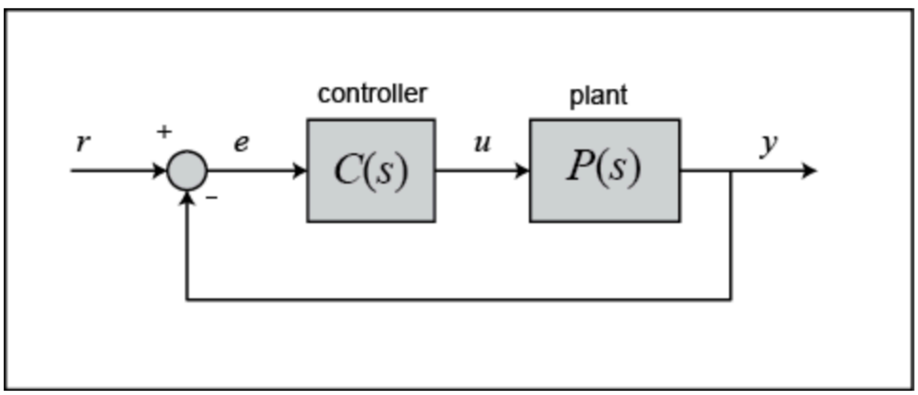
ECE 327: Lecture 5

PID Controller Design

In this lecture we will introduce the Proportional-Integral-Derivative (PID) controller. The PID controller is widely employed because it is very understandable and because it is quite effective. One attraction of the PID controller is that all engineers understand conceptually differentiation and integration, so they can implement the control system even without a deep understanding of control theory. Further, even though the compensator is simple, it is quite sophisticated in that it captures the history of the system (through integration) and anticipates the future behavior of the system (through differentiation). In this laboratory, we will discuss the effect of each of the PID parameters on the dynamics of a closed-loop system and will show how to use a PID controller to improve a system's performance.

PID Overview

Consider the following unity-feedback system:



The output of a PID controller, which is equal to the control input to the plant, is calculated in the time domain from the feedback error as follows:

$$u(t) = k_p e(t) + k_i \int e(t)dt + k_d \frac{de}{dt}.$$

First, let's take a look at how the PID controller works in a closed-loop system using the schematic shown above. The variable (e) represents the tracking error, the difference between the desired output (r) and the actual output (y). This error signal (e) is fed to the PID controller, and the controller computes both the derivative and the integral of this error signal with respect to time. The control signal (u) to the plant is equal to the proportional gain (k_p) times the magnitude of the error plus the integral gain (k_i) times the integral of the error plus the derivative gain (k_d) times the derivative of the error.

This control signal (u) is fed to the plant and the new output (y) is obtained. The new output (y) is then fed back and compared to the reference to find the new error signal (e). The controller takes this new error signal and computes an update of the control input. This process continues while the controller is in effect.

The transfer function of a PID controller is found by taking the Laplace transform of the above equation.

$$k_p + \frac{k_i}{s} + k_d s = \frac{k_d s^2 + k_p s + k_i}{s}$$

where k_p is the proportional gain, k_i is the integral gain, and k_d is the derivative gain.

The Characteristics of the P, I, and D Terms

Increasing the proportional gain (k_p) has the effect of proportionally increasing the control signal for the same level of error. The fact that the controller will "push" harder for a given level of error tends to cause the closed-loop system to react more quickly, but also to overshoot more. Another effect of increasing (k_p) is that it tends to reduce, but not eliminate, the **steady-state error**.

The addition of a derivative term to the controller (k_d) adds the ability of the controller to "anticipate" error. With simple proportional control, if $(k_p)$is fixed, the only way that the control will increase is if the error increases. With derivative control, the control signal can become large if the error begins sloping upward, even while the magnitude of the error is still relatively small. This anticipation tends to add damping to the system, thereby decreasing overshoot. The addition of a derivative term, however, has no effect on the steady-state error.

The addition of an integral term to the controller (k_i) tends to help reduce steady-state error. If there is a persistent, steady error, the integrator builds and builds, thereby increasing the control signal and driving the error down. A drawback of the integral term, however, is that it can make the system more sluggish (and oscillatory) since when the error signal changes sign, it may take a while for the integrator to "unwind."

The general effects of each controller parameter (k_p, k_d, k_i) on a closed-loop system are summarized in the table below. Note, these guidelines hold in many cases, but not all. If you truly want to know the effect of tuning the individual gains, you will have to do more analysis, or will have to perform testing on the actual system.

CL RESPONSE	RISE TIME	OVERSHOOT	SETTLING TIME	S-S ERROR
Kp	Decrease	Increase	Small Change	Decrease
Ki	Decrease	Increase	Increase	Decrease
Kd	Small Change	Decrease	Decrease	No Change

General Tips for Designing a PID Controller

When you are designing a PID controller for a given system, follow the steps shown below to obtain a desired response.

1. Obtain an open-loop response and determine what needs to be improved
2. Add a proportional control to improve the rise time
3. Add a derivative control to reduce the overshoot
4. Add an integral control to reduce the steady-state error
5. Adjust each of the gains k_p, k_i and k_d until you obtain a desired overall response.

Lastly, keep in mind that you do not need to implement all three controllers (proportional, derivative, and integral) into a single system, if not necessary. For example, if a PI controller meets the given requirements, then you don't need to implement a derivative controller on the system. Keep the controller as simple as possible.