Model Predictive Control for Cyber-Physical Systems

Alberto Bemporad

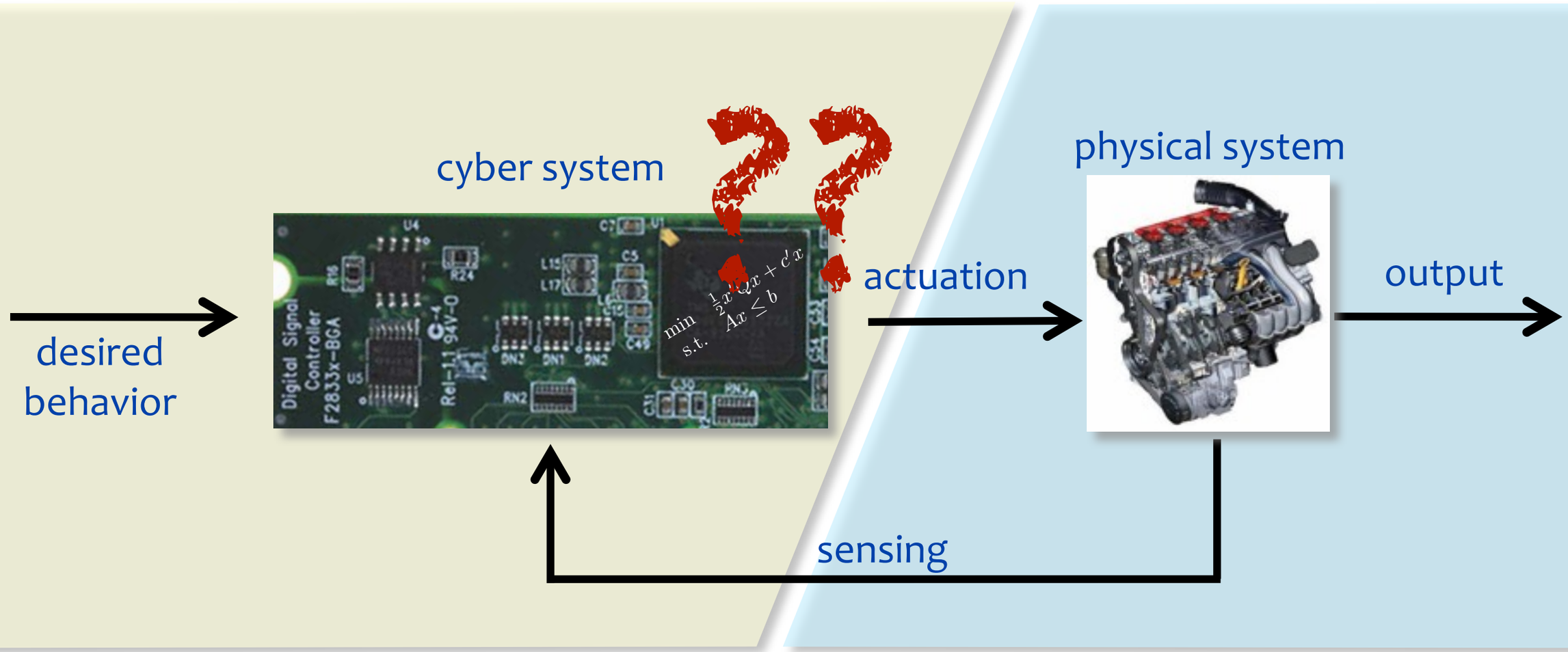
http://imtlucca.it/alberto.bemporad

IMT INSTITUTE FOR ADVANCED STUDIES LUCCA

ODYS Advanced Controls & Optimization

www.odys.it

Role of control in Cyber-Physical Systems



What kind of "intelligence" to embed in the "cyber" component to make the overall CPS behave autonomously, robustly, safely, and optimally ?

Model Predictive Control (MPC)

Quadratic Program (QP)

$$\begin{aligned} \min_z \quad & \frac{1}{2} z'Qz + c'z \\ \text{s.t.} \quad & Az \le b \end{aligned}$$

embedded model-based optimizer

reference $r(t)$ → input $u(t)$ → process → output $y(t)$

measurements

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

Model Predictive Control (MPC)

- At time t: consider optimal control problem over a future horizon of N steps

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

$$\text{s.t.} \quad x_{k+1} = Ax_k + Bu_k$$

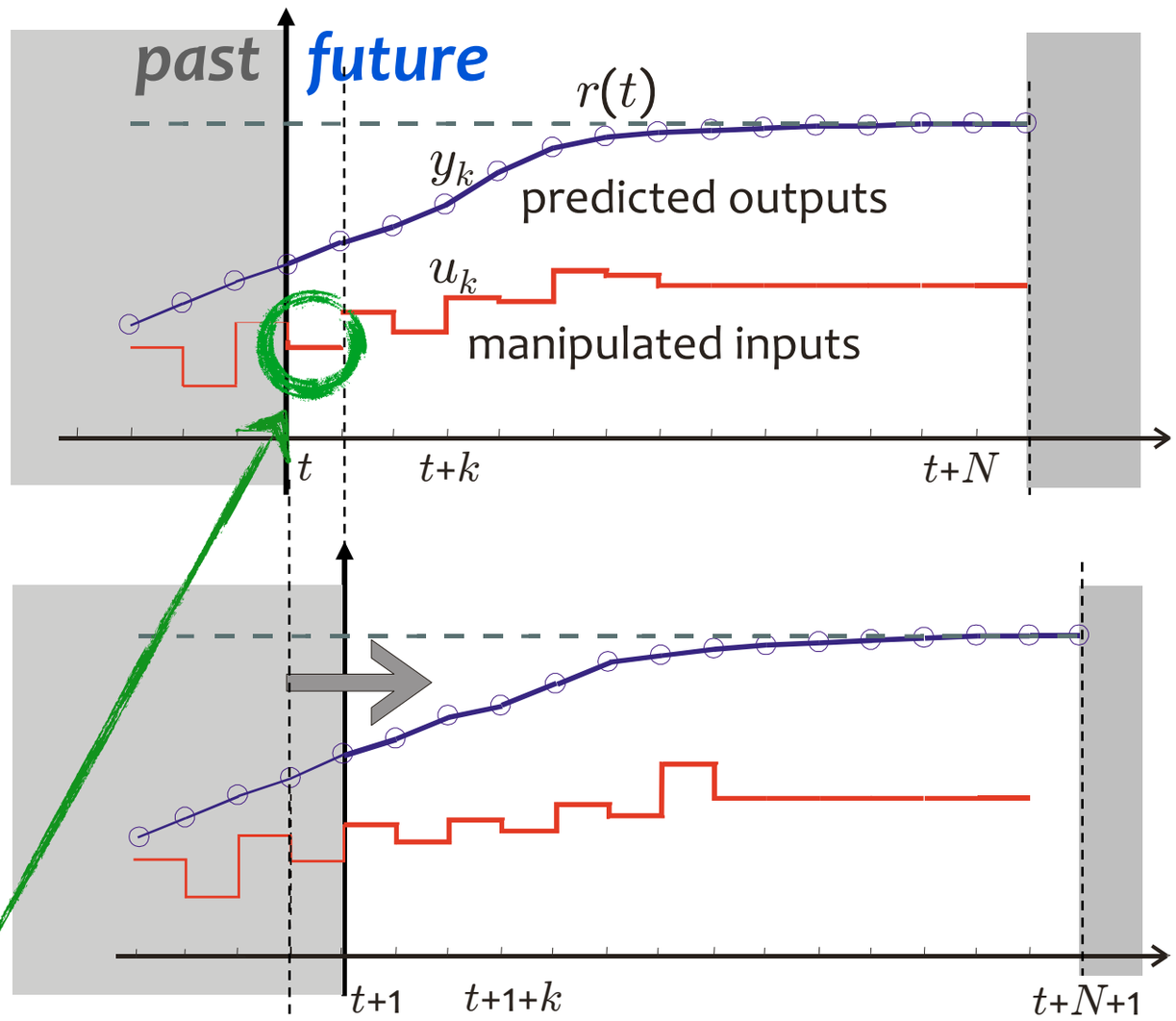
$$y_k = Cx_k$$

$$u_{\min} \le u_k \le u_{\max}$$

$$y_{\min} \le y_k \le y_{\max}$$

$$x_0 = x(t) \quad \leftarrow \text{feedback !}$$

Quadratic Program (QP)



- Solve QP w.r.t. $\{u_0,\ldots,u_{N-1}\}$
- Apply the first optimal move $u(t)=u_0^*$, throw the rest of the sequence away
- At time $t+1$: Get new measurements, repeat the optimization. And so on …

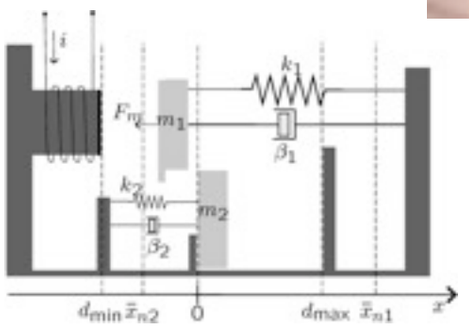
Used in process industries since the 80's. Now spreading in auto & aero

MPC of CPS: Automotive systems

Bemporad, Bernardini, Borrelli, Cimini, Di Cairano, Esen, Giorgetti, Hrovat, Kolmanovsky, Ripaccioli, Trimboli, Tseng, Yanakiev, … (2001-2015)

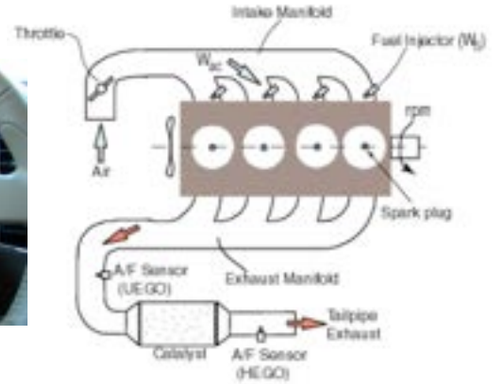
Powertrain
- direct-inj. engine control
- A/F ratio control
- magnetic actuators
- robotized gearbox
- power MGT in HEVs
- cabin heat control in HEVs
- electrical motors

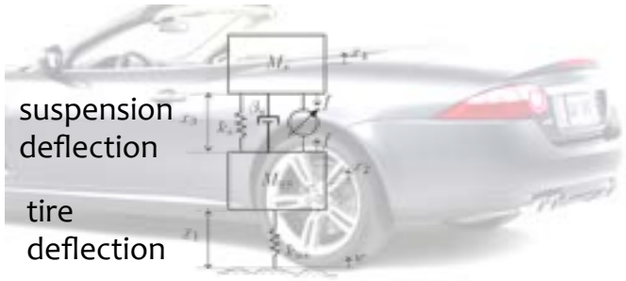
Vehicle dynamics
- traction control
- active steering
- semiactive suspensions



Homogeneous

Stratified





MPC of CPS: Aerospace systems

- ESA projects funded to explore MPC capabilities in new space apps:
  - cooperating UAVs, planetary rover locomotion, impulsive hopping
  - powered descent, sloshing-aware attitude control

- New MATLAB MPC Toolboxes developed for ESA (Bemporad, 2010) (Bemporad, 2012)

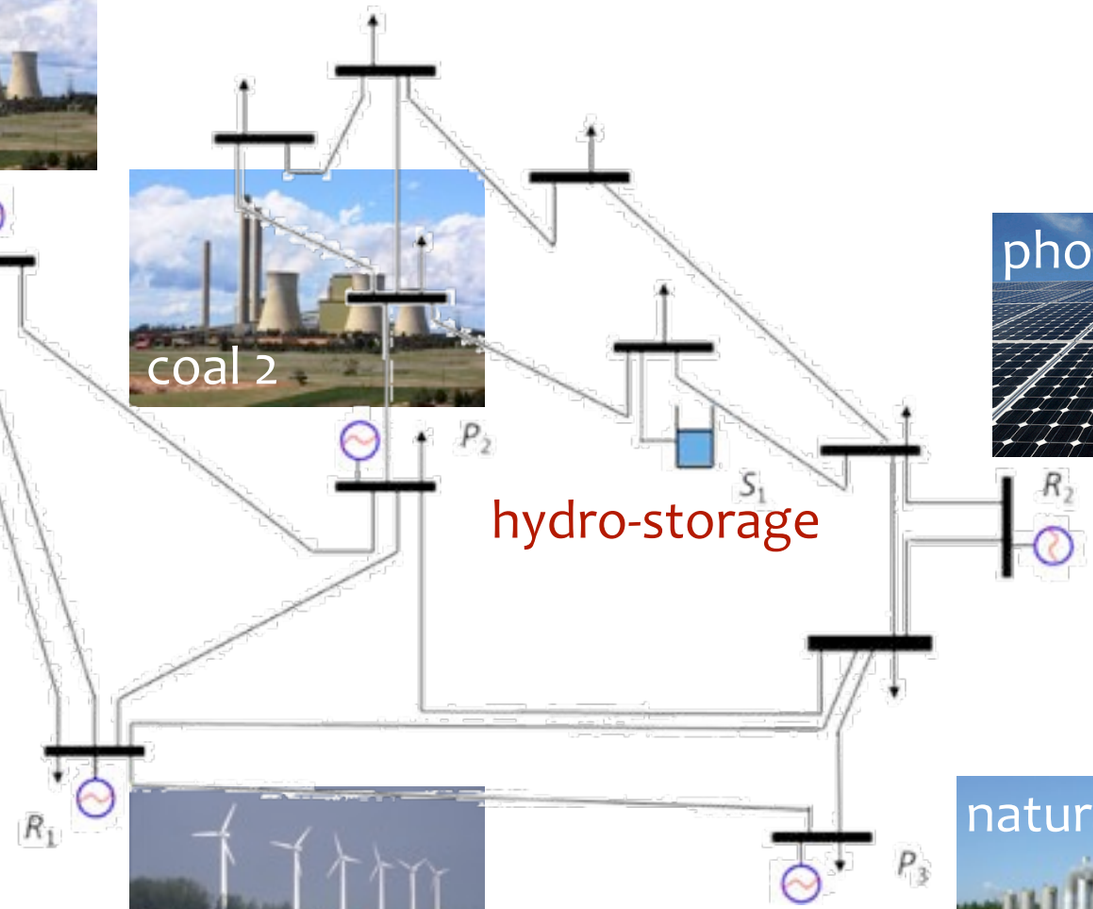
Curiosity Rover

Mission Animation

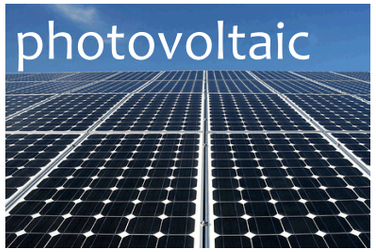
NASA Mars Science Laboratory

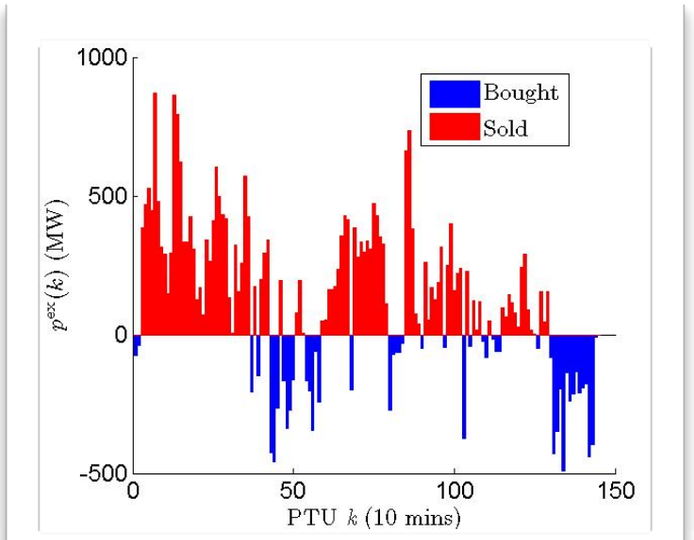
(Açıkmeşe et al.)

MPC of CPS: Smart Grids



hydro-storage

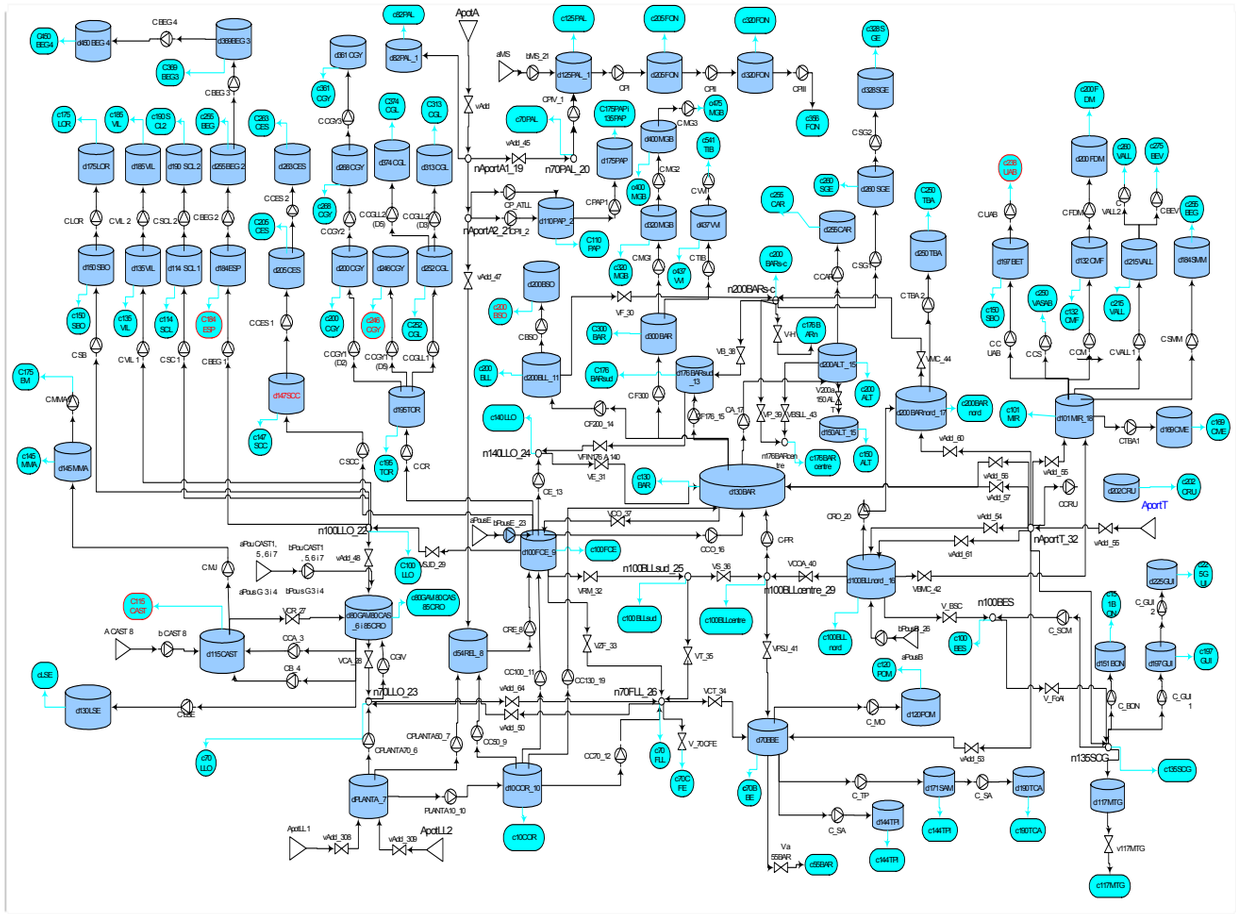




- **Aggregate distributed energy resources**, including **renewables**, to trade energy on the day-ahead and real-time **markets**
- **Dispatch power** in smart distribution grids in real-time

MPC of CPS: Water distribution systems

Drinking water network of Barcelona:
(81 tanks, 64 valves, 180 pumps)



- Control of large-scale **urban drinking water networks** (flows, storage, constraints)
- **Optimize** network's operating costs under **stochastic** uncertainty (water demand and energy prices)

MPC in industrial automotive production

ODYS
Advanced Controls & Optimization

Contents of my lecture

- Model Predictive Control (MPC) *for* CPS's
- Embedded quadratic optimization algorithms (*inside* the CPS)
- Hybrid MPC = supervisory control *of* CPSs

MPC for Cyber-Physical Systems

Linear MPC

- Linear prediction model:

$$\begin{cases} x_{k+1} = Ax_k + Bu_k \\ y_k = Cx_k \end{cases}$$

$$x_0 = x(t)$$

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

- Constraints to enforce:

$$\begin{cases} u_{\min} \le u(t) \le u_{\max} \\ y_{\min} \le y(t) \le y_{\max} \end{cases}$$

- Constrained optimal control problem (quadratic performance index):

$$\begin{aligned} \min_z \quad & x_N' P x_N + \sum_{k=0}^{N-1} x_k' Q x_k + u_k' R u_k \\ \text{s.t.} \quad & u_{\min} \le u_k \le u_{\max}, \quad k = 0, \dots, N-1 \\ & y_{\min} \le y_k \le y_{\max}, \quad k = 1, \dots, N \end{aligned}$$

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

Linear MPC

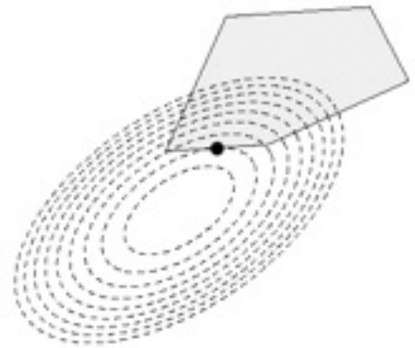
- Optimization problem:

$$V(x_0) = \tfrac{1}{2}x_0' Y x_0 + \begin{array}{ll} \min\limits_{z} & \tfrac{1}{2}z'Hz + x_0'F'z \\ \text{s.t.} & Gz \le W + Sx_0 \end{array}$$

(quadratic)

(linear)

Convex QUADRATIC PROGRAM (QP)

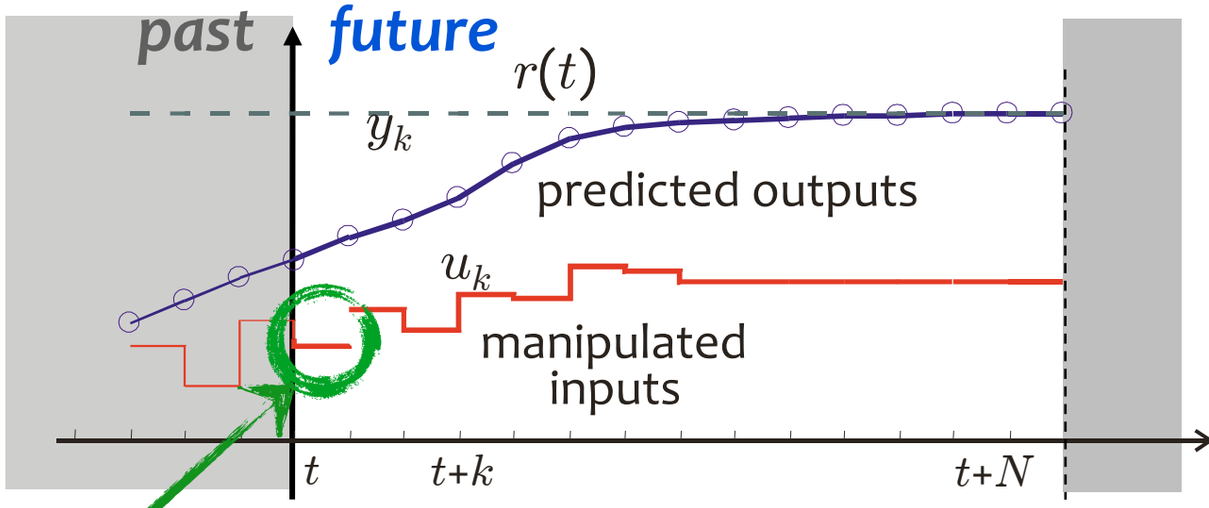


- $z = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix} \in \mathbb{R}^s, \;\; s \triangleq Nm$ is the optimization vector

- $H = H' \succ 0$ and H, F, Y, G, W, S depend on weights Q, R, P, upper and lower bounds $u_{\min}$, $u_{\max}$, $y_{\min}$, $y_{\max}$, and model matrices A, B, C

Linear MPC algorithm

@ each sampling step t:



- Measure (or estimate) the current state $x(t)$
- Get the solution $z^* = \{u_0^*, \ldots, u_{N-1}^*\}$ of the QP
- Apply only $u(t) = u_0^*$, discard remaining optimal inputs $u_1^*, \ldots, u_{N-1}^*$

feedback !

$$\min_z \quad \tfrac{1}{2} z' H z + x'(t) F' z$$

$$\text{s.t.} \quad Gz \le W + Sx(t)$$

Linear MPC - Tracking

- Optimal control problem (quadratic performance index):

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

$$[\Delta u_k \triangleq u_k - u_{k-1}],\ u_{-1} = u(t-1)$$

$$\text{subj. to} \quad u_{\min} \le u_k \le u_{\max},\ k = 0,\dots,N-1$$
$$y_{\min} \le y_k \le y_{\max},\ k = 1,\dots,N$$
$$\Delta u_{\min} \le \Delta u_k \le \Delta u_{\max},\ k = 0,\dots,N-1$$

optimization vector

$$z = \begin{bmatrix} \Delta u_0 \\ \Delta u_1 \\ \vdots \\ \Delta u_{N-1} \end{bmatrix}$$

- Note: $\|Wa\|^2 = (Wa)'(Wa) = a'(W'W)a = a'Qa$

⟹ same formulation as before (W = Cholesky factor of weight matrix Q)

- Optimization problem:

Convex Quadratic Program (QP)

$$\min_z \quad J(z, x(t)) = \tfrac{1}{2}z'Hz + [x'(t)\ \ r'(t)\ \ u'(t-1)]F'z$$

$$\text{s.t.} \quad Gz \le W + S\begin{bmatrix} x(t) \\ r(t) \\ u(t-1) \end{bmatrix}$$

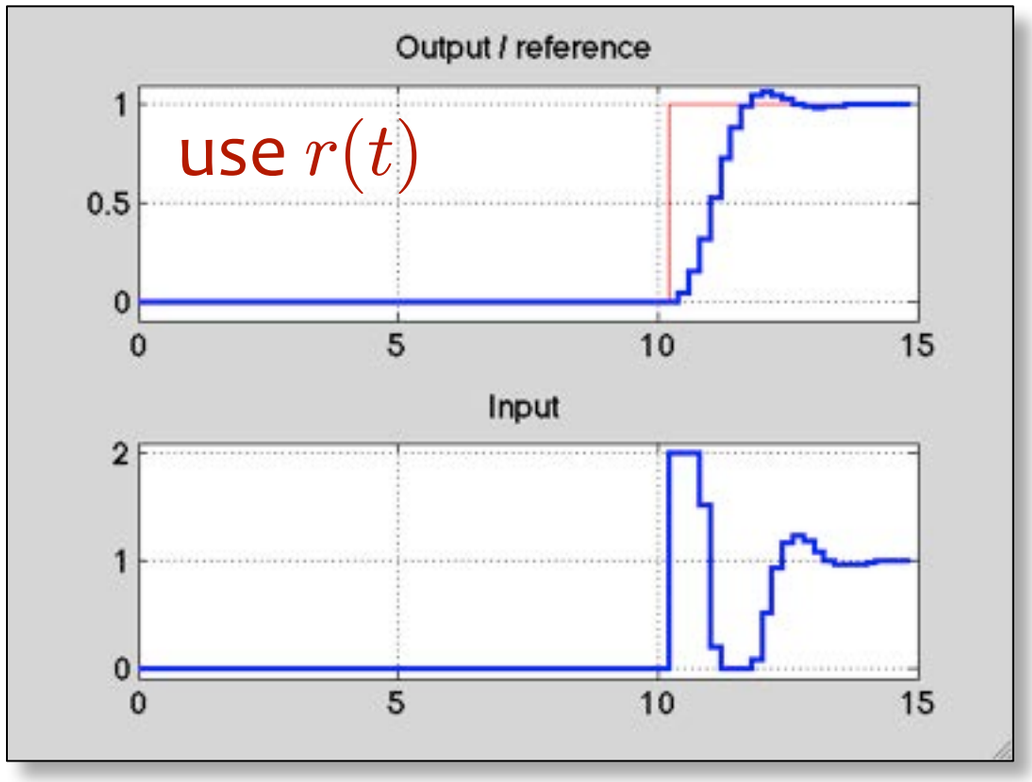
- Constraints on tracking errors can be also included: $e_{\min} \le y_k - r(t) \le e_{\max}$

Anticipative action (a.k.a. "preview")

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

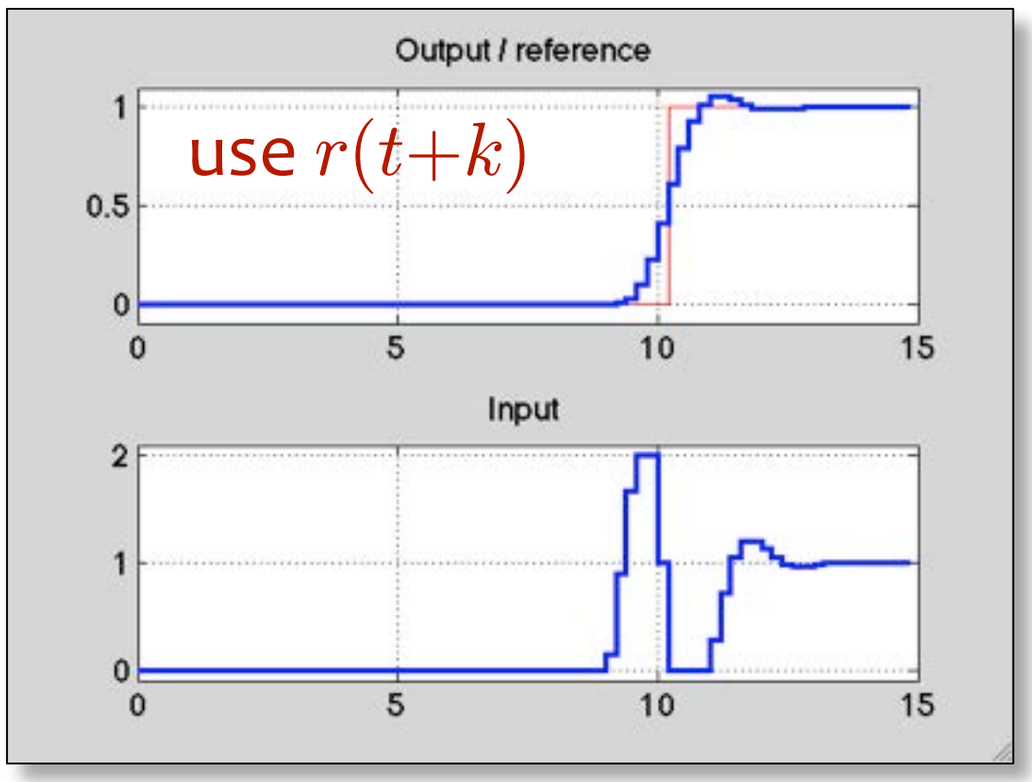
- Reference not known in advance (causal):

$$r_k \equiv r(t), \forall k = 0, \ldots, N-1$$



- Future reference samples (partially) known in advance (anticipative action):

$$r_k = r(t+k), \forall k = 0, \ldots, N-1$$



go to demo **`mpcpreview.m`** (MPC-Tbx)

Model Predictive Control Toolbox

(Bemporad, Ricker, Morari, 1998-2015)

- **MPC Toolbox** (The Mathworks, Inc.)

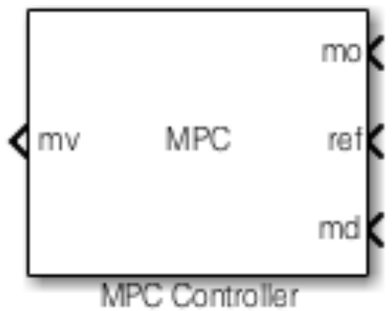
  – Linear MPC
  – Explicit MPC
  – Adaptive (LPV) MPC

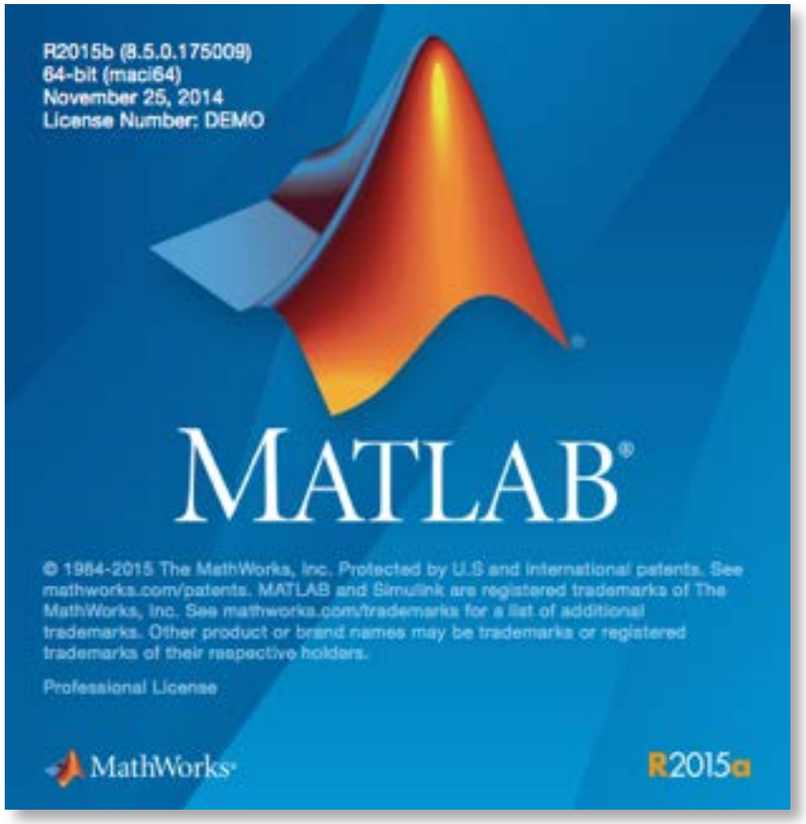
  – MPC Simulink library

  – MPC Graphical User Interface

  – Code generation [RTW, xPC Target, dSpace, etc.]

  – Linked to OPC Toolbox, System ID Toolbox, ...





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

MPC theory

- MPC theory came after the industrial success of MPC

- Areas of theoretical research:
  - Linear MPC — linear prediction model
  - Nonlinear MPC — nonlinear prediction model
  - Robust MPC — uncertain (linear) prediction model
  - Stochastic MPC — stochastic prediction model
  - Hybrid MPC — prediction model integrating logic and dynamics
  - Explicit MPC — off-line (exact/approximate) computation of MPC
  - Solvers for MPC — on-line computation of MPC

- Theoretical issues:
  - Feasibility
  - Convergence and stability
  - Solution algorithms

(Mayne, *Automatica*, 2014)

- Research on solution algorithms with guaranteed theoretical properties has the most impact in industrial practice

Feasibility

$$
\begin{aligned}
\min_{z} \quad & \sum_{k=0}^{N-1} \|W^y(y_{k+1} - r(t))\|^2 + \|W^{\Delta u}\Delta u_k\|^2 \\
\text{subj. to} \quad & u_{\min} \le u_k \le u_{\max}, \;\; k = 0, \ldots, N-1 \\
& y_{\min} \le y_k \le y_{\max}, \;\; k = 1, \ldots, N \\
& \Delta u_{\min} \le \Delta u_k \le \Delta u_{\max}, \;\; k = 0, \ldots, N-1
\end{aligned}
$$

QUADRATIC PROGRAM (QP)

- **Feasibility**: Will the QP problem be feasible at all sampling instants t ?

- **Input constraints only:** no feasibility issues !

- **Hard output constraints:**
  - When $N<\infty$ there is no guarantee that the QP problem will remain feasible at all future time steps t
  - $N=\infty$ ⟹ infinite number of constraints !
  - Maximum output admissible set theory: $N<\infty$ is enough !

(Gilbert & Tan, 1991) (Kerrigan & Maciejowski, 2000) (Chmielewski & Manousiouthakis, 1996)

Soft constraints

- To prevent QP infeasibility, relax output constraints:

$$\begin{aligned}
\min_{z} \quad & \sum_{k=0}^{N-1} \|W^y(y_{k+1} - r(t))\|^2 + \|W^{\Delta u}\Delta u_k\|^2 + \rho_\epsilon \epsilon^2 \\
\text{subj. to} \quad & u_{\min} \le u_k \le u_{\max}, \;\; k = 0, \ldots, N-1 \\
& \Delta u_{\min} \le \Delta u_k \le \Delta u_{\max}, \;\; k = 0, \ldots, N-1 \\
& y_{\min} - \epsilon V_{\min} \le y_k \le y_{\max} + \epsilon V_{\max}, \;\; k = 1, \ldots, N
\end{aligned}$$

ϵ = "panic" variable

$$z = [\Delta u'(0) \;\; \Delta u'(1) \;\; \ldots \;\; \Delta u'(N-1) \;\; \epsilon]'$$

$$\rho_\epsilon \gg W^y, W^{\Delta u}$$

$V_{\min}$, $V_{\max}$ = vectors with entries ≥ 0 (the larger the i-th entry of vector V, the relatively softer the corresponding i-th constraint)

- Infeasibility can be due to:
  - modeling errors
  - disturbances
  - wrong MPC setup (e.g., prediction horizon is too short)

Basic convergence properties

Theorem. Consider the linear system $\begin{cases} x(t+1) = Ax(t)+Bu(t) \\ y(t) = Cx(t) \end{cases}$

and the MPC control law based on optimizing

$$V^*(x(t)) = \min \sum_{k=0}^{N-1} x_k' Q x_k + u_k' R u_k$$

$$\text{s.t.} \quad x_{k+1} = Ax_k + Bu_k$$

$$u_{\min} \le u_k \le u_{\max}$$

$$y_{\min} \le Cx_k \le y_{\max}$$

$$x_N = 0 \quad \text{“terminal constraint”}$$

with $R,Q>0$. If the optimization problem **is feasible at time** $t=0$ then

$$\lim_{t\to\infty} x(t) = 0$$

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

and the constraints are satisfied at all time $t\ge 0$, for all $R,Q>0$

(Keerthi and Gilbert, 1988) (Bemporad, Chisci, Mosca, 1994)

For general **stability result** see (Lazar, Heemels, Weiland, Bemporad, IEEE TAC, 2006)

Convergence proof

Main idea: Use **value function** $V^*(x(t))$ as a **Lyapunov** function

- Let z_t = optimal control sequence at time t, $z_t = [u_0^t \ \dots \ u_{N-1}^t]'$
- By construction $\bar{z}_{t+1} = [u_1^t \ \dots \ u_{N-1}^t \ 0]'$ is a feasible sequence at time $t+1$
- The cost of $\bar{z}_{t+1}$ is $V^*(x(t)) - x'(t)Qx(t) - u'(t)Ru(t) \geq V^*(x(t+1))$
- $V^*(x(t))$ is monotonically decreasing and ≥ 0, so $\exists \lim_{t\to\infty} V^*(x(t)) \triangleq V_\infty$
- Hence $0 \leq x'(t)Qx(t) + u'(t)Ru(t) \leq V^*(x(t)) - V^*(x(t+1)) \to 0$ for $t \to \infty$
- Since $R, Q > 0$, $\lim_{t\to\infty} x(t) = 0$, $\lim_{t\to\infty} u(t) = 0$ □

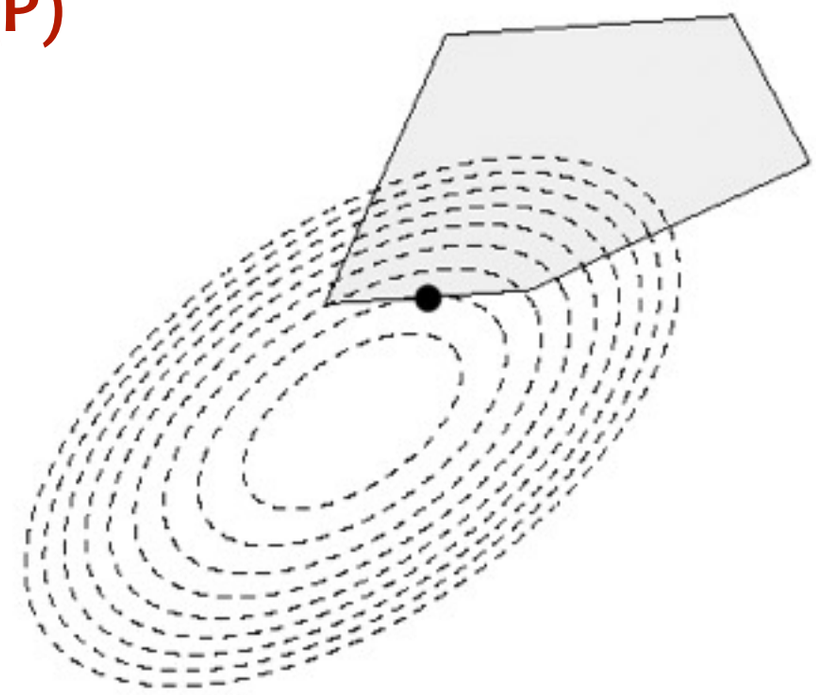
Global optimum is not needed to prove convergence !

Embedded QP solvers for Cyber-Physical Systems

Quadratic programming

$$\begin{array}{ll}\min & \frac{1}{2}x'Px + f'x \\ \text{s.t.} & Ax \leq b, \;\; x \in \mathbb{R}^n\end{array}$$

quadratic program (QP)

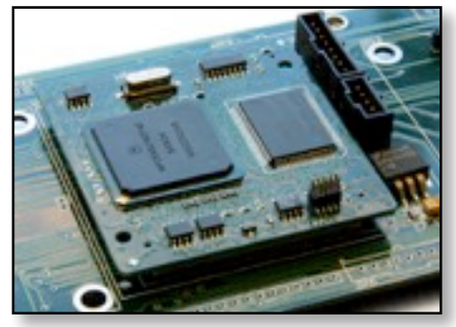


- Convex optimization if $P \geq 0$ ($P =$ positive semidefinite matrix)
- Hard problem if $P \not\geq 0$ ($P =$ indefinite matrix)

Requirements for embedded optimization code

1. **Speed (throughput)**: fast enough to provide the control decision within short sampling intervals (e.g., 10-100 ms)

2. Require **simple/cheap hardware** (microcontroller, microprocessor, FPGA) and **little memory** to store problem **data *and* code**



3. **Worst-case execution time** must be (tightly) estimated for embedding the controller in a **real-time platform**

4. **Code simple** enough to be **verifiable/certifiable** (or at least understandable by production engineers)

```
54    # Main iteration loop
55    while keepgoing & (nu<maxiter):
56
57        # Compute theta, beta
58        beta=th*(1/th0-1)
59
60        w=y+beta*(y-y0)
61        zhat=-(dot(MG,w)+gP)
62        if -onlyzhat:
63            z=(1-th)*z+th*zhat
64        th0=th # Update th0
65        th2=th**2
66        th=(sqrt(th2**2+4.0*th2)-th2)/2.0
```

Reqs for embedded optimization algorithms

- **Fast** to execute

- Require **little memory**

- **Numerically robust**: must perform well in single precision or even in fixed-point computations

- **Limited linear algebra operations** (in on-line computations) for **code simplicity** (example: at most matrix-vector products)

Reqs for embedded optimization algorithms

- **Dense problems**, no need for sparse linear algebra

  Typical problem size: 10÷50 variables, 10÷100 constraints

- **Suboptimal solutions are usually ok.** Optimality is of less concern than feasibility (the prediction model is approximate, the cost function usually comes from trial & error tuning)

- Need to be able to estimate the **worst-case number of iterations**

Gradient projection method for QP (very simple!)

(Goldstein, 1964) (Levitin & Poljak, 1965)

- QP problem:

$$\begin{aligned} \min_z \quad & \frac{1}{2} z' H z + x' F z \\ \text{s.t.} \quad & Gz \le W + Sx \end{aligned}$$

$z =$ optimization vector

$x =$ parameter vector

- Apply **gradient-projection to dual QP**

$$\min_{y \ge 0} \quad \frac{1}{2} y' M y + (Dx + W)' y$$

L = max eigenvalue of M, or $L = \sqrt{\sum_{i,j=1}^{m} |M_{i,j}|^2}$ (Frobenius norm)

- Iterate $y_{k+1} = \max\{(I - \frac{1}{L}M)y_k - \frac{1}{L}(Dx + W), 0\}$ from $y_0 = 0$

until convergence to optimal y^* (guaranteed if QP is feasible)

- Set $z^* = -H^{-1}G'y^* - H^{-1}F'x$

- Convergence rate: $f(y_k) - f^* \le \frac{L}{2k}\|y_0 - y^*\|$

Fast gradient projection for (dual) QP

(Nesterov, 1983)
(Patrinos, Bemporad, IEEE TAC, 2014)

- Apply **fast gradient method** to dual QP:

$$\begin{aligned} w_k &= y_k + \beta_k(y_k - y_{k-1}) \\ z_k &= -Kw_k - Jx \\ s_k &= \tfrac{1}{L}Gz_k - \tfrac{1}{L}(Sx + W) \\ y_{k+1} &= \max\{y_k + s_k, 0\} \end{aligned}$$

$$\begin{aligned} K &= H^{-1}G' \\ J &= H^{-1}F' \end{aligned} \qquad y_{-1} = y_0 = 0$$

$$\beta_k = \begin{cases} 0 & k = 0 \\ \frac{k-1}{k+2} & k > 0 \end{cases}$$

- Termination criterion #1: **primal feasibility**

feasibility tol

$$s_k^i \le \frac{1}{L}\epsilon_G, \ \ \forall i = 1, \dots, m$$

- Termination criterion #2: **primal optimality**

$$f(z_k) - f^* \le f(z_k) - \phi(w_k) = -w_k' s_k L \le \epsilon_V$$

dual function

optimality tol

$$-w_k' s_k \le \frac{1}{L}\epsilon_V$$

Fast gradient projection for (dual) QP

(Patrinos, Bemporad, IEEE TAC, 2014)

- Main on-line operations involve only **simple linear algebra**

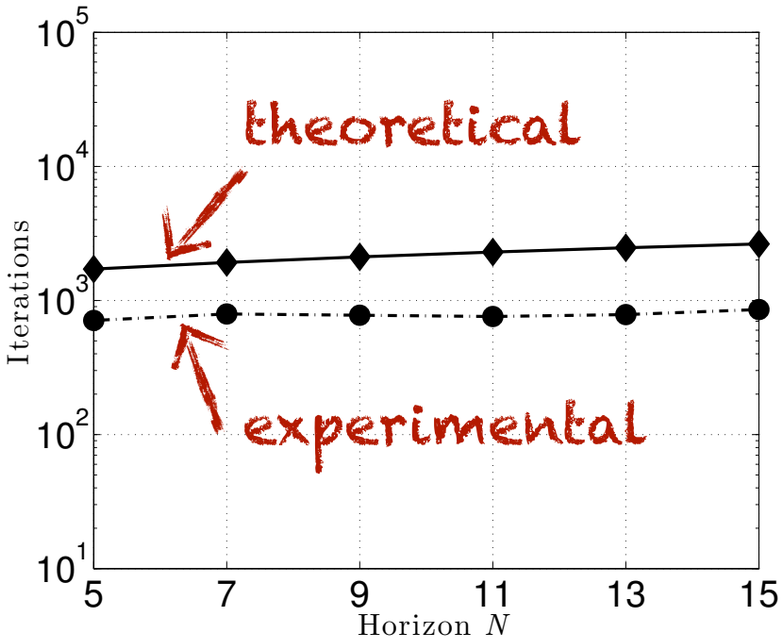
```
while keepgoing && (i<maxiter),

    beta=(i-1)/(i-2).*(i>0);

    w=y+beta*(y-y0);
    z=-(iMG*w+iMc);
    s=GLz-bL;

    y0=y;

    % Check termination conditions
    if all(s<=epsGL),
        gapL=-w'*s;
        if gapL<=epsVL,
            return
        end
    end

    y=max(w+s,0);

    i=i+1;
end
```

- For MPC problems, using Riccati-like iterations complexity per iteration is $O(N)$
  $[N = \text{prediction horizon}]$

- Convergence rate:

$$f(y_k) - f^* \leq \frac{2L}{(k+1)^2}\|y_0 - y^*\|^2$$

- Tight bounds on maximum number of iterations

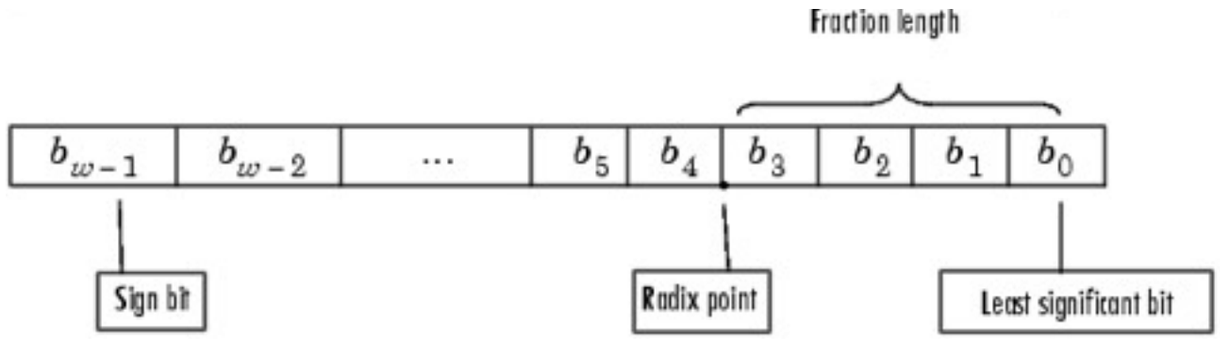


- Can be very useful to warm-start other methods !

Finite-precision arithmetics

How about numerical robustness ?

- **Fixed-point arithmetics** is very attractive for embedded control:
  - Computations are fast and cheap
  - Hardware support in all platforms



- Cons:
  - Accumulation of quantization errors
  - Limited range (numerical overflow)

Gradient projection in fixed-point arithmetics

(Patrinos, Guiggiani, Bemporad, 2013)

- Convergence to feasibility

$$\max_i g_i(z_k) \le \frac{2LD^2}{k+1} + L_v\epsilon_z^2 + 4D\epsilon_\xi$$

→ exponentially decreasing with p

- Convergence to optimality

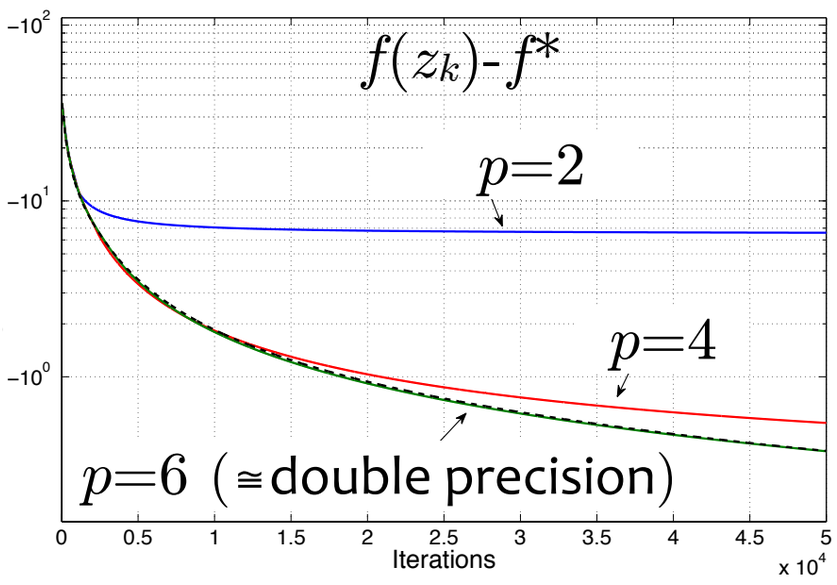
$$f(z_k) - f^\star \le \frac{L}{2(k+1)}(\|y^\star\|^2 + \|y_0\|^2) + \delta$$

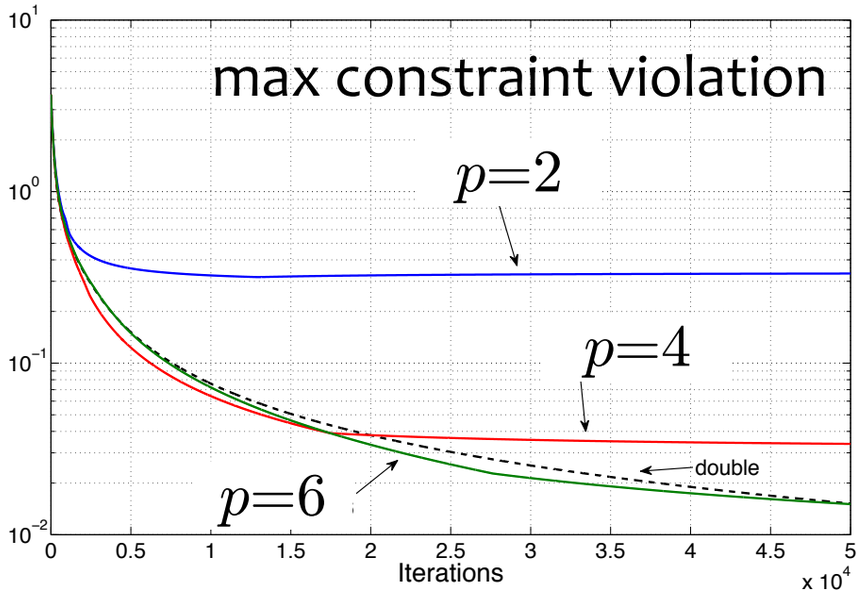
- Design guidelines (precision)

$$p \ge \log_2 \frac{m\sqrt{n}}{\sqrt{\frac{\varepsilon}{L_V} + \frac{n}{m}\left(\frac{2D}{L_V}\right)^2} - \sqrt{\frac{n}{m}}\frac{2D}{L_V}} - 1$$

fractional bits

desired solution "quality"





- Design guidelines for required #**integer bits** to avoid **overflow** also available

Hardware tests (floating vs fixed point)

(Patrinos, Guiggiani, Bemporad, 2013)

32-bit Atmel SAM3X8E **ARM Cortex-M3** processing unit:
84 MHz, 512 KB of flash memory and 100 KB of RAM.

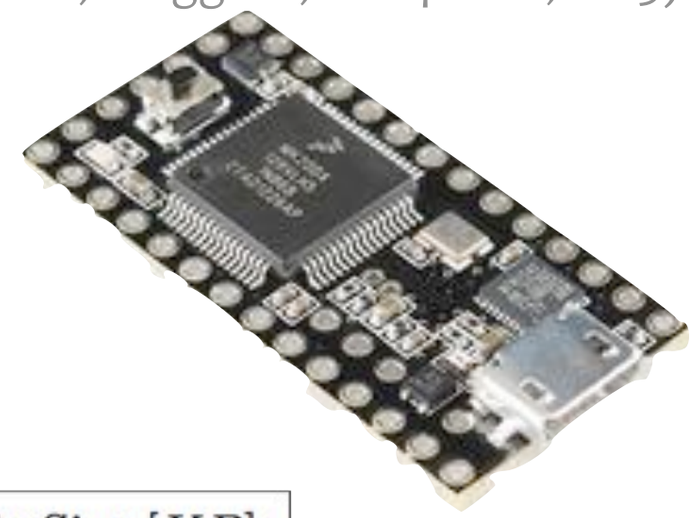


Table 1
Fixed-point hardware implementation

Size [variables/constraints]	Time $[ms]$	Time per iteration $[\mu s]$	Code Size $[KB]$
10/20	22.9	226	15
20/40	52.9	867	17
40/80	544.9	3382	27
60/120	1519.8	7561	43

fixed point

Table 2
Floating-point hardware implementation

Size [variables/constraints]	Time $[ms]$	Time per iteration $[\mu s]$	Code Size $[KB]$
10/20	88.6	974	16
20/40	220.1	3608	21
40/80	2240	13099	40
60/120	5816	30450	73

floating point

fixed-point about 4x faster than floating-point

ADMM method for QP

(Boyd et al., 2010)

- Alternating Directions Method of Multipliers (ADMM) for QP

$$\begin{array}{ll}\min & \frac{1}{2}x'Qx + q'x \\ \text{s.t.} & \ell \leq Ax \leq u\end{array}$$

- Separable form of QP:

$$\begin{array}{ll}\min & \frac{1}{2}x'Qx + q'x + g(z) \\ \text{s.t.} & Ax - z = 0\end{array}$$

$$g(z) = \begin{cases} 0 & \text{if } -\ell \leq Az \leq u \\ +\infty & \text{otherwise} \end{cases}$$

- Scaled ADMM iterations:

$$\begin{array}{rcl} x^{k+1} & = & -(Q + \gamma A^T A)^{-1}(\gamma A^T(y^k - z^k) + q) \\ z^{k+1} & = & \min\{\max\{Ax^{k+1} + y^k, \ell\}, u\} \\ y^{k+1} & = & y^k + Ax^{k+1} - z^{k+1} \end{array}$$

Scaling (or preconditioning)

- **Preconditioning** can improve convergence rate of iterative algorithms (in particular first-order methods are very sensitive to scaling)

primal QP $\min_z \ \frac{1}{2} z'Hz + f'z \quad \text{s.t.} \quad Gz \le W$

$$M = GH^{-1}G'$$
$$d = GH^{-1}f + W$$

- **Dual scaling:**

$$\min_y \ \frac{1}{2} y'My + d'y \quad \text{s.t.} \quad y \ge 0$$

scaling $y = Py_s$

$$P = \text{diag}\left(\frac{1}{\sqrt{M_{ii}}}\right)$$

$$\min_{y_s} \ \frac{1}{2} y_s'(PMP)y_s + d'Py_s \quad \text{s.t.} \quad y_s \ge 0$$

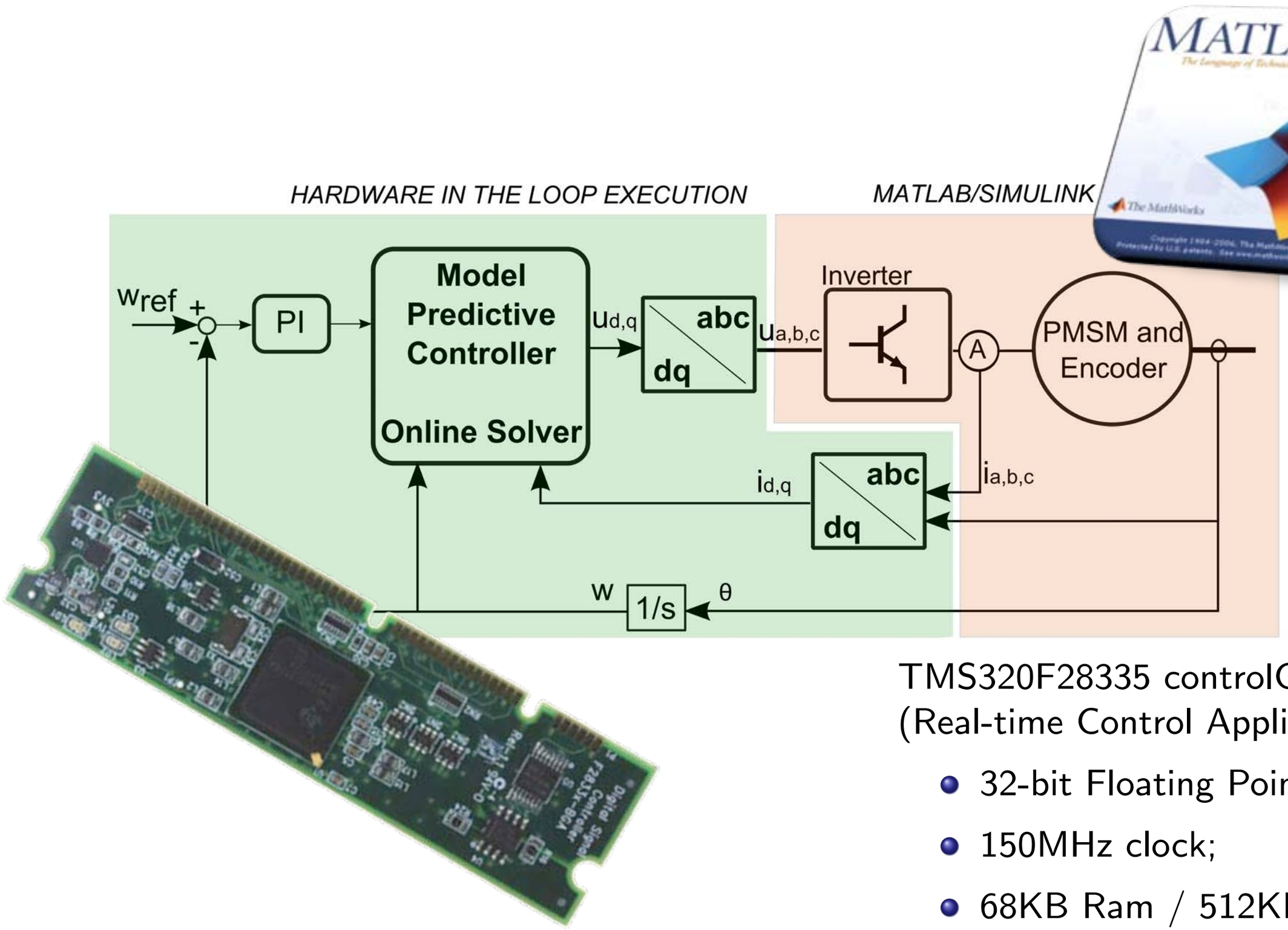
scaled dual QP

- This is equivalent to scale constraints in primal problem: $\frac{1}{\sqrt{M_{ii}}} G_i z \le \frac{1}{\sqrt{M_{ii}}} W_i$

- Primal solution: $z^* = -H^{-1}(G'Py_s^* + f)$

Hardware-In-the-Loop experiments (DSP)

- Control of a brushless DC motor



TMS320F28335 controlCARD
(Real-time Control Applications)

- 32-bit Floating Point (IEEE-754);
- 150MHz clock;
- 68KB Ram / 512KB Flash.

Hardware-In-the-Loop experiments (DSP)

- Linear MPC setup:

$$\begin{cases} x(k+1) &= Ax(k) + B_u u(k) + B_v(k) \\ y(k) &= Cx(k) \end{cases}$$

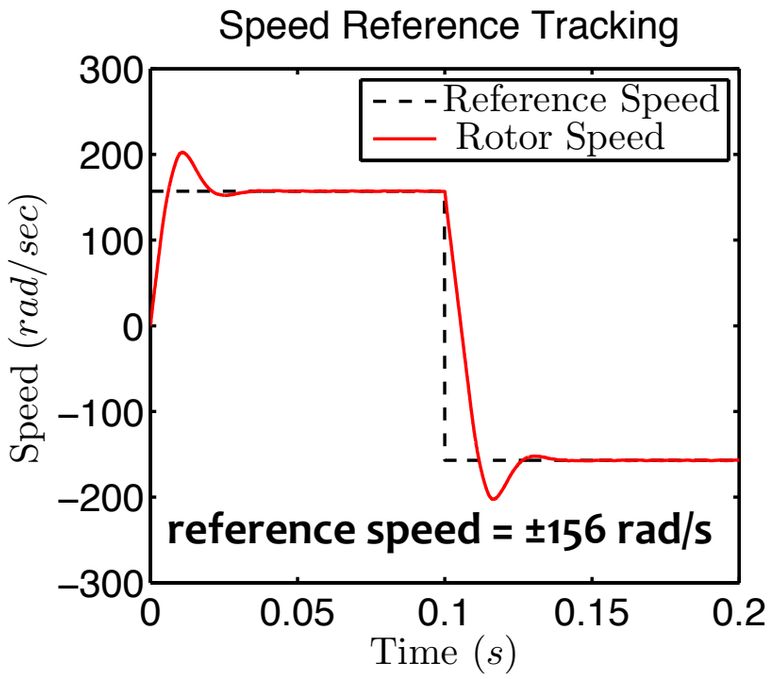
$$x = \begin{bmatrix} i_d \\ i_q \end{bmatrix} \qquad u = \begin{bmatrix} u_d \\ u_q \end{bmatrix}$$

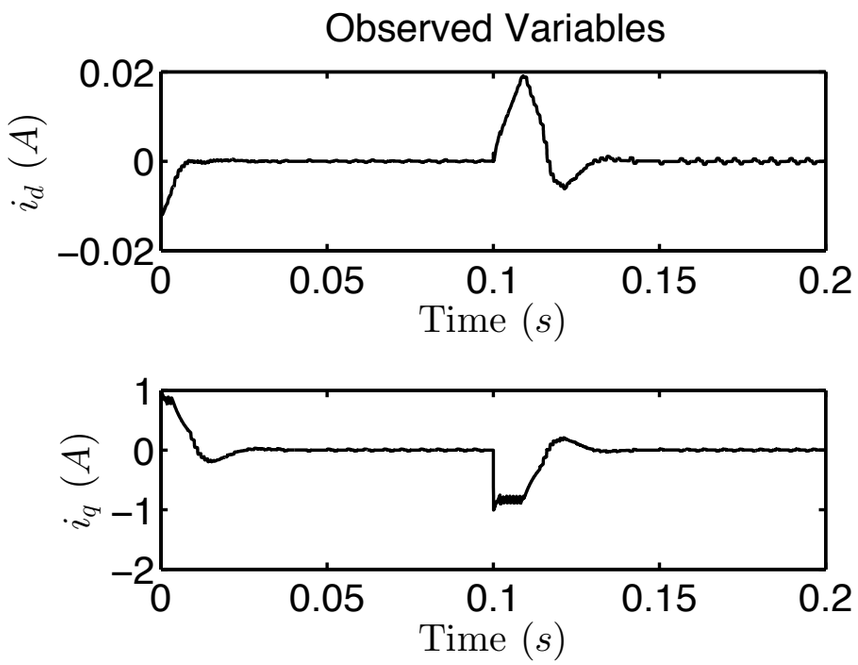
$$A = \begin{bmatrix} -\frac{R}{L} & \bar{\omega} \\ 0 & -\frac{R}{L} \end{bmatrix},\ B_u = \begin{bmatrix} \frac{1}{L} & 0 \\ 0 & \frac{1}{L} \end{bmatrix},\ B_v = \begin{bmatrix} 0 \\ -\omega^2 \frac{\lambda}{L} \end{bmatrix},\ C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

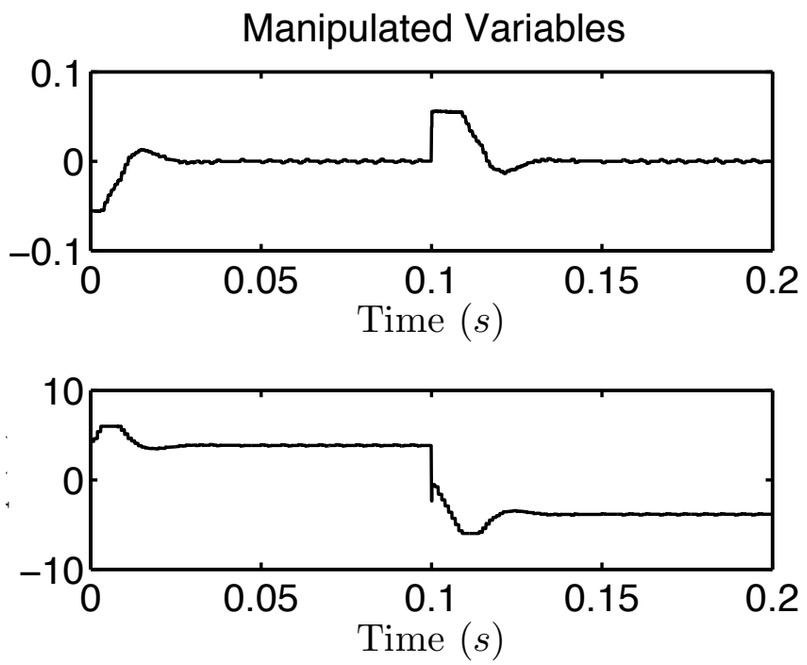
R = stator resistance [Ω]
L = stator inductance [H]
ω = rotor speed [rad/s]
λ = motor flux leakage [Wb]

prediction horizon = 4
control horizon = 2
box constraints on i_q and u_q

$$-\epsilon I_{\max} \le i_d \le \epsilon I_{\max}$$
$$V_{\max} \le u_q \le V_{\max}$$



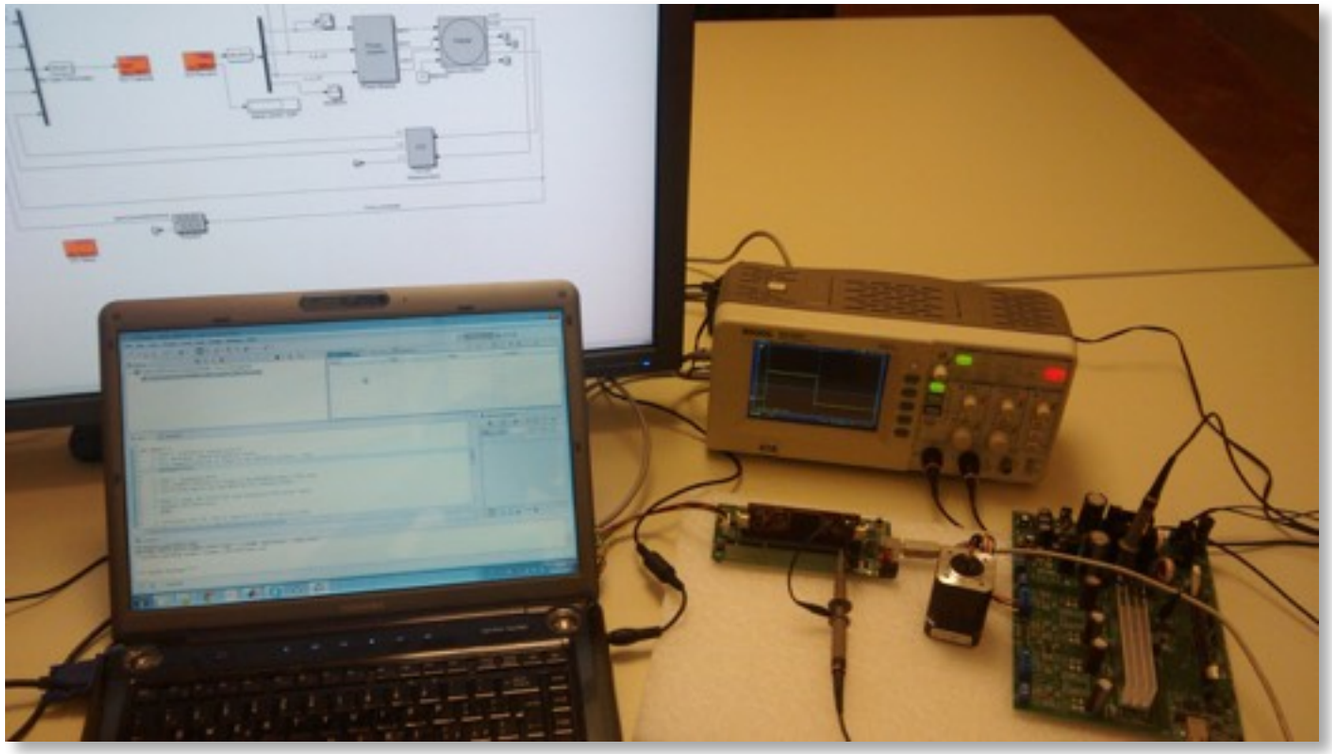




Hardware-In-the-Loop experiments (DSP)

- Experimental setup:

  - PC with MATLAB/Simulink
  - RS232 adapter
  - TMS320F28335 Experimenter Kit



var × constr.	GPAD	AS	ADMM	FBN
4 × 16	332 μs (18)	120 μs (3)	1.42 ms (62)	208 μs (2)
8 × 24	1.1 ms (22)	446 μs (5)	4 ms (77)	396 μs (2)
12 × 32	2.59 ms (27)	1.19 ms (7)	8.25 ms (82)	652 μs (2)

- Active set dominates on small problems (Goldfarb, Idnani, 1983)
- Forward-Backwards Newton (proximal method) better for larger problems

Solving QP's via nonnegative least squares

(Bemporad, IEEE TAC, accepted)

- Use NNLS to also solve strictly convex QP's !

$$\begin{array}{ll}\min\limits_{z} & \frac{1}{2}z'Qz + c'z \\ \text{s.t.} & Gz \le g\end{array}$$

QP

complete the squares

$$u \triangleq Lz + L^{-T}c$$
$$Q = L'L$$

$$\begin{array}{ll}\min\limits_{u} & \frac{1}{2}\|u\|^2 \\ \text{s.t.} & Mu \le d\end{array}$$

$$M = GL^{-1}$$

Least Distance Problem

residual $= 0$?

yes

QP problem infeasible

no

$$\begin{array}{ll}\min\limits_{y} & \frac{1}{2}\left\|\begin{bmatrix} M' \\ d' \end{bmatrix} y + \begin{bmatrix} 0 \\ 1 \end{bmatrix}\right\|_2^2 \\ \text{s.t.} & y \ge 0\end{array}$$

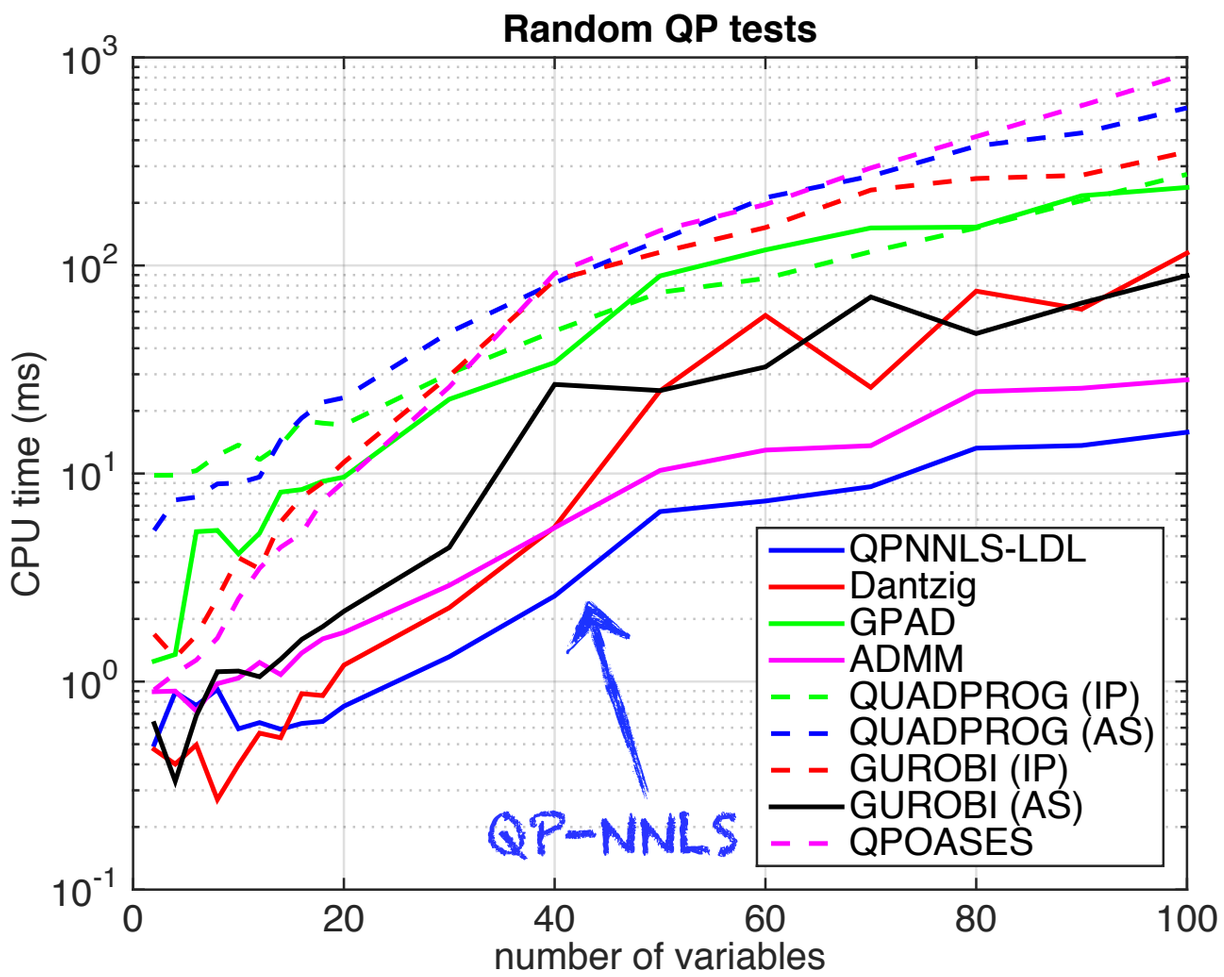
Nonnegative Least Squares

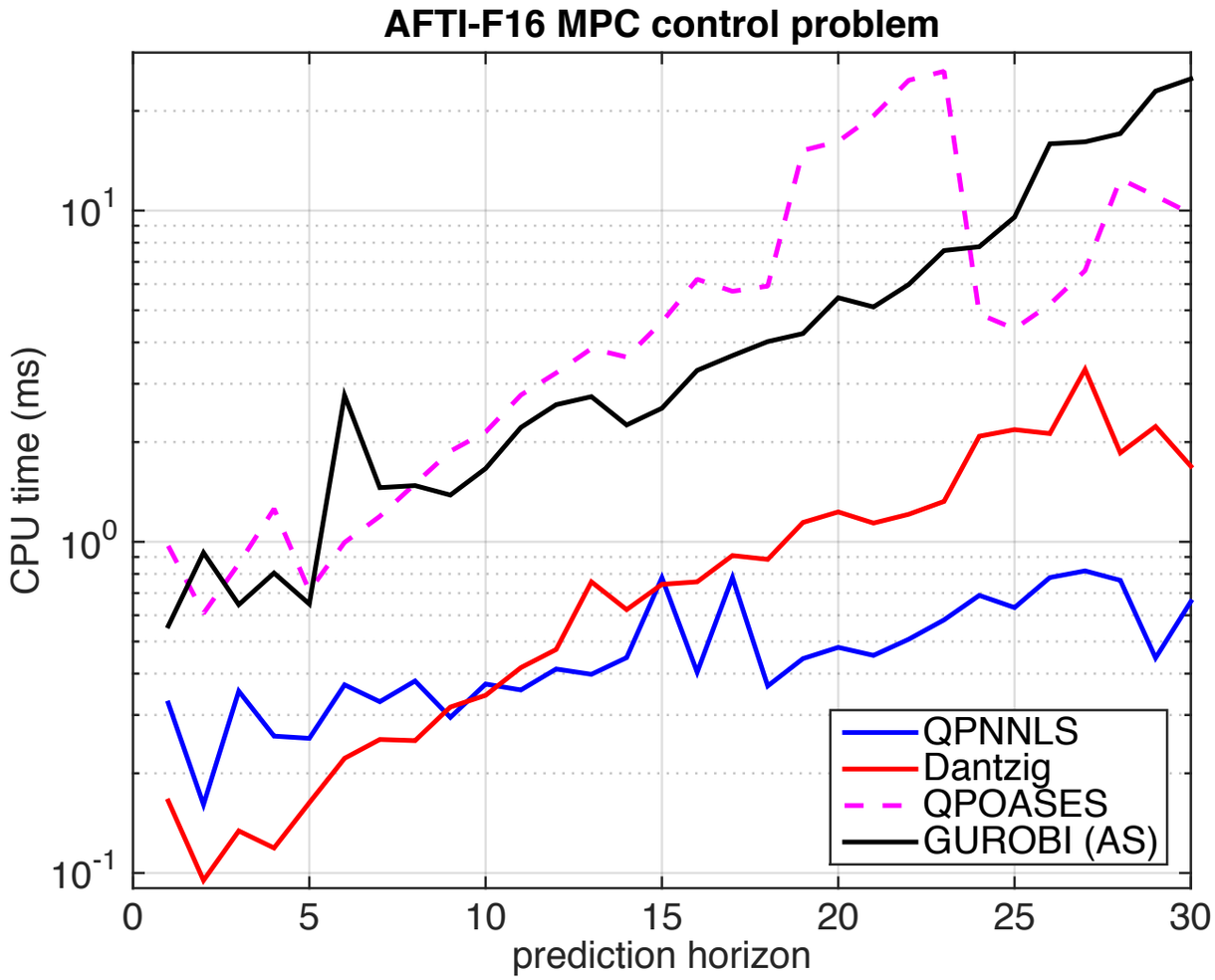
$$z^* = -\frac{1}{1 + d'y^*}L^{-1}M'y^* - Q^{-1}c$$

retrieve solution

Numerical results

(Bemporad, IEEE TAC, accepted)





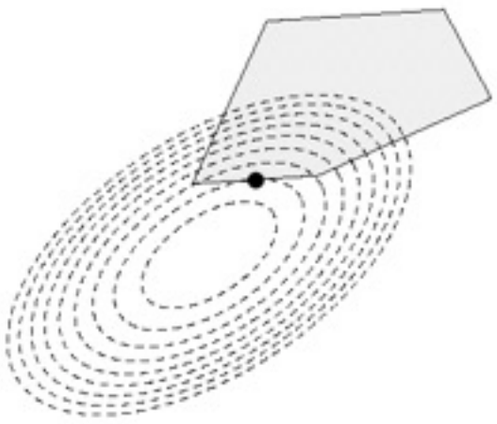
- Rather fast and simple-to-code QP solver
- Good numerical robustness in double precision
- Extended to Mixed-Integer QP (Bemporad, submitted NMPC'15)

Explicit model predictive control

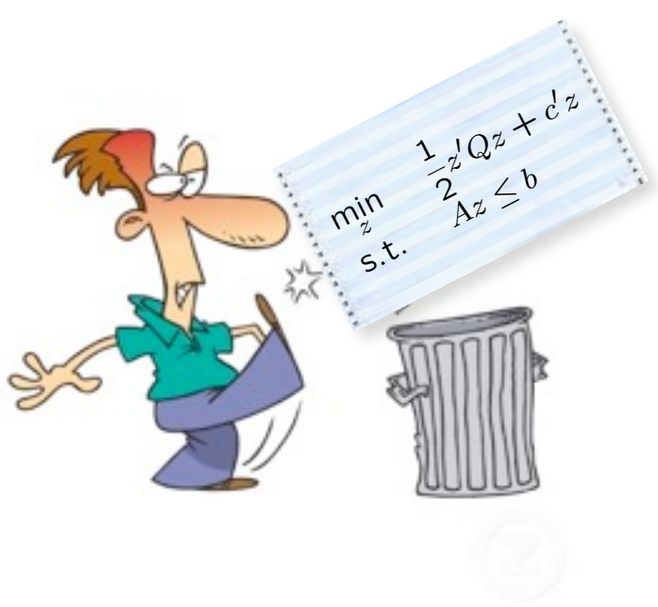
- Linear MPC requires solving a **Quadratic Program (QP)**

$$\begin{aligned} \min_z \quad & \frac{1}{2} z' H z + x'(t) F' z + \frac{1}{2} x'(t) Y x(t) \\ \text{s.t.} \quad & Gz \le W + S x(t) \end{aligned}$$

$$z = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}$$



The control law $u(t) = u_0^*(x(t))$ is **implicitly** defined by the QP solver



Can we implement MPC without solving a QP ?

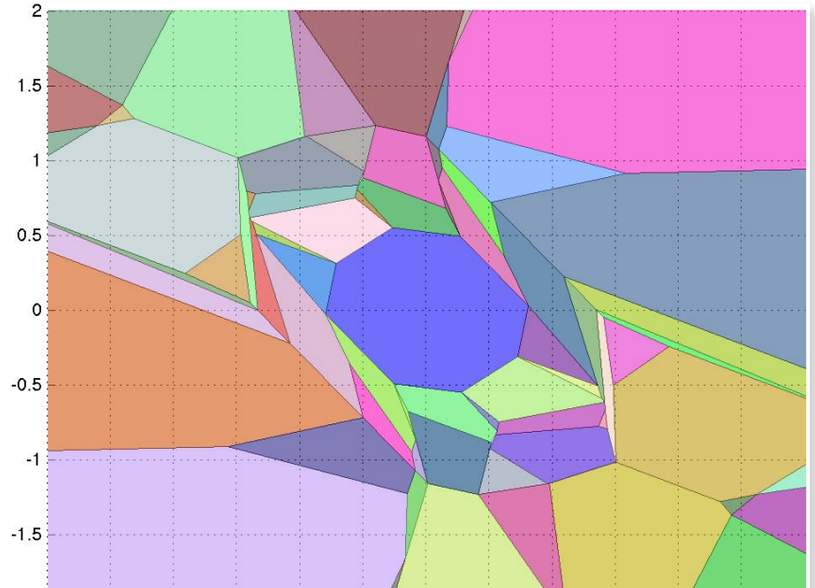
- **Explicit MPC**: pre-solve the QP **off-line for all** $x(t)$ to find the control law $u_0^*(x)$ **explicitly** via **multiparametric quadratic programming** (mpQP)

Properties of multiparametric QP

(Bemporad, Morari, Dua, Pistikopoulos, 2002)

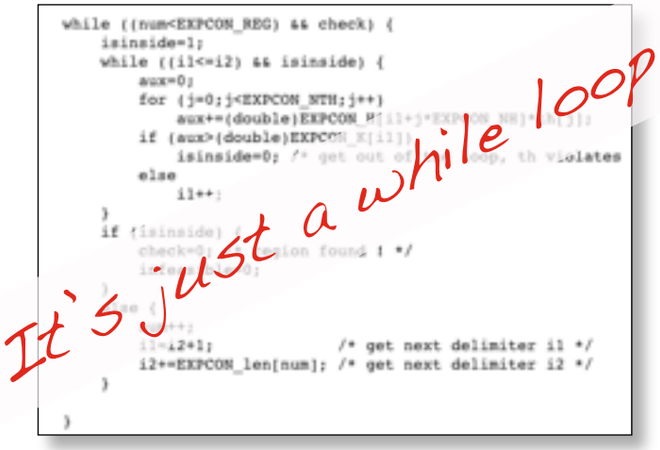
The multiparametric solution of a strictly convex QP is **continuous** and **piecewise affine**

$$z^*(x) = \arg\min_z \quad \tfrac{1}{2} z' H z + x' F' z \quad \text{s.t.} \quad Gz \le W + Sx$$



Corollary: The linear MPC control law is continuous & piecewise affine !

$$z^* = \begin{bmatrix} u_0^* \\ u_1 \\ \vdots \\ u_{N-1}^* \end{bmatrix} \qquad u(x) = \begin{cases} F_1 x + g_1 & \text{if} \quad H_1 x \le K_1 \\ \vdots & \vdots \\ F_M x + g_M & \text{if} \quad H_M x \le K_M \end{cases}$$



Multiparametric QP solvers

- A variety of mpQP solvers is available

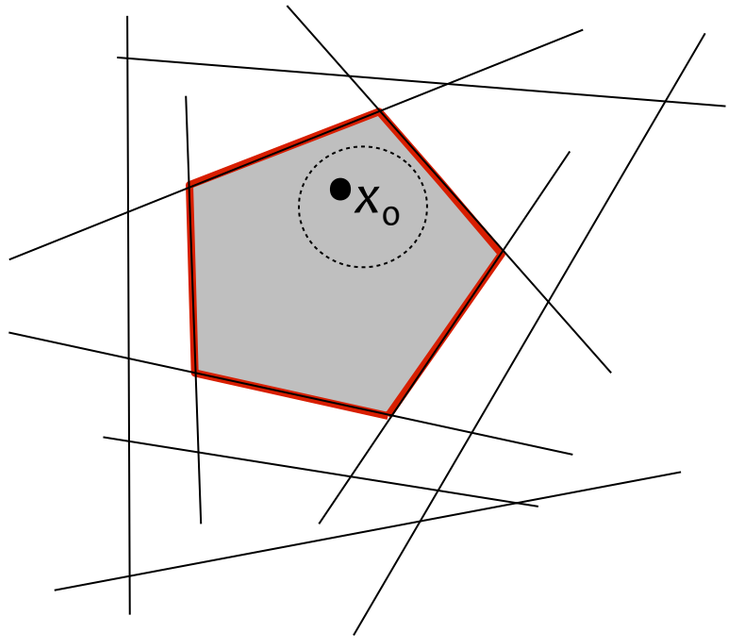
(Bemporad *et al.*, 2002) (Baotic, 2002)
(Tøndel, Johansen, Bemporad, 2003)
(Spjøtvold *et al.*, 2006)(Patrinos, Sarimveis, 2010)

- Most computations are spent in **operations on polyhedra** (=critical regions)

$$\begin{aligned} \hat{G}z^*(x) &\leq \hat{W} + \hat{S}x \\ \tilde{\lambda}^*(x) &\geq 0 \end{aligned}$$

← feasibility of primal solution
← feasibility of dual solution

- checking **emptiness of polyhedra**
- removal of **redundant inequalities**
- checking **full-dimensionality** of polyhedra



- All such operations are usually done via **linear programming (LP)**

A new mpQP algorithm based on Least Squares

The **Least Squares (LS)** problem is probably the most studied problem in numerical linear algebra

(Legendre, 1805)

(Gauss, <= 1809)

In MATLAB: >> `A\b` $= \arg\min \|Ax - b\|_2^2$
(1 character !)

- **Nonnegative Least Squares (NNLS):**

$$\begin{array}{ll}\min_v & \|Av - b\|_2^2 \\ \text{s.t.} & v \geq 0\end{array}$$

- **Partially Nonnegative Least Squares (PNNLS):**

$$\begin{array}{ll}\min_{v,u} & \|Av + Bu - c\|_2^2 \\ \text{s.t.} & v \geq 0,\ u \text{ free}\end{array}$$

PNNLS = pseudo-inverse of B
+ solve NNLS

Active-set method for Nonnegative Least Squares

(Lawson, Hanson, 1974)

$$\begin{aligned} \min_v \quad & \|Av-b\|_2^2 \\ \text{s.t.} \quad & v \geq 0 \end{aligned}$$

1) $\mathcal{P} \leftarrow \emptyset$, $v \leftarrow 0$;
2) $w \leftarrow A'(Av-b)$;
3) **if** $w \geq 0$ **or** $\mathcal{P} = \{1,\dots,m\}$ **then go to** Step 11;
4) $i \leftarrow \arg\min_{i \in \{1,\dots,m\}\setminus\mathcal{P}} w_i$, $\mathcal{P} \leftarrow \mathcal{P} \cup \{i\}$;
5) $y_{\mathcal{P}} \leftarrow \arg\min_{z_{\mathcal{P}}} \|((A')_{\mathcal{P}})' z_{\mathcal{P}} - b\|_2^2$, $y_{\{1,\dots,m\}\setminus\mathcal{P}} \leftarrow 0$;
6) **if** $y_{\mathcal{P}} \geq 0$ **then** $v \leftarrow y$ and **go to** Step 2;
7) $j \leftarrow \arg\min_{h \in \mathcal{P}:\ y_h \leq 0} \left\{ \frac{v_h}{v_h - y_h} \right\}$;
8) $v \leftarrow v + \frac{v_j}{v_j - y_j}(y - v)$;
9) $\mathcal{I} \leftarrow \{h \in \mathcal{P} : v_h = 0\}$, $\mathcal{P} \leftarrow \mathcal{P} \setminus \mathcal{I}$;
10) **go to** Step 5;
11) $v^* \leftarrow v$; **end**.

- NNLS algorithm is very simple (750 chars in Embedded MATLAB), the key operation is to solve a standard LS problem at each iteration (via a QR, LDL', or Cholesky factorization)

A new mpQP algorithm based on NNLS

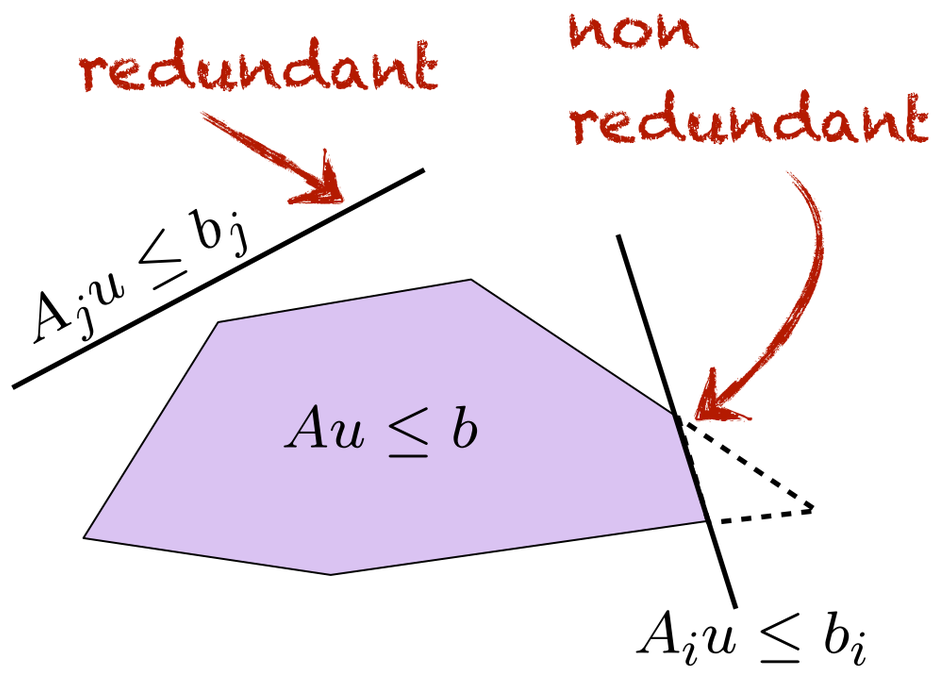
(Bemporad, IEEE TAC 2015)

- Key result:

A polyhedron $P = \{u \in \mathbb{R}^n : \ Au \le b\}$
is nonempty iff the PNNLS problem

$$(v^*, u^*) = \arg\min_{v,u} \ \|v + Au - b\|_2^2 \quad \text{s.t.} \ \ v \ge 0, u \text{ free}$$

has zero residual $\|v^* + Au^* - b\|_2^2 = 0$



- Numerical results on elimination of redundant inequalities:

m	NNLS	LP
2	0.0006	0.0046
4	0.0019	0.0103
6	0.0038	0.0193
8	0.0071	0.0340
10	0.0111	0.0554
12	0.0178	0.0955
14	0.0263	0.1426
16	0.0357	0.1959

random polyhedra of $\mathbb{R}^m$ with $10m$ inequalities

NNLS = compiled Embedded MATLAB
LP = compiled C code (GLPK)

CPU time = seconds (this Mac)

- Many other polyhedral operations can be also tackled by NNLS

A new mpQP algorithm based on NNLS

(Bemporad, IEEE TAC, 2015)

- New mpQP algorithm based on **Nonnegative Least Squares (NNLS)** + **dual QP formulation** to compute active sets and deal with degeneracy

- Comparison with:
  - Hybrid Toolbox (Bemporad, 2003)
  - Multiparametric Toolbox 2.6 (with default opts) (Kvasnica, Grieder, Baotic, 2006)

- Included in MPC Toolbox 5.0 (R2014b)

The MathWorks (Bemporad, Morari, Ricker, 1998-2015)

q	m	Hybrid Tbx	MPT	NNLS
4	2	0.0174	0.0256	0.0026
4	3	0.0263	0.0356	0.0038
4	4	0.0432	0.0559	0.0061
4	5	0.0650	0.0850	0.0097
4	6	0.0827	0.1105	0.0126
8	2	0.0347	0.0396	0.0050
8	3	0.0583	0.0680	0.0092
8	4	0.0916	0.0999	0.0140
8	5	0.1869	0.2147	0.0322
8	6	0.3177	0.3611	0.0586
12	2	0.0398	0.0387	0.0054
12	3	0.1121	0.1158	0.0191
12	4	0.2067	0.2001	0.0352
12	5	0.6180	0.6428	0.1151
12	6	1.2453	1.3601	0.2426
20	2	0.1029	0.0763	0.0152
20	3	0.3698	0.2905	0.0588
20	4	0.9069	0.7100	0.1617
20	5	2.2978	1.9761	0.4395
20	6	6.1220	6.2518	1.2853

Complexity of explicit MPC

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

(Intel Centrino 1.4 GHz)

Average time on 100 random 3D parameters (2N constraints)

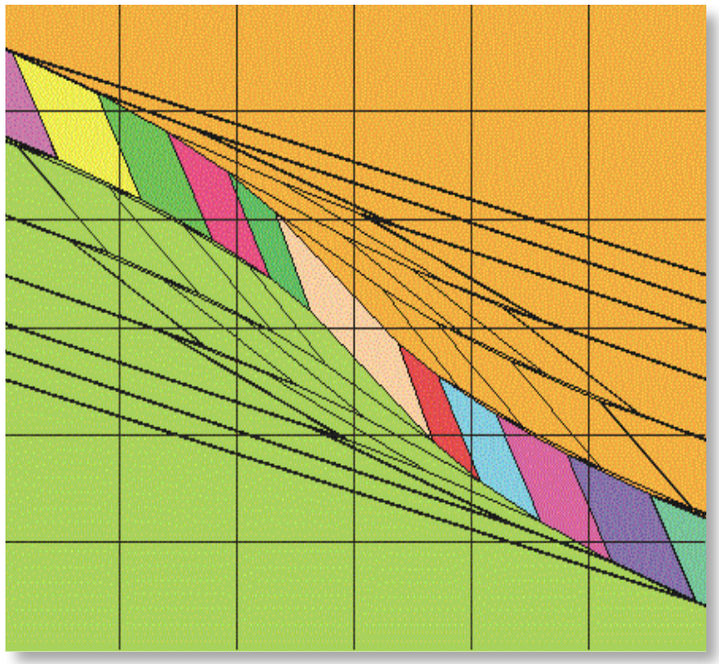
Worst-case time on 100 random 3D parameters (2N constraints)

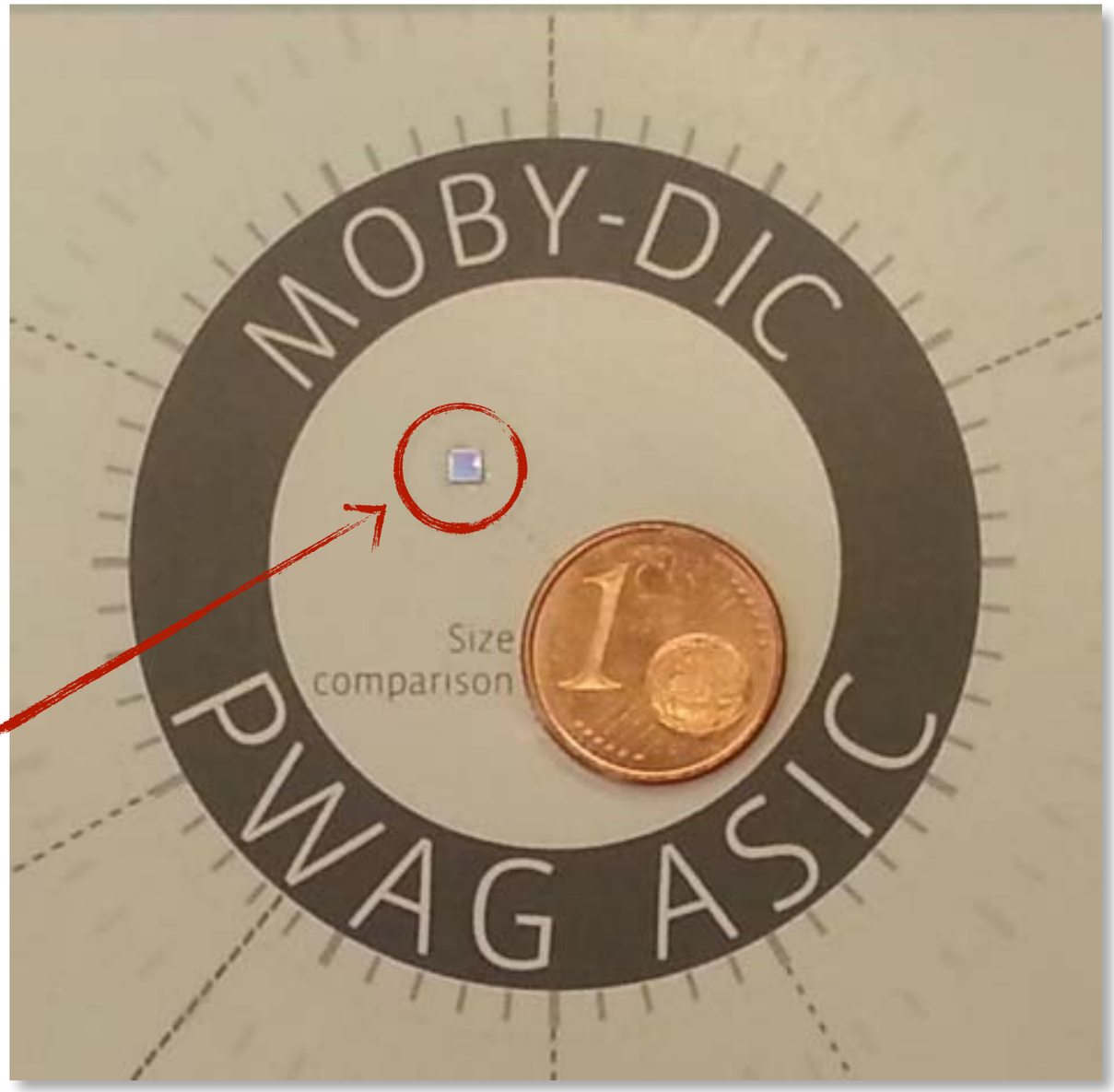
Explicit MPC typically practical for max 6-8 free control moves, 10-20 constraints, and 8-12 parameters (states+references)

⟹ Regions can be visited more efficiently than linear search

⟹ Number of regions can be usually reduced (e.g. suboptimal solutions)

Hardware (ASIC) implementation of explicit MPC





Technology: 90 nm, 9 metal-layer from Taiwan Semiconductor Manufacturing Company

Chip size: 1860x1860 μm^2

Memory sizes: 24 KB (tree memory) ; 30 KB (parameter memory)

Power supply: 2.5V (ring); 1.2V (core)

Maximum frequency: 107.5 MHz (with two inputs)

Power consumption: 38.08 mW@107.5MHz;

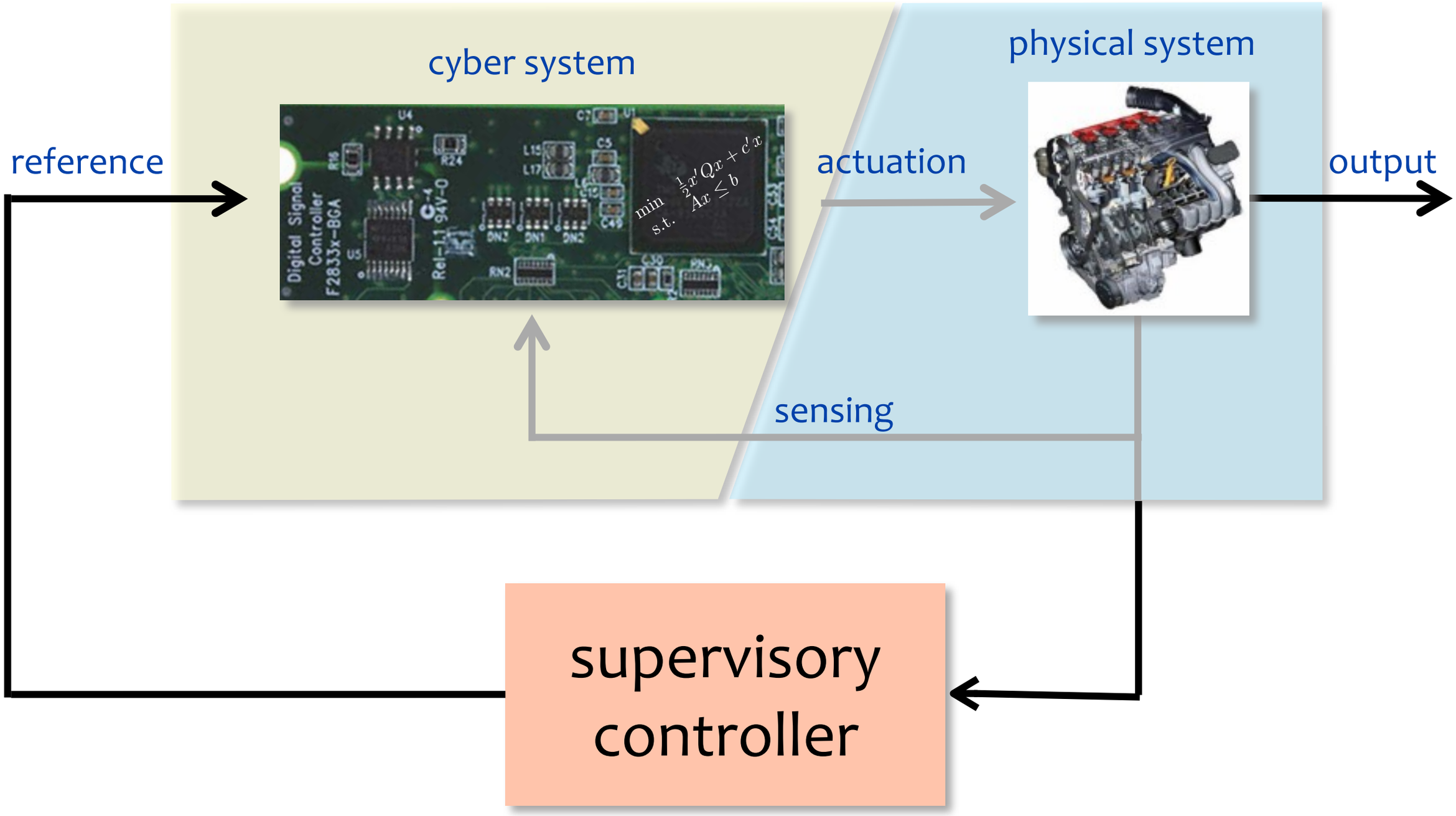
$$\frac{1}{107.5\ \text{MHz}} = 9.3\ \text{ns}$$

SEVENTH FRAMEWORK PROGRAMME

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

Hybrid MPC of Cyber-Physical Systems

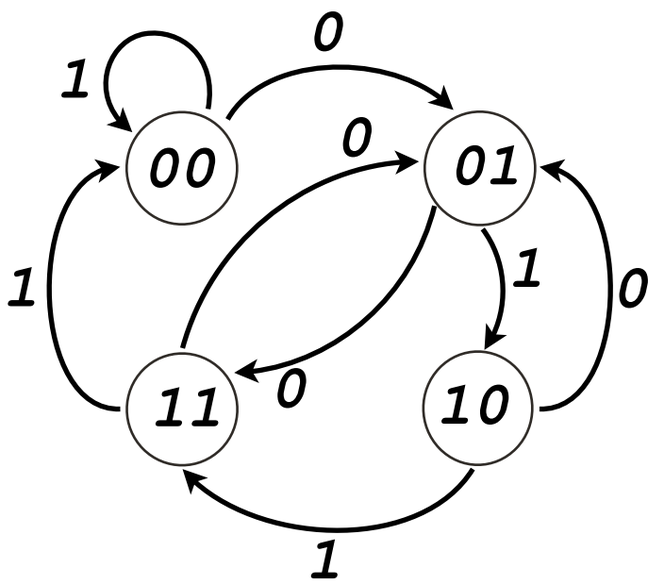
Control of Cyber-Physical Systems



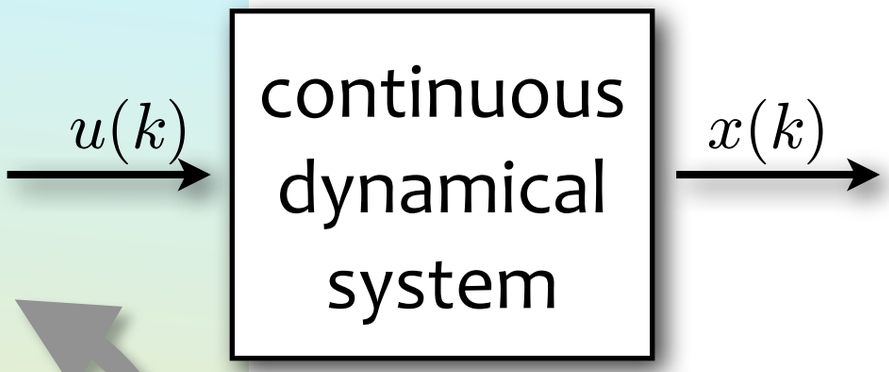
What is a good model of a CPS for supervisory control purposes ?

Hybrid dynamical systems

cyber system



physical system



hybrid dynamical system

- 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

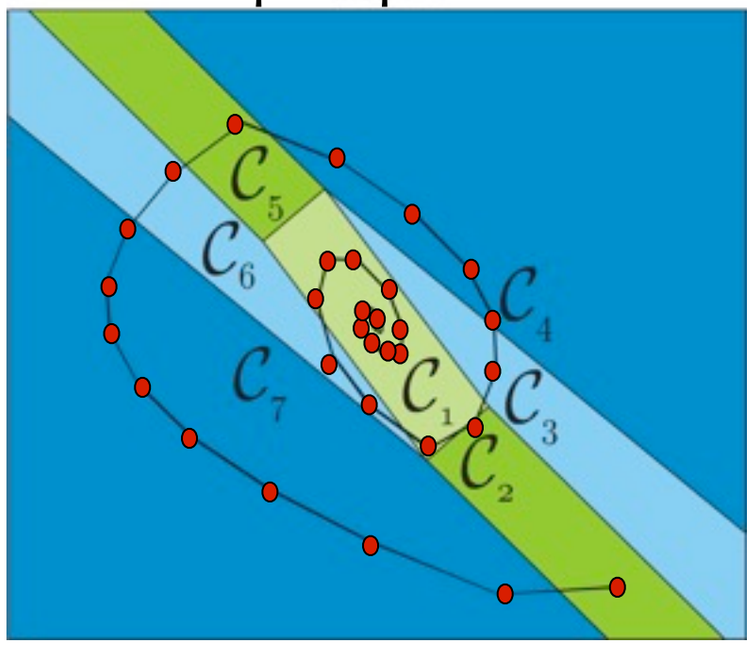
Piecewise affine systems

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

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

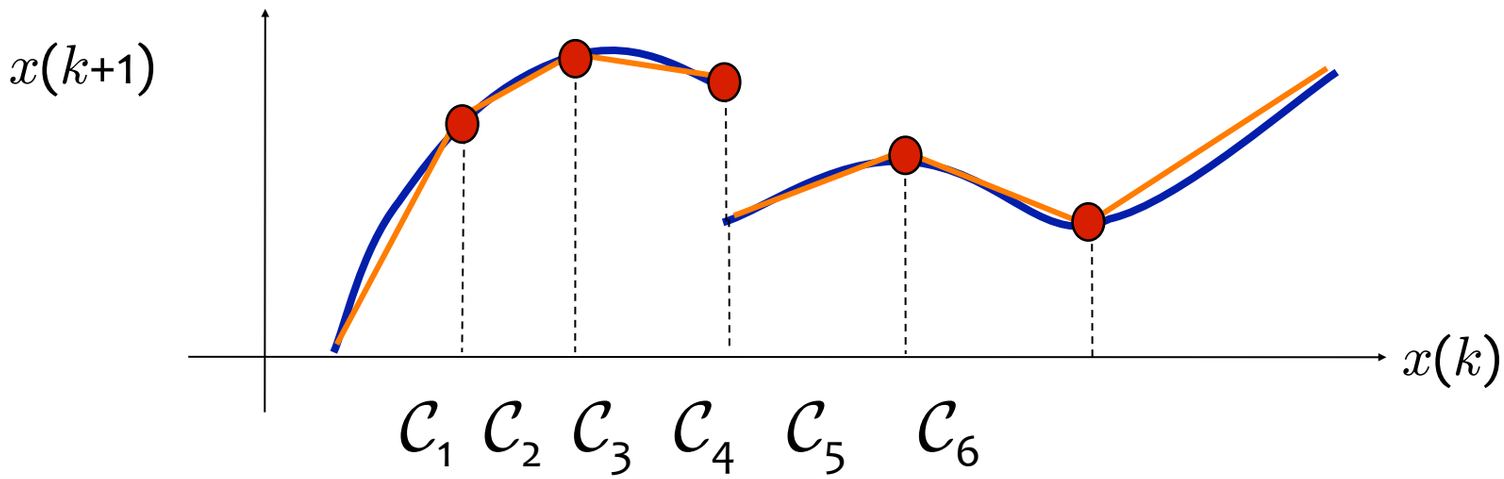
$i(k) \in \{1, \ldots, s\}$

state+input space



(Sontag 1981)

Can approximate nonlinear and/or discontinuous dynamics arbitrarily well



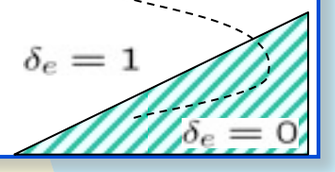
Discrete Hybrid Automaton (DHA)

(Torrisi, Bemporad, 2004)

discrete

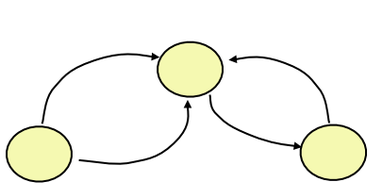
Event Generator

$$[\delta_e = 1] \leftrightarrow [Hx_c + Ku_c \leq W]$$



Finite State Machine

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



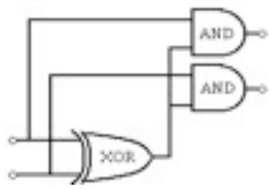
Switched Affine System

$$x_c(k+1) = A_i x_c(k) + B_i u_c(k) + f_i$$

$$i = i(k)$$

discrete time counter

Mode Selector



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

$x_c(k)$

$u_c(k)$

$x_c(k)$

$\delta_e(k)$

$u_\ell(k)$

$x_\ell(k)$

$u_\ell(k)$

$\delta_e(k)$

mode $i(k)$

continuous

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

$u_\ell \in \{0,1\}^{m_\ell}$ = 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

Logic $\rightarrow$ inequalities

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

(Glover 1975, Williams 1977, Hooker 2000)

0. Given a Boolean statement

$$F(X_1, X_2, \ldots, X_n) = \text{TRUE}$$

1. Convert to Conjunctive Normal Form (CNF):

$$\bigwedge_{j=1}^{m} \left(\bigvee_{i \in P_j} X_i \bigvee_{i \in N_j} \bar{X}_i \right) = \text{TRUE}$$

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

2. Transform into inequalities:

$$\begin{aligned} \sum_{i \in P_1} \delta_i + \sum_{i \in N_1} (1 - \delta_i) &\geq 1 \\ \vdots \quad \vdots \quad &\vdots \\ \sum_{i \in P_m} \delta_i + \sum_{i \in N_m} (1 - \delta_i) &\geq 1 \end{aligned}$$

$\Rightarrow$ polyhedron

$$A\delta \leq b, \quad \delta \in \{0,1\}^n$$

Any logic proposition can be translated into **integer linear inequalities**

Logic and inequalities

$X_1 \vee X_2 = \text{TRUE} \longrightarrow \delta_1 + \delta_2 \geq 1, \quad \delta_1, \delta_2 \in \{0,1\}$ (Glover 1975, Williams 1977, Hooker 2000)

Any logic statement
$f(X) = \text{TRUE}$

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

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

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

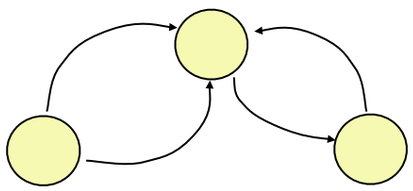
$[\delta_e^i(k) = 1] \leftrightarrow [H^i x_c(k) \leq W^i]$

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

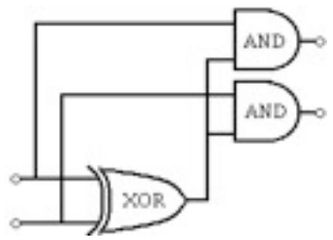
IF $[\delta = 1]$ THEN $z = a_1^T x + b_1^T u + f_1$
ELSE $z = a_2^T x + b_2^T u + f_2$

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

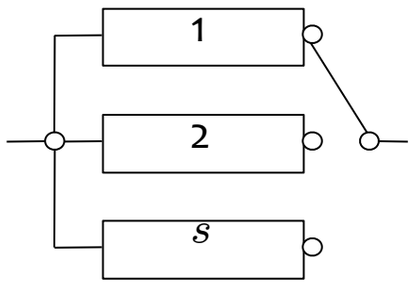
Finite State Machine



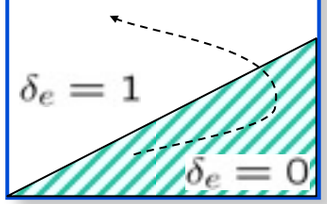
Mode Selector



Switched Affine System

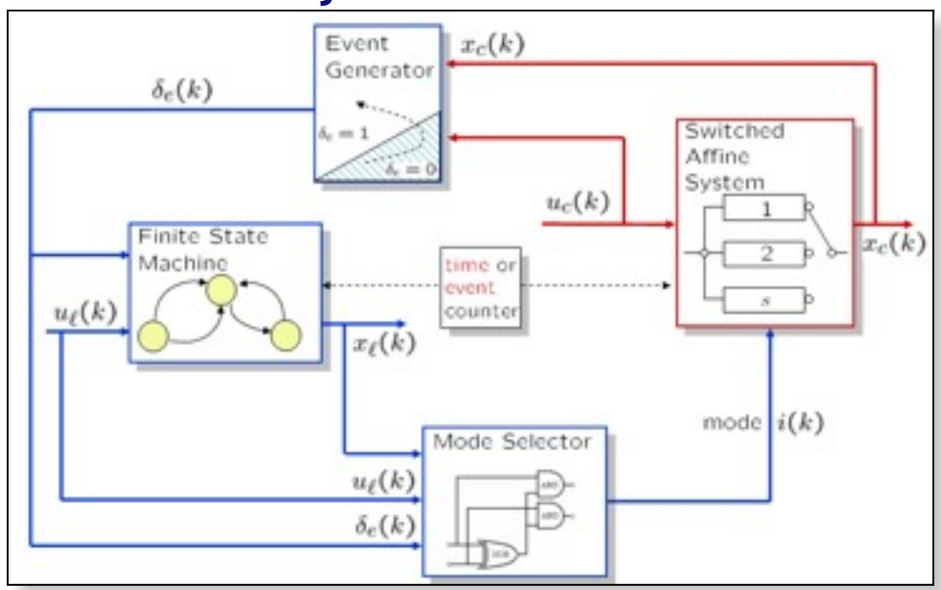


Event Generator



Mixed Logical Dynamical (MLD) systems

Discrete Hybrid Automaton



HYSDEL (Torrisi, Bemporad, 2004)

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

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

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

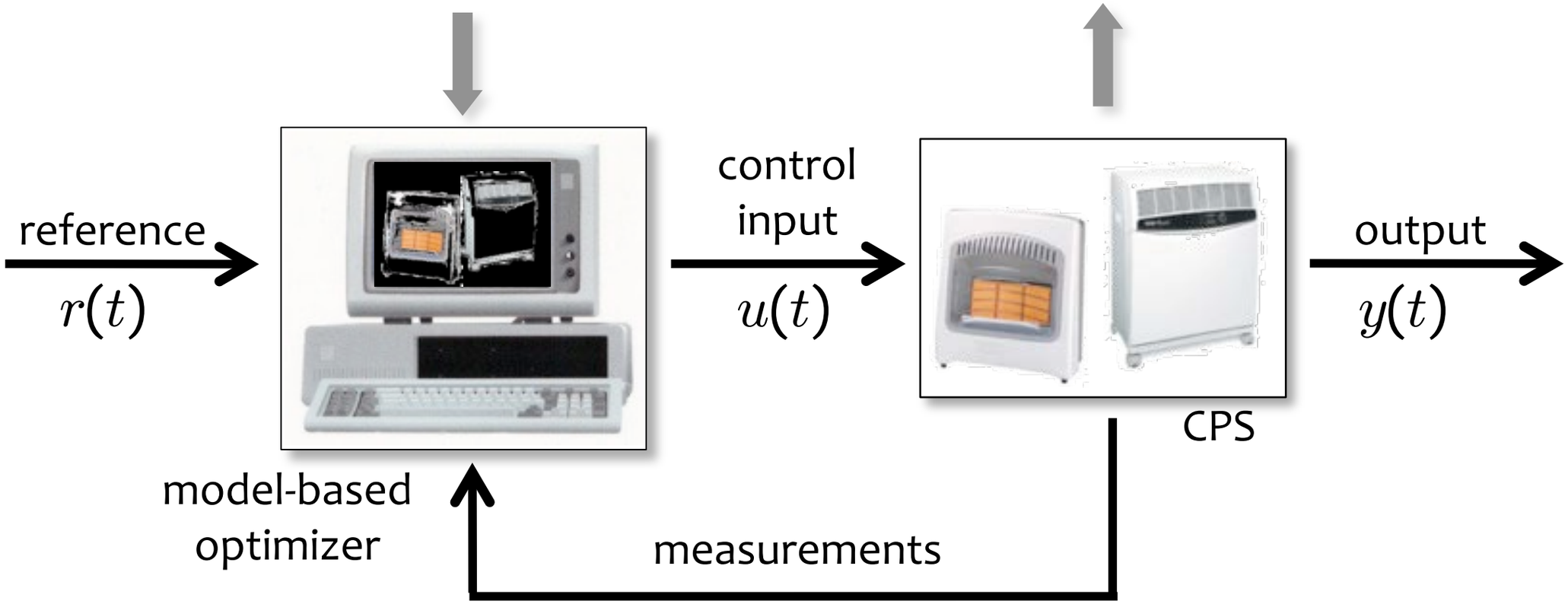
Continuous and binary variables

$$x \in \mathbb{R}^{n_c} \times \{0,1\}^{n_b},\ u \in \mathbb{R}^{m_c} \times \{0,1\}^{m_b}$$
$$y \in \mathbb{R}^{p_c} \times \{0,1\}^{p_b},\ \delta \in \{0,1\}^{r_b},\ z \in \mathbb{R}^{r_c}$$

- Computationally oriented model (mixed-integer programming)
- Suitable for MPC control, verification, state estimation, fault detection, ...

MPC of hybrid systems

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



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

MPC for hybrid systems

- At time t solve the finite-horizon optimal control problem w.r.t. $U \triangleq \begin{bmatrix} u_0 \\ \vdots \\ u_{N-1} \end{bmatrix}$

$$\min_{u_0,\dots,u_{N-1}} \quad \sum_{k=0}^{N-1} \|y_k - r\|_Q^2 + \|u_k - u_r\|_R^2 + \sigma\left(\|\delta_k - \delta_r\|^2 + \|z_k - z_r\|^2 + \|x_k - x_r\|^2\right)$$

subject to MLD equations

$$x_0 = x(t)$$

$$x_N = x_r$$

notation:

$\|v\|_Q^2 \triangleq v'Qv$

where the equilibrium condition $(x_r, u_r, \delta_r, z_r)$ is obtained by solving the following **mixed-integer feasibility problem**

$$\begin{cases} x_r = Ax_r + B_1u_r + B_2\delta_r + B_3z_r \\ r = Cx_r + D_1u_r + D_2\delta_r + D_3z_r \\ E_2\delta_r + E_3z_r \le E_4x_r + E_1u_r + E_5 \end{cases}$$

- Apply only $u(t)$=u_0* (discard the remaining optimal inputs)
- At time $t+1$: get new measurements, repeat optimization

MIQP formulation of MPC

(Bemporad, Morari, 1999)

$$\begin{aligned}\min_{\xi} \quad & \sum_{k=0}^{N-1} y_k' Q y_k + u_k' R u_k \\ \text{s.t.} \quad & \begin{cases} x_{k+1} = Ax_k + B_1 u_k + B_2 \delta_k + B_3 z_k + B_5 \\ y_k = Cx_k + D_1 u_k + D_2 \delta_k + D_3 z_k + D_5 \\ E_2 \delta_k + E_3 z_k \le E_4 x_k + E_1 u_k + E_5 \end{cases}\end{aligned}$$

- Optimization vector: $\xi = [u_0, \ldots, u_{N-1}, \delta_0, \ldots, \delta_{N-1}, z_0, \ldots, z_{N-1}]$

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

Mixed Integer Quadratic Program (MIQP)

$u \in \mathbb{R}^{m_c} \times \{0,1\}^{m_b}$

$\delta \in \{0,1\}^{r_b}$

$z \in \mathbb{R}^{r_c}$

$\xi \in \mathbb{R}^{(m_c+r_c)N} \times \{0,1\}^{(m_b+r_b)N}$

vector ξ has both **real** and **binary** values

Closed-loop convergence

Theorem Let $(x_r, u_r, \delta_r, z_r)$ be the equilibrium values corresponding to set point r. Assume $x(0)$ is such that the MPC problem is feasible at time $t=0$.

Then $\forall Q, R \succ 0,\ \forall \sigma > 0$ the closed-loop hybrid MPC loop **converges asymptotically**

$$\lim_{t\to\infty} y(t) = r \qquad \lim_{t\to\infty} x(t) = x_r$$
$$\lim_{t\to\infty} \delta(t) = \delta_r$$
$$\lim_{t\to\infty} u(t) = u_r \qquad \lim_{t\to\infty} z(t) = z_r$$

and all constraints are fulfilled at each time $t \geq 0$.

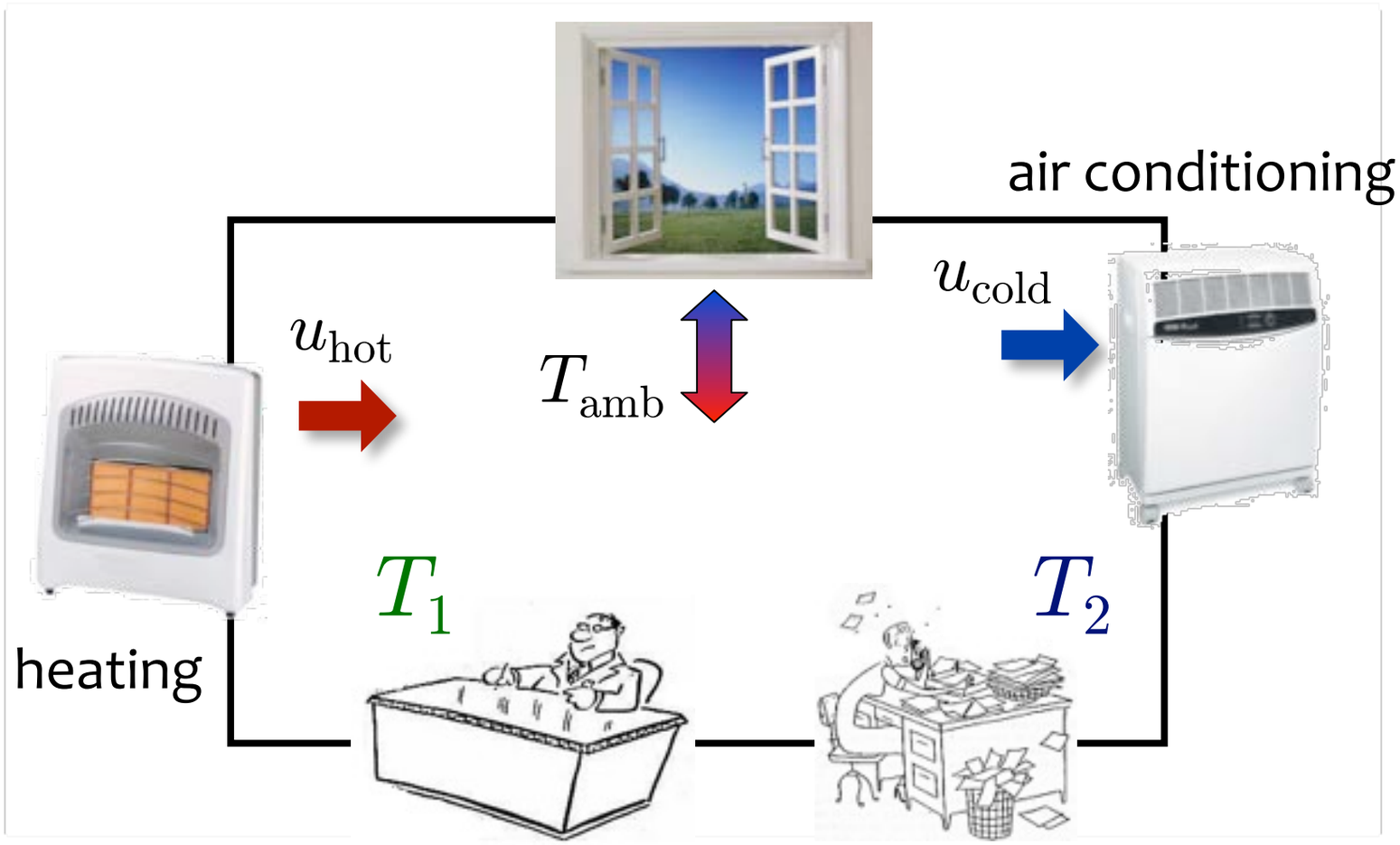
(Bemporad, Morari 1999)

Proof: Easily follows from standard Lyapunov arguments

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

(Lazar, Heemels, Weiland, Bemporad, 2006)

Example: Room temperature control



cyber dynamics

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

physical dynamics

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

$$i = 1, 2$$

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

HYSDEL model

```
SYSTEM heatcool {

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

```
>>S=mld('heatcoolmodel',Ts)
```

get the MLD model in MATLAB

```
>>[XX,TT]=sim(S,x0,U);
```

simulate the MLD model

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)

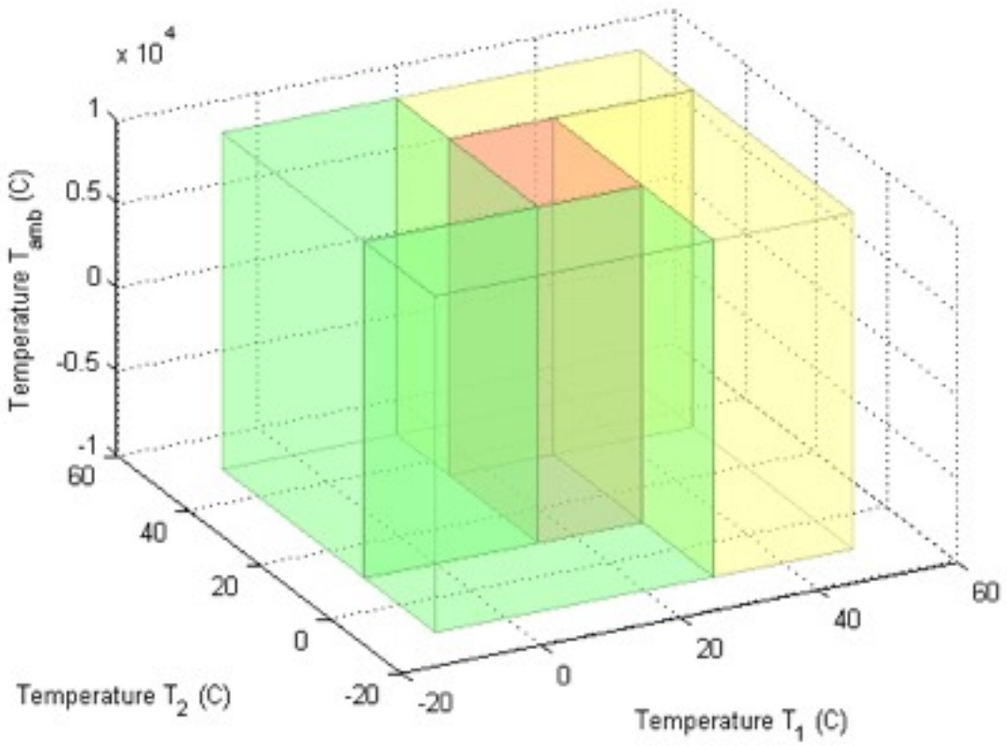
Hybrid PWA model

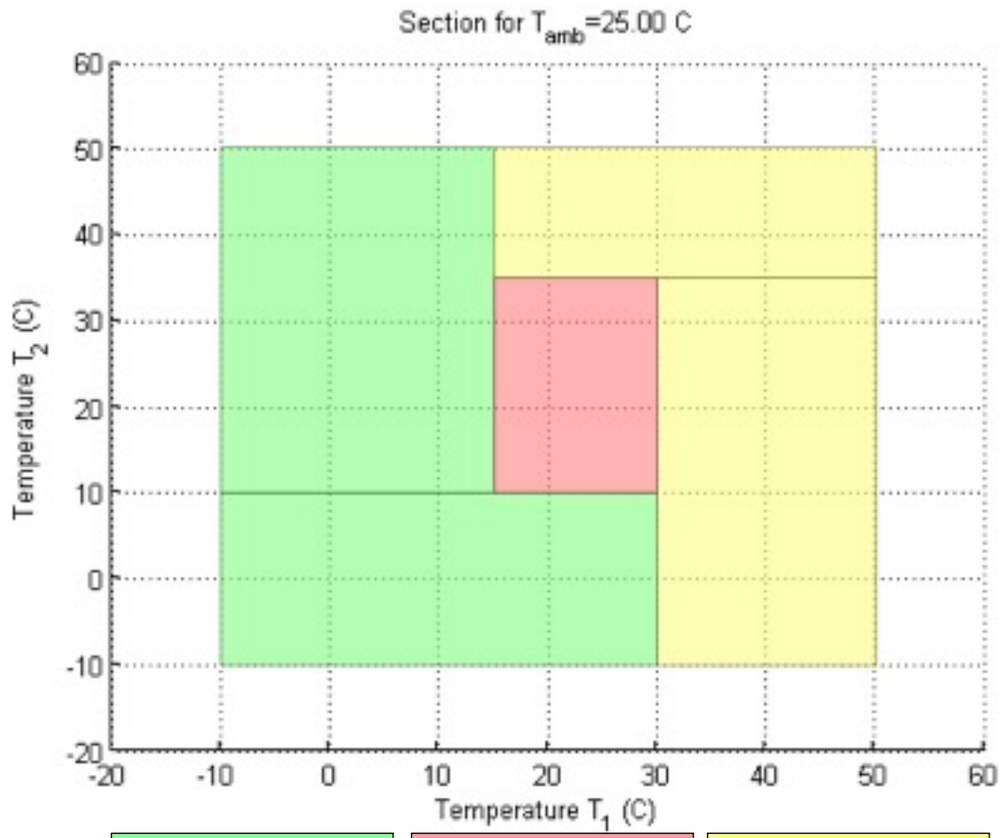
- PWA model

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

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

- 2 continuous states:
  (temperatures T_1,T_2)
- 1 continuous input:
  (room temperature T_{amb})





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

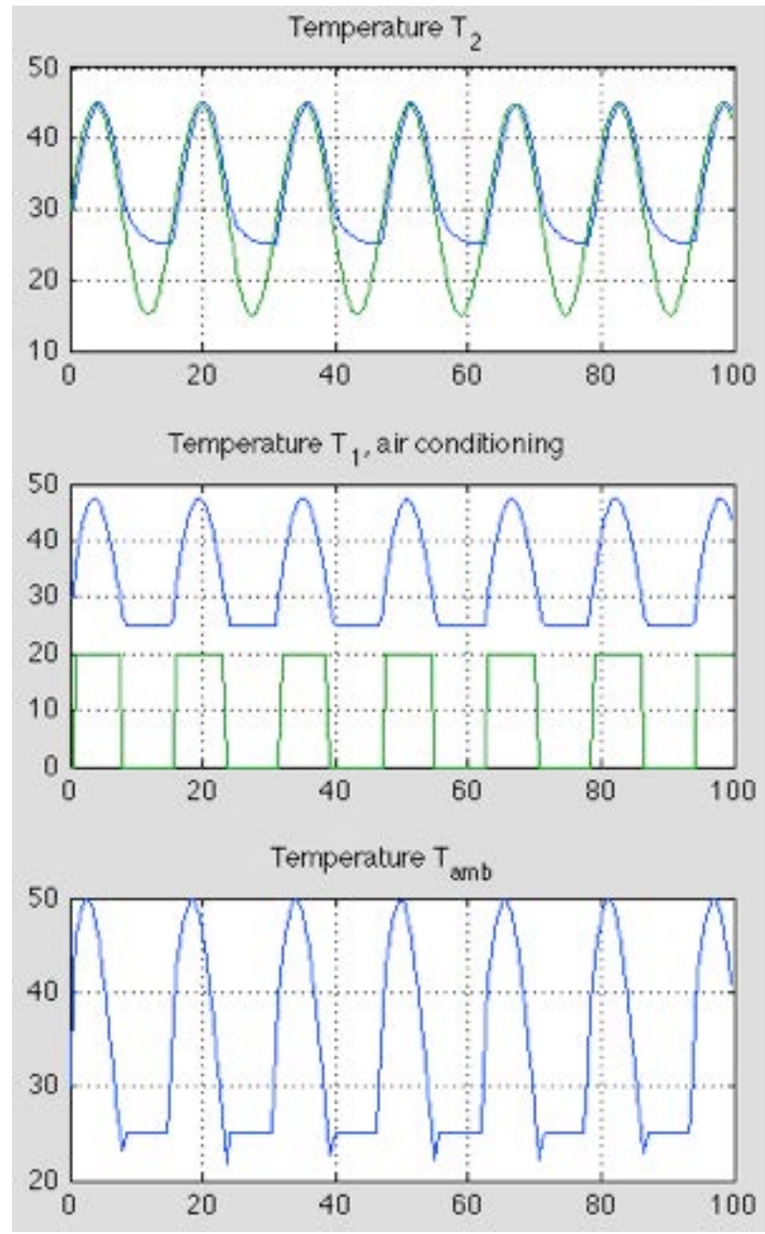
Hybrid MPC – Temperature control

```
>>refs.x=2;    % just weight state #2
>>Q.x=1;       % unit weight on state #2
>>Q.rho=Inf;   % hard constraints
>>Q.norm=Inf; % infinity norms
>>N=2;         % prediction horizon
>>limits.xmin=[25;-Inf];
```

$$\min \sum_{k=0}^{2} \|x_{2k} - r(t)\|_\infty$$
$$\text{s.t. } \begin{cases} x_{1k} \geq 25, \ k = 1,2 \\ \text{MLD model} \end{cases}$$

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

```
>> C

Hybrid controller based on MLD model S <heatcoolmodel.hys> [Inf-norm]

  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);
```

Explicit hybrid MPC

- It is possible to write hybrid MPC laws in explicit form too !
- The explicit MPC law is still piecewise affine on polyhedra
- The control law may be discontinuous, polyhedra may overlap

(Bemporad, Borrelli, Morari, 2000)

(Mayne, ECC 2001)

(Mayne, Rakovic, 2002)

(Bemporad, *Hybrid Toolbox*, 2003)

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

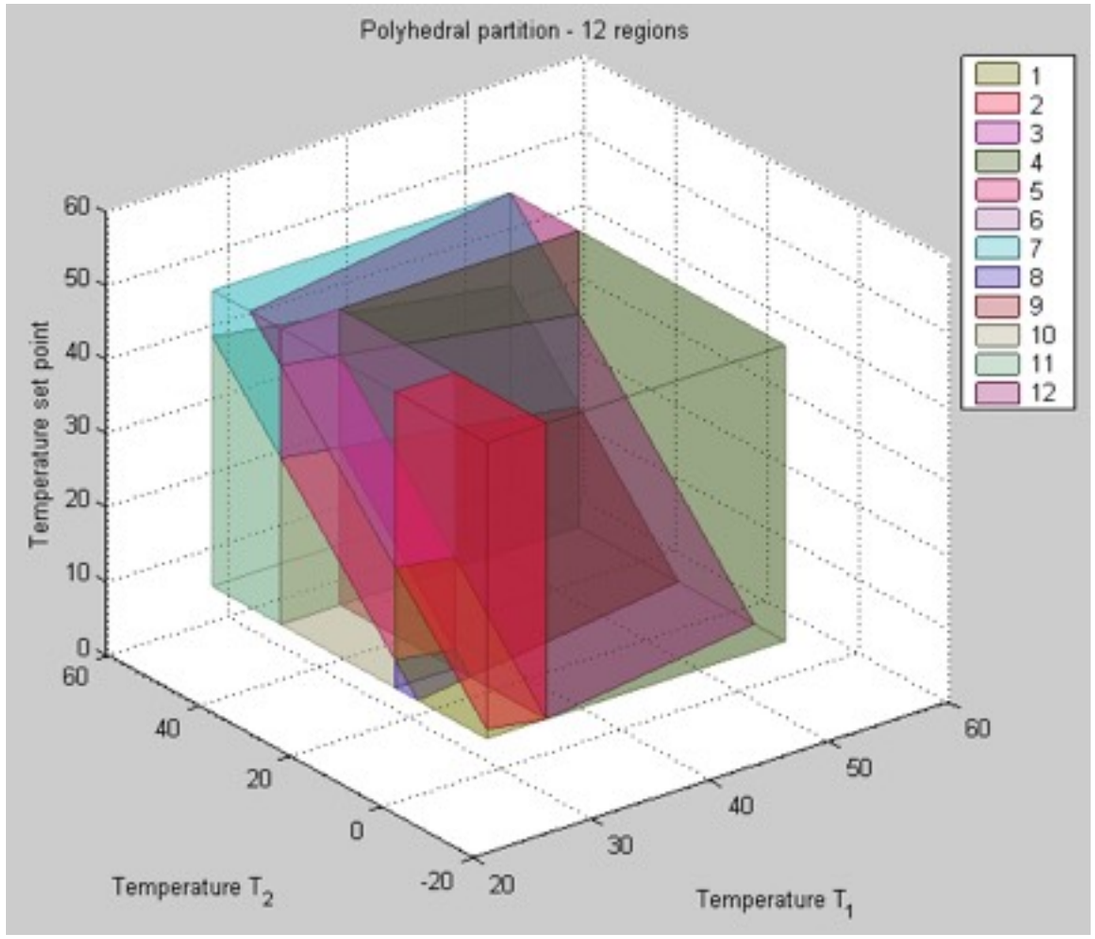
(Alessio, Bemporad, ADHS 2006)

Explicit MPC – Temperature control

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

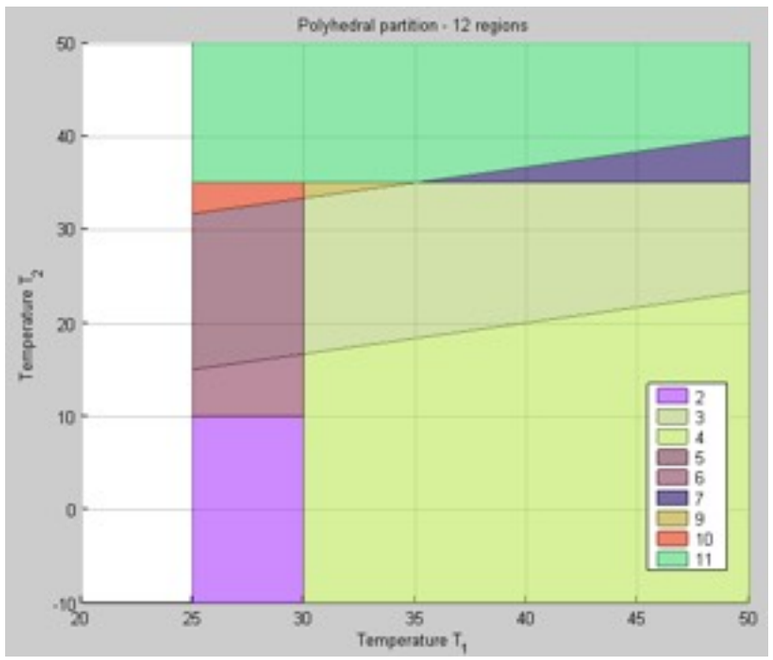
```
>> E

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

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

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

384 numbers to store in memory



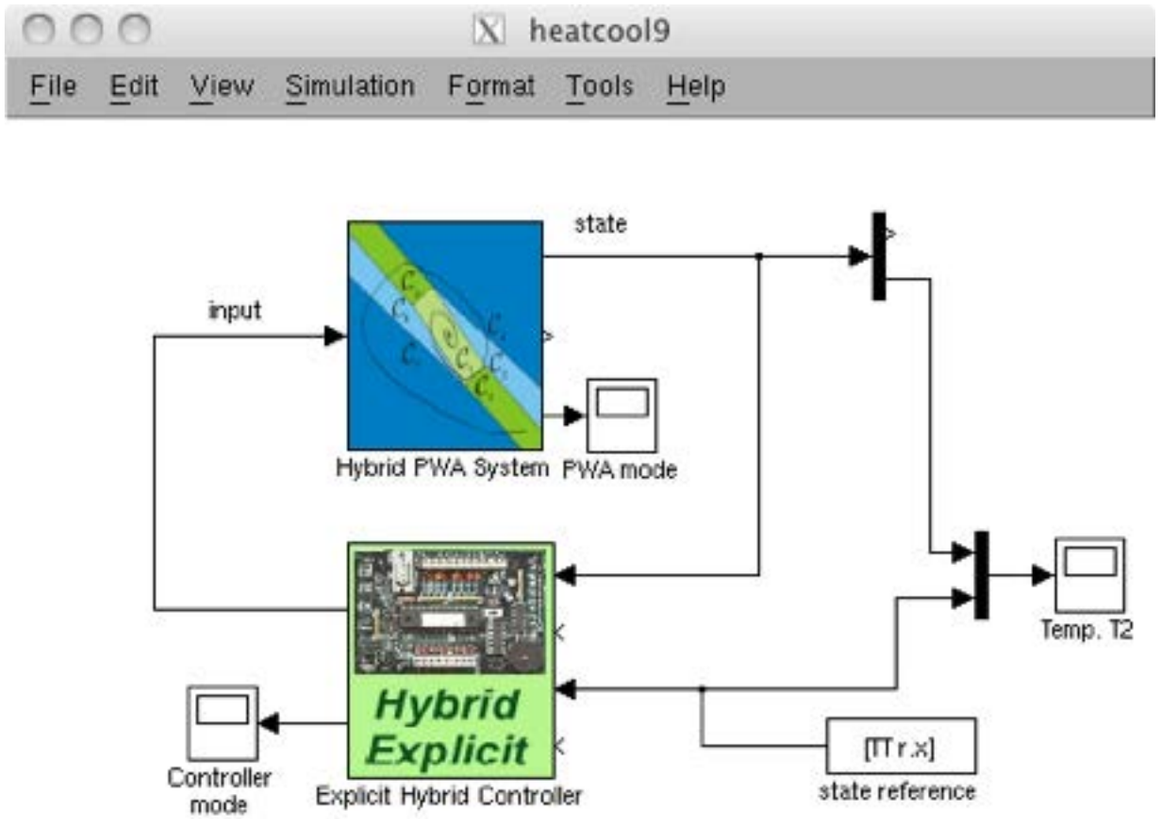
$$\min \sum_{k=0}^{2} \|x_{2k} - r(t)\|_\infty$$

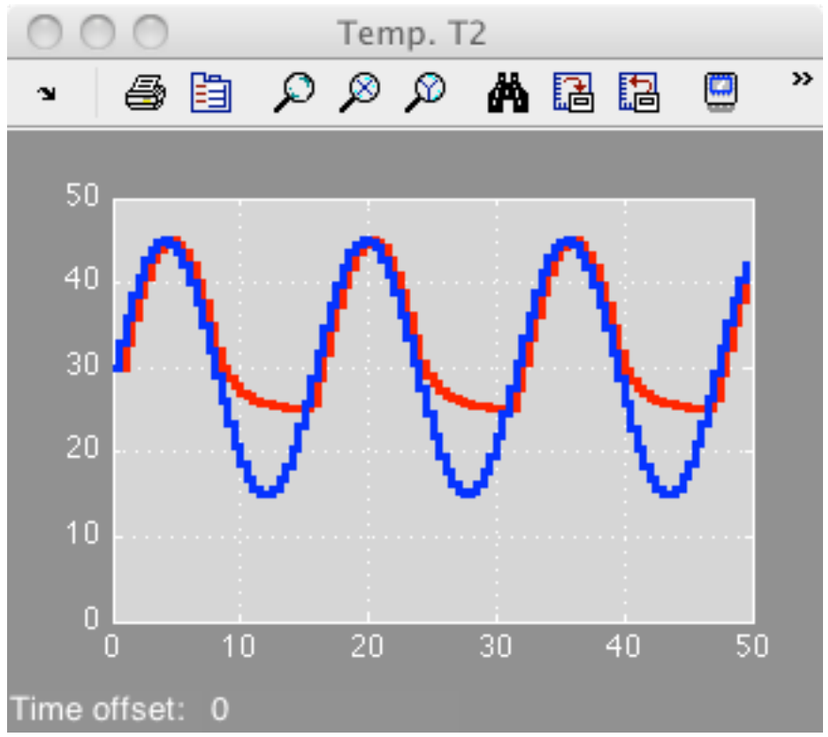
$$\text{s.t.} \begin{cases} x_{1k} \geq 25, \ k = 1,2 \\ \text{hybrid model} \end{cases}$$



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

Explicit MPC – Temperature control

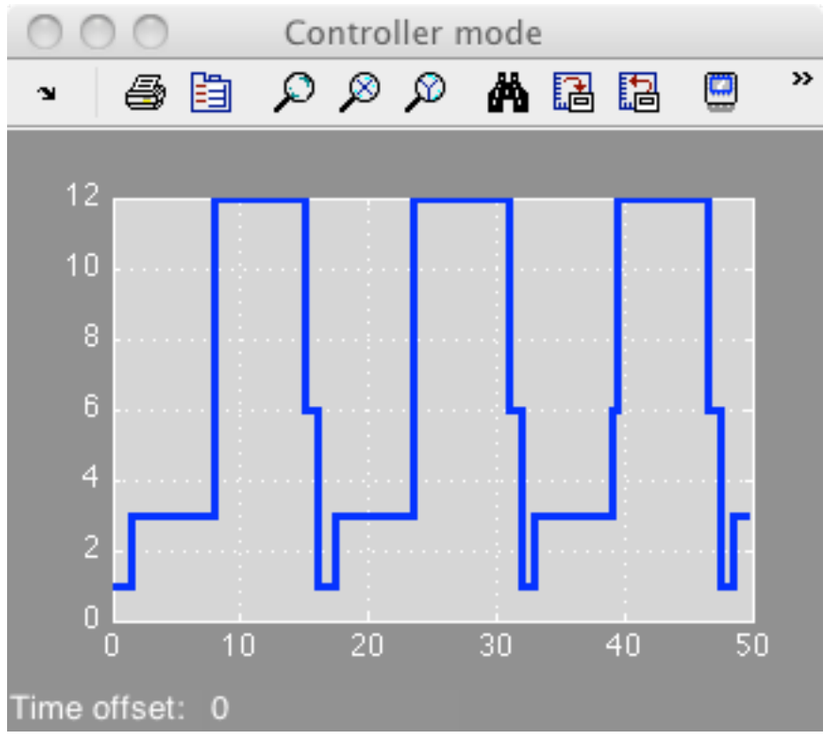




generated C-code

utils/expcon.h

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

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

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



Conclusions

- MPC is a **very versatile** technique to provide "intelligence" to a large class of cyber-physical systems

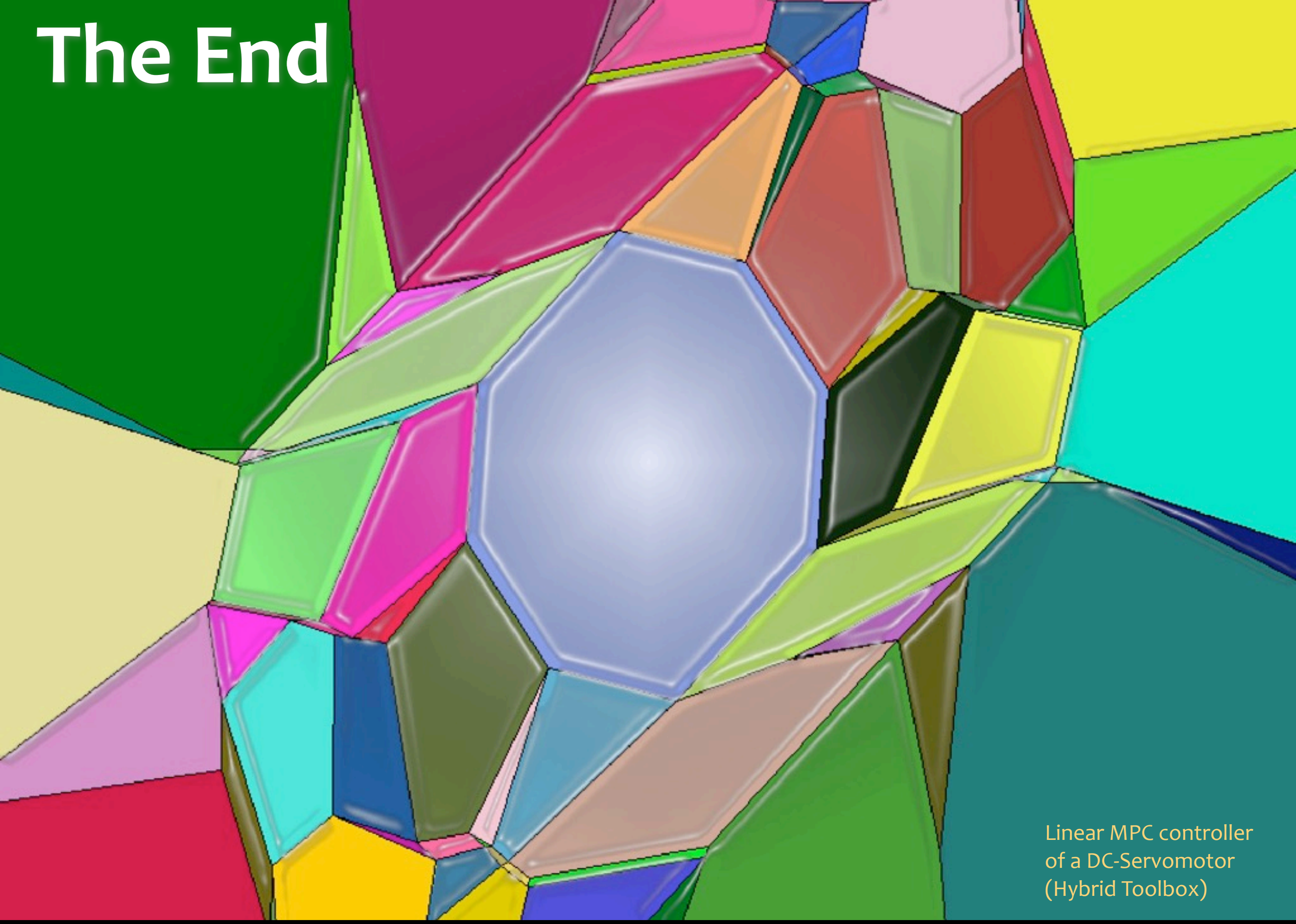
- MPC can easily handle **multiple** inputs and outputs, **hybrid** model abstractions, **constraints** on variables for safety, **optimal** performance

- Routinely used in the *process industries* from the 80's, it's gaining increasing impact in automotive, aerospace, energy and other sectors

- A library of **solvers tailored to embedded MPC** applications is available that are very **simple to code**, **fast**, amenable for **low-precision arithmetic** (single or fixed-point), and with **proved bounds** on real-time execution

MPC is a winning control technology for modern cyber-physical systems !

The End



Linear MPC controller
of a DC-Servomotor
(Hybrid Toolbox)

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

Bibliography

General references on MPC

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

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

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

[4] A. Bemporad, *Hybrid Toolbox – User's Guide*, Jan. 2004, `http://www.ing.unitn.it/~bemporad/hybrid/toolbox`

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

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

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

Bibliography

MLD and HYSDEL Modeling

[1] A. Bemporad, "Hybrid Toolbox – User's Guide," Dec. 2003, `http://cse.lab.imtlucca.it/~bemporad/hybrid/toolbox`

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

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

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

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

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

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

Bibliography

Identification of hybrid systems

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

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

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

Model predictive control

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

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

[13] A. Bemporad, "A multiparametric quadratic programming algorithm with polyhedral computations based on nonnegative least squares," *IEEE Trans. Automatic Control*, 2015, In press.

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

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

Bibliography

Reachability and observability

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

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

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

Selected Applications

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

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

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

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

more on **`http://cse.lab.imtlucca.it/~bemporad/publications`**