KTH Electrical Engineering

Model Predictive Control Course - EL2700

Lab 2 - MPC for double tank process

MIKAEL JOHANSSON,
PEDRO F. LIMA, VALERIO TURRI, MARTIN BIEL

Stockholm September 28, 2018

Automatic Control
School of Electrical Engineering
Kungliga Tekniska Högskolan

Contents

Introduction

The goal of this laboratory exercise is to study Model Predictive Control (MPC) by applying it to the double-tank process. We are going to study a few aspects of designing MPC controllers, such as the choice of weight functions, prediction parameters and constraint handling. Also, we compare the performance of MPC against traditional PID control and LQR design. The controllers will be evaluated both in simulation and in experiments on a real process.

1 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. Only the first step in the calculated control sequence is applied to the plant. The calculations are then repeated at the next sampling instant. This principle is known as receding horizon control. A great advantage of solving the optimal control problem on-line is that MIMO plants and constraints can be handled explicitly. Two important considerations are the computation time required to solve the optimal control problem at each sampling instant, and the requirement of a plant model. Traditionally, MPC has been applied to systems with rather slow dynamics, such as process control plants, where the computation time is negligible compared to the sampling time. However, with the emergence of powerful computing hardware, MPC has also been applied to systems with faster dynamics, such as air planes and combustion engines.

2 The Process

The process to be controlled consists of two water tanks and a pump, as illustrated in Figure 1. Water is pumped from a reservoir to the upper tank, from which it flows through a restriction to the lower tank. From the lower tank the water flows through a second restriction back to the reservoir. The upper tank also has a tap from which, when opened, water is allowed to flow directly to the reservoir. The water levels in both tanks are measured using pressure sensors. The process and equipment are manufactured by Quanser Consulting in Canada.

The control system is implemented in Matlab on a PC, using the toolboxes Realtime Workshop and Realtime Target. This lab is partially based on material from the course in EL1000 Basic course on Automatic Control.

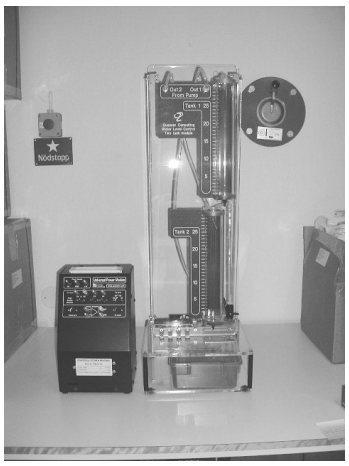

Figure 1: Coupled tanks with pump.

3 Modeling

Assume that both tanks have the same cross-sectional area, i.e., $A1 = A2 = A$, let a_1 and a_2 be the effective outlet areas. The relationship between the outflow velocity $v(t)$ [m/s] and the water level $h(t)$ [m] in a tank is given by Bernoulli's law

$$v(t) = \sqrt{2gh(t)}.$$

The model can be written as

$$\frac{dh_1(t)}{dt} = -\alpha_1\sqrt{2gh_1(t)} + \beta u(t), \tag{1}$$

$$\frac{dh_2(t)}{dt} = \alpha_1\sqrt{2gh_1(t)} - \alpha_2\sqrt{2gh_2(t)}, \tag{2}$$

where $h_1(t)$ and $h_2(t)$ are the water levels in the two tanks, and $u(t)$ is the pump voltage. The parameters $\alpha_1 = a_1/A$ and $\alpha_2 = a_2/A$.

The system is linearized around equilibrium points h_1^0 , h_2^0, and u^0 and the resulting transfer functions are given by

$$\Delta H_1(s) = \frac{k}{1+\tau s}\Delta U(s), \tag{3}$$

$$\Delta H_2(s) = \frac{\gamma}{1+\gamma\tau s}\Delta X_1(s), \tag{4}$$

where $\tau = \frac{1}{\alpha_1}\sqrt{\frac{2h_1^0}{g}}$, $\gamma = \frac{h_2^0}{h_1^0}$, and $k = 60\beta\tau$.

4 Lab overview

The lab is composed by two parts:

- The first part is conducted in simulation and should be used as preparation before the scheduled lab session. In this part, you will get familiar with the system under control and design multiple controllers, i.e., PID, LQR,

and MPC. The simulation part is divided into multiple tasks to guide you in the design of the controllers and understand their relation.

- The second part is conducted in the water tank lab according to the course schedule. In this part, you will first implement the LQR and MPC controller on the real system and, second, explore how different formulations of the MPC controller affect the controller performance.

At the end of the lab session, you should report the results from the experimental part to the TAs. We encourage you to work with the preperatory simuliation in groups of two people (For example, your design task groups) to boost discussions and critical thinking. In the water tank lab, you will possibly work in larger groups depending on how many water tanks are available.

5 GUI reference

The necessary files to run the lab are contained in the zipped folder `EL2700_MpcTankLab` that are available on the computers in the water tank lab (look for the link on the desktop named `EL2700_MPCTankLab`).

In this folder you find this Lab documentation (the current file), some extra folders containing MATLAB functions, .m files, and a .mlapp file. To run the lab GUI you only need to run the script `lab_start.m`.

It is important that you have Matlab 2017a or above to be able to run the GUI. Moreover, you can run the simulation part entirely outside the lab (e.g., in your own personal computer).

The GUI was developed during the summer in 2017 by Shuqi Xu. It consists of several panels containing switches, checkboxes, text boxes, buttons and a big plot area. In the following, we summarize the main features of the GUI:

- To start the simulation, change the first switch from `Stop` to `Start` in the upper-left corner. To start the experiment, change the third switch from `Simulator` to `Realtime` and the first switch to `Start`.
- In the second switch, you can choose between `Manual` and `Name_of_the_controller`. This allows you to navigate and change the parameters in the controller tabs (`PID`, `LQR`, `MPC`) without applying the control immediately.
- The fourth switch is only used in simulation and simulates the effect of the opening the tap.
- The middle-top panel consists of several tabs with the different controllers that can be used. These controllers will only be active when the second switch on the panel to the left is not set to `Manual`.
- To evaluate the performance of different controllers, you will be asked to control the lower tank as it is subjected to a step reference of amplitude 20, starting from a water level of 40. To do so, use the manual control to stabilize the lower tank at 40 ± 1, then choose a reference value of 60 inside the `Name_of_the_controller` tab, and finally switch from Manual to

<Name of the controller> to apply the controller settings. You can speed up the simulation up to five times when not using MPC.

- To capture the result of a simulation or an experiment, choose how much time you want to capture and press the `Capture` button on the right-hand side. This button opens a new window where a plot of the selected signals are shown. You can also save this figure.

- You can analyze a simulation or an experiment externally by saving the data to a `.mat` file. Everytime the GUI is started, all the `.mat` in the current directory are deleted. This is to avoid cluttering the workspace in the lab room computers. Please be advised to save important data in a separate folder when working locally on your own computers, to avoid loosing results if the GUI is restarted.

- Only the `PID` tab contains a switch to control the upper or lower tank. In the `LQR` or the `MPC` tab there is no switch to choose the lower tank or the upper tank. However, you can choose to penalize the deviations from the reference in only one of the tanks.

- Inside the `MPC` tab, there is a `Modelconfigure` button. There you can modify the double tank model parameters. Note that, to save the modifications you need to click `Savemodel` in the main GUI!

Chapter 1

Simulation

In this part, you will get familiar with the system under control and design multiple controllers, i.e., PID, LQR, and MPC. The simulation part is divided in multiple tasks to guide you in the design of the controllers and in understanding their relation.

You are encouraged to go through this part before the scheduled lab session as preparation. You do not have to present your simulation results, but you are expected to finish the experimental part of the lab in 2 hours. Working through the simulation assignments will make you familiar with the GUI and the water tank process, so that you will be able to finish the experimental part in time.

1.1 PID-Control

A PID-controller is obtained by introducing proportional, integral, and differential effects

$$u(t) = K\left[e(t) + \frac{1}{T_I}\int_0^t e(\tau)d\tau + T_D\frac{d}{dt}e(t)\right].$$

From now on, you will be asked to control the **lower tank**.

Task 1.1

(a) Use a PID controller (choose the `PID` tab) to control the lower tank (choose `Lowertank` in the switch). Define the gains $K = 5$, $T_D = 5$, and $T_I = 80$. Input a step reference of amplitude 20 starting at level 40. Press `Capture`, save the figure, and compute the **settling time** (5%), **rise time** (from 10% to 90%) and **overshoot** of the step response.

(b) Use the knowledge from the basic control course to tune the PID controller to get a better step response (**do not over do this task**), i.e., faster rise

time and settling time and less overshoot. Avoid overflow of any of the tanks. Explain your reasoning. Evaluate your tuning with a step of amplitude 20 starting at level 40. Press `Capture`, save the figure, and compute the new **settling time** (5%), **rise time** (from 10% to 90%), and **overshoot** of the step response. Also, write down the new PID gains.

(c) Draw the closed-loop block diagram.

1.2 Linear-Quadratic Regulator

A Linear-Quadratic Regulator (LQR) is a state feedback optimal controller. Typically, the cost function of an LQR is defined as a sum of the deviations from a reference value over an infinite horizon. The algorithm finds the controller gains that minimize these deviations. Also, the magnitude of the control action itself can also be included in the cost function.

From now on, you will be asked to control the **lower tank**.

Task 1.2

(a) Propose an LQR controller formulation such that the water level can track a desired reference in discrete-time.

(b) Draw the closed-loop block diagram.

(c) Control the system using an LQR (choose the tab `LQR`). Input a step reference of amplitude 20 starting at level 40. Make sure that the `Referencechoiceswitch` is in `Constant` inside the `LQR` tab. Use `Upper_tank_penalty`= 0, `Lower_tank_penalty`= 1, and `Input_penalty`= 0.01. Press `Capture`, save the figure, and compute the **settling time** (5%), **rise time** (from 10% to 90%), and **overshoot** of the step response.

(d) Tune the LQR controller by modifying the **tanks and input penalties**. Reason about their influence in the step response properties. You can save several figures to illustrate your argument.

(e) Compare the performance of the **PID** and the **LQR**? Which are the pros and cons of each controller? Which one gives better results? Why?

1.3 Unconstrained Model Predictive Controller

In Model Predictive Control (MPC) the horizon is limited to a finite time. At every step, the optimization formulated in the previous task is solved and the first optimal input is sent to the system - this is called receding-horizon control.

From now on, you will be asked to control the **lower tank**.

Make sure that the input and state constraint checkboxes are **unchecked**.

Task 1.3

(a) Formulate an **unconstrained MPC**, such that the water level can track a desired reference in discrete-time by limiting the horizon of the LQR to a **finite time**. In your formulation, include a state terminal cost.

(b) Control the lower using an MPC (choose the tab `MPC`). Input a step reference of amplitude 20 starting at level 40. Use `Upper_tank_penalty`= 0, `Lower_tank_penalty`= 1, and `Input_penalty`= 0.01. Also, use a closed-loop prediction horizon of 10 steps. Make sure that the `Referencechoice` switch is in `Constant` inside the `MPC` tab. Press `Capture`, save the figure, and compute the **settling time** (5%), **rise time** (from 10% to 90%), and **overshoot** of the step response.

(c) Without changing the tank penalties or the input, try different closed-loop prediction horizons (by inserting a value in the corresponding text box), e.g., 3, 5, 10, 15. Explain the similarities and differences between the **LQR** and **MPC** formulations.

(d) Lets study the influence of the **Riccati terminal penalty** (turn `On` and `Off` the `Riccati_terminal_penalty` switch). Use again a step reference of amplitude 20 starting at level 40. Use a **prediction horizon** of 5 steps.

Press `Capture`, save the figure, and compute the **settling time** (5%), **rise time** (from 10% to 90%), and **overshoot** of the step response. How does it compare with the LQR performance?

(e) Explain the similarities and differences between the **LQR** and **MPC** performance.

1.4 Constrained MPC

In a real system, physical constraints have to be taken into account. For example, in the specific case of the water tanks, the input is a water pump that has a voltage limit. Also, the tanks have limited height, and hence the water levels are limited.

From now on, you will be asked to control the **lower tank**. By default, the input and state constraints are off.

Task 1.4

(a) Assume that it is not desirable to exceed an input of 70%.

(i) **Without using input constraints**, retune the MPC controller such that the input is always below 70%, while aiming at good performance (fast response, short settling time and no (or low) overshoot). Control the lower tank using a step reference of amplitude 20 starting at level 40. Make sure that the `Reference_choice` switch is in `Constant` inside the `MPC` tab.

(ii) Repeat the experiment (and retune the MPC), but now explicitly activating **only** the input constraints between 0 and 70.

Press `Capture`, save the figure, and compute the **settling time** (5%), **rise time** (from 10% to 90%), and **overshoot** of the step responses. Compare the two experiments.

(b) Now, you can use the full range of the input, i.e., from 0 to 100. Also, the input constraints should be on. Assume that it is not desirable to exceed a water level of 70% for both tanks.

(i) **Without using state constraints**, retune the MPC in order to satisfy the new constraints, while aiming at good performance (fast response, short settling time and no (or low) overshoot). Input a step reference of amplitude 20 starting at level 40. Make sure that the `Reference_choice` switch is in `Constant` inside the `MPC` tab.

(ii) Repeat the experiment (and retune the MPC), but now explicitly activating the state constraints, such that the water level cannot exceed a level of 70% for both tanks.

Press `Capture`, save the figure, and compute the **settling time** (5%), **rise time** (from 10% to 90%), and **overshoot** of the step responses. Compare the two experiments.

(c) Can you limit the water level with the other controllers (PID and LQR)? Explain why.

1.5 Open-loop vs closed-loop MPC

The goal of this last task is to answer the following question: why do we need to solve the optimization problem every step? We will now compare the performance of the controller by using the optimal control input sequence computed by **one** optimization of the MPC problem, with the performance of the closed-loop MPC.

Task 1.5

(a) In the MPC tab, select the input and state constraints between 0 and 100. Also, switch the MPC choice to `Open-loop` and choose a prediction horizon

of 100. The MPC will then recompute a solution every 100 sampling times. The open-loop MPC solution is input to the system from the moment the step is given. Input a step reference of amplitude 20 starting at level 40. Make sure that the `Reference_choice` switch is in `Constant` inside the `MPC` tab. Press `Capture` and save the figure. Using the same tuning (you can set the input penalty quite small to allow faster convergence), compare with the receding-horizon (or closed-loop) MPC (turn to `Closed-loop` in the `MPC_choice` switch).

(b) Do you expect the results obtained in (a) to be similar experimentally? Why?

Chapter 2

Experiment

The experimental part will be conducted during the lab session, according to the course schedule. If possible, maintain the same groups as when you worked with the simulation part. You will test the MPC in practice and analyze how the MPC formulation can be changed to improve the controller performance.

We strongly recommend to use the opportunity of using a real system to explore the MPC framework. Do not be (or feel) limited by the proposed tasks. Change the state and input penalization matrices, modify the prediction horizon, and open the water tank tap are just some of the possibilities that you have. Discuss your results with the other groups and with the TAs. Be critical.

2.1 Identify the system

We will now start with the experimental part of the lab. Before starting, make sure that you select `Real-time` in the corresponding switch.

As seen in the introduction part the double tank system can be described by the following linearized model:

$$\Delta H_1(s) = \frac{k}{1+\tau s}\Delta U(s), \tag{2.1}$$

$$\Delta H_2(s) = \frac{\gamma}{1+\gamma\tau s}\Delta H_1(s), \tag{2.2}$$

where k, γ, and τ are model parameters that depend on the physical system properties, e.g., pump performance, tube friction, and outlet size. These parameters can significantly vary from system to system and, therefore, they need to be identified. In order to do that, we will follow the procedure explained in the following task.

Task 2.1

(a) In manual mode (choose the `Manual` tab), variate the control input u, until you find value that at steady-state takes the **lower-tank level** to 40 ± 1. Capture and save the figure. Write down such value (u_0) and the corresponding steady-state values of the upper and lower tanks, h_{10} and h_{20}.

We will use these values as equilibrium point for the linearized prediction model that we will use in the MPC.

(b) Change the input signal to $u_0 + 4$. Wait until the water tank levels reach steady-state. Capture and save the figure. Explain how we can use the result from the previous test to compute the system parameters k, γ and τ. If you need some help, you can refer to the documentation for Lab 2 (Tasks 2.7, 3.1, and 3.2) of the basic course in automatic control (available at `https://www.kth.se/social/upload/4fce1aa9f2765441d5000005/lab2_080818.pdf`).

Compute k, γ and τ.

2.2 Open-loop vs closed-loop MPC

In Task 1.5, we have compared the system response while using the optimal control input sequence computed in a single optimization with the system response while using MPC. We will run now the same comparison using the real double tank system.

Task 2.2

(a) We will now study the importance of using a good model in the MPC prediction. Repeat the experiment in **Task 1.5**, but this time on the real system. Press capture and save the figure. Reason about the results and compare them with your expectations. How is the open-loop and the close-loop MPC performance effected?

(b) Repeat the previous subtask but this time define the system parameters values u_0, h_{10}, h_{20}, k, γ, τ as identified and computed in the previous task. You can do this by pressing the button `Model_configure` and then `Model_save`. Press capture and save the figure. Compare with the performance in the previous task.

2.3 MPC with preview

MPC is extremely suitable for control problems where the reference or the constraints vary over time. If the future reference and constraints are known, they can be included in the optimization problem formulation. Imagine, for example, the case where we want to control a car driving along a sharp turn or avoiding an obstacle ahead. In this lab, we will track a sinusoidal reference.

Task 2.3

(a) How would you change your MPC formulation to include reference preview?

(b) First use an LQR (choose the `LQR` tab) to control the lower tank using a sinusoid as a reference (select `Sinusoid` in the `Reference_choice` switch). Set the reference value to 50, this defines the bias of the sinusoid. Press capture and save the figure. Describe the behaviour of the system. Is it satisfactory?

(c) Now, switch to the MPC controller (choose the `MPC` tab). Again, control the lower tank and choose a sinusoid as a reference. Set the reference value to 50. Press capture and save the figure. Did the sinusoid tracking improve?

(d) Let's now exploit the knowledge of the future reference trajectory by including it in the optimization problem. You can activate this in the MPC controller by turning `On` the `Reference_preview` switch. How does the tracking behave this time? Why?

2.4 MPC with disturbance compensation

Using the MPC in closed-loop makes it much more robust to model errors. However, no matter how well we identify the system, there will be always disturbance and noise affecting the performance of the controller.

Task 2.4

(a) How would you change your MPC formulation to include disturbance compensation?

(b) As in the previous task, control the lower tank using an MPC controller and a sinusoid as a reference. Turn `On` the `Disturbance_compensation` switch. Press capture and save the figure. Do you notice any improvement?

This is now time to report to the TAs your results of the experimental part. Call one of them if available...