Formulation of the MPC problem as a QP problem

PBPKSim Application Note

Pantelis Sopasakis

June 12, 2014

Scope

In this short application note we present how one can formulate the offset-free MPC control problem as a quadratic optimisation problem which can be solved with any solver in MATLAB, Python or other languages.

Formulation of the problem

Notation

Let $\mathbb{N}$, $\mathbb{R}$, $\mathbb{R}^n$, $\mathbb{R}^{n\times m}$ denote the set of non-negative integers, the set of real number, the set of n-vectors, and the set of n-by-m matrices. Let $\mathbb{N}_{[k_1,k_2]}$ denote the set of non-negative integers that are between k_1 and k_2. We denote by I_m the unit matrix of size m-by-m; the simpler notation I will sometimes be used without explicitly stating the dimension of the matrix.

General formulation

The cost function is defined as:

$$V_N(\pi, \hat{c}_j, \hat{d}_j) = \|c_N - \bar{c}_j\|_P^2 + \sum_{k=0}^{N-1} \left(\|c_k - \bar{c}_j\|_Q^2 + \|u_k - \bar{u}_j\|_R^2 \right), \tag{1}$$

where $\|\cdot\|_Q$ is the Q-(semi)norm defined as $\|c\|_Q^2 = c'Qc$ for a positive (semi)definite matrix Q, $c_k \in \mathbb{R}^n$ is the (predicted) state of the system, $u_k \in \mathbb{R}^m$ is a (predicted) input, $\bar{c}_j \in \mathbb{R}^n$ and $\bar{u}_j \in \mathbb{R}^m$ are reference values computed as explained in [1, Eq. (33)] and P is a properly calculated terminal cost matrix [1, Eq. (32)]. In (1), π stands for a sequence of inputs $\pi = \{u_i\}_{i=0}^{N-1}$ of length N, $Q \in \mathbb{R}^{n\times n}$ is a positive semidefinite matrix, $R \in \mathbb{R}^{m\times m}$ is a positive definite matrix and $P \in \mathbb{R}^{n\times n}$.

We assume that a constant additive disturbance $d_k \in \mathbb{R}^{n_d}$ acts on the system, where n_d is less than of equal to the number of measured outputs of the system. The system dynamics is described by:

$$V_N^\star(\hat{c}_j, \hat{d}_j) = \min_{\pi=\{u_i\}_{i=0}^{N-1}} V_N(\pi, \hat{c}_j, \hat{d}_j), \tag{2}$$

subject to the constraints:

$$
\begin{aligned}
c_{k+1} =& Ac_k + Bu_k + B_d d_k, \forall k \in \mathbb{N}_{[0,N-1]}, && \text{(3a)}\\
d_{k+1} =& d_k, \forall k \in \mathbb{N}_{[0,N-1]}, && \text{(3b)}\\
Gc_k + Hu_k \leq& M, \forall k \in \mathbb{N}_{[0,N-1]}, && \text{(3c)}\\
G_f c_N \leq& M_f, && \text{(3d)}\\
c_0 =& \hat{c}_j, && \text{(3e)}\\
d_0 =& \hat{d}_j, && \text{(3f)}
\end{aligned}
$$

but where the mixed state-input constraints of (3c) and the terminal constraints (3d) can sometimes be simplified and written in the form:

$$
\begin{aligned}
0 \leq& c_k \leq c_{\max}, \forall k \in \mathbb{N}_{[1,N]}, && \text{(3g)}\\
0 \leq& u_k \leq u_{\max}, \forall k \in \mathbb{N}_{[0,N-1]}. && \text{(3h)}
\end{aligned}
$$

In the above equations $G \in \mathbb{R}^{n_c \times n}$, $H \in \mathbb{R}^{n_c \times m}$ and $M \in \mathbb{R}^{n_c}$, where $n_c \in \mathbb{N}$ is the number of mixed state-input constraints. Also, $G_f \in \mathbb{R}^{n_f \times n}$ and $M_f \in \mathbb{R}^{n_f}$, where $n_f \in \mathbb{N}$ is the number of terminal constraints. What is more, it is $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ and $B_d \in \mathbb{R}^{n \times n_d}$.

Let us define the optimisation variable z as follows:

$$
z = (u_0, c_1, u_1, c_2, \ldots, c_{N-1}, u_{N-1}, c_N)', \tag{4}
$$

where $z \in \mathbb{R}^{N(n+m)}$. Then, the cost function in (1) can be written as follows:

$$
\begin{aligned}
V_N =& (c_N - \bar{c}_j)' P (c_N - \bar{c}_j) + \sum_{k=0}^{N-1} (c_k - \bar{c}_j)' Q (c_k - \bar{c}_j) + (u_k - \bar{u}_j)' R (u_k - \bar{u}_j)\\
=& c_N' P c_N + \bar{c}_j' P \bar{c}_j - \bar{c}_j' P c_N - c_N' P \bar{c}_j +\\
& + \sum_{k=0}^{N-1} c_k' Q c_k + \bar{c}_j' Q c_k + u_k' R u_k + \bar{u}_j' R \bar{u}_j - 2\bar{u}_j R u_k\\
=& c_N' P c_N + \sum_{k=0}^{N-1} \begin{bmatrix} c_k' & u_k' \end{bmatrix} \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} \begin{bmatrix} c_k \\ u_k \end{bmatrix} +\\
& + \left(-2\bar{c}_j' P c_N + \sum_{k=0}^{N-1} \begin{bmatrix} -2\bar{c}_j Q & -2\bar{u}_j R \end{bmatrix} \begin{bmatrix} c_k \\ u_k \end{bmatrix} \right) + \gamma,
\end{aligned}
$$

where the terms in red are quadratic terms, the blue ones are linear and there is also a constant term γ which we shall omit.

The cost function V_N can be now written as a function of z in the following form:

$$V_N(z) = \frac{1}{2} z' \Theta z + q' z, \tag{5}$$

where $\Theta \in \mathbb{R}^{N(n+m)\times N(n+m)}$ is the following square positive definite matrix:

$$\Theta = 2 \begin{bmatrix} R & & & & & & & \\ & Q & & & & & & \\ & & R & & & & & \\ & & & Q & & & & \\ & & & & \ddots & & & \\ & & & & & Q & & \\ & & & & & & R & \\ & & & & & & & P \end{bmatrix}, \tag{6}$$

and $q \in \mathbb{R}^{N(n+m)}$ is:

$$q = -2 \begin{bmatrix} \bar{u}_j' R \\ \bar{c}_j' Q \\ \bar{u}_j' R \\ \bar{c}_j' Q \\ \vdots \\ \bar{u}_j' Q \\ \bar{c}_j' R \\ \bar{c}_j' P \end{bmatrix}. \tag{7}$$

If we only have constraints of the simple form (3g) and (3h), then, the constraints on z are also bounds of the form $0 \leq z \leq z_{\max}$, where $z_{\max} \in \mathbb{R}^{N(n+m)}$ is the vector:

$$z_{\max} = \begin{bmatrix} u_{\max} \\ c_{\max} \\ \vdots \\ u_{\max} \\ c_{\max} \end{bmatrix}. \tag{8}$$

Given that the disturbance dynamics is (3b), the state-update equation (3a) can be written as follows:

$$\begin{aligned} c_1 - Bu_0 &= B_d d_0 + Ac_0, \\ c_2 - Ac_1 - Bu_1 &= B_d d_0, \\ c_3 - Ac_2 - Bu_2 &= B_d d_0, \\ &\vdots \\ c_N - Ac_{N-1} - Bu_{N-1} &= B_d d_0, \end{aligned}$$

which is compactly written in the matrix form $Kz = L$ where $K \in \mathbb{R}^{Nn \times N(n+m)}$ is:

$$K = \begin{bmatrix} -B & I & & & & & \\ & -A & -B & I & & & \\ & & -A & -B & I & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & -A & -B & I \end{bmatrix}, \tag{9}$$

and $L \in \mathbb{R}^{Nn}$ is

$$L = \begin{bmatrix} B_d d_0 + Ac_0 \\ B_d d_0 \\ B_d d_0 \\ \vdots \\ B_d d_0 \end{bmatrix}, \tag{10}$$

where, of course $d_0 = \hat{d}_j$ and $c_0 = \hat{c}_j$, that is d_0 is the current disturbance estimate and c_0 is the current concentration (state) estimate.

Now, overall the problem has been formulated as a convex QP as follows:

$$V_N^\star = \min_z \frac{1}{2} z' \Theta z + q' z, \tag{11a}$$

subject to the constraints:

$$0 \leq z \leq z_{\max} \tag{11b}$$

$$Kz = L. \tag{11c}$$

Mixed state-input constraints

There are cases where we need to impose mixed state-input constraints as in (3c) and/or terminal constraints as in (3d). These constraints may be imposed additionally to the bounds (11b). Constraints of the form (3c) can be represented, in terms of z, in the form:

$$\Phi z \leq \phi, \tag{12}$$

where $\Phi \in \mathbb{R}^{Nn_c \times N(n+m)}$ is

$$\Phi = \begin{bmatrix} H & 0 & & & & \\ & G & H & & & \\ & & G & H & & \\ & & & \ddots & \ddots & \\ & & & & G & H \end{bmatrix}, \tag{13}$$

and $\phi \in \mathbb{R}^{Nn_c}$ is

$$\phi = \begin{bmatrix} M - Gc_0 \\ M \\ \vdots \\ M \end{bmatrix}. \tag{14}$$

Partial constraints

Constraints may not be necessarily imposed on all concentrations but only on some c^j for $j \in J$, where c^j stands for the j-th element of the vector $c \in \mathbb{R}^n$. Let $J = \{j_1, j_2, \ldots, j_l\}$ with $l \leq n$. Let us define the matrix $\Delta_J \in \mathbb{R}^{l \times n}$ as

$$\Delta_J = \begin{bmatrix} e(j_1)' \\ e(j_2)' \\ \vdots \\ e(j_k)' \end{bmatrix}, \tag{15}$$

where $e(j)$ is a vector of $\mathbb{R}^n$ whose j-th entry is 1 and all other entries are equal to 0. The constraints on c are then written as:

$$\Delta_J c_k \leq c_{\max}^J, \forall k \in \mathbb{N}_{[0,N]}, \tag{16a}$$

$$0 \leq c_k, \forall k \in \mathbb{N}_{[0,N]}, \tag{16b}$$

where $c^J_{\max} \in \mathbb{R}^l$ stands for the bounds on c^j with $j \in J$. In terms of the variable z, the state and input bounds (3g) and (3h) can be written as $\Xi_J z \leq \xi_J$, where

$$\Xi_J = \begin{bmatrix} I_m & & & & \\ & \Delta_J & & & \\ & & \ddots & & \\ & & & I_m & \\ & & & & \Delta_J \end{bmatrix}, \tag{17a}$$

and $\xi_J \in \mathbb{R}^{N(m+l)}$ is

$$\xi_J = \begin{bmatrix} u_{\max} \\ c^J_{\max} \\ u_{\max} \\ c^J_{\max} \\ \vdots \\ u_{\max} \\ c^J_{\max} \end{bmatrix}. \tag{17b}$$

Other details

The vectors $\bar{c}_j$ and $\bar{u}_j$ are computed online by the following equation:

$$\begin{bmatrix} A - I & B \\ C & 0 \end{bmatrix} \begin{bmatrix} \bar{c}_j \\ \bar{u}_j \end{bmatrix} = \begin{bmatrix} -B_d \hat{d}_j \\ r_j - C_d \hat{d}_j \end{bmatrix}, \tag{18}$$

where $\hat{d}_j$ is the disturbance estimate (which comes from the augmented state observer [1, Eq. (24)]), and r_j is the desired set-point. Hence, a linear system has to be solved at every time instant.

Matrix P in the definition of V_N is a matrix which solves the following Ricatti-type algebraic equation:

$$P = A'PA - A'PB(B'PB + R)^{-1}B'PA + Q. \tag{19}$$

The solution of (19) is unique and can be found through the iterative procedure:

$$P_{k+1} = A'P_kA - A'P_kB(B'P_kB + R)^{-1}B'P_kA + Q, \tag{20}$$

with $P_0 = I$. Algorithm (20) terminates if the norm of two successive iterates $\|P_\nu - P_{\nu-1}\|$ is smaller than a given desired (small) tolerance.

Bibliography

[1] P. Sopasakis, P. Patrinos, and H. Sarimveis, "Model predictive control for optimal continuous drug administration," *Computer Methods and Programs in Biomedicine*, 2014. Accepted for publication on 29 May 2014.