**Executive Summary**

The purpose of this experiment was to accomplish two tasks; to analyze a Controller under three different modes of operation: Proportional (P), Proportional + Integral (PI), and Proportional + Derivate (PD), and using benchmark parameters of Proportional Control to improve the output of the latter two modes of operation.

The second task involved combining of all three modes of operation to achieve a stable Proportional + Integral + Derivative (PID) Controller, and, using one of three methods of tuning: trial-and-error, Ziegler-Nichols “Ultimate Gain” Method or Ziegler-Nichols “Process Reaction” Method, to improve the closed loop response to find the best parameters to meet given specifications.

Certain specifications were to be obtained from the controllers under different gains. The specifications included Percent Overshoot (PO), Settling Time, Rise Time, and Steady State Error for both step and ramp input. In the first part of the lab, all of the transfer functions of each controllers were tested with different gain values and the PI and PD controllers were tested on different time constants as well. It is observed that the specifications will change from gain to another and from time constant as well as the difference between each controller compared to the benchmarked response. Further details will be supported within the report for clarification purposes.

In the last part of this experiment, the three controllers wee all summed up in a parallel connection to come up with what is called the PID controller. The “Ultimate Gain” Method was chosen for designing/tuning the controller which states manipulating the value of the gain of the P controller and getting the period off the plot and then using the modified equations to obtain both integral and derivative time constants that can be achieved of the PID. For our case, the value of the gain of the P controller was 7.1946 and the chosen value was the benchmarked gain which was 3.05 V/V.

**Part 1: Exploring Control Modes (P, PI and PD)**

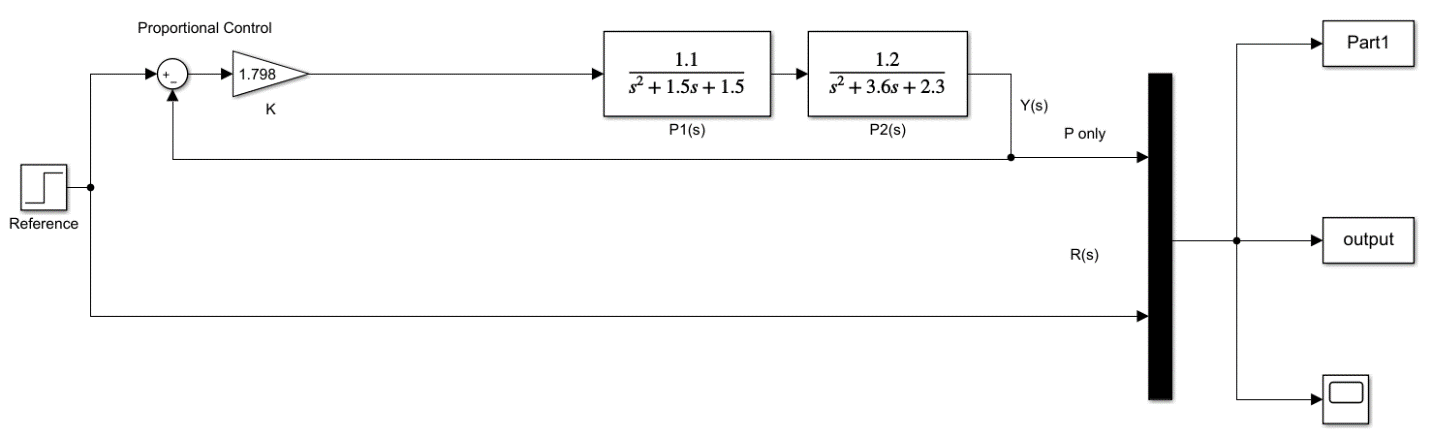
**1.1 Proportional (P) Control**

Figure 1: P Control SIMULINK Diagram

The transfer function used is as follows:

The characteristic equation is:

It was found that the Critical Gain is 7.198.

Different gain values were used, and the results were tabulated in the following table:

Table 1: Steady State Error of P Controller under Different Operational Gain

|  |  |  |
| --- | --- | --- |
| **Proportional Gain Value** “Kp” | **Steady State Output Value** “yss” | **Steady State Error** **Percentage** “ess”(%) |
| 1 | 0.2768 | 73.32 |
| 2 | 0.4335 | 56.65 |
| 3 | 0.5345 | 46.55 |
| 4 | 0.6049 | 39.51 |
| 5 | 0.6568 | 34.32 |
| 6 | 0.6945 | 30.55 |
| 7 | 0.7948 | 20.52 |

The steady state error percentage was calculated using the following equation:

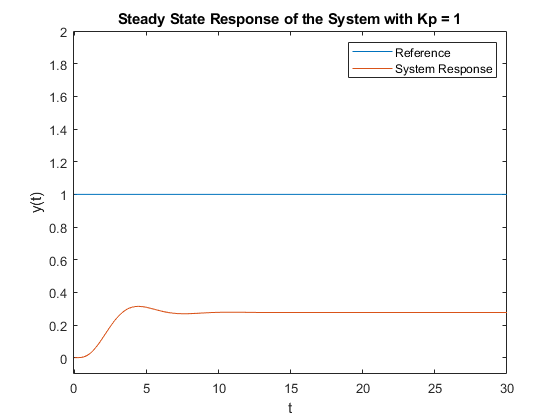
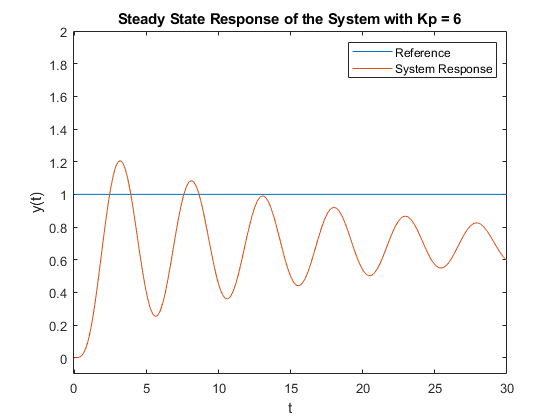
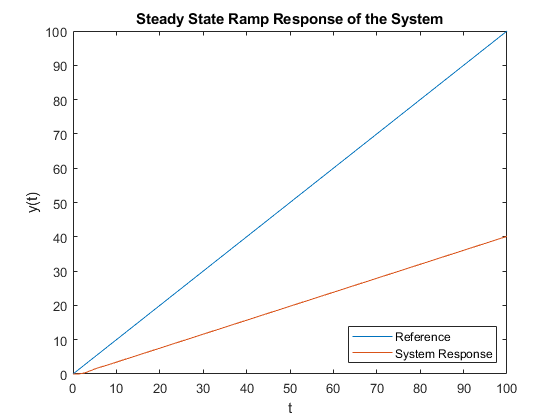
A ramp response was also studied to determine the system steady state response parameters outside of a step response. The results are as follows:

Figure 4: Steady State Response of P Controller with Unit Ramp Reference

Figure 2: Steady State Response of P Controller with Gain = 1

Figure 3: Steady State Response of P Controller with Gain = 6

As seen in the figure above, the ramp response is not an accurate way of determining response parameters as it will have a velocity constant, , of infinity due to the infinitely different slopes between the two signals. This is situation is modeled by the equation:

Furthermore, a unit step response can be used to measure the transient response of the system. These transient response parameters include settling time, percentage overshoot and peak time values. The table below describes the parameters at different values.

Table 2: Transient Parameters of P Controller under Different Operational Gain

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Proportional Gain Value** “” | **Percentage Overshoot** “PO” (%) | **Overshoot Peak Value** “PO” | **TSettle 2%** (s) | **TRise 0-100%** (s) |
| 1 | 13.44 | 0.3140 | 8.3612 | 3.3445 |
| 2 | 27.34 | 0.5520 | 10.70 | 3.01 |
| 3 | 40.29 | 0.7498 | 15.38 | 2.6756 |
| 4 | 51.78 | 0.9181 | 22.40 | 2.3411 |
| 5 | 62.94 | 1.0701 | 34.11 | 2.3411 |
| 6 | 72.37 | 1.1971 | 67.89 | 2.0067 |
| 7 | 67.2623 | 1.3294 | 99.66 | 2.0067 |

To calculate the Percentage Overshoot, the following equation was used:

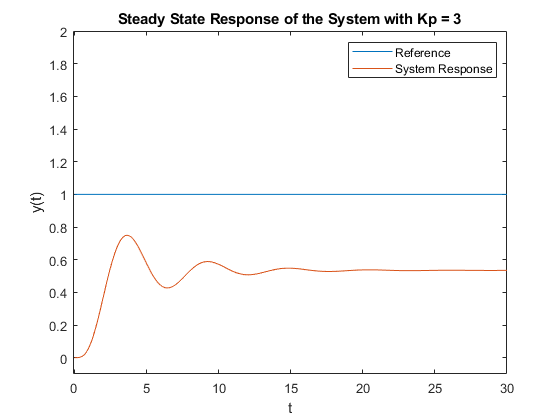
Note that placing reference lines within the MATLAB scope plot and using the cursor function to solve for the time in which the 2% settling occurs was the technique used to determine this settling time. An example plots with proportional gain of 3 can be found below:

Figure 5: Steady State Response of P Controller with Gain = 3

Table 3: Effect on Transient Response of P Controller with Change in Gain

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Proportional Controller Gain** | **ess Step** (%) | **ess Ramp** (%) | **PO** | **TSettle 2%** (s) | **TRise 0-100%** (s) |
| Increase | Decrease | Decrease | Increase | Increase | Decrease |
| Decrease | Increase | Increase | Decrease | Decrease | Increase |

“Benchmarking” the System

The system was “benchmarked” using the “Quarter Decay” response of the Proportional Controller [1]. The Operational Gain to achieve Quarter Decay was done through Trial-And-Error such that:

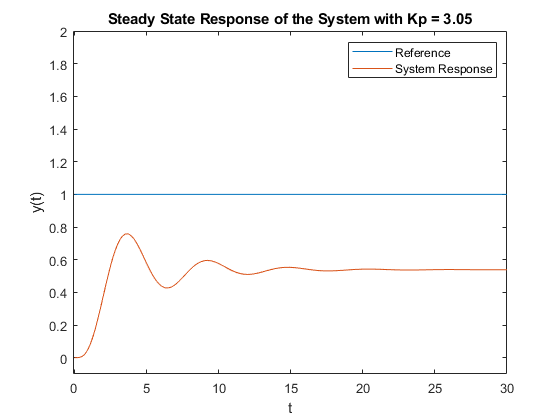


Figure 6: Steady State Response of P Controller when Kp = 3.05

Table 4: Transient Parameters of Benchmarked P Controller

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Proportional Gain Value** “” | **Percentage Overshoot** “PO” (%) | **Overshoot Peak Value** “PO” | **Steady State Output Value** “yss” | **Steady State Error** **Percentage** “ess”(%) | **TSettle 2%** (s) | **TRise 0-100%** (s) |
| 3.05 | 40.88 | 0.7587 | 0.5386 | 46.14 | 15.38 | 2.6756 |

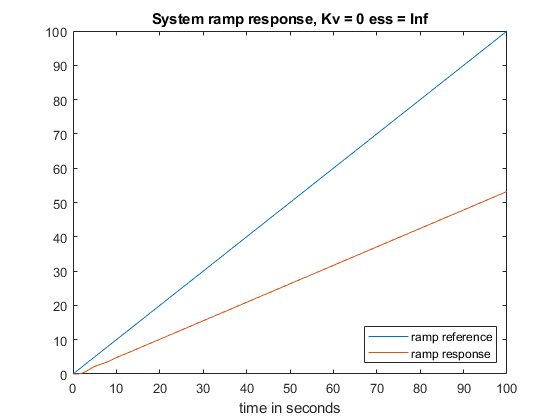


Figure 7: Ramp Response for Benchmark

**Proportional + Integral (PI) Control**

**Proportional + Derivative (PD) Control**

**Part 2: PID Control**

**Discussion**

**References**

[1]"PID Controller Tuning Techniques | ECE Tutorials", *Ecetutorials.com*. [Online]. Available: http://ecetutorials.com/process-control/pid-controller-tuning-techniques/.