FRTN15 Predictive Control
Laboratory experiment 3
Model Predictive Control

Department of Automatic Control

Lund Institute of Technology

1. Introduction

The goal of this laboratory exercise is to study Model Predictive Control (MPC) by applying it to the quadruple-tank process. In the configuration used in this exercise the process has nonminimum-phase behaviour and there are also some constraints which need to be taken into account. We are going to study a few aspects of designing MPC controllers, such as the choice of weight functions, prediction parameters and constraint handling. The controllers will be evaluated both in simulation and in experiments on a real process.

For an introduction to MPC and information about MPCtools, the toolbox used in this exercise, see Åkesson [2006]. A good text book on MPC is Maciejowski [2001].

1.1 Login

Log onto the lab machines with user your LTH student account.

1.2 Preparations

Read this manual carefully and perform the preparatory exercises.

1.3 Model Predictive Control

MPC differs from most other control strategies in the way the control action is calculated. A finite horizon optimal control problem is solved at each sampling instant. The first step in the calculated control sequence is applied to the plant, while the resulting steps are discarded. The calculations are then repeated at the next sampling instant. This principle is known as receding horizon control. A great advantage with solving the optimal control problem on-line is that it makes it possible to handle MIMO plants and constraints explicitly. Two obstacles that need to be considered are the computation time required to solve the optimal control problem at each sampling instant, and that a model of the plant is required.

Traditionally, MPC has been applied to plants with rather slow dynamics so that the computation time is negligible compared to the sampling intervals used. One typical application is process control. The ability to handle constraints explicitly and the MIMO-capabilities of MPC make it a suitable choice for this type of applications. However, MPC has also been applied to plants with much faster dynamics such as air planes and combustion engines.

1.4 MPCtools

MPCtools is a freely available MATLAB/Simulink-based toolbox for simulation of MPC controllers. It is developed by Johan Åkesson at the Department of Automatic Control, Lund University. In combination with MATLAB, MPCtools allows the user to create and simulate MPC controllers. The controllers may also be applied to physical plants and nonlinear models via Simulink.

1.5 Files needed

The following files are available through the course homepage, and needed

```
mpc_controller.m
mpc_tank_simulated.mdl
mpc_tank_real.mdl
```

Also, you need to download MPCtools, which is freely available at

```
http://www.control.lth.se/user/johan.akesson/mpctools/index.html
```

Once MATLAB is started, the directory where MPCtools is located needs to be added with the following code

```
>>addpath('/path/to/MPCtools-1.0')
```

2. Modeling

The quadruple-tank process is depicted in Figure 1. The goal is to control the water levels in the lower tanks by applying voltage to the two pumps. The flow through pump 1 is divided between tanks 1 and 4 and the flow through pump 2 is divided between tanks 2 and 3. The tubes connected to the lower tanks are narrower than those connected to the upper tanks so that only 30% of the flows from the pumps enter the lower tanks. Because of this, the system has nonminimum-phase behaviour (zeros in the right half plane). Additionally, the pumps have limited capacity and you should avoid flooding the tanks. In summary, the quadruple tank process is a nonminimum-phase MIMO system with constraints on the control signals as well as on the states.

2.1 Nonlinear Model

For tank i, the cross-section is denoted A_i [cm^2], the cross-section of the outlet is denoted a_i [cm^2] and the water level is denoted h_i [cm]. The control signals, u_1 [V] and u_2 [V], control the flow through the corresponding pumps. The flow through pump i is given by $q_i = k_i u_i$ [cm^{3}s^{-1}] where k_i [cm^{3}s^{-1}v^{-1}] is a constant. The flow is then divided by γ_1 [1] and γ_2 [1] so that $\gamma_1 k_1 u_1$ enters Tank 1 and $(1-\gamma_1)k_1u_1$ enters Tank 4. The flows into Tank 2 and 3 are given by similar expressions with γ_2, k_2, and u_2 instead. Process specific constants are shown in Table 1

The water levels in the tanks are described by the following nonlinear equations[1].

[1] These equations are readily derived using a mass balance and Bernoulli's law for each tank.

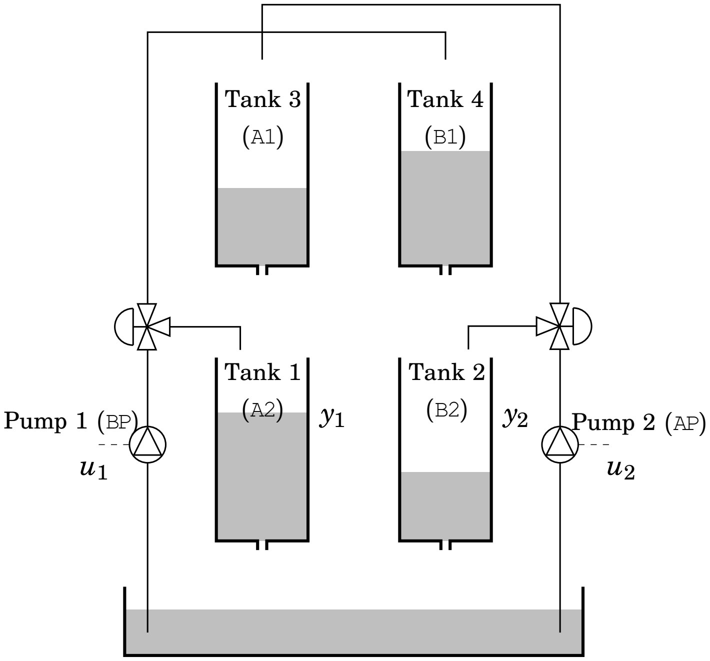

Figure 1 The quadruple-tank process.

$$
\begin{aligned}
\frac{dh_1}{dt} &= -\frac{a_1}{A_1}\sqrt{2gh_1} + \frac{a_3}{A_1}\sqrt{2gh_3} + \frac{\gamma_1 k_1}{A_1}u_1 \\
\frac{dh_2}{dt} &= -\frac{a_2}{A_2}\sqrt{2gh_2} + \frac{a_4}{A_2}\sqrt{2gh_4} + \frac{\gamma_2 k_2}{A_2}u_2 \\
\frac{dh_3}{dt} &= -\frac{a_3}{A_3}\sqrt{2gh_3} + \frac{(1-\gamma_2)k_2}{A_3}u_2 \\
\frac{dh_4}{dt} &= -\frac{a_4}{A_4}\sqrt{2gh_4} + \frac{(1-\gamma_1)k_1}{A_4}u_1
\end{aligned}
$$

2.2 Linear Model

The measured heights are converted into voltages by a factor k_c [V cm^{-1}]. After a linearization around a stationary operating point $(h_1^0, h_2^0, h_3^0, h_4^0, u_1^0, u_2^0)$ the following state-space representation is obtained, where $\Delta x = x - x^0$, $\Delta y = y - y^0$ and $\Delta u = u - u^0$. The state vector x contains the heights $h_1 \ldots h_4$ and we have $u = \begin{pmatrix} u_1 & u_2 \end{pmatrix}^T$.

Name	Value	Unit
A_i	4.9	cm^2
a_i	0.03	cm^2
k_i	1.6	cm^3 V^{-1} s^{-1}
γ_i	0.3	1
k_c	0.5	V cm^{-1}

Table 1 Process specific constants.

$$
\begin{aligned}
\Delta \dot{x} &= \begin{pmatrix} -\frac{1}{T_1} & 0 & \frac{A_3}{A_1 T_3} & 0 \\ 0 & -\frac{1}{T_2} & 0 & \frac{A_4}{A_2 T_4} \\ 0 & 0 & -\frac{1}{T_3} & 0 \\ 0 & 0 & 0 & -\frac{1}{T_4} \end{pmatrix} \Delta x + \begin{pmatrix} \frac{\gamma_1 k_1}{A_1} & 0 \\ 0 & \frac{\gamma_2 k_2}{A_2} \\ 0 & \frac{(1-\gamma_2)k_2}{A_3} \\ \frac{(1-\gamma_1)k_1}{A_4} & 0 \end{pmatrix} \Delta u, \\
\Delta y &= \begin{pmatrix} k_c & 0 & 0 & 0 \\ 0 & k_c & 0 & 0 \end{pmatrix} \Delta x,
\end{aligned}
$$

where

$$T_i = \frac{A_i}{a_i}\sqrt{\frac{2x_i^0}{g}}$$

The choice of γ_1 and γ_2 will affect the zeros of the system. Throughout this laboratory exercise we will have $\gamma_i = 0.3$ which yields a nonminimum-phase system.

Since the tanks have limited heights (16 cm) and the pumps have limited capacity (10 V) the constraints in Table 2 will be used in this laboratory exercise.

Signal	Min	Max	Unit
u_i	0	10	V
h_i	0	15	cm

Table 2 Constraints on the control signals and water levels.

3. Controller Design

Start MATLAB and open the file

```
mpc_controller.m
```

The goal of this section is to understand what this file does and to make some initial parameter choices. We will do this by going over the different parts of the file.

3.1 Model

The first part of the file defines the linear process model with $\gamma_i = 0.3$, $h_1^0 = h_2^0 = 10$ cm. A linearized state-space model is then discretized with sample time T_s [s]. It is worth noting that three C-matrices and two D-matrices are defined; C_{yd}, C_{zd}, C_{cd}, D_{zd}, and D_{cd}.

```
% Discretize model
[Ad, Bd, Cyd, Dzd]=ssdata(c2d(ss(A,B,C,D),Ts));
Czd = Cyd;
Ccd = kc*eye(4);
Dcd = zeros(4,2);
```

In this case $C_{zd} = C_{yd}$ which means that the controlled outputs are the same as the measured outputs. Since all states are constrained, C_{cd} is set to the identity matrix times the conversion factor k_c; D_{cd} is set to zero since there is no direct term from the control signals to the constrained outputs.

Exercise 1 (Preparatory Exercise) Consider the linearized model and the stationary point. The controller measures the deviations from this point. What measurement does the controller receive when the tanks are empty? What measurement does the controller receive when the tanks are full? Which output from the controller generates the maximum (minimum) allowed voltage (10 V and 0 V respectively) to the pumps? The measurement constant is $k_c = 0.5$ Vcm^{-1}. Use the constraints of Table 2, rather than the real tank heights in this exercise.

3.2 Constraints

The second part of the file defines the parameters needed for the MPC-controller. First, the constraints are defined.

```
% Constraints
% limit delta u
du_max = []';
du_min = []';
% limit absolute value of u
u_max = []';
u_min = []';
% limit controlled outputs
z_max = []';
z_min = []';
```

Exercise 2 (Preparatory Exercise) First make sure that you understand what the different constraints mean. Then enter values so that the constraints in Table 2 are satisfied. Note that the constraints should be entered in terms of deviations from the linearization point. Will there be any limitations on du_{max}/du_{min}?

3.3 Prediction Parameters

Next, the prediction parameters need to be defined.

Exercise 3 (Preparatory Exercise) As a starting point, enter the following values. Make sure that you understand what the parameters mean. Draw a figure with time on the x-axis which shows H_p and H_u.

```
% Prediction parameters
Hp=30;   % Prediction horizon
Hu=10;   % Horizon for varying input signal
Hw=1;    % First penalty sample
zblk=2;  % blocking factor
ublk=2;  % blocking factor
```

3.4 Weighting Matrices

Exercise 4 (Preparatory Exercise) Enter the following values for the weighting matrices; Q and R for the controlled outputs and the control signals, and W and V for the Kalman filter design[2]. Also, explain what this choice of matrices means in the optimization problem.

```
% Weights
Q=diag([1 1]);
R=diag([1 1]);
W=diag([1 1 1 1]);
V=0.1*diag([1 1]);
```

Finally, a controller is created using the following command

```
md = MPCInit(Ad,Bd,Cyd,Czd,Dzd,Ccd,Dcd,Hp,Hw,zblk,Hu,ublk, ...
    du_max,du_min,u_max,u_min,z_max, ...
    z_min,Q,R,W,V,Ts,2,'qp_as');
```

The second last argument is set to 2, which means that the controller includes a Kalman filter in order to estimate the unmeasured states. The last argument specifies which solver to use for the optimization problem. MPC-tools includes an active set solver, *qp_as*, an interior point solver, *qp_ip*, and also supports use of of the function *quadprog* in the Optimization Toolbox. For further reading, see Åkesson [2006] and the references therein.

[2] If all states were available for measurement, the W and V matrices would not be needed.

4. Simulation Experiments

Open the file

```
mpc_tank_simulated.mdl
```

The block named `MPCController` uses the object `md` which we created in the previous section. Every time you make changes to the controller parameters, you need to rerun the file `mpc_controller.m`. We will now simulate various MPC controllers to see the effects of different parameter choices.

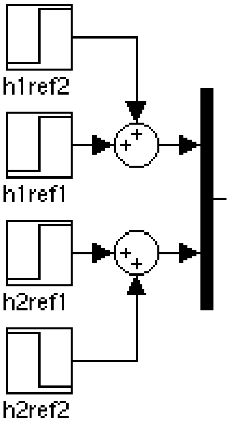

Figure 2 The reference trajectory is specified in these blocks.

Exercise 5 The reference trajectory is specified in the blocks depicted in Figure 2 with final values set to `h1ref1=h1+1`, `h2ref1=h2+3`, `h1ref2=`3, and `h2ref2=`1. Run the simulation. The water levels should remain in $[0, 15]$ cm and the control signals in $[0, 10]$ V. If there are constraint violation, try to explain them.

Exercise 6 Now change `z_max` so that the maximum allowed water level in Tank 2 is 14 cm (instead of 15 cm) and run the simulation. How does this affect the results?

Exercise 7 Restore the constraints on the water level and change `u_max` so that the maximum allowed value of u_2 is $u_2 = 5$ V instead of $u_2 = 10$ V. Does the controller still manage to follow the reference trajectory?

Exercise 8 Restore the constraints on the control signal. The MPC is based on a linearized model, wheareas the real proecss is de fact non-linear. Deviations from the linearization point introduce a model error, which can be viewed as a disturbance. The controller you have designed will not be able to suppress the effects of this, since it lacks integral action. One way of achieving integral action is to use a disturbance observer. To do this, the second last input argument to `MPCinit` needs to be set to 4 instead of 2. Also, the weighting matrix W must contain weights for the disturbance states as well. A suitable starting point is

```
W=diag([1 1 1 1 1 1]);
md = MPCInit(Ad,Bd,Cyd,Czd,Dzd,Ccd,Dcd,Hp,Hw,zblk,Hu,ublk, ...
    du_max,du_min,u_max,u_min,z_max, ...
    z_min,Q,R,W,V,Ts,4,'qp_as');
```

Exercise 9 Try varying the Q- and R-matrices. Are the results consistent with your expectations for the following configurations?

```
Q=diag([4 1]);  R=diag([1 1]);
Q=diag([4 1]);  R=100*diag([1 1]);
Q=diag([1 40]); R=diag([1 1]);
Q=daig([1 40]); R=diag([1 100]);
```

Choose a configuration that you find yields good results.

Exercise 10 The choice of `Hp` is very important in order to preserve performance and even stability. The blocking factor `zblk` may be used in order to reduce the complexity of the optimization problem. To see the effects of `Hp`, try (at least) the following configurations and comment on the results. Note that the computation times may be quite extensive for some configurations.

```
Hp=30; zblk=2;
Hp=90; zblk=6;
Hp=10; zblk=1;
```

Exercise 11 The sample time T_s affects the choice of `Hp`. To see this, change the sample time to $T_s = 0.1$ s and repeat the experiments in the previous exercise.

5. Experiments

Restore the sample time to $T_s = 0.5$ s and open the file `mpc_tank_real.mdl`. Make sure a serial cable connects the process and PC and that the process is connected to the grid. Also verify that water is filled to the mark. Valves `AV3`, `AV4`, `BV3`, `BV4`, `V5` should be pressed down, valves `AV1`, `BV2`, `BV1` should be pulled out and valve `AV2` should be pressed in.

Exercise 12 Once you have found parameters that yield good results in simulation, try your controller on the real process. Are the results similar to those obtained in simulation? Are the constraints violated at any point? Try to explain any dissimilarities.

The controller is sensitive to process variations. The real processes have indiviual variations in γ_1, γ_2 as well as a_1, a_2, a_3, a_4.
(Parameters A_1, A_2, A_3, A_4, k_c are not subject to significant variation between process instances. Sensitivity towards variations in k_1, k_2 is low due to a flow control loop embedded in the quadruple tank model.)

Exercise 13 Conduct a parameter identification experiment using the simulink model '`tank_manual.mdl`'. This model is equipped with 'slider gain' blocks, with which you can control pump flows during ongoing simulations and scopes for monitoring water levels in tanks 1-4. Device experiments using '`tank_manual.mdl`' and any of the `A` and `B` valves on the process to identify the parameters and enter them into `mpc_controller.m`. Verify that previously used controller parameters still yield satisfactory results in simulation and then try it on the real process.

References

Johan Åkesson. MPCtools 1.0—Reference Manual. Technical Report ISRN LUTFD2/TFRT--7613--SE, Department of Automatic Control, Lund Institute of Technology, Sweden, January 2006.

J.M. Maciejowski. *Predictive Control with Constraints*. Prentice-Hall, 2001.

Document history:
Created by Anders Widd
Revised September 2009 by Kristian Soltesz