Computing the explicit MPC solution using the Hasse diagram of the lifted feasible domain

Florin Stoican, Ștefan S. Mihai

POLITEHNICA University of Bucharest (UPB) - Automatic Control and Systems Engineering Department

25.05.2023 - RTE workshop (CentraleSupelec)
Outline

1 Motivation

2 Preliminaries

3 Geometrical interpretation of the explicit MPC representation

4 Explicit MPC toolbox
Outline

1 Motivation

2 Preliminaries

3 Geometrical interpretation of the explicit MPC representation

4 Explicit MPC toolbox
Motivation

- The Explicit MPC allows to solve the optimization problem off-line.

- The optimal control is an "explicit" function of the state $\rightarrow$ the on-line operations become simple function evaluations.

- Usually, the control law is a piecewise affine (PWA) function $\rightarrow$ the controller is stored in a lookup table of affine gains.

- Since both the control law and the cost surface are known (piecewise affine and, respectively, quadratic), stability and performance can be analyzed offline.

Idea

Exploit the geometrical structure of the problem to reduce the computation time of the eMPC!
Outline

1 Motivation

2 Preliminaries
   • Polytopic sets
   • The face lattice
   • The MPC problem

3 Geometrical interpretation of the explicit MPC representation

4 Explicit MPC toolbox
Polyhedron – general notions

- Given $A \in \mathbb{R}^{d_H \times d}$, $b \in \mathbb{R}^{d_H}$ and $V \in \mathbb{R}^{d \times d_V}$, $R \in \mathbb{R}^{d \times d_R}$, a polyhedral set can be represented using the\(^1\):
  - **half-space representation**, i.e., the intersection of $d_H$ linear inequalities
    \[
P(A, b) = \left\{ x \in \mathbb{R}^d : A_i x \leq b_i, \forall i \in \{1, \ldots, d_H\} \right\};
    \]
  - **generator representation**, i.e., the convex sum of $d_V$ vertices added to the linear sum of $d_R$ rays
    \[
P(V, R) = \left\{ x \in \mathbb{R}^d : x = \sum_{j=1}^{d_V} \alpha_j V_j + \sum_{k}^{d_R} \beta_k R_k, \; \alpha_j, \beta_k \geq 0; \sum_{j=1}^{d_V} \alpha_j = 1, \; \forall j \in \{1, \ldots, d_V\}, \; k \in \{1, \ldots, d_R\} \right\}.
    \]
- A bounded polyhedron is called a **polytope**.

---

Preliminaries | Polytopic sets

Polytope – faces/ $f$-vector

A face $F$ may be represented in$^2$:

- **half-space form**, by a combination of active/inactive constraints, indexed by sets $A$ – active, $I$ – inactive, where $A \cup I = \{1, \ldots, d_H\}$ and $A \cap I = \emptyset$, as

$$F(A) = \left\{ A_i x = b_i, \forall i \in A, \ A_i x \leq b_i, \forall i \in I \right\};$$

- **generator form**, by a collection of vertices indexed by the set $V \subset \{1 : d_V\}$, as

$$F(V) = \left\{ x = \sum_{j \in V} \alpha_j V_j, \sum_{j \in V} \alpha_j = 1, \ \alpha_j \geq 0, \forall j \in V \right\}.$$  

A face $F$ is called a $k$–face if it is embedded in a $k$–subspace in $\mathbb{R}^d$.

The $f$-vector of $P$ is given as:

$$f(P) = (f_{-1}, f_0, f_1, \ldots, f_d)$$

$f_k$ is the number of $k$–dimensional faces.

the convention is that $f_{-1} = f_d = 1$

---

DOI: 10.3929/ethz-b-000426218.
Preliminaries

The face lattice

- The collection of all faces of a polytope $P$ is called its face lattice.
- It enjoys a partial ordering relation $\rightarrow$ graph representation, the so-called Hasse diagram.

The Hasse diagram (right) of the polytope (left)
The cube and its polar, the cross-polytope
Comparison times for Hasse diagram computation

![Graph showing comparison times for Hasse diagram computation]

- **Original**
- **Polar**
- **n-skeleton**

<table>
<thead>
<tr>
<th>Prediction horizon N</th>
<th>Time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>10^{-2}</td>
</tr>
<tr>
<td>3</td>
<td>10^{0}</td>
</tr>
<tr>
<td>4</td>
<td>10^{1}</td>
</tr>
<tr>
<td>5</td>
<td>10^{2}</td>
</tr>
</tbody>
</table>
The MPC problem

- Consider the linear time-invariant (LTI) discrete system:

\[ x_{k+1} = Ax_k + Bu_k, \quad y_k = Cx_k. \]

- Then the typical (quadratic cost and linear constraints) MPC problem is:

\[
\begin{align*}
\left. \begin{array}{l}
\mathbf{u}^*_N = \arg \min_{\mathbf{u}_N} \mathbf{x}_N^T S \mathbf{x}_N + \sum_{k=0}^{N-1} \left( \mathbf{x}_k^T Q \mathbf{x}_k + \mathbf{u}_k^T R \mathbf{u}_k \right), \\
\text{s.t.} \quad \mathbf{x}_{k+1} = A \mathbf{x}_k + B \mathbf{u}_k, \\
\quad \mathbf{y}_k = C \mathbf{x}_k, \\
\quad \mathbf{x}_k \in \mathcal{X}, \quad \mathbf{u}_k \in \mathcal{U}, \quad \mathbf{y}_k \in \mathcal{Y}, \\
\quad \mathbf{x}_N \in \mathcal{X}_f.
\end{array} \right\}
\end{align*}
\]

- The equivalent multi-parametric quadratic program (mp-QP):

\[
\begin{align*}
\left. \begin{array}{l}
\mathbf{u}^*_N(\mathbf{x}_0) = \arg \min_{\mathbf{u}_N} \frac{1}{2} \mathbf{u}_N^T \mathbf{Q} \mathbf{u}_N + \mathbf{x}_0^T \mathbf{H} \mathbf{u}_N, \\
\text{s.t.} \quad A \mathbf{u}_N \leq \mathbf{b} + E \mathbf{x}_0,
\end{array} \right\}
\end{align*}
\]
The compact form

For further use, we introduce auxiliary notation:

\[
    x_N = [x_1^T \ldots x_N^T]^T, y_N = [y_1^T \ldots y_N^T]^T, \\
    \Theta_N = \begin{bmatrix}
        A \\
        A^2 \\
        \vdots \\
        A^N
    \end{bmatrix}, \Phi_N = \\
    \begin{bmatrix}
        B & 0 & \ldots & 0 \\
        AB & B & \ldots & 0 \\
        \vdots & \vdots & \ddots & \vdots \\
        A^{N-1}B & A^{N-2}B & \ldots & B
    \end{bmatrix}, \\
    X_N = \text{diag}\{X, \ldots, X\}, \forall X \in \{Q, R, C\}, N = [0, \ldots, 0, 1].
\]

This allows to rewrite the MPC problem compactly:

\[
    u_N^* = \arg \min_{u_N} u_N^T \left[ \Phi_N^T (Q_N + S_N^T) \Phi_N + R_N \right] u_N \\
    + 2x_0^T \Theta_N^T (Q_N + S_N^T) \Phi_N u_N + x_0^T \Theta_N^T (Q_N + S_N) \Theta_N x_0, \\
    \text{s.t.} \quad \Theta_N x_0 + \Phi_N u_N \in X_N, \\
    u_N \in U_N, \\
    C_N(\Theta_N x_0 + \Phi_N u_N) \in Y_N, \\
    N(\Theta_N x_0 + \Phi_N u_N) \in X_f.
\]
Outline

1 Motivation

2 Preliminaries

3 Geometrical interpretation of the explicit MPC representation
   - Main idea
   - Non-emptiness (visibility) tests
   - Illustrative example

4 Explicit MPC toolbox
The KKT conditions

The usual approach is to rewrite the initial problem in its dual form via the Karush-Kuhn-Tucker (KKT) optimality conditions

\[ \tilde{Q}u_N^* + \tilde{H}^T x_0 + A^T \lambda^* = 0, \]
\[ Au_N^* - Ex_0 \leq b, \]
\[ \lambda^* \geq 0, \]
\[ \lambda^* \times (Au_N^* - Ex_0 - b) = 0. \]

Taking a particular subset \( \mathcal{A} \subset \{1, 2, \ldots \} \) of inequalities to be active, we arrive at

\[ \tilde{Q}u_N^* + \tilde{H}^T x_0 + A_\mathcal{A}^T \lambda^* = 0, \]
\[ A_\mathcal{A} u_N^* - E_\mathcal{A} x_0 = b_\mathcal{A}, \quad A_\mathcal{I} u_N^* - E_\mathcal{I} x_0 \leq b_\mathcal{I} \]
\[ \lambda^*_\mathcal{A} \geq 0, \quad \lambda^*_\mathcal{I} = 0 \]

where \( \mathcal{I} = \{1, 2, \ldots \} \setminus \mathcal{A} \).
Geometrical interpretation of the explicit MPC representation

The explicit MPC solution

The mp-QP may be exploited (using the KKT conditions) to explicitly and offline give the input in terms of the current value of the state (piecewise formulation):

- for a given $x_0$, a subset of constraints, $\mathcal{A}$, is active;
- to it, corresponds a critical region: $CR_A = \{Z_A x_0 \leq z_A \}$
- over which, the constrained optimum is defined:
  \[ u_N^*(x_0) = L_A x_0 + l_A \]

To find all critical regions/control laws, we need to iterate all possible candidate sets of active constraints $\mathcal{A}$!

Issue

- the number of critical regions increases exponentially with prediction horizon.

---

Geometrical interpretation – I

Define

- The subspace of all the unconstrained solutions of the mp-QP:
  \[ \overline{U} = \left\{ \begin{bmatrix} u_N \\ x_0 \end{bmatrix} : \begin{bmatrix} -\tilde{Q}^{-1} & \tilde{H}^T \\ I \end{bmatrix} x_0 , \forall x_0 \in \mathbb{R}^n \right\} ; \]

- The lifted feasible domain:
  \[ P \left( \begin{bmatrix} A & -E \end{bmatrix} , b \right) . \]

Main idea

- Inspired by the work of Seron\(^4\)\(^5\), we state that a subset of faces of the lifted feasible domain correspond to the critical regions.

- The faces which do not correspond to the critical regions are hidden → we developed a visibility test.

---


Geometrical interpretation – II

- At each value of $x_0$ a different instance of the feasible domain is active.
- We can look at the problem in a lifted space (input plus initial state).
- We now have three elements:
  - the subspace of the unconstrained optimum
  - the lifted feasible domain
  - the piecewise constrained optimal solution

```
A_Au = E_Ax_0 + b_A \quad \text{(active constraints)}
```

```
\bar{u}_N(x_0) \quad u^*_N(x_0) = \bar{u}_N(x_0)
```

```
Au_N \leq Ex_0 + b \quad \text{(feasible domain)}
```

```
\text{cost iso-levels}
```

Geometrical interpretation of the problem – III

\[ x_0 = -2.35 \]
\[ x_0 = -1.35 \]
\[ x_0 = 1.75 \]
Associated Hasse diagram
Complexity bounds

- The candidate set $\mathcal{A}$ of active constraints is in fact an intersection of faces ($\mathcal{F}(\mathcal{A}) = \{h_i x = k_i, \forall i \in \mathcal{A}\} \cap \{h_i x \leq k_i, \forall i \notin \mathcal{A}\}$).

- The upper bound for the number of candidate sets / critical regions is given by the complexity of the feasible domain (its number of faces):

$$l(P) = \sum_{\mathcal{A} \subset \{1, \ldots, N_h\}, \mathcal{F}(\mathcal{A}) \neq \emptyset} 1 = \sum_{k=0}^{d-1} f_k(P),$$

where $f_k(P)$ is the number of $k$-order faces of the polyhedron $P$.

- The bound for the $k$-order face for a polyhedron is [Fukuda 2020]:

$$f_k(P) \leq f_{d-k-1}(c(d, N_h)), \forall k = 0, 1, \ldots, d - 2,$$

where:

$$f_k(c(d, N_h)) = \sum_{r=0}^{\lfloor d/2 \rfloor} \binom{r}{d-k-1} (N_h - d + r - 1) + \sum_{r=\lfloor d/2 \rfloor + 1}^{d} \binom{r}{d-k-1} (N_h - r - 1),$$

and $c(d, N_h)$ denotes the cyclic polytope.
The algorithm sketch

- compute the k-skeleton of the Hasse diagram for the lifted feasible domain (in its polar form!, using cdd and Kaibel’s algorithm)

- as stop conditions in the Hasse construction, use geometric/algebraic (Truffet’s algorithm) methods

- for all active sets of constraints construct the critical regions and associated affine control laws

- use the graph structure to efficiently store and retrieve the active region at runtime (the point location problem)
Face visibility idea
Visibility test

- Each face of $P([A - E, b])$ is either:
  - hidden by the other faces of the polytope, or
  - projects onto the sub-space $\overline{U}$ into a region which corresponds bijectively with a critical region of the explicit solution.

- A face is visible if no segment $[v_j, \overline{v}_j]$ intersects the feasible domain $P([A - E, b])$; here, $v_j$ is a vertex of $P$ and $\overline{v}_j$ is its projection on $\overline{U}$.

Face visibility test

Consider a face $F \subset P([A - E, b])$ given in both half-space $- F(A)$, and generator $- F(V)$, forms. Then, verifying:

$$\exists i \in A \text{ s.t. } [A - E], \overline{v}_j \leq b_i, \forall j \in V,$$

is a sufficient condition for the visibility of the face $F$ w.r.t. the subspace $\overline{U}$. 
Visibility test

Consider a face $F \subset P(A - E, b)$ given in both half-space $-F(A)$, and generator $-F(V)$, forms. Then, verifying:

$$\exists i \in A \text{ s.t. } [A - E]_i \bar{v}_j \leq b_i, \forall j \in V,$$

is a sufficient condition for the visibility of the face $F$ w.r.t. the subspace $\bar{U}$. 

Face visibility test

$$\exists i \in A \text{ s.t. } A_i \bar{v}_1 \leq b_i \rightarrow \text{hidden}$$

$$\nexists i \text{ s.t. } A_i \bar{v}_j \leq b_i, \forall j \in \{2, 3\} \rightarrow \text{visible}$$
Algebraic test for critical region emptiness

Rationale

- The process of selecting the candidate polyhedral sets from the face lattice continues after the visibility test.

- Thus, the remaining candidate sets have to be non-empty.

- We used an algebraic method to tackle this problem\(^6\).

Assumptions

- The matrix \( Z_A \) has no null row;

- that \( q > n \) and that \( Z_A \in \mathbb{R}^{q\times n} \) has full column rank, i.e., \( \text{rank}(Z_A) = n \).

---

Illustrative example

Approach

- compute the face lattice and export the Hasse graph using Polymake – a powerful tool “designed for the algorithmic treatment of polytopes and polyhedra”;
- apply successively the test procedures to retrieve the non-empty critical regions and their associated affine control laws.
- we validate our results by comparing with the Parametric Optimization toolbox (POP) – a MATLAB toolbox with efficient implementations of mp-QP problem solvers (problem 82 from POP dataset).

<table>
<thead>
<tr>
<th>POP problem</th>
<th>POP toolbox</th>
<th>Current approach</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Polymake</td>
</tr>
<tr>
<td>82</td>
<td>13.27</td>
<td>3.06</td>
</tr>
</tbody>
</table>

**Tools:** Polymake\(^7\), POP Toolbox\(^8\).


Outline

1 Motivation
2 Preliminaries
3 Geometrical interpretation of the explicit MPC representation
4 Explicit MPC toolbox
Mpqp-lattice toolbox

- We are working on a toolbox in C++ specialized in solving multi-parametric quadratic programs with linear constraints (stable release via QR)

- It uses Eigen 3 and libigl for linear algebra and the cddlib [Fukuda 2003] library for polyhedral computations (vertex/facet enumeration and nonemptyness check)

- Algebraic test (experimental) for nonemptyness check [Truffet 2020]

- Containerized application with Docker (can be compiled on different platforms)
A medium-sized example

- The state-space model

\[
x^+ = \begin{bmatrix} I_3 & 0.1I_3 \\ O_3 & I_3 \\ \end{bmatrix} x + \begin{bmatrix} 0.005I_3 \\ 0.1I_3 \\ \end{bmatrix} u
\]

- Constraints: \( |\begin{bmatrix} I_3 & O_3 \end{bmatrix} x| \leq [1.5 \ 1.5 \ 1.5]^\top \) and \( |u| \leq [0.8154 \ 0.8154 \ 3.2700]^\top \)

- Prediction horizon \( N = 3 \) and weights \( Q = \text{diag}(50I_3, 5I_3) \), \( R = 5I_3 \)

- We used the POP Toolbox [Oberdieck, Diangelakis, Papathanasiou, Nascu și Pistikopoulos 2016b] to generate the mpQP and to solve the problem for reference

Results

<table>
<thead>
<tr>
<th></th>
<th>Execution time (sec)</th>
<th>Number of nonempty regions</th>
</tr>
</thead>
<tbody>
<tr>
<td>POP Toolbox</td>
<td>3611</td>
<td>11838</td>
</tr>
<tr>
<td>mpqpp-lattice</td>
<td>1044</td>
<td>18016</td>
</tr>
</tbody>
</table>

- We generated more solutions than POP; all POP solutions can be found within ours
Comparison between POP/our toolbox (simple example)

**Execution time (seconds)**

**Cases**

**POP : \(N(\mu = 3.084, \sigma = 1.961)\)**

**Lattice: \(N(\mu = 0.472, \sigma = 0.274)\)**
References


