pyMPC Documentation

Marco Forgione

February 20, 2021

1 Mathematical formulation

1.1 Dynamical system

We consider a linear, discrete-time dynamical system with n_u inputs and n_x states:

$$x_{k+1} = Ax_k + Bu_k, \tag{1}$$

with $A \in \mathbb{R}^{n_x \times n_x}$ and $B \in \mathbb{R}^{n_x \times n_u}$.

1.2 MPC cost function

The Model Predictive Control (MPC) problem solved by pyMPC is:

$$\begin{aligned}
&\arg\min_{U} \overbrace{\frac{1}{2}\big(x_N - x_{\rm ref}\big)^\top Q_{x_N}\big(x_N - x_{\rm ref}\big)}^{=J_{Q_{x_N}}} + \overbrace{\frac{1}{2}\sum_{k=0}^{N_p-1}\big(x_k - x_{\rm ref}\big)^\top Q_x\big(x_k - x_{\rm ref}\big)}^{J_{Q_x}} + \\
&+ \overbrace{\frac{1}{2}\sum_{k=0}^{N_p-1}\big(u_k - u_{\rm ref}\big)^\top Q_u\big(u_k - u_{\rm ref}\big)}^{J_u} + \overbrace{\frac{1}{2}\sum_{k=0}^{N_p-1}\Delta u_k^\top Q_{\Delta u}\Delta u_k}^{J_{\Delta u}} && \text{(2a)}\\
&\text{subject to :}\\
&x_{k+1} = Ax_k + Bu_k && \text{(2b)}\\
&u_{\min} \leq u_k \leq u_{\max} && \text{(2c)}\\
&x_{\min} \leq x_k \leq x_{\max} && \text{(2d)}\\
&\Delta u_{\min} \leq \Delta u_k \leq \Delta u_{\max} && \text{(2e)}\\
&x_0 = \bar{x} && \text{(2f)}\\
&u_{-1} = \bar{u}, && \text{(2g)}
\end{aligned}$$

where $\Delta u_k = u_k - u_{k-1}$ and the optimization variables are the elements of the input sequence:

$$U = \left\{u_0,\ u_1,\ \ldots,\ u_{N_p-1}\right\}. \tag{3}$$

According to the solution strategy, the elements of the state sequence:

$$X = \left\{x_0,\ x_1,\ \ldots,\ x_{N_p}\right\} \tag{4}$$

may be either included as optimization variables (sparse implementation), or eliminated (dense implementation).

1.3 Notation

The column vector $\mathbf{u} \in \mathbb{R}^{N_p n_u \times 1}$ contains all the stacked entries of the input sequence U:

$$\mathbf{u}^\top = \begin{bmatrix} u_0^\top & u_1^\top & \ldots & u_{N_p-1}^\top \end{bmatrix}^\top. \tag{5}$$

Similarly, the column vector $\mathbf{x} \in \mathbb{R}^{N_p n_x \times 1}$ is defined as:

$$\mathbf{x}^\top = \begin{bmatrix} x_0^\top & x_1^\top & \ldots & x_{N_p}^\top \end{bmatrix}^\top, \tag{6}$$

the column vector $\mathbf{x}_{\mathrm{ref}} \in \mathbb{R}^{N_p n_x \times 1}$ is defined as:

$$\mathbf{x}_{\mathrm{ref}}^\top = \begin{bmatrix} x_{\mathrm{ref}}^\top & x_{\mathrm{ref}}^\top & \ldots & x_{\mathrm{ref}}^\top \end{bmatrix}^\top, \tag{7}$$

and finally the column vector $\mathbf{u}_{\mathrm{ref}}$ is defined as:

$$\mathbf{u}_{\mathrm{ref}}^\top = \begin{bmatrix} u_{\mathrm{ref}}^\top & u_{\mathrm{ref}}^\top & \ldots & u_{\mathrm{ref}}^\top \end{bmatrix}^\top. \tag{8}$$

Note that we consider a constant state reference x_{ref} over the prediction horizon for notation simplicity. The extension to a varying reference is straightforward (and actually implemented in pyMPC).

1.4 Receding horizon implementation

In a typical implementation, the MPC input is applied in *receding horizon*. At each time step i, the problem (2) is solved with $x_0 = x[i],\ u_{-1} = u[i-1]$ and an optimal input sequence $u_0, \ldots, u_{N_p}$ is obtained. The first element of this sequence u_0 is the control input that is actually applied at time instant i. At time instant $i+1$, a new state $x[i+1]$ is measured (or estimated), and the process is iterated.

Thus, formally, the MPC control law is a (static) function of the current state and the previous input:

$$u_{MPC} = K(x[i], u[i-1]). \tag{9}$$

Note that this function also depends on the references x_{ref} and u_{ref} and on the system matrices A and B.

2 Quadratic Programming Formulation

The OSQP Quadratic Programming (QP) solver expects a problem with form:

$$\min \frac{1}{2}\mathbf{z}^\top \mathbf{P}\mathbf{z} + \mathbf{q}^\top \mathbf{z} \tag{10a}$$

subject to

$$l \leq \mathbf{A}\mathbf{z} \leq u \tag{10b}$$

To implement the MPC controller using the OSQP solver, we need to re-write the MPC optimization problem (2) in form (10).

3 Sparse Implementation

In the sparse implementation, the dynamic constraints given by the linear dynamics equations (2b) are included explicitly and the state sequence X is considered as an optimization variable, along with the input sequence U.

3.1 Cost function

3.1.1 Terms in Q_x and Q_{x_N}

By direct inspection, the non-constant terms of the cost function in Q_x and Q_{x_N} are:

$$\begin{aligned} J_{Q_x} = \frac{1}{2}\begin{bmatrix} x_0^\top & x_1^\top & \dots & x_{N_p}^\top \end{bmatrix}^\top \overbrace{\text{blkdiag}(Q_x, Q_x, \dots, Q_{x_N})}^{=\mathcal{Q}_x} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N_p} \end{bmatrix} + \\ + \begin{bmatrix} -x_{\text{ref}}{}^\top Q_x & -x_{\text{ref}}{}^\top Q_x & \dots & -x_{\text{ref}}{}^\top Q_{x_N} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N_p} \end{bmatrix}^\top . \end{aligned} \tag{11}$$

More compactly, this is equivalent to:

$$J_{Q_x} = \frac{1}{2}\mathbf{x}^\top \mathcal{Q}_x \mathbf{x} - \mathbf{x}_{\text{ref}}^\top \mathcal{Q}_x \mathbf{x}. \tag{12}$$

3.1.2 Terms in Q_u

Similarly, for the term J_{Q_u}:

$$J_{Q_u} = \frac{1}{2}\begin{bmatrix}u_0^\top & u_1^\top & \dots & u_{N_p-1}^\top\end{bmatrix} \overbrace{\text{blkdiag}(Q_u, Q_u, \dots, Q_u)}^{\mathcal{Q}_u} \begin{bmatrix}u_0 \\ u_1 \\ \vdots \\ u_{N_p-1}\end{bmatrix} + \\ + \begin{bmatrix}-u_{\text{ref}}{}^\top Q_u & -u_{\text{ref}}{}^\top Q_u & \dots & -u_{\text{ref}}{}^\top Q_u\end{bmatrix} \begin{bmatrix}u_0 \\ u_1 \\ \vdots \\ u_{N_p-1}\end{bmatrix} \tag{13}$$

More compactly, this is equivalent to:

$$J_{Q_u} = \frac{1}{2}\mathbf{u}^\top \mathcal{Q}_u \mathbf{u} - \mathbf{u}_{\text{ref}}^\top \mathcal{Q}_u \mathbf{u}. \tag{14}$$

3.1.3 Terms in $Q_{\Delta u}$

As for the terms in $Q_{\Delta u}$ we have instead:

$$J_{\Delta u} = \frac{1}{2}\begin{bmatrix}u_0 & u_1 & \dots & u_{N_p-1}\end{bmatrix}^\top \begin{bmatrix} 2Q_{\Delta u} & -Q_{\Delta u} & 0 & \dots & \dots & 0 \\ -Q_{\Delta u} & 2Q_{\Delta u} & -Q_{\Delta u} & 0 & \dots & 0 \\ 0 & -Q_{\Delta u} & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & -Q_{\Delta u} & 2Q_{\Delta u} & -Q_{\Delta u} \\ 0 & 0 & 0 & 0 & -Q_{\Delta u} & Q_{\Delta u} \end{bmatrix} \begin{bmatrix}u_0 \\ u_1 \\ \vdots \\ u_{N_p-1}\end{bmatrix}^\top \\ - u_{-1}^\top Q_{\Delta u} u_0 \tag{15}$$

More compactly, this is equivalent to:

$$J_{\Delta u} = \frac{1}{2}\mathbf{u}^\top \mathcal{Q}_{\Delta u} \mathbf{u} - u_{-1}^\top Q_{\Delta u} u_0$$

It is convenient to write the above expression using stacked vectors only:

$$J_{\Delta u} = \frac{1}{2}\mathbf{u}^\top \mathcal{Q}_{\Delta u} \mathbf{u} + \begin{bmatrix}-u_{-1}{}^\top Q_{\Delta u} & 0 & \dots & 0\end{bmatrix}\mathbf{u} \tag{16}$$

3.2 Constraints

3.2.1 Linear dynamics

Let us consider the linear equality constraints (2b) representing the system dynamics. These can be written in matrix form as:

$$\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N_p-1} \\ x_{N_p} \end{bmatrix} = \overbrace{\begin{bmatrix} 0 & 0 & \dots & 0 \\ A_d & 0 & \dots & 0 \\ 0 & A_d & \dots & 0 \\ \vdots & 0 & \ddots & 0 \\ 0 & 0 & \dots & A_d \end{bmatrix}}^{=\mathcal{A}} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N_p-1} \\ x_{N_p} \end{bmatrix} + \overbrace{\begin{bmatrix} 0 & 0 & \dots & 0 \\ B_d & 0 & \dots & 0 \\ 0 & B_d & \dots & 0 \\ \vdots & 0 & \ddots & 0 \\ 0 & 0 & \dots & B_d \end{bmatrix}}^{=\mathcal{B}} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N_p-2} \\ u_{N_p-1} \end{bmatrix} + \overbrace{\begin{bmatrix} \bar{x} \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix}}^{=\mathcal{C}}. \tag{17}$$

Thus, we get a set of linear equality constraints representing the system dynamics (2b). These constraints can be written as

$$\begin{bmatrix} (\mathcal{A} - I) & \mathcal{B} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} = \mathcal{C}. \tag{18}$$

3.2.2 Variable bounds: x and u

The bounds on x and u are readily implemented as:

$$\begin{bmatrix} x_{\text{min}} \\ u_{\text{min}} \end{bmatrix} \leq \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} \leq \begin{bmatrix} x_{\text{max}} \\ u_{\text{max}} \end{bmatrix}. \tag{19}$$

3.2.3 Variable bounds: Δu

$$\begin{bmatrix} u_{-1} + \Delta u_{\text{min}} \\ \Delta u_{\text{min}} \\ \vdots \\ \Delta u_{\text{min}} \end{bmatrix} \leq \begin{bmatrix} I & 0 & \dots & \dots & 0 & 0 \\ -I & I & 0 & \dots & 0 & 0 \\ 0 & -I & I & \dots & 0 & 0 \\ \vdots & & & & & \\ 0 & 0 & \dots & 0 & -I & I \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N_c-1} \end{bmatrix} \leq \begin{bmatrix} u_{-1} + \Delta u_{\text{max}} \\ \Delta u_{\text{max}} \\ \vdots \\ \Delta u_{\text{max}} \end{bmatrix} \tag{20}$$

3.3 Soft constraints

Bounds on x may result in an problem unfeasible! A common solution is to transform the hard constraints in x into soft constraints by means of *slack* *variables* $\boldsymbol{\epsilon}$. In the current implementation, there are as many slack variables as state variables, i.e. $\boldsymbol{\epsilon} \in \mathbb{R}^{N_p n_x \times 1}$. We use the constraint:

$$\begin{bmatrix} x_{\min} \\ u_{\min} \end{bmatrix} \leq \begin{bmatrix} I & 0 & I \\ 0 & I & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \\ \boldsymbol{\epsilon} \end{bmatrix} \leq \begin{bmatrix} x_{\max} \\ u_{\max} \end{bmatrix}. \tag{21}$$

In order to penalize constraint violation, we have a penalty term in the cost function:

$$J_\epsilon = \frac{1}{2} \boldsymbol{\epsilon}^\top \mathcal{Q}_\epsilon \boldsymbol{\epsilon}, \tag{22}$$

where

$$\mathcal{Q}_\epsilon = \text{blkdiag}(Q_\epsilon, Q_\epsilon, \dots Q_\epsilon) \tag{23}$$

and Q_ϵ is σI_{n_x}, with σ a "large" constant (e.g. $\sigma = 10^4$).

3.4 Control Horizon

Sometimes, we may want to use a control horizon $N_c < N_p$ instead of the standard $N_c = N_p$. In this case, the input considered is constant for $N_c \geq N_p$.

$$\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N_p-1} \\ x_{N_p} \end{bmatrix} = \overbrace{\begin{bmatrix} 0 & 0 & \dots & 0 \\ A_d & 0 & \dots & 0 \\ 0 & A_d & \dots & 0 \\ \vdots & 0 & \ddots & 0 \\ 0 & 0 & \dots & A_d \end{bmatrix}}^{=\mathcal{A}} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N_p-1} \\ x_{N_p} \end{bmatrix} + \overbrace{\begin{bmatrix} 0 & 0 & \dots & 0 \\ B_d & 0 & \dots & 0 \\ 0 & B_d & \dots & 0 \\ \vdots & 0 & \ddots & 0 \\ 0 & 0 & \dots & B_d \\ 0 & 0 & \dots & \vdots \\ 0 & 0 & \dots & B_d \end{bmatrix}}^{=\mathbb{B}} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{N_c-1} \\ \vdots \\ u_{N_c-1} \end{bmatrix} + \overbrace{\begin{bmatrix} \bar{x} \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix}}^{=\mathcal{C}} \tag{24}$$

The contributions J_{Q_u} of the cost function also changes:

$$J_{Q_u} = \frac{1}{2}\begin{bmatrix} u_0^\top & u_1^\top & \dots & u_{N_p-1}^\top \end{bmatrix} \text{blkdiag}\Big(Q_u, Q_u, \dots, (N_p{-}N_c{+}1)Q_u\Big) \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N_p-1} \end{bmatrix} +$$

$$+ \begin{bmatrix} -u_{\text{ref}}{}^\top Q_u & -u_{\text{ref}}{}^\top Q_u & \dots & -(N_p - N_c + 1)u_{\text{ref}}{}^\top Q_u \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N_p-1} \end{bmatrix} \tag{25}$$

Instead, $J_\Delta u$ does not change (because the input is constant for $k \geq N_c$!

4 Dense Implementation

In the dense implementation, the elements of the input sequence U are the only optimization variables. The state variables are eliminated using the so-called Lagrange equations:

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N_p-1} \\ x_{N_p} \end{bmatrix} = \overbrace{\begin{bmatrix} A_d \\ A_d^2 \\ \vdots \\ A_d^{N_p-1} \\ A_d^{N_p} \end{bmatrix}}^{=\mathcal{A}} x_0 + \overbrace{\begin{bmatrix} B_d & 0 & 0 & \dots & 0 & 0 \\ A_d B_d & B_d & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & 0 & 0 \\ A_d^{N_p-2} B_d & A_d^{N_p-3} B_d & A_d^{N_p-4} B_d & \dots & B_d & 0 \\ A_d^{N_p-1} B_d & A_d^{N_p-2} B_d & A_d^{N_p-3} B_d & \dots & A_d B_d & B_d \end{bmatrix}}^{=\mathcal{B}} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{N_p-1} \end{bmatrix}. \tag{26}$$

In vector notation, this is simply:

$$\mathbf{x} = \mathcal{A}x_0 + \mathcal{B}\mathbf{u} \tag{27}$$

The elements of the cost function may be written compactly as:

$$J_{Q_x} = \frac{1}{2}\left(\mathcal{A}x_0 + \mathcal{B}\mathbf{u} - \mathbf{x}_{\text{ref}}\right)^\top \mathcal{Q}_x \left(\mathcal{A}x_0 + \mathcal{B}\mathbf{u} - \mathbf{x}_{\text{ref}}\right) \tag{28}$$

$$J_{Q_u} = \frac{1}{2}(\mathbf{u} - \mathbf{u}_{\text{ref}})^\top \mathcal{Q}_u (\mathbf{u} - \mathbf{u}_{\text{ref}}) \tag{29}$$

$$J_{Q_{\Delta u}} = \frac{1}{2}\mathbf{u}^\top \mathcal{Q}_{\Delta u}\mathbf{u} + \begin{bmatrix} -u_{-1}{}^\top Q_{\Delta u} & 0 & \dots & 0 \end{bmatrix} \mathbf{u} \tag{30}$$

Summing up and expanding terms:

$$J = \frac{1}{2}\mathbf{u}^\top \mathcal{B}^\top \mathcal{Q}_x \mathcal{B}\mathbf{u} + \frac{1}{2}(\mathcal{A}x_0 - \mathbf{x}_{\mathrm{ref}})^\top \mathcal{Q}_x(\mathcal{A}x_0 - \mathbf{x}_{\mathrm{ref}}) + (\mathcal{A}x_0 - \mathbf{x}_{\mathrm{ref}})^\top \mathcal{Q}_x \mathcal{B}\mathbf{u} + \\ + \frac{1}{2}\mathbf{u}^\top \mathcal{Q}_u \mathbf{u} + \frac{1}{2}\mathbf{u}_{\mathrm{ref}}{}^\top \mathcal{Q}_u \mathbf{u}_{\mathrm{ref}} + -\mathbf{u}_{\mathrm{ref}}{}^\top \mathcal{Q}_u \mathbf{u} + \\ + \frac{1}{2}\mathbf{u}^\top \mathcal{Q}_{\Delta u}\mathbf{u} + \begin{bmatrix} -u_{-1}{}^\top Q_{\Delta u} & 0 & \dots & 0 \end{bmatrix} \mathbf{u} \tag{31}$$

Neglecting constant terms and collecting:

$$J = C + \frac{1}{2}\mathbf{u}^\top \overbrace{\left(\mathcal{B}^\top \mathcal{Q}_x \mathcal{B} + \mathcal{Q}_u + \mathcal{Q}_{\Delta u}\right)}^{=\mathbf{P}} \mathbf{u} + \\ \overbrace{\left[(\mathcal{A}x_o - \mathbf{x}_{\mathrm{ref}})^\top \mathcal{Q}_x \mathcal{B} - \mathbf{u}_{\mathrm{ref}}{}^\top \mathcal{Q}_u - \begin{bmatrix} u_{-1}{}^\top Q_{\Delta u} & 0 & \dots & 0 \end{bmatrix}\right)}^{=\mathbf{q}^\top} \mathbf{u} \tag{32}$$

Thus, we have

$$\mathbf{q} = \mathcal{B}^\top \mathcal{Q}_x(\mathcal{A}x_0 - \mathbf{x}_{\mathrm{ref}}) - \mathcal{Q}_u \mathbf{u}_{\mathrm{ref}} + \begin{bmatrix} -Q_{\Delta u} u_{-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{33}$$

expanding

$$\mathbf{q} = \overbrace{\mathcal{B}^\top \mathcal{Q}_x \mathcal{A}}^{=p_{x_0}} x_0 \; + \; \overbrace{-\mathcal{B}^\top \mathcal{Q}_x}^{p_{\mathbf{x}_{\mathrm{ref}}}} \mathbf{x}_{\mathrm{ref}} \; + \; \overbrace{-\mathcal{Q}_u}^{=p_{\mathbf{u}_{\mathrm{ref}}}} \mathbf{u}_{\mathrm{ref}} + \overbrace{\begin{bmatrix} -Q_{\Delta u} \\ 0 \\ \vdots \\ 0 \end{bmatrix}}^{=p_{u_{-1}}} u_{-1} \tag{34}$$

4.1 Unconstrained case

For an unconstrained problem, the optimal value of $\mathbf{u}$ is:

$$\mathbf{u}^{\mathrm{opt}} = -\mathbf{P}^{-1}\mathbf{q} \tag{35}$$

Expanding, we obtain:

$$\mathbf{u}^{\mathrm{opt}} = k_{x_0}x_0 + k_{\mathbf{x}_{\mathrm{ref}}}\mathbf{x}_{\mathrm{ref}} + k_{\mathbf{u}_{\mathrm{ref}}}\mathbf{u}_{\mathrm{ref}}k_{u_{-1}}u_{-1} \tag{36}$$

with:

$$k_{x_0} = -\mathbf{P}^{-1}p_{x_0} \tag{37a}$$
$$k_{\mathbf{x}_{\mathrm{ref}}} = -\mathbf{P}^{-1}p_{\mathbf{x}_{\mathrm{ref}}} \tag{37b}$$
$$k_{\mathbf{u}_{\mathrm{ref}}} = -\mathbf{P}^{-1}p_{\mathbf{u}_{\mathrm{ref}}} \tag{37c}$$
$$k_{u_{-1}} = -\mathbf{P}^{-1}p_{u_{-1}}. \tag{37d}$$

Thus, the control law is a simple linear function of x_0, $\mathbf{x}_{\mathrm{ref}}$, $\mathbf{u}_{\mathrm{ref}}$, and u_{-1}.