کنترل پیش بین
Model Predictive Control

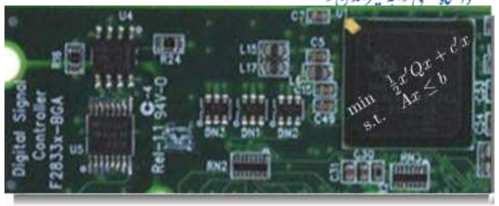
ارائه کننده: امیرحسین نیکوفرد
مهندسی برق و کامپیوتر دانشگاه خواجه نصیر

دانشگاه صنعتی خواجه نصیرالدین طوسی

Explicit Model Predictive Control

- Introduction
- Multiparametric Programming
- Explicit Linear MPC
- Explicit Hybrid MPC
- Summery

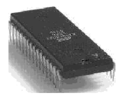
Introduction



Key requirements for deploying MPC in production:

1. speed (throughput)
   - — worst-case execution time less than sampling interval
   - — also fast on average (to free the processor to execute other tasks)

2. limited memory and CPU power (e.g., 150 MHz/50 kB)



3. numerical robustness (single precision arithmetic)

4. certification of worst-case execution time

5. code simple enough to be validated/verified/certified (library-free C code, easy to check by production engineers)

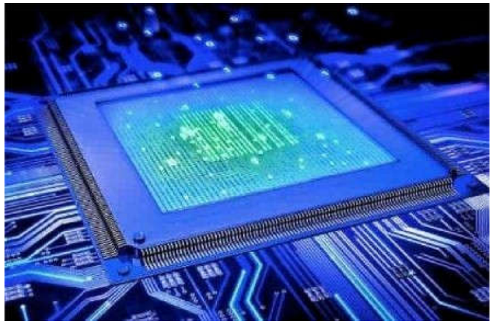
Introduction

Embedded solvers in industrial production

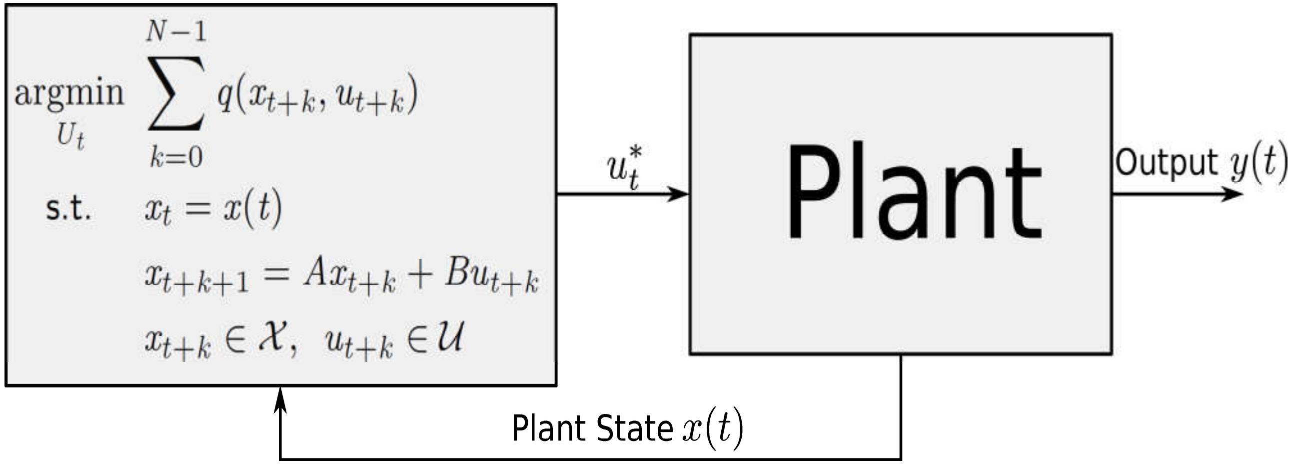
- Multivariable MPC controller
- Sampling frequency = 40 Hz (= 1 QP solved every 25 ms)
- Vehicle operating ≈1 hr/day for ≈360 days/year on average
- Controller running on 10 million vehicles

~520,000,000,000,000 QP/yr

and none of them should fail.



Introduction



Standard Model Predictive Control

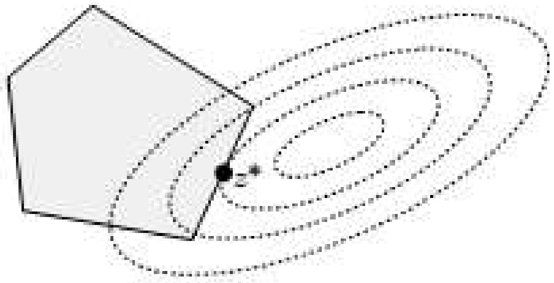
- **Standard MPC**: requires solving an optimization problem online (at each time step t) with updated state measurement

Introduction

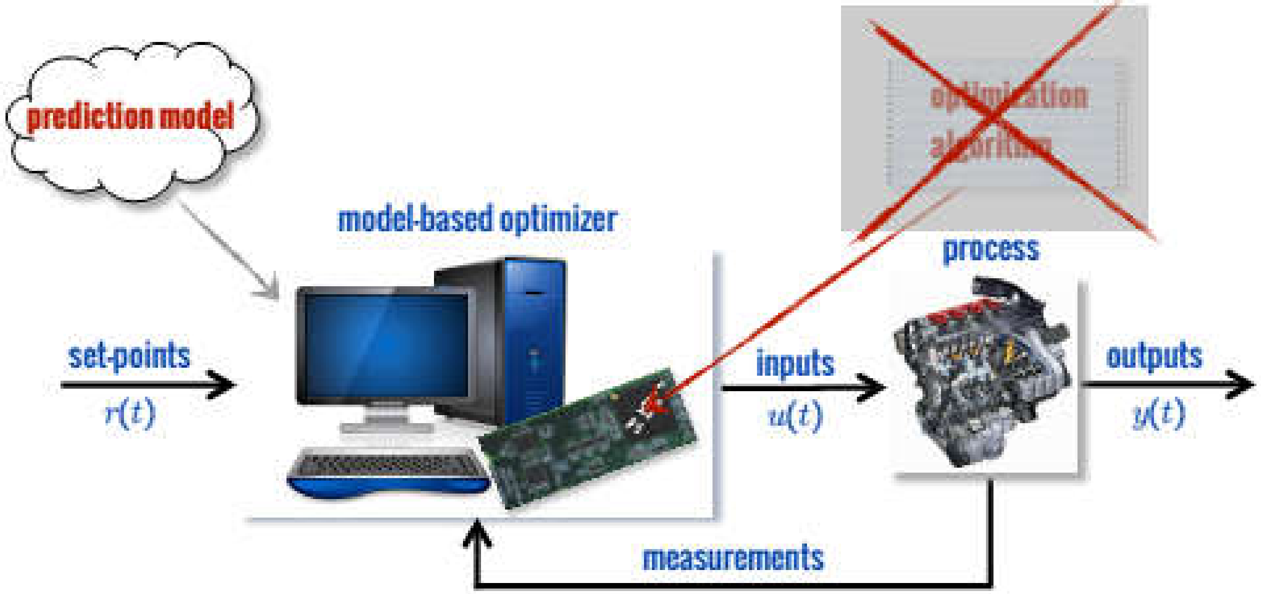
- Most used algorithms for solving QP problems:
  - active set methods
  - interior-point methods
  - gradient projection methods
  - alternating direction method of multipliers (ADMM)
  - ...

$$\begin{aligned} \min_z \quad & \tfrac{1}{2} z' Q z + x' F' z \\ \text{s.t.} \quad & G z \leq W + S x \end{aligned}$$

Quadratic Program (QP)



Introduction



- Can we implement constrained linear MPC without an on-line QP solver ?

YES !

Introduction

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

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

- On-line optimization: given $x(t)$ solve the problem at each time step t (the control law $u = u_0^*(x)$ is implicitly defined by the QP solver)

→ Quadratic Programming (QP)

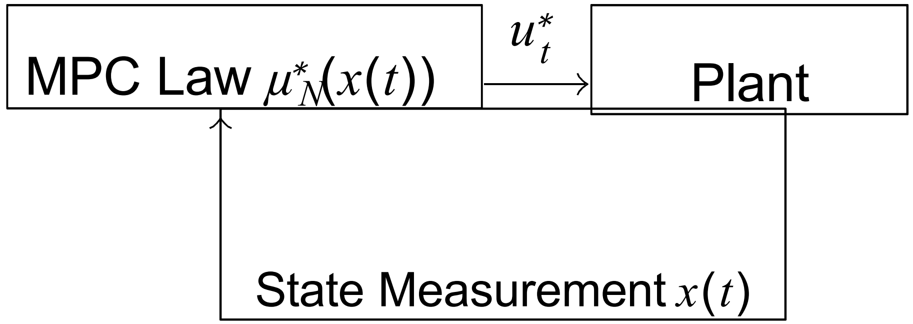
- Off-line optimization: solve the QP in advance for all $x(t)$ in a given range to find the control law $u = u_0^*(x)$ explicitly

→ multi-parametric Quadratic Programming (mpQP)

Introduction

Precomputed Offline

$$
\begin{aligned}
\mu_N(x(t)) \quad & = \operatorname{argmin}_{U_0} J_N(x(t), U_0) \\
\text{subj. to:} \quad & x_{k+1} = f(x_k, u_k) \\
& x_k \in X, \; u_k \in U \\
& x_N \in X_f, \; x_0 = x(t)
\end{aligned}
$$



Explicit Model Predictive Control

- Pre-compute control law μ_N^* as function of state x so that online computation is dramatically reduced
- Main Tool: Multiparametric Programming
- May not work when the state dimension is large or the system dynamics is nonlinear.

Explicit Model Predictive Control

- Introduction
- Multiparametric Programming
- Explicit Linear MPC
- Explicit Hybrid MPC
- Summery

Multiparametric Programming

- Multiparametric programming problem:

$$J^*(x) = \begin{array}{ll} & \min_v J(x,\ v) \\ \text{subj. to.} & Gv \leq w + Sx \end{array} \qquad (1)$$

- Optimization variable is $v \in \mathbb{R}^{n_v}$, while $x \in \mathbb{R}^n$ is viewed as a parameter of the problem

- Minimum cost $J^*(x)$ is called the *value function*

- The minimizer function is parameter dependent:

$$v^*(x) = \text{argmin}_v \{J(x, v) : Gv \leq w + Sx\} \qquad (2)$$

Multiparametric Programming

- Multiparametric programming studies properties of the value function $J^*(x)$ and minimizer function $v^*(x)$ w.r.t. parameter x

- Region of feasible parameters:

$$\mathcal{K}^* = \{x \in \mathbb{R}^n : \exists v \in \mathbb{R}^{n_v} \text{ s.t. } Gv \leq w + Sx\}$$

- **Assumptions:**

1. $\mathcal{K}^*$ is full dimensional polytope in $\mathbb{R}^n$

2. S is full column rank

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

Multiparametric Programming

Definition 1 (Active Set).

We define the *active set* at x, $\mathcal{A}(x)$, and its complement, $\mathcal{A}^c(x)$, as

$$\mathcal{A}(x) \triangleq \{i \in I : G_i v^*(x) - S_i x = w_i\}$$
$$\mathcal{A}^c(x) \triangleq \{i \in I : G_i v^*(x) - S_i x < w_i\}$$

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

Definition 2 (Critical Region).

$\mathcal{CR}_J$ is the set of parameters x for which the same set of constraints $J \subseteq I$ is active at the optimum, i.e.,

$$\mathcal{CR}_J \triangleq \{x \in \mathcal{K}^* : \mathcal{A}(x) = J\}$$

Multiparametric Programming

- **Standard mp-LP form:**

$$J^*(x) = \min_v c^T v \quad \text{subj. to} \quad Gv \leq w + Sx$$

- Optimizer function: $v^*(x) = \text{argmin}_v\{c^T v : \text{subj. to } Gv \leq w + Sx\}$
- $G, w, , S, c$ are known constants
- v is optimization variable
- x is parametric variable

Multiparametric Programming

- The following theorem summarizes the properties of the mp-LP solution.

Theorem 1 (Solution of mp-LP).

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

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

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* $\mathcal{CR}_i$.

Multiparametric Programming

- Multiparametric Quadratic Programming (mp-QP)
  - Standard mp-QP form:

$$J^*(x) = \begin{cases} & \min_v v^T H v \\ \text{subj. to} & Gv \leq w + Sx \end{cases}$$

  - $v^*(x) = \text{argmin}_v \{v^T H v : \text{ subj. to } Gv \leq w + Sx\}$
  - $G, w, , S, H$ are known constants
  - v is optimization variable
  - x is parametric variable

Multiparametric Programming

- Multiparametric Quadratic Programming (mp-QP)
  - The following theorem summarizes the properties of the mp-QP solution.

Theorem 2 (Solution of mp-QP).

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

ii) *The optimizer function* $v^*(x) : \mathcal{K}^* \to \mathbb{R}^{n_v}$ *is:*
- **continuous**
- **polyhedral piecewise affine over** $\mathcal{K}^*$, *it is affine in each* $\mathcal{CR}_i$.

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* $\mathcal{CR}_i$.

Multiparametric Programming

- Multiparametric Quadratic Programming (mp-QP)

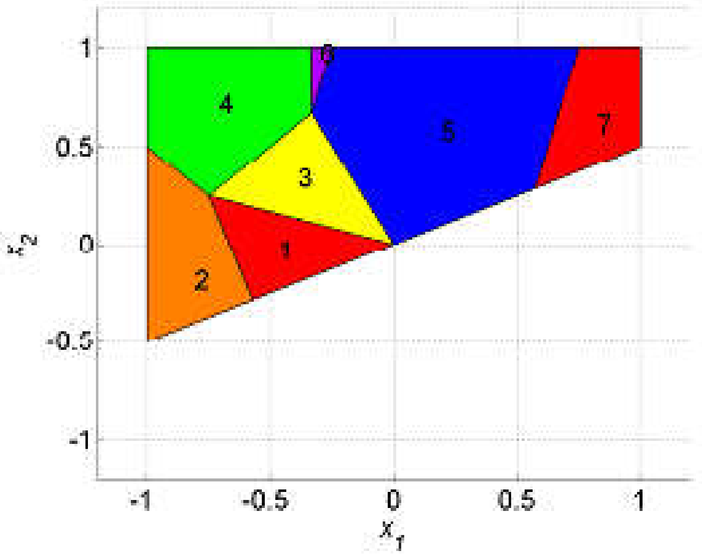
Consider the example

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

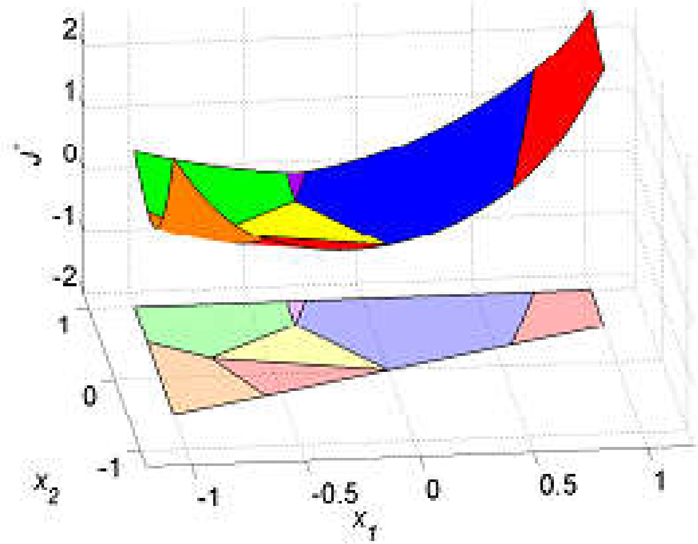
Multiparametric Programming

- Multiparametric Quadratic Programming (mp-QP)

The explicit solution is defined over $i = 1, \ldots, 7$ 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



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

Multiparametric Programming

- Multiparametric Quadratic Programming (mp-QP)

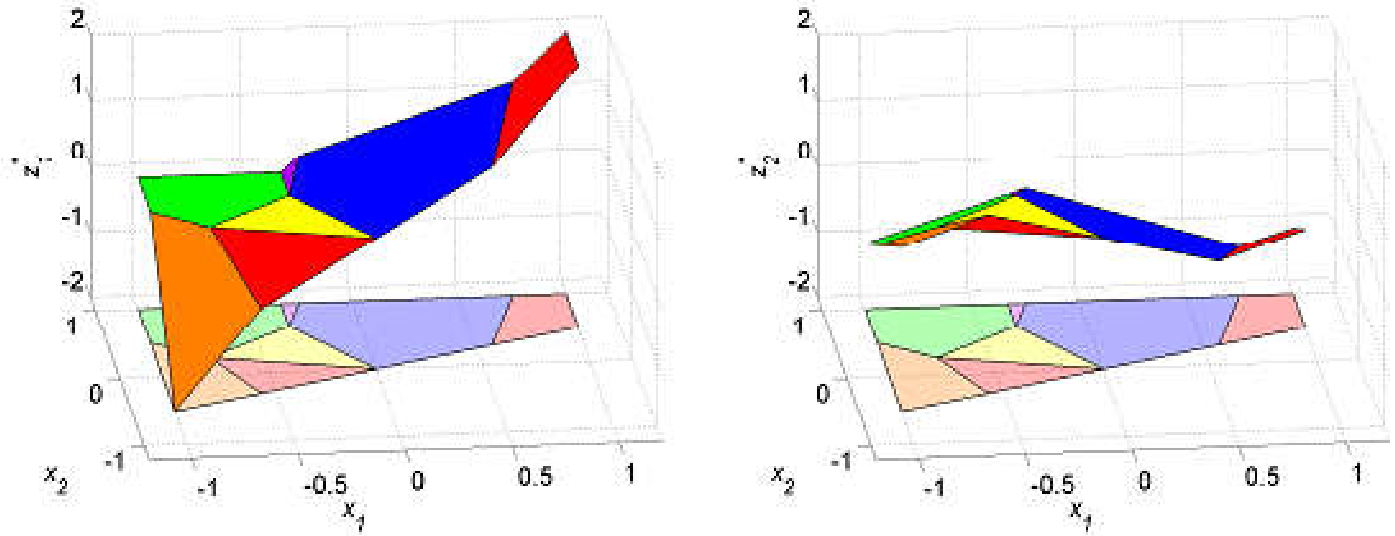
Primal solution is given as piecewise affine function $z(x) = F_i + g_i x$ if $x \in \mathcal{P}_i$.

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

Multiparametric Programming

- Multiparametric Quadratic Programming (mp-QP)

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 $z_2^*(x)$

Multiparametric Programming

- Multiparametric Mixed-Integer Linear Programming (mp-MILP)
  - Continuous and discrete optimization variables: $v = \{v_c, v_d\}$, $v_c \in \mathbb{R}^{n_c}$ $v_d \in \{0,1\}^{n_d}$, and $n_v = n_c + n_d$.
  - Standard mp-MILP form:

$$J^*(x) = \begin{cases} & \min_v c^T v \\ \text{subj. to} & Gv \leq w + Sx \\ & v = \{v_c, v_d\}, v_c \in \mathbb{R}^{n_c}, v_d \in \{0,1\}^{n_d} \end{cases}$$

  - Optimizer function: $v^*(x)$
  - G, w, S, c are known constants
  - v is optimization variable
  - x is parametric variable

Multiparametric Programming

- Multiparametric Mixed-Integer Linear Programming (mp-MILP)
  - The following theorem summarizes the properties of the mp-MILP solution.

Theorem 3 (Solution of mp-MILP).

i) *The feasible set $\mathcal{K}^*$ is the* **union of a finite number of (possibly open) polyhedra**.

ii) *If the optimal solution v^* is unique for $\forall x \in \mathcal{K}^*$, the optimizer function $v_c^*(x) : \mathcal{K}^* \to \mathbb{R}^{n_c}$ and $v_d^*(x) : \mathcal{K}^* \to \mathbb{R}^{n_d}$ are:*

- **polyhedral piecewise affine and polyhedral piecewise constant over $\mathcal{K}^*$, respectively**, *they are affine and constant in each $\mathcal{CR}_i$, respectively.*

Otherwise, it is always possible to define a piecewise affine optimizer function $v^*(x)$ for all $x \in \mathcal{K}^*$.

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

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

- Note that, different from the mp-LP case, the set $\mathcal{K}^*$ can be non-convex and even disconnected.

Explicit Model Predictive Control

- Introduction
- Multiparametric Programming
- Explicit Linear MPC
- Explicit Hybrid MPC
- Summery

Explicit Linear MPC

- Linear MPC: Linear systems with polyhedral constraints

- At time t: solve the following N-horizon optimal control problem:

$$\mathcal{P}_N(x(t)) : V_N(x(t)) = \begin{cases} \min_{\mathbf{u}} & J(x(t), U_0) \triangleq J_f(x_N) + \sum_{k=0}^{N-1} l(x_k, u_k) \\ \text{subj. to:} & x_{k+1} = Ax_k + Bu_k, k \leq N-1 \\ & A_x x_k \leq b_x, A_u u_k \leq b_u, k \leq N-1 \\ & A_f x_N \leq b_f, \quad x_0 = x(t) \end{cases} \quad (3)$$

- Cost function:

$$J_N(x(t), U_0) = \|Q_f x_N\|_p + \sum_{k=0}^{N-1} (\|Qx_k\|_p + \|Ru_k\|_p) \quad (4)$$

- We will consider cases: $p = 2, \infty, 1$

Explicit Linear MPC

- For the 2-norm case, $\mathcal{P}_N(x(t))$ boils down to a quadratic programming (QP) problem:

$$V_N(x(t)) = \begin{cases} \min_{U_0} & J_N(x(t), U_0) = [U_0^T, x(t)^T] \begin{bmatrix} H & F^T \\ F & Y \end{bmatrix} \begin{bmatrix} U_0 \\ x(t) \end{bmatrix} \\ \text{subj. to} & G_0 U_0 \leq w_0 + E_0 x(t) \end{cases} \quad (5)$$

- Online solution: given $x(t)$ at each time t compute a **control action** u_0^* by solving the QP (5)

- Offline solution (Explicit MPC): pre-compute the **control law** $U_0^*(z)$ for all possible state location $x(t) = z$. The online implementation of the controller only involves evaluating the law at a given state $u_0^* = U_0^*(x(t))$

- Simplest case: no state and control constraint $\Rightarrow U_0^*(x(t)) = -H^{-1}F^T x(t)$

Explicit Linear MPC

- With polyhedral state and control constraints, explicit solution can be obtained through multiparametric programming

- Problem (5) can be transformed to standard mp-QP problem with $v = U_0 + H^{-1}F^T x(t)$.

- $\Rightarrow$ explicit MPC law is of the form:

$$U_0^*(x(t)) = \begin{cases} F_1 x(t) + f_1 & \text{if } H_1 x(t) \leq h_1 \\ \vdots & \vdots \\ F_M x(t) + f_M & \text{if } H_M x(t) \leq h_M \end{cases} \qquad (6)$$

- The above control law can be computed using the Multi-Parametric Toolbox

Explicit Linear MPC

- (∞-norm) case:

$$U_0^*(x(t)) = \text{argmin}_{U_0}\{J_N(x(t),U_0) : G_0U_0 \leq w_0 + S_0x(t)\}$$

$$\text{where } J_N(x(t),U_0) = \sum_{k=0}^{N-1}(\|Qx_k\|_\infty + \|Ru_k\|_\infty) \qquad (7)$$

- For notation simplicity, the terminal cost is dropped

- Note: $\min_x\{|x| : h(x) \leq 0\} \Leftrightarrow \min_{x,\epsilon}\{\epsilon : \epsilon \geq x, \epsilon \geq -x, h(x) \leq 0\}$

- Introduce slack variables:

$$\begin{cases} \epsilon_k^x \geq \|Qx_k\|_\infty \\ \epsilon_k^u \geq \|Ru_k\|_\infty \end{cases} \Rightarrow \begin{cases} \epsilon_k^x \geq [Qx_k]_i, & i = 1,\dots,n \quad k = 1,\dots,N-1 \\ \epsilon_k^x \geq -[Qx_k]_i & i = 1,\dots,n \quad k = 1,\dots,N-1 \\ \epsilon_k^u \geq [Ru_k]_i & i = 1,\dots,m \quad k = 0,\dots,N-1 \\ \epsilon_k^u \geq -[Ru_k]_i & i = 1,\dots,m \quad k = 0,\dots,N-1 \end{cases}$$

Explicit Linear MPC

- The problem reduces to an optimization problem of the following form:

$$\begin{cases} \min\limits_{\xi} & \sum\limits_{k}^{N-1} \epsilon_k^x + \epsilon_k^u \\ \text{subj. to} & G\xi \leq W + Sx(t) \end{cases} \tag{8}$$

  where $\xi = \{\epsilon_1^x, \ldots, \epsilon_{N-1}^x, \epsilon_0^u, \ldots, \epsilon_{N-1}^u, u_0, \ldots, u_{N-1}\}$.

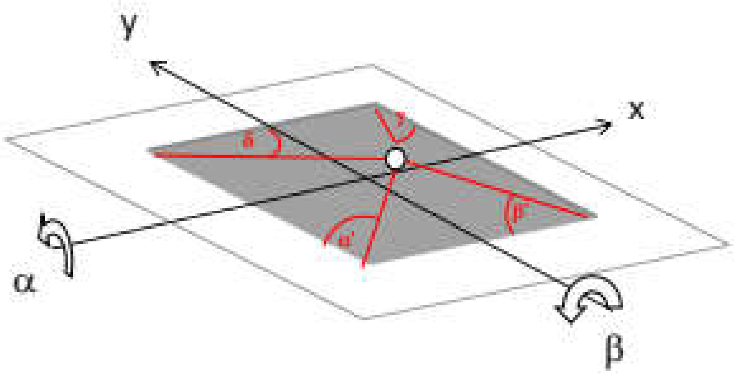
- Therefore, the ∞-norm linear MPC problem can be reformulated as an mp-LP problem

- $U_0^*(x)$ is also piecewise affine that can be computed using the Multi-Parametric Toolbox

- 1-norm explicit MPC can also be formulated as an mp-LP problem

Explicit Linear MPC

Ball and Plate

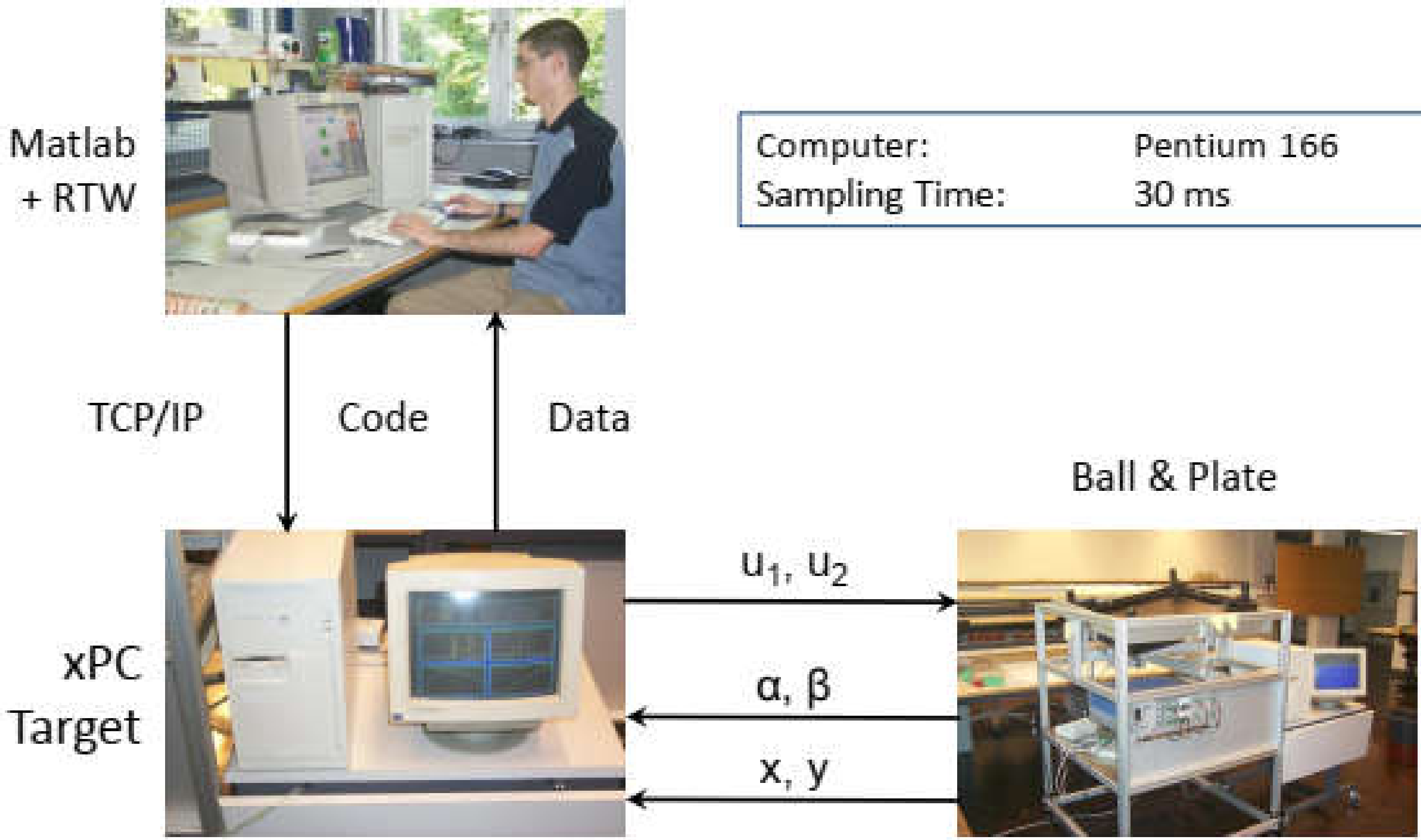


- Linearized model: four states for each axis: plate angle, ball position, plate angular speed, ball speed.
- Constraints on inputs and states
  - Plate angle
  - Ball position
  - Acceleration
- MPC objective: path tracking

Explicit Linear MPC

Ball and Plate - System

Computer:	Pentium 166
Sampling Time:	30 ms



Explicit Linear MPC

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

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

Explicit Linear MPC

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 )
```

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

Explicit Linear MPC

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 )
```

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

Explicit Linear MPC

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 )
```

$$\min \sum_{i=0}^{9} 100\|y_i - y_t\|_2^2 + 0.1\|u_i\|_2^2$$

$$\begin{aligned} \text{s.t.}\quad & x_0 = x \\ & x_{i+1} = Ax_i + Bu_i \\ & y_i = Cx_i \\ & u_{\min} \leq u_i \leq u_{\max} \\ & y_{\min} \leq y_i \leq y_{\max} \\ & x_{\min} \leq x_i \leq x_{\max} \\ & u_{i+1} = u_i, \quad i = \{1, \ldots, 9\} \end{aligned}$$

Explicit Linear MPC

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 )
```

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

Explicit Linear MPC

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 )
```

$$\min \sum_{i=0}^{9} 100\|y_i - y_t\|_2^2 + 0.1\|u_i\|_2^2$$

$$\begin{aligned}
\text{s.t. } & x_0 = x \\
& x_{i+1} = Ax_i + Bu_i \\
& y_i = Cx_i \\
& u_{\min} \le u_i \le u_{\max} \\
& y_{\min} \le y_i \le y_{\max} \\
& x_{\min} \le x_i \le x_{\max} \\
& u_{i+1} = u_i, \quad i = \{1, \ldots, 9\}
\end{aligned}$$

Explicit Linear MPC

Explicit Solution

MPT

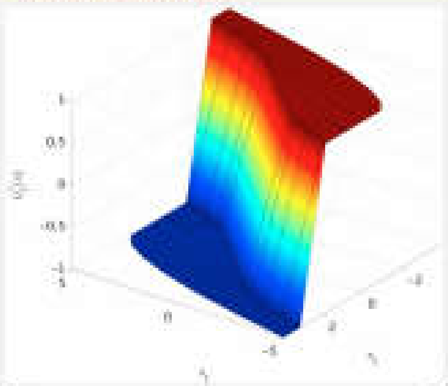
```
% Compute explicit solution
explicit_ctrl = online_ctrl.toExplicit()
```

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

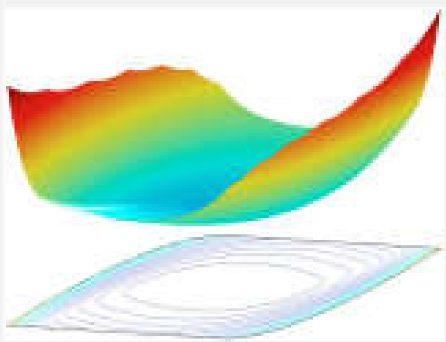
```
% plot the objective function
explicit_ctrl.optimizer.fplot('obj')
```

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

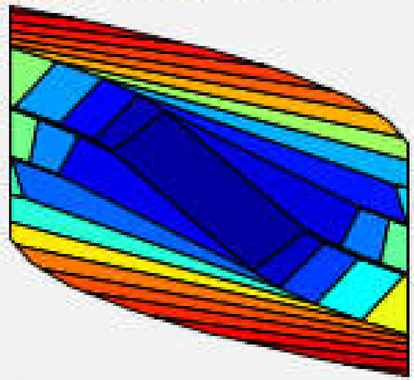
Control input



Value function



Controller partition



Explicit Linear MPC

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

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

Explicit solution (per dimension)

Regions : 529

Storage : 48'000 numbers
(192 kB)

Computation : 89'000 FLOPS
(~1ms)

Explicit Model Predictive Control

- Introduction
- Multiparametric Programming
- Explicit Linear MPC
- Explicit Hybrid MPC
- Summery

Explicit Hybrid MPC

- Let us use the Mixed Logical Dynamical (MLD) representation

$$\begin{cases} x(t+1) = Ax(t) + B_1u(t) + B_2\delta(t) + B_3z(t) + B_5 \\ y(t) = Cx(t) + D_1u(t) + D_2\delta(t) + D_3z(t) + D_5 \\ E_2\delta(t) + E_3z(t) \leq E_1u(t) + E_4x(t) + E_5 \end{cases} \quad (9)$$

  - $x(t) \in \mathbb{R}^{n_c} \times \{0,1\}^{n_l}$, $u(t) \in \mathbb{R}^{m_c} \times \{0,1\}^{m_l}$
  - $y(t) \in \mathbb{R}^{p_c} \times \{0,1\}^{p_l}$, $\delta(t) \in \{0,1\}^{r_l}$, $z(t) \in \mathbb{R}^{r_c}$

- Recall that MLD is equivalent to Piecewise affine system and Dynamical Hybrid Automaton model.

Explicit Hybrid MPC

∞-norm case:

$$\begin{cases} \min_{U_0} J(x(t), U_0) = \sum_{k=0}^{N-1} (\|Qx_k\|_\infty + \|Ru_k\|_\infty) \\ \text{subj. to. MLD (9)} \end{cases} \tag{10}$$

- Introduce slack variables:

$$\begin{cases} \epsilon_k^x \geq \|Qx_k\|_\infty \\ \epsilon_k^u \geq \|Ru_k\|_\infty \end{cases} \Rightarrow \begin{cases} \epsilon_k^x \geq [Qx_k]_i, & i = 1, \ldots, n \quad k = 1, \ldots, N-1 \\ \epsilon_k^x \geq -[Qx_k]_i & i = 1, \ldots, n \quad k = 1, \ldots, N-1 \\ \epsilon_k^u \geq [Ru_k]_i & i = 1, \ldots, m \quad k = 1, \ldots, N-1 \\ \epsilon_k^u \geq -[Ru_k]_i & i = 1, \ldots, m \quad k = 1, \ldots, N-1 \end{cases}$$

- The problem reduces to an optimization problem of the following form:

$$\begin{cases} \min_{\xi} & \sum_{k}^{N-1} \epsilon_k^x + \epsilon_k^u \\ \text{subj. to} & G\xi \leq W + Sx(t) \end{cases} \tag{11}$$

where $\xi = \{\epsilon_1^x, \ldots, \epsilon_{N-1}^x, \epsilon_0^u, \ldots, \epsilon_{N-1}^u, u_0, \ldots, u_{N-1}, \delta_0, \ldots, \delta_{N-1}, z_0, \ldots, z_{N-1}\}$.

Explicit Hybrid MPC

- Problem (11) is a Mixed Integer Linear Programming (MILP) problem.

- Explicit MPC solutions can also be computed offline using multiparametric MILP (mp-MILP).

- **Explicit Hybrid MPC:** Accordingly to the properties of mp-MILP, the hybrid MPC controller is piecewise affine

$$U_0^*(x(t)) = \begin{cases} F_1 x(t) + g_1 & \text{if } x(t) \in \text{ partition } 1 \\ \vdots & \\ F_M x(t) + g_M & \text{if } x(t) \in \text{ partition } M \end{cases}$$

- The partition is polyhedral.

Explicit Hybrid MPC

- 2-norm case:

$$\begin{cases} \min_{U_0} J(x(t), U_0) = x_N^T Q_f x_N + \sum_{k=0}^{N-1} \left(x_k^T Q x_k + u_k^T R u_k \right) \\ \text{subj. to. MLD (9)} \end{cases} \tag{12}$$

- This can be reduced to

$$\begin{cases} \min_{\xi} & \xi^T H \xi + 2x(t)^T F \xi + x(t)^T Y x(t) \\ \text{subj. to} & G\xi \leq W + Sx(t) \end{cases} \tag{13}$$

where $\xi = \{u_0, \ldots, u_{N-1}, \delta_0, \ldots, \delta_{N-1}, z_0, \ldots, z_{N-1}\}$.

Explicit Hybrid MPC

- Explicit MPC: The optimal control law can be computed through Multiparametric Mixed Integer Quadratic Programming (mp-MIQP).

- It can be shown that optimizer function of mp-MIQP is still piecewise affine.

- Therefore, the explicit hybrid MPC controller is piecewise affine

$$U_0^*(x(t)) = \begin{cases} F_1 x(t) + g_1 & \text{if } x(t) \in \text{ partition } 1 \\ \vdots & \\ F_M x(t) + g_M & \text{if } x(t) \in \text{ partition } M \end{cases}$$

- However, for the 2-norm case, the partitions may not be polyhedral.

Explicit Model Predictive Control

- Introduction
- Multiparametric Programming
- Explicit Linear MPC
- Explicit Hybrid MPC
- Summery

Explicit MPC: Summary

- Explicit MPC: compute the explicit optimal control law μ^*_N offline (the first control law in the N-horizon optimal policy π^*_N). Applying this control law repetitively online is equivalent to the standard MPC controller.
- Main theoretical tool: multi-parametric programming.
- Explicit Linear MPC:
  - 1-norm and ∞-norm ⇒ mp-LP ⇒ piecewise affine control law with polyhedral partitions
  - 2-norm: mp-QP ⇒ piecewise affine control law with polyhedral partitions
- Explicit Hybrid MPC (with Mixed Logical Dynamical Systems):
  - 1-norm and ∞-norm ⇒ mp-MILP ⇒ piecewise affine control law with polyhedral partitions
  - 2-norm: mp-MIQP ⇒ piecewise affine control law with non-polyhedral partitions
- Very useful software: Multi-Parametric Toolbox [Herceg2013]
- M. Herceg et al."Multi-Parametric Toolbox 3.0".In Proc. of the European Control Conference. http://control.ee.ethz.ch/~mpt. Zurich, Switzerland, 2013.