Explicit Model Predictive Control

Alberto Bemporad

(joint survey paper with Alessandro Alessio)

University of Siena
(founded in 1240)

Department of Information Engineering

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

Pavia, September 6, 2008

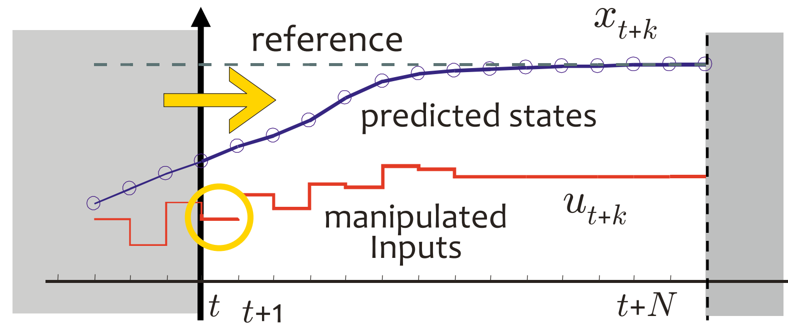
Model Predictive Control

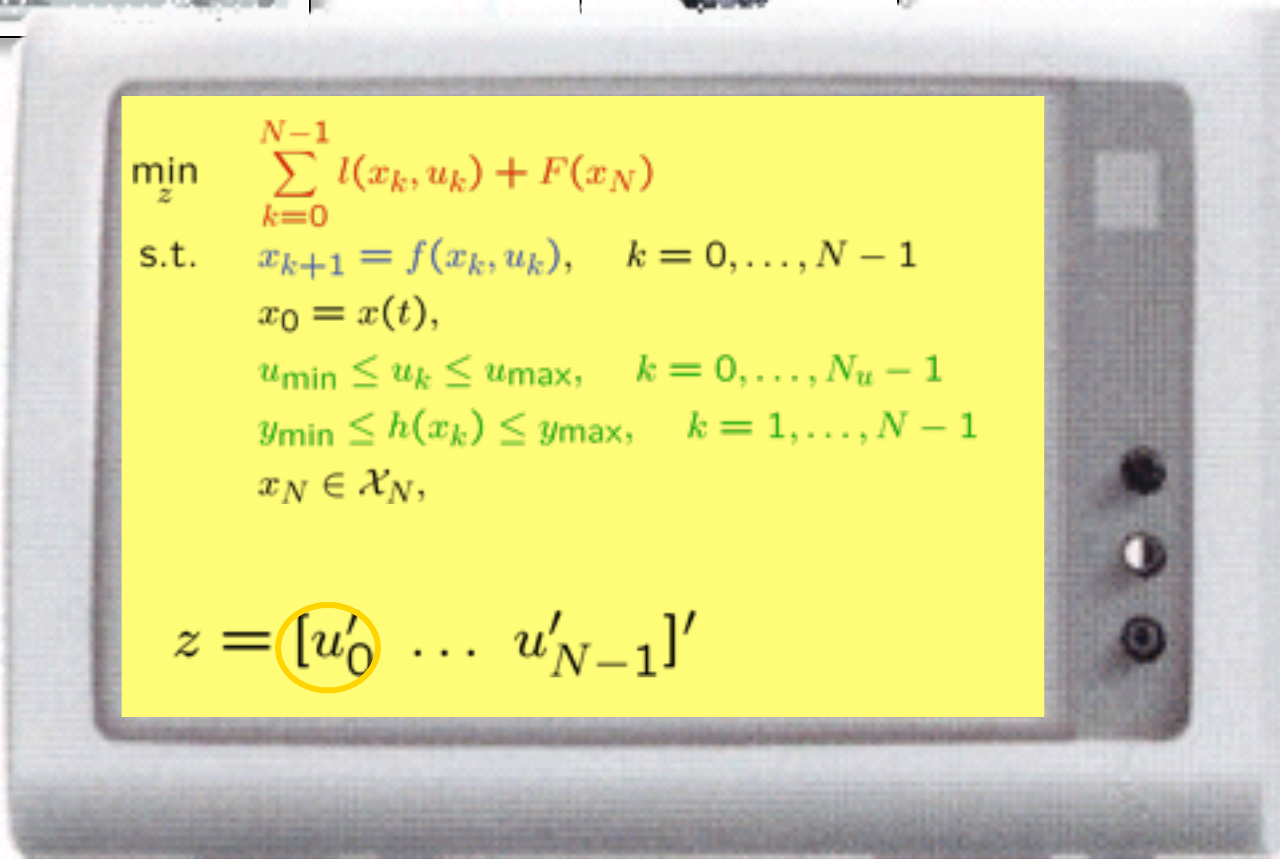
model-based optimizer

input

$u(t)$

process





$$
\begin{aligned}
\min_{z} \quad & \sum_{k=0}^{N-1} l(x_k, u_k) + F(x_N) \\
\text{s.t.} \quad & x_{k+1} = f(x_k, u_k), \quad k = 0, \dots, N-1 \\
& x_0 = x(t), \\
& u_{\min} \leq u_k \leq u_{\max}, \quad k = 0, \dots, N_u - 1 \\
& y_{\min} \leq h(x_k) \leq y_{\max}, \quad k = 1, \dots, N-1 \\
& x_N \in \mathcal{X}_N,
\end{aligned}
$$

$$z = [u_0' \ \dots \ u_{N-1}']'$$

Only apply $u^*(t)$

“Recede” horizon

& repeat @ t+1

MPC of Linear Systems – Quadratic Costs

Linear model

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

Performance index

$$\min_z x'_N P x_N + \sum_{k=0}^{N-1} x'_k Q x_k + u'_k R u_k \qquad x_0 = x(t)$$

$$\begin{array}{ll} \min\limits_z & \frac{1}{2} z' H z + x'(t) F' z + \frac{1}{2} x(t) Y x(t) \\ \text{subj. to} & Gz \leq W + Sx(t), \end{array} \qquad z = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}$$

Constraints

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

MPC law defined by the solution of a
(convex) Quadratic Program (QP)

$$R = R' \succ 0$$
$$Q = Q' \succeq 0$$
$$P = P' \succeq 0$$

MPC of Linear Systems – "Linear" Costs

Linear model

$$\begin{cases} x_{k+1} = Ax_k + Bu_k \\ y_k = Cx_k \end{cases} \qquad x_0 = x(t)$$

Performance index

$$\min_z \|Px_N\|_\infty + \sum_{k=0}^{N-1} \|Qx_k\|_\infty + \|Ru_k\|_\infty$$

$$\begin{array}{ll} \min_z & \left[1 \ \dots \ 1 | \ 0 \ \dots \ 0\right]' z \\ \text{subj. to} & Gz \leq W + Sx(t), \end{array}$$

$$z = \begin{bmatrix} \epsilon_0^u \\ \vdots \\ \epsilon_{N-1}^u \\ \epsilon_1^x \\ \vdots \\ \epsilon_N^x \\ \hline u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}$$

Constraints

$$\epsilon_k^x \geq \|Qx_{t+k|t}\|_\infty, \quad \epsilon_k^u \geq \|Ru_{t+k|t}\|_\infty, \quad \epsilon_N^x \geq \|Px_{t+N|t}\|_\infty$$

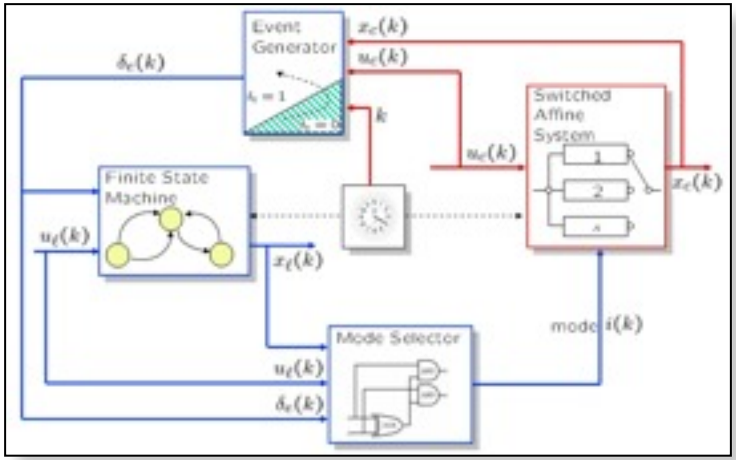
$$\begin{cases} u_{\min} \leq u_k \leq u_{\max} & \text{(more generally: } u_k \in \mathcal{U} \text{ polyhedron)} \\ x_{\min} \leq x_k \leq x_{\max} & \text{(more generally: } x_k \in \mathcal{X} \text{ polyhedron)} \end{cases}$$

MPC law defined by the solution of a **Linear Program (LP)**

(holds for **any** sum of convex piecewise affine costs)

(Schechter, 1987)

MPC of Hybrid Systems



Discrete Hybrid Automata

(Torrisi, Bemporad, 2004)

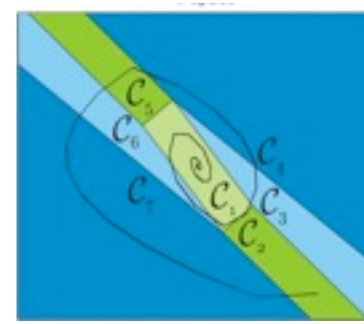
Piecewise Affine (PWA) Systems

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

(Bemporad, 2002)

(Geyer, Torrisi, Morari, 2003)

state+input space



(Sontag 1981)

Mixed Logical Dynamical (MLD) Systems

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

$\delta_i(t) \in \{0, 1\}$

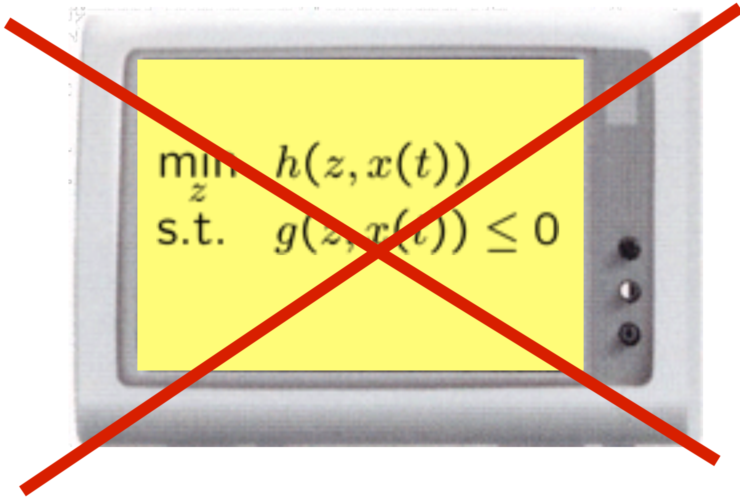
(Bemporad, Morari 1999)

MPC law defined by the solution of a

Mixed Integer Linear or Quadratic Program (MILP/MIQP)

Main Drawbacks of On-line Optimization

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



but ...

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

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

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

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

Off-line optimization

- Determine **off-line** the optimizer **function** $z^*: X \to \mathbb{R}^{n_z}$

$$z^*(x) = \arg\min_z \; h(z,x) \quad \text{s.t.} \quad g(z,x) \leq 0$$

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

where X is a given set of states x of interest

- Characterizing the optimizer means to write the MPC control law in an **explicit** form

$$u(x) = [I \;\; 0 \;\; \ldots \;\; 0] \; z^*(x)$$

Can we compute $u(x)$ in advance, in explicit terms ?

Known properties of (linear) MPC

(before year 2000)

Unconstrained linear MPC

- Linear model, quadratic costs, **no constraints**:

$$\min_z h(z, x(t)) = \frac{1}{2} z' H z + x'(t) F z \qquad z = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}$$

$$\nabla_z h(x(t), z) = Hz + F'x(t) = 0 \quad \longrightarrow \quad z^* = -H^{-1} F' x(t)$$

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

unconstrained linear MPC $\equiv$ standard linear state-feedback

Linear MPC and LQR

- Linear model, quadratic costs, Riccati terminal weight and gain:

$$\min_z h(x(t),z) = x_N' P x_N + \sum_{k=0}^{N-1} x_k' Q x_k + u_k' R u_k$$

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

Jacopo Francesco Riccati (1676-1754)

- Then, for any choice of the horizon N

(Unconstrained) MPC $\equiv$ LQR

- **With constraints**: For any compact set of states $\exists N$ such that MPC $\equiv$ constrained LQR

(Chmielewski, Manousiouthakis, 1996)

(Scokaert and Rawlings, 1998)

Linear dependence on the state

- Paper by Zafiriou (1990):

The optimization problem of the QDMC algorithm can be written as a standard quadratic programming problem:

$$\min_{v} q(v) = \tfrac{1}{2} v^{\mathrm{T}} G v + g^{\mathrm{T}} v, \quad (4)$$

subject to

$$A^{\mathrm{T}} v \geqslant b, \quad (5)$$

where

$$v = [\Delta u(\bar{k}) \cdots \Delta u(\bar{k} + M - 1)]^{\mathrm{T}} \quad (6)$$

and the matrices G, A and vectors g, b are functions of the tuning parameters (weights, horizon P, M, some of the hard constraints). The vectors g, b are also linear functions of $y(\bar{k}), u(\bar{k}-1), \ldots, u(\bar{k}-N)$. For the optimal solution v^* we have (Fletcher, 1981):

$$\begin{bmatrix} G & -\hat{A} \\ -\hat{A}^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} v^* \\ \lambda^* \end{bmatrix} = -\begin{bmatrix} g \\ \hat{b} \end{bmatrix}, \quad (7)$$

where $\hat{A}^{\mathrm{T}}$, $\hat{b}$ consist of the rows of A^{T}, b that correspond to the constraints that are active at the optimum and λ^* is the vector of the Lagrange multipliers corresponding to these constraints. The

$$\begin{bmatrix} G & -\hat{A} \\ -\hat{A}^{\mathrm{T}} & 0 \end{bmatrix}^{-1} = \begin{bmatrix} H & -T \\ -T^{\mathrm{T}} & U \end{bmatrix}. \quad (8)$$

Then:

$$v^* = -Hg + T\hat{b}, \quad (9)$$

$$\lambda^* = T^{\mathrm{T}} g - U\hat{b} \quad (10)$$

and

$$u(\bar{k}) = u(\bar{k}-1) + \underbrace{[I \quad 0 \cdots 0]}_{M} v^*$$

$$\stackrel{\text{def}}{=} f[y(\bar{k}), u(\bar{k}-1), \ldots, u(\bar{k}-N), r_{\mathrm{P}}(k)]. \quad (11)$$

Explicit (linear) MPC

(year 2000)

Full characterization of linear MPC law

Three papers appeared in 2000 on the solution $u(x)$ of linear MPC:

(Johansen, Petersen, Slupphaug, CDC 2000)

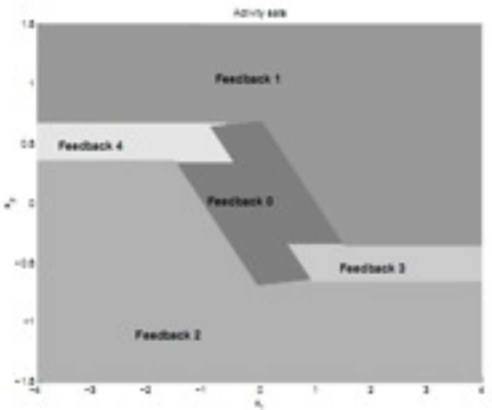


Figure 3: Activity regions for the five constituent affine feedbacks in the constrained LQR for the double integrator with boundary layers.

suboptimal way of solving constrained LQR:

- select a number of active constraint combinations
- compare on-line where regions overlap

(Trondheim, Norway)

(Seron, De Doná, Goodwin, CDC 2000)

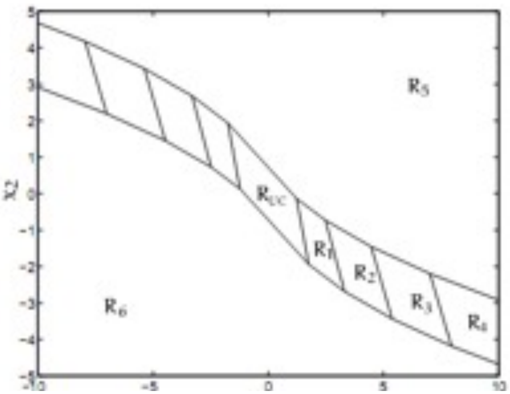


Figure 3: State-space partition for Example 4.1 for N = 5.

enumerate all possible locations of optimizer in an equivalent minimum distance problem

(input constraints only)

(Newcastle, Australia)

(Bemporad, Morari, Dua, Pistikopoulos, ACC 2000)

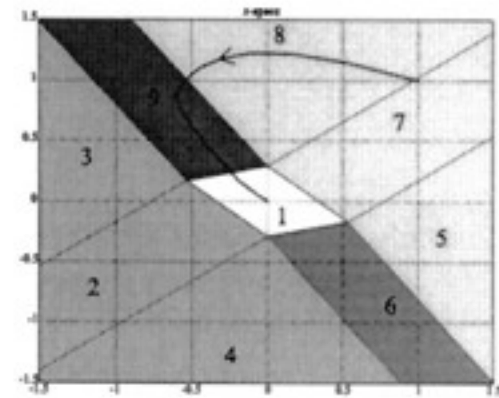


Figure 1: Closed-loop MPC and partition of the state-space

recursively partition the state-space in a set of polyhedra, each one corresponding to a unique optimal solution affine in x

⇨ **multiparametric QP**

(Zurich, Switzerland)

Multiparametric programming problem

Given the optimization problem

$$\begin{array}{ll} \min_{z} & h(z, x) \\ \text{s.t.} & g(z, x) \leq 0 \end{array}$$

determine:

- The **feasible parameter set** X_f of all $x \in X$ for which the problem admits a solution ($g(z, x) \leq 0$ for some z)
- The **value function** $V^* : X_f \to \mathbb{R}$ that at each x associates the optimal value $V^*(x)$
- An **optimizer function** $z^* : X_f \to \mathbb{R}^\ell$

Multiparametric Quadratic Programming

(Bemporad, Morari, Dua, Pistikopoulos, 2002)

$$\begin{array}{ll} \min_z & \frac{1}{2}z'Hz + x'F'z + \frac{1}{2}xYx \\ \text{subj. to} & Gz \leq W + Sx, \end{array}$$

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

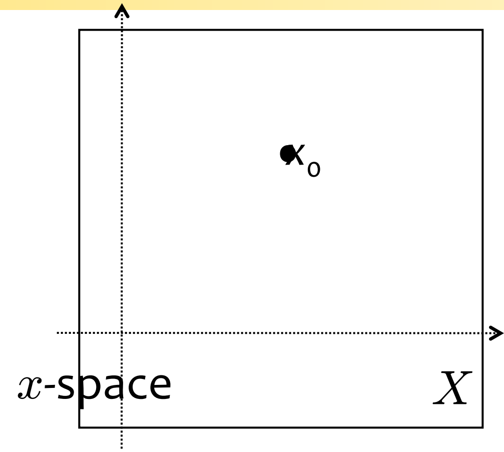
- Objective: solve the QP off line **for all** $x \in X$ to find the MPC control law $u=u(x)$ **explicitly**

- Assumptions:

$\begin{bmatrix} H & F \\ F' & Y \end{bmatrix} \succeq 0$ — always satisfied if mpQP comes from an optimal control problem !

$H \succ 0$ — always satisfied if weight matrix $R>0$

Multiparametric QP algorithm



- Fix $x_0 \in X$

  ⇨ 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: $\tilde G z^*(x_0) - \tilde W - \tilde S x_0 = 0$

- For the fixed combination of active constraints get z, $\tilde G,\ \tilde W,\ \tilde S$ from the KKT conditions of optimality:

$$\tilde\lambda(x) = -(\tilde G H^{-1} \tilde G')^{-1}(\tilde W + (\tilde S + \tilde G H^{-1} F)x).$$

$$z(x) = H^{-1}[\tilde G'(\tilde G H^{-1}\tilde G')^{-1}(\tilde W + (\tilde S + \tilde G H^{-1}F)x) - Fx]$$

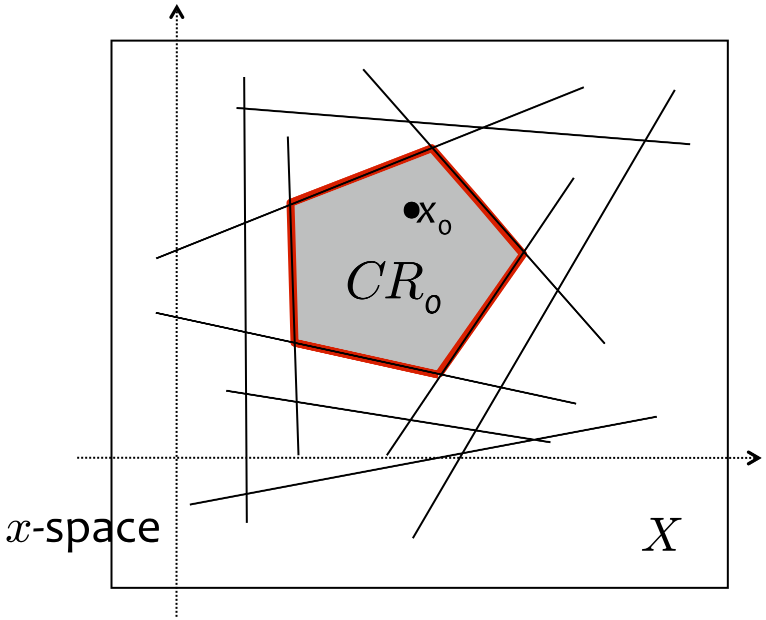
➡ In some neighborhood of x_0, λ and z are *explicit* **affine** functions of x (Zafiriou, 1990)

Multiparametric QP algorithm

- Impose primal and dual feasibility:

$$\hat{G}z(x) \leq \hat{W} + \hat{S}x$$
$$\tilde{\lambda}(x) \geq 0$$

→ linear inequalities in x !



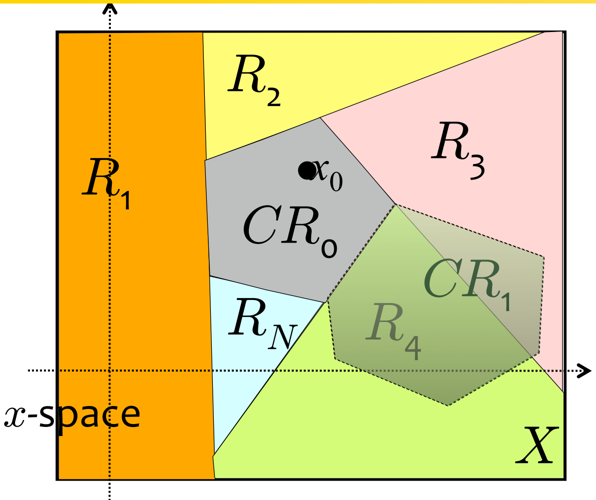
- Remove redundant constraints (this requires solving LP's):

→ 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 algorithm



Method #1: Split and proceed iteratively

Recursions terminate because combinations of active constraints are finite

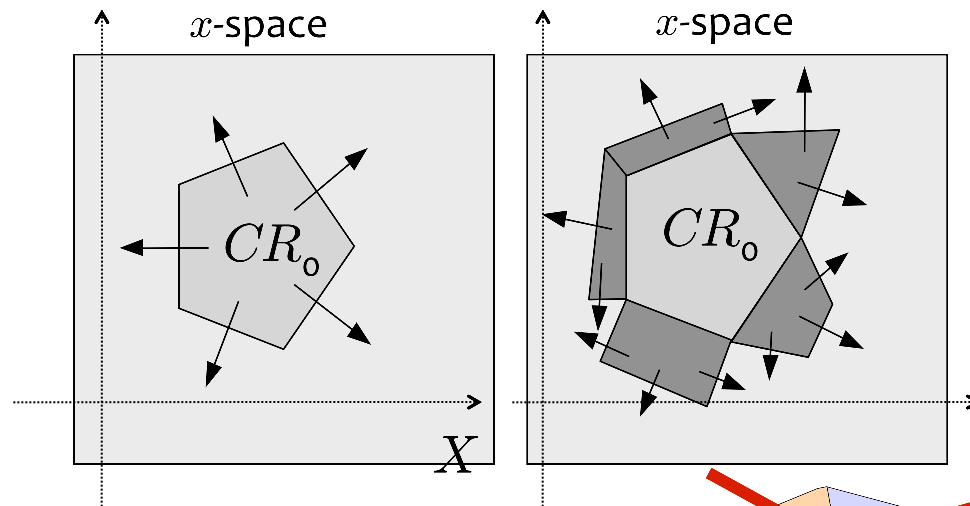
(Bemporad, Morari, Dua, Pistikopoulos, 2002)

Method #2: add/withdraw constraints from active set

$\hat{G}^i z(x) \le \hat{W}^i + \hat{S}^i x$ $\Rightarrow$ add

$\tilde{\lambda}_j(x) \ge 0$ $\Rightarrow$ withdraw

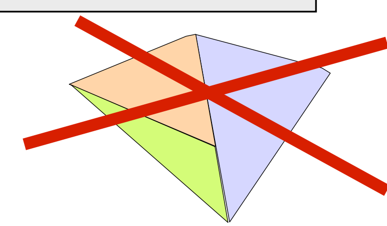
(Tøndel, Johansen, Bemporad, 2003)



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

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

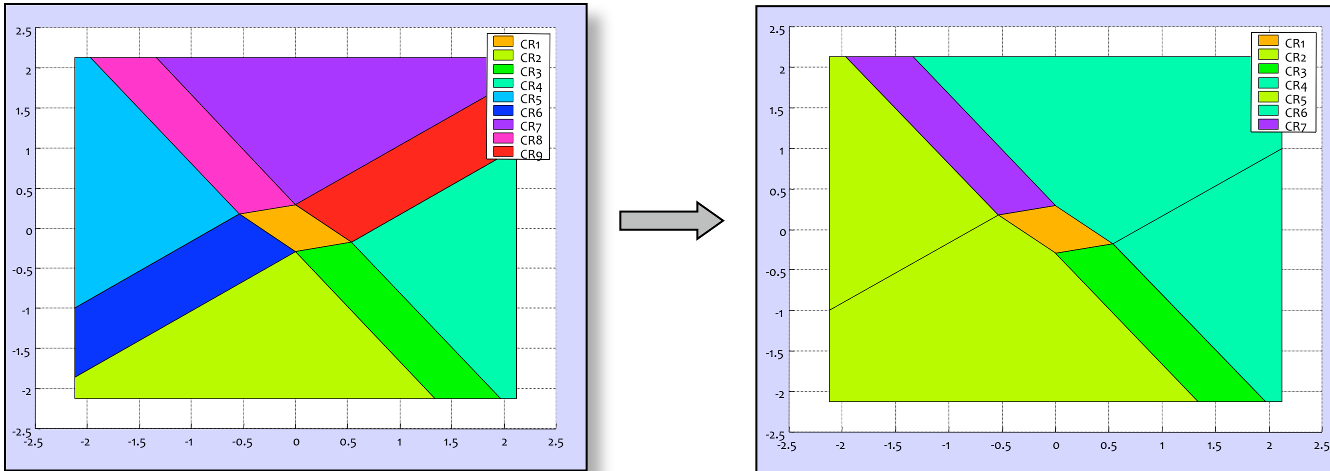
(Spjøtvold, 2008)



Step out ϵ outside each facet, solve QP, get new region, iterate.

(Baotic, 2002)

Get explicit MPC law



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

Regions where the first component of the solution is the same can be joined (when their union is convex).

(Bemporad, Fukuda, Torrisi, 2001)

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

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

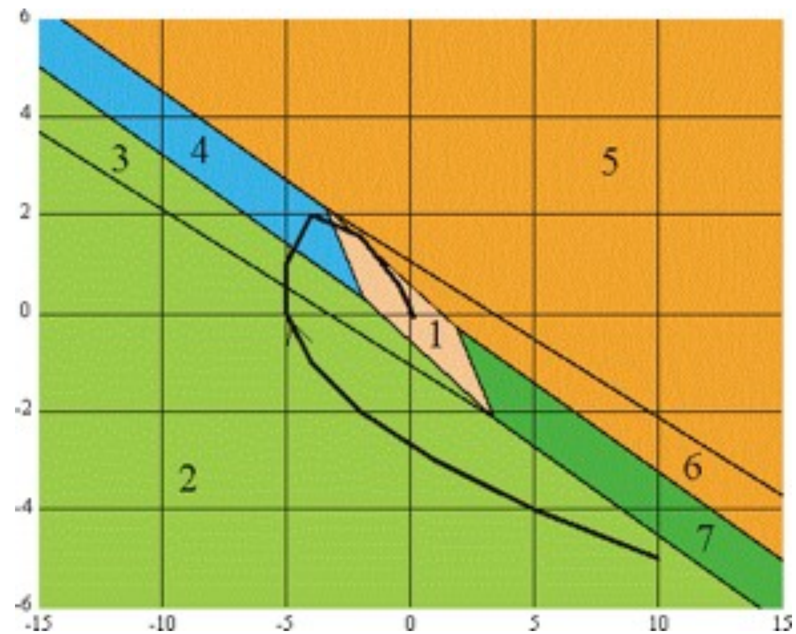
Double integrator example

$$
\begin{cases}
x_1(t+1) = x_1(t) + x_2(t) \\
x_2(t+1) = x_2(t) + u(t) \\
y(t) = x_1(t)
\end{cases}
$$

$$\min x_2' P x_2 + \sum_{k=0}^{1} y_k^2 + \frac{1}{100} u_k^2$$

$$-1 \leq u_0, u_1 \leq 1$$

P = solution Riccati equation



$$
u(x) = \begin{cases}
\begin{bmatrix} -0.8166 & -1.7499 \end{bmatrix} x & \text{if } \begin{bmatrix} -0.8166 & -1.7499 \\ 0.8166 & 1.7499 \\ 0.6124 & 0.4957 \\ -0.6124 & -0.4957 \end{bmatrix} x \leq \begin{bmatrix} 1.0000 \\ 1.0000 \\ 1.0000 \\ 1.0000 \end{bmatrix} & \text{(Region \#1)} \\
1.0000 & \text{if } \begin{bmatrix} 0.3864 & 1.0738 \\ 0.2970 & 0.9333 \end{bmatrix} x \leq \begin{bmatrix} -1.0000 \\ -1.0000 \end{bmatrix} & \text{(Region \#2)} \\
1.0000 & \text{if } \begin{bmatrix} 0.9712 & 2.6991 \\ -0.2970 & -0.9333 \\ 0.8166 & 1.7499 \end{bmatrix} x \leq \begin{bmatrix} -1.0000 \\ 1.0000 \\ -1.0000 \end{bmatrix} & \text{(Region \#3)} \\
\begin{bmatrix} -0.5528 & -1.5364 \end{bmatrix} x + 0.4308 & \text{if } \begin{bmatrix} -0.9712 & -2.6991 \\ 0.3864 & 1.0738 \\ 0.6124 & 0.4957 \end{bmatrix} x \leq \begin{bmatrix} 1.0000 \\ 1.0000 \\ -1.0000 \end{bmatrix} & \text{(Region \#4)} \\
-1.0000 & \text{if } \begin{bmatrix} -0.3864 & -1.0738 \\ -0.2970 & -0.9333 \end{bmatrix} x \leq \begin{bmatrix} -1.0000 \\ -1.0000 \end{bmatrix} & \text{(Region \#5)} \\
-1.0000 & \text{if } \begin{bmatrix} -0.9712 & -2.6991 \\ 0.2970 & 0.9333 \\ -0.8166 & -1.7499 \end{bmatrix} x \leq \begin{bmatrix} -1.0000 \\ 1.0000 \\ -1.0000 \end{bmatrix} & \text{(Region \#6)} \\
\begin{bmatrix} -0.5528 & -1.5364 \end{bmatrix} x - 0.4308 & \text{if } \begin{bmatrix} -0.3864 & -1.0738 \\ 0.9712 & 2.6991 \\ -0.6124 & -0.4957 \end{bmatrix} x \leq \begin{bmatrix} 1.0000 \\ 1.0000 \\ -1.0000 \end{bmatrix} & \text{(Region \#7)}
\end{cases}
$$

The origins of (multi)parametric programming

(mono)-parametric LP

THE COMPUTATIONAL ALGORITHM FOR THE PARAMETRIC OBJECTIVE FUNCTION[1]

Saul Gass
U. S. Air Force[2]
and
Thomas Saaty
Melpar, Inc.[3]

(Gass and Saaty, 1955)

Let $\delta \leq \lambda \leq \phi$ be an arbitrary interval on the real line
For each λ in this interval, find a vector $x = (x_1, x_2, \ldots . x_n)$ which minimizes

$$\sum_{j=1}^{n} (d_j + \lambda d'_j)\, x_j ,$$

$$x_j \geq 0 \qquad j = 1, \ldots, n,$$

$$\sum_{j=1}^{n} a_{ij} x_j = a_{i0} \qquad i = 1, \ldots, m,$$

$$\begin{aligned} \min_{z} \quad & (c_1 + x \cdot c_2)' z \\ \text{s.t.} \quad & Gz = W \\ & z \geq 0 \end{aligned} \qquad x \in \mathbb{R}$$

Also extended to 2 parameters in 1955 by Gass and Saaty

multi-parametric convex programming

INEQUALITIES FOR STOCHASTIC NONLINEAR PROGRAMMING PROBLEMS

O. L. Mangasarian and J. B. Rosen*

Shell Development Company, Emeryville, California

(Received December, 1962)

(Mangasarian and Rosen, 1964)

$$V^*(\color{red}{x}) = \min_z \ h(z,\color{red}{x}) \quad \text{s.t.} \quad g(z,\color{red}{x}) \le 0$$

h, g **convex** in (z, x)

Lemma 1. *The scalar function* $\alpha(a) \equiv \min_z\{\theta(z,a) \mid f(z,a) \geqq 0\}$ *is a convex function of the vector a provided that* θ *is a convex function of the vector* $[z'a']$ *and each component of f is a concave function of* $[z'a']$.

h, g convex in $(z, x) \Rightarrow V^*(\color{red}{x})$ convex

Lemma 2. *The scalar function* $\alpha(a) \equiv \min_z\{\theta(z,a)(z,a) \mid f(z,a) \geqq 0\}$ *is a convex and continuous function of the vector a provided that* θ *is a convex and continuous function of the vector* $[z'a']$, *and each component of f is a concave and continuous function of* $[z'a']$.

h, g convex and continuous in $(z, x) \Rightarrow V^*(\color{red}{x})$ convex and continuous

multi-parametric LP

MANAGEMENT SCIENCE
Vol. 18, No. 7, March, 1972
Printed in U.S.A.

MULTIPARAMETRIC LINEAR PROGRAMMING*

TOMAS GAL† AND JOSEF NEDOMA‡

The multiparametric linear programming (MLP) problem for the right-hand sides (RHS) is to maximize $z = c^T x$ subject to $Ax = b(\lambda), x \geqq 0$, where $b(\lambda)$ can be expressed in the form

$$b(\lambda) = b^* + F\lambda,$$

where F is a matrix of constant coefficients, and λ is a vector-parameter.

The multiparametric linear programming (MLP) problem for the prices or objective function coefficients (OFC) is to maximize $z = c^T(\nu)x$ subject to $Ax = b$, $x \geqq 0$, where $c(\nu)$ can be expressed in the form $c(\nu) = c^* + H\nu$, and where H is a matrix of constant coefficients, and ν a vector-parameter.

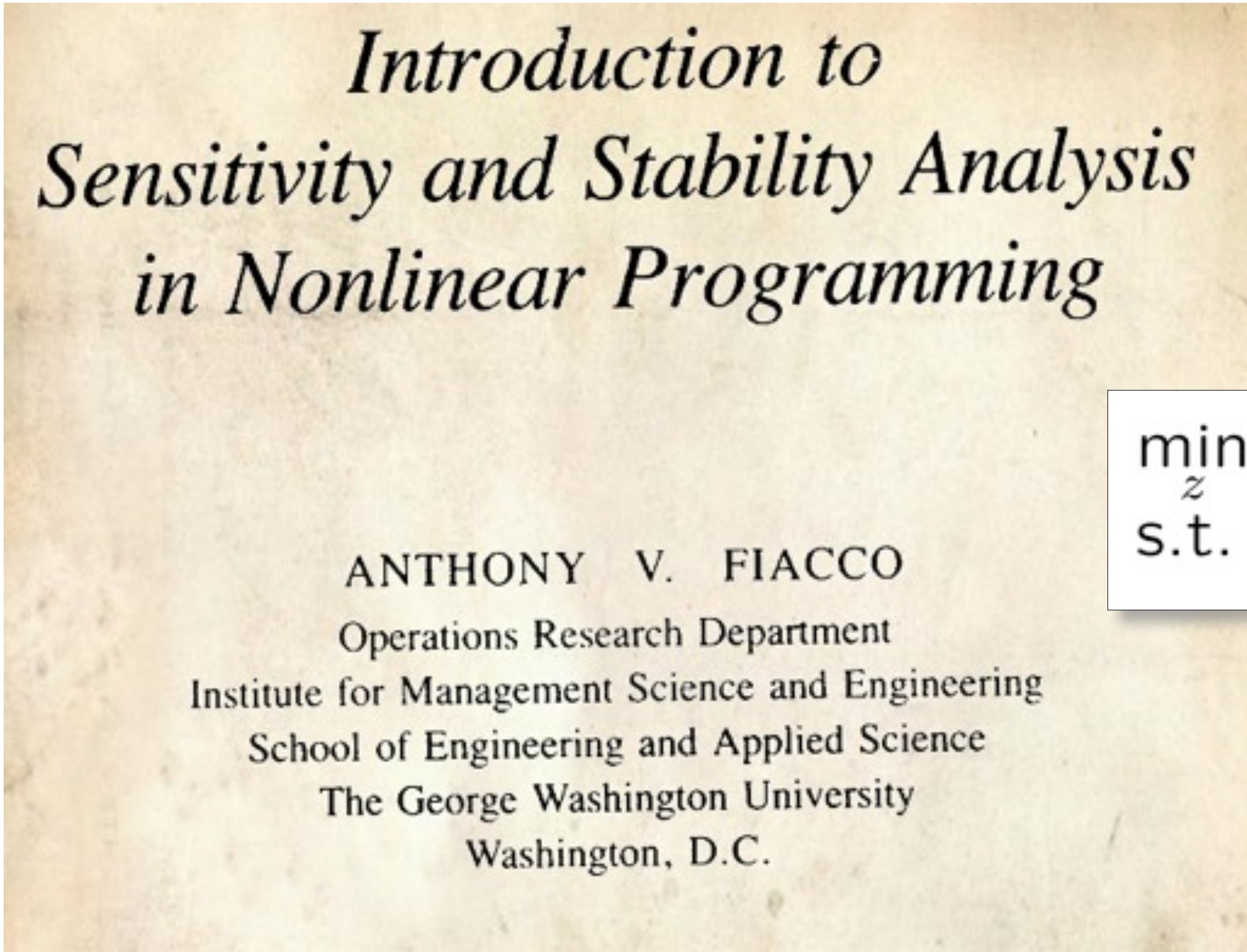
(Gal and Nedoma, 1972)

$$\begin{aligned} \min_z \quad & c'z \\ \text{s.t.} \quad & Gz = W + Sx \\ & z \geq 0 \end{aligned}$$

$$\begin{aligned} \min_z \quad & (c_1 + x'c_2)'z \\ \text{s.t.} \quad & Gz = W \\ & z \geq 0 \end{aligned}$$

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

multi-parametric NLP - 1983



Introduction to
Sensitivity and Stability Analysis
in Nonlinear Programming

ANTHONY V. FIACCO

Operations Research Department
Institute for Management Science and Engineering
School of Engineering and Applied Science
The George Washington University
Washington, D.C.

$$\begin{aligned} \min_{z} \quad & h(z, x) \\ \text{s.t.} \quad & g(z, x) \leq 0 \end{aligned}$$

(Fiacco, 1983)

Very general treatment of multiparametric programming

Multiparametric programming algorithms

Problem	$z^*(x)$	$V^*(x)$	
mp-LP	continuous, PWA	convex (cont.), PWA	(Gal, Nedoma, 1972) (Gal 1995) (Borrelli, Bemporad, Morari, 2003)
mp-QP	continuous, PWA	convex (cont.) piecewise quadratic, C^1 (if no degen.)	(Bemporad, Morari, Dua, Pistikopoulos, 2002) (Tøndel, Bemporad, Johansen, 2003a) (Seron, De Doná, Goodwin, 2000) (Baotic, 2002)
mp-MILP	PWA	(nonconvex) PWA	(Acevedo, Pistikopoulos, 1997) (Dua, Pistikopoulos, 2000)
mp-LCP	continuous, PWA	[undefined]	(Jones, Morari, 2006) (Columbano, Fukuda, Jones, 2008)
mp-convex (mp-SDP)	PWA (approx.)	convex (approx.)	(Bemporad, Filippi. 2003)
mp-IP	PW constant	PWA	(Bemporad, 2003) (Crema, 1999)

Ways to handle *degeneracy* in mpQP/mpLP have been studied

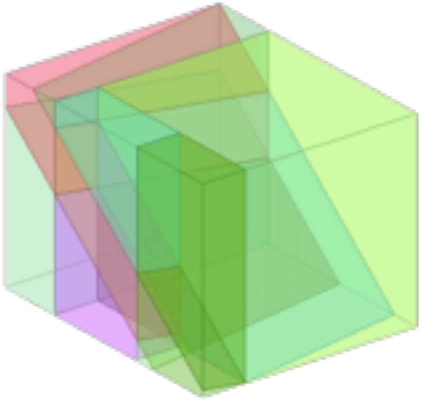
(Tøndel, Bemporad, Johansen, 2003b)

(Jones, Kerrigan, Maciejowski. 2007)

Multiparametric solutions: Hybrid MPC

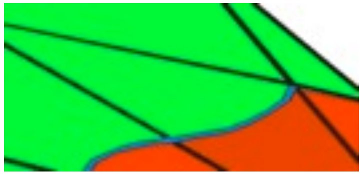
The **hybrid MPC** controller is **piecewise affine** in x
(control law may be *discontinuous*)

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



- Use mixed-integer (MLD) model and solve mp-MILP
  (Bemporad, Borrelli, Morari, 2000)
- Use PWA model + dynamic programming + mpLP/mpQP
  (Borrelli, Baotic, Bemporad, Morari, 2005)
- PWA + enumerate switching sequences + mpQP
  (Mayne, ECC 2001) (Mayne, Rakovic, 2002)
- PWA + DP for enumeration + mpQP/mpLP + cost comparison
  (Bemporad, 2004) (Alessio, Bemporad, ADHS 2006)

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



Multiparametric solutions: Min-max MPC

$$x_{k+1} = A(w_k)x_k + B(w_k)u_k + Ev_k$$ uncertain linear model

$$A(w) = A_0 + \sum_{i=1}^{q} A_i w_i, \quad B(w) = B_0 + \sum_{i=1}^{q} B_i w_i$$ w, v belong to polytopes

- open-loop prediction, ∞-norms: solved via mpLP ($A(w) \equiv A_0$)

  (Bemporad, Borrelli, Morari, 2003)

- closed-loop prediction, ∞-norms:

  - mpLP iterations (dynamic programming solution)

    (Bemporad, Borrelli, Morari, 2003)

  - mpLP solving single LP problem of Scokaert-Mayne

    (Kerrigan, Maciejowski, 2004)

- min-max MPC with quadratic costs

  (Ramirez, Camacho, 2006)

  (Munoz, Alamo, Ramirez, Camacho, 2007)

Explicit min-max MPC control law is piecewise affine

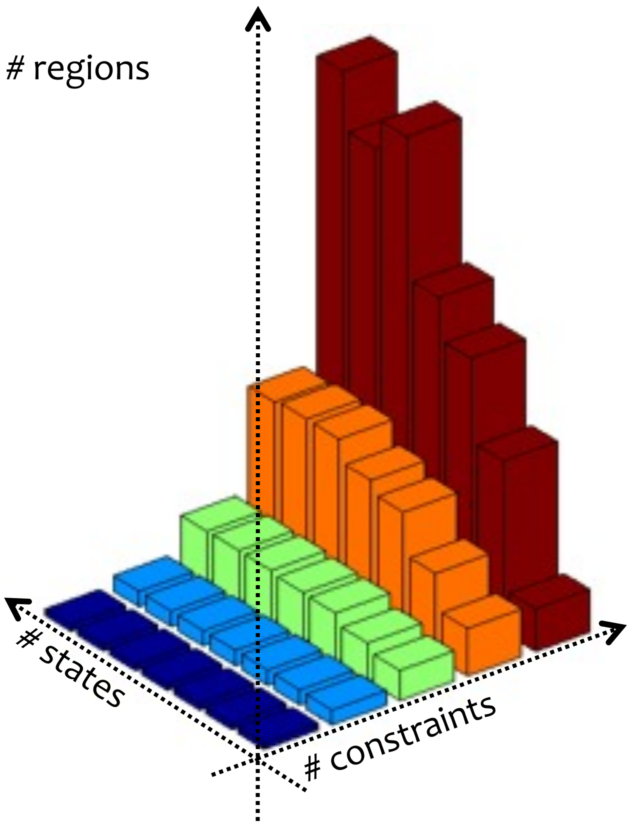
Complexity

- Number of regions $\leq$ # combinations of active constraints

  - Largely depends on #constraints (usually combinatorially)

  - Also depends on #free variables

  - Weakly depends on #states



- Example

average on 20 random SISO systems (input saturation)

states\horizon	$N=1$	$N=2$	$N=3$	$N=4$	$N=5$
n=2	3	6.7	13.5	21.4	19.3
n=3	3	6.9	17	37.3	77
n=4	3	7	21.65	56	114.2
n=5	3	7	22	61.5	132.7
n=6	3	7	23.1	71.2	196.3
n=7	3	6.95	23.2	71.4	182.3
n=8	3	7	23	70.2	207.9

Explicit MPC typically limited to 6-8 free control moves and 8-12 states+references

Complexity - QP vs. Explicit

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

(Intel Centrino 1.4 GHz)

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

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

→ Need to visit regions more efficiently than linear search

→ Need to reduce number of regions

Point location problem

In which region of the partition is $x(t)$?

- Store all regions and search linearly through them

- Exploit properties of mpLP solution to locate $x(t)$ from value function (also extended to mpQP) (Baotic, Borrelli, Bemporad, Morari 2008)

- Organize regions on a tree for logarithmic search (Tøndel, Johansen, Bemporad, 2003)

- For mpLP, recast as weighted nearest neighbour problem (logarithmic search) (Jones, Grieder, Rakovic, 2003)

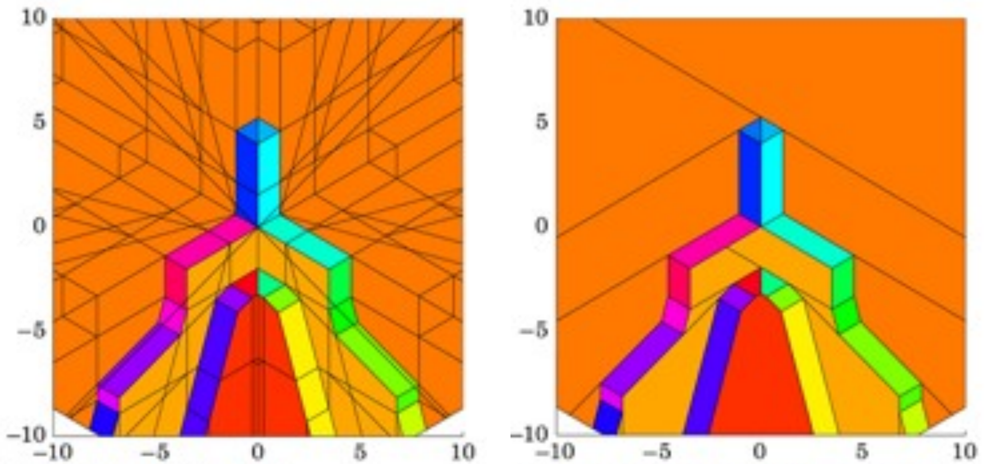
- Exploit reachability analysis (Spjøtvold, Rakovic, Tøndel, Johansen, 2006) (Wang, Jones, Maciejowski, 2007)

- Use bounding boxes and trees (Christophersen, Kvasnica, Jones, Morari, 2007)

Region reduction

- Join regions more efficiently

  (Geyer, Torrisi, Morari, 2008)



(a) Original set of 252 polyhedra. (c) Set of 39 polyhedra resulting from OCR.

- Change cost function (e.g. minimum time)

  (Grieder, Morari, 2003)

- Relax KKT conditions (suboptimal mpQP)

  (Bemporad, Filippi, 2003)

- Use orthogonal trees (suboptimal solutions)

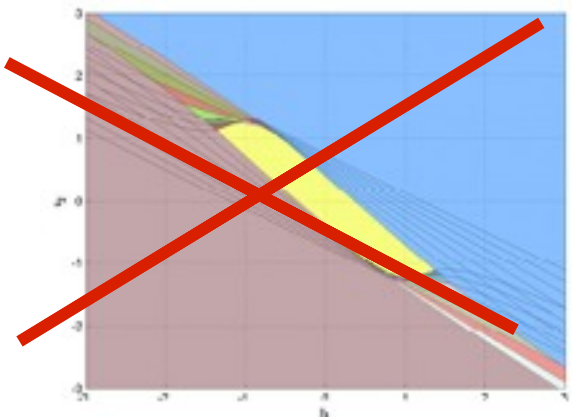
  (Johansen, Grancharova, 2003)

Region reduction - Interpolation

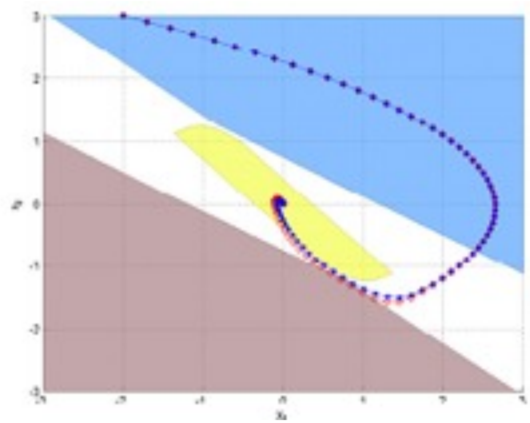
- Interpolate solution from reduced number of regions

(Pannocchia, Rawlings, Wright, 2007) (Christophersen, Zeilinger, Jones, Morari, 2007)

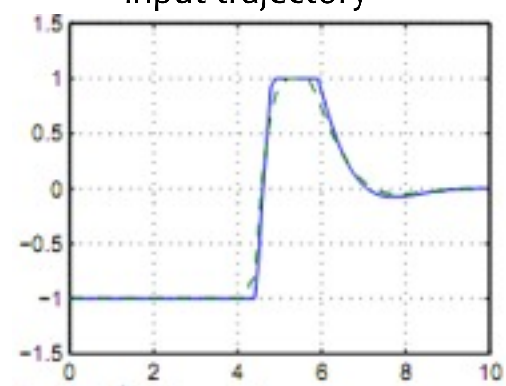
exact solution (99 regions)



3 most visited combinations of active constraints are stored



input trajectory



(Alessio, Bemporad, NMPC 2008)

$$\beta_i(x) = \max_j \{H_i^j x - K_i^j\}$$

max violation (how much x is outside region $H_i x \leq K_i$

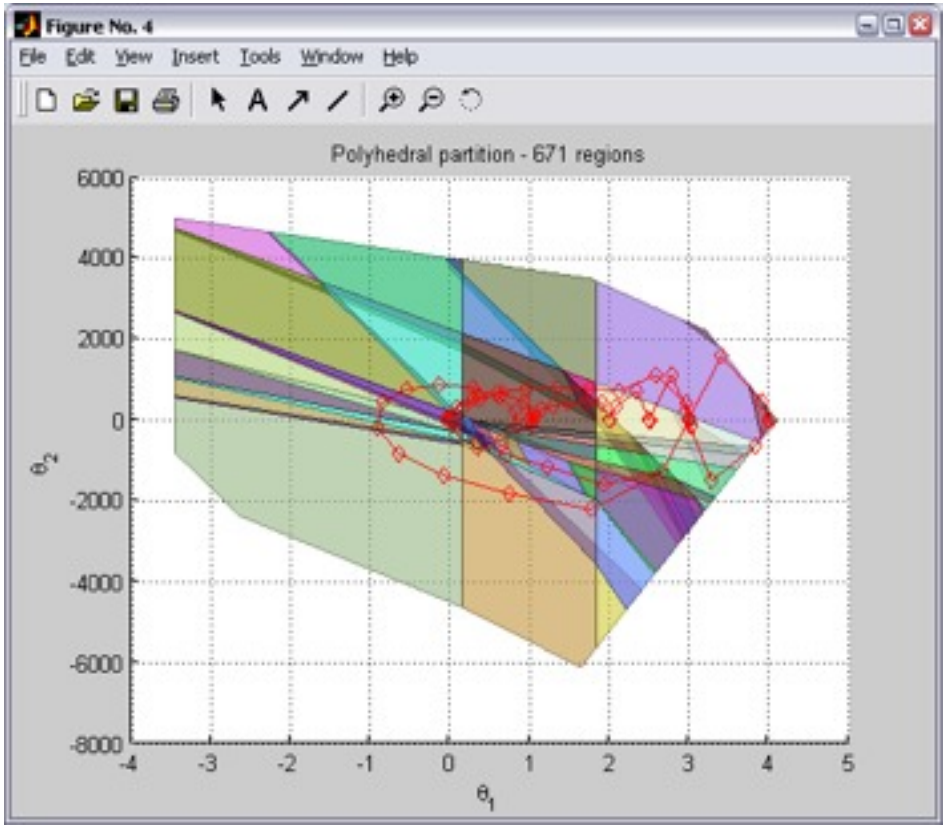
$$\bar{u}(x) = \left(\sum_{i=1,\ldots,L} \frac{1}{\beta_i(x)} \right)^{-1} \sum_{i=1,\ldots,L} \frac{1}{\beta_i(x)} (F_i x + g_i)$$

or set $\bar{u}(x) = F_h x + g_h$ $\quad \beta_h(x) = \min_{i=1,\ldots,L} \beta_i(x)$

Region reduction - Interpolation

Example: hybrid MPC of magnetic actuators

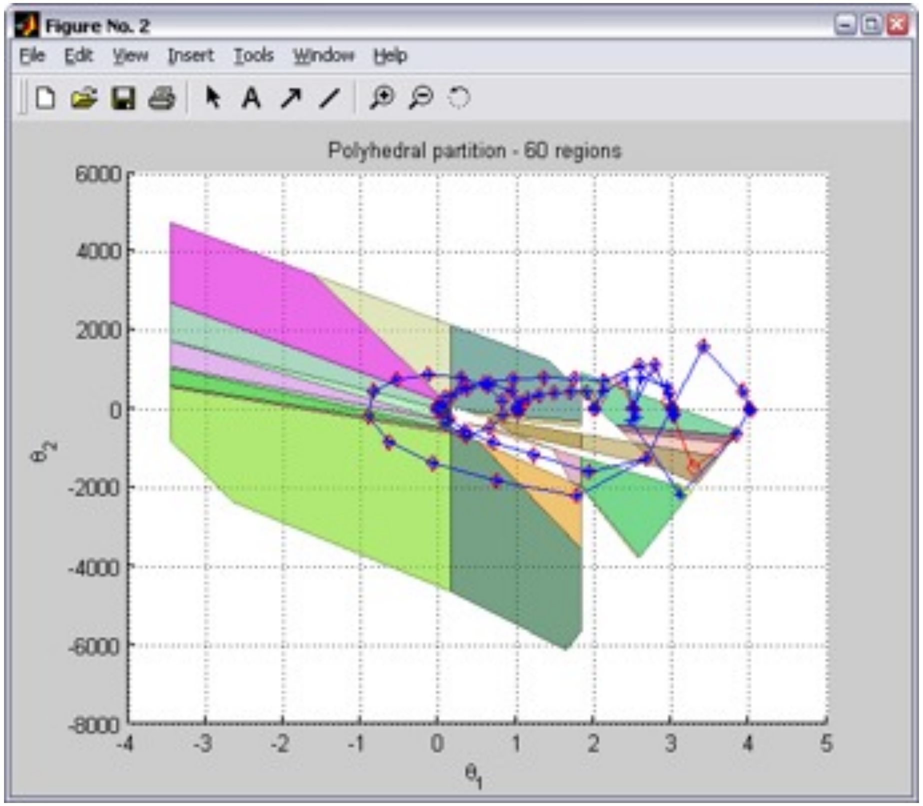
(Di Cairano, Bemporad, Kolmanovsky, Hrovat, 2006)

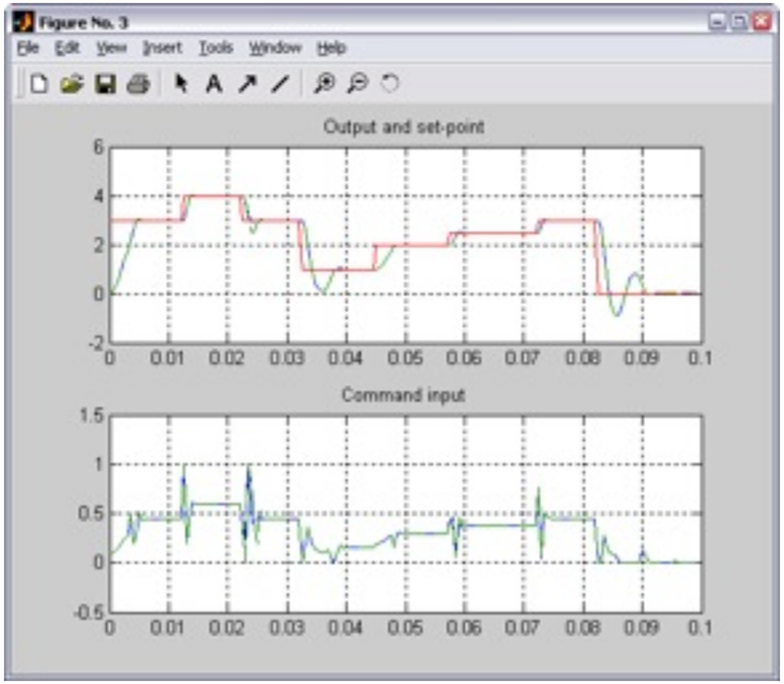


Exact explicit solution (671 regions)

Region reduction - Interpolation

Suboptimal solution (Bemporad, summer 2007)

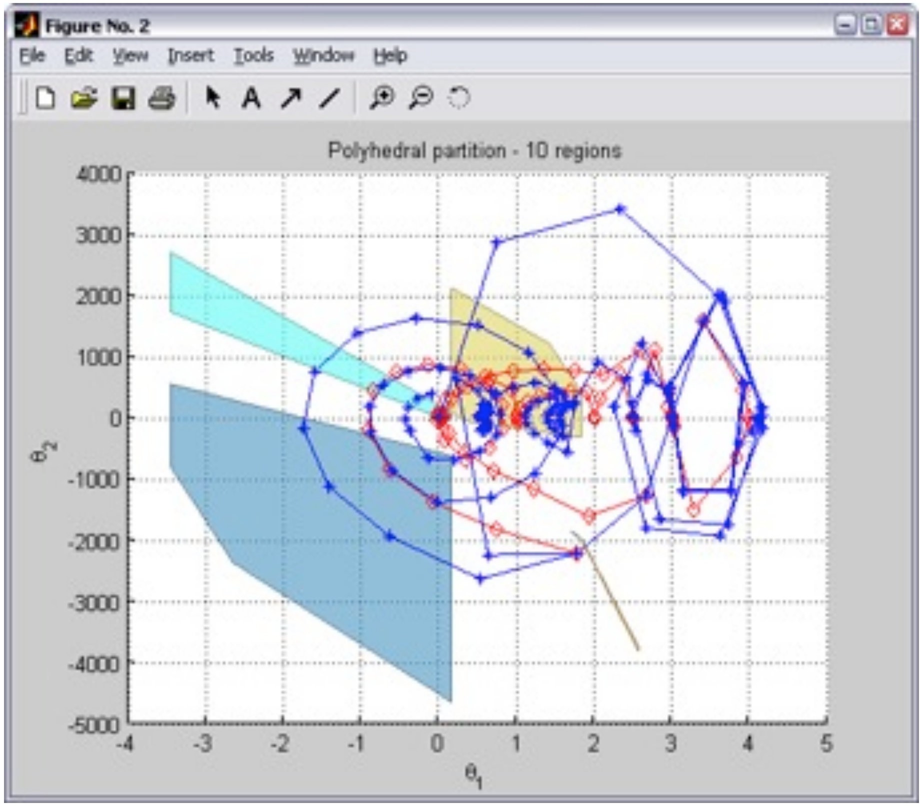




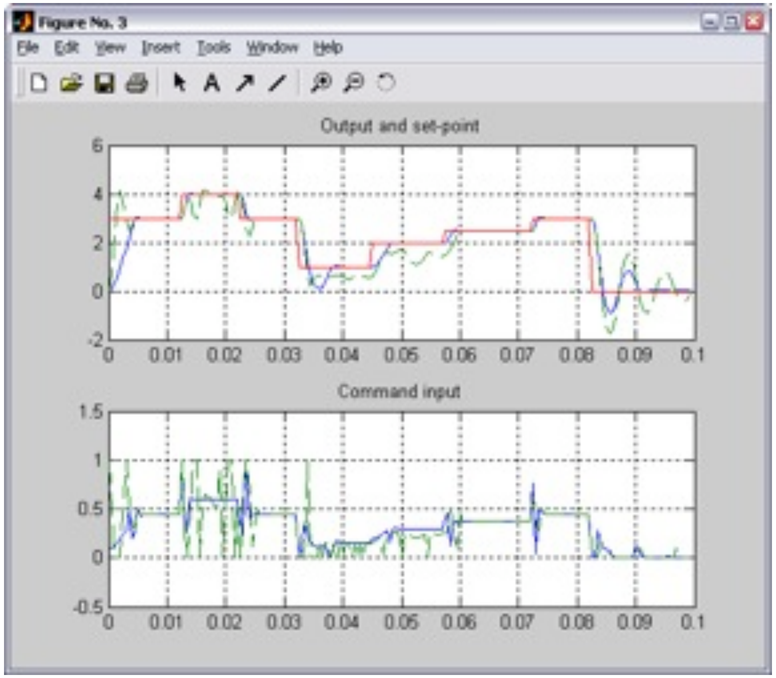
N=60 regions out of 671 kept

Region reduction - Interpolation

Suboptimal solution (Bemporad, summer 2007)



N=10 regions out of 671 kept



(Bemporad, summer 2007)

Region reduction - Interpolation

- Use any *approximation technique* to get MPC control law from N samples $u_i = u^*(x_i)$, then check performance and prove stability:

  - Lookup tables (linear interpolation)
  - Neural networks (Parisini, Zoppoli 1995)
  - PWA identification
  - NL identification (Canale, Fagiano, Milanese NMPC'08)

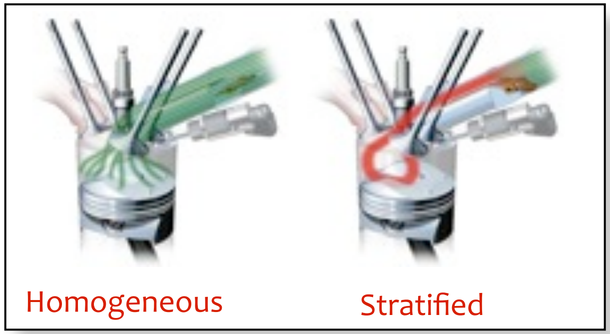
- Use gridding methods from dynamic programming

curse of dimensionality may be an issue

Automotive applications of explicit MPC

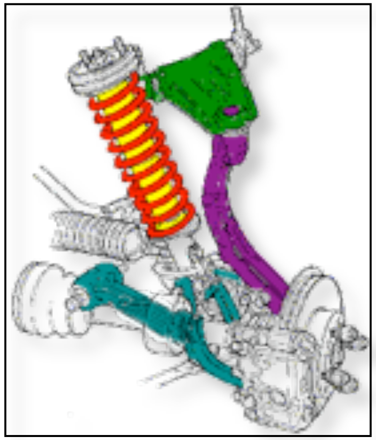
traction control

(Borrelli, Bemporad, Fodor, Hrovat, 2001)



engine control

(N.Giorgetti,G. Ripaccioli, AB, I. Kolmanovsky, D.Hrovat, 2006)

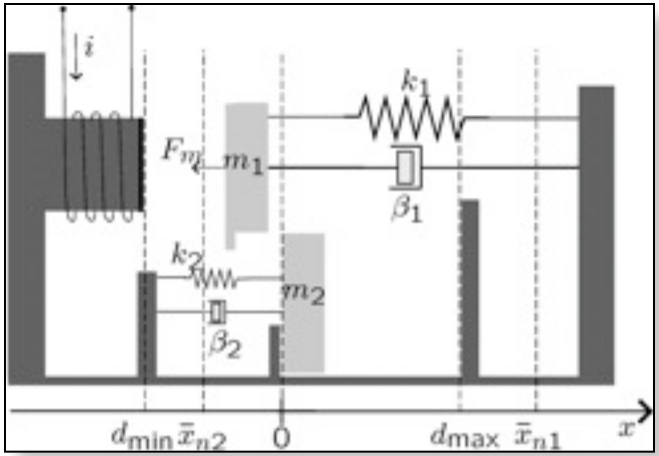


semiactive suspensions

(Giorgetti, Bemporad, Tseng, Hrovat, 2005)

idle speed control

(Di Cairano, Yanakiev, Bemporad, Kolmanovsky, Hrovat, 2008)



magnetic actuators

(Di Cairano, Bemporad, Kolmanovsky, Hrovat, 2006)

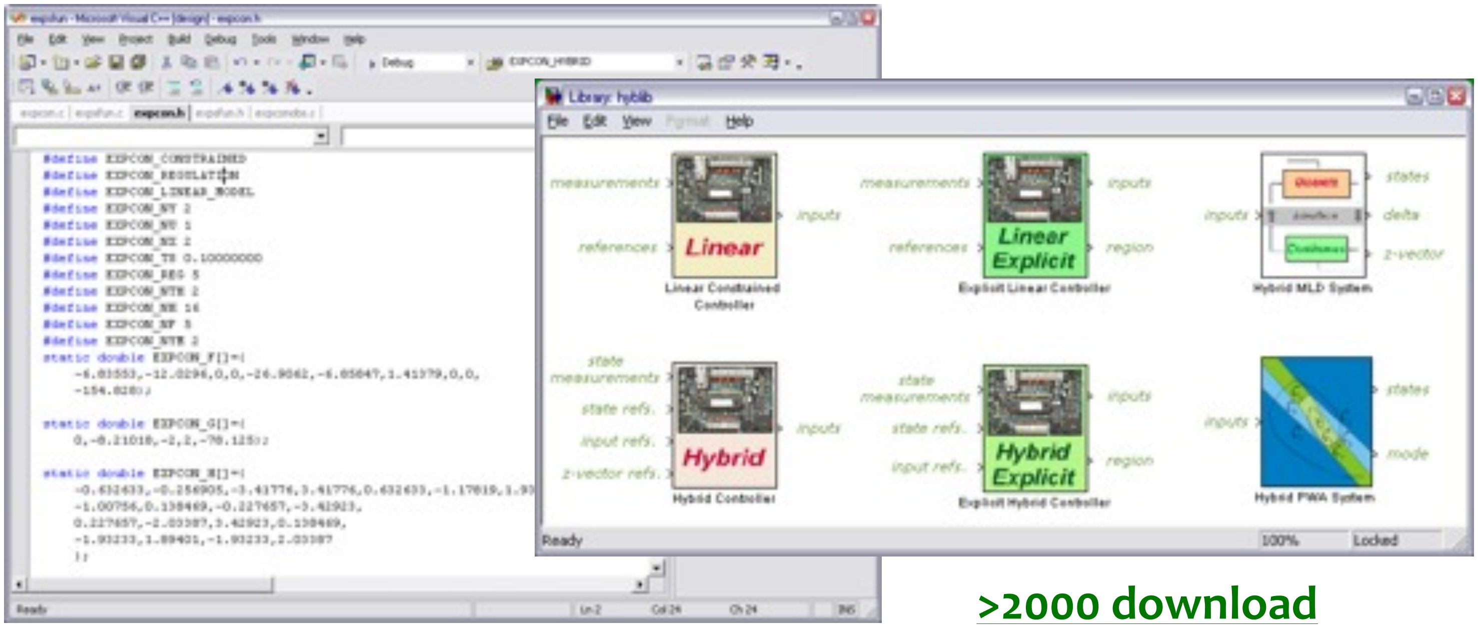
Hybrid Toolbox for Matlab

(Bemporad, 2004)

- Computer-aided hybrid MPC design in Matlab/Simulink
- Explicit controller synthesis and C-code generation

Support:

FORD RESEARCH LABORATORY



>2000 download requests in 4 years

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

- Other tools: **Multi-Parametric Toolbox** (Kvasnica, Grieder, Baotic, Morari)
  `http://control.ee.ethz.ch/~mpt`

Explicit MPC for idle speed control

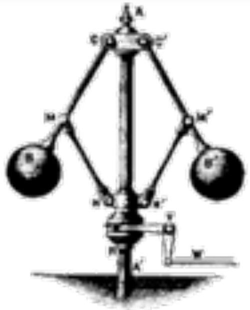
(Di Cairano, Yanakiev, Bemporad, Kolmanovsky, Hrovat, CDC 2008)

- 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)

- Related to most classical problem in control: Watt's governor (1787)

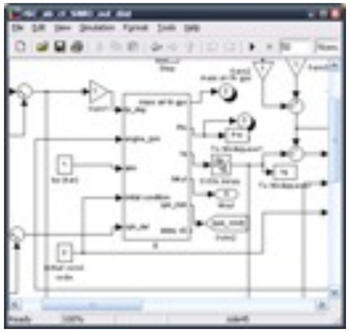


- Problem suitable for MPC design

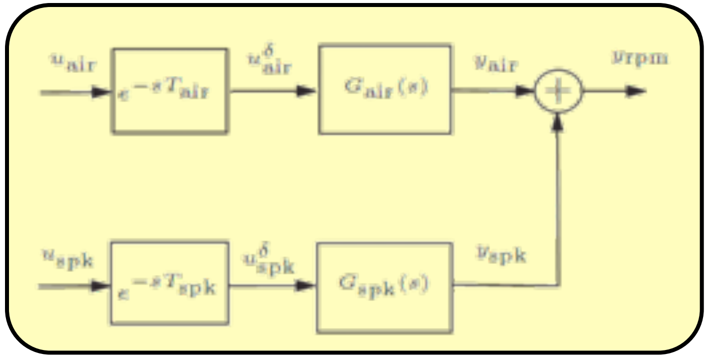
(Hrovat, 1996)

Explicit MPC Design Flow

Simulink model



prediction model



control specs

add integral action

MPC setup
(on-line QP)

closed-loop simulation

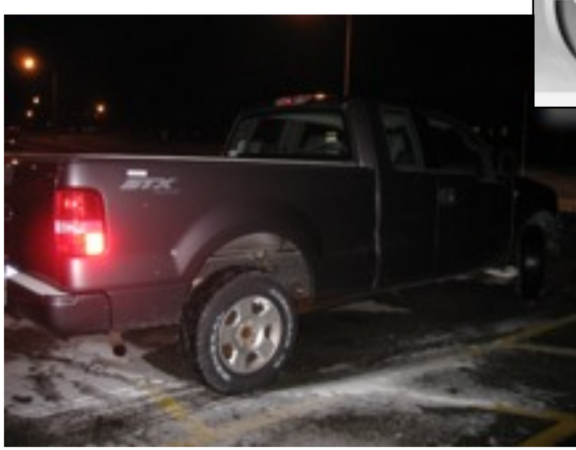
multi-parametric solver

```
#define EXPCON_NTH 2
#define EXPCON_NH 16
#define EXPCON_NF 5
#define EXPCON_NYM 2
static double EXPCON_F[]={
   -6.83553,-12.0296,0,0,-26.9062,-
   -154.828};
```

PWA control law
(C-code)

identification

revise **weights**
& **observer**



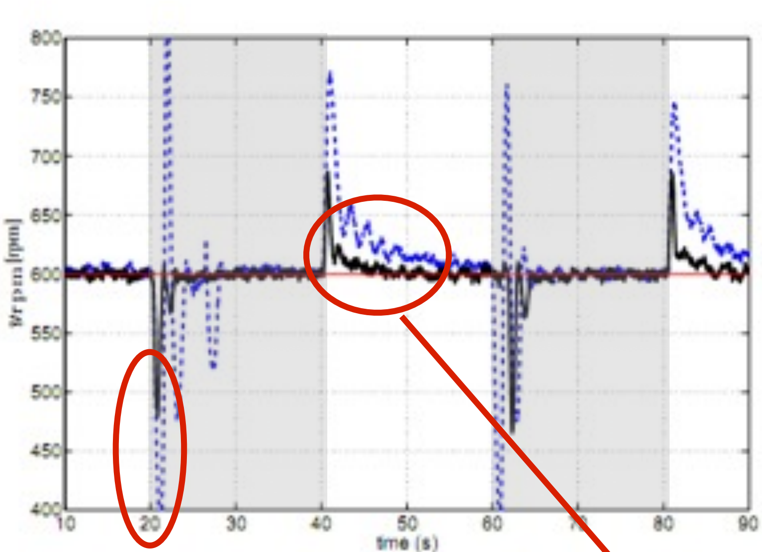
experiments

dSpace platform

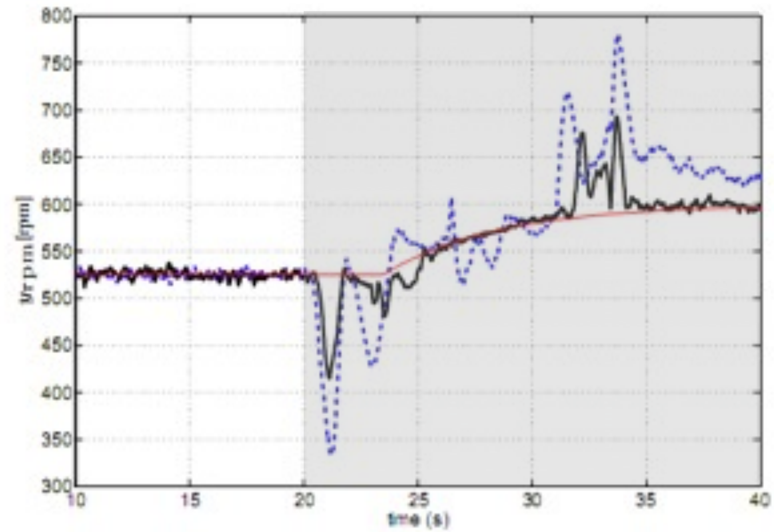
Matlab tool:
Hybrid Toolbox

Explicit MPC for idle speed - Experiments

(Di Cairano, Yanakiev, Bemporad, Kolmanovsky, Hrovat, CDC 2008)



Load torque (power steering)



Power steering + air conditioning

peak reduced by 50%

convergence 10s faster

explicit MPC

baseline controller (linear)

set-point

mpQP in portfolio optimization

(Bemporad, 2008)

Markowitz portfolio optimization

$$\begin{aligned} \min \quad & z'\Sigma z \\ \text{s.t.} \quad & p'z \geq x \\ & [1 \; \ldots \; 1]\, z = 1 \\ & z \geq 0 \end{aligned}$$

z_i = money invested in asset i

p_i = expected return of asset i

Σ_{ij} = covariance of assets i‹ j

x = expected minimum return of portfolio

Objective: minimize variance (=risk)

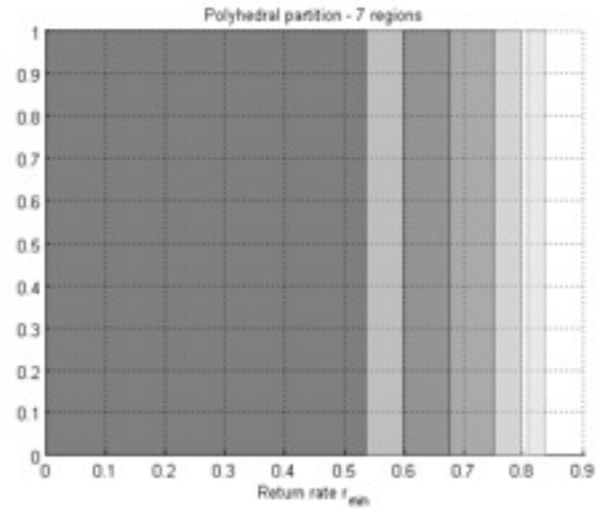
Constraint: guarantee a minimum expected return

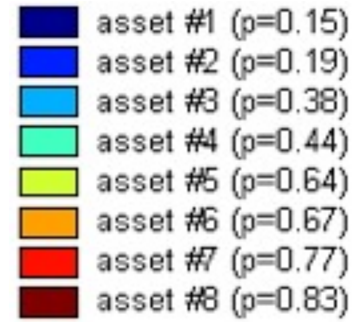
mpQP in portfolio optimization

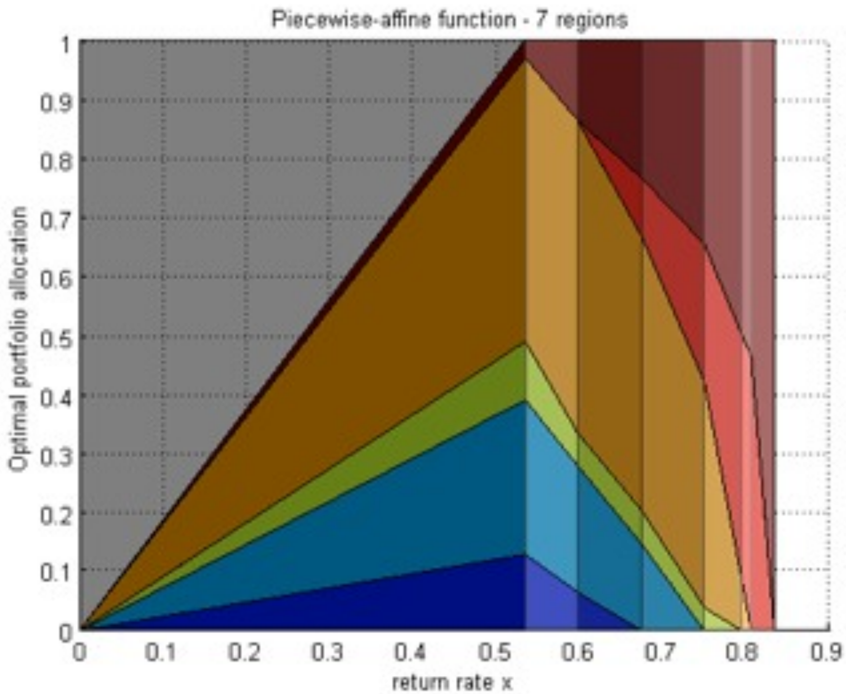
(Bemporad, 2008)

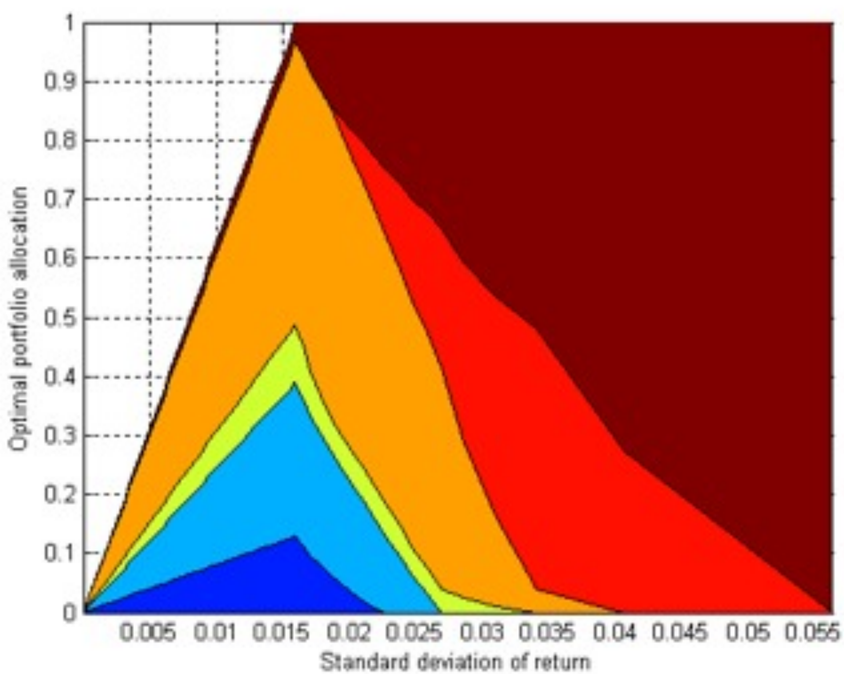
Multiparametric QP solution

$$
\begin{aligned}
\min \quad & z'\Sigma z \\
\text{s.t.} \quad & p'z \geq x \\
& [1 \ \dots \ 1]\, z = 1 \\
& z \geq 0
\end{aligned}
$$







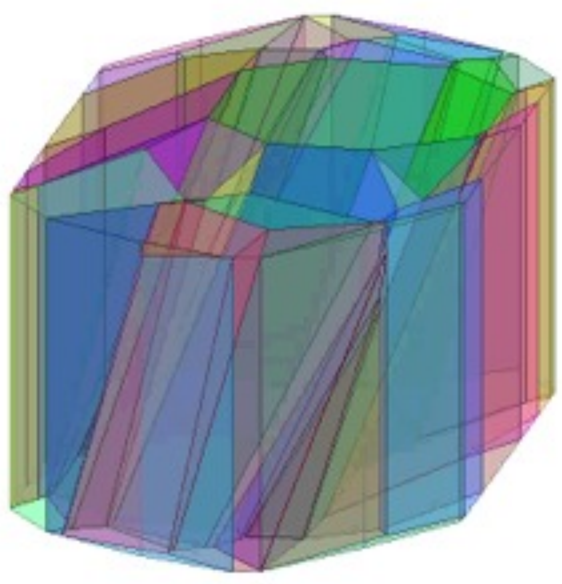


Conclusions and open issues

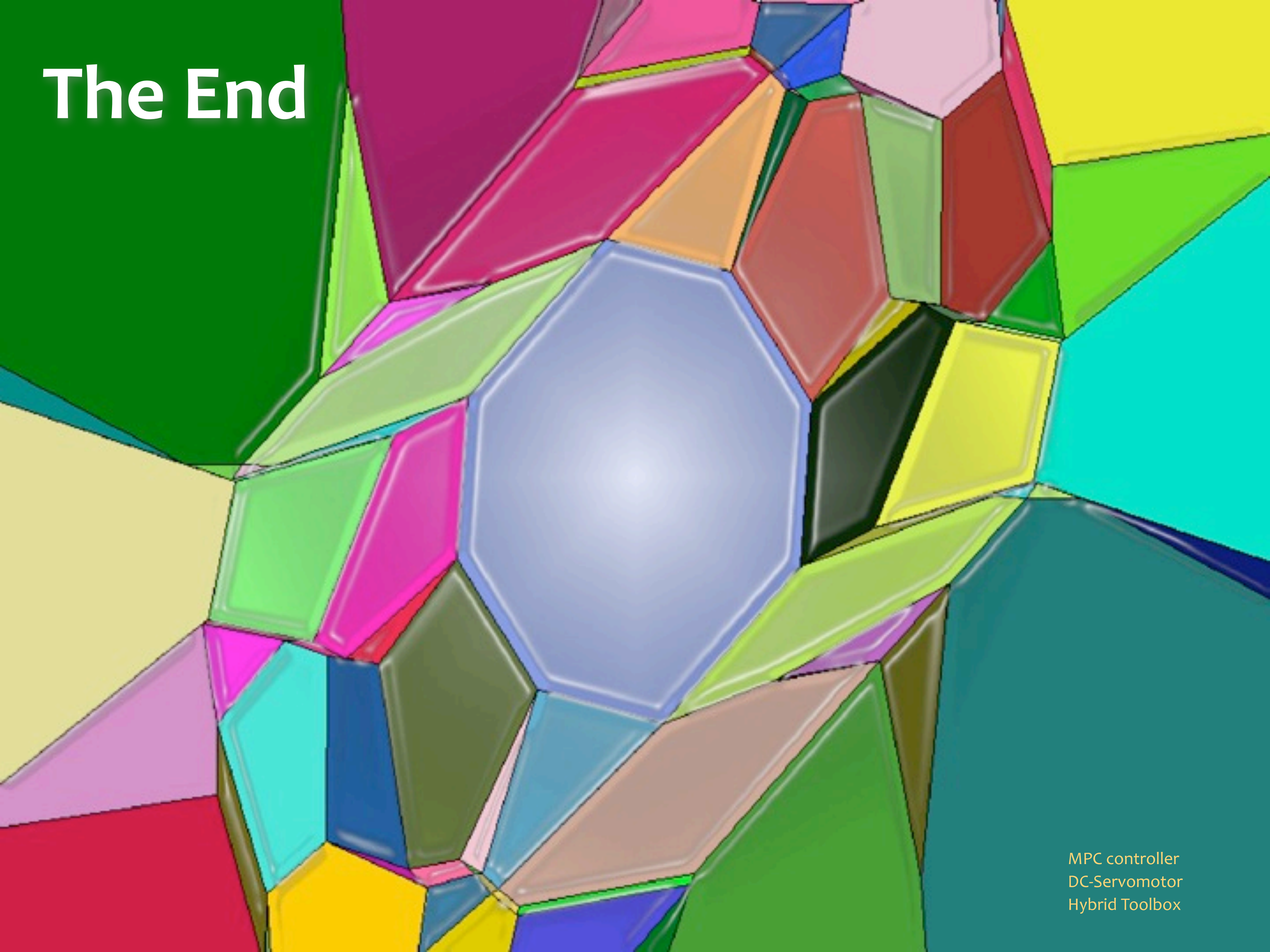
- Linear and hybrid MPC: exact solutions well studied
- Explicit MPC not good for big problems, but extremely good for small size / fast-sampling problems (e.g.: automotive)
- Possible research directions:
  - **Suboptimal solutions**: why getting the (complex) exact solution if MPC weights are usually chosen very roughly ?
  - **Semi-explicit approaches**: get off-line a simpler *partial* solution, do a little optimization on-line
  - **More robust multiparametric solvers**: need better polyhedral computation (e.g.: exact arithmetics)
    - **Piecewise-nonlinear solutions**: PWA may not be the best representation in all cases
    - **Spread tools in industrial practice**

Acknowledgements

A. Alessio
M. Baotic
D. Bernardini
F. Borrelli
S. Di Cairano
C. Filippi
N. Giorgetti
M. Heemels
D. Hrovat
T.A. Johansen
I. Kolmanovsky
M. Lazar
D. Mignone
M. Morari
D. Munoz de la Pena
G. Pannocchia
G. Ripaccioli
P. Tøndel
F.D. Torrisi
E. Tseng



The End



MPC controller
DC-Servomotor
Hybrid Toolbox

Acknowledgements

A. Alessio
M. Baotic
D. Bernardini
F. Borrelli
S. Di Cairano
C. Filippi
N. Giorgetti
M. Heemels
D. Hrovat
T.A. Johansen
I. Kolmanovsky
M. Lazar
D.Q. Mayne
D. Mignone
M. Morari
D. Munoz de la Pena
G. Ripaccioli
P. Tøndel
F.D. Torrisi
E. Tseng

Announcement

- **3rd WIDE PhD School on Networked Control Systems**
  (Siena, mid July 2009)

European project FP7
INFSO-ICT-224168