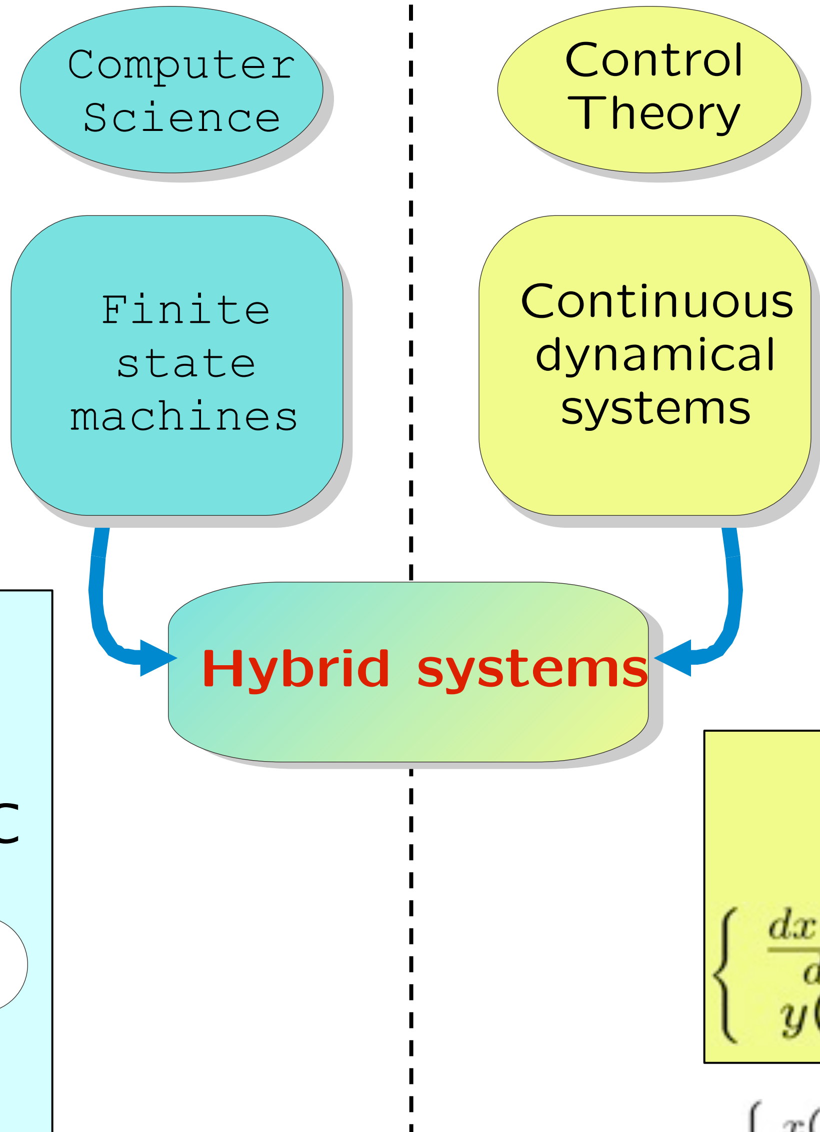
CONTROLO 2006
7TH PORTUGUESE CONFERENCE ON AUTOMATIC CONTROL
LISBON • IST • CONGRESS CENTER

Lisbon, September 11, 2006

Contents

- Models of hybrid systems
- Model predictive control of hybrid systems
- Explicit reformulation
- Automotive applications

Hybrid Systems



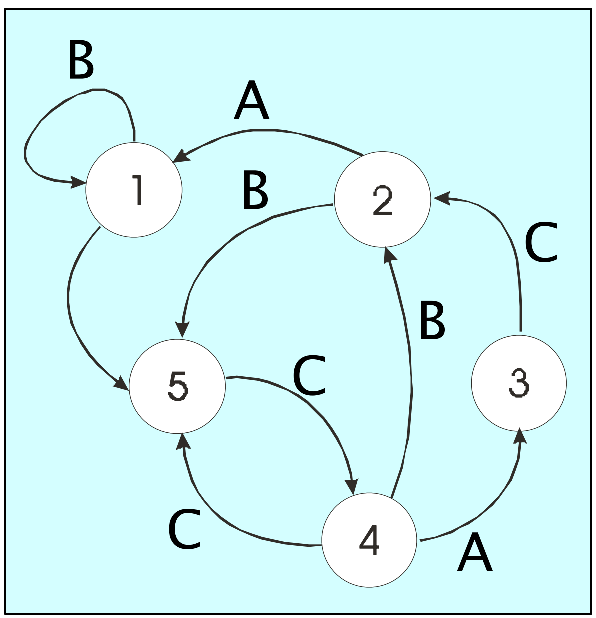
$\mathbf{x} \in \{1, 2, 3, 4, 5\}$

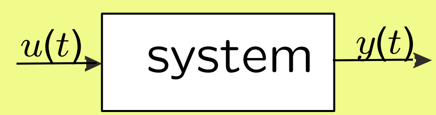
$\mathbf{u} \in \{A, B, C\}$

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

$u \in \mathbb{R}^m$

$y \in \mathbb{R}^p$





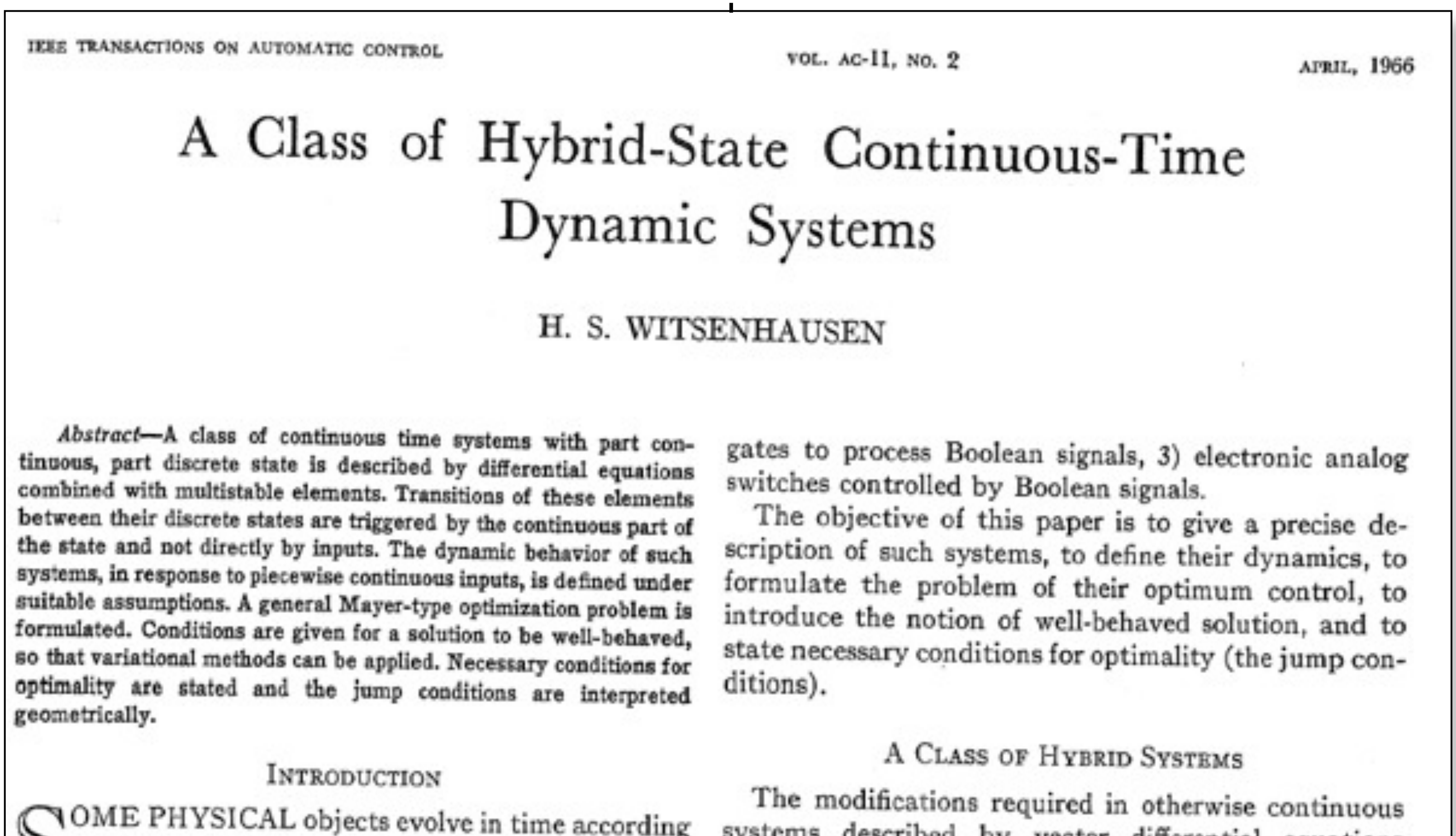
$$\begin{cases} \frac{dx(t)}{dt} & = & f(x(t), u(t)) \\ y(t) & = & g(x(t), u(t)) \end{cases}$$

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

Hybrid Systems

Computer Science

Control Theory


IEEE TRANSACTIONS ON AUTOMATIC CONTROL VOL. AC-11, NO. 2 APRIL, 1966

A Class of Hybrid-State Continuous-Time Dynamic Systems

H. S. WITSENHAUSEN

Abstract—A class of continuous time systems with part continuous, part discrete state is described by differential equations combined with multistable elements. Transitions of these elements between their discrete states are triggered by the continuous part of the state and not directly by inputs. The dynamic behavior of such systems, in response to piecewise continuous inputs, is defined under suitable assumptions. A general Mayer-type optimization problem is formulated. Conditions are given for a solution to be well-behaved, so that variational methods can be applied. Necessary conditions for optimality are stated and the jump conditions are interpreted geometrically.

Introduction

SOME PHYSICAL objects evolve in time according

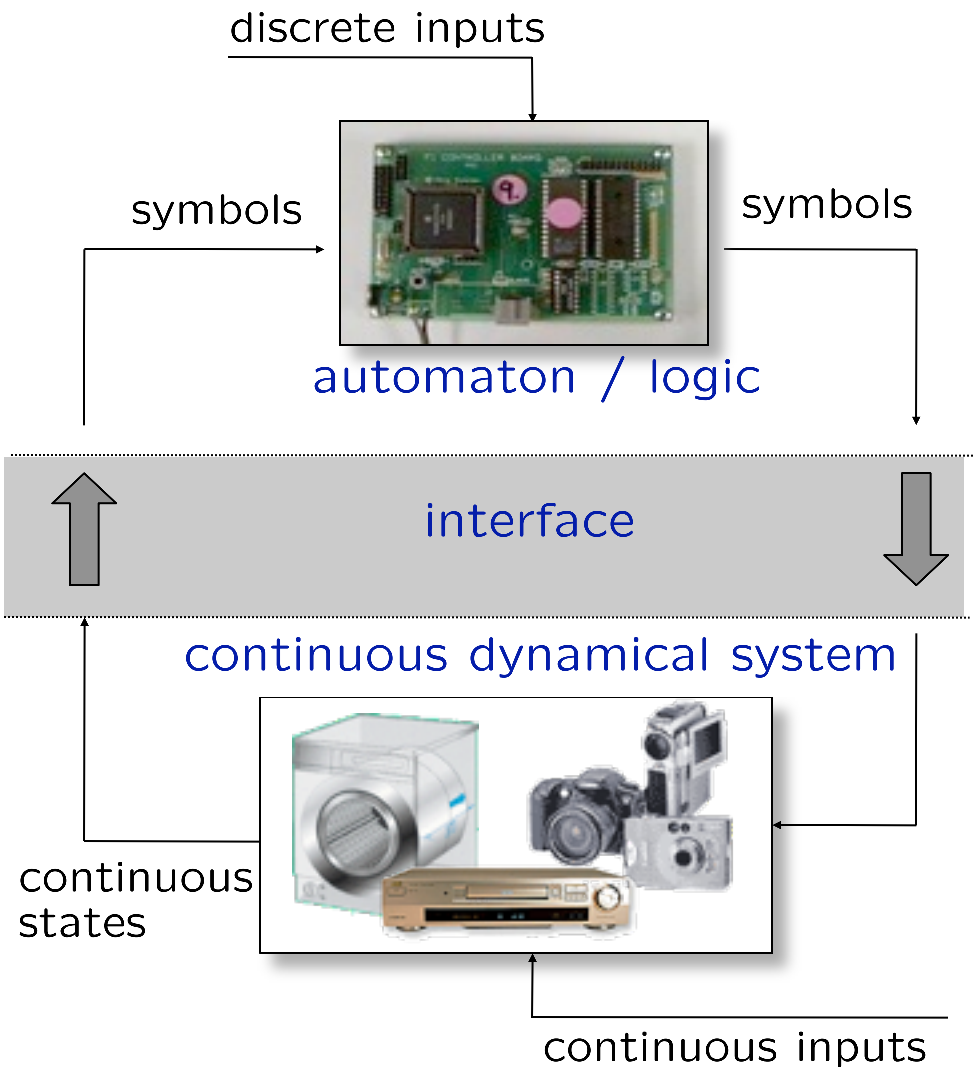
gates to process Boolean signals, 3) electronic analog switches controlled by Boolean signals.

The objective of this paper is to give a precise description of such systems, to define their dynamics, to formulate the problem of their optimum control, to introduce the notion of well-behaved solution, and to state necessary conditions for optimality (the jump conditions).

A Class of Hybrid Systems

The modifications required in otherwise continuous systems described by vector differential equations

Embedded Systems

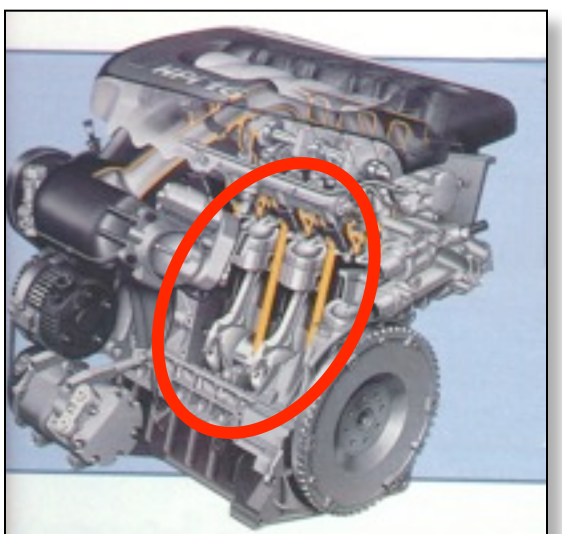


- Consumer electronics
- Home appliances
- Oce automation
- Automobiles
- Industrial plants
- ...

Motivation: “Intrinsically Hybrid”

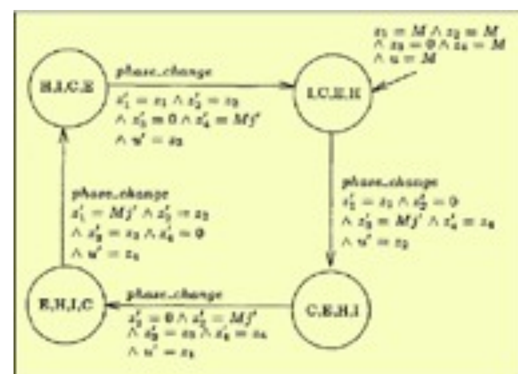
- Transmission

Discrete command (R,N,1,2,3,4,5) + Continuous dynamical variables (velocities, torques)



- Four-stroke engines

Automaton, dependent on power train motion



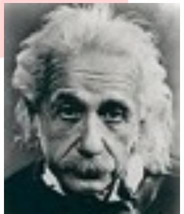
Key Requirements for Hybrid Models

- 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

linear hybrid systems

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

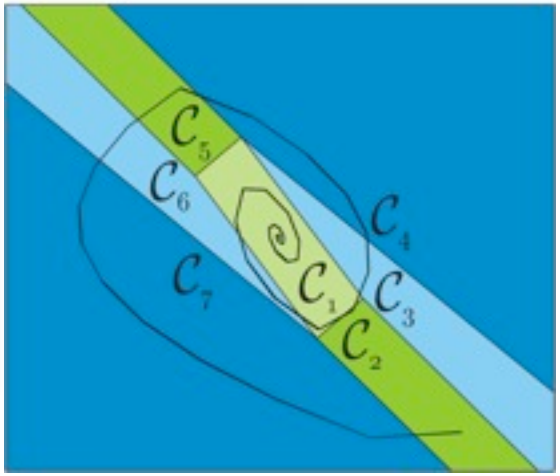


Piecewise Ane Systems

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

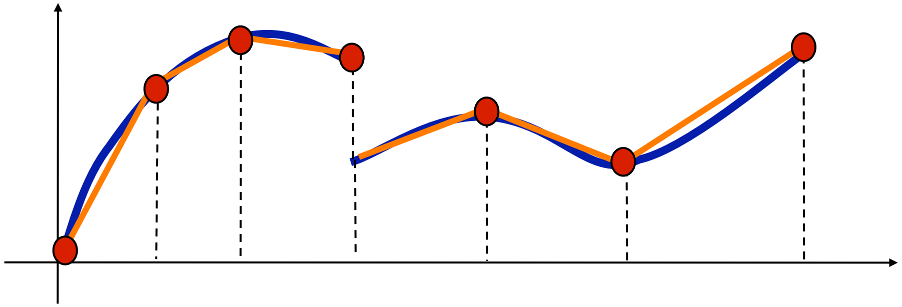
$x \in \mathcal{X} \subseteq \mathbb{R}^n$, $u \in \mathcal{U} \subseteq \mathbb{R}^m$, $y \in \mathcal{Y} \subseteq \mathbb{R}^p$
$i(k) \in \{1, \ldots, s\}$

state+input space



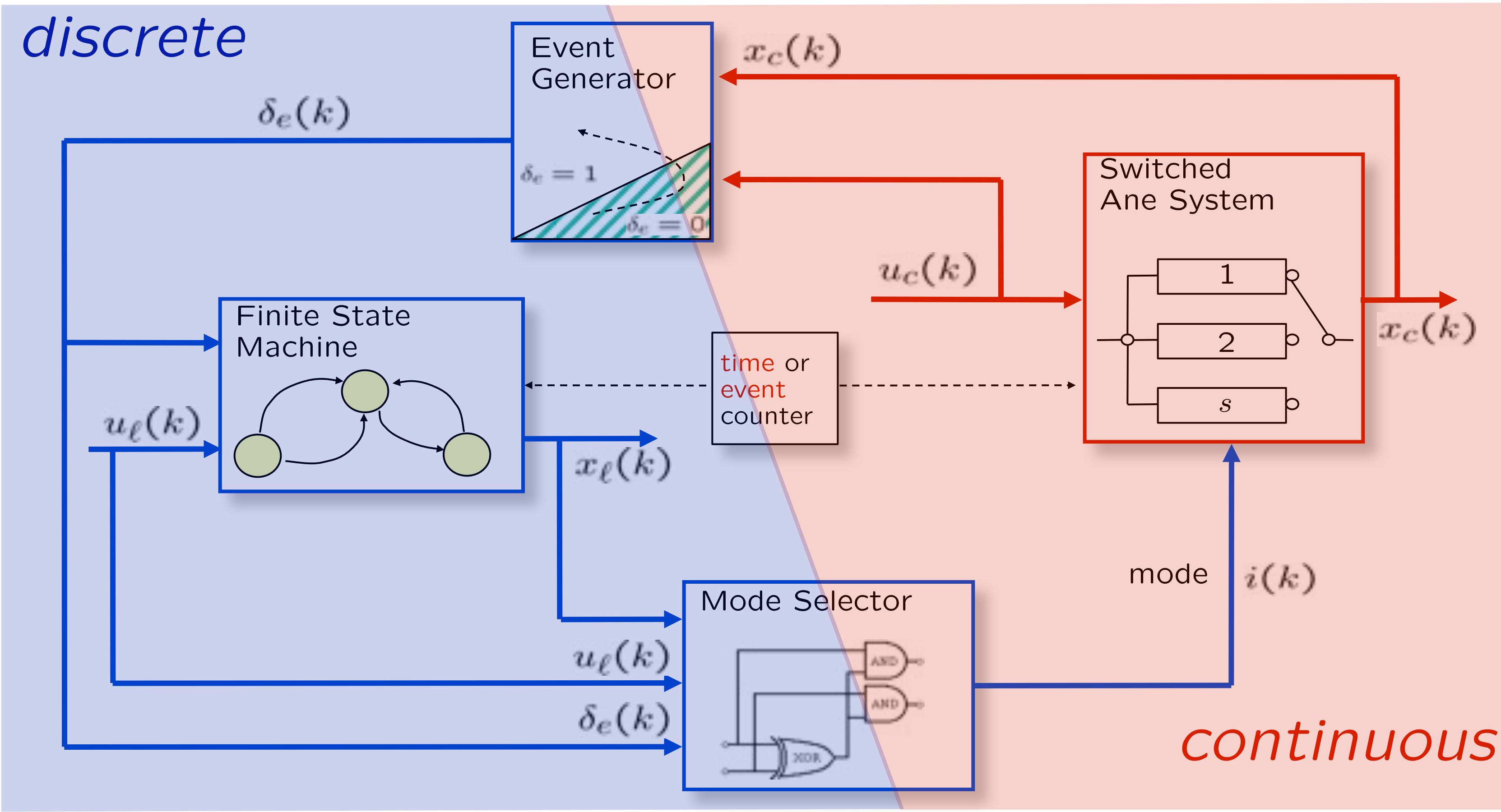
(Sontag 1981)

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

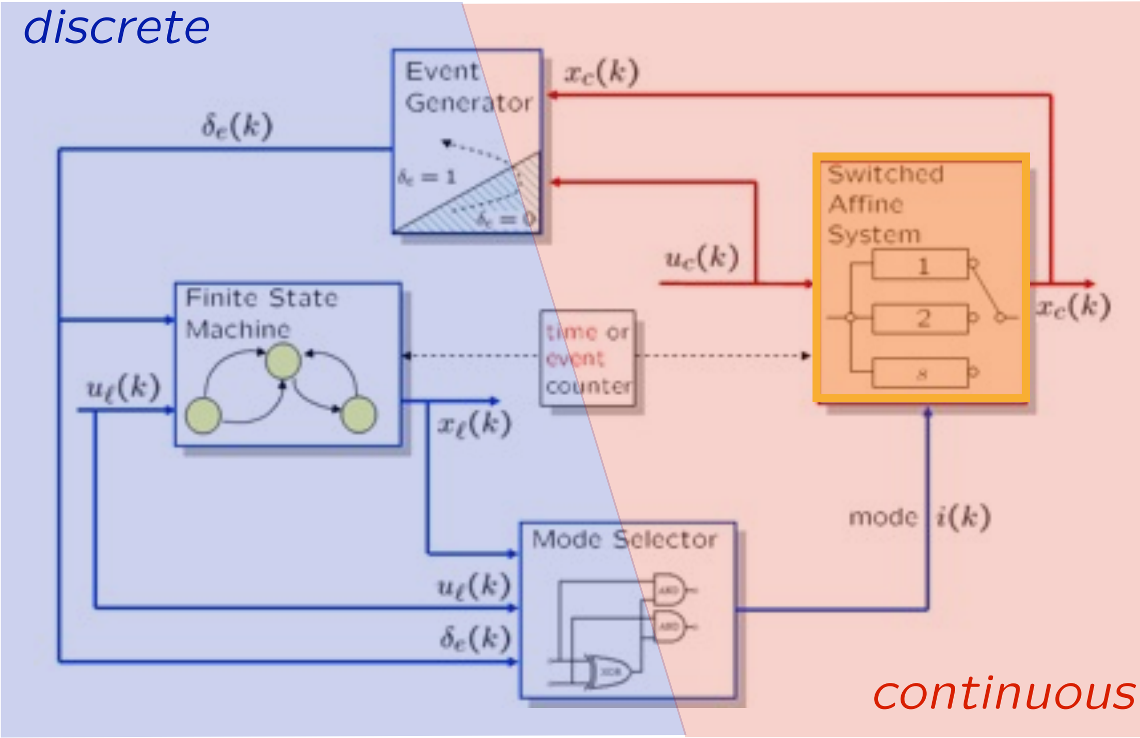
$\delta_e \in \{0,1\}^{n_e}$ = event variables

$x_c \in \mathbb{R}^{n_c}$ = continuous states

$u_c \in \mathbb{R}^{m_c}$ = continuous inputs

$i \in \{1, 2, \ldots, s\}$ = current mode

Switched Ane System

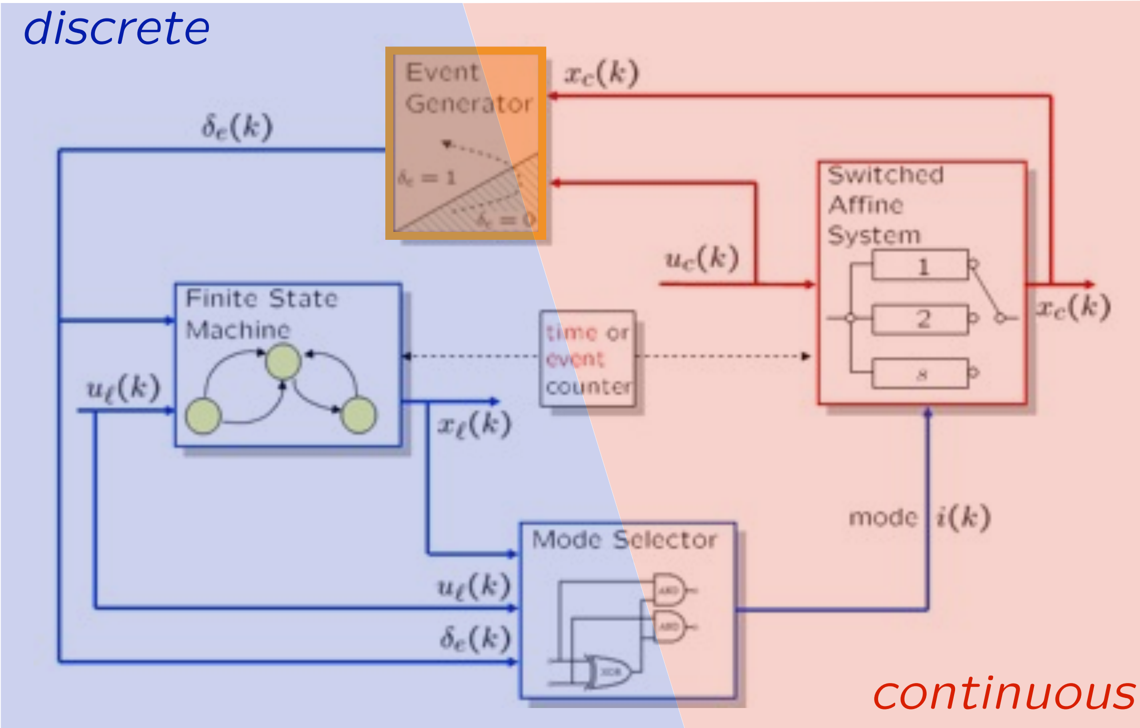


The ane dynamics depend on the current mode $i(k)$:

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

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

Event Generator



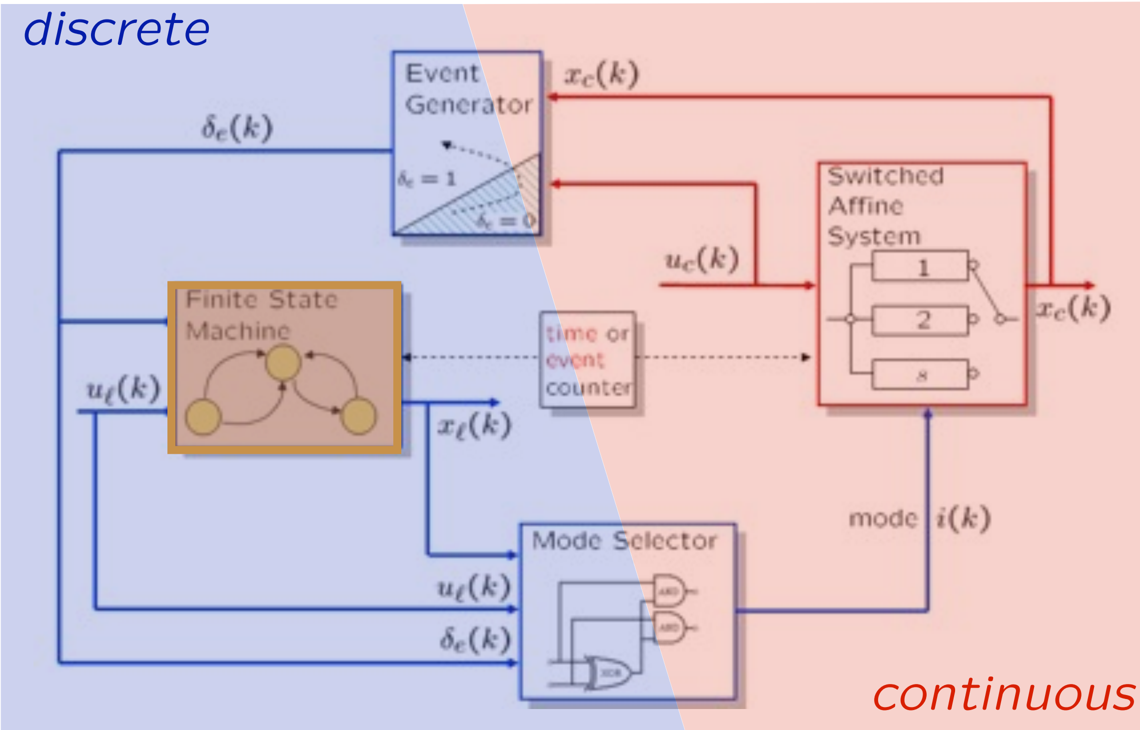
Event variables are generated by linear threshold conditions over continuous states, continuous inputs, and time:

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

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

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

Finite State Machine



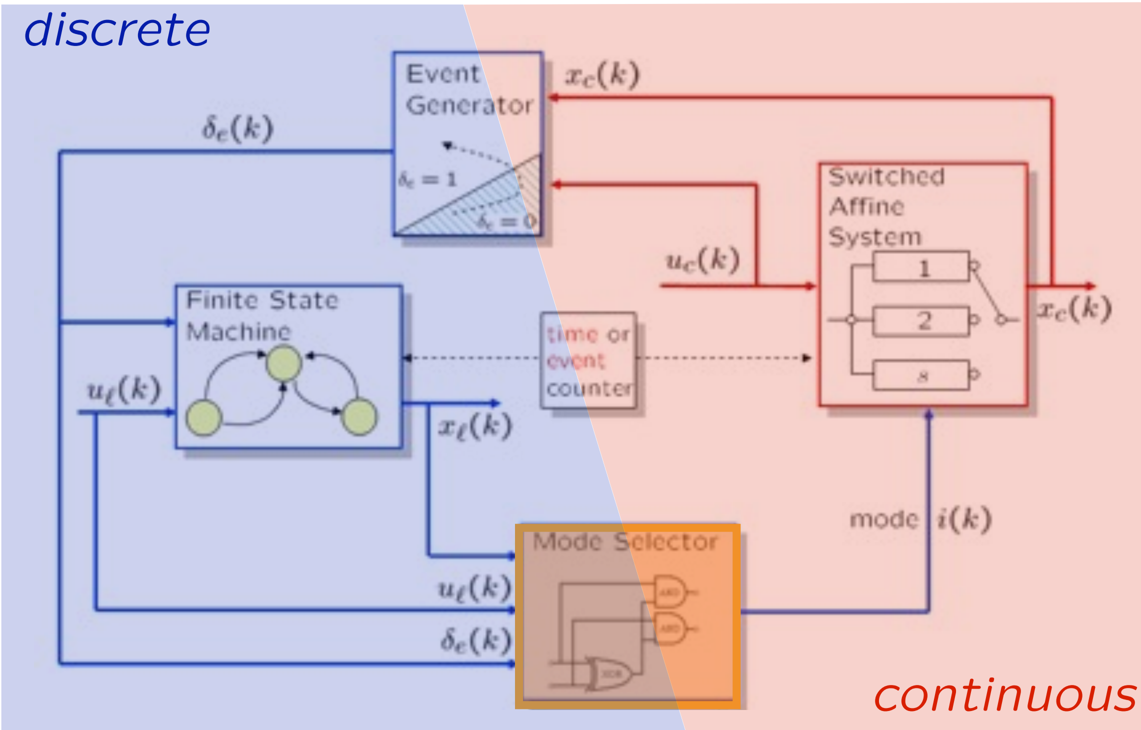
The binary state of the nite state machine evolves according to a Boolean state update function:

$$x_\ell(k+1) = f_{\mathrm{B}}(x_\ell(k), u_\ell(k), \delta_e(k))$$

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

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

Mode Selector



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:

$$i(k) = f_{\mathrm{M}}(x_\ell(k), u_\ell(k), \delta_e(k))$$

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

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	$i = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$i = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$
1	$i = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$i = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

the system has 3 modes

Logic and Inequalities

(Glover 1975, Williams 1977, Hooker 2000)

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

Any logic statement

$f(X) = \text{TRUE}$

$$\bigwedge_{j=1}^{m} \left(\vee_{i \in P_j} X_i \vee_{i \in N_j} \neg X_i \right) \quad \text{(CNF)}$$

$N_j, P_j \subseteq \{1, \ldots, n\}$

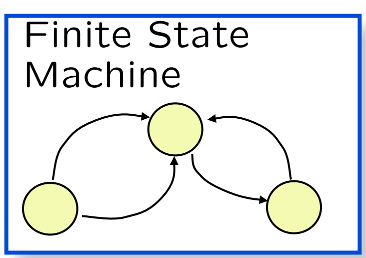
$$\begin{cases} 1 \leq \sum_{i \in P_1} \delta_i + \sum_{i \in N_1} (1 - \delta_i) \\ \vdots \\ 1 \leq \sum_{i \in P_m} \delta_i + \sum_{i \in N_m} (1 - \delta_i) \end{cases}$$

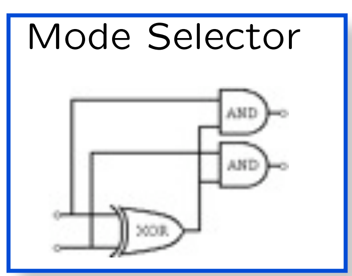
$[\delta_e^i(k) = 1] \leftrightarrow [H^i x_c(k) \leq W^i]$

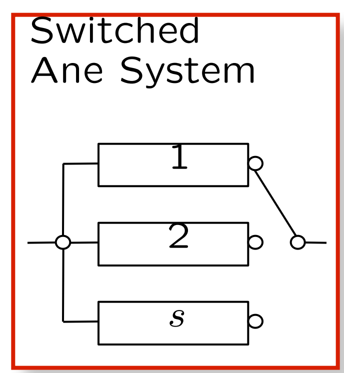
$$\begin{cases} H^i x_c(k) - W^i \leq M^i (1 - \delta_e^i) \\ H^i x_c(k) - W^i > m^i \delta_e^i \end{cases}$$

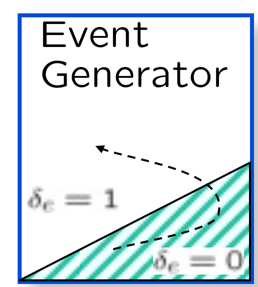
IF $[\delta = 1]$ THEN $z = a_1^T x + b_1^T u + f_1$
ELSE $z = a_2^T x + b_2^T u + f_2$

$$\begin{cases} (m_2 - M_1)\delta + z \leq a_2 x + b_2 u + f_2 \\ (m_1 - M_2)\delta - z \leq -a_2 x - b_2 u - f_2 \\ (m_1 - M_2)(1 - \delta) + z \leq a_1 x + b_1 u + f_1 \\ (m_2 - M_1)(1 - \delta) - z \leq -a_1 x - b_1 u - f_1 \end{cases}$$



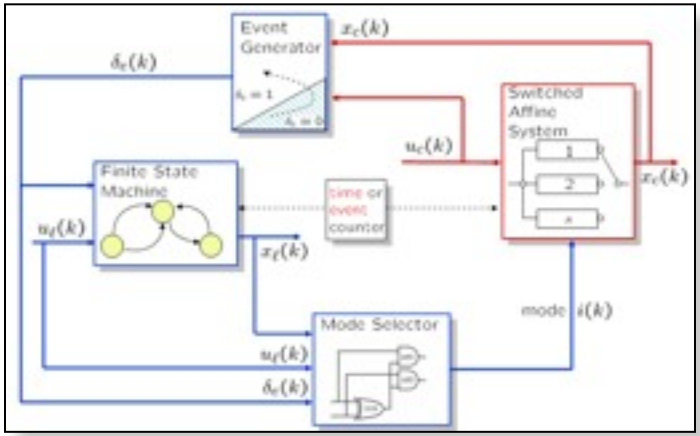






Mixed Logical Dynamical Systems

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

$x \in \mathbb{R}^{n_r} \times \{0,1\}^{n_b}$, $u \in \mathbb{R}^{m_r} \times \{0,1\}^{m_b}$

$y \in \mathbb{R}^{p_r} \times \{0,1\}^{p_b}$, $\delta \in \{0,1\}^{r_b}$, $z \in \mathbb{R}^{r_r}$

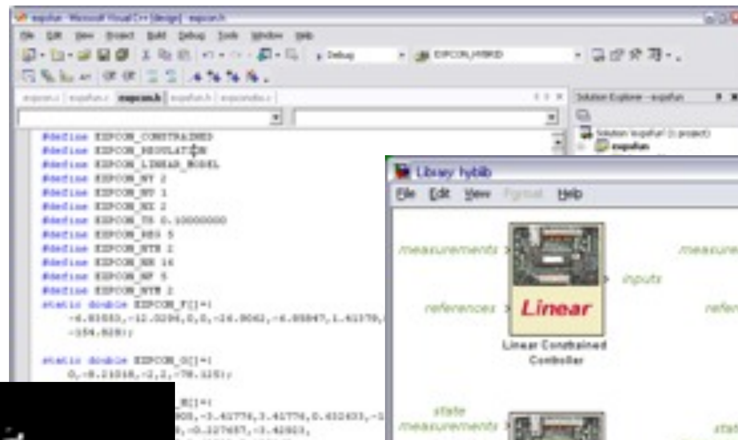
- Computationally oriented (mixed-integer programming)
- Suitable for controller synthesis, verication, ...

Hybrid Toolbox for Matlab

(Bemporad, 2003-2006)

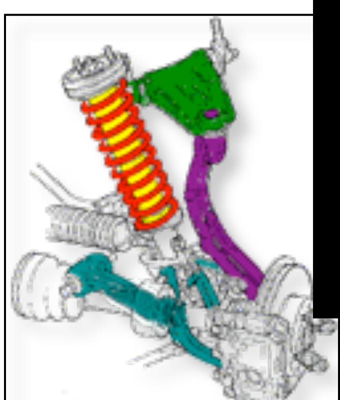
Features:

- Hybrid model (MLD and PWA) design, simulation, verication
- 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



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

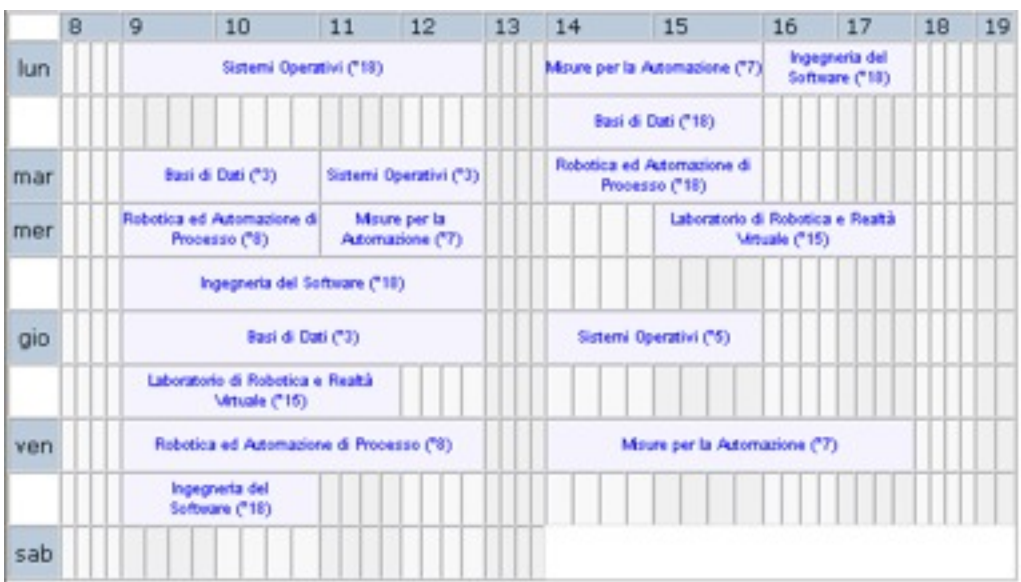
Mixed-Integer Models in OR

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

Example: Optimal multi-period investments for maintenance and upgrade of electrical energy distribution networks

(Bemporad, Muñoz, Piazzesi, 2006)

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



CPU time: 0.2 s

Major Advantages of Linear Hybrid Models

Many problems of analysis:

- Stability (Johansson, Rantzer, 1998)
- Safety / Reachability (Torrisi, Bemporad, 2001)
- Observability (Bemporad, Ferrari-Trecate, Morari, 2000)
- Passivity (Bemporad, Bianchini, Brogi, 2006)

Many problems of synthesis:

- Controller design (Bemporad, Morari, 1999)
- Robust control design (Silva, Bemporad, Botto, Sá da Costa, 2003)
- Filter design (state estimation/fault detection)
  (Bemporad, Mignone, Morari, 1999)
  (Ferrari-Trecate, Mignone, Morari, 2002)
  (Pina, Botto, 2006)

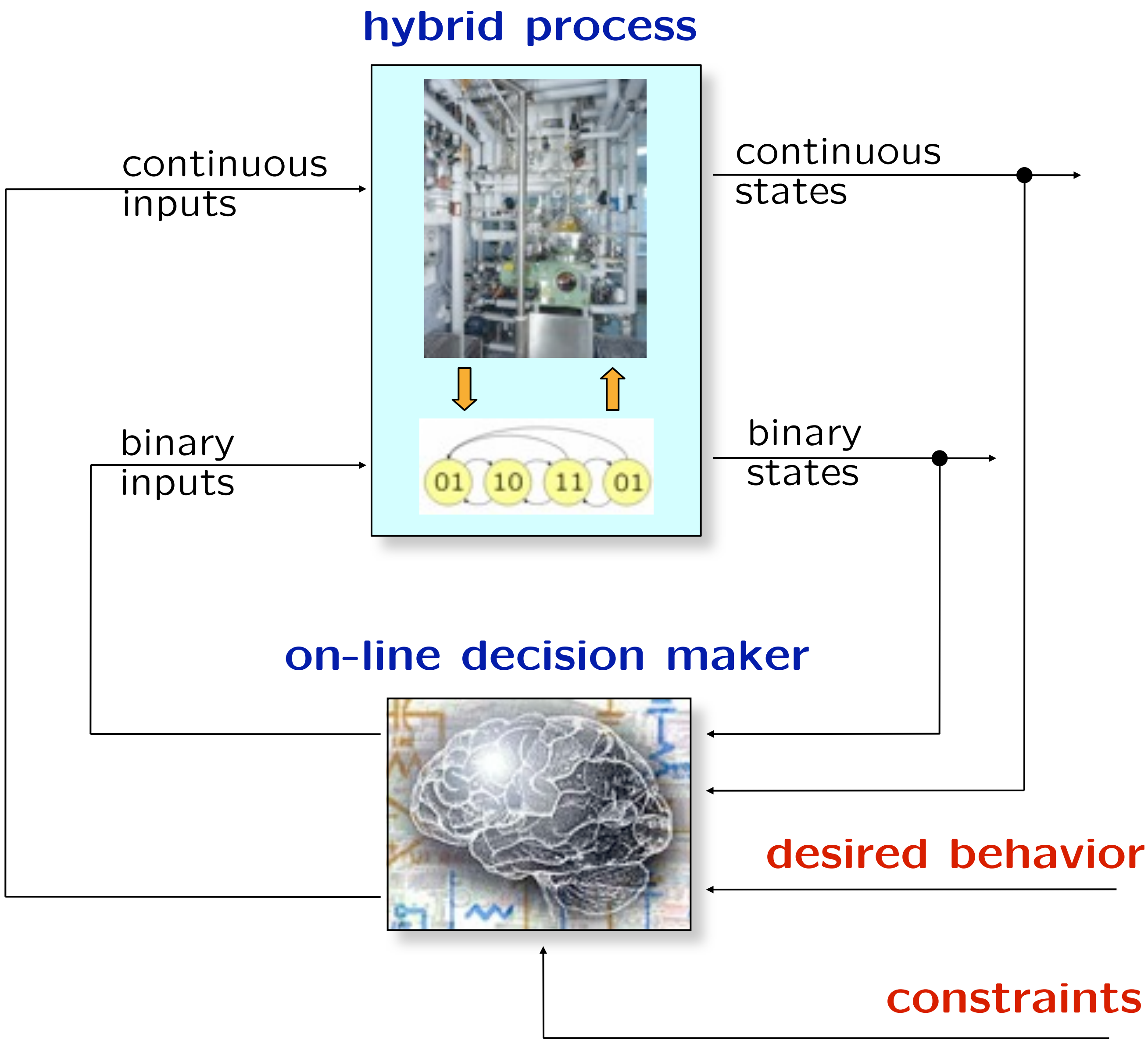
can be solved through mathematical programming

(However, all these problems are NP-hard !)

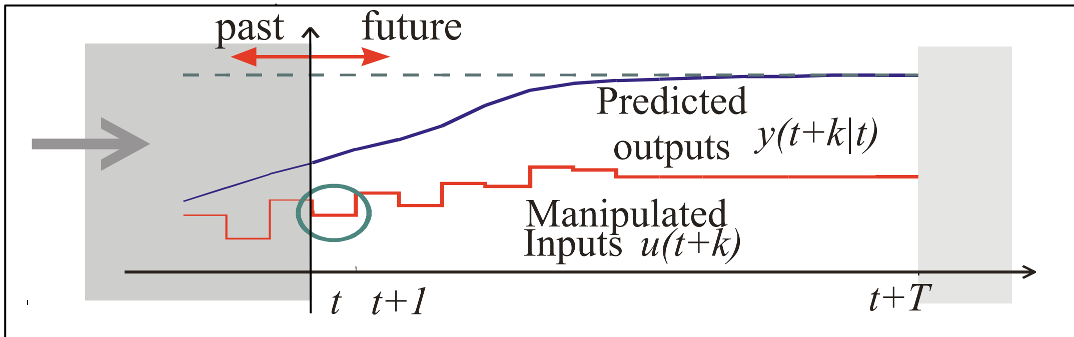
Contents

- ✓ Models of hybrid systems
- Model predictive control of hybrid systems
- Explicit reformulation
- Automotive applications

Hybrid Control Problem



MPC for Hybrid Systems



Model
Predictive (MPC)
Control

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

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

subject to MLD model

$$x(t|t) = x(t)$$
$$x(t+T|t) = x_r$$

- Apply only $u(t)=u^*(t)$ (discard the remaining optimal inputs)
- Repeat the whole optimization at time $t+1$

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)

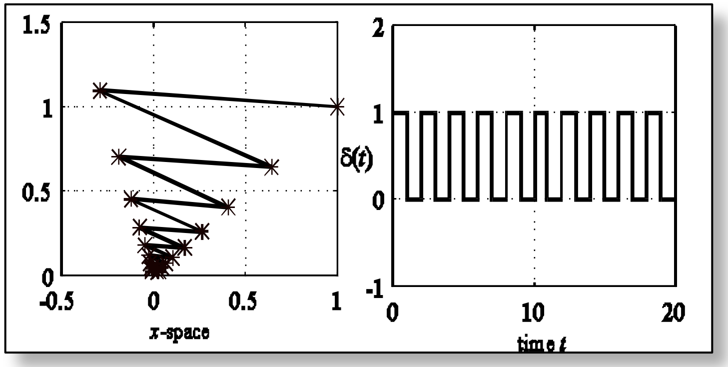
Hybrid MPC - Example

PWA system:

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

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

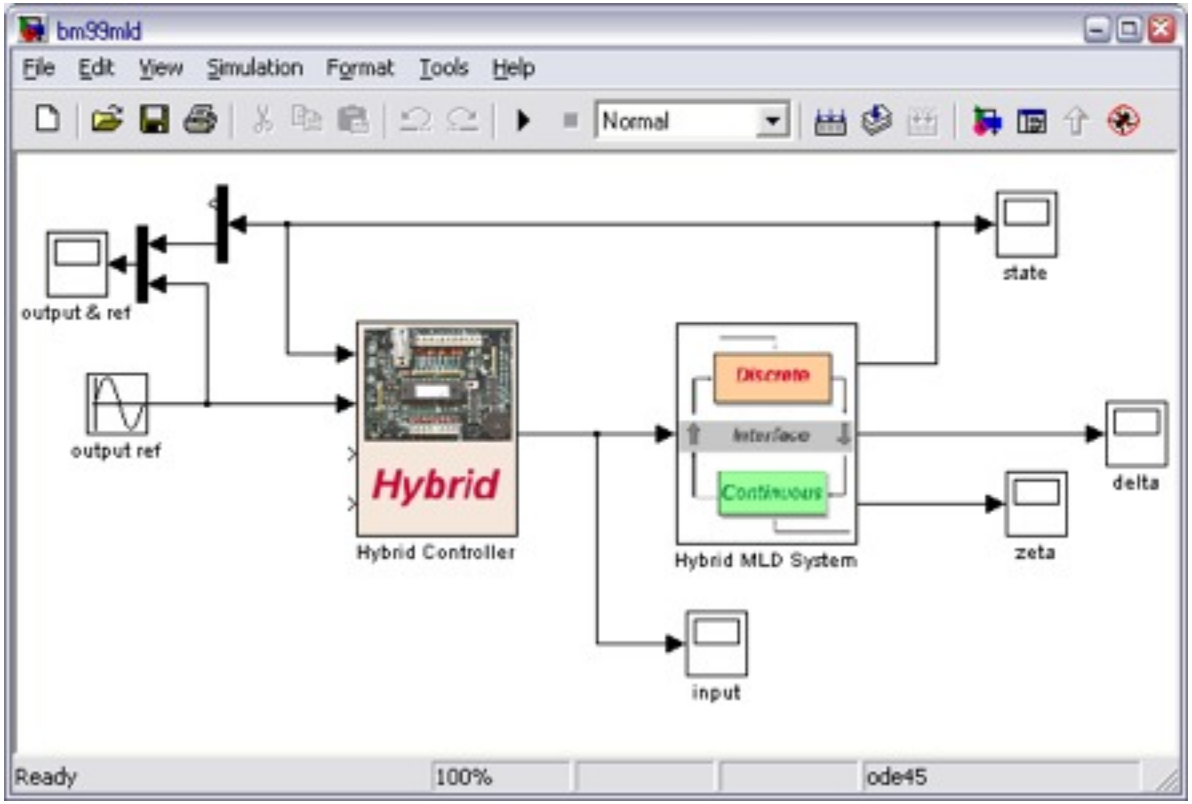
Open loop behavior

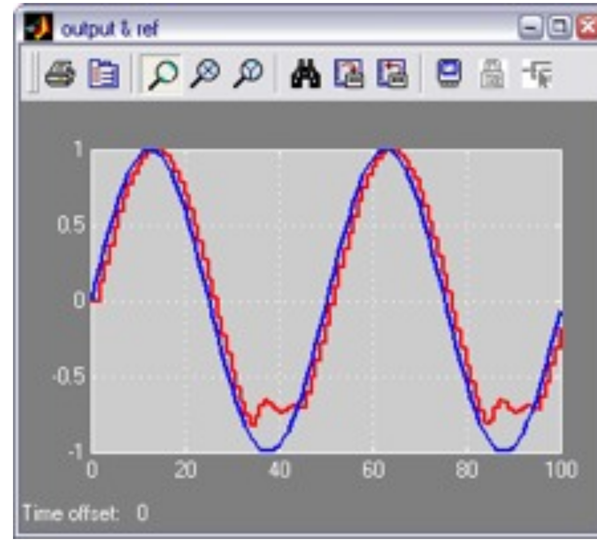


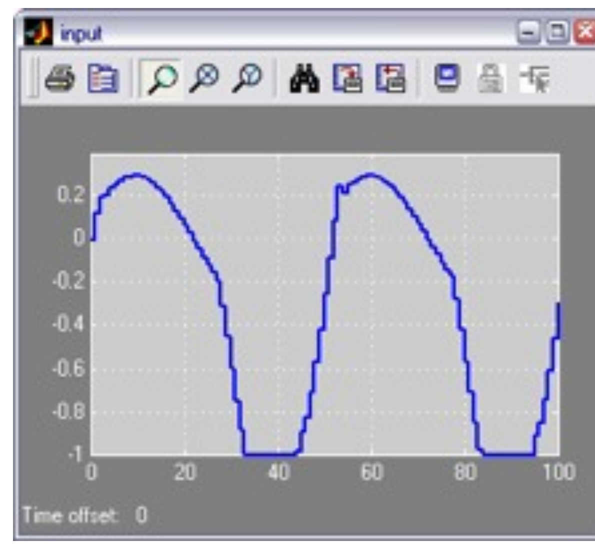
/demos/hybrid/bm99sim.m

Hybrid MPC - Example

Closed loop:



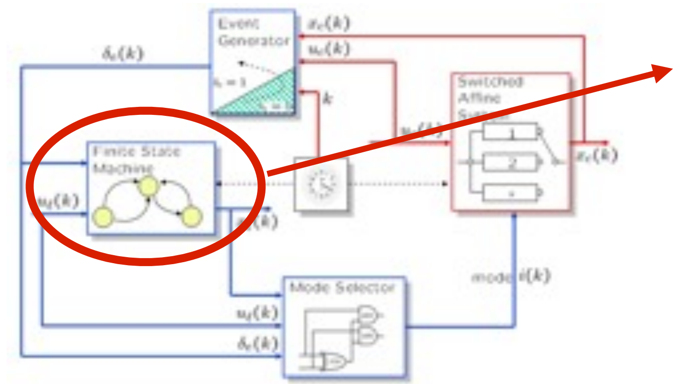




Performance index: $\min \sum_{k=1}^{2} |y(t+k|t) - r(t)|$

Hybrid MPC - Extensions

Discrete-time Hybrid Stochastic Automaton (DHSA)

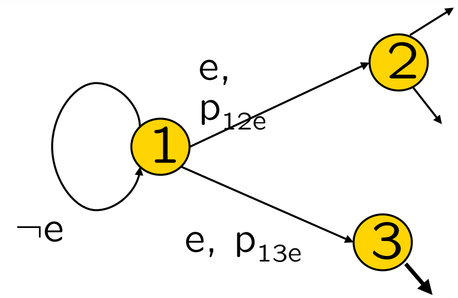


(Bemporad, Di Cairano, HSCC'05)

Stochastic Finite State Machine (sFSM)

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

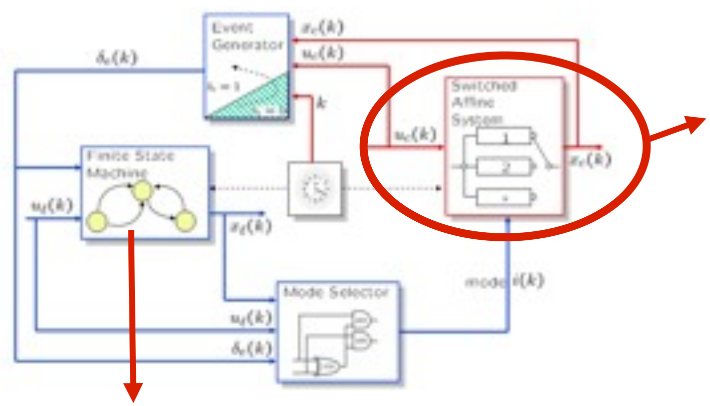
k = discrete-time counter



All MPC techniques described earlier can be applied !

Event-based Continuous-time Hybrid Automaton (icHA)

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



Switched integral dynamics

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

Asynchronous FSM

k = event counter

Optimal Control of Hybrid Systems: Computational Aspects

MPC for Linear Systems

Linear model

$$\begin{cases} x_{k+1} = Ax_k + Bu_k \\ y_k = Cx_k \end{cases} \qquad x_0 = x(t)$$

Quadratic performance index

$$\min_U J(x(t), U) = \sum_{k=0}^{N-1} \left[x_k' Q x_k + u_k' R u_k \right] + x_N' P x_N$$

$$\begin{array}{ll} \min_U & \frac{1}{2} U' H U + x'(t) F' U + \frac{1}{2} x(t) Y x(t) \\ \text{subj. to} & GU \leq W + Sx(t), \end{array} \qquad U = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}$$

Constraints

$$\begin{cases} u_{\min} \leq u_k \leq u_{\max} \\ y_{\min} \leq y_k \leq y_{\max} \end{cases}$$

This is a **(convex) Quadratic Program (QP)**

MIQP Formulation of MPC

(Bemporad, Morari, 1999)

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

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

- Optimization vector:

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

$$\min_\xi \frac{1}{2}\xi' H\xi + x(0)'F\xi + \frac{1}{2}x'(0)Yx(0)$$

$$\text{subj. to} \quad G\xi \leq W + Sx(t)$$

(the term $\frac{1}{2}x'(0)Yx(0)$ is struck through)

Mixed Integer Quadratic Program (MIQP)

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

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:

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

- Generalization: $\begin{cases} \epsilon_k^x \geq \|Qy(t+k|t)\|_\infty \\ \epsilon_k^u \geq \|Ru(t+k)\|_\infty \end{cases}$

- Optimization vector:

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

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

Mixed Integer Linear Program (MILP)

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

Mixed-Integer Program Solvers

- Mixed-Integer Programming is NP-hard

BUT

- Extremely rich literature in Operations Research (still very active)

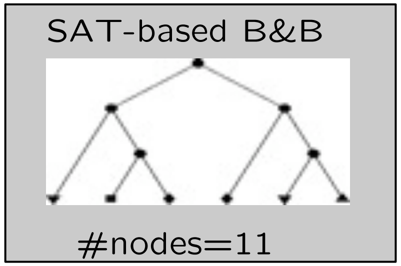
MILP/MIQP is nowadays a technology (CPLEX, Xpress-MP, BARON, GLPK, see e.g. http://plato.la.asu.edu/bench.html for a comparison)

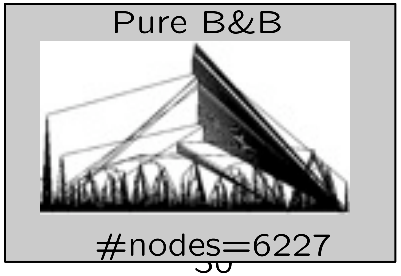
- No need to reach the global optimum for stability of MPC (see proof of the theorem), although performance deteriorates

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

(Bemporad, Giorgetti, IEEE TAC 2006)

		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





Contents

- ✓ Models of hybrid systems
- ✓ Model predictive control of hybrid systems
- Explicit reformulation
- Automotive applications
- Hybrid MPC over wireless sensor networks

On-Line vs Off-Line Optimization

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

$$\text{subject to} \begin{cases} \text{MLD model} \\ x(t|t) = x(t) \end{cases}$$

- On-line optimization: given $x(t)$ solve the problem at each time step t.

Mixed-Integer Linear/Quadratic Program (MILP/MIQP)

➢ Good for large sampling times (e.g., 1 h) / expensive hardware ...

... but not for fast sampling (e.g. 10 ms) / cheap hardware !

- Off-line optimization: solve the MILP/MIQP for all $x(t)$

$$\min_\zeta J(\zeta, x(t)) = \begin{cases} f'\zeta & \infty\text{-norm} \\ \zeta' H\zeta + f'\zeta & 2\text{-norm} \end{cases}$$

$$\text{s.t. } G\zeta \leq W + Fx(t)$$

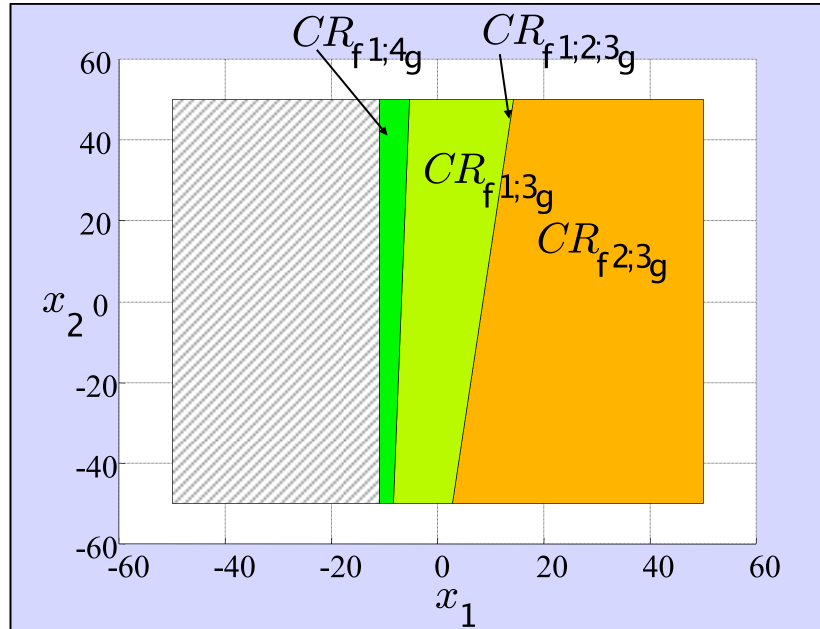
multi-parametric programming

Example of Multiparametric Solution

Multiparametric LP (ø2 R^2)

$$\min_{\xi} \quad -3\xi_1 - 8\xi_2$$

$$\text{s.t.} \quad \begin{cases} \xi_1 + \xi_2 & \leq & 13 + x_1 \\ 5\xi_1 - 4\xi_2 & \leq & 20 \\ -8\xi_1 + 22\xi_2 & \leq & 121 + x_2 \\ -4\xi_1 - \xi_2 & \leq & -8 \\ -\xi_1 & \leq & 0 \\ -\xi_2 & \leq & 0 \end{cases}$$



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

MPC for Linear Systems

Linear model

$$\begin{cases} x_{k+1} = Ax_k + Bu_k \\ y_k = Cx_k \end{cases} \qquad x_0 = x(t)$$

Quadratic performance index

$$\min_U J(x(t), U) = \sum_{k=0}^{N-1} \left[x_k' Q x_k + u_k' R u_k \right] + x_N' P x_N$$

$$\begin{array}{ll} \min\limits_U & \frac{1}{2} U'HU + x'(t)F'U + \frac{1}{2} x(t)Yx(t) \\ \text{subj. to} & GU \leq W + Sx(t), \end{array} \qquad U = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}$$

Constraints

$$\begin{cases} u_{\min} \leq u_k \leq u_{\max} \\ y_{\min} \leq y_k \leq y_{\max} \end{cases}$$

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

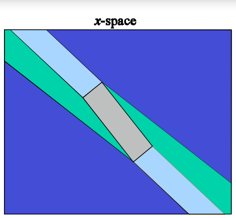
Properties of multiparametric-QP

(Bemporad et al., 2002)

Optimizer	$U^*(x) = \arg\min_U \ \frac{1}{2}U'HU + x'F'U$ subj. to $GU \leq W + Sx$	continuous, piecewise ane
Value function	$V^*(x) = \frac{1}{2}x'Yx + \min_U \ \frac{1}{2}U'HU + x'F'U$ subj. to $GU \leq W + Sx$	convex continuous, piecewise quadratic, C^1 (if no degeneracy)
Feasible state set	$X^* = \{x : \ \exists U \text{ such that } Gx \leq W + SU\}$	convex polyhedral

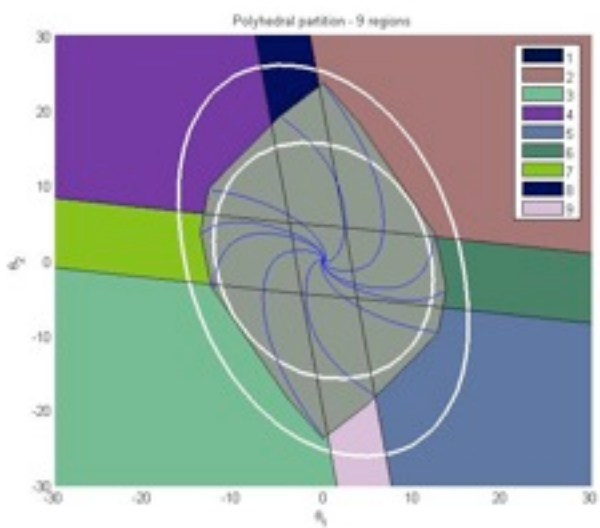
Corollary: The linear MPC controller is a continuous piecewise ane function of the state

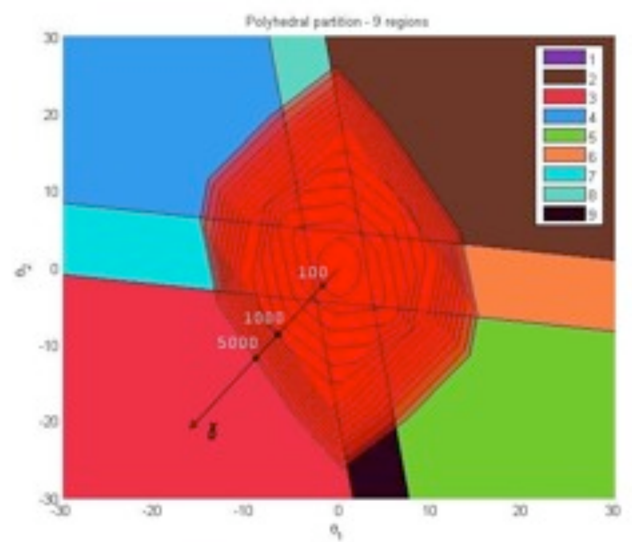
$$u(x) = \begin{cases} F_1 x + g_1 & \text{if } H_1 x \leq K_1 \\ \vdots & \vdots \\ F_M x + g_M & \text{if } H_M x \leq K_M \end{cases}$$



Polyhedral Invariant Sets for Closed-loop Linear MPC Systems

- Convexity of value function implies convexity of the piecewise ellipsoidal sets $\{x : V^*(x) \leq \gamma\}$ and $\{x : V^*(x) - x'Qx \leq \gamma\}$
- Any polyhedron P contained in between is a **positively invariant** set for the closed-loop MPC system





- By changing γ invariant polyhedra of arbitrary size can be constructed for the closed-loop MPC system

(Alessio, Bemporad, Lazar, Heemels, CDC'06)

(Note: explicit form of MPC not required)

Explicit Hybrid MPC (PWA)

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

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

$p = 1, 2, \infty$

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

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

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

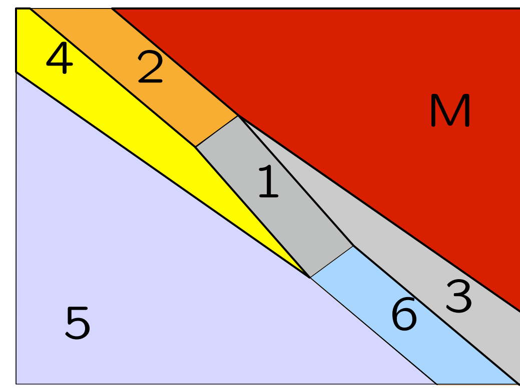
- **The MPC controller is piecewise ane in x,r**

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

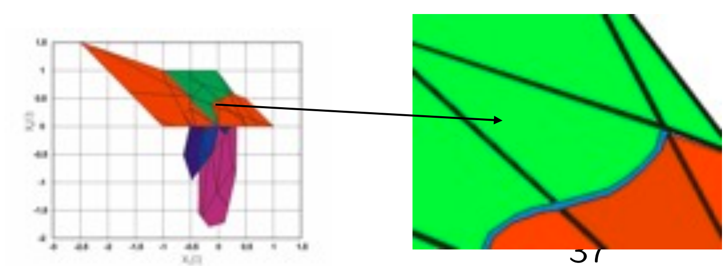
(Borrelli, Baotic, Bemporad, Morari, *Automatica*, 2005)

(Mayne, ECC 2001) (Alessio, Bemporad, ADHS 2006)

(x,r)-space



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



Hybrid Control - Example

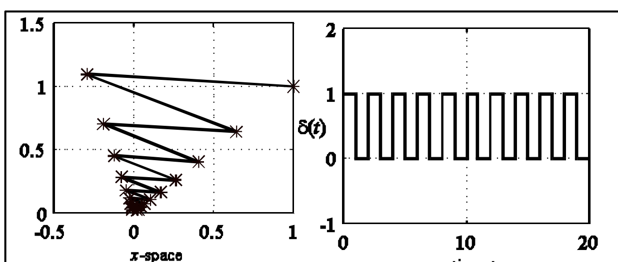
PWA system:

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

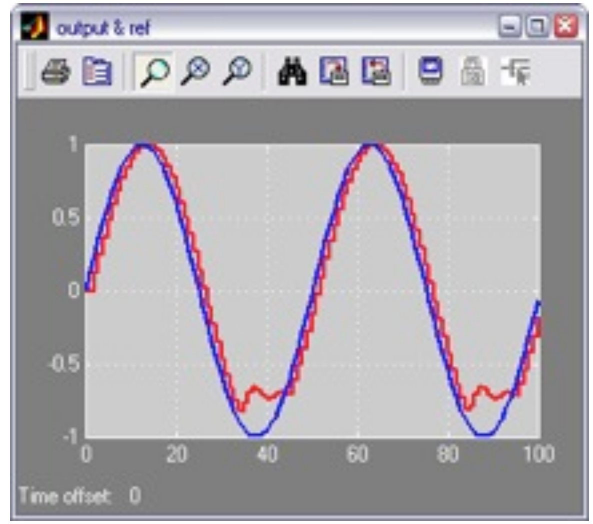
Constraints: $-1 \leq u(t) \leq 1$

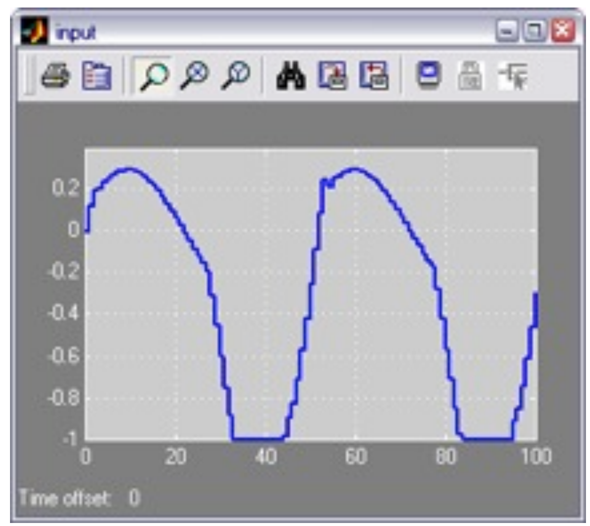
Objective: $\min \sum_{k=1}^{2} |y(t+k|t) - r(t)|$

Open loop behavior:



Closed loop:

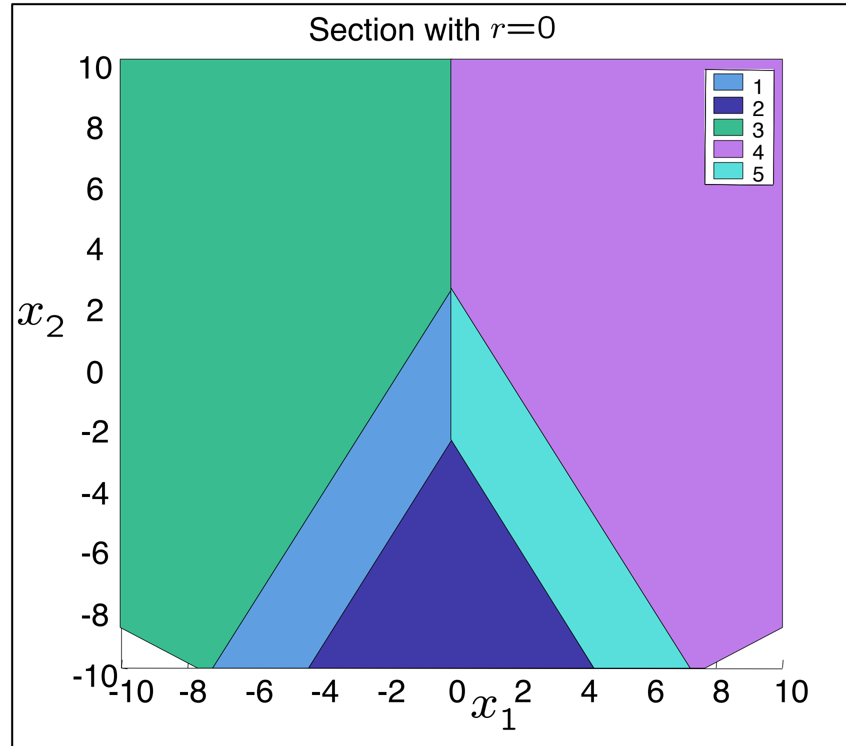




HybTbx: **`/demos/hybrid/bm99sim.m`**

Explicit PWA Controller

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



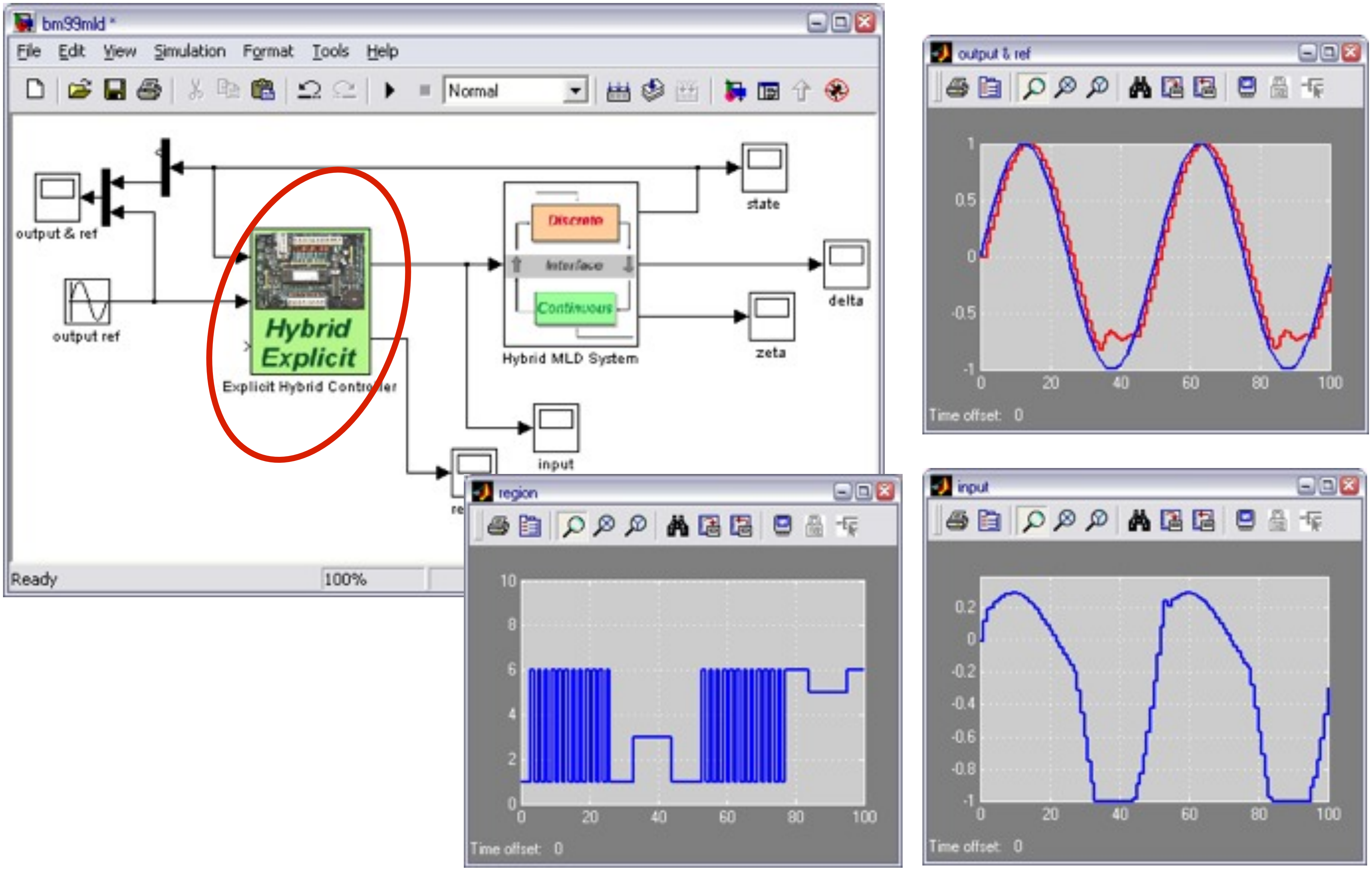
PWA law ≡ MPC law

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

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

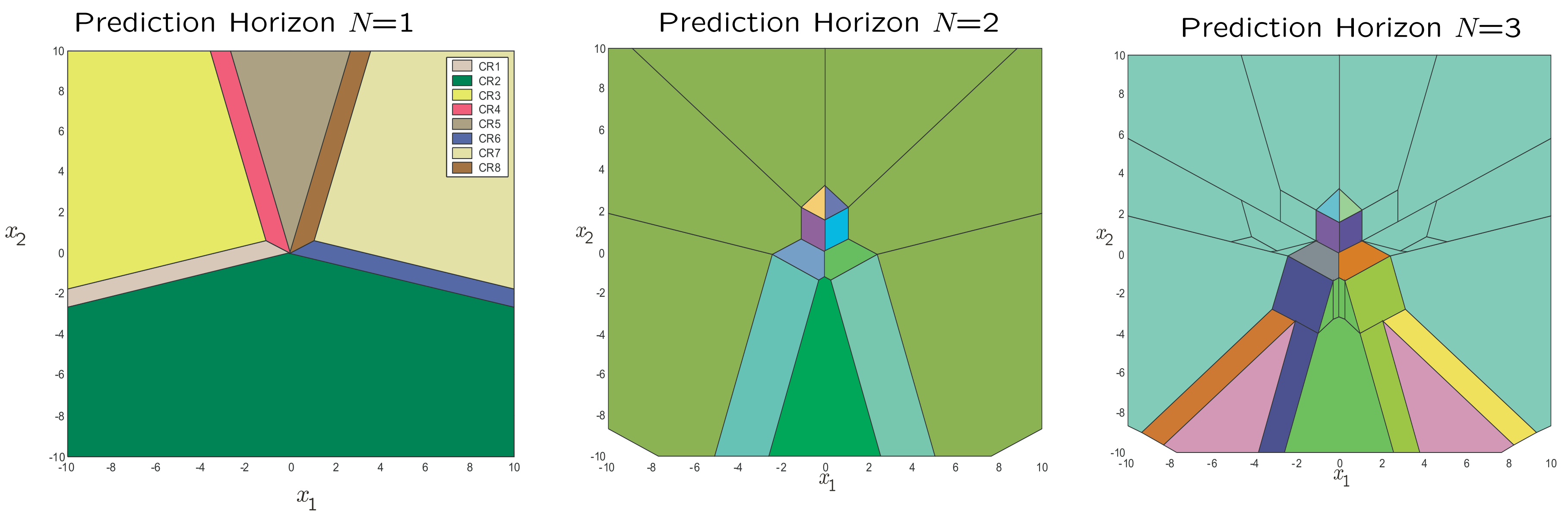
Hybrid MPC - Example

Closed loop:



Explicit PWA Regulator

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



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

Comments on Explicit Solutions

- Alternatives: either (1) solve an MIP on-line
  or (2) evaluate a PWA function

- For problems with many variables and/or long horizons: MIP may be preferable

- For simple problems (short horizon/few constraints):
  - 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

Contents

✓ Models of hybrid systems

✓ Model predictive control of hybrid systems

✓ Explicit reformulation

- Automotive applications

(Photo: Courtesy Mitsubishi)

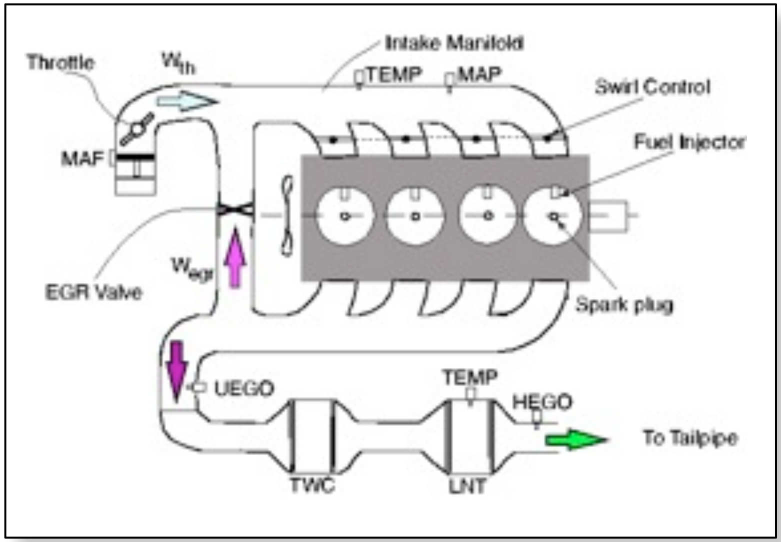
Hybrid Control of a DISC Engine

(joint work with N. Giorgetti, G. Ripaccioli, I. Kolmanovsky, and D. Hrovat)

DISC Engine

Two distinct regimes:

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



Objective: Design a controller for the engine that

- Automatically choose operating **mode** (homogeneous/stratied)
- Can cope with **nonlinear** dynamics
- Handles **constraints** (on A/F ratio, air-ow, spark)
- Achieves **optimal** performance

Hybridization of DISC Dynamics

- Proprietary nonlinear model of the DISC engine developed and validated at Ford Research Labs (Dearborn) *(Kolmanovsky, Sun, ...)*
- Model good for simulation, not good for control design!

MODEL HYBRIDIZATION

DYNAMICS (intake pressure, air-to-fuel ratio, torque):

- Denition of two operating points;
- Numerical linearization of nonlinear dynamics;
- Time discretization of the linear models.

→ ρ-dependent dynamic equations

CONSTRAINTS on:

- Air-to-Fuel Ratio: $\lambda_{min}(\rho) \leq \lambda(t) \leq \lambda_{max}(\rho)$;
- Mass of air through the throttle: $0 \leq W_{th} \leq K$;
- Spark timing: $0 \leq \delta(t) \leq \delta_{mbt}(\lambda, \rho)$

→ ρ−dependent constraints

Hybrid system with 2 modes (switched ane system)

Integral Action

Integrators on torque error and air-to-fuel ratio error added to obtain zero osets in steady-state:

$$\epsilon_\tau(t+1) = \epsilon_\tau(t) + T \cdot (\tau_{ref} - \tau)$$
$$\epsilon_\lambda(t+1) = \epsilon_\lambda(t) + T \cdot (\lambda_{ref} - \lambda)$$

T = sampling time

τ_{ref}, λ_{ref} brake torque and air-to-fuel references

Simulation based on nonlinear model conrms zero osets in steady-state

(despite the model mismatch)

Reference values are automatically generated from τ_{ref} and λ_{ref} by numerical computation based on the nonlinear model

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

$$\min_{\xi} J(\xi, x(t)) = \sum_{k=0}^{N-1} u_k' R u_k + y_k' Q y_k + x_{k+1}' S x_{k+1}$$

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

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

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

Weights:

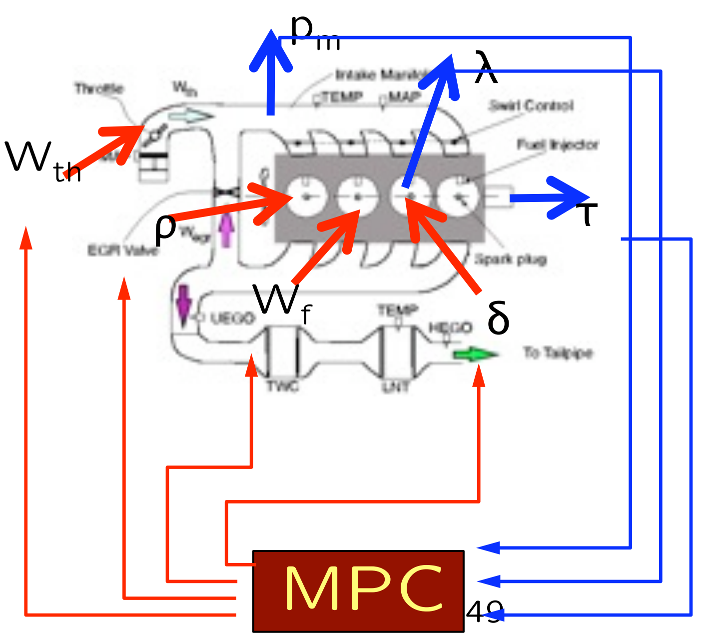
$$R = \begin{pmatrix} 0.01 & 0 & 0 & 0 \\ 0 & 0.001 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

r_ρ (prevents unneeded chattering)

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

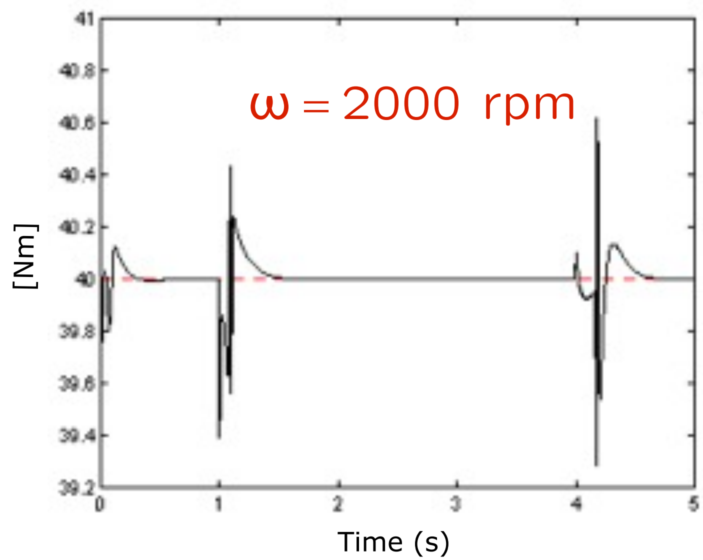
q_τ q_λ s_{ε_τ} s_{ε_λ}

main emphasis on torque

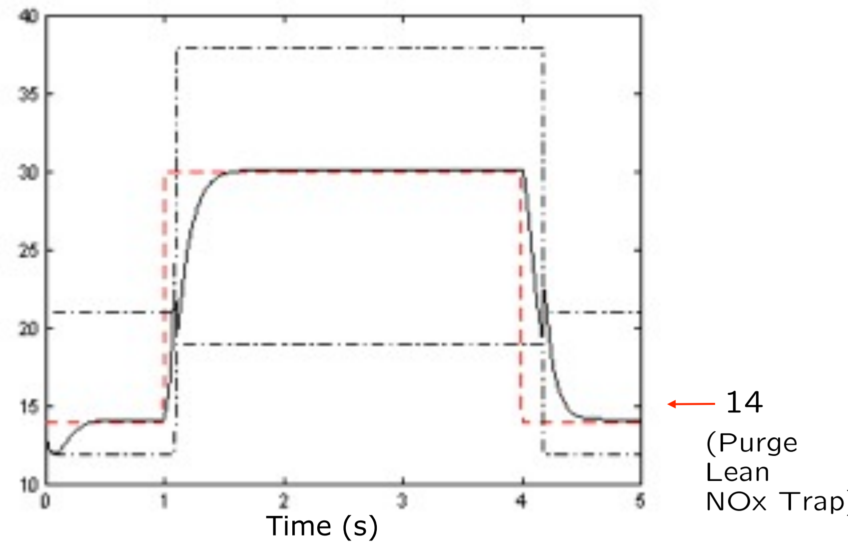


Simulation Results (nominal engine speed)

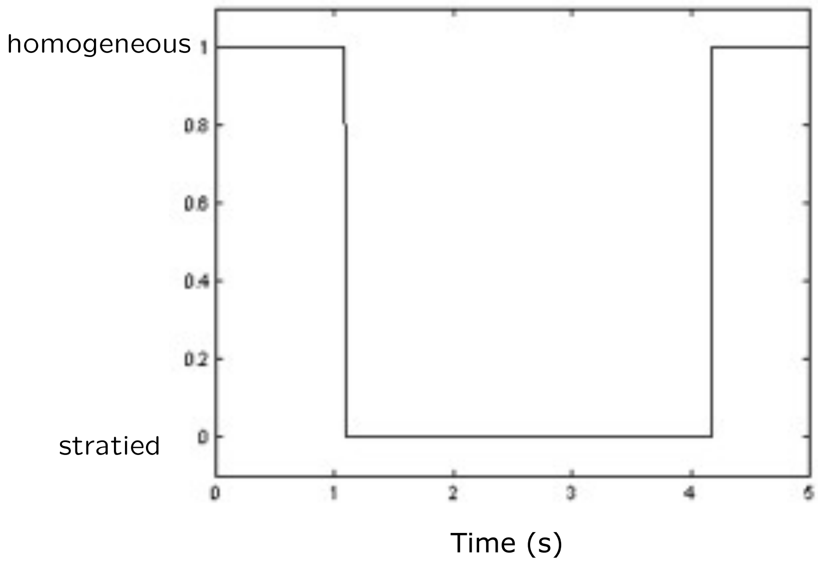
Engine Brake Torque



Air-to-Fuel Ratio



Combustion mode

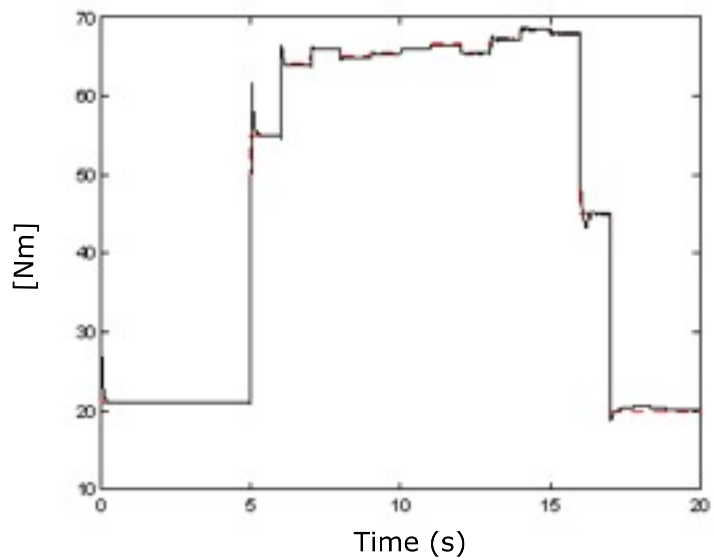


- Control horizon N=1;
- Sampling time Ts=10 ms;
- PC Xeon 2.8 GHz + Cplex 9.1

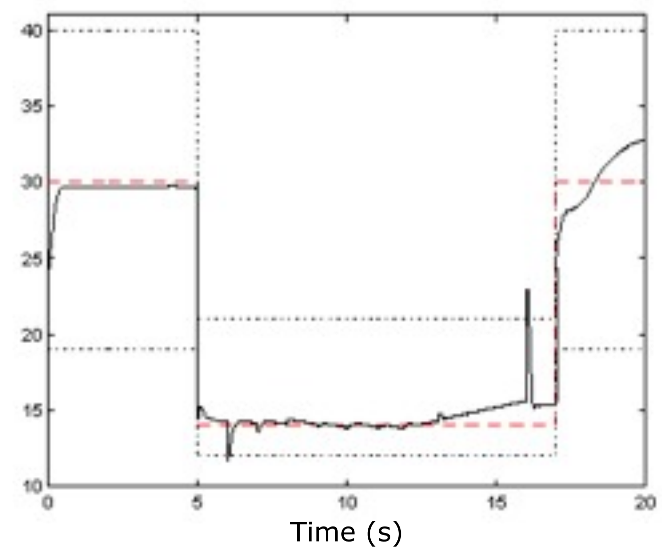
$\simeq$ 3 ms per time step

Simulation Results (varying engine speed)

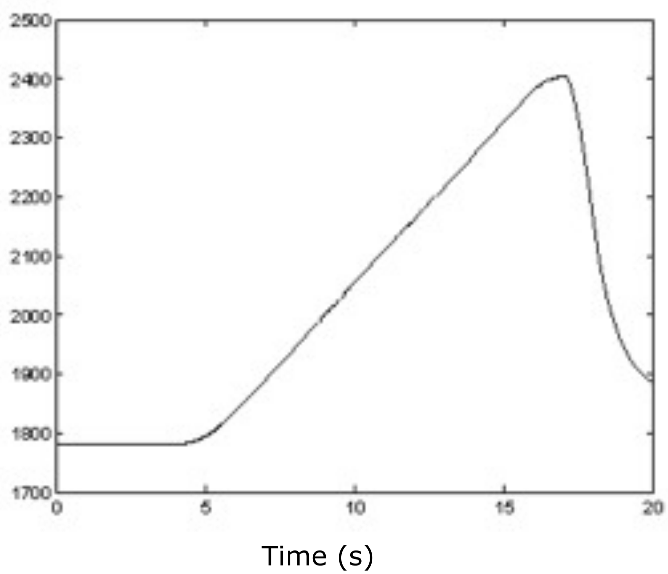
Engine Brake Torque



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

Control code too complex (MILP) $\Rightarrow$ not implementable !

Explicit MPC Controller

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

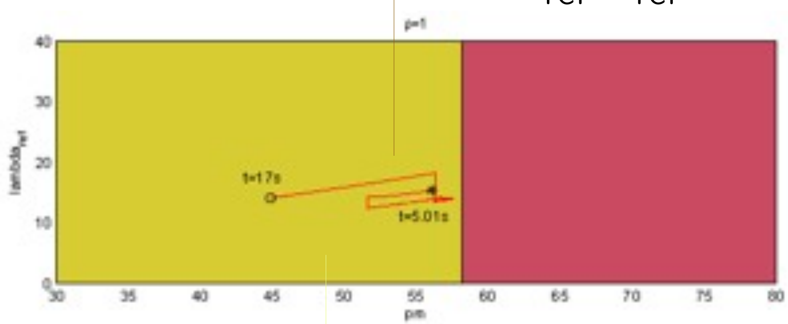
where: $u = [W_{th}\ W_f\ \delta\ \rho]'$

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

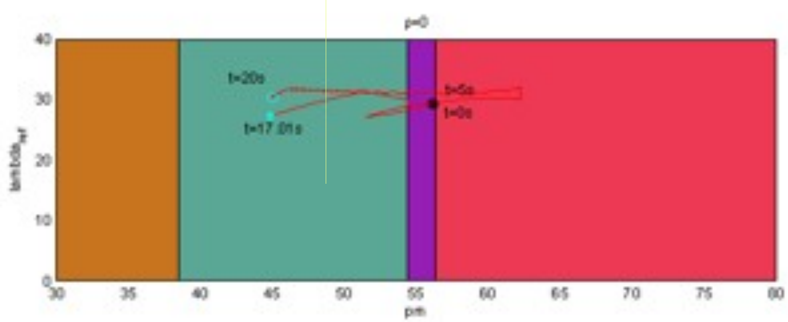
N=1 (control horizon)

75 partitions

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



ρ=0



ρ=1

- Time to compute the explicit MPC: 3.4750 s;
- Sampling time Ts=10 ms;
- PC Xeon 2.8 GHz + Cplex 9.1

$\simeq$ 8 μs per time step

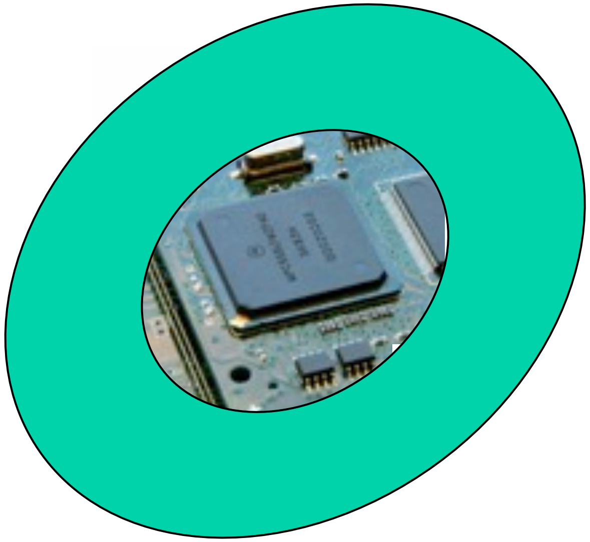
Microcontroller Implementation

- C-code **automatically** generated by the Hybrid Toolbox

- Microcontroller Motorola MPC 555 (custom made for Ford)

- 43 Kb memory available

- Floating point arithmetic



→ $\approx$ 3ms execution time

sampling period = 10ms → **Implementable !**

- Further reduction of number of partitions possible

(Alessio, Bemporad, 2005)

- C-code can be further optimized

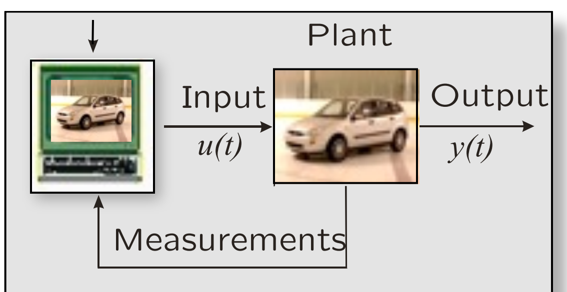
(Tøndel, Johansen, Bemporad, 2003)

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 o-line via multiparametric programming for fast-sampling applications



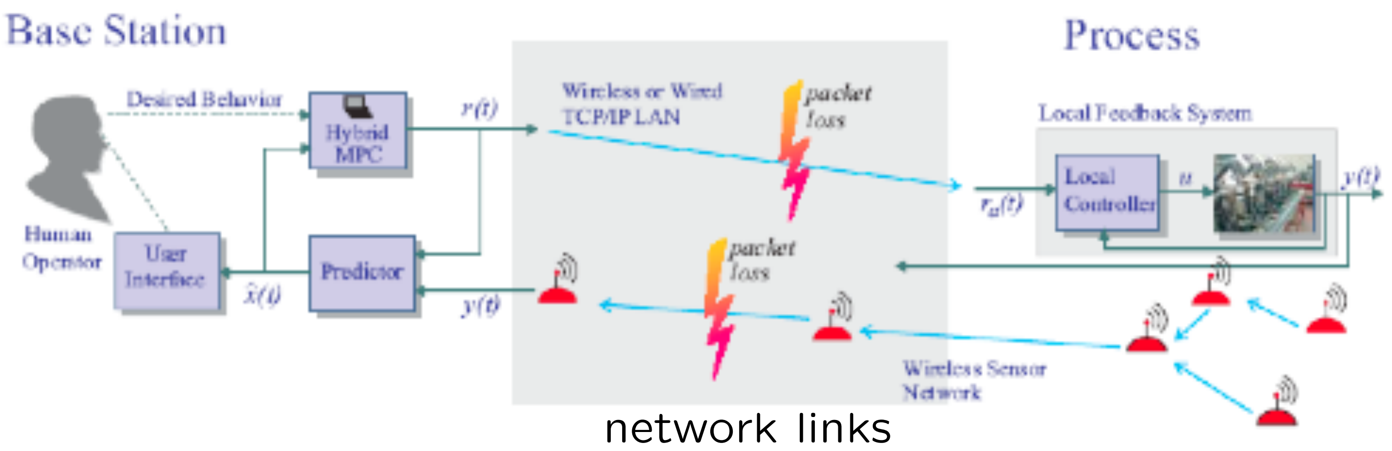
Hybrid modeling and MPC design

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

Hybrid MPC & Wireless Sensor Networks



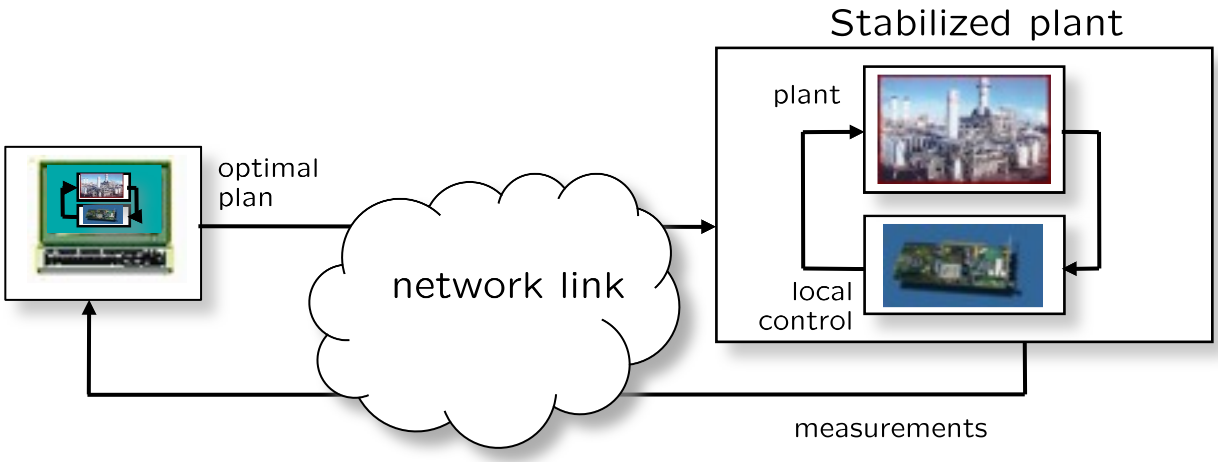
- Measurements acquired and sent to base station (MPC) by wireless sensors

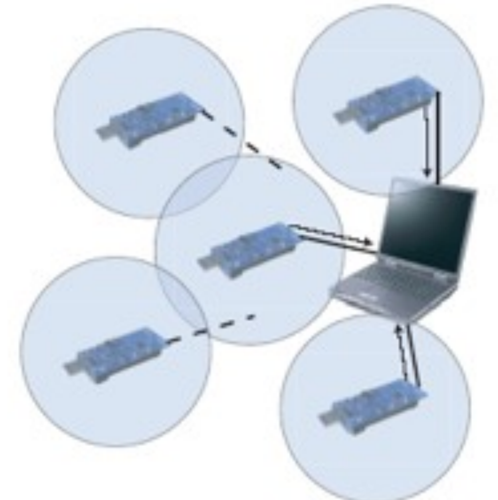
- MPC computes the optimal plan when new measurements arrive

- Optimal plan implemented by local controller if received in time, otherwise previous plan still kept

> Packet loss possible along both network links,
> delayed packets must be discarded (out-of-date data)

Challenges in Wireless MPC





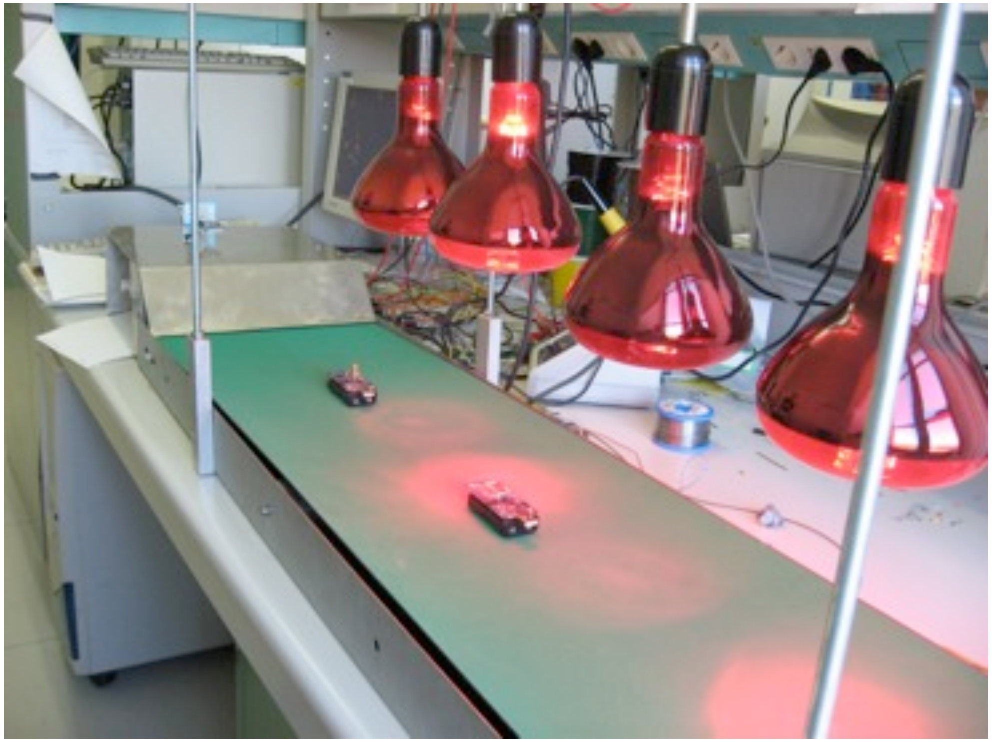
- Synchronization schemes must ensure correct prediction in spite of packet loss

- MPC algorithm must be robust w.r.t. packet loss
  $\Rightarrow$ stochastic hybrid MPC, robust hybrid MPC

- Wireless sensors must be interfaced to optimization tools

Demo Application in Wireless Automation

(Bemporad, Di Cairano, Henriksson)



(Automatic Control Lab, Univ. Siena)

Telos motes

- Telos motes provide wireless temperature feedback in Matlab

- Hybrid MPC algorithm adjust belt speed and coordinate linear motors (via Simulink/xPC-Target link)

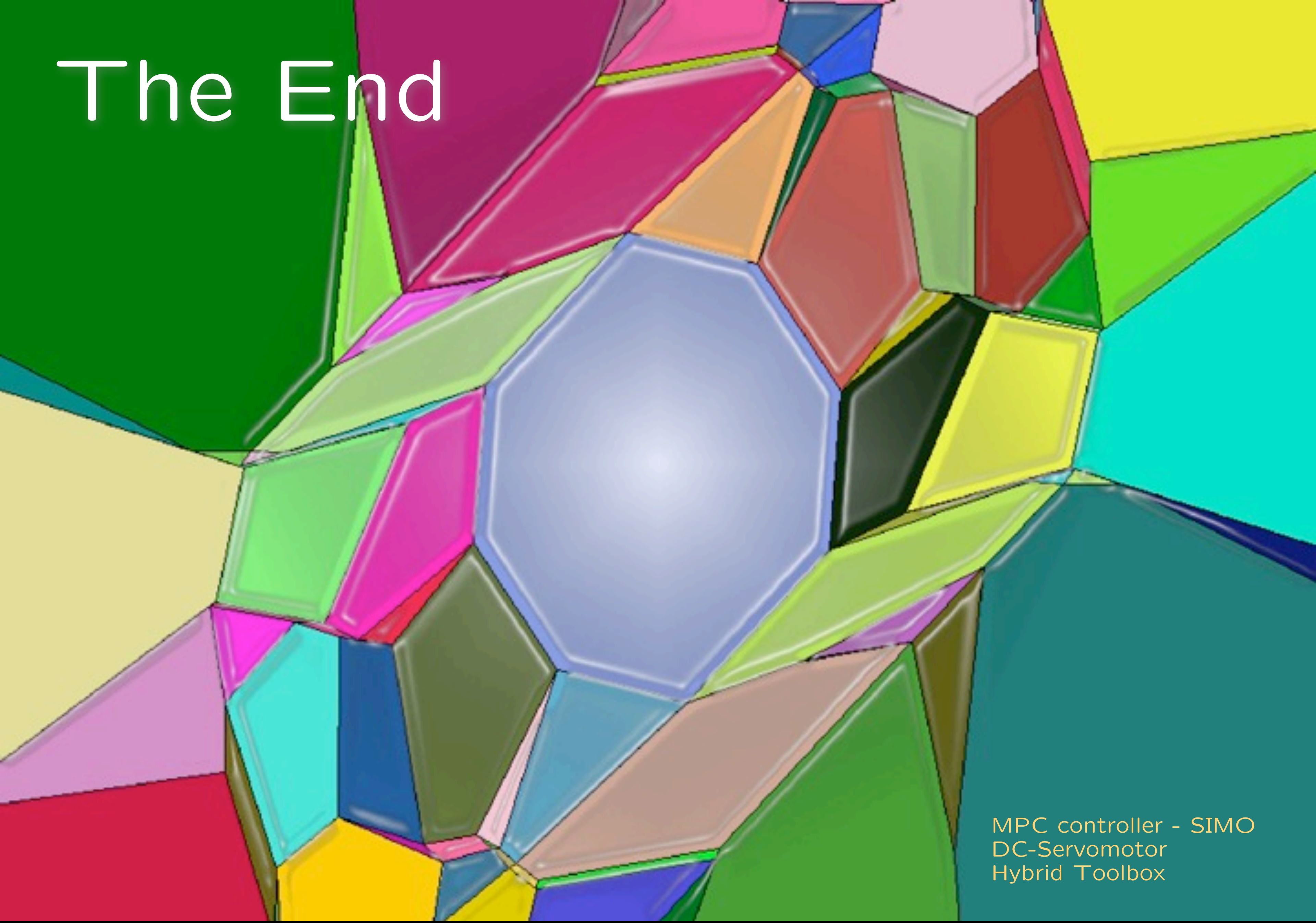
Acknowledgements

A. Alessio, M. Baotic, F. Borrelli, B. De Schutter, S. Di Cairano, G. Ferrari-Trecate, K. Fukuda, N. Giorgetti, M. Heemels, D. Hrovat, J. Julvez, I. Kolmanovski, M. Lazar, L. Ljung, D. Mignone, M. Morari, D. Munoz de la Pena, S. Paoletti, G. Ripaccioli, J. Roll, F.D. Torrisi

Announcements

- 10th "Hybrid Systems: Computation and Control" Conference (Pisa, Italy, April 3-5, 2006). **Submission: Oct. 9, 2006**
- 2nd PhD School on Hybrid Systems, Siena, 2007

The End



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