1. Brief review on the basic concepts of Linear Matrix Inequalities

A Linear Matrix Inequality (LMI) is an inequality of the following form

\[ F(x) = F_0 + \sum_{i=1}^{l} x_i F_i > 0 \]  

(1)

where

\[ x_i, i = 1, 2, \ldots, l, \] are the variables of the problems.

\[ F_i = F_i^T \in \mathbb{R}^{n \times n} \] are known matrices

\[ F(x) > 0 \] means that \( F(x) \) is positive definite for any value of \( x \)

Several LMIs can be grouped into a single LMI, as follows:

\[ F_1(x) > 0, F_2(x) > 0, \ldots, F_n(x) > 0 \Rightarrow \text{diag} \left( F_1(x) > 0, \ldots, F_n(x) > 0 \right) \]

Convex quadratic inequalities can be converted into LMIs with the help of the Schur’s complement. Let \( Q(x) = Q(x)^T, R(x) = R(x)^T \) and \( S(x) \) be linear dependent on \( x \), then, the LMI:

\[
\begin{bmatrix}
Q(x) & S(x) \\
S(x)^T & R(x)
\end{bmatrix} > 0
\]  

(2)

is equivalent to

\[ R(x) > 0, \quad Q(x) - S(x) R(x)^{-1} S(x)^T > 0 \]  

(3)

or

\[ Q(x) > 0, \quad R(x) - S(x)^T Q(x)^{-1} S(x) > 0 \]  

(4)

Proof: Consider the auxiliary matrix

\[
T = \begin{bmatrix}
I & 0 \\
S^T Q^{-1} & I
\end{bmatrix}
\]

Observe that \( T > 0 \) because all the eigenvalues are equal to 1. Then, it is easy to show that
Thus, inequality (4) is equivalent to inequality (2).

Analogously, using

\[ T = \begin{bmatrix} I & 0 \\ SR^{-1} & I \end{bmatrix} \]

we can prove the equivalence between (3) and (2). □

2. Nominal MPC with LMIs

Suppose that the system model is represented in the state space form:

\[ x(k+1) = A x(k) + B \Delta u(k) \quad (5) \]
\[ y(k) = C x(k) \quad (6) \]

where \( x \in \mathbb{R}^{nx} \) is the state of the predicting model, \( k \) is the present time step, \( u \in \mathbb{R}^{nu} \) is the manipulated input, \( \Delta u(k) = u(k) - u(k-1) \) is the input increment, \( y \in \mathbb{R}^{ny} \) is the controlled output. \( A, B, C \) are matrices of appropriate dimensions.

The objective function of the MPC controller is defined as follows:

\[
J_k = \sum_{j=0}^{np} e^T(k+j)Qe(k+j) + \sum_{j=0}^{m-1} \Delta u^T(k+j)R \Delta u(k+j)
\]

(7)

where \( e(k+j) = y^{sp} - y(k+j) \), \( np \) is the output optimization horizon, \( m \) is the control horizon, \( y(k+j) \) is the predicted output at sampling time \( k+j \) taking into account the future control actions, \( Q \) and \( R \) are positive definite weighting matrices. Associated to the output setpoint \( y^{sp} \), we can obtain a state setpoint \( v \) and the output error can also be represented as

\[ e(k+j) = C(v - x(k+j)) = Cx^e(k+j) . \]
With model equations (1) and (2), the objective function defined in (7) can be represented as follows

\[ J_k = (\bar{A} x^e(k) - \bar{B} \Delta u)^T \bar{Q}_{np}(\bar{A} x^e(k) - \bar{B} \Delta u) + \Delta u^T R_m \Delta u \]  

where

\[
\bar{A} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{np} \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ CB & 0 & 0 & \cdots & 0 \\ CAB & CB & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{np-1}B & CA^{np-2}B & \cdots & CA^{np-m}B \end{bmatrix}
\]

\[
\Delta u = \begin{bmatrix} \Delta u(k)^T \\ \vdots \\ \Delta u(k+m-1)^T \end{bmatrix}^T
\]

\[
Q_{np} = \text{diag}(Q_{np}) \quad \text{and} \quad R_m = \text{diag}(R_m)
\]

Thus, the optimization problem of the nominal MPC can be written in the following form:

\[
\min_{\gamma, \Delta u} \gamma \\
\text{subject to}
\]

\[
\gamma - x^e(k)^T \bar{A}^T \bar{Q}_{np} \bar{A} x^e(k) + x^e(k)^T \bar{A}^T \bar{Q}_{np} \bar{B} \Delta u + \Delta u^T \bar{B}^T \bar{Q}_{np} \bar{A} x^e(k) - \\
- \Delta u^T (\bar{B}^T \bar{Q}_{np} \bar{B} + R_m) \Delta u > 0
\]

\[-\Delta u_{\text{max}} \leq \Delta u(k + j) \leq \Delta u_{\text{max}}, \quad j = 0,1,\ldots, m-1
\]

\[u_{\text{min}} \leq u(k + j) \leq u_{\text{max}}, \quad j = 0,1,\ldots, m-1
\]

Applying the Schur’s complement to (9), this inequality can be written in the form of a LMI. Consequently, the nominal MPC optimization problem can be written as follows:

\[
\min_{\gamma, \Delta u} \gamma \\
\text{subject to}
\]
\[
\begin{bmatrix}
(B^T Q_{np} B + R_m)^{-1} \\
\Delta u^T
\end{bmatrix}
\begin{bmatrix}
\Delta u \\
\gamma - x^e(k)^T A^T Q_{np} A x^e(k) + x^e(k) A^T Q_{np} B \Delta u \\
+ \Delta u B^T Q_{np} A x^e(k)
\end{bmatrix} > 0
\] (11)

\[-\Delta u_{max} \leq \Delta u(k+j) \leq \Delta u_{max}, \quad j = 0,1,..., m-1\]

\[u_{min} \leq u(k+j) \leq u_{max}, \quad j = 0,1,..., m-1\]

3. Multimodel MPC (MMPC)

The controller of the previous section can be extended to the case in which the true model of the plant is not exactly known, but it is known to belong to a set constituted by \(L\) different models. To present the method, we assume that the true plant can be represented by the model defined in (1) and (2) and has model parameters designated as \(\theta_T = (A_T, B_T)\). This means that

\[\theta_T \in \Omega \equiv \{\theta_1, \theta_2, \cdots, \theta_L\}\] (12)

This kind of uncertainty is usually designated as the multiple-plant uncertainty where each plant may represent the real process at a particular operating condition. With this notation, for instance, plant \(i\) corresponds to \(\theta_i = (A_i, B_i)\) and the extended MPC control problem can be formulated as follows:

\[
\max_{\theta_i} \min_{\Delta u} J_{k, \theta_i}
\]

\[
J_{k, \theta_i} = (\bar{A}_i x^e(k) - \bar{B}_i \Delta u)^T Q_{np} (\bar{A}_i x^e(k) - \bar{B}_i \Delta u) + \Delta u^T R_m \Delta u, \quad i = 1, 2, ..., L
\]

\[-\Delta u_{max} \leq \Delta u(k+j) \leq \Delta u_{max}, \quad j = 0,1,..., m-1\]

\[u_{min} \leq u(k+j) \leq u_{max}, \quad j = 0,1,..., m-1\]

Based on the structure of the nominal controller optimization problem defined in (10), the optimization problem of the multi-model controller defined in (13) can be reformulated as follows:
\[
\begin{align*}
\min_{\gamma, \Delta u} & \quad \gamma \\
\text{subject to} & \\
\left( \bar{B}_i^T Q_{np} \bar{B}_i + R_m \right)^{-1} \Delta u & > 0 \\
\Delta u^T & \\
\begin{bmatrix}
\gamma - x^e(k)^T \bar{A}_i^T Q_{np} \bar{A}_i x^e(k) + x^e(k)^T \bar{A}_i^T Q_{np} \bar{B}_i \Delta u \\
+ \Delta u^T \bar{B}_i^T Q_{np} \bar{A}_i x^e(k)
\end{bmatrix}
\end{align*}
\]  
\[i = 1, 2, \ldots, L \] 

\[-\Delta u_{\text{max}} \leq \Delta u(k + j) \leq \Delta u_{\text{max}}, \quad j = 0, 1, \ldots, m - 1 \]

\[u_{\text{min}} \leq u(k + j) \leq u_{\text{max}}, \quad j = 0, 1, \ldots, m - 1 \]

4. Example: The debutanizer column

To illustrate the application of the Multi-Model Predictive Control consider the debutanizer column by Rodrigues and Odloak (2002, 2003). We consider that, depending on the operating conditions, the system may be represented by one of the two following models:

\[
G_A(s) = \begin{bmatrix}
-0.2623 & 0.1368 \\
60s^2 + 59.2s + 1 & 1164s^2 + 99.7s + 1 \\
0.1242 & -0.1351 \\
218.7s^2 + 16.2s + 1 & 70s^2 + 20s + 1
\end{bmatrix}
\]

\[
G_B(s) = \begin{bmatrix}
-0.3544 & 0.2044 \\
218.6s^2 + 50.1s + 1 & 1150s^2 + 93.86s + 1 \\
0.0685 & -0.1256 \\
100.2s^2 + 11.32s + 1 & 20s^2 + 15s + 1
\end{bmatrix}
\]

Suppose the case in which the predicting model and the true plant are represented by model \(A\) and the tuning parameters of the controller are the following: \(np=30; m=2; q=[.2 .2]\) and \(r=[1. 1.]\). In this case, the controller does not consider the possible existence of model \(B\) and the system responses are represented in figure 1. This should correspond to the best possible performance of the system.
Consider now the case where we have the same scenario as in the previous case, but now control law is computed taking into account the possibility that the true plant may be $A$ or $B$. The closed loop responses are represented in figure 2. We note that the control actions become more conservative and the response for output $y_2$ is much slower than in the previous case.

Finally, consider the case where the predicting model is $A$ and the true plant is $B$ but the control law is computed considering both models. The closed loop responses are represented in figure 3. It is clear that the system responses are even slower than in the previous case, but the system is stable and performance is acceptable. It is not shown here but the closed loop system is unstable when the predicting model is $A$ and the plant is $B$, but $B$ is not considered in the computation of the control law.
Figure 1. MPC with model $A$ only and plant represented by model $A$. 
Figure 2. MPC with models $A$ and $B$ and plant represented by model $A$. 
Figure 3. MPC with models A and B and plant represented by model B.

References:


1. Using the Matlab program *MMPC.m* and the debutanizer column presented in the classnotes, try to tune a MPC controller that is based on model A only, and that would stabilize a plant represented by model B. Compare the performance of the new controller with the performance of MMPC with its best tuning parameters.

2. Extend the Matlab program *MMPC.m* to include a third possible model for the debutanizer column. This model has the following transfer function:

\[
G_c(s) = \begin{bmatrix}
-0.0279 & 0.005 \\
59.77s^2 + 99.6s + 1 & 499.8s^2 + 73.8s + 1 \\
0.195 & -0.1351 \\
220.1s^2 + 18.9s + 1 & 29.7s^2 + 20.7s + 1
\end{bmatrix}
\]

Simulate the closed loop system with the MMPC that includes models A, B and C, for cases where the true plant is represented by models A or B or C. Assume the same set of tuning parameters for the 3 cases. Verify if the controller is able to satisfy the constraints on \( \Delta u_{\text{max}} \) and on \( (u_{\text{min}}, u_{\text{max}}) \).
OPOM model for time-delayed systems

Consider a system with \( ny \) controlled outputs and \( nu \) manipulated inputs, let \( \theta_{i,j} \) be the time delay between input \( u_j \) and output \( y_i \) and define \( p > \max \left( \frac{\theta_{i,j}}{T} \right) \) where \( T \) is the sampling period. To build the OPOM extended to delayed time systems (González and Odloak, 2011; Odloak, 2004), assume that the Laplace transfer function relating input \( u_j \) and output \( y_i \) is given by:

\[
G_{i,j}(s) = \frac{B_{i,j}(s)e^{-\theta_{i,j}T}}{A_{i,j}(s)} \tag{0-1}
\]

where

\[
B_{i,j}(s) = b_{i,j,0} + b_{i,j,1}s + b_{i,j,2}s^2 + \ldots + b_{i,j,nb}s^{nb}
\]

\[
A_{i,j}(s) = 1 + a_{i,j,1}s + a_{i,j,2}s^2 + \ldots + a_{i,j,na}s^{na}
\]

The step response of the system defined in eq. (0-1) can be represented as follows:

\[
S_{i,j}(k) = d_{i,j}^0 + \sum_{l=1}^{nu} d_{i,j,l}^d e^{\theta_{i,j}(kT-\theta_{i,j})} \tag{0-2}
\]

where \( d_{i,j}^0, d_{i,j,1}^d, \ldots, d_{i,j,na}^d \) are obtained by partial fractions expansion of the model defined in eq. (0-1) and \( r_{i,j,1}, \ldots, r_{i,j,na} \) are the poles of \( A_{i,j}(s) \), which are assumed to be distinct. Note that this model will be used for open-loop stable systems so that all poles are different from 0.

The extended OPOM is based on the step responses defined in eq. (0-2) and can be expressed as follows:

\[
x(k+1) = Ax(k) + Bu(k)
\]

\[
y(k) = Cx(k) \tag{0-3}
\]

where \( x(k) = \begin{bmatrix} y(k)^T & y(k+1)^T & \ldots & y(k+p)^T & x^s(k)^T & x^d(k)^T \end{bmatrix}^T \).
In the state vector defined in eq. (0-3), the first \( p + 1 \) components are associated with the output predictions at future time steps, \( x^s \) corresponds to the predicted output at steady state and \( x^d \) are the states corresponding to the stable modes of the system that tend to zero when the system approaches steady state, \( nd \) being the number of stable modes of the system.

\[
B^d = D^d F^d \tilde{N}
\]

with

\[
D^d = \text{diag}(d^d_{1,1} \ldots, d^d_{1,1,nu}, d^d_{1,2}, \ldots, d^d_{1,2,na}, d^d_{1,2,1}, \ldots, d^d_{1,2,1,nu}, d^d_{2,1}, \ldots, d^d_{2,1,nu}, \ldots, d^d_{ny,1} \ldots, d^d_{ny,1,nu}, \ldots, d^d_{ny,1,nu,na})
\]

\[
\tilde{N} = \left[ \begin{array}{cc}
N & \vdots \\
\vdots & \ddots \\
N & \end{array} \right]_{ny}, \quad N = \left[ \begin{array}{cccc}
1 & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
1 & 0 & 0 & \cdots & 0 \\
0 & 1 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & 1 \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & 1
\end{array} \right]
\]
$S_1, \ldots, S_{p+1}$ are the step response coefficients of the system. Matrix $\Psi$ is defined as follows:

$$
\Psi(t) = \begin{bmatrix}
\phi_1(t) & 0 & \ldots & 0 \\
0 & \phi_2(t) & \ldots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \ldots & \phi_{ny}(t)
\end{bmatrix},
$$

where

$$
\phi_i(t) = \begin{bmatrix}
e^{z_{1,1}(t-\theta_1)} & \ldots & e^{z_{1,m}(t-\theta_1)} & \ldots & e^{z_{n,1}(t-\theta_n)} & \ldots & e^{z_{n,m}(t-\theta_n)}
\end{bmatrix}.
$$

With this model formulation, it is easy to show that, if the system reaches a steady state with the output at $y_{ss}$, the system state will stabilize at the following equilibrium point:

$$
x_{ss}(k) = \begin{bmatrix}
y_{ss}^T & y_{ss}^T & \ldots & y_{ss}^T & y_{ss}^T & 0
\end{bmatrix}^T
$$

Observe that the components of the state defined in eq. (0-3) cannot be measured and then have to be estimated by a state observer. The way a state observer is implemented and the state observer that will be used in this thesis will be presented in the two next sections.

**LMI-based robust model predictive control of process systems**

This chapter addresses the solution to the problem of robust model predictive control of chemical processes, for which, to guarantee stability, nonlinear convex constraints are added to the control problem so that the control problem becomes an NLP problem that can be computationally expensive for high dimension systems. Here, the robust MPC problem is formulated as an LMI problem that can be solved in real time with a fraction of the computer effort. The proposed approach is compared with the conventional robust MPC and tested through the simulation of a reactor system and a C3/C4 splitter of the process industry.

Let us note that the controller shown in this section is not based on the realigned model but on the extended OPOM presented in section 0. Indeed, the objective here is to reduce the computational burden of the conventional robust control and that can be performed independently of the type of model chosen.
In the model structure defined in eq. (0-3), the model uncertainty corresponds to uncertainties in matrices $F^d, B^s, B^d$ and in the entries of matrix $\Theta$ that represents the time delays. These uncertainties also reflect in uncertainties in the step response coefficients $S_1, \ldots, S_{p+1}$ that appear in matrix $B$.

In the development of the proposed LMI-based robust MPC, it will be shown that the controller is robust to polytopic uncertainty in some of the parameters of the model defined in eq. (0-3), while, for the other parameters, robustness is only assured for the multi-plant uncertainty. In addition, in the development presented here, it is assumed that the current estimated state corresponds to the true plant.

**Nominal MPC with zone control and optimizing targets**

In this section, it is revised the IHMPC for state space models in the incremental form as in eq. (0-3) considering the zone control of the outputs and assuming that there are targets for some of the inputs (González and Odloak, 2009). This controller is based on the following control objective:

$$V_k = \sum_{j=0}^{\infty} \left( y(k+j|k) - y^{op}_k - \delta_{y,k} \right)^T Q_y \left( y(k+j|k) - y^{op}_k - \delta_{y,k} \right)$$

$$+\sum_{j=0}^{\infty} \left( u(k+j|k) - u_{des,k} - \delta_{u,k} \right)^T Q_u \left( u(k+j|k) - u_{des,k} - \delta_{u,k} \right)$$

$$+\sum_{j=0}^{m-1} \Delta u(k+j|k)^T R \Delta u(k+j|k) + \delta_{y,k}^T S_y \delta_{y,k} + \delta_{u,k}^T S_u \delta_{u,k}$$

where $\Delta u(k+j|k)$ is the control move computed at time $k$ to be applied at time $k+j$, $m$ is the control horizon, $Q_y, Q_u, R, S_y, S_u$ are positive weighting matrices of appropriate dimension, $y^{op}_k$ and $u_{des,k}$ are respectively the output and input targets, $\delta_{y,k}$ and $\delta_{u,k}$ are slack variables that extend the attraction domain of the controller to the whole definition set of the states.

The cost defined in eq. (0-1) can be developed as follows:
\[
V_k = \sum_{j=0}^{p} (y(k+j|k) - y_k^{sp} - \delta_{y,k})^T Q_y (y(k+j|k) - y_k^{sp} - \delta_{y,k}) \\
+ \sum_{j=1}^{\infty} (y(k+j|k) - y_k^{sp} - \delta_{y,k})^T Q_y (y(k+j|k) - y_k^{sp} - \delta_{y,k}) \\
+ \sum_{j=0}^{m-1} (u(k+j|k) - u_{des,k} - \delta_{u,k})^T Q_u (u(k+j|k) - u_{des,k} - \delta_{u,k}) \\
+ \sum_{j=0}^{m-1} \Delta u(k+j|k)^T R \Delta u(k+j|k) + \delta_{y,k}^T S_{y} \delta_{y,k} + \delta_{u,k}^T S_{u} \delta_{u,k}
\]  

(0-2)

The first term of the right hand side of eq. (0-2) can also be expressed as follows

\[
V_{k,1} = \left( \tilde{y}_k - \bar{I}_y y_k^{sp} - \bar{I}_y \delta_{y,k} \right)^T \tilde{Q}_y \left( \tilde{y}_k - \bar{I}_y y_k^{sp} - \bar{I}_y \delta_{y,k} \right)
\]

where

\[
\tilde{y}_k = N_x x(k) + \tilde{S} \Delta u_k
\]

(0-3)

\[
\tilde{y}_k = \begin{bmatrix}
    y(k|k) \\
    y(k+1|k) \\
    \vdots \\
    y(k+p|k)
\end{bmatrix}, \quad
N_x = \begin{bmatrix}
    I_{(p+1)ny} & 0
\end{bmatrix} \in \mathbb{R}^{(p+1)ny \times nx}, \quad
\tilde{S} = \begin{bmatrix}
    0 & 0 & \cdots & 0 \\
    S_1 & 0 & \cdots & 0 \\
    S_2 & S_1 & \cdots & 0 \\
    \vdots & \vdots & \ddots & \vdots \\
    S_p & S_{p-1} & \cdots & S_{p-m+1}
\end{bmatrix}
\]

\[
\Delta u_k = \begin{bmatrix}
    \Delta u(k|k) \\
    \Delta u(k+1|k) \\
    \vdots \\
    \Delta u(k+m-1|k)
\end{bmatrix}, \quad
\bar{I}_y = \begin{bmatrix}
    I_{ny} & \cdots & I_{ny}
\end{bmatrix}^T, \quad
\bar{I}_y \in \mathbb{R}^{(p+1)ny \times ny},
\]

\[
\tilde{Q}_y = \text{diag} \begin{bmatrix}
    Q_y & \cdots & Q_y
\end{bmatrix}_{p+1}, \quad
nx = (p+1)ny + ny + nd.
\]

Then, considering eq. (0-3), \(V_{k,1}\) can be written as follows

\[
V_{k,1} = \left( N_x x(k) + \tilde{S} \Delta u_k - \bar{I}_y y_k^{sp} - \bar{I}_y \delta_{y,k} \right)^T \tilde{Q}_y \left( N_x x(k) + \tilde{S} \Delta u_k - \bar{I}_y y_k^{sp} - \bar{I}_y \delta_{y,k} \right)
\]

(0-4)

The term corresponding to the infinite horizon output error in eq. (0-2) can be written as follows

\[
V_{k,2} = \sum_{j=1}^{\infty} \left( x'(k+m|k) + \Psi(p+j-m)x^d(k+m|k) - y_k^{sp} - \delta_{y,k} \right)^T Q_y \left( x'(k+m|k) + \Psi(p+j-m)x^d(k+m|k) - y_k^{sp} - \delta_{y,k} \right)
\]

(0-5)

where
\[ x'(k+m | k) = x'(k) + \tilde{B}' \Delta u_k \text{ with } \tilde{B}' = \begin{bmatrix} B' & \ldots & B' \\ \vdots & \ddots & \vdots \\ B' & \ldots & B' \end{bmatrix}, \]

\[ x^d(k+m | k) = F^m x^d(k) + \tilde{B}^d \Delta u_k \text{ with } \tilde{B}^d = \begin{bmatrix} F^{-1}B^d & F^{m-2}B^d & \ldots & B^d \end{bmatrix} \]

\[ \Psi(p + j - m) = \Psi(p - m)F^j \]

Then, in order to guarantee that \( V_{k,2} \) will be bounded, the following constraint has to be included in the control problem:

\[ x'(k+m | k) - y^p_k - \delta_{y,k} = 0 \]

or

\[ x'(k) + \tilde{B}' \Delta u_k - y^p_k - \delta_{x,k} = 0 \quad (0-6) \]

Now, assuming that eq. (0-6) is satisfied, eq. (0-5) becomes

\[ V_{k,2} = \sum_{j=1}^{m} \left( \Psi(p - m)F^j x^d(k+m | k) \right)^T Q_y \left( \Psi(p - m)F^j x^d(k+m | k) \right) \]

\[ V_{k,2} = \left( F^m x^d(k) + \tilde{B}^d \Delta u_k \right)^T Q_d \left( F^m x^d(k) + \tilde{B}^d \Delta u_k \right) \quad (0-7) \]

where

\[ Q_d = \sum_{j=1}^{m} \left( \Psi(p - m)F^j \right)^T Q_y \left( \Psi(p - m)F^j \right) \]

Let us now consider the error in the input along the prediction horizon in eq. (0-2):

\[ V_{k,3} = \sum_{j=0}^{m} \left( u(k + j | k) - u_{der,k} - \delta_{u,k} \right)^T Q_u \left( u(k + j | k) - u_{der,k} - \delta_{u,k} \right) \quad (0-8) \]

In order to force the cost defined in eq. (0-8) to be bounded, the following constraint is included in the control problem:

\[ u(k+m | k) - u_{der,k} - \delta_{u,k} = 0 \]

that is equivalent to

\[ u(k-1) + \tilde{I}_u^T \Delta u_k - u_{der,k} - \delta_{u,k} = 0 \quad (0-9) \]

where \( \tilde{I}_u^T = \begin{bmatrix} I_{nu} & \ldots & I_{nu} \end{bmatrix} \).

Then, assuming that eq. (0-9) is satisfied, the cost defined in eq. (0-8) can be written as follows:

\[ V_{k,3} = \left( \tilde{I}_u u(k-1) + M \Delta u_k - \tilde{I}_u u_{der,k} - \tilde{I}_u \delta_{u,k} \right)^T \tilde{Q}_u \left( \tilde{I}_u u(k-1) + M \Delta u_k - \tilde{I}_u u_{der,k} - \tilde{I}_u \delta_{u,k} \right) \quad (0-10) \]
where \( M = \begin{bmatrix} I_{nu} & 0 & \cdots & 0 \\ I_{nu} & I_{nu} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ I_{nu} & I_{nu} & \cdots & I_{nu} \end{bmatrix} \) \( M \in \mathbb{R}^{(nu,m)\times(nu,m)} \) and \( \tilde{Q}_u = \text{diag}(Q_u, \cdots, Q_u) \).

The term penalizing the input effort in eq. (0-2) can be expressed as follows:

\[
V_{k_A} = \Delta u_k^\top \tilde{R} \Delta u_k
\]

(0-11)

where \( \tilde{R} = \text{diag} \left( \frac{R}{m} \cdots R \right) \).

Finally, using eqs. (0-4), (0-7), (0-10) and (0-11), the control cost defined in eq. (0-2) can be written as follows:

\[
V_k = \left[ N_y \Delta u_k + \tilde{S} \Delta u_k - \tilde{I}_y y^m_k - \tilde{I}_y \delta_{yk} \right]^\top \tilde{Q}_y \left[ N_y \Delta u_k + \tilde{S} \Delta u_k - \tilde{I}_y y^m_k - \tilde{I}_y \delta_{yk} \right] + \left( F^m \chi^d(k) + \tilde{B}^d \Delta u_k \right)^\top Q_d \left( F^m \chi^d(k) + \tilde{B}^d \Delta u_k \right) \\
+ \left( \tilde{I}_u u(k-1) + M \Delta u_k - \tilde{I}_u u_{des,k} - \tilde{I}_u \delta_{uk} \right)^\top \tilde{Q}_u \left( \tilde{I}_u u(k-1) + M \Delta u_k - \tilde{I}_u u_{des,k} - \tilde{I}_u \delta_{uk} \right) \\
+ \Delta u_k^\top \tilde{R} \Delta u_k + \delta_{yk}^\top S_y \delta_{yk} + \delta_{uk}^\top S_u \delta_{uk}
\]

(0-12)

In the formulation of the infinite horizon MPC with zone control and input targets, the output setpoint is considered as an additional decision variable of the control problem and the controller results from the solution to the following optimization problem:

\[
\min_{\Delta u_k, y^m_k, \delta_{yk}, \delta_{uk}} V_k
\]

subject to eqs. (0-6), (0-9), (0-12) and

\[
y_{\min} \leq y^m_k \leq y_{\max}
\]

(0-14)

\[
\Delta u_{\min} \leq \Delta u(k + j | k) \leq \Delta u_{\max}
\]

(0-15)

\[
j = 0, \ldots, m-1
\]

\[
u_{\min} \leq u(k-1) + \sum_{i=0}^{j} \Delta u(k+i | k) \leq u_{\max}
\]

(0-16)

\[
j = 0, \ldots, m-1
\]

Constraints defined in eqs. (0-14) - (0-16) define the output zone, the input move limitation and the input range, respectively. The problem defined in eq. (0-13) is a QP
problem that can be easily solved with the available QP solvers. Besides, it is easy to show that the problem defined in eq. (0-13) can be transformed into the following equivalent optimization problem:

**Problem P1**

\[
\min_{\Delta u_k, y_k, \delta_{y,k}, \delta_{u,k}, \gamma_k} \gamma_k \\
\text{subject to eqs. (0-6), (0-9), (0-12), (0-14), (0-15), (0-16) and}\\
V_k \leq \gamma_k, \quad \gamma_k \geq 0
\]

(0-17)

Observe that applying the Schur complement presented in section Erro! Fonte de referência não encontrada. to the inequality defined in eq. (0-18) and using eq. (0-12), the following expression can be obtained:

\[
\begin{bmatrix}
I & 0 & 0 & 0 & 0 & 0 \\
0 & I & 0 & 0 & 0 & 0 \\
0 & 0 & I & 0 & 0 & 0 \\
0 & 0 & 0 & I & 0 & 0 \\
0 & 0 & 0 & 0 & I & 0 \\
0 & 0 & 0 & 0 & 0 & I \\
\end{bmatrix}
\begin{bmatrix}
\sqrt{Q_y} \left( N_y x(k) + \hat{S} \Delta u_k - \tilde{I}_y y_{k} - \tilde{I}_y \delta_{y,k} \right) \\
\sqrt{Q_d} \left( F_x d(k) + \hat{B} d \Delta u_k \right) \\
\sqrt{Q_a} \left( \tilde{I}_a u(k-1) + M \Delta u_k - \tilde{I}_a u_{des,k} - \tilde{I}_a \delta_{a,k} \right) \\
\sqrt{R} \Delta u_k \\
\sqrt{S_y} \delta_{y,k} \\
\sqrt{S_u} \delta_{u,k} \\
\end{bmatrix} \geq 0
\]

(0-19)

It is clear that eq. (0-19) is an LMI in terms of the decision variables of Problem P1, and if used in place of eq. (0-18), Problem P1 becomes an LMI optimization problem that can be solved with available LMI solvers as the MATLAB LMI Toolbox.

The solution of the problem P1 produces a control law that stabilizes the nominal system as established in the theorem below.

**Theorem 1:** For the stable system represented in eq. (0-3), when the model is perfectly known, the state is measured and the desired target is reachable, Problem P1 is always feasible and the successive solutions of this problem produce a sequence of inputs that drives the system to its target while maintaining the remaining system inputs and outputs inside their bounds.

**Proof:** It is easy to show that Problem P1 is always feasible as the slack variables \( \delta_{y,k} \) and \( \delta_{u,k} \) are unbounded and assure that the equality constraints can always be satisfied. Now,
assume that the state is known and there are no disturbances affecting the system. Then, consider the optimum solution to Problem P1 at time step $k$:

$$
\Delta u_k^* = \begin{bmatrix}
\Delta u^*(k | k)^T \\
\vdots \\
\Delta u^*(k + m - 1 | k)^T
\end{bmatrix},
\gamma_{y,k}^*, \delta_{y,k}^*, \delta_{u,k}^* \text{ and } \gamma_k^*.
$$

Next, suppose that the first control move $\Delta u^*(k | k)$ is injected into the real plant and one moves to time step $k + 1$. Then, consider the following set of variables

$$
\Delta \tilde{u}_{k+1} = \begin{bmatrix}
\Delta u^*(k + 1 | k)^T \\
\vdots \\
\Delta u^*(k + m - 1 | k)^T
\end{bmatrix}, \tilde{y}_{k+1}^p = y_{k}^p, \delta_{y,k+1}^* = \delta_{y,k}^*, \delta_{u,k+1}^* = \delta_{u,k}^* \\
$$

and

$$
\tilde{y}_{k+1} = y_k^* - (y(k | k) - y_{k}^p - \delta_{y,k}^*)^T Q_y (y(k | k) - y_{k}^p - \delta_{y,k}^*) \\
- (u^*(k | k) - u_{des,k} - \delta_{u,k}^*)^T Q_u (u^*(k | k) - u_{des,k} - \delta_{u,k}^*) - \Delta u^*(k | k)^T R \Delta u^*(k | k) \tag{0-20}
$$

It is easy to show that the set of variables defined above satisfies the constraints defined in eqs. (0-6), (0-9), (0-14), (0-15), (0-16). Now, to verify that the LMI defined in eq. (0-19) is also satisfied by these variables, define the following matrices

$$
Y = 
\begin{bmatrix}
I & 0 & 0 & 0 & 0 & 0 & \sqrt{Q_y (N_x x(k + 1) + \tilde{S} \Delta \tilde{u}_{k+1} - \tilde{I} y_{k+1}^* - \tilde{I} y_{y,k+1}^*)} \\
0 & I & 0 & 0 & 0 & 0 & \sqrt{Q_u (F^m x^d (k + 1) + \tilde{B}^d \Delta \tilde{u}_{k+1})} \\
0 & 0 & I & 0 & 0 & 0 & \sqrt{Q_u (\tilde{I} u(k) + M \Delta \tilde{u}_{k+1} - \tilde{I} u_{des,k} - \tilde{I} u_{des,k+1})} \\
0 & 0 & 0 & I & 0 & 0 & \sqrt{R \Delta \tilde{u}_{k+1}} \\
0 & 0 & 0 & 0 & I & 0 & \sqrt{S_y \delta_{y,k+1}} \\
0 & 0 & 0 & 0 & 0 & I & \sqrt{S_u \delta_{u,k+1}} \\
(*)^T & (*)^T & (*)^T & (*)^T & (*)^T & (*)^T & (*)^T
\end{bmatrix}
$$

and
It is easy to show that

\[
X^T Y X = \begin{bmatrix} I & 0 & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & 0 & \gamma_k - V_k^* \end{bmatrix}
\]

Consequently, it is clear that

\[X^T Y X \geq 0\]

Now, since \(X\) is non-singular, one concludes that \(Y \geq 0\) and that the LMI defined in eq. (0-19) is also satisfied. Thus, the solution proposed in eq. (0-20) is feasible and corresponds to an upper bound \(\bar{\gamma}_k\) to the cost function and, unless the desired steady state (or target) has been reached, one has \(\bar{\gamma}_k < \gamma_k^*\) and \(\dot{\gamma}_k < \dot{\gamma}_k^*\). This means that the optimum upper bound \(\gamma_k\) to the cost function is a Lyapunov function that guarantees the stability of the closed-loop system with the controller resulting from the solution to Problem P1.

A similar analysis as in (González and Odloak, 2009) can be performed here to show that if the input weight \(S_u\) is large enough and the desired steady state is reachable, then slacks \(\delta_{y,k}\) and \(\delta_{u,k}\) will converge to zero and the targets will be reached. □

There is no apparent advantage in obtaining the stable control law through the solution to Problem P1 instead of solving the QP problem defined in eq. (0-13). The advantage of the
LMI gadgetry to produce a stable MPC becomes more evident when model uncertainty is included in the control problem as presented in the next section.

Robust MPC with zone control and optimizing targetd

Now, consider the multi-model uncertainty defined in section Erro! Fonte de referência não encontrada. that is here characterized by a set of parameters defined as

\[ \Theta_n = \{ B^e_n, B^d_n, F^e_n, F^d_n, \theta_n \}, n = 1, \ldots, L. \]

Also assume that in this case \( p > \max \theta_n(i, j) + m \) (this condition guarantees that the state vector of all models have the same dimension). Then, for each model \( \Theta_n \), following the same steps as in the previous section, one can define the following cost function:

\[
V_k(\Theta_n) = \left( N_n x(k) + \bar{S}(\Theta_n) \Delta u_k - \bar{I}_y y^p(\Theta_n) - \bar{I}_y \delta_{y,k}(\Theta_n) \right)^T \bar{Q}_y \\
\times \left( N_n x(k) + \bar{S}(\Theta_n) \Delta u_k - \bar{I}_y y^p(\Theta_n) - \bar{I}_y \delta_{y,k}(\Theta_n) \right) \\
+ \left( (F(\Theta_n))^m x^d(k) + B^e_n(\Theta_n) \Delta u_k \right)^T Q_u(\Theta_n) \left( (F(\Theta_n))^m x^d(k) + B^d_n(\Theta_n) \Delta u_k \right) \\
+ \left( \bar{I}_a u(k - 1) + M \Delta u_k - \bar{I}_u u_{des,k} - \bar{I}_u \delta_{u,k}(\Theta_n) \right)^T \bar{Q}_u \left( \bar{I}_a u(k - 1) + M \Delta u_k - \bar{I}_u u_{des,k} - \bar{I}_u \delta_{u,k} \right) \\
+ \Delta u_k^T R \Delta u_k + \delta_{y,k}(\Theta_n)^T S_y \delta_{y,k}(\Theta_n) + \delta_{u,k}^T S_u \delta_{u,k}
\]

Also, following the same steps as in the nominal system case, assume that the following constraints are included in the control problem:

\[
\begin{align*}
x^e(k) + \bar{B}^e(\Theta_n) \Delta u_k - y^p(\Theta_n) - \bar{I}_y \delta_{y,k}(\Theta_n) &= 0 \\
u(k - 1) + \bar{I}_u^T \Delta u_k - u_{des,k} - \bar{I}_u \delta_{u,k} &= 0
\end{align*}
\]

Then, at any time step \( k \), the robust MPC for stable systems with time delays and multi-model uncertainty is obtained from the solution to the following optimization problem (González and Odloak, 2011):

Problem P2

\[
\min_{\Delta u_k, y^p(\Theta_n), \delta_{y,k}(\Theta_n), \delta_{u,k}} V_k(\Theta_n) \\
\text{subject to}
\begin{align*}
\Delta u^\min &\leq \Delta u(k + j | k) \leq \Delta u^\max & j = 0, 1, \ldots, m - 1 \\
u^\min &\leq u(k + j | k) \leq u^\max & j = 0, 1, \ldots, m - 1 \\
y^\min &\leq y^p(\Theta_n) \leq y^\max & n = 1, \ldots, L
\end{align*}
\]
\[ x'(k) + \tilde{B}' \Delta u_k - y_{k}^{\text{op}}(\Theta_n) - \delta_{y,k}(\Theta_n) = 0 \quad n = 1, \ldots, L \]

\[ u(k-1) + \tilde{I}_u' \Delta u_k - u_{\text{des},k} - \delta_{u,k} = 0 \]

\[ V_k(\Delta u_k, \delta_{y,k}(\Theta_n), \delta_{u,k}, y_{k}^{\text{op}}(\Theta_n), \Theta_n) \leq V_k(\Delta \bar{u}_k, \bar{\delta}_{y,k}(\Theta_n), \bar{\delta}_{u,k}, \bar{y}_{k}^{\text{op}}(\Theta_n) - \Theta_n) \]

\[ n = 1, \ldots, L \quad (0-21) \]

where, assuming that \((\Delta u_{k-1}^{*}, y_{k-1}^{*}(\Theta_n), \delta_{u,k-1}, \delta_{y,k-1}(\Theta_n))\) is the optimal solution at the previous time step \(k-1\), one defines

\[ \Delta \bar{u}_k = \begin{bmatrix} \Delta u^*(k|k-1)^T \quad \ldots \quad \Delta u^*(k+m-2|k-1)^T \end{bmatrix}^T, \quad \bar{y}_{k}^{\text{op}}(\Theta_n) = y_{k-1}^{*}(\Theta_n), \quad \bar{\delta}_{u,k} \text{ such that} \]

\[ u(k-1) + \tilde{I}_u' \Delta \bar{u}_k - u_{\text{des},k} - \bar{\delta}_{u,k} = 0 \]

and \(\bar{\delta}_{y,k}(\Theta_n)\) such that

\[ x'(k) + \tilde{B}' \Delta \bar{u}_k - \bar{y}_{k}^{\text{op}}(\Theta_n) - \bar{\delta}_{y,k}(\Theta_n) = 0 \quad n = 1, \ldots, L \]

Observe that in Problem P2, \(\Theta_n\) corresponds to the nominal or most probable model of the system. So, in this problem, the objective is to minimize the cost of the most probable plant.

Compared to the nominal control problem, the multi-model control problem includes the nonlinear constraints represented in eq. \((0-21)\). These constraints turn the control problem into an NLP problem, the resolution of which is more complex than the resolution of the QP problem that is obtained in the nominal MPC. For systems with large dimensions, the online resolution of Problem P2 may be too computer demanding. In the next section, Problem P2 is recast as an LMI problem that, as it will be shown later, has a lower computation burden.

**LMI formulation of the robust MPC**

Here, the controller resulting from the solution to problem P2 that was formulated for the multi-plant uncertainty case is re-casted as an LMI problem. Before presenting the problem that defines the controller, one can extend the representation of the model uncertainty.

Suppose that the process to be controlled is represented by the model defined in eq. \((0-3)\), where the model parameters are defined as in the previous section by the set
\[ \Omega = \{ \Theta_1, \cdots, \Theta_L \} \] where each element corresponds to a set of parameters
\[ \Theta_n = \left( F^d, B^s, B^d, \theta \right)_n, \quad n = 1, \ldots, L. \]

Also suppose that the true model is such that
\[ \left( B^s, B^d \right)_T = \sum_{i=1}^L \lambda_i \left( B^s, B^d \right)_i, \quad \sum_{i=1}^L \lambda_i = 1, \lambda_i \geq 0, \quad F^d_T = \left( F^d_1, \ldots, F^d_L \right) \text{ and } \theta_T \in (\theta_1, \ldots, \theta_L). \]

Then, it is interesting to separate \( \left( B^s, B^d \right) \) from the remaining model parameters and to define the two following subsets:
\[ \Upsilon_n = \left( B^s, B^d \right)_n, \quad n = 1, \ldots, L \] and
\[ \Lambda_n = \left( F^d, \theta \right)_n, \quad n = 1, \ldots, L. \]

Observe that the uncertainty described above is more general than the multi-plant uncertainty considered in Problem P2. This means that the controller proposed here will be robust to a larger class of process gains than in the previous section. For the remaining model parameters \( \left( F^d, \theta \right)_n, \quad n = 1, \ldots, L \), the same sort of uncertainty as in the controller defined through Problem P2 is considered, which means that the multi-plant uncertainty is assumed for these parameters.

Now, following the same steps as in the nominal MPC, the robust MPC defined through Problem P2 can be reformulated as an LMI problem that minimizes the upper bound to the control cost of the nominal system and forces the contraction of the upper bound associated with each of the possible process models. Then, the LMI formulation of the robust MPC can be written as follows:

Problem P3
\[
\min \gamma_k \left( \Upsilon_N, \Lambda_N \right)
\]
subject to
\[
\begin{pmatrix}
I & 0 & 0 & 0 & 0 & 0 & X_1 \\
0 & I & 0 & 0 & 0 & 0 & X_2 \\
0 & 0 & I & 0 & 0 & 0 & X_3 \\
0 & 0 & 0 & I & 0 & 0 & \sqrt{R} \Delta u_k \\
0 & 0 & 0 & 0 & I & 0 & \sqrt{S_y} \delta_{y,k} (Y_n) \\
0 & 0 & 0 & 0 & 0 & I & \sqrt{S_u} \delta_{u,k} \\
\end{pmatrix} \geq 0
\]  \quad (0-22)

where

\[
X_1 = \sqrt{Q_y} (N_x x(k) + \tilde{S} (Y_n) \Delta u_k - \tilde{I}_y y_{yp} (Y_n, \Lambda_i) - \tilde{I}_y \delta_{y,k} (Y_n))
\]

\[
X_2 = \sqrt{Q_d} \left( F (\Lambda_i) \right)^m x_d (k) + \tilde{B}^d (Y_n) \Delta u_k \right)
\]

\[
X_3 = \sqrt{O_u} \left( \tilde{I}_u u(k-1) + M \Delta u_k - \tilde{I}_u u_{des,k} - \tilde{I}_u \delta_{u,k} \right)
\]

\[
\Delta u^\min < \Delta u(k+j | k) < \Delta u^\max \quad j = 0,1,\ldots,m-1
\]

\[
u^\min < u(k+j | k) < u^\max \quad j = 0,1,\ldots,m-1
\]

\[
y^\min < y_{yp}^\min (Y_n, \Lambda_i) < y^\max
\]

\[
n = 1,\ldots,L \quad i=1,\ldots,L
\]

\[
x^\prime (k) + \tilde{B}^\prime (Y_n) \Delta u_k - y_{yp}^\prime (Y_n, \Lambda_i) - \delta_{y,k} (Y_n) = 0
\]  \quad (0-23)

\[
u(k-1) + \tilde{I}_u \Delta u_k - u_{des,k} - \delta_{u,k} = 0
\]  \quad (0-24)

\[
-\gamma_k (Y_n, \Lambda_i) + \tilde{V}_k (\Delta u_k, y_{yp}^\prime (Y_n, \Lambda_i), Y_n, \Lambda_i) \geq 0
\]

\[
n = 1,\ldots,L \quad i=1,\ldots,L
\]  \quad (0-25)

It can be shown that the constraint defined in eq. (0-22) together with the constraint defined in eq. (0-25) is equivalent to the following inequality:
The constraints defined in eq. (0-22) are LMIs in the decision variables of the control problem, and so, Problem P3 can be solved with an LMI solver. At this point, it is interesting to observe how the uncertainty in the model parameters affects the constraints of Problem P3. It is not difficult to show that the parameters that appear in the input matrix $B$ of the model defined in eq. (0-3) will appear linearly in all the constraints of problem P3. This means that if these constraints are satisfied by a finite set of models characterized by different values of parameters: $S_i, ..., S_{p+1}, B^s$ and $B^d$, then they will also be satisfied by any convex combination of these models. For the model defined in eq. (0-3), uncertainty in these parameters can be interpreted as uncertainty in the static gains of the process system, which is quite common in the process industries. On the other hand, uncertainty in the parameters that appear in the state matrix $A$ of the model defined in eq. (0-3) is related to the dynamic modes of the system, which, as it can be easily shown, do not appear linearly in the constraints of Problem P3. Consequently, only the multi-plant uncertainty can be considered for these parameters in the controller defined through Problem P3. It is quite straightforward to show that uncertainty in the time delays will also appear nonlinearly in matrix $A$ and consequently follows the same pattern.

Because of the inclusion of slack variables, the controller defined through Problem P3 is always feasible and if the state of the true system is measured, the stability of the closed-loop system can be guaranteed by the following theorem.

Theorem 2: Suppose that the process to be controlled is represented by the model defined in eq. (0-3), where the true model is unknown but is known to belong to $\Omega$, then, the controller resulting from the solution to Problem P3 stabilizes the true plant. This means that, for the true plant, if the desired steady state is reachable, then the system inputs and outputs with targets will converge to these targets while the remaining inputs and outputs will converge to values inside their respective zones.

Proof:
Suppose that the true plant can be represented by a model as defined in eq. (0-3) where matrices $A, B$ and $C$ are such that $(B', B')_T = \sum_{i=1}^{L} \lambda_i (B', B')_T$, $\sum_{i=1}^{L} \lambda_i = 1, \lambda_i \geq 0$, $F'_d \in (F_d', ..., F_d'_L)$ and $\theta_T \in (\theta_1, ..., \theta_L)$. Suppose also that, at any time step $k$, the solution to Problem P3 is represented by: $\{\Delta u^*_k, y^{sp}_k(Y_n, \Lambda_r), \delta^{w}_{y,k}(Y_n, \Lambda_r), \delta^{a}_{u,k}, \gamma^*_k(Y_n, \Lambda_r)\}$ with $n = 1, ..., L$ and $i = 1, ..., L$, then, it is easy to show that the following set of variables

$$\left\{\Delta u^*_k, \sum_{n=1}^{L} \lambda_n y^{sp}_k(Y_n, \Lambda_r), \sum_{n=1}^{L} \lambda_n \delta^{w}_{y,k}(Y_n, \Lambda_r), \delta^{a}_{u,k}, \sum_{n=1}^{L} \lambda_n \gamma^*_k(Y_n, \Lambda_r)\right\} \quad (0-26)$$

is a feasible solution to problem P3 if this problem is written only for the true plant (assuming that the true plant is known). Observe that in this case, the upper bound to the true process cost function is given by $\gamma^*_k(Y_r, \Lambda_r) = \sum_{n=1}^{L} \lambda_n \gamma^*_k(Y_n, \Lambda_r)$.

Now, consider that $\Delta u^*(k | k)$ is injected in the real process and one moves to time step $k+1$ where Problem P3 has to be solved again. Now, following a similar procedure as in Theorem 1, one can show that if $\Delta u^*_{k+1}$ is defined as in Theorem 1 and one defines

$$\tilde{y}^{sp}_{k+1}(Y_n, \Lambda_r) = y^{sp}_k(Y_n, \Lambda_r), \quad \tilde{\delta}^{w}_{y,k+1}(Y_n, \Lambda_r) = \delta^{w}_{y,k}(Y_n, \Lambda_r), \quad \tilde{\delta}^{a}_{u,k+1} = \delta^{a}_{u,k},$$

$$\tilde{\gamma}^*_k(Y_r, \Lambda_r) = \gamma^*_k(Y_r, \Lambda_r)$$

$$-\left( y(k | k) - y^{sp}_k(Y_r, \Lambda_r) - \delta^{w}_{y,k}(Y_r, \Lambda_r) \right)^T Q_y \left( y(k | k) - y^{sp}_k(Y_r, \Lambda_r) - \delta^{w}_{y,k}(Y_r, \Lambda_r) \right)$$

$$-\left( u^*(k | k) - u^*_d - \delta^{a}_{u,k} \right)^T Q_u \left( u^*(k | k) - u^*_d - \delta^{a}_{u,k} \right) - \Delta u^*(k | k)^T R \Delta u^*(k | k)$$

then

$$\left\{\Delta u^*_{k+1}, \sum_{n=1}^{L} \lambda_n \tilde{y}^{sp}_{k+1}(Y_n, \Lambda_r), \sum_{n=1}^{L} \lambda_n \tilde{\delta}^{w}_{y,k+1}(Y_n, \Lambda_r), \tilde{\delta}^{a}_{u,k+1}, \sum_{n=1}^{L} \lambda_n \tilde{\gamma}^*_k(Y_n, \Lambda_r)\right\}$$

is a feasible solution to Problem P3 at time $k+1$ for the true process system and obviously $\tilde{\gamma}^*_k(Y_r, \Lambda_r) \leq \gamma^*_k(Y_r, \Lambda_r)$ where equality holds only if a steady state has been reached. Consequently $\gamma^*_{k+1}(Y_r, \Lambda_r) \leq \gamma^*_k(Y_r, \Lambda_r)$ and $\gamma^*_k(Y_r, \Lambda_r)$ is a Lyapunov function to the closed-loop system that is asymptotically stable. □
Implementation of the robust MPC in the LMI framework

Observe that in Problem P3, the constraints defined in eqs. (0-23) and (0-24) represent equality constraints that cannot be included in some of the available LMI solvers, as for instance the MATLAB LMI Toolbox, for which all the constraints need to be expressed in terms of inequality constraints. However, since the slack variables $\delta_{y,k} (Y_n, \Lambda_i)$ and $\delta_{u,k}$ are unbounded, eqs. (0-23) and (0-24) can be used to eliminate these variables from the remaining constraints of the control problem. Then, the cost function can be represented in terms of a reduced set of decision variables as follows:

$$V_k(Y_n, \Lambda_i) = \left[ N_s x(k) - \tilde{I}_s x'(k) + (\tilde{S}(Y_n) - \tilde{B}^*(Y_n))\Delta u_k \right]^T \tilde{Q}_y$$

$$+ \left[ N_s x(k) - \tilde{I}_s x'(k) + (\tilde{S}(Y_n) - \tilde{B}^*(Y_n))\Delta u_k \right]$$

$$+ \left( (F(\Lambda_i))^m x^d(k) + \tilde{B}^d(Y_n, \Lambda_i)\Delta u_k \right)^T Q_d(\Lambda_i) \left( (F(\Lambda_i))^m x^d(k) + \tilde{B}^d(Y_n, \Lambda_i)\Delta u_k \right)$$

$$+ \left[ (M - \tilde{I}_u)\Delta u_k \right]^T \tilde{Q}_u \left[ (M - \tilde{I}_u)\Delta u_k \right] + \Delta u_k^T \tilde{R}\Delta u_k$$

$$+ \left( x'(k) + \tilde{B}^* (Y_n) \Delta u_k - y_{kp}^p(Y_n, \Lambda_i) \right)^T S_y \left( x'(k) + \tilde{B}^* (Y_n) \Delta u_k - y_{kp}^p(Y_n, \Lambda_i) \right)$$

$$+ (u(k-1) + \tilde{I}_u^T \Delta u_k - u_{des,k})^T S_u \left( u(k-1) + \tilde{I}_u^T \Delta u_k - u_{des,k} \right)$$

that can be written in the following quadratic form

$$V_k(Y_n, \Lambda_i) = \left[ \Delta u_k^T y^p_{kp} (Y_n, \Lambda_i)^T \right] \begin{bmatrix} H_{11}(Y_n, \Lambda_i) & H_{12}(Y_n) \\ H_{21}(Y_n) & H_{22} \end{bmatrix} \begin{bmatrix} \Delta u_k \\ y^p_{kp} (Y_n, \Lambda_i) \end{bmatrix}$$

$$+ 2 \begin{bmatrix} C_{jk,1}(Y_n, \Lambda_i) \\ C_{jk,2} \end{bmatrix} \begin{bmatrix} \Delta u_k \\ y^p_{kp} (Y_n, \Lambda_i) \end{bmatrix} + c_k(Y_n, \Lambda_i) \quad \text{(0-27)}$$

where

$$H_{11}(Y_n, \Lambda_i) = \left[ \tilde{S}(Y_n) - \tilde{B}^* (Y_n) \right]^T \tilde{Q}_y \left[ \tilde{S}(Y_n) - \tilde{B}^* (Y_n) \right]$$

$$+ \tilde{B}^d(Y_n, \Lambda_i)^T Q_d(\Lambda_i) \tilde{B}^d(Y_n, \Lambda_i) + \tilde{R} + \tilde{B}^* (Y_n)^T S_y \tilde{B}^* (Y_n) + \tilde{I}_u S_u \tilde{I}_u^T$$

$$H_{12}(Y_n) = -\tilde{B}^* (Y_n)^T S_y$$

$$H_{21}(Y_n) = H_{12}(Y_n)^T$$

$$H_{22} = S_y$$

$$C_{jk,1}(Y_n, \Lambda_i) = \left[ N_s x(k) - \tilde{I}_s x'(k) \right]^T \tilde{Q}_y \left[ \tilde{S}(\Theta_n) - \tilde{B}^* (\Theta_n) \right]$$

$$+ \left[ (F(Y_n, \Lambda_i))^m x^d(k) \right]^T Q_d(\Lambda_i) \tilde{B}^d(Y_n, \Lambda_i)$$

$$+ x'(k)^T S_y \tilde{B}^* (Y_n) + \left[ u(k-1) - u_{des,k} \right]^T S_u \tilde{I}_u^T$$
\[ C_{jk,2} = -x'(k)^T S_y \]
\[
c_k(Y_n, \Lambda_i) = \left[ N_{xi}(k) - \tilde{I}_{xi}x'(k) \right]^T \tilde{Q}_y \left[ N_{xi}(k) - \tilde{I}_{xi}x'(k) \right] + \left[ (F(\Lambda_i))^m x^d(k) \right]^T Q_d(\Lambda_i) \left[ (F(\Lambda_i))^m x^d(k) \right] + x'^T(k)S_yx'(k) + \left[ u(k-1) - u_{des,k} \right]^T S_u \left[ u(k-1) - u_{des,k} \right] \]

Eq. (0-27) can also be written as follows:
\[ V_k(Y_n, \Lambda_i) = Z_k(Y_n, \Lambda_i)^T H(Y_n, \Lambda_i) Z_k(Y_n, \Lambda_i) + 2C_{jk}(Y_n, \Lambda_i) Z(Y_n, \Lambda_i) + c_k(Y_n, \Lambda_i) \]
where
\[ Z_k(Y_n, \Lambda_i) = \begin{bmatrix} \Delta u_k \\ y_{jp}^{sp}(Y_n, \Lambda_i) \end{bmatrix}, \quad H(Y_n, \Lambda_i) = \begin{bmatrix} H_{11}(Y_n, \Lambda_i) & H_{12}(\Theta_n) \\ H_{21}(Y_n) & H_{22} \end{bmatrix} \]
\[ C_{jk}(Y_n, \Lambda_i) = \begin{bmatrix} C_{jk,1}(Y_n, \Lambda_i) & C_{jk,2} \end{bmatrix}. \]

Finally, to be solved with an LMI solver such as the one available in the MATLAB LMI Toolbox, Problem P3 is reformulated as follows:

Problem P4
\[
\min_{\Delta u_i, y_{jp}^{sp}(Y_n, \Lambda_i), \ n=1,...,L \ i=1,...,L} \gamma_k(Y_n, \Lambda_N) \\
\text{subject to} \\
\left[ I \begin{bmatrix} Z_k(Y_n, \Lambda_i)\sqrt{H(Y_n, \Lambda_i)} \end{bmatrix} \begin{bmatrix} \gamma_k(Y_n, \Lambda_i) - 2C_{jk}(Y_n, \Lambda_i) Z(Y_n, \Lambda_i) - c_k(Y_n, \Lambda_i) \end{bmatrix} \right] > 0 \\
\begin{bmatrix} \dot{V}_k(Y_n, \Lambda_i) - \gamma_k(Y_n, \Lambda_i) > 0 \end{bmatrix} \\
\begin{bmatrix} \Delta u_i(k + j \mid k) - \Delta u_i^{min} > 0 \end{bmatrix} \\
\begin{bmatrix} \Delta u_i^{max} - \Delta u_i(k + j \mid k) > 0 \end{bmatrix} \\
\begin{bmatrix} u_i(k + j \mid k) - u_i^{min} > 0 \end{bmatrix} \\
\begin{bmatrix} u_i^{max} - u_i(k + j \mid k) > 0 \end{bmatrix} \\
\begin{bmatrix} y_{jp}^{sp}(Y_n, \Lambda_i) - y_{jp}^{min} > 0 \end{bmatrix} \\
\begin{bmatrix} y_{jp}^{max} - y_{jp}^{sp}(Y_n, \Lambda_i) > 0 \end{bmatrix} \\
\end{bmatrix} > 0 \\
\text{where} \\
\begin{bmatrix} \dot{V}_k(Y_n, \Lambda_i) \end{bmatrix} = \begin{bmatrix} \Delta \hat{u}_k^T & \tilde{y}_{jp}^{sp}(Y_n, \Lambda_i) \end{bmatrix} \begin{bmatrix} H_{11}(Y_n, \Lambda_i) & H_{12}(\Theta_n) \\ H_{21}(Y_n) & H_{22} \end{bmatrix} \begin{bmatrix} \Delta \hat{u}_k \\ \tilde{y}_{jp}^{sp}(Y_n, \Lambda_T) \end{bmatrix} + 2\begin{bmatrix} C_{jk,1}(Y_n, \Lambda_i) & C_{jk,2} \end{bmatrix} \begin{bmatrix} \Delta \hat{u}_k \\ \tilde{y}_{jp}^{sp}(Y_n, \Lambda_T) \end{bmatrix} + c_{k-1}(Y_n, \Lambda_i)
Observe that Problem P4 is equivalent to Problem P3, but now the robust control problem is an LMI problem as defined in section Error! Fonte de referência não encontrada.. This problem can then be solved, for instance, with the MATLAB LMI Toolbox. Besides being more general in terms of considering a broader class of model uncertainties, the robust LMI-based MPC that results from the solution to Problem P4 shows a better potential in terms of numerical efficiency when compared to the robust NLP-based MPC based on the solution to Problem P2. In the next section, low order simulation examples are used to compare the performance of these two methods.

Simulation results

Application to a Fluid Catalytic Cracking (FCC) system

The system considered here is part of the FCC system studied in (Sotomayor and Odloak, 2005) where more details about this system can be found. A schematic representation of a FCC system is shown in Figure 0-1.
The FCC system is a highly nonlinear system, so that a conventional MPC, based on a single linear model, may have a poor performance in a context where the optimization of the plant may result in frequent operating point changes.

The subsystem considered here has 2 inputs and 3 outputs. In this reduced system, the manipulated inputs correspond to: $u_1$ the air flow rate to the catalyst regenerator and $u_2$ the opening of the regenerated catalyst valve, and the controlled outputs are: $y_1$ the riser temperature, $y_2$ the regenerator dense phase temperature and $y_3$ the regenerator dilute phase temperature.

Three models corresponding to different operating points of the FCC system were considered based on the one obtained experimentally. These models are assumed to constitute the multi-model set $\Omega$ on which the robust controller is based. Based on these models, one may construct the polytope in which the gain of the true model is supposed to
The parameters corresponding to each of the models can be seen in the following transfer functions:

\[
G_1(s) = \begin{bmatrix}
0.4515e^{-2s} & 0.2033e^{-4s} \\
2.9846s+1 & 1.7187s+1 \\
1.5e^{-6s} & (0.1886s - 3.8087)e^{-3s} \\
20s+1 & 17.7347s^2 + 10.8348s + 1 \\
1.7455e^{-6s} & -6.1355e^{-5s} \\
9.1085s+1 & 10.9088s+1
\end{bmatrix}, \quad G_2(s) = \begin{bmatrix}
0.25e^{-2s} & 0.135e^{-5s} \\
3.5s+1 & 2.77s+1 \\
0.9e^{-3s} & (0.1886s - 2.8)e^{-4s} \\
25s+1 & 19.7347s^2 + 10.8348s + 1 \\
1.25e^{-5s} & -5e^{-6s} \\
11.1085s+1 & 12.9088s+1
\end{bmatrix},
\]

\[
G_3(s) = \begin{bmatrix}
0.7e^{-3s} & 0.5e^{-4s} \\
1.98s+1 & 2.7s+1 \\
2.3e^{-5s} & (0.1886s - 4.8087)e^{-3s} \\
25s+1 & 15.7347s^2 + 10.8348s + 1 \\
3e^{-4s} & -8.1355e^{-6s} \\
7s+1 & 7.9088s+1
\end{bmatrix}.
\]

In the first simulation shown in this section, the performances of the LMI-based robust MPC resulting from the solution to Problem P4 and of the NLP-based robust MPC resulting from the solution to Problem P2 are compared. In this simulation, the multi-plant uncertainty is considered, model \(G_1\) is the true model \(\Theta_T\) and model \(G_3\) represents the nominal model \(\Theta_N\). For the implementation of this case, the MATLAB routine “mincx” was used to solve Problem P4 while the MATLAB “fmincon” routine was used to solve Problem P2. Of course, other algorithms may have been used for the resolution of problems P2 and P4 and different results in terms of computational efficiency may have been obtained, nevertheless the results that will be obtained with these algorithms should indicate a clear tendency.

In order to provide a fair comparison of the performances and computation costs of the two controllers, the same algorithm stopping criteria were implemented for both controllers.

The following tuning parameters were adopted for both controllers.

\[
T = 1\text{min}, \quad m = 3, \quad Q_y = \text{diag}(0.5, 0.5, 0.5), \quad R = \text{diag}(10, 10), \quad Q_u = \text{diag}(1, 1),
\]

\[
S_y = \text{diag}(1, 1, 1) \times 10^3, \quad S_u = \text{diag}(1, 1) \times 10^3.
\]

Figure 0-2 and Figure 0-3 show the responses of both robust controllers for the case where the input target is \(u_{des} = (225, 71)\), the output zone limits are \(y_{min}^T = [545 \ 685 \ 670]^T\) and \(y_{max}^T = [550 \ 700 \ 685]^T\), the input bounds are \(u_{min}^T = [75 \ 25]^T\) and \(u_{max}^T = [250 \ 100]^T\),
and the input move bound is $\Delta u^{\text{max}} = [5\ 2]^T$. The reactor system starts from the initial operating point defined by $u_0 = [230\ 70]^T$ and $y_0 = [549.5\ 704.3\ 690.6]^T$. The responses of the two controllers are quite similar and there are no major differences between the performances of the NLP-based MPC and the LMI-based MPC. Note that $y_2$ starts from a value outside the control zone and that both controllers can easily bring this output back to its control zone. At the same time the inputs are driven to their targets after a few time steps while all the outputs converge to steady state values inside their respective control zones. Here, the input targets are reachable, which means that at steady state, when the inputs are at their targets, the corresponding values of the outputs lie inside their control zones. The only significant difference between the two controllers is the computer time to perform the simulation. With the NLP-based MPC, the simulation takes 45.4s, while, with the LMI-based MPC, the simulation lasts only 10.5s. Then, these results confirm that the proposed LMI-based approach is capable of a significant reduction of the computer time when compared to the conventional approach. At time instant $k = 50\text{min}$, the output bounds are changed to $y^{\text{min}} = [545\ 695\ 680]^T$ and $y^{\text{max}} = [550\ 710\ 695]^T$. With these new bounds, the steady state values of $y_2$ and $y_3$ corresponding to the input targets would lie outside their bounds, which means that, with the new control specifications, the input targets are unreachable. The controllers have to change the system inputs in order to bring the outputs to a point inside the new zones. Both controllers reach the same steady state for which the distance to the input target values is minimized. Figure 0-4 shows, for the LMI-based MPC, the variable $\gamma_k(\Theta_T)$ that is the upper bound to the cost function of the true model that in this simulation is $G_1$. One can observe that for $k<50$ where the target is reachable, $\gamma_k(\Theta_T)$ asymptotically decreases to zero, as guaranteed by Theorem 2. However, for $k>50$, as the target becomes unreachable, $\gamma_k(\Theta_T)$ tends to a value that is not null confirming that the closed-loop system is stable but does not converge to the desired steady state.
Figure 0-2: Outputs with the LMI-based MPC (——) and the NLP-based MPC (—), and bounds (− − −). Multi-plant uncertainty.
Figure 0-3: Inputs with the LMI-based MPC (——) and the NLP-based MPC (---) and targets (· · · · ·). Multi-plant uncertainty
The diagram shows a plot of $\text{Gam T}$ versus time (min). The y-axis is labeled $\text{Gam T}$, scaled from $-1 \times 10^6$ to $6 \times 10^6$. The x-axis represents time in minutes, ranging from 0 to 50. The plot appears to be a straight line at a constant value on the y-axis.
In the second simulation case presented here, the multi-plant uncertainty is still considered and the LMI-based MPC is tested for the case where there are targets to output $y_1$ and to input $u_2$, while the remaining outputs should be controlled inside their zones. It is also investigated the effect of switching between MPC controllers that are based on different nominal models. The tuning of the LMI-based MPC is the same as in the first simulation case except for the input error and the input slack weights that are changed to $Q_u = diag(0,1)$ and $S_u = diag(0,1) \times 10^3$ respectively.

Figure 0-5 and Figure 0-6 show the outputs and inputs responses for the closed-loop system with the LMI-based MPC when $u_{des,2} = 71$ and the output zone limits are initially $y_{min} = [548 \ 685 \ 670]^T$ and $y_{max} = [548 \ 715 \ 710]^T$. The input bounds are the same as in the first case. Observe that the lower and upper bounds to output $y_1$ are the same, which corresponds to considering a fixed target for this output. The system represented by model $G_1$ starts from the same initial steady state as in the first case and is driven by the controller.
to the desired targets smoothly and without offset since the steady state corresponding to
the selected targets is reachable. At time step $k = 50\text{ min}$, the target of output $y_1$ and the
bounds to the remaining outputs are changed as shown in Figure 0-5. Although the target of
input $u_2$ remained the same, the new desired steady state becomes unreachable. The
closed-loop is still stable but there are offsets in $y_1$ and $u_2$. To test the effect of switching
between controllers that are based on different nominal models $\Theta_N$, the simulation is
started with a controller based on model $G_1$, which means that in Problem P4, the objective
function is the upper bound corresponding to model $G_1$. Then, at time step $k = 10\text{ min}$, the
nominal model is switched to $G_2$ and, finally, at time step $k = 60\text{ min}$, the nominal model
is switched to $G_3$. Figure 0-7 shows $\gamma_k(\Theta_T)$, the upper bound to the true plant, that is
assumed to be $G_1$. One can observe that $\gamma_k(\Theta_T)$ is strictly decreasing even when there is a
switch in the nominal model considered by the controller. This interesting property of the
proposed controller can be easily proved and allows the optimization of the performance of
the robust controller through the online selection of the most appropriate nominal model.
Figure 0-5: Outputs with the LMI-based MPC (solid), bounds (dashed) and setpoints (dotted). Switching nominal model.
Figure 0-6: Inputs with the LMI-based MPC (—) and targets (······). Switching nominal model
In the third simulation case presented here, the polytopic uncertainty model is considered for the system gain and the LMI-based MPC is tested for the case where there are fixed setpoints for two of the three outputs. To simulate this case, one considers models $G_1$, $G_2^*$ and $G_3^*$ where the denominator and time delays of the last two models are made equal to the denominator and time delays of model $G_1$ while the numerators are made equal to the numerators of models $G_2$ and $G_3$ respectively. The tuning parameters of the LMI-based MPC are the same as in the previous simulation, except for weights $Q_u$ and $S_u$ that are both made equal to $\text{diag}([0 \ 0])$. Figure 0-8 and Figure 0-9 show the system responses when the true model is assumed to be the following:

$$G_f = 0.3G_1 + 0.2G_2^* + 0.5G_3^*$$

Since the desired steady states defined by the output fixed setpoints are reachable, the LMI-based MPC can easily drive $y_1$ and $y_2$ to their setpoints without offset. Figure 0-10 shows
the bound to the cost of the true plant and it is clear that this bound is asymptotically decreasing and converges to zero, which shows, as demonstrated in Theorem 2, that the closed-loop system remains stable.
Figure 0-8: Outputs with the LMI-based MPC (---) and bounds (−−−). Polytopic uncertainty.
Figure 0-9: Inputs with the LMI-based MPC. Polytopic uncertainty.
Figure 0-10: Upper bound to the cost function of the true plant with the LMI-based MPC. Polytopic uncertainty.