Modelling and Optimization-based Control of Hybrid Dynamical Systems

Alberto Bemporad

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

University of Siena

Department of Information Engineering

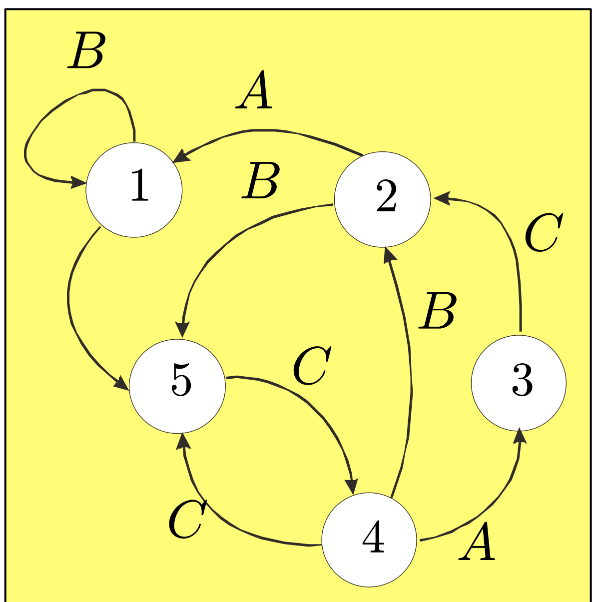
Talk outline

- Models of hybrid systems
- Model predictive control (MPC) of hybrid systems
- Automotive applications
- New research directions

Hybrid dynamical systems

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

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



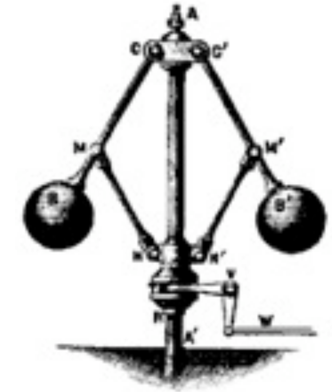
Computer Science

Finite state machines

Control Theory

Continuous dynamical systems

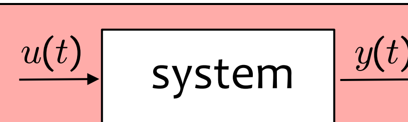
Hybrid systems



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

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

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

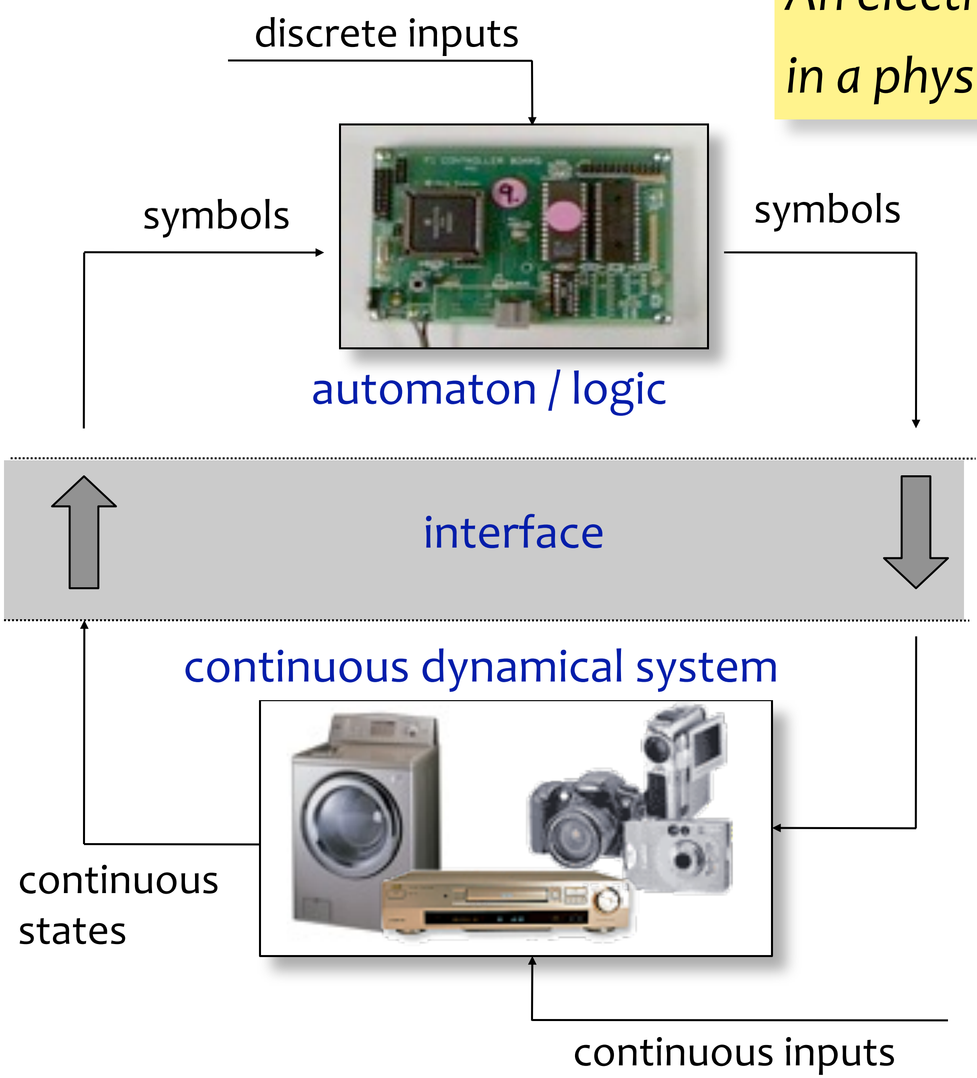


$$\begin{cases} \frac{dx(t)}{dt} & = & f(x(t),u(t)) \\ y(t) & = & g(x(t),u(t)) \end{cases}$$

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

Embedded systems

*An electronic (control) device is "**embedded**" in a physical process and interacts with it*



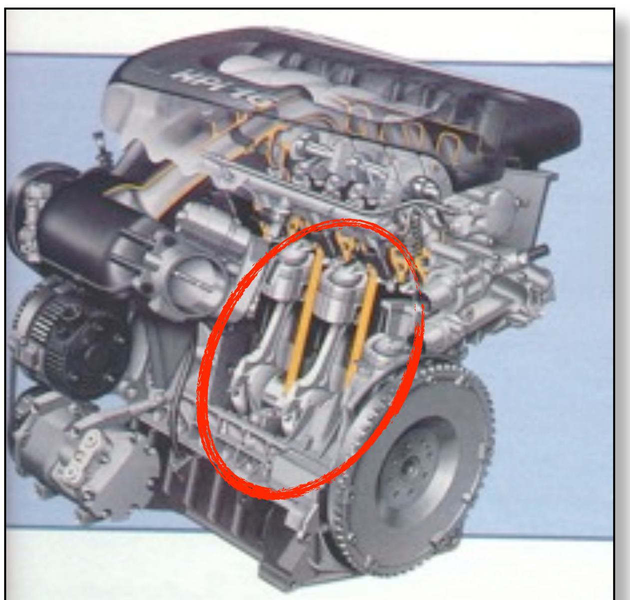
Examples: automobiles, industrial processes, consumer electronics, home appliances, ...

Sensor-based image stabilization

"Intrinsically hybrid" systems

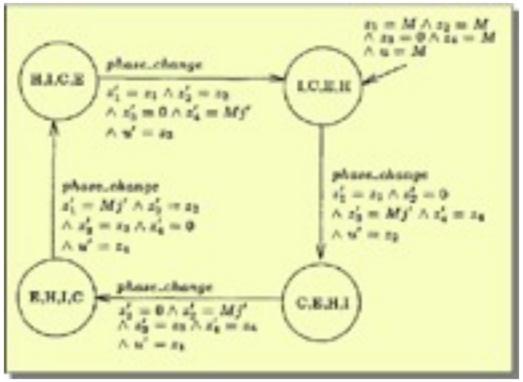
- Transmission

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



- Four-stroke engines

automaton, dependent on crankshaft angle

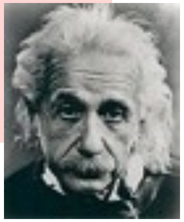


Key requirements for hybrid models

- **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

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



linear hybrid systems

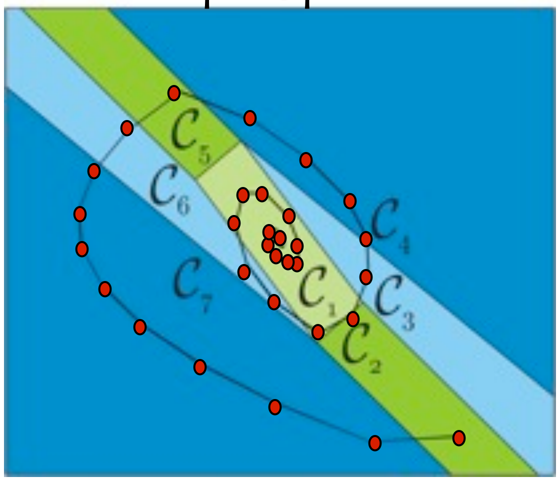
Piecewise affine (PWA) systems

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

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

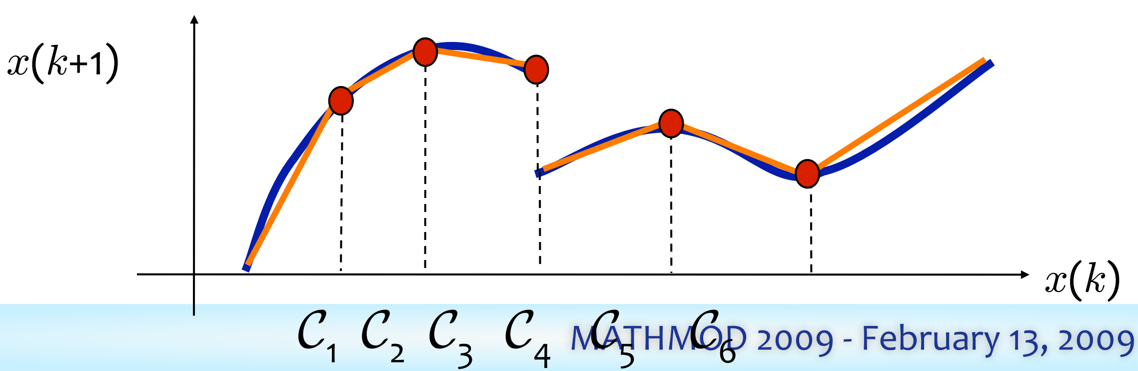
$i(k) \in \{1, \ldots, s\}$

state+input space



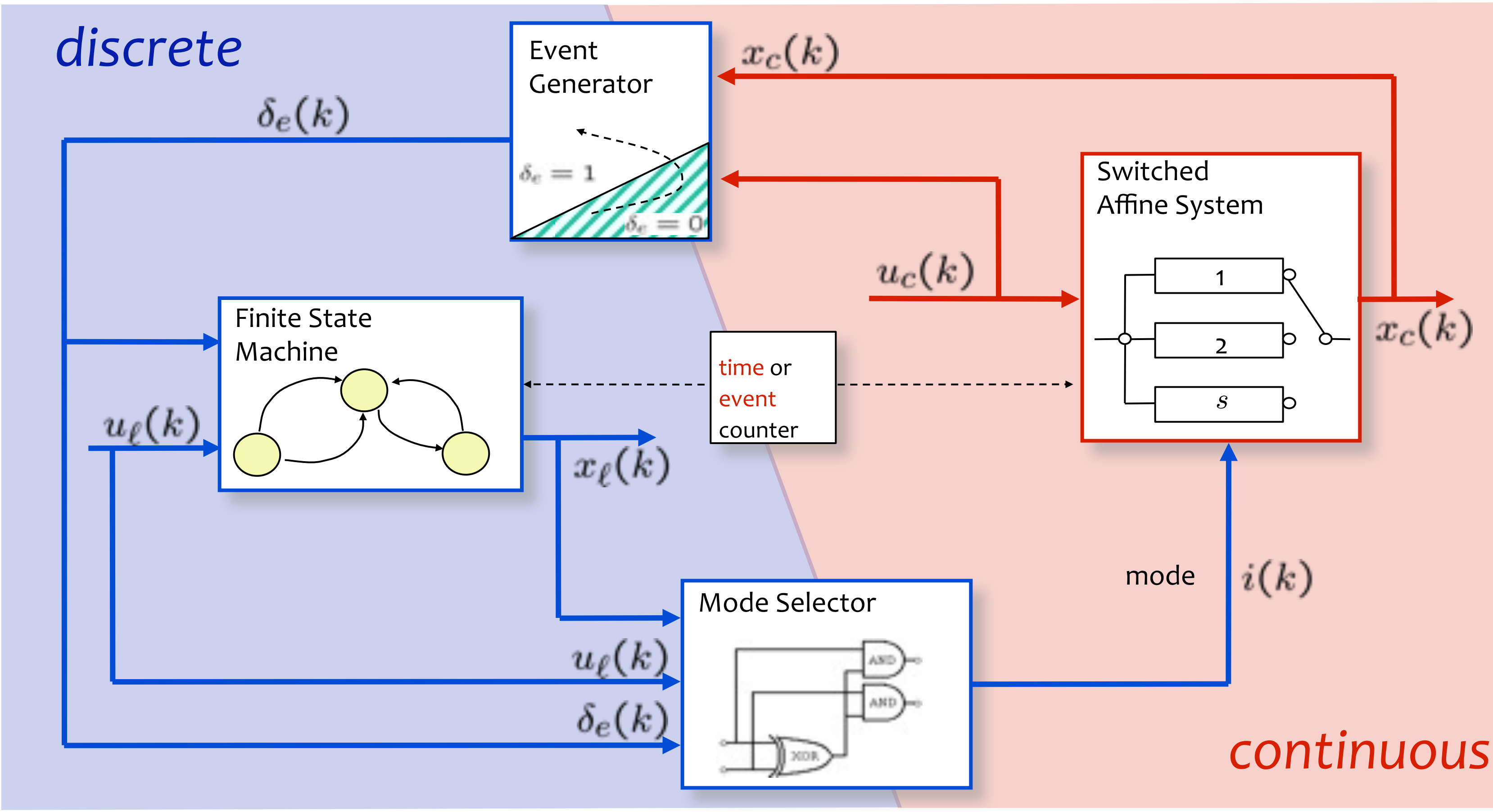
(Sontag 1981)

Can approximate nonlinear and/or discontinuous dynamics arbitrarily well



Discrete Hybrid Automata (DHA)

(Torrisi, Bemporad, 2004)



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

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

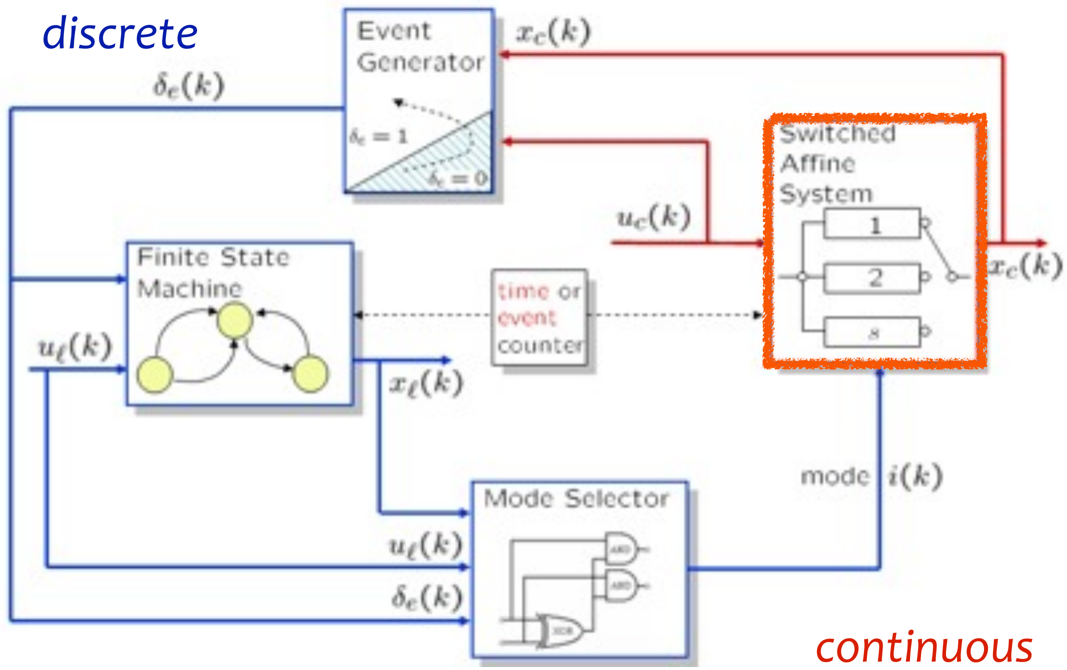
$\delta_e \in \{0,1\}^{n_e}$ = event variables

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

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

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

Switched affine system

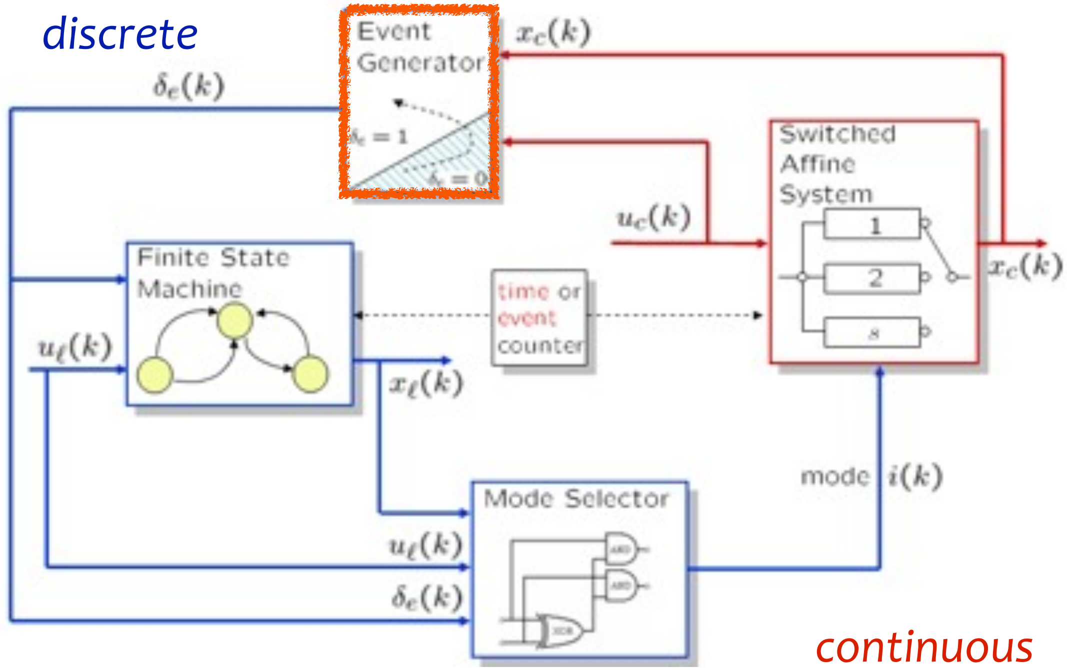


The affine dynamics depend on the current mode $i(k)$:

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

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

Event generator



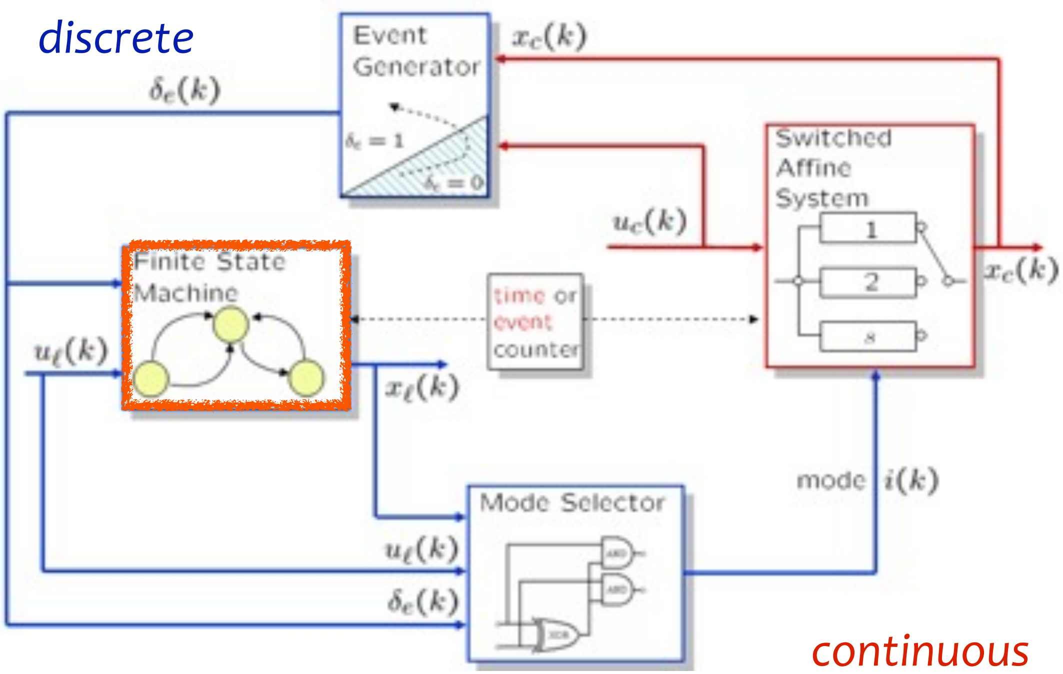
Event variables are generated by linear threshold conditions over continuous states and inputs (time events can be also modeled):

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

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

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

Finite state machine



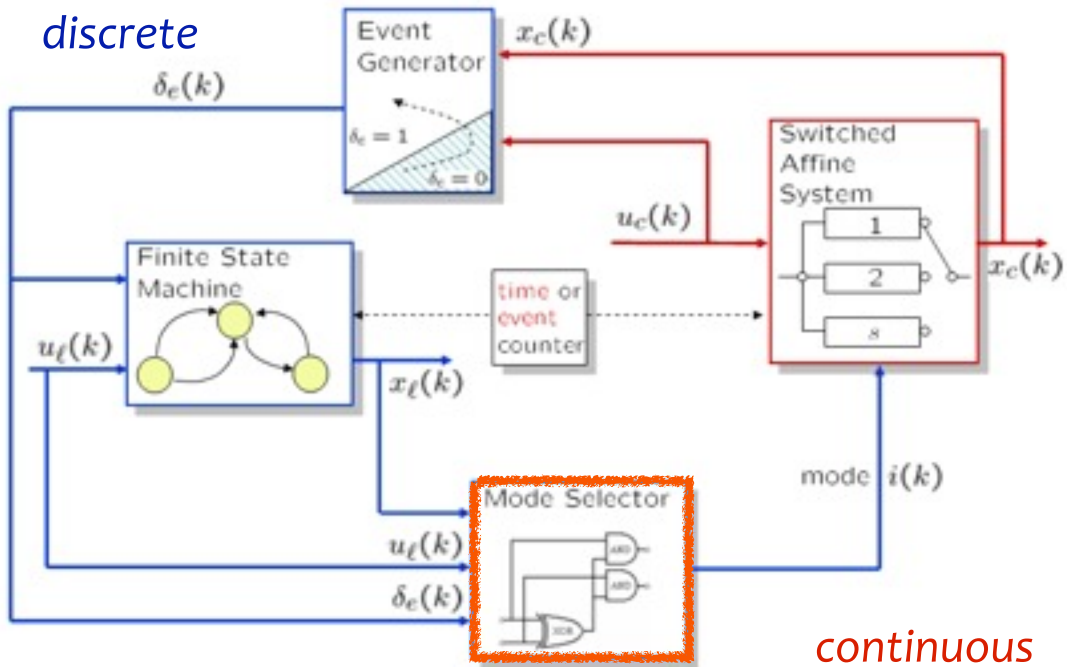
The binary state of the finite state machine evolves according to a Boolean state update function:

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

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

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

Mode selector



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

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

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

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

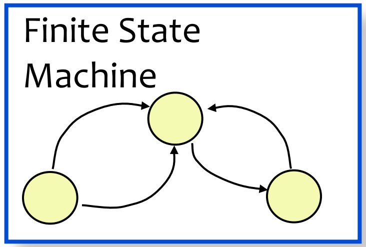
Logic and linear inequalities

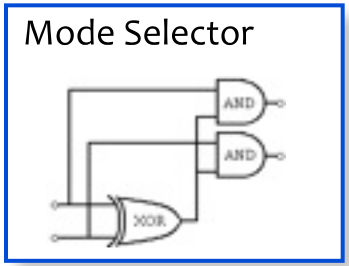
$$X_1 \vee X_2 = \text{TRUE} \longrightarrow \delta_1 + \delta_2 \geq 1, \qquad \delta_1, \delta_2 \in \{0, 1\}$$

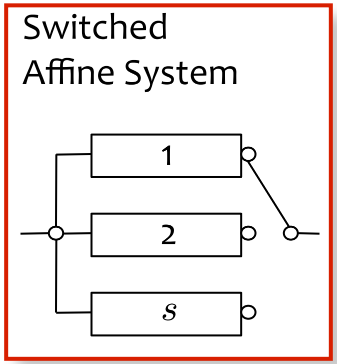
(Glover 1975, Williams 1977, Hooker 2000)

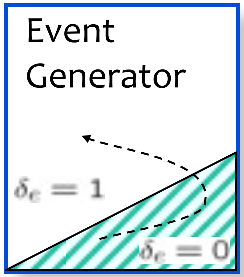
Any logic formula involving **Boolean variables** and **linear combinations of continuous variables** can be translated into a set of **(mixed-)integer linear (in)equalities**

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



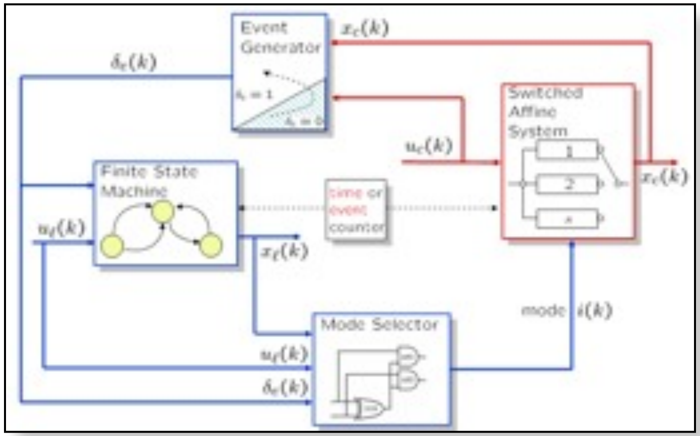


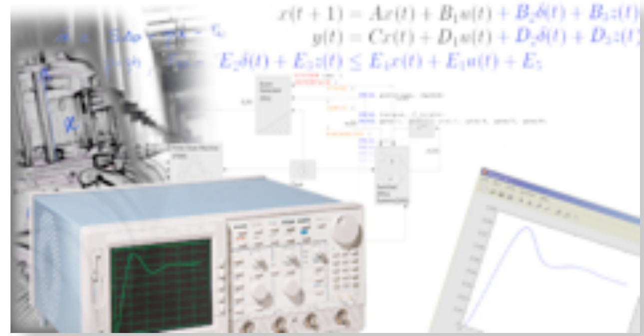




Mixed-logical dynamical systems

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

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

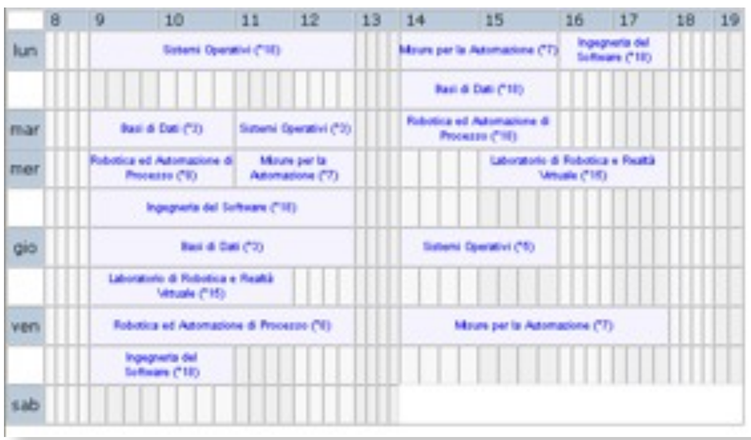
Suitable for solving optimization problems (mixed-integer programming)

Mixed-integer models in operations research

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: Timetable generation (for demanding professors …)

CPU time: 0.2 s



Effort: 10% mathematical problem setup (mixed-integer linear model)
30% database & web interfaces
60% deal with professors' complaints, complaints, complaints …

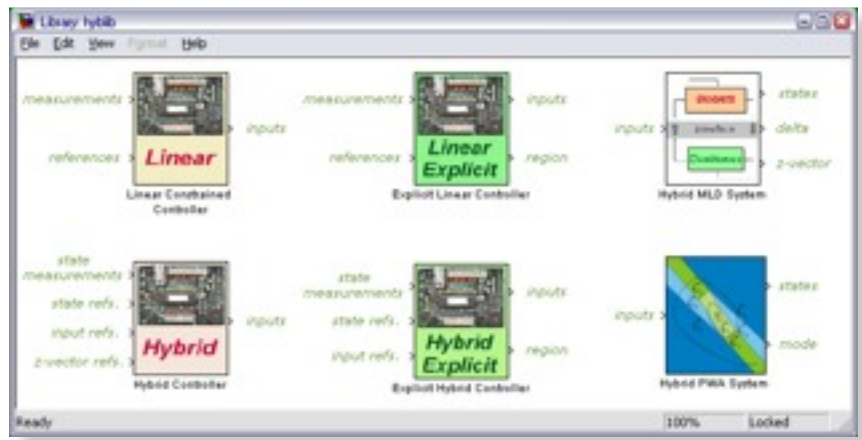
Hybrid Toolbox for MATLAB

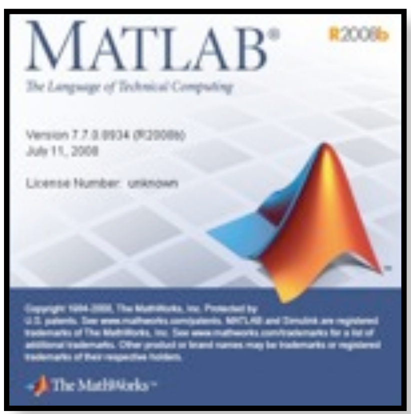
Features:

(Bemporad, 2003-2009)

- Hybrid models: design, simulation, verification
- Control design for linear systems w/ constraints
  and hybrid systems (on-line optimization via QP/MILP/MIQP)
- Explicit MPC control (via multi-parametric programming)
- C-code generation
- Simulink library

Support:



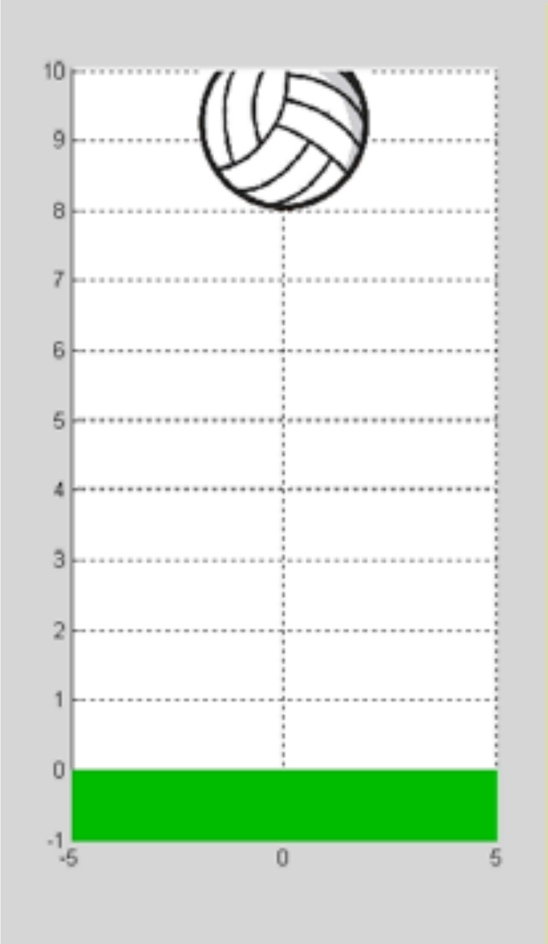


2200+ download requests since October 2004

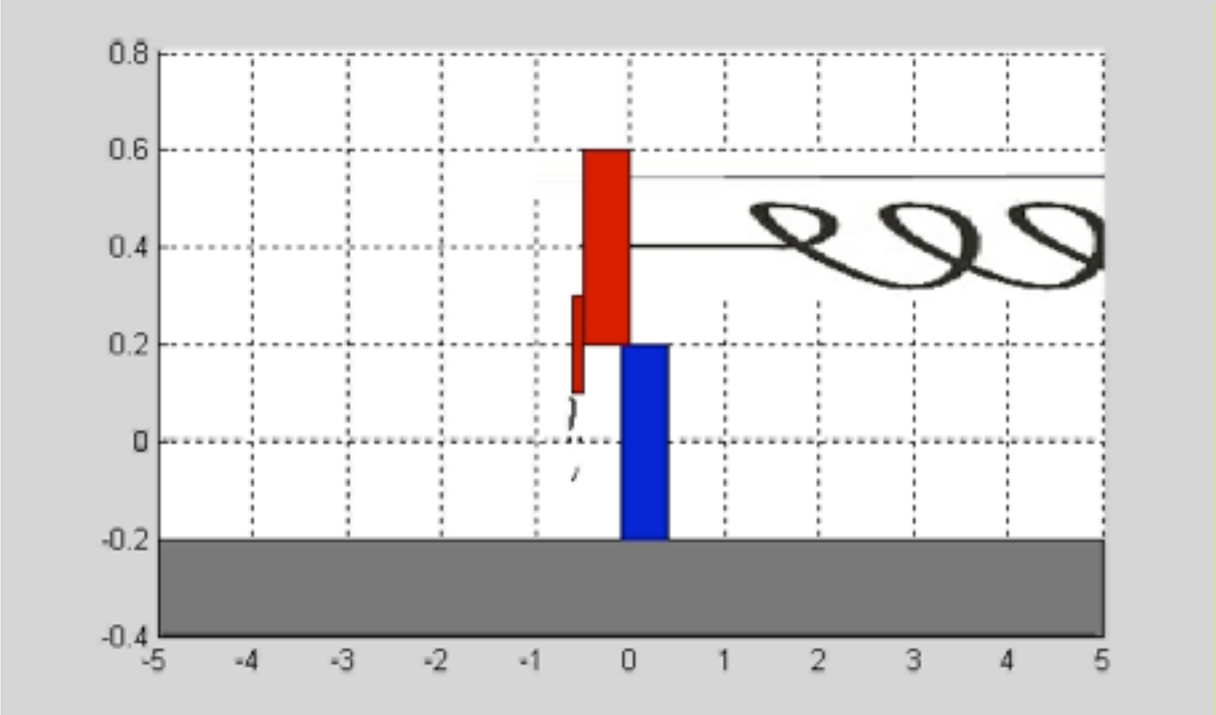
http://www.dii.unisi.it/hybrid/toolbox

Examples: systems with impacts

bouncing ball

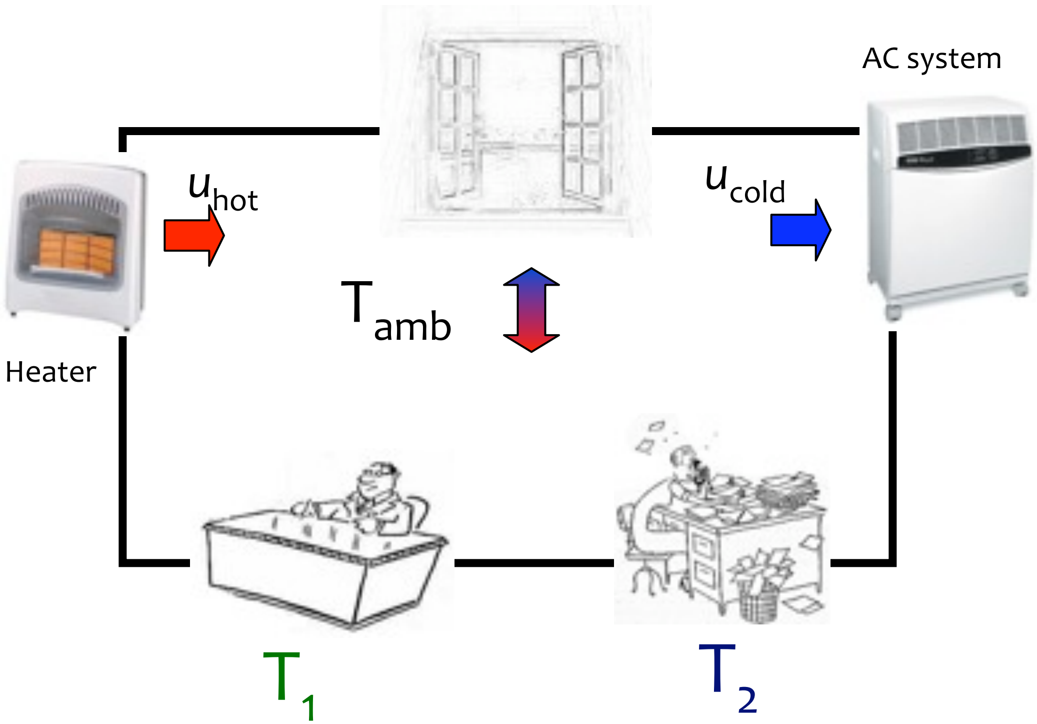


magnetically actuated fuel injector



Easily modeled as discrete-time linear hybrid systems

Example: room temperature



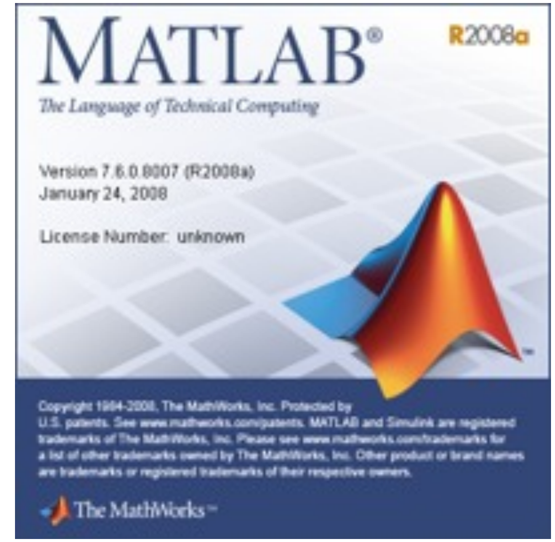
Hybrid dynamics

- #1 turns the heater (A/C) 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 A/C on, unless #2 is cold
- Otherwise, heater and A/C are off

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

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

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

- 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

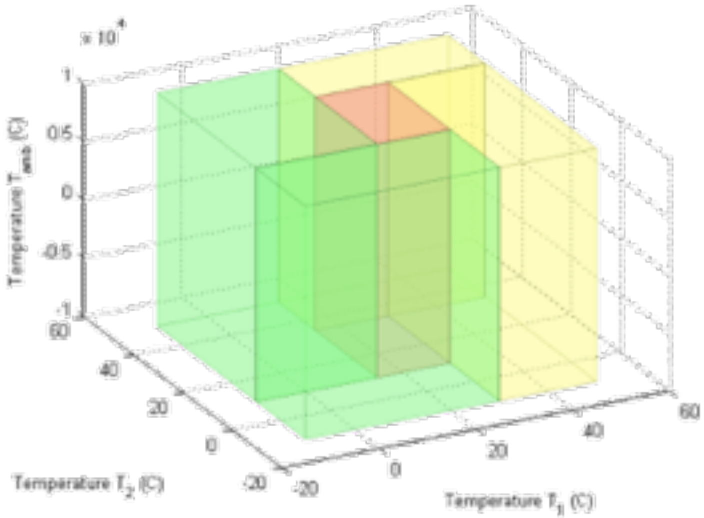
- PWA model

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

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

- 2 continuous states:
  (temperatures T_1,T_2)

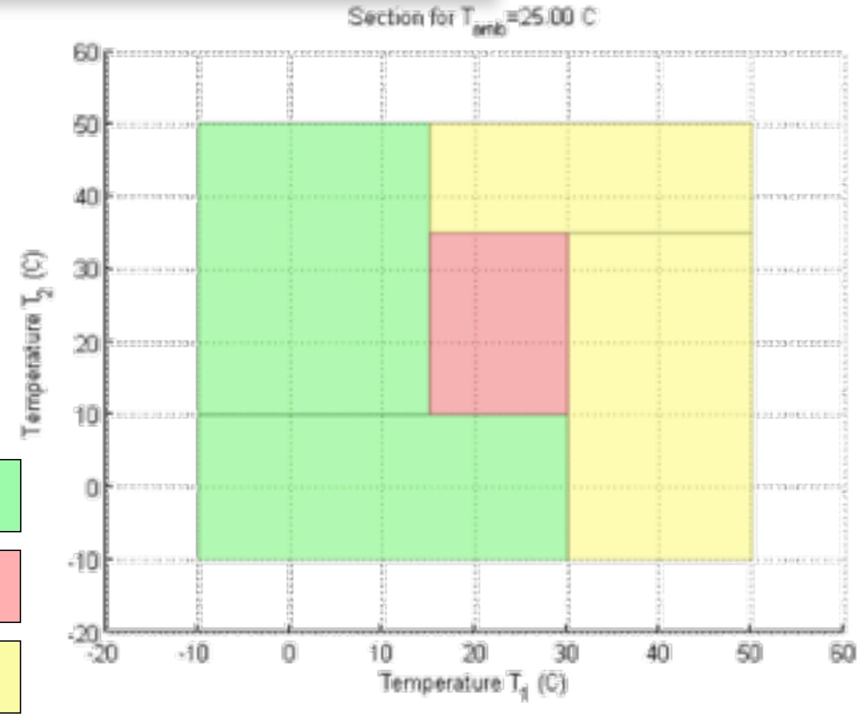
- 1 continuous input:
  (room temperature T_{amb})



heater on

both off

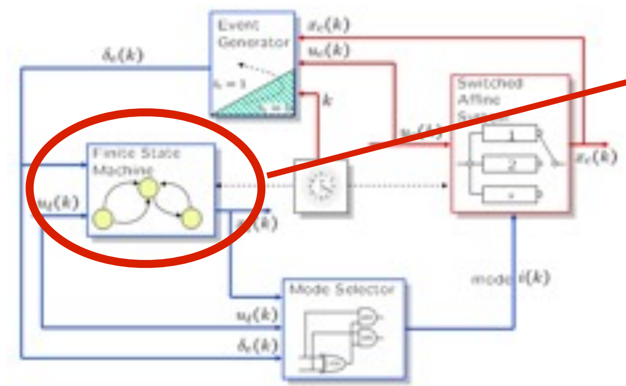
A/C on



- 5 polyhedral regions
  (partition does not depend on input)

DHA extensions

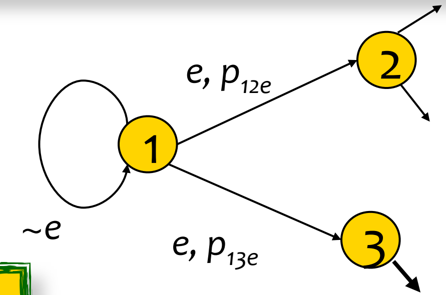
Discrete-time Hybrid Stochastic Automaton (DHSA)



Stochastic Finite State Machine (sFSM)

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

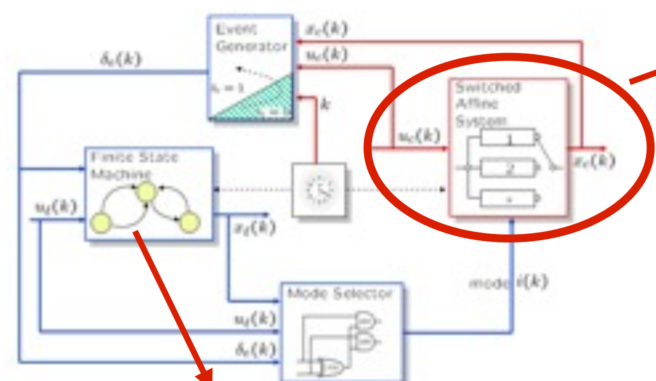
k = discrete-time counter



(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**

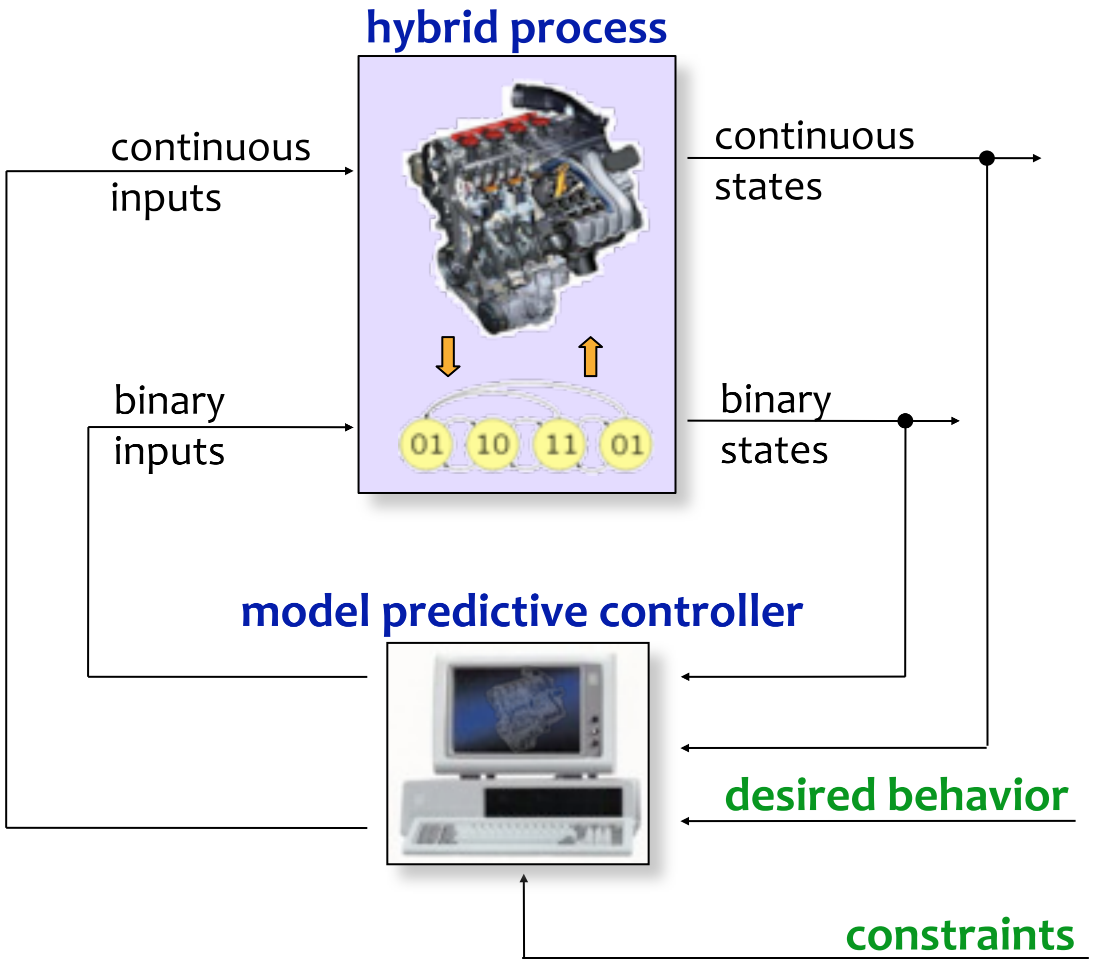
u_c constant between events k and k+1

(Bemporad, Di Cairano, Julvez, 2009)

Talk outline

- ✓ Models of hybrid systems
- Model predictive control (MPC) of hybrid systems
- Automotive applications
- New research directions

Model Predictive Control (MPC)



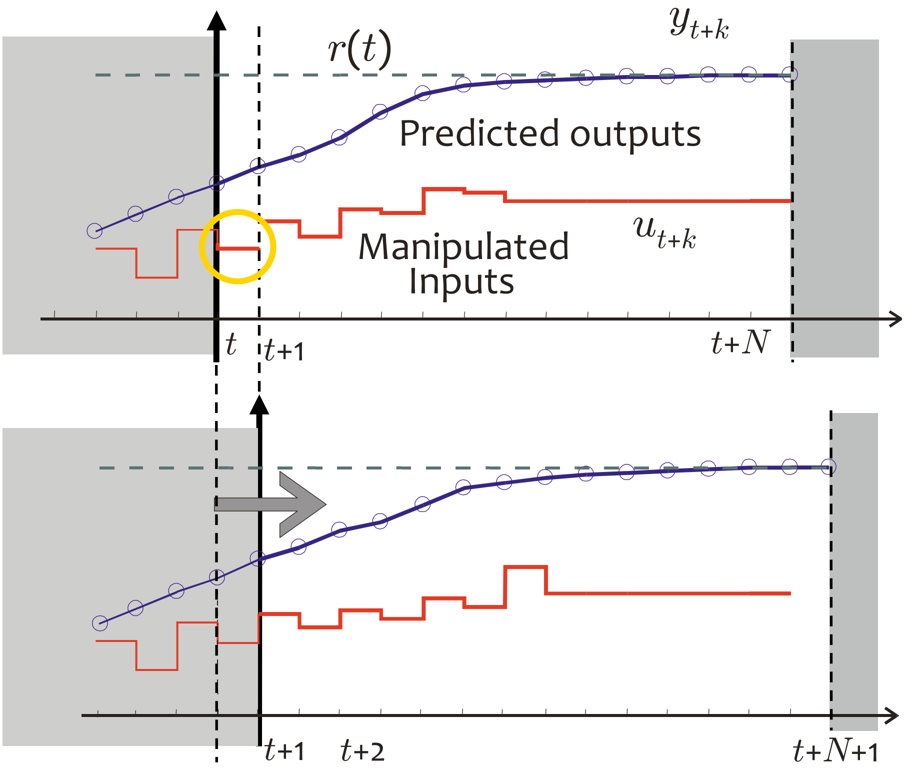
A **model** of the process is used to **predict** the future evolution of the process to decide the **control** signal

Receding horizon philosophy

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

$$\min_{u_t,\ldots,u_{t+N-1}} \left\{ \sum_{k=0}^{N-1} \|y_{t+k} - r(t)\|^2 + \rho\|u_{t+k} - u_r(t)\|^2 \right\}$$

$$\begin{aligned}
\text{s.t.} \quad & x_{t+k+1} = f(x_{t+k}, u_{t+k}) \\
& y_{t+k} = g(x_{t+k}, u_{t+k}) \\
& u_{\min} \le u_{t+k} \le u_{\max} \\
& y_{\min} \le y_{t+k} \le y_{\max} \\
& x_t = x(t), \ k = 0, \ldots, N-1
\end{aligned}$$



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

- At time t+1: Get new measurements, repeat the optimization. And so on …

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

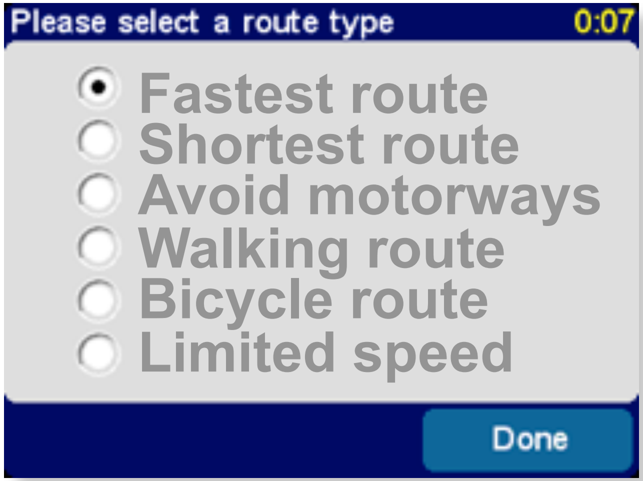
Receding horizon example

- **prediction model** how vehicle moves on the map
- **constraints** drive on roads, respect one-way roads, etc.
- **disturbances** mainly driver's inattention !
- **set point** desired location
- **cost function** minimum time, minimum distance, etc.
- **receding horizon mechanism**

  event-based
  (optimal route re-planned when path is lost)

x = GPS position

u = navigation commands



MPC of hybrid systems

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

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

(Bemporad, Morari, 1999)

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

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

Mixed Integer Quadratic Program (MIQP)

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

<u>Alternative</u>: Mixed-integer **linear** (MILP) formulations

Closed-loop convergence results of hybrid MPC available

(Bemporad, Morari 1999) (Lazar, Heemels, Weiland, Bemporad, 2006) (Di Cairano, Lazar, Bemporad, Heemels, 2008)

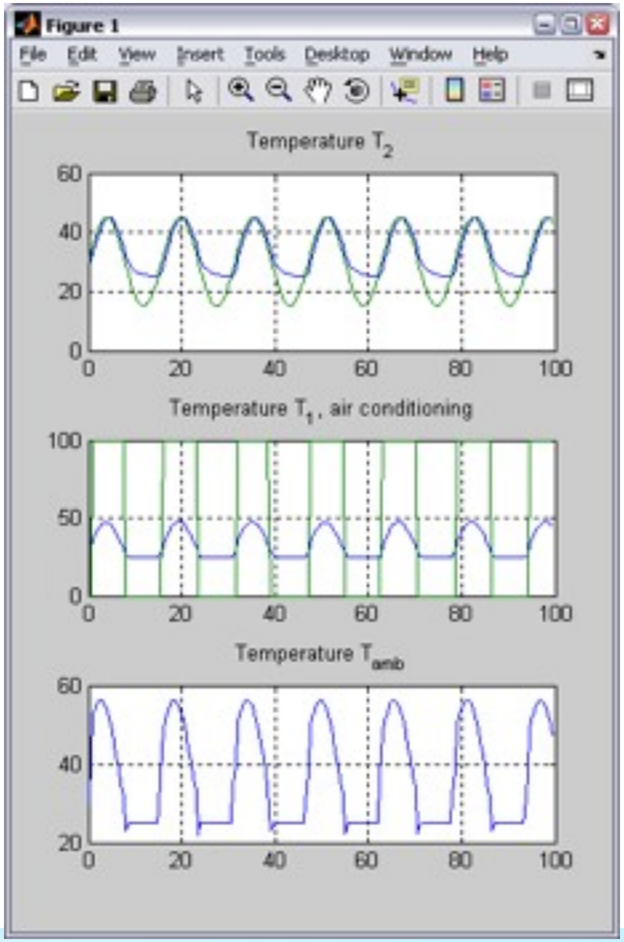
Hybrid MPC – Room temperature example

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

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

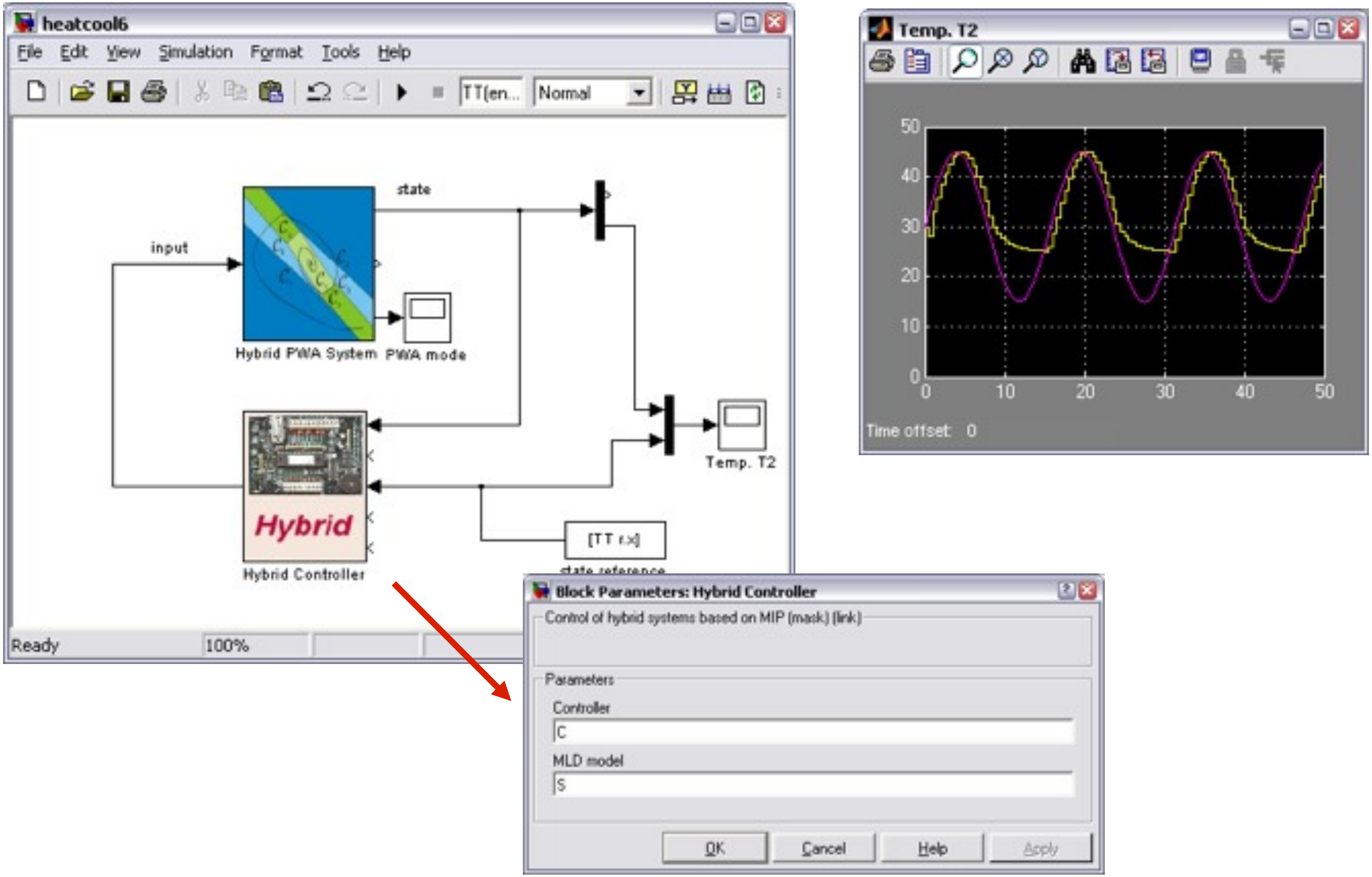
```
>>C=hybcon(S,Q,N,limits,refs);
```

```
>> 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 – Room temperature example



Main drawbacks of on-line implementation

- Excellent LP/QP/MIP/NLP solvers exist today (*"LP is a technology" – S. Boyd*)

but ...

- Computation time may be too long: ok for large sampling times (>0.1s) but not for fast-sampling applications (<1ms). Worst-case CPU time hard to estimate

- Requires relatively expensive hardware (not suitable on inexpensive 8-bit μ-controllers with few kB RAM)

- Software complexity: control profile $u(x)$ hard to understand, solver code difficult to certify (bad in safety critical apps)

Any way to use MPC **without** on-line solvers ?

Explicit model predictive control

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

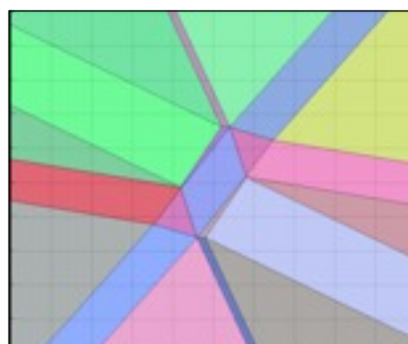
Idea: solve the QP **for all** $x(t)$ within a given range of $\mathbb{R}^n$ **off-line**

→ multi-parametric programming problem

The linear MPC controller is a **continuous piecewise affine** function of the state vector

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

(Bemporad *et al.*, 2002)



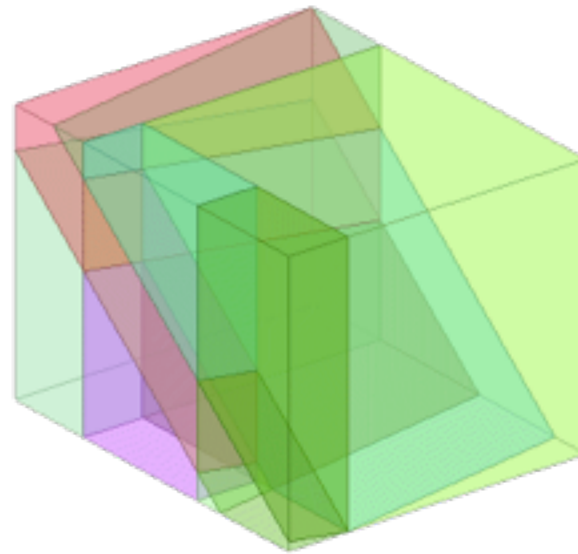
```
while ((num<EXPCON_REG) && check) {
    isinside=1;
    while ((i1<=i2) && isinside) {
        aux=0;
        for (j=0;j<EXPCON_NTH;j++)
            aux+=(double)EXPCON_H[i1+j*EXPCON_NH]*th[j];
        if (aux>(double)EXPCON_K[i1])
            isinside=0; /* get out of the loop, th violates
    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

}
```

Explicit hybrid model predictive control

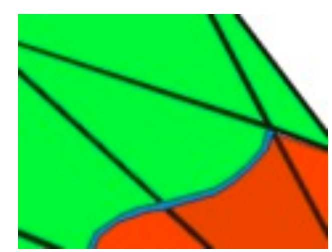
- The **hybrid MPC** controller is **piecewise affine in** x,r (control law may be discontinuous)

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

(Borrelli, Baotic, Bemporad, Morari, *Automatica*, 2005)
(Mayne, ECC 2001) (Alessio, Bemporad, ADHS 2006)



Note: With quadratic costs, partition may not be fully polyhedral. Then better keep overlapping polyhedra

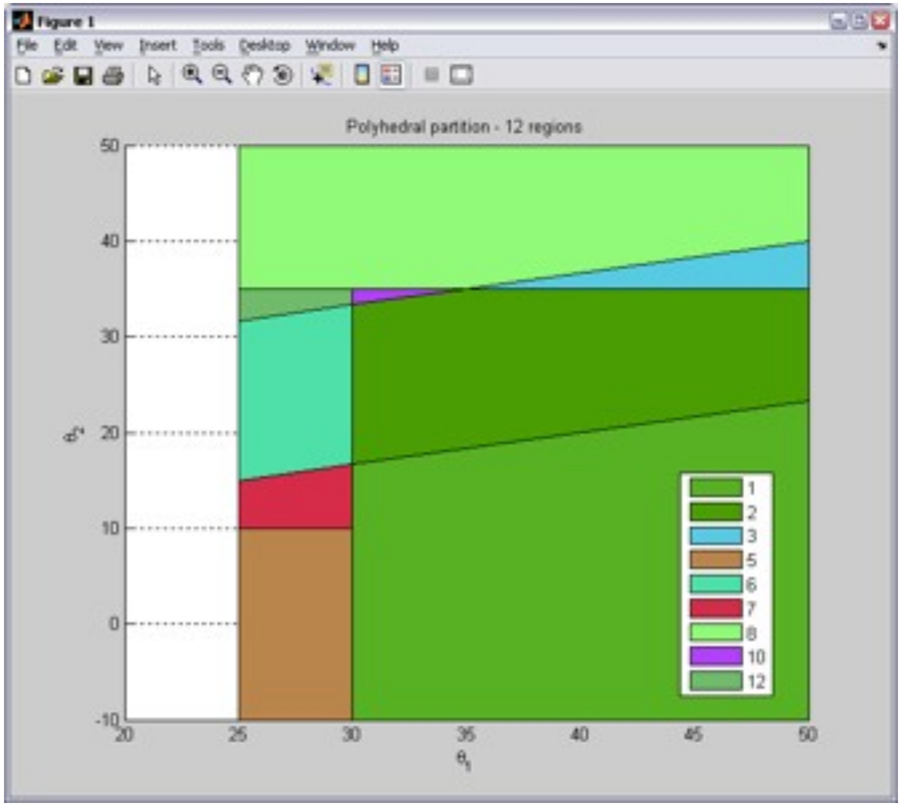


Explicit Hybrid MPC – Room temperature

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

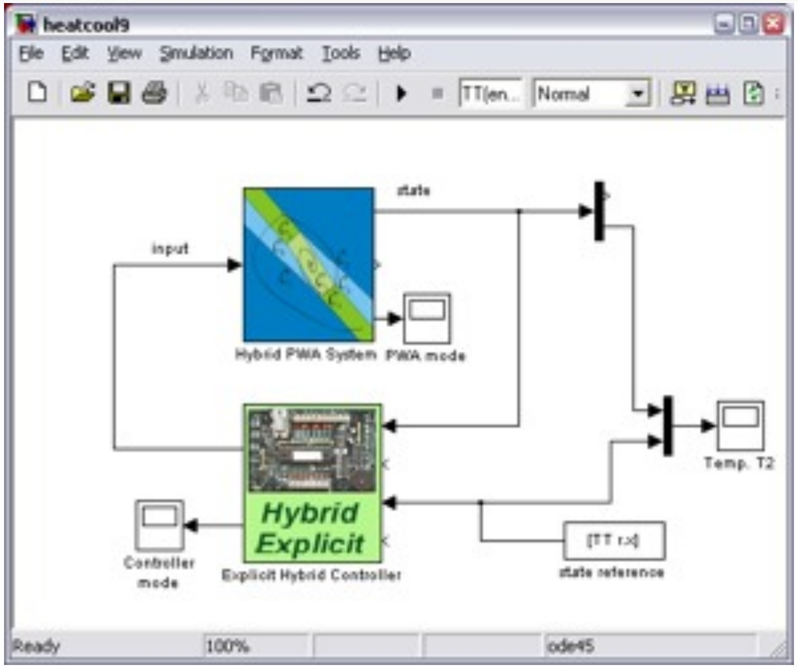
```
>> E

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.
>>
```

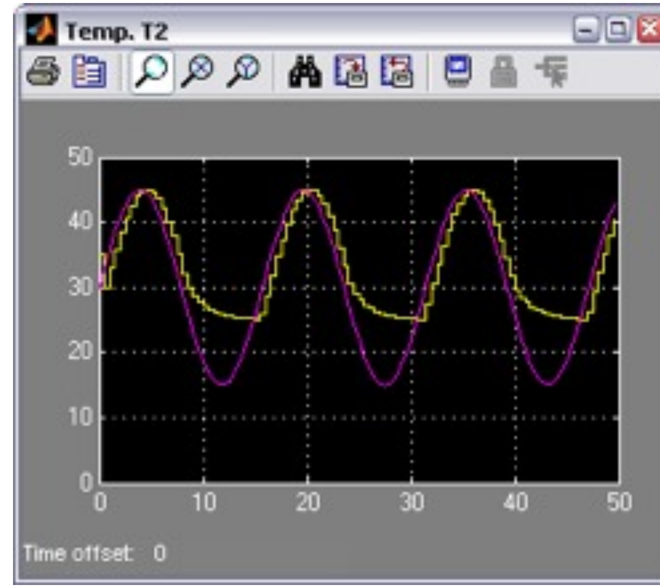
$$
\begin{aligned}
\min \quad & \sum_{k=1}^{2} x_2^2(k) \\
\text{s.t.} \quad & x_1(k) \geq 25 \quad k = 1, 2 \\
& \text{PWA model}
\end{aligned}
$$



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

Explicit Hybrid MPC – Room temperature

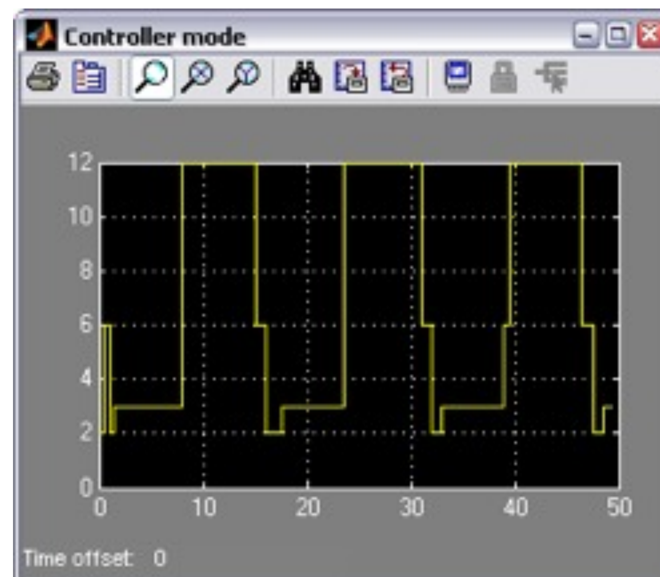




generated C-code

utils/expcon.h

```c
#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,
```



Talk outline

✓ Models of hybrid systems

✓ Model predictive control (MPC) of hybrid systems

- Automotive applications

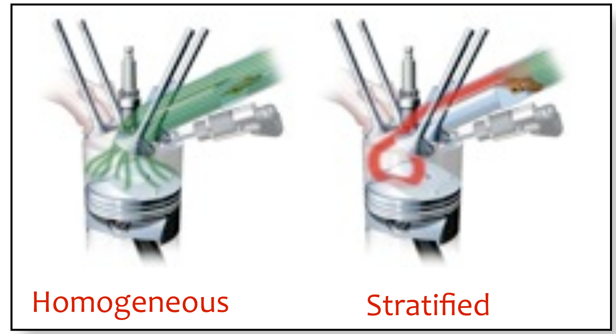
- New research directions

Automotive applications of MPC

PhD students: Bernardini, Borrelli, Di Cairano, Giorgetti, Ripaccioli, Trimboli (2001-2009)

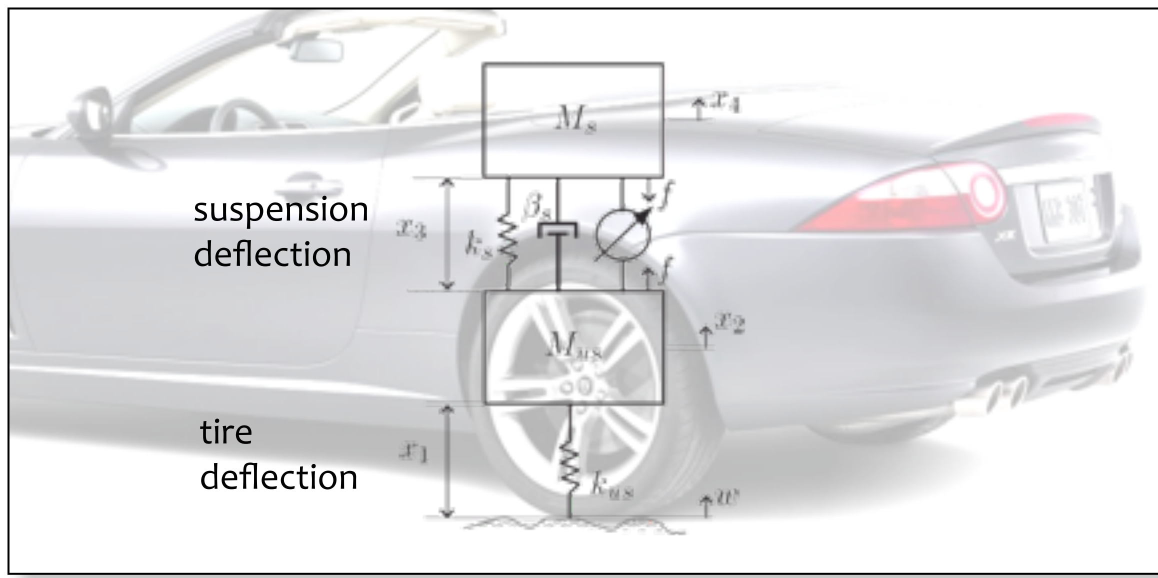
& Hrovat, Kolmanovsky, Tseng (Ford)

traction control



engine control

idle speed control

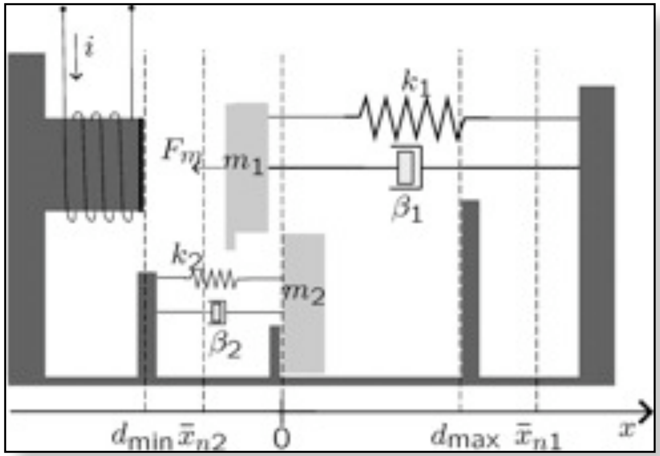


semiactive suspensions

Automotive applications of MPC

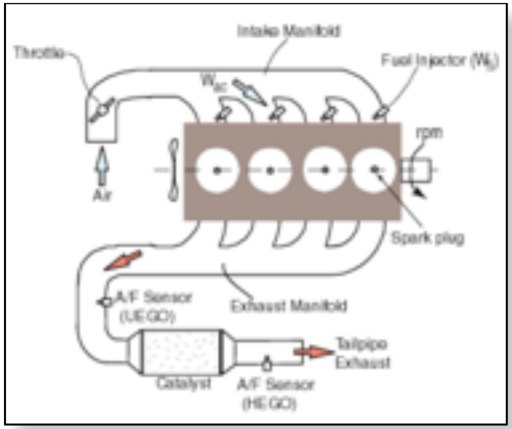
(PhD students: Bernardini, Borrelli, Di Cairano, Giorgetti, Ripaccioli, Trimboli - 2001-2009)

& Hrovat, Kolmanovsky, Tseng (Ford)



magnetic actuators

hybrid electric vehicles



air-to-fuel ratio

active steering

robotized gearbox

CENTRO RICERCHE FIAT

(Borodani, Mannelli, CRF)

Energy management of HEVs

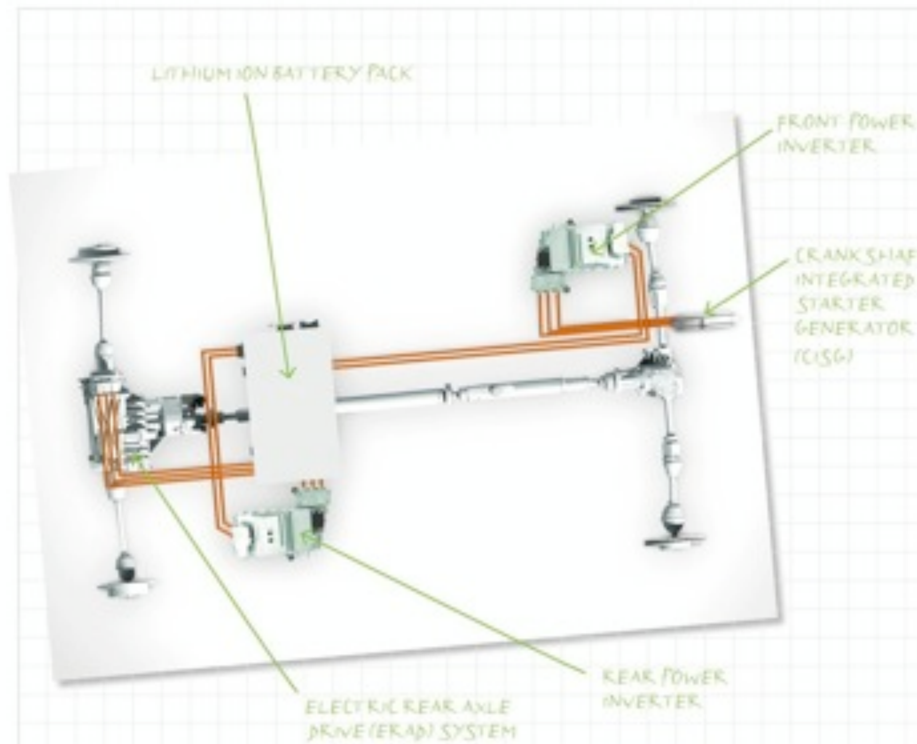
Why Hybrid Electric Vehicles (HEV) ?

- rising fuel prices
- tightening emission regulations
- performance improvement

Advanced Power Systems

- more components
- more controlled variables
- more constraints
- several operating modes

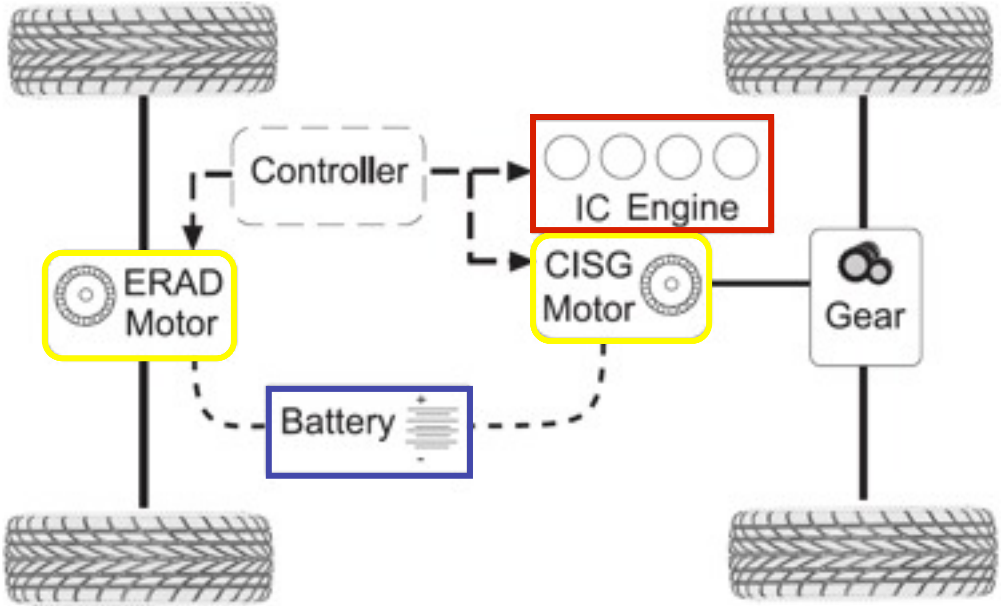
Use MPC as systematic modeling and model-based control approach



Hybrid MPC for Energy management of HEVs

High-fidelity industrial model of an advanced 4x4 hybrid electric car with:

- ✓ turbocharged diesel engine
- ✓ high voltage NiMH battery
- ✓ two electric motors acting (one per axle)



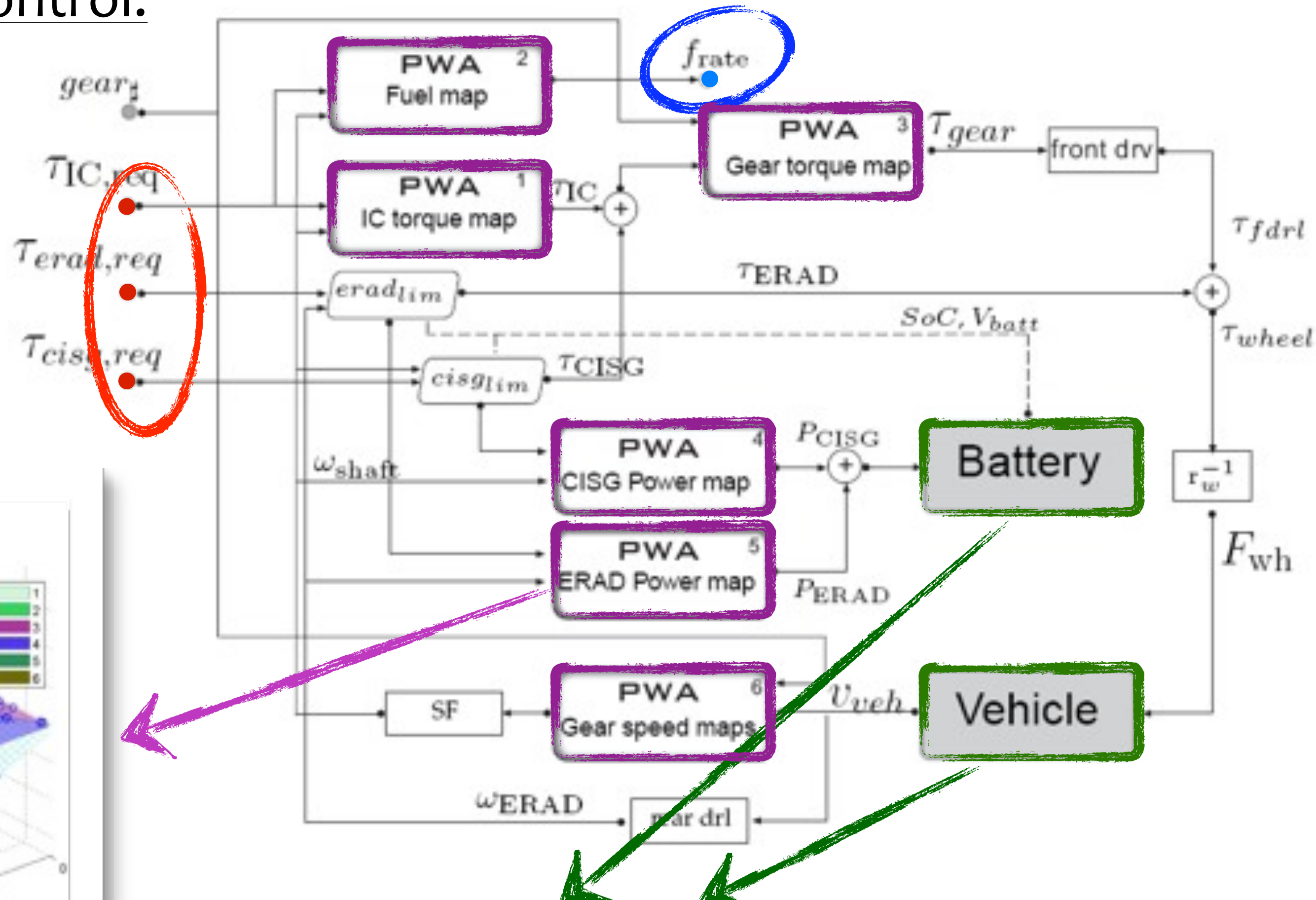
Objectives:

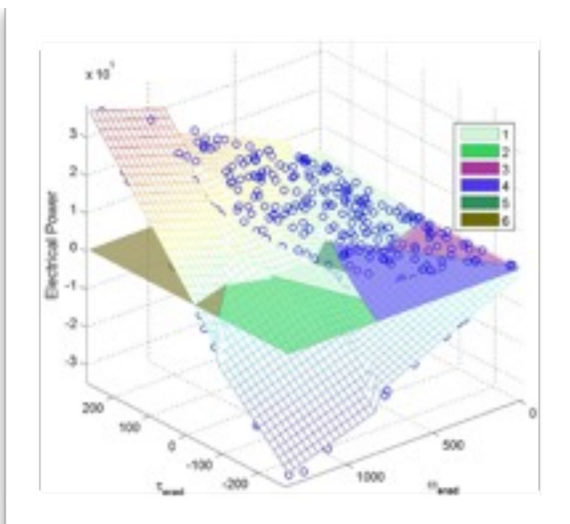
- manage IC, ERAD, and CISG power requests
- minimize fuel consumption
- keep battery preferably within 30-70% of full charge
- fulfill various mechanical & electrical constraints

Hybrid MPC for Energy management of HEVs

HEV model for MPC control:

- 3 real inputs (torques)
- 1 real output (fuel)
- 9 real states (dynamics)
- 7 binary states (gears)
- 32 binary inputs (maps)





≈ piecewise affine maps

(Bemporad *et al.*, IEEE TAC, 2005)

$$\frac{dx(t)}{dt} = Ax(t) + Bu(t)$$

≈ linear dynamics

MLD hybrid model:

- 56 aux. real variables
- 490 mixed integer ineq.

Hybrid MPC for Energy management of HEVs

- Hybrid optimal control problem for MPC:

$$\min_{\xi} J(\xi, x(t)) \triangleq Q_\rho \rho^2 + \sum_{k=1}^{N} (\Gamma_x x_k - x_{ref})^T S (\Gamma_x x_k - x_{ref}) + \\ + \sum_{k=0}^{N-1} (\Gamma_u u_k - u_{ref})^T R (\Gamma_u u_k - u_{ref}) + (y_k - y_{ref})^T Q (y_k - y_{ref})$$

$$\text{subj. to} \begin{cases} x_0 = x(t) \\ x_{k+1} = A x_k + B_1 u_k + B_3 z_k \\ y_k = C x_k + D_1 u_k + D_3 z_k \\ E_3 z_k \leq E_1 u_k + E_4 x_k + E_5 \\ 0.3 - \rho \leq SoC_k \leq 0.7 + \rho \end{cases}$$

hybrid MLD model

constraint on SoC

- Set points:

$$y_{ref} \triangleq f_{\text{rate,ref}}$$
$$u_{ref} \triangleq [\tau_{\text{IC,req}} \ \tau_{\text{ERAD,req}} \ \tau_{\text{CISG,req}}]'$$
$$x_{ref} \triangleq [v_{\text{veh,ref}} \ SoC_{\text{ref}} \ 0]'$$

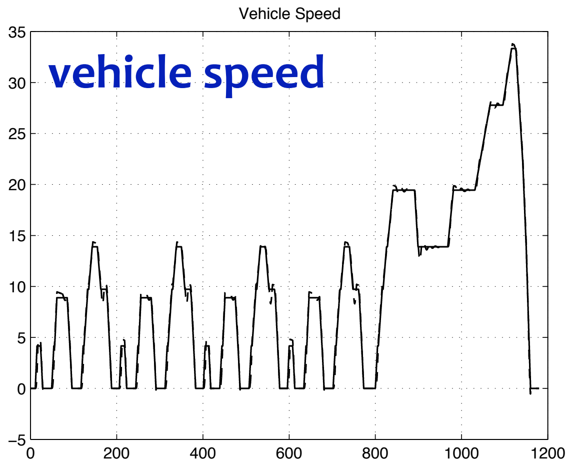
- Weights:

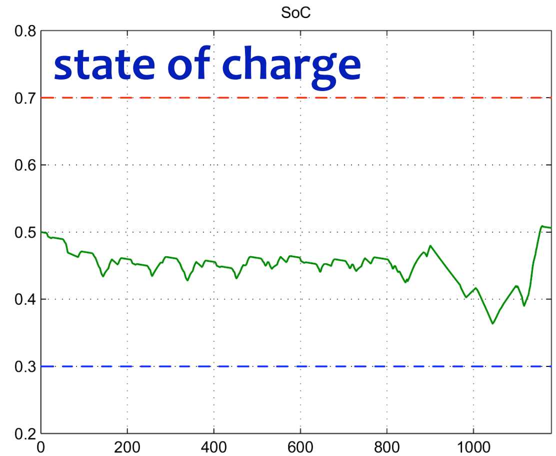
$$Q = q_{\text{fuel}}, \ R = \begin{bmatrix} r_{\tau,\text{IC}} & 0 & 0 \\ 0 & r_{\tau,\text{CISG}} & 0 \\ 0 & 0 & r_{\tau,\text{ERAD}} \end{bmatrix} \ S = \begin{bmatrix} s_{\text{v,veh}} & 0 & 0 \\ 0 & s_{\text{SoC}} & 0 \\ 0 & 0 & s_{\text{v,int}} \end{bmatrix}, \ Q_\rho = 10^5$$

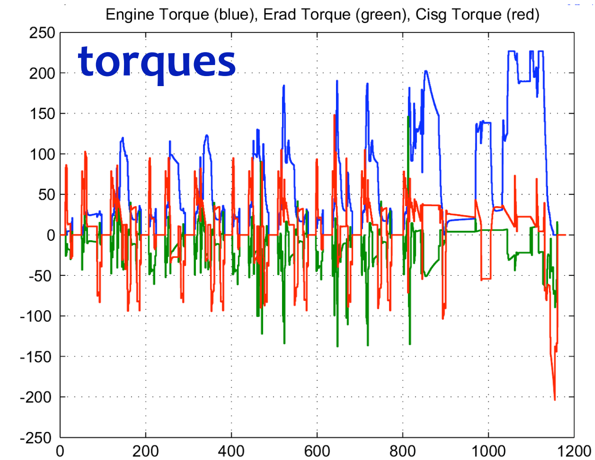
Hybrid MPC for Energy management of HEVs

- Simulations (MPC#1)

(Ripaccioli, Bemporad, Assadian, Dextreit, Di Cairano, Kolmanovsky, HSCC 2009)







- CPU time: average 0.13 s per time step (worst: 0.29 s) on PC 2GHz + CPLEX
- Simulations based on high-fidelity nonlinear model + hybrid MPC

- Performance analysis

	q_{fuel}	s_{SoC}	$s_{\text{v,int}}$	fuel cons (norm)	$\max\|v_{veh} - v_{veh,ref}\|$	$\max\|SoC - SoC_{ref}\|$	
0	*	*	*	1	*	*	conventional vehicle
1	1e-2	2e6	10	0.79	2.105	0.1364	MPC #1
2	1e1	1e6	1	0.76	2.789	0.2484	MPC #2

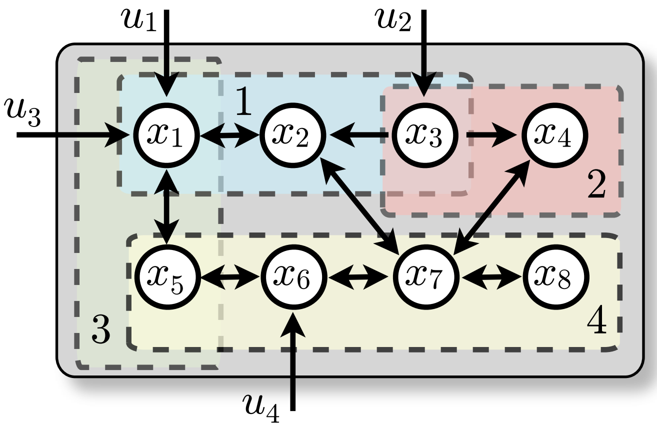
Note: driving cycle not known in advance !

Talk outline

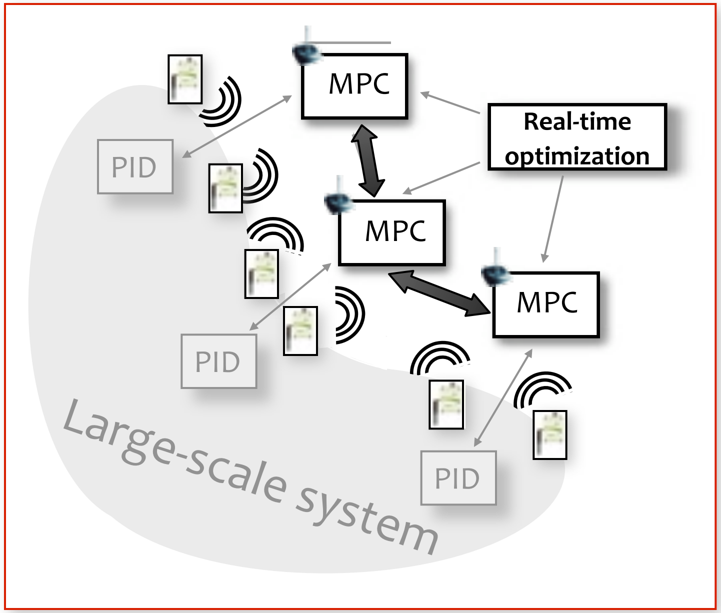
- ✓ Models of hybrid systems
- ✓ Model predictive control (MPC) of hybrid systems
- ✓ Automotive applications
- New research directions

MPC in Wireless Automation

- Characterized by **spatial distribution**
- Need for **decentralized control** computations



Decentralized MPC



- Need for **flexible information gathering** system

Wireless sensor networks

- Wireless communications introduce **new problems in control design** to take care of (packet drops, delays & jitter, battery energy consumption, etc.)

MPC in Wireless Automation

SP100 - The Standard for Industrial Wireless: Interoperability / Coexistence / WSN

Companies/organizations participating in this Working Group include:

3e Technologies International	EPRI Charlotte	Phoenix Contact
Adalet Wireless	ESAII-UPC	Proto-Power
Adaptive Instruments	ESensors	ProSoft Technology
Advanced Industrial Networks	Exxon/Mobil	Putnam Media
Apprion	Frontline Test Equipment	Rice Lake Weighing Systems
ARC Advisory Group	GE Global Research	Rockwell Automation Global
Argonne National Laboratory	General Monitors	Rosemount
Aujas Systems	Honeywell	Safety Control Solutions
Automation Electronics	IoSelect	Saudi Aramco
Automation World	Invensys	Schneider Electric
Bayer	Kinney Consulting	Sensicast Systems
Boeing	Lyondell Equistar Chemicals	Shell Global Solutions
BP America	MaCT USA	Shindengen America
Cambridge Silicon Radio	Michigan Technological University	SMAR International
Chevron/Texaco	Motorola	Smart Sensor Systems
CMC Associates	NIST	StatSignal Systems
Compressor Controls	NuFlo Measurement Systems	Syncrude Canada
Crossbow Technology	Oak Ridge National Labs	UniTorq Actuators
Dust Networks	Occidental Petroleum Qatar	University of Alabama
Eaton	Oceana Sensor	Wunderlich-Malec Engineering
ELPRO Technologies	Omnex Controls	Yokogawa Electric
Emerson Process Management	OPTI Canada	Zone Automation
Endress+Hauser	Parsons Brinckerhoff	

www.isa.org/community/sp100

European project “WIDE”

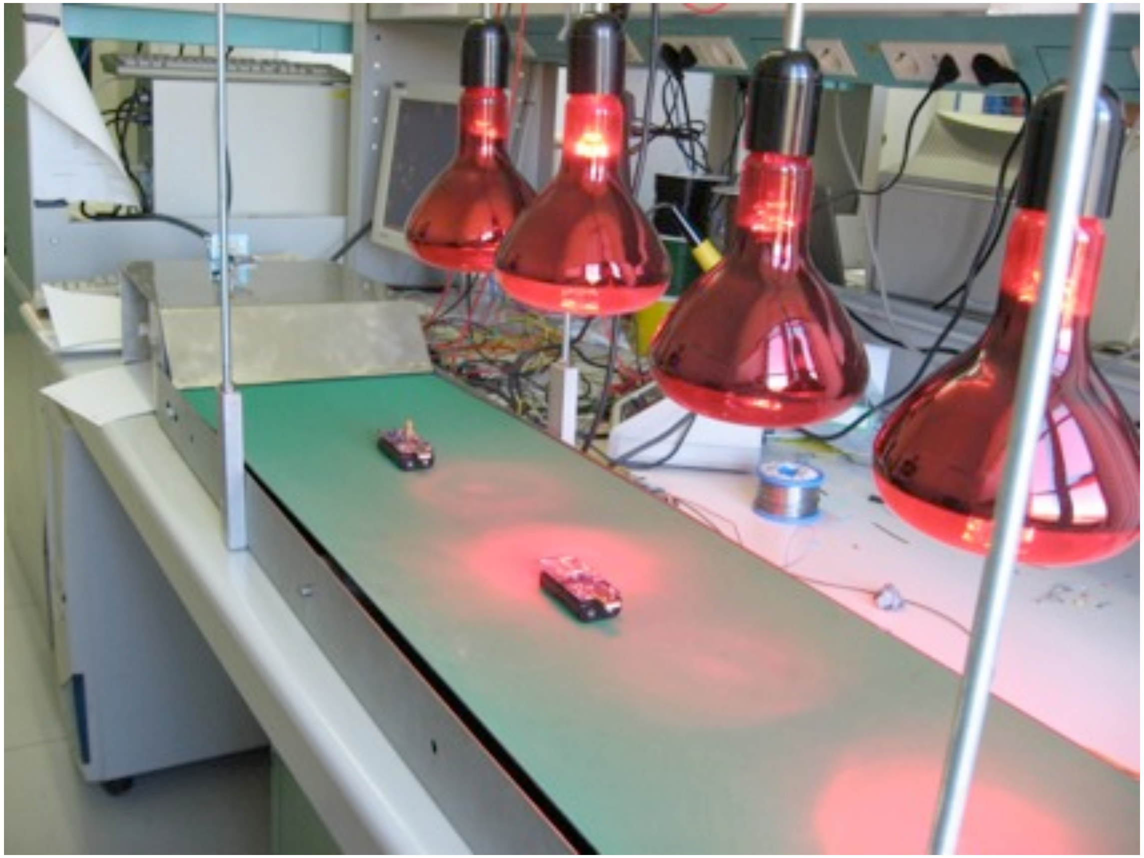
DEcentralized and WIreless Control of Large-scale Systems

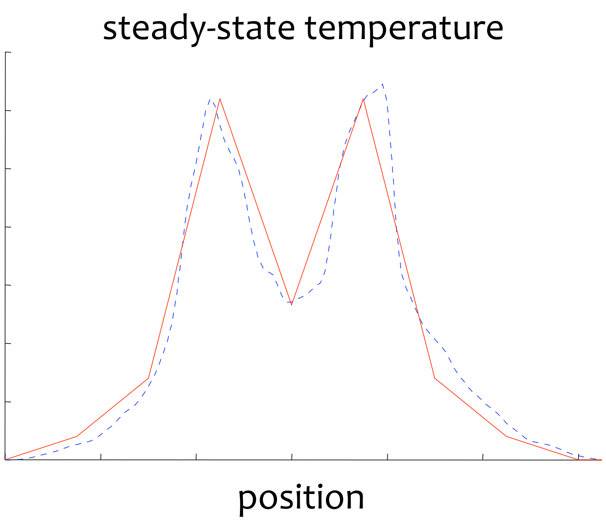
- ICT-FP7-Call 2 (2008-2011), total budget 2.7M€. Started Sept. 1, 2008
- Objective ICT-2007.3.7 “Networked Embedded and Control Systems”

3rd WIDE PhD School on Networked Control Systems, Siena, July 7-9, 2009

http://ist-wide.dii.unisi.it

MPC in Wireless Automation





Hybrid MPC problem:
- 2 binary inputs (lamps)
- 1 continuous input (speed)
- PWL state function (heating)
- Outputs: temp, position
- Sampling = 4Hz

Objective: track **position** and **temperature** references while enforcing safety constraints

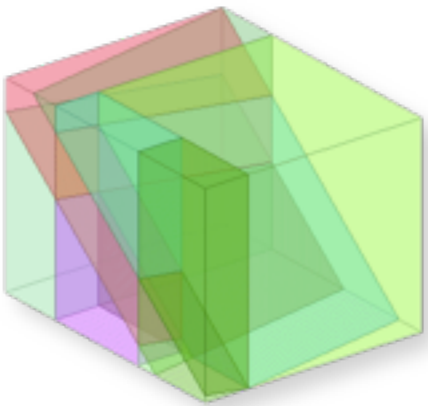
Conclusions

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

- **Model predictive control (MPC)** is a rather general and systematic control design methodology for multi-variable systems with constraints

- **Implementation: (1)** *on-line* optimization (QP, mixed-integer), or
  (2) *off-line* multi-parametric optimization and PWL control

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



- **Matlab/Simulink tools** are available to assist the whole design process

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