Trenton Cathcart
Date: 10/01/2023

MATLAB, Simulink & LabVIEW Control System Configurations and Controller Design

Table of Contents

I. List of Tables:

A. N/A

II. List of Figures:

III. Objectives:

The objective of this experiment, conducted in the Control Systems Laboratory (EE 3321), is to delve into the intricacies of control system configurations and controller design using prominent software tools: MATLAB, MATLAB Simulink, and LabVIEW. Through this hands-on exploration, participants will construct and analyze series, parallel, and feedback systems of transfer functions. The experiment aims to provide a comprehensive understanding of the proportional gain controller and the PID (Proportional-Integral-Differential) controller, emphasizing their real-world applications and effects. By the end of this experiment, participants are expected to

have a solid grasp of control system configurations, their practical implementations, and the impact of varying controller parameters on system performance.

IV. Equipment Used:

- Matlab Program
- Labview Software
- Computer

V. Background Theory:

At the heart of this theory are transfer functions, mathematical representations that capture the relationship between the input and output of a system. These systems can be interconnected in various configurations, namely series, parallel, and feedback. Each configuration offers a unique way of combining individual system behaviors to achieve a collective response. The series configuration is a sequential connection, resulting in the multiplication of individual transfer functions. In contrast, the parallel configuration combines systems side-by-side, leading to the summation of their transfer functions. The feedback configuration, on the other hand, introduces a loop where the output of a system influences its input, often used to achieve stability or specific dynamic characteristics. Furthermore, controllers, such as the proportional gain controller and the more complex PID controller, play a pivotal role in adjusting and refining system behavior. These controllers work by comparing the system's actual output to a desired reference and making necessary adjustments based on the difference or error. The proportional gain controller acts on the present error, while the PID controller considers the present, past (integral), and future (derivative) values of the error, offering a more nuanced control. Through understanding these fundamental theories, one can design and optimize control systems for a myriad of applications, from simple household appliances to complex industrial machinery.

VI. Preliminary Calculations:

For our prelab of Lab #5b, we need to concentrate on questions 3 and 5. We'll begin by revisiting the Proportional Gain Controller Theory on page 2 and then delve deeper into the PID controller concepts on pages 3-4. In question 3, we'll engage with the transfer function of the Armature Controlled DC Motor, focusing on the relationship between Speed (ω(s)) and Voltage (V(s)), ensuring we exclude the 1/s term. Our aim is to adjust the proportional gain and observe its effects, utilizing values from Experiment #3 and making specific assumptions. For question 5, we'll integrate the PID controller with the transfer function from Experiment #4 in LabVIEW, combining the model with series and feedback blocks. This preparation will lay the groundwork for a successful lab experience.

VII. Procedure/Result/Analysis:

I. Submarine Depth Control System Analysis:

In this section, we delved into the intricacies of the submarine depth control system. We began by examining the transformed block diagram representation, where C(s) symbolized the transform of the commanded depth signal and D(s) represented the transform of the actual

depth signal of the submarine. To simulate a real-world scenario, a 50-foot dive was modeled, initiated at t = 0, using the equation c(t) = -50 u(t). MATLAB was then employed to compute the submarine's depth over time, specifically focusing on three distinct values of K: 0.4, 0.08, and 0.04. The chosen time scale for this analysis ranged from -10 to 180 seconds. The results of this analysis, including the MATLAB code and the three corresponding graphs for each K value, are vividly illustrated in **Figure 1**. The graphs not only showcase the submarine's depth over time but also highlight the significant behavioral differences based on the varying K values. The command signal is also prominently displayed in each graph, serving as a reference point for the depth commanded versus the depth achieved.

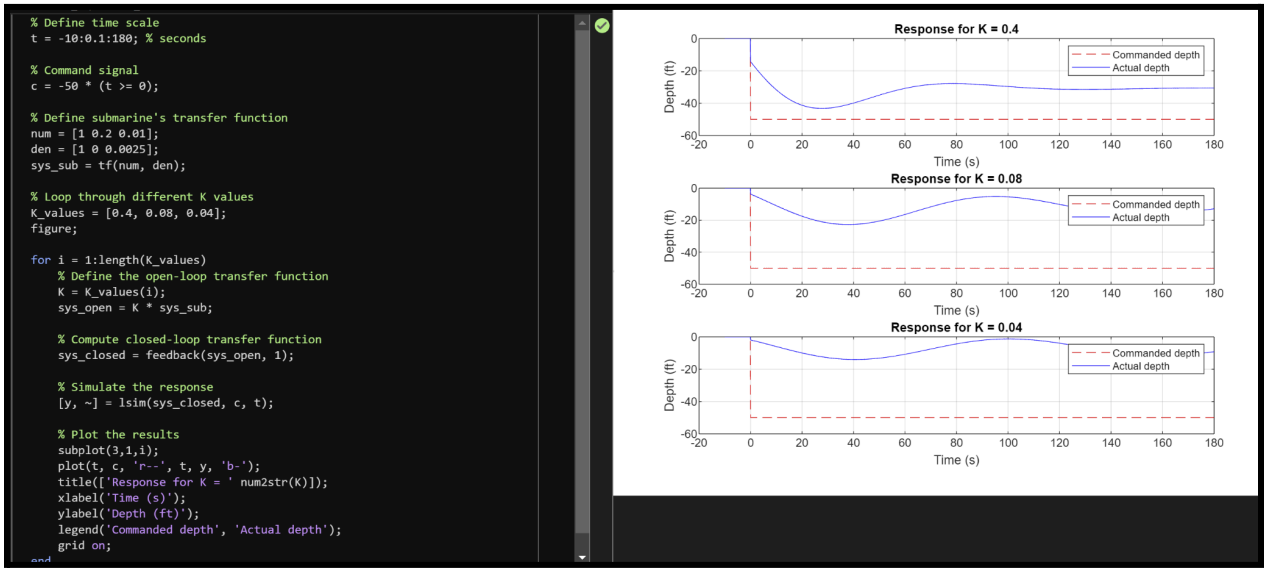

Figure 1: Submarine Matlab Script

II. MATLAB Simulink Analysis:

Following the initial MATLAB analysis, we further explored the submarine depth control system using MATLAB Simulink. Through Simulink's graphical interface, we constructed a block diagram representation of the system that mirrored the structure and components of our previous MATLAB model. The desired configuration for this system is depicted in **Figure 2**. After setting up the model in Simulink, we simulated the system's response and generated the output. The intricacies of the Simulink block diagram, along with its corresponding output, are showcased in **Figure 3**. Upon comparison, the Simulink output appeared to be more discretized than the MATLAB results. This distinction provided a different perspective on the system's behavior, especially when observing the effects of varying the K values.

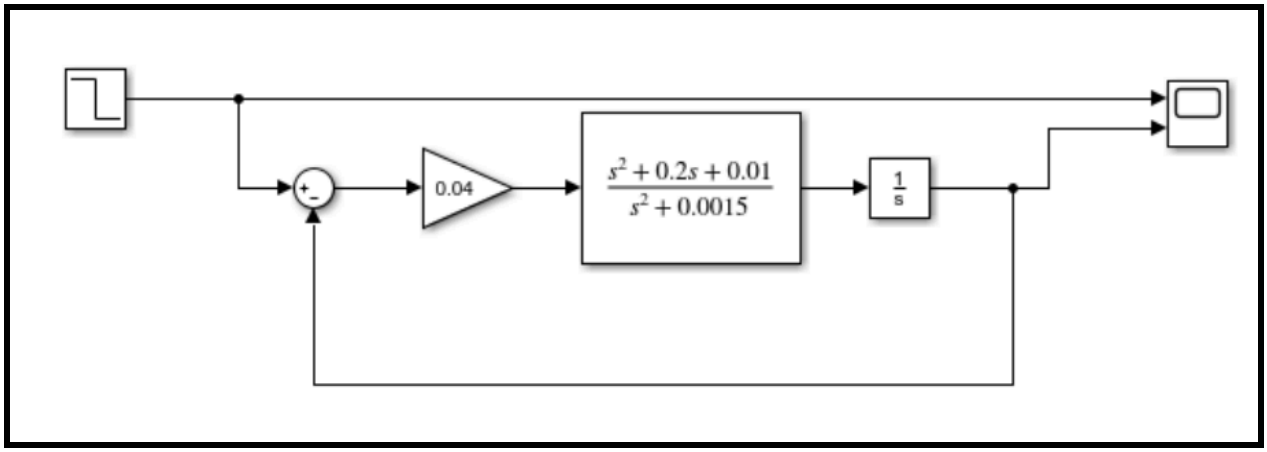

Figure 2: Desired Simulink

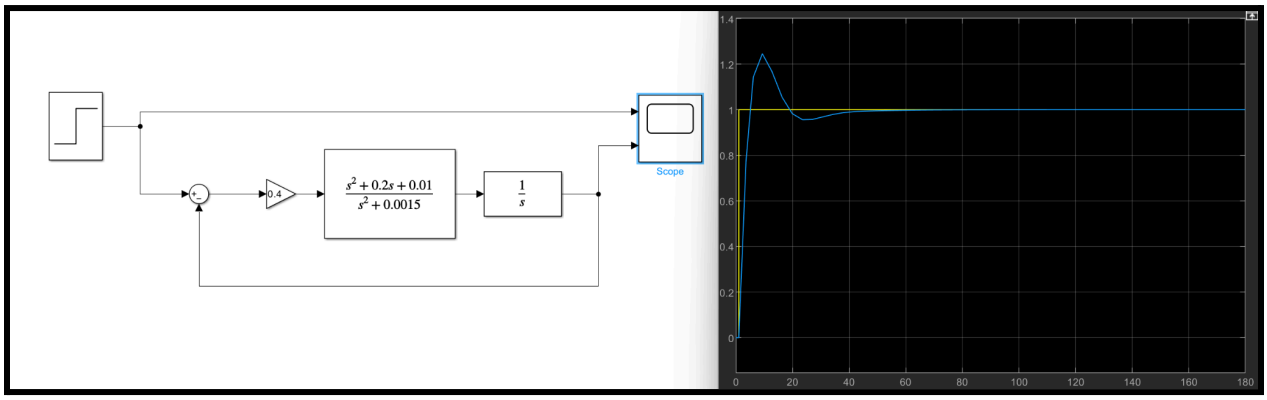

Figure 3: Programmed Simulink and Output

III. Proportional Gain Controller for DC Motor:

In the third segment of our experimental procedure, we focused on the development of a proportional gain controller for an armature-controlled DC motor, as referenced in experiments 3 and 4. The primary objective was to regulate the motor's speed by applying a specific voltage from the controller. For this task, we utilized the transfer function representing Speed (ω(s)) over Voltage (V(s)). It's crucial to note that our focus was on ω(s)/V(s) and not θ(s)/Voltage (V(s)), leading to the exclusion of the 1/s term.

To achieve our goal, we varied the proportional gain constant and keenly observed its effects on the system. The parameters for this experiment, including Ra, La, J, and Kb, were sourced from the data in Experiment #3. For the purpose of this experiment, we assumed Td to be 0 and considered V as u(t).

Both MATLAB and MATLAB Simulink were employed to carry out this task. The results derived from the Simulink model, which include the system's response and behavior under varying gain constants, can be thoroughly examined in **Figure 4**.

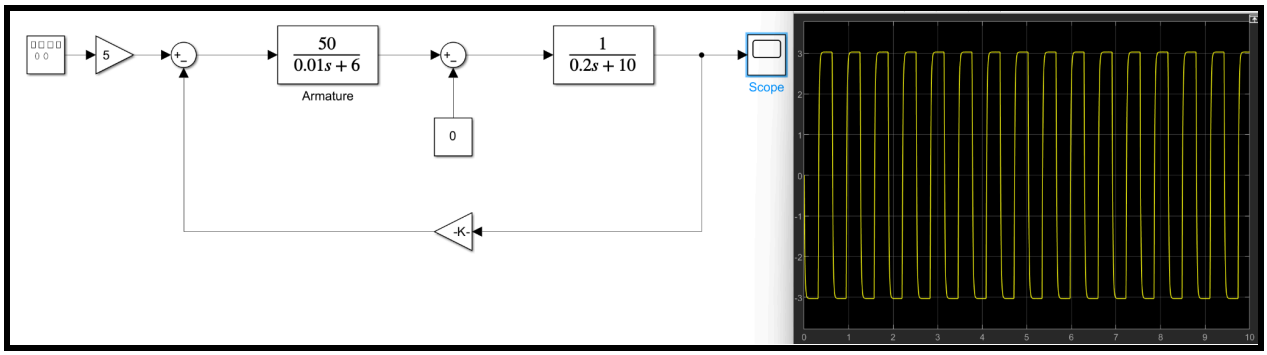

Figure 4: Simulink Block and Output

IV. Proportional Gain Controller for DC Motor:

In the subsequent phase of our experiment, we ventured into the realm of LabVIEW to construct and display a series connection for the system defined by H3(s) = H1(s) H2(s). LabVIEW, with its intuitive graphical interface, facilitated the seamless integration of these transfer functions. We began by setting up the series connection, ensuring that the individual systems H1(s) and H2(s) were appropriately interconnected to produce the desired H3(s) output.

Following the series connection, we further expanded our exploration by constructing parallel and feedback connections for the same two systems, H1(s) and H2(s). The LabVIEW environment provided the necessary tools and blocks to achieve these configurations with precision.

The culmination of our efforts in LabVIEW, including the series, parallel, and feedback configurations, along with their respective outputs, can be meticulously observed in **Figure 5.**

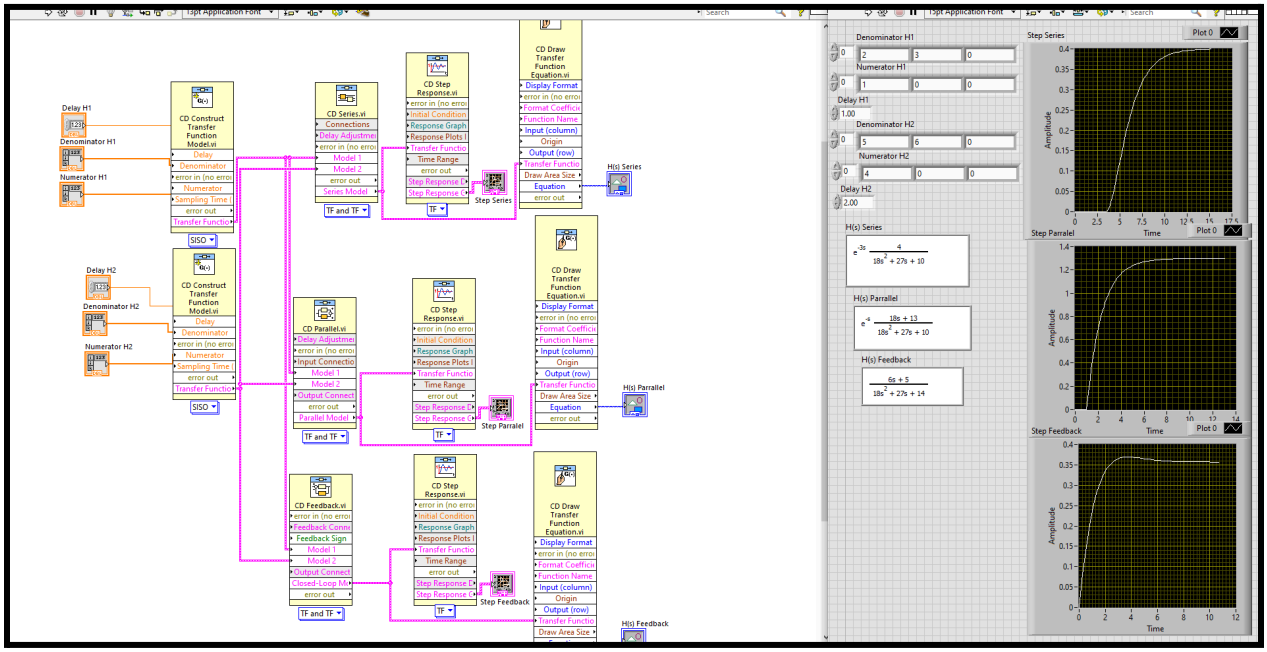

Figure 5: Labview All Responses

V. Proportional Gain Controller for DC Motor:

In the final segment of our experimental procedure, we delved into the construction of a PID controller using the LabVIEW Control Design and Simulation Toolkit. This controller was specifically tailored to be compatible with the motor transfer function we previously established in Experiment 4. With the PID controller in place, we embarked on a comprehensive analysis of both the time and frequency responses.

Our observations were meticulously recorded as we navigated through the system's dynamics. A pivotal aspect of this phase involved adjusting the proportional, integral, and differential gain constants. By tweaking these constants, we were able to discern the distinct effects each one imparted on the system's behavior. The nuances in system performance, as influenced by these adjustments, provided valuable insights into the intricacies of PID control.

The results of our endeavors, including the time response and frequency response visualizations, are vividly captured in **Figures 6 and 7**, respectively.

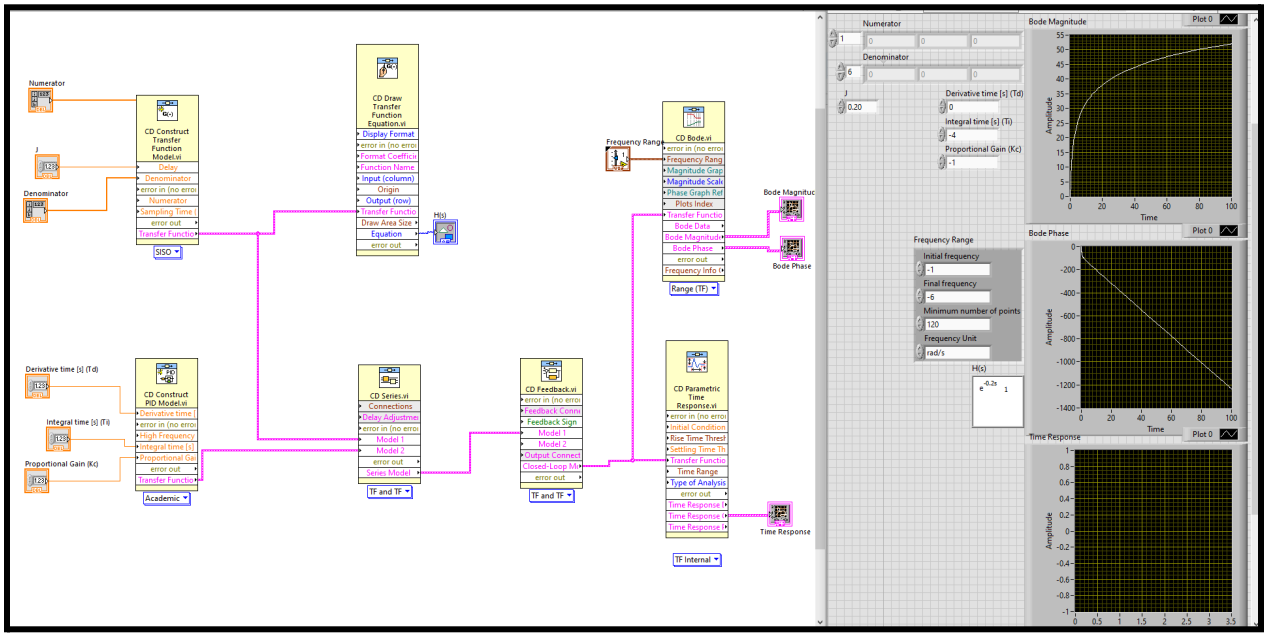

Figure 6: Labview PID

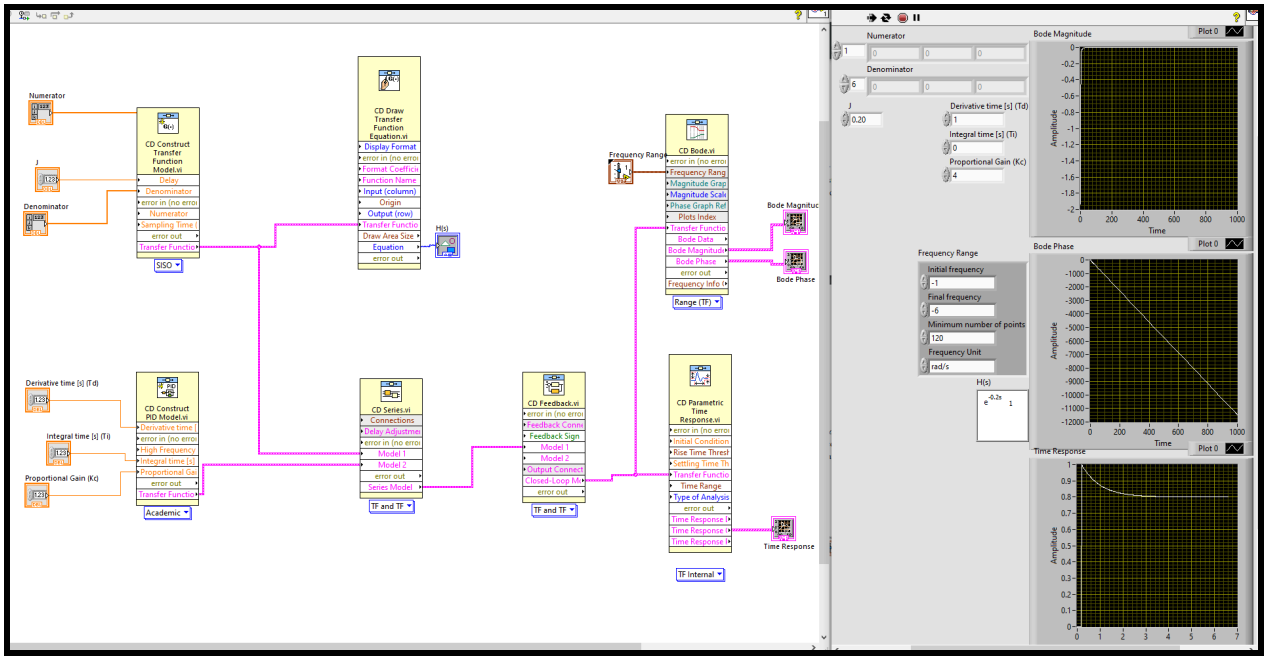

Figure 7: Labview PID Adjusted

VIII. Conclusion:

Throughout the experiment, it became evident that varying the gain constants in a control system significantly influences its behavior and stability. The proportional gain directly impacted the system's responsiveness to errors, with higher values leading to a quicker response. However, excessive proportional gain could introduce overshoot and potential oscillations. The integral gain aimed to eliminate steady-state errors, but when set too high, it might cause the system to become sluggish or even unstable. The differential gain provided a predictive action, counteracting rapid changes in the system. However, it's crucial to fine-tune it to prevent excessive sensitivity to noise. During our tests, there were instances where certain combinations of gain values led to unstable system behavior, underscoring the importance of meticulous tuning and understanding of PID controllers in control systems.

IX. References:

None

X. Appendix:

Appendix 1: N/A