# Explicit Model Predictive Control

F. Borrelli, C. Jones, M. Morari

UC Berkeley, EPFL, ETHZ

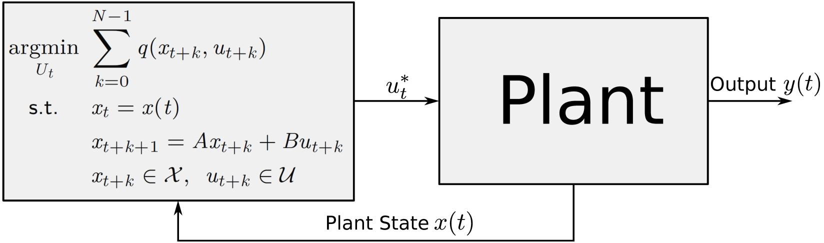
Fall Semester 2015

# Table of Contents

# Table of Contents

# Table of Contents

# Introduction



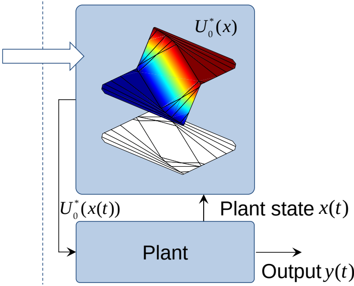

- Requires at each time step on-line solution of an optimization problem

# Introduction

OFFLINE

$$
\begin{aligned}
U_0^*(x(t)) = \text{argmin} \ \ & x_N^T P x_N + \sum_{k=0}^{N-1} x_k' Q x_k + u_k' R u_k \\
\text{subj. to} \ \ & x_0 = x(t) \\
& x_{k+1} = A x_k + B u_k, \ \ k = 0, \ldots, N-1 \\
& x_k \in \mathcal{X}, \ \ u_k \in \mathcal{U}, \ \ k = 0, \ldots, N-1 \\
& x_N \in \mathcal{X}_f
\end{aligned}
$$

ONLINE




- Optimization problem is parameterized by state
- Pre-compute control law as function of state $x$
- Control law is piecewise affine for linear system/constraints

Result: Online computation dramatically reduced and *real-time*
Tool: *Parametric programming*

# Table of Contents

# mpQP - Problem formulation

$$\begin{aligned} J^*(x) = \min_z \quad & \tfrac{1}{2} z' H z, \\ \text{subj. to} \quad & Gz \le w + Sx \end{aligned}$$

where $H > 0$, $z \in \mathbb{R}^s$, $x \in \mathbb{R}^n$ and $G \in \mathbb{R}^{m \times s}$.

Given a closed and bounded polyhedral set $\mathcal{K} \subset \mathbb{R}^n$ of parameters denote by $\mathcal{K}^* \subseteq \mathcal{K}$ the region of parameters $x \in \mathcal{K}$ such that the problem is feasible

$$\mathcal{K}^* \triangleq \{x \in K : \exists z,\ Gz \le w + Sx\}$$

Goals:

1. find $z^*(x) = \text{argmin}_z \ J(z, x)$,
2. find all $x$ for which the problem has a solution
3. compute the value function $J^*(x)$

# Active Set and Critical Region

Let $I \triangleq \{1, \ldots, m\}$ be the set of constraint indices.

## Definition: Active Set

We define the active set at $x$, $A(x)$, and its complement, $NA(x)$, as

$$A(x) \triangleq \{i \in I : G_i z^*(x) - S_i x = w_i\}$$
$$NA(x) \triangleq \{i \in I : \ G_i z^*(x) - S_i x < w_i\}.$$

$G_i$, $S_i$ and $w_i$ are the $i$-th row of $G$, $S$ and $w$, respectively.

## Definition: Critical Region

$CR_A$ is the set of parameters $x$ for which the same set $A \subseteq I$ of constraints is active at the optimum. For a given $\bar{x} \in \mathcal{K}^*$ let $(A, NA) \triangleq (A(\bar{x}), NA(\bar{x}))$. Then,

$$CR_A \triangleq \{x \in \mathcal{K}^* : A(x) = A\}.$$

# mpQP - Global properties of the solution

The following theorem summarizes the properties of the mpQP solution.

## Theorem: Solution of mpQP

i) The feasible set $\mathcal{K}^*$ is a **polyhedron**.

ii) The optimizer function $z^*(x) : \mathcal{K}^* \to \mathbb{R}^m$ is:
- **continuous**
- **polyhedral piecewise affine over** $\mathcal{K}^*$. It is affine in each critical region $\mathcal{CR}_i$, every $\mathcal{CR}_i$ is a polyhedron and $\bigcup \mathcal{CR}_i = \mathcal{K}^*$.

iii) The value function $J^*(x) : \mathcal{K}^* \to \mathbb{R}$ is:
- **continuous**
- **convex**
- **polyhedral piecewise quadratic over** $\mathcal{K}^*$, it is quadratic in each $\mathcal{CR}_i$

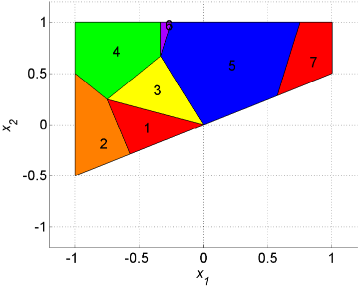
# mpQP - Example (1/4)

Consider the example

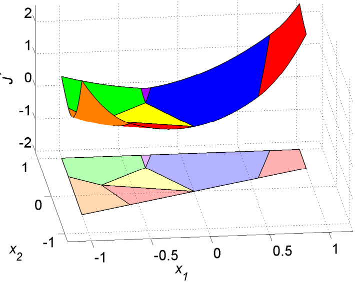
$$
\begin{aligned}
\min_{z(x)} \quad & \tfrac{1}{2}(z_1^2 + z_2^2) \\
\text{subj. to} \quad & z_1 \leq 1 + x_1 + x_2 \\
& -z_1 \leq 1 - x_1 - x_2 \\
& z_2 \leq 1 + x_1 - x_2 \\
& -z_2 \leq 1 - x_1 + x_2 \\
& z_1 - z_2 \leq x_1 + 3x_2 \\
& -z_1 + z_2 \leq -2x_1 - x_2 \\
& -1 \leq x_1 \leq 1,\ -1 \leq x_2 \leq 1
\end{aligned}
$$

# mpQP - Example (2/4)

The explicit solution is defined over $i = 1, \ldots, 7$ regions $\mathcal{P}_i = \{x \in \mathbb{R}^2 \mid A_i x \le b_i\}$ in the parameter space $x_1 - x_2$.




Critical regions




Piecewise quadratic objective function $J^*(x)$

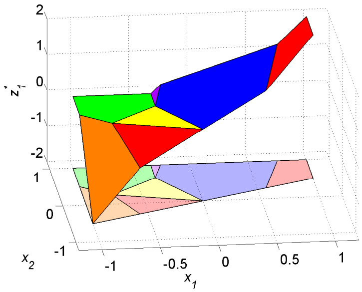
# mpQP - Example (3/4)

Primal solution is given as piecewise affine function $z(x) = F_i + g_i x$ if $x \in \mathcal{P}_i$.

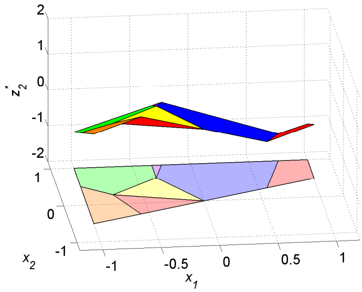
$$z^*(x) = \begin{cases} \begin{pmatrix} 0.5 & 1.5 \\ -0.5 & -1.5 \end{pmatrix} x \text{ if } x \in \mathcal{P}_1 \\ \begin{pmatrix} 2 & 2 \\ 1 & -1 \end{pmatrix} x + \begin{pmatrix} 1 \\ 1 \end{pmatrix} \text{ if } x \in \mathcal{P}_2 \\ \vdots \\ \vdots \end{cases}$$

# mpQP - Example (4/4)

Primal solution is given as piecewise affine function $z(x) = F_i + g_i x$ if $x \in \mathcal{P}_i$.




Piecewise affine function $z_1^*(x)$




Piecewise affine function $z_2^*(x)$

# Table of Contents

# mpLP - Problem formulation

$$
\begin{aligned}
J^*(x) = \min_{z} \quad & c'z \\
\text{subj. to} \quad & Gz \le w + Sx
\end{aligned}
$$

where $z \in \mathbb{R}^s$, $x \in \mathbb{R}^n$ and $G \in \mathbb{R}^{m \times s}$.

Given a closed and bounded polyhedral set $\mathcal{K} \subset \mathbb{R}^n$ of parameters, denote by $\mathcal{K}^* \subseteq \mathcal{K}$ the region of parameters $x \in \mathcal{K}$ such that the problem is feasible

$$\mathcal{K}^* \triangleq \{x \in K : \exists z, \ Gz \le w + Sx\}$$

Goals:

1. find $z^*(x) = \text{argmin}_z \ J(z, x)$,
2. find all $x$ for which the problem has a solution
3. compute the value function $J^*(x)$

# mpLP - Global properties of the solution

The following theorem summarizes the properties of the mpLP solution.

## Theorem: Solution of mpLP

i) The feasible set $\mathcal{K}^*$ is a **polyhedron**.

ii) If the optimal solution $z^*$ is unique $\forall x \in \mathcal{K}^*$, the optimizer function $z^*(x) : \mathcal{K}^* \to \mathbb{R}^m$ is:

- **continuous**
- **polyhedral piecewise affine over** $\mathcal{K}^*$. It is affine in each critical region $\mathcal{CR}_i$, every $\mathcal{CR}_i$ is a polyhedron and $\bigcup \mathcal{CR}_i = \mathcal{K}^*$.

Otherwise, it is always possible to choose such a continuous and PPWA optimizer function $z^*(x)$.

iii) The value function $J^*(x)\ :\ \mathcal{K}^* \to \mathbb{R}$ is:

- **continuous**
- **convex**
- **polyhedral piecewise affine over** $\mathcal{K}^*$, it is affine in each $\mathcal{CR}_i$.

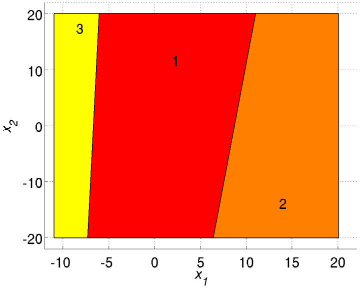
# mpLP - Example (1/4)

Consider the example

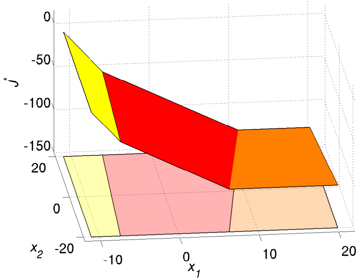
$$
\begin{aligned}
\min_{z(x)} \quad & -3z_1 - 8z_2 \\
\text{subj. to} \quad & z_1 + z_2 \le 13 + x_1 \\
& 5z_1 - 4z_2 \le 20 \\
& -8z_1 + 22z_2 \le 121 + x_2 \\
& -4z_1 - z_2 \le -8 \\
& -z_1 \le 0 \\
& -z_2 \le 0 \\
& -1 \le x_1 \le 1 \\
& -1 \le x_2 \le 1
\end{aligned}
$$

# mpLP - Example (2/4)

The explicit solution is defined over $i = 1, \ldots, 3$ regions $\mathcal{P}_i = \{x \in \mathbb{R}^2 \mid A_i x \leq b_i\}$ in the parameter space $x_1 - x_2$.




Critical regions $\mathcal{P}_i$




Piecewise affine objective function $J^*(x)$

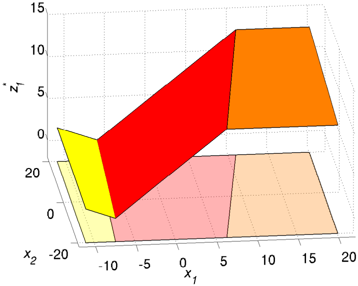
# mpLP - Example (3/4)

Primal solution is given as piecewise affine function $z^*(x) = F_i x + g_i$ if $x \in \mathcal{P}_i$.

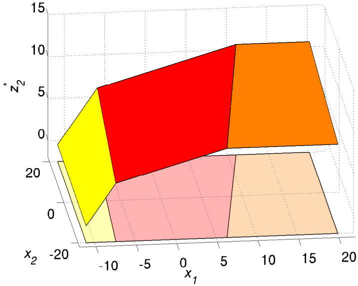
$$z^*(x) = \begin{cases} \begin{pmatrix} 0.733 & -0.033 \\ 0.267 & 0.033 \end{pmatrix} x + \begin{pmatrix} 5.5 \\ 7.5 \end{pmatrix} & \text{if } x \in \mathcal{P}_1 \\ \begin{pmatrix} 0 & 0.051 \\ 0 & 0.064 \end{pmatrix} x + \begin{pmatrix} 11.846 \\ 9.808 \end{pmatrix} & \text{if } x \in \mathcal{P}_2 \\ \begin{pmatrix} -0.333 & 0 \\ 1.333 & 0 \end{pmatrix} x + \begin{pmatrix} -1.667 \\ 14.667 \end{pmatrix} & \text{if } x \in \mathcal{P}_3 \end{cases}$$

# mpLP - Example (4/4)

Primal solution is given as piecewise affine function $z^*(x) = F_i x + g_i$ if $x \in \mathcal{P}_i$.




Piecewise affine function $z_1^*(x)$




Piecewise affine function $z_2^*(x)$

# Table of Contents

# Problem Formulation - Quadratic Cost

Quadratic cost function

$$J_0(x(0), U_0) = x_N' P x_N + \sum_{k=0}^{N-1} x_k' Q x_k + u_k' R u_k \tag{1}$$

with $P \succeq 0$, $Q \succeq 0$, $R \succ 0$.
Constrained Finite Time Optimal Control problem (CFTOC).

$$\boxed{\begin{array}{lll} J_0^*(x(0)) = & \min\limits_{U_0} & J_0(x(0), U_0) \\ & \text{subj. to} & x_{k+1} = Ax_k + Bu_k,\ k = 0, \ldots, N-1 \\ & & x_k \in \mathcal{X},\ u_k \in \mathcal{U},\ k = 0, \ldots, N-1 \\ & & x_N \in \mathcal{X}_f \\ & & x_0 = x(0) \end{array}} \tag{2}$$

$N$ is the time horizon and $\mathcal{X}$, $\mathcal{U}$, $\mathcal{X}_f$ are polyhedral regions.

# Construction of the QP with substitution

- **Step 1**: Rewrite the cost as (see lectures on Day 1 & 2)

$$
\begin{aligned}
J_0(x(0), U_0) &= U_0' H U_0 + 2x(0)' F U_0 + x(0)' Y x(0) \\
&= [U_0' \; x(0)'] \left[\begin{smallmatrix} H & F' \\ F & Y \end{smallmatrix}\right] [U_0{}' \; x(0)']'
\end{aligned}
$$

  Note: $\left[\begin{smallmatrix} H & F' \\ F & Y \end{smallmatrix}\right] \succeq 0$ since $J_0(x(0), U_0) \geq 0$ by assumption.

- **Step 2**: Rewrite the constraints compactly as (details provided on the next slide)

$$
G_0 U_0 \leq w_0 + E_0 x(0)
$$

- **Step 3**: Rewrite the optimal control problem as

$$
\boxed{
\begin{aligned}
J_0^*(x(0)) = \min_{U_0} \quad & [U_0' \; x(0)'] \left[\begin{smallmatrix} H & F' \\ F & Y \end{smallmatrix}\right] [U_0{}' \; x(0)']' \\
\text{subj. to} \quad & G_0 U_0 \leq w_0 + E_0 x(0)
\end{aligned}
}
$$

# Solution

$$
\begin{aligned}
J_0^*(x(0)) = \min_{U_0} & \quad [U_0' \; x(0)'] \left[\begin{smallmatrix} H & F' \\ F & Y \end{smallmatrix}\right] [U_0{}' \; x(0)']' \\
\text{subj. to} & \quad G_0 U_0 \leq w_0 + E_0 x(0)
\end{aligned}
$$

For a given x(0) $U_0^*$ can be found via a QP solver.

# Construction of QP constraints with substitution

If $\mathcal{X}$, $\mathcal{U}$ and $\mathcal{X}_f$ are given by:

$$\mathcal{X} = \{x \,|\, A_x x \leq b_x\} \qquad \mathcal{U} = \{u \,|\, A_u u \leq b_u\} \qquad \mathcal{X}_f = \{x \,|\, A_f x \leq b_f\}$$

Then $G_0$, $E_0$ and $w_0$ are defined as follows

$$G_0 = \begin{bmatrix} A_u & 0 & \dots & 0 \\ 0 & A_u & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A_u \\ 0 & 0 & \dots & 0 \\ A_x B & 0 & \dots & 0 \\ A_x A B & A_x B & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_f A^{N-1} B & A_f A^{N-2} B & \dots & A_f B \end{bmatrix}, E_0 = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ -A_x \\ -A_x A \\ -A_x A^2 \\ \vdots \\ -A_f A^N \end{bmatrix}, w_0 = \begin{bmatrix} b_u \\ b_u \\ \vdots \\ b_u \\ b_x \\ b_x \\ b_x \\ \vdots \\ b_f \end{bmatrix}$$

# 2-Norm State Feedback Solution

Start from QP with substitution.

- **Step 1**: Define $z \triangleq U_0 + H^{-1}F'x(0)$ and transform the problem into

$$\boxed{\begin{array}{rll} \hat{J}^*(x(0)) = & \min_z & z'Hz \\ & \text{subj. to} & G_0 z \leq w_0 + S_0 x(0), \end{array}}$$

  where $S_0 \triangleq E_0 + G_0 H^{-1} F'$, and
  $\hat{J}^*(x(0)) = J_0^*(x(0)) - x(0)'(Y - FH^{-1}F')x(0)$.

  The CFTOC problem is now a **multiparametric quadratic program (mp-QP)**.

- **Step 2**: Solve the mp-QP to get explicit solution $z^*(x(0))$

- **Step 3**: Obtain $U_0^*(x(0))$ from $z^*(x(0))$

# 2-Norm State Feedback Solution

## Main Results

1. The **Open loop optimal control function** can be obtained by solving the mp-QP problem and calculating $U_0^*(x(0)), \ \forall x(0) \in \mathcal{X}_0$ as $U_0{}^* = z^*(x(0)) - H^{-1}F'x(0)$.
2. The first component of the multiparametric solution has the form

$$u^*(0) = f_0(x(0)), \ \ \forall x(0) \in \mathcal{X}_0,$$

$f_0 : \mathbb{R}^n \to \mathbb{R}^m$, is continuous and piecewise affine on polyhedra

$$f_0(x) = F_0^i x + g_0^i \quad \text{if} \quad x \in CR_0^i, \ \ i = 1, \dots, N_0^r$$

3. The polyhedral sets $CR_0^i = \{x \in \mathbb{R}^n | H_0^i x \leq K_0^i\}$, $i = 1, \dots, N_0^r$ are a partition of the feasible polyhedron $\mathcal{X}_0$.
4. The value function $J_0^*(x(0))$ is convex and piecewise quadratic on polyhedra.

# Example

Consider the double integrator

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

subject to constraints

$$-1 \le u(k) \le 1, \;\; k = 0, \ldots, 5$$

$$\begin{bmatrix} -10 \\ -10 \end{bmatrix} \le x(k) \le \begin{bmatrix} 10 \\ 10 \end{bmatrix}, \;\; k = 0, \ldots, 5$$

Compute the **state feedback** optimal controller $u^*(0)(x(0))$ solving the CFTOC

problem with $N = 6$, $Q = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$, $R = 0.1$, $P$ the solution of the ARE, $\mathcal{X}_f = \mathbb{R}^2$.

# Example

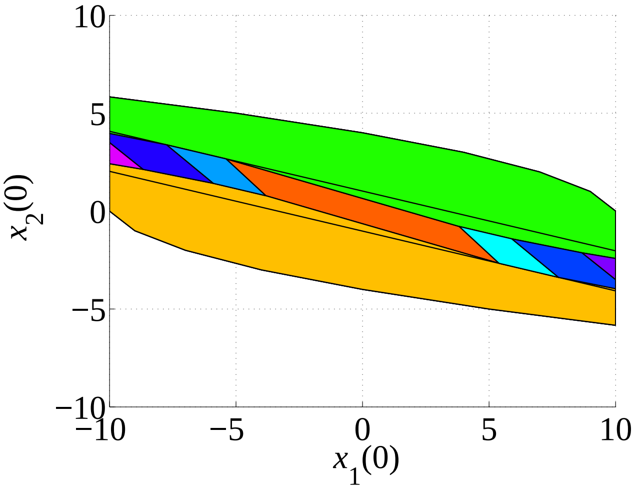



Figure: Partition of the state space for the piecewise affine control law $u^*(0)$ ($N_0^r = 13$)

# Problem Formulation - Piecewise Linear Cost

Piecewise linear cost function

$$J_0(x(0), U_0) := \|Px_N\|_p + \sum_{k=0}^{N-1} \|Qx_k\|_p + \|Ru_k\|_p \tag{3}$$

with $p = 1$ or $p = \infty$, $P$, $Q$, $R$ full column rank matrices

Constrained Finite Time Optimal Control Problem (CFTOC)

$$\begin{array}{rll} J_0^*(x(0)) = & \min\limits_{U_0} & J_0(x(0), U_0) \\ & \text{subj. to} & x_{k+1} = Ax_k + Bu_k,\ k = 0, \ldots, N-1 \\ & & x_k \in \mathcal{X},\ u_k \in \mathcal{U},\ k = 0, \ldots, N-1 \\ & & x_N \in \mathcal{X}_f \\ & & x_0 = x(0) \end{array} \tag{4}$$

$N$ is the time horizon and $\mathcal{X}$, $\mathcal{U}$, $\mathcal{X}_f$ are polyhedral regions.

# Construction of the LP with substitution

Recall that the $\infty-$norm problem can be equivalently formulated as

$$
\begin{aligned}
\min_{z_0} \quad & \varepsilon_0^x + \ldots + \varepsilon_N^x + \varepsilon_0^u + \ldots + \varepsilon_{N-1}^u \\
\text{subj. to} \quad & -\mathbf{1}_n \varepsilon_k^x \leq \pm Q \left[ A^k x_0 + \sum_{j=0}^{k-1} A^j B u_{k-1-j} \right], \\
& -\mathbf{1}_r \varepsilon_N^x \leq \pm P \left[ A^N x_0 + \sum_{j=0}^{N-1} A^j B u_{N-1-j} \right], \\
& -\mathbf{1}_m \varepsilon_k^u \leq \pm R u_k, \\
& A^k x_0 + \sum_{j=0}^{k-1} A^j B u_{k-1-j} \in \mathcal{X}, \;\; u_k \in \mathcal{U}, \\
& A^N x_0 + \sum_{j=0}^{N-1} A^j B u_{N-1-j} \in \mathcal{X}_f, \\
& k = 0, \ldots, N-1 \\
& x_0 = x(0)
\end{aligned}
$$

# Construction of the LP with substitution

The problem yields the following standard LP

$$
\begin{array}{ll}
\min\limits_{z_0} & c_0' z_0 \\
\text{subj. to} & \bar{G}_0 z_0 \leq \bar{w}_0 + \bar{S}_0 x(0)
\end{array}
$$

where $z_0 := \{\varepsilon_0^x, \dots, \varepsilon_N^x, \varepsilon_0^u, \dots, \varepsilon_{N-1}^u, u_0', \dots, u_{N-1}'\} \in \mathbb{R}^s$,
$s \triangleq (m+1)N + N + 1$ and

$$
\bar{G}_0 = \begin{bmatrix} G_\varepsilon & 0 \\ 0 & G_0 \end{bmatrix}, \; \bar{S}_0 = \begin{bmatrix} S_\varepsilon \\ S_0 \end{bmatrix}, \; \bar{w}_0 = \begin{bmatrix} w_\varepsilon \\ w_0 \end{bmatrix}
$$

For a given $x(0)$ $U_0^*$ can be obtained via an LP solver (the $1-$norm case is similar).

# 1- /∞-Norm State Feedback Solution

## Main Results

1. The **Open loop optimal control function** can be obtained by solving the mp-LP problem and calculating $z_0^*(x(0))$
2. The component $u_0^* = [0 \ \ldots 0 \ I_m \ 0 \ \ldots \ 0] z_0^*(x(0))$ of the multiparametric solution has the form

$$u^*(0) = f_0(x(0)), \quad \forall x(0) \in \mathcal{X}_0,$$

$f_0 : \mathbb{R}^n \to \mathbb{R}^m$, is continuous and piecewise affine on polyhedra

$$f_0(x) = F_0^i x + g_0^i \quad \text{if} \quad x \in CR_0^i, \quad i = 1, \ldots, N_0^r$$

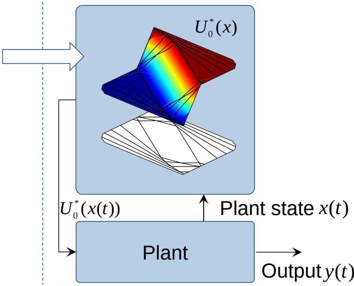
3. The polyhedral sets $CR_0^i = \{x \in \mathbb{R}^n | H_0^i x \leq K_0^i\}$, $i = 1, \ldots, N_0^r$ are a partition of the feasible polyhedron $\mathcal{X}_0$.
4. In case of multiple optimizers a piecewise affine control law exists.
5. The value function $J_0^*(x(0))$ is convex and piecewise affine on polyhedra.

# Explicit MPC

OFFLINE

$$
\begin{aligned}
U_0^*(x(t)) = \text{argmin} \quad & x_N^T P x_N + \sum_{k=0}^{N-1} x_k' Q x_k + u_k' R u_k \\
\text{subj. to} \quad & x_0 = x(t) \\
& x_{k+1} = A x_k + B u_k,\ k = 0, \ldots, N-1 \\
& x_k \in \mathcal{X},\ u_k \in \mathcal{U},\ k = 0, \ldots, N-1 \\
& x_N \in \mathcal{X}_f
\end{aligned}
$$

ONLINE




- Optimization problem is parameterized by state
- Pre-compute control law as function of state $x$
- Control law is piecewise affine for linear system/constraints

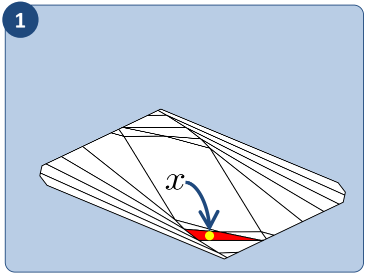
Result: Online computation dramatically reduced and *real-time*
Tool: *Parametric programming*

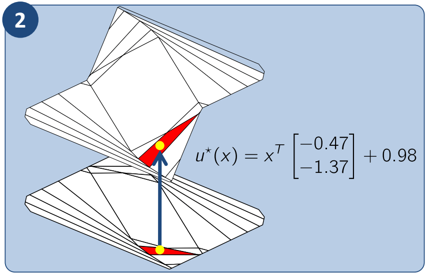
# Table of Contents

# Online evaluation: Point location

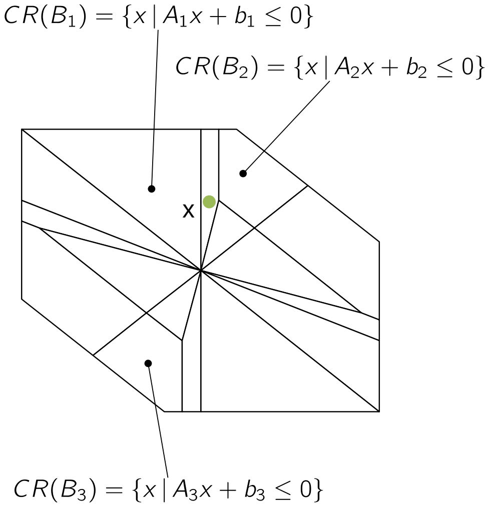
Calculation of piecewise affine function:

1. Point location
2. Evaluation of affine function



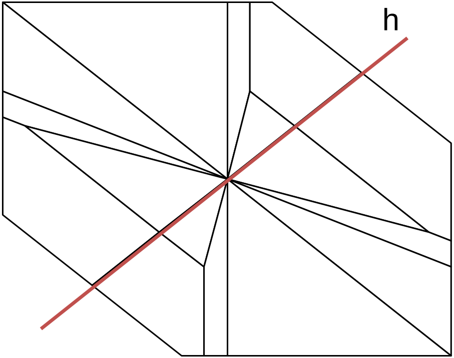
# Sequential search




Sequential search
for each $i$
  if $A_i x + b_i \leq 0$ then
  $x$ is in region $i$

- Very simple
- Linear in number of regions

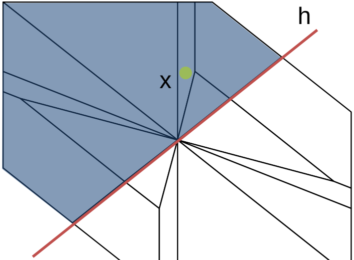
# Logarithmic search (1/6)

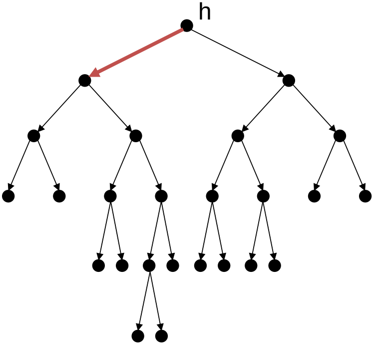



Offline construction of search tree

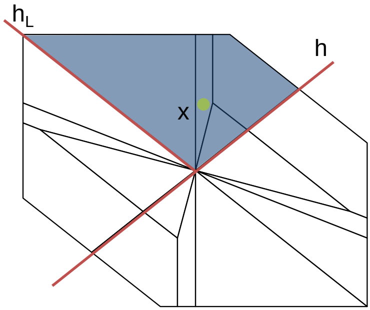
- Find hyperplane that separates regions into two equal sized sets
- Repeat for left and right sets

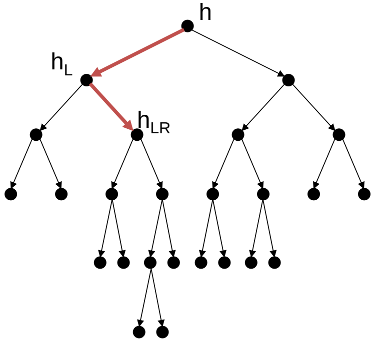
# Logarithmic search (2/6)



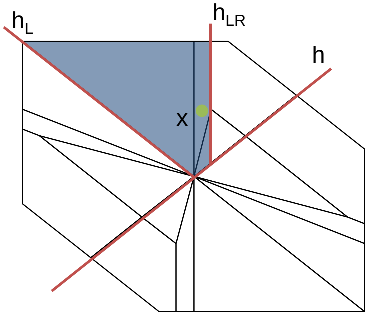



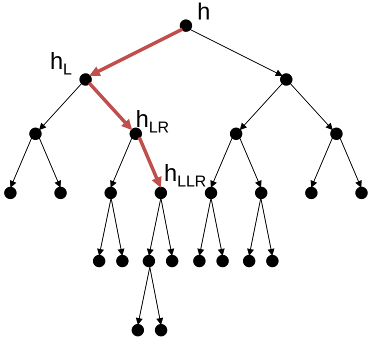
# Logarithmic search (3/6)



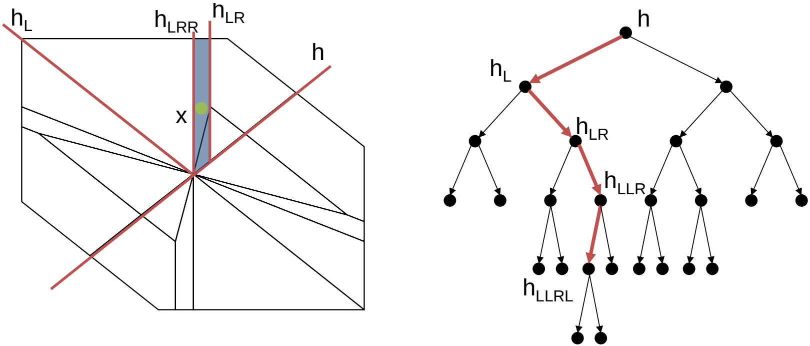



# Logarithmic search (4/6)

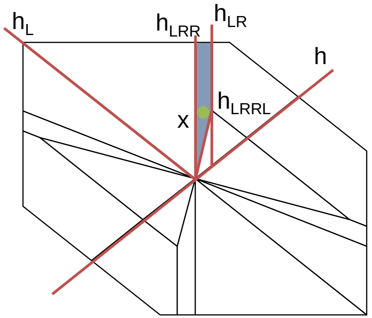


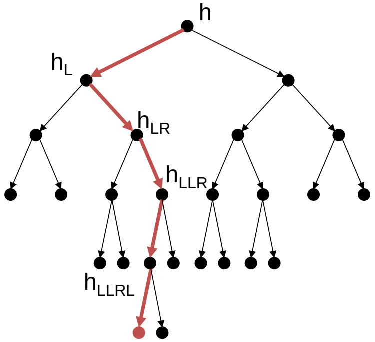



# Logarithmic search (5/6)



# Logarithmic search (6/6)



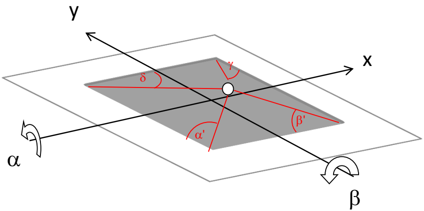



# Point Location - Summary

- Sequential search
  - Very simple
  - Works for all problems
- Search tree
  - Potentially logarithmic
  - Significant offline processing (reasonable for $< 1'000$ regions)
- Many other options for special cases

# Table of Contents

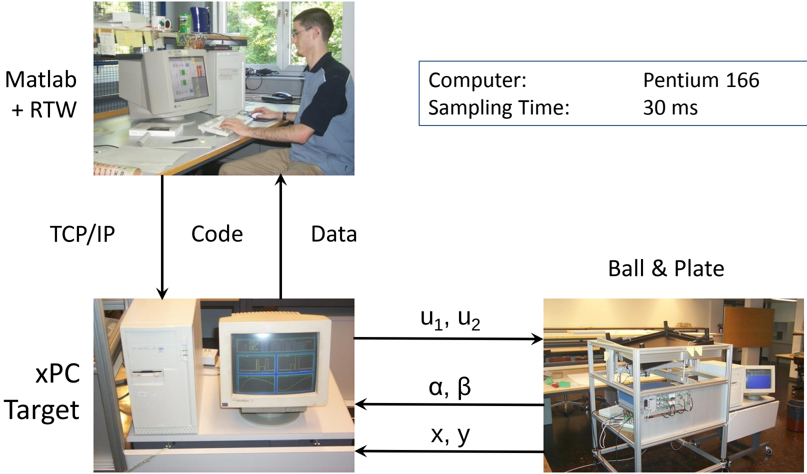
# Ball and Plate




- Linearized model: four states for each axis: plate angle, ball position, plate angular speed, ball speed.
- Constraints on inputs and states
  - Plate angle
  - Ball position
  - Acceleration
- MPC objective: path tracking

# Ball and Plate - System

| | |
|---|---|
| Computer: | Pentium 166 |
| Sampling Time: | 30 ms |

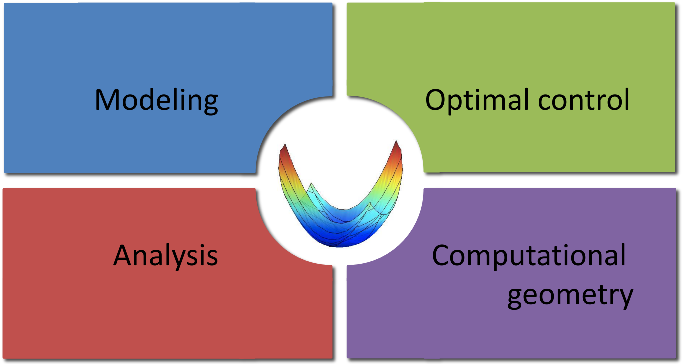


# Ball and Plate - MPC Problem

- $4$ states $+$ $1$ tracking variable $=$ 5 parameters
- Move-blocking reduces complexity
  - Horizon of $10$
  - Inputs $2-10$ must be equal

$$
\boxed{
\begin{aligned}
\min \quad & \sum_{i=0}^{9} 100\|y_i - y_t\|_2^2 + 0.1\|u_i\|_2^2 \\
s.t. \quad & x_0 = x \\
& x_{i+1} = Ax_i + Bu_i \\
& y_i = Cx_i \\
& u_{\text{min}} \leq u_i \leq u_{\text{max}} \\
& y_{\text{min}} \leq y_i \leq y_{\text{max}} \\
& x_{\text{min}} \leq x_i \leq x_{\text{max}} \\
& u_{i+1} = u_i, i = \{1, \ldots, 9\}
\end{aligned}
}
$$

# Multi-Parametric Toolbox




control.ee.ethz.ch/~mpt

# Multi-Parametric Toolbox 3.0 Formulation

MPT

```
% Linear discrete-time prediction model
model=LTISystem('A', A, 'B', B, 'C', C);

% Input constraints
model.u.min = -10; model.u.max = 10;

% Output constraints
model.y.min = -30; model.y.max = 30;

% State constraints
model.x.min = [-30; -15; -15*pi/180; -1];
model.x.max = [30; 15; 15*pi/180; 1];

% Penalties in the cost function
model.y.penalty = QuadFunction(100);
model.u.penalty = QuadFunction(0.1);

% Adjustment via input blocking
model.u.with('block'); model.u.from = 1; model.u.to = 9;

% Time varying reference signal
model.y.with('reference'); model.y.reference = 'free';
```

```
% Online MPC object
online_ctrl = MPCController( model, 9 )
```

$$
\begin{aligned}
\min \quad & \sum_{i=0}^{9} 100\|y_i - y_t\|_2^2 + 0.1\|u_i\|_2^2 \\
\text{s.t.} \quad & x_0 = x \\
& x_{i+1} = Ax_i + Bu_i \\
& y_i = Cx_i \\
& u_{\min} \leq u_i \leq u_{\max} \\
& y_{\min} \leq y_i \leq y_{\max} \\
& x_{\min} \leq x_i \leq x_{\max} \\
& u_{i+1} = u_i, \quad i = \{1, \ldots, 9\}
\end{aligned}
$$

# Multi-Parametric Toolbox 3.0 Formulation

MPT

```
% Linear discrete-time prediction model
model=LTISystem('A', A, 'B', B, 'C', C);

% Input constraints
model.u.min = -10; model.u.max = 10;

% Output constraints
model.y.min = -30; model.y.max = 30;

% State constraints
model.x.min = [-30; -15; -15*pi/180; -1];
model.x.max = [30; 15; 15*pi/180; 1];

% Penalties in the cost function
model.y.penalty = QuadFunction(100);
model.u.penalty = QuadFunction(0.1);

% Adjustment via input blocking
model.u.with('block'); model.u.from = 1; model.u.to = 9;

% Time varying reference signal
model.y.with('reference'); model.y.reference = 'free';
```

```
% Online MPC object
online_ctrl = MPCController( model, 9 )
```

$$
\begin{aligned}
\min \ & \sum_{i=0}^{9} 100\|y_i - y_t\|_2^2 + 0.1\|u_i\|_2^2 \\
\text{s.t. } & x_0 = x \\
& x_{i+1} = Ax_i + Bu_i \\
& y_i = Cx_i \\
& u_{\min} \le u_i \le u_{\max} \\
& y_{\min} \le y_i \le y_{\max} \\
& x_{\min} \le x_i \le x_{\max} \\
& u_{i+1} = u_i, \quad i = \{1, \dots, 9\}
\end{aligned}
$$

# Multi-Parametric Toolbox 3.0 Formulation

MPT

```
% Linear discrete-time prediction model
model=LTISystem('A', A, 'B', B, 'C', C);

% Input constraints
model.u.min = -10; model.u.max = 10;

% Output constraints
model.y.min = -30; model.y.max = 30;

% State constraints
model.x.min = [-30; -15; -15*pi/180; -1];
model.x.max = [30; 15; 15*pi/180; 1];

% Penalties in the cost function
model.y.penalty = QuadFunction(100);
model.u.penalty = QuadFunction(0.1);

% Adjustment via input blocking
model.u.with('block'); model.u.from = 1; model.u.to = 9;

% Time varying reference signal
model.y.with('reference'); model.y.reference = 'free';
```

```
% Online MPC object
online_ctrl = MPCController( model, 9 )
```

$$
\begin{aligned}
\min \ & \sum_{i=0}^{9} 100\|y_i - y_t\|_2^2 + 0.1\|u_i\|_2^2 \\
\text{s.t. } & x_0 = x \\
& x_{i+1} = Ax_i + Bu_i \\
& y_i = Cx_i \\
& u_{\min} \leq u_i \leq u_{\max} \\
& y_{\min} \leq y_i \leq y_{\max} \\
& x_{\min} \leq x_i \leq x_{\max} \\
& u_{i+1} = u_i, \quad i = \{1, \ldots, 9\}
\end{aligned}
$$

# Multi-Parametric Toolbox 3.0 Formulation

MPT

```
% Linear discrete-time prediction model
model=LTISystem('A', A, 'B', B, 'C', C);

% Input constraints
model.u.min = -10; model.u.max = 10;

% Output constraints
model.y.min = -30; model.y.max = 30;

% State constraints
model.x.min = [-30; -15; -15*pi/180; -1];
model.x.max = [30; 15; 15*pi/180; 1];

% Penalties in the cost function
model.y.penalty = QuadFunction(100);
model.u.penalty = QuadFunction(0.1);

% Adjustment via input blocking
model.u.with('block'); model.u.from = 1; model.u.to = 9;

% Time varying reference signal
model.y.with('reference'); model.y.reference = 'free';
```

```
% Online MPC object
online_ctrl = MPCController( model, 9 )
```

$$
\begin{aligned}
\min \quad & \sum_{i=0}^{9} 100\|y_i - y_t\|_2^2 + 0.1\|u_i\|_2^2 \\
\text{s.t.} \quad & x_0 = x \\
& x_{i+1} = Ax_i + Bu_i \\
& y_i = Cx_i \\
& u_{\min} \leq u_i \leq u_{\max} \\
& y_{\min} \leq y_i \leq y_{\max} \\
& x_{\min} \leq x_i \leq x_{\max} \\
& u_{i+1} = u_i, \quad i = \{1, \dots, 9\}
\end{aligned}
$$

# Multi-Parametric Toolbox 3.0 Formulation

MPT

```
% Linear discrete-time prediction model
model=LTISystem('A', A, 'B', B, 'C', C);

% Input constraints
model.u.min = -10; model.u.max = 10;

% Output constraints
model.y.min = -30; model.y.max = 30;

% State constraints
model.x.min = [-30; -15; -15*pi/180; -1];
model.x.max = [30; 15; 15*pi/180; 1];

% Penalties in the cost function
model.y.penalty = QuadFunction(100);
model.u.penalty = QuadFunction(0.1);

% Adjustment via input blocking
model.u.with('block'); model.u.from = 1; model.u.to = 9;

% Time varying reference signal
model.y.with('reference'); model.y.reference = 'free';
```

```
% Online MPC object
online_ctrl = MPCController( model, 9 )
```

$$\begin{aligned} \min \ & \sum_{i=0}^{9} 100\|y_i - y_t\|_2^2 + 0.1\|u_i\|_2^2 \\ \text{s.t.} \ & x_0 = x \\ & x_{i+1} = Ax_i + Bu_i \\ & y_i = Cx_i \\ & u_{\min} \leq u_i \leq u_{\max} \\ & y_{\min} \leq y_i \leq y_{\max} \\ & x_{\min} \leq x_i \leq x_{\max} \\ & u_{i+1} = u_i, \quad i = \{1, \dots, 9\} \end{aligned}$$

# Explicit Solution

MPT

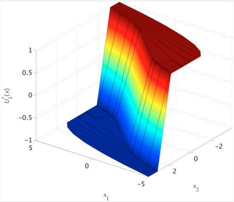
```
% Compute explicit solution
explicit_ctrl = online_ctrl.toExplicit()
```

```
% Plot control law (primal solution)
explicit_ctrl.optimizer.fplot('primal')
```

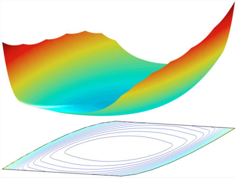
```
% plot the objective function
explicit_ctrl.optimizer.fplot('obj')
```

```
% Plot controller partition
explicit_ctrl.optimizer.plot
```

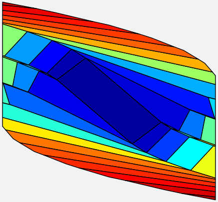
## Control input




## Value function



## Controller partition



# Exporting Explicit Solutions

**Search tree and export to flat c-file**

MPT

```
% generate binary search tree
tree = BinTreePolyUnion(explicit_ctrl.optimizer)

% export control law to C
tree.toC('primal')
```

**Export to flat c-file, sequential search**

```
% export control law to C
explicit_ctrl.optimizer.toC('primal')
```

```
#define MPT_NU 1
#define MPT_NX 2
static float MPT_ST[] = {
-5.225216e-01,  8.526260e-01,   2.007907e+00,   2.000000e+00,   3.000000e+00,
-6.669647e-01,  7.450893e-01,   3.576296e+00,   4.000000e+00,   5.000000e+00,
6.347758e-01,   -7.726964e-01,  2.699058e+00,   6.000000e+00,   7.000000e+00,
-2.990218e-01,  -9.542463e-01,  -2.286671e-01,  8.000000e+00,   9.000000e+00,
-4.546596e-01,  -8.906653e-01,  3.576036e-01,   1.000000e+01,   1.100000e+01,
3.630512e-01,   9.317692e-01,   4.490581e-02,   1.200000e+01,   1.300000e+01,
4.546596e-01,   8.906653e-01,   3.576036e-01,   1.400000e+01,   1.500000e+01,
-2.528581e-01,  -9.675034e-01,  -4.781395e-01,  -5.000000e+00,  -2.600000e+01
-6.811331e-01,  7.321596e-01,   4.632449e+00,   1.600000e+01,   1.700000e+01,
-3.630512e-01,  -9.317692e-01,  4.490581e-02,   -5.000000e+00,  -6.000000e+00
-6.347758e-01,  7.726964e-01,   2.699058e+00,   1.800000e+01,   1.900000e+01,
6.811331e-01,   -7.321596e-01,  4.632449e+00,   2.000000e+01,   2.100000e+01,
6.669647e-01,   -7.450893e-01,  3.576296e+00,   2.200000e+01,   2.300000e+01,
5.845669e-01,   8.113455e-01,   7.438927e-01,   -4.000000e+00,  -1.000000e+00
5.225216e-01,   -8.526260e-01,  2.007907e+00,   2.400000e+01,   2.500000e+01,
2.528581e-01,   9.675034e-01,   1.703078e+00,   -4.000000e+00,  -2.600000e+01
2.990218e-01,   9.542463e-01,   1.489549e+00,   -4.000000e+00,  -1.400000e+01
3.630512e-01,   9.317692e-01,   1.263985e+00,   -4.000000e+00,  -6.000000e+00
4.546596e-01,   8.906653e-01,   1.018374e+00,   -4.000000e+00,  -2.000000e+00
2.528581e-01,   9.675034e-01,   -4.781395e-01,  -4.000000e+00,  -3.100000e+01
2.990218e-01,   9.542463e-01,   -2.286671e-01,  -4.000000e+00,  -1.900000e+01
```

```
#ifdef tmwtypes_h
  /* RTW is used, switch to real_T data types to avoid problems with TLC compi
lation */
  static long mpt_searchTree(const real_T *X, real_T *U)
#else
  static long mpt_searchTree(const float *X, float *U)
#endif
{
    int ix, iu;
    long node = 1, row;
    float hx, k;

    /* initialize U to zero*/
    for (iu=0; iu<MPT_NU; iu++) {
        U[iu] = 0;
    }

    /* find region which contains the state x0  */
    while (node > 0) {
        hx = 0;
        row = (node-1)*(MPT_NX+3);
        for (ix=0; ix<MPT_NX; ix++) {
            hx = hx + MPT_ST[row+ix]*X[ix];
        }
        k = MPT_ST[row+MPT_NX];
```

# Ball and Plate - Explicit Controller

- $4$ states $+ 1$ tracking variable $= 5$ parameters
- Move-blocking reduces complexity
  - Horizon of 10
  - Inputs 2-10 must be equal

$$
\begin{aligned}
J^\star(x, y_t) = \min \; & \sum_{i=0}^{9} 100\|y_i - y_t\|_2^2 + 0.1\|u_i\|_2^2 \\
\text{s.t. } & x_0 = x \\
& x_{i+1} = Ax_i + Bu_i \\
& y_i = Cx_i \\
& u_{\min} \le u_i \le u_{\max} \\
& y_{\min} \le y_i \le y_{\max} \\
& u_{i+1} = u_i, \quad i = \{1, \ldots, 9\}
\end{aligned}
$$

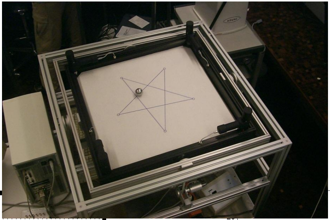
**Explicit solution (per dimension)**

Regions : 529

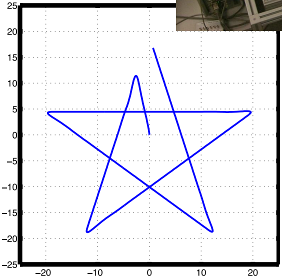
Storage : 48’000 numbers
(192 kB)

Computation : 89’000 FLOPS
(~1ms)

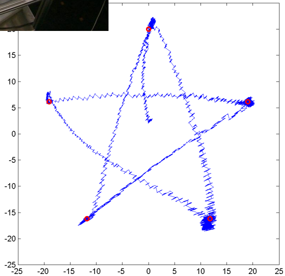
# Ball and Plate - Pentagram



Simulation




Experiment



# Real-time MPC Software Toolbox

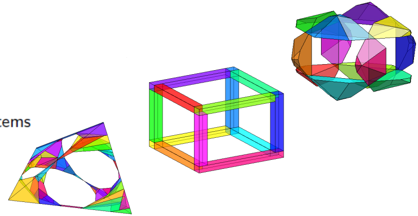
**Formal specification**
- YALMIP
- HYSDEL
- Linear + Hybrid models

**Control law**
- Explicit MPC
- Fixed-complexity solutions

**Software synthesis**
- Real-time workshop
- Bounded-time solvers
- Verifiable code generation

**Verified controller**

Multi-Parametric Toolbox (MPT)
- Computational geometry
- Multi-parametric programming
- Control of linear and hybrid systems



# Table of Contents

# Summary

- Linear MPC $+$ Quadratic or linear-norm cost $\Rightarrow$ Controller is PWA function
- We can pre-compute this function offline efficiently
- Online evaluation of a PWA function is very fast (ns - $\mu$s)
- We can only do this for very small systems! (3-6 states)