Lab 3: Basic Control Actions

Introduction

The proportional, PI, and PID controllers are among the most basic of feedback controllers. They are easily implemented and perform effectively in light of plant uncertainties. These controllers are widely used in industry for applications in motion control and power systems. The PID controllers are a valuable pedagogical tool as it requires an intuitive understanding of the feedback mechanism, which serves as the basis for most material in classical control.

This laboratory exercise will focus on the hardware implementation and physical effects of basic feedback controllers. Additionally, the individual effects of the proportional, integral, and derivative contributions to the control effort will be observed. An intuitive feel for P, PI, and PID controllers will be developed. Mathematical background for the three types of controllers will not be covered.

Objectives

Required:

1. Identify each portion of the unity feedback closed-loop control system (Figure 2 of the Prelab). Match each portion to the corresponding component in the physical system.
2. Tune proportional, integral, and derivative gains to achieve a desirable output response.
3. Interpret the individual contributions of the proportional, integral, and derivative terms.

Experiment 1: Speed Regulation of the Motor using P Control

Tune the proportional motor controller and plot motor speed vs. time for two step reference speeds.

Software Setup:

1. Recall the usage of the Simulink software. If you need to refresh, refer to the Lab 2 manual. Refer to the 'B&R Code Generation Guide' to remember how to build and run your model.
2. Start MATLAB and then Simulink. Copy the file 'N:\HydralulicsLab\ME460\Lab3_MotorPID.slx' and place it in your own working directory 'C:\ME460_SPxx\ABx\Lab3\Motor'.
3. Lab3_MotorPID.slx is an incomplete block diagram. Complete the closed-loop system with a proportional controller. Set the k_p gain to 1 initially. Do not modify the conversion blocks.
4. Set the reference speed to be a step of 500 RPM and a duration of 5 seconds using the *Custom Step Input* block. (Leave the start time and initial value as zero).
5. Find the scope that plots both the reference speed and measured speed.
    a. Confirm the scope's time span is 15 seconds and the y-axis ranges from -50 RPM to 800 RPM by viewing the scope's settings, .

b. Note this scope is logging data to the MATLAB workspace, identify the variables that are logged to the workspace.

Hardware Setup:

1. Connect the hydraulic hoses. Use the BD90 valve as the actuator and the motor as the plant.
2. Connect the signal wiring. Connect the MW2000's Analog Output Ch.1 to the BD90 valve driver box using a BNC cable. Connect the MW2000's Analog Input Ch. 7 (Tach) to the tachometer's feedback signal labelled "Tachometer" on the sensor box.
3. Have your instructor check the wiring, block diagram and hose connections.
4. Power on the valve driver box, the feedback sensor box, the MW2000, and the B&R control rack.
5. Start the hydraulic trainer following the steps on the whiteboard. Set the pressure to 300 psi.

Run Controller:

1. Build, connect, and run your input command from your Simulink model.
2. Observe the difference between the output speed and the reference speed at steady state. Record the error.
3. Without disconnecting from target, change the k_p gain from 1.0 to 1.5. Again, record the error. Do the same for a gain of 2.0.
4. Calculate percentages of the relative error. Make a table to track gain, reference speed, error magnitude, and error percentage.
5. Change the reference speed from 500 to 700 rpm. Repeat steps 2 through 4 for all three gains.
6. Make sure to *Disconnect from Target* when completed. Leave the hydraulic trainer running.

Experiment 2: Speed Regulation of Motor using PI Control

Software Setup:

1. Use the same model 'Lab3_MotorPID.slx'. Modify your model to create a proportional-integral controller. You must use the provided *Custom Integrator* block that is commented out. Set the proportional gain to 1.0 and integral gain to 0.0 initially.
2. Set the reference speed to be a step of 500 RPM.

Hardware Setup:

3. Same setup as Experiment 1.

Run Controller:

1. Build, connect, and run your input command from your Simulink model.
2. Observe the plant's response for your controller.
3. While remaining connected to the target, re-run the step input command for an integral gain of 10.0 and 25.0 and again observe the response.

4. *Disconnect from Target* and shut down the hydraulics.

Experiment 3: Position Regulation of Cylinder using PI Control

A PI controller will now be used on a different plant, the hydraulic cylinder, to further demonstrate the proportional and integral effects.

Software Setup:

1. Copy the file 'N:\HydralulicsLab\ME460\Lab3_CylinderPID.slx' and place it in your own working directory 'C:\ME460_SPxx\ABx\ Lab3\Cylinder'.
2. Leave the k_i and k_d gains zero and set the k_p gain to 3.0.
3. Set the reference position to be a step of 2" for a duration of 5 seconds with the *Custom Step Input* block. Leave the start time and initial position as zero.
4. Use the manual switch to change the input to a square wave to observe a continuous response.

Hardware Setup:

1. Disconnect the hoses from the motor and connect them to the cylinder. **Note!! Value pot *C1* must go to the top cylinder port, while port *C2* must go to the bottom cylinder port.**
2. Change the signal wiring. Connect the MW2000's Analog Output Ch.2 to the valve driver box using the BNC cable. Connect the MW2000's Analog Input Ch.6 to the string potentiometer feedback label "Linear Pot." on the sensor box.
3. Have your instructor check the wiring, block diagram and hose connections.
4. Start the hydraulic trainer following the steps on the whiteboard. Set the pressure to 300 psi.

Run Controller:

1. Build, connect, and run your input command from your Simulink model.
2. Observe the plant's response to your controller.
3. While remaining connected to the target, change the proportional gain to a higher value such as 10.0. Observe the response.
4. Repeat the two proportional gains with integral gains of 3.0 and 5.0. It may be useful to increase the step duration or square wave period.
5. Try to develop a feel for what integral control does for the system.
6. *Halt* your command input and *Disconnect from Target* when completed, but leave the hydraulic trainer running.

Experiment 4: Position Regulation of Cylinder using PID Control

Software Setup:

1. Use the same model 'Lab3_CylinderPID.slx'.

Hardware Setup:

1. Same setup as Experiment 3.

Run Controller:

1. Set the proportional gain to 10.0. Set the integral gain to 0.5. Set the derivative gain to 0.0.
2. Connect and run your input command from your Simulink model.
3. Now, set the derivative gain to 2.0 leaving the proportional and integral gains the same. Observe the new response.
4. Note the difference between the full PID controller versus the PI controller in the last experiment.
5. *Halt* your command input and *Disconnect from Target*.
6. Power off all electrical equipment and shutdown the hydraulic trainer.

Report Questions

1. Relate all blocks and connections labeled in Prelab Figure 2 to the physical system implementation for Speed Regulation of the Motor using PI Control (Experiment 2).
2. For proportional control of motor speed, a steady state error between measured speed and the reference speed $R(s)$ was observed. Was it possible to decrease this error? Why does this error exist? Is it possible to remove this error completely using just proportional control?
3. Is it possible to remove steady state error for position regulation of the cylinder using only proportional control?
4. What effect does integral control have on the plant's step response and steady state error? Consider both the motor and cylinder systems.
5. What effect does derivative control have? Why is it necessary to filter the measured voltage signal before taking its derivative? When is the derivative gain useful?

References

[1]. Ogata, Katsuhiko. Modern Control Engineering, 5th ed. Prentice Hall, New Jersey: 2009.

[2]. Horowitz & Hill. The Art of Electronics 2nd ed. Cambridge University Press, MA: 1989.