RECENT ADVANCES IN EMBEDDED AND STOCHASTIC MODEL PREDICTIVE CONTROL

Alberto Bemporad

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

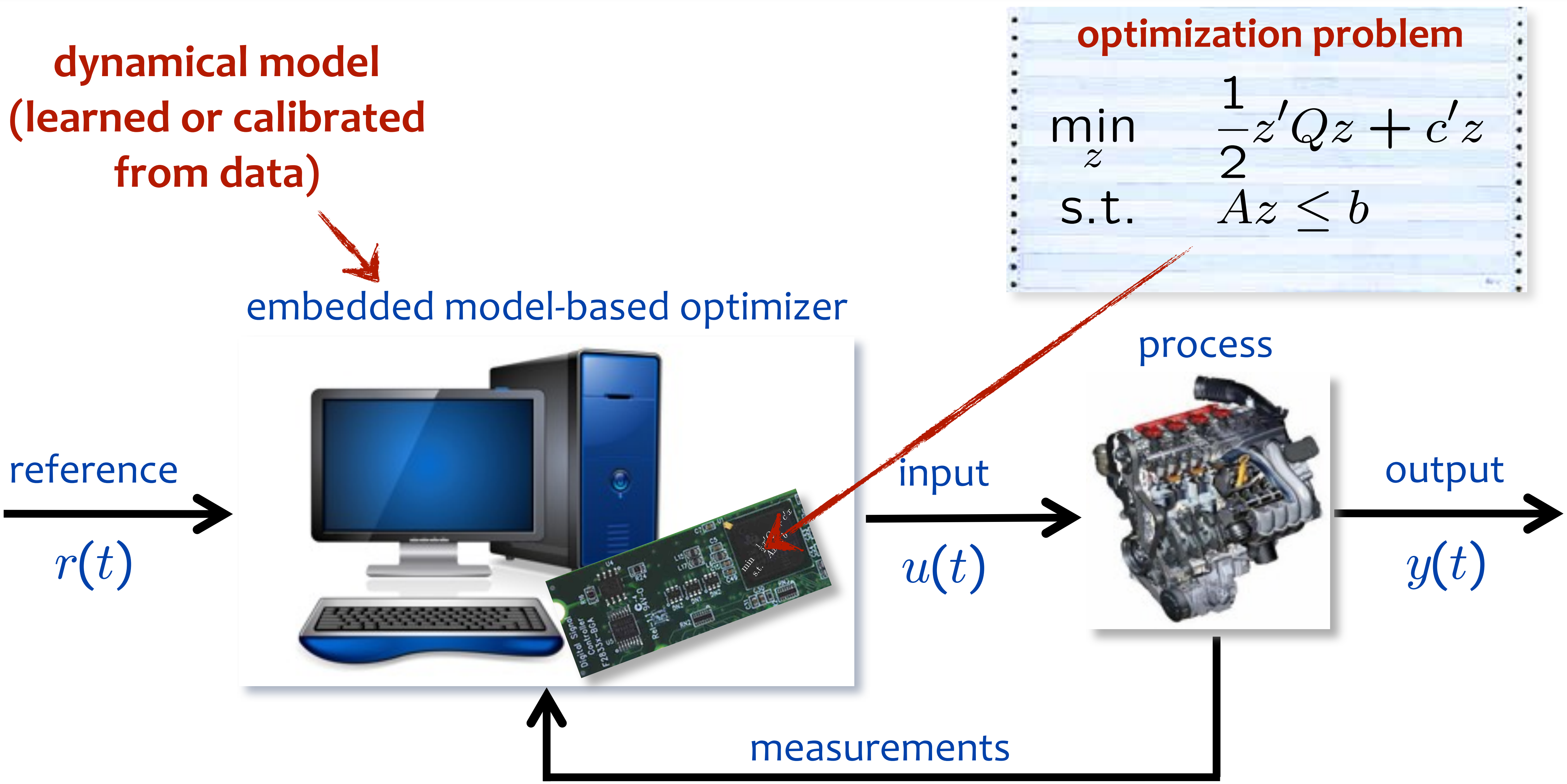
www.odys.it

IMT INSTITUTE FOR ADVANCED STUDIES LUCCA

March 21, 2016

ODYS Advanced Controls & Optimization

MODEL PREDICTIVE CONTROL (MPC)



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

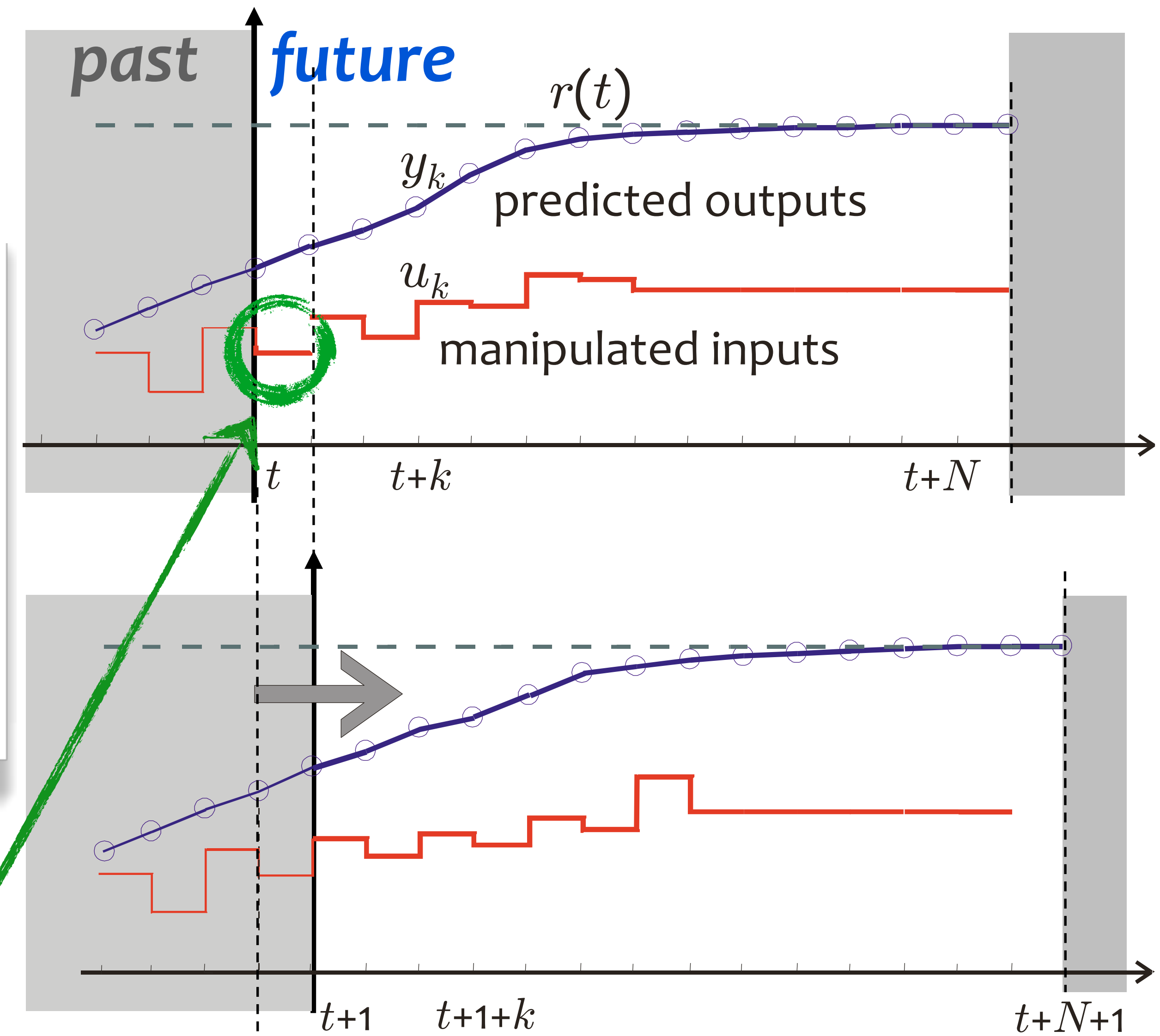
penalty on tracking error

penalty on actuation effort

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

← feedback !

optimization problem



- Solve problem 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

FIRST PAPER ON MPC ...

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

Discrete Dynamic Optimization Applied to On-Line Optimal Control

MARSHALL D. RAFAL and WILLIAM F. STEVENS
Northwestern University, Evanston, Illinois

A general method has been developed for controlling deterministic systems described by linear or linearized dynamics. The discrete problem has been treated in detail. Step-by-step optimal controls for a quadratic performance index have been derived. The method accommodates upper and lower limits on the components of the control vector.

A small binary distillation unit was considered as a typical application of the method. The control vector was made up of feed rate, reflux ratio, and reboiler heat load. Control to a desired state and about a load upset was effected.

Calculations are performed quite rapidly and only grow significantly with an increase in the dimension of the control vector. Extension to much larger distillation units with the same controls thus seems practical.

The advent of high-speed computers has made possible the on-line digital control of many chemical engineering processes. In on-line control a three-step procedure is adhered to:

1. Sense the current state.
2. Calculate a suitable control action.
3. Apply this control for a period of time known as the sampling period.

The present study proposes a method for performing step 2. The technique developed is based on linearized dynamics. The strongly nonlinear binary distillation unit provides a suitable system for this study. While much has been published recently (*2, 3, 8*) on modeling distillation, little if anything has appeared on the optimal control of such units.

In recent years, a good deal has been published by Kalman, Lapidus, and others (*4* to *7*) on the control of linear or linearized nonlinear systems by minimizing a quadratic function of the states resulting from a sequence of control actions. Their controls are always unconstrained, although the introduction of a quadratic penalty function limits this effect somewhat. The general constrained problem has been treated numerically (*1*) for a single control variable. It was Wanninger (*10, 11*) who first chose to look at the problem on a one-step-at-a-time basis rather than considering a sequence of controls. However, he made no attempt to solve completely the resulting quadratic programming problem.

The approach taken in the present work is to set up the problem on a one-step basis. This is quite compatible with the on-line digital control scheme. The problem is then shown to be a special case of the quadratic programming problem and as such has a special solution. The particulars concerning the theory underlying the solution scheme and its implementation on a digital computer have been presented (9). In addition, a derivation of the theorems upon which the computational algorithm is based is presented in the Appendix.

The authors wish to be very careful to point out that *optimal*, as used herein, refers only to a single step of control. Even for truly linear systems, the step-by-step optimal control need not be overall optimal. A recent text by Athans and Falb (*1a*) presents both the virtues and defects of such a one-step method. In the present work, the one-step approach is taken because it is amenable to practical solution of the problem and is well suited to nonlinear situations where updating linearization is useful.

THE PROBLEM

The system under consideration is described by a set of matrix differential equations:

$$\dot{X}(t) = AX(t) + BM(t) + \delta(t) \tag{1}$$

Marshall D. Rafal is with Esso Research and Engineering Company., Florham Park, New Jersey.

AUTOMOTIVE APPLICATIONS OF MPC

Bemporad, Bernardini, Borrelli, Cimini, Di Cairano, Esen, Giorgetti, Hrovat, Kolmanovsky, Ripaccioli, Trimboli, Tseng, Yanakiev, ... (2001-2016)

Powertrain

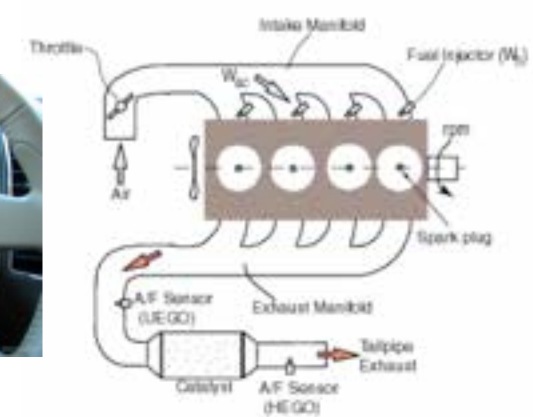
- direct-inj. engine control
- A/F ratio control
- magnetic actuators
- robotized gearbox
- power MGT in HEVs
- cabin heat control in HEVs
- electrical motors

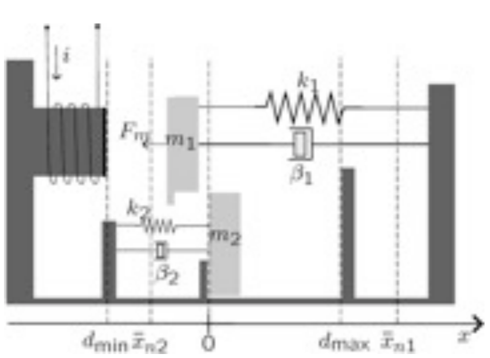
Vehicle dynamics

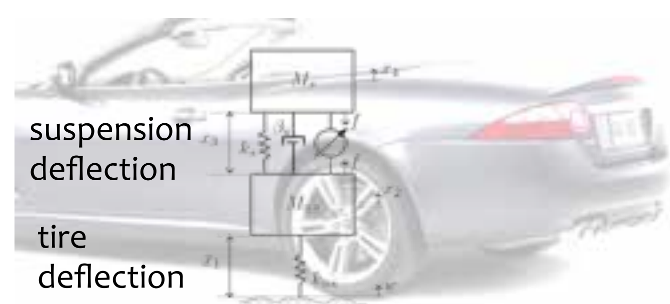
- traction control
- active steering
- semiactive suspensions


Homogeneous


Stratified







AEROSPACE APPLICATIONS OF MPC

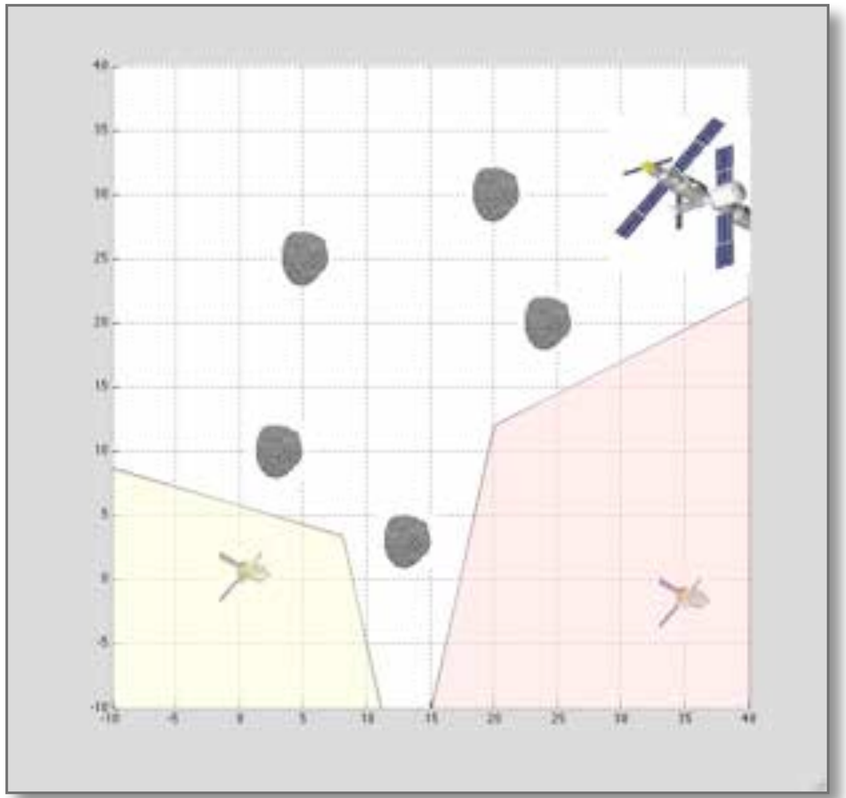
- **Main goal:** **explore MPC capabilities in new space applications:**
- New MATLAB MPC Toolboxes developed (**MPCTOOL** and **MPCSofT**)

(Bemporad, 2010) (Bemporad, 2012)

powered descent

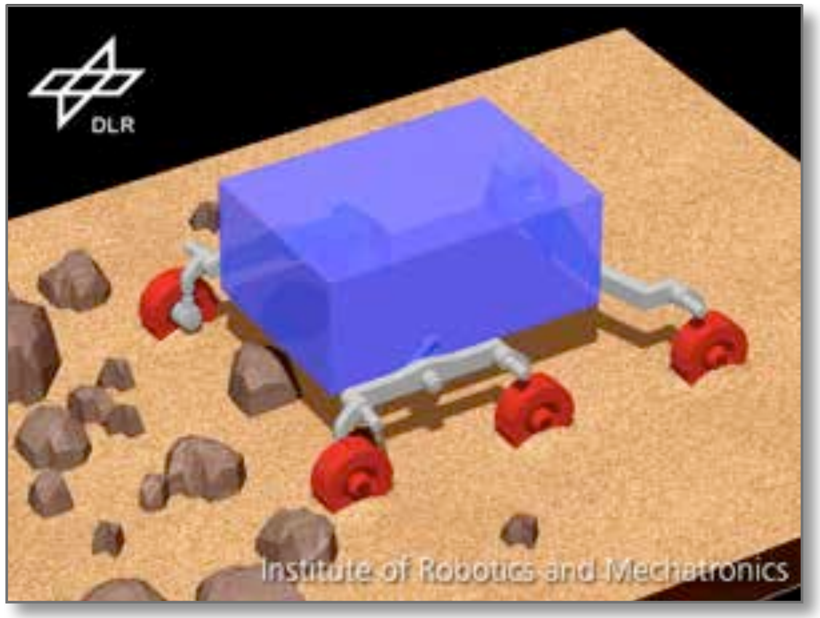
(Pascucci, Bennani, Bemporad, 2016)

cooperating UAVs



(Bemporad, Rocchi, 2011)

planetary rover

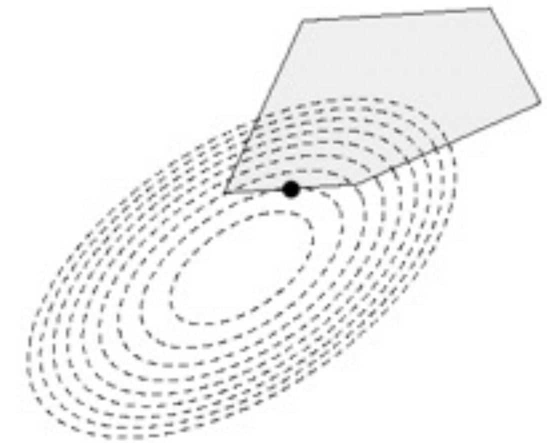


EMBEDDED LINEAR MPC

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

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

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



- Several algorithms exist to solve the QP **on-line** given $x(t)$:

  active set (AS), interior point (IP), gradient projection (GP), alternating direction method of multipliers (ADMM), proximal methods, ...

FIRST PAPER ON QP …

(Beale, 1955)

On Minimizing a Convex Function Subject to Linear Inequalities

By E. M. L. Beale

Admiralty Research Laboratory, Teddington, Middlesex

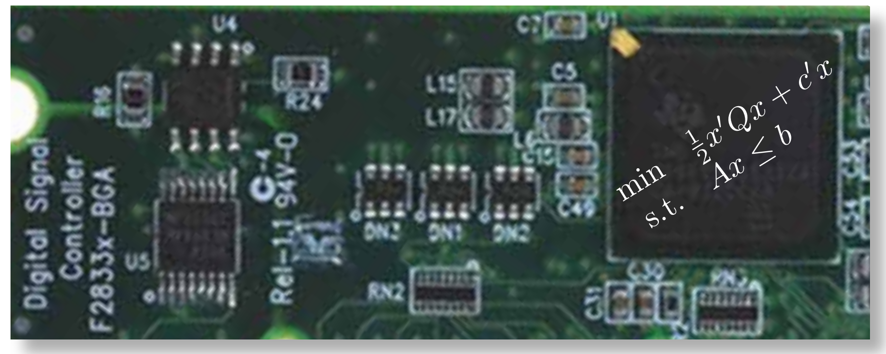
SUMMARY

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.

A rich set of good QP algorithms is available today, but still more research is needed to have an impact in real applications !

MPC IN A PRODUCTION ENVIRONMENT

embedded model-based optimizer



Requirements for production:

1. **Speed (throughput)**: solve optimization problem within sampling interval

2. **Robustness** with respect to finite-precision arithmetics

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

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

5. **Code simple** enough to be validated/verified/certified
   (in general, it must be understandable by production engineers)

MPC IN INDUSTRIAL AUTOMOTIVE PRODUCTION

FAST GRADIENT PROJECTION FOR (DUAL) QP

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

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

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

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

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

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

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

- 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

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

FAST GRADIENT PROJECTION FOR (DUAL) QP

(Patrinos, Bemporad, IEEE TAC, 2014)

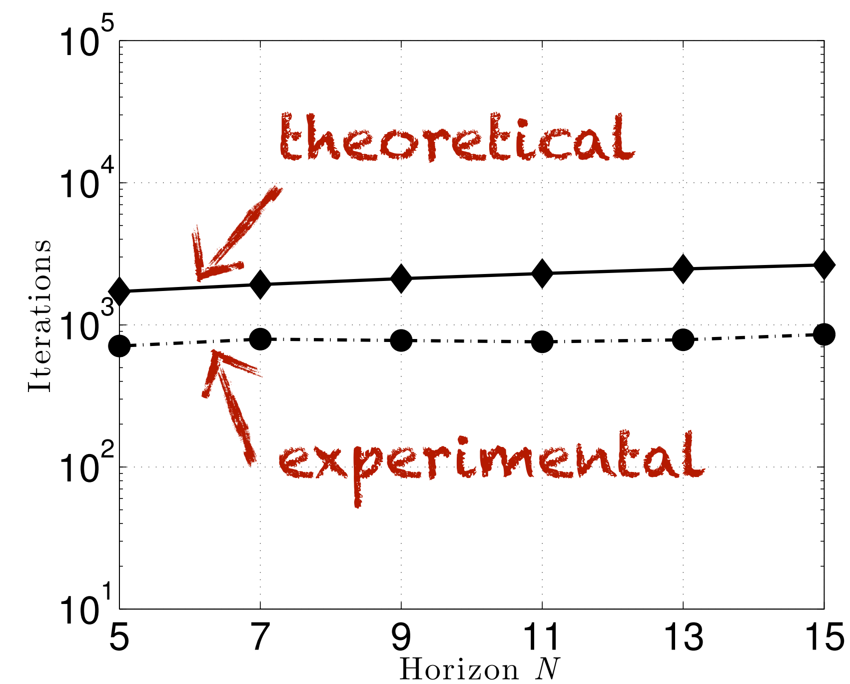
- Main on-line operations involve only **simple linear algebra**

```
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
```

- Convergence rate:

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

- Tight bounds on maximum number of iterations



- Can be used to warm-start other methods

- Currently extended to mixed-integer problems

(Naik, Bemporad, work in progress)

HARDWARE TESTS (FLOATING VS FIXED POINT)

(Patrinos, Guiggiani, Bemporad, 2013)

- Gradient projection works in fixed-point arithmetics

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

exponentially decreasing with number p of fractional bits

max constraint violation

Table 1
Fixed-point hardware implementation

Size [variables/constraints]	Time [ms]	Time per iteration [μs]	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 [μs]	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

32-bit Atmel SAM3X8E

ARM Cortex-M3 processing unit

84 MHz, 512 KB of flash memory

and 100 KB of RAM

fixed-point about 4x faster than floating-point

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)

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

1) $\mathcal{P} \leftarrow \emptyset$, $v \leftarrow 0$;
2) $w \leftarrow A'(Av - b)$;
3) **if** $w \geq 0$ **or** $\mathcal{P} = \{1, \ldots, m\}$ **then go to** Step 11;
4) $i \leftarrow \arg\min_{i \in \{1,\ldots,m\} \setminus \mathcal{P}} w_i$, $\mathcal{P} \leftarrow \mathcal{P} \cup \{i\}$;
5) $y_{\mathcal{P}} \leftarrow \arg\min_{z_{\mathcal{P}}} \|((A')_{\mathcal{P}})' z_{\mathcal{P}} - b\|_2^2$, $y_{\{1,\ldots,m\} \setminus \mathcal{P}} \leftarrow 0$;
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**.

Algorithm: *While maintaining primal var v feasible, keep switching active set until 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, IEEE TAC, 2016)

- Use NNLS to solve strictly convex QP

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

QP

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

complete the squares

$$\begin{aligned} Q &= L'L \\ M &= GL^{-1} \\ d &= b + GQ^{-1}c \end{aligned}$$

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

Least Distance Problem

(Lawson, Hanson, 1974)

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

Nonnegative Least Squares

residual $= 0$?

yes

QP problem infeasible

no

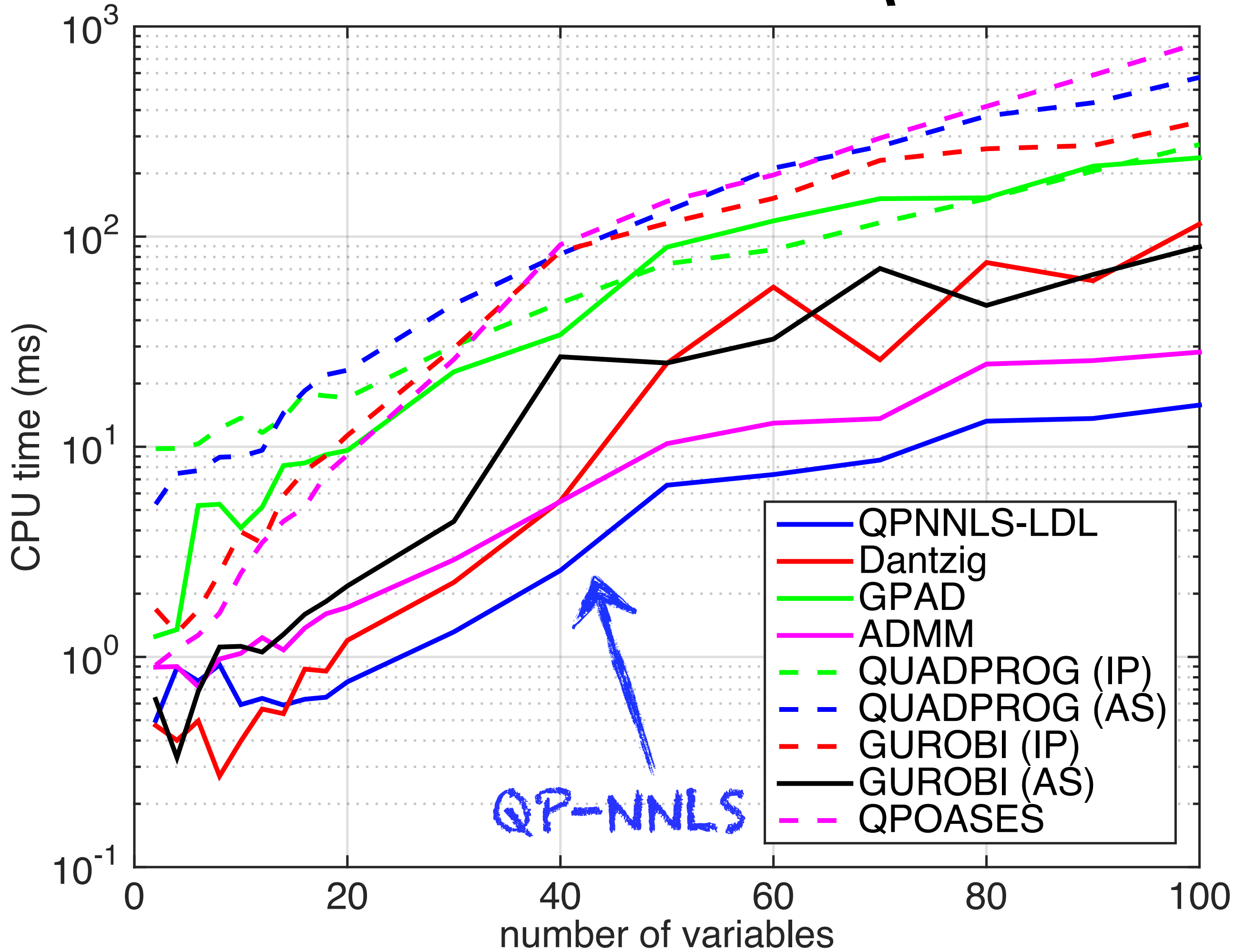
$$z^* = -\frac{1}{1 + d'y^*}L^{-1}M'y^* - Q^{-1}c$$

retrieve primal solution

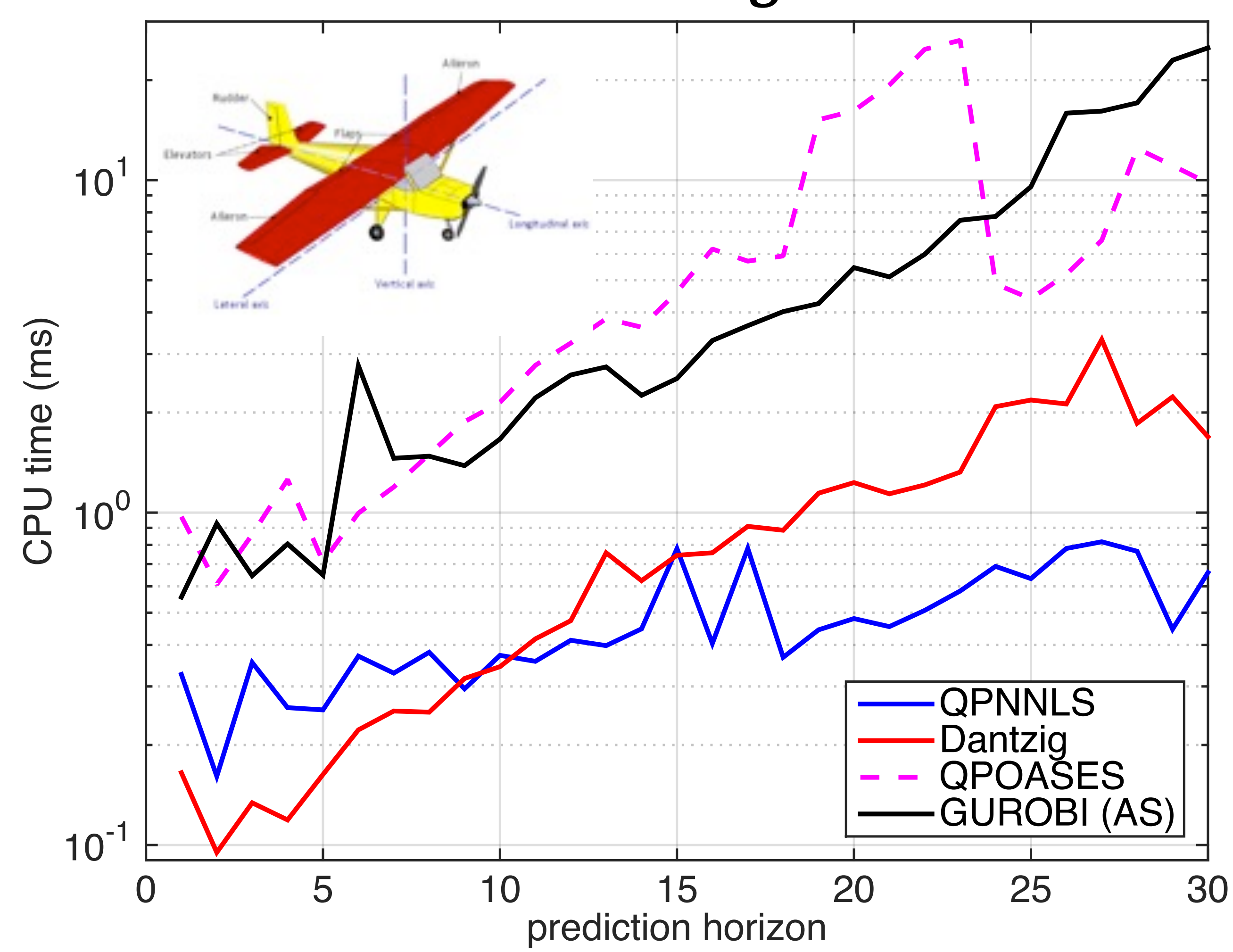
SOLVING QP VIA NNLS: NUMERICAL RESULTS

(Bemporad, IEEE TAC, 2016)

worst-case over 100 random QP instances



worst-case occurred during entire simulation*



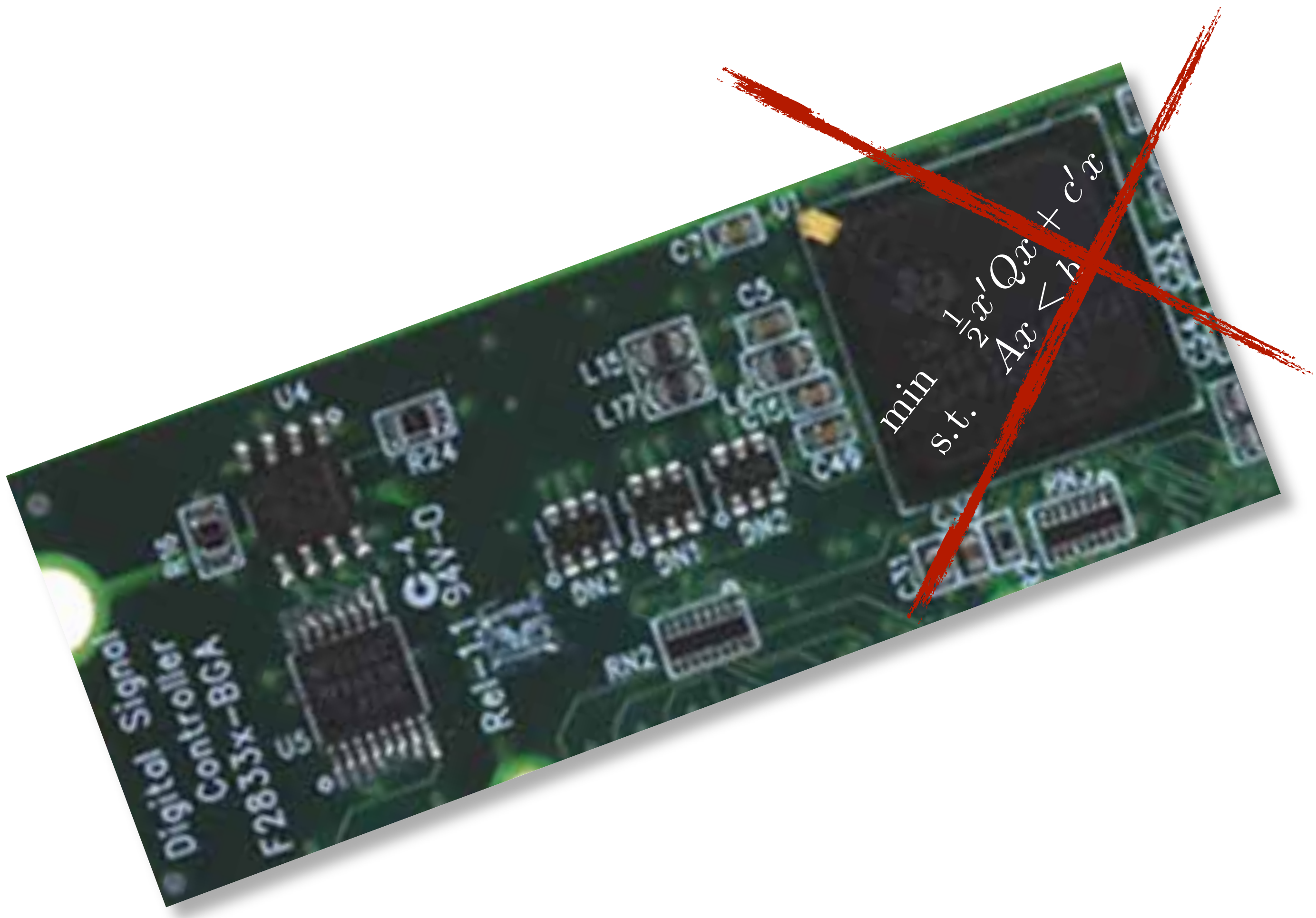
* Step t=0 not considered for QPOASES not to penalize the benefits of the method with warm starting

- A rather **fast** and relatively **simple-to-code** QP solver !

- Extended to solving mixed-integer QP’s (Bemporad, NMPC 2015)

EXPLICIT MPC

- Can we implement optimization-based controllers like MPC **without** an optimization solver running in real-time ?



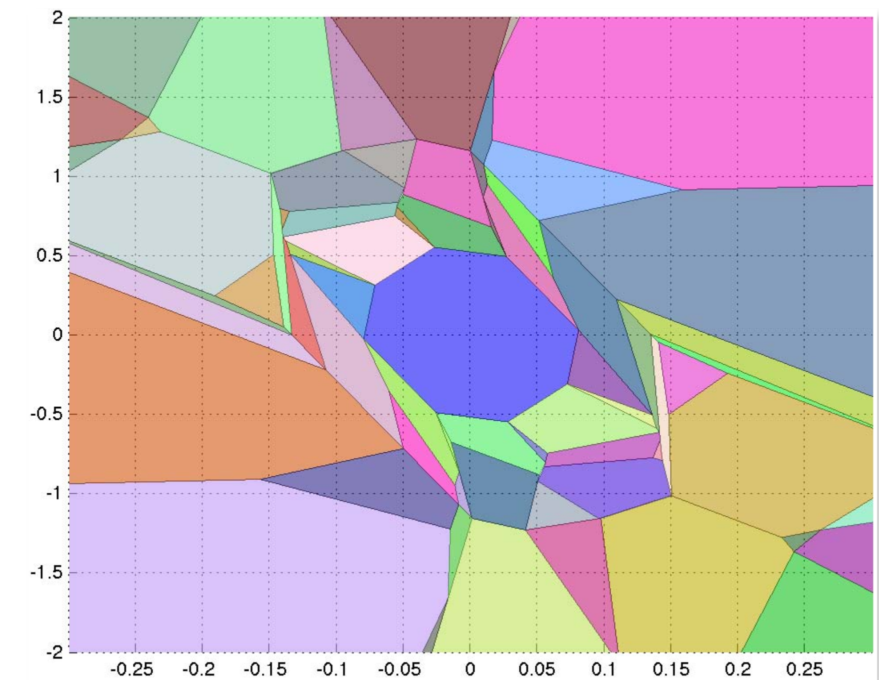
YES !

EXPLICIT MODEL PREDICTIVE CONTROL AND MULTIPARAMETRIC QP

(Bemporad, Morari, Dua, Pistikopoulos, 2002)

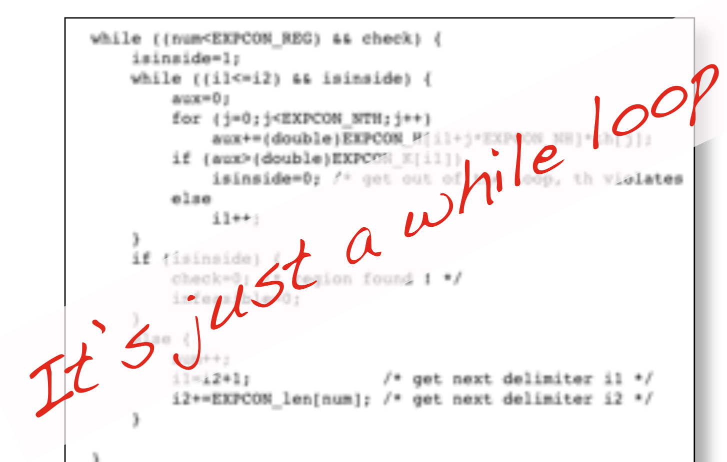
The multiparametric solution of a strictly convex QP is **continuous** and **piecewise affine**

$$z^*(x) = \arg\min_z \quad \frac{1}{2}z'Hz + x'F'z \qquad \text{s.t.} \quad Gz \leq W + Sx$$



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

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



NNLS FOR MULTIPARAMETRIC QP

- A variety of mpQP solvers is available

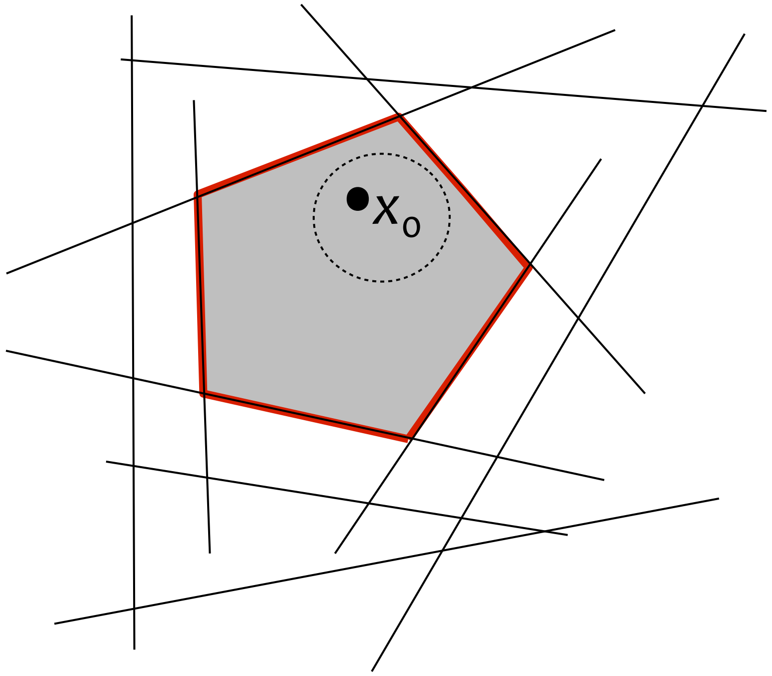
(Bemporad *et al.*, 2002) (Baotic, 2002)
(Tøndel, Johansen, Bemporad, 2003)
(Spjøtvold *et al.*, 2006)(Patrinos, Sarimveis, 2010)

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

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

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)**

NNLS FOR MULTIPARAMETRIC QP

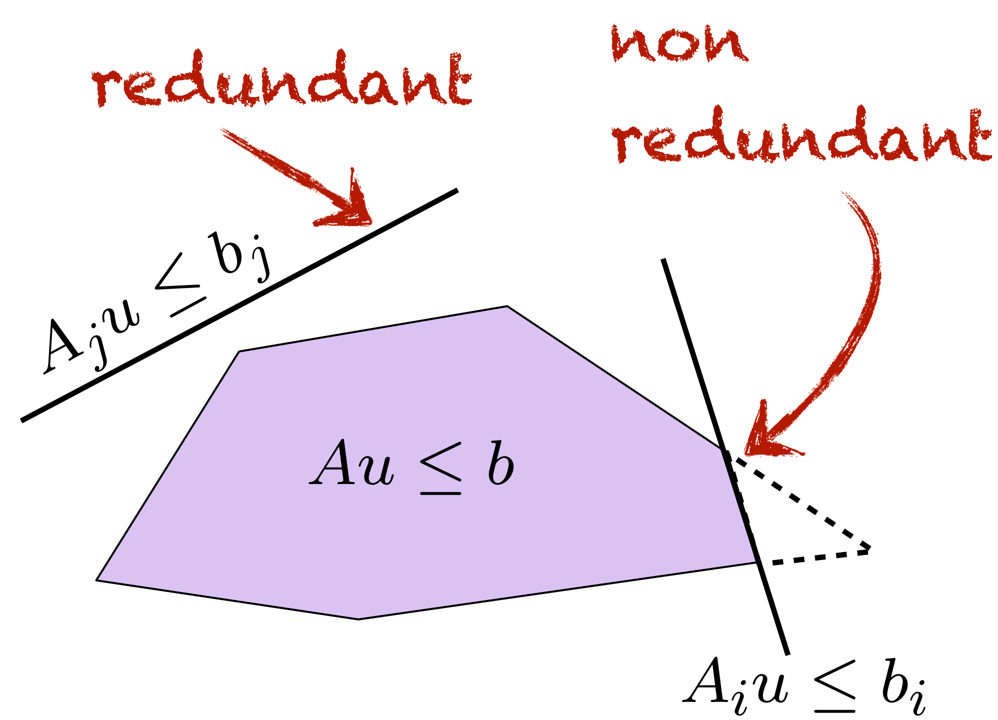
(Bemporad, IEEE TAC 2015)

- Key result:

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

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

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

NNLS FOR SOLVING MPQP PROBLEMS

(Bemporad, IEEE TAC, 2015)

- New mpQP algorithm based on **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)

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

- Included in MPC Toolbox 5.0 (≥R2014b)

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

OPTIMIZE DECISIONS UNDER UNCERTAINTY

renewable power

prices

water

demand

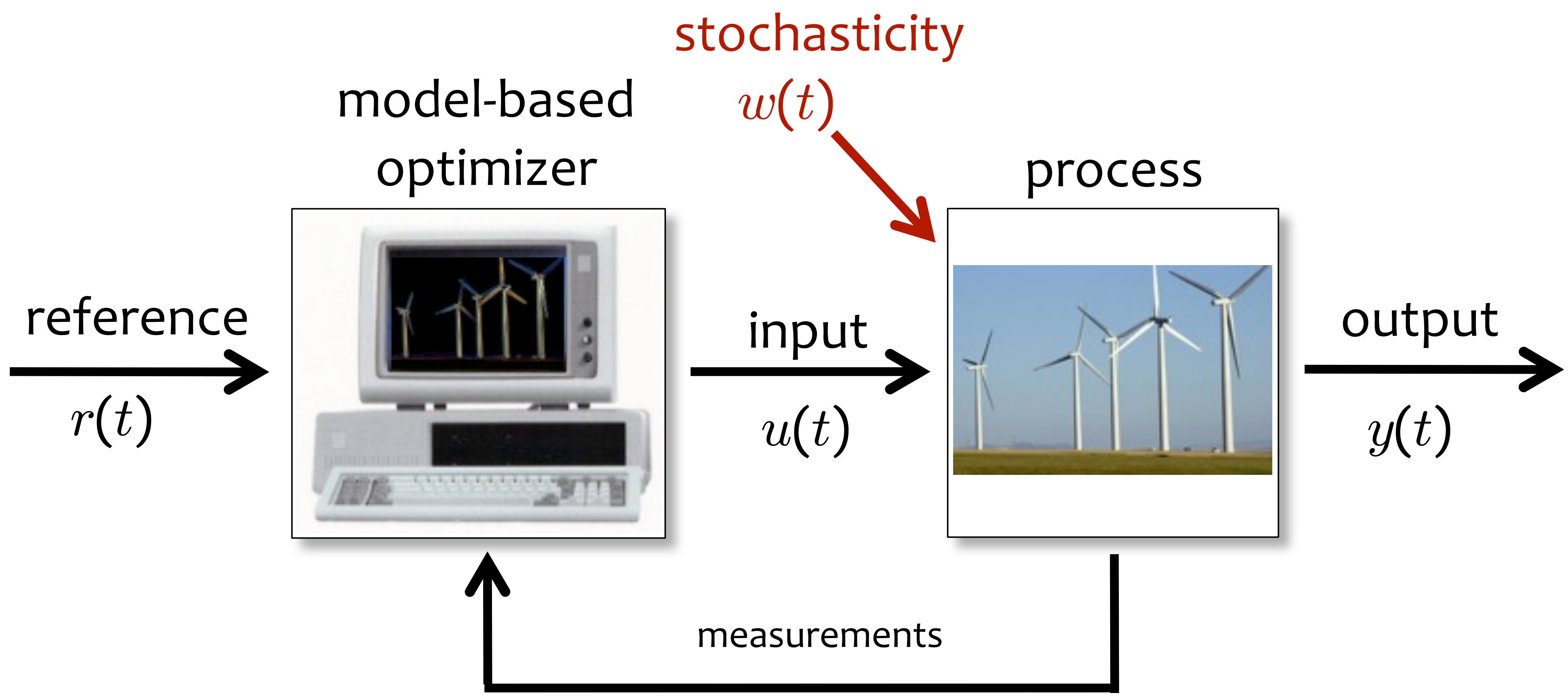
human (inter)action

- **Deterministic** (=certainty equivalence) approaches often inadequate (e.g.: portfolio management)

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

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

STOCHASTIC MODEL PREDICTIVE CONTROL (SMPC)



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

A FEW SAMPLE APPLICATIONS OF SMPC

- **Energy systems**: power dispatch in smart grids, optimal bidding on electricity markets

  (Patrinos, Trimboli, Bemporad 2011)

  (Puglia, Bernardini, Bemporad 2011)

- **Financial engineering**: dynamic hedging of portfolios replicating synthetic options

  (Bemporad, Bellucci, Gabbriellini, 2009)

  (Bemporad, Gabbriellini, Puglia, Bellucci, 2010)

  (Bemporad,Puglia, Gabbriellini, 2011)

- **Water networks**: pumping control in urban drinking water networks, under uncertain demand & minimizing costs under varying electricity prices

  (Sampathirao, Sopasakis, Bemporad, 2014)

- **Automotive control**: energy management in HEVs, adaptive cruise control (human-machine interaction)

  (Di Cairano, Bernardini, Bemporad, Kolmanovsky, 2014)

- **Networked control**: improve robustness against communication imperfections

  (Bernardini, Donkers, Bemporad, Heemels, NECSYS 2010)

LINEAR STOCHASTIC MPC W/ DISCRETE DISTURBANCE

- **Linear stochastic** prediction model

$$\begin{cases} x_{k+1} = A(w_k)x_k + B(w_k)u_k + f(w_k) \\ y_k = C(w_k)x_k + D(w_k)u_k + g(w_k) \end{cases}$$

(A,B,C,D) are can be sparse (ex: network of interacting subsystems)

- **Discrete disturbance** $w_k \in \{w^1, \dots, w^s\}$

$$p_j = \mathbf{Pr}[w_k = w^j]$$

$$p_j \geq 0, \quad \sum_{j=1}^{s} p_j = 1$$

Often w_k is low-dimensional (ex: electricity price, weather, etc.)

LINEAR STOCHASTIC MPC FORMULATION

Existing literature on stochastic MPC

(Schwarme & Nikolaou, 1999) (Munoz de la Pena, Bemporad, Alamo, 2005) (Oldewurtel, Jones, Morari, 2008)
(Wendt & Wozny, 2000) (Couchman, Cannon, Kouvaritakis, 2006) (Ono, Williams, 2008)
(Batina, Stoorvogel, Weiland, 2002) (Primbs, 2007) (van Hessem & Bosgra 2002) (Bemporad, Di Cairano, 2005)
(Bernardini, Bemporad, 2012)

- Performance index $$\min E_w \left[x_N' P x_N + \sum_{k=0}^{N-1} x_k' Q x_k + u_k' R u_k \right]$$

- Goal: ensure **mean-square convergence** $\lim_{t\to\infty} E[x'(t)x(t)] = 0$ (for $f(w(t))=0$)

- The existence of a **stochastic Lyapunov function** $V(x) = x'Px$

$$E_{w(t)}[V(x(t+1)] - V(x(t)) \leq -x(t)'Lx(t), \quad \forall t \geq 0 \qquad L = L' > 0$$

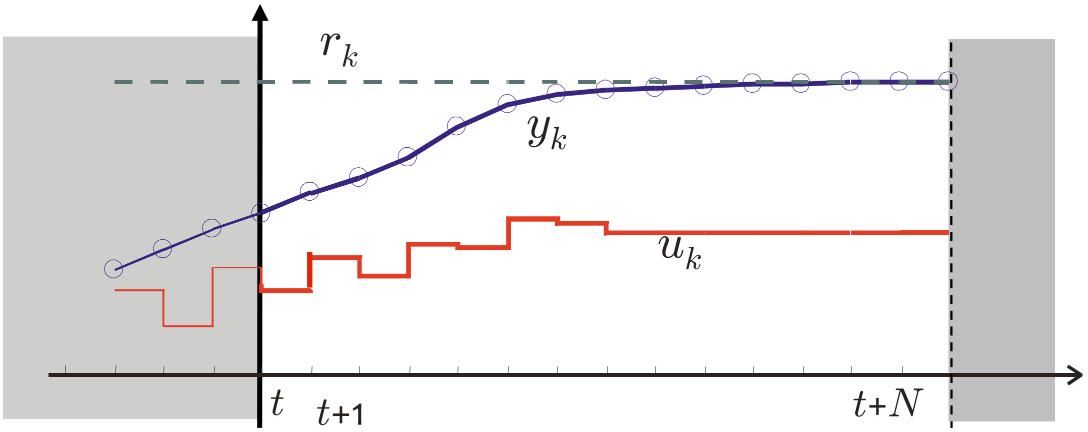
ensures mean-square stability

(Morozan, 1983) (Bernardini, Bemporad, 2012)

COST FUNCTIONS FOR SMPC TO MINIMIZE

- **Expected performance**

$$\min_{u} \sum_{k=0}^{N-1} E_w\left[(y_k - r_k)^2\right]$$



- Tradeoff between **expectation & risk**

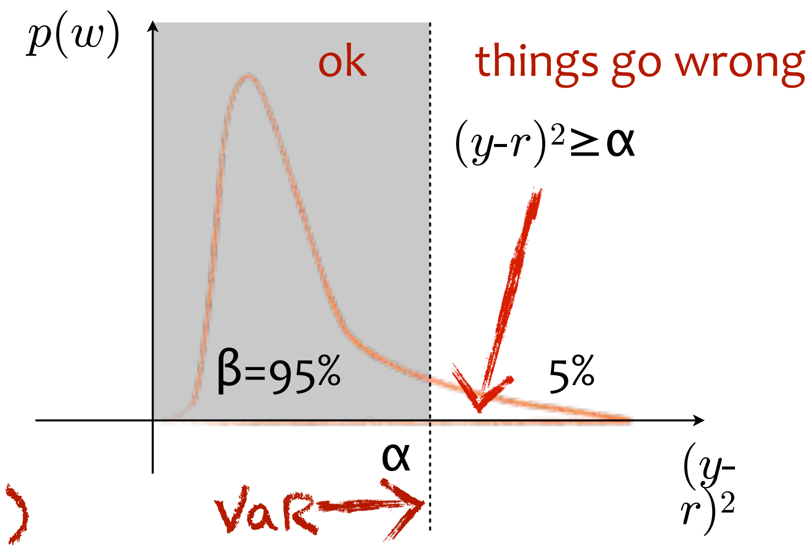
$$\min_{u} \sum_{k=0}^{N-1} (E_w\left[y_k - r_k\right])^2 + \alpha \mathrm{Var}_w\left[y_k - r_k\right] \qquad \alpha \geq 0$$

- Note that they coincide for α=1, since

$$\mathrm{Var}_w E\left[y_k - r_k\right] = E_w\left[(y_k - r_k)^2\right] - (E_w\left[y_k - r_k\right])^2$$

COST FUNCTIONS FOR SMPC TO MINIMIZE

- Conditional Value-at-Risk (**CVaR**) (Rockafellar, Uryasev, 2000)



$$\min_{u,\alpha} \sum_{k=0}^{N-1} \alpha_k + \frac{1}{1-\beta} E_w \left[\max \left\{ (y_k - r_k)^2 - \alpha_k, 0 \right\} \right]$$

= minimize expected loss when things go wrong (convex !)

can be cast to a linear programming problem

- **Min-max** = minimize worst case performance

$$\min_u \sum_{k=0}^{N-1} \max_w |y_k - r| + \rho |u_k|$$

can be cast to a linear programming problem

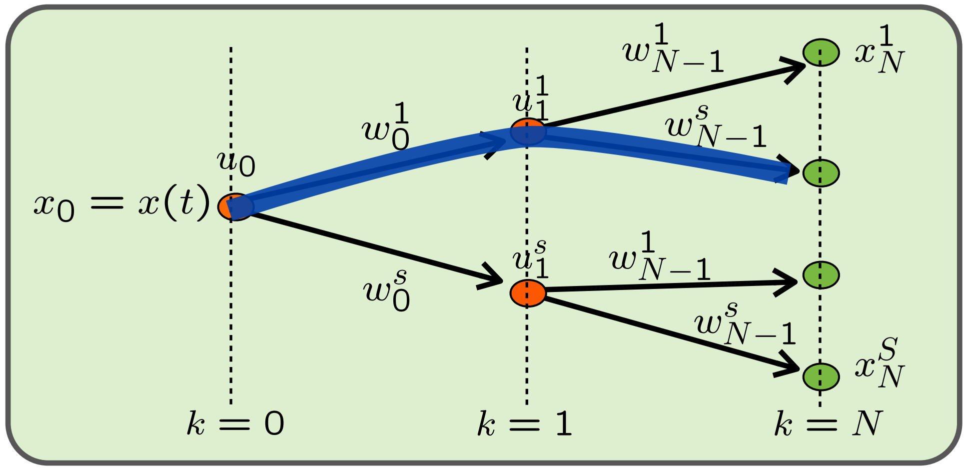
SCENARIO-BASED STOCHASTIC MPC

(Bernardini, Bemporad, 2012)

- Each scenario has its own evolution

$$x_{k+1}^j = A(w_k^j)x_k^j + B(w_k^j)u_k^j + f(w_k^j)$$

(=linear time-varying system)



- Expectations become simple sums !

Ex: $\min E_w \left[x_N' P x_N + \sum_{k=0}^{N-1} x_k' Q x_k + u_k' R u_k \right]$

$$\min \sum_{j=1}^{S} p^j \left((x_N^j)' P x_N^j + \sum_{k=0}^{N-1} (x_k^j)' Q x_k^j + (u_k^j)' R u_k^j \right)$$

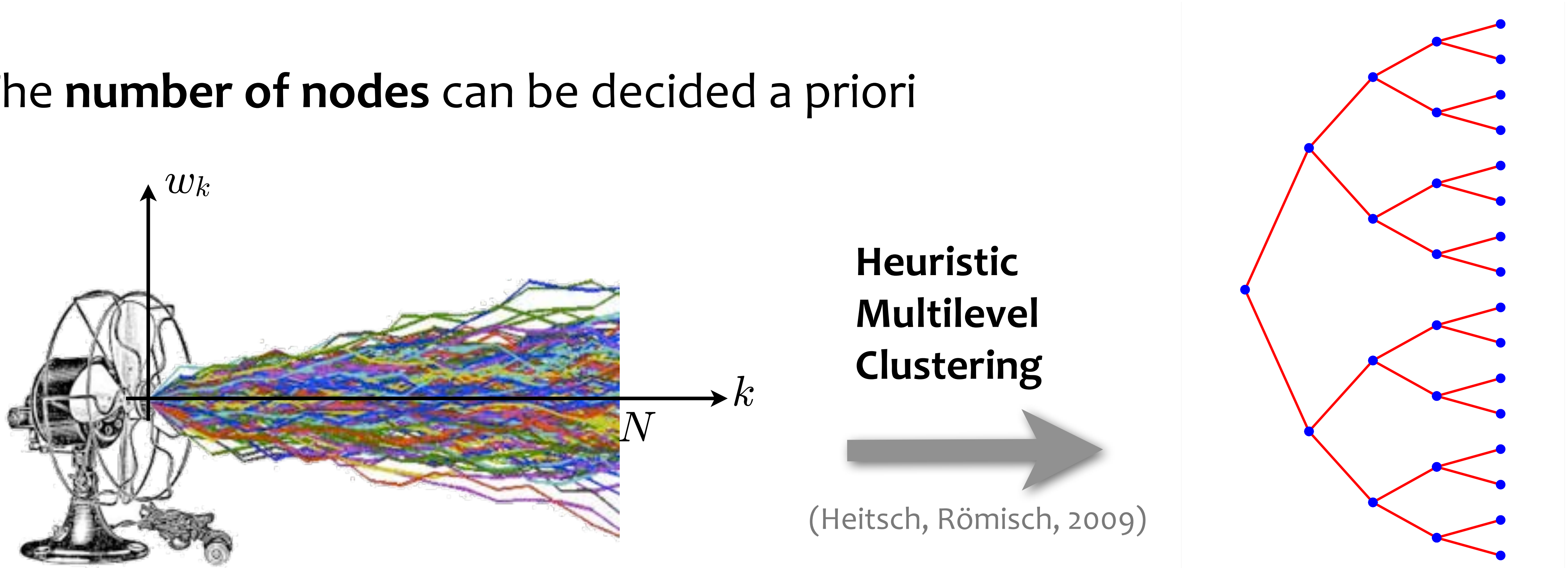
Expectations of quadratic costs remain quadratic costs

SCENARIO TREE GENERATION FROM DATA

- Scenario trees can be generated by **clustering** sample paths

- Paths can be obtained by **Monte Carlo simulation** of (estimated) models, or from **historical data**

- The **number of nodes** can be decided a priori



scenario "fan" (collection of sample paths)

scenario tree

- Alternative (simpler/less accurate) approach: **k-means** clustering

SMPC FOR MARKET-BASED OPTIMAL POWER DISPATCH

(Patrinos, Trimboli, Bemporad 2011)

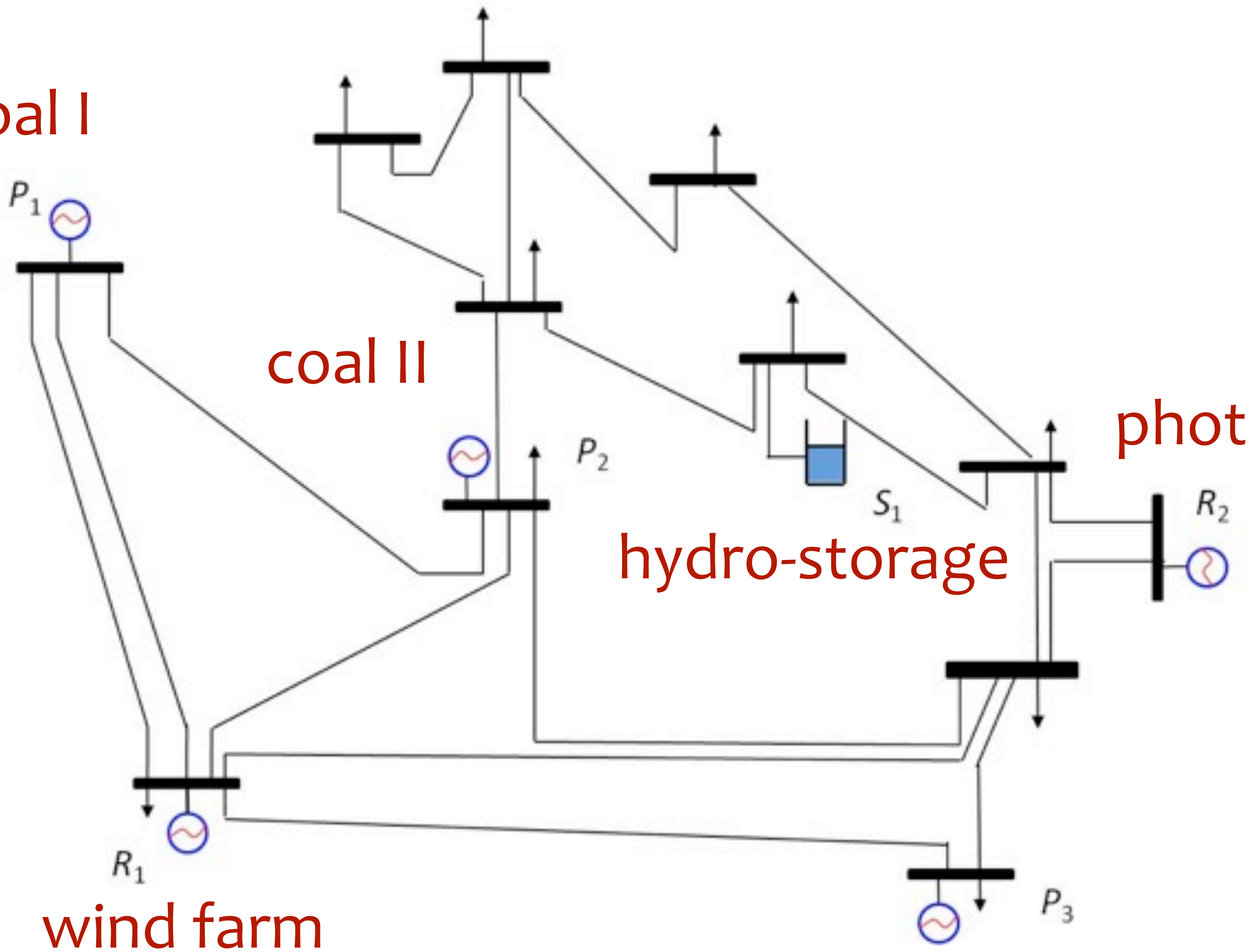


TABLE I: Generator Cost Data

Unit	Q_i (\$/MWh^2)	q_i (\$/MWh)	c_i (\$)
P1	0.009	30.375	398.025
P2	0.0225	73.35	292.275
P3	0.0488	61.488	489.952

TABLE II: Generator Data

Unit	$p_i^{\min}$	$p_i^{\max}$	$\Delta p_i^{\min}$	$\Delta p_i^{\min}$
P1	450	1100	-250	250
P2	50	500	-200	200
P3	50	100	-75	75

TABLE III: Storage Data

Unit	$x_i^{\min}$	$x_i^{\max}$	$\Delta x_i^{\min}$	$\Delta x_i^{\min}$	α_i	α_i^{c}	α_i^{d}
S1	15	300	-120	120	0.95	0.85	0.90
	$u_i^{c,\min} = u_i^{d,\min}$			$u_i^{c,\max} = u_i^{d,\max}$			
	0			300			

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

eprice
ENABLING THE FUTURE ENERGY SYSTEM

SEVENTH FRAMEWORK PROGRAMME

SMPC FOR MARKET-BASED OPTIMAL POWER DISPATCH

(Patrinos, Trimboli, Bemporad 2011)

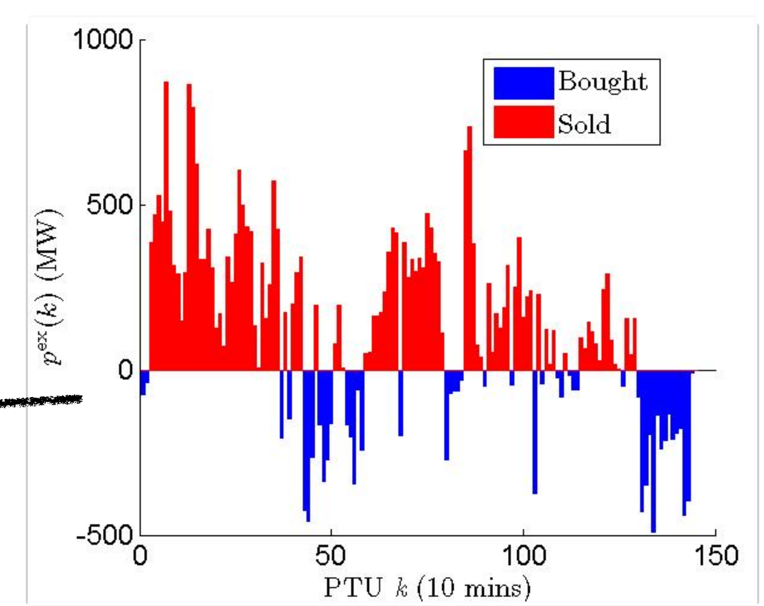
Exact knowledge of future uncertainty

Deterministic: time-dependent expectations used for future uncertainty

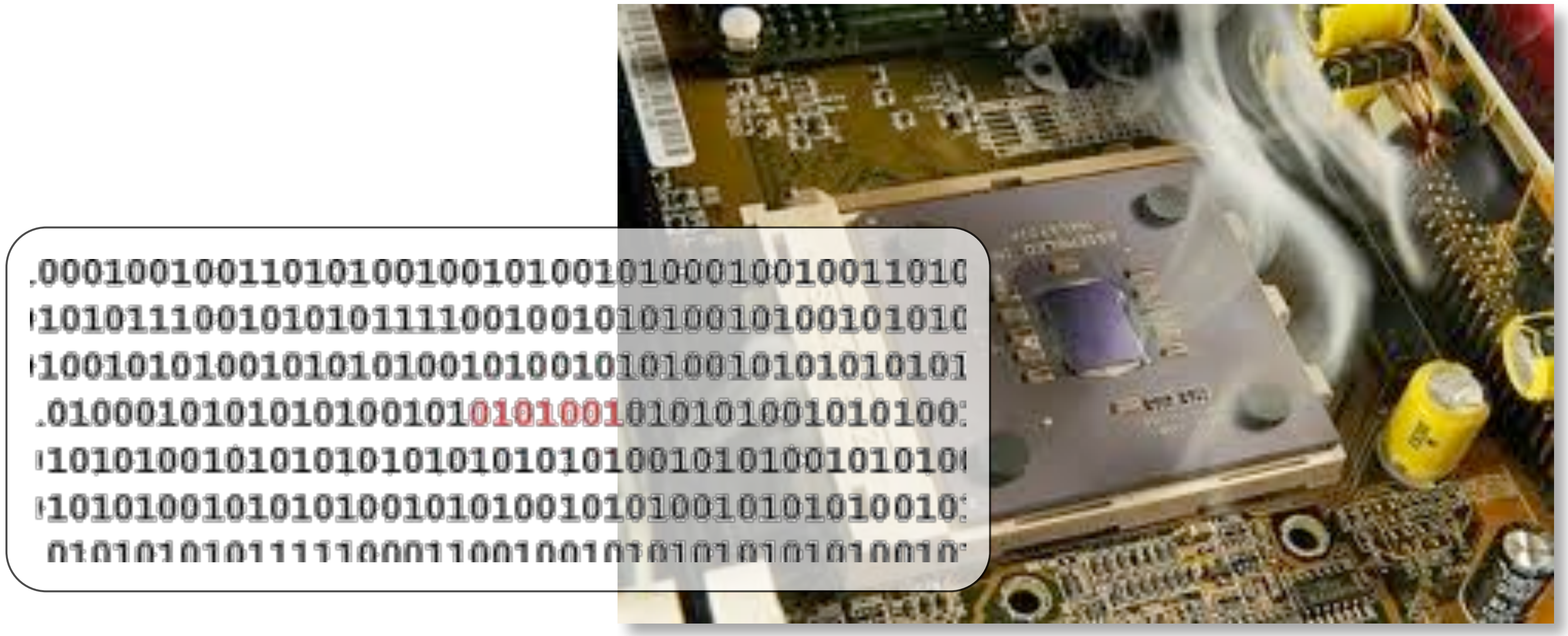
Stochastic formulation

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

power exchanged with grid

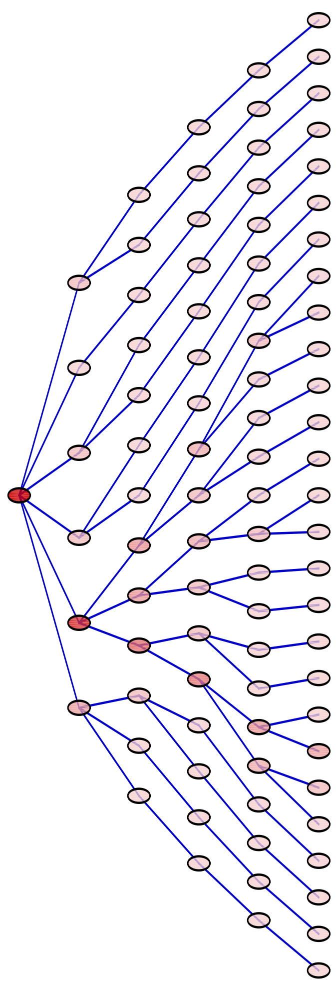


HOW ABOUT COMPUTATION COMPLEXITY ?



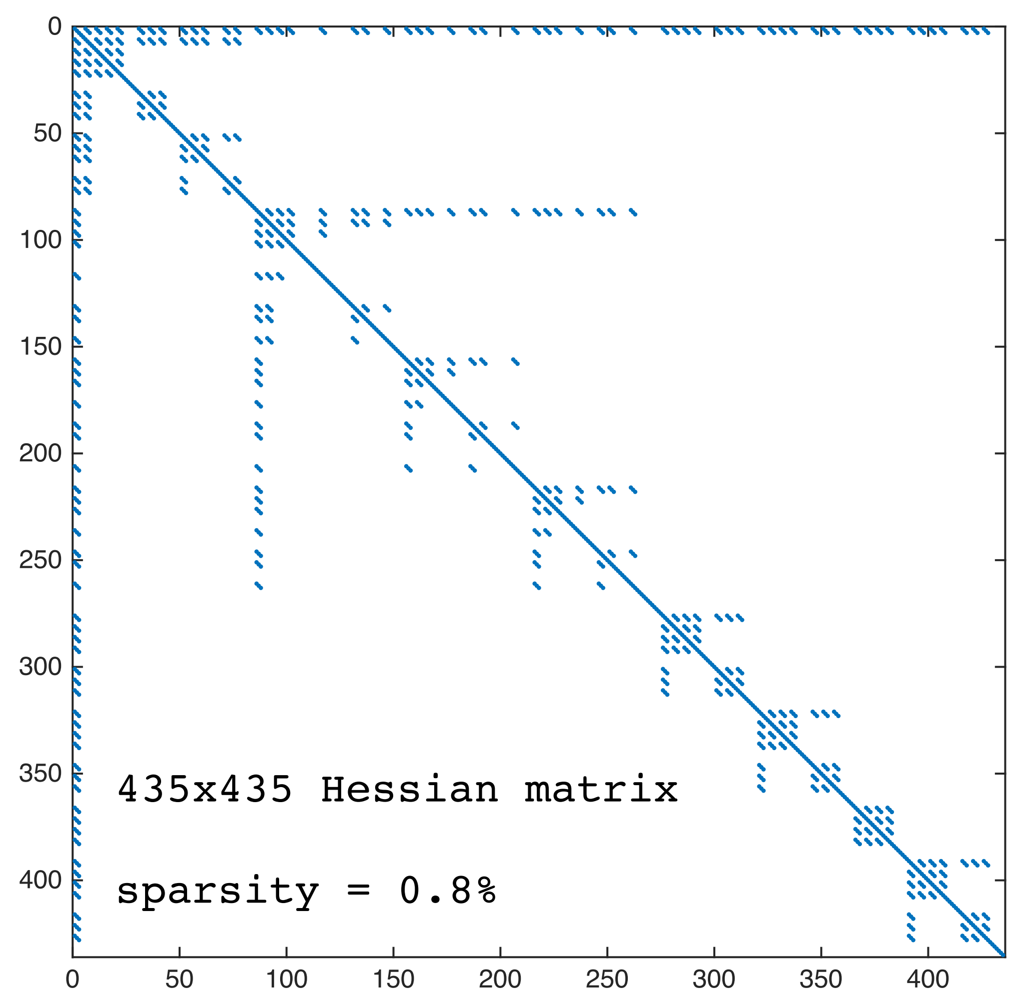
COMPLEXITY OF STOCHASTIC OPTIMIZATION PROBLEM

- #optimization variables = #nodes x #inputs (in condensed version)
- Problems are **very sparse** (well exploited by **interior point methods**)
- Example: SMPC with quadratic cost and linear constraints

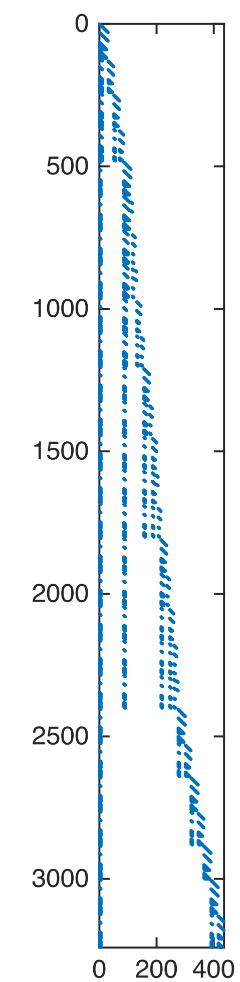


Tree=87 nodes

Branching factor M=[6 3 2 2 2]



435 free variables (5 inputs x node)



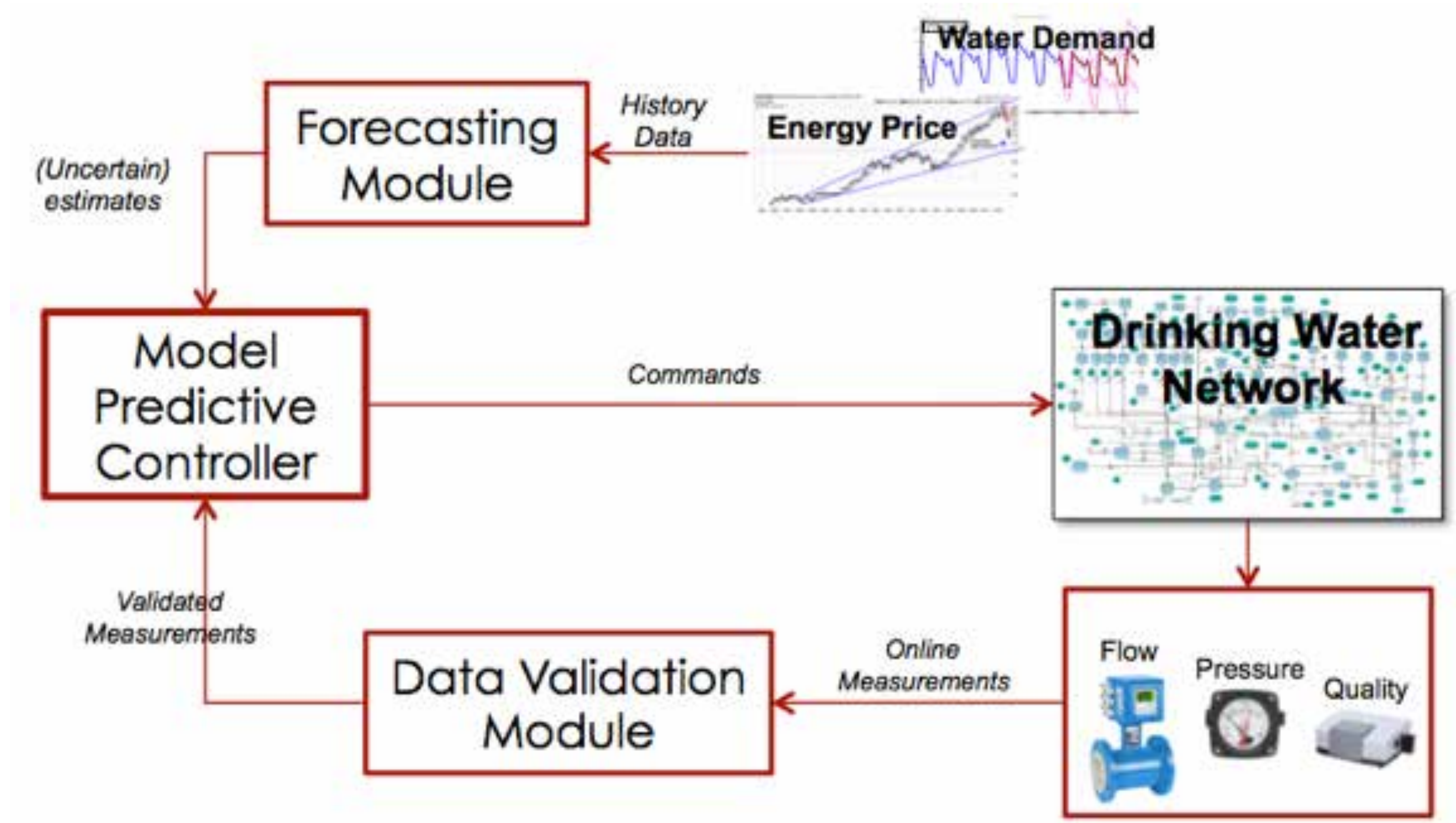
3240x435 constraint matrix
sparsity = 1.1%

SMPC OF THE DRINKING WATER NETWORKS

(Sampathirao, Sopasakis, Bemporad, Patrinos, 2015)

Main goals:

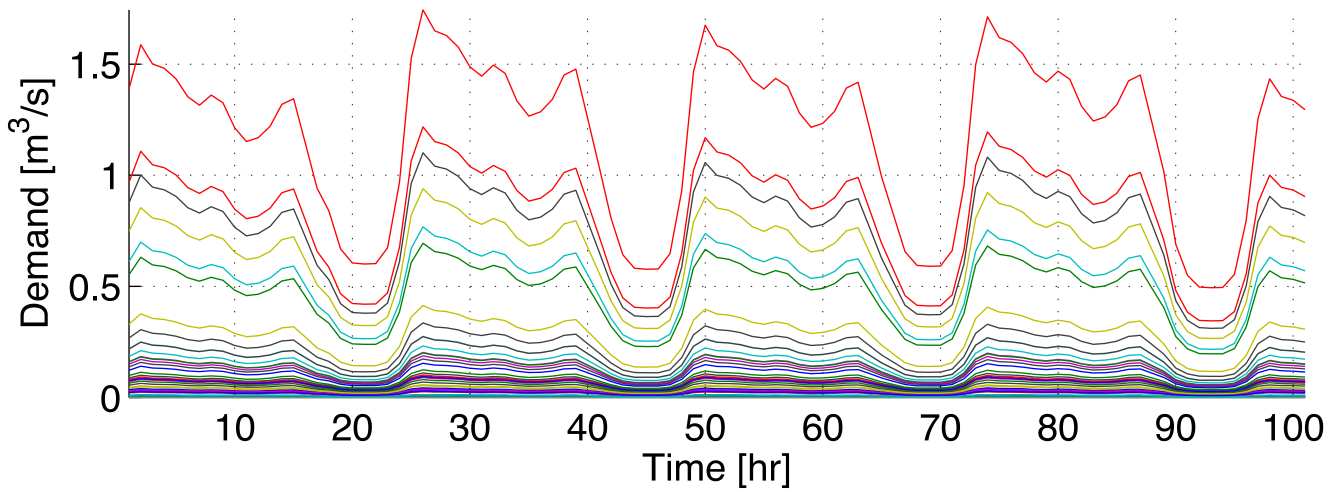
- Reduce **electricity consumption** for pumping (€€€)
- Meet **demand** requirements
- Deliver **smooth** control actions
- Keep storage tanks above safety **limits**
- Respect the technical **limitations:** pressure limits, overflow limits & pumping capabilities



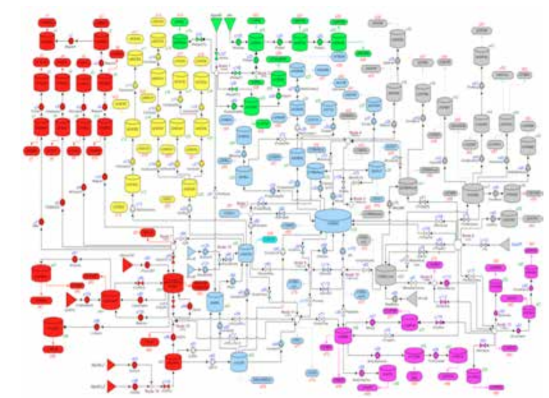
EFFINET

STOCHASTIC MPC AND PARALLEL COMPUTATIONS ON GPU

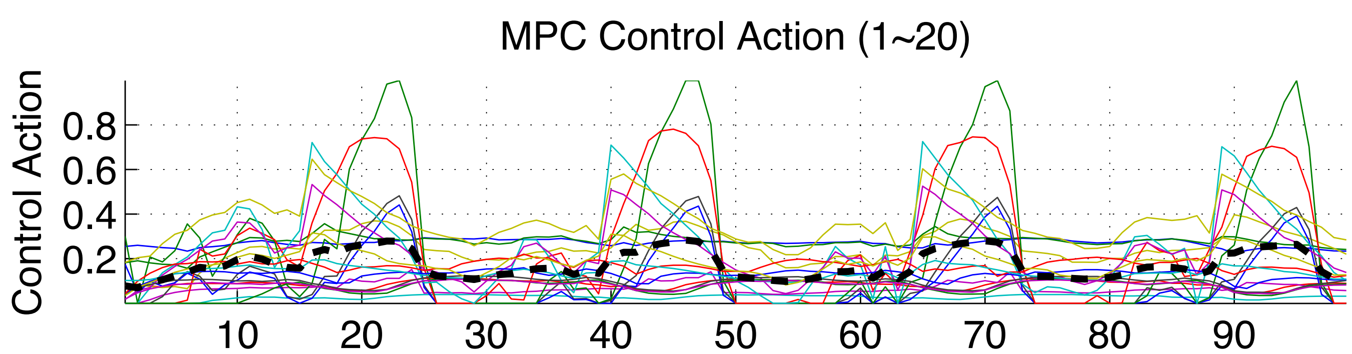
(Sampathirao, Sopasakis, Bemporad, Patrinos, 2016)



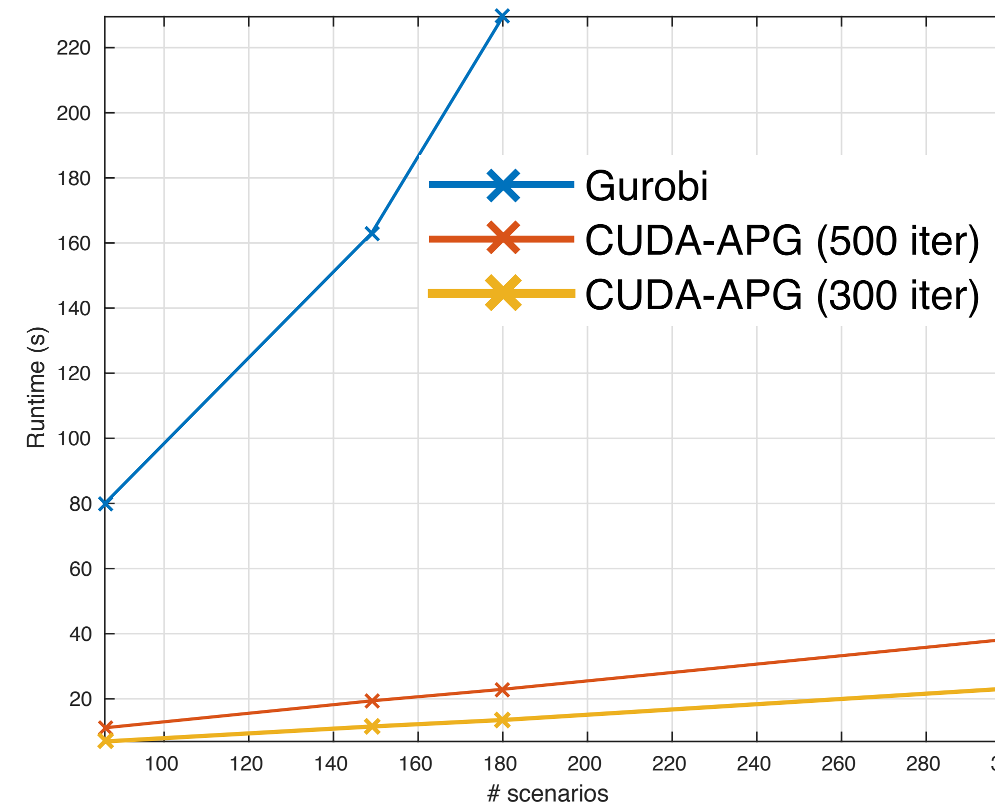
Drinking water network of Barcelona:



63 tanks
114 controlled flows
17 mixing nodes



CPU time (s)



MPC-controlled network:

- Minimum pressure requirement hardly violated
- ~5% savings on energy cost w.r.t. current practice
- Smooth control actions
- sampling time = 1 hour

APG = Accelerated Proximal Gradient, parallel implemented on NVIDIA Tesla 2075 CUDA platform

FP7-ICT project "**EFFINET - Efficient Integrated Real-time Monitoring and Control of Drinking Water Networks**" (2012-2015)

EFFINET

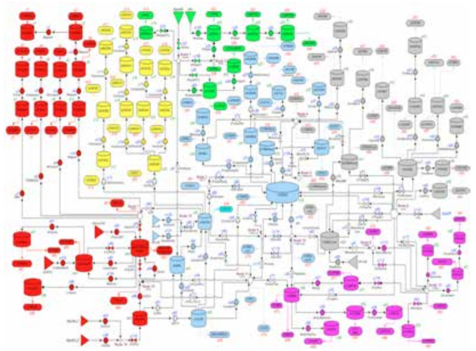
CONCLUSIONS

- MPC can easily handle **multivariable control** problems with **constraints** in an **optimized** way, it's **easy to design** and **reconfigure,** it handles **uncertainty**

- Long history of success in the *process industries*

Is linear MPC really a mature technology for production in fast embedded applications ?

YES !



Is stochastic MPC mature too ?

WE'RE VERY CLOSE ...