Team 33 Emulation

Neuromorphic Hardware

Daniel Ainley, Madison Freeman, Omar Mostafa, David Owen

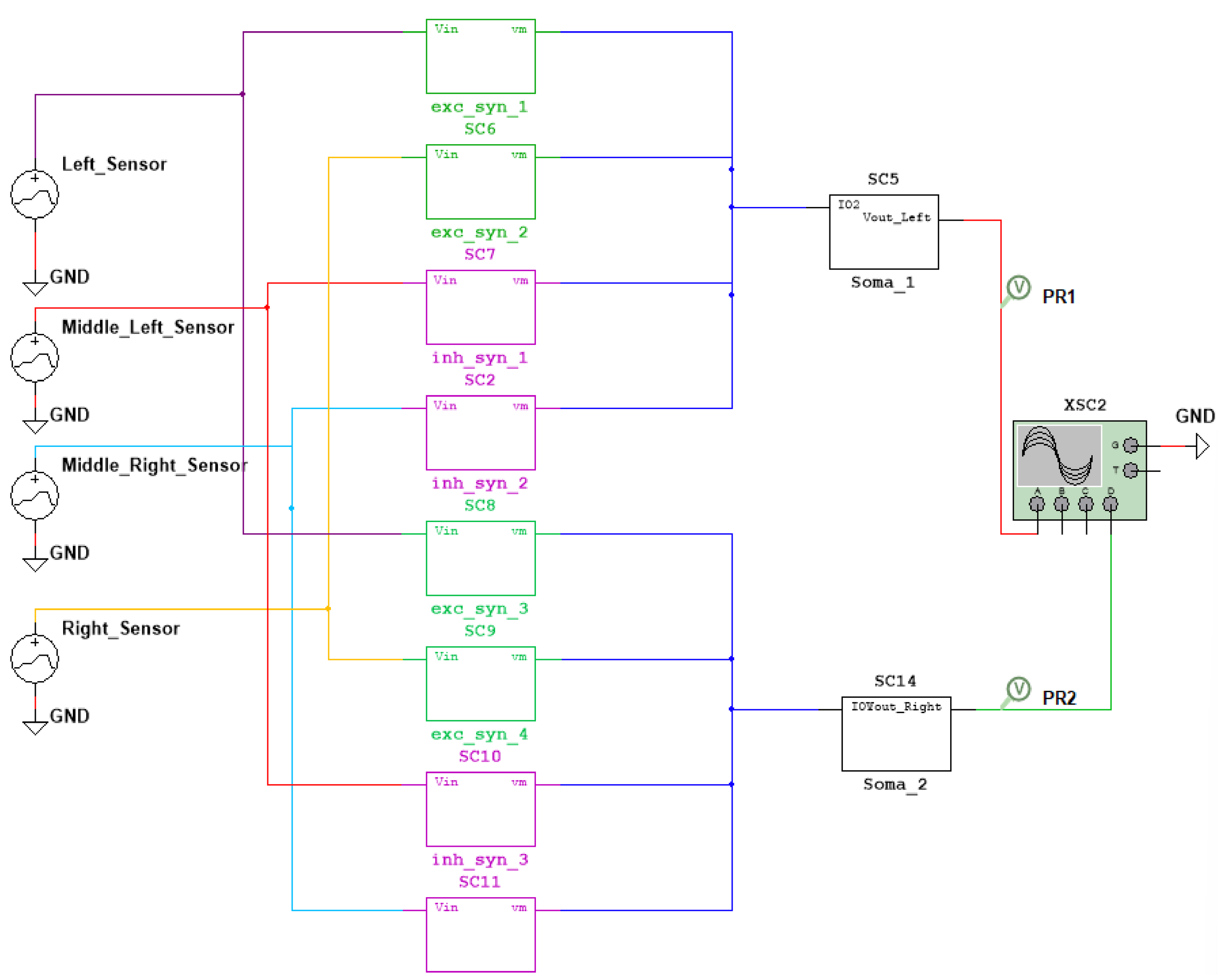
PURDUE UNIVERSITY

Polytechnic Institute

We would like to extend our sincere gratitude to Masoome Fatahi for her enduring patience and direction throughout the project. We also thank Nick Wacyk and Toby Yoo for their guidance and valuable insights from previous project knowledge. Finally, we are grateful to Professor Nawrocki, Professor Milton, and Dr. Berry for their mentorship and support.

OBJECTIVE

The objective of this project was to collect and verify the performance of a line following EV by using an existing FPGA based Spiking Neural Network (SNN) and then developing an analog SNN. This involved ensuring that the analog SNN could replicate the behavior of the digital FPGA version while interfacing with analog motor drivers and correctly following the line. In addition, the team aimed to create a reliable method for error detection and data collection to support system validation



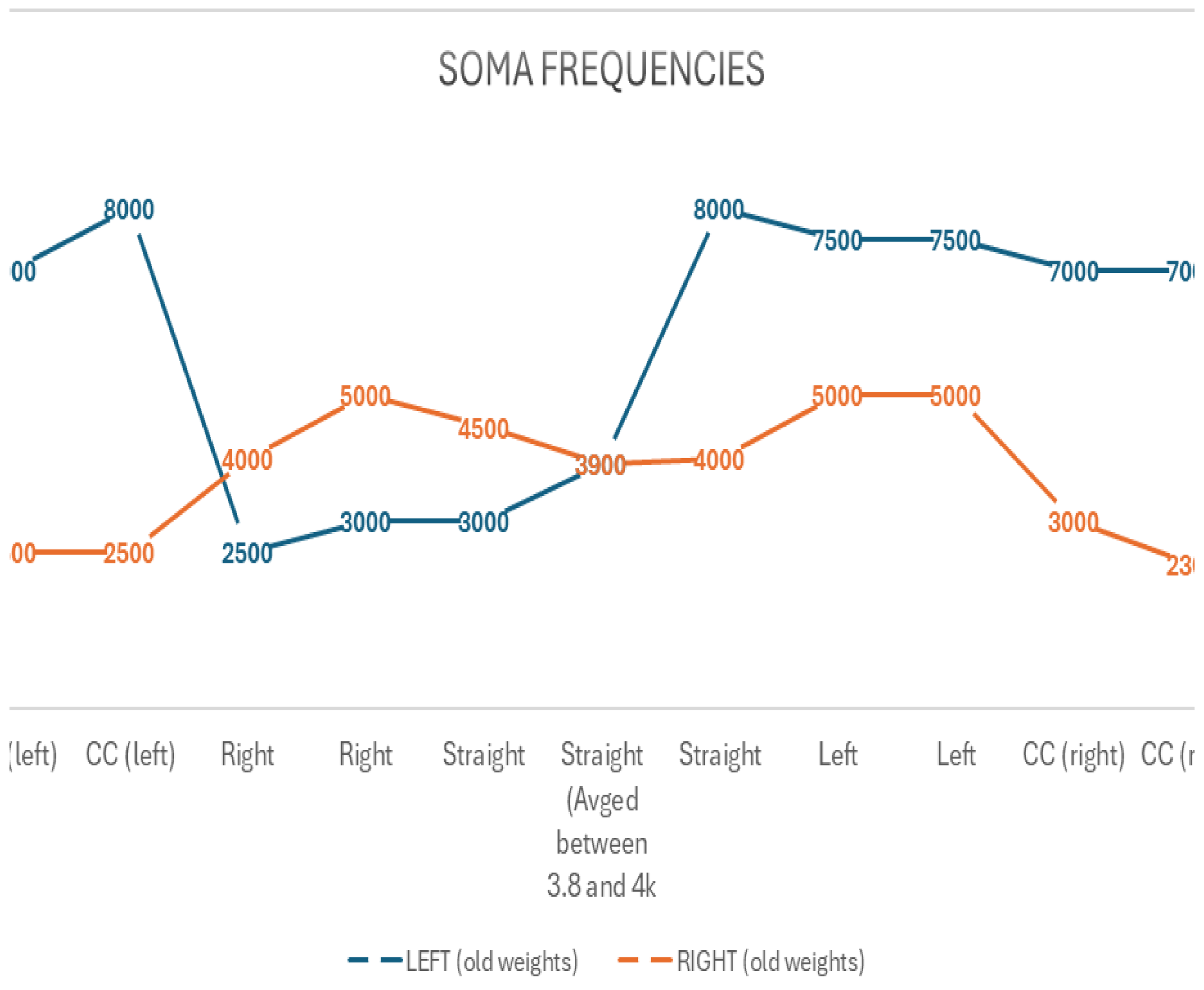
CUSTOMER PROBLEM AND BACKGROUND

Spiking neural networks (SNNs) represent a major advancement by mimicking the brain's biological processes, transmitting information through discrete, time-dependent spikes rather than continuous signals. This spike-based approach improves computational efficiency and drastically reduces power consumption, making SNNs ideal for real-time applications such as autonomous vehicle control systems. Unlike ANNs, which encode information through firing rates, SNNs use precise spike timing, allowing for a faster and more dynamic decision-making process. These advantages make SNNs a promising solution for enhancing performance and reducing the energy footprint in autonomous systems.

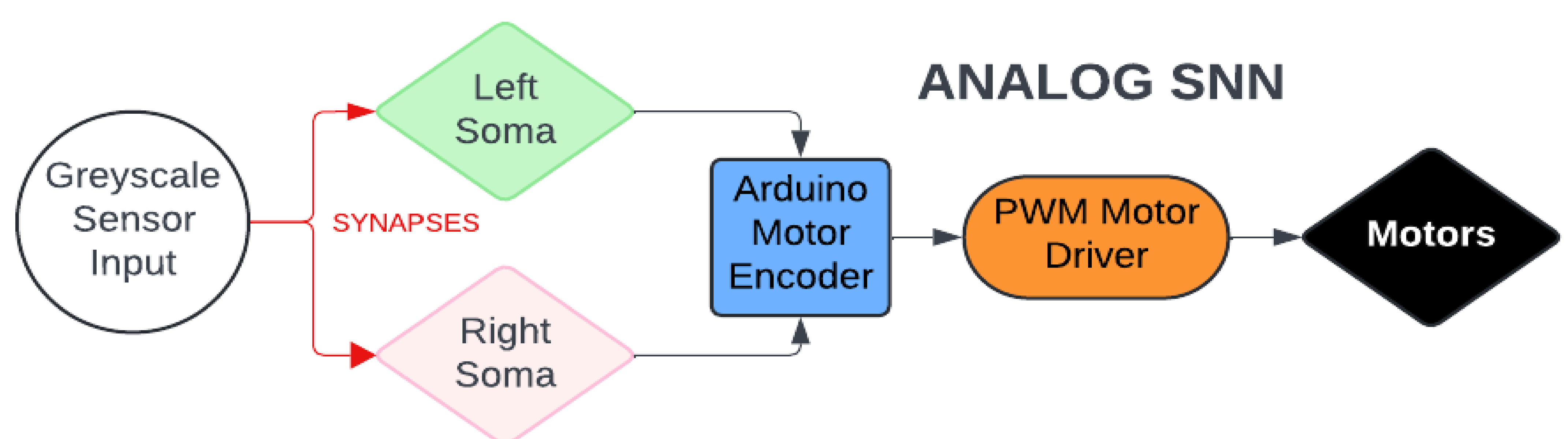
The existing FPGA-based model, although functional, was limited by its size, power consumption, and digital architecture. By creating a breadboard-based analog SNN, the team aimed to replicate the digital system's performance while improving responsiveness and energy efficiency. This project also worked to establish a foundation for future compact circuit integration, offering a scalable, low-power alternative for autonomous applications.

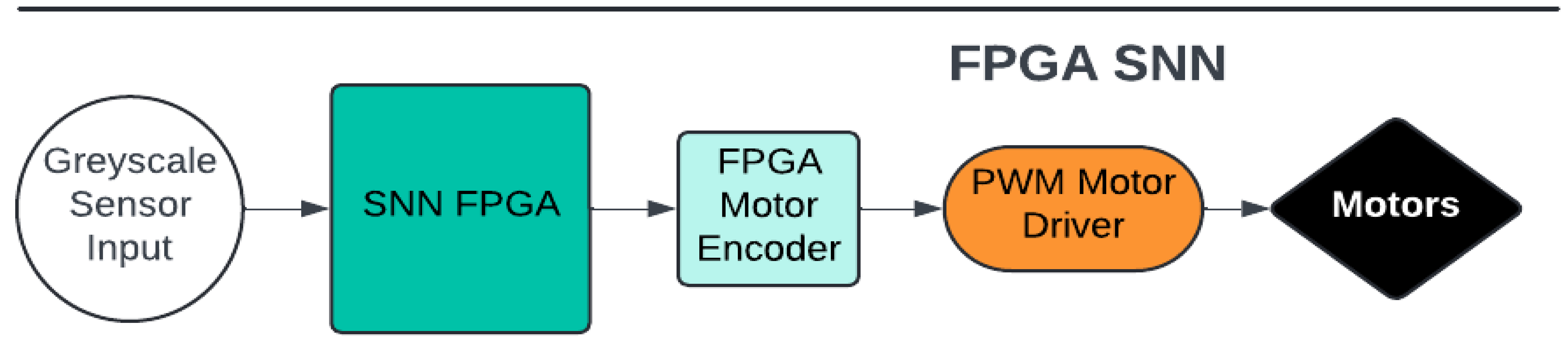
CONCEPTS AND EXPERIMENTATION

Frequency data from the SNN somas was first collected using an oscilloscope while manually testing the car on the line at different angles. This data was graphed to visualize the "zero crossing" points where the car should ideally drive straight. Frequency "bins" were created to define control cases. As seen in Figure 2, when the left soma frequency was higher, the car would turn left; when the right soma frequency was higher, the car would turn right. Although a Straight function was implemented, the car failed to utilize that set of frequencies indicating an issue with the voltage weights. Additionally, an error detection system was developed using a Raspberry Pi and camera setup which tracked the centermost point of the line in real time and recorded timestamped video data during each run. This allowed for side-by-side comparisons of vehicle behavior across tests and provided insight into inconsistencies or deviations.

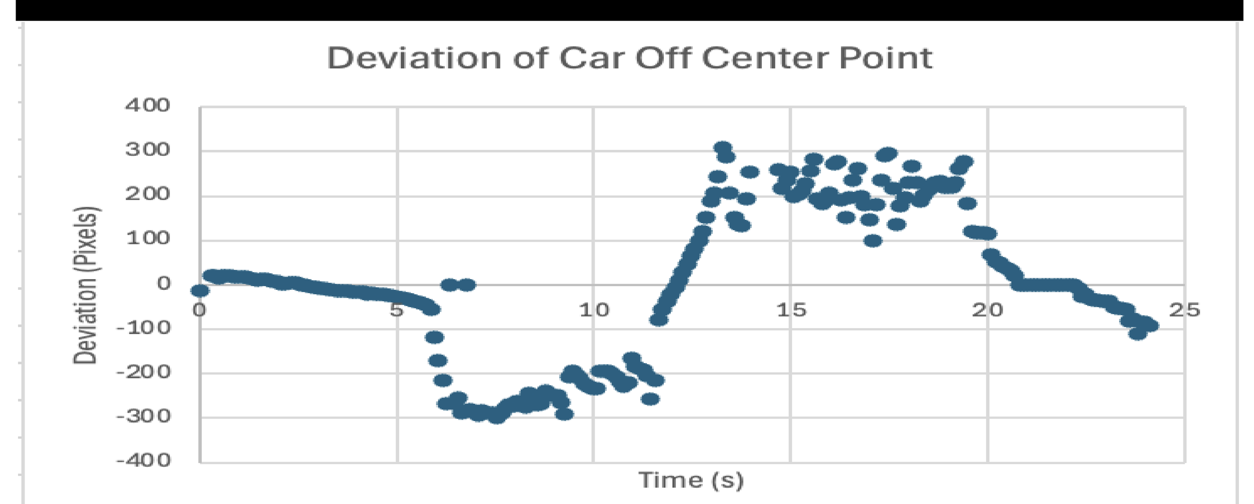


REQUIREMENTS AND FINAL DESIGN





TESTING RESULTS



CONCLUSION AND RECOMMENDATIONS

The team successfully integrated the analog spiking neural network (SNN) into the autonomous vehicle by building an embedded system with an Arduino microcontroller. The Arduino served as a motor encoder, interpreting the SNN soma spike frequencies and converting them into motor control commands. Using this setup, the vehicle was able to move down the line, although only Left and Right directional commands were implemented, with no Straight driving function. As a result, the car exhibited slow, jittery movement as it constantly corrected its path without a neutral driving state. Future improvements include refining the voltage weight settings, which determine the strength of inhibitory and excitatory synapses, to create a smoother zero-crossing transition between left and right frequency thresholds. This adjustment would allow for more stable driving behavior and quicker response to sensor input. Additionally, creating a printed circuit board (PCB) would help organize the analog wiring, minimize signal noise, and reduce confusion during troubleshooting. A PCB would also significantly shorten testing and adjustment time, improving both system reliability and ease of integration for future autonomous applications.