ECSE 307 Linear Systems and Control
Lab 4 Report

Hai XU, 260661832 Wenjie WEI, 260685967

October 25^{th}, 2017

1 Introduction

THIS lab introduces the methodologies to analyze the behavior of a dynamic system. Proportional, Integral, and Derivative (PID) controls are so far the most common type of controllers that are used. They are simple yet still able to give promising performance. Figure 1 shows what will happen if a PID controller is implemented.

Parameter	Rise Time	Overshoot	Settling Time	e_{ss}
k_P	Decrease	Increase	Small Change	Decrease
k_I	Decrease	Increase	Increase	Eliminate
k_D	Small Change	Decrease	Decrease	No Change

Figure 1: Key Changes that PID Systems Can Bring

2 Finding the PID Gains

Consider a system transfer function as below:

$$G(s) = \frac{1}{(s+1)(s+2)(s+3)}$$

Plot the step response using MATLAB. The step response of the system is shown in Figure 2:

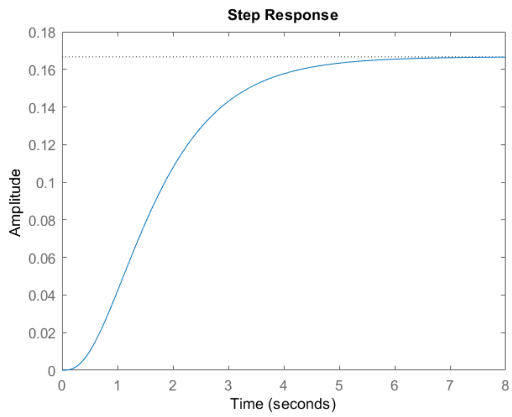

Figure 2: Step Response of System G(s)

Apply the MATLAB command:

```
stepinfo(G);
```

A vector containing some critical values will be generated. Table 1 shows the most critical data for future analysis of the system:

e_{ss}	t_r	t_s	M_p
	2.7428	5.0039	0

Table 1: Table of Some Critical Values

Next, the root locus plot will be needed to explore the stability of the system if we change the gain of the system.

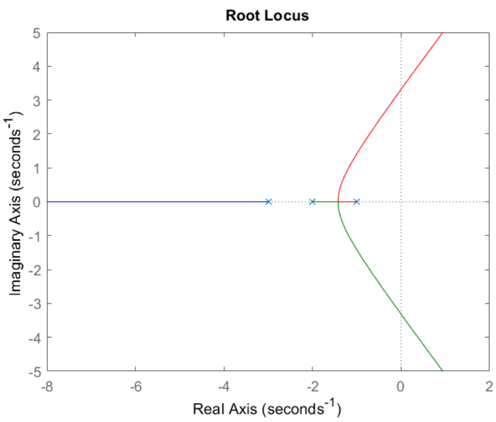

Figure 3: Root-Locus Plot of the System

Figure 3 shows the root locus plot of the system G(s). From the MATLAB plot, we can find that at the marginal stability, gain $K = 60$ and frequency $\omega = 3.31 rad/s$.

Using the information above, we are able to design a stable system with a proper gain. However, as can be seen from Table 1, the rising time t_r is really large, which means that this system responds slowly to input signals. Idealy, we would like to design a controller that reduces the rise time, settling time, and eliminates the steady-state error. According to Figure 1, a proportional controller decreases the rising time, and reduces the steady state error as well. Add a proportional controller by implementing the MATLAB code in the Section 3.2 of the Appendix, and find the step response characteristics for $k_p = 40$.

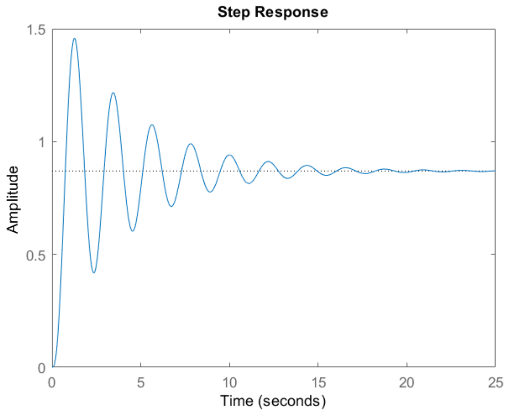

Figure 4: Step Response of the P-Controller

Figure 4 shows the step response and Table shows the key response characteristics of the system after adding a proportional controller.

e_{ss}	t_r	t_s	M_p
	0.4368	15.6105	67.6273

Table 2: Table of the Key Characteristics of the P-Controller

Compare the data in Table 2 and 1, it can be seen clearly that by adding the P-Controller, the system is responding to an input signal much faster. However, the giant overshoot of this system makes the system not as stable as it was before, and the settling time becomes much longer meaning that it is taking longer for the system to reach a steady state.

3 Appendix: MATLAB Code for Modeling the PID Controller

3.1 Analysis of the System with Transfer Function *G(s)*

```
G = tf([1], [1 6 11 6]);
figure(1)
hold on;
step(G);
stepinfo(G);
figure(2)
rlocus(G);
```

3.2 Analysis of a Proportional Controller

```
C_P = pid(40);
open_loop = series(C_P, G);
H1 = feedback(open_loop, 1);
hold on;
figure(1)
step(H1);
stepinfo(H1);
```