Gesture Based Turn Signaling System

ECE 445 Individual Progress Report

Kaylan Wang

kaylanw2@illinois.edu
Team 20
Professor: Viktor Gruev
TA: Sanjana Pingali

Spring 2024

Contents

1 Introduction

For our ECE 445 Senior Design project, our group is designing a gesture-based turn signaling system for riders. The system will use a combination of sensors and a microcontroller to detect the rider's hand gestures and activate the appropriate turn signal. The goal of the project is to improve the safety of cyclists by providing a more intuitive and convenient way to signal turns, slowing down, and accidents.

My responsibility within this project has been the sensor subsystem, focusing on the design and integration fo the IMU breakout board. The sensor subsytem's role is to capture the user's arm movements through the IMUs and communicate this data to the control unit via the I2C protocol. Initially, our design incorporated 2 LSM9DS1 IMUs [1]; however, due to some supply constraints, we have since decided to use 4 ICM-20948 IMUs [2] (2 on each arm) to better model the user's arms. This shift required a redesign of the breakout board and the communication protocol between the IMUs and the control unit.

Currently, I am focusing on testing and verifying the communication between the IMU and the control unit, making sure the data is being transmitted correctly and minimizes noise through data filtering. Filtering is crucial to ensure that the control unit can accurately interpret the user's gestures, and will be discussed in more detail in the following sections. Additionally, I am doing research on how to communicate the data between the IMUs and the control unit, as well as possibly using a Recurrent Neural Network (RNN) to interpret the data.

2 Individual Design Work

2.1 Design Considerations

2.1.1 Sensor Subsystem

The first thing I had to consider when designing the sensor subsystem was the availability of the IMUs. As mentioned earlier, we initially planned to use the

LSM9DS1 IMUs, but they are no longer available to purchase, so we had to switch to the ICM-20948 IMUs. Additionally, we decided to switch to the I2C from SPI due to the simplicity of the I2C protocol when designing the PCB for the IMU breakout board (see figure 1). The I2C protocol requires fewer pins and is easier to route on the PCB, which is crucial for the small size of the breakout board. The design of the PCB is based on the typical operating circuit from the datasheet [2] while omitting the unnecessary components for the SPI interface.

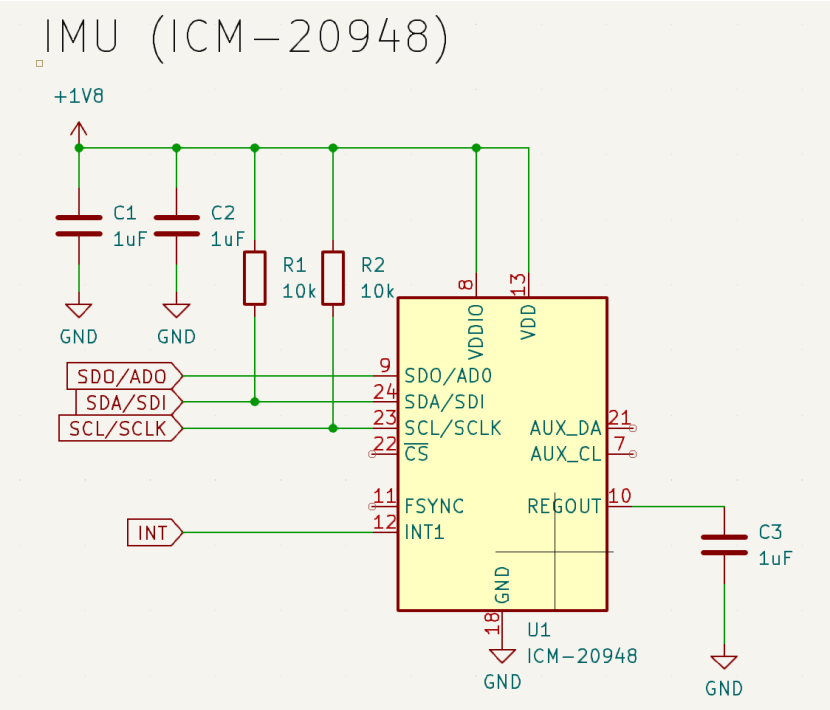

Figure 1: IMU schematic

2.1.2 Software

It is likely that the data from the IMU will be noisy and require filtering to accurately interpret the user's gestures. The noise will cause the data from the IMU to be inaccurate after a short period of time. I am currently researching different filtering techniques to implement on the control unit to

minimize noise. One possible filtering technique is the Kalman filter, which is a recursive algorithm that estimates the state of a system from a series of noisy measurements. The Kalman filter is particularly useful for systems with a high degree of uncertainty, such as the IMU data [3].

I am also considering using a Recurrent Neural Network (RNN) with long short term memory (LSTM) to interpret the data from the IMUs. An RNN is a type of neural network that is well-suited for sequential data, such as the time-series data from the IMUs because it can learn the patterns in the data and make predictions based on the previous data points. The LSTM architecture is particularly useful for time-series data because it can remember information from previous time steps and use it to make predictions about future time steps [4]. A RNN with LSTM has a connection going forward to learn from previous data. Additionally, see figure 2 for a high-level overview of how an RNN with BiLSTM works, where there are two connections, one from the original LSTM going forward, and one going backwards to learn from future data.

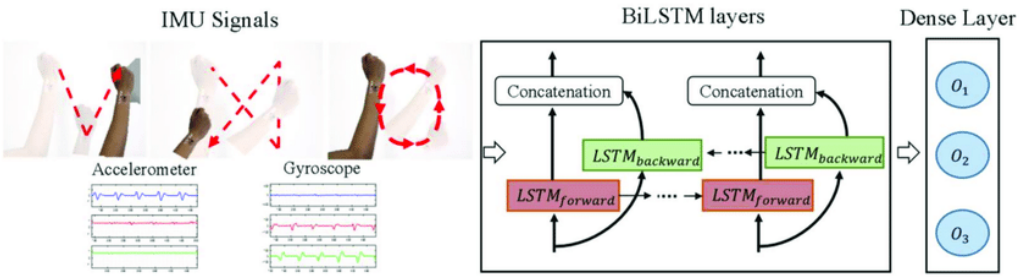

Figure 2: RNN with BiLSTM high level diagram [5]

The RNN would be trained on a dataset of gestures to learn the patterns in the data and predict the user's intended gesture. This approach would require a significant amount of data to train the RNN, so it is likely we adapt an existing gesture recognition model to our specific use case. This is still in the beginning stages of research, and I will continue to explore this option as we progress in the project.

2.2 Testing/Verification

Testing and verification for the sensor subsystem will be done in two stages: first, testing the communication between the IMUs and the control unit, and second, testing the filtering of the IMU data. The first stage will involve sending data from the IMUs to the control unit and verifying that the data is being transmitted correctly. This will involve writing test code to read the data from the IMUs and send it to the control unit via the I2C protocol. The second stage will involve filtering the data from the IMUs to minimize noise and ensure that the control unit can accurately interpret the user's gestures. This will involve writing test code to apply different filtering techniques to the IMU data and verifying that the filtered data is more accurate than the raw data.

2.2.1 Communication

For the first stage, we will write test code to read the data from the IMUs and send it to the control unit via the I2C protocol. We will first be using the dev board for the ESP32 and pre made breakout boards purchased from Amazon to test the communication between the IMUs and the control unit. Once we have verified that the communication is working correctly, we will complete the custom PCB for the IMU breakout board and test the communication with the custom PCB for the control unit. The communication will be tested by sending data from the IMUs to the control unit and verifying that the data is being transmitted correctly. In the case that the communication does not work properly, we will spend some time debugging the issue and if it is not resolved, we will consider switching to a different communication protocol such as SPI, and if that doesn't work, we will likely just use the pre made breakout boards for the IMUs.

2.2.2 Software

For the second stage, we will write test code to apply different filtering techniques to the IMU data and verify that the filtered data is more accurate than the raw data. We will first test the filtering techniques on the dev board for the ESP32 and pre made breakout boards purchased from Amazon. Once

we have verified that the filtering is working correctly, we will complete the custom PCB for the control unit and test the filtering with the custom PCB. The filtering will be tested by applying different filtering techniques to the IMU data and verifying that the filtered data is more accurate than the raw data. In the case that the filtering does not work properly, we will spend some time debugging the issue and if it is not resolved, there isn't much we can do other than try different filtering techniques. The data will likely need to be filtered before it is sent to out algorithm for interpretation or else we will get inaccurate results.

For the RNN, we will first train the model on a dataset of gestures to learn the patterns in the data and predict the user's intended gesture. We will then test the model on a separate dataset of gestures to verify that the model can accurately interpret the user's gestures. This will involve writing test code to apply the model to the dataset of gestures and verifying that the model can accurately predict the user's intended gesture. Once the model performs well on the test dataset, we will integrate it into the control unit and test it with the IMUs to verify that it can accurately interpret the user's gestures in real-time. It is likely that we will need to optimize the model fro deployment on the microcontroller to ensure that it can run in real-time; this might involve reducing the complexity of the model via techniques such as pruning or using a more efficient implementation of the model. In the case that the model does not perform well on the test dataset, we will spend some time debugging the issue and if it is not resolved, we will consider a more straightforward algorithm for gesture recognition, such as a decision tree.

3 Conclusion

Throughout the progression of this project, I have been focusing on the sensor subsystem, specifically the IMU breakout board and the communication between the IMUs and the control unit. I have designed the PCB for the IMU breakout board and am currently testing the communication between the IMUs and the control unit. I am also researching different filtering techniques with my teammmates to minimize noise in the IMU data and considering using an RNN with LSTM to interpret the data from the IMUs. I would say this part of the design is equal in workload compared to

my teammates, as we all have different responsibilities in the project.

Currently, we are slightly behind schedule due to the switch in IMUs and the need to redesign the breakout board and communication protocol. Additionally we were unable to test the custom PCBs (control unit and IMU) before the second round of orders, so we missed out on a potential time saver there. However, we are confident that we can catch up and complete the project on time.

Moving forward, our plan comprises of testing of the IMU-control unit communication and the implementation of data filtering techniques. Specifically, within the next two weeks, we aim to finalize the communication protocols and begin integrating data filtering algorithms. Concurrently, I will advance my research into utilizing an RNN for gesture data interpretation, with a goal to have a preliminary model ready for training on gesture data within the next two weeks as well. This approach is ambitious, considering the complexity of neural network models and the need for a substantial dataset for training. However, it is also realistic given our current progress and the resources available to us, including pre-existing datasets and models that can be adapted to our needs.

The ultimate objective is to integrate the RNN model with the sensor and control subsystems, ensuring real-time, accurate gesture recognition. By the project's conclusion, we anticipate meeting this goal and delivering a fully functional gesture-based turn signaling system that enhances the safety and convenience of cyclists.

References

[1] *LSM9DS1: iNEMO inertial module: always-on 3D accelerometer and 3D gyroscope*, STMicroelectronics, Geneva, Switzerland, 2015, rev 4. [Online]. Available: https://www.st.com/resource/en/datasheet/lsm9ds1.pdf

[2] TDK Invensense, *ICM-20948 Datasheet*, TDK Invensense, 2016, v1.3. [Online]. Available: https://invensense.tdk.com/wp-content/uploads/2016/06/DS-000189-ICM-20948-v1.3.pdf?ref_disty=digikey

[3] N. P. Palanisamy, "Filtering of imu data using kalman filter," Masters Project, California State University, Sacramento, Sacramento, California, Dec. 2016, available from Sacramento State University's Scholarly Works. https://hdl.handle.net/10211.3/182681.

[4] IBM, "What are recurrent neural networks?" 2024, accessed: 2024-03-27. [Online]. Available: https://www.ibm.com/topics/recurrent-neural-networks

[5] E. Valarezo Añazco, S. Han, K. Kim, P. Rivera, T.-S. Kim, and S. Lee, "Hand gesture recognition using single patchable six-axis inertial measurement unit via recurrent neural networks," *Sensors*, vol. 21, p. 1404, 02 2021.