MPC : Homework Part

Homework : Model Predictive Control

Molin LIU
Xinjian OUYANG
Candela ARCE

P2022 CentraleSupélec

09 May 2021

1 Settings chosen

1.1 Operating points added

Considering the operating points : $u_0(k+1) = u_0(k)(=5)$ and $y_0(k+1) = y_0(k)(=5)$,we define the new state variable $x(k)$ as :

$$x(k) = \begin{pmatrix} x(k) \\ u_0(k) \\ y_0(k) \end{pmatrix} \tag{1.1.1}$$

The initial problem : $\begin{cases} x(k+1) = Ax(k) + Bu(k) - Bu_0(k) \\ y(k) = Cx(k) + y_0(k) \end{cases}$ can be transformed into :

$\begin{cases} x(k+1) = A_a x(k) + B_a u(k) \\ y(k) = C_a x(k) \end{cases}$ with :

$$A_a = \begin{pmatrix} A & -B & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{pmatrix} \quad B_a = \begin{pmatrix} B \\ 0 \end{pmatrix} \quad C_a = \begin{pmatrix} C & 0 & I \end{pmatrix} \tag{1.1.2}$$

1.2 Setting up an observer

As shown in the MATLAB file, we derive the observable part of the state matrices as A_o B_o and C_o. Once the observer is introduced, the problem is rewritten as :

$\begin{cases} \hat{x}(k+1) = A_o\hat{x}(k) + B_o u(k) + L_0(y(k) - \hat{y}(k)) \\ \hat{y}(k) = C_o\hat{x}(k) \end{cases}$

To satisfy the corresponding form in SIMULINK : $\begin{cases} x_{n+1} = A_{obs}x_n + B_{obs}u_n \\ y_n = C_{obs}x_n + D_{obs}u_n \end{cases}$ with :

$$x_n = \hat{x}(k) \quad u_n = \begin{pmatrix} u(k) \\ y(k) \end{pmatrix} \quad y_n = \hat{x}(k) \tag{1.2.1}$$

We get the forms of those observable state matrices :

$$A_{obs} = A_o - L_0 C_o \quad B_{obs} = \begin{pmatrix} B_o & L_0 \end{pmatrix} \quad C_{obs} = I \quad D_{obs} = 0 \tag{1.2.2}$$

1.3 Sampling Period & Prediction Horizon size

Since we have a limited duration of 1.5 seconds. With a define simple period Te, the size of the prediction horizon is :

$$Np = \frac{1.5s}{Te} \tag{1.3.1}$$

Taking in account that **we have chosen a simple period equal to 0,05s our prediction horizon is going to be 30.**

1.4 Optimization criteria

We implemented three different optimization methods to solve this problem : Linear Programming, Quadratic Programming and Q.P. with slack variables.

1.4.1 Linear Programming

$$J^{Np} = \sum[\lambda_1(y - ymin) + \lambda_2 u] = f^T Z \tag{1.4.1.1}$$

where

$$f = \begin{pmatrix} T1 & -T1 & T2 \end{pmatrix}, Z = \begin{pmatrix} Y \\ Ymin \\ U \end{pmatrix} \tag{1}$$

with

$$T1 = \begin{pmatrix} \lambda_1 \\ \lambda_1 \\ \dots \\ \lambda_1 \end{pmatrix}^{\mathrm{T}}_{2Np\times 1}, T2 = \begin{pmatrix} \lambda_2 \\ \lambda_2 \\ \dots \\ \lambda_2 \end{pmatrix}^{\mathrm{T}}_{2Np\times 1} \tag{1.4.1.2}$$

1.4.2 Quadratic Programming (original version)

$$J^{Np} = (Y - Y_{min})^T (Y - Y_{min}) + \lambda_u U^T U = \frac{1}{2} Z^T W Z + f^T Z \tag{1.4.2.1}$$

with

$$Z = \begin{pmatrix} Y \\ U \end{pmatrix} \quad W = \begin{pmatrix} I & 0 \\ 0 & \lambda_u I \end{pmatrix} \quad and \quad f = \begin{pmatrix} -Y_{min} \\ 0 \end{pmatrix} \tag{1.4.2.2}$$

1.4.3 Quadratic Programming (with slack variables ϵ)

$$J^{Np} = (Y - Y_{min})^T (Y - Y_{min}) + \lambda_u U^T U + \lambda_e \epsilon^T \epsilon = \frac{1}{2} Z^T W Z + f^T Z \tag{1.4.3.1}$$

with

$$Z = \begin{pmatrix} Y \\ U \\ \epsilon \end{pmatrix} \quad W = \begin{pmatrix} I & 0 & 0 \\ 0 & \lambda_u I & 0 \\ 0 & 0 & \lambda_e I \end{pmatrix} \quad and \quad f = \begin{pmatrix} -Y_{min} \\ 0 \\ 0 \end{pmatrix} \tag{1.4.3.2}$$

Notice that λ_e , compared withλ_u, should be very large to ensure that the slackers are very small.

1.5 Solver

1.5.1 Linear Programming

The constraints for linear programming are as following :

$$\begin{pmatrix} I & 0 & -H \end{pmatrix} Z = F X_o \tag{1.5.1.1}$$

$$\begin{pmatrix} 0 & I & 0 \end{pmatrix} Z = Y_{min} \tag{1.5.1.2}$$

$$\begin{pmatrix} 0 & 0 & T \end{pmatrix} Z \leq UTOTmax \tag{1.5.1.3}$$

$$\begin{pmatrix} -I & I & 0 \end{pmatrix} Z \leq 0 \tag{1.5.1.4}$$

$$\begin{pmatrix} P & -P & 0 \end{pmatrix} Z \leq \begin{pmatrix} 2 \\ 2 \\ \dots \\ 2 \end{pmatrix}_{Np \times 1} \tag{1.5.1.5}$$

with (we take Np =3 for example)

$$T = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}_{Np=3} \qquad P = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}_{Np=3} \tag{1.5.1.6}$$

However, the linprog solver in MATLAB always find **no feasible solution** with this strategy. Therefore, we refer to quadratic method to reconsider it.

1.5.2 Quadratic Programming

At first, we don't introduce the slack variable, and the constraints are similar with those above :

$$\begin{pmatrix} I & -H \end{pmatrix} Z = FX_o \tag{1.5.2.1}$$

$$\begin{pmatrix} 0 & T \end{pmatrix} Z \leq UTOTmax \tag{1.5.2.2}$$

$$\begin{pmatrix} -I & 0 \end{pmatrix} Z \leq -Ymin \tag{1.5.2.3}$$

$$\begin{pmatrix} P & 0 \end{pmatrix} Z \leq PYmin + 2 \tag{1.5.2.4}$$

While we try this strategy again, **but still could not find a feasible solution.**
Thus, we decide to introduce the slack variable to solve it.

1.5.3 Quadratic Programming with slack variables

As shown in 1.4, ϵ refers to the slack variable and its dimension is $3Np \times 1$ as the flow measurement doesn't have a maximum constraint.

$$\begin{pmatrix} I & -H & 0 \end{pmatrix} Z = FX_o \tag{1.5.3.1}$$

$$\begin{pmatrix} 0 & T & 0 \end{pmatrix} Z \leq UTOTmax \tag{1.5.3.2}$$

$$\begin{pmatrix} -I & 0 & -Q \end{pmatrix} Z \leq -Ymin \tag{1.5.3.3}$$

$$\begin{pmatrix} P & 0 & -R \end{pmatrix} \leq PYmin + 2 \tag{1.5.3.4}$$

with (we take Np =2 for example)

$$Q = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}_{Np=2} \qquad R = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}_{Np=2} \tag{1.5.3.5}$$

Luckily and finally, we find a optical solution with this strategy, which will be detailed in next part.

2 Matlab/Simulink : implentation of Q.P. with slack variables

u1 : flow control, u2 : temperature control
y1 : flow measure, y2 : temperature measure

We have tried two kinds of poles :
$poles_{obs} = eig(Ao)$;
$poles_{obs} = exp(-3 * Te * [1 : 1 : 10])$

We got that the $poles_{obs} = eig(Ao)$ is the better one. Then we take eig(Ao) as the poles where Ao is the observable part of the state matrices.

2.1 Simulink

After setting up the "*process model*" block and "*control*" block, where we replaced the matrices(A,B,C,D) in state-space function by the matrices we got from matlab scipt file, we got the results in Simulink as followings, with the configuration $Np = 30, Te = 0.05, \lambda_u = 0.1, \lambda_\epsilon = 100$.

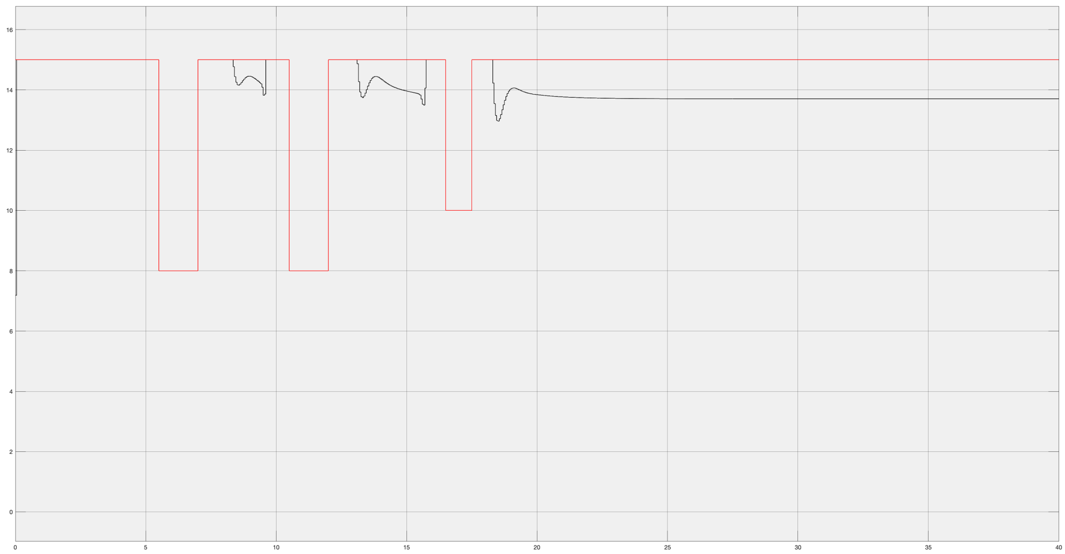

FIGURE 1 – Scope of input control : u1+u2

The red curve is $UTOTmax$, and the black curve is $u1 + u2$ denoted as u_{tot}.

It can be seen that u_{tot} converges to around 13.7 finally, and the control constraint are well respected.

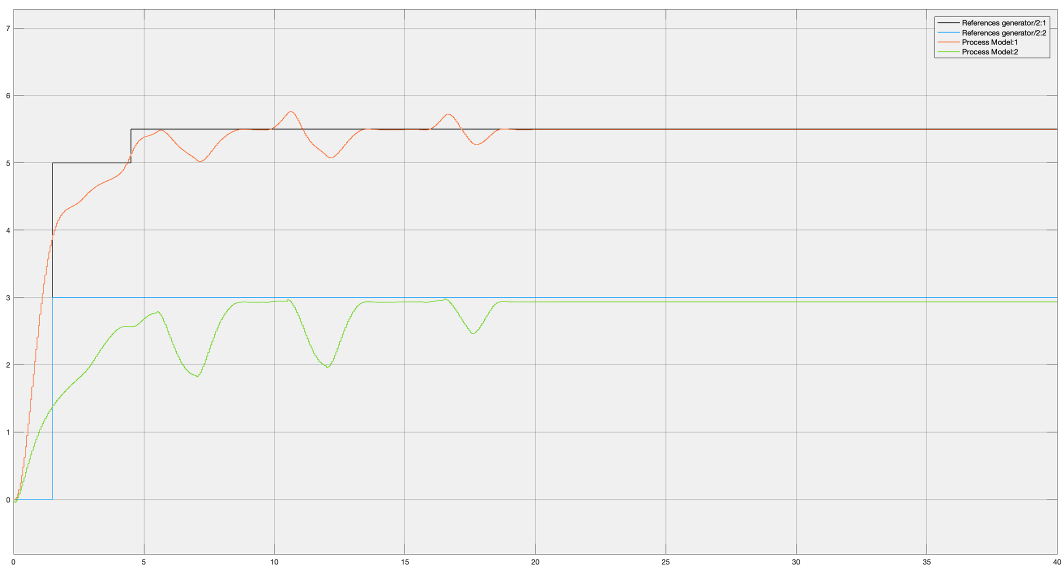

FIGURE 2 – Scope of output measurement : y

The black curve is the reference of y1 while the red curve is the output y1 ;(flow measure)
The blue curve is the reference of y2 while the green curve is the output y2 ;(temperature measure)
It can be seen from Figure 2 that output y1 and y2 converge around the references eventually. However, since the influence of slackers, the outputs are not strictly consistent with the references.

2.2 Matlab

In Matlab, we set the bound of contraints **manually** as following :

y1min = 0.2 ;
y2min = 0.1 ;
UTOTmax = 20*ones(Np,1).

Notice that to better observe the convergence of output, we set Np as 100.

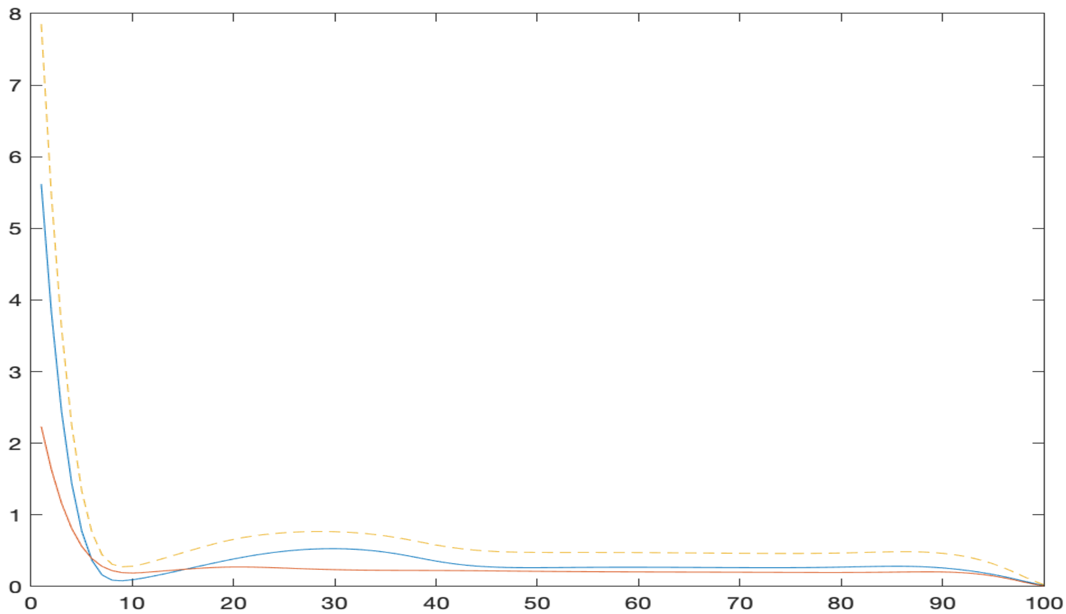

FIGURE 3 – Input control : u1, u2 and u1+u2

where the blue line is the input control $u1$, the orange line is the input control $u2$ and the dotted line is the sum of input u1+u2, noted as u_{tot}.

It can be seen that u_{tot} is always lower that UTOTmax, so **the input constraint are satisfied**.

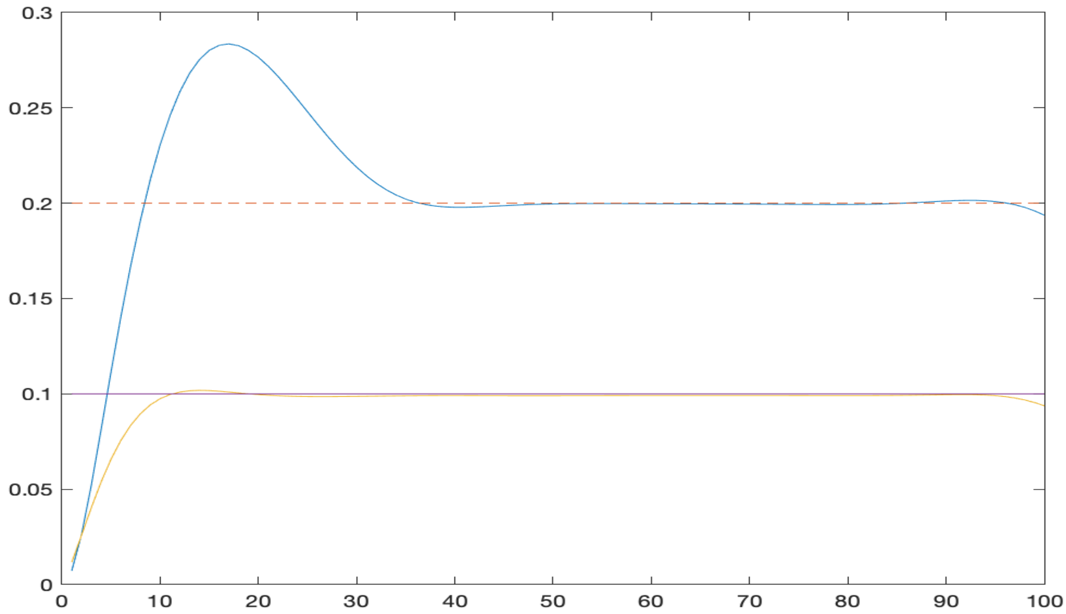

FIGURE 4 – output of process model : y

where the purple line is y1min(set as 0.1), the dotted line is y2min(set as 0.1), the blue line is the output y1, and the yellow line is the output y2.

From Figure 4, it can be seen that output y1 and y2 converge around the references(Ymin) eventually. However, since the influence of slackers, the outputs are not strictly consistent with the references.**Anyway, the constraints of outputs are well respected.**

For more details of the realization of the whole process, please refer to the MATLAB file.