ELEC ENG 4CL4 – CONTROL SYSTEM DESIGN

Lab #3: PID tuning in MATLAB/Simulink

Objectives:

To gain experience in applying empirical methods for determining PID controller parameters within MATLAB/Simulink.

Assessment:

Your grade for this lab will be based on your ability to create Simulink models and MATLAB code to apply empirical methods for determining PID controller parameters for the plant model described below and on your reporting of the results. The report should contain the results of your pre-lab derivations and calculations, schematics of your Simulink models with brief descriptions, MATLAB plots of results with brief descriptions, and answers to specific questions below.

The total grade will be out of 20 points. The written report will be worth 15 points, and the remaining 5 points will be based on your demonstration of your Simulink model and MATLAB code to the TA and/or a pre-lab quiz on the theory required for this lab.

Clearly label all plots and their axes (points for style will be deducted otherwise)

DON'T COPY OTHERS BLINDLY!!

Please attend the lab section to which you have been assigned.

You should complete this lab with one lab partner. If there are an odd number of students, then one group of three will be created by the TA, or you can choose to work on your own if a computer is available.

You may choose to complete the lab assignment partially or entirely in your own time (in groups preferably of two students but definitely no larger than three). However, if you choose to do this, you must show up at the start of your scheduled lab time to give the TA a brief demonstration of your MATLAB/Simulink code and model.

Each pair of students should complete one lab report together, which is to be submitted <u>one week</u> from the date of the lab.

Nominal plant model description:

A plant can be described by the nominal model:

$$G_o(s) = \frac{\mathrm{e}^{-s}}{s^2 + 9s + 3},$$

which includes a one-second signal delay (indicated by the term e^{-s}), complicating the synthesis of a suitable controller[1]. Consequently, this plant may be suitable for the application of empirical methods for tuning PID controllers. The plant is open-loop stable, and we will assume that it is safe and practical to drive the plant to the point of critical stability, so we will investigate all three PID tuning methods described in Chapter 6 of Goodwin et al.: the *Ziegler-Nichols Oscillation Method* and both of the *Reaction Curve Based Methods*. In the latter case, we will compare the results obtained with the PID parameters suggested by Ziegler-Nichols to the results obtained with the Cohen-Coon parameters.

---

[1] A Smith predictor, as described in Chapter 7 of Goodwin et al., could be used to eliminate the plant signal delay in the controller synthesis procedure, but we will not make use of that option in this lab.

 Continued on page 2

Pre-lab:

- You should re-read the notes for Lectures #13–15 before your lab session and bring them with you to the lab—you will need to refer to several equations and tables from the notes.
- You should also think about how to implement the PID controllers (of type **P**, **PI** and **PID**) in Simulink. One possibility is to work out the transfer function forms (i.e., the numerator and denominator polynomials) as I did on the board in class. Another is to implement them as sets of connected basic blocks, i.e., gain, integration, approximate derivation.
- You should derive an equation for the maximum slope tangent (*m.s.t.*) of the process reaction curve $y(t)$ for Section 2, part a of the lab, along with the resulting parameters t_0, t_1, t_2, u_0, u_∞, y_0 and y_∞.. The process reaction curve can be found by calculating the open-loop step response $y_1(t)$ of the nominal plant model *without* the delay term e^{-s}, i.e., $G_1(s)=\dfrac{1}{s^2+9s+3}$ and $Y_1(s)=G_1(s)R(s)$, and then adding the known time delay of 1 second to give $y(t)=y_1(t-1)\mu(t-1)$, where $\mu(t)$ is the unit step function.

1. Ziegler-Nichols oscillation method

a. In Simulink, place the nominal plant model given above in a unity-feedback loop with a proportional controller.

b. Increase the controller gain until the point of critical stability is reached, i.e., until a sustained oscillation is generated at the *controller output* $u(t)$. Record the controller critical gain K_c and the oscillation period P_c of the *controller output*, and save the MATLAB plot that you used to determine these parameters.

c. From Table 6.1 of Goodwin et al., calculate the values of K_p, T_r and T_d for each of the three controller types given: **P**, **PI** and **PID**. Let $\tau_D = 0.1T_d$.

d. Using the parameters obtained in part c, simulate the response of a unity-feedback loop with each of the controller types (**P**, **PI** and **PID**) to (i) a unit step reference and (ii) a sinusoidal reference $r(t)=\sin(1.5t)$. Compare the results for the three different controller types, and explain the differences in terms of the **P** action, the **I** action and the **D** action.

e. Demonstrate what the step response would be with the **PID** type controller with parameters from part c if the *true plant* actually had a signal delay of 2 seconds, rather than the 1-second delay included in the nominal model.

 Continued on page 3

2. Reaction curve based methods

a. With the nominal plant model in *open loop*, apply a unit step input to the plant. (Don't forget to return the plant signal delay to 1 second!)

b. Plot the resulting *process reaction curve*, and find the maximum slope tangent (*m.s.t.*). You should be able to work out how to determine the *m.s.t.* from the numerical values of the plant output. If you cannot get this to work, you can try to determine the *m.s.t.* graphically, but it will not be as accurate. Does the *m.s.t.* calculated here agree with what you derived in the pre-lab?

c. Calculate the process reaction curve parameters t_0, t_1, t_2, u_0, u_∞, y_0 and y_∞. Again, you should be able to determine all of these from the numerical values of the plant output and the *m.s.t.* obtained in part b. If you cannot get this to work, you can try to determine these parameters graphically, but they will not be as accurate. Do the values calculated here agree with what you derived in the pre-lab?

d. Compute the parameter model:

$$K_0 = \frac{y_\infty - y_0}{u_\infty - u_0}; \qquad \tau_0 = t_1 - t_0; \qquad \nu_0 = t_2 - t_1 .$$

e. Determine the Ziegler-Nichols PID controller parameters (just for the **PID** type) from Table 6.2 of Goodwin et al. Let $\tau_D = 0.1T_d$.

f. Using the parameters obtained in part e, simulate the response of a unity-feedback loop with the controller type **PID** to (i) a unit step reference and (ii) a sinusoidal reference $r(t) = \sin(1.5t)$.

g. Determine the Cohen-Coon PID controller parameters (just for the **PID** type) from Table 6.3 of Goodwin et al. Let $\tau_D = 0.1T_d$.

h. Using the parameters obtained in part g, simulate the response of a unity-feedback loop with the controller type **PID** to (i) a unit step reference and (ii) a sinusoidal reference $r(t) = \sin(1.5t)$.

i. Compare the results obtained with the Cohen-Coon reaction curve method to those obtained with the Ziegler-Nichols reaction curve method. How do each of these compare with the results obtained with the Ziegler-Nichols oscillation method (in section 1 above)?

j. Consider the situation where a non-minimum-phase zero is added to the nominal plant model, such that the transfer function is now:

$$G_o(s) = \frac{\mathrm{e}^{-s}(-s+2)}{s^2 + 9s + 3}.$$

k. Compute new *Cohen-Coon* PID parameters (just for the **PID** type) from the process reaction curve of this new plant model.

l. Using the parameters obtained in part k, simulate the response of a unity-feedback loop with the controller type **PID** to a unit step reference. You should find that in this case the Cohen-Coon method has produced an unstable closed-loop system! By how much do you need to reduce the proportional gain K_p in order to stabilize the system?