Extending IMU Degrees of Freedom for Pose Estimation Using AI on Chip

Team 71

Chirag Rastogi and Lukas Zscherpel

Introduction

Who Are We?

Chirag Rastogi

- Computer Engineer
- Created software to run the algorithms and designed the Neural Network

Lukas Zscherpel

- Computer Engineer
- Designed and Programmed PCB

Objective

The Problem

What is an IMU?

Inertial Measurement Units identify position in space using 3 sensors:

- **Accelerometer**
  - Measures linear acceleration in three dimensions
  - Used to estimate velocity and position over time
- **Gyroscope**
  - Measures angular acceleration in three dimensions
  - Used to estimate orientation over time
- **Magnetometer**
  - Measures the direction of the Earth's magnetic field
  - Used to estimate orientation with respect to Earth's magnetic field

Issues With Current IMU's

- **Accuracy**
  - To estimate position in space, you differentiate sensor outputs
  - Differentiating the data causes small errors to explode
  - Most affordable IMU's have low levels of accuracy and high error
- **Price**
  - There are many IMU's with high accuracy available
  - High accuracy IMU's typically run for hundreds of dollars

The Solution

Using Traditional Filters

- **Kalman**
- **Madgwick**
- **Mahony**

Using AI

- **RNN**
- **LSTM**
- **Bidirectional LSTM**

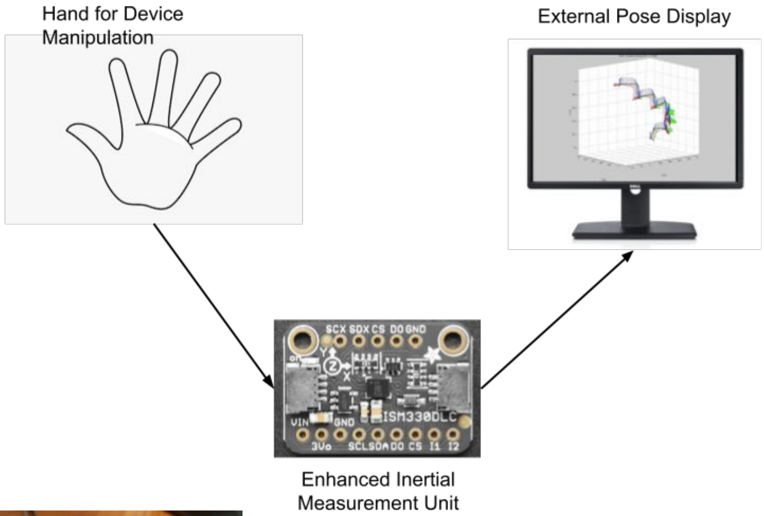
An AI on chip solution may have the potential to reduce the cost of 9DOF IMU sensors by enabling the integration of multiple sensors and processing functions onto a single chip, which can simplify the design, reduce the bill of materials, and lower the manufacturing costs.

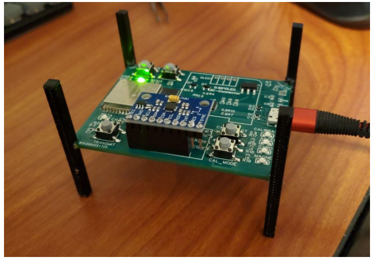
High Level Overview

High Level Overview

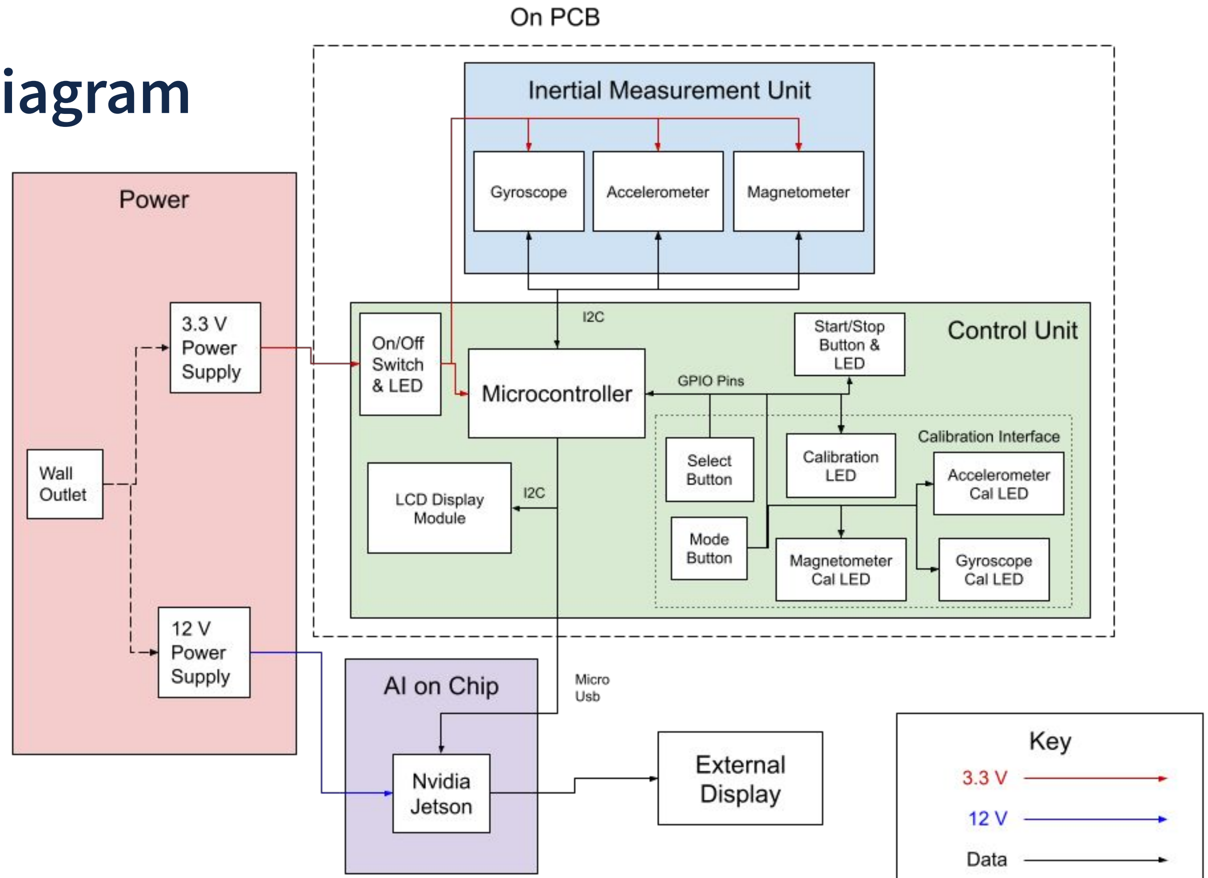
Project Components:

- **IMU Dev Board**
  - Provides sensor data from IMU to external device through USB
  - Allows for IMU to be easily swapped
  - Provides UI for IMU calibration
- **Data Processing and Display**
  - Processes data on GPU or CPU
  - Improves accuracy of the data using filtering or AI
  - Displays the data through a 3D visual of device orientation





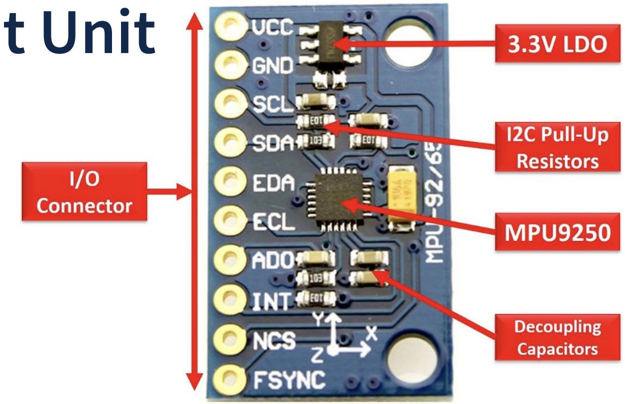
Block Diagram

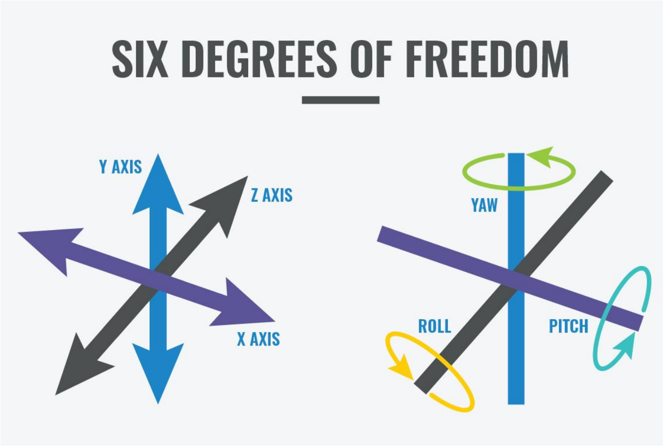


Subsystem 1: Inertial Measurement Unit

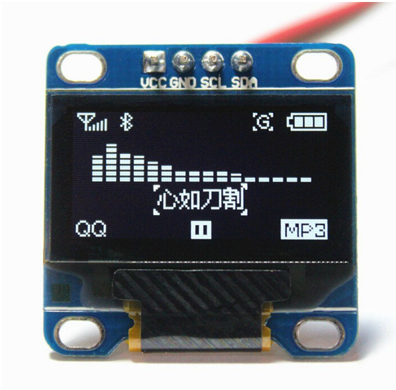
Standard IMU’s

- **MPU 6050**
  - 6 DOF
  - Includes accelerometer and gyroscope
  - ~$3
- **MPU 9250**
  - 9 DOF
  - Includes accelerometer, gyroscope and magnetometer
  - Offers improved sensor accuracy over MPU 6050
  - ~$6



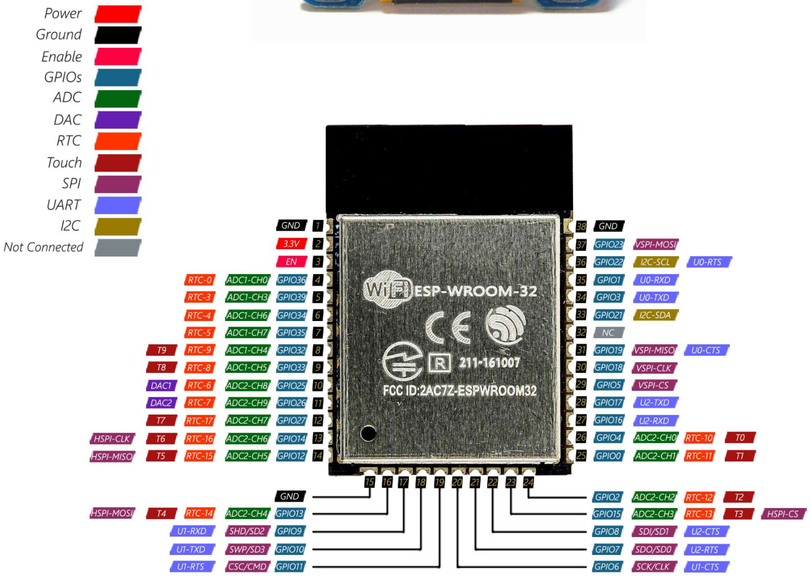


Subsystem 2: Control Unit



Components

- **ESP32-WROOM-32 microcontroller**
  - Reads data from IMU through I2C protocol
  - Standard microcontroller
  - Affordable ~ $5
  - Low power consumption
- **USB to UART IC - CP2102N**
  - Interface between microcontroller and USB connection
  - Compatible with standard serial IO protocols and libraries
- **128x64 OLED display**
  - Helps interface with user for calibration
  - Displays data collected in real time



Subsystem 3: Position Estimation using AI on Chip

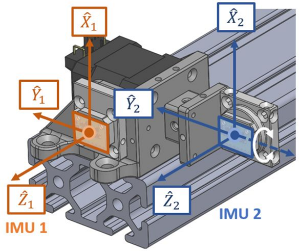
The data we used for this project is collected from 2 IMUs as shown below. The IMUs used are MPU-6050.

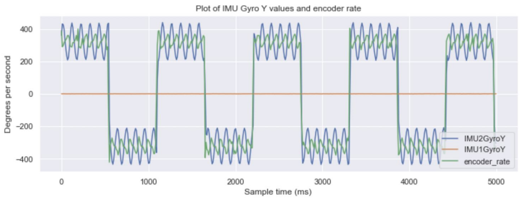
We investigated to what extent the limitations of standard algorithms can be overcome by means of neural networks

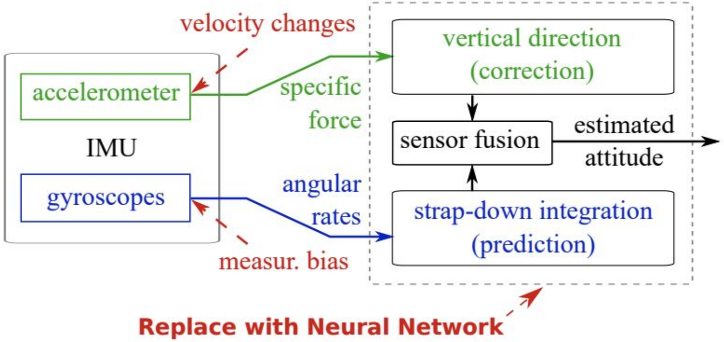
```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_2 (LSTM)               (None, 64)                18176

 dense_1 (Dense)             (None, 1)                 65

=================================================================
Total params: 18,241
Trainable params: 18,241
Non-trainable params: 0
_________________________________________________________________
```







Subsystem 4: Power Supply

Components

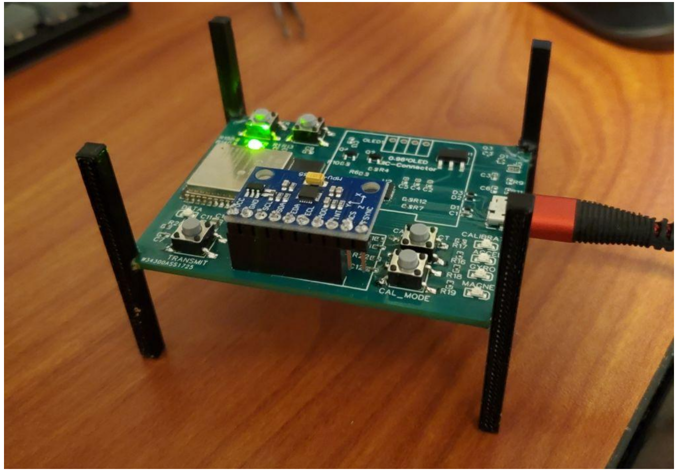
- Power is supplied from Micro-USB adapter at 5V
- Voltage Regulator reduces 5V to 3.3V
- 3.3V is supplied to other parts through copper lines on PCB

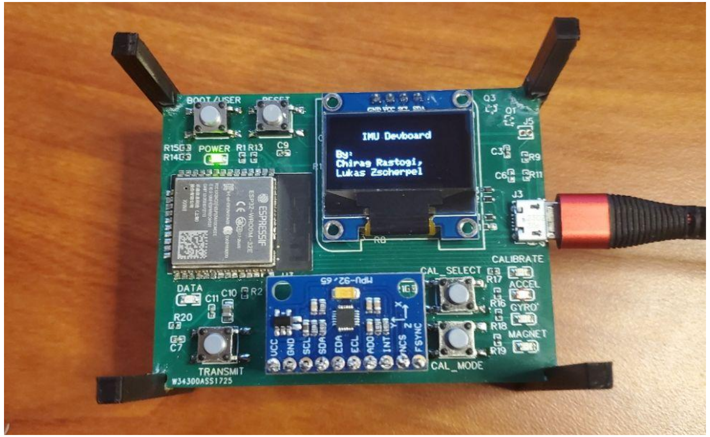
Results

PCB Dev Board

Development

- **Design**
  - Uses surface mount parts for compact design
  - Uses 2 transistor method to automatically enable boot mode
  - 3D printed legs allow for more stable calibration
  - Plug and Play IMU connection
- **Soldering**
  - Failed when trying to hand solder each part
  - Successfully soldered using solder paste and oven
- **Programming**
  - Programmed using the Arduino IDE
  - Uses standard Arduino libraries





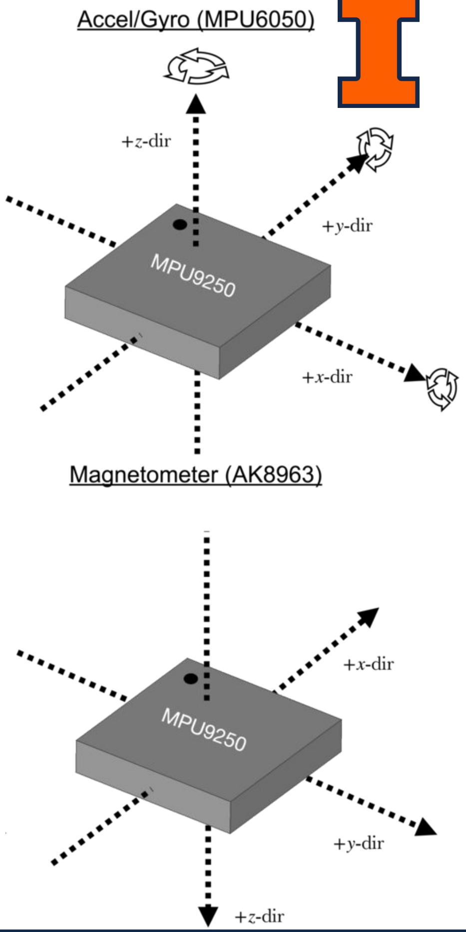
Programming and Calibration

Data Transmission State

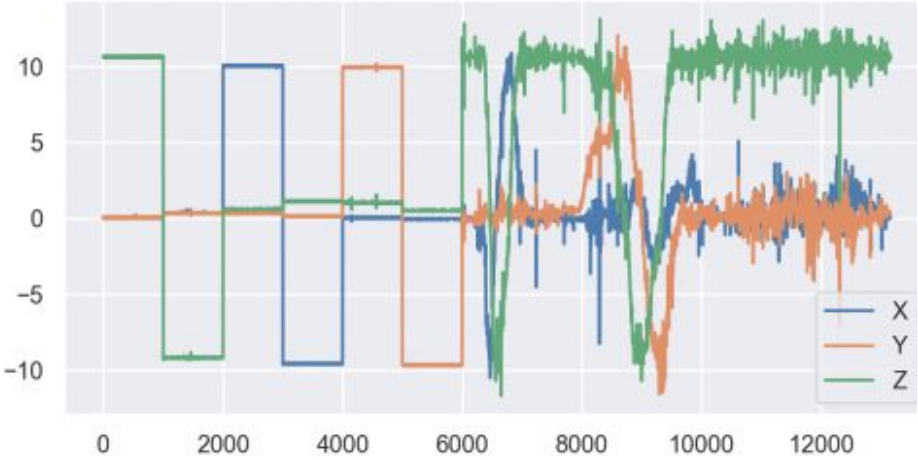
- **Sends data at 250 Hz through serial port**
- **Data includes:**
  - Microcontroller State
  - Accelerometer x, y, and z
  - Gyroscope x, y, and z
  - Magnetometer x, y, and z
  - Timestamp
- **Receiving computer displays attitude estimation**

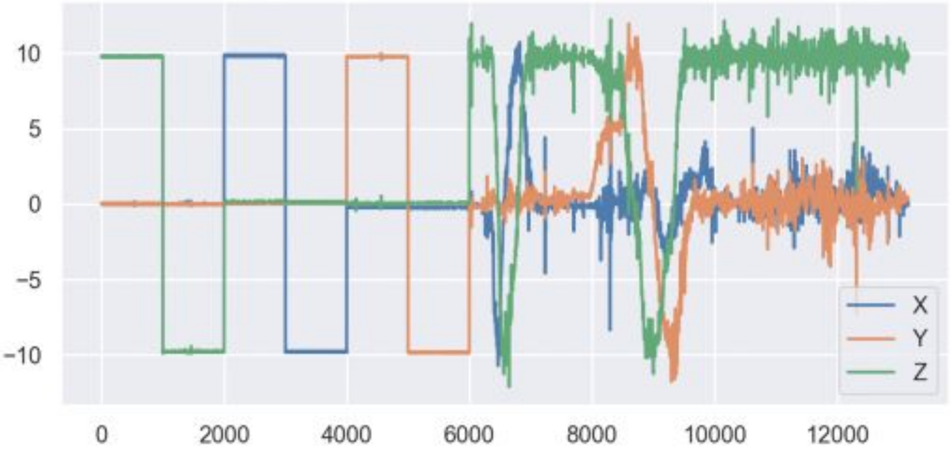
Calibration State

- **Accelerometer**
  - Expect sensors to read zero when device is stationary
  - Calibration is done on each axis (X, Y, Z)
- **Gyroscope**
  - Ferraris calibration method
  - Involves rotating IMU in 360°
- **Magnetometer**
  - Hard Iron Offset
  - Rotate the IMU in a sphere to get magnetometer readings in each orientation



Calibration Test Results





Filtering and AI

Standard Algorithms

Extended Kalman Filter 40Hz,

Madgwick Filter 125Hz, Mahony Filter 160Hz

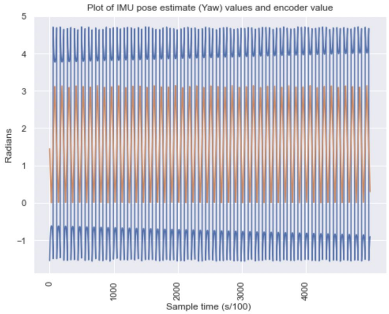
9250 at 400HZ (I2C at 400kHz or SPI at 1MHz)

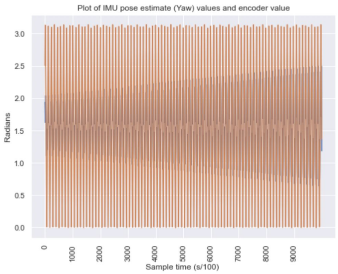
LSTM

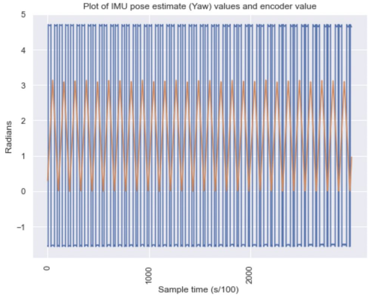
Got an RMSE of < 0.1 degrees over 20 mins

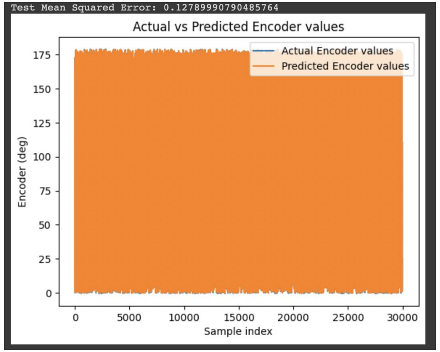
110Hz

13Gb RAM

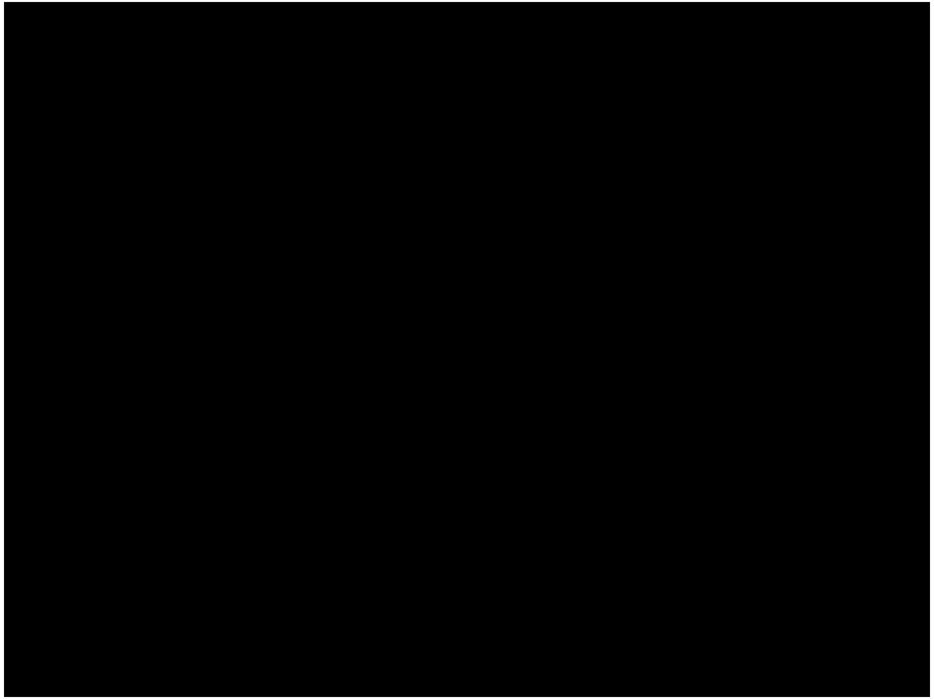








Demo

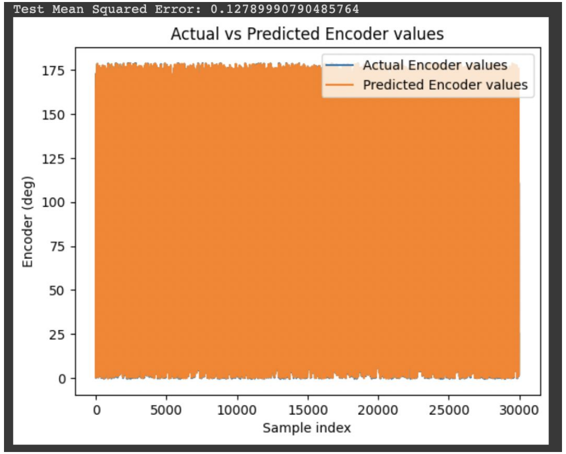


Successes and Challenges

Successes

We were able to meet all our high-level Requirements

- For the Neural Network, the IMU interfacing system was able to accurately measure and monitor the orientation and movement of a device to within a maximum error of 3 degrees and 0.5 cm/s
- The system was able to perform calibration on the data outputted by the IMU to within a maximum error of 0.5 degrees and 0.2 cm/s.
- Due to the performance of the standard algorithms, we chose to stick with the Neural Network as the primary algorithm



Challenges

NVidia Jetson Implementation

- **Booting Issues**
  - We had issues getting started with the Jetson Board

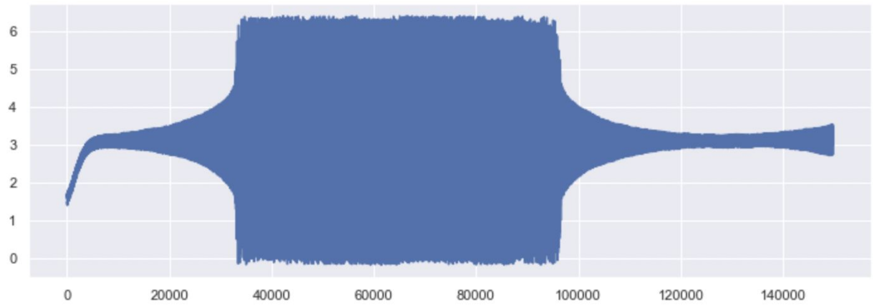
Assembling the PCB

- **Delays**
  - Shipping delays made parts come in last minute
  - Unable to iterate PCB design due to delays
  - Limited time for testing algorithms on data coming from PCB
- **Soldering**
  - Surface mounted IC's are HARD to solder
  - Took multiple attempts to get working
- **Programming**
  - Had issues programming using micropython in Thonny IDE
  - Had to redesign program from initial IMU testing breadboard

Failed Requirements

- We could not get any standard algorithm to give us a MSE of <5% over 20 minutes, therefore we could not perform selection

Error in Radians



Conclusions

What We Learned

Soft Skills

- Scope, schedule, and resources are all important when designing a project
- Good communication between team members is vital to succeed
- Starting early helps to prepare better for the unexpected
- Most important: things never go as planned

Hard Skills

- How to solder surface mount IC's
- How do design a PCB from scratch
- Learned how the three main algorithms worked and their applications
- Learned how to communicate between computer and pcb dev board

What We Would Change

Moving Forward

Thank You

Questions?