Introduction

In this lab we were tasked with developing a GPS driver in ROS2. We began by writing the actual driver itself in Python. We created a Node called GPSDriver which takes in the GPS serial port and baud rate. We then created the callback function to trigger every .1 second which is 10 Hz. Our GPSmsg data message consists of the ROS standard message header which is a timestamp of seconds and nanoseconds. The rest of the message was the latitude, longitude, altitude, UTM easting and northing directions, zone and letter. This entire message was published to the /gps at a rate of 10Hz as that's the ideal frequency for the GPS. Once the driver itself was completed, we developed a launch script to make running the driver easier. As real robot systems can incorporate many different sensors, the launch script would make it easier to run all of them.

We then collected 10 minutes of stationary data in one spot, and then also data of the GPS moving for around 100 meters. We used the rosbag library to record the data in a format that is easily to parse we can analyze the data effectively.

Analysis and Results

We began our stationary data analysis by calculating the mean and standard deviation of the UTM easting and UTM northing to be 327919.68 m, 1.77 m and 4689477.18 m, 3.83 respectively. This data shows that even when the GPS is not moving at all, there is still variation in the recorded position. The variation is relatively small as shown by our mean and standard deviation calculations below. We also plotted the data points on a scatter plot to show the recorded position graphically. This is shown in Fig. 1.

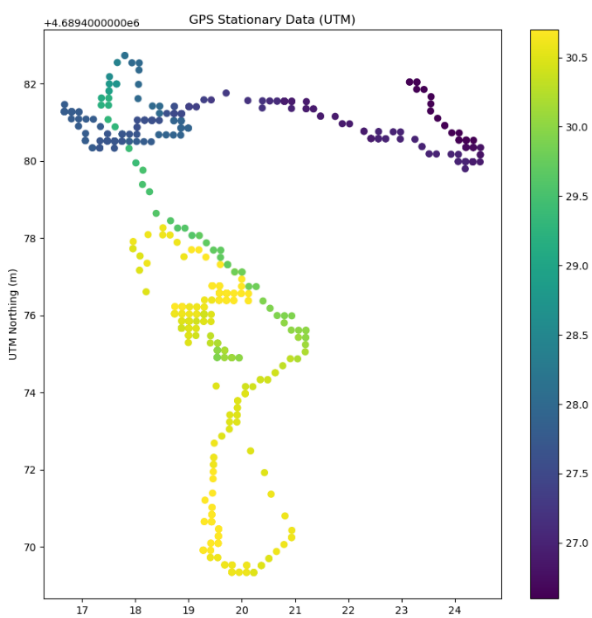

Figure 1: UTM Stationary Data Scatter Plot

As we can see from Fig. 1, there is decent amount of variation in the data in both the 2D directions and in altitude, although most GPS are more accurate in the 2D directions then the altitude. Although, the data does seem to follow some type as pattern as the it seems to have smoother transitions between sequential points instead of it being completely random. Now that we know there is error in the recorded position, we tried to analyze the error to learn more about it and where it comes from. We calculated the error of the 2D data by finding the distance between the actual point and the median. The mean of the 2D stationary data was 3.63 m and had a standard deviation of 2.15 m. As shown in Fig. 2, the error does not fit a Gaussian distribution which must mean the data is coming from another source. We believe that the error is caused by Kalman filtering. The GPS performs Kalman filtering which is useful for moving data as it assumes that velocity and distance don't change too abruptly, which helps to filter out noise. For stationary data, this causes a problem because the filter interprets noise as movement which then compounds error over time. This error can also explain why Fig. 1 seems to follow a smoother motion as the filter makes the changes gradual and not random. We also believe that the peaks and valleys shown in the error graph is from the filter as it will settle on spots and then begin to compound error again.

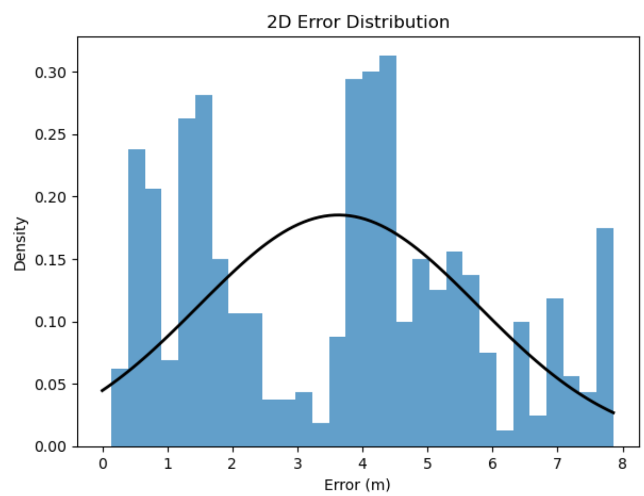

Figure 2. 2D Stationary Data Error

Now that we have analyzed the stationary data, we can use that analysis to infer some things about the analysis of our walking data, specifically relating how the Kalman filter affects moving data differently. As shown in Fig. 3, the GPS data recorded a relatively straight line. The total walking distance was calculated to be 81.89 m which is accurate according to a measurement taken with Google Maps. We then calculated the error of the data by fitting a straight line to the data and measuring the distance from the actual data to the fitted line. The mean of the error was calculated to be .44 m with a standard deviation of .39 m. We plotted a histogram of the error as shown in Fig. 4. The histogram shows that most of the error is under 1 m with a couple samples taken from 1.5 m to 2.25 m. As we assumed before, the Kalman filter works a lot better for stationary data then moving data as it is designed to filter out small fluctuations that are not normal to moving data.

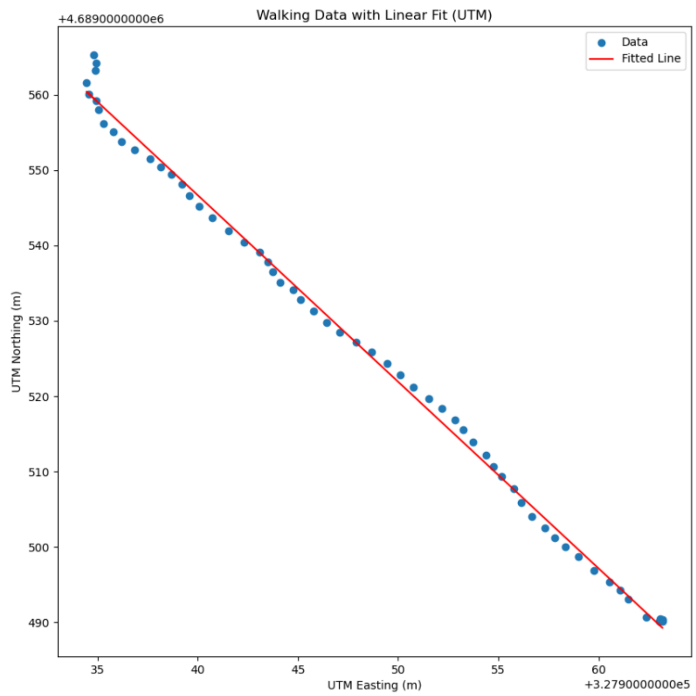

Figure 3. UTM GPS Walking Data

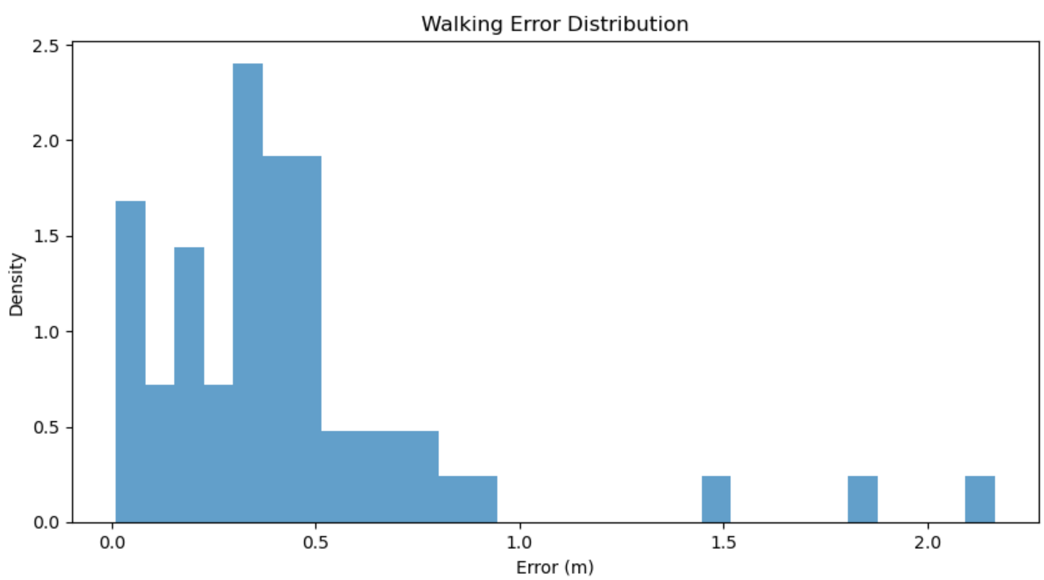

Figure 4. UTM Walking Error Distribution