Control Systems Laboratory (EE 3321) — Experiment 5

MATLAB, and Simulink Control System Configurations and Controller Design

Overview of Experimental Procedure

This experiment utilizes MATLAB, and Simulink to construct series, parallel, and/or feedback systems of transfer functions. Additionally, the concepts of a proportional gain controller, PID controller are introduced and the effects of the controllers are observed.

System Configuration Theory

System transfer function models can be interconnected in one of three ways: series, parallel, or feedback. A series configuration is shown in Figure 5.1, and is simplified to a transfer function of *G(s)H(s)*.

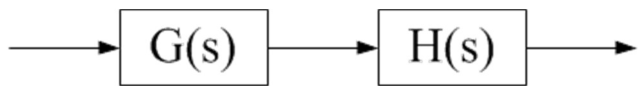

FIGURE 5.1: SERIES CONFIGURATION

A parallel configuration is shown in Figure 5.2, and is simplified to a transfer function of *G(s) + H(s)*.

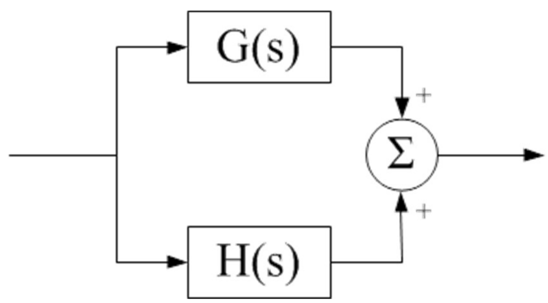

FIGURE 5.2: PARALLEL CONFIGURATION

A feedback configuration is shown in Figure 5.3, and is simplified to a transfer function *G(s) / [1 + G(s)H(s)]*.

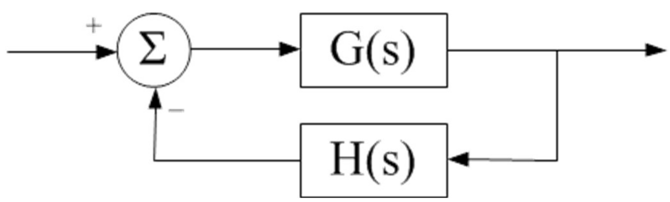

FIGURE 5.3: NEGATIVE FEEDBACK CONFIGURATION

To arrange a transfer function into one of the aforementioned configurations in MATLAB, use the series(), parallel(), or feedback() functions. Each function requires two transfer function systems as inputs to the function, and will yield the simplified transfer function as an output.

Proportional Gain Controller Theory

A proportional gain controller uses the error of the system as an input and drives the system that it is controlling based upon the proportional gain constant of the controller. The system error is the difference between the system reference and the actual system output. As we will see in this experiment, varying the proportional gain of a control system can drastically affect the system performance.

PID Controller Theory

PID is an acronym for Proportional-Integral-Differential. A PID controller is a control loop feedback controller widely used in industrial control systems. PID controllers provide the excitation for the plant; the gains K_P, K_I, and K_D of the controller are set to control the overall system behavior. The variable "e" shown on the system block diagram represents the **tracking error**, or the difference between the desired output and the actual output. This error signal will be sent to the PID controller; the controller then computes a signal that is the sum of the signals directly proportional to, proportional to the integral of, and proportional to the derivative of this error signal. This signal, which is the output of the PID controller, is fed to the plant, which generates its own output (oftentimes the system output), and a new error is calculated, which, again, is sent to the PID controller and the process is repeated. The diagram of a general system with a PID controller is presented below.

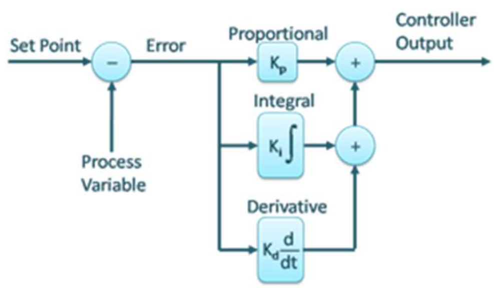

FIGURE 5.6: PID CONTROLLER CONFIGURATION

A proportional controller (with gain K_P) will have the effect of reducing the rise time and steady-state error. A derivative controller (with gain K_D) will have the effect of increasing the stability of the system, reducing the overshoot, and improving the transient response. Finally, an integral controller (with gain K_I) will have the effect of eliminating the steady-state error, reducing rise time, but at the same time increasing settling time.

Experimental Procedure

1. The transformed block diagram representation for a submarine depth control system is shown below. In the block diagram, *C(s)* is the transform of the commanded depth signal, and *D(s)* is the transform of the signal representing the submarine's actual depth.

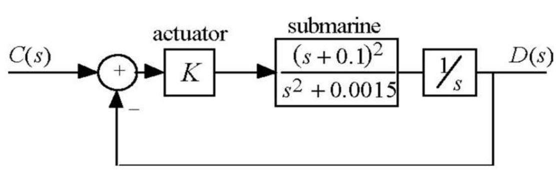

FIGURE 5.10: SUBMARINE DEPTH CONTROL SYSTEM BLOCK DIAGRAM

A quick 50 foot dive commanded at $t = 0$ can be modeled as $c(t) = -50\, u(t)$. Using MATLAB, **find the resulting submarine depth** as a function of time for $K = 0.4$, $K = 0.08$, and $K = 0.04$. Use a time scale interval of 0 to 180 seconds. Note the significant characteristic differences between the submarine's behaviors for the values of K. Be sure to include the command signal in each graph for reference.

2. Complete procedure 1 using MATLAB Simulink. Compare the results of each method.
3. Develop a proportional gain controller for an armature-controlled DC motor (see experiments 3 and 4 for reference) that controls the speed of the motor by applying a voltage to the system from the controller. Vary the proportional gain constant, and note the effect. Complete this task in both MATLAB and MATLAB Simulink.

4. Using the **Simulink** file constructed in part 3, build a PID controller for the armature-controlled DC motor. Note the effect that varying each of the proportional, integral, and derivative terms has on the system response.

Conclusion

Describe the effect of varying the gain constants in a control system. Be sure to be thorough in your explanation, and note whether or not you encountered any unstable system behavior as a result of this controller.