Explicit Model Predictive Control

Alberto Bemporad

http://www.imtlucca.it/alberto.bemporad

IMT Institute for Advanced Studies Lucca

Linear MPC - Unconstrained case

- Linear prediction model:

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

$$x_0 = x(t)$$

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

- Performance index

$$J(x_0, U) = \min_U x_N' P x_N + \sum_{k=0}^{N-1} x_k' Q x_k + u_k' R u_k$$

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

- Goal: find sequence $U^* = \begin{bmatrix} u_0^* \\ u_1^* \\ \vdots \\ u_{N-1}^* \end{bmatrix}$ that steers the state to the origin "optimally"

[computation of cost function]

$$J(x_0,U) = x_0'Qx_0 + \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-1} \\ x_N \end{bmatrix}' \overbrace{\begin{bmatrix} Q & 0 & 0 & \dots & 0 \\ 0 & Q & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & Q & 0 \\ 0 & 0 & \dots & 0 & P \end{bmatrix}}^{\bar{Q}} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-1} \\ x_N \end{bmatrix} + \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}' \underbrace{\begin{bmatrix} R & 0 & \dots & 0 \\ 0 & R & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & R \end{bmatrix}}_{\bar{R}} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}$$

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \overbrace{\begin{bmatrix} B & 0 & \dots & 0 \\ AB & B & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \dots & B \end{bmatrix}}^{\bar{S}} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix} + \underbrace{\begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}}_{\bar{T}} x_0$$

$$\begin{aligned} J(x_0,U) &= x_0'Qx_0 + (\bar{S}U + \bar{T}x_0)'\bar{Q}(\bar{S}U + \bar{T}x_0) + U'\bar{R}U \\ &= \frac{1}{2}U' \underbrace{2(\bar{R} + \bar{S}'\bar{Q}\bar{S})}_{H} U + x_0' \underbrace{2\bar{T}'\bar{Q}\bar{S}}_{F} U + \frac{1}{2}x_0' \underbrace{2(Q + \bar{T}'\bar{Q}\bar{T})}_{Y} x_0 \end{aligned}$$

Linear MPC - Unconstrained case

$$J(x_0,U) = \frac{1}{2}U'HU + x_0'FU + \frac{1}{2}x_0'Yx_0 \qquad U = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}$$

The optimum is obtained by zeroing the gradient

$$\nabla_U J(x_0,U) = HU + F'x_0 = 0$$

and hence $U^* = \begin{bmatrix} u_0^* \\ u_1^* \\ \vdots \\ u_{N-1}^* \end{bmatrix} = -H^{-1}F'x_0$ **(batch least squares)**

Alternative approach: use dynamic programming to find U^* (Riccati iterations)

Linear MPC - Constrained case

- Linear prediction model:

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

$$x_0 = x(t)$$

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

- Constraints to enforce:

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

- Constrained optimal control problem (quadratic performance index):

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

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

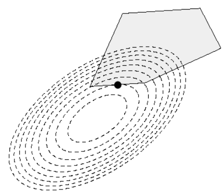
Linear MPC - Constrained case

- Optimization problem:

$$V(x_0) = \tfrac{1}{2} x_0' Y x_0 + \min_U \ \tfrac{1}{2} U' H U + x_0' F U \quad \text{(quadratic)}$$

$$\text{s.t.} \quad GU \leq W + S x_0 \quad \text{(linear)}$$

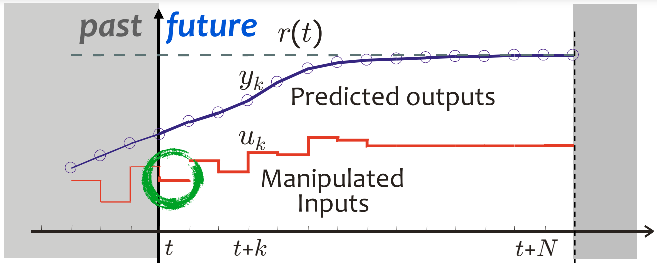
Convex QUADRATIC PROGRAM (QP)



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



At time t:

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

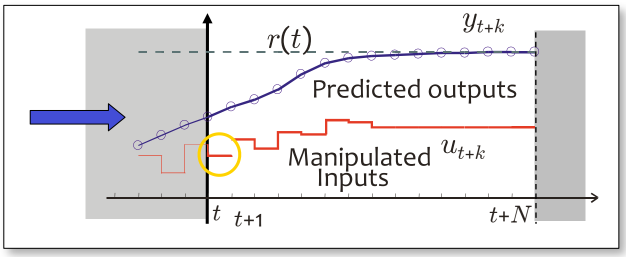
$$\min_U \ \frac{1}{2}U'HU + x'(t)FU$$
$$\text{s.t.} \quad GU \leq W + Sx(t)$$

Let $U=\{u^*(0),\ldots,u^*(N\text{-}1)\}$ be the solution

- Apply only $u(t)=u^*(0)$ and discard the remaining optimal inputs
- Repeat optimization at time $t+1$. And so on ...

Unconstrained Linear MPC

- Assume no constraints
- Problem to solve on-line:



$$\min_U \ J(x(t),U) = \frac{1}{2}U'HU + x'(t)FU$$

- Solution: $\nabla_U J(x(t),U) = HU + F'x(t) = 0$

$$\longrightarrow \quad U^* = -H^{-1}F'x(t) \qquad U = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}$$

$$\Longrightarrow \quad u(t) = -[I \ 0 \ \ldots \ 0]H^{-1}Fx(t) \triangleq Kx(t)$$

Unconstrained linear MPC is nothing else than a standard linear state-feedback law !

MPC and Linear Quadratic Regulation (LQR)

- Consider again the MPC control law based on minimizing

$$J(x_0, U) = \min_U x_N' P x_N + \sum_{k=0}^{N-1} x_k' Q x_k + u_k' R u_k$$

Jacopo Francesco Riccati (1676 - 1754)

- Given $R = R' \succ 0,\ Q = Q' \succeq 0$, choose matrix P by solving the Riccati equation

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

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

MPC and Linear Quadratic Regulation (LQR)

- Consider again the constrained MPC law based on minimizing

$$\min_U \quad x_N' P x_N + \sum_{k=0}^{N-1} x_k' Q x_k + u_k' R u_k$$

$$\begin{aligned} \text{s.t.} \quad & u_{\min} \leq u_k \leq u_{\max}, \quad k = 0, \ldots, N-1 \\ & y_{\min} \leq y_k \leq y_{\max}, \quad k = 1, \ldots, N \\ & u_k = K x_k, \quad k = N_u, \ldots, N-1 \end{aligned}$$

Jacopo Francesco Riccati (1676 - 1754)

- Choose matrix P and terminal gain K by solving the LQR problem

$$\begin{aligned} K &= -(R + B'PB)^{-1}B'PA \\ P &= (A + BK)'P(A + BK) + K'RK + Q \end{aligned}$$

- In a polyhedral region around the origin **constrained MPC = LQR** (for any choice of the prediction and control horizons N,N_u)

(Chmielewski, Manousiouthakis, 1996)(Scokaert and Rawlings, 1998)

- The larger the horizon, the larger the region where MPC=LQR

Linear MPC - Tracking

- Objective: make the output $y(t)$ track a reference signal $r(t)$
- Idea: parameterize the problem using input increments

$$\boxed{\Delta u(t) = u(t) - u(t-1)} \Rightarrow u(t) = u(t-1) + \Delta u(t)$$

- Extended system: let $x_u(t)=u(t\text{-}1)$

$$\begin{cases} x(t+1) &= Ax(t) + Bu(t-1) + B\Delta u(t) \\ x_u(t+1) &= x_u(t) + \Delta u(t) \end{cases}$$

$$\Downarrow$$

$$\begin{cases} \begin{bmatrix} x(t+1) \\ x_u(t+1) \end{bmatrix} &= \begin{bmatrix} A & B \\ 0 & I \end{bmatrix} \begin{bmatrix} x(t) \\ x_u(t) \end{bmatrix} + \begin{bmatrix} B \\ I \end{bmatrix} \Delta u(t) \\ y(t) &= \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ x_u(t) \end{bmatrix} \end{cases}$$

Again a linear system with states $x(t)$, $x_u(t)$ and input $\Delta u(t)$

Linear MPC - Tracking

- Optimal control problem (quadratic performance index):

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

optimization vector

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

- Note: $\|Wz\|^2 = (Wz)'(Wz) = z'(W'W)z = z'Qz$

$\Rightarrow$ same formulation as before (W=Cholesky factor of weight matrix Q)

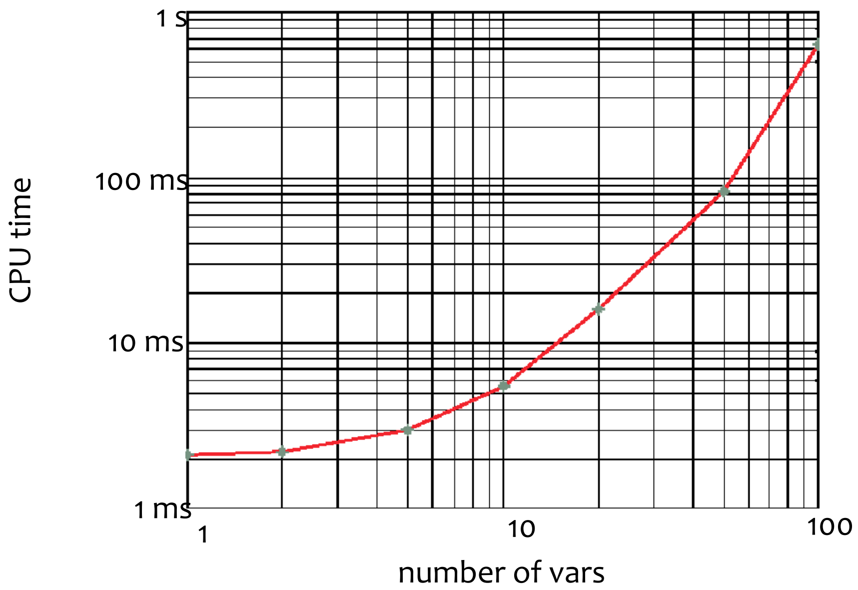
- Optimization problem:

Convex Quadratic Program (QP)

$$\begin{aligned} \min_{\Delta U} \quad & J(\Delta U, x(t)) = \tfrac{1}{2}\Delta U' H \Delta U + [x'(t)\ r'(t)\ u'(t-1)] F \Delta U \\ \text{s.t.} \quad & G\Delta U \le W + S \begin{bmatrix} x(t) \\ r(t) \\ u(t-1) \end{bmatrix} \end{aligned}$$

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

QP Complexity



George Dantzig
(1914 - 2005)

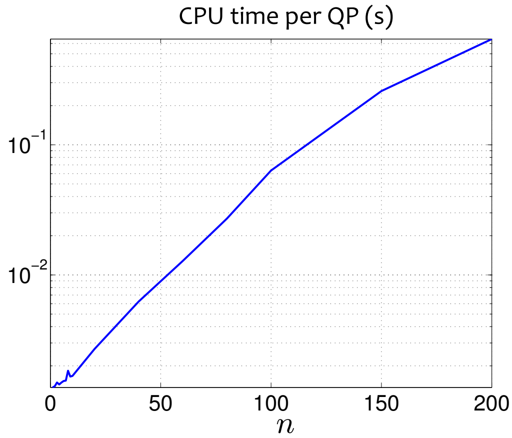
- Non-optimized Dantzig's routine (MPC Toolbox for MATLAB)

(Ricker, *Ind. Eng. Chem. Process Des. Dev.*, 1985)

- Tests in MATLAB 5.3 on Pentium II 300 MHz

Linear MPC - Numerical example

- Linear MPC of random square MIMO systems
  - n outputs, n inputs, $3n$ states
  - prediction horizon N=10, control horizon m=2
  - constraints: $-1 \leq u_k \leq 1, \quad -1 \leq y_k \leq 1$
  - QP size: $(mn+1)$ variables, $(2Nn+2mn)$ constraints



n	#vars	# constraints	CPU time (s)
1	3	24	0.00136
5	11	120	0.00149
20	41	480	0.00270
100	201	2400	0.06432
150	301	3600	0.25873
200	401	4800	0.64981

Macbook Air 2.13 GHz (this mac !)

Inter Core 2 Duo 4GB RAM

MPC Toolbox 4.0, MATLAB R2011b

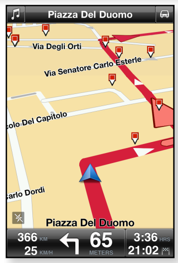
New active set QP in EML (dense matrices)

Large-scale system with **200** inputs and **200** outputs w/ constraints: **less than 1 s** !

- For LSS, MPC **CPU time** is not a problem. Getting the **model** is the largest effort !

Pros and cons of on-line optimization

✓ Continuous update of best decision under constraints, reaction to unexpected events (disturbances)



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

✖ Computation time may be too long: ok for large sampling times (>10 ms) but not for fast-sampling applications (< 1 ms).

✖ Requires relatively expensive hardware (microprocessor)

✖ Software complexity: solver code must be embedded

✖ Real-time: Worst-case CPU time often hard to estimate

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

On-Line vs off-Line optimization

$$\min_U \ \tfrac{1}{2}U'HU + x'(t)_0 FU + \tfrac{1}{2}x'(t)Yx(t)$$
$$\text{s.t.} \quad GU \leq W + Sx(t)$$

- **On-line** optimization: given $x(t)$ solve the problem at each time step t (the control law u=$u(x)$ is **implicitly** defined by the QP solver)

→ Quadratic Program (QP)

- **Off-line** optimization: solve the QP **for all** $x(t)$ to find the control law u=$u(x)$ **explicitly**

→ multi-parametric Quadratic Program (mp-QP)

Multiparametric programming problem

Given the optimization problem

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

and a set X of parameters, determine:

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

Multiparametric Quadratic Programming

(Bemporad et al., 2002)

$$\begin{array}{ll} \min\limits_{U} & \frac{1}{2}U'HU + x'(t)_0 FU + \frac{1}{2}x'(t)Yx(t) \\ \text{s.t.} & GU \leq W + Sx(t) \end{array}$$

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

- 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 MPC problem !

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

KKT optimality conditions

$$\begin{array}{ll} \min_U & f(U) \\ \text{s.t.} & g_i(U) \leq 0, \quad \forall i = 1, \dots, m \\ & h_j(U) = 0, \quad \forall j = 1, \dots, p \end{array}$$

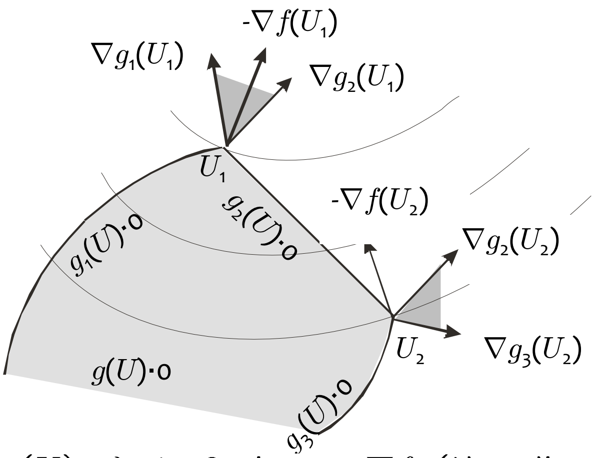
KKT Conditions (necessary) Let U^* be a feasible solution, let $I = \{i : g_i(U^*) = 0\}$. Suppose f and g_i, h_j differentiable at U^*, Suppose $\nabla g_i(U^*)$, $\nabla h_j(U^*)$ linearly independent for $i \in I$.

Then, if U^* is optimal, there exists vector of **Lagrange multipliers** $\lambda \in \mathbb{R}^m$, $\nu \in \mathbb{R}^p$ such that

$$\begin{aligned} \nabla f(U^*) + \sum_{i=1}^{m} \lambda_i \nabla g_i(U^*) + \sum_{j=1}^{p} \nu_j \nabla h_j(U^*) &= 0 && \text{(1a)} \\ \lambda_i g_i(U^*) &= 0, \quad \forall i = 1, \dots, m && \text{(1b)} \\ \lambda_i &\geq 0 && \text{(1c)} \\ g_i(U^*) &\leq 0 && \text{(1d)} \\ h_j(U^*) &= 0, \quad \forall j = 1, \dots, p && \text{(1e)} \end{aligned}$$

When f, g_i are convex functions and h_j are linear, the condition is also sufficient

KKT - Geometric interpretation



- $-\nabla f(U) = \sum_{i \in I} \lambda_i \nabla g_i(U)$, $\lambda_i \geq 0$, i.e. $-\nabla f$ (the direction of maximum decrease of f) belongs to the convex cone spanned by ∇g_i's, where g_i is a binding constraint

- f can only decrease if some constraint is violated, Therefore U must be optimal (see point U_1)

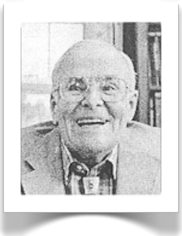
- Conversely, if some $\lambda_i < 0$ one can move within the set of feasible points $g(U) \leq 0$ and decrease f, which implies that U is not optimal (see point U_2).

KKT conditions for QP

$$\min_U \quad f(U) \triangleq \frac{1}{2}U'HU + c'U$$
$$\text{s.t.} \quad AU \leq b$$

$U \in \mathbb{R}^n,\ H \succeq 0 \in \mathbb{R}^{n\times n}$
$A \in \mathbb{R}^{m\times n}$.

Harold W. Kuhn (1925 -)



Albert W. Tucker (1905 - 1995)

William Karush (1917 - 1997)

$\nabla f(U) = HU + c$
$g_i(U) = A_i'U - b_i$ (A_i' is the i-th row of A)
$\nabla g_i(U) = A_i$

$$\begin{aligned} HU + c + A'\lambda &= 0 \\ \lambda_i(A_i'U - b_i) &= 0 \\ \lambda &\geq 0 \\ AU - b &\leq 0 \end{aligned}$$

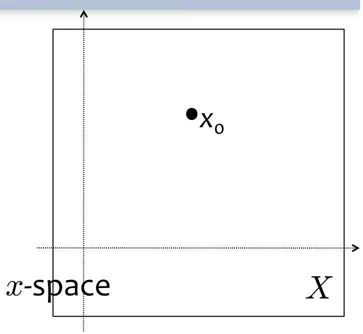
Linearity of the Solution

Fix $x_0 \in X$

⇨ solve QP to find $U^*(x_0),\ \lambda^*(x_0)$

⇨ identify active constraints at $U^*(x_0)$

⇨ 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:

$$\begin{array}{ll} (1)\ HU + Fx + G'\lambda = 0, & (2)\ \tilde{G}U - \tilde{W} - \tilde{S}x = 0 \\ (3)\ \lambda_i(G^iU - W^i - S^ix) = 0, & (4)\ \hat{G}U \leq \hat{W} + \hat{S}x \\ (5)\ \tilde{\lambda}_i \geq 0,\ \hat{\lambda}_i = 0 & \end{array}$$

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

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

From (2):

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

➡ In some neighborhood of x_0, λ and U are explicit affine functions of x !

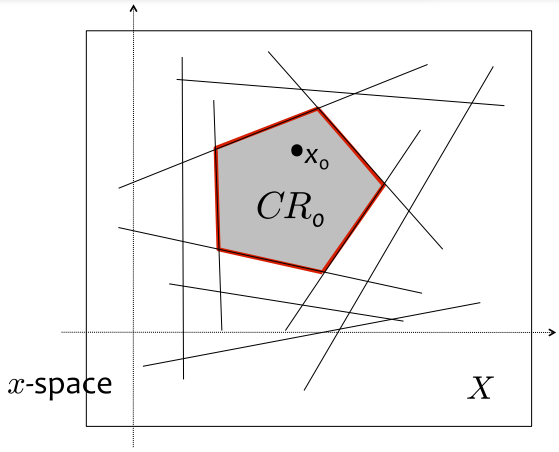
(Zafiriou, 1990)

Critical region

- Impose primal and dual feasibility:

$$\hat{G}U(x) \leq \hat{W} + \hat{S}x \quad \text{from (4)}$$
$$\tilde{\lambda}(x) \geq 0 \quad \text{from (5)}$$

➡ linear inequalities in x !



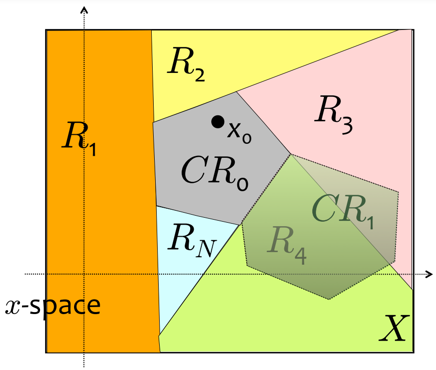
- Remove redundant constraints (this requires solving LP's):

➡ 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 solver #1



Method #1: Split and proceed iteratively

(Bemporad, Morari, Dua, Pistikopoulos, 2002)

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

$$R_i = \{x \in X : \ \mathcal{A}^i x > \mathcal{B}^i, \ \mathcal{A}^j x \leq \mathcal{B}^j, \ \forall j < i\}$$

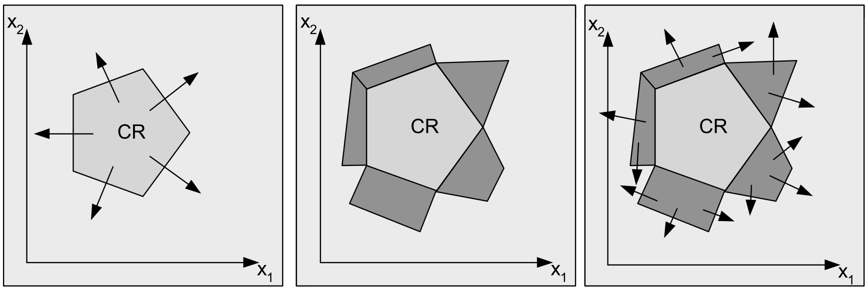
Note: while CR_i is characterizing a set of active constraints, R_i is not

➡ 1) Use the above splitting only as a search procedure, don't split the CR

2) Remove duplicates of CR already found

Multiparametric QP solver #2, #3

(Tøndel, Johansen, Bemporad, 2003)



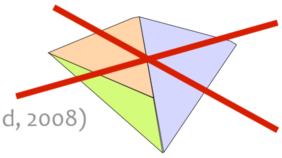
Active set of neighboring region obtained by adding/removing active constraints based on knowledge of the type of crossed hyperplane in x-space

$\hat{G}^i U(x) \leq \hat{W}^i + \hat{S}^i x$ → constraint $\#i$ **added** to active set (to maintain **feasibility** of solution)

$\tilde{\lambda}_j(x) \geq 0$ → constraint $\#j$ **withdrawn** from active set (to maintain **optimality** of solution)

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 $U^* : X^* \mapsto \mathbb{R}^r$ is piecewise affine and continuous.*

$$U^*(x) = \arg\min_U \ \tfrac{1}{2}U'HU + x'F'U \quad \text{subj. to } GU \leq W + Sx$$

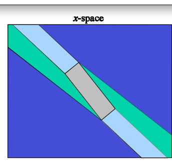
continuous, piecewise affine

$$V^*(x) = \tfrac{1}{2}x'Yx + \min_U \ \tfrac{1}{2}U'HU + x'F'U \quad \text{subj. to } GU \leq W + Sx$$

convex, continuous, piecewise quadratic, C¹ (if no degeneracy)

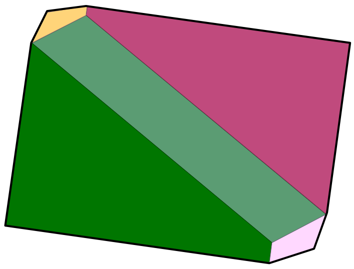
Corollary: The linear MPC controller is a continuous piecewise affine function of the state

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



Set of Feasible Parameters X^*

Why is the set X^* of parameters for which the QP problem is solvable a convex polyhedral set ?



$$X^* = \{x : \exists U \text{ such that } GU \leq W + Sx\}$$

X^* is the projection of a polyhedron onto the parameter space. Therefore X^* is a polyhedron.

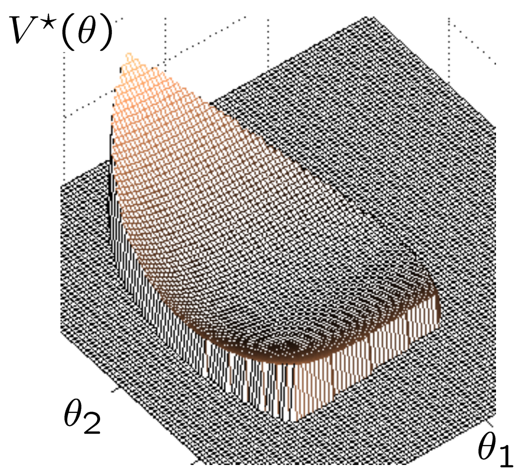
$$X^* = \text{Proj}_x \left\{ \begin{bmatrix} U \\ x \end{bmatrix} : \begin{bmatrix} G & -S \end{bmatrix} \begin{bmatrix} U \\ x \end{bmatrix} \leq W \right\}$$

Multiparametric Convex Programming

$$\begin{array}{ll} \min_x & f(U,x) \\ \text{s.t.} & g_i(U,x) \leq 0 \quad (i=1,\dots,p) \\ & AU + Bx + d = 0 \end{array}$$

Lemma *Let f, g_i be jointly convex functions of (U,x) $(\forall i = 1,\dots,p)$. Then X^* is a convex set and $V^\star$ is a convex function of x.*

(Mangasarian, Rosen, 1964)

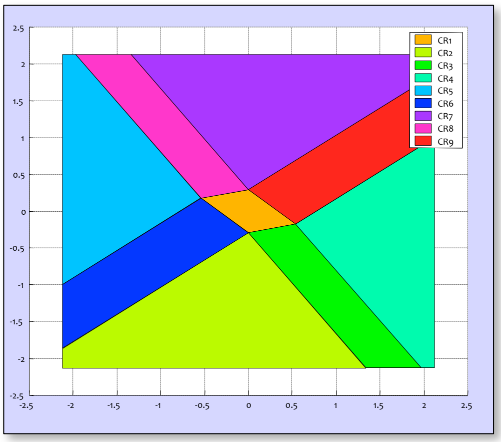


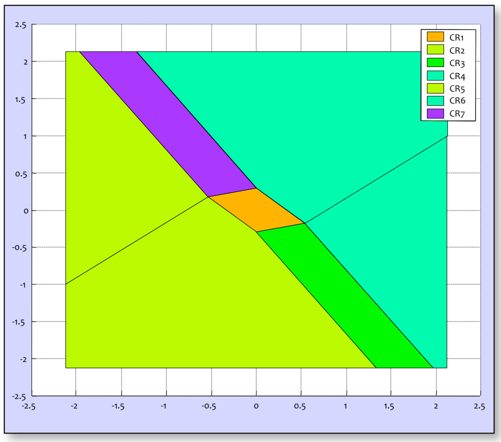
V^* and X^* may not be easy to express analytically

(approximate solutions possible)

(Bemporad, Filippi, 2003)

Complexity Reduction





$$U(x) \triangleq [u_0'(x) \; u_1'(x) \; \dots \; u_{N-1}'(x)]'$$

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

(Bemporad, Fukuda, Torrisi, *Computational Geometry*, 2001)

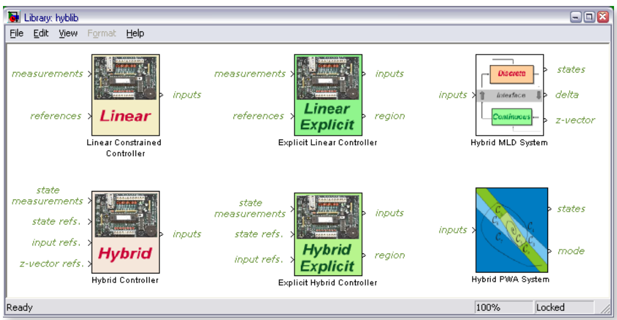
Hybrid Toolbox for MATLAB

(Bemporad, 2003-2011)

Features:

- **Hybrid** models: design, simulation, verification
- Control design for linear systems w/ constraints and hybrid systems (on-line optimization via QP/MILP/MIQP)
- **Explicit** MPC control (via multi-parametric programming)
- C-code generation
- Simulink library

Supported by



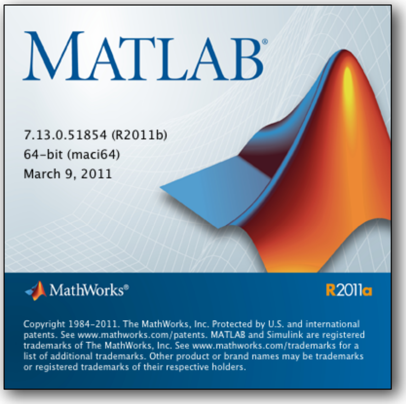
3500+ download requests since October 2004

http://www.ing.unitn.it/~bemporad/hybrid/toolbox

Model Predictive Control Toolbox

(Bemporad, Ricker, Morari, 1998-2011)

- **MPC Toolbox 4.0** (The Mathworks, Inc.)
  - Object-oriented implementation (MPC object)
  - MPC Simulink Library
  - MPC Graphical User Interface
  - Code generation [RTW, xPC Target, dSpace, etc.]
  - Linked to OPC Toolbox, System ID Toolbox, ...



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

Double Integrator Example

- System: $y(t) = \frac{1}{s^2}u(t)$ $\Longrightarrow$ (sampling + ZOH, T_s=1 s)

$$\begin{aligned} x(t+1) &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) \\ y(t) &= \begin{bmatrix} 1 & 0 \end{bmatrix} x(t) \end{aligned}$$

- Constraints: $-1 \leq u(t) \leq 1$

- Control objective: min $\sum_{t=0}^{\infty} y^2(t) + \frac{1}{100}u^2(t)$ $\quad u(k) = K_{LQ}x(k),\ \forall k \geq N = 2$ (LQ gain)

$$\Longrightarrow \left(\sum_{k=0}^{1} y^2(k) + \frac{1}{100}u^2(k) \right) + x'(2) \begin{bmatrix} 2.1429 & 1.2246 \\ 1.2246 & 1.3996 \end{bmatrix} x(2)$$

(solution of algebraic Riccati equation)

- Optimization problem

$$H = \begin{bmatrix} 0.8365 & 0.3603 \\ 0.3603 & 0.2059 \end{bmatrix}, \quad F = \begin{bmatrix} 0.4624 & 1.2852 \\ 0.1682 & 0.5285 \end{bmatrix}$$

(cost function was normalized by max svd(H))

$$G = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix}, \quad W = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad S = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Double Integrator Example

- System: $y(t) = \frac{1}{s^2} u(t)$ sampling + ZOH T_s=1 s

$$\begin{aligned} x(t+1) &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) \\ y(t) &= \begin{bmatrix} 1 & 0 \end{bmatrix} x(t) \end{aligned}$$

- Constraints: $-1 \le u(t) \le 1$

- Control objective: minimize $\sum_{t=0}^{\infty} y^2(t) + \frac{1}{100} u^2(t)$

$$u(t+k) = K_{LQ} x(t+k|t), \ \forall k \ge N_u$$

- Optimization problem: for N_u=2

$$H = \begin{bmatrix} 0.8365 & 0.3603 \\ 0.3603 & 0.2059 \end{bmatrix}, \quad F = \begin{bmatrix} 0.4624 & 1.2852 \\ 0.1682 & 0.5285 \end{bmatrix}$$

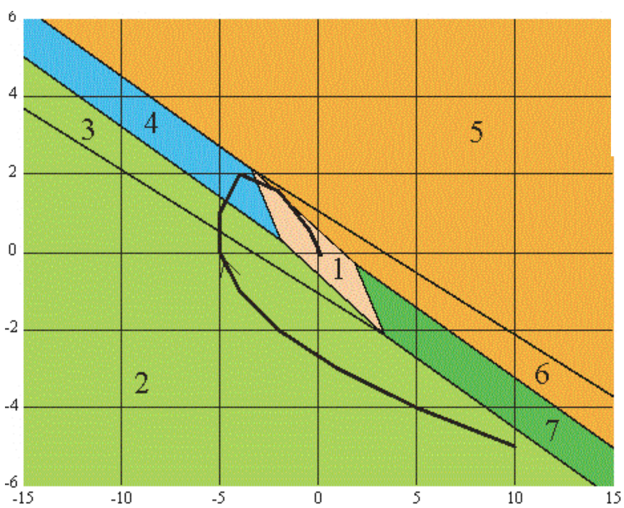
$$G = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix}, \quad W = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad S = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

(cost function is normalized by max svd(H))

mp-QP solution

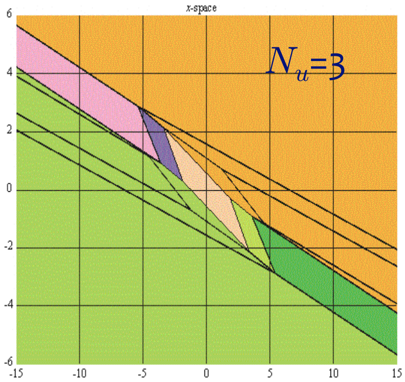
N_u=2

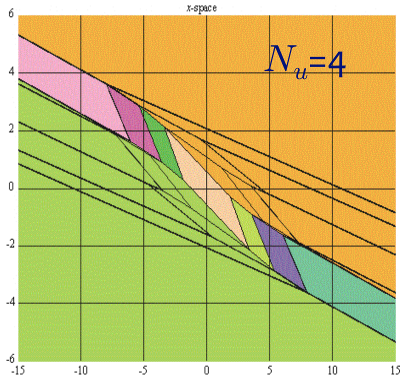
$$u(x) = \begin{cases} \begin{bmatrix} -0.8166 & -1.7499 \end{bmatrix} x & \text{if } \begin{bmatrix} -0.8166 & -1.7499 \\ 0.8166 & 1.7499 \\ 0.6124 & 0.4957 \\ -0.6124 & -0.4957 \end{bmatrix} x \le \begin{bmatrix} 1.0000 \\ 1.0000 \\ 1.0000 \\ 1.0000 \end{bmatrix} & \text{(Region \#1)} \\ 1.0000 & \text{if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

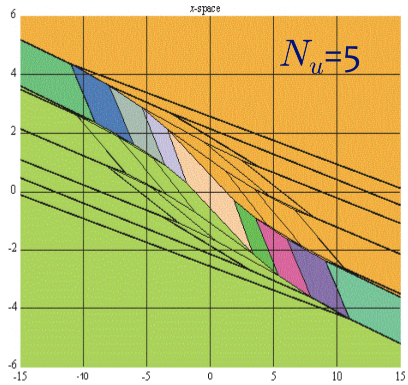


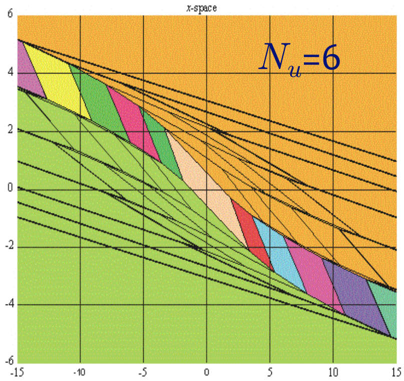
go to demo /**demos/linear/doubleintexp.m** (Hyb-Tbx)

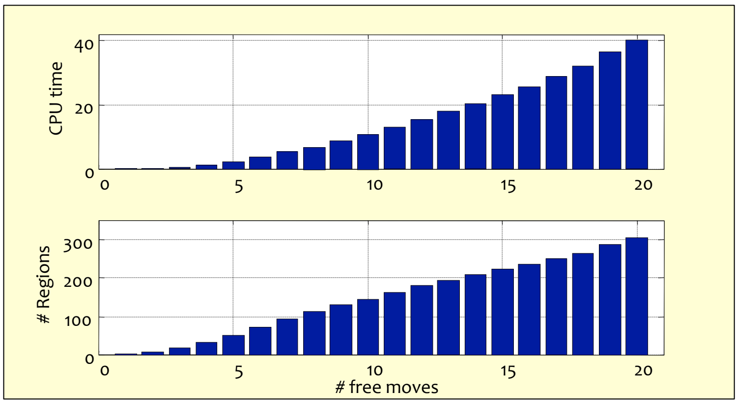
Complexity











(is the number of regions finite for $N_u \to \infty$?)

Complexity

- Worst-case complexity analysis:

$$M \triangleq \sum_{\ell=0}^{q} \binom{q}{\ell} = 2^q$$ combinations of active constraints

- Usually the number of regions is much smaller, as many combinations of active constraints are never feasible and optimal at any parameter vector x
- Strongest dependence on the number q of constraints
- Strong dependence on the number N_u of free moves
- Weak dependence on the number n of parameters x
- Example:

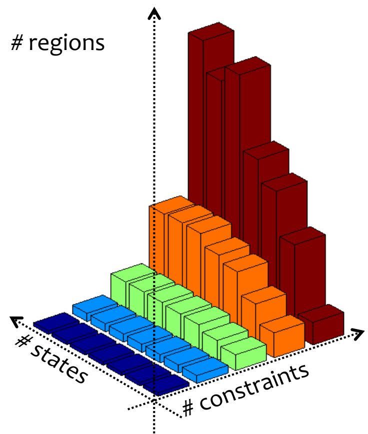
states\horizon	$N=1$	$N=2$	$N=3$	$N=4$	$N=5$
n=2	3	6.7	13.5	21.4	19.3
n=3	3	6.9	17	37.3	77
n=4	3	7	21.65	56	114.2
n=5	3	7	22	61.5	132.7
n=6	3	7	23.1	71.2	196.3
n=7	3	6.95	23.2	71.4	182.3
n=8	3	7	23	70.2	207.9

Data averaged over 20 randomly generated single-input single-output systems subject to input saturation (q=2N)

Complexity

- Number n_r of regions = number of combinations of active constraints at optimality
  - Mainly depends on number q of constraints: $n_r \leq \sum_{h=0}^{q} \binom{q}{h} = 2^q$

    (this is a worst-case estimate, most of the combinations are never optimal !)
  - 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)

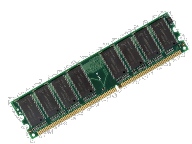
Average time on 100 random 3D parameters (2N constraints)

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

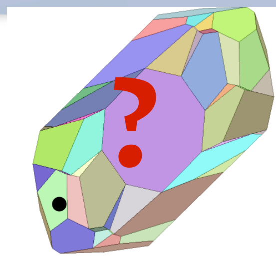
Explicit MPC typically limited to 6-8 free control moves and 8-12 states+references

⇨ Regions can be visited more efficiently than linear search

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



Point location problem

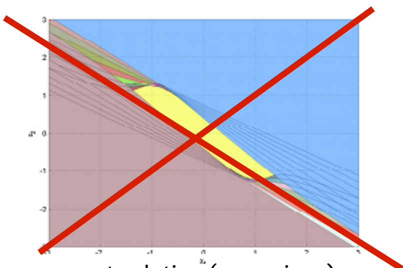


In which region of the partition is $x(t)$?

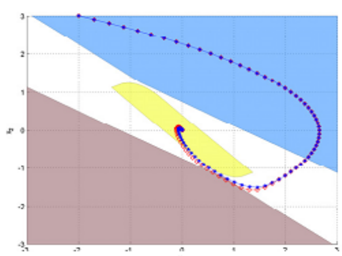
- Store all regions and search linearly through them
- Exploit properties of mpLP solution to locate $x(t)$ from value function (also extended to mpQP) (Baotic, Borrelli, Bemporad, Morari 2008)
- Organize regions on a tree for logarithmic search (Tøndel, Johansen, Bemporad, 2003)
- For mpLP, recast as 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)

Approximate MPC solutions

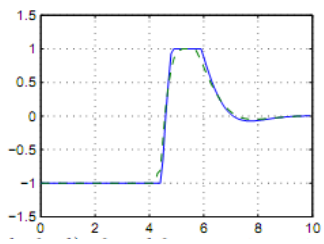
- Change cost function (e.g. minimum time) (Grieder, Morari, 2003)
- Relax optimality conditions (suboptimal mpQP) (Bemporad, Filippi, 2003)
- Use orthogonal trees (suboptimal solutions) (Johansen, Grancharova, 2003+)
- Interpolate solution from a reduced number of regions (Pannocchia, Rawlings, Wright, 2007) (Christophersen, Zeilinger, Jones, Morari, 2007) (Alessio, Bemporad, 2008)



exact solution (99 regions)



3 most visited combinations of active constraints are stored



input trajectory

- Other approaches: (Jones, Morari, 2010) (Kvasnica, Fikar, 2010)

Approximate MPC solutions

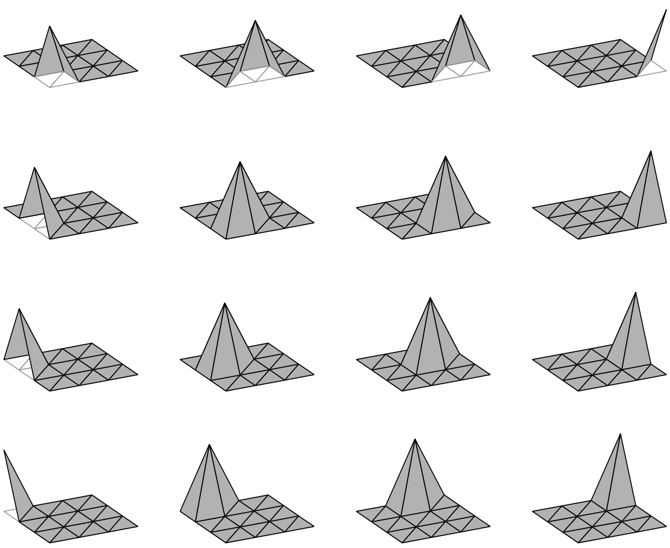
- Use gridding methods from dynamic programming

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

  - Lookup tables (linear interpolation, inverse distance weighting, ...)
  - Neural networks (Parisini, Zoppoli 1995)
  - Hybrid (PWA) system identification
  - NL identification (Canale, Fagiano, Milanese NMPC'08)

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 , IEEE TAC, 2011)



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

approximate MPC law

(Julian, Desages, Agamennoni, 1999)

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

PWA approximation of MPC over simplices

- **Extremely cheap**: **PWAS functions can be directly implemented on FPGA**, or even **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

MIMO system dynamics

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

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

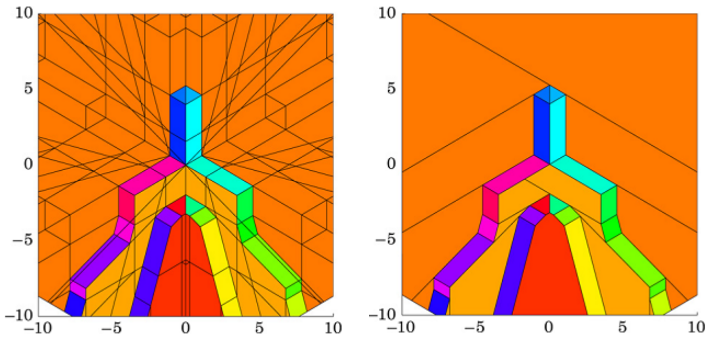
subject to saturation on u and y

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

- Certified closed-loop stability by constructing a **PWA Lyapunov fnc**
- Fulfillment of constraints on inputs (soft constraints on states)

Region reduction

- Join regions more efficiently
  (Geyer, Torrisi, Morari, 2008)



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

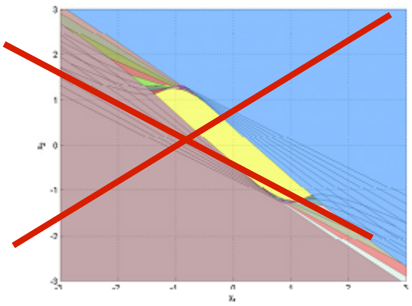
- Change cost function (e.g. minimum time)
  (Grieder, Morari, 2003)
- Relax KKT conditions (suboptimal mpQP)
  (Bemporad, Filippi, 2003)
- Use orthogonal trees (suboptimal solutions)
  (Johansen, Grancharova, 2003)

Region reduction - Interpolation

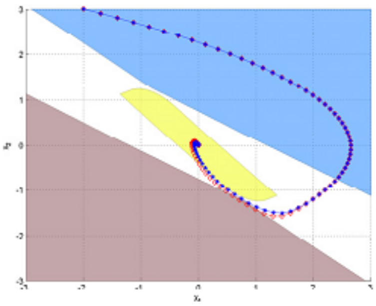
- Interpolate solution from reduced number of regions

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

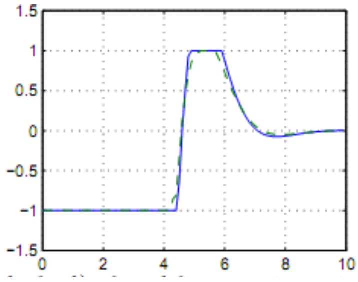
exact solution (99 regions)



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

Example: AFTI-16

- Linearized model:

$$\begin{cases} \dot{x} = \begin{bmatrix} -.0151 & -60.5651 & 0 & -32.174 \\ -.0001 & -1.3411 & .9929 & 0 \\ .00018 & 43.2541 & -.86939 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x + \begin{bmatrix} -2.516 & -13.136 \\ -.1689 & -.2514 \\ -17.251 & -1.5766 \\ 0 & 0 \end{bmatrix} u \\ y = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x, \end{cases}$$

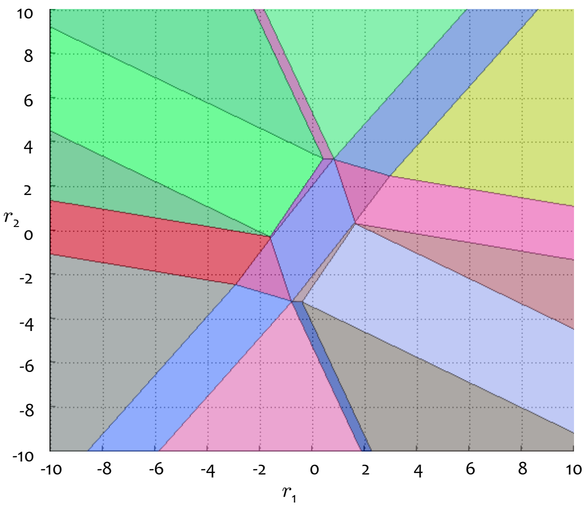
- Inputs: elevator and flaperon angle
- Outputs: attack and pitch angle
- Sampling time: $T_s = .05$ s (+ zero-order hold)
- Constraints: max 25° on both angles
- Open-loop response: unstable

(open-loop poles: $-7.6636, -0.0075 \pm 0.0556j, 5.4530$)

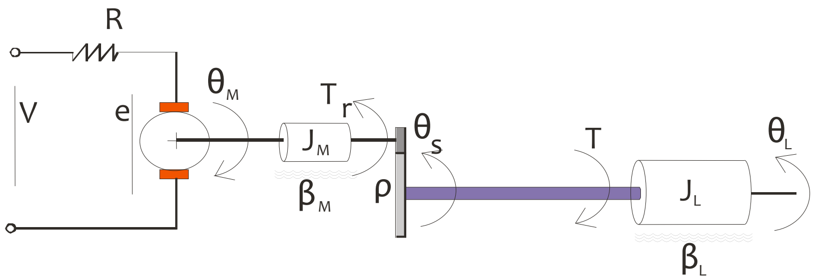
Explicit controller: 8 parameters, 51 regions

go to demo **`linear/afti16.m`** (Hyb-Tbx)

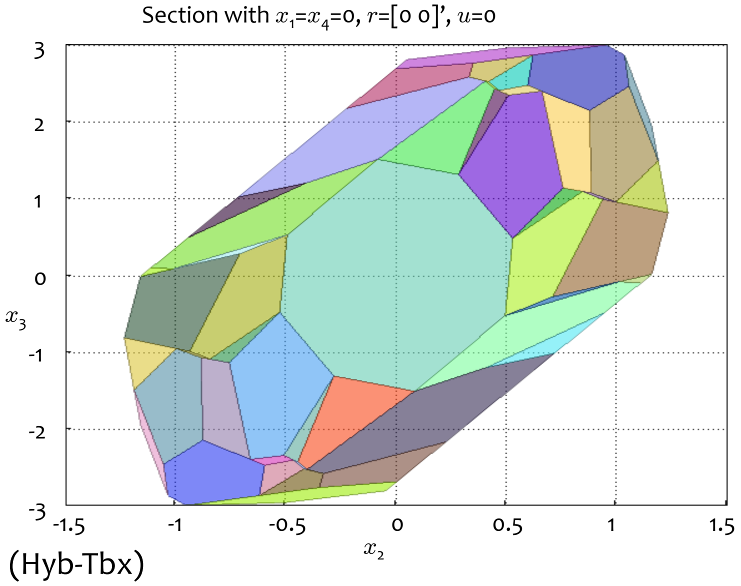
Section with x=[0 0 0 0]', u=[0 0]'



MPC of a DC Servomotor



$$\begin{aligned} N &= 10 \\ N_u &= 2 \\ w_y &= \{1000, 0\} \\ w_{\delta u} &= .05 \\ u &\in [-220, 220] \text{ V} \\ y_2 &\in [-78.5398, 78.5398] \text{ Nm} \end{aligned}$$



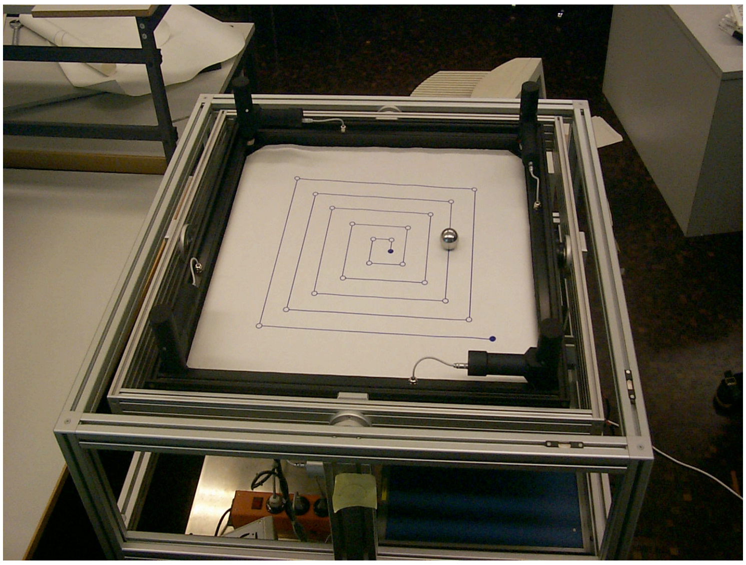
Explicit controller:
7 parameters, 101 regions

go to demo **`linear/dcmotor.m`** (Hyb-Tbx)

MPC Regulation of a Ball on a Plate

Task:

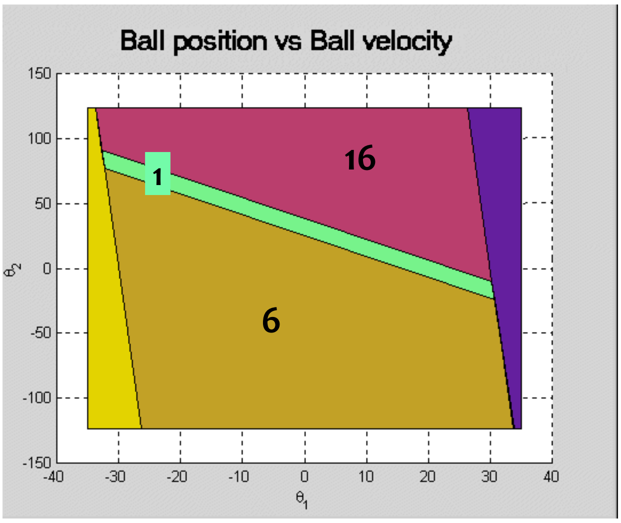
- Tune an MPC controller by simulation, using the ***MPC Simulink Toolbox***
- Get the ***explicit solution*** of the MPC controller.
- Validate the controller on ***experiments***.

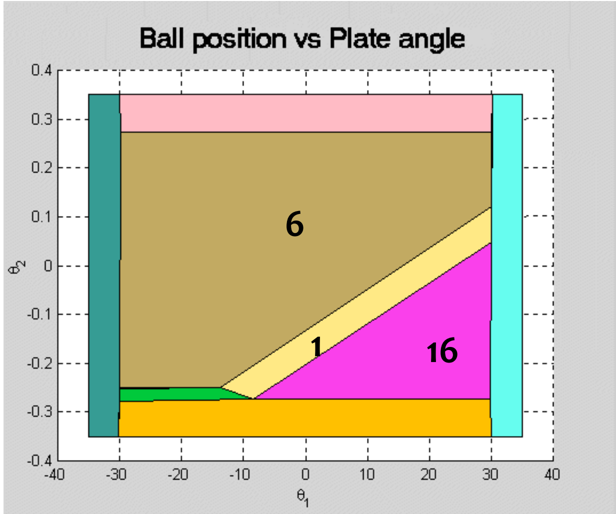


Explicit MPC Solution

Controller: x: 22 Regions, y: 23 Regions

x-MPC: sections at α_x=0, $\dot{\alpha}_x$=0, u_x=0, r_x=18, r_α=0



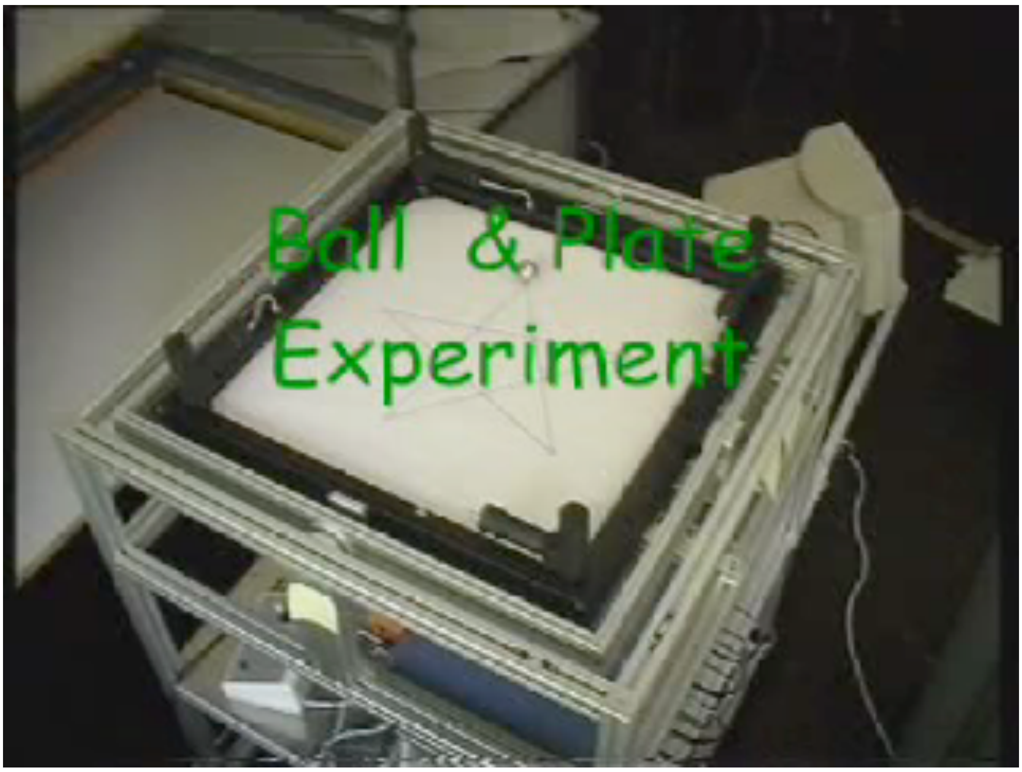


Region 1: LQR Controller (near equilibrium)

Region 6: Saturation at -10

Region 16: Saturation at +10

Ball and plate experiment



Ball and plate experiment

Ball and plate experiment in LEGO, using explicit MPC and Hybrid Toolbox

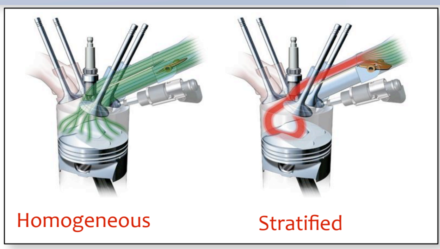
(by Daniele Benedettelli, Univ. of Siena, July 2008)

- 20Hz sampling frequency
- camera used for position feedback
- explicit MPC coded using integer numbers

Automotive applications of explicit MPC

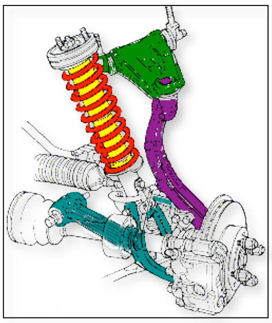
traction control

(Borrelli, Bemporad, Fodor, Hrovat, 2001)



engine control

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

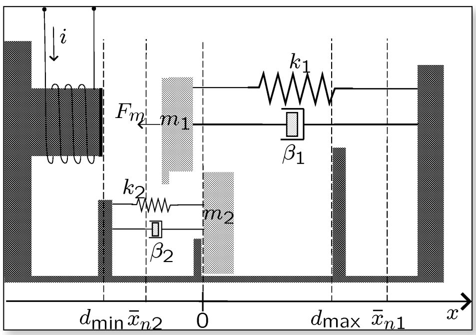


semiactive suspensions

(Giorgetti, Bemporad, Tseng, Hrovat, 2005)

idle speed control

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



magnetic actuators

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

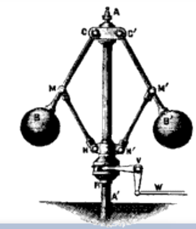
Explicit MPC for idle speed control

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

- Ford pickup truck, V8 4.6L gasoline engine
- Process:
  - *1 output* (engine speed) to regulate
  - *2 inputs* (airflow, spark advance)
  - input *delays*

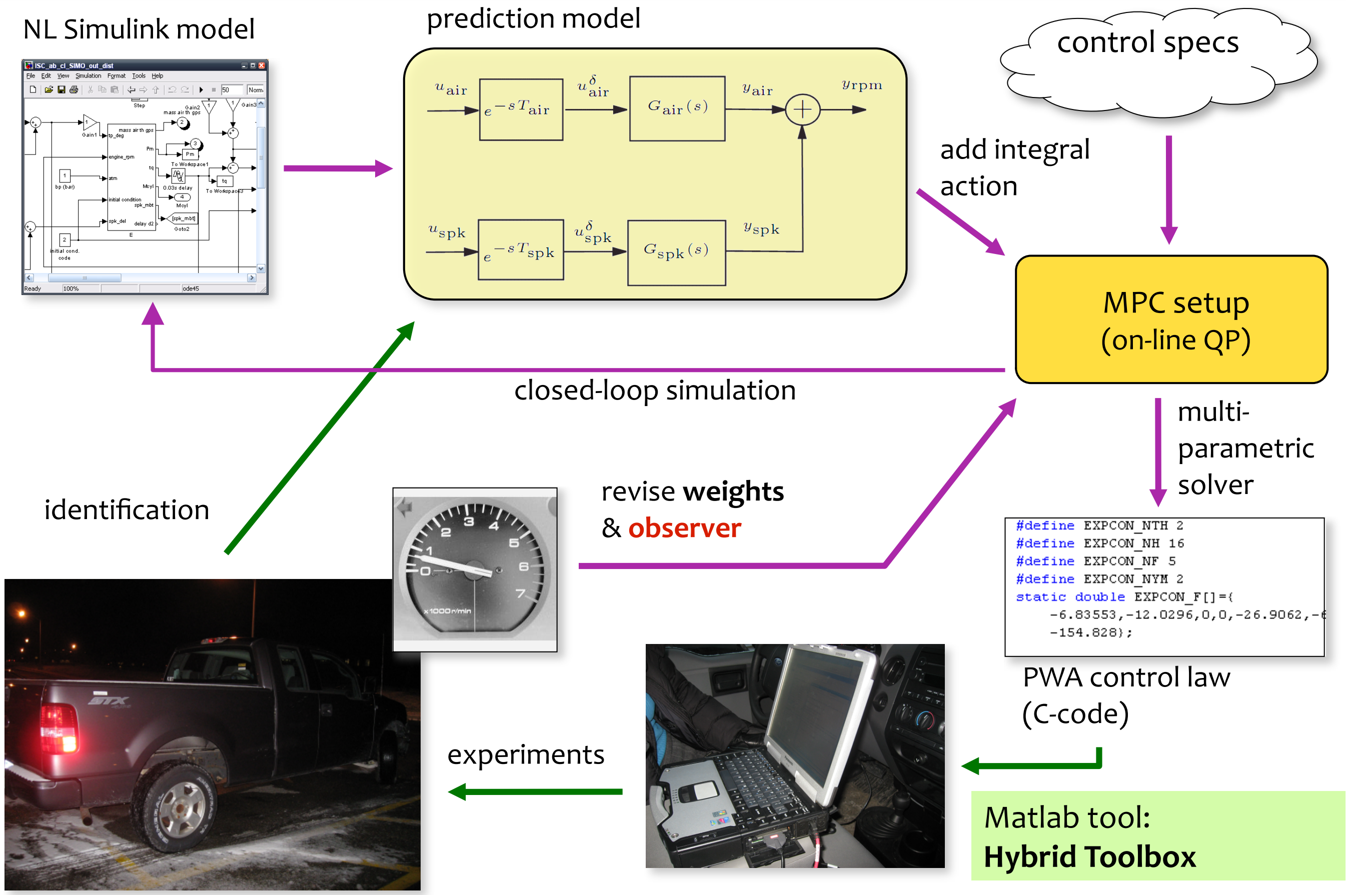
- Objectives and specs:
  - *regulate engine speed* at constant rpm
  - *saturation* limits on airflow and spark
  - *lower bound* on engine speed ($\geq$450 rpm)

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



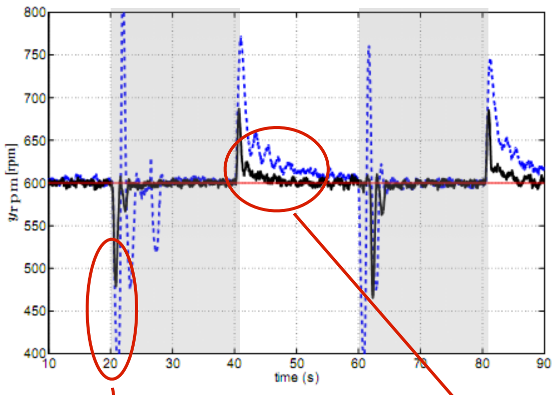
- Problem suitable for MPC design (Hrovat, 1996)

Explicit MPC Design Flow

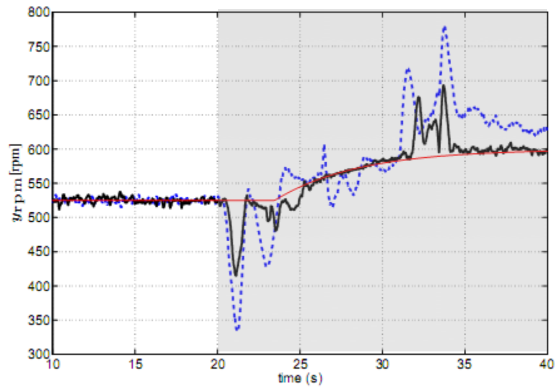


Explicit MPC for idle speed - Experiments

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



Load torque (power steering)



Power steering + air conditioning

peak reduced by 50%

convergence 10s faster

explicit MPC

baseline controller (linear)

set-point

Linear MPC Based on LP

Linear MPC Based on LP

(Bemporad, Borrelli, Morari, 2003)

- Linear Model:

$$\begin{cases} x(t+1) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \end{cases}$$

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

- Constraints:

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

- Optimal control problem (infinity-norm performance index):

$$\begin{aligned} \min_{u_t,\ldots,u_{t+N-1}} \quad & \sum_{k=0}^{N-1} \left(\|Qx_{t+k|t}\|_\infty + \|Ru_{t+k}\|_\infty\right) + \|Px_{t+N|t}\|_\infty \\ \text{s.t.} \quad & y_{min} \leq y_{t+k|t} \leq y_{max}, \quad k = 1,\ldots,N \\ & u_{min} \leq u_{t+k} \leq u_{max}, \quad k = 0,\ldots,N-1 \\ & x_{t|t} = x(t) \\ & x_{t+k+1|t} = Ax_{t+k|t} + Bu_{t+k}, \quad k = 0,\ldots,N-1 \\ & y_{t+k|t} = Cx_{t+k|t}, \quad k = 0,\ldots,N-1 \end{aligned}$$

Q, R full rank matrices

Linear MPC Based on LP

- Basic trick:

$$\begin{array}{l} \min |x| \\ x \in \mathbb{R} \end{array} \quad \Longleftrightarrow \quad \begin{array}{ll} \min & \epsilon \\ \text{s.t.} & \epsilon \geq x \\ & \epsilon \geq -x \end{array}$$

- Introduce slack vars:

$$\begin{array}{rcl} \epsilon_k^x & \geq & \|Qx_{t+k|t}\|_\infty \\ \epsilon_k^u & \geq & \|Ru_{t+k|t}\|_\infty \\ \epsilon_N^x & \geq & \|Px_{t+N|t}\|_\infty \end{array} \quad \Rightarrow \quad \begin{array}{rcll} \epsilon_k^x & \geq & Q^i x_{t+k|t} & i = 1,\ldots,n \\ \epsilon_k^x & \geq & -Q^i x_{t+k|t} & k = 1,\ldots,N \\ \epsilon_k^u & \geq & R^i u_{t+k|t} & i = 1,\ldots,m \\ \epsilon_k^u & \geq & -R^i u_{t+k|t} & k = 0,\ldots,N-1 \\ \epsilon_N^x & \geq & P^i x_{t+N|t} & i = 1,\ldots,n \\ \epsilon_N^x & \geq & -P^i x_{t+N|t} & \end{array}$$

$Q^i = i$th row of matrix Q

Linear MPC Based on LP

- Substitution: $x_{t+k|t} = A^k x(t) + \sum_{j=0}^{k-1} A^j B u_{t+k-1-j}$
- Optimization problem:

$$V(x(t)) = \min_z \begin{bmatrix} 1 & \dots & 1 & 0 & \dots & 0 \end{bmatrix} z \quad \text{(linear)}$$

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

LINEAR PROGRAM (LP)

- $z \triangleq [\epsilon_0^u \ \dots \ \epsilon_{N-1}^u \ \epsilon_1^x \ \dots \ \epsilon_N^x \ u'_t, \dots, u'_{t+N-1}]' \in \mathbb{R}^s$, $s \triangleq N(m+2)$, is the optimization vector
- G, W, S are obtained from weights Q, R, P, and model matrices A, B, C
- Q, R, P can be selected to guarantee closed-loop stability (Bemporad, Borrelli, Morari, 2003)

On-Line vs. Off-Line Optimization

$$V(x(t)) = \min_z \begin{bmatrix} 1 & \dots & 1 & 0 & \dots & 0 \end{bmatrix} z$$

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

$$z \triangleq [\epsilon_0^u \ \dots \ \epsilon_{N-1}^u \ \epsilon_1^x \ \dots \ \epsilon_N^x \ u'_t \ \dots \ u'_{t+N-1}]'$$

- **On-line** optimization: given $x(t)$ solve the problem at each time step t (the control law $u=u(x)$ is **implicitly** defined by the LP solver)

  → Linear Program (LP)

- **Off-line** optimization: solve the LP **for all** $x(t)$ to find the control law $u=u(x)$ **explicitly**

  → multi-parametric Linear Program (mp-LP)

Multiparametric LP

(Gal, Nedoma, 1972)
(Borrelli, Bemporad, Morari, 2003)

Primal Problem

$$\begin{array}{ll} \min_{\xi} & f'\xi \\ \text{s.t.} & G\xi \leq W + Sx \end{array}$$

Dual problem

$$\begin{array}{ll} \max_{\lambda} & (W + Sx)'\lambda \\ \text{s.t.} & G'\lambda = f \\ & \lambda \leq 0 \end{array}$$

Optimality conditions:

- Primal feasibility: $G\xi \leq W + Sx$
- Dual feasibility: $G'\lambda = f, \ \lambda \leq 0$
- Complementary slackness: $\lambda_j(G_j\xi - W_j - S_jx) = 0, \ \forall j$

For a given parameter x_0:

⇒ solve LP to find $\xi_0^*, \ \lambda_0^*$ (suppose no degeneracy)

⇒ identify active constraints

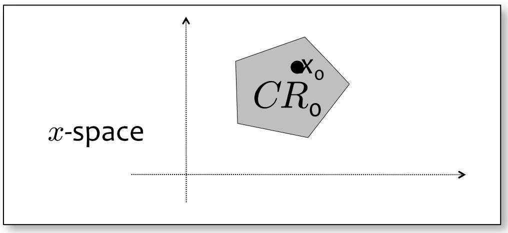
⇒ form submatrices $\tilde{G}, \tilde{W}, \tilde{S}$ of active constraints

Multiparametric LP

- Primal feasibility condition: $\tilde{G}\xi = \tilde{W} + \tilde{S}x$

⇒ $\xi^*(x) = (\tilde{G}^{-1}\tilde{S})x + (\tilde{G}^{-1}\tilde{W})$ optimizer

⇒ $\hat{G}(\tilde{G}^{-1}\tilde{S})x + \hat{G}(\tilde{G}^{-1}\tilde{W}) \leq \hat{W} + \hat{S}x$ critical region

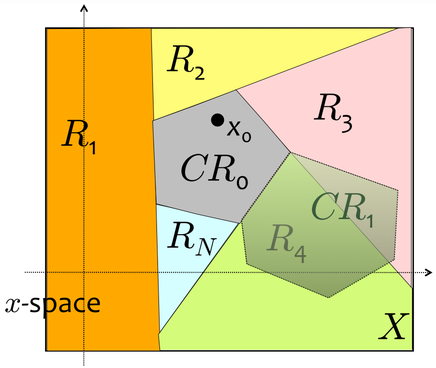


- Dual feasibility condition: $\tilde{\lambda}^*(x) = (\tilde{G}')^{-1}f = (\tilde{G}^{-1})'f, \ \hat{\lambda}^*(x) = 0$ dual variables

- Primal cost = dual cost:

⇒ $V^*(x) = f'\xi^*(x) = f'\tilde{G}^{-1}(\tilde{W} + \tilde{S}x)$ value function

Multiparametric LP



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

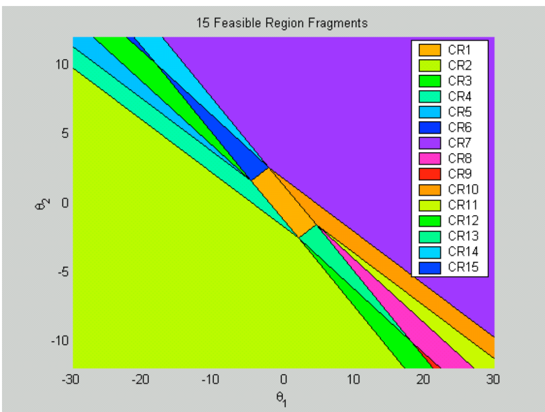
$$R_i = \{x \in X : \mathcal{A}^i x > \mathcal{B}^i, \ \mathcal{A}^j x \leq \mathcal{B}^j, \ \forall j < i\}$$

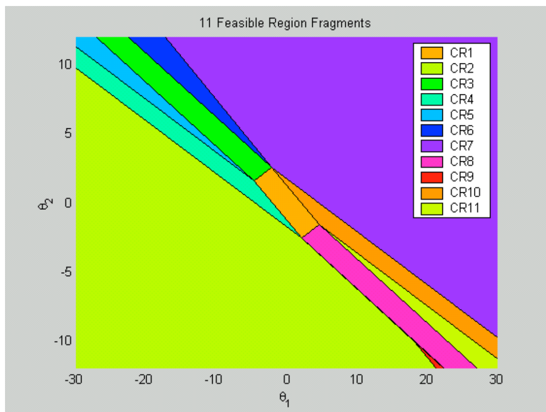
Note: while CR_i is characterizing a set of active constraints, R_i is not

1) Use the above splitting only as a search procedure, don't split the CR

2) Remove duplicates of CR already found

Union of Regions





$$z(x) \triangleq [\epsilon_0^u(x) \ \dots \ \epsilon_{N-1}^u(x) \ \epsilon_1^x(x) \ \dots \ \epsilon_N^x(x) \ u_0'(x) \ \dots \ u_{N-1}'(x)]'$$

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

(Bemporad, Fukuda, Torrisi, *Computational Geometry*, 2001)

Linearity, Convexity, and Continuity

Theorem The set X^* of parameters for which the mpLP problem is feasible is a convex polyhedron. The functions $V^*(x)$ and $\xi^*(x)$ are piecewise affine and continuous over X^*, and $V^*(x)$ is also convex on X^*.

$$\begin{aligned} \xi^*(x) &= \arg\min f'\xi \\ & \quad \text{s.t.} G\xi \leq W + Sx \end{aligned}$$

piecewise affine, continuous
(if optimizer is always unique)

$$\begin{aligned} V^*(x) &= \min f'\xi \\ & \quad \text{s.t.} G\xi \leq W + Sx \end{aligned}$$

continuous, piecewise affine, convex

Corollary 1: The value function $V^*(x)$ is continuous piecewise affine

Corollary 2: The MPC controller is continuous piecewise affine !

Triple Integrator Example

- System:

$$y(t) = \frac{1}{s^3}u(t)$$

sampling + ZOH
T=0.1 s

$$\begin{aligned} x(t+1) &= \begin{bmatrix} 1.05 & 0.46 & 1.02 \\ 0 & 0.73 & -0.01 \\ 0 & 0.86 & 0.99 \end{bmatrix} x(t) + \begin{bmatrix} 0.15 \\ 0.86 \\ 0.45 \end{bmatrix} u(t) \\ y(t) &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x(t) \end{aligned}$$

- Constraints: $-1 \leq u(t) \leq 1$

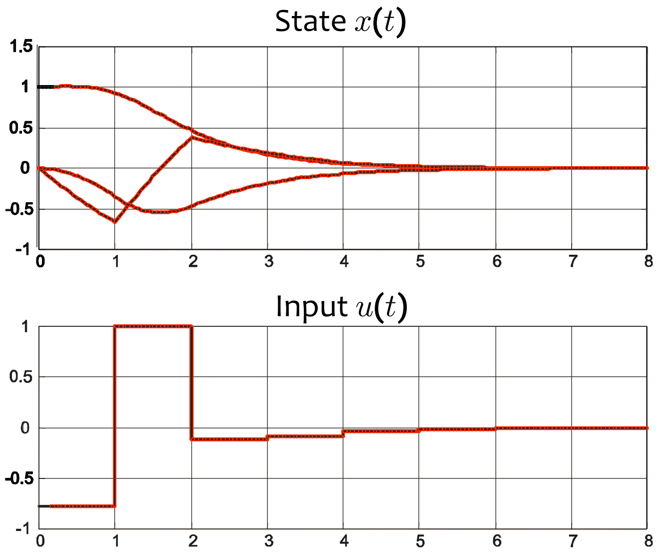
- Control objective: minimize: $\sum_{k=0}^{2} |y(k)| + \left|\frac{1}{8000}u(k)\right|$

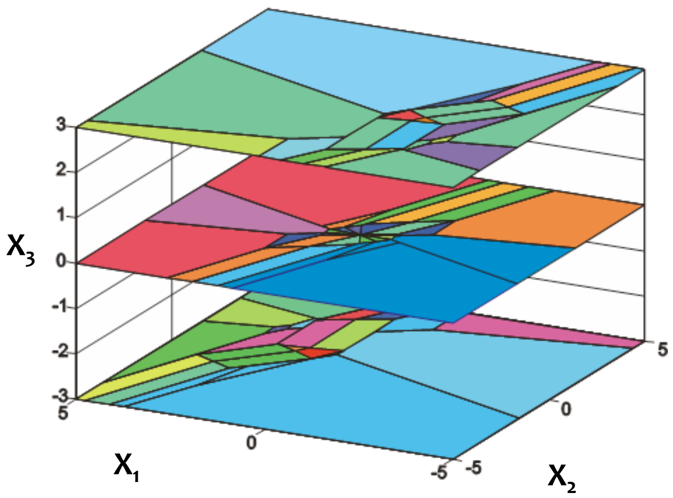
- LP problem:

$$f = [1\ 1\ 1\ 1\ 0\ 0],$$

$$G = \begin{bmatrix} -1 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 & -1332.1 & 0 \\ 0 & 0 & -1 & 0 & -7319.1 & 0 \\ 0 & 0 & -1 & 0 & -3848.8 & 0 \\ 0 & 0 & -1 & 0 & 1332.1 & 0 \\ 0 & 0 & -1 & 0 & 7319.1 & 0 \\ 0 & 0 & -1 & 0 & 3848.8 & 0 \\ 0 & 0 & 0 & -1 & -8706.3 & -1332.1 \\ 0 & 0 & 0 & -1 & -5295.8 & -7319.1 \\ 0 & 0 & 0 & -1 & -10116 & -3848.8 \\ 0 & 0 & 0 & -1 & 8706.3 & 1332.1 \\ 0 & 0 & 0 & -1 & 5295.8 & 7319.1 \\ 0 & 0 & 0 & -1 & 10116 & 3848.8 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}, \quad W = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad S = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 8935.8 & 3915.4 & 8689.4 \\ 0 & 6227.3 & -146.38 \\ 0 & 7319.1 & 8423 \\ -8935.8 & -3915.4 & -8689.4 \\ 0 & -6227.3 & 146.38 \\ 0 & -7319.1 & -8423 \\ 9394 & 14467 & 17678 \\ 0 & 4436.2 & -252.3 \\ 0 & 12615 & 8220.7 \\ -9394 & -14467 & -17678 \\ 0 & -4436.2 & 252.3 \\ 0 & -12615 & -8220.7 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Explicit MPC Controller





# constraints	# variables	# critical regions
20	6	68

Robust (Explicit) MPC

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

The origins of (multi)parametric programming

Cave painting, Lascaux, France, 15,000 to 10,000 B.C.

(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

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

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

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

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

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)

$$\begin{array}{rll} V^*(x) = & \min_z & h(z, x) \\ & \text{s.t.} & g(z, x) \leq 0 \end{array}$$

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 $(z, x) \Rightarrow V^*(x)$ 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 $(z, x) \Rightarrow V^*(x)$ 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)

Bibliography

[1] 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.

[2] P. Tøndel, T. A. Johansen, and A. Bemporad, "An algorithm for multi-parametric quadratic programming and explicit MPC solutions," Automatica, vol. 39, no. 3, pp. 489–497, Mar. 2003.

[3] P. Tøndel, T. A. Johansen, and A. Bemporad, "Evaluation of piecewise affine control via binary search tree," Automatica, vol. 39, no. 5, pp. 945–950, May 2003.

[4] A. Bemporad, F. Borrelli, and M. Morari, "Model predictive control based on linear programming — The explicit solution," IEEE Trans. Automatic Control, vol. 47, no. 12, pp. 1974–1985, 2002.

[5] F. Borrelli, A. Bemporad, and M. Morari, "A geometric algorithm for multi-parametric linear programming," Journal of Optimization Theory and Applications, vol. 118, no. 3, pp. 515–540, Sept. 2003.

[6] A. Alessio and A. Bemporad. A survey on explicit model predictive control. In D.M. Raimondo L. Magni, F. Allgower, editor, Nonlinear Model Predictive Control: Towards New Challenging Applications, volume 384 of Lecture Notes in Control and Information Sciences, pages 345–369, Berlin Heidelberg, 2009. Springer-Verlag.

[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.

[8] A. Bemporad, F. Borrelli, and M. Morari, "Min-max control of constrained uncertain discrete-time linear systems," IEEE Trans. Automatic Control, vol. 48, no. 9, pp. 1600–1606, 2003

[9] 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

Bibliography

Toolbox

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

Book

[11] F. Borrelli, A. Bemporad, and M. Morari, *Predictive control for linear and hybrid systems*, Cambridge University Press, 2011, In press.

Publications available for download at **http://www.ing.unitn.it/~bemporad/publications**