CHE4400: Process Control
Extra Credit HW: PID Controller Tuning

Instructor: Jay H. Lee
Due: Thursday, April 17, 2003

This whole exercise is worth up to 5.0% point of the overall grade. The simulation should be done using MATLAB and SIMULINK. **The results must be submitted in a report format and should include all the simulation files and plots**. You may work in pair if you would like and share the credit (according to the percentage distribution of the labor you agree on).

Important: Using someone else's simulation results in preparing your own report constitutes an act of academic dishonesty. When such a violation is perceived, the relevant materials will have to be turned over to the appropriate university officer.

Problem 1

Consider the following third-order system with time-delay:

$$\tilde{G}_p = \frac{(1-2s)}{(3s+1)(0.5s+1)(2s+1)}e^{-10s}$$

For the above system:

- Perform a step response test and fit a **first-order-plus-time-delay** (FOPTD) model.
- Compare the output responses of the process $\tilde{G}_p$ and the FOPTD model, for a unit step change in the input.
- Compare the **Bode plots** for the above process $\tilde{G}_p$ and the FOPTD approximation you obtained from the step test. How do they match at the *critical frequency* of $\tilde{G}_p$, ω_c ?

  Plot the Bode diagrams using MATLAB tools.
- Using the FOPTD model parameters obtained, compute PID controller parameters from **Cohen-Coon Design Relations**. Implement the controller and simulate the closed-loop response under a step setpoint change and a step disturbance entering the output (both of size 1).

- Use the **Ziegler-Nichols Tuning Method**. Find the ultimate gain and the ultimate period from *the Bode plot of the obtained FOPTD model* and compute PID controller parameters using the Ziegler-Nichols tuning table. Implement the controller and simulate the closed-loop response under a step setpoint change and a step disturbance entering the output (both of size 1).

- Use the **IMC Method**. Find the PID parameters based on the obtained FOPTD model and desired time constant λ. Vary λ and examine its effect. What is the most appropriate value? How did you arrive at this value?

- Implement a **Smith Predictor** and use the same PID tuning parameters you got from Cohen-Coon design. Simulate the performance as before.

Problem 2

Consider again the same system as Problem 1:

$$\tilde{G}_p = \frac{(1-2s)}{(3s+1)(0.5s+1)(2s+1)}e^{-10s}$$

For the above system:

- Implement a P-only controller. Now by **Trial and Error Tuning**, determine the *ultimate gain*, $\mathbf{K_u}$ and *ultimate period*, $\mathbf{P_u}$. Try K_c values between 1.0 and 1.5.

- Use the **Relay Controller** (with an appropriately chosen deadzone) to obtain the *ultimate gain*, $\mathbf{K_u}$ and *ultimate period*, $\mathbf{P_u}$. **Relay Controller** can be found in the *Nonlinear* library of SIMULINK. Obtain the data and read the **relay amplitude**, d, and the **amplitude of the process oscillation**, a. Refer to your class notes (or your textbook) for details on the relay controller and required formulae to compute K_u and P_u.

- Obtain the *real values* for the *ultimate gain*, $\mathbf{K_u}$ and *ultimate period*, $\mathbf{P_u}$ of the process, $\tilde{G}_p$, from its bode diagram. Compare with the values obtained from the *trial and error* tuning and the *relay controller*. How do the values from the three methods compare?

- From the knowledge of *ultimate gain*, $\mathbf{K_u}$ and *ultimate period*, $\mathbf{P_u}$, obtained from either of the experiments, determine the **Ziegler-Nichols** PID tuning parameters. Implement the controller and verify the closed-loop response under a step setpoint change and a step disturbance entering the output (both of size 1).

- From the knowledge of *ultimate gain*, $\mathbf{K_u}$ and *ultimate period*, $\mathbf{P_u}$, determine the **Modified Ziegler-Nichols** PID tuning parameters (which can be found in your classnotes). Implement the controller and verify the closed-loop response under a step setpoint change and a step disturbance entering the output (both of size 1).
- Implement a **Smith Predictor** and use the same PID tuning parameters you got from Ziegler-Nichols design. Simulate the performance as before.

Problem 3

- Give a qualitative/comparative discussion of the simulation results obtained in problems 1 and 2. Did one approach work better than the other? Give your reasoning.
- What is the effect of the Smith Predictor?

Note: (About the PID Implementation in SIMULINK)
The implementation of PID controller in the SIMULINK library is different from the classical PID algorithm we learned in the class. The difference is in how the specified parameters are utilized in the computation of the input.

The classical PID algorithm:

$$u(s) = K_c(1 + \frac{1}{\tau_I s} + \tau_D s)e(s)$$

The SIMULINK PID algorithm:

$$u(s) = (P + \frac{I}{s} + Ds)e(s)$$

where P, I and D are the values you will be entering. When you specify parameters you should keep this in mind. The proportional gain is same in both cases ($P = K_c$). But for the integral term, I, you should be entering $I = \frac{K_c}{\tau_I}$ and for the derivative term, D, you should be entering $D = K_c\tau_D$.

An alternative is to use the PID block from the lab (PIDGT). This block takes K_c, τ_I and τ_D as the inputs.