Trenton Cathcart
Date: 09/27/2023

LabVIEW Control Design & Simulation Toolkit

Table of Contents

I. List of Tables:

II. List of Figures:

III. Objectives:

The primary goal of Experiment 4 is to bridge the foundational knowledge acquired from previous sessions and extend its application to the LabVIEW environment, particularly utilizing its Control Design & Simulation Toolkit. Students will immerse themselves in an intricate exploration of control system and dynamic system analysis, leveraging tools that echo functionalities present in both MATLAB and Octave. Through a comprehensive approach, the experiment aims to equip participants with the proficiency to construct system models in both graphic and textual formats. Moreover, the experiment emphasizes the practical aspect by enabling students to observe and interpret system responses under varying conditions. This hands-on experience, backed by state-of-the-art toolkits, seeks to instill a deeper understanding of control systems, readying students for real-world engineering challenges.

IV. Equipment Used:

- Matlab Program
- Labview Software
- Computer

V. Background Theory:

In control system engineering, understanding the dynamics of a system is pivotal. At the heart of these dynamics lies the concept of the Transfer Function (TF), which is a mathematical representation showcasing the system's output to a given input. This experiment revolves around the integration of LabVIEW, a visual programming language platform, with traditional control system principles. The Control Design & Simulation Toolkit within LabVIEW offers a rich set of functionalities that mirror many found in MATLAB's Control System Toolbox, catering to both graphic and textual enthusiasts. By representing systems in State-Space (SS) and Zero-Pole Gain (ZPK) models, alongside the conventional TF, we obtain a holistic view of a system's dynamics. Observing how systems respond, especially through step and frequency response analyses, becomes crucial in predicting their behavior under varying conditions. Furthermore, the armature-controlled DC motor system, characterized by various parameters such as back-emf constant, inertia, resistance, and torque constant, serves as an epitome of a real-world system. By modeling and simulating this system in LabVIEW, students get an opportunity to witness the intricacies of its dynamics. Ultimately, the theory underscored in this experiment aims to blend mathematical principles with practical applications, ensuring students not only comprehend but also implement the essence of control system dynamics.

VI. Preliminary Calculations:

Before diving into the core of the experiment, it is essential to have a foundational setup in place. From our last experiment, we had developed a MATLAB script tailored for a particular system's dynamics. To ensure a seamless transition and integration in this experiment, students should confirm that they have retained the MATLAB program. Given that LabVIEW's Control Design & Simulation Toolkit offers the capability to integrate MATLAB scripts via the MathScript Node, it becomes vital that our MATLAB program is both accessible and compatible. In essence, the preliminary work primarily revolves around ensuring the MATLAB script from the previous experiment is readily available and, if needed, reviewing its logic to guarantee its consistency with the current experimental objectives.

VII. Procedure/Result/Analysis:

I. Graphical Modeling of the DC Motor System:

In the first segment of our experimental procedure, our focus was on constructing a detailed graphical model of the armature-controlled DC motor system. Drawing from the provided equation, we worked meticulously to represent each variable accurately. These variables included the input voltage (Va), the output angular velocity (ωm), the back-emf constant (Kb), the system's inherent inertia (Jeq), armature resistance (Ra), and the torque constant (Ka). The parameters used in this modeling phase were derived from a previous experiment. We were specifically instructed to refer to **Figure 1** as a reference for our construction. After diligent work and ensuring all elements were appropriately represented, our final graphical model was represented in **Figure 2.**

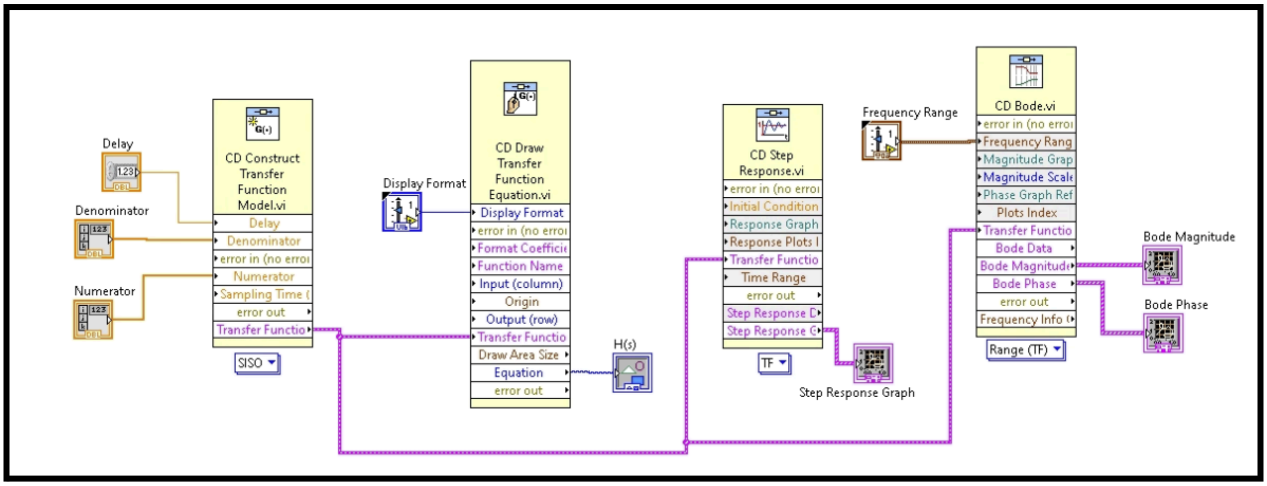

Figure 1: Example Construction

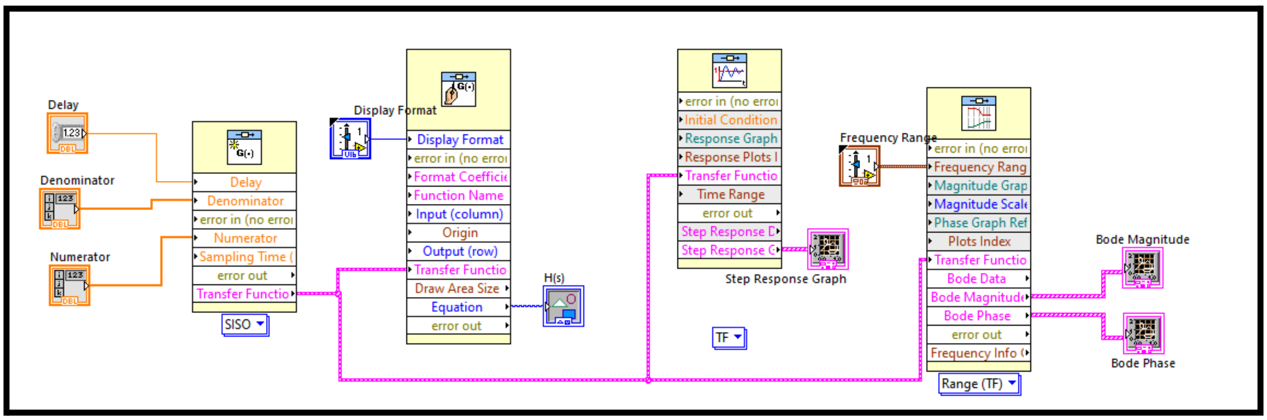

Figure 2: Constructed Labview Armature Controlled DC Motor

II. Transfer Function Creation & MATLAB Integration:

For the second segment of our procedure, we embarked on re-constructing the block diagram formed in the initial part, with a special emphasis on integrating the transfer function through LabVIEW's MathScript Node. This integration aimed to seamlessly marry the capabilities of both MATLAB and LabVIEW, creating a more comprehensive and detailed representation of our system. To achieve this, we began by drafting a MATLAB script, incorporating the parameters set in the first part of the experiment. Once our script was fine-tuned and saved, we proceeded to import it directly into the MathScript Node within LabVIEW. This integration proved instrumental in simulating and visualizing the transfer functions and system dynamics accurately. The amalgamation of MATLAB's robust scripting with LabVIEW's intuitive graphical interface resulted in our final representation, as showcased in **Figure 3.**

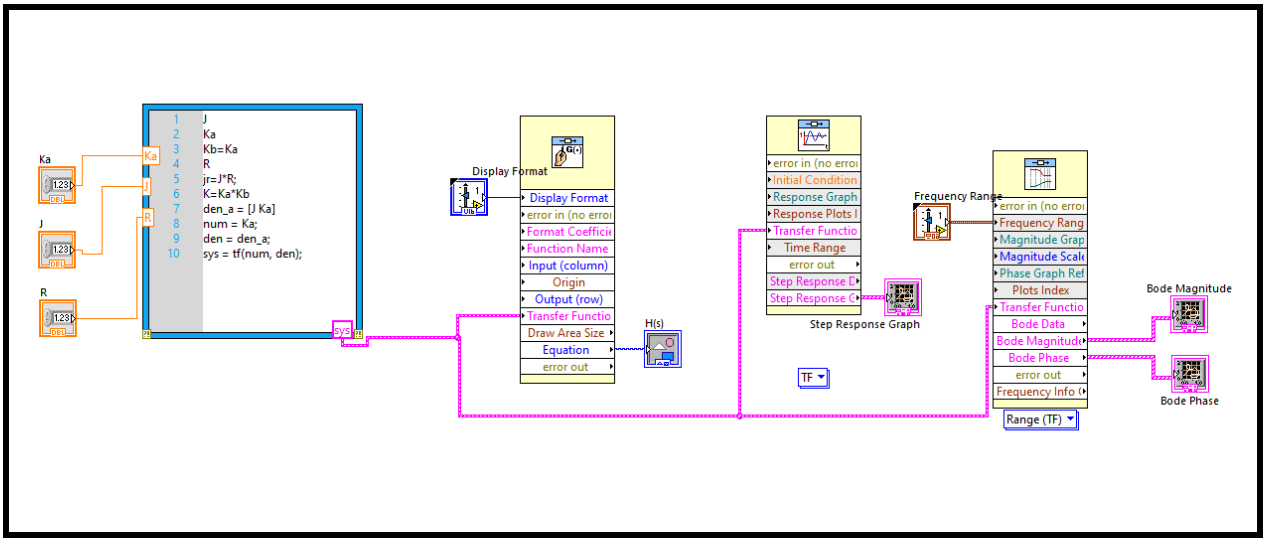

Figure 3: Labview Motor With Matlab Script

III. System Response Analysis:

For the third and final portion of our experimental procedure, our focus shifted to keenly observing both the step and frequency responses of the model we meticulously constructed in the prior segments. To begin, we launched the LabVIEW software and initiated our system simulation. The immediate output, using the parameters provided at the outset of the experiment, yielded a distinct response pattern that we captured and documented as **Figure 4.** With this benchmark established, we took the initiative to modify the system by varying the armature resistance, Ra, a pivotal component that influences the system's overall behavior. Adjusting the resistance allowed us to observe distinct alterations in the step and frequency responses of our system. These variations and their consequential impacts on the system were captured and have been showcased in **Figure 5.** The comparisons between the two figures underline the profound influence of resistance on the system dynamics and underscore the importance of intricate calibrations in control systems.

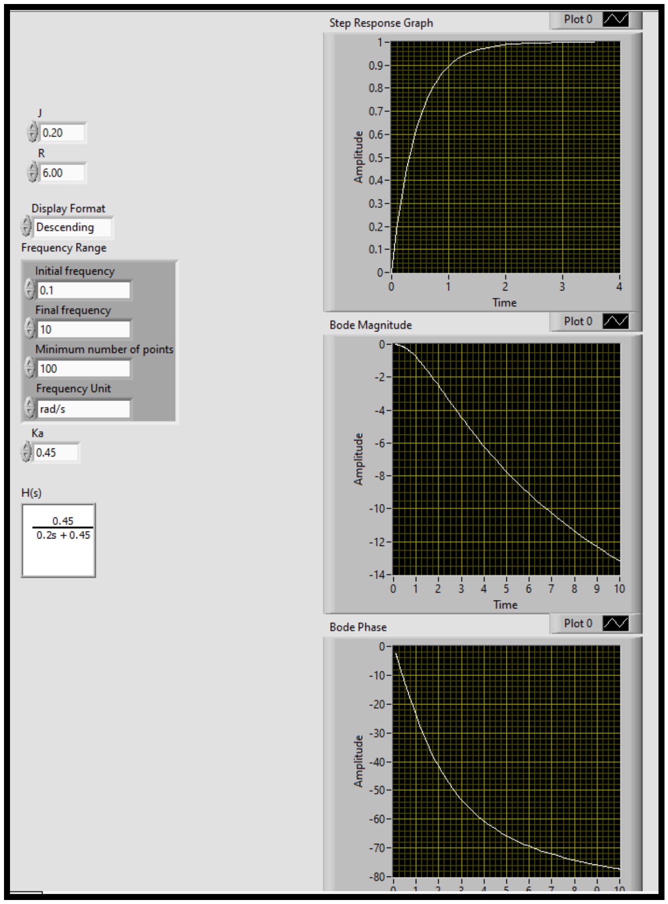

Figure 4: Labview Original Parameter Output

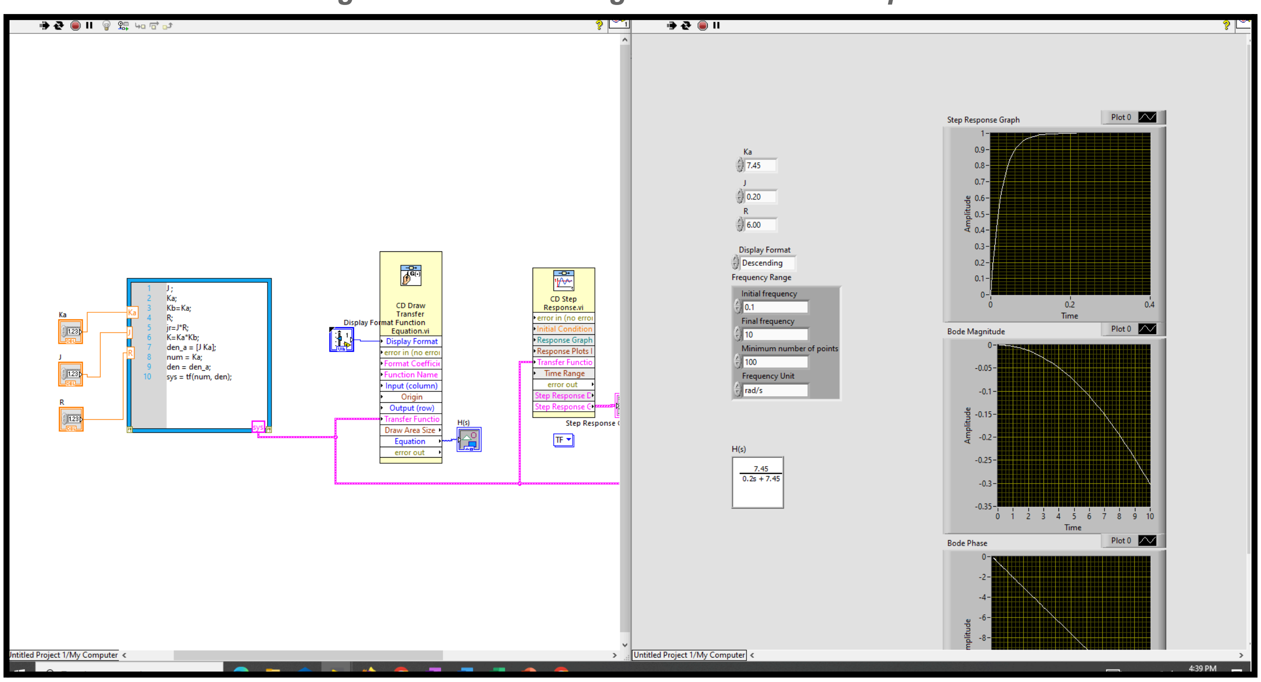

Figure 5: Labview, Adjusted Resistance Value

VIII. Conclusion:

Upon analyzing the results derived from the step responses associated with different values of Ra, it's evident that the armature resistance, Ra, plays a crucial role in the system's transient behavior. As Ra was varied, we noticed tangible modifications in both the rise time and settling time of the system’s step response. Specifically, an increase in Ra seemed to lead to a more sluggish response, suggesting that the system might be more resistant to changes. Conversely, a decrease in Ra quickened the system's reaction to inputs. Without the direct visualization provided by pole-zero plots, it's speculative yet educated to presume that as Ra is increased, the poles of the system might move closer to the imaginary axis in the s-plane, leading to a slower system response. Conversely, reducing Ra might shift the poles further left, making the system inherently more agile and responsive. This observation reinforces the criticality of armature resistance in controlling the dynamics of DC motor systems and accentuates its value in fine-tuning performance in real-world applications.

IX. References:

None

X. Appendix:

Appendix 1: N/A