EL2450: Hybrid and Embedded Control Systems: Homework 1

Introduction

The objective of this homework is to understand the basics of digital control including modelling, controller design and implementation. The process to be controlled is a double tank system, shown in Figure 1. The water is pumped into the upper tank by an electric pump. The water flows through a hole in the bottom of the upper tank to the lower tank. In the lower tank a similar hole is located letting the water flow back to the main reservoir. Let A be the cross-sectional areas of the tanks and a_1 and a_2 the effective outlet areas. Assume that the outgoing flow q from the pump is directly proportional to the voltage u applied to the motor. Let β be the proportionally coefficient. The water level of the upper and lower tanks can be described by the following differential equations:

$$\begin{aligned}
\frac{dh_1(t)}{dt} &= -\alpha_1\sqrt{2gh_1(t)} + \beta u(t) \\
\frac{dh_2(t)}{dt} &= \quad \alpha_1\sqrt{2gh_1(t)} - \alpha_2\sqrt{2gh_2(t)}
\end{aligned}$$

where h_1 is the level of the upper tank, h_2 the level of the lower tank, $\alpha_1 = a_1/A$ and $\alpha_2 = a_2/A$. In this homework the water level of the lower tank is to be controlled.

Exercises

Download the package `H1_src.zip` from the course homepage. You will edit the files during this homework. After you change a file you have to execute it for the changes to take effect.

Coupled tank model

1. Verify the model by opening the Simulink model `tanks.mdl`. Open the Two tanks block by double-clicking on it. Open the Tank 1 block. What is modelled by the gain Tap that has the value zero? **[2p]**

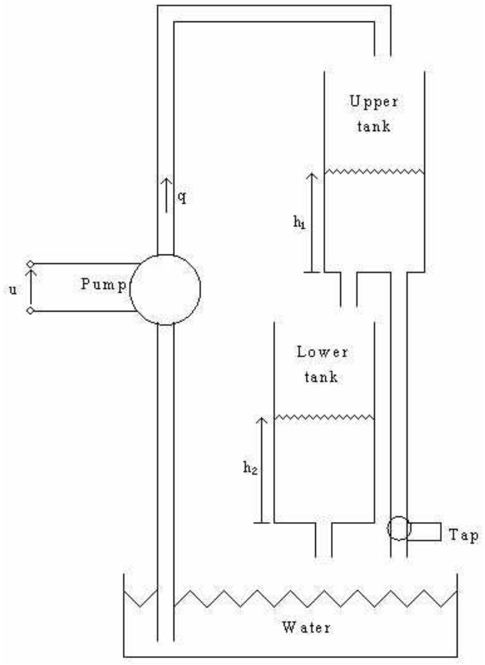

Figure 1: Coupled tanks with pump.

Continuous control design

The scaled and linearized system is described by the transfer function

$$\begin{aligned}\Delta X_1(s) &= \frac{k}{1+\tau s}\Delta U(s)\\ \Delta X_2(s) &= \frac{\gamma}{1+\gamma\tau s}\Delta X_1(s)\\ G(s) &= \Delta X_2(s)\end{aligned}$$

2. Edit the file `pid_design.m` and fill in the transfer functions for the upper and lower tank. **[1p]**

We should now try to control the system. The design method that we will use is pole-placement. We have the following step response requirements on the closed loop system:

- rise time less than 6s
- overshoot less than 35%
- settling-time less than 30s

The PID-controller to be used is given by

$$F(s) = K\left[1 + \frac{1}{T_I s} + \frac{T_D N s}{s+N}\right]$$

The closed loop system will have 4 poles. The poles will be placed using the closed-loop characteristic equation

$$(s+\chi)^2(s^2 + 2\zeta\omega_0 s + \omega_0^2) = 0$$

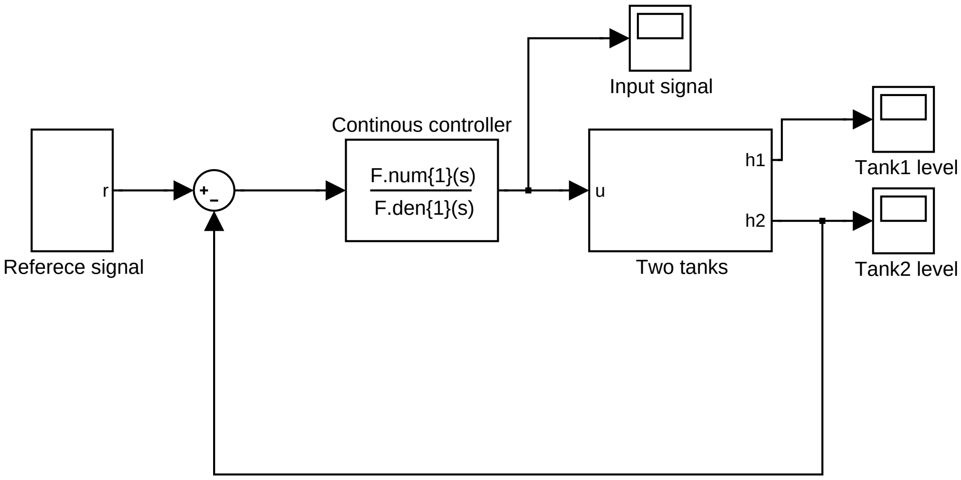

Figure 2: Simulink model of closed loop system.

The calculation of the PID parameters is implemented in the function `polePlacePID.m`. Open the file and make sure you understand how to use it.

3. Open the Simulink model by typing `tanks`. Also open the Simulink model `controller`. Move the continuous controller to the tanks model and connect them as shown in the Figure 2. Check the reference signal block and describe what the reference signal will look like. **[2p]**

4. Edit the file `pid_design.m`. Use the function `[K_pid,Ti,Td,N]=polePlacePID(chi,omega0,zeta,Tau,Gamma,K)`

   to derive the PID parameters. Also fill in the transfer function for the controller F. **[2p]**

χ	ζ	ω_0	T_r	M	$T_{settling}$
0.5	0.7	0.1			
0.5	0.7	0.2			
0.5	0.8	0.2			

5. Simulate the system for the different values of the parameters as specified in the table above. The first 100s of the simulation the system is initializing, this part can be neglected when evaluating the control performance. Which one of the parameter settings gives the best control performance? Please motivate your answer by the simulation results. **[2p]**

6. What is the cross over frequency of the open loop system? Please motivate how you derive it. **[1p]**

Digital control design

The controller you just designed will be implemented digitally. This system can be described as in the Figure 3. In this part of the homework, the A/D

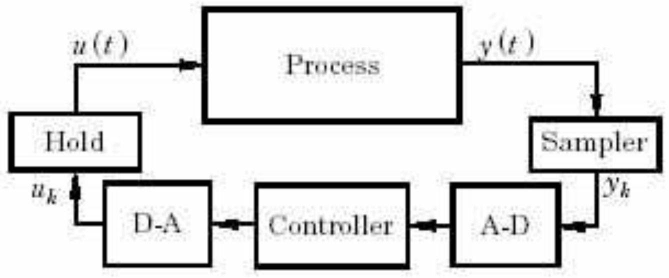

Figure 3: Model of digital control system.

and D/A converters can be neglected. This will be covered in the last part of the homework. In Simulink, the sampling block is included at the input connector of all discrete blocks. Hence no explicit sampler is needed when simulating the system with a discrete controller.

7. Open the Simulink model. Connect a zero-order hold block after the continuous controller and before the double-tank process. Try to change the sample time of the zero-order hold. What are the differences in the control performance, compared to the continues case? **[5p]**

8. Discretize the continuous controller into state space form (using MATLAB function `c2d`) under certain sampling time. Then move the discretized controller (from `controller.mdl`) into the loop, replace the continuous controller and remove the zero-order hold. Compare the simulation results with the previous case in Question 7 (under different sampling time). Are there any differences in control performance? Please motivate your answer by simulation results. **[5p]**

9. When implementing a continuous controller digitally, what sampling time should be used to keep the control performance? (*Hint*: Use the cross over frequency of the open loop system). **[2p]**

10. How long is the sampling interval possible without affecting control performance in Question 8? Compare with the calculated sampling time in Question 9 and comment on the possible differences. **[3p]**

Discrete control design

Now suppose that the sampling time is $T_s = 4s$.

11. Simulate the closed-loop system in Question 8 when the sampling time $T_s = 4s$. How is the control performance? **[1p]**

We will now try a different approach. We will derive a sampled model of the process for which we design a discrete controller. The sampled model should be on the form:

$$G_d(z) = \frac{a_1 z + a_2}{z^2 + b_1 z + b2}$$

The closed loop system should have the same performance as the continuous-time closed loop, i.e., the systems should have the same poles. The discrete-time system poles are located at $z_i = e^{T_s p_i}$, where p_i are the poles of the continuous-time system. The controller to be designed is on the form:

$$F_d(z) = \frac{c_o z^2 + c_1 z + c_2}{(z-1)(z+r)}$$

12. Sample the system G, with the sampling time $T_s = 4s$, using the MATLAB function `c2d`, with the zero-order hold method. Edit the file `pid_design.m` and save the sampled system in the parameter Gd. What are the coefficients a_i and b_i? **[3p]**

13. Where should the poles of a discrete time system be located for it to be stable? **[1p]**

14. Convert the poles for the continuous closed loop system to discrete-time poles. What are the corresponding poles for the discrete-time closed loop system? Also calculate the corresponding pole polynomial $z^4 + d_0 z^3 + d_1 z^2 + d_2 z + d_3$. (Hint: Use the Matlab functions `poly` and `minreal`.) **[3p]**

15. Now determine the controller parameters so that the closed loop system gets the desired poles. Calculate the pole polynomial of the closed loop system $(1+F_d G_d)^{-1} F_d G_d$, set it equal to the desired discrete pole polynomial and identify the coefficients. Show that this leads to the following linear equations: **[3p]**

$$\begin{bmatrix} 1 & a_1 & 0 & 0 \\ b_1 - 1 & a_2 & a_1 & 0 \\ b_2 - b_1 & 0 & a_2 & a_1 \\ -b_2 & 0 & 0 & a_2 \end{bmatrix} \begin{bmatrix} r \\ c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} d_0 - b_1 + 1 \\ d_1 - b_2 + b_1 \\ d_2 + b_2 \\ d_3 \end{bmatrix}.$$

16. Solve the above equations and save the discrete controller in the variable F_d. Verify that the poles of the discrete-time closed loop system are located where they should be. **[2p]**

17. Open the Simulink model and change to the discrete designed controller (from `controller.mdl`). Simulate the system again and compare the outcome with the previous results in Question 11. Conclusions? Please motivate your answer by simulation results. **[2p]**

Quantization

We will next investigate the effect of quantization. In a digital control system quantization appears in three different parts. When the signal is converted from analog to digital (A/D) the signal is quantized. The quantization level depends on the number of bits of the converter. In the control algorithm the output signal is computed. Here the size of the memory, used to store the signal value, contributes to the quantization. The less memory used the greater quantization. The conversion of the signal back to analog (D/A) gives the same effect.

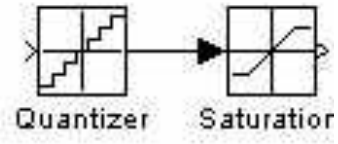

Figure 4: Quantizer with saturation.

18. Suppose you want to construct a A/D converter. The signal to be converted vary between 0 and 100. What will the quantization level be if the the number of bits used is 10? **[3p]**

19. In Simulink, the quantization block does not have any upper or lower limits. Try to construct a quantization block by connecting a quantizer with a saturation as shown in Figure 4. **[2p]**

20. Open the Simulink model with the discrete designed controller. Add your own quantization block before and after the controller, both with the same quantization level. This corresponds to a A/D and a D/A converter. Set the saturation for both blocks to -100 for the lower and 100 for the upper limit. Simulate the system for different values of the quantization level. For which quantization level will the control performance start to be degraded? How many bits does this correspond to? **[5p]**