Hybrid Modeling, Analysis, and Optimization-based Control

Alberto Bemporad

http://cse.lab.imtlucca.it/~bemporad

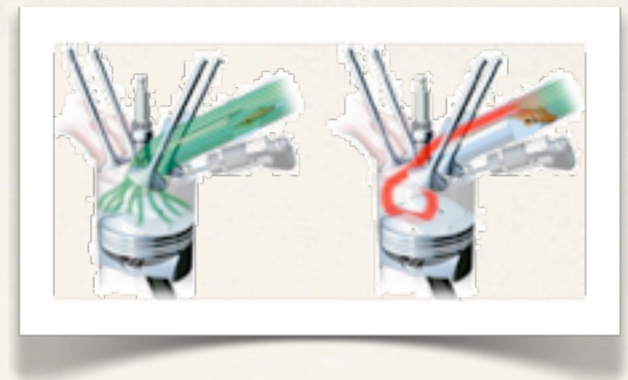
Institutions
Markets
Technologies
IMT
INSTITUTE
FOR ADVANCED
STUDIES
LUCCA

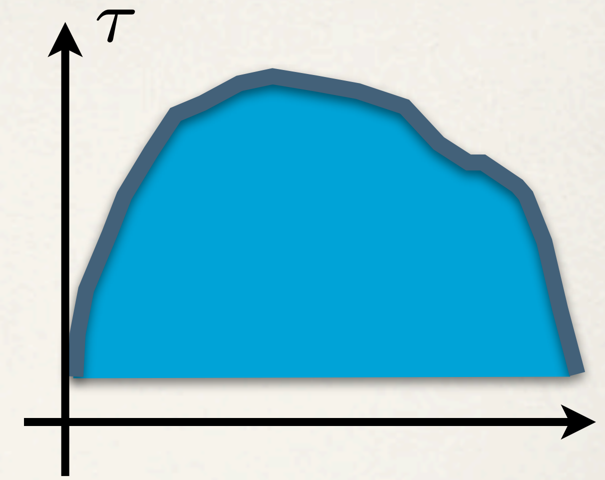
Eindhoven, June 6-8 2012

ADHS12

Motivation: typical control problems in practice

- **Nonlinear** and **switching dynamics**





- **Constraints** on manipulated and state variables

- **Optimal closed-loop performance**

- Solid **theoretical certification** (stability guarantees)

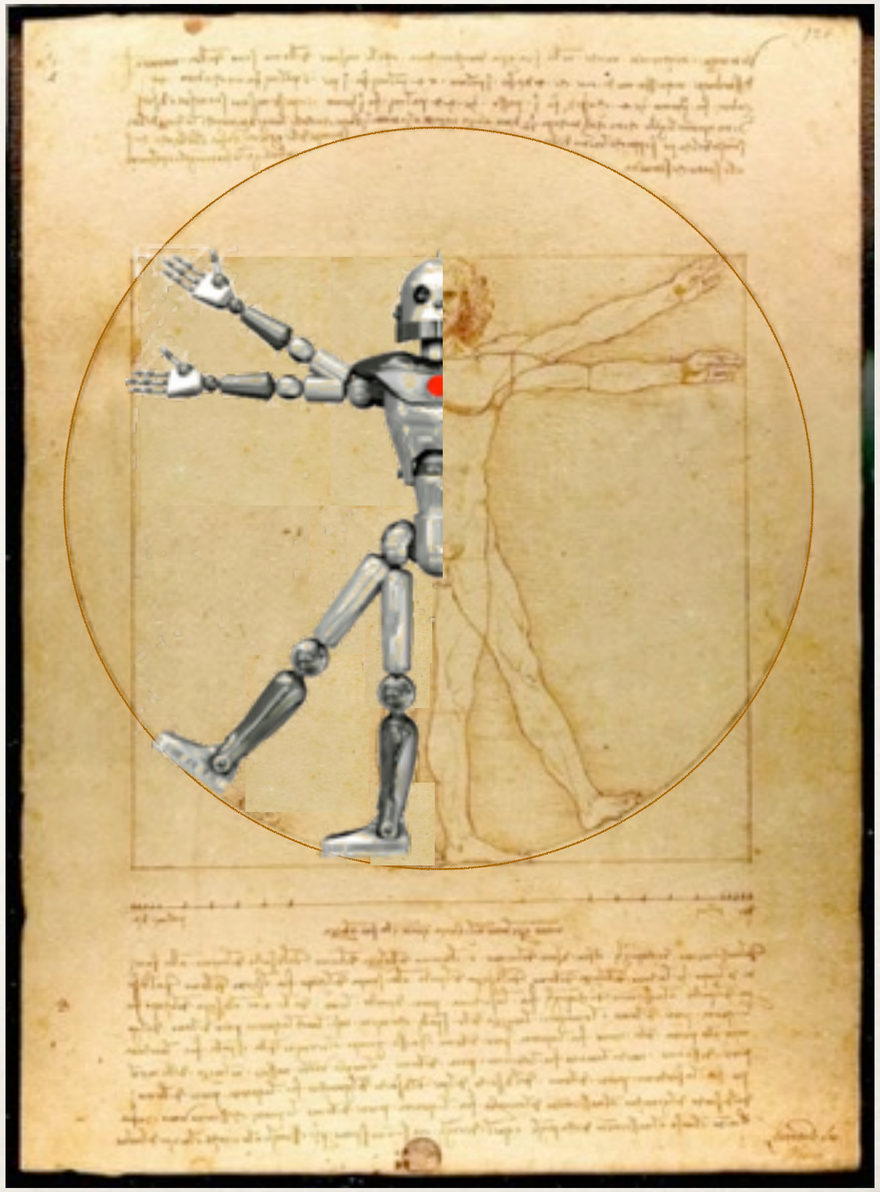
- **Fast dynamics** ⇒ **short sample time** (e.g. milliseconds)

- **Uncertainty**

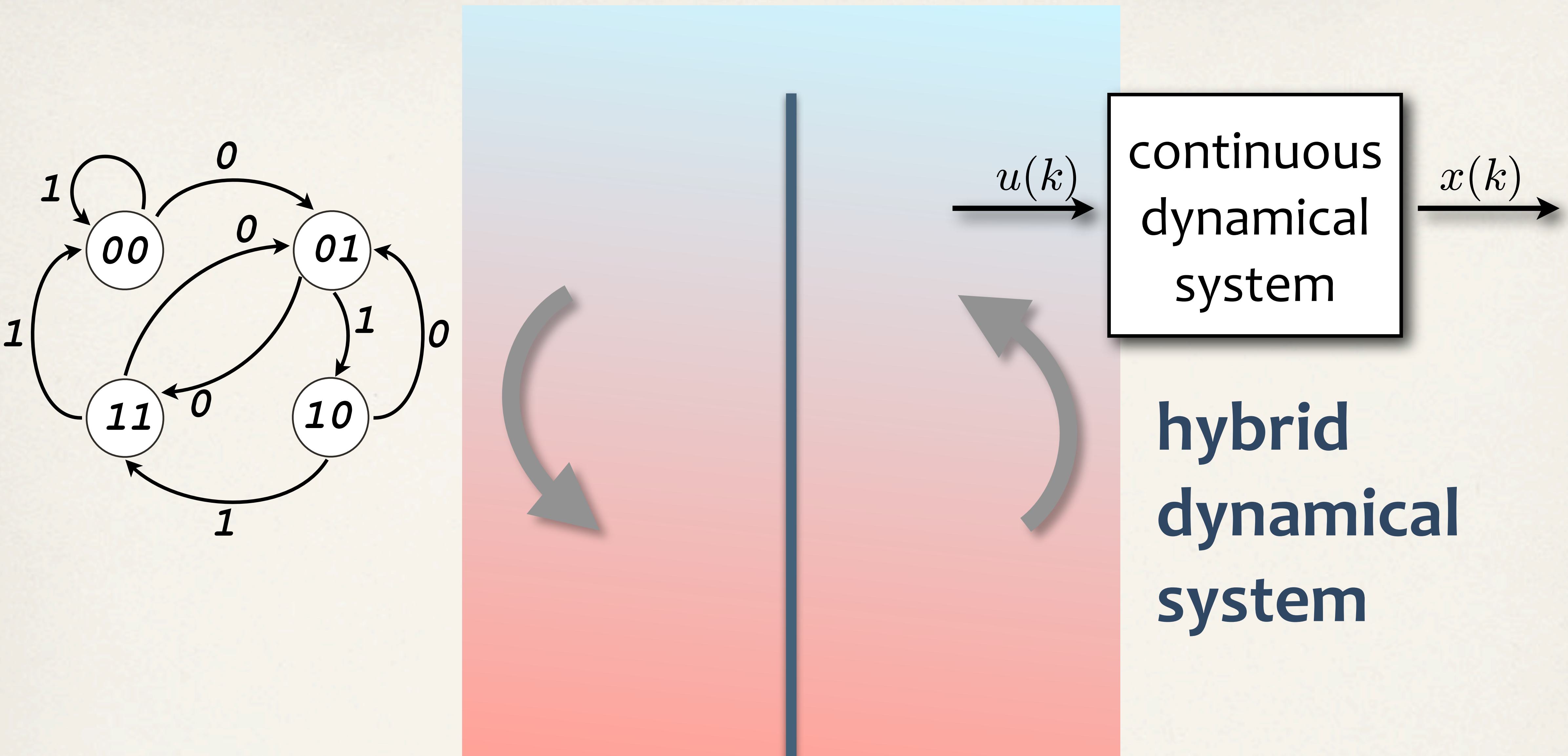
Hybrid systems & optimization-based control

Outline

- Modeling hybrid dynamical systems
- Optimization-based control
  - Model Predictive Control (MPC)
  - Piecewise Affine (PWA) control
- Applications



Hybrid dynamical systems



- Variables are **discrete-valued**

  $x \in \{0,1\}^{n_b}, \;\; u \in \{0,1\}^{m_b}$

- Dynamics = **finite state machine**

- **Logic** constraints

- Variables are **real-valued**

  $x \in \mathbb{R}^{n_c}, \;\; u \in \mathbb{R}^{m_c}$

- **Difference/differential equations**

- **Linear inequality** constraints

Hybrid dynamical systems

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.

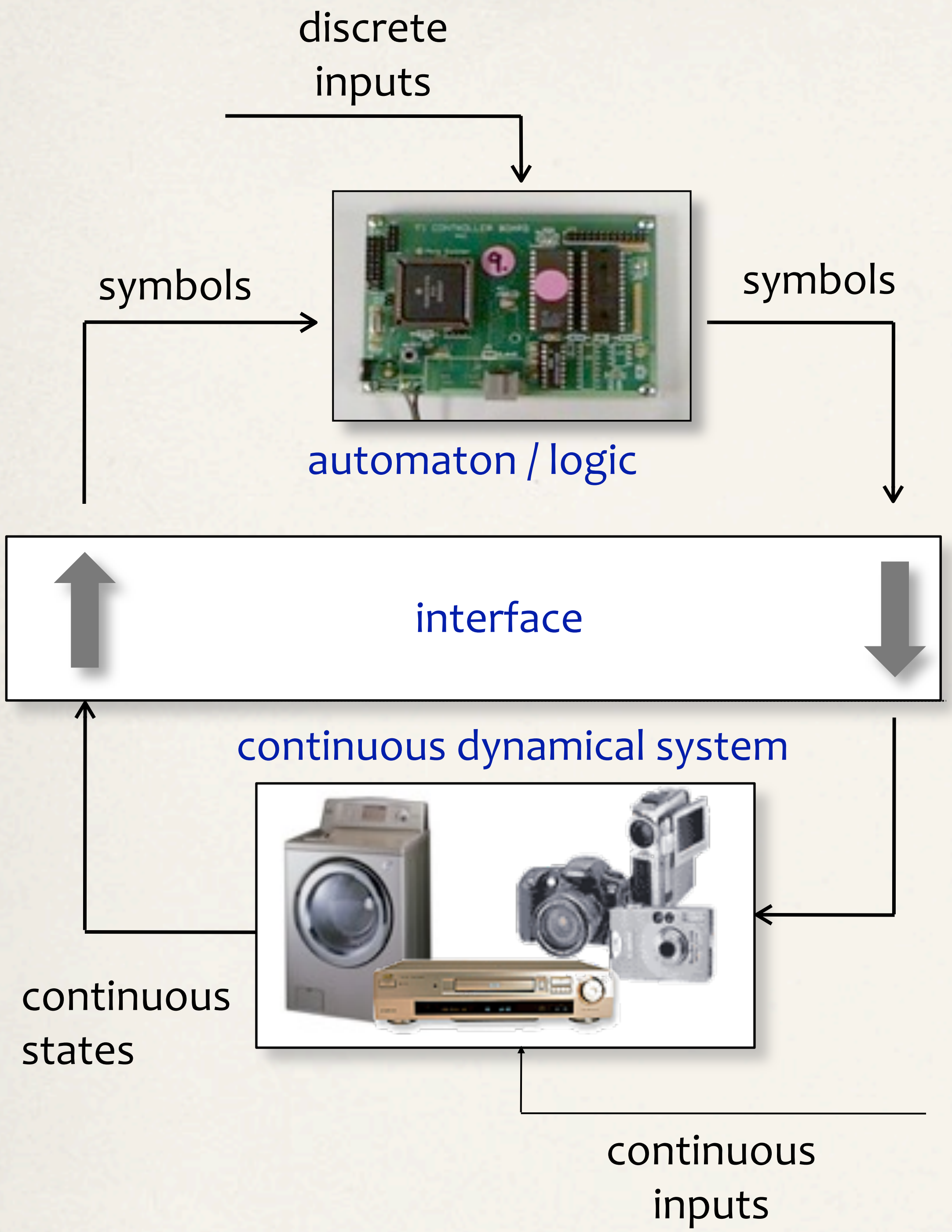
The objective of this paper is to give a precise description of such systems, to define their dynamics, to formulate the problem of their optimum control, to introduce the notion of well-behaved solution, and to state necessary conditions for optimality (the jump conditions).

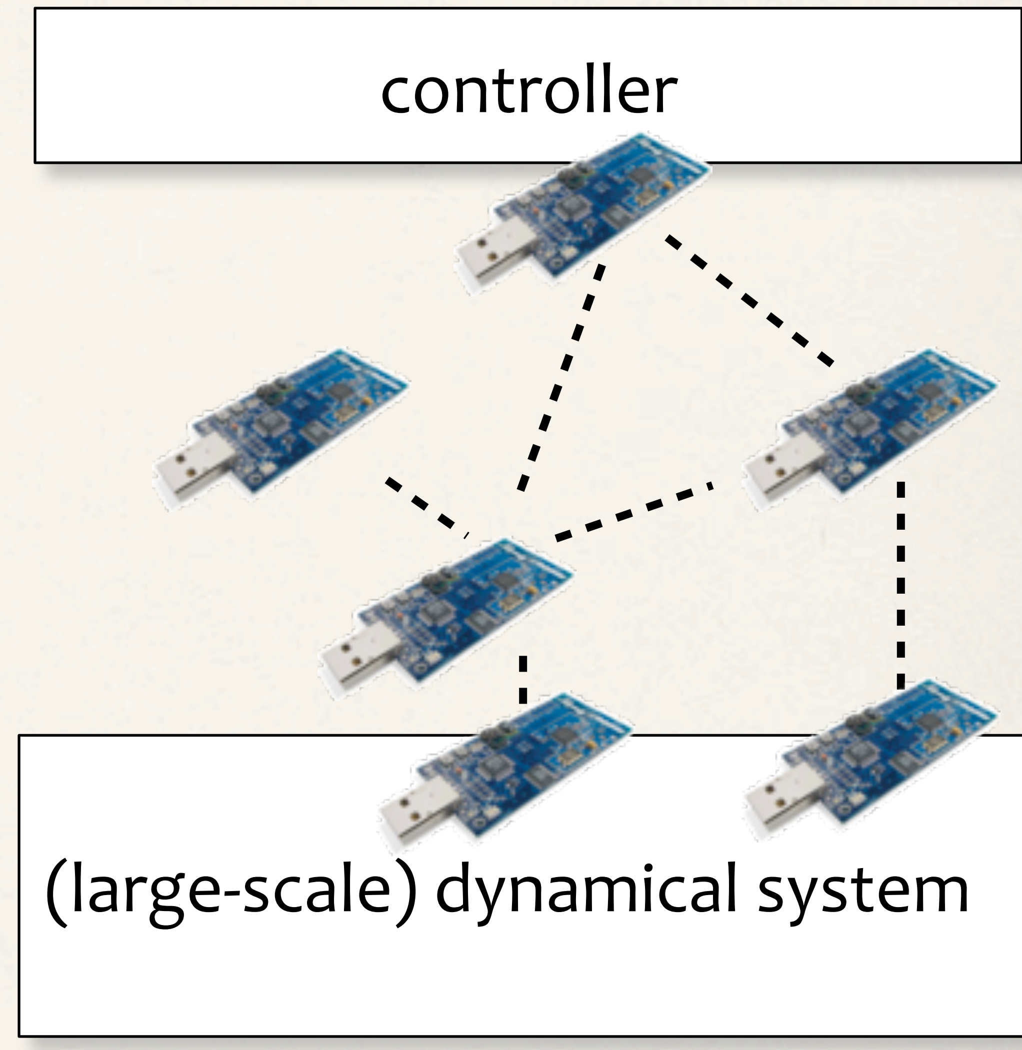
A Class of Hybrid Systems

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

Sontag (1981)
Lee & Arapostathis (1987)
Ezzine & Haddad (1989)
Gollu & Varaiya (1989)
Peleties & DeCarlo (1991)
Nerode & Kohn (1993)
Brockett (1993)
Branicky (1994)
...many others (≥1995)

Technological push for studying hybrid systems





embedded systems

networked control systems

>1995

>2005

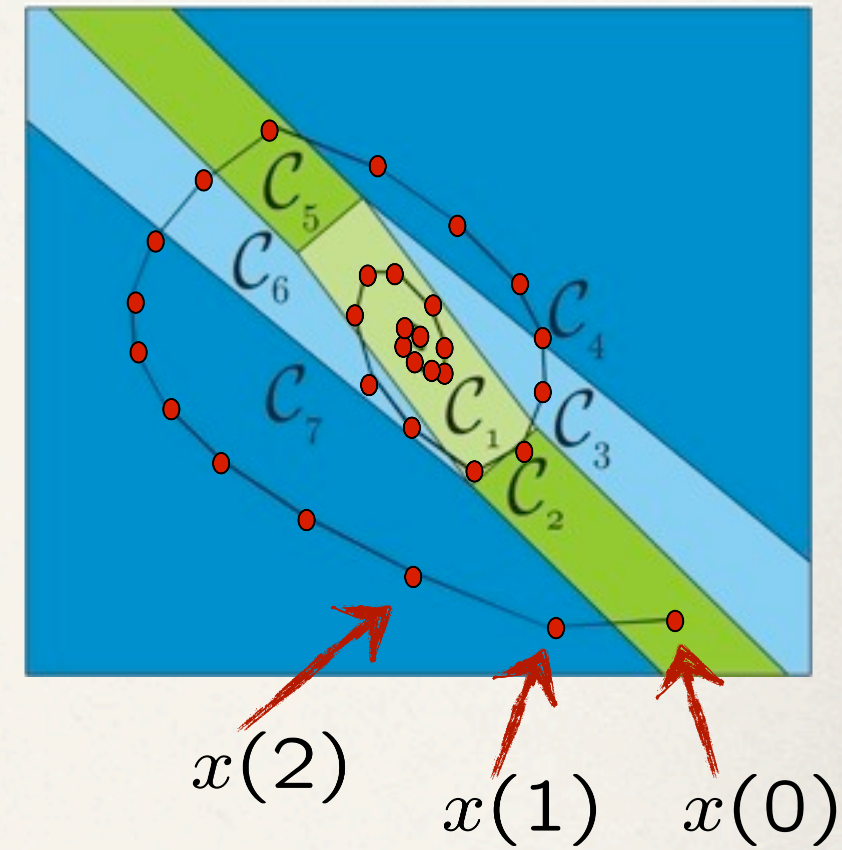
Computation-oriented models of hybrid systems

Piecewise affine systems

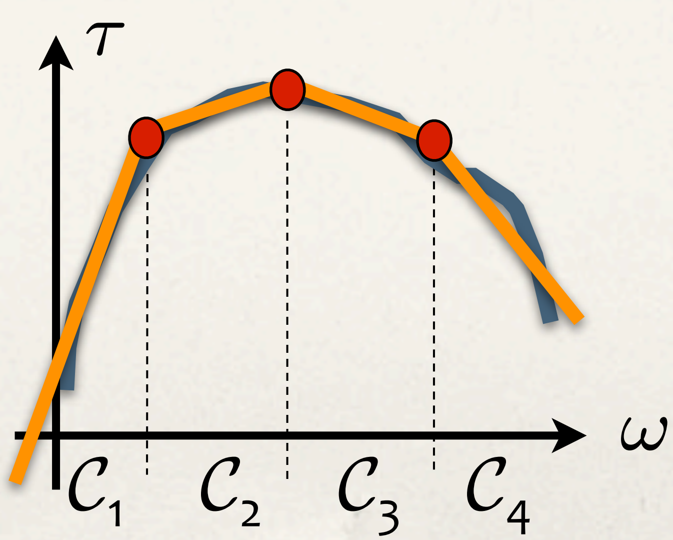
$$
\begin{aligned}
x(k+1) &= A_{i(k)}x(k) + B_{i(k)}u(k) + f_{i(k)} \\
y(k) &= C_{i(k)}x(k) + D_{i(k)}u(k) + g_{i(k)} \\[1em]
i(k) \ \text{s.t.} &\quad H_{i(k)}x(k) + J_{i(k)}u(k) \leq K_{i(k)}
\end{aligned}
$$

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

state+input space

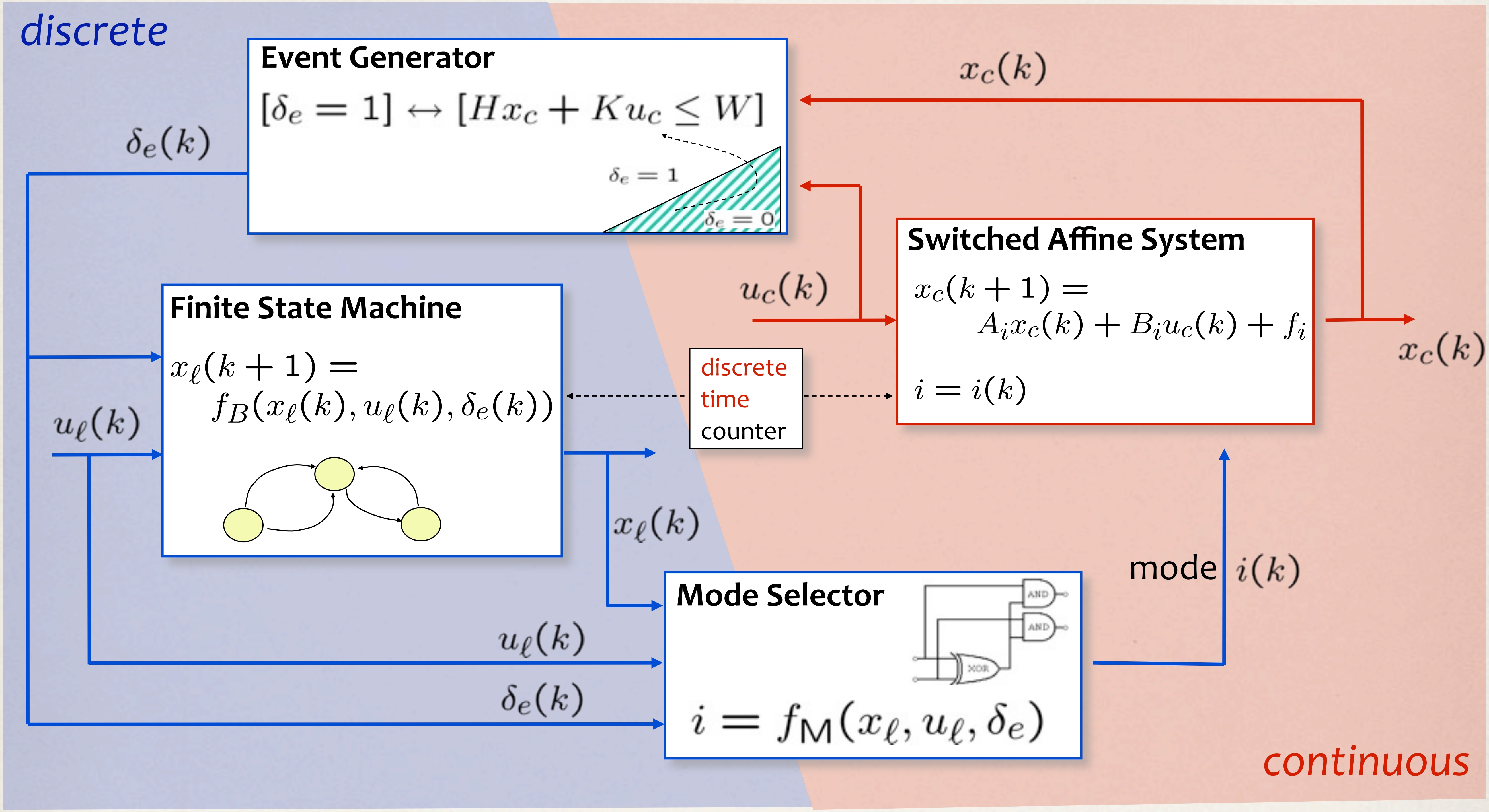


Can approximate nonlinear and/or discontinuous dynamics arbitrarily well



Discrete Hybrid Automaton (DHA)

(Torrisi, Bemporad, 2004)



$x_\ell \in \{0,1\}^{n_b}$ = binary state

$u_\ell \in \{0,1\}^{m_b}$ = binary input

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

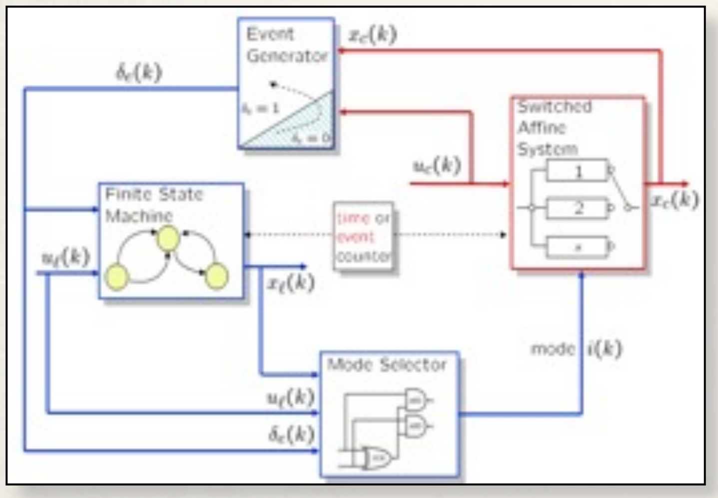
$x_c \in \mathbb{R}^{n_c}$ = real-valued state

$u_c \in \mathbb{R}^{m_c}$ = real-valued input

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

Mixed Logical Dynamical (MLD) systems

- Any **logic formula** involving Boolean variables can be translated into a set of **integer linear (in)equalities** (Raman, Grossmann, 1991)



Example: $\delta_1 \text{ OR } \delta_2 = \text{TRUE}$ $\Longleftrightarrow$ $\delta_1 + \delta_2 \geq 1$

HYSDEL
modeling language
(Torrisi, Bemporad, 2004)

Mixed Logical Dynamical (MLD) system

$$\begin{cases} x_{k+1} = Ax_k + B_1 u_k + B_2 \delta_k + B_3 z_k + B_5 \\ y_k = Cx_k + D_1 u_k + D_2 \delta_k + D_3 z_k + D_5 \\ E_2 \delta_k + E_3 z_k \leq E_4 x_k + E_1 u_k + E_5 \end{cases}$$

(Bemporad, Morari 1999)

- x,u,y,z,δ contain **mixed-integer** (=**real** and **binary**) components

Equivalence of hybrid models

- **MLD** systems and **PWA** systems are equivalent

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

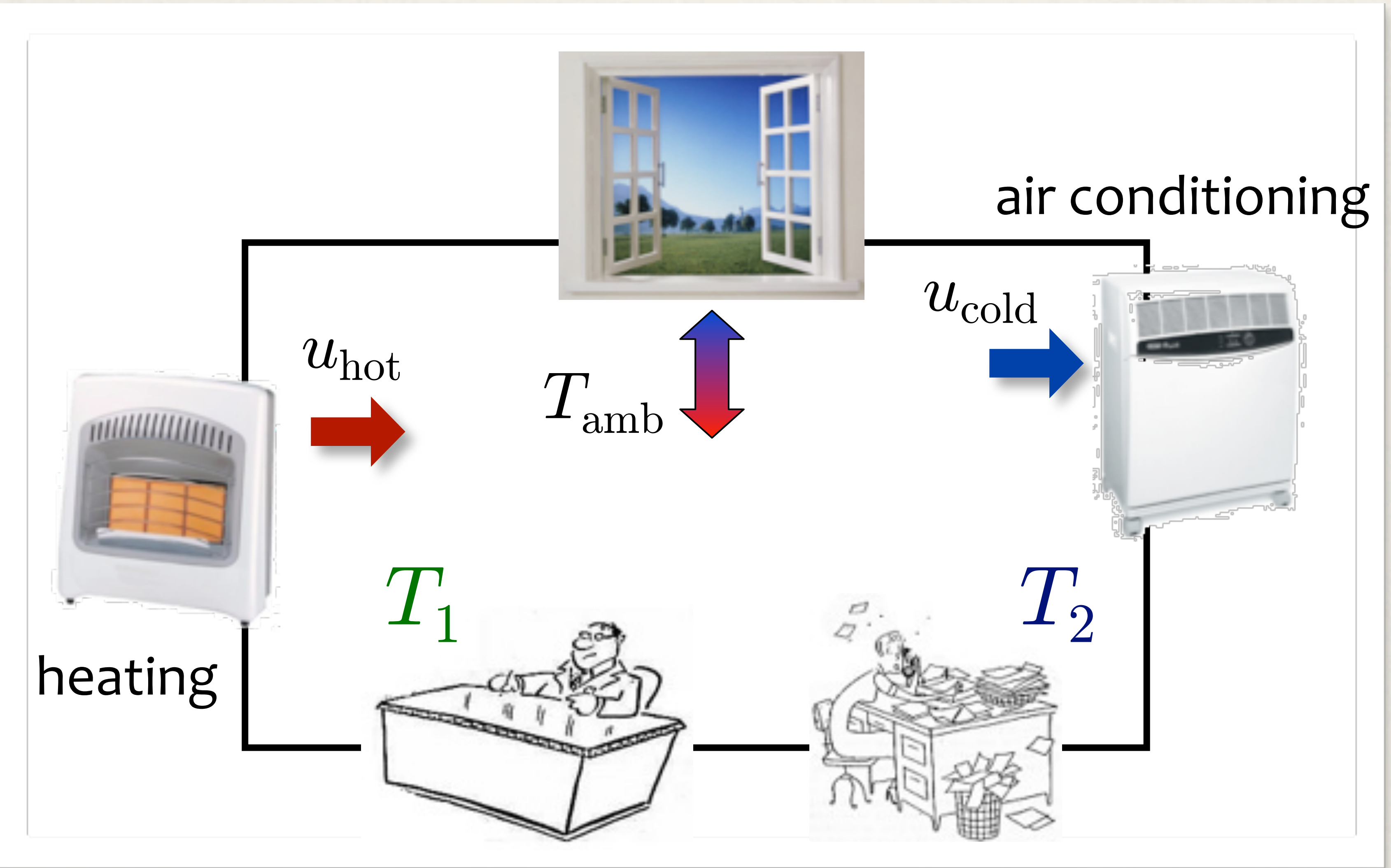
- Efficient algorithms for **converting** MLD models into PWA form exist

(Bemporad, *IEEE TAC, 2004*) (Geyer, Torrisi, Morari, HSCC, 2003)

- **Further equivalences** exist with other classes of hybrid dynamical systems, such as Linear Complementarity (**LC**) systems

(Heemels, De Schutter, Bemporad, *Automatica,* 2001)

Example: Room temperature control



discrete dynamics

- #1=cold ➞ heater=on
- #2=cold ➞ heater=on **unless** #1=hot
- A/C activation has similar rules

continuous dynamics

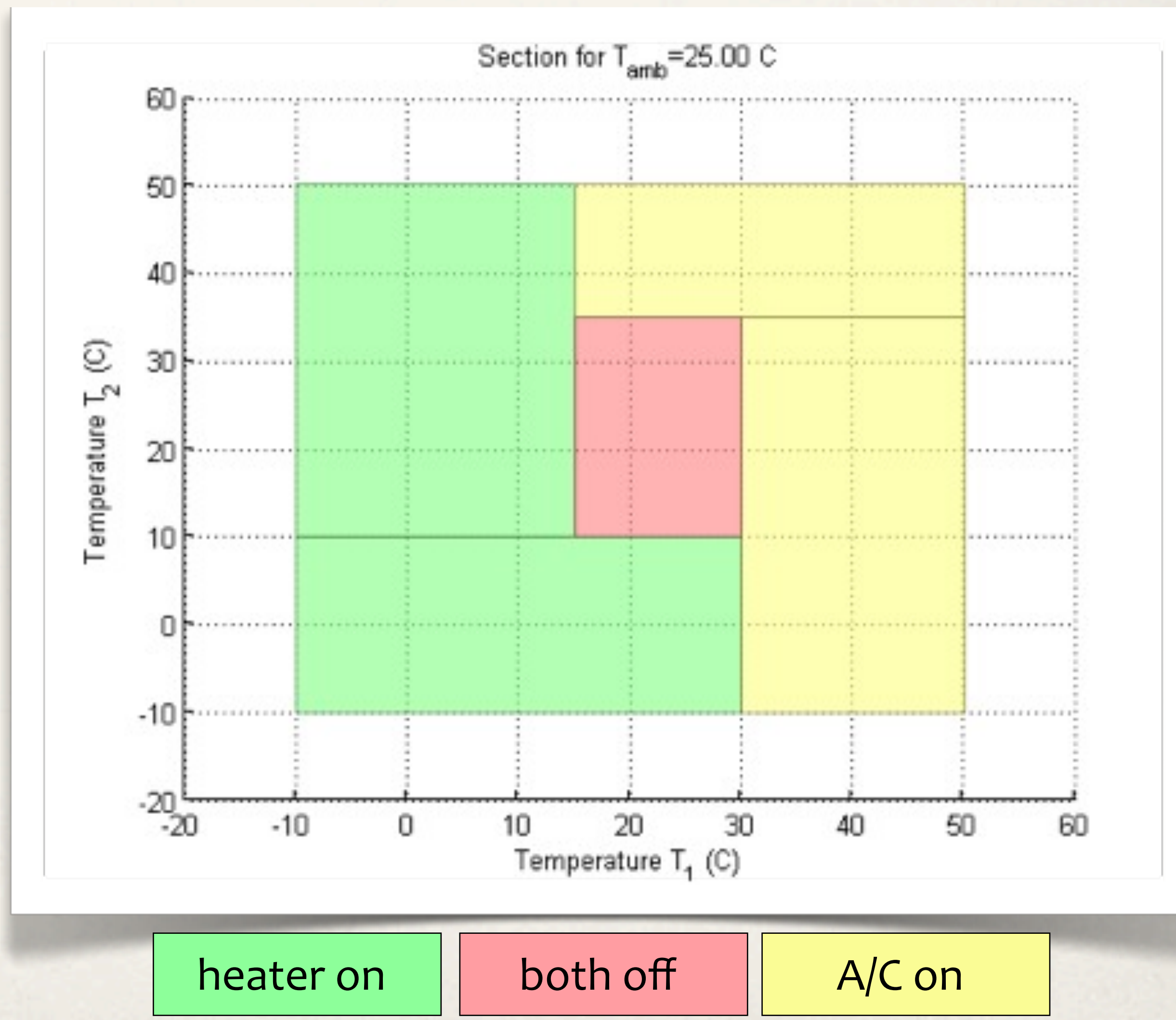
$$\frac{dT_i}{dt} = -\alpha_i(T_i - T_{\text{amb}}) + k_i(u_{\text{hot}} - u_{\text{cold}})$$

$$i = 1, 2$$

Example: Room temperature control

```
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));
        }
    }
}
```

- Model described in **HYSDEL** and converted to **MLD** form (**20 mixed-integer inequalities**)

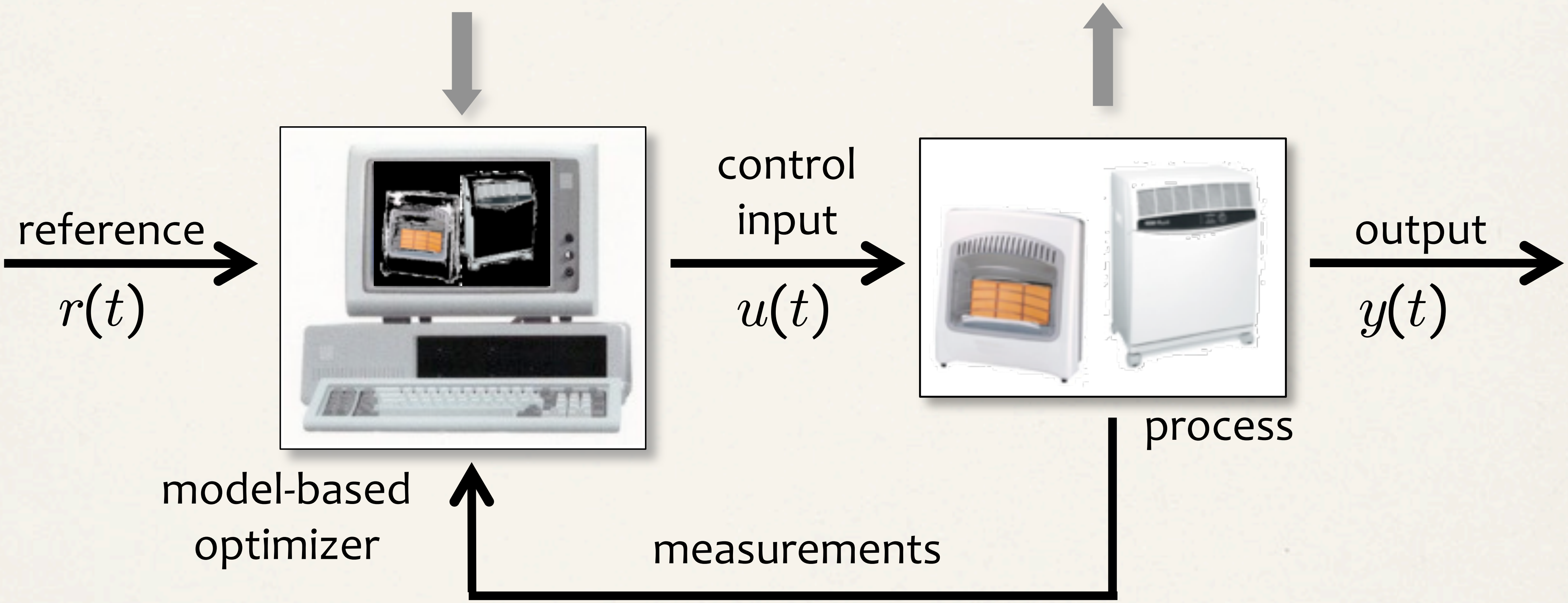


- Equivalent PWA model consists of **5 regions**

Are linear discrete-time hybrid models useful for control design ?

Model Predictive Control (MPC)

$$\left\{\begin{array}{rcl} x_{k+1} & = & Ax_k + B_1u_k + B_2\delta_k + B_3z_k + B_5 \\ y_k & = & Cx_k + D_1u_k + D_2\delta_k + D_3z_k + D_5 \\ E_2\delta_k & + & E_3z_k \leq E_4x_k + E_1u_k + E_5 \end{array}\right.$$

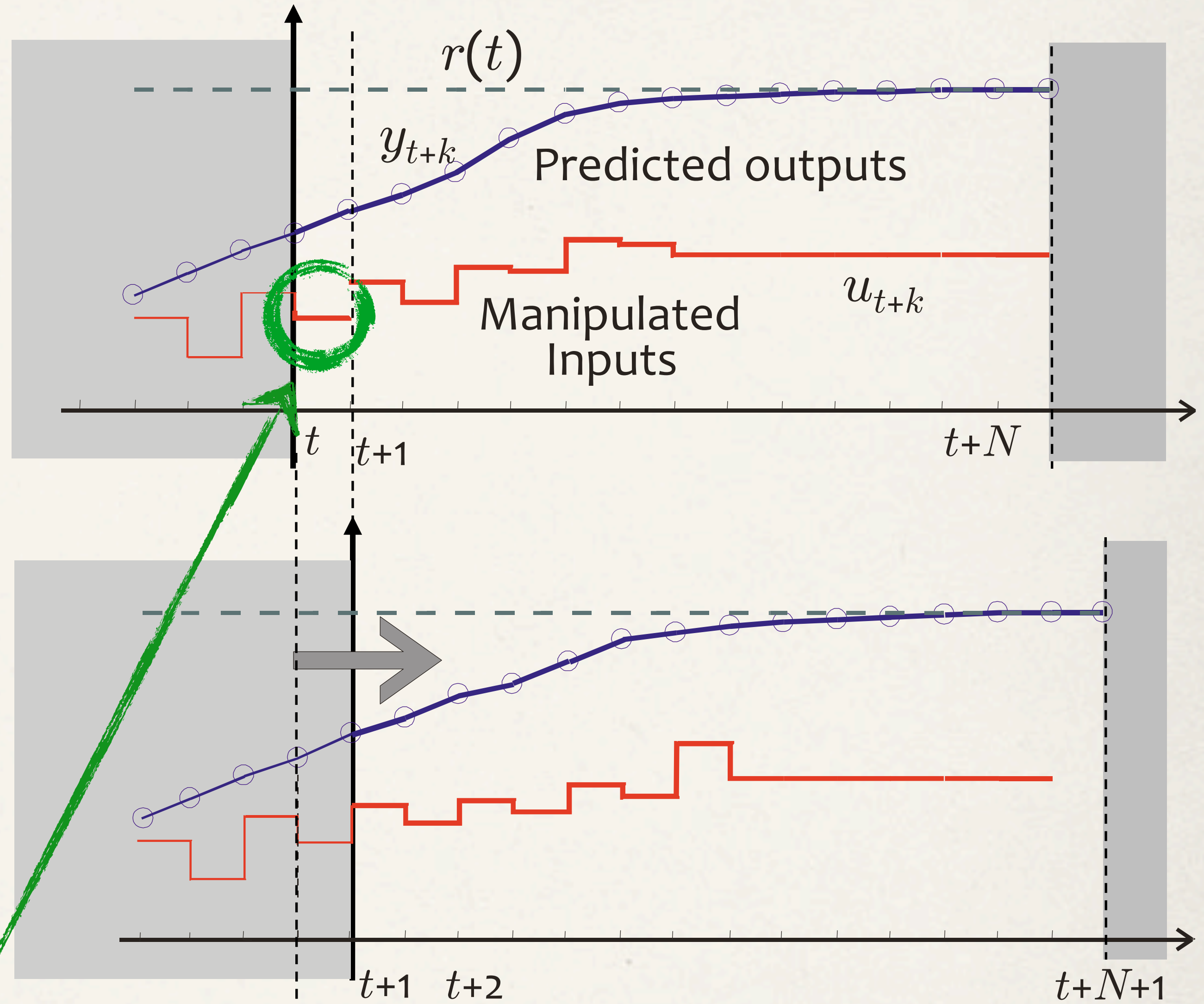


Use a hybrid dynamical ***model*** *of the process to* ***predict*** *its future evolution and choose the "best"* ***control*** *action*

MPC algorithm

- **At time** t, solve an **optimal control** problem over a future horizon of N steps

$$\begin{aligned}
\min \quad & \sum_{k=0}^{N-1} \ell(y_{t+k} - r_{t+k}, u_{t+k}) \\
\text{s.t.} \quad & x_{t+k+1} = f(x_{t+k}, u_{t+k}) \\
& y_{t+k} = g(x_{t+k}, u_{t+k}) \\
& \text{constraints on } u_{t+k}, x_{t+k}, y_{t+k} \\
& x_t = x(t)
\end{aligned}$$



- Apply only the first optimal move $u^*(t)$, trash the rest of the optimal sequence
- **At time** t**+1:** Get new measurements, repeat the optimization. And so on ...

MPC transforms open-loop optimal control into a feedback control law

MPC of linear systems

linear model
$$\left\{\begin{array}{rcl} x_{k+1} &=& Ax_k + Bu_k \\ y_k &=& Cx_k \end{array}\right. \qquad x_0 = x(t)$$

performance index
$$\min_U x'_N P x_N + \sum_{k=0}^{N-1} x'_k Q x_k + u'_k R u_k \qquad \begin{array}{rcl} R &=& R' \succ 0 \\ Q &=& Q' \succeq 0 \\ P &=& P' \succeq 0 \end{array}$$

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

linear constraints
$$\left\{\begin{array}{rcccl} u_{\min} &\leq& u_k &\leq& u_{\max} \\ y_{\min} &\leq& Cx_k &\leq& y_{\max} \end{array}\right.$$

MPC implemented by solving a (convex) Quadratic Program (QP)

Routinely used in the process industries

Honeywell aspentech ABB

MPC of hybrid systems

hybrid
MLD model

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

performance index

$$\min_U x'_N P x_N + \sum_{k=0}^{N-1} x'_k Q x_k + u'_k R u_k$$

$$\begin{aligned} R &= R' \succ 0 \\ Q &= Q' \succeq 0 \\ P &= P' \succeq 0 \end{aligned}$$

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

$$\xi = \begin{bmatrix} u_0 \\ \vdots \\ u_{N-1} \\ \delta_0 \\ \vdots \\ \delta_{N-1} \\ z_0 \\ \vdots \\ z_{N-1} \end{bmatrix}$$

linear and logic constraints

MPC implemented by solving a **Mixed-Integer Quadratic Program (MIQP)**

Alternative: Mixed-Integer Linear Program (**MILP**) formulation

What are the stability properties of hybrid MPC ?

Closed-loop properties

- Convergence result

Assume proper choice of weights and constraints on terminal state.
If MPC optimization problem is feasible at $t=0$, then:

- MPC problem is also **feasible** for all $t>0$
- The closed-loop system (MLD system + hybrid MPC controller) **converges asymptotically** to the equilibrium

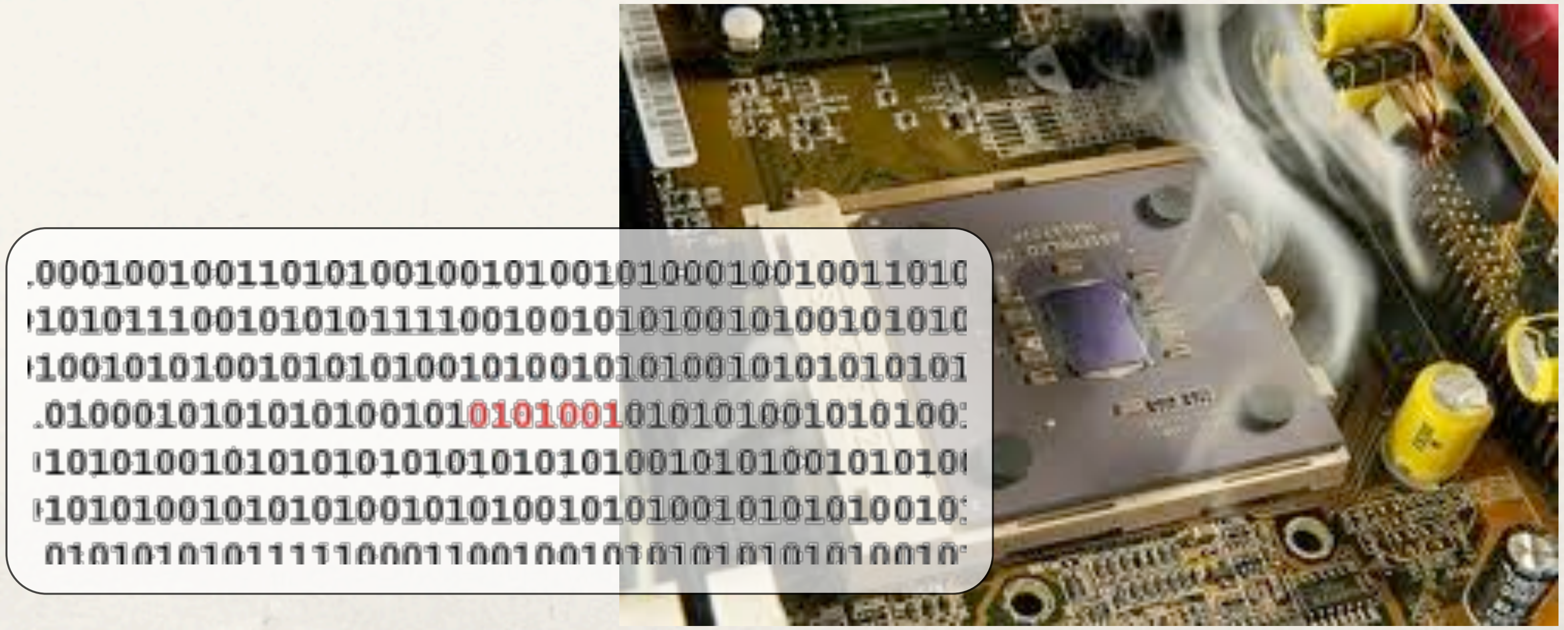
(Bemporad, Morari, 1999)

- Stability result

Lyapunov asymptotic stability and **exponential stability** can be guaranteed by choosing a proper terminal cost and constraint set

(Lazar, Heemels, Weiland, Bemporad, 2006)

How about computational load ?



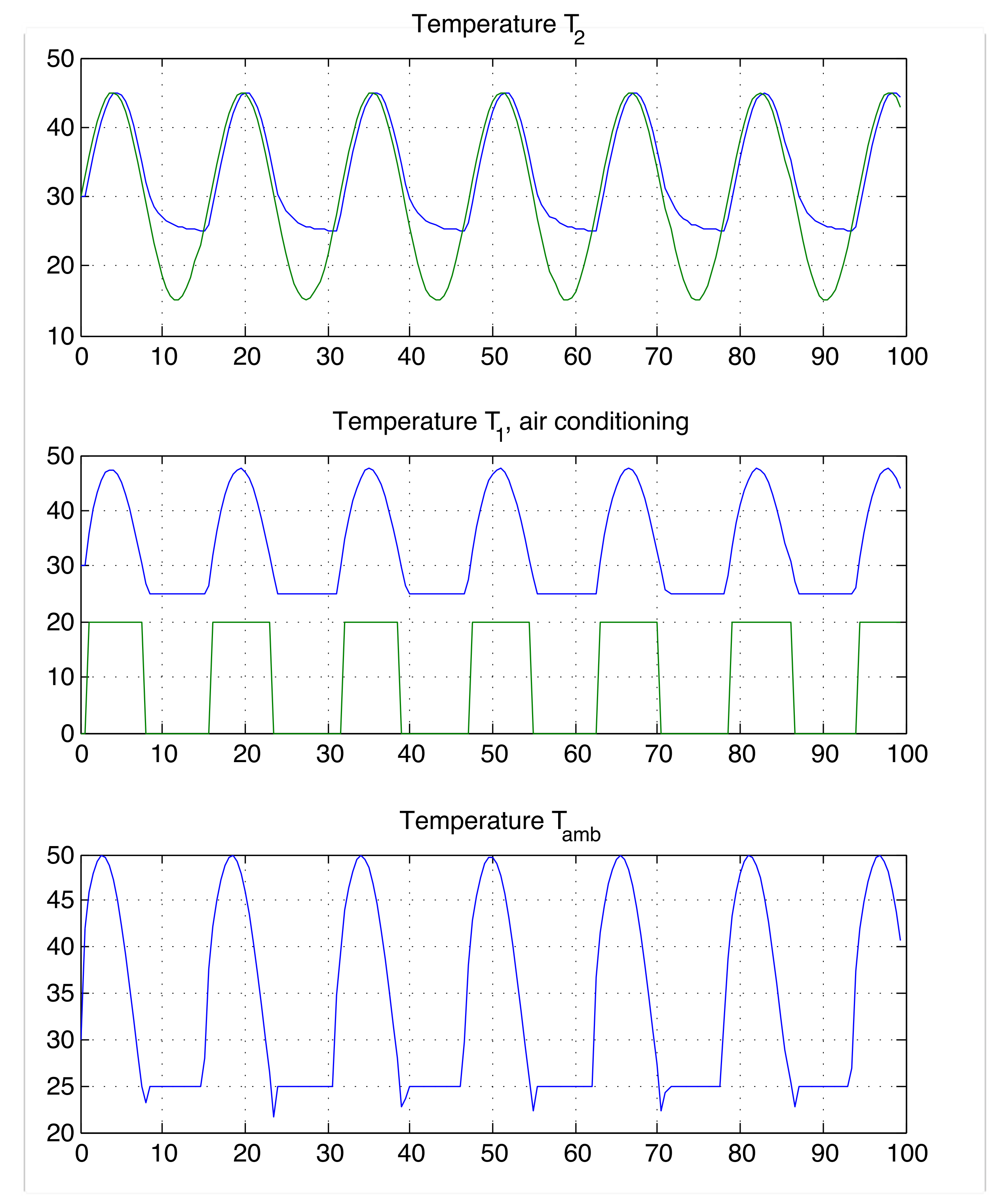
Example: Room temperature control

- MPC problem formulation

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

- MPC optimization problem (MILP)

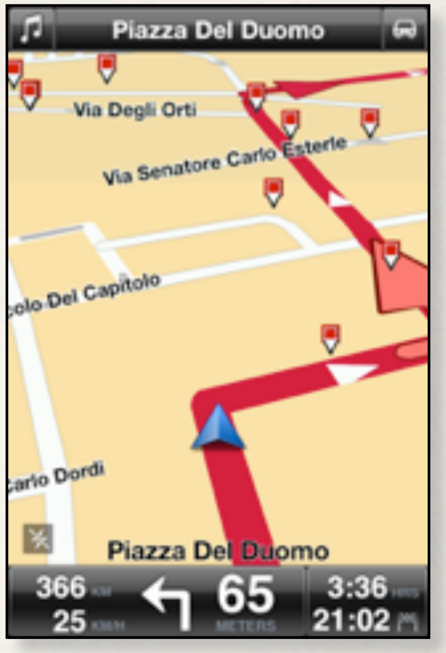
```
20 optimization variables
   (8 continuous, 12 binary)

46 mixed-integer linear inequalities
```



CPU time = **1.3 ms** per time step

(IBM CPLEX 11.2, this Mac)

Pros and cons of on-line optimization



PROS

✓ Continuously update the best decision, reacting to unexpected events (disturbances, faults, obstacles,...)

✓ Excellent QP and MILP/MIQP solvers exist today (http://plato.la.asu.edu/bench.html)

CONS

✖ Computation time may be too long for large-problems/fast sampling

✖ Requires relatively expensive hardware (such as a microprocessor)

✖ Software complexity: solver code must be embedded in the control code

✖ Real-time: worst-case CPU time often hard to estimate

Explicit model predictive control

$$
\begin{array}{ll}
\min\limits_{U} & \frac{1}{2}U'HU + \mathbf{x}'(t)F'U + \cancel{\frac{1}{2}x'(t)Yx(t)} \\
\text{subj. to} & GU \leq W + S\mathbf{x}(t)
\end{array}
$$

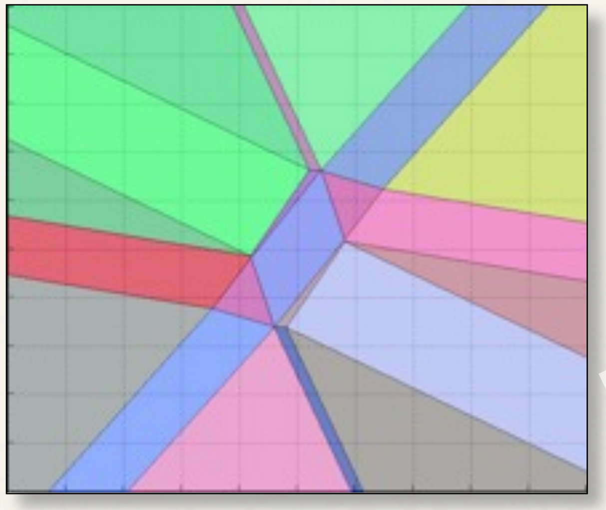
Idea: solve the QP **for all $x(t)$** within a given range of $\mathbb{R}^n$ **off-line**

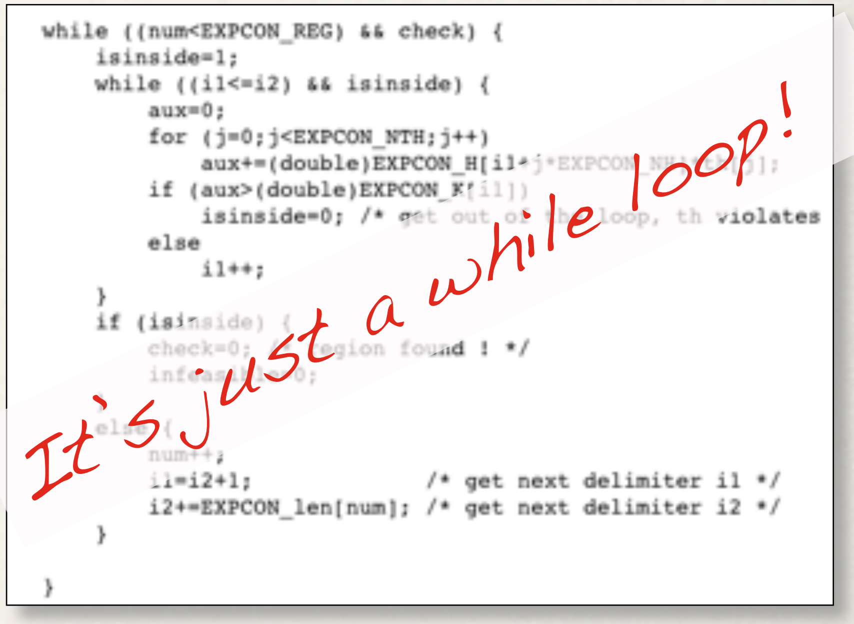
→ multi-parametric programming problem

Linear MPC is a **continuous** and **piecewise affine** control law !

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

(Bemporad, Morari, *et al.*, 2002)





```
while ((num<EXPCON_REG) && check) {
    isinside=1;
    while ((i1<=i2) && isinside) {
        aux=0;
        for (j=0;j<EXPCON_NTH;j++)
            aux+=(double)EXPCON_H[i1+j*EXPCON_NH]*th[j];
        if (aux>(double)EXPCON_K[i1])
            isinside=0; /* get out of the loop, th violates
        else
            i1++;
    }
    if (isinside) {
        check=0; /* region found ! */
        infeasible=0;
    }
    else{
        num++;
        i1=i2+1;              /* get next delimiter i1 */
        i2+=EXPCON_len[num]; /* get next delimiter i2 */
    }

}
```


Hybrid MPC based on linear costs is a **piecewise affine** control law

(Borrelli, Baotic, Bemporad, Morari, 2005) (Mayne, Rakovic, 2002) (Alessio, Bemporad, ADHS 2006)

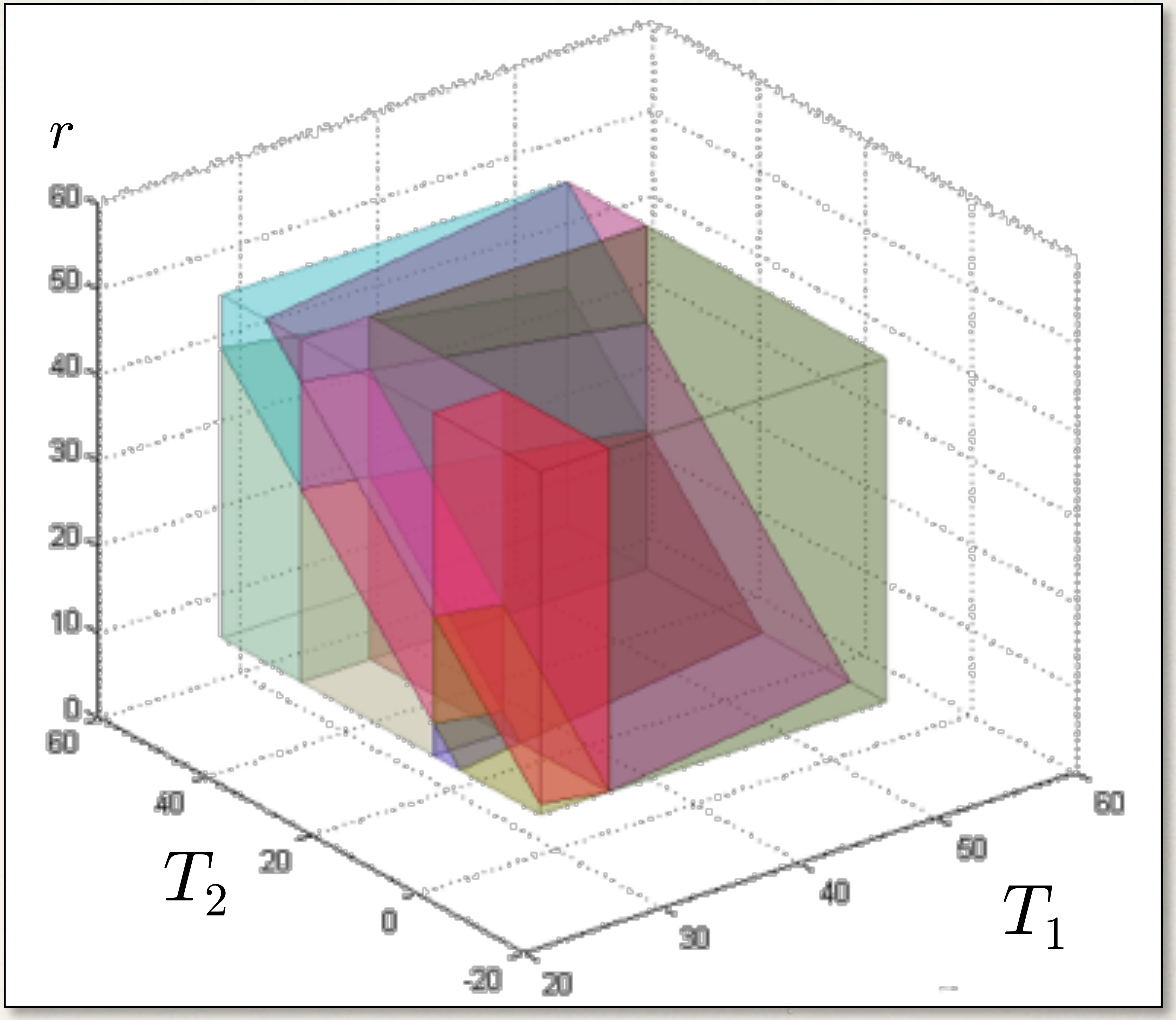
Example: Room temperature control

- Explicit solution

12 polyhedral regions and linear gains

↓

384 numbers to store in memory



CPU time = **0.8 ms**

(compiled C-code, this Mac)

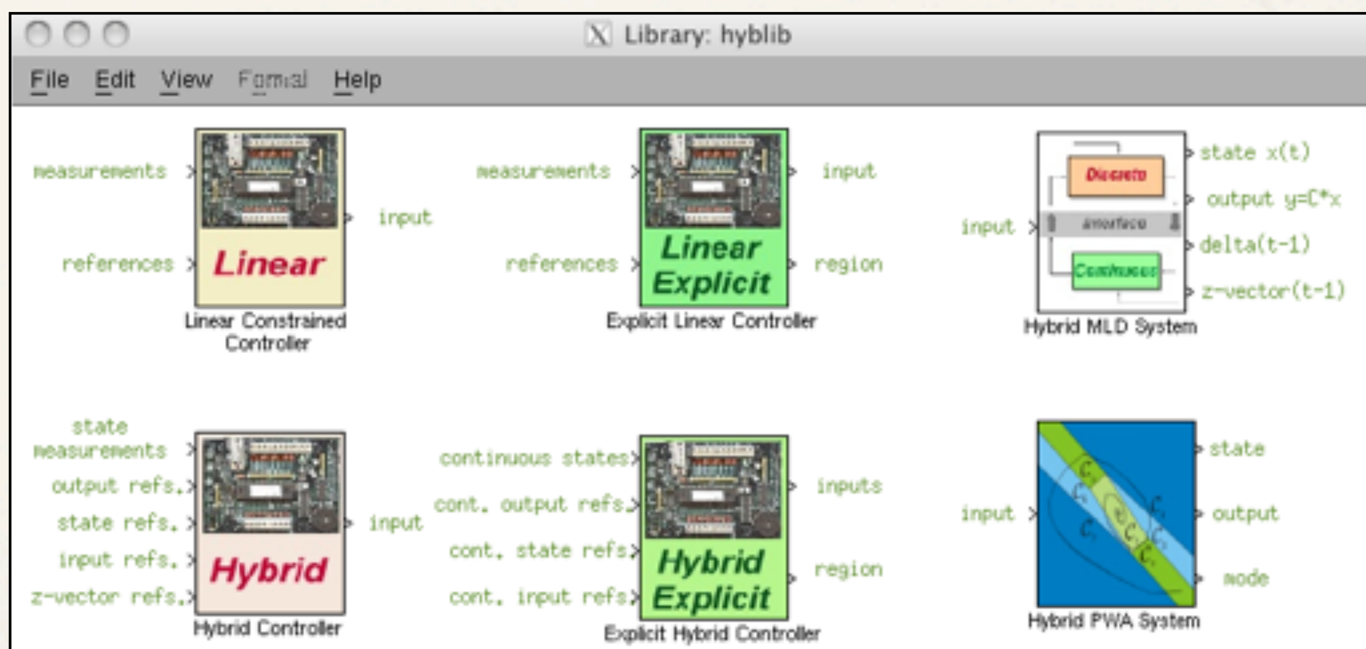
Note: explicit form does not change the control law at all !

Hybrid Toolbox for MATLAB

(Bemporad, 2003-2012)

Features:

- **Hybrid** models: design, simulation, verification
- MPC design for linear and hybrid systems
- Interfaces to several linear, quadratic and mixed-integer programming solvers
- **Explicit** MPC (via multi-parametric programming)
- Simulink library
- C-code generation



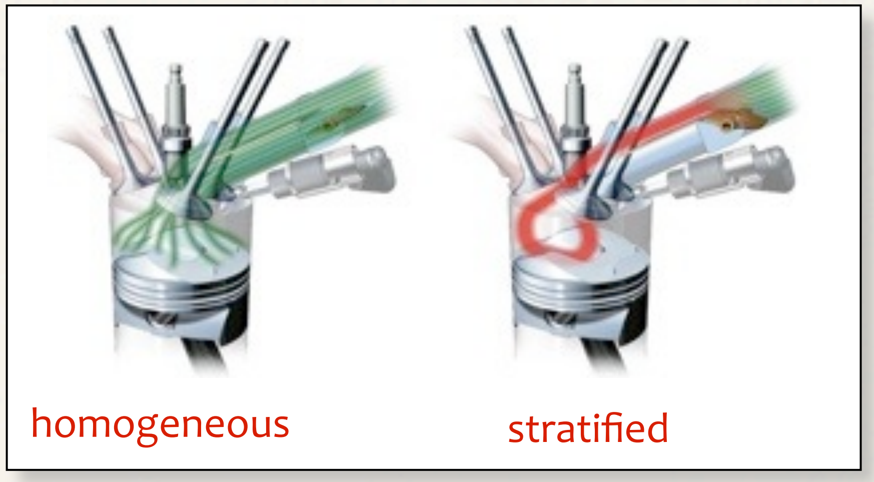
4000+ downloads

Supported by

http://cse.lab.imtlucca.it/~bemporad/hybrid/toolbox/

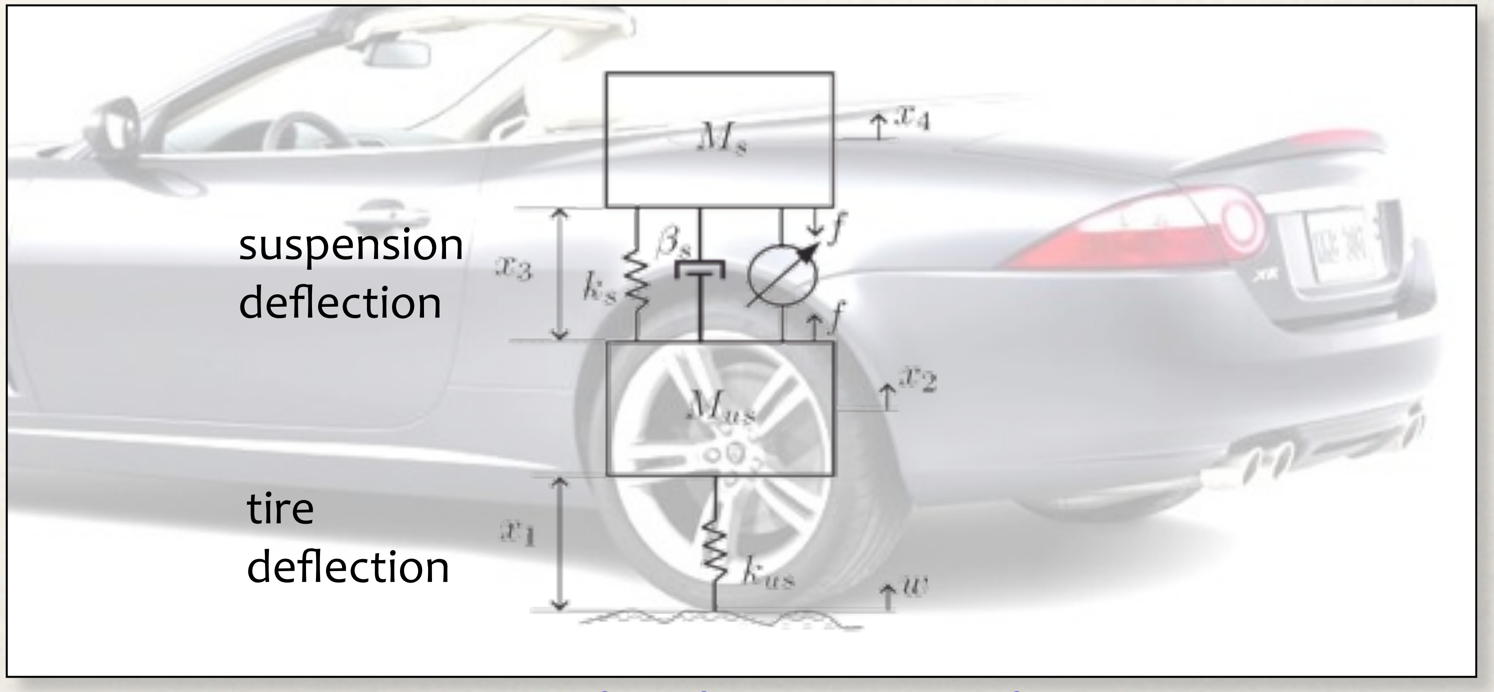
Is hybrid MPC (or its PWA form) applicable in practice ?

Automotive applications of MPC



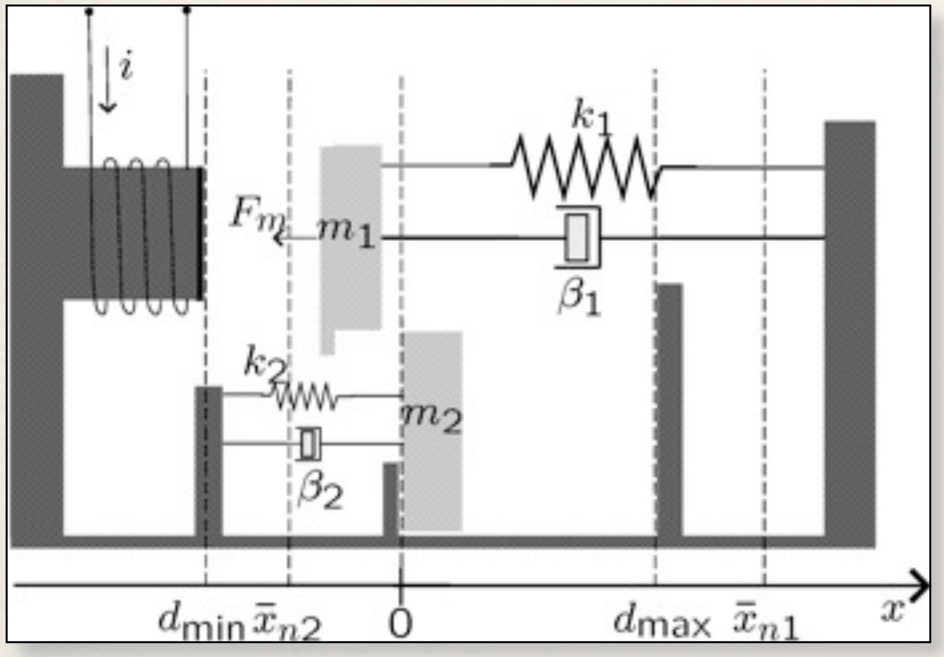
engine control

idle speed control



semiactive suspensions

Automotive applications of MPC



magnetic actuators

hybrid electric vehicles

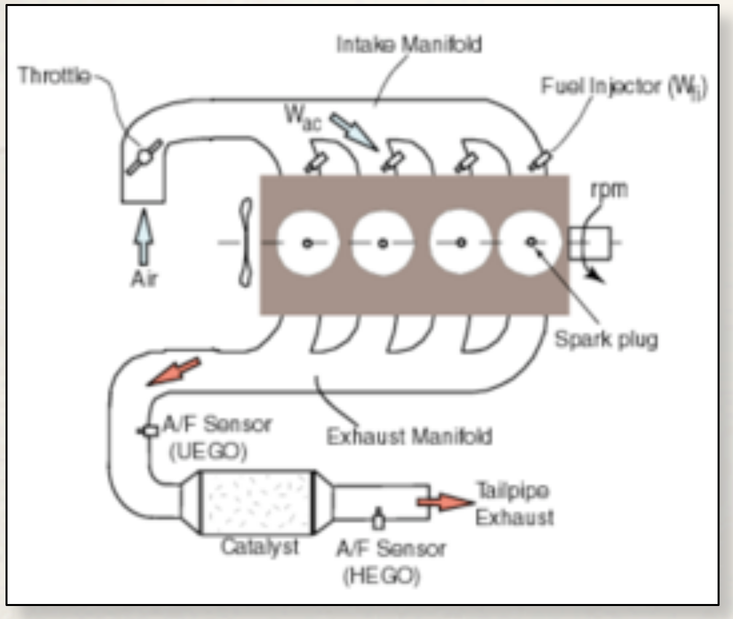
Ford

JAGUAR

DENSO

Institutions Markets Technologies IMT INSTITUTE FOR ADVANCED STUDIES LUCCA

CENTRO RICERCHE FIAT



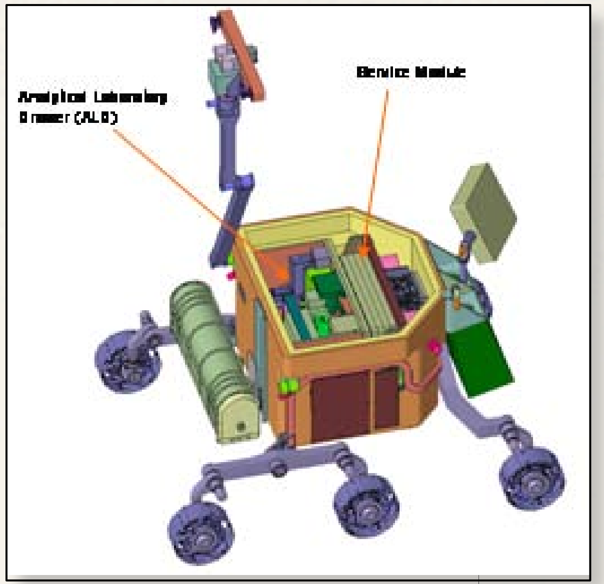
air-to-fuel ratio

active steering

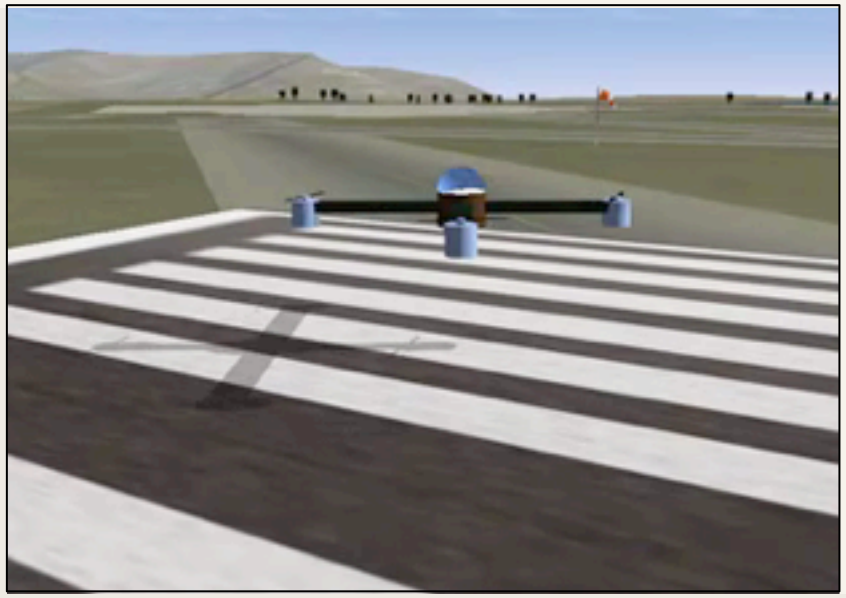
robotized gearbox

Aerospace applications of MPC

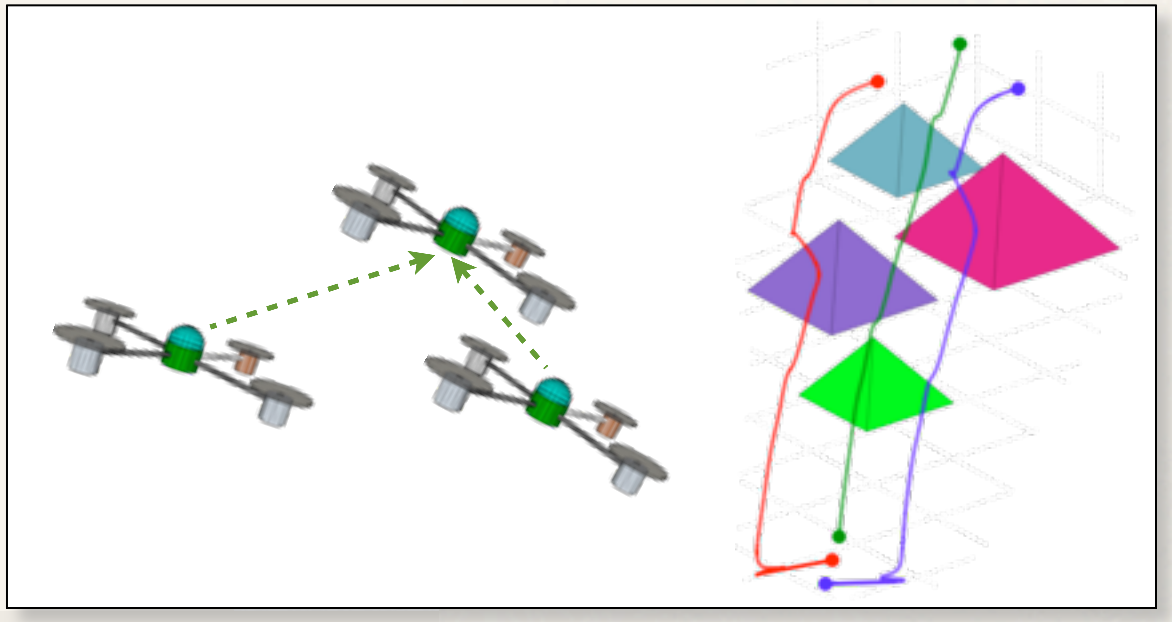
satellite attitude control



planetary rover locomotion



UAV guidance and control



formation flying

How to efficiently implement PWA controllers (like explicit MPC) in hardware ?

Complexity of explicit MPC

- Number of regions depends on number of possible combinations of active constraints
- Weak dependence on number of states and references
- On-line QP vs explicit MPC comparison:

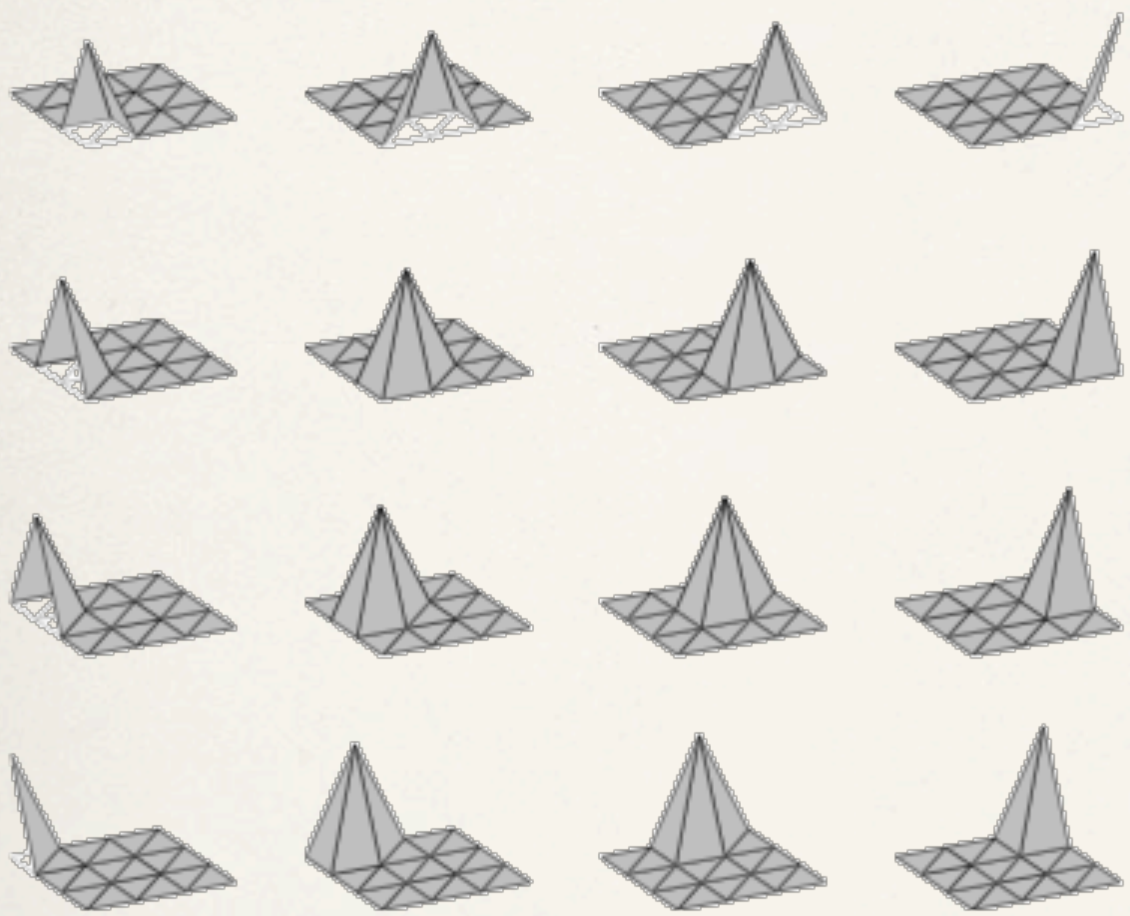
$2N$	QP (ms) average	worst	explicit (ms) average	worst	regions	[storage kb]
4	1.1	1.5	0.005	0.1	25	16
8	1.3	1.9	0.023	1.1	175	78
20	2.5	2.6	0.038	3.3	1767	811
30	5.3	7.2	0.069	4.4	5162	2465
40	10.9	13.0	0.239	15.6	11519	5598

(Intel Centrino 1.4 GHz)

Explicit MPC typically limited to 6-8 free control moves and 8-12 states+references

PWA approximation of MPC over simplices

- *Approximate* a given linear MPC controller by using **canonical PWA** functions over **simplicial partitions** (**PWAS**) (Bemporad, Oliveri, Poggi, Storace , IEEE TAC, 2011)



(Julian, Desages, Agamennoni, 1999)

$$\hat{u}(x) = \sum_{k=1}^{N_v} w_k\,\phi_k(x) = w'\phi(x)$$

approximate MPC law

Weights w_k optimized **off-line**
to best approximate a given MPC law

PWA approximation of MPC over simplices

- **Extremely cheap**: **PWAS functions can be directly implemented on FPGA**, or even **ASIC** (Application Specific Integrated Circuits)

- **Extremely fast** computations (**10-100 nanoseconds**)

fit criterion

Control	p	Latency (A - B) [ns]
	7	170 - 31
L^2	31	238 - 45
	63	272 - 46
	7	170 - 31
L^∞	31	238 - 45
	63	272 - 46

online time to compute MPC (Xilinx Spartan 3 FPGA)

Architecture A: mainly serial

Architecture B: fully parallel

partitions per dimension

Exact explicit MPC: 52 regions
383 ns (avg) - 486 ns (max)

MIMO system dynamics

$$x_{k+1} = \begin{bmatrix} 1.2 & 1 \\ 0 & 1.1 \end{bmatrix} x_k + \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} u_k$$

$$y_k = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} x_k$$

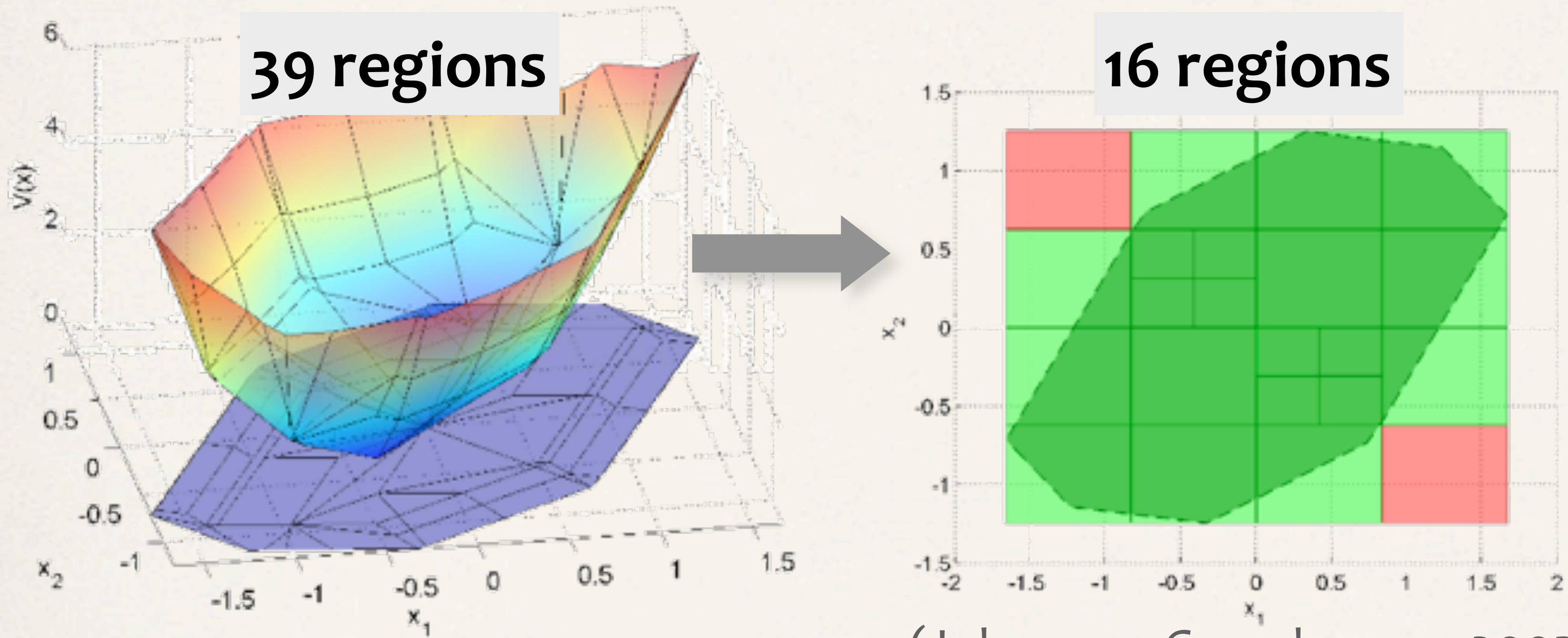
subject to saturation on u and y

- Closed-loop stability certified *a posteriori* via a **PWA Lyapunov function**

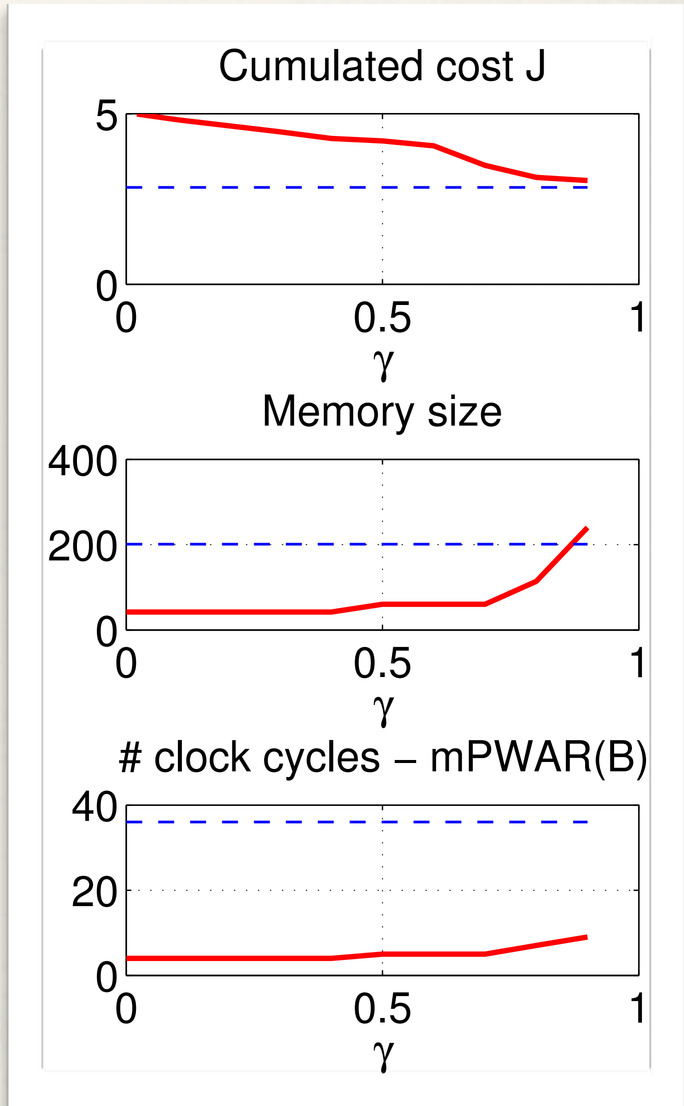
✖ Curse of dimensionality (wrt to state dimension)

PWA approximation of MPC over rectangles

- *Approximate* a given linear MPC controller by using **regular PWA** functions over **hyper-rectangular partitions** (**PWAS**) (Liang, Heemels, Bemporad, 2011)



(Johansen, Grancharova, 2003)

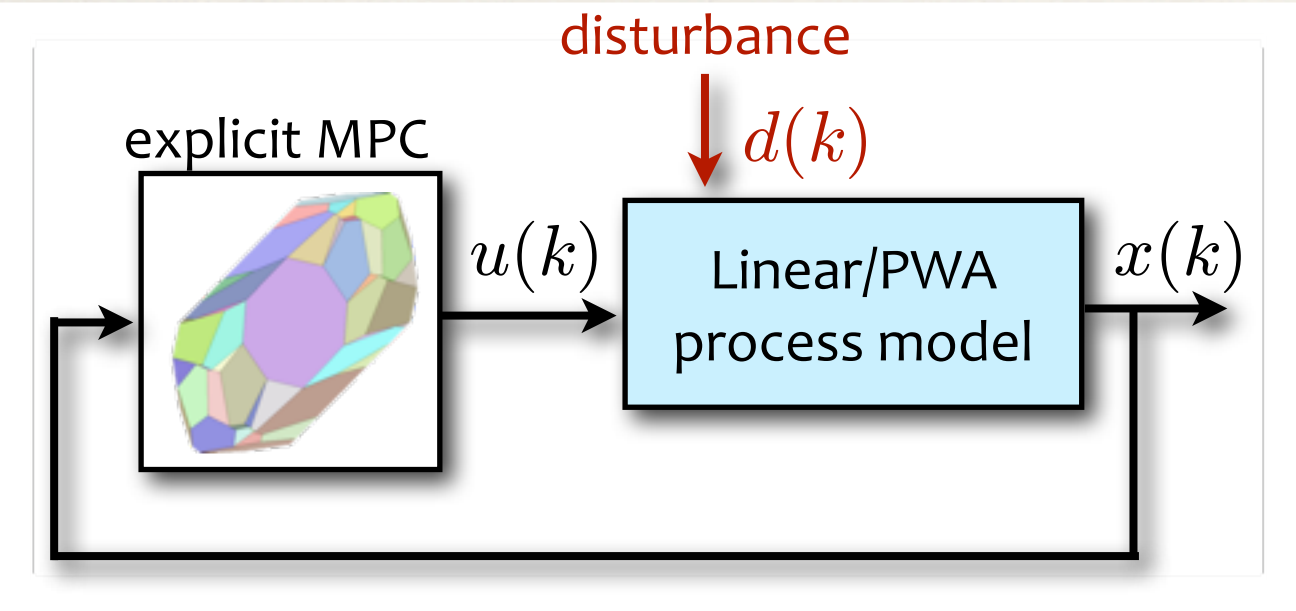


- Key idea: **use optimal PWA cost as a control Lyap. fnc**

- Main features of suboptimal control law:
  - Can be computed (off-line) by solving a **linear program** (LP)
  - Closed-loop **stability** and degree of **performance loss** are guaranteed a priori
  - Fulfillment of **input and state constraints** is guaranteed a priori
  - Tradeoff between controller complexity and suboptimality is selectable

Stability analysis of PWA closed-loop systems in non-nominal conditions

Stability analysis of (perturbed) PWA systems



$$x(k+1) = A_i x(k) + a_i + E_i d(k) \quad \text{if } x(k) \in \mathcal{X}_i$$

Closed-loop uncertain PWA model

- A priori **stability certificates** (if any is available) only hold for **nominal** conditions (prediction model = process model)

- Is explicit MPC controller in closed-loop with a more realistic process model **stable** (and **robustly stable**) ?

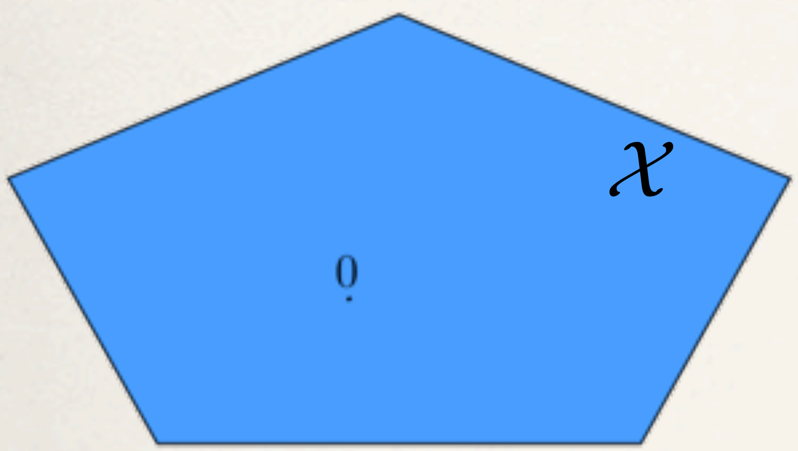
- Trajectories may exit the domain of definition of the explicit MPC law. Need to find a (possibly large) **robustly positive invariant set**

Stability analysis of (perturbed) PWA systems

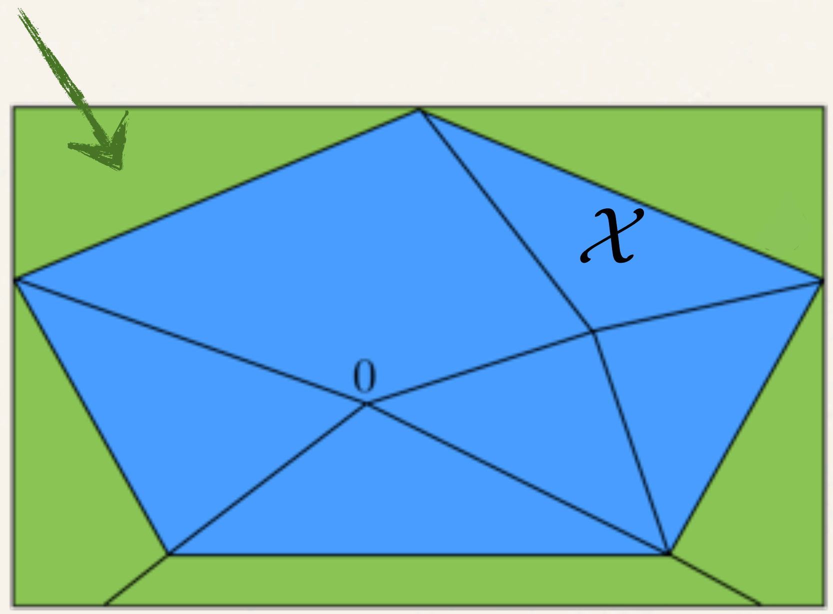
(Rubagotti, Trimboli, Bemporad, 2011)

A new analysis method to check **uniform ultimate boundedness** and evaluate the **region of attraction** based on **PWA Lyapunov functions**

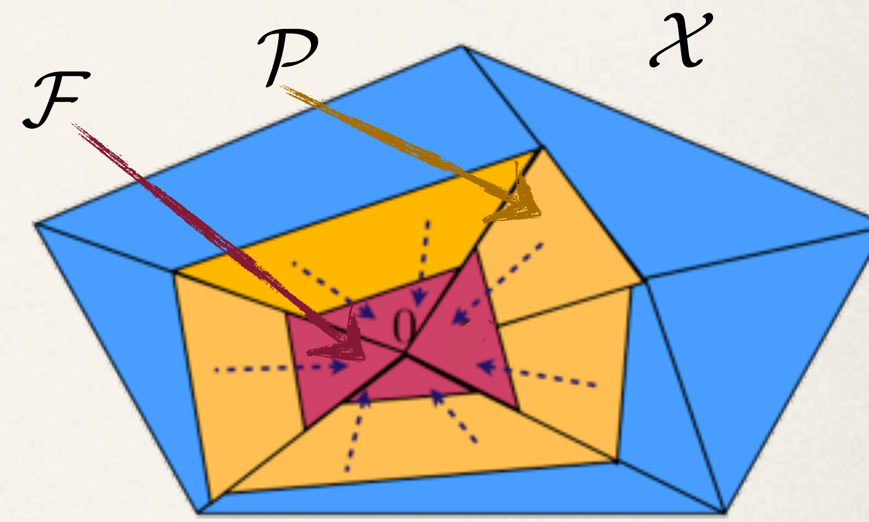
$$x(k+1) = \rho x(k),\ 0 < \rho < 1$$



domain of definition
of closed-loop PWA system
(e.g.: domain of explicit MPC)



extend to invariant set
by introducing “fake” dynamics

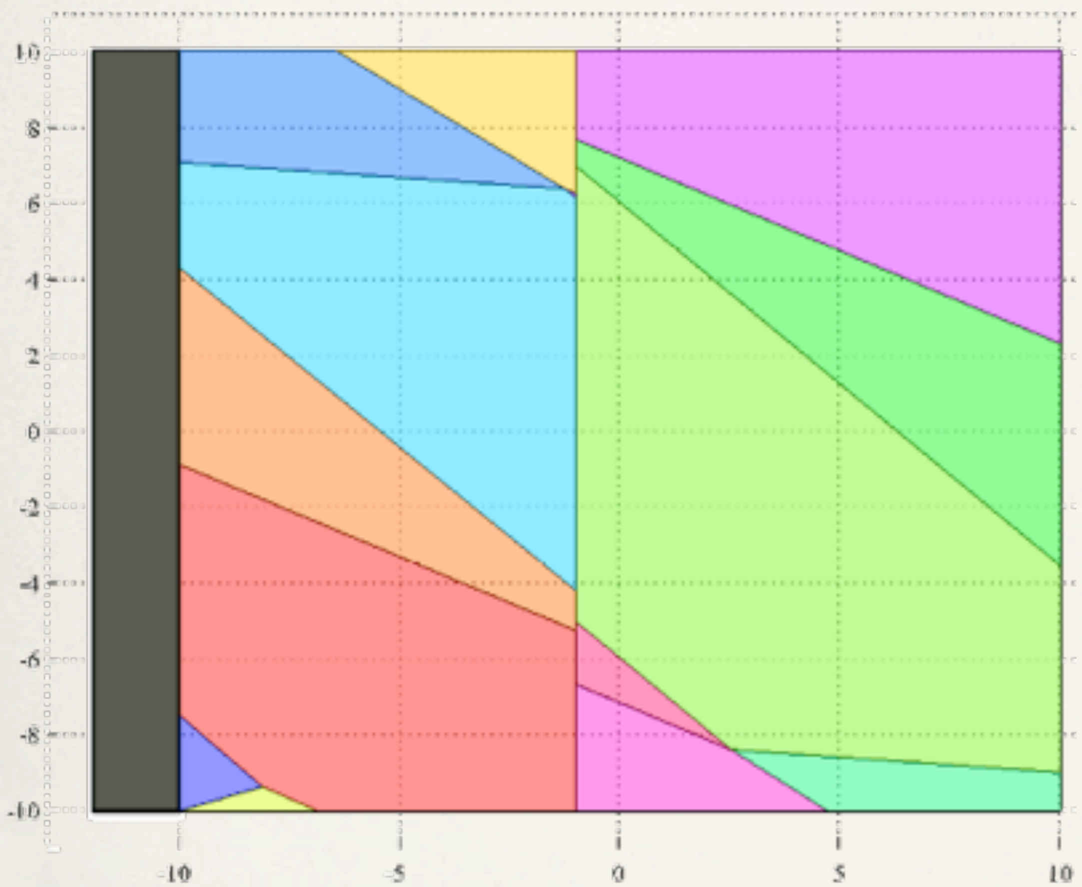


synthesize a PWA
Lyapunov function V
via linear programming
➡ $\mathcal{P}$, $\mathcal{F}$ = level sets of V

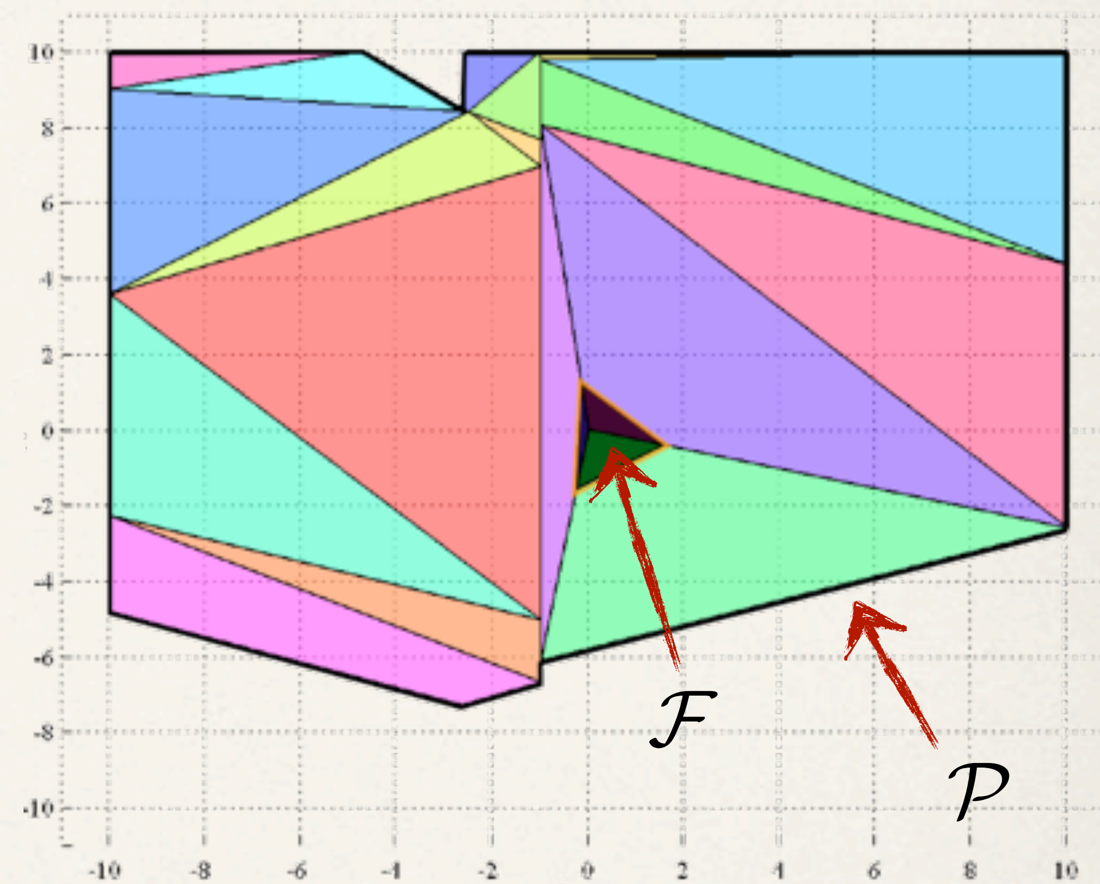
- $\mathcal{P} \subseteq \mathcal{X}$ of V is invariant for the original dynamics
- the system is uniformly ultimately bounded from $\mathcal{P}$ to $\mathcal{F}$

Example: stability of switched linear MPC

(Rubagotti, Trimboli, Bemporad, 2011)



switched linear MPC controller
(no a priori stability)



domain of attraction $\mathcal{P}$ and
set $\mathcal{F}$ of asymptotic convergence

http://www.mobydic-project.eu/

MOBY-DIC Toolbox for MATLAB

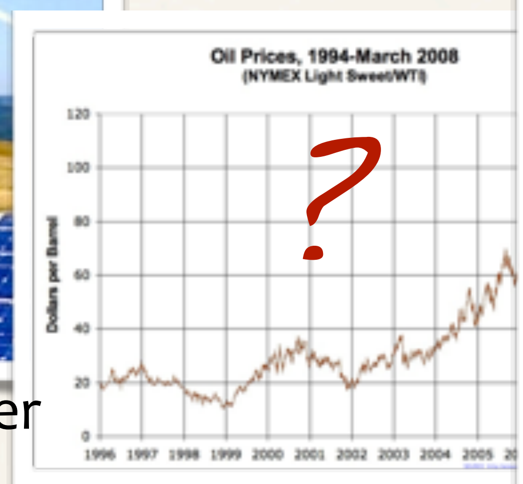
(Oliveri *et al.*, NMPC, 2012)

Can we include stochastic models in MPC formulation ?

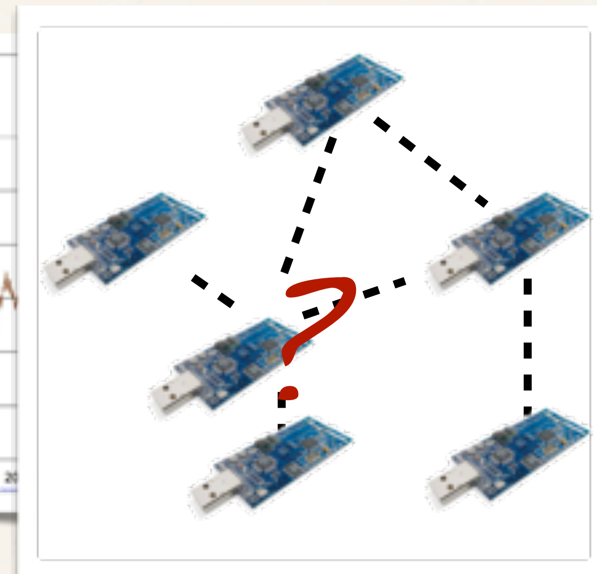
Stochastic systems

- In many control problems decisions must be taken under **uncertainty**

wind/solar power



prices



wireless networks

demand

human (inter)action

- **Robust** control approaches do not model uncertainty (only assume that is bounded) and pessimistically consider the worst case

- **Stochastic models** provide instead additional information about uncertainty

Stochastic MPC formulation

- At each time t solve a finite-time **stochastic optimal control** problem

$$
\begin{array}{ll}
\min_{u} & E_w\left[\sum_{k=0}^{N-1} \|y_{t+k} - r(t)\|^2 + \rho\|u_{t+k}\|^2\right] \\
\text{s.t.} & x_{t+k+1} = f(x_{t+k}, u_{t+k}, w_{t+k}) \\
& y_{t+k} = g(x_{t+k}, u_{t+k}, w_{t+k}) \\
& u_{\min} \leq u_{t+k} \leq u_{\max} \\
& Prob(y_{\min} \leq y_{t+k} \leq y_{\max}) \geq p \\
& x_t = x(t),\ \ k = 0, \ldots, N-1
\end{array}
$$

$x(t)$ = process state

$u(t)$ = manipulated vars

$y(t)$ = controlled output

$w(t)$ = stochastic dist.

- Apply optimal move for current time t, repeat optimization at time $t+1$

Scenario-based stochastic MPC

Existing literature on stochastic MPC

(Schwarme & Nikolaou, 1999) (Munoz de la Pena, Bemporad, Alamo, 2005) (Oldewurtel, Jones, Morari, 2008)
(Wendt & Wozny, 2000) (Couchman, Cannon, Kouvaritakis, 2006) (Ono, Williams, 2008)
(Batina, Stoorvogel, Weiland, 2002) (Primbs, 2007) (van Hessem & Bosgra 2002) (Bemporad, Di Cairano, 2005)

Stochastic prediction model

$$x(k+1) = A(w(k))x(k) + B(w(k))u(k) + Hw(k)$$

(Bernardini, Bemporad, IEEE TAC, 2012)

$$w(k) \in \{w_1, w_2, \dots, w_s\}$$

$$P[w(k) = w_i] = p_i(k)$$

Stochastic MPC features

- Less conservative control action w.r.t. *robust* MPC
- Extremely general discrete probability distribution
- Guarantee stochastic convergence and recursive feasibility $\boxed{\lim_{k\to\infty} E[x'(k)x(k)] = 0}$ (for H=0)

Main idea: decouple performance optimization and stability issues

Stochastic MPC for dynamic hedging

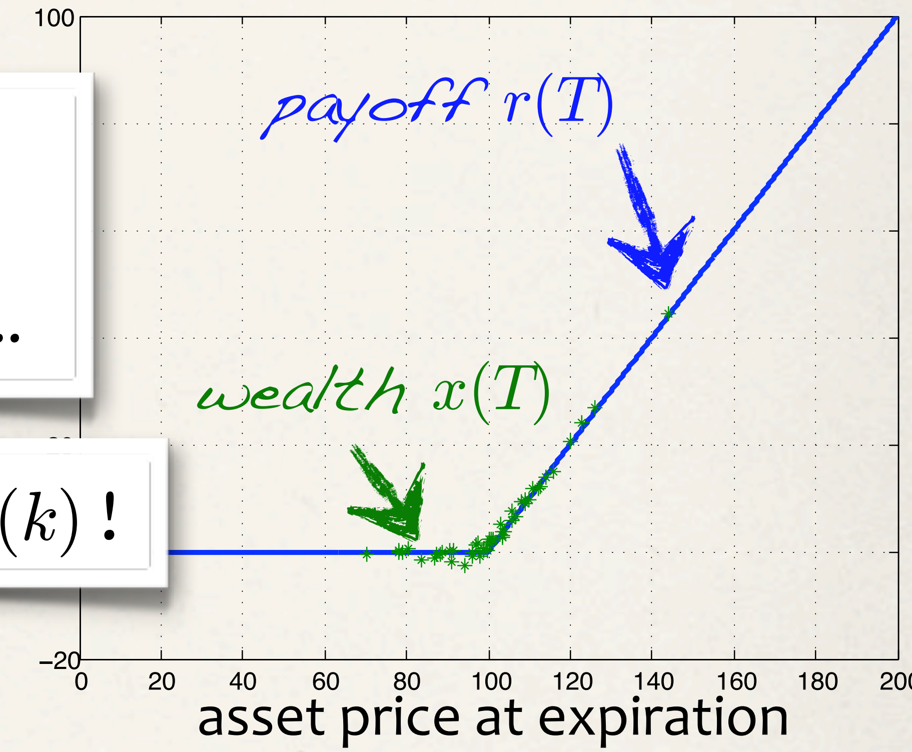
- Dynamic hedging of **financial options**

(Bemporad, Bellucci, Gabbriellini, *Quant Finance*, 2012)

(Bemporad, Gabbriellini, Puglia, Bellucci, CDC'10)

SMPC **adjusts dynamically** the portfolio so that **wealth** $x(T) \cong$ **payoff** $r(T)$ ...

.. for any price realization $w_i(k)$!



On-line optimization: very simple least squares problem with n variables

($n =$ number of traded assets)

MPS CAPITAL SERVICES

SMPC for real-time market-based power dispatch

(Patrinos, Trimboli, Bemporad, CDC'11)

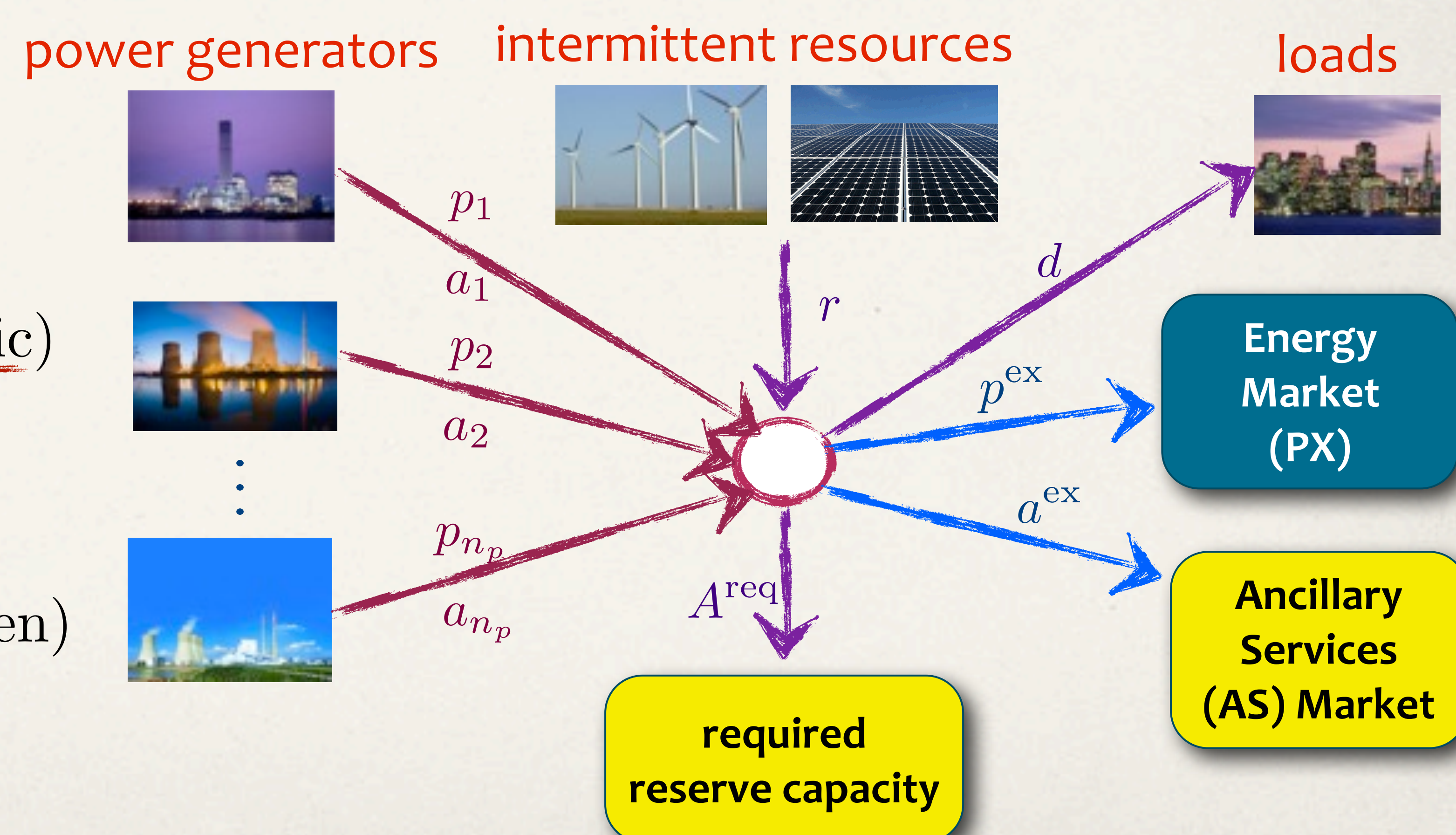
- We are a legal entity trading on the *energy (PX)* and *ancillary service (AS)* markets

- **Objective**: Minimize costs via efficient use of intermittent resources, and maximize profits by trading on electricity (PX, AS) markets

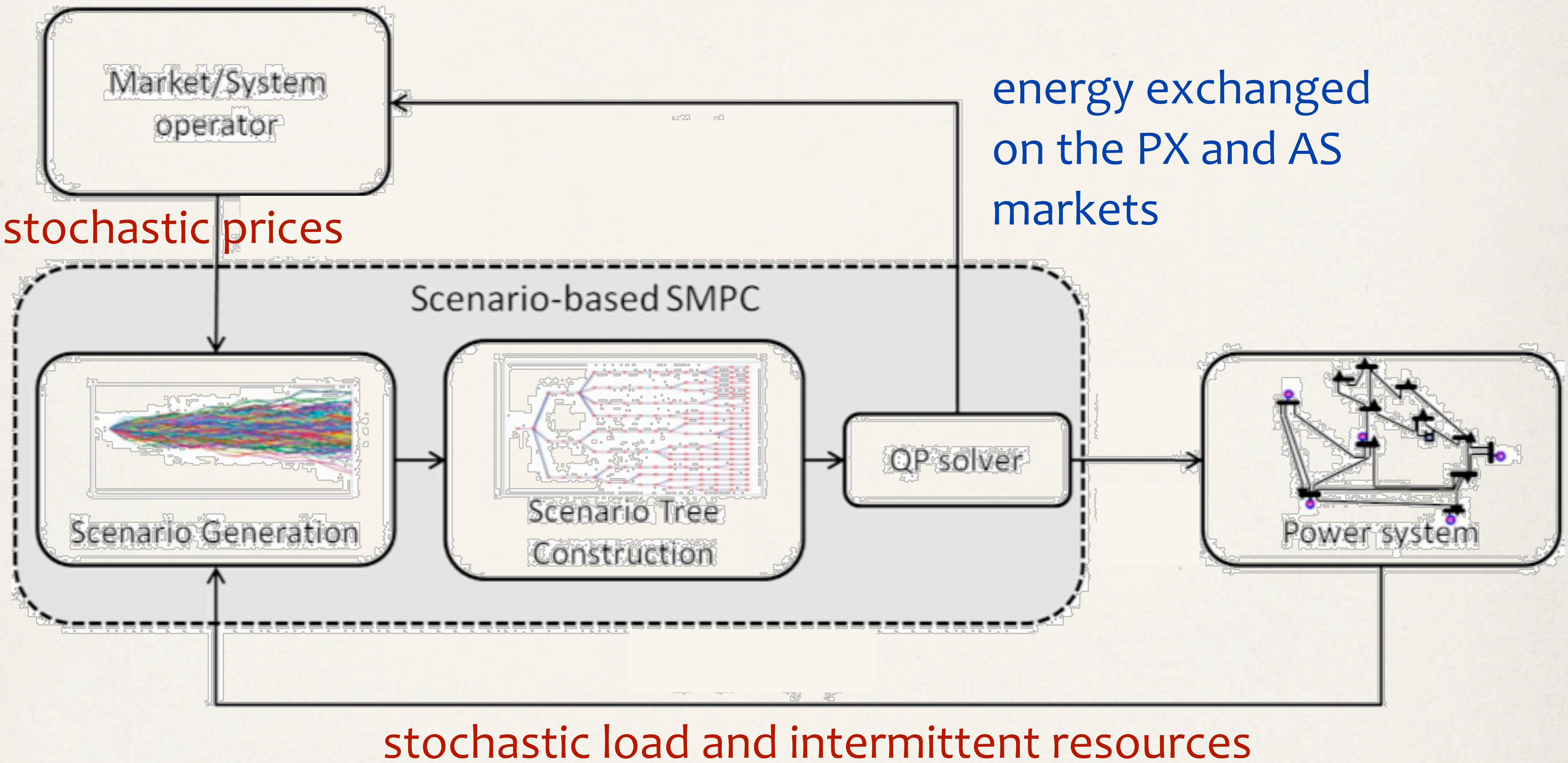
- **Constraints**: Grid capacity, rate limits, load balancing, AS balancing

p_i = power injection (decision)
a_i = reserve capacity (decision)
r = intermittent generation (stochastic)
d = demand (stochastic)
p^{ex} = power exchanged (decision)
p^{as} = power exchanged (decision)
A^{req} = required reserve for BRP (given)

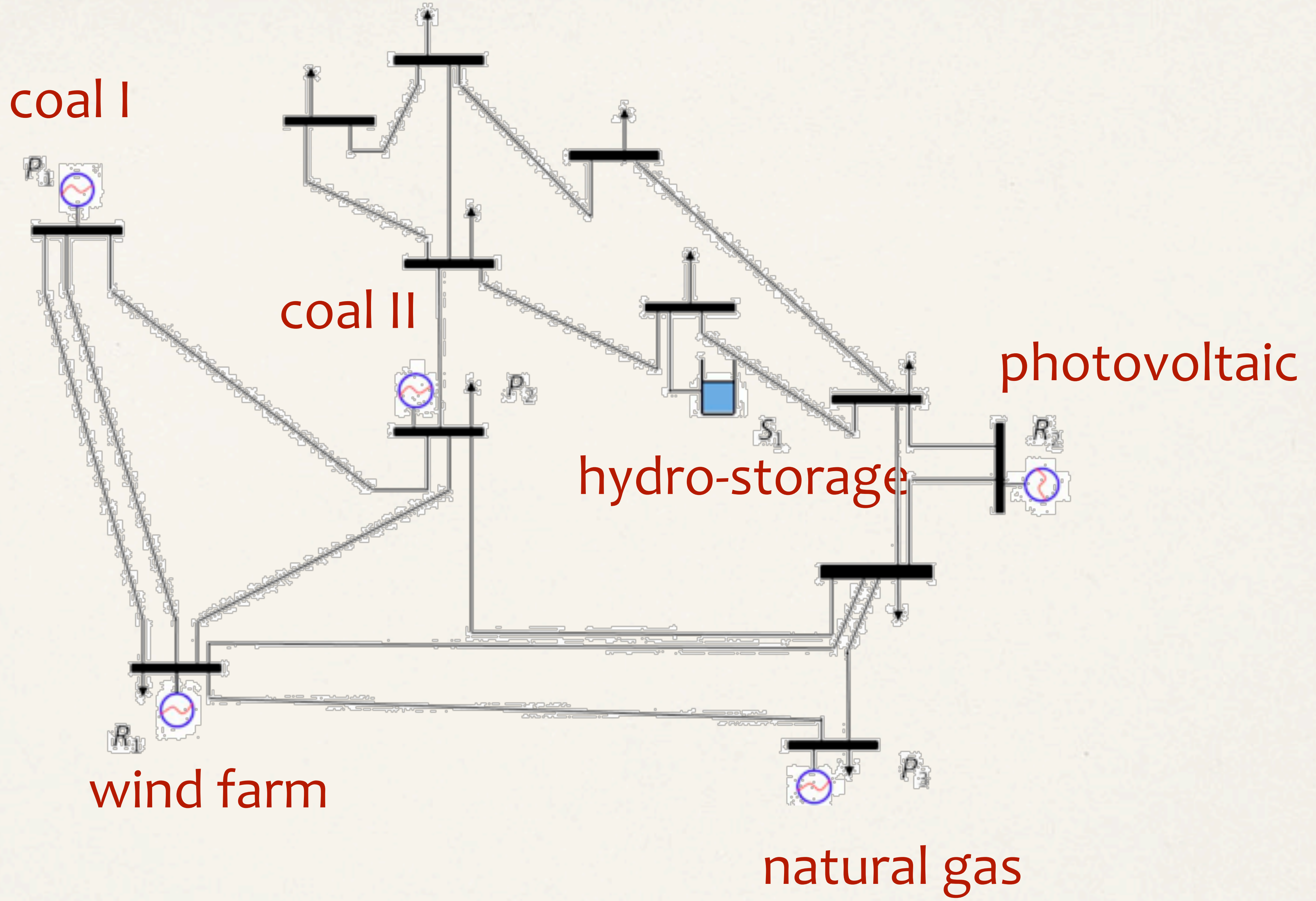
energy prices are also stochastic



SMPC architecture



SMPC for market-based optimal power dispatch



ENABLING THE FUTURE ENERGY SYSTEM

http://www.e-price-project.eu/

SEVENTH FRAMEWORK PROGRAMME

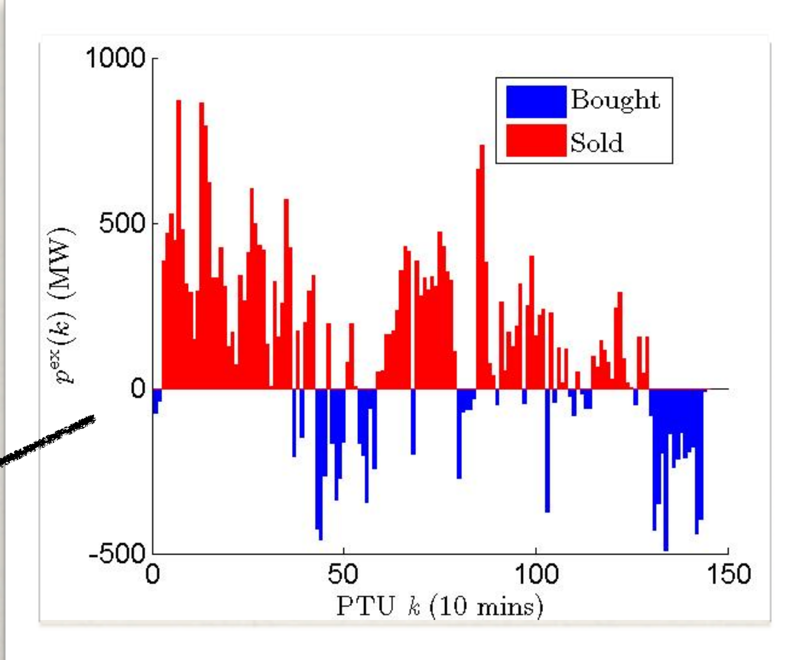
SMPC for market-based optimal power dispatch

Exact knowledge of future uncertainty

Deterministic: time-dependent expectations used for future uncertainty

Stochastic formulation

Algorithm	Storage Cost	No Storage Cost	Avg # of nodes
Prescient-OC	6427979	6879741	
CE-MPC	9778750	9819518	
SSMPC ($e_{\mathrm{rel}} = 0.1$)	7134582	7245962	350
SSMPC ($e_{\mathrm{rel}} = 0.2$)	7144011	7249401	335
SSMPC ($e_{\mathrm{rel}} = 0.3$)	7148494	7250207	172
SSMPC ($e_{\mathrm{rel}} = 0.4$)	7179848	7264505	87
SSMPC ($e_{\mathrm{rel}} = 0.5$)	7224912	7267497	50
SSMPC ($e_{\mathrm{rel}} = 0.6$)	7239985	7277410	38
SSMPC ($e_{\mathrm{rel}} = 0.7$)	7259491	7298023	31
SSMPC ($e_{\mathrm{rel}} = 0.8$)	7255246	7312092	26
SSMPC ($e_{\mathrm{rel}} = 0.9$)	7260424	7318643	22
SSMPC ($e_{\mathrm{rel}} = 1.0$)	7260424	7318642	20



power exchanged with grid

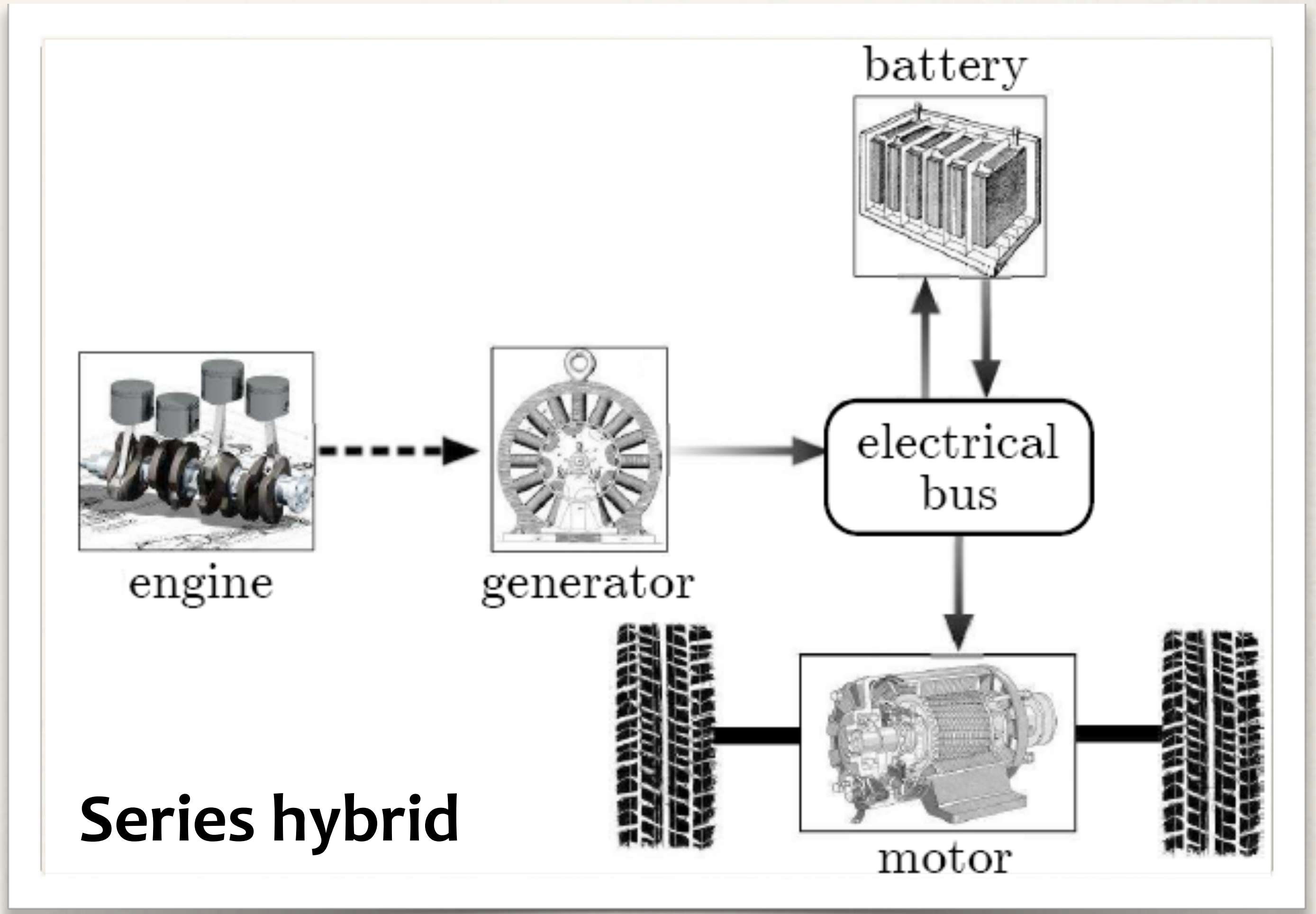
SMPC for hybrid electric vehicles (HEVs)

(Bichi, Ripaccioli, Di Cairano, Bernardini, Bemporad, Kolmanovsky, CDC'10)

Control problem:

Decide optimal generation of **mechanical power** (from engine) and **electrical power** (from battery) to satisfy **driver's power request**

What will the future power request from the driver be ?



Learning a stochastic model of the driver

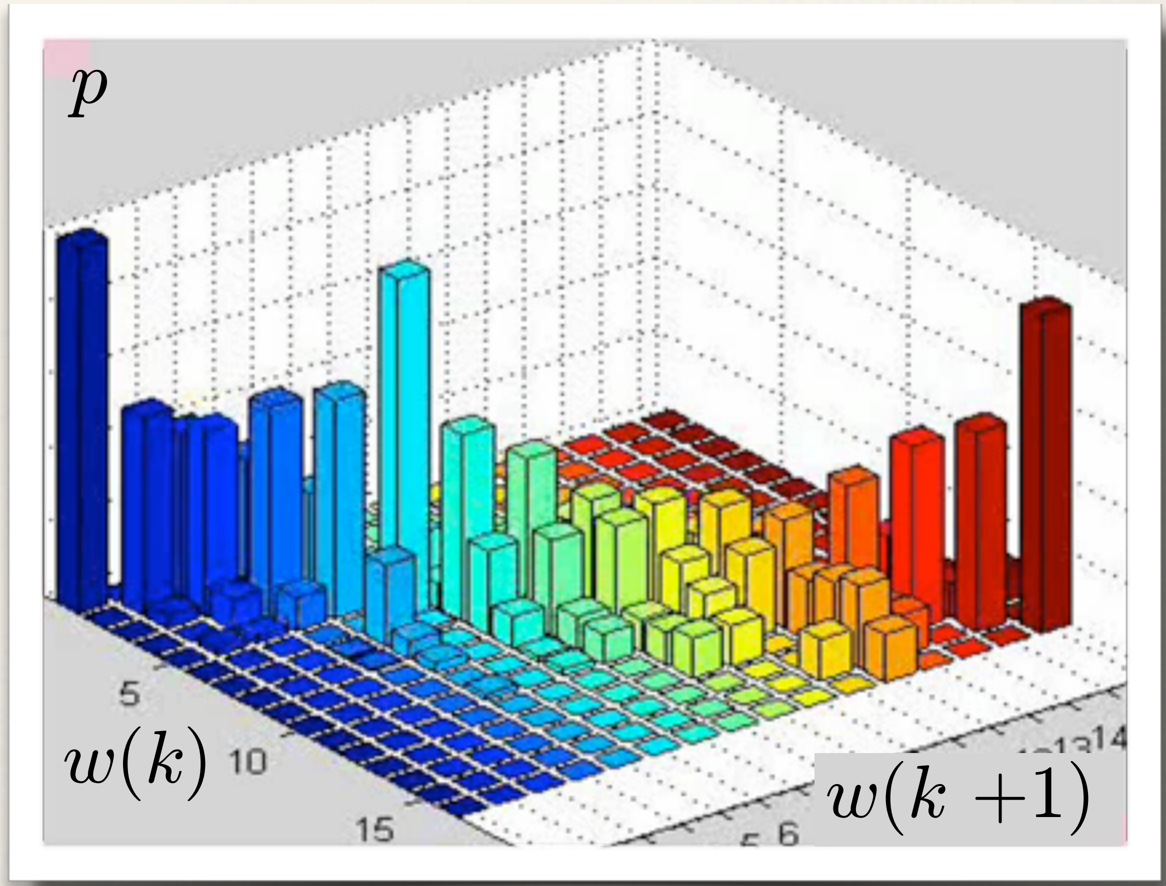
- Driver's action (power request) modeled by a **stochastic** process $w(k)$

- Good model for control purposes: $w(k)$ generated by a Markov chain

$$[T]_{ij} = \mathbf{P}[w(k+1) = w_j | w(k) = w_i]$$

Number of states in Markov chain determines the **trade-off** between complexity *and* accuracy

Transition probability matrix T is easily estimated from driver's data



Stochastic MPC results

Deterministic: future power request ≡ current one ≡ constant

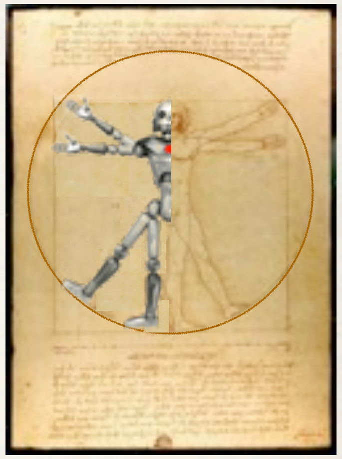
Comparison on a standard driving cycle

	Fuel cons. [kg]	% Fuel improv.
FTMPC	0.281	–
SMPC (static)	0.243	13.5%
SMPC (adaptive)	0.199	29.2%
PMPC	0.197	29.9%

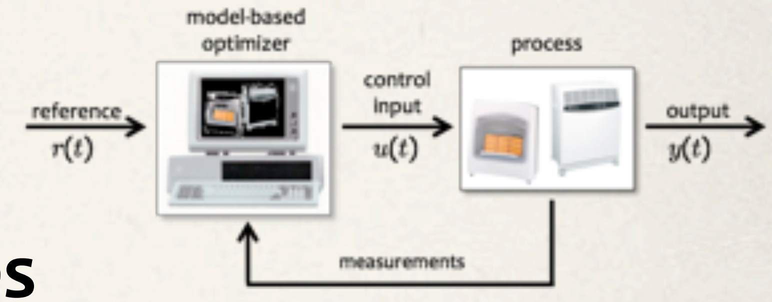
Stochastic formulation

Exact knowledge of future uncertainty

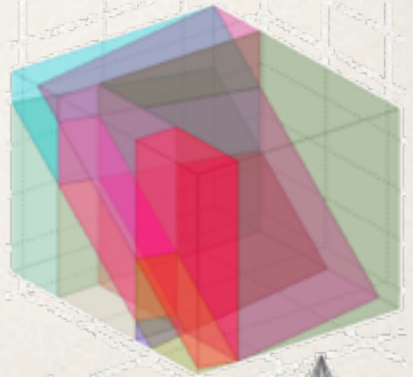
Conclusions

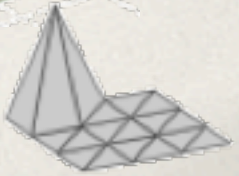


- **Hybrid models** (and in particular **piecewise affine** models)
  - ✓ provide a very **powerful modeling framework**
  - ✓ allow the setup of tractable **optimization problems** (MPC)
  - ✓ powerful **analysis** and **software tools** are available
  - ✓ **stochastic** switched affine models optimally describe uncertainty



- **Hybrid MPC**
  - ✓ very good to study achievable **performance limits**
  - ✓ directly implementable for controlling **slow processes**
  - ✖ rarely applicable to fast processes, unless **suboptimal** solutions are adopted (switched linear MPC, LTV-MPC)





- **Explicit MPC** based on **piecewise affine** representations
  - ✓ applicable to **very fast processes**
  - ✖ limited to **small-size** processes (8-10 states, 1-2 inputs)
  - ✖ curse of dimensionality when using **regular partitions**