Explicit MPC: Basics, Fast Implementations, Advantages and Limitations

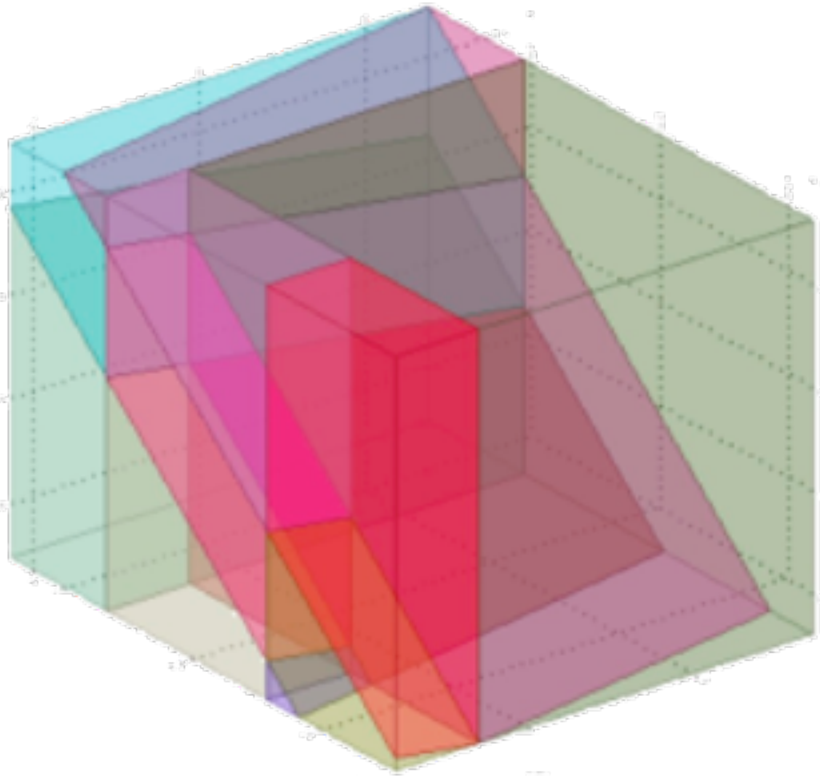
Alberto Bemporad

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

IMT INSTITUTE FOR ADVANCED STUDIES LUCCA

Outline

- Formulation of linear MPC design framework
- Multiparametric quadratic programming (explicit MPC)
- An automotive control example
- Fast circuit implementations
- Final conclusions (explicit vs implicit MPC)



Finite-time optimal control of linear systems

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

$R = R' \succ 0$

$Q = Q' \succeq 0$

$P = P' \succeq 0$

performance index $\min_{u_0,\ldots,u_{N-1}} x_N' P x_N + \sum_{k=0}^{N-1} x_k' Q x_k + u_k' R u_k$

$$\begin{array}{ll} \min_z & \frac{1}{2} z' H z + x_0' F' z + \cancel{\frac{1}{2} x_0' Y x_0} \\ \text{s.t.} & Gz \leq W + Sx_0 \end{array}$$

$H \succ 0$

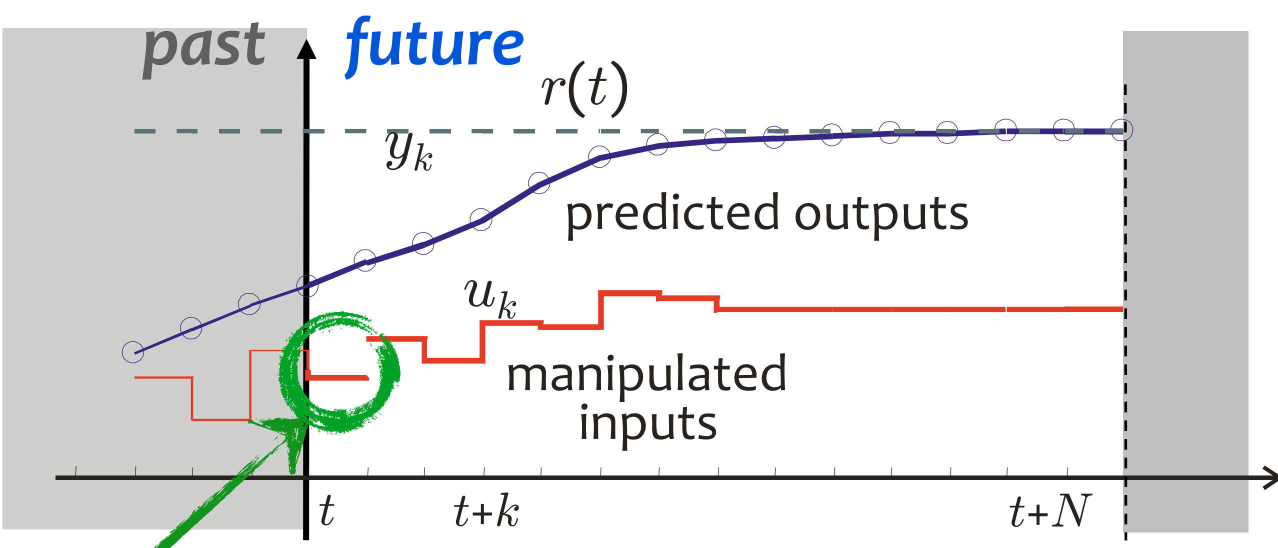
$$z = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}$$

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

(convex) Quadratic Program (QP)

Linear MPC algorithm

@ each sampling step t:



- Measure (or estimate) the current state $x(t)$

feedback !

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

$$\min_z \quad \frac{1}{2} z' H z + x'(t) F' z$$
$$\text{s.t.} \quad Gz \leq W + Sx(t)$$

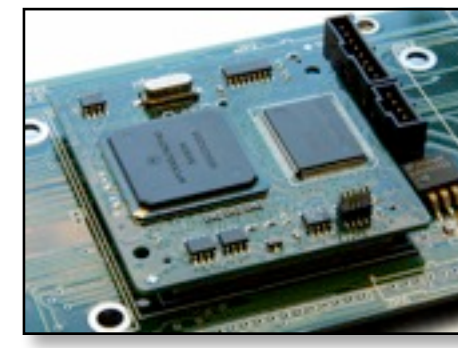
- Apply only $u(t) = u_0^*$, discard remaining optimal inputs $u_1^*, \ldots, u_{N-1}^*$

Routinely used in **process control**, now spreading in **automotive/aerospace**

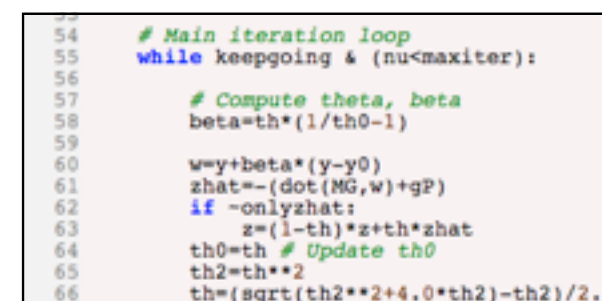
Requirements for embedded MPC

1. **Speed**: fast enough to provide a solution within short sampling intervals (such as 10-100 ms)

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



3. **Code simple** enough to be **verifiable/certifiable**



4. **Worst-case execution time** must be (tightly) estimated for embedding MPC in a **real-time platform**

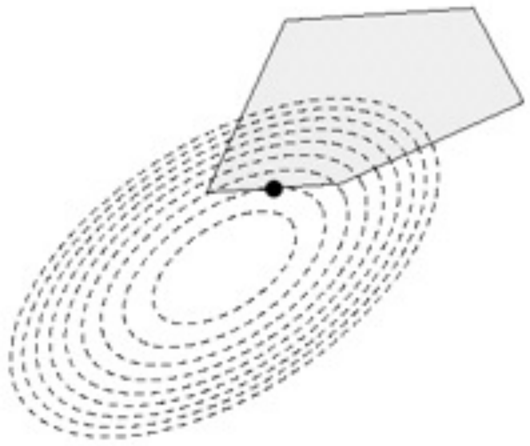
Computing the MPC command action

Algorithms for solving QP **on-line** given $x(t)$:

- *active set (AS) methods*
- *interior point (IP) methods*
- *gradient projection (GP) methods*
- *conjugate gradient (CG) methods*
- *alternating direction method of multipliers (ADMM)*

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

Quadratic Program (QP)



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

Alternative: solve the QP **off-line** **for all** $x(t)$ to find the control law u=$u(x)$ **explicitly** via **multiparametric programming**

Offline multiparametric QP algorithm

(Bemporad, Morari, Dua, Pistikopoulos, 2002)

Fix a point $x_0 \in X$ in the parameter space

$$\min_z \quad \frac{1}{2}z'Hz + x'F'z$$
$$\text{s.t.} \quad Gz \leq W + Sx$$

- solve QP to find $z^*(x_0), \;\; \lambda^*(x_0)$
- identify active constraints at $z^*(x_0)$
- form matrices $\tilde{G}, \tilde{W}, \tilde{S}$ by collecting active constraints: $\tilde{G}z^*(x_0) - \tilde{W} - \tilde{S}x_0 = 0$

KKT optimality conditions:

$$\begin{array}{llll} (1) & Hz + Fx + G'\lambda = 0 & (2) & \tilde{G}z - \tilde{W} - \tilde{S}x = 0 \\ (3) & \lambda_i(G^i z - W^i - S^i x) = 0 & (4) & \hat{G}z \leq \hat{W} + \hat{S}x \\ (5) & \tilde{\lambda}_i \geq 0, \;\; \hat{\lambda}_i = 0 & & \end{array}$$

$\hat{G}$=rows of G not in $\tilde{G}$ (inactive constraints)

From (1) :

$$z = -H^{-1}(Fx + \tilde{G}'\tilde{\lambda})$$

From (2) :

$$\boxed{\begin{array}{l} \tilde{\lambda}(x) = -(\tilde{G}H^{-1}\tilde{G}')^{-1}(\tilde{W} + (\tilde{S} + \tilde{G}H^{-1}F)x). \\ z(x) = H^{-1}\left[\tilde{G}'(\tilde{G}H^{-1}\tilde{G}')^{-1}(\tilde{W} + (\tilde{S} + \tilde{G}H^{-1}F)x) - Fx\right] \end{array}}$$

- In some neighborhood of x_0, λ and z are **explicit affine functions** of x

(Zafiriou, 1990)

Multiparametric QP algorithm

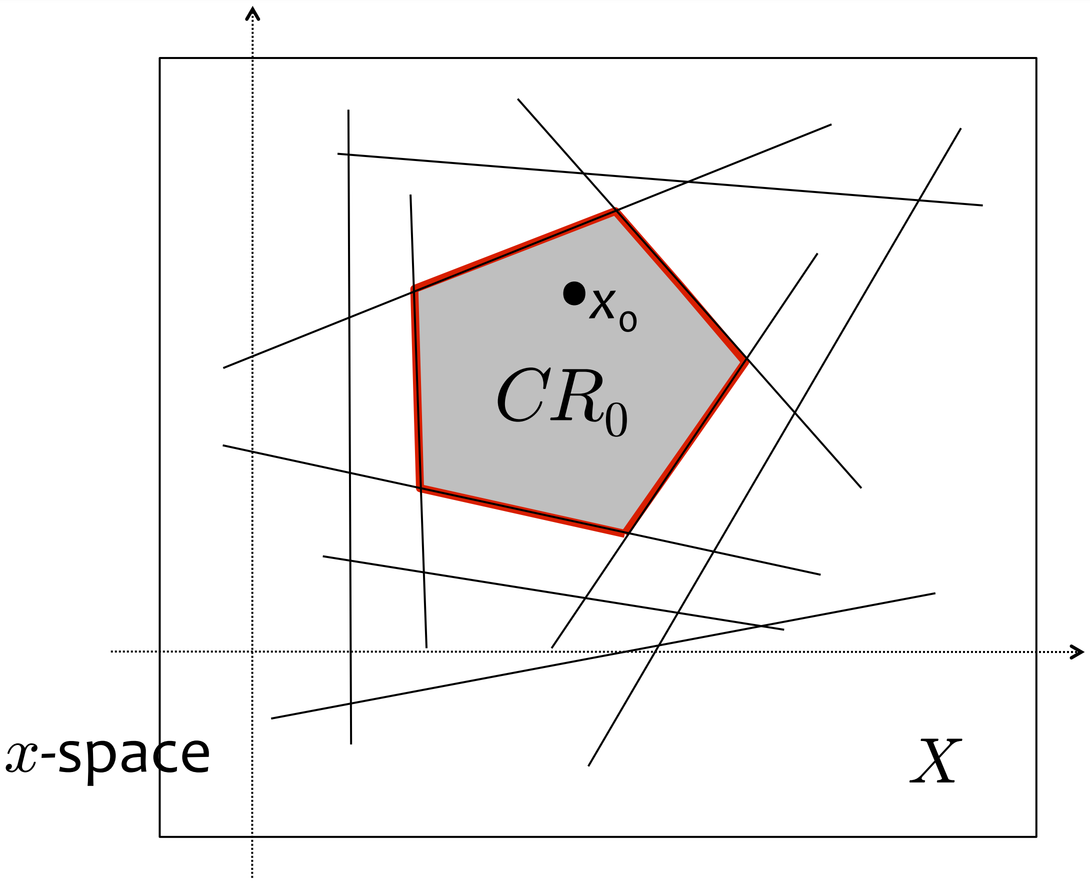
- Impose primal and dual feasibility:

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

from (4)

from (5)

→ linear inequalities in x !



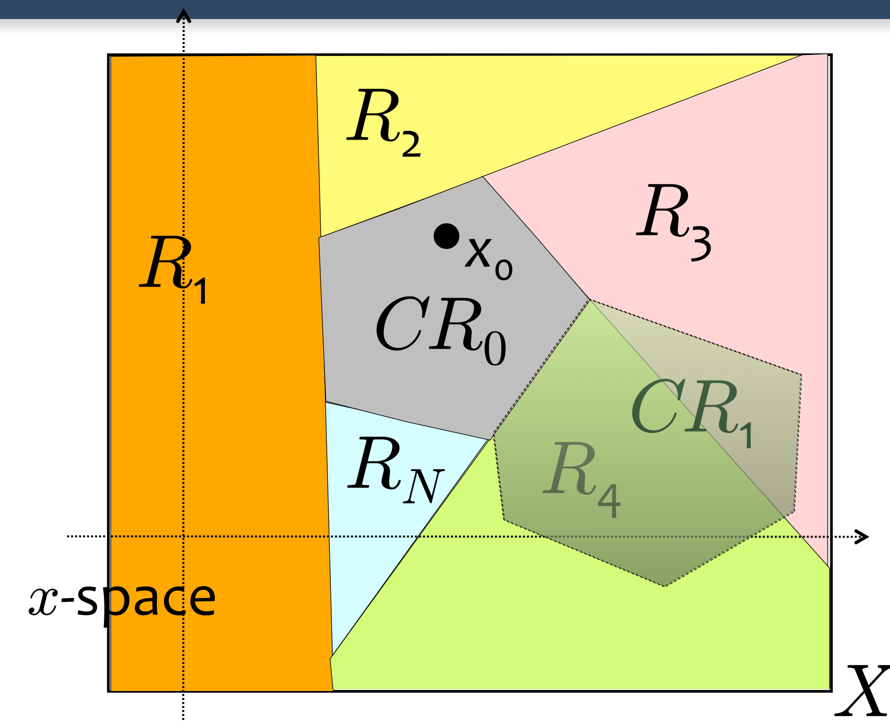
- Remove redundant constraints (this requires solving LP's):

→ critical region CR_0

$$CR_0 = \{x \in X : \ \mathcal{A}x \leq \mathcal{B}\}$$

- CR_0 is the set of all and only parameters x for which $\tilde{G}$, $\tilde{W}$, $\tilde{S}$ is the optimal combination of active constraints at the optimizer

Multiparametric QP solvers

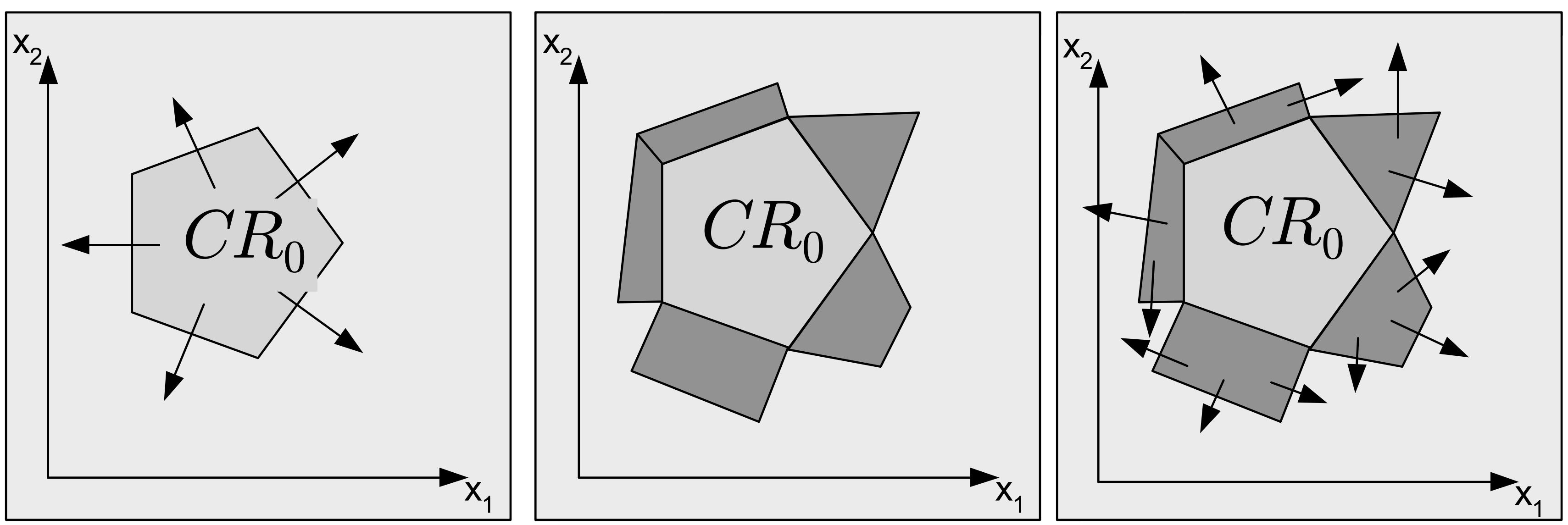


Method #1: Split and proceed iteratively

(Bemporad, Morari, Dua, Pistikopoulos, 2002)

Method #2: Split and proceed iteratively

(Tøndel, Johansen, Bemporad, 2003)



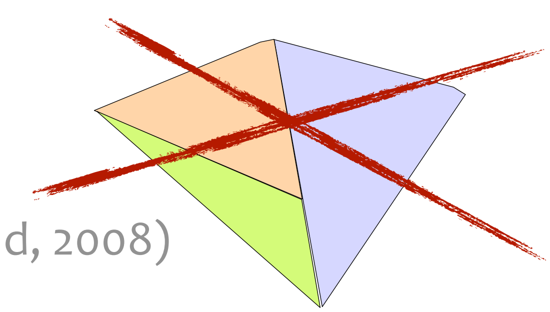
add constraint $\#i$

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

remove constraint $\#j$

Method #3: exploit the *facet-to-facet* property

(Spjøtvold, Kerrigan, Jones, Tøndel, Johansen, 2006)



(Spjøtvold, 2008)

Step out ϵ outside each facet, solve QP, get new region, iterate. (Baotic, 2002)

Properties of multiparametric QP

Theorem 1 *Consider a multi-parametric quadratic program with* $H \succ 0$, $\left[\begin{smallmatrix} H & F \\ F' & Y \end{smallmatrix}\right] \succeq 0$. *The set* X^* *of parameters* x *for which the problem is feasible is a polyhedral set, the value function* $V^* : X^* \mapsto \mathbb{R}$ *is piecewise quadratic, convex and continuous and the optimizer* $z^* : X^* \to \mathbb{R}^s$ *is piecewise affine and continuous.*

(Bemporad, Morari, Dua, Pistikopoulos, 2002)

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

continuous,
piecewise affine

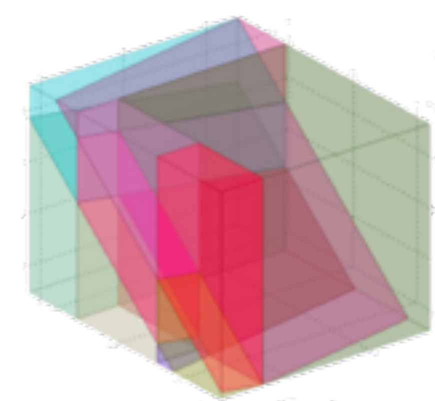
$$V^*(x) = \tfrac{1}{2} x'Yx + \min_z \ \tfrac{1}{2} x'Hz + x'F'z \quad \text{s.t. } Gz \le W + Sx$$

convex, continuous,
piecewise quadratic,
$\mathcal{C}^1$ (if no degeneracy)

Corollary: The linear MPC controller is a continuous piecewise affine function of the state

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

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



Applicability of explicit MPC approach

- Consider the following general MPC formulation

$$\min_{z} \quad \sum_{k=0}^{N-1} \frac{1}{2}(y_k - r(t+k)'S(y_k - r(t+k)) + \frac{1}{2}\Delta u_k' T \Delta u_k + (u_k - u_r(t+k))'V(u_k - u_r(t+k))' + \rho_\epsilon \epsilon^2$$

$$\begin{aligned}
\text{subj. to} \quad & x_{k+1} = Ax_k + Bu_k + B_v v(t+k), \ k = 0, \ldots, N-1 \\
& y_k = Cx_k + Du_k + D_v v(t+k), \ k = 0, \ldots, N-1 \\
& u_{\min}(t+k) \leq u_k \leq u_{\max}(t+k), \ k = 0, \ldots, N-1 \\
& \Delta u_{\min}(t+k) \leq \Delta u_k \leq \Delta u_{\max}(t+k), \ k = 0, \ldots, N-1 \\
& y_{\min}(t+k) - \epsilon V_{\min} \leq y_k \leq y_{\max}(t+k) + \epsilon V_{\max}, \ k = 1, \ldots, N
\end{aligned}$$

- Everything marked in red can be **time-varying** in explicit MPC

- Not applicable to time-varying problems (weights and/or system matrices)

- Can be extended to **hybrid systems** (continuity of control law may be lost)
  (Bemporad, Borrelli, Morari, 2001) (Mayne, Rakovic, 2002) (Mayne, ECC 2001)
  (Borrelli, Baotic, Bemporad, Morari, 2005) (Alessio, Bemporad, 2006)

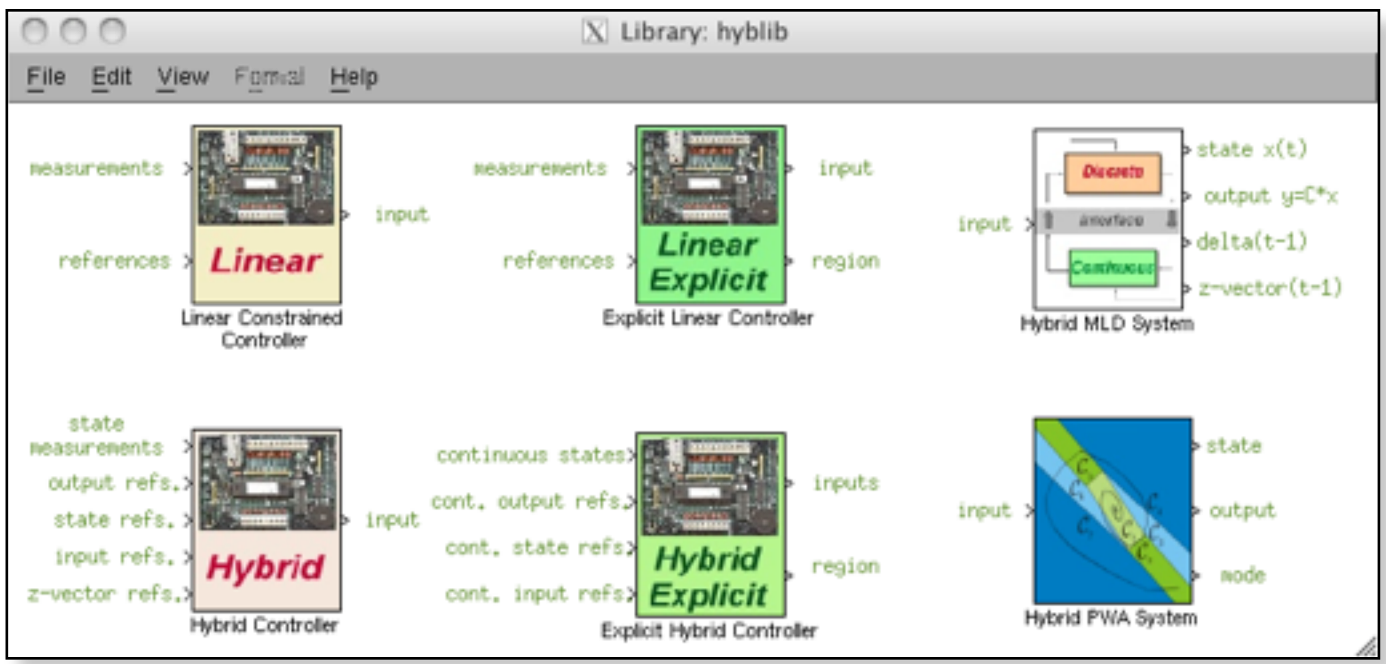
Hybrid Toolbox for MATLAB

(Bemporad, 2003-2013)

Supported by

Features:

- **Explicit** MPC control (via multi-parametric programming) and C-code generation
- **Hybrid** models: design, simulation, verification



5000+ download requests since October 2004

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

Other good tools: **Multiparametric Toolbox** (see next talk !)

Explicit MPC for idle speed control

(Di Cairano, Yanakiev, Bemporad, Kolmanovsky, Hrovat, 2011)

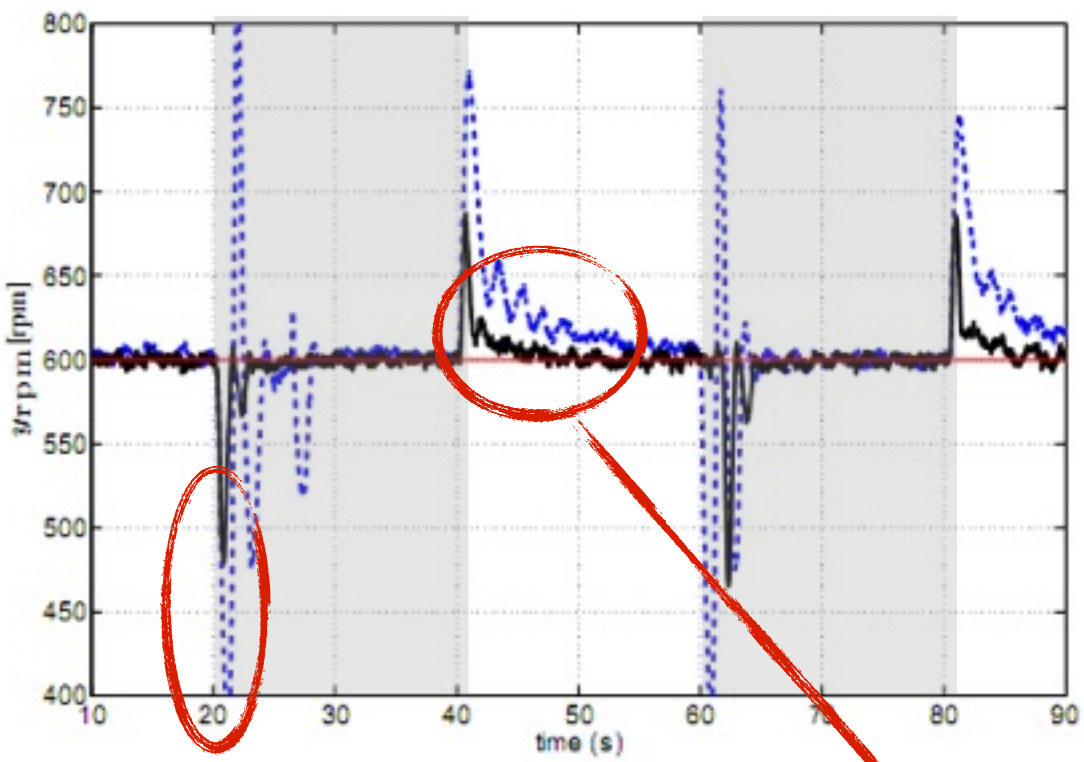
- Ford pickup truck, V8 4.6L gasoline engine

- Process:
  - *1 output* (engine speed) to regulate
  - *2 inputs* (airflow, spark advance)
  - input *delays*

- Objectives and specs:
  - **regulate engine speed** at constant rpm
  - **saturation limits** on airflow and spark
  - **lower bound** on engine speed ($\geq$450 rpm)

Explicit MPC for idle speed - Experiments

(Di Cairano, Yanakiev, Bemporad, Kolmanovsky, Hrovat, 2011)



Load torque (power steering)

peak reduced by 50%

convergence 10s faster

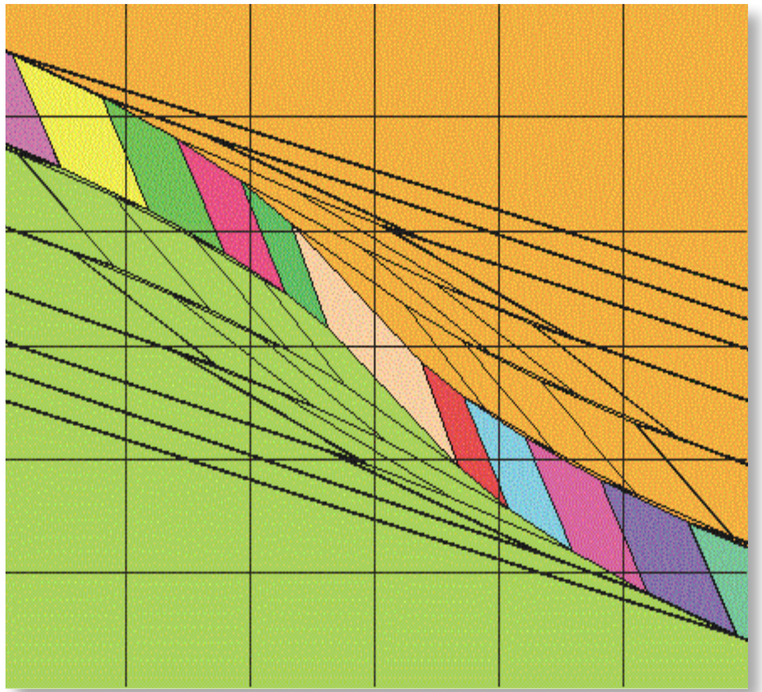
- Sampling time = 30 ms
- Explicit MPC implemented in dSPACE MicroAutoBox rapid prototyping unit
- Observer tuning as much important as tuning of MPC weights !

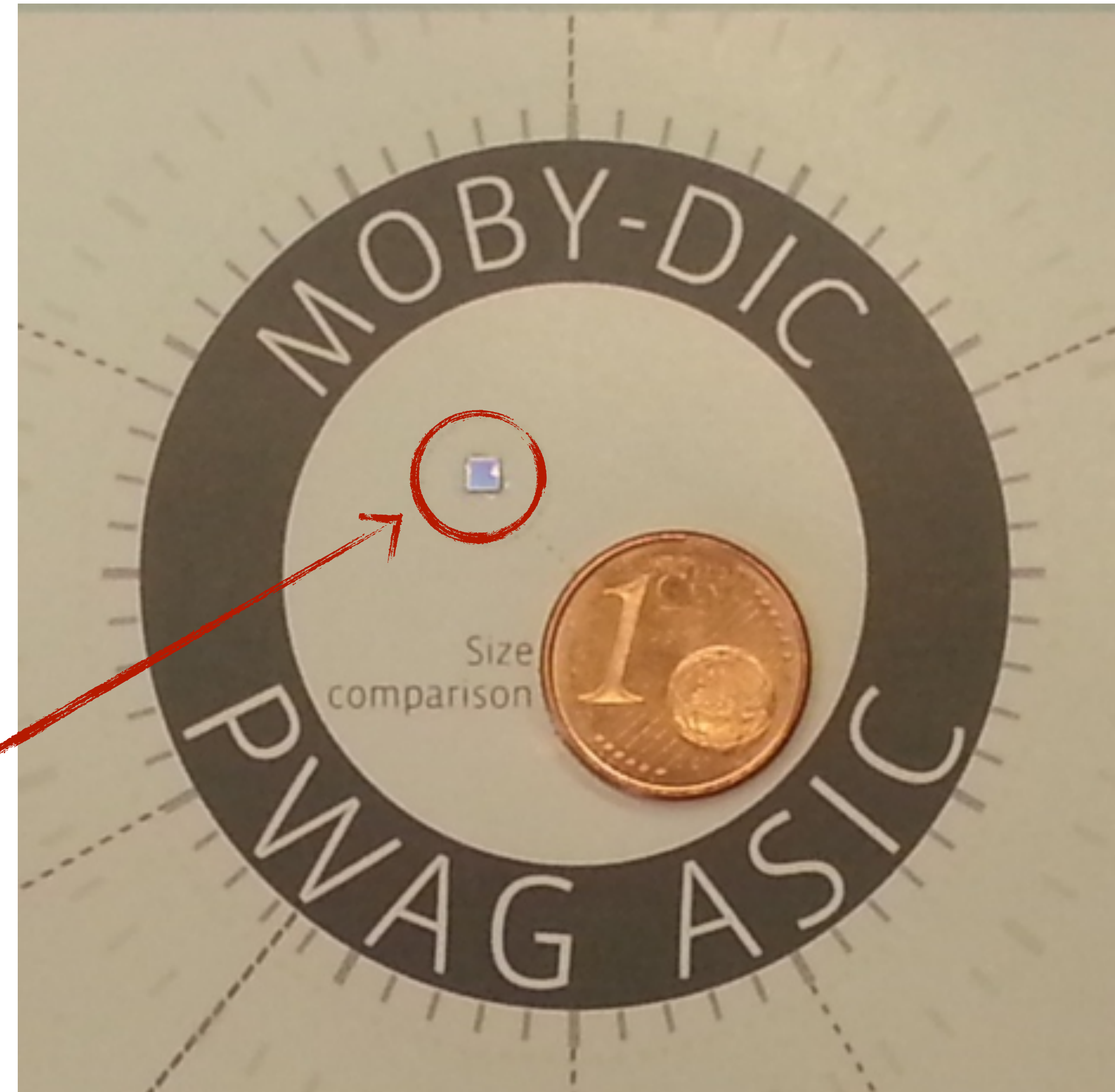
explicit MPC

baseline controller (linear)

set-point

Hardware (ASIC) implementation of explicit MPC





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

Chip size: 1860x1860 μm²

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

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

Maximum frequency: 107.5 MHz (with two inputs); 98 MHz (with four inputs)

Power consumption: 38.08 mW@107.5MHz; 41.91 mW@98 MHz

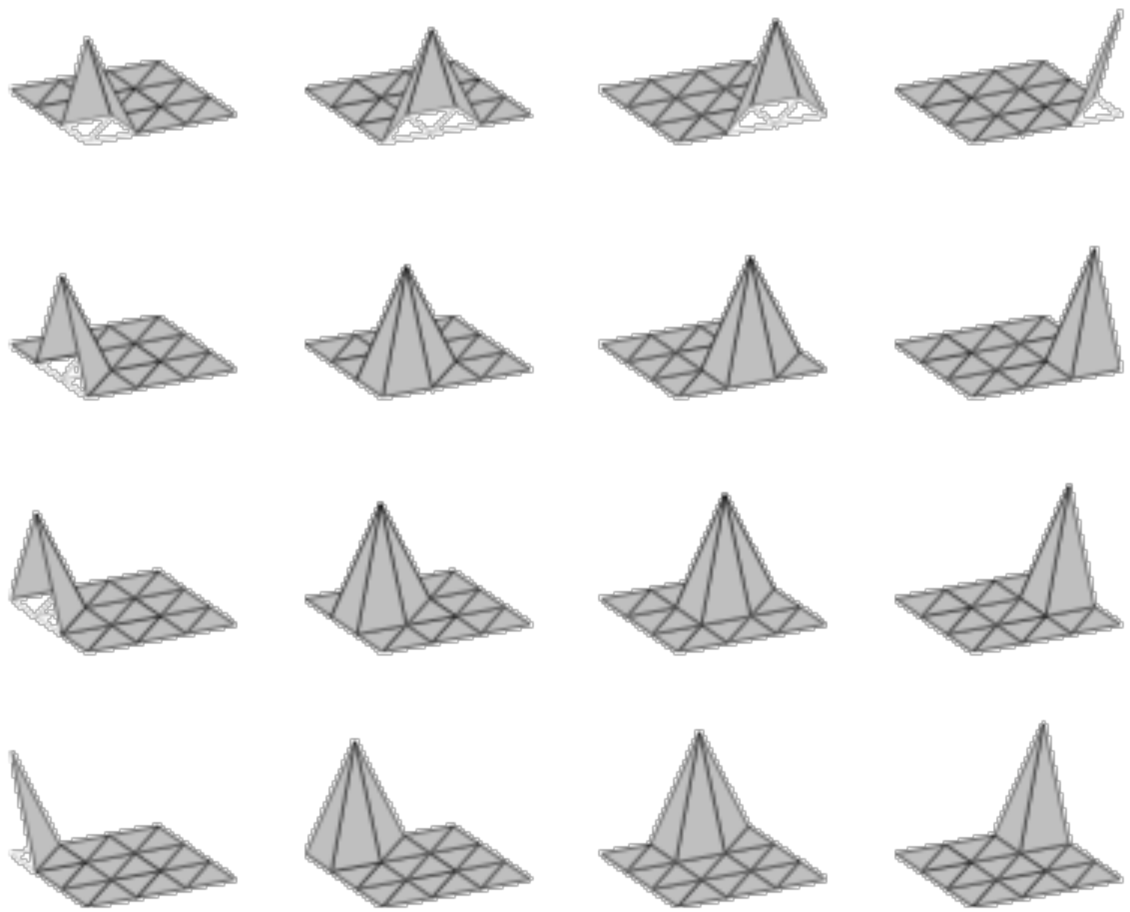
MOBY-DIC

SEVENTH FRAMEWORK PROGRAMME

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

PWA approximation of MPC over simplices

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



(Julian, Desages, Agamennoni, 1999)

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

approximate MPC law

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

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

PWA approximation of MPC over simplices

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

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

fit criterion

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

partitions per dimension

online time to compute MPC (Xilinx Spartan 3 FPGA)

Architecture A: mainly serial

Architecture B: fully parallel

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

MIMO system dynamics

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

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

subject to saturation on u and y

- Closed-loop stability results are available (Bemporad, Oliveri, Poggi, Storace , 2011) (Rubagotti, Barcelli, Bemporad, 2012)

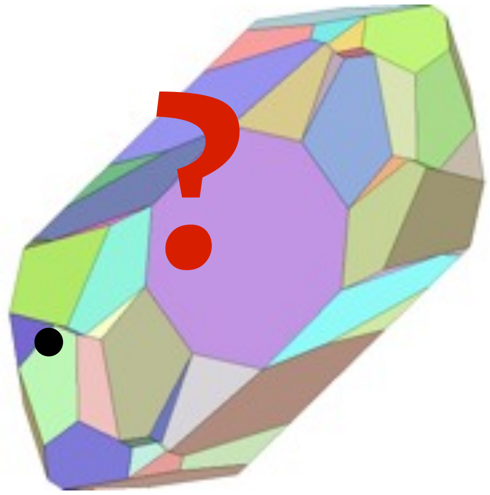
✖ Curse of dimensionality (with respect to to state dimension)

Complexity of multiparametric solutions

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

- Weak dependence on the number of states and references

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



- Fast and simple on-line QP may be preferable otherwise

Fast gradient projection for (dual) QP

- Solve dual QP using fast gradient projection
  (Nesterov, 1983)

- Main on-line operations involve only
  simple linear algebra

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

```
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
```

(Patrinos, Bemporad, IEEE TAC, 2013)

- Tight bounds on maximum number of iterations

- Similar approaches exist (Richter, Morari, Jones, 2009/2011/2012) (Giselsson, 2012)

- Dual gradient projection methods for QP work very well in fixed-point !
  (Patrinos, Guiggiani, Bemporad, ECC 2013)

Conclusions

Explicit or implicit ?

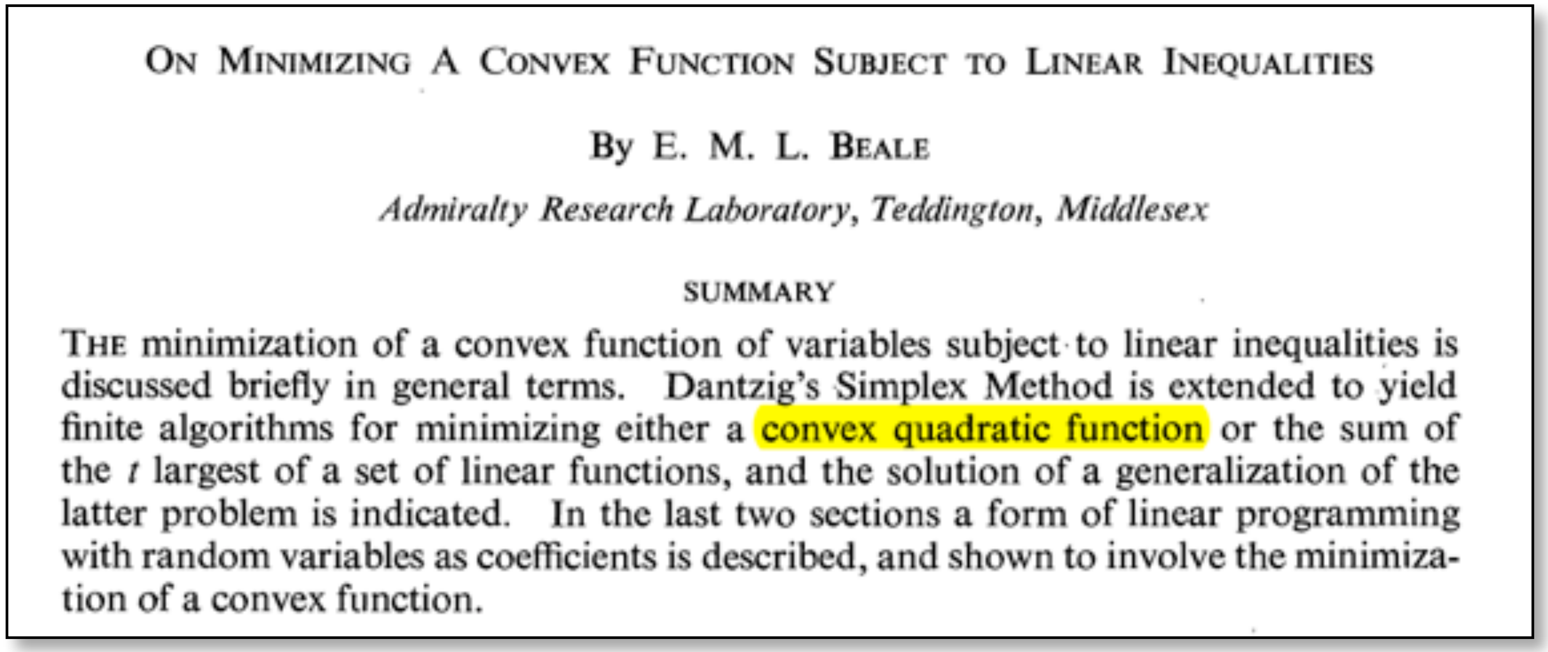
	Explicit MPC	Implicit (=on-line QP)		
		active set	interior point	gradient projection
CPU time / throughput	🙂	😐	😐	😐
worst-case exec estimates	🙂	😠	😠	😐
numerical arithmetics	🙂	😐	😐	🙂
off-line computations	😠	🙂	🙂	🙂
solution quality (feas/opt)	🙂	🙂	😐	😐
data memory	😠	🙂	🙂	🙂
control code	🙂	😐	😠	🙂
problem size	`small`	`medium`	`large`	`medium`

- Explicit typically limited **6÷8 free control moves** and **8÷12 parameters** (states+references)

- Embedded QP methods are preferable otherwise

Conclusions

Research on **QP solution methods** started ~60 years ago …



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)

… still more research on QP can have an impact in real applications !