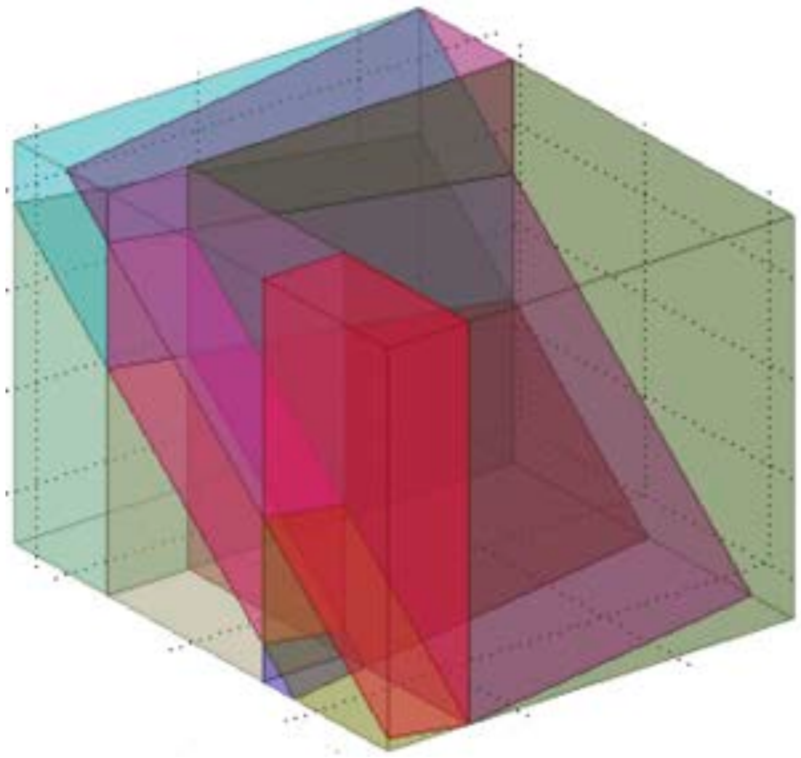
MODEL PREDICTIVE CONTROL FOR CYBER-PHYSICAL SYSTEMS

Alberto Bemporad

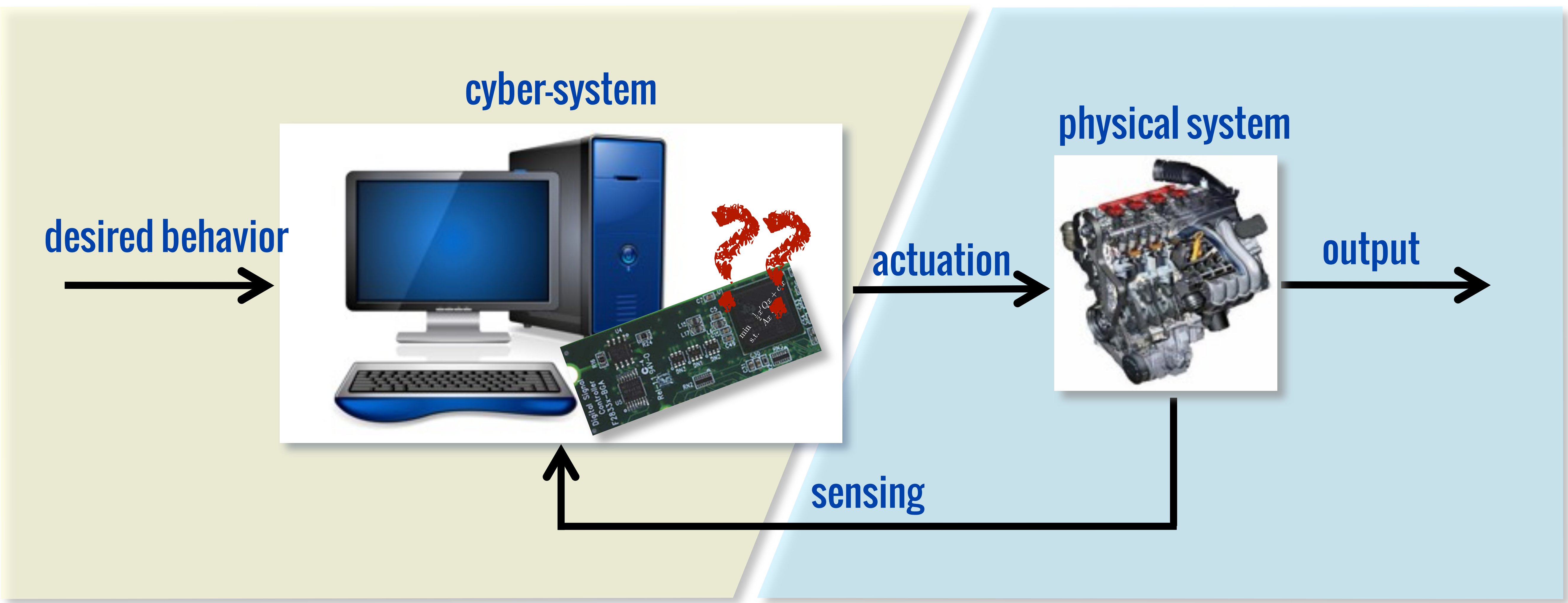
http://cse.lab.imtlucca.it/~bemporad

IMT SCHOOL FOR ADVANCED STUDIES LUCCA



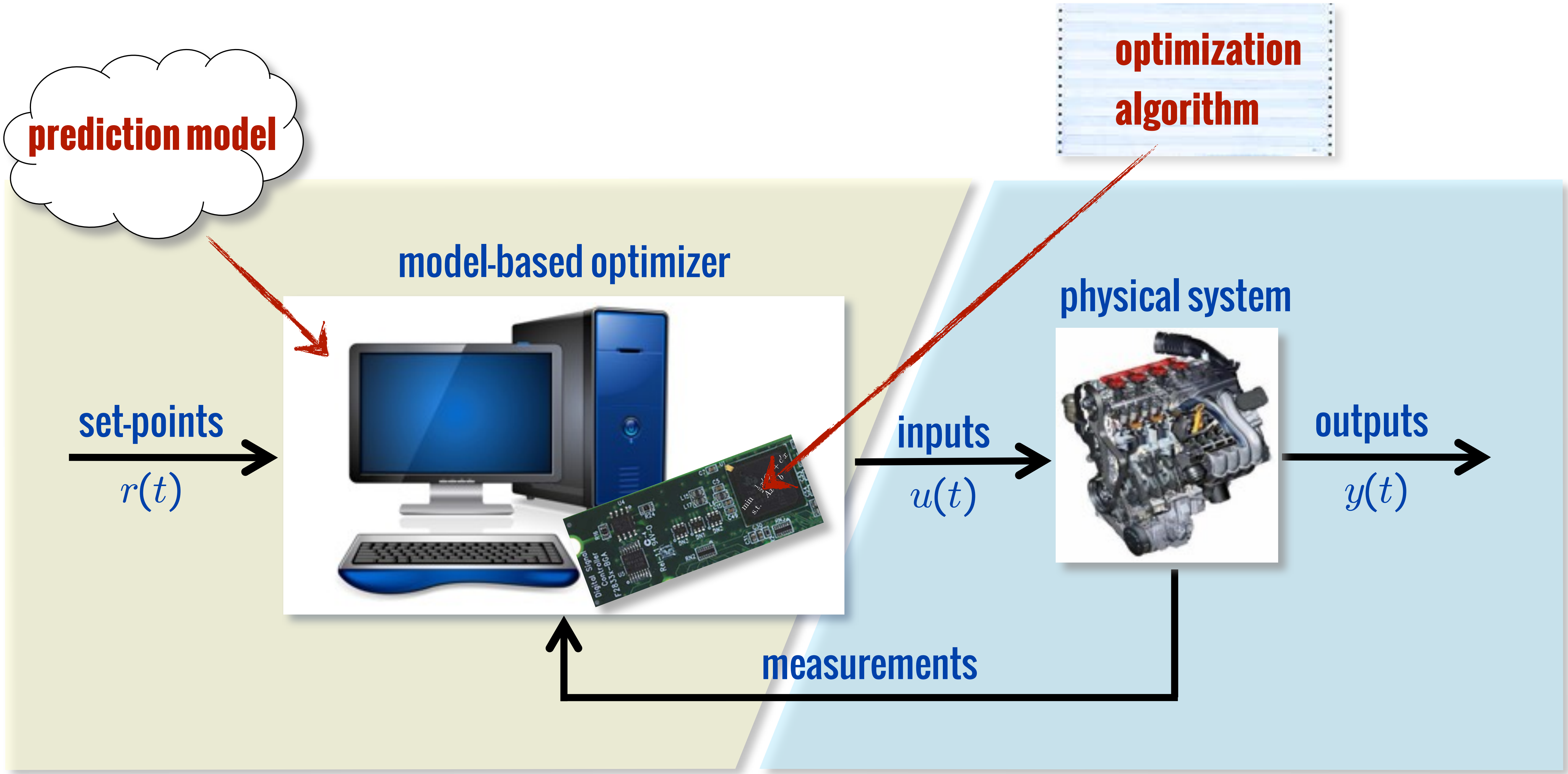
oCPS PhD School - Lucca, June 14, 2017

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)



simplified likely

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

a good

MODEL PREDICTIVE CONTROL (MPC)

- At each time t, find the best control sequence over a future horizon of N steps

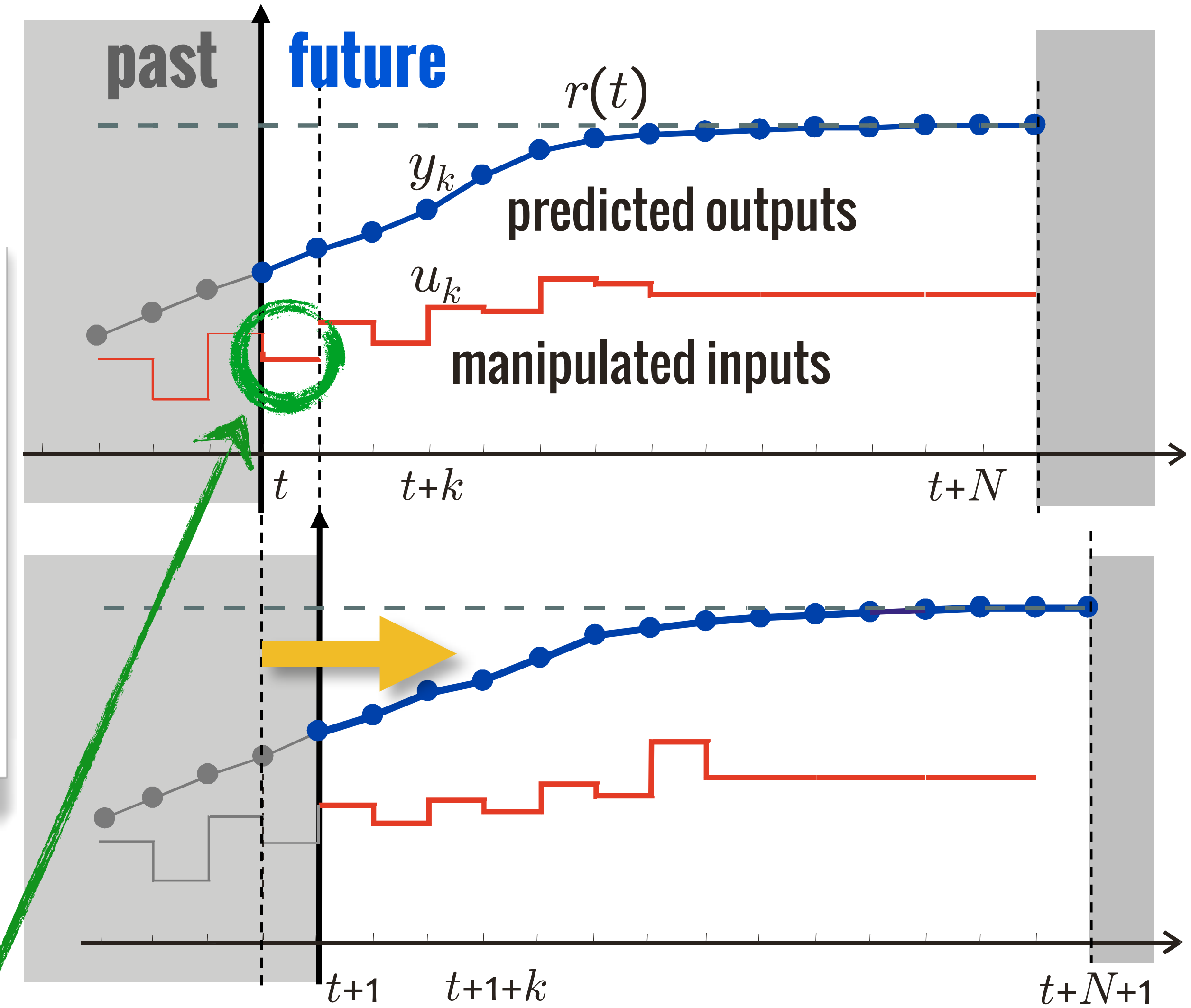
penalty on tracking error

penalty on actuation

$$
\begin{aligned}
\min \quad & \sum_{k=0}^{N-1} \|W^y(y_k - r(t))\|^2 + \|W^u(u_k - u^{\text{ref}}(t))\|^2 \\
\text{s.t.} \quad & x_{k+1} = f(x_k, u_k, t) \\
& y_k = g(x_k, u_k, t) \\
& \text{constraints on } u_k, y_k \\
& x_0 = x(t)
\end{aligned}
$$

$\longleftarrow$ feedback !

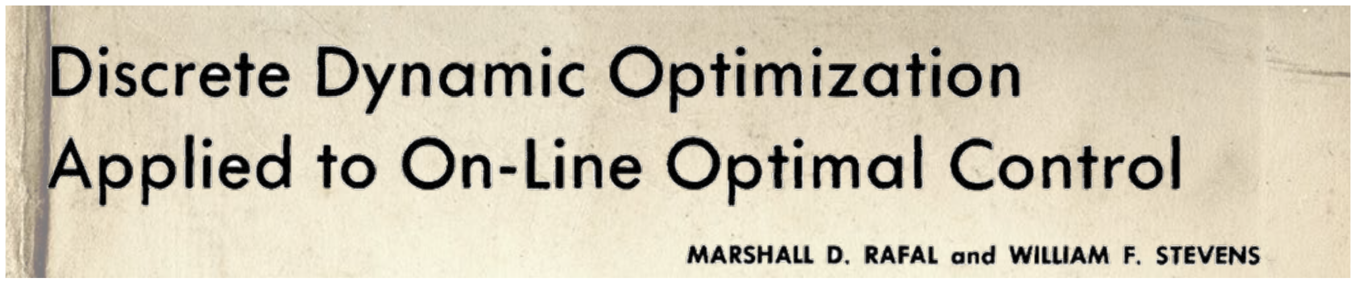
optimization problem



- Problem solved 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 …

MPC IN INDUSTRY

- The MPC concept for process control dates back to the 60's



Discrete Dynamic Optimization
Applied to On-Line Optimal Control

MARSHALL D. RAFAL and WILLIAM F. STEVENS

(Rafal, Stevens, *AiChE Journal*, 1968)

- MPC used in the process industries since the 80's

©SimulateLive.com

MPC is the standard for advanced control in the process industry.

(Qin, Badgewell, 2003) (Bauer & Craig, 2008)

- Research in MPC is still very active !

MPC IN INDUSTRY

(Samad, *IEEE CS Magazine*, Feb 2017)

- Impact of advanced control technologies in industry

TABLE 1 A list of the survey results in order of industry impact as perceived by the committee members.

Rank and Technology	High-Impact Ratings	Low- or No-Impact Ratings
PID control	100%	0%
Model predictive control	78%	9%
System identification	61%	9%
Process data analytics	61%	17%
Soft sensing	52%	22%
Fault detection and identification	50%	18%
Decentralized and/or coordinated control	48%	30%
Intelligent control	35%	30%
Discrete-event systems	23%	32%
Nonlinear control	22%	35%
Adaptive control	17%	43%
Robust control	13%	43%
Hybrid dynamical systems	13%	43%

AUTOMOTIVE APPLICATIONS OF MPC

Bemporad, Bernardini, Borrelli, Cimini, Di Cairano, Esen, Giorgetti, Graf-Plessen, Hrovat, Kolmanovsky, Levijoki, Ripaccioli, Trimboli, Tseng, Yanakiev, ... (2001-present)

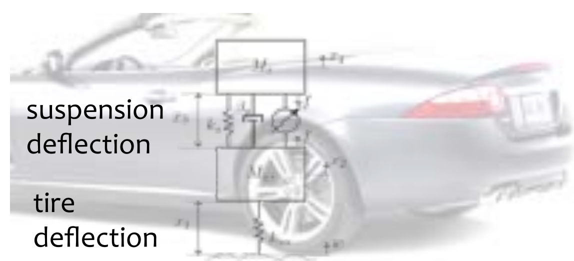
Powertrain

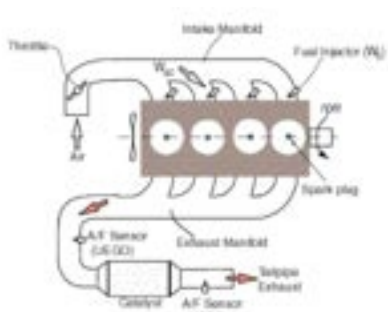
- direct-inj. engine control
- A/F ratio control
- magnetic actuators
- robotized gearbox
- power MGT in HEVs
- cabin heat control in HEVs
- electrical motors

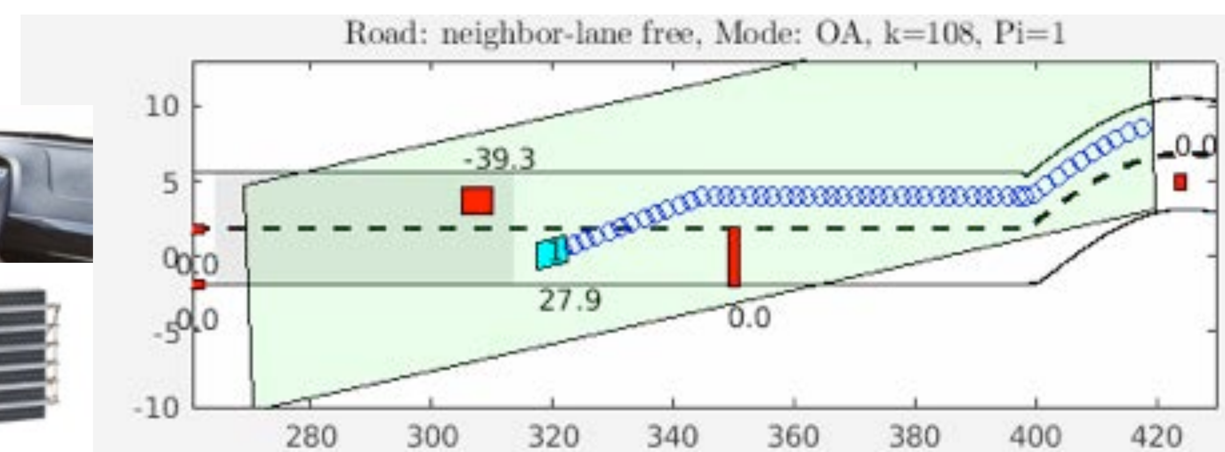
Vehicle dynamics

- traction control
- active steering
- semiactive suspensions
- autonomous driving







Most automotive OEMs are adopting MPC solutions today

MPC IN THE AERONAUTIC INDUSTRY

PRESS RELEASE

Pratt & Whitney's F135 Advanced Multi-Variable Control Team Receives UTC's Prestigious George Mead Award for Outstanding Engineering Accomplishment

EAST HARTFORD, CONN., THURSDAY MAY 27, 2010

Pratt & Whitney engineers Louis Celiberti, Timothy Crowley, James Fuller and Cary Powell won the George Mead Award – United Technologies Corp.'s highest award for outstanding engineering achievement – for their pioneering work in developing the world's first advanced multi-variable control (AMVC) design for the only engine that powers the F-35 Lightning II flight test program. Pratt & Whitney is a United Technologies Corp. (NYSE:UTX) company.

The AMVC, which uses a proprietary model predictive control methodology, is the most technically advanced propulsion system control ever produced by the aerospace industry, demonstrating the highest pilot rating for flight performance and providing independent control of vertical thrust and pitch from five sources. This innovative and industry-leading advanced design is protected with five broad patents for Pratt & Whitney and UTC, and is the new standard for propulsion system control for Pratt & Whitney military and commercial engines.

http://www.pw.utc.com/Press/Story/20100527-0100/2010

Pratt & Whitney
A United Technologies Company

CONTENTS OF MY LECTURE

- Model Predictive Control (MPC) ***for*** CPS's

- Embedded quadratic optimization algorithms (***inside*** the CPS)

- Hybrid MPC = supervisory control ***of*** CPSs

LINEAR MPC

- Linear prediction model:

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

$$x_0 = x(t)$$

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

- Constraints to enforce:

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

- Constrained optimal control problem (quadratic performance index):

$$\begin{array}{ll} \min\limits_{z} & x_N' P x_N + \sum\limits_{k=0}^{N-1} x_k' Q x_k + u_k' R u_k \\ \\ \text{s.t.} & u_{\min} \leq u_k \leq u_{\max}, \ k = 0, \ldots, N-1 \\ & y_{\min} \leq y_k \leq y_{\max}, \ k = 1, \ldots, N \end{array}$$

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

LINEAR MPC - CONSTRAINED CASE

- State response: $x_k = A^k x_0 + \sum_{i=0}^{k-1} A^i B u_{k-1-i}$

- Optimization problem:

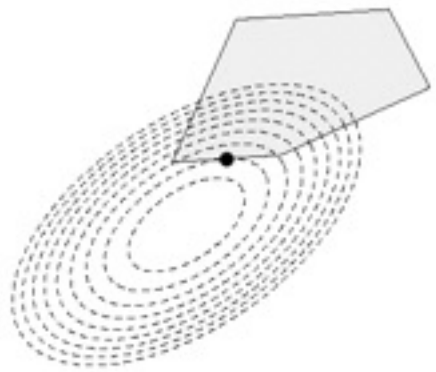
$$
\begin{aligned}
V(x_0) = \tfrac{1}{2}x_0' Y x_0 + \quad \min_z \quad & \tfrac{1}{2} z' H z + x_0' F' z \\
\text{s.t.} \quad & Gz \leq W + S x_0
\end{aligned}
$$

(quadratic)

(linear)

Convex QUADRATIC PROGRAM (QP)

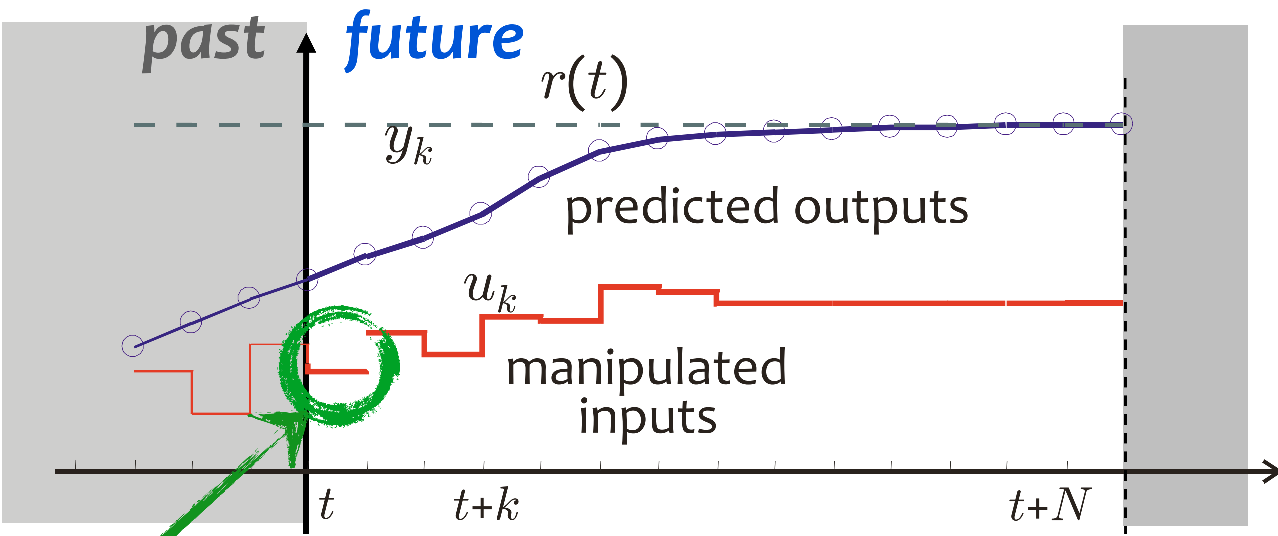
- $z = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix} \in \mathbb{R}^s, \quad s \triangleq Nm$ is the optimization vector



- $H = H' \succ 0$ 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 $x(t)$

feedback !

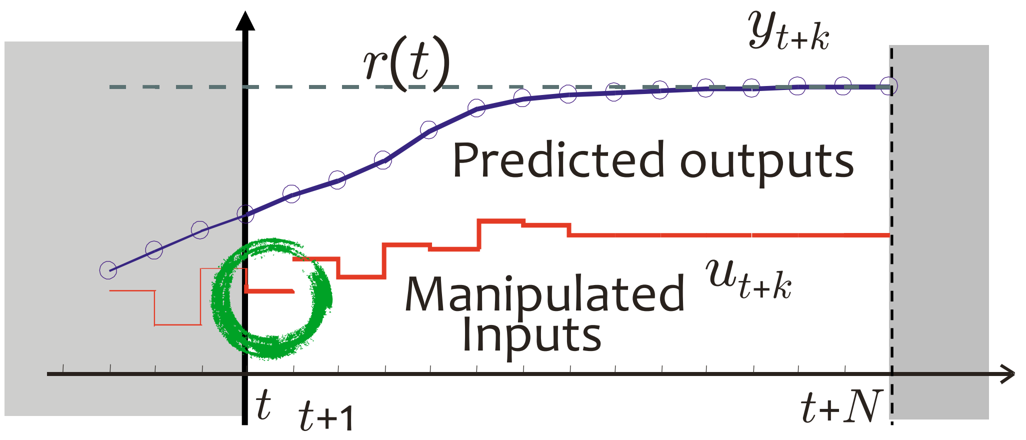
$$\begin{array}{ll} \min_{z} & \frac{1}{2} z' H z + x'(t) F' z \\ \text{s.t.} & G z \le W + S x(t) \end{array}$$

- Get the solution $z^* = \{ u_0^*, \ldots, u_{N-1}^* \}$ of the QP

- Apply only $u(t) = u_0^*$, discard remaining optimal inputs $u_1^*, \ldots, u_{N-1}^*$

LINEAR MPC - UNCONSTRAINED CASE

- Minimize quadratic function, no constraints



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

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

- Solution: $\nabla f(z) = Hz + Fx(t) = 0$

$$\longrightarrow \quad z^* = -H^{-1} F x(t)$$

$$\Rightarrow \quad u(t) = -[I \ 0 \ \ldots \ 0] H^{-1} F x(t) \triangleq K x(t)$$

Unconstrained linear MPC = linear state-feedback !

MPC AND LINEAR QUADRATIC REGULATION (LQR)

- Special case: $J(z, x_0) = \min_z x_N' P x_N + \sum_{k=0}^{N-1} x_k' Q x_k + u_k' R u_k$

with matrix P solving the Algebraic Riccati Equation

$$P = A'PA - A'PB(B'PB + R)^{-1}B'PA + Q$$

Jacopo Francesco Riccati (1676 - 1754)

- **(unconstrained) MPC = LQR** (for any choice of the prediction horizon N)

Proof. Easily follows from Bellman's principle of optimality (dynamic programming): $x_N' P x_N$ = optimal "cost-to-go" from time N to ∞.

BASIC CONVERGENCE PROPERTIES

Theorem. Consider the linear system $\left\{\begin{array}{rcl} x(t+1) & = & Ax(t)+Bu(t) \\ y(t) & = & Cx(t) \end{array}\right.$

and the MPC control law based on optimizing

$$\begin{array}{rl} V^*(x(t)) = \min & \displaystyle\sum_{k=0}^{N-1} x_k' Q x_k + u_k' R u_k \\ \text{s.t.} & x_{k+1} = Ax_k + Bu_k \\ & u_{\min} \le u_k \le u_{\max} \\ & y_{\min} \le Cx_k \le y_{\max} \\ & x_N = 0 \end{array}$$

"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 $t\geq 0$, 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)

LINEAR MPC - TRACKING

- Optimal control problem (quadratic performance index):

$$\begin{array}{ll} \min\limits_{z} & \displaystyle\sum_{k=0}^{N-1} \|W^y(y_{k+1}-r(t))\|^2 + \|W^{\Delta u}\Delta u_k\|^2 \\ & \qquad\qquad [\Delta u_k \triangleq u_k - u_{k-1}],\ u_{-1} = u(t-1) \\ \text{subj. to} & u_{\min} \le u_k \le u_{\max},\ k = 0,\ldots,N-1 \\ & y_{\min} \le y_k \le y_{\max},\ k = 1,\ldots,N \\ & \Delta u_{\min} \le \Delta u_k \le \Delta u_{\max},\ k = 0,\ldots,N-1 \end{array}$$

optimization vector

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

- Optimization problem:

Convex QUADRATIC PROGRAM (QP)

$$\begin{array}{ll} \min\limits_{z} & J(z,x(t)) = \frac{1}{2}z'Hz + [x'(t)\ r'(t)\ u'(t-1)]F'z \\ \text{s.t.} & Gz \le W + S\begin{bmatrix} x(t) \\ r(t) \\ u(t-1) \end{bmatrix} \end{array}$$

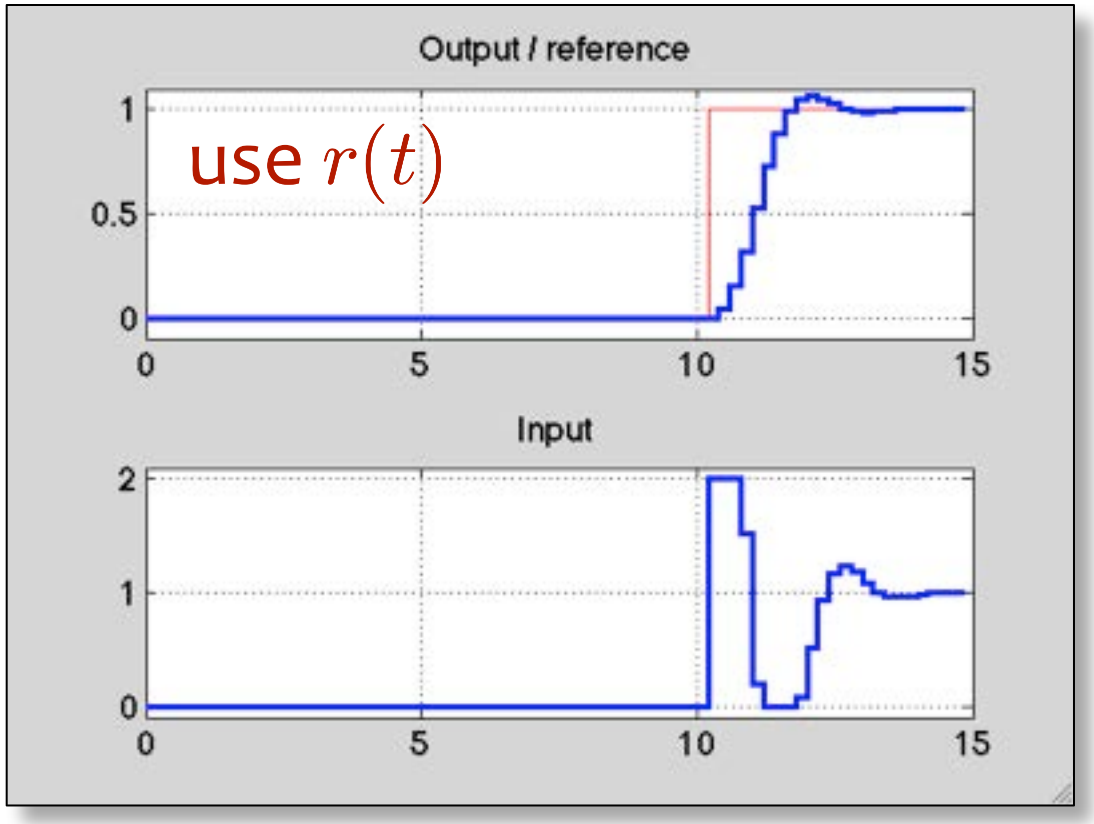
- Input references can be also handled by adding the extra penalty $\|W^u(u_k - u_{\text{ref}}(t))\|^2$
- Constraints on tracking errors can be also included: $e_{\min} \le y_k - r(t) \le e_{\max}$

ANTICIPATIVE ACTION (A.K.A. "PREVIEW")

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

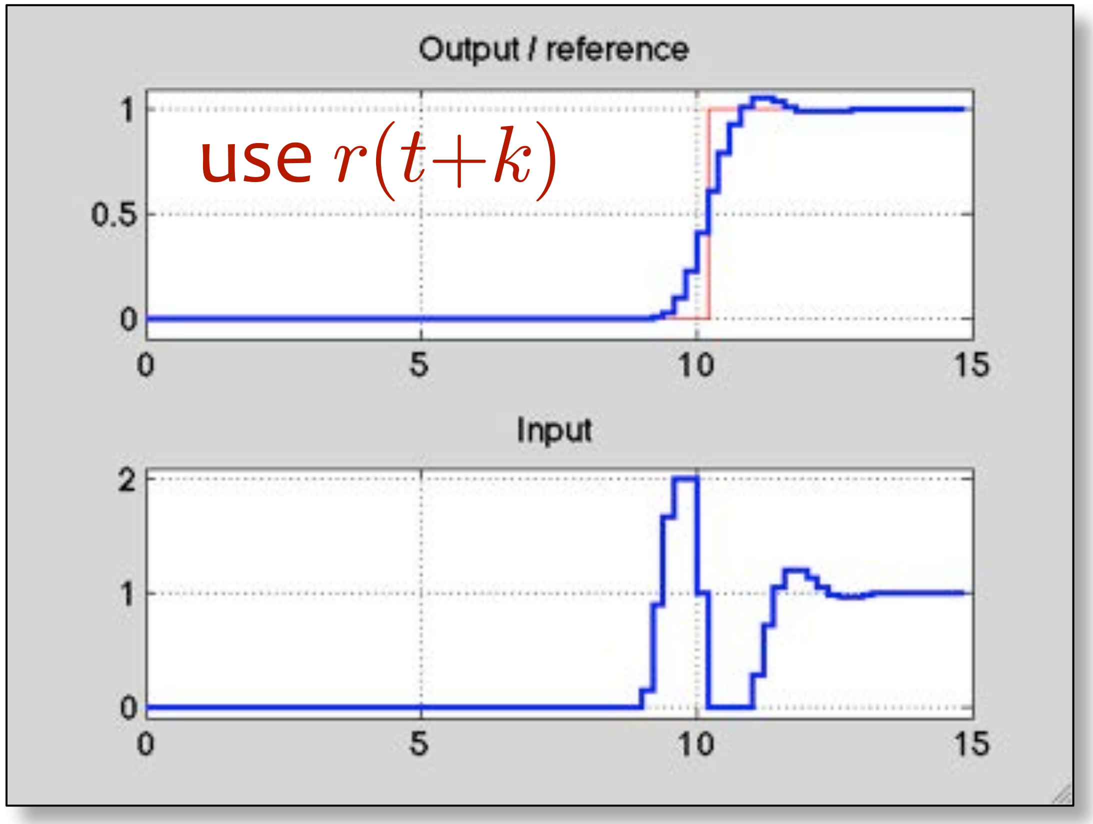
- Reference not known in advance (causal):

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



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

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



Same idea also applies to reject **measured disturbances** entering the process

MPC Toolbox for MATLAB, `mpcpreview.m`

LINEAR PARAMETER-VARYING (LPV) MPC

LTI prediction model

$$\begin{cases} x_{k+1} = A(p(t))x_k + B_u(p(t))u_k + B_v(p(t))v_k \\ y_k = C(p(t))x_k + D_v(p(t))v_k \end{cases} \qquad x_0 = x(t)$$

Model depends on time t but does not change in prediction

quadratic performance index

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

QP

$$\begin{array}{ll} \min\limits_z & \frac{1}{2} z' H(p(t)) z + \theta'(t) F(p(t))' z \\ \text{s.t.} & G(p(t)) z \leq W(p(t)) + S(p(t))\theta(t) \end{array}$$

constraints

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

All QP matrices are constructed on line

- LPV models can be obtained from linearization of nonlinear models or from black-box LPV system identification

LINEARIZATION AND TIME-DISCRETIZATION

- Assume model is nonlinear and continuous-time

$$\frac{dx}{dt} = f(x(t), u(t))$$

- Linearize around a nominal state $\bar{x}(t)$ and input $\bar{u}(t)$, such as:

  - an **equilibrium**
  - a **reference** value
  - the **current** value

$$\begin{aligned}\frac{dx}{dt}(t+\tau) \quad &\simeq \quad \left.\frac{\partial f}{\partial x}\right|_{\bar{x}(t),\bar{u}(t)} (x(t+\tau) - \bar{x}(t)) \\ &\quad + \left.\frac{\partial f}{\partial u}\right|_{\bar{x}(t),\bar{u}(t)} (u(t+\tau) - \bar{u}(t)) + f(x(t), u(t))\end{aligned}$$

- Conversion to discrete-time linear prediction model

discrete-time LPV model

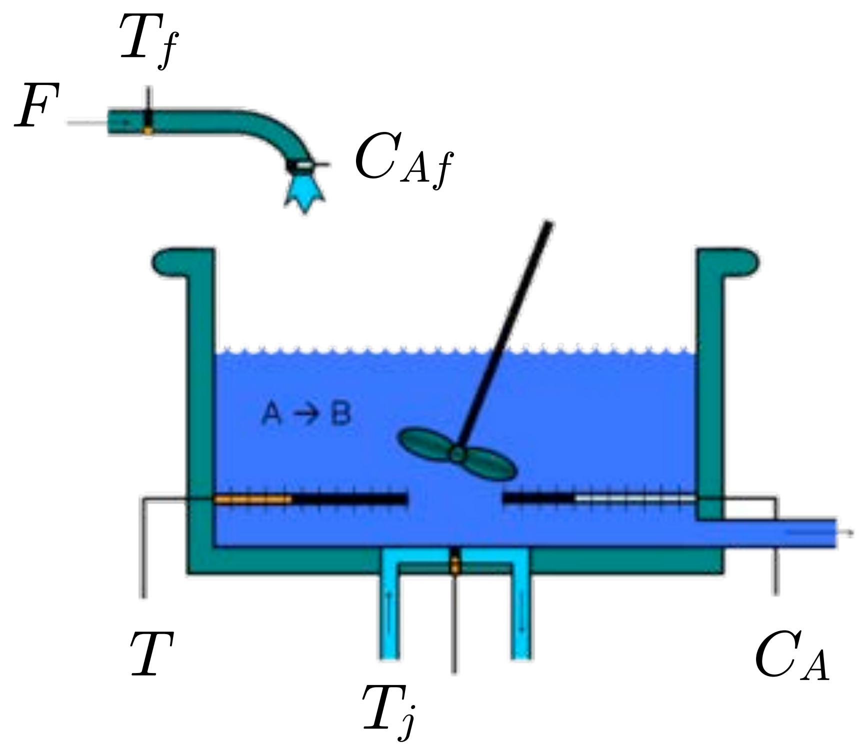
$$x_{k+1} = \underbrace{\left(I + T_s \left.\frac{\partial f}{\partial x}\right|_{\bar{x}(t),\bar{u}(t)}\right)}_{A(t)} x_k + \underbrace{\left(T_s \left.\frac{\partial f}{\partial u}\right|_{\bar{x}(t),\bar{u}(t)}\right)}_{B(t)} u_k + \underbrace{f_k}_{f(t)}$$

model matrices depend on current time t

EXAMPLE: LPV-MPC OF A NONLINEAR CSTR SYSTEM

- MPC control of a diabatic continuous stirred tank reactor (CSTR)

- Process model is nonlinear:



$$
\begin{aligned}
\frac{dC_A}{dt} &= \frac{F}{V}(C_{Af} - C_A) - C_A k_0 e^{-\frac{\Delta E}{RT}} \\
\frac{dT}{dt} &= \frac{F}{V}(T_f - T) + \frac{UA}{\rho C_p V}(T_j - T) - \frac{\Delta H}{\rho C_p} C_A k_0 e^{-\frac{\Delta E}{RT}}
\end{aligned}
$$

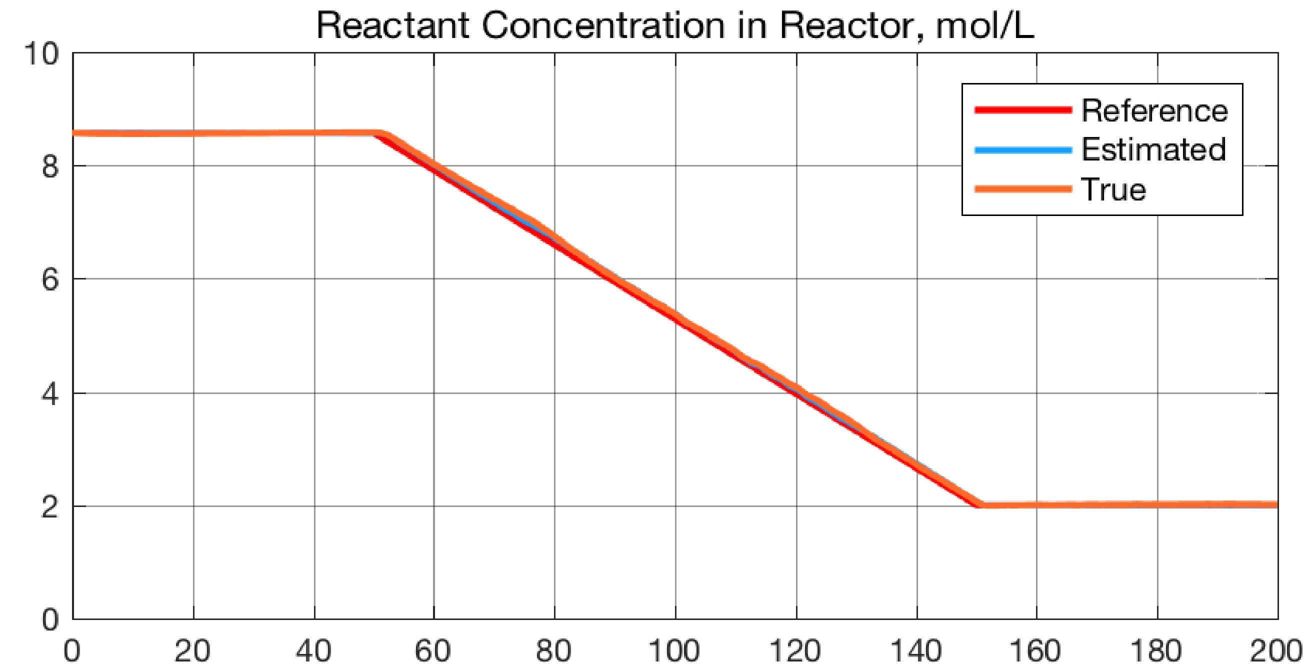
- T : temperature inside the reactor $[K]$ (state)
- C_A : concentration of the reagent in the reactor $[kgmol/m^3]$ (state)
- T_j : jacket temperature $[K]$ (input)
- T_f : feedstream temperature $[K]$ (measured disturbance)
- C_{Af} : feedstream concentration $[kgmol/m^3]$ (measured disturbance)

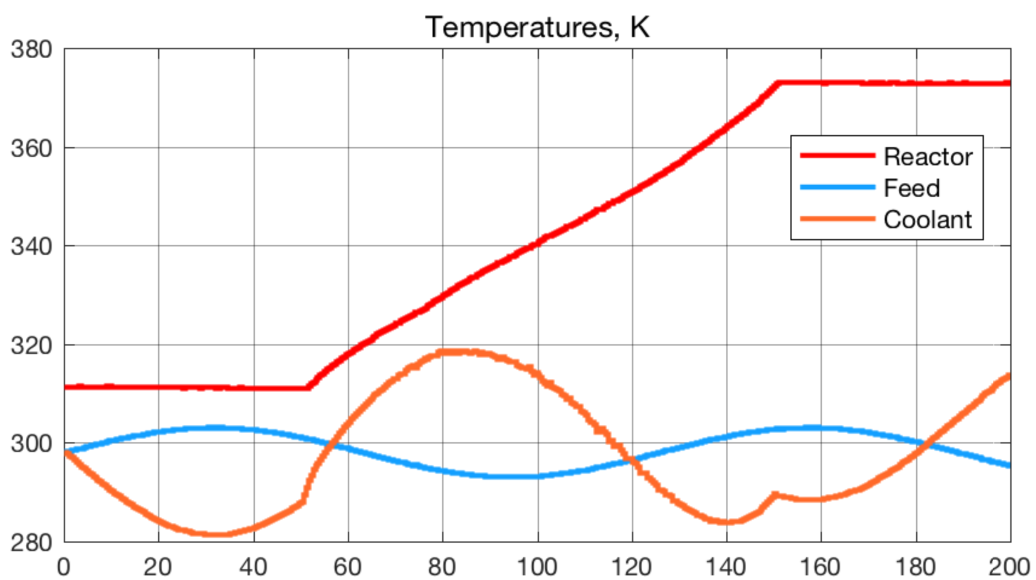
- **Objective:** manipulate T_j to regulate C_A on desired set-point

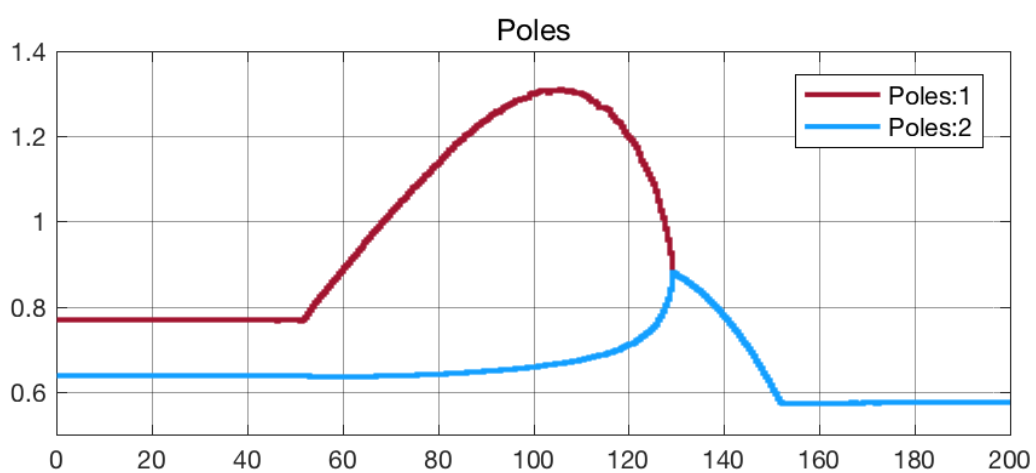
`ampccstr_linearization` (MPC Toolbox)

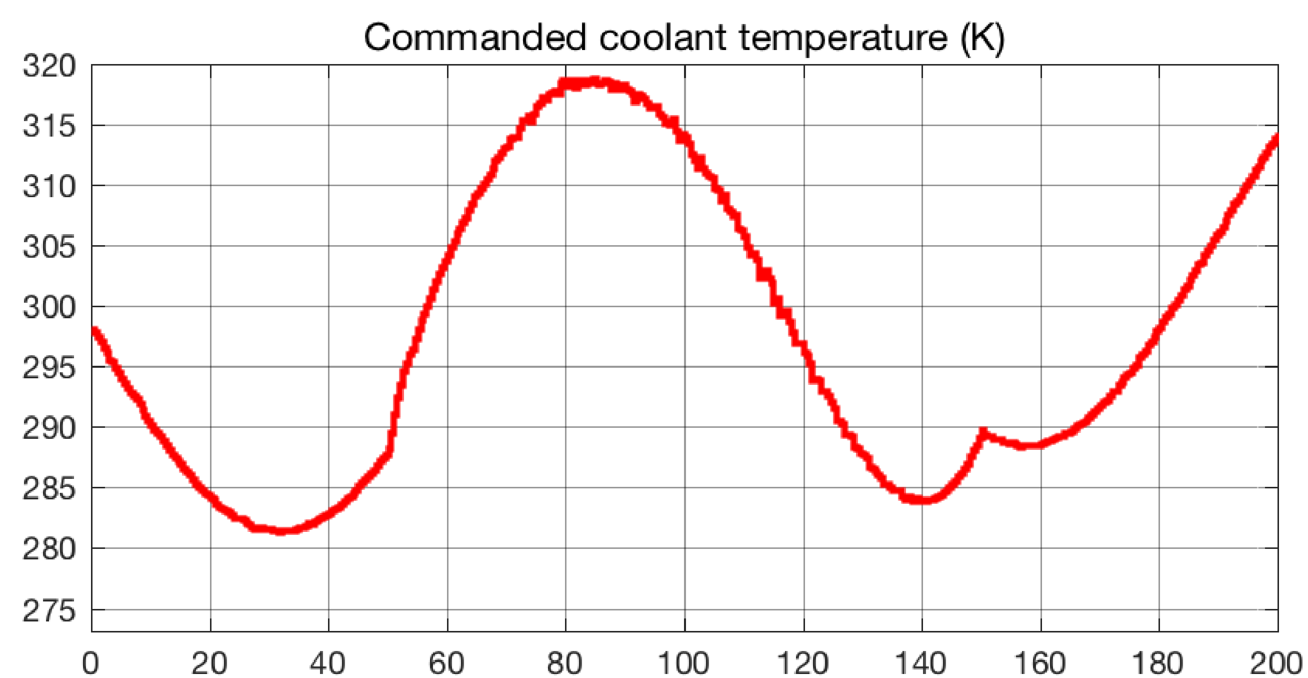
EXAMPLE: LPV-MPC OF A NONLINEAR CSTR SYSTEM

- Closed-loop results



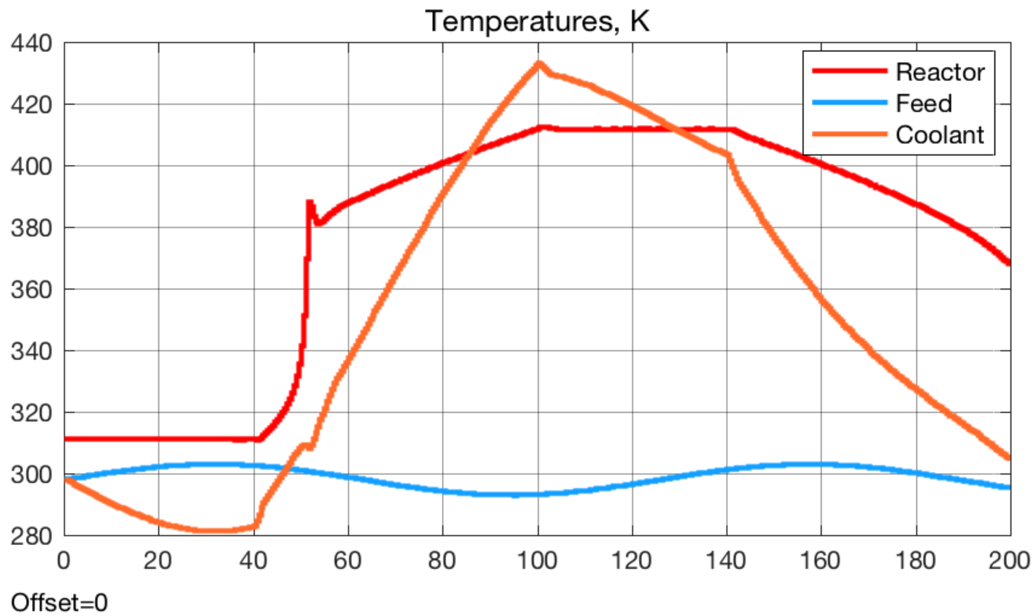


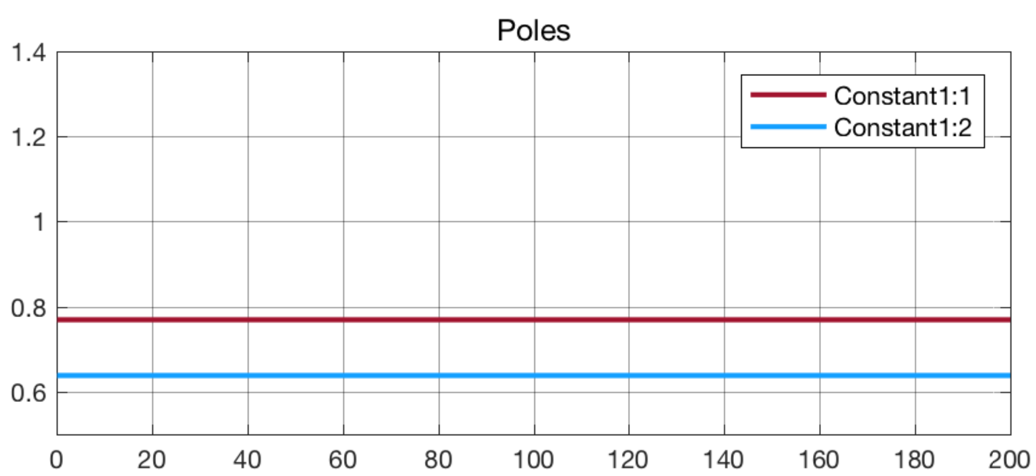


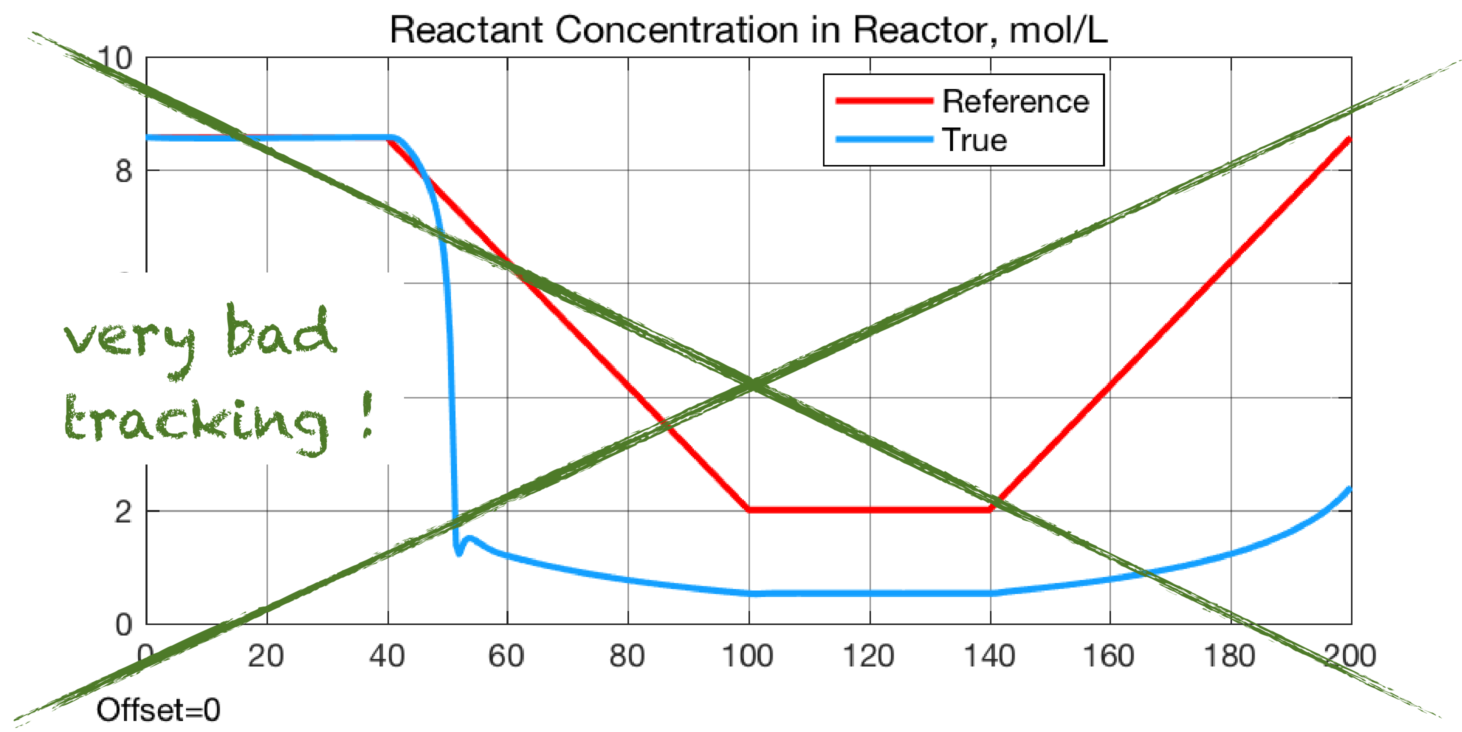


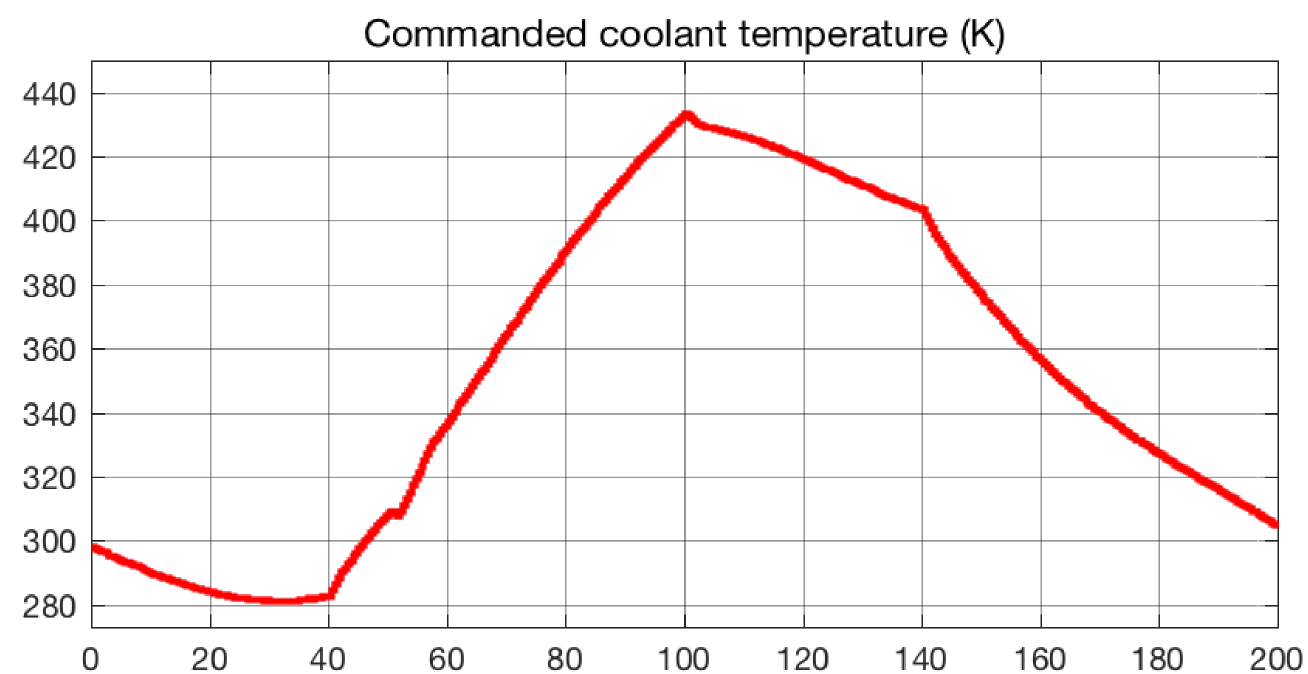
EXAMPLE: LTI-MPC OF A NONLINEAR CSTR SYSTEM

- Closed-loop results





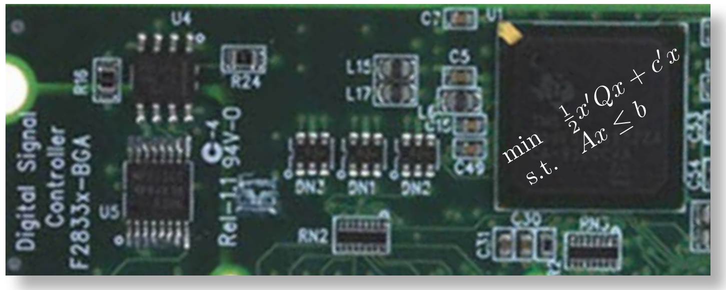




EMBEDDED QP SOLVERS

MPC IN A PRODUCTION ENVIRONMENT

Key **requirements** for optimization-based controllers:



1. **Speed (throughput)**

   a. Execution time must be less than sampling interval

   b. Also fast on average (to free the processor to execute other tasks)

2. Be able to run on **limited hardware** (e.g., 150 MHz) with **little memory**

3. **Robustness** (e.g., with respect to numerical errors)

4. **Worst-case** execution time must be (tightly) **estimated**

5. **Code simple** enough to be validated/verified/certified
   (Library-free C code, easily understandable by production engineers)

EMBEDDED SOLVERS IN INDUSTRIAL PRODUCTION

- Multivariable MPC controller
- Sampling frequency = 40 Hz (= 1 QP solved every 25 ms)
- Vehicle operating ~1 hr/day for ~360 days/year on average
- Controller may be running on 10 million vehicles

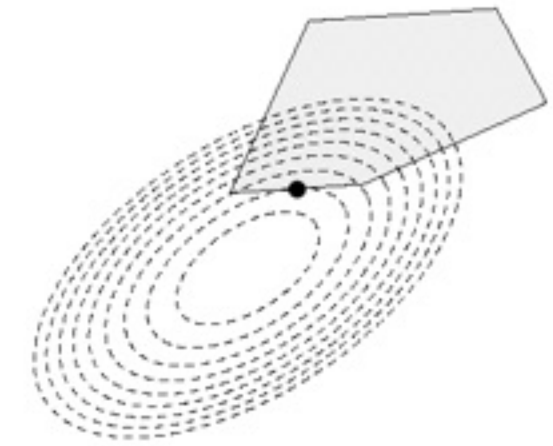
≈ 520,000,000,000,000 QP/yr

and none of them should fail.

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

$$\begin{array}{ll}\min\limits_{z} & \frac{1}{2}z'Hz + x'(t)F'z + \cancel{\frac{1}{2}x'(t)Yx(t)} \\ \text{s.t.} & Gz \leq W + Sx(t)\end{array}$$

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



ON MINIMIZING A CONVEX FUNCTION SUBJECT TO LINEAR INEQUALITIES

By E. M. L. BEALE

Admiralty Research Laboratory, Teddington, Middlesex

SUMMARY

THE minimization of a convex function of variables subject to linear inequalities is discussed briefly in general terms. Dantzig's Simplex Method is extended to yield finite algorithms for minimizing either a convex quadratic function or the sum of the *t* largest of a set of linear functions, and the solution of a generalization of the latter problem is indicated. In the last two sections a form of linear programming with random variables as coefficients is described, and shown to involve the minimization of a convex function.

(Beale, 1955)

A rich set of good QP algorithms is available today

Still a lot of research is going on to address **real-time requirements** ...

SOLUTION METHODS FOR QP

Most used algorithms for solving QP problems:

- ***active set*** methods (small/medium size)
- ***interior point*** methods (large scale)
- ***conjugate gradient*** methods
- ***gradient projection*** methods
- ***alternating direction method of multipliers*** **(ADMM)**
- **...**

$$\begin{array}{ll} \text{min}_z & \frac{1}{2}z'Hz + x'Fz \\ \text{s.t.} & Gz \le W + Sx \end{array}$$

Quadratic Program (QP)

→ Read Dimitri's books for much more on this !

FAST GRADIENT PROJECTION METHOD

(Nesterov, 1983)

- Optimization problem: $\boxed{\min_{z \in Z} f(z)}$ $\quad f : \mathbb{R}^s \to \mathbb{R}$, $\quad Z \subseteq \mathbb{R}^s$

- f convex and ∇f Lipschitz continuous with constant L

$$\|\nabla f(z_1) - \nabla f(z_2)\| \leq L\|z_1 - z_2\|$$

- Accelerated gradient projection iterations:

$$\boxed{\begin{array}{rcl} w_k & = & z_k + \beta_k(z_k - z_{k-1}) \\ z_{k+1} & = & \mathcal{P}_Z\left(w_k - \frac{1}{L}\nabla f(w_k)\right) \end{array}} \qquad \beta_k = \begin{cases} 0 & k = 0 \\ \frac{k-1}{k+2} & k > 0 \end{cases}$$

$$z_{-1} = z_0$$

- Convergence rate: $$f(z_{k+1}) - f^* \leq \frac{2L}{(k+2)^2}\|z_0 - z^*\|^2$$

FAST GRADIENT PROJECTION FOR (DUAL) QP

(Patrinos, Bemporad, IEEE TAC, 2014)

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

$$\begin{array}{ll}\min\limits_{z} & \frac{1}{2}z'Hz + x'F'z \\ \text{s.t.} & Gz \le W + Sx\end{array}$$

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

$$\begin{array}{rcl} M & = & GH^{-1}G' \\ D & = & GH^{-1}F+S \end{array}$$

prepared off-line

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

- Iterations:

$$\begin{array}{rcl} K & = & H^{-1}G' \\ J & = & H^{-1}F' \end{array}$$

$$y_{-1} = y_0 = 0$$

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

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

```
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
```

FAST GRADIENT PROJECTION FOR (DUAL) QP

(Patrinos, Bemporad, IEEE TAC, 2014)

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

feasibility tol

$$s_k^i \leq \frac{1}{L}\epsilon_G, \quad \forall i = 1, \ldots, m$$

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

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

dual function

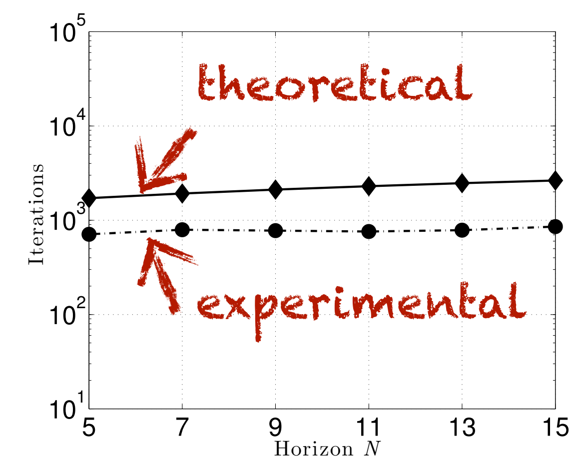
optimality tol

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

- **Convergence rate:**

$$f(z_{k+1}) - f^* \leq \frac{2L}{(k+2)^2}\|z_0 - z^*\|^2$$

- **Tight bounds** on maximum number of iterations



ADMM METHOD FOR QP

(Boyd et al., 2010)

- Alternating Directions Method of Multipliers (ADMM) for QP

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

- Scaled ADMM iterations:

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

(ρy = dual vector)

"integral action"

```
while i<maxiter,
    i=i+1;
    x=-iM*(c+A'*(rho*(u-z)));
    if i<maxiter,
        Ax=A*x;
        z=max(min(Ax+u,ub),lb);
        u=u+Ax-z;
    end
end
```

(9 lines EML code)

(~40 lines of C code)

REGULARIZED ADMM METHOD FOR QP

(Banjac, Stellato, Moehle, Goulart, Bemporad, Boyd, 2017)

- Scaled and regularized ADMM iterations:

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

$Q \succeq 0$

$\epsilon \geq 0$

ρy = dual vector

- Infeasibility detection: $\left\| \dfrac{y_k}{-u' \max\{y_k, 0\} + l' \max\{-y_k, 0\}} \right\|_\infty \leq \epsilon_I$

- Unboundedness detection:

$$v_k = \frac{x_k}{-c' x_k}, \; \|Qv_k\|_\infty \leq \epsilon_U, \; \begin{cases} A^i v_k \leq \epsilon_U \;\&\; u^i < +\infty \\ A^i v_k \geq -\epsilon_U \;\&\; \ell^i > -\infty \end{cases}$$

- **Simple, fast, robust.** Only needs to factorize $\begin{bmatrix} Q+\epsilon I & A' \\ A & -\gamma I \end{bmatrix}$ once

osQP solver https://github.com/oxfordcontrol/osqp

SCALING (OR PRECONDITIONING)

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

(Giselsson, Boyd, 2015)

primal QP

$$\begin{array}{ll} \min_z & \frac{1}{2}z'Hz + f'z \\ \text{s.t.} & Gz \leq W \end{array}$$

$$\begin{array}{rcl} M &=& GH^{-1}G' \\ d &=& GH^{-1}f + W \end{array}$$

- **Dual scaling** (Jacobi scaling): (Bertsekas, 2009)

$$\begin{array}{ll} \min_y & \frac{1}{2}y'My + d'y \\ \text{s.t.} & y \geq 0 \end{array}$$

dual QP

$$\xrightarrow{\text{scaling } y = Py_s}$$

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

$$\begin{array}{ll} \min_{y_s} & \frac{1}{2}y_s'(PMP)y_s + d'Py_s \\ \text{s.t.} & y_s \geq 0 \end{array}$$

scaled dual QP

- Equivalent to just **scale constraints** in primal problem: $\frac{1}{\sqrt{M_{ii}}}G_i z \leq \frac{1}{\sqrt{M_{ii}}}W_i$

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

CAN WE SOLVE QP'S USING LEAST SQUARES ?

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

(Legendre, 1805)

(Gauss, <= 1809)

$$v = \arg\min \|Av - b\|_2^2$$

In MATLAB: >> `v=A\b % (1 character ! )`

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

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

ACTIVE-SET METHOD FOR NONNEGATIVE LEAST SQUARES

(Lawson, Hanson, 1974)

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

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:
While maintaining the primal var v feasible, keep switching the active set until the dual var w is also feasible

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

SOLVING QP'S VIA NONNEGATIVE LEAST SQUARES

(Bemporad, 2016)

- Use NNLS to solve strictly convex QP

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

QP

$$u \triangleq Lz + L^{-T}c$$

complete the squares

$$Q = L'L$$

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

Least Distance Problem

$$M = GL^{-1}$$
$$d = b + GQ^{-1}c$$

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

Nonnegative Least Squares

residual $= 0$?

yes

QP problem infeasible

no

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

retrieve primal solution

- Fast and relatively simple active-set QP solver. But not very robust ...

SOLVING QP VIA NNLS: ROBUST ALGORITHM

(Bemporad, 2017)

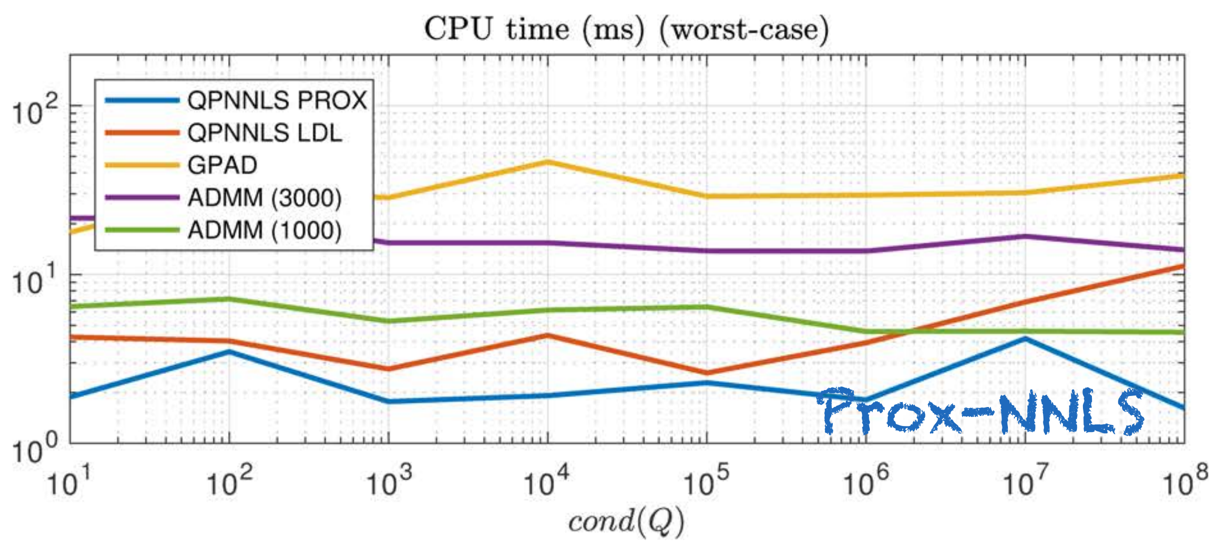
- **Key idea:** solve a sequence of regularized QP problems

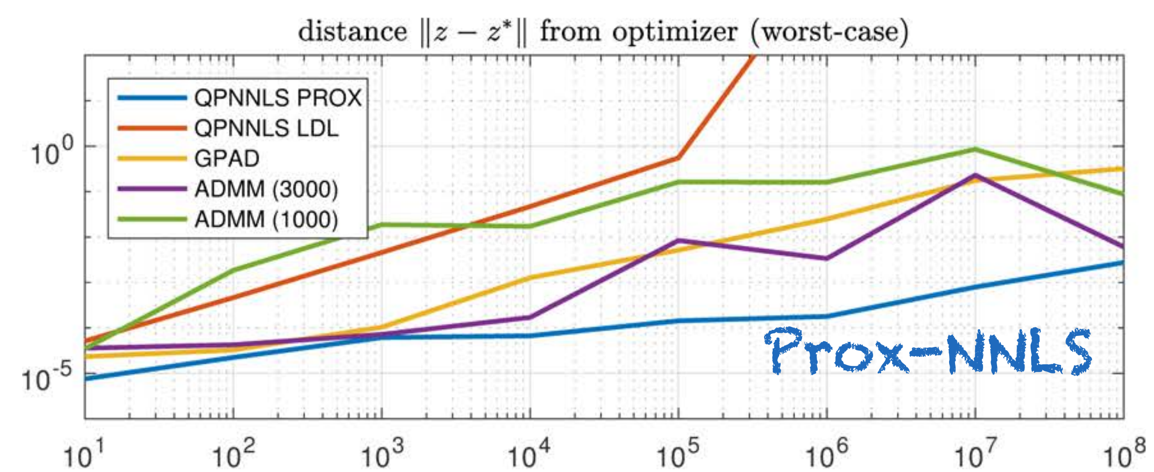
$$
\begin{aligned}
z_{k+1} = \arg\ \min_z\ & \tfrac{1}{2}z'Qz + c'z + \tfrac{\epsilon}{2}\|z - z_k\|_2^2 \\
\text{s.t.}\ & Az \leq b \\
& Gx = g
\end{aligned}
$$

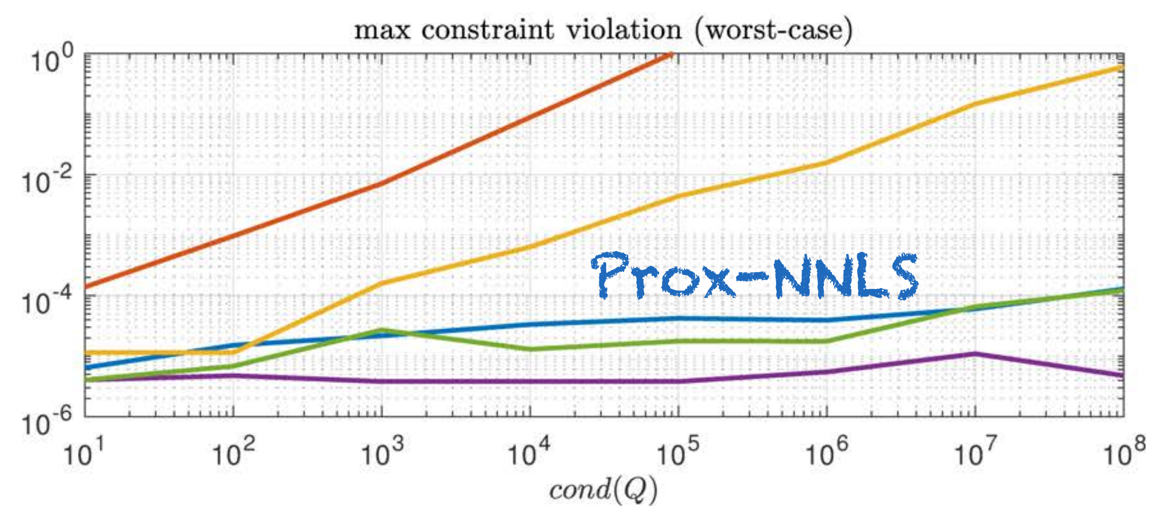
proximal-point algorithm, iterations converge to the optimal solution

- Main advantage: primal Hessian $Q + \epsilon I$ can be arbitrarily well conditioned !

(tradeoff robustness/CPU time)

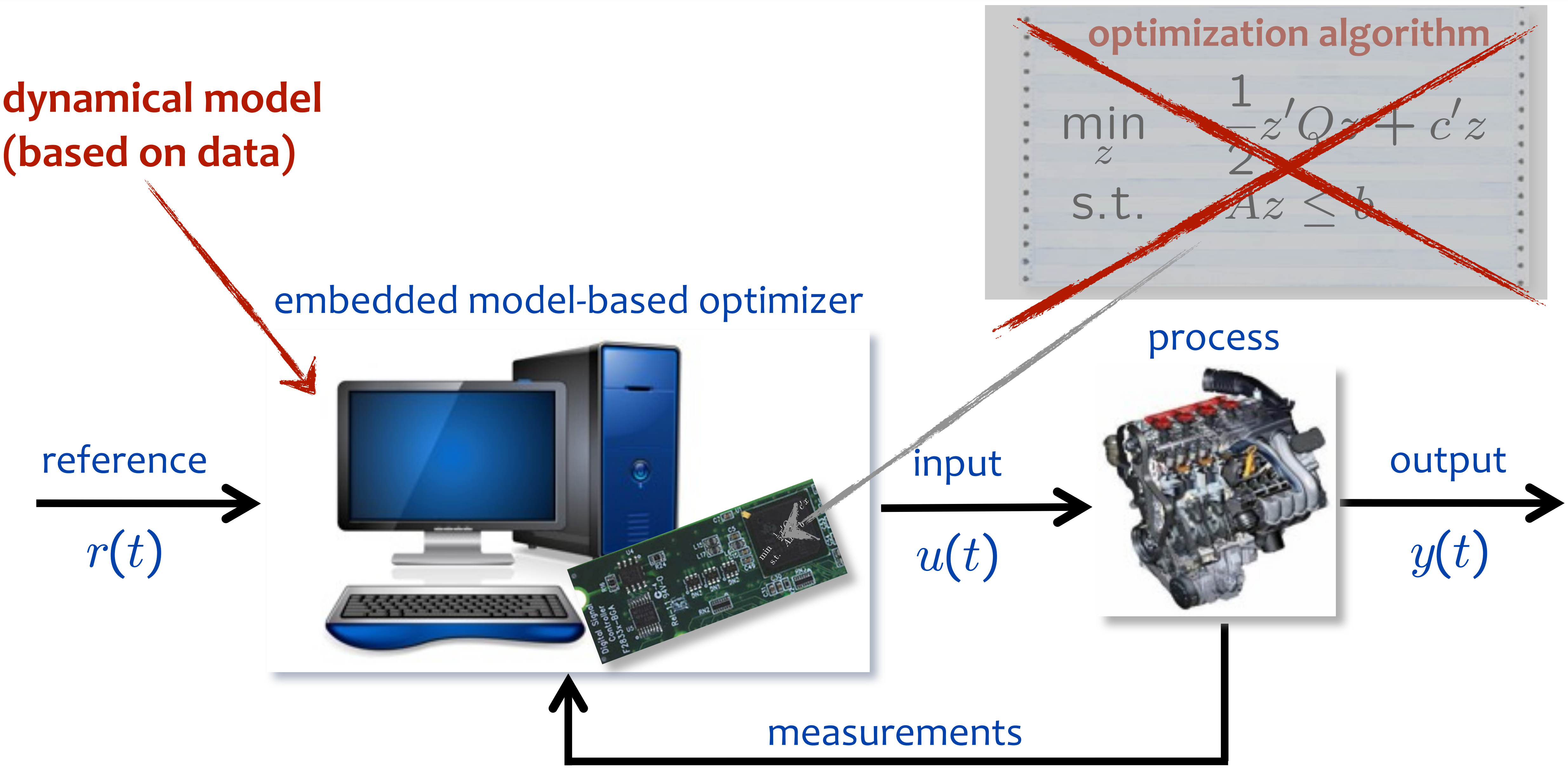


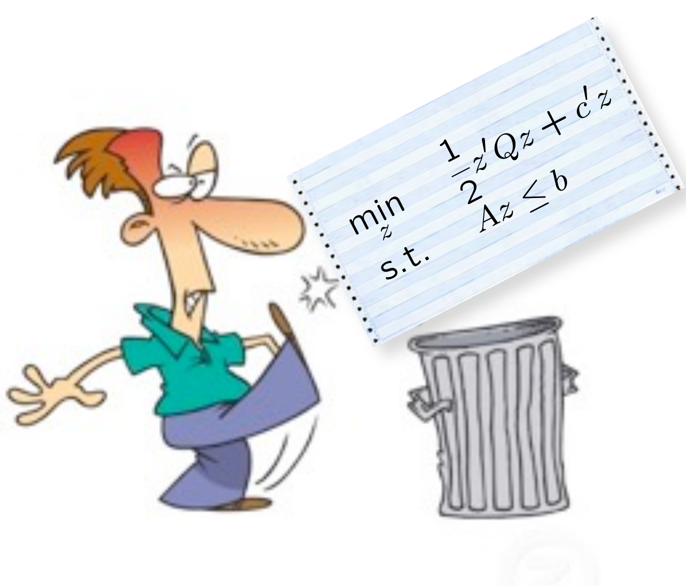




single precision arithmetic, random QPs
30 vars, 100 constraints (this Mac)

EMBEDDED MPC WITHOUT SOLVING QP'S ON LINE





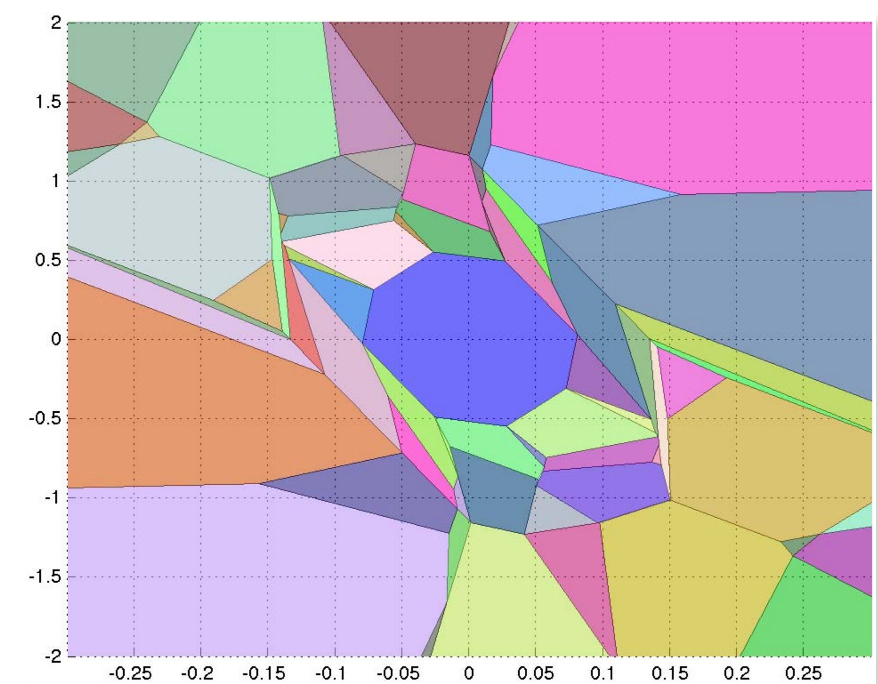
- Can we implement MPC **without** an embedded **optimization solver** ?

YES !

(Bemporad, Morari, Dua, Pistikopoulos, 2002)

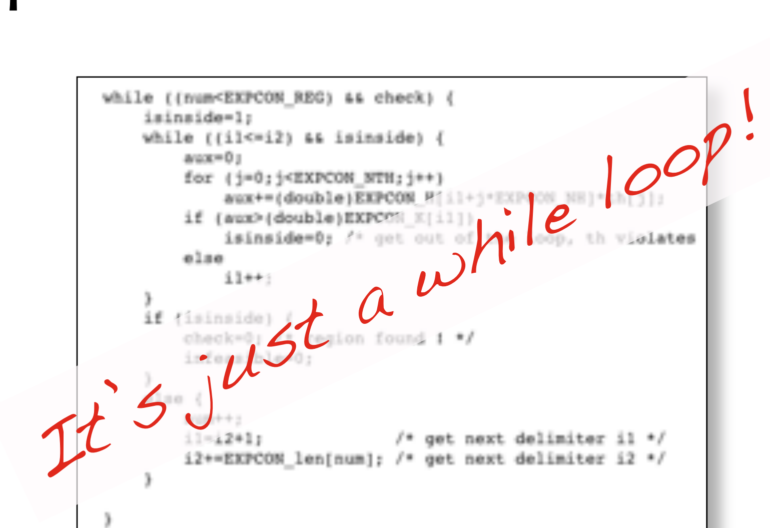
The multiparametric solution of a strictly convex QP is **continuous** and **piecewise affine**

$$
\begin{aligned}
z^*(x) = \arg\min_z &\quad \tfrac{1}{2} z' H z + x' F' z \\
\text{s.t.} &\quad G z \le W + S x
\end{aligned}
$$



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

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



MULTIPARAMETRIC QUADRATIC PROGRAMMING

- A variety of mpQP solvers is available (Bemporad *et al.*, 2002) (Baotic, 2002) (Tøndel, Johansen, Bemporad, 2003) (Jones, Morari, 2006) (Spjøtvold *et al.*, 2006) (Patrinos, Sarimveis, 2010)

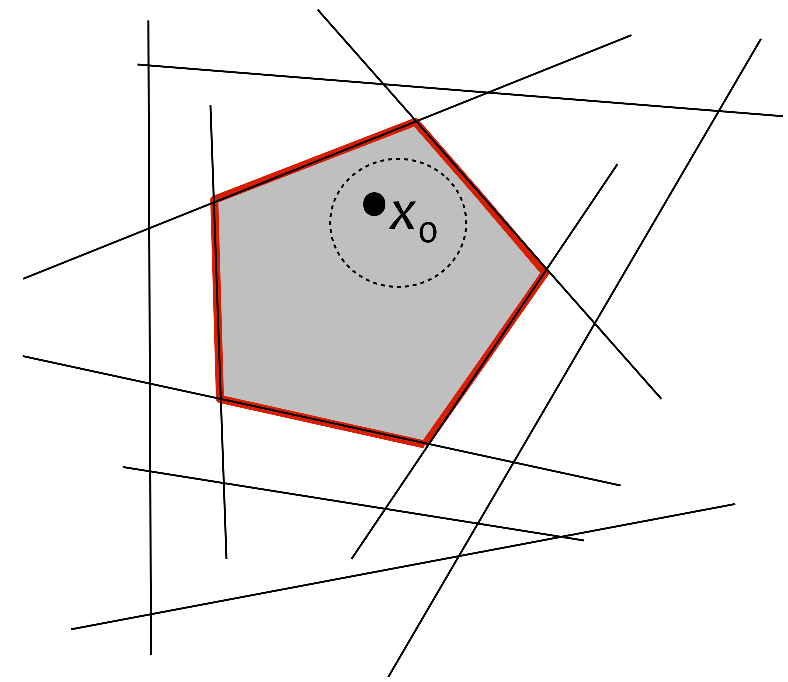
- Most computations are spent in **operations on polyhedra** (=critical regions)

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

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)**

- Can be also performed via **nonnegative least squares (NNLS)** (Bemporad, 2015)

NNLS FOR SOLVING MPQP PROBLEMS

- Comparison of mpQP solvers
  - **Hybrid Toolbox** (Bemporad, 2003)
  - **Multiparametric Toolbox 2.6** (with default opts) (Kvasnica, Grieder, Baotic, 2004) (Herceg, Kvasnica, Jones, Morari, 2013)
  - **NNLS - MPC Toolbox** (≥R2014b)

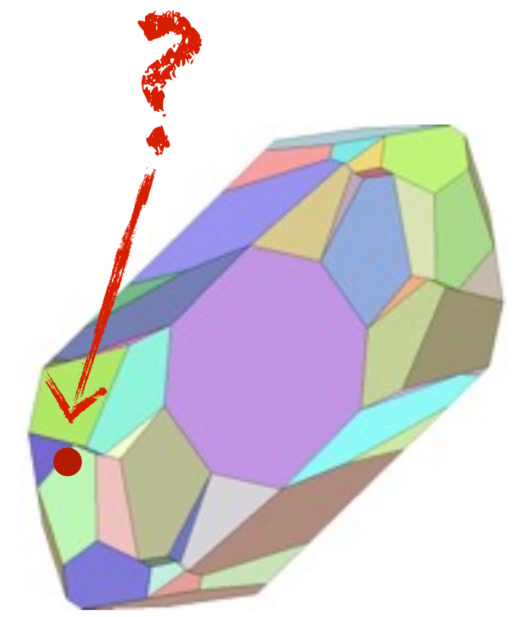
The MathWorks (Bemporad, Morari, Ricker, 1998-present)

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 MULTIPARAMETRIC SOLUTIONS

- The number of regions depends (exponentially) on the number of possible **combinations of active constraints**

- Explicit MPC gets less attractive when number of regions grows: too much **memory** required, too much **time** to locate state $x(t)$



- Fast **on-line** QP solvers (=implicit MPC) may be preferable

When is implicit preferable to explicit MPC ?

COMPLEXITY CERTIFICATION FOR ACTIVE SET QP SOLVERS

(Cimini, Bemporad, IEEE TAC, 2017)

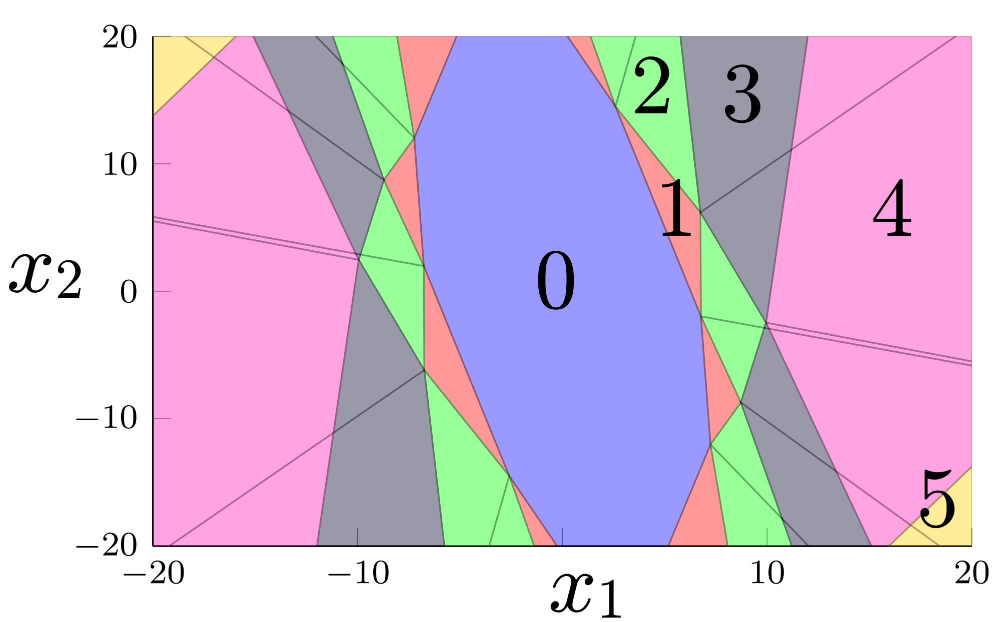
- Consider a dual active-set QP solver
  (Goldfarb, Idnani, 1983)

$$\begin{array}{rl} z^*(x) = \arg\min_z & \frac{1}{2} z' H z + x' F' z \\ \text{s.t.} & Gz \leq W + Sx \end{array}$$

- What is the worst-case number of iterations over x to solve the QP ?

- **Key result:**

The number of iterations to solve the QP is a piecewise constant function of the parameter x



We can **exactly** quantify how many iterations (flops) the QP solver takes in the worst-case !

COMPLEXITY CERTIFICATION FOR ACTIVE SET QP SOLVERS

- Examples (from MPC Toolbox):

	inv. pend.	DC motor	nonlin. demo	AFTI 16
# vars	5	3	6	5
# constraints	10	10	18	12
# params	9	6	10	10
Explicit MPC				
# regions	87	67	215	417
max flops	3382	1689	9184	16434
max memory (kb)	55	30	297	430
Implicit MPC				
max iters	11	9	13	16
max flops	3809	2082	7747	7807
sqrt	27	9	37	33
max memory (kb)	15	13	20	16

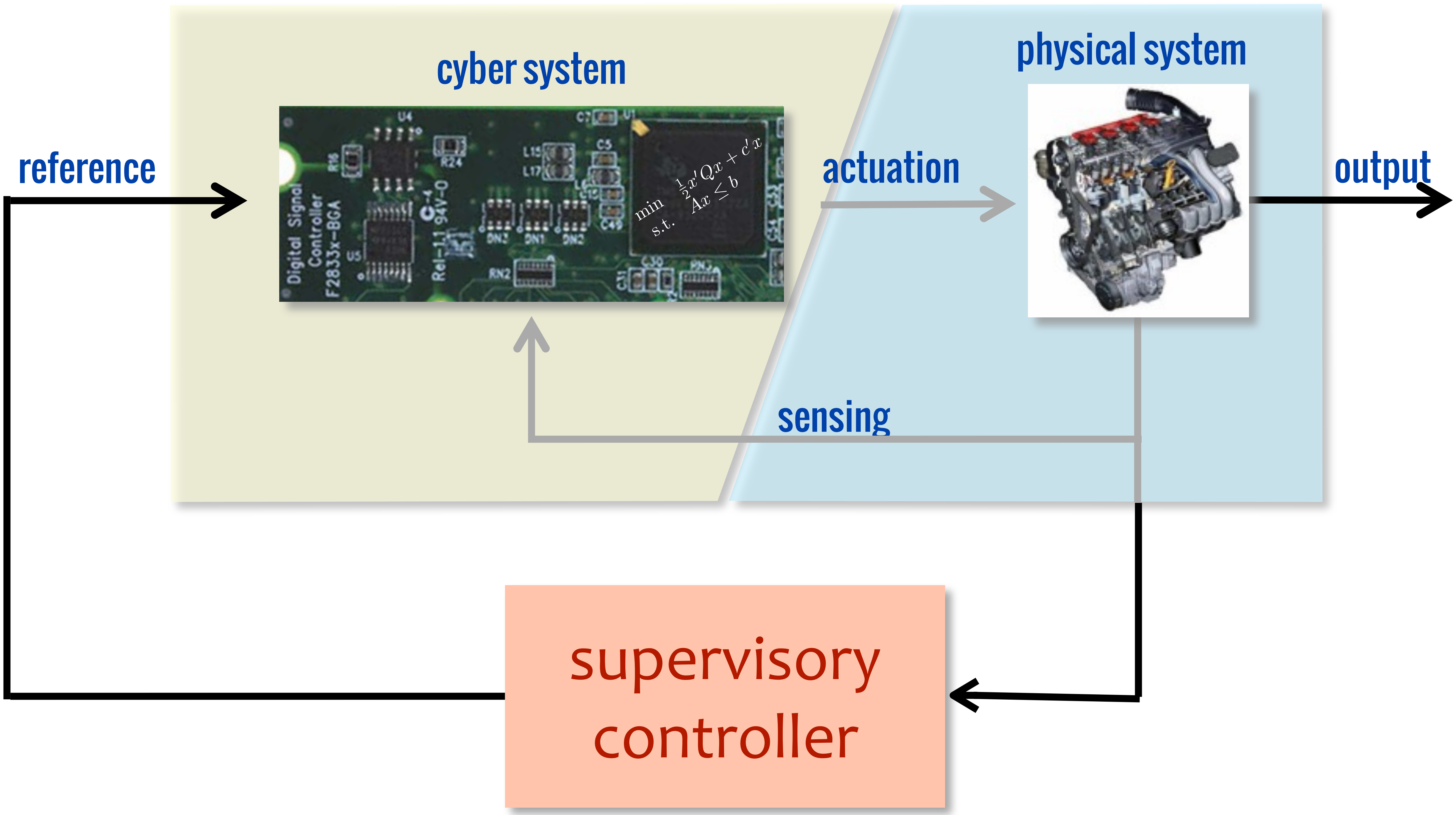
explicit MPC is faster in the worst-case

online QP is faster in the worst-case

- It is possible to combine explicit and on-line QP for best tradeoff

HYBRID MPC OF CYBER-PHYSICAL SYSTEMS

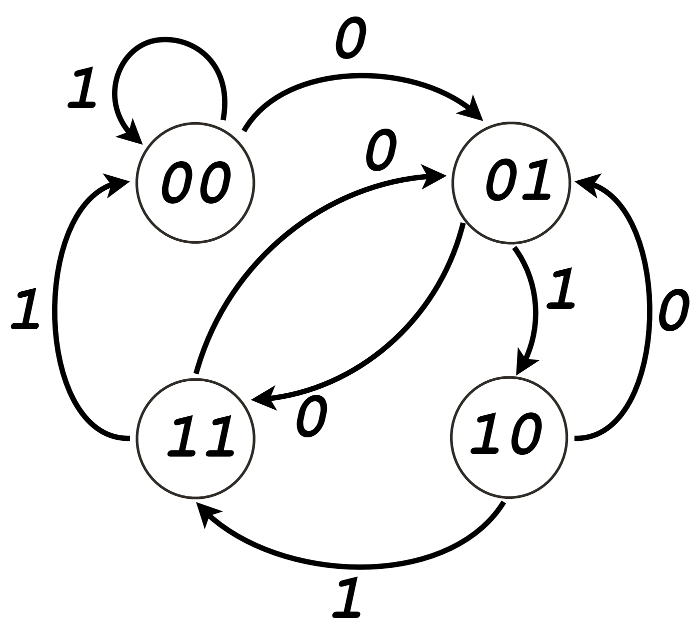
CONTROL OF CYBER-PHYSICAL SYSTEMS



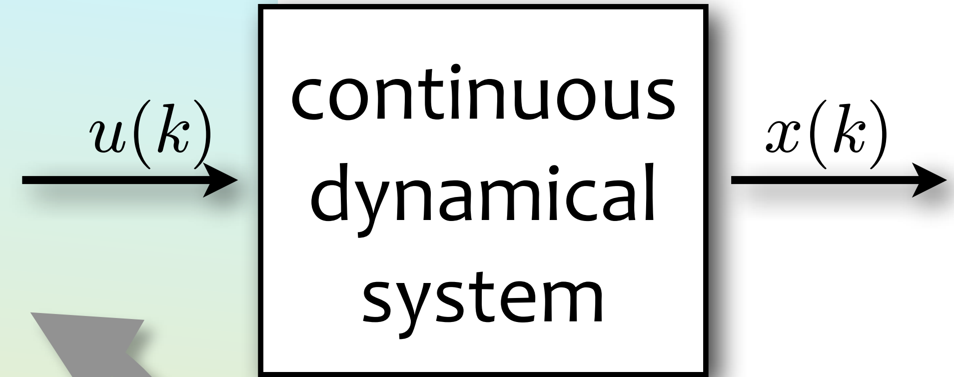
What is a good model of a CPS for supervisory control purposes ?

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**

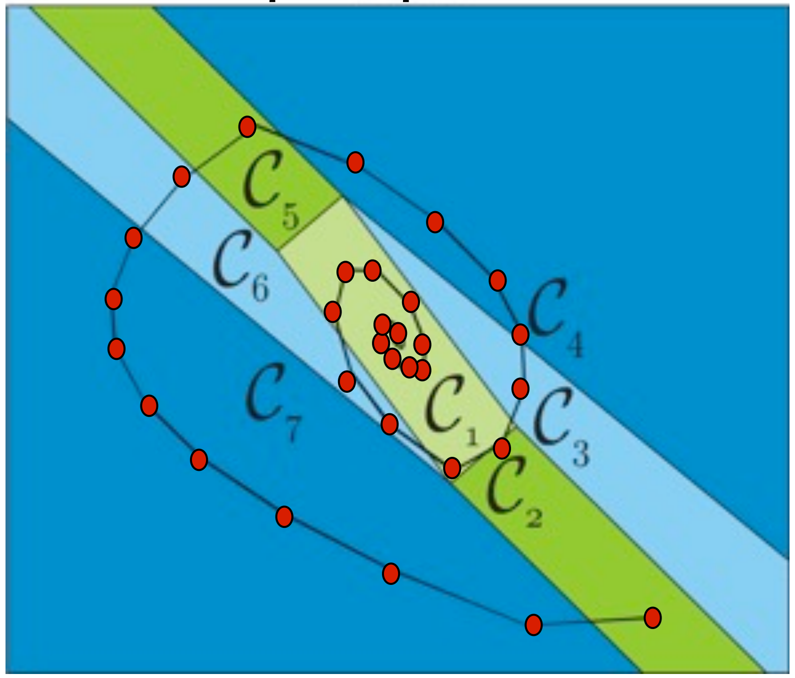
  $x \in \mathbb{R}^{n_c}, \;\; u \in \mathbb{R}^{m_c}$

- **Difference/differential equations**
- **Linear inequality** constraints

PIECEWISE AFFINE SYSTEMS

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

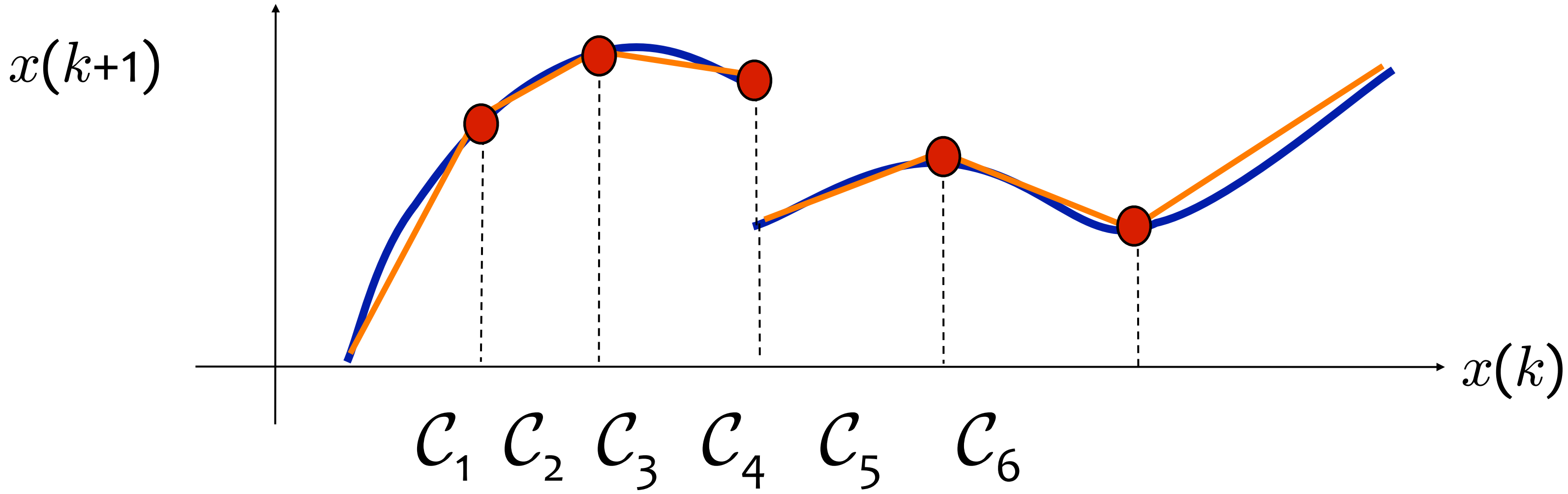
state+input space



(Sontag 1981)

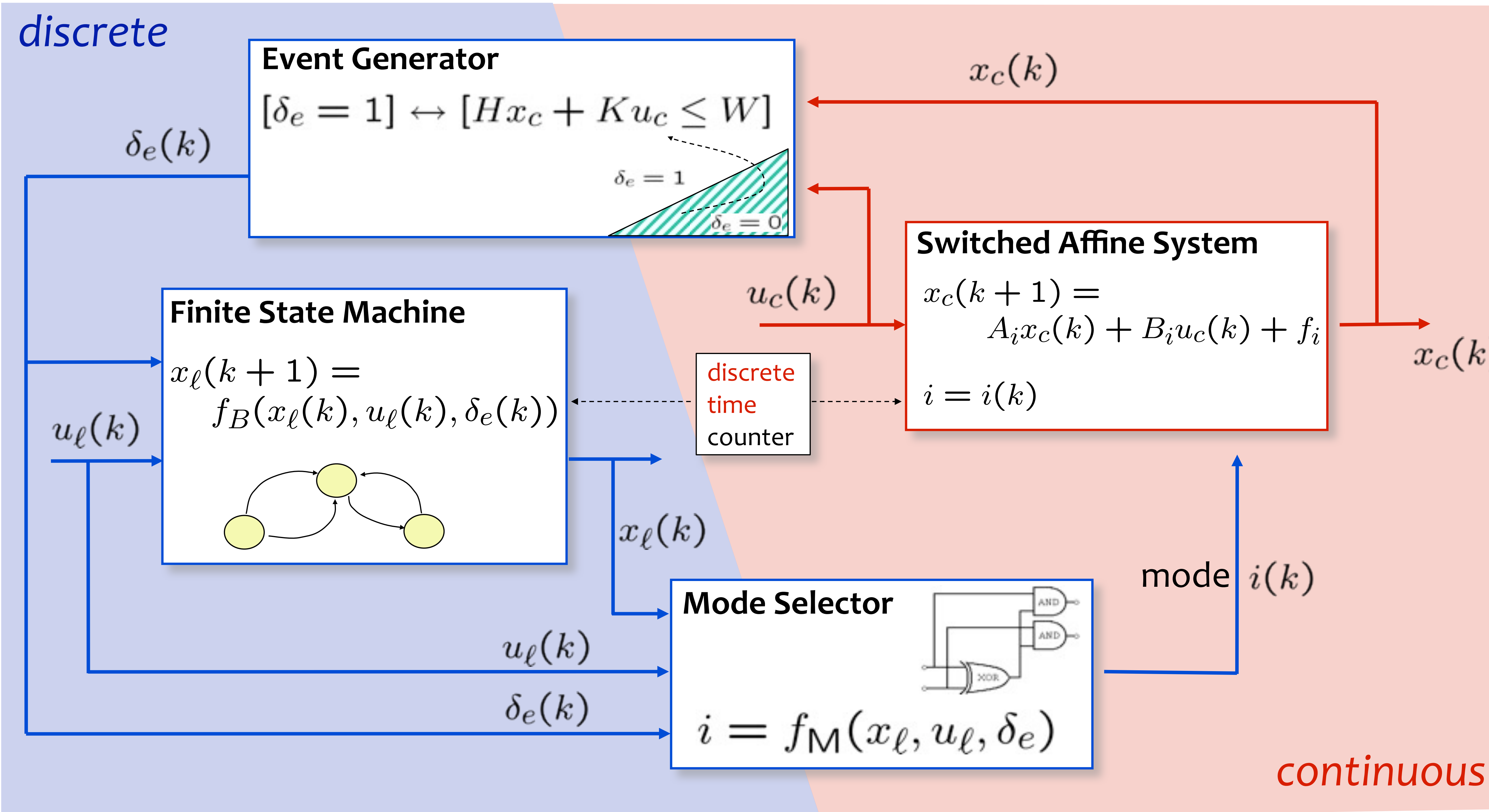
$x \in \mathbb{R}^n, \ u \in \mathbb{R}^m, \ y \in \mathbb{R}^p$
$i(k) \in \{1, \ldots, s\}$

Can approximate nonlinear and/or discontinuous dynamics arbitrarily well



DISCRETE HYBRID AUTOMATON (DHA)

(Torrisi, Bemporad, 2004)



$$x_\ell \in \{0,1\}^{n_\ell} = \text{binary state}$$
$$u_\ell \in \{0,1\}^{m_\ell} = \text{binary input}$$
$$\delta_e \in \{0,1\}^{n_e} = \text{event variable}$$

$$x_c \in \mathbb{R}^{n_c} = \text{real-valued state}$$
$$u_c \in \mathbb{R}^{m_c} = \text{real-valued input}$$
$$i \in \{1,\dots,s\} = \text{current mode}$$

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

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

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

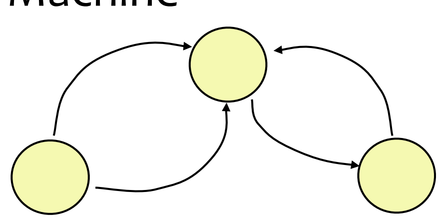
$$\begin{cases} 1 \leq \sum_{i \in P_1} \delta_i + \sum_{i \in N_1} (1 - \delta_i) \\ \vdots \\ 1 \leq \sum_{i \in P_m} \delta_i + \sum_{i \in N_m} (1 - \delta_i) \end{cases}$$

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

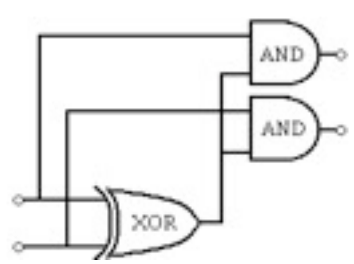
IF $[\delta = 1]$ THEN $z = a_1^T x + b_1^T u + f_1$
ELSE $z = a_2^T x + b_2^T u + f_2$

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

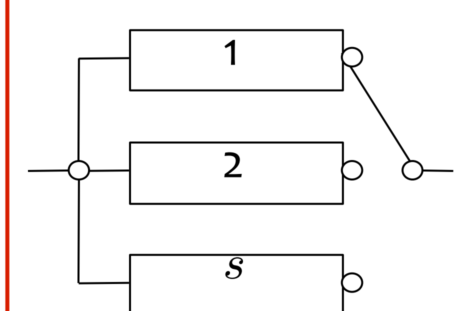
Finite State Machine



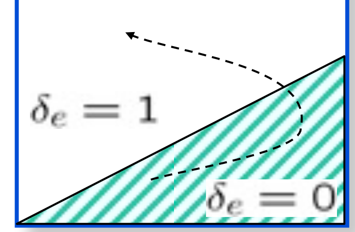
Mode Selector



Switched Affine System



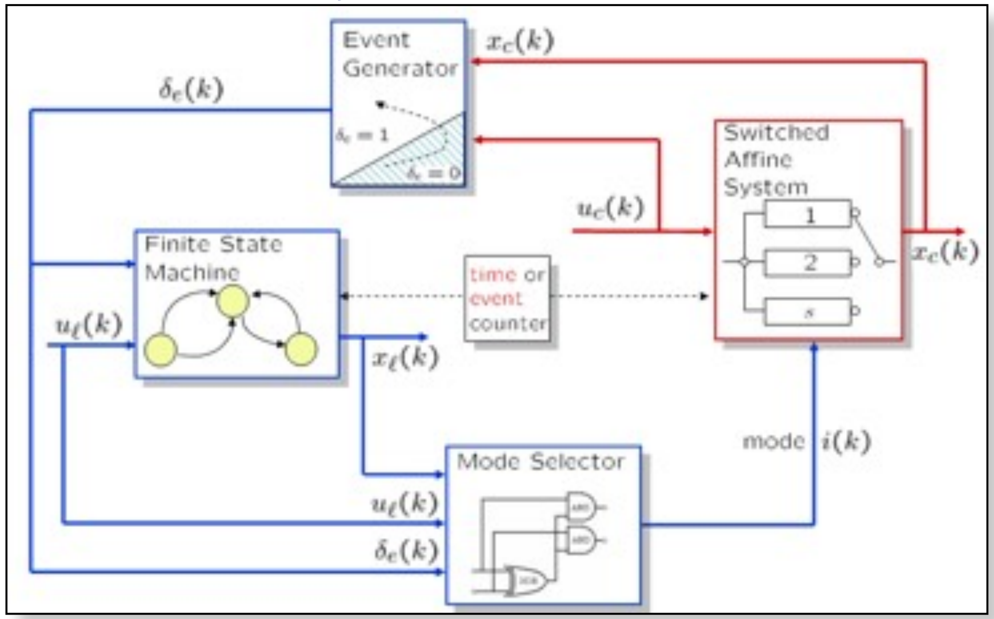
Event Generator



MIXED LOGICAL DYNAMICAL (MLD) SYSTEMS

(Bemporad, Morari 1999)

Discrete Hybrid Automaton



HYSDEL (Torrisi, Bemporad, 2004)

convert logic propositions to mixed-integer linear inequalities

Mixed Logical Dynamical (MLD) systems

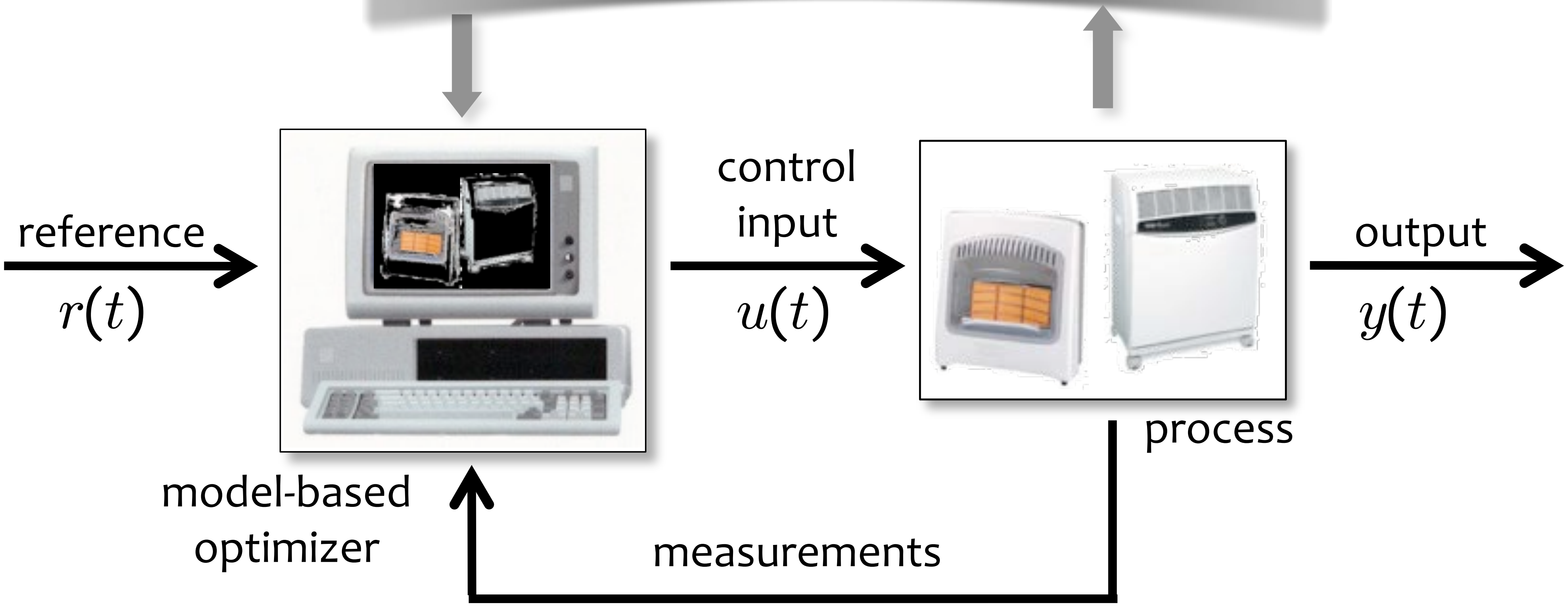
$$\begin{cases} x_{k+1} = Ax_k + B_1 u_k + B_2 \delta_k + B_3 z_k + B_5 \\ y_k = Cx_k + D_1 u_k + D_2 \delta_k + D_3 z_k + D_5 \\ E_2 \delta_k + E_3 z_k \leq E_4 x_k + E_1 u_k + E_5 \end{cases}$$

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

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



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

MIQP FORMULATION OF MPC

(Bemporad, Morari, 1999)

$$
\begin{array}{ll}
\min_{\xi} & \sum_{k=0}^{N-1} y_k' Q y_k + u_k' R u_k \\
\text{s.t.} & \left\{ \begin{array}{rcl} x_{k+1} & = & A x_k + B_1 u_k + B_2 \delta_k + B_3 z_k + B_5 \\ y_k & = & C x_k + D_1 u_k + D_2 \delta_k + D_3 z_k + D_5 \\ E_2 \delta_k + E_3 z_k & \leq & E_4 x_k + E_1 u_k + E_5 \end{array} \right.
\end{array}
$$

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

$$
\begin{array}{ll}
\min_{\xi} & \frac{1}{2}\xi' H \xi + x'(t) F \xi + \cancel{\frac{1}{2} x'(t) Y x(t)} \\
\text{s.t.} & G\xi \leq W + S x(t)
\end{array}
$$

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 $t \geq 0$.

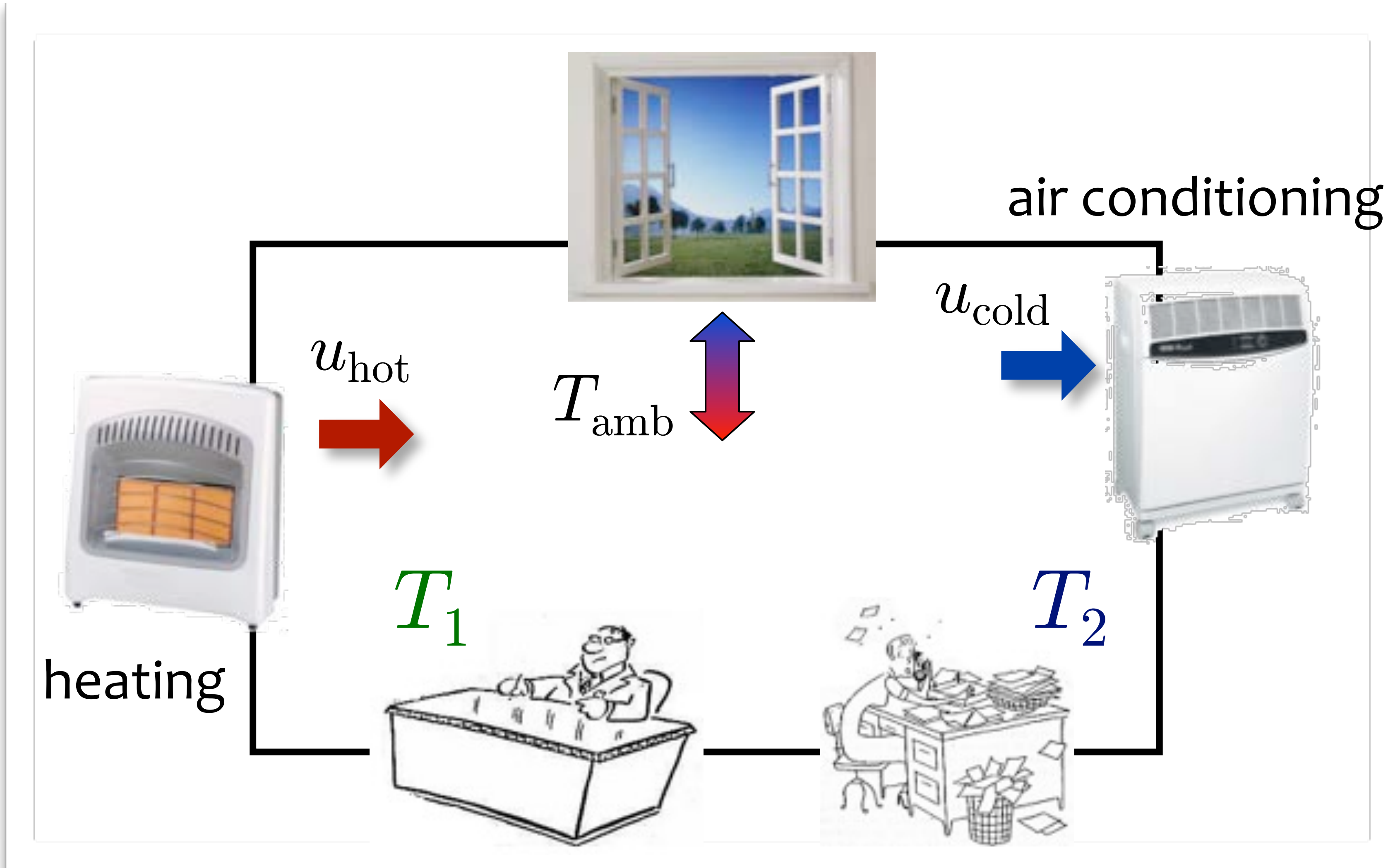
(Bemporad, Morari 1999)

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



discrete dynamics

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

continuous dynamics

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

$$i = 1, 2$$

Hybrid Toolbox for MATLAB, /`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

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];
```

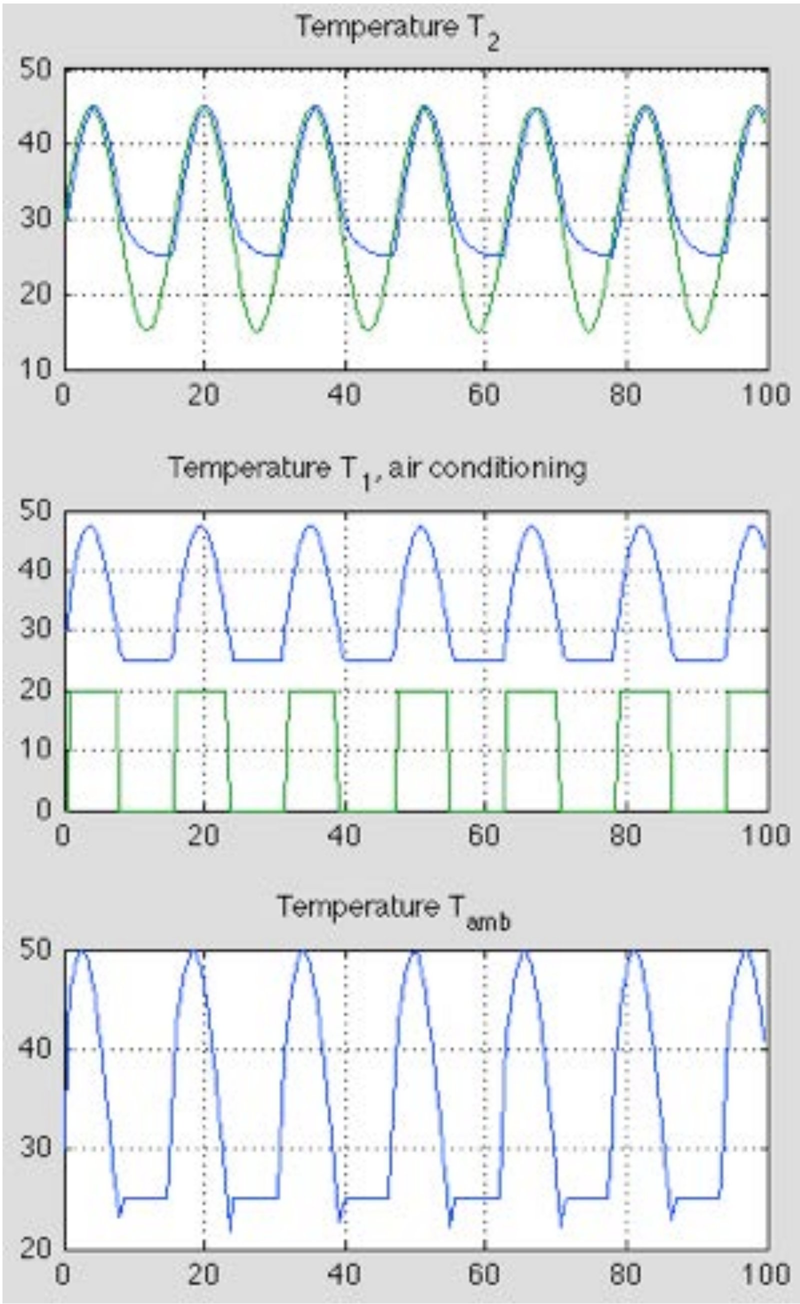
$$\min \sum_{k=0}^{2} \|x_{2k} - r(t)\|_\infty$$

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

```
>>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);
```



MIXED-INTEGER PROGRAM (MIP) SOLVERS

- Mixed-Integer Programming is *NP*-complete

BUT

- General purpose branch & bound / branch & cut solvers available for MILP and MIQP (CPLEX, GLPK, Xpress-MP, CBC, Gurobi, ...)

More solvers and benchmarks: http://plato.la.asu.edu/bench.html

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

BRANCH & BOUND FOR MIQP (USING NNLS SOLVER FOR QP)

(Bemporad, NMPC, 2015)

- Consider a MIQP problem of the form

$$\begin{array}{ll} \min_z & V(z) \triangleq \frac{1}{2} z' Q z + c' z \\ \text{s.t.} & \ell \leq Az \leq u \\ & Gz = g \\ & \bar{A}_i z \in \{\bar{\ell}_i, \bar{u}_i\}, \; i = 1, \dots, q \end{array}$$

$$Q = Q' \succ 0$$

- Binary constraints on z are a special case:

$$\bar{\ell}_i = 0, \; \bar{u}_i = 1, \; \bar{A}_i = [0 \dots 0 \;\, 1 \;\, 0 \dots 0]$$

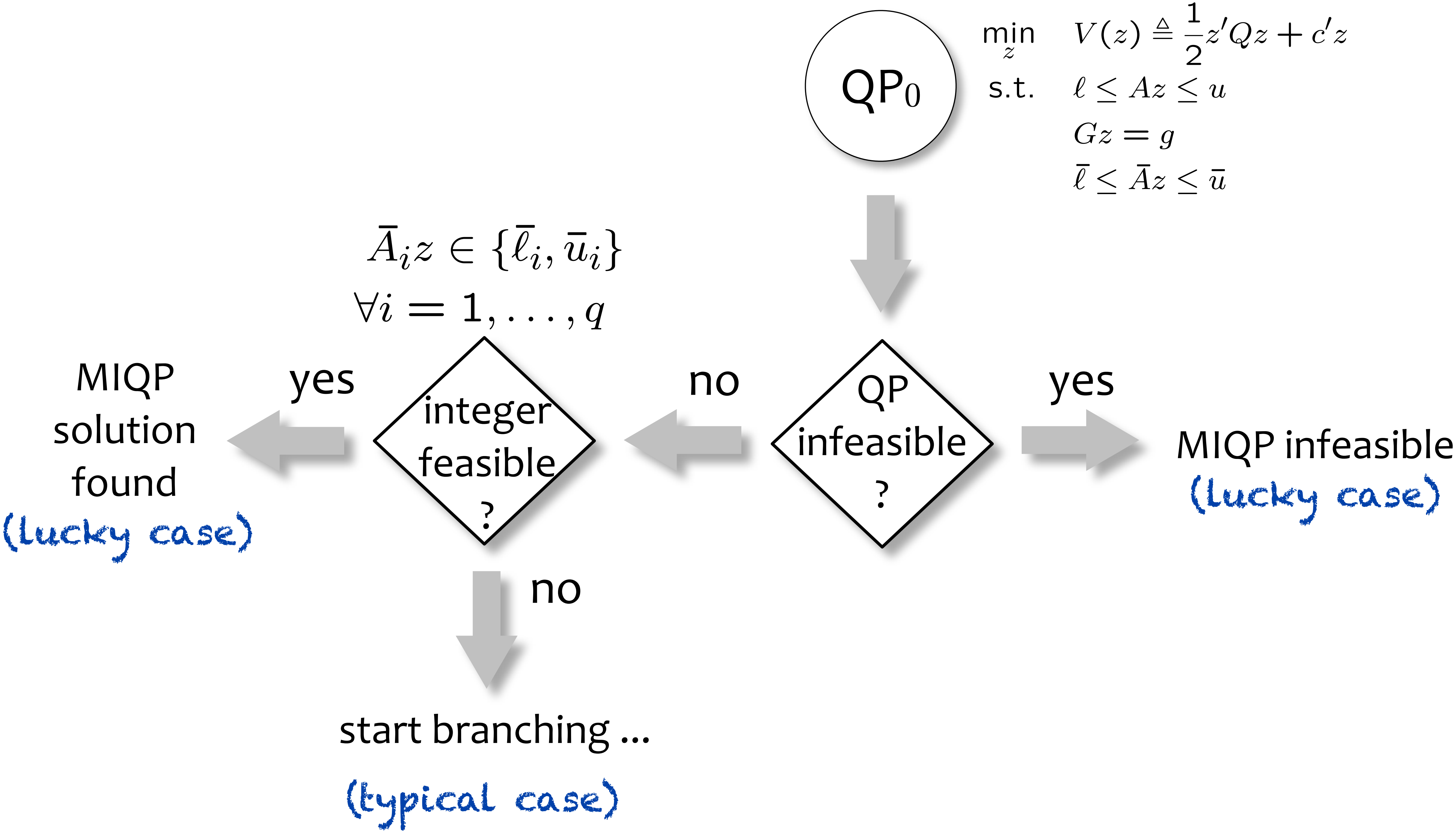
- QP algorithm based on NNLS is used to solve MIQP relaxations $\bar{\ell}_i \leq \bar{A}_i z \leq \bar{u}_i$

BRANCH & BOUND FOR MIQP (USING NNLS SOLVER FOR QP)

- **Branch and bound** scheme:

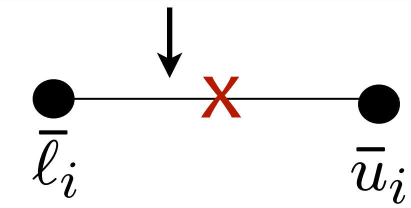
QP relaxation

$$\begin{aligned} \min_z \quad & V(z) \triangleq \frac{1}{2} z' Q z + c' z \\ \text{s.t.} \quad & \ell \leq A z \leq u \\ & G z = g \\ & \bar{\ell} \leq \bar{A} z \leq \bar{u} \end{aligned}$$

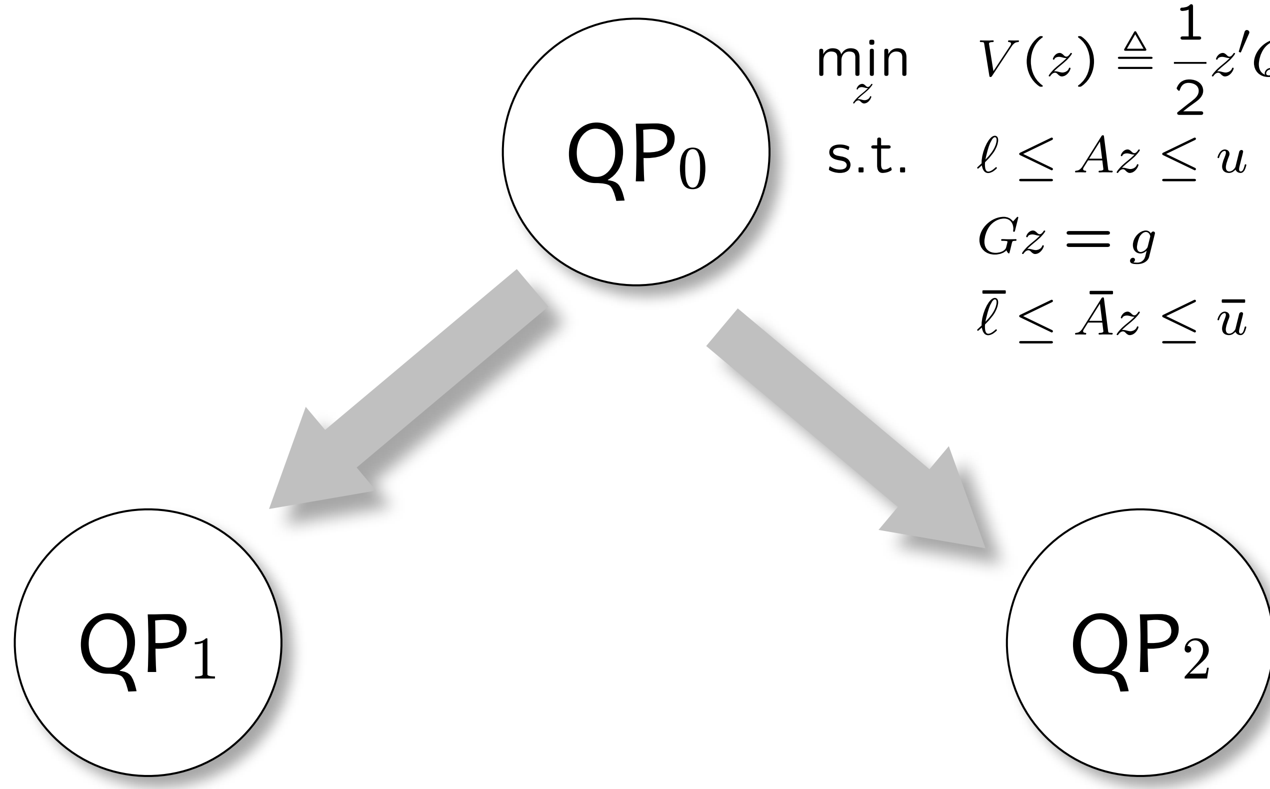


BRANCH & BOUND FOR MIQP (USING NNLS SOLVER FOR QP)

- **Branching:** pick up index i such that $\bar{A}_i z$ is closest to $\frac{\bar{\ell}_i + \bar{u}_i}{2}$



- Solve two new QP problems:



QP$_0$:

$$\begin{array}{ll} \min_z & V(z) \triangleq \frac{1}{2} z' Q z + c' z \\ \text{s.t.} & \ell \leq Az \leq u \\ & Gz = g \\ & \bar{\ell} \leq \bar{A} z \leq \bar{u} \end{array}$$

QP$_1$:

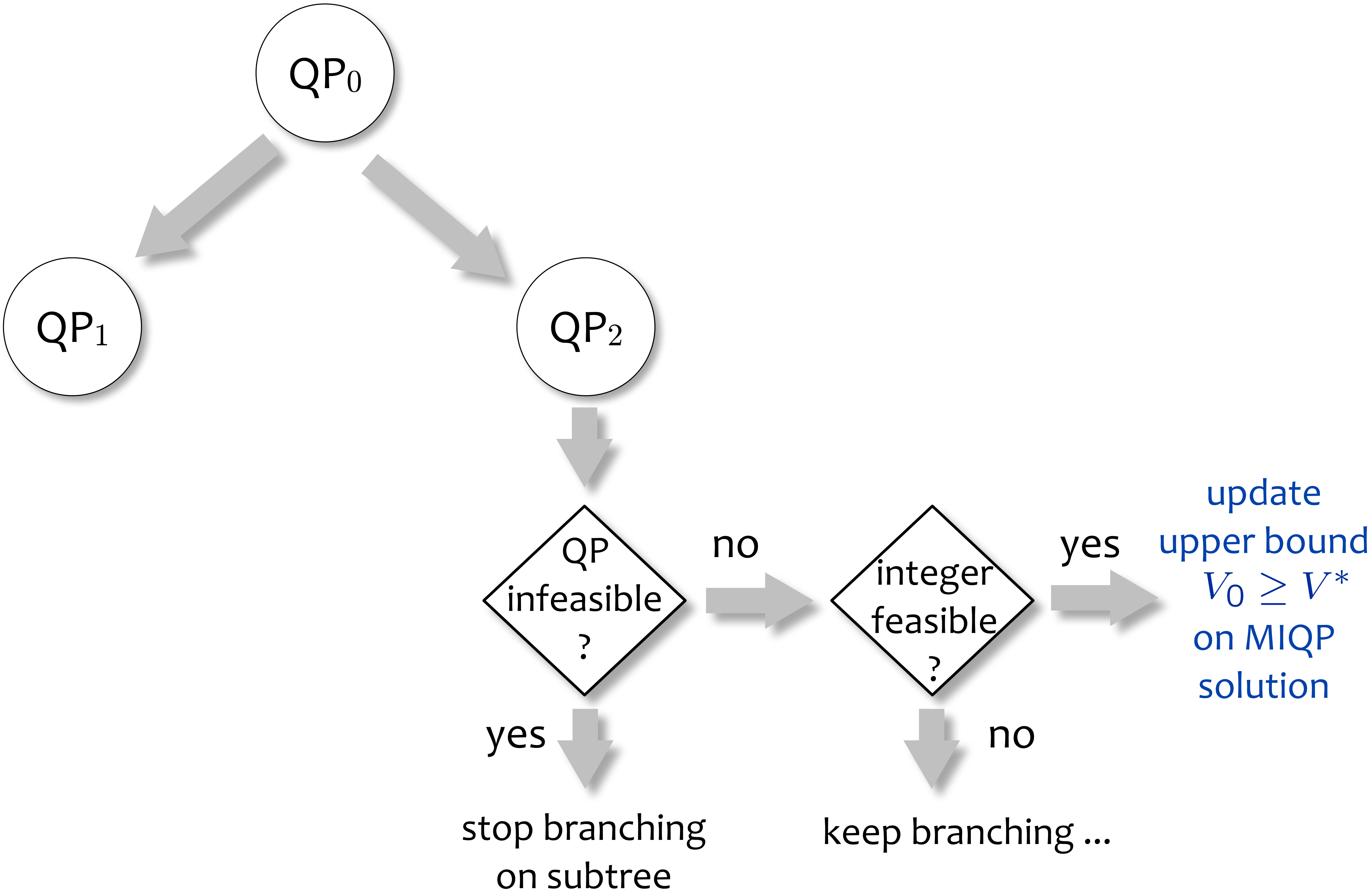
$$\begin{array}{ll} \min_z & V(z) \triangleq \frac{1}{2} z' Q z + c' z \\ \text{s.t.} & \ell \leq Az \leq u \\ & Gz = g \\ & A_i z = \bar{\ell}_i \\ & \bar{\ell}_j \leq \bar{A}_j z \leq \bar{u}_j, \ \ j \neq i \end{array}$$

QP$_2$:

$$\begin{array}{ll} \min_z & V(z) \triangleq \frac{1}{2} z' Q z + c' z \\ \text{s.t.} & \ell \leq Az \leq u \\ & Gz = g \\ & A_i z = \bar{u}_i \\ & \bar{\ell}_j \leq \bar{A}_j z \leq \bar{u}_j, \ \ j \neq i \end{array}$$

Warm start from previous solution of QP$_0$ helps solving QP$_1$, QP$_2$

BRANCH & BOUND FOR MIQP (USING NNLS SOLVER FOR QP)



MIQP VIA NNLS: NUMERICAL RESULTS

(Bemporad, NMPC 2015)

- Worst-case CPU time on random **MIQP** problems:

n	m	q	NNLS_{LDL}	NNLS_{QR}	GUROBI	CPLEX
10	5	2	2.3	1.2	1.4	8.0
10	100	2	5.7	3.3	6.1	31.4
50	25	5	4.2	6.1	14.1	30.1
50	200	10	68.8	104.4	114.6	294.1
100	50	2	4.6	10.2	37.2	69.2
100	200	15	137.5	365.7	259.8	547.8
150	100	5	15.6	49.2	157.2	260.1
150	300	20	1174.4	3970.4	1296.1	2123.9

n = # variables, m = # inequality constraints, no equalities, q = # binary constraints

QP algorithm in compiled Embedded MATLAB code, B&B in interpreted MATLAB code.
CPU time measured on this Mac

NNLS_{LDL} = **recursive LDL factorization** used to solve least-square problems in QP solver
NNLS_{QR} = **recursive QR factorization** used instead (numerically more robust)

FAST GRADIENT PROJECTION FOR MIQP

(Naik, Bemporad, 2016)

- MIQP problem

$$\begin{array}{ll} \min_z & V(z) \triangleq \frac{1}{2} z' Q z + c' z \\ \text{s.t.} & \ell \leq Az \leq u \\ & A_{eq} z = b_{eq} \\ & \bar{A}_i z \in \{\bar{\ell}_i, \bar{u}_i\}, \ i = 1, \dots, p \end{array}$$

special case:
binary constraints $z_i \in \{0,1\}$

- Use branch & bound, relax binary constraints to $\bar{\ell}_i \leq \bar{A}_i z \leq \bar{u}_i$

- Only projection changes from one QP relaxation to another:

constraint is relaxed $\quad \bar{A}_i z \leq \bar{u}_i \ \rightarrow \ y_{k+1}^i = \max\left\{y_k^i + s_k^i, 0\right\} \quad y_i \geq 0$

constraint is fixed $\quad \bar{A}_i z = \bar{u}_i \ \rightarrow \ y_{k+1}^i = y_k^i + s_k^i \quad y_i \gtrless 0$

constraint is ignored $\quad \bar{A}_i z = \bar{\ell}_i \ \rightarrow \ y_i^{k+1} = 0 \quad y_i = 0$

FAST GRADIENT PROJECTION FOR MIQP

(Naik, Bemporad, 2016)

- Same dual QP matrices, preconditioning only computed at root node
- Warm-start exploited, dual cost used to stop QP relaxations earlier
- Criterion based on Farkas lemma to detect QP infeasibility
- Numerical results (time in ms):

n	m	p	q	miqpGPAD	GUROBI
10	100	2	2	15.6	6.56
50	25	5	3	3.44	8.74
50	150	10	5	63.22	46.25
100	50	2	5	6.22	26.24
100	200	15	5	164.06	188.42
150	100	5	5	31.26	88.13
150	200	20	5	258.80	274.06
200	50	15	6	35.08	144.38

HEURISTIC ADMM METHOD FOR (SUBOPTIMAL) MIQP

(Takapoui, Moehle, Boyd, Bemporad, ACC'16)

- MIQP problem:

$$
\begin{array}{ll}
\min & \frac{1}{2}x'Qx + q'x \\
\text{s.t.} & \ell \leq Ax \leq u \\
 & A_i x \in \{\ell_i, u_i\}, \ \ i \in I
\end{array}
$$

- ADMM iterations:

quantization

$$
\begin{aligned}
x^{k+1} &= -(Q+\rho A^T A)^{-1}(\rho A^T(y^k - z^k) + q) \\
z^{k+1} &= \min\{\max\{Ax^{k+1} + y^k, \ell\}, u\} \\
z_i^{k+1} &= \begin{cases} \ell_i & \text{if } z_i^{k+1} < \frac{\ell_i+u_i}{2} \\ u_i & \text{if } z_i^{k+1} \geq \frac{\ell_i+u_i}{2} \end{cases}, \ \ i \in I \\
y^{k+1} &= y^k + Ax^{k+1} - z^{k+1}
\end{aligned}
$$

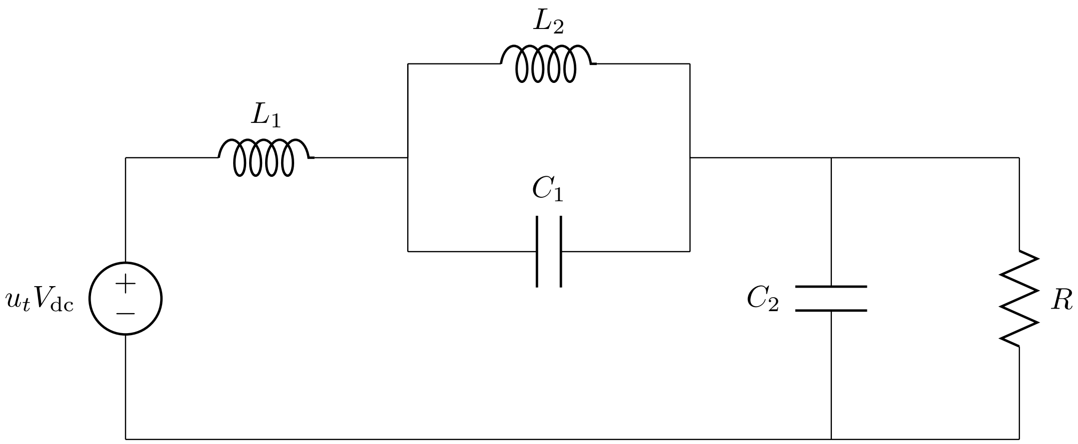
- Iterations converge to a (local) solution

- Similar idea also applicable to fast gradient methods (Naik, Bemporad, 2016)

ADMM METHOD FOR (SUBOPTIMAL) MIQP

(Takapoui, Moehle, Boyd, Bemporad, ACC'16)

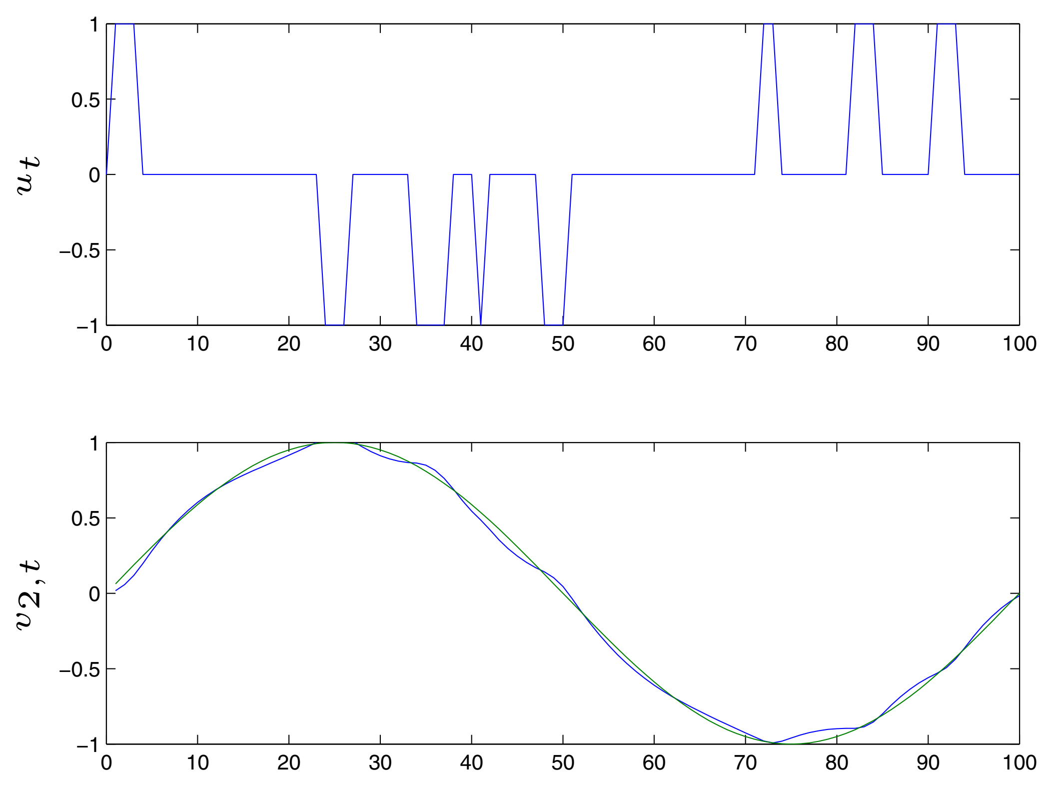
- Example: power converter control



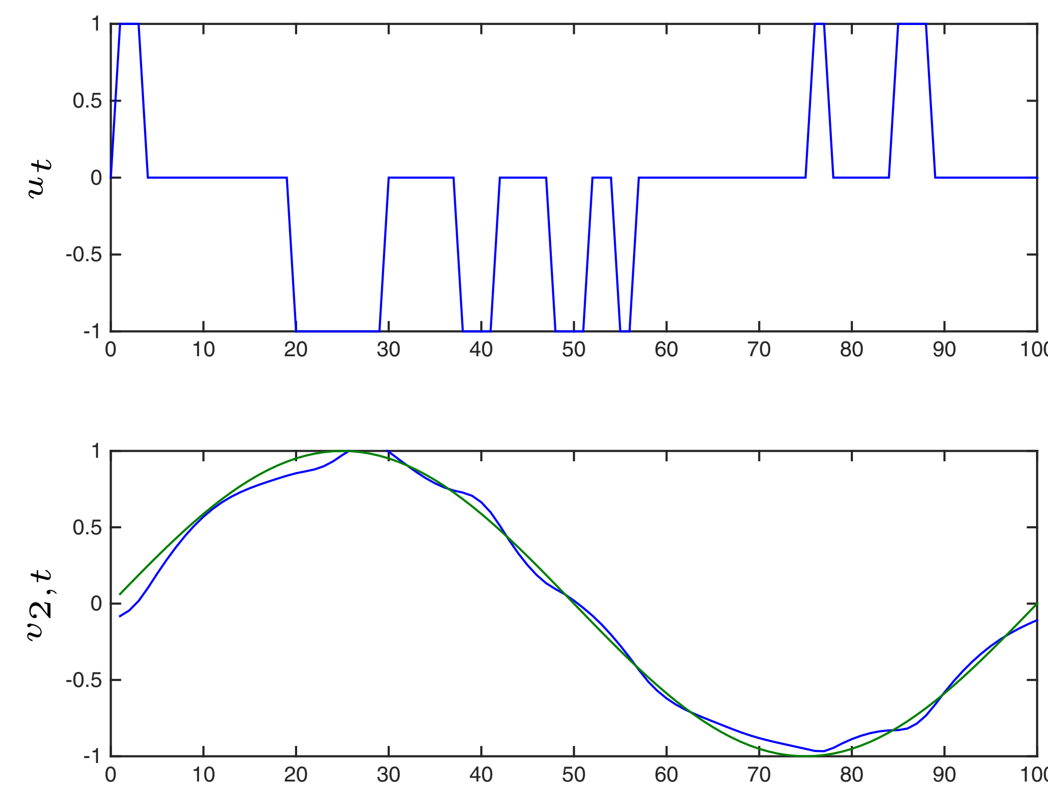
$$\begin{array}{ll} \text{minimize} & \sum_{t=0}^{T} (v_{2,t} - v_{\rm des})^2 + \lambda |u_t - u_{t-1}| \\ \text{subject to} & \xi_{t+1} = G\xi_t + Hu_t \\ & \xi_0 = \xi_T \\ & u_0 = u_T \\ & u_t \in \{-1, 0, 1\}, \end{array}$$

input voltage sign u_t

output voltage v_2



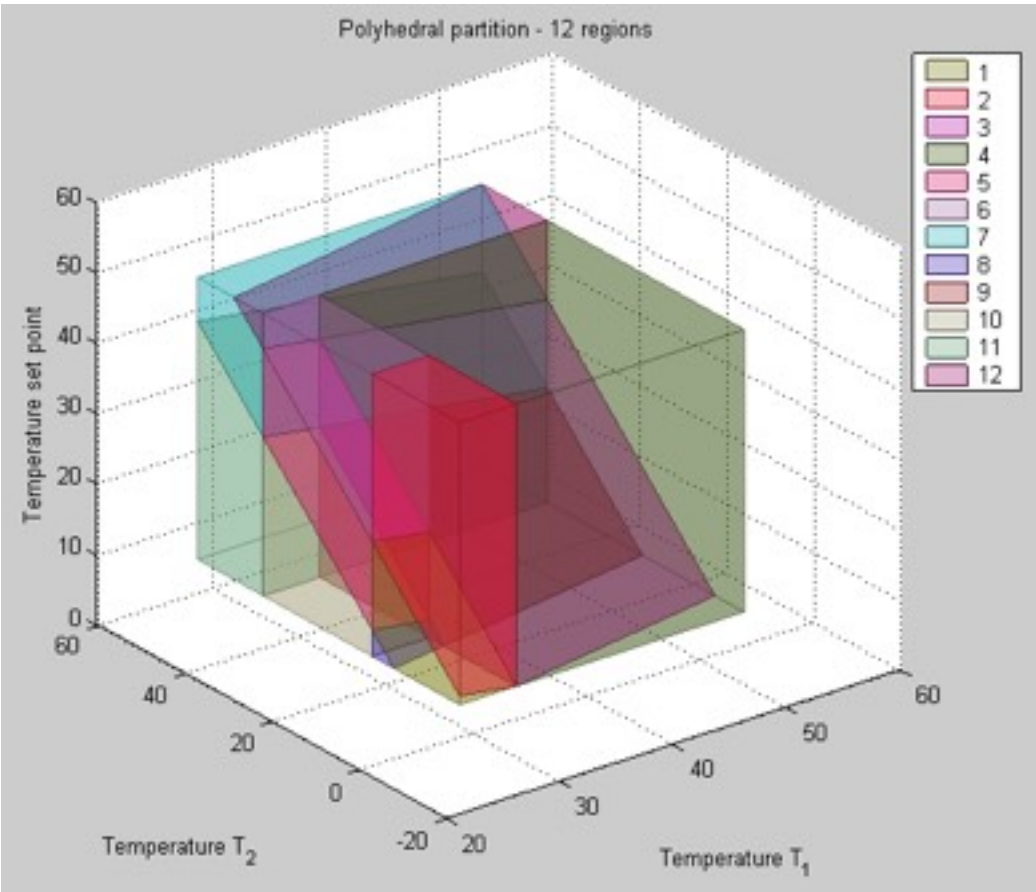
optimal solution



ADMM solution

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

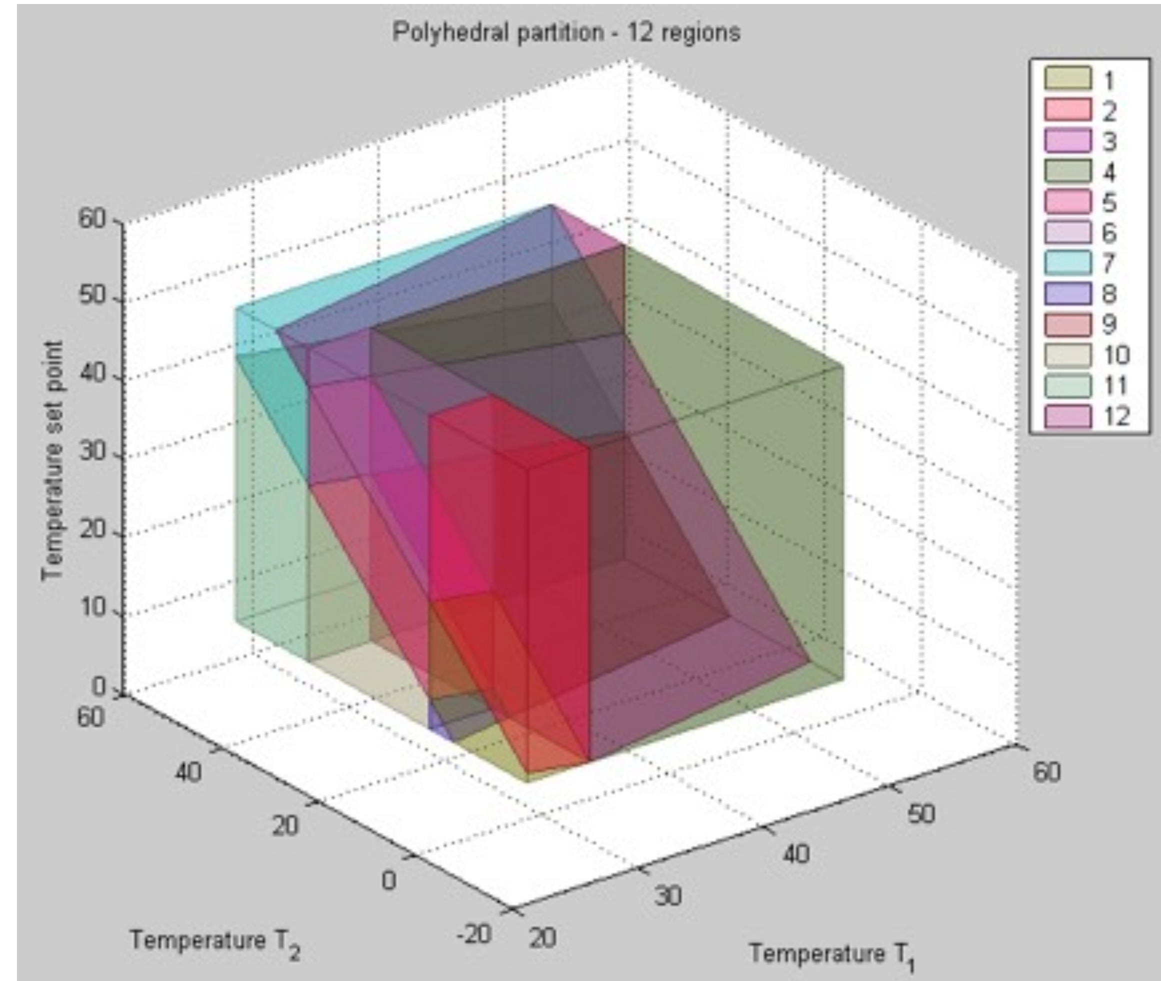
```
>>E=expcon(C,range,options);
```

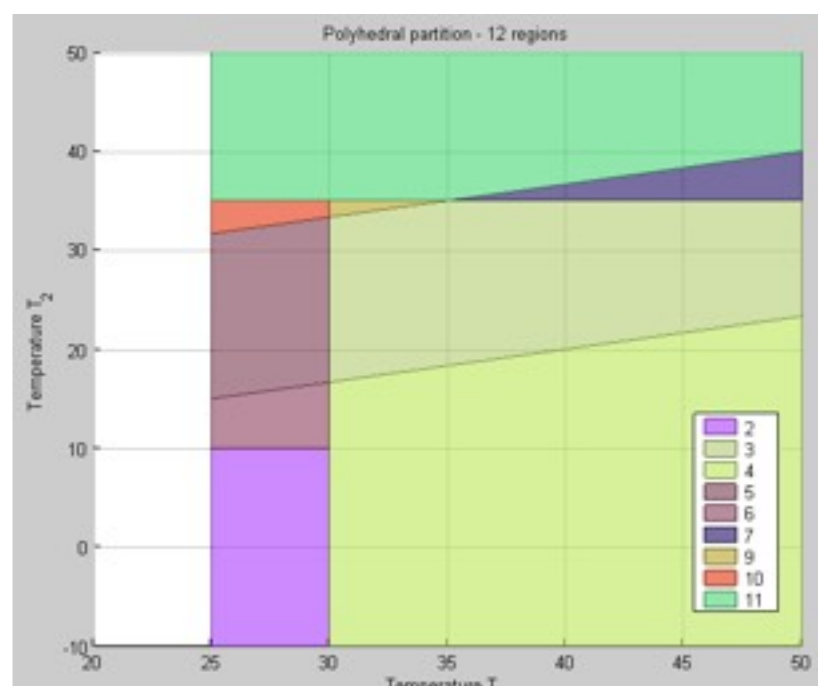
```
>> E

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

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

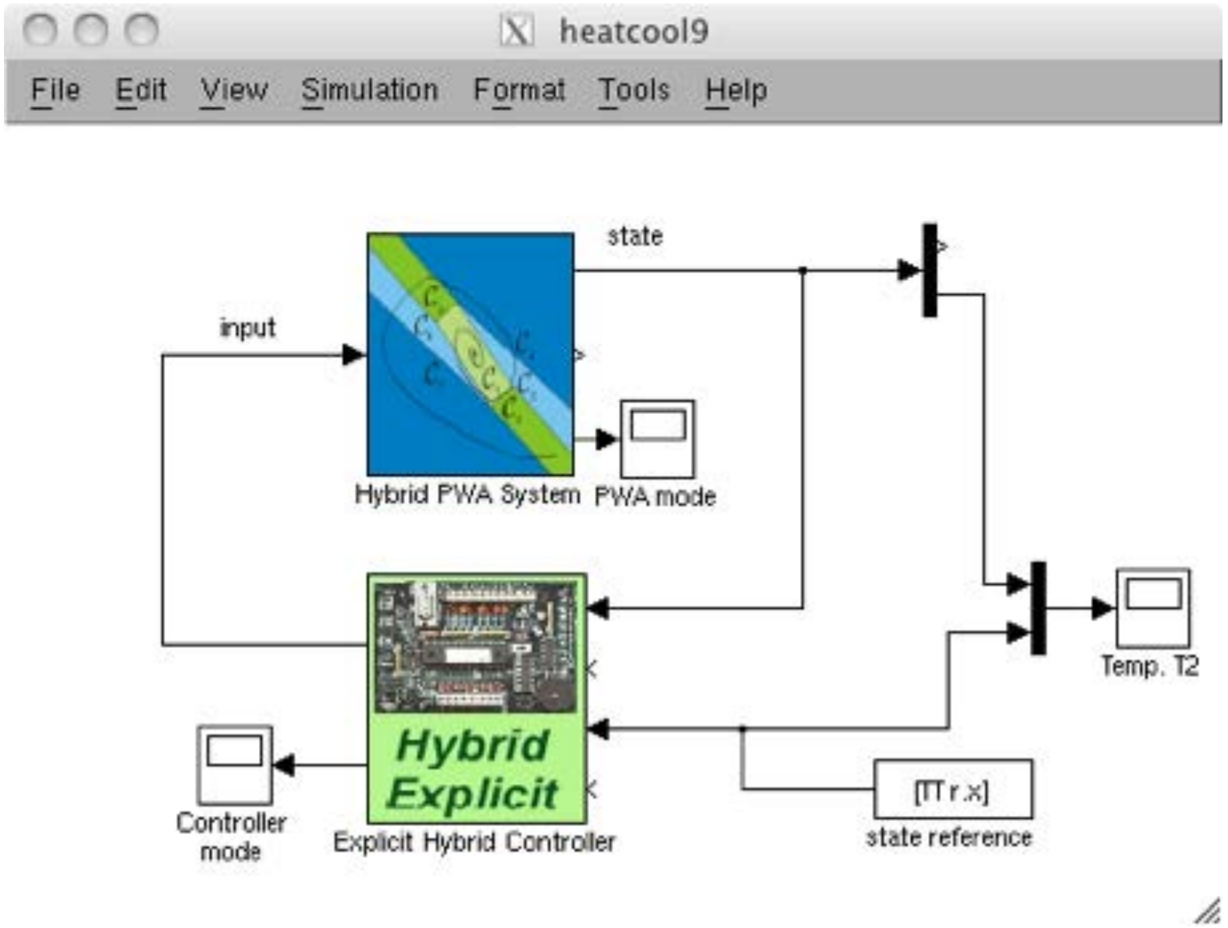
$$\text{s.t.} \begin{cases} x_{1k} \geq 25, & k = 1,2 \\ \text{hybrid model} \end{cases}$$

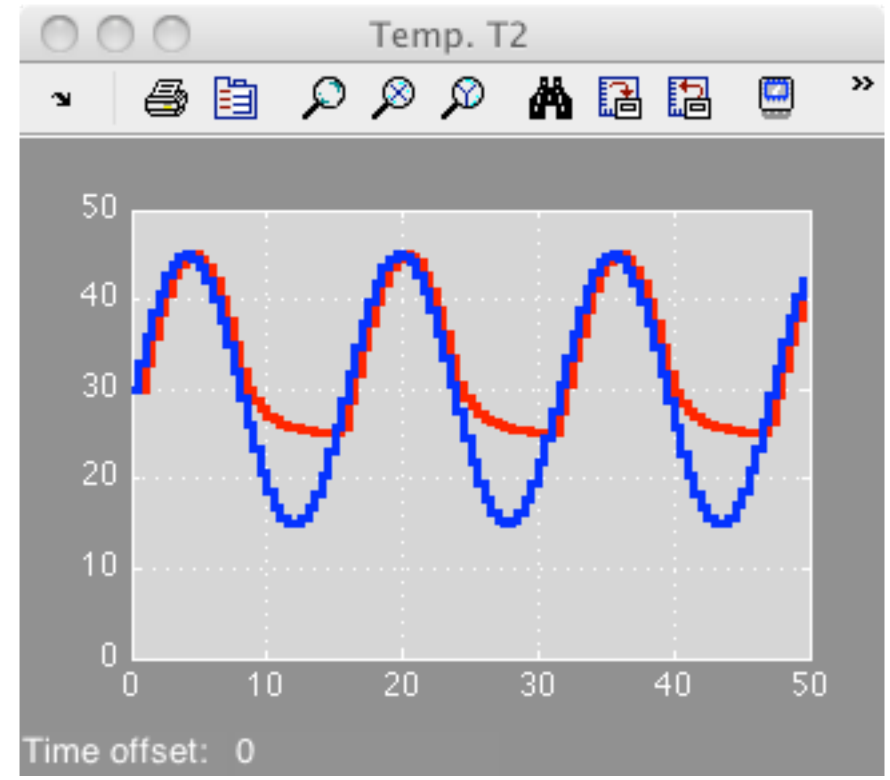




Section in the (T_1,T_2)-space for $T_{ref} = 30$

EXPLICIT MPC – TEMPERATURE CONTROL

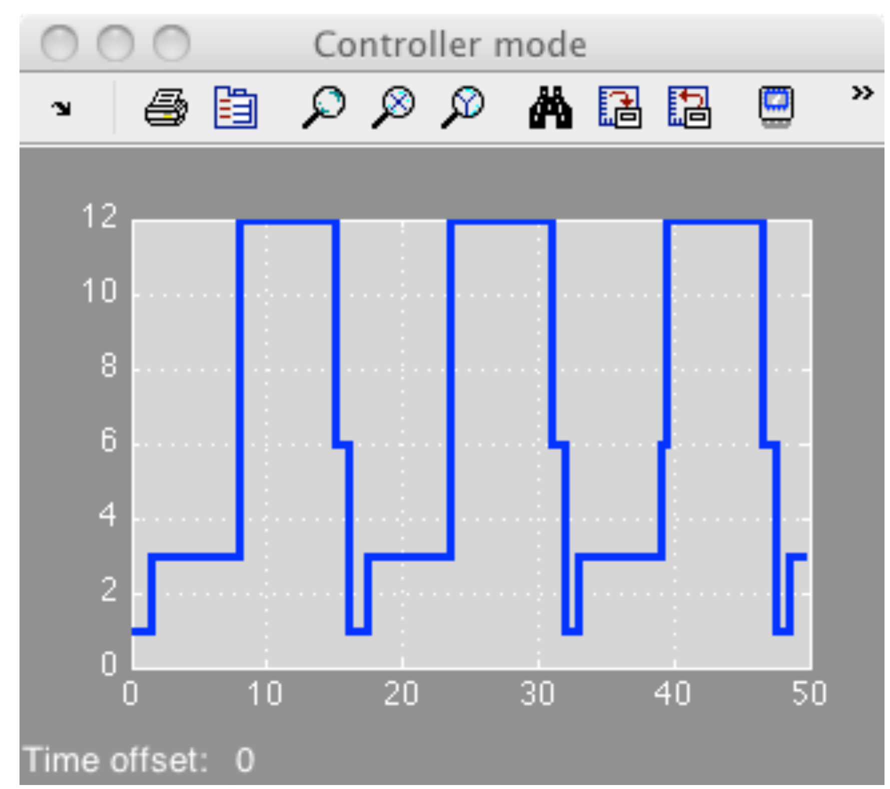




generated C-code

utils/expcon.h

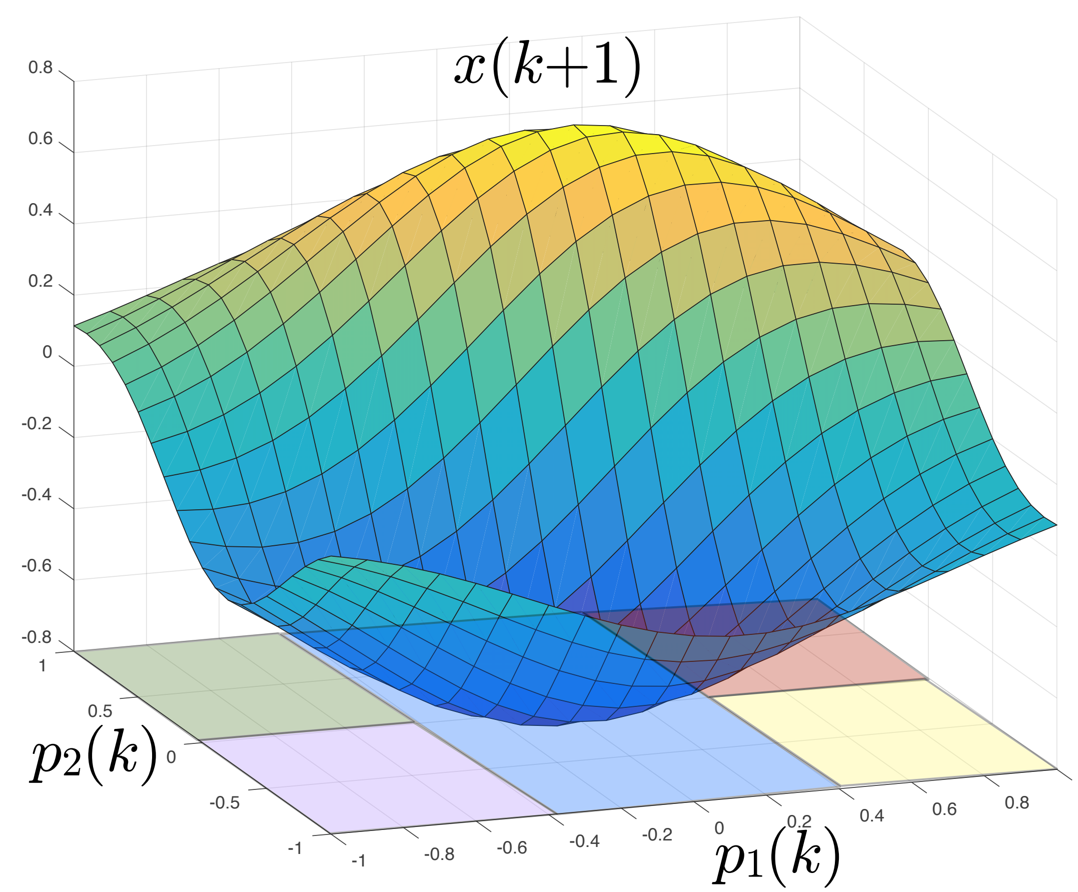
```
#define EXPCON_REG 12
#define EXPCON_NTH 3
#define EXPCON_NYM 2
#define EXPCON_NH 72
#define EXPCON_NF 12
static double EXPCON_F[]={
    -1,0,0,0,-1,0,
    -1,-1,-1,-1,-1,-1,0,-3,-3,
    -3,0,-3,0,0,0,0,0,
    0,0,4,4,4,0,4,0,0,
    0,0,0,0};

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,
```



HYBRID SYSTEMS IDENTIFICATION

- Model Predictive Control requires a **model** of the process.
- Models are usually obtained from **data** via systems identification (*offline* and/or *online*)
- Models may depend on parameters (e.g., ambient conditions)

In industrial MPC applications, most of the effort is spent in **identifying (multiple) linear** prediction models from data



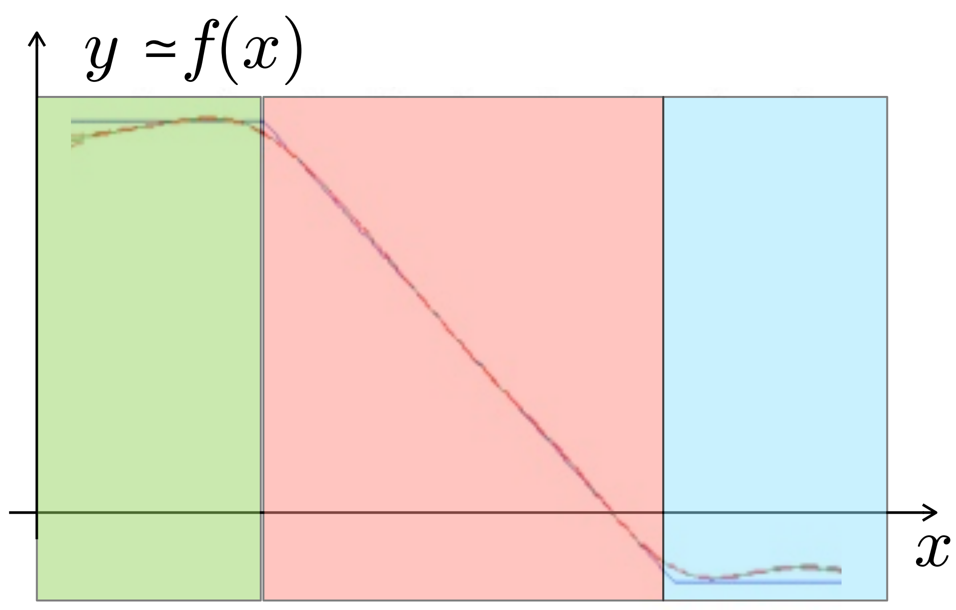
HYBRID SYSTEMS IDENTIFICATION

- **Problem:** given input/output pairs $\{x(k), y(k)\}$, $k = 1, \dots, N$ and number s of models, compute an approximation $y \simeq f(x)$

$$f(x) = \begin{cases} F_1 x + g_1 & \text{if } H_1 x \leq K_1 \\ \vdots \\ F_s x + g_s & \text{if } H_s x \leq K_s \end{cases}$$

PWA model (PieceWise Affine)

- Need to learn **both** the parameters (F_i, g_i) of the affine submodels **and** the partition (H_i, K_i) of the PWA map from data (*off-line* learning)



- Possibly need to update model and partition as new data are collected (*on-line* learning)

APPROACHES TO PWA IDENTIFICATION

- Mixed-integer linear or quadratic programming (Roll, Bemporad, Ljung, 2004)
- Partition of infeasible set of inequalities (Bemporad, Garulli, Paoletti, Vicino, 2005)
- K-means clustering in a feature space (Ferrari-Trecate, Muselli, Liberati, Morari, 2003)
- Bayesian approach (Juloski, Wieland, Heemels, 2004)
- Kernel-based approaches (Pillonetto, 2016)
- Hyperplane clustering in data space (Münz, Krebs, 2002)
- Recursive multiple least squares & PWL separation (Breschi, Piga, Bemporad, 2016)

PWA REGRESSION ALGORITHM

(Breschi, Piga, Bemporad, 2016)

1. **Estimate** the parameter matrices (F_i, g_i) **recursively**, by only updating one model $F_{i(k)}, g_{i(k)}$ at the time such that

$$i(k) \leftarrow \arg\min_{i=1,\ldots,s} e_i(k)'\Lambda_e^{-1}e_i(k) + (x(k)-c_i)'R_i^{-1}(x(k)-c_i)$$

one-step prediction error of model #i

proximity to centroid of cluster #i

recursive least squares based on inverse QR decomposition (Alexander, Ghirnikar, 1993)

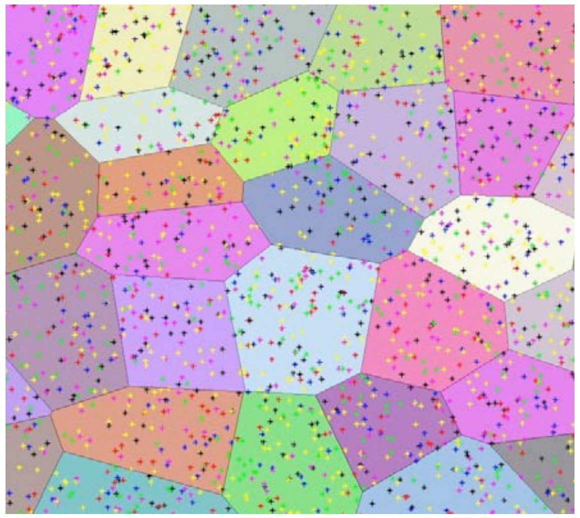
This also splits the data points $x(k)$ in **clusters** $C_i = \{x(k) : \ i(k) = i\}$

2. Compute a **polyhedral partition** (H_i, K_i) of the regressor space via multi-category linear separation

$$\phi(x) = \max_{i=1,\ldots,s} \{w_i'x - \gamma_i\}$$

(Bennet, Mangasarian, 1994)

- **Robust linear programming**
- **Piecewise-smooth Newton method**
- **Averaged stochastic gradient descent**



PWA REGRESSION EXAMPLES

(Breschi, Piga, Bemporad, 2016)

- **Identification** of **piecewise-affine LPV-ARX** model

$$\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} -0.83 & 0.20 \\ 0.30 & -0.52 \end{bmatrix} \begin{bmatrix} y_1(k-1) \\ y_2(k-1) \end{bmatrix} + \begin{bmatrix} -0.34 & 0.45 \\ -0.30 & 0.24 \end{bmatrix} \begin{bmatrix} u_1(k-1) \\ u_2(k-1) \end{bmatrix} + \begin{bmatrix} 0.20 \\ 0.15 \end{bmatrix} + \max\left\{ \begin{bmatrix} 0.20 & -0.90 \\ 0.10 & -0.42 \end{bmatrix} \begin{bmatrix} y_1(k-1) \\ y_2(k-1) \end{bmatrix} + \begin{bmatrix} 0.42 & 0.20 \\ 0.50 & 0.64 \end{bmatrix} \begin{bmatrix} u_1(k-1) \\ u_2(k-1) \end{bmatrix} + \begin{bmatrix} 0.40 \\ 0.30 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\} + e_o(k),$$

Results:

quality of fit

		$N = 4000$	$N = 20000$	$N = 100000$
BFR_1	(Off-line) RLP [8]	96.0 %	96.5 %	99.0 %
	(Off-line) RPSN	96.2 %	96.4 %	98.9 %
	(On-line) ASGD	86.7 %	95.0 %	96.7 %
BFR_2	(Off-line) RLP [8]	96.2 %	96.9 %	99.0 %
	(Off-line) RPSN	96.3 %	96.8 %	99.0 %
	(On-line) ASGD	87.4 %	95.2 %	96.4 %

RLP = robust linear programming
RPSN = piecewise-smooth Newton
ASGD = (one-pass) averaged stochastic gradient

$$\mathrm{BFR}_i = \max\left\{1 - \frac{\|y_{o,i} - \hat{y}_i\|_2}{\|y_{o,i} - \bar{y}_{o,i}\|_2}, 0\right\}$$

(Best Fit Rate)

CPU time for computing the partition

	$N = 4000$	$N = 20000$	$N = 100000$
(Off-line) RLP [8]	0.308 s	3.227 s	112.435 s
(Off-line) RPSN	0.016 s	0.086 s	0.365 s
(On-line) ASGD	0.013 s	0.023 s	0.067 s

(Breschi, Piga, Bemporad, 2016)

- **Identification** of **piecewise-linear LPV-ARX** model

$$\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} \bar{a}_{1,1}(p(k)) & \bar{a}_{1,2}(p(k)) \\ \bar{a}_{2,1}(p(k)) & \bar{a}_{2,2}(p(k)) \end{bmatrix} \begin{bmatrix} y_1(k-1) \\ y_2(k-1) \end{bmatrix} + \begin{bmatrix} \bar{b}_{1,1}(p(k)) & \bar{b}_{1,2}(p(k)) \\ \bar{b}_{2,1}(p(k)) & \bar{b}_{2,2}(p(k)) \end{bmatrix} \begin{bmatrix} u_1(k-1) \\ u_2(k-1) \end{bmatrix} + e_o(k)$$

$$\bar{a}_{1,1}(p(k)) = \begin{cases} -0.3 & \text{if} \quad 0.4\,(p_1(k)+p_2(k)) \le -0.3, \\ 0.3 & \text{if} \quad 0.4\,(p_1(k)+p_2(k)) \ge 0.3, \\ 0.4\,(p_1(k)+p_2(k)) & \text{otherwise}, \end{cases}$$

$$\bar{a}_{1,2}(p(k)) = 0.5\,(|p_1(k)|+|p_2(k)|)\,, \quad \bar{a}_{2,1}(p(k)) = p_1(k)-p_2(k)$$

$$\bar{a}_{2,2}(p(k)) = \begin{cases} 0.5 & \text{if} \quad p_1(k) < 0, \\ 0 & \text{if} \quad p_1(k) = 0, \\ -0.5 & \text{if} \quad p_1(k) > 0, \end{cases}$$

$$\bar{b}_{1,1}(p(k)) = 3p_1(k) + p_2(k),$$

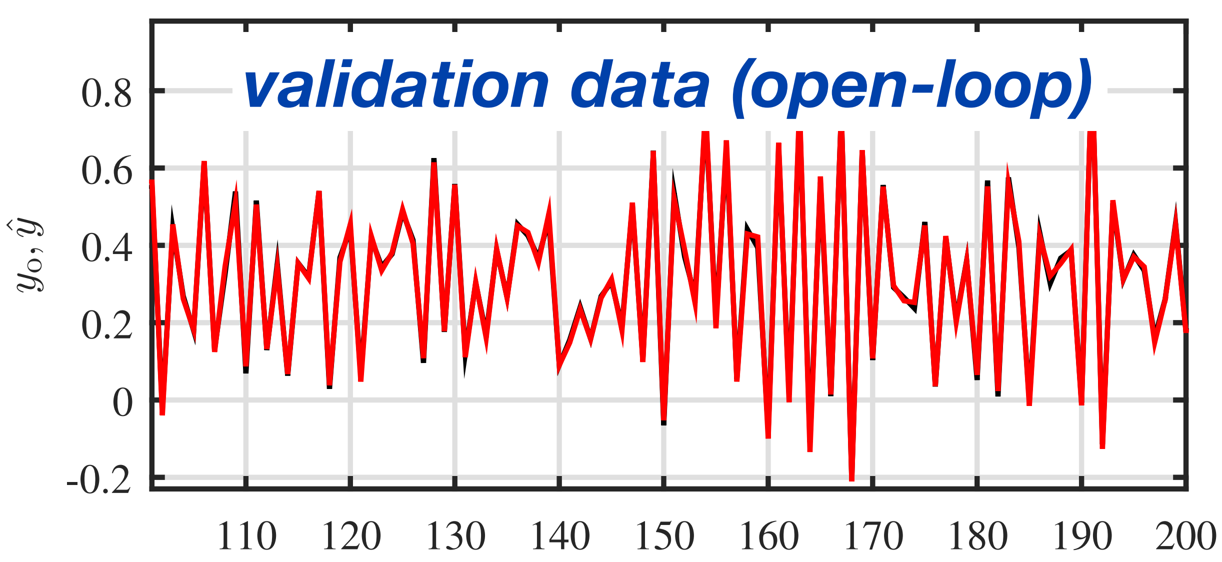
$$\bar{b}_{1,2}(p(k)) = \begin{cases} 0.5 & \text{if} \quad 2\,(p_1^2(k)+p_2^2(k)) \ge 0.5, \\ 2\,(p_1^2(k)+p_2^2(k)) & \text{otherwise}, \end{cases}$$

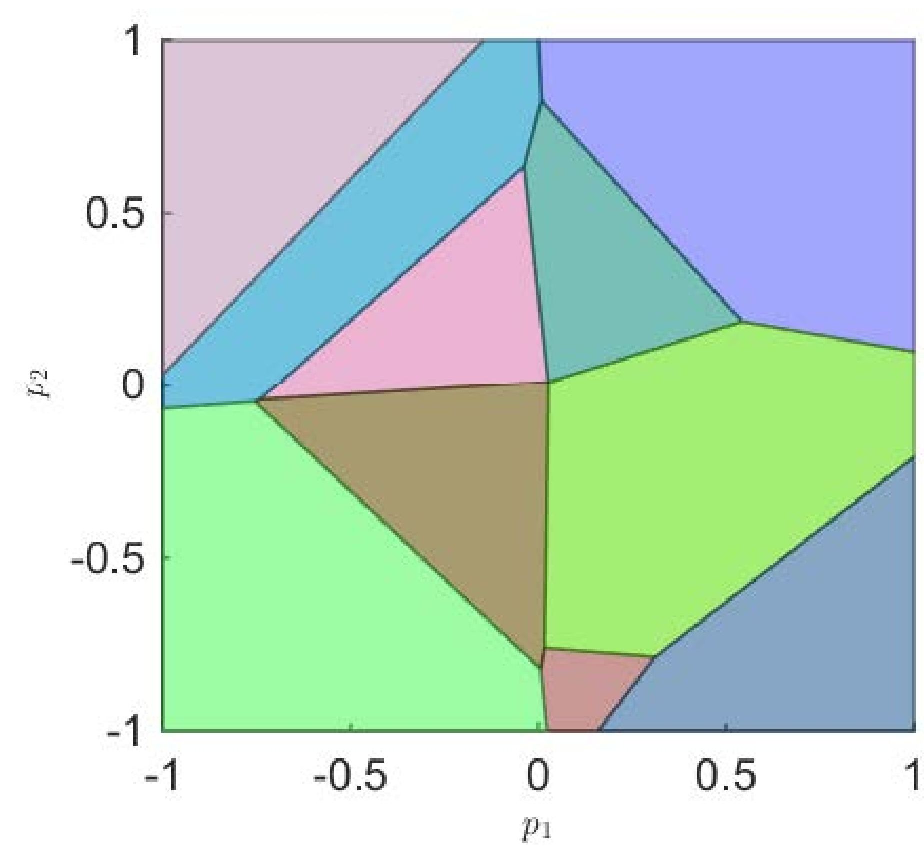
$$\bar{b}_{2,1}(p(k)) = 2\sin\{p_1(k)-p_2(k)\}\,, \quad \bar{b}_{2,2}(p(k)) = 0.$$

Results: ***quality of fit***

	BFR_1	BFR_2
PWA regression	87 %	84 %
parametric LPV [3]	80 %	70 %

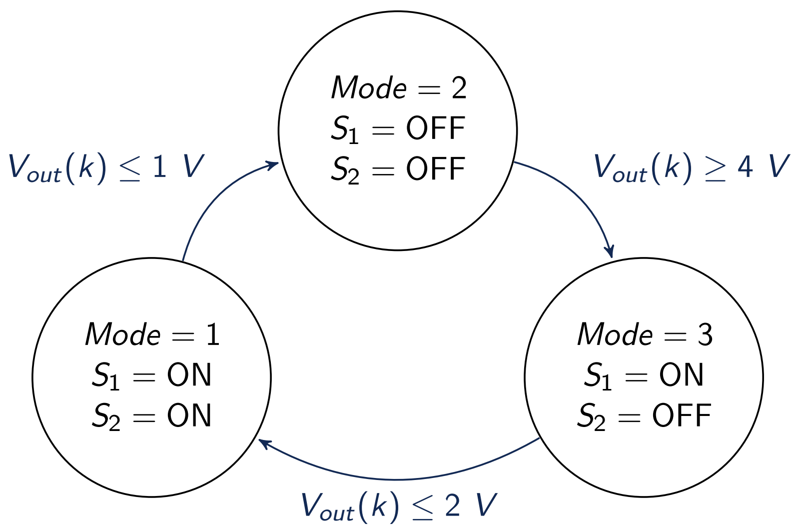
[3] = Bamieh, Giarré (2002)

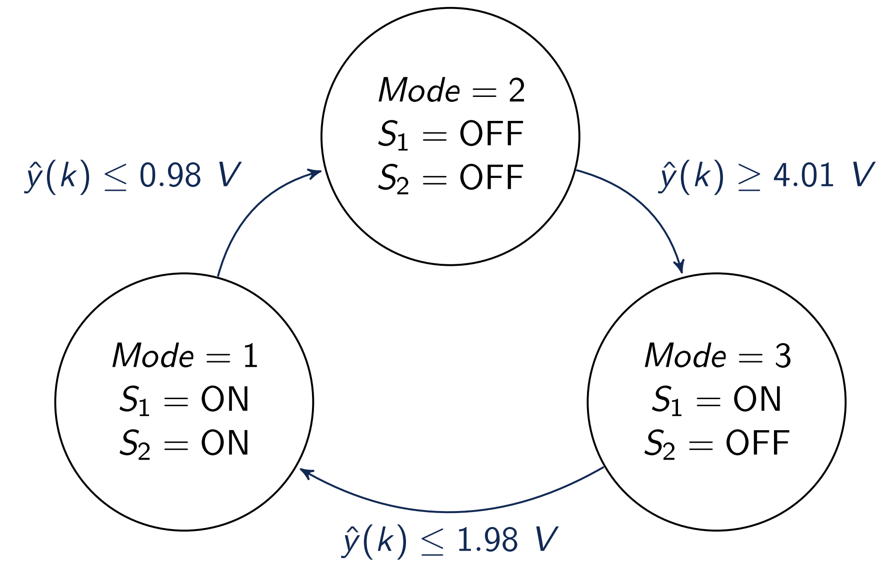




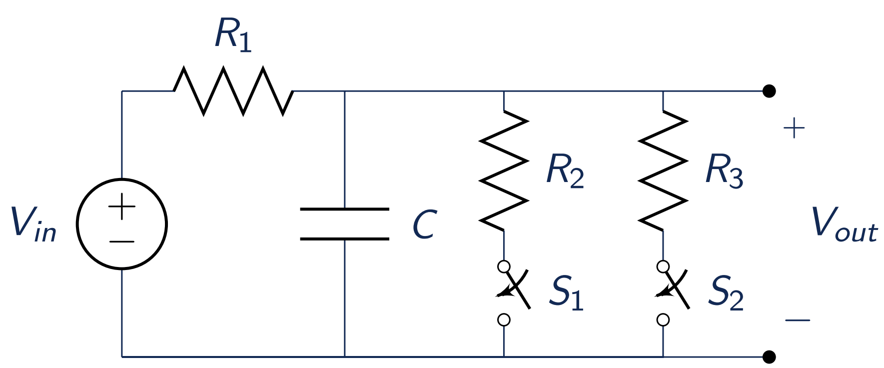
(Breschi, Piga, Bemporad, CDC 2016)

- **Identification** of **hybrid** models from data



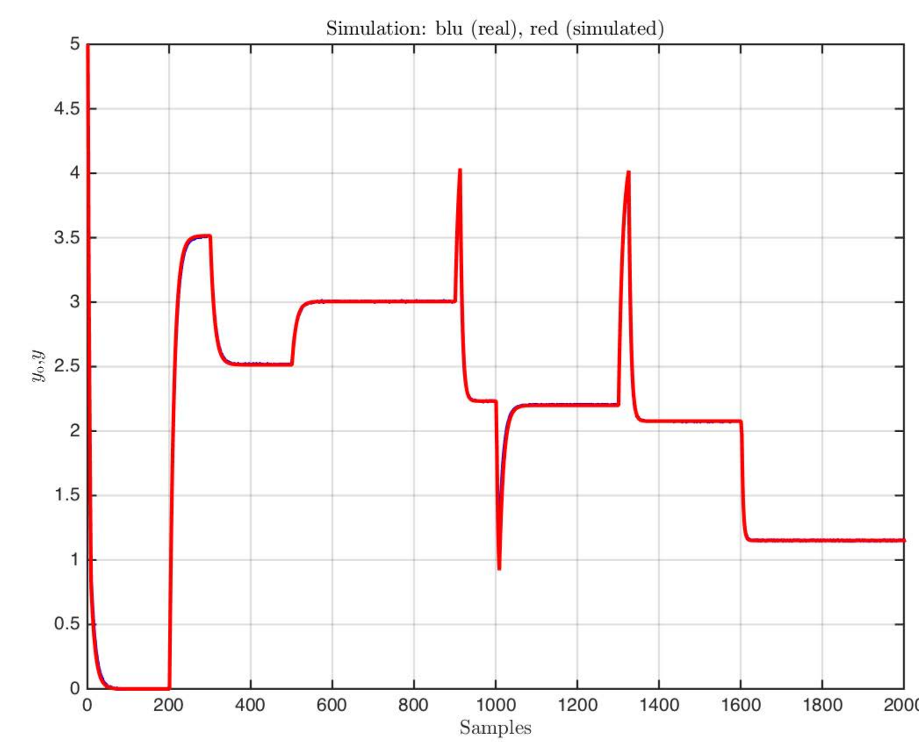


identified system



true system

CPU time to compute the partitions: **0.033 s**
CPU required to identify the DHA: **0.078 s***



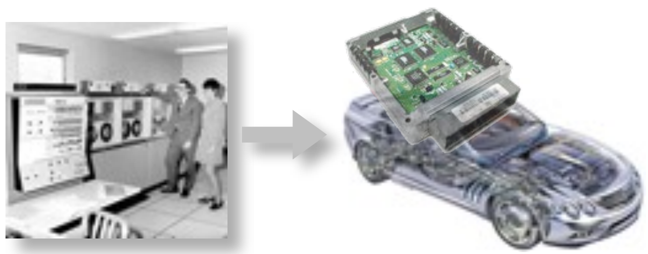
Validation set of $N_V = 2000$ samples:
$\mathrm{BFR} = 98.64\ \%$

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. Increasingly used in automotive, aerospace, energy, ...



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

Several control problems in real-world cyber-physical systems can be (and many are) well solved by MPC !

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

General references on MPC

[1] D.Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, n.12, p. 2967-2986, 2014

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

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

[4] 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/`.

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

[6] 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.

[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.

[11] V. Breschi, D. Piga, and A. Bemporad, "Piecewise affine regression via recursive multiple least squares and multicategory discrimination," Automatica, vol. 73, pp. 155–162, Nov. 2016.

Model predictive control

[12] 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

[13] 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.

[14] A. Bemporad, "A multiparametric quadratic programming algorithm with polyhedral computations based on nonnegative least squares," IEEE Trans. Automatic Control, vol. 60, no. 11, pp. 2892–2903, 2015.

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

[16] 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

[17] 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.

[18] 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.

[19] 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 automotive applications

[20] 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.

[21] S. Di Cairano, H.E. Tseng, D. Bernardini, and A. Bemporad, "Vehicle yaw stability control by coordinating active front steering and differential braking in the tire sideslip angles domain," IEEE Trans. Contr. Systems Technology, vol. 21, no. 4, pp. 1236–1248, July 2013.

[22] 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.

[23] S. Di Cairano, D. Bernardini, A. Bemporad, and I.V. Kolmanovsky, "Stochastic MPC with learning for driver-predictive vehicle control and its application to HEV energy management," IEEE Trans. Contr. Systems Technology, vol. 22, pp. 1018–1031, 2014.