Lab 3, Analysis and Design of Discrete Controllers

IE1304 Control Theory

1 Goal

The goal is to learn how to design a discrete controller, using pole placement. You will design the controller and analyze its characteristics (settling time, stability, overshoot, steady-state error).

2 The Process

The process to control is exactly the same as in lab 2 and is depicted in fig 1. It consists of two water tanks, the lower tank is filled from the upper tank, which in turn is filled by a pump. There is an outlet from the lower tank and, to introduce a disturbance, also the upper tank has an outlet. The measured value is the level of the lower tank.

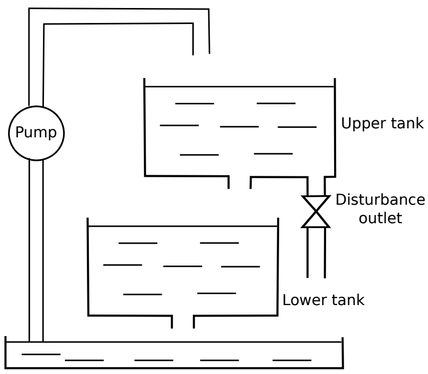

Figure 1: The controlled process

3 Preparation Tasks, to be solved BEFORE the lab

Task 1, Reading

- Read Chapter 19.1-19.5 in the course text book, and understand how a controller can be designed with pole placement.
- Read Section 4 in this tutorial and try to understand what to do during the lab.

Task 2, Controller Design

a) Pole Placement

In this lab we will treat the process as the second order model it actually is, $G_P = \frac{K_P}{(1+T_1 s)(1+T_2 s)}$. For now, you can set $K_P = 1$, $T_1 = 6$ and $T_2 = 21$. These values are close to the actual model values. Set the sampling time to 10% of the process' fastest (smallest) time constant and discretize the transfer function G_P. For discretization, use Matlab and the command `c2d`.

Recall the effects of different pole locations mentioned at lecture 11. Use the controller design described at lecture 11 and on page 351 in the text book which gives the transfer function from reference value to measured value, $H(z) = \frac{K_r B(z)}{A(z)C(z)+B(z)D(z)}$.

Place the poles in the area showed in the figure on page 355 in the course text book. The process is quite slow, therefore we must place the poles quite close to the unit circle or we will get too high values for control input, $u(k)$. Since both $A(z)$ and $B(z)$ is of second order the pole placement method described in the text book will tell that three poles are needed, you can place one of the poles in origo.

Calculate $C(z)$, $D(z)$ and K_r as described at lecture 11 and on page 353-354 in the text book. Use Matlab for the calculations, it will save you a lot of time.

- Define the model's transfer function.

```
s = tf('s')
GP = 1/(1+6*s)/(1+21*s)
```

- Discretize

```
HP=c2d(GP,0.6)
```

- Decide where the poles should be placed and simplify the pole placement equation, $A(z)C(z) + B(z)D(z) = P(z)$. This can not be done in Matlab, since Matlab can not do algebraic calculations. If we place the poles in $z = 0.7 \pm 0.2j$, the equation will be:
  $1 + (c1 + 0.001369 * d0 - 1.877) * z^{-1} + (-1.877 * c1 + 0.001312 * d0 + 0.001369 * d1 + 0.8794) * z^{-2} + (0.8794 * c1 + 0.001312 * d1)z^{-3} = 1 - 1.4z^{-1} + 0.53z^{-2}$

- Write the equations for each of the powers of z:

$$\begin{cases} c1 + 0.001369 * d0 - 1.877 = -1.4 \\ -1.877 * c1 + 0.001312 * d0 + 0.001369 * d1 + 0.8794 = 0.53 \\ 0.8794 * c1 + 0.001312 * d1 = 0 \end{cases}$$

- Solve for $c1$, $d0$, $d1$. This can be done in matlab as follows.

```
koeff = [1,0.001369,0;-1.877,0.001312,0.001369;0.8794,0,0.001312]
y = [-1.4+1.877;0.53-0.8794;0]
koeff \ y
```

The commands above solved the equation

$$\begin{bmatrix} 1 & 0.001369 & 0 \\ -1.877 & 0.001312 & 0.001369 \\ 0.8794 & 0 & 0.001312 \end{bmatrix} \begin{bmatrix} c1 \\ d0 \\ d1 \end{bmatrix} = \begin{bmatrix} -1.4 + 1.877 \\ 0.53 - 0.8794 \\ 0 \end{bmatrix}$$

which gives

$$\begin{cases} c1 = 0.2149 \\ d0 = 191.4 \\ d1 = -144.0 \end{cases}$$

- Finally, calculate $K_r = \frac{P(1)}{B(1)}$

```
Kr=(1-1.4+0.53)/(0.001369+0.001312)
```

This gives $K_r = 48.49$

b) Evaluation

Use Simulink to simulate the step response of the whole feedback loop (controller and process together). Remember that the control input is the pump voltage. A resonable value for the step is from 5V to 6V. Are the system properties (settling time, stability, overshoot and steady-state error) acceptable? Remember to check the step response also of the control input, $u(k)$. At the lab, the highest possible value of $u(k)$ is about 25V. The simulink simulation is showed in fig 2.

c) Integrating Controller

Improve your controller by introducing an integrator as described in section 19.4 in the text book. Again use Simulink to simulate the step response of $y(k)$ and $u(k)$. Also again evaluate the system's settling time, stability, overshoot and steady-state error.

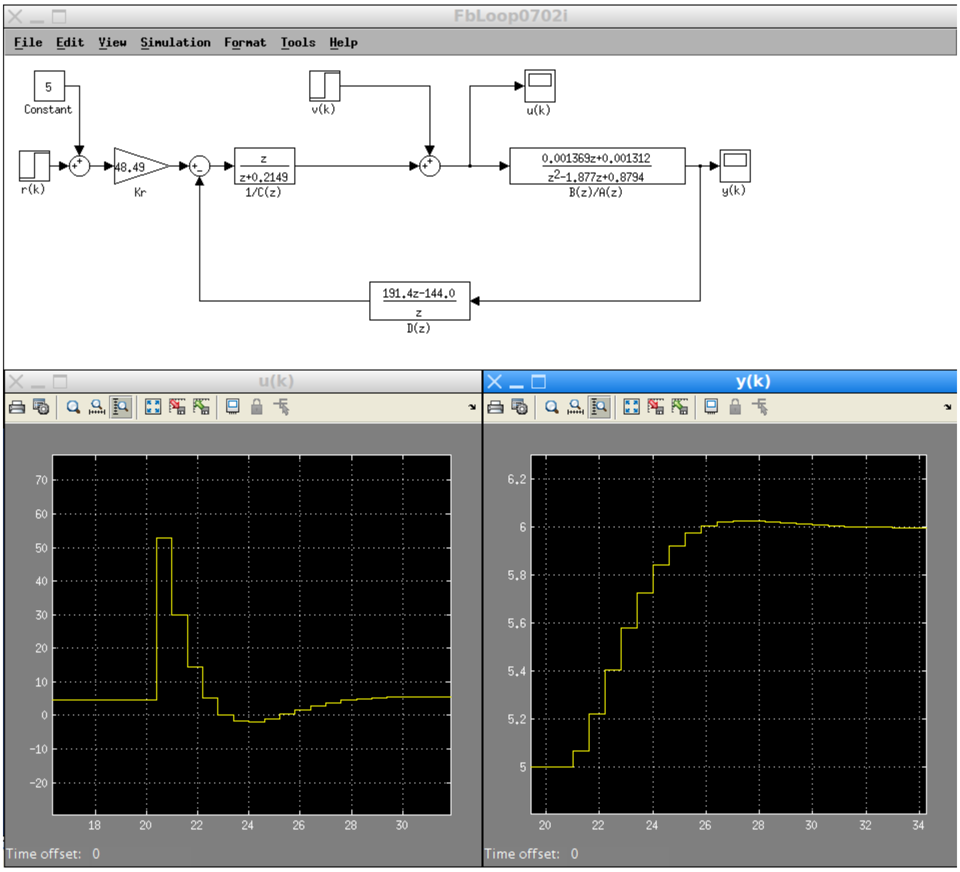

Figure 2: Simulation of reference value step from 5 to 6 at t = 20s. There is no disturbance, $v(k) = 0$. $HP = \frac{0.001369z^{-1}+0.001312z^{-2}}{1-1.877z^{-1}+0.8794z^{-2}}$, poles in $z = 0.7 \pm 0.2i$

Task 3, Controller Implementation

The controller will be the same PLC system that was used during lab 2. Implement the controller you designed in the previous task in a program in ISaGRAF. Use the program in fig 3, which is complete apart from the control law in step GS3.

4 Lab Tasks, to be solved at the lab

Task 1, Process Transfer Function

- Set up the lab equipment the same way you did in lab 2, see the tutorial for lab 2 for details.
- Decide K_p, T_1 and T_2 of the process. How to do this is described on pages 112-113 in the text book.
- Design a pole placement controller the same way you did in preparation task 2.

Task 2, Analysis of Non-Integrating Controller

a) Download your controller program to Smart I/O and set the reference value to for example 5000. Let the system stabilize and then make a small change in reference value, for example to 5500. Plot the step response of the system using Fluke View and measure settling time, stability, overshoot and steady-state error of the process when using your controller.

b) Measure settling time, stability, overshoot and steady-state error of the process when using your controller and introducing a disturbance.

c) Does the system behave as expected? If it does not, why do you think it behaves differently?

d) If your system has a steady-state error, try to eliminate it by adjusting the controller gain, K_r.

Task 3, Analysis of Integrating Controller

If time allows, evaluate the integrating controller the same way you evaluated the non-integrating controller in Task 2.

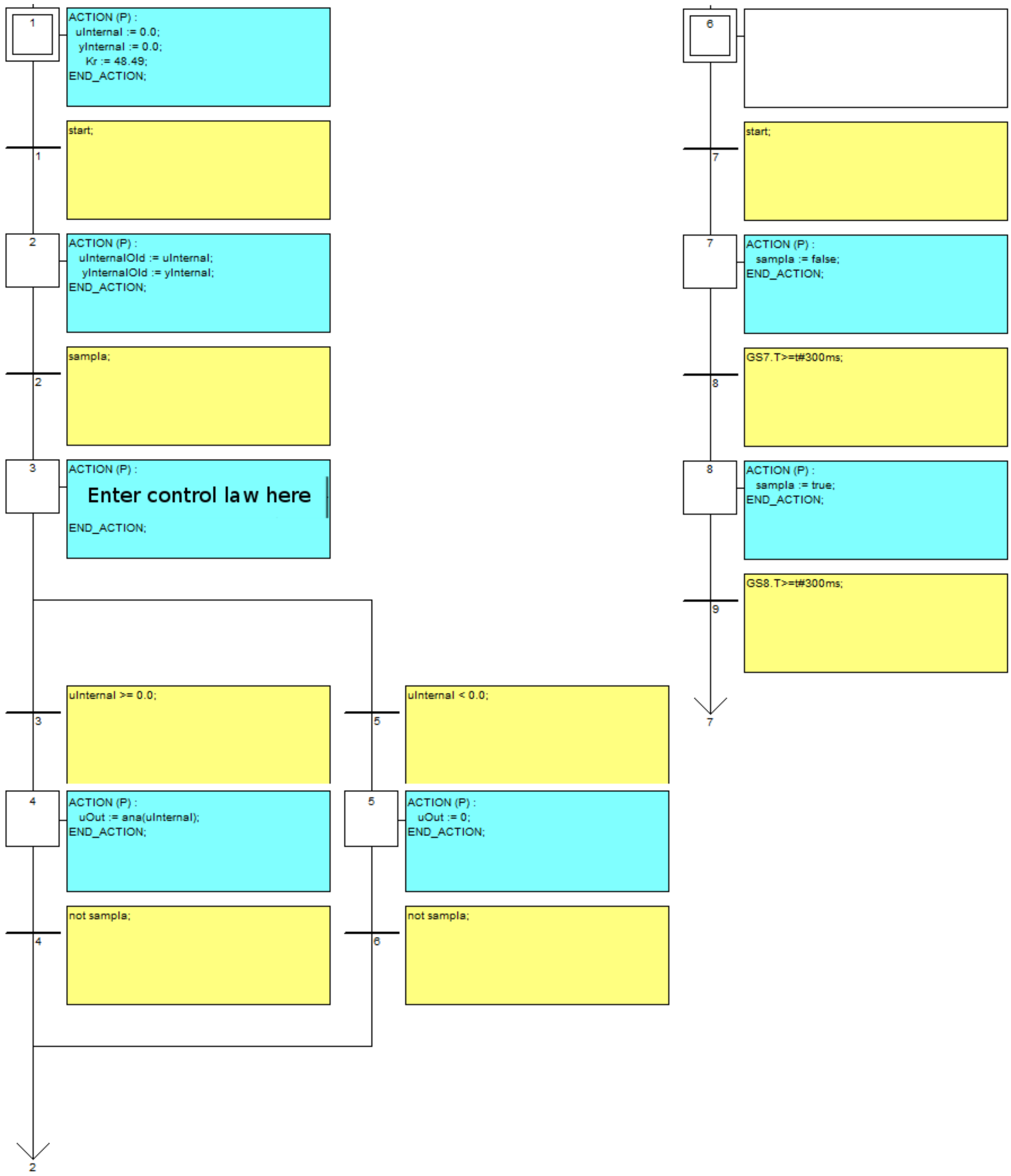

Figure 3: Pole placement controller program