ELECTRIC VEHICLE DRIVE SIMULATION

Team Kappa | Cole Helmer & Mason Reilly

Widener University | RE 403 | Final Project Report | 5/7/2025

Table of Contents

Abstract

The project centered on replicating and validating a simulation of an electric vehicle (EV) motor-drive system, originally introduced by McDonald in his work, *Electric Vehicle Drive Simulation with MATLAB/Simulink*. The main goal was to reconstruct the foundational Simulink model to explore how power behaves during both motoring and regenerative braking operations. The recreated setup integrates a DC permanent magnet motor, an idealized motor controller paired with a proportional-integral (PI) controller, and a streamlined battery model. Using the parameters and modeling strategies outlined in McDonald's study, the team developed a stable and operational simulation environment. This updated model was then used to examine the flow of energy and system performance across specific speed and torque profiles, echoing the approach used in the original research. Ultimately, the exercise offered deeper insight into the dynamics of EV powertrains, highlighting control strategies and reinforcing the value of model-based design in analyzing vehicle systems.

Introduction

The automotive industry is undergoing a major transformation, with electric vehicles becoming a cornerstone in the pursuit of sustainable transportation. As the shift toward EVs continues to accelerate, there is an increasing need for advanced engineering education and development tools that can address the complexity of their powertrain systems. Simulation, especially through model-based design, has become a critical asset in this field. It enables analysis, testing, and refinement of vehicle components and control strategies in a cost-effective and efficient way before moving to physical prototypes.

This project sets out to recreate and explore an electric vehicle drive simulation inspired by David McDonald's work, *Electric Vehicle Drive Simulation with MATLAB/Simulink*. The original model provides a foundation for studying the behavior of a basic EV motor-drive system, with a focus on energy flow during both acceleration and regenerative braking. By reconstructing this model, the project aims to develop a practical understanding of how key subsystems interact, including the motor, controller, and battery.

The main objective is to create a fully functional version of the Simulink model described in the reference paper. This involves applying the relevant equations and parameters for each major component: a DC permanent magnet motor, an idealized motor controller, a proportional-integral controller for system stability, and a simplified battery representation. Once completed and validated, the simulation will be used to examine energy flow and overall system performance under specific drive cycle conditions. Special attention will be given to evaluating power transfer efficiency during both motoring and regenerative phases. Through this process, the project not only strengthens theoretical knowledge of EV systems but also provides direct experience with industry-standard simulation tools for vehicle system analysis.

Modeling

To understand how an electric vehicle (EV) works, we built a model of its drive system using MATLAB/Simulink. We've constructed the drive model in Simulink by modeling its individual components and connecting them to the required inputs and outputs, seen in Figures 1 through 5. With the motor drive modelled, the next step is to input real-world data and run the simulations to see how it performs. Our model of the EV drive system is essentially broken down into four main parts, or subsystems. Each of these subsystems represents a key piece of the car's powertrain. We followed the general ideas and simplifications laid out by McDonald in "Electric Vehicle Drive Simulation with MATLAB/Simulink" to build the new model.

First, there is the matter of the electric motor as seen in Figure 1. For our project, we decided to use a DC permanent magnet motor. In the real world, most new EVs are using permanent magnet synchronous motors (PMSMs) as the main drive component, but the DC motor kept modelling simple, and it's what the original article used. Our motor model does a decent job of showing things like the energy lost in the windings (resistance) and the slight delay due to inductance. Since it’s a permanent magnet type, there is no worry about power loss in the field. To keep our model focused, we decided to exclude losses from things like friction, the inertia of the rotor, and other magnetic aspects.

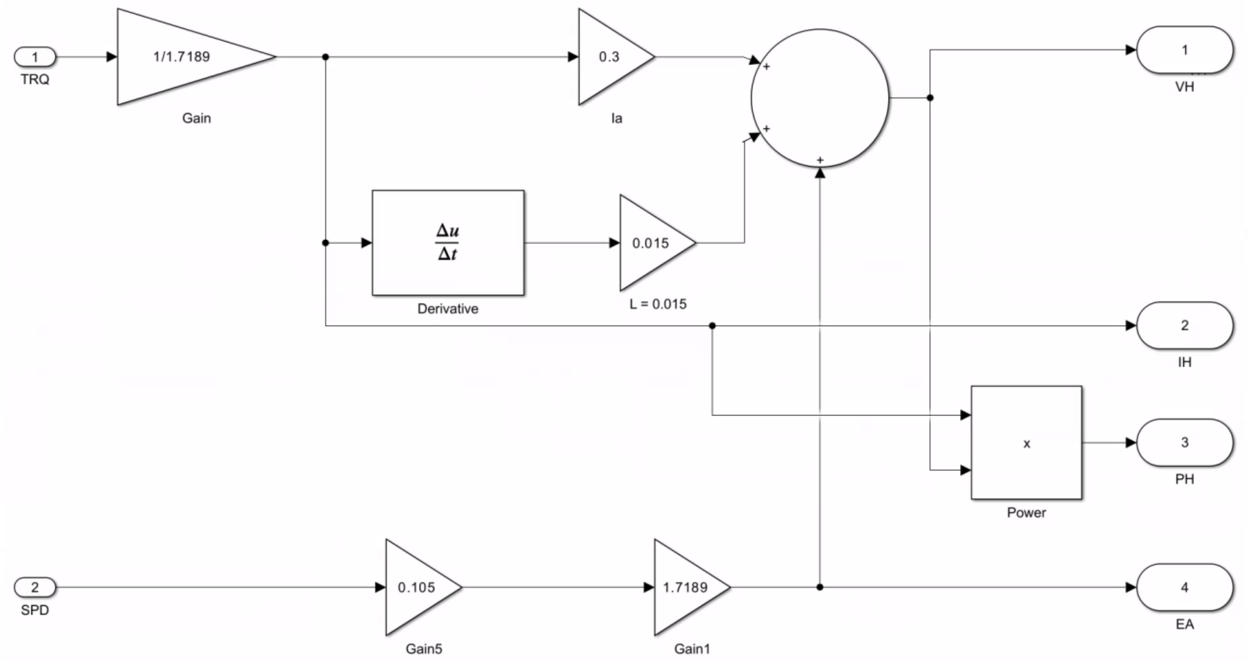

Figure 1. Motor Subsystem

Next is the motor controller subsystem, shown in Figure 2. We've treated this as a perfect, ideal controller – meaning we're assuming it doesn't lose any power and reacts instantly. Its main job in our simulation is to make sure the motor gets the right amount of voltage from the battery when it needs it. A key setting in this component is a gain referred to as 'K', which changes depending on the motor’s function at any point in time. This 'K' value also helps adjust how much the current system draws.

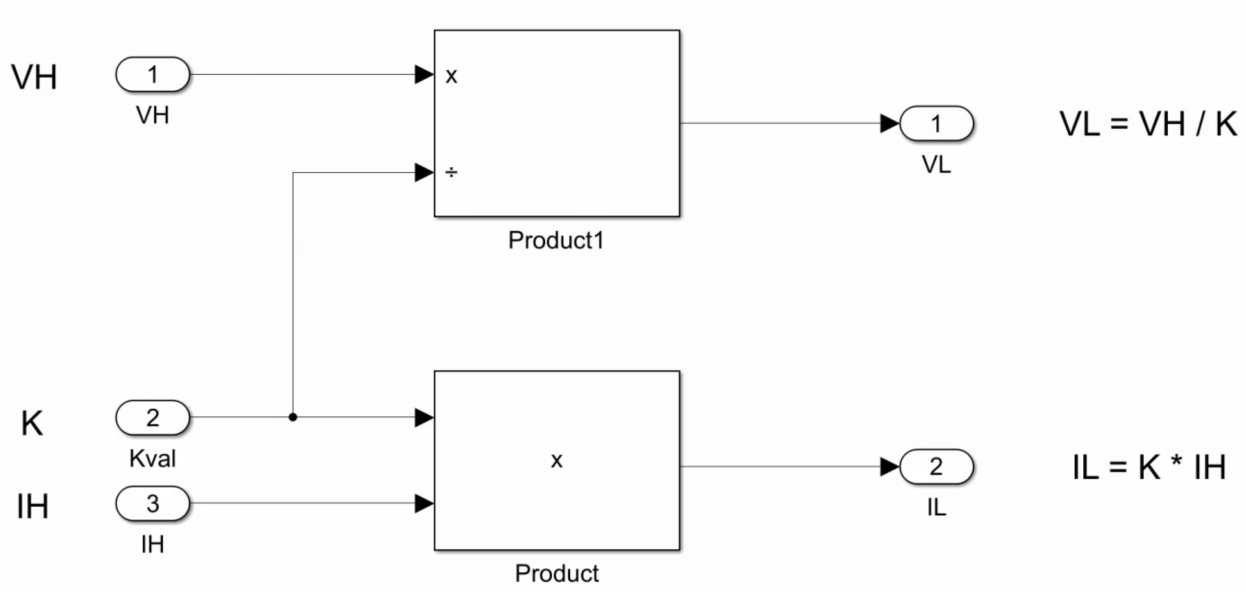

Figure 2. Motor Controller Subsystem

Then we have the battery, which Figure 3 lays out. We've modeled it as a basic voltage source (Eb) that also has some internal resistance. This is a standard way to show that real batteries aren't ideal and lose some power internally. We're assuming our battery responds instantly and that its internal voltage (Eb) stays constant. This part of the model takes the current voltage numbers from the motor controller to figure out what the battery's actual output voltage is. Then, it compares this to the ideal internal voltage (Eb) to find out if there's any difference, which we call the battery voltage error (Berr). This error signal is important for keeping the whole system in check.

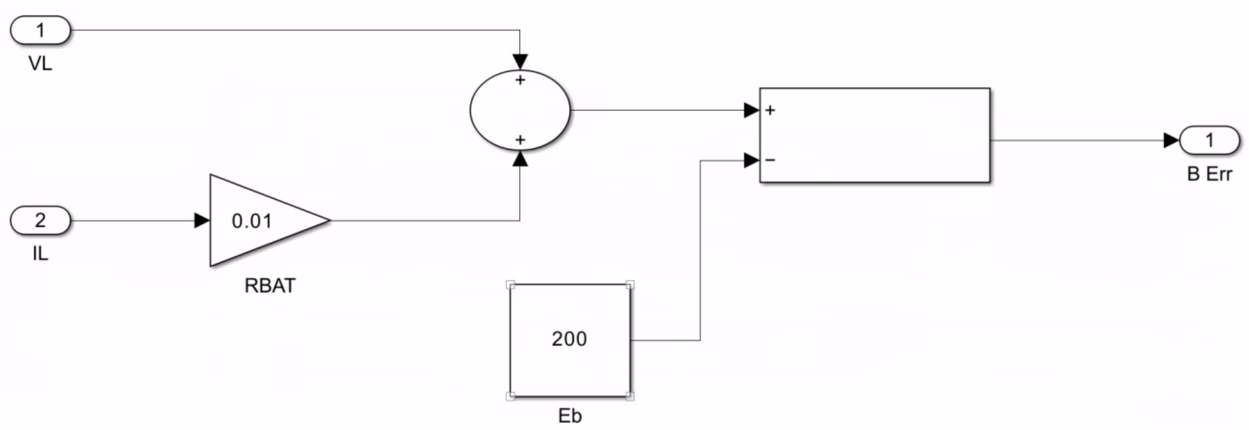

Figure 3. Battery Subsystem

Figure 4 shows the subsystem that serves as the central component to the motor drive – the Proportional-Integral (PI) controller. This unit is key to making sure the whole system runs smoothly and responds well. It takes the battery error signal (Berr) from the battery subsystem and, using set tuning values (proportional gain, Kp, and integral gain, Ki), the PI controller calculates the proper value for 'K'. This 'K' then goes back to the motor controller and determines further adjustments.

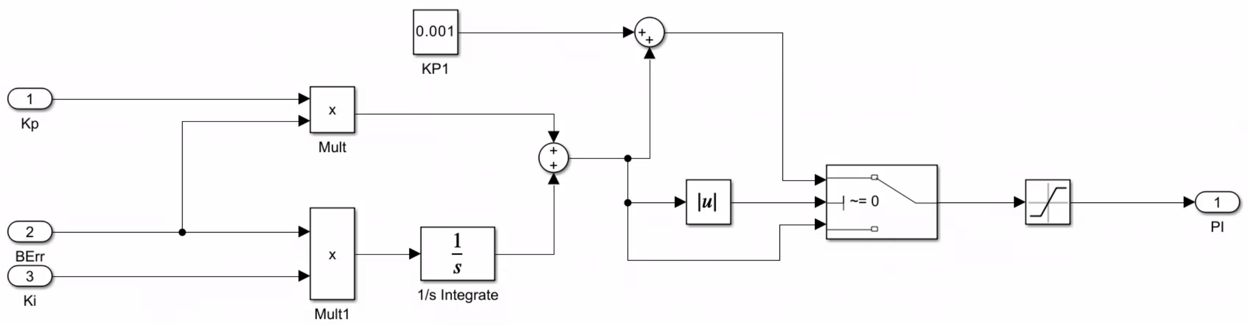

Figure 4. Proportional-Integral Controller

Finally, these individual parts are connected to create our complete model of the EV drive system as seen in Figure 5. With everything connected, we can run simulations to see how power moves through the system and how all the components work together when the vehicle performs a certain behavior, like speeding up or slowing down.

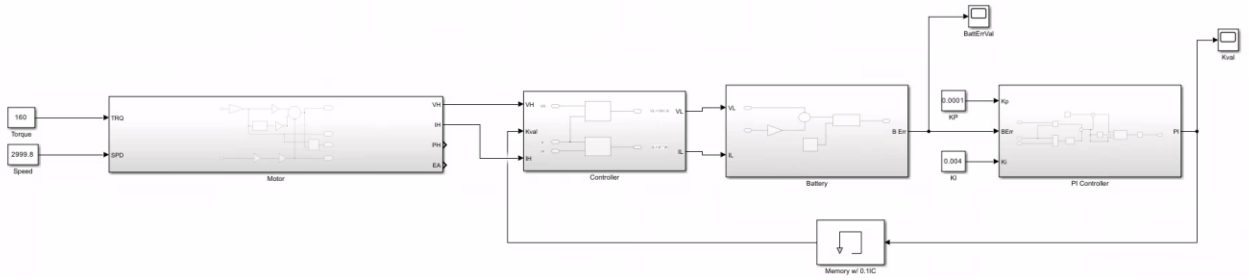

Figure 5. Overall System

The motor simulation would then be run using set values for the speed and torque inputs:

- Speed
  - Svals = [0 2000 3000 1000 1000]
  - Stime = [0 5 50 85 100]
- Torque
  - Tvals = [0 330 330 160 160 -220 -220 0 0]
  - Ttime = [0 5 10 15 50 55 80 85 100]

These values would be inserted into their own lookup tables, which provide a map of input values from data arrays. The input tables for speed and torque were then used as the input data for the modelling of the motor's behavior at certain values.

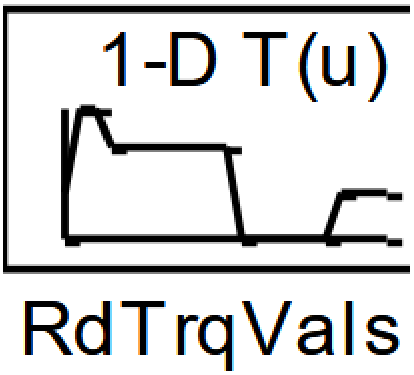

Figure 6. Graphical representation of torque data via lookup table

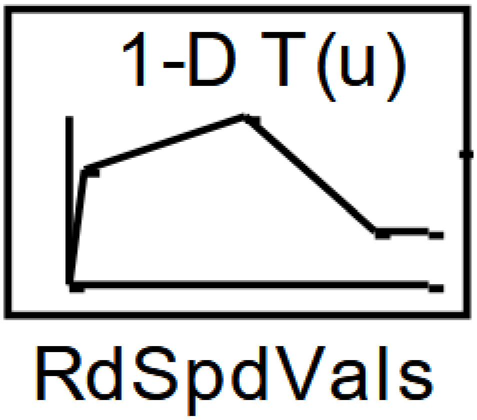

Figure 7. Graphical representation of speed data via lookup table

Testing Procedure

To validate the electric vehicle drive simulation, a test was run over a simulated span of 0 to 100 seconds, with data captured every 0.01 seconds. This high-resolution sampling offered a clear view of how the system responded dynamically over time.

The first approach involved running the Simulink model using the MATLAB script provided in the appendix of the reference paper. However, this immediately triggered a warning related to the derivative block in the motor subsystem. The warning stated, 'Unable to resolve name [scope name].signals.values', which stemmed from the block receiving a discrete input signal when it was expecting a continuous one. To address this, the block was replaced with its discrete-time counterpart.

Once this fix was in place, the simulation was executed successfully, drawing input values from predefined lookup tables. However, a more persistent problem surfaced when trying to export data from Simulink scopes to the MATLAB workspace. Specifically, the error 'Unable to resolve name [scope name].signals.values' kept arising. Upon further digging, it became clear that the core model file (e.g., motor.mdl) was being shadowed by another file of the same base name, but with a .slx extension. Removing the .slx file cleared up the shadowing issue, yet the original data export error remained, in turn blocking the script-based extractions.

To get around this, the scopes in the Simulink model were reconfigured to generate direct graphical outputs. These were set up to display all essential signals with fully labeled axes for time and amplitude. Running the simulation with this setup yielded real-time plots that visually showcased the system's behavior across the full test window. A careful side-by-side comparison was then made between these plots and the corresponding figures from the original paper which is discussed in the following section.

Results Comparison

The results from the Testing phase of the experiment are displayed below, as we go through a thorough visual comparison between the results the team were able to produce, and the ones produced in the reference paper. Through the comparison, it was shown that the findings in the testing phase correspond quite accurately to the original findings for all the key parameters, which are discussed throughout this section.

Below, in Figure 8, the graphs for road speed, torque, and power can be found. The graphs depicted from our Simulink model closely represented similar findings to those of the reference paper. Specifically, within the reference paper, looking at Figure 9, which shows the originally produced graphs for road speed, torque and power. It can then be seen that the team's produced simulation graphs take on a similar shape, depicting near same peak values and transition points.

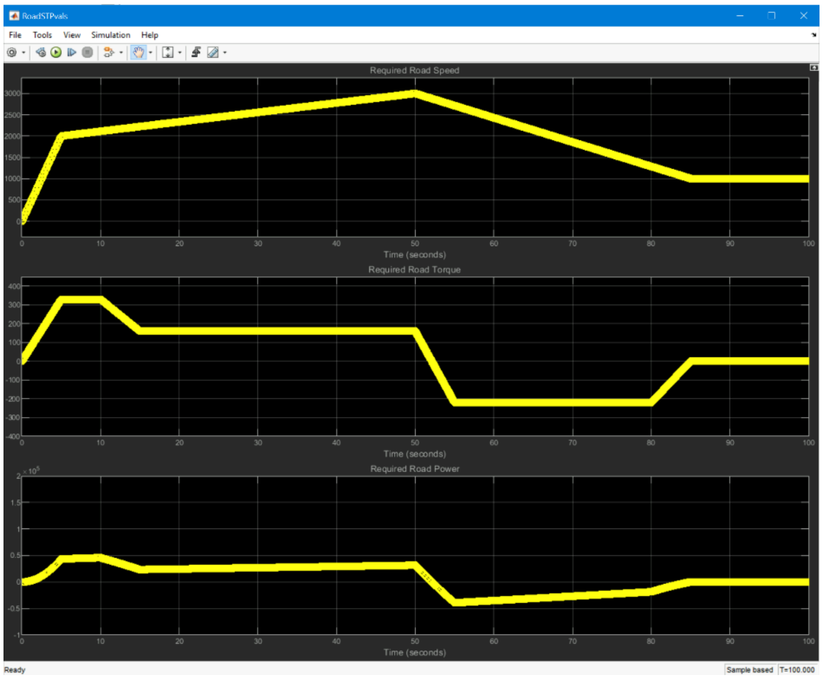

Figure 8. Resulting Graphs for Road Speed, Torque, and Power

The team then turned towards some electrical aspects of the simulation, such as the motor voltage, current, and power. Our results for the graphs of those values throughout the sim can be found on the following page in Figure 9. As for the results originally produced by the reference article, please refer to their figure number 10. As the two are compared, it can easily be seen that the team was on the right track as the graphs produced share a similar representation reflecting the motors response to the drive cycle.

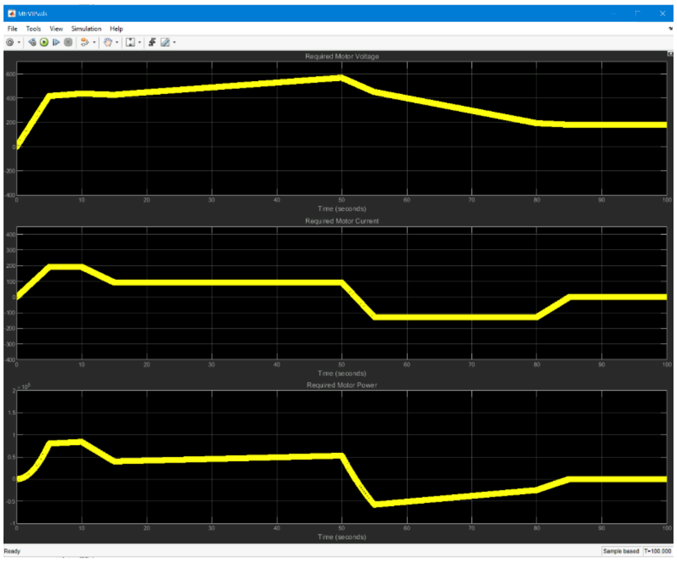

Figure 9. Resulting Graphs for Required Motor Voltage, Current, and Power

Then moving to the comparison of the graphs for the battery component of the team's drive simulation. Found in Figure 10 in this report and its counterpart to compare would be Figure 11 in the reference. The graphs shown depict representations for the battery voltage, current, and power throughout the simulation of a drive cycle. Once again, the team was able to clearly represent similar findings to those produced in the reference.

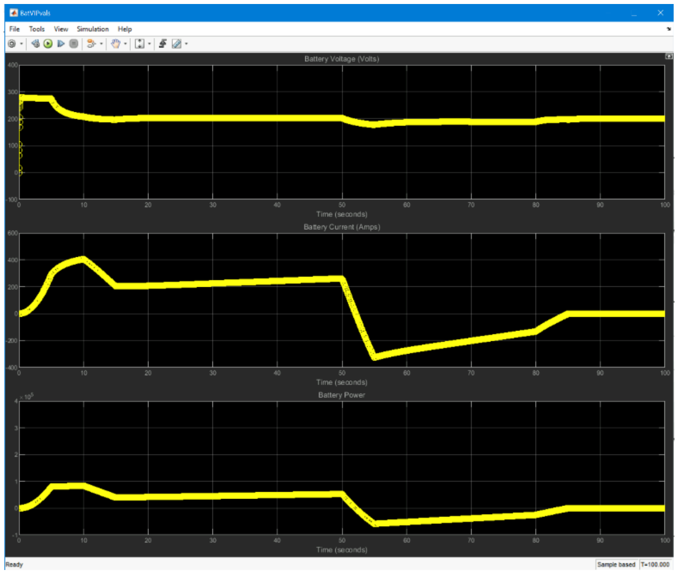

Figure 10. Resulting Graphs for Battery Voltage, Current, and Power

The image depicted in Figure 11 shows the ‘battery voltage error’. It shows an initial spike, followed by a stabilizing pattern almost identical to that of the reference (reference article Figure 12). The team’s graph for the controller gain can also be found below in Figure 12. The ‘controller gain’, represented as K, closely aligns with the profile found in Figure 13 of the reference paper, including the peak values and its corresponding responses to changing speed and torque demands.

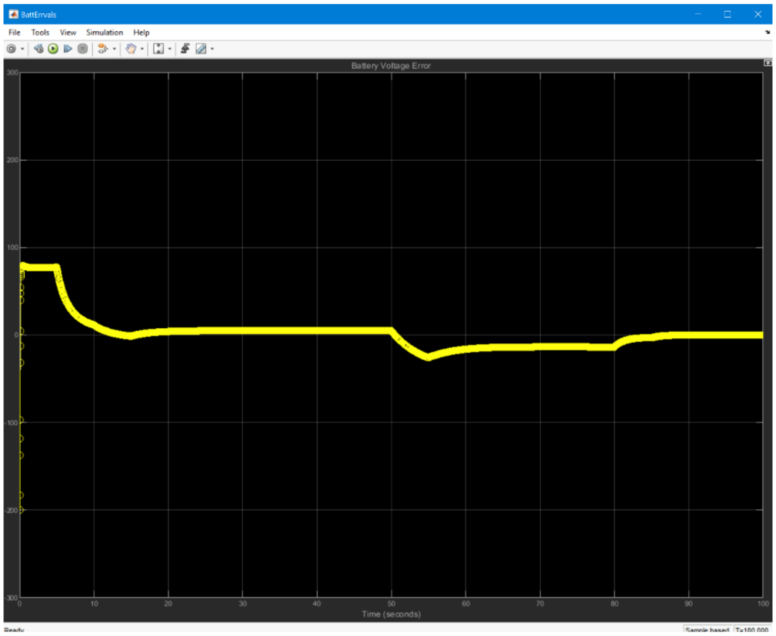

Figure 11. Resulting Graph from The Battery Voltage Error

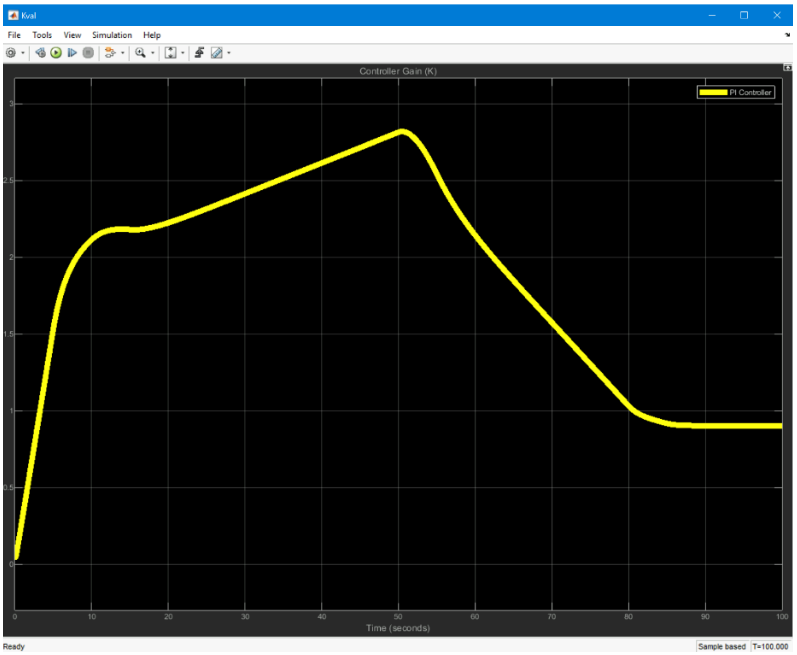

Figure 12. Resulting Graph for The Controller Gain(K)

Conclusion

This project provided the team with substantial insights into the operational principles of electric vehicle powertrains and offered practical experience in the application of model-based design using MATLAB along with its counterpart Simulink. One of the main things the team got out of this project was a much better grasp on how the different parts of an EV work together. This includes the DC permanent magnet motor, a simplified motor controller, a basic battery model, and the very important Proportional-Integral (PI) controller that keeps the whole system stable. We took the theories and formulas from the reference article and put them into practice by building a working model in Simulink. This meant creating each part of the system as a separate block and then connecting them all to make a complete model that could simulate the EV running (motoring) and slowing down (regenerative braking).

When we ran our simulation using the same kind of speed and torque input as the original study, the way our system behaved lined up well with what was reported in the original paper. Key things like how the motor's voltage, current, and power changed, how the battery's voltage, current, and power responded, and how the controller's gain (K) adjusted all showed a strong similarity. This was specifically shown within our ‘Results Comparison’ section of the report. Depicted within this part are all the necessary graphs needed to show our working Electric Vehicle Motor Drive. Our graphs, corresponding with various values throughout the simulation, lined up very similarly with those produced within the original reference paper. We were particularly pleased to see important actions like the switch between powering the wheels and recharging the battery, the current changing direction as expected, and the PI controller making the necessary adjustments. This similarity shows we understood the original material and built a solid version of their simulation.

References

[1] D. McDonald, "Electric Vehicle Drive Simulation with MATLAB/Simulink," in Proceedings of the 2012 North-Central Section Conference, 2012.

Appendix

Electric Vehicle Drive Simulation with MATLAB Simulink.pdf

Link to a pdf formatted file of the scholarly article used as reference for the above experiment/project.