Trenton Cathcart
Date: 10/18/2023

Experiment 6: QUARC Software, Hardware Integration, Filtering, and Constant Determination

Table of Contents

I. List of Tables:

II. List of Figures:

III. Objectives:

The objective of the experiment is to familiarize with the QUARC software and its integration with the QUBE-Servo 2 hardware. Through the experiment, we will design a Simulink model using QUARC blocks to drive a DC motor and measure its angular position or speed. Subsequently, they will design a model to measure the servo velocity, analyze the angular velocity response, determine the equipment's gain and time constants, and create a comprehensive transfer function that represents the system.

IV. Equipment Used:

- Matlab Program

- Simulink and Quarc Software
- Computer

V. Background Theory:

Control systems are vital in engineering and technology as they govern the behavior of devices and systems to achieve desired outputs. At the heart of these systems is the transfer function, a mathematical model representing the relationship between the system's input and output. This experiment utilizes the QUBE-Servo 2 hardware and QUARC software to simulate and measure responses of a DC motor, a fundamental component in many control systems. By designing and analyzing the Simulink model, students will bridge the gap between theoretical principles and real-world applications, gaining insights into the dynamic behavior of control systems, including gain and time constants.

VI. Preliminary Calculations:

No Preliminary calculations required.

VII. Procedure/Result/Analysis:

I. Reading the Encoder/Driving the DC Motor:

In our experiment, we interfaced with the QUBE-Servo 2 using MATLAB's Simulink and QUARC. We began by initiating a new Simulink diagram within MATLAB Simulink and navigating through the SIMULINK Library Browser to access the QUARC Targets and then the Data Acquisition | Generic | Configuration folder, as showcased in Figure 6.4. Here, we integrated the HIL Initialize block into our model, vital for the data acquisition device configuration. With the QUBE-Servo 2 connected and its power LED glowing green, we specified the board type as 'qube_servo2_usb'. Adjusting the Real-Time Workshop parameters through the QUARC menu, we prepped our model for external operation and compiled it, resulting in the QUARC controller. Upon its activation, the QUBE-Servo 2's power LED blinked. Our next phase involved encoder readings, noting that the encoder reset to zero upon each start or cessation. For a complete rotation, it recorded roughly 2000 counts, evident in **Figure 1**. Shifting our attention to the DC motor, we enhanced our SIMULINK diagram with the HIL Write Analog block from the Data Acquisition | Generic | Immediate I/O category. This inclusion allowed signal output from the analog output channel #0 connected to the onboard PWM amplifier driving the DC motor. We further enriched the Simulink model by introducing the Constant block from its sources folder. Linking the Constant and HIL Write Analog blocks as represented in Figure 6.5, we ran the QUARC controller post-building. Setting the Constant block at 0.5 applied 0.5 volts to the QUBE-Servo 2's DC motor. We verified the positive correlation between signal input and measurement, crucial for control system designs. Conclusively, a positive input caused the disc to rotate clockwise, as depicted in **Figure 2**: Encoder Reading Simulink Model.

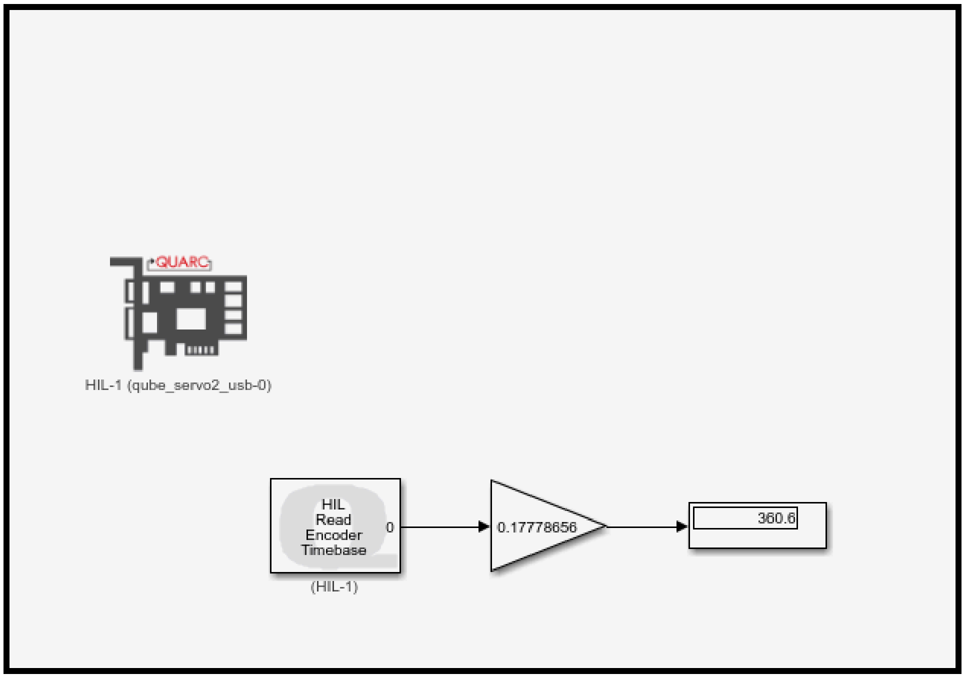

Figure 1: Qube Simulink Configuration Model

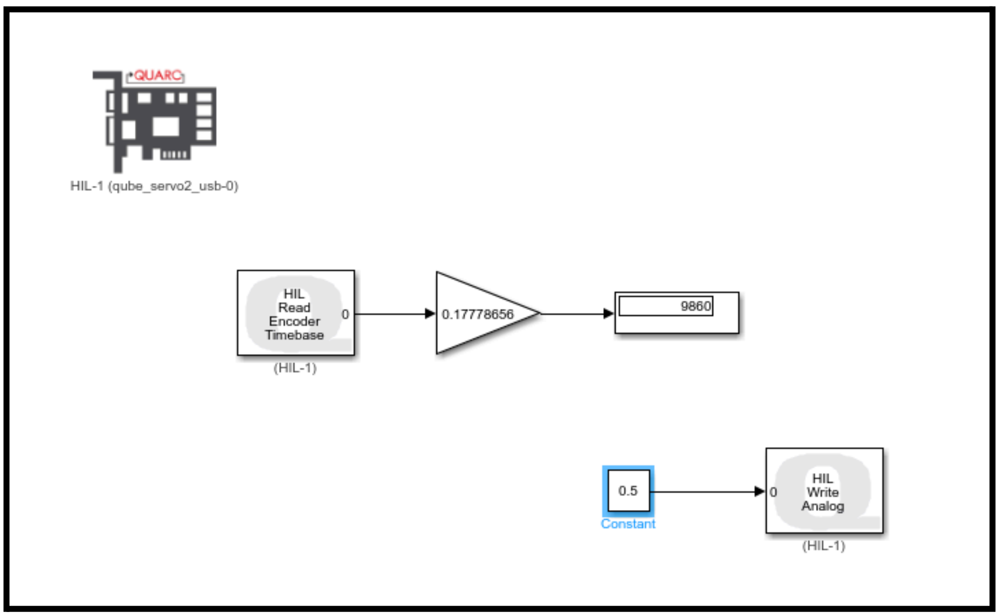

Figure 2: Encoder Reading Simulink Model

II. Measuring Servo Velocity:

In our experiment, we proceeded to measure the servo velocity using the model we developed earlier. We altered the encoder calibration gain to calculate the gear position in radians rather than degrees. For the purpose of this experiment, and as instructed in section VIII, we constructed the SIMULINK diagram as depicted in Figure 6.6, deliberately excluding the

Transfer Fcn block for the time being. Within the Solver selection area, we designated the type as Fixed-step and selected the solver as ode1 (Euler).

To gauge the gear speed using the encoder (in rad/s), we incorporated a Derivative block connected to the encoder calibration gain output. We then linked the Derivative's output to a Scope. We configured the source blocks to generate a step voltage transitioning from 1 V to 3 V at a frequency of 0.4 Hz. After building and running the QUARC controller, we analyzed the encoder speed response, which can be seen in our provided sample responses, resembling Figure 6.7.

Upon closer examination, we observed that the encoder-based measurement appeared noisy. To understand the underlying reason, we measured the encoder position using a new Scope. On zooming into the position response, we pondered if the signal was continuous, recalling its subsequent entry into the derivative. To mitigate some of the high-frequency components, we employed a lowpass filter (LPF) to the derivative output. We integrated a Transfer Fcn block from the Simulink | Continuous Simulink libraries post the Derivative output and link the LPF to the Scope. We configured the Transfer Fcn block as 50/(s + 50), shown in Figure 6.6. After executing the QUARC controller again, we noticed an improved filtered encoder-based speed response along with the motor voltage, as depicted in **Figures 4 to 7.**

The variations in the cutoff frequency, ωf, from 10 to 200 rad/s (or 1.6 to 32 Hz), showed diverse effects on the filtered response. We observed the 10 Hz cutoff presented the cleanest waveform, albeit with gentler rises and falls. Conversely, 200 Hz exhibited considerable noise but had abrupt cutoffs and spikes. To derive the DC gain and time constant values (K, τ), we employed the Cursor Measurements tool within the SIMULINK Scope or the Data Tips tool in the MATLAB figure, drawing direct measurements from our response plots. Finally, we halted the QUARC controller.

As we reviewed our results, **Figures 3 through 7** provided invaluable insights into our experiment. Specifically, **Figures 4 to 6** showcased the oscillation results at varying cutoff frequencies: 50 Hz, 200 Hz, and 10 Hz, respectively. Among our observations, the 10 Hz cutoff emerged as the cleanest, while 200 Hz presented sharp rises and drops amidst significant noise.

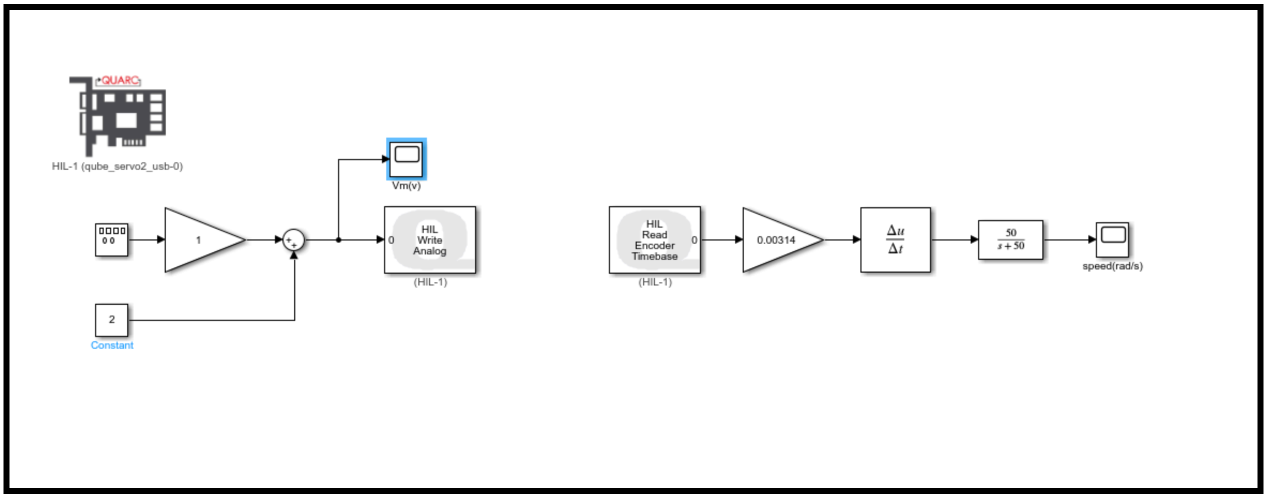

Figure 3: Simulink Block Construction

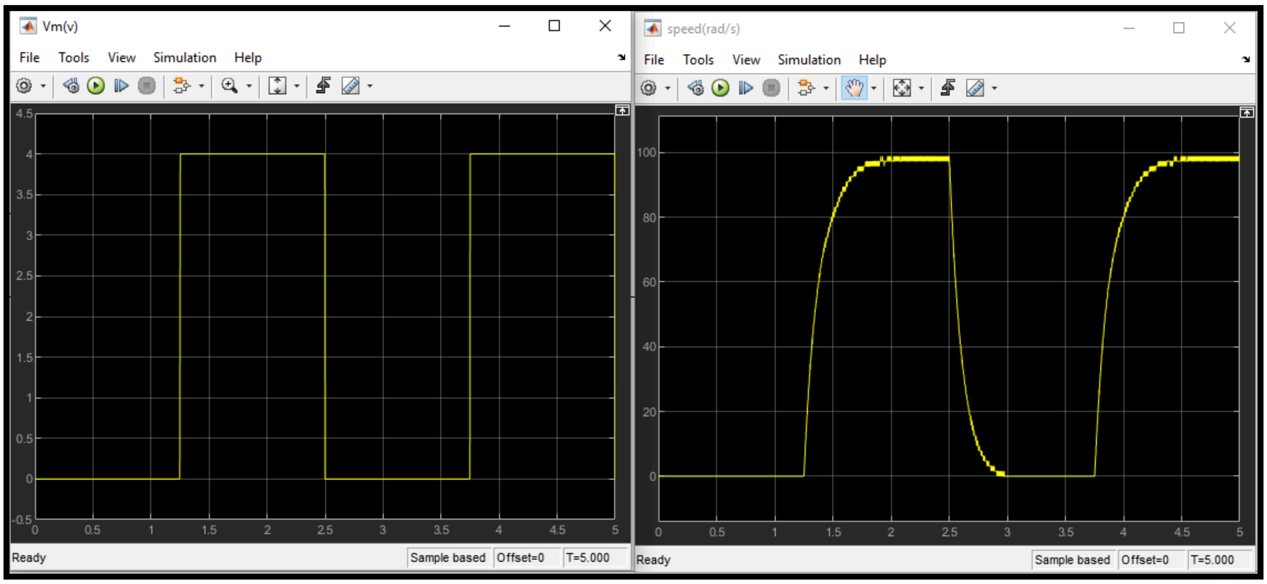

Figure 4: Oscilloscope Results; No Transfer Function

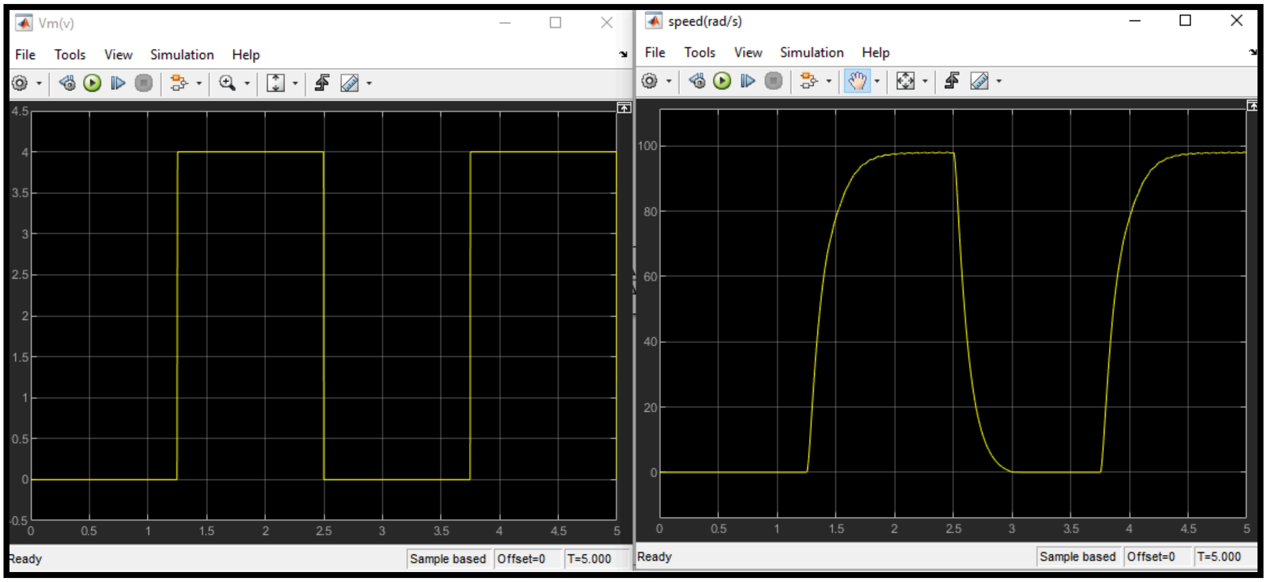

Figure 5: Oscilloscope Results; Cutoff 50 Hz

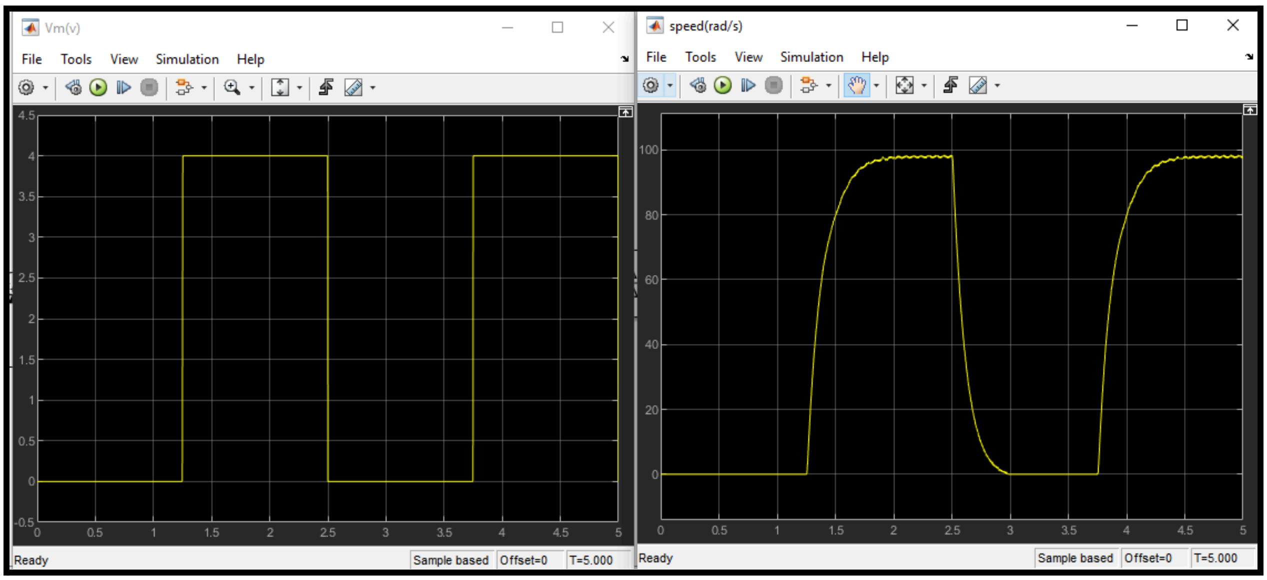

Figure 6: Oscilloscope Results; Cutoff 200 Hz

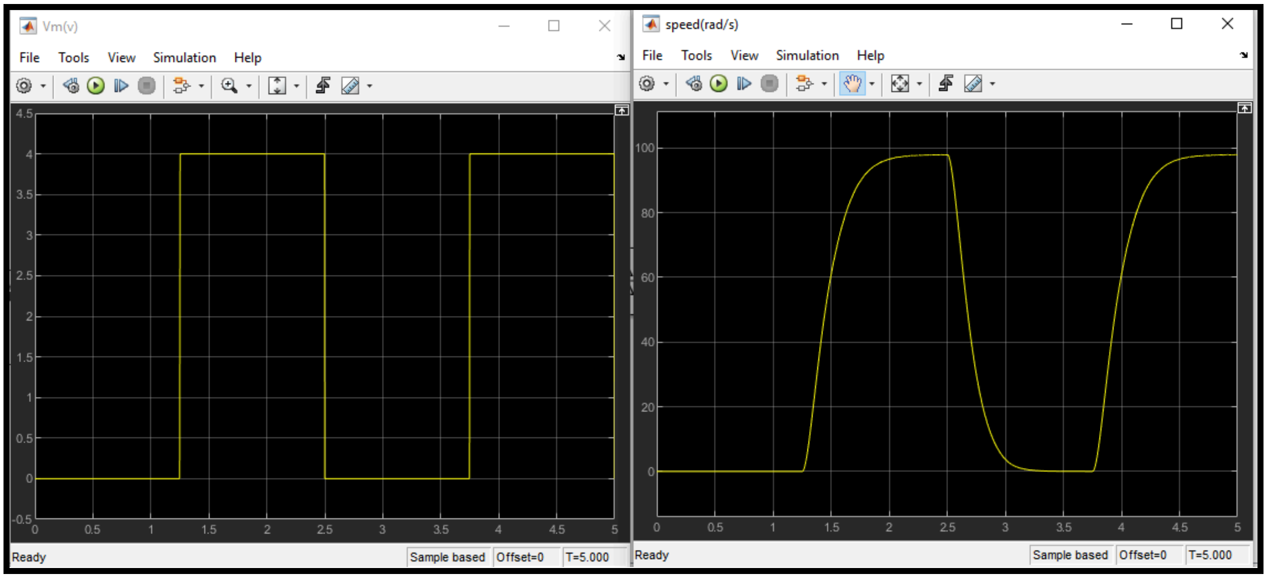

Figure 7: Oscilloscope Results; Cutoff 10 Hz

III. Bump Test:

In our subsequent phase of the experiment, we embarked on the Bump Test. Our objective was to devise a transfer function representative of the QUBE-Servo 2, correlating a voltage input with an angular speed output. As illustrated in Figure 6.8, we constructed the model, ensuring we incorporated the precise values we determined for K and τ, rather than default ones. For our experiment, K was set to 22.5, and τ was fixed at 0.16.

To facilitate simultaneous viewing of both the measured and the simulated QUBE-Servo 2 responses, we utilized a Mux block, sourced from the Signal Routing category, and connected them to the scope. With the model set, we proceeded to build and execute our QUARC controller. The subsequent outputs provided an enlightening comparison between our actual measurements and the simulated predictions. This visual representation was encapsulated in a MATLAB figure, presenting both the measured and simulated responses within a singular plot, alongside the input voltage. This visualization can be closely observed in **Figure 8**: Bump Test Model and Output.

Upon scrutinizing our outcomes and the model parameters, we asked ourselves if we had accurately derived the values for K and τ. Based on the close correlation between the measured and simulated results, we deduced that our parameters for K and τ were indeed correctly derived. The graph exhibited only minor deviations, affirming our confidence in our calculations and methodologies. Concluding this segment of our experiment, we halted the QUARC controller.

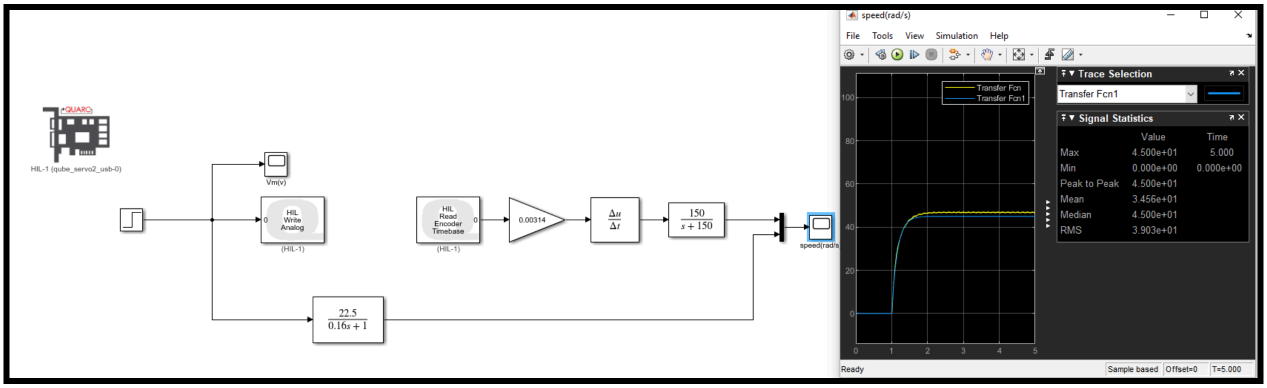

Figure 8: Bump Test Model and Output

VIII. Conclusion:

In concluding our experiment, we reflected on several pivotal observations and understandings we garnered throughout our investigation. Every time we initiated the QUARC controller, we noted that the encoder reading would reset to zero. This characteristic played an instrumental role in ensuring consistent measurements and facilitating accurate comparisons across different iterations.

Furthermore, our exploration into the efficacy of low-pass filters on the measured speed of the servo proved enlightening. Implementing a low-pass filter undoubtedly enhanced the clarity of our readings by filtering out high-frequency noise, leading to smoother, more interpretable results. However, it wasn't without its shortcomings. By employing the filter, there's a potential risk of losing sharpness in rapid fluctuations or transitions within the data. Striking a balance was crucial. For our experiment, we found the cutoff frequency of 10 Hz to be the most favorable tradeoff. At this frequency, the output was the cleanest, and while there were less pronounced rises and drops compared to higher cutoff frequencies, it provided a reliable middle ground between clarity and preserving essential data.

The bump test, an integral component of our experiment, was undertaken to validate the accuracy of our derived system parameters. In our words, the bump test serves as a benchmarking tool, juxtaposing our predicted model outputs with actual measured responses. The resulting comparison provided invaluable insights into the accuracy of our model's parameters and the system's behavior. Our results from this test underscored the precision of our derived values, reaffirming our confidence in the robustness and reliability of our experimental procedures and findings.