Department of Systems and Computer Engineering

SYSC5702 W: Sensor Fusion
for Autonomous Vehicles

Project #2: Implementing IMU-based Forward and Inverse Kinematics Equations

Date: February 24, 2025
Name: Alexander Crain
Student ID: 100848550

Problem Statement

This report details the implementation of inverse and dynamic equations for an Inertial Measurement Unit (IMU) on two datasets. The first dataset (#1) was obtained from a sensor model, while the second dataset (#2) was obtained from a real sensor. In this report, the inverse kinemtics equations are implemented in software to calculate the accelerometer and gyroscope data for both datasets using the position, velocity, and orientation data. The dynamic equations are then implemented and used to re-calculate the position, velocity, and orientation data using the "measurements" obtained using the inverse kinematics. The resulting state estimates are compared to their dataset counterparts, and conclusions are drawn regarding any observed errors. Finally, noise and bias factors are added to the "measurements" obtained from the inverse kinematics prior to passing this data through the dynamics equations. The overall effect of these noise and bias factors on the state estimates is then analysed.

Developed Solution

The following section contains the solutions to the problems outlines in the Problem Statement. All code was written in Python 3.10.2 and has been provided separately from the report.

Task #1: Inverse Kinematics on Dataset #1

In this subsection of the report, the results of the inverse kinematics calculations on dataset #1 are summarized. The inverse kinematics equations are used to obtain the approximate IMU acceleration and gyroscope data from position, velocity, and orientation data.

Part #1: Summary of Salient Theory - Inverse Kinematics

The inverse kinematics algorithm can be summarized as follows:

1. From the provided velocity v^L (in L frame), and the time between each data sample Δt, the acceleration a^L in the L frame can be calculated from:

$$a^L = \frac{\mathrm{d}v^L}{\mathrm{d}t} = \frac{v^L(k) - v^L(k-1)}{\Delta t} \tag{1}$$

2. The acceleration in the L frame must be corrected for the effects of gravity and the Coriolis effect. The corrected acceleration $\hat{a}^L$ is calculated as follows:

$$\hat{a}^L = a^L - g^L + (\omega^L_{EL} + 2\omega^L_{IE}) \times v^L \tag{2}$$

where g^L is the acceleration due to gravity in the L frame, ω^L_{EL} is the angular velocity of the Earth in the L frame, and ω^L_{IE} is the angular velocity of the Earth relative to the inertial frame in the L frame. These are calculated using the following equations:

$$g^L = \begin{bmatrix} 0 & 0 & -9.81 \end{bmatrix} \tag{3}$$

$$\omega^L_{EL} = \begin{bmatrix} \frac{v_e}{R_n+h} & \frac{-v_n}{R_m+h} & \frac{-v_e}{R_n+h}\tan(\phi) \end{bmatrix} \tag{4}$$

$$\omega^L_{IE} = \begin{bmatrix} w^e\cos(\phi) & 0 & -w^e\sin(\phi) \end{bmatrix} \tag{5}$$

where v_e and v_n are the east and north velocities respectively, R_n and R_m are the normal and meridian radii of curvature of Earth at the current latitude ϕ, h is the altitude, and w^e is the Earth's angular velocity–which is assumed to be 7.292115×10^{-5} rad/s.

3. The IMU acceleration is not in the L frame, but rather in the body frame B. To convert the corrected acceleration to the B frame, the direction cosine matrix (DCM) C^B_L is used:

$$\hat{a}^B = C^B_L \hat{a}^L \tag{6}$$

4. The gyroscope data can be calculated by calculating the rotation difference between two consecutive DCMs and then divide by sampling period, which results in the skew matrix s_g:

$$s_g = \frac{C^B_L(k-1)C^L_B(k) - I}{T} \tag{7}$$

where I is the identity matrix, k is the sample, and T is the sampling period. From the skew matrix s_g, the gyroscope data can be extracted:

$$\omega^B_{LB_{SIM}} = \begin{bmatrix} s_g(3,2), & s_g(1,3), & s_g(2,1) \end{bmatrix} \tag{8}$$

5. Finally, the gyroscope data must be corrected for the Earth's rate and the Transport rate using:

$$\omega_{LB}^{B} = \omega_{LB_{SIM}}^{B} + C_{L}^{B}(\omega_{EL}^{L} + \omega_{IE}^{L}) \tag{9}$$

Part #2: Calculated Acceleration and Gyroscopic Data

Figure 1 compares the calculated acceleration obtained from the inverse kinematics to the data provided on the outset of the project. The match between the calculated and measured acceleration is near-perfect, indicating that the inverse kinematics have been successfully implemented. However, note the overshoot present in the measured data which is not present in the calculated data. Note, the "data" was in fact obtained from models of an IMU and are not true measurements. As such, the overshoot is most likely the result of sensor dynamics in the model. These would not be captured by the inverse kinematics, as these equations represent an idealized system. Possibly, this indicates the presence of a filter in the model, which would explain the underdamped response.

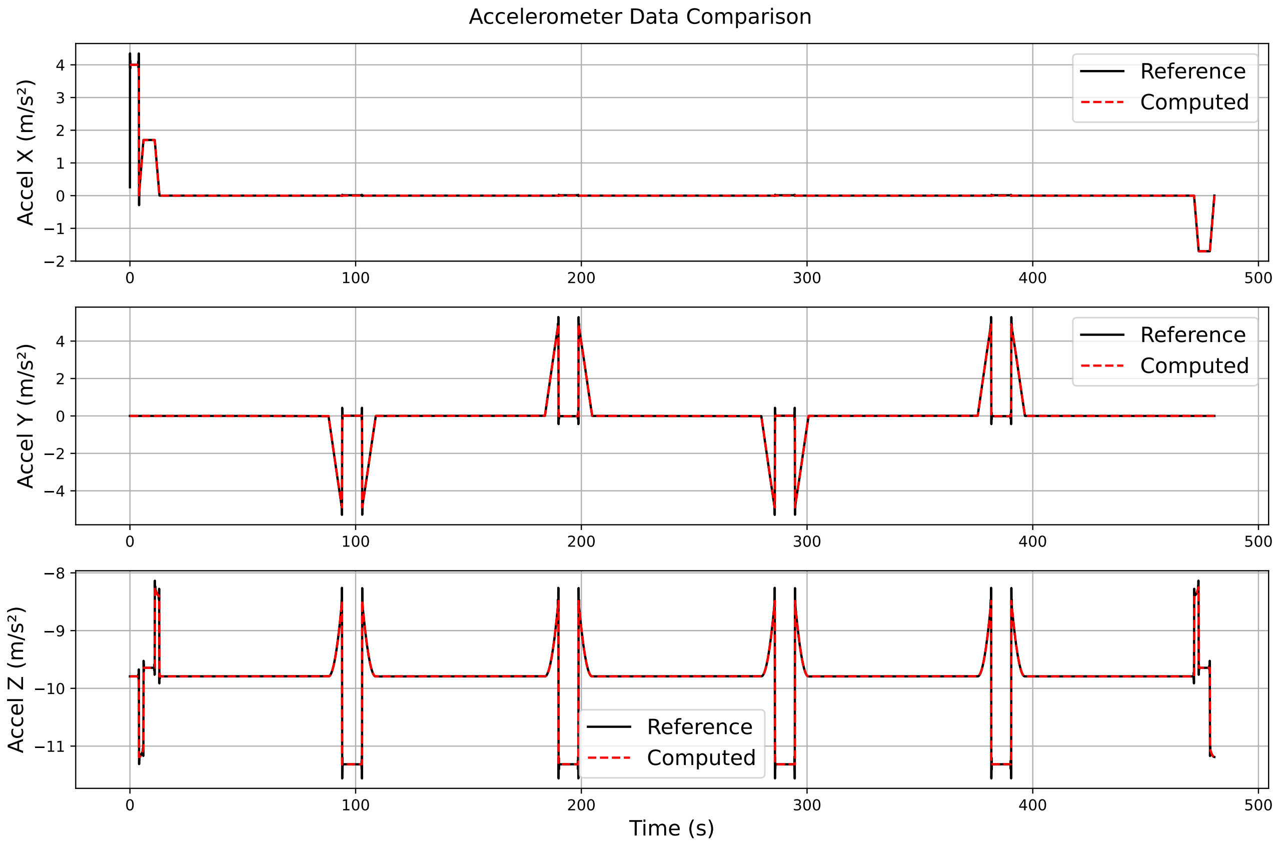

Figure 1: Comparison between the acceleration data provided in dataset #1 and the acceleration data calculated using the inverse kinematics.

Figure 2 compares the calculated acceleration obtained from the inverse kinematics to the data provided on the outset of the project. Similar to the previous conclusions around the acceleration data, the match between the calculated and measured gyroscope data is excellent, except for a similar overshoot. The conclusions drawn above regarding differences between the model and the inverse kinematics would also apply here.

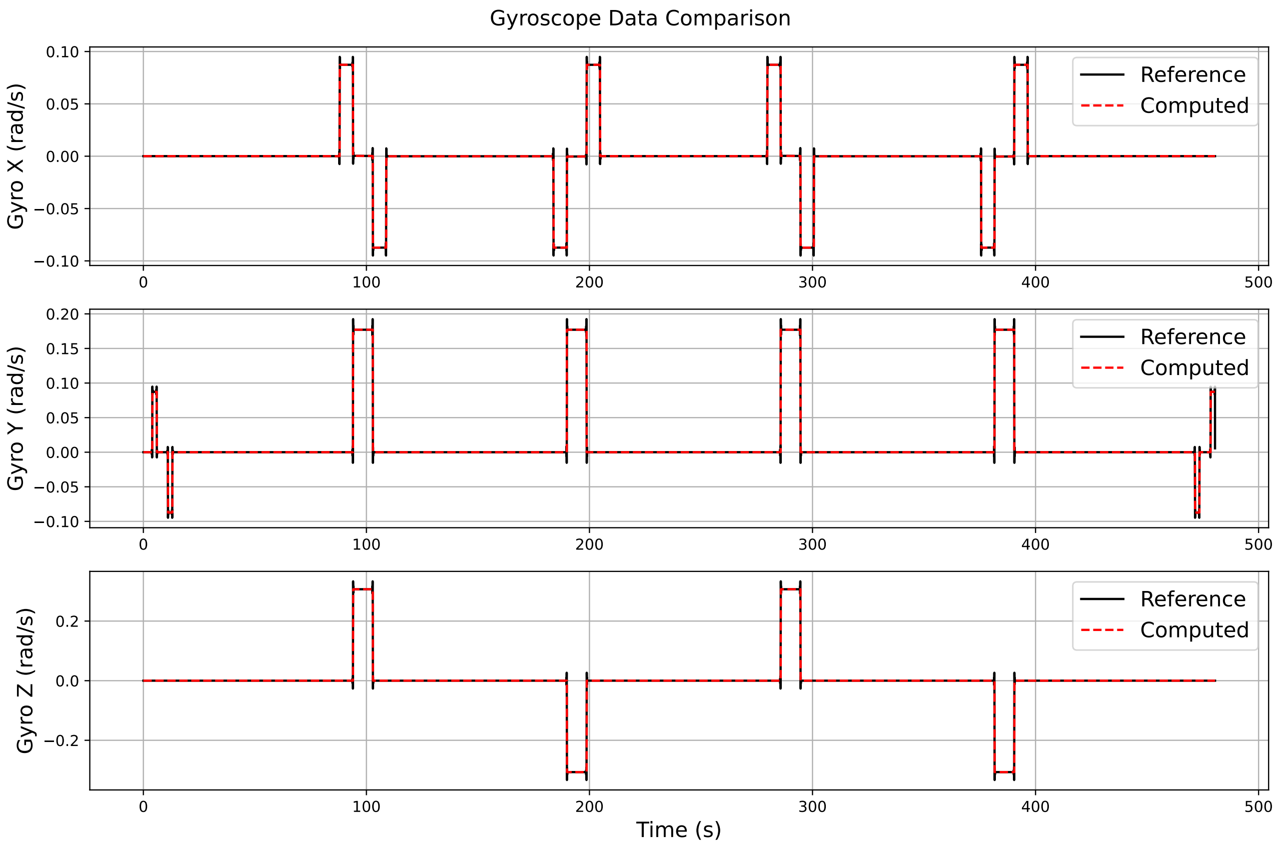

Figure 2: Comparison between the gyroscope data provided in dataset #1 and the gyroscope data calculated using the inverse kinematics.

Task #2: INS Dynamics Model on Dataset #1

In this subsection, IMU-based motion equations will be used to generate the position, velocity, and orientation data using the previously obtained acceleration and gyroscope data. The results will be compared to the provided data in dataset #1.

Part #1: Summary of Salient Theory - Dynamics

The INS dynamic algorithm can be summarized as follows:

1. The position, velocity, and orientations are initialized to the first data point in the dataset. The DCM C_B^L is initialized using the initial orientation data.
2. The radii of curvature R_n and R_m are calculated using the estimated latitude ϕ and altitude h. The equations are not provided here, but they are available in the code.
3. The quaternion q_L^B is calculated from the current DCM C_B^L. The equations will not be provided here, but they are available in the code.
4. The gyroscope compensation equations from the previous section are used to calculate ω_{IL}^L. Using the gyroscope measurements and the correction, the compensated gyroscope data is calculated:

$$\omega_{LB}^B = \frac{\hat{\omega}_{IB}^B - b_g}{1 + s_g} - C_L^B \omega_{IL}^L \tag{10}$$

where b_g is the gyroscope bias, s_g is the scale factor, and $\hat{\omega}_{IB}^B$ is the gyroscope measurement.

5. The time derivative of the quaternion is calculated using the compensated gyroscope data using:

$$\dot{q}_B^L = \frac{1}{2}(W_{LB}^B \times) q_B^L \tag{11}$$

where the matrix $W_{LB}^B \times$ is defined as:

$$W_{LB}^B \times = \begin{bmatrix} 0 & -\omega_{LB_X}^B & -\omega_{LB_Y}^B & -\omega_{LB_Z}^B \\ \omega_{LB_X}^B & 0 & \omega_{LB_Z}^B & -\omega_{LB_Y}^B \\ \omega_{LB_Y}^B & -\omega_{LB_Z}^B & 0 & \omega_{LB_X}^B \\ \omega_{LB_Z}^B & \omega_{LB_Y}^B & -\omega_{LB_X}^B & 0 \end{bmatrix} \tag{12}$$

6. The derivative of the quaternion at time step t_i is numerically integrated to calculate the quaternion at time $t_i + \Delta t$. This quaternion is then normalized via:

$$||q_B^L|| = \frac{q_B^L}{\sqrt{a^2 + b^2 + c^2 + d^2}} \tag{13}$$

where b, c, and d are the elements of the quaternion, and a is the magnitude of the quaternion.

7. The DCM C_B^L is calculated from the quaternion $||q_B^L||$ using the following equations:

$$C_B^L(1,1) = a^2 + b^2 - c^2 - d^2 \tag{14}$$

$$C_B^L(1,2) = 2(bc - ad) \tag{15}$$

$$C_B^L(1,3) = 2(bd + ac) \tag{16}$$

$$C_B^L(2,1) = 2(bc + ad) \tag{17}$$

$$C_B^L(2,2) = a^2 - b^2 + c^2 - d^2 \tag{18}$$

$$C_B^L(2,3) = 2(cd - ab) \tag{19}$$

$$C_B^L(3,1) = 2(bd - ac) \tag{20}$$

$$C_B^L(3,2) = 2(cd + ab) \tag{21}$$

$$C_B^L(3,3) = a^2 - b^2 - c^2 + d^2 \tag{22}$$

8. The time derivative of the velocity in the L frame is calculated using the acceleration data and the DCM C_B^L:

$$\dot{v}^L = C_B^L a^B + g_L - (\omega_{EL}^L + 2\omega_{IE}^L) \times v^L \tag{23}$$

where a^B is obtained from $\hat{a}^B$ through:

$$a^B = \frac{\hat{a}^B - b_a}{1 + s_a} \tag{24}$$

where b_a is the accelerometer bias, s_a is the accelerometer scale factor, and $\hat{a}^B$ is the accelerometer measurement obtained using the inverse kinematics.

9. The velocity at time step t_i is numerically integrated to calculate the velocity at time $t_i + \Delta t$. Using this estimate of velocity, the position can then be calculated using:

$$\dot{r} = \begin{bmatrix} \frac{1}{R_n + h} & 0 & 0 \\ 0 & \frac{1}{(R_m + h)\cos(\phi)} & 0 \\ 0 & 0 & -1 \end{bmatrix} v^L \tag{25}$$

10. Once again, the position at time step t_i is numerically integrated to calculate the position at time $t_i + \Delta t$. Finally, the orientation data is extracted from the DCM C_B^L using the following equations:

$$\phi = \arctan 2\left(C_B^L(3,2), C_B^L(3,3)\right) \tag{26}$$

$$\theta = -\arcsin(C_B^L(3,1)) \tag{27}$$

$$\psi = \arctan 2\left(C_B^L(2,1), C_B^L(1,1)\right) \tag{28}$$

11. This process is then repeated using the calculated position, velocity, and orientation dat at $t_i+\Delta t$; in other words, the algorithm restarts the process from Step #2.

Part #2: Calculated Position, Velocity, and Orientation Data

In this section, the results of the INS dynamic model on dataset #1 are presented. The position, velocity, and orientation data are compared to the provided data in the dataset. For the results in this section, it was assumed that there is no noise, bias, or scale factor on the gyroscope and accelerometer data. Figure 3 compares the position (latitute, longitude, and altitude) data from dataset #1 to the results obtained using the INS dynamics model. The match between the calculated and measured data is reasonable, but there is some drift in all three position values over time. This drift is the result of several factors:

- Any small unaccounted for errors in the sensor data (bias and scale factors) will cause small errors which accumulated over time. The longer the intefgration period, the more signficant the drift.
- Similarly, any unaccounted noise factors in the data will also cause small errors which accumulate over time.
- Finally, sensor misalignment is also often a cause for drift during the numerical integration.

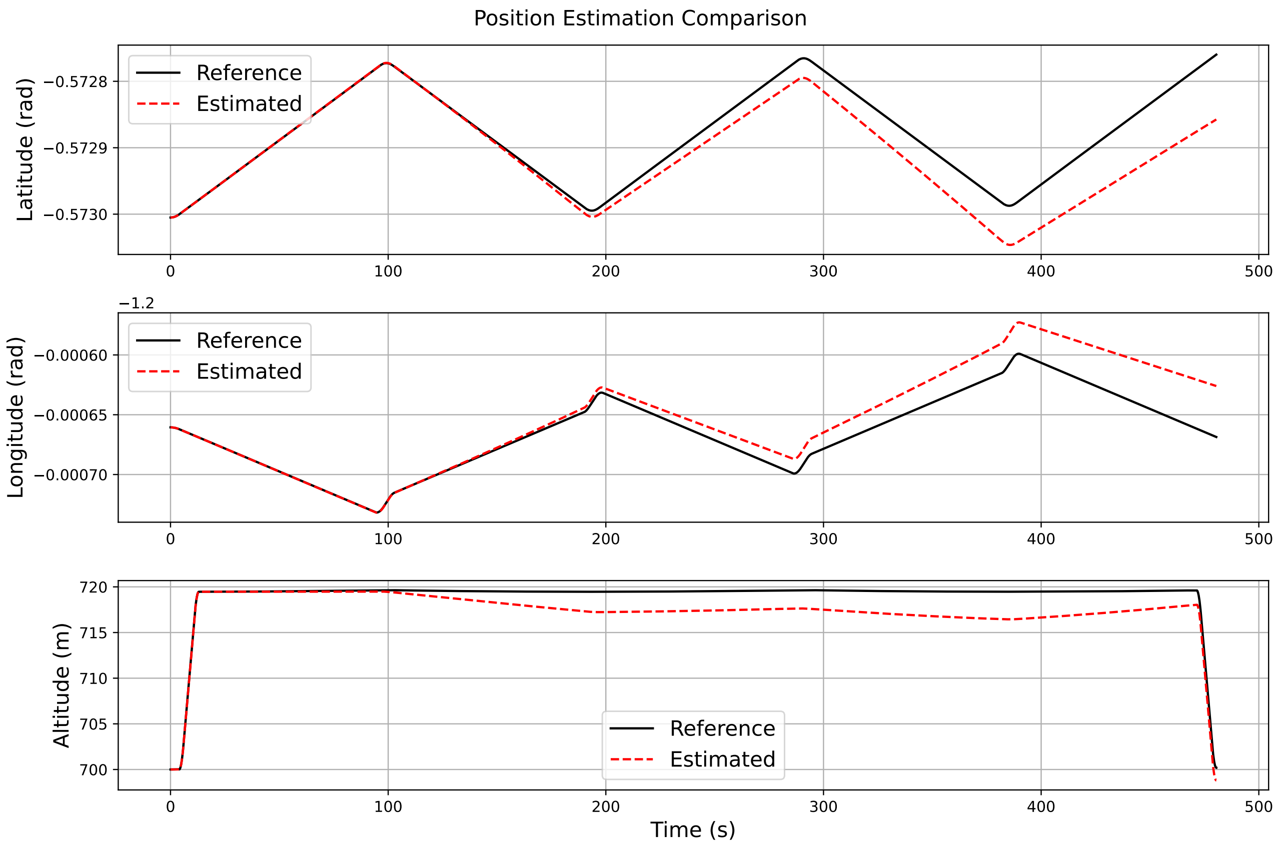

Figure 3: Comparison between the position data provided in dataset #1 and the position calculated using the INS dynamic model.

However, the drift is overall small, with the RMSE for the latitude, longitude and altitude being 4.2×10^{-5} radians, 1.8×10^{-5} radians, and 1.8 meters respectively. These results are summarized in Table 1.

Table 1: Root Mean Squared Error (RMSE) values for position, velocity, and orientation states.

State	**RMSE**
Latitude	4.2×10^{-5} radians
Longitude	1.8×10^{-5} radians
Altitude	1.8 meters
Velocity North	1.6 m/s
Velocity East	0.60 m/s
Velocity Down	0.015 m/s
Roll	3.1×10^{-4} radians
Pitch	9.5×10^{-4} radians
Heading	9.6×10^{-4} radians

Figure 4 compares the velocity data from dataset #1 to the results obtained using the INS dynamics model. Overall, the match is similarly reasonable, and while some drift is still present, the overall magnitude of the drift is smaller. This is expected given that the velocity is the result of two integration steps, wheras the position is the result of three integrations steps. Thus, the small errors that contribute to the drift will accumulate more slowly in the velocity estimations. The overall RMSE for the velocity components is listed in Table 1.

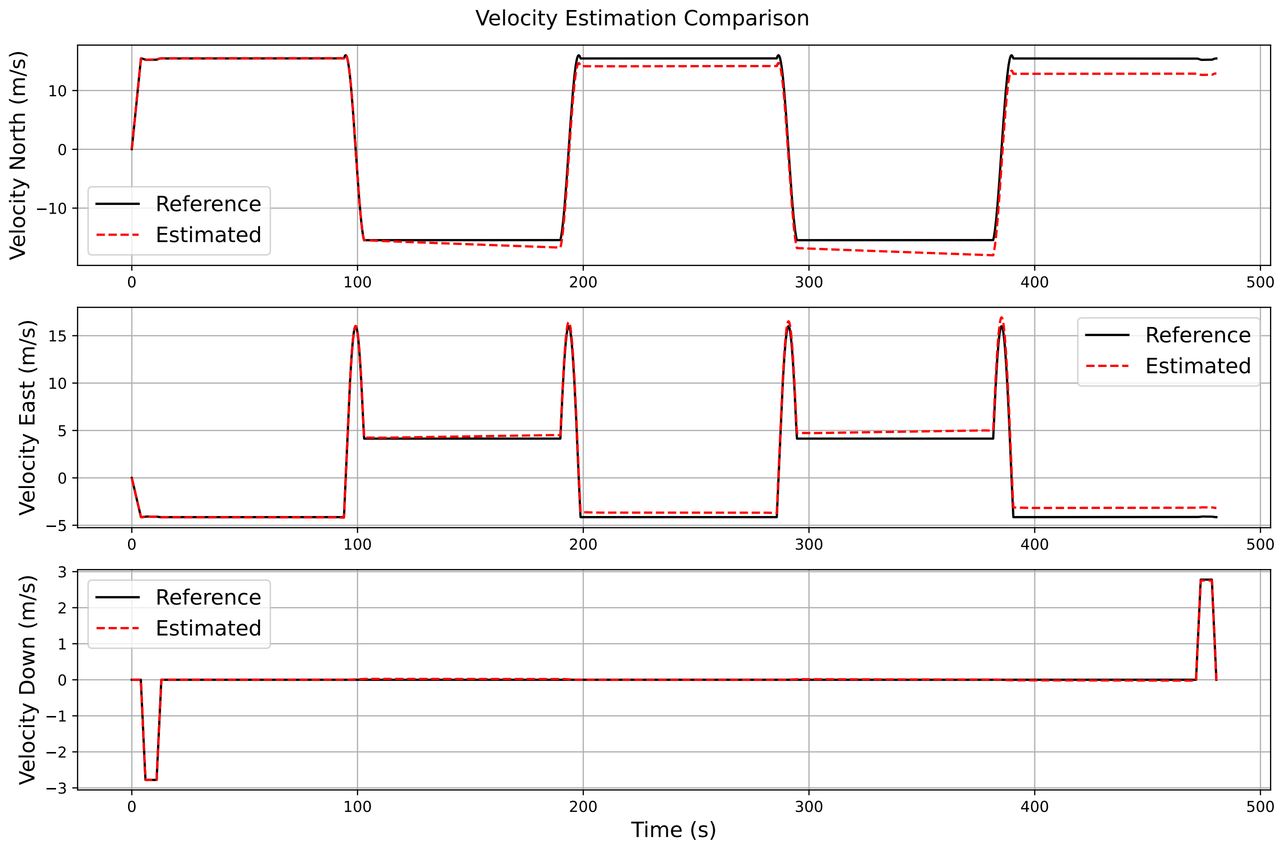

Figure 4: Comparison between the velocity data provided in dataset #1 and the velocity calculated using the INS dynamic model.

Lastly, Fig. 5 compares the orientation data from dataset #1 to the results obtained using the INS dynamics model. The match between the calculated and measure ddata is excellent, with no visually identafiable drift in the estimations. This is consistent with the fact that the orientation data is obtained through a single numerical integral, thereby reducing the accumulation of errors over time. The RMSE for the orientation data is also listed in Table 1, with the average error on the order of 10^{-4} radians.

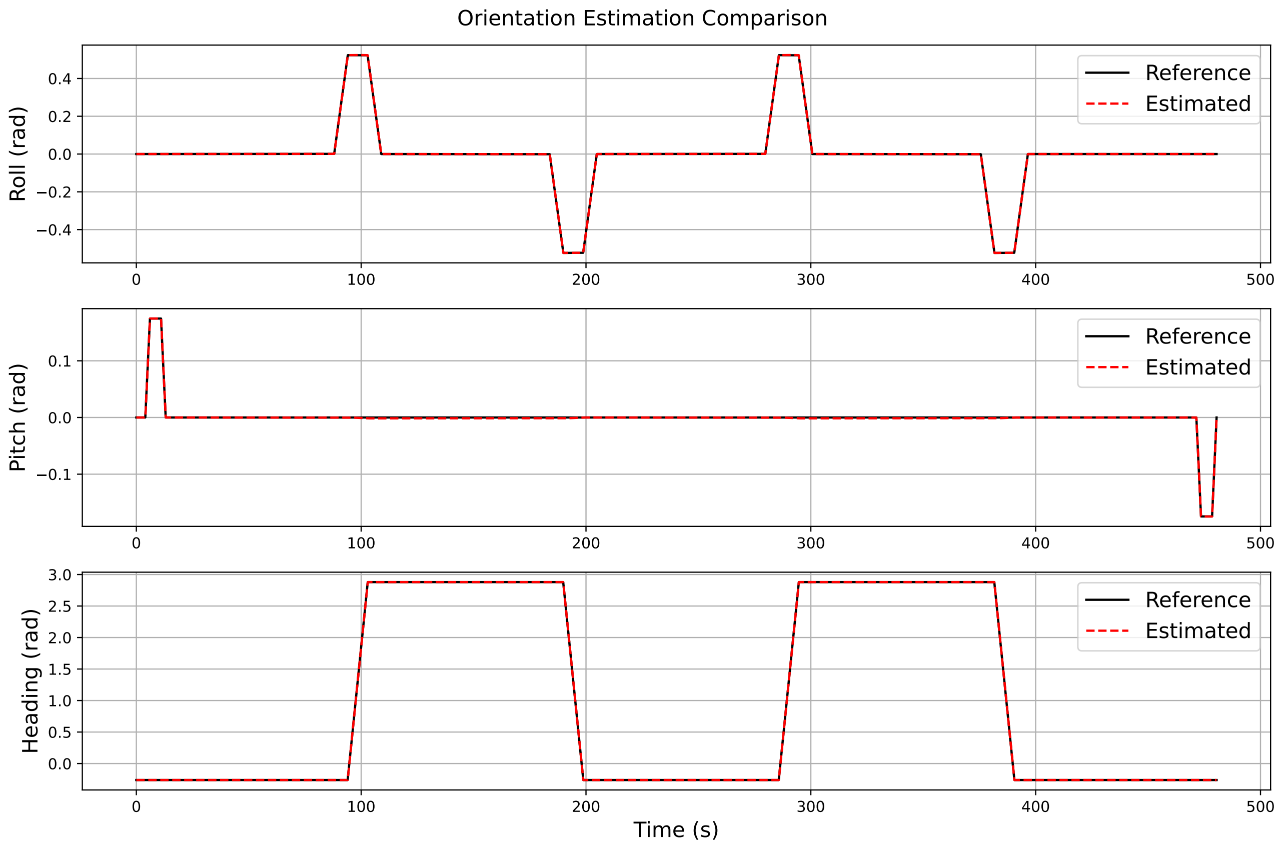

Figure 5: Comparison between the orientation data provided in dataset #1 and the orientation calculated using the INS dynamic model.

Task #3: Effect of Noise on Dynamics - Dataset #1

In this section, noise and bias factors will be added to the gyroscope and accelerometer data obtained from the inverse kinematics. The results from the INS dynamics model will be compared to the provided data in dataset #1 and the impact of the noise and bias factors will be discussed. To properly understand the impact of these noise and bias factors, it is necessary to apply them on a single-axis basis. However, this would generate a significant number of figures, most fo which would not provide critically important informations. Thus, to streamline the analysis, most of the results will be summarized in tables. However, for the sake of demonstrating the impact, Fig. 6 shows the impact of adding some noise and a bias to the z-axis of the accelerometer data. The impact is primarily on the altitude estimation, which diverges signficantly from the measurement. This makes sense, as the noise and bias will increase the errors that accumulate over each integration step, leading to an overall increase in drift for the chosen axis.

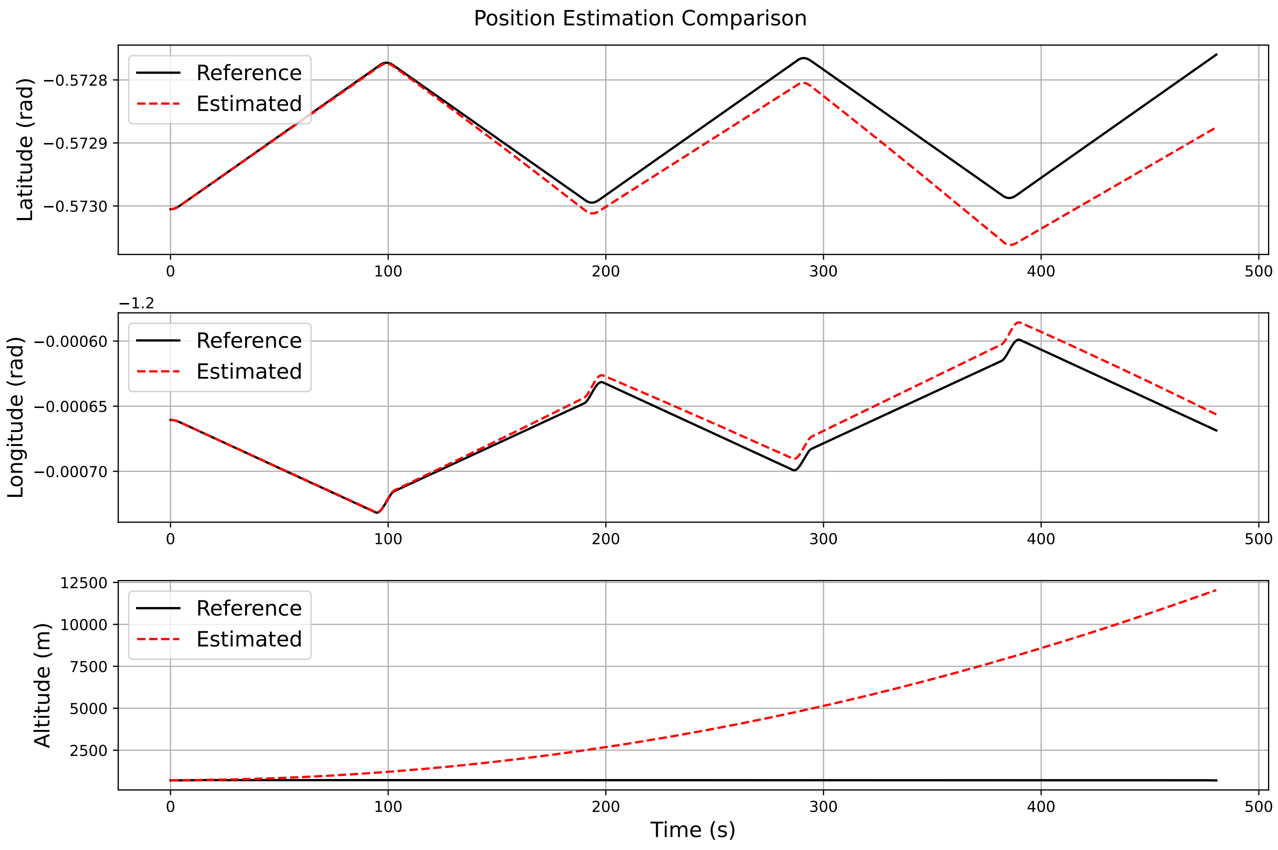

Figure 6: Comparison between the position data provided in dataset #1 and the position calculated using the INS dynamic model when noise and a bias factor are added to A_z.

Table 2 summarizes the RMSE values for the position states when noise and bias factors are applied to the accelerometer and gyroscope data from the inverse kinematics analysis. Under default conditions (labelled as `Default`), the overall RMSE is quite low, with an altitude error of only 1.84 m. However, when adding noise and bias to the three acceleromter axes, the drift in the estimated states increases–this is particularly true for the altitude, whose RMSE increases to over 5,000 m when noise and a bias are added to the z-axis. In the case of the gyroscope data, the addition of noise and a bias factor have to the x- and y-axis has a much greater impact then when it is added to the z-axis data.

Table 2: Root Mean Squared Error values for position states, with and without noise and bias factors (dataset #1).

Case	**Latitude (radians)**	**Longitude (radians)**	**Altitude (m)**
DS1, Default	4.20×10^{-5}	1.81×10^{-5}	1.84
DS1, A_x	2.50×10^{-4}	4.70×10^{-5}	2.65×10^{1}
DS1, A_y	6.86×10^{-5}	2.31×10^{-4}	5.77×10^{1}
DS1, A_z	5.12×10^{-5}	8.52×10^{-6}	5.07×10^{3}
DS1, G_x	5.49×10^{-3}	4.22×10^{-2}	7.50×10^{4}
DS1, G_y	3.66×10^{-2}	5.45×10^{-3}	6.67×10^{4}
DS1, G_z	6.59×10^{-4}	1.55×10^{-3}	5.10×10^{2}

Table 3 summarizes the RMSE values for velocity states. The default errors are again quite small, with the largest error being the north velocity at 1.62 m/s. Adding noise and bias factors to the accelerometer directly increases the corresponding velocity; in other words, adding noise and bias to A_x primarily increases the north velocity error, adding noise and bias to A_y primarily increases the east velocity error, and adding noise and bias to A_z primarily increases the down velocity error. This direct correlation is not present for the gyroscope data, where the errors are more evenly distributed accross the velocities. This is likely due to the fact that the gyroscope data is used to estimate the quaternions, which are then used to estimate the DCM. The DCM is used to calculate the acceleration in the L frame, and the time derivative of the velocity is then numerically integrated. Because of this complex coupling, any additional bias or noise in the gyroscope data is likely to impact all three velocity components.

Table 3: Root Mean Squared Error values for velocity states, with and without noise and bias factors (dataset #1).

Case	**North Velocity (m/s)**	**East Velocity (m/s)**	**Down Velocity (m/s)**
DS1, Default	1.62	0.60	0.015
DS1, A_x	6.69	1.08	8.97×10^{-2}
DS1, A_y	1.82	5.11	3.70×10^{-1}
DS1, A_z	1.81	2.15×10^{-1}	2.73×10^{1}
DS1, G_x	1.39×10^{2}	1.21×10^{3}	3.96×10^{2}
DS1, G_y	1.26×10^{3}	1.16×10^{2}	3.49×10^{2}
DS1, G_z	3.54×10^{1}	5.85×10^{1}	3.12

Finally, Table 4 summarizes the RMSE values for orientation states. Once again, under the default conditions the RMSE is negligibly small accross the three orientations. In fact, contrary to the previous states, adding noise and a bias factor to the accelerometer data has no significant effect on the estimated orientation states. This is consistent the equations for orientation having no direct dependence on the accelerometer data; the orientations are estimated from the DCM matrix, which itself comes from the quaternions estimates. These quaternions only utilize the gyroscope measurements. However, as expected, adding noise and a bias factor to the gyroscope data has a significant impact on the orientation estimates. Given the spread of the gyroscope measurements across the DCM, it is unsuprising that the orientation errors are more evenly spread across all three orientations. This highlights the sensitivity of the orientation calculations to errors in the gyroscope measurements.

Table 4: Root Mean Squared Error values for orientation states, with and without noise and bias factors (dataset #1).

Case	**Roll (radian)**	**Pitch (radian)**	**Azimuth (radian)**
DS1, Default	3.09×10^{-4}	9.49×10^{-4}	9.61×10^{-4}
DS1, A_x	2.92×10^{-4}	8.54×10^{-4}	9.61×10^{-4}
DS1, A_y	3.28×10^{-4}	9.66×10^{-4}	9.73×10^{-4}
DS1, A_z	3.08×10^{-4}	9.45×10^{-4}	9.61×10^{-4}
DS1, G_x	5.39×10^{-1}	1.57×10^{-1}	1.02×10^{-1}
DS1, G_y	2.36×10^{-1}	5.31×10^{-1}	7.58×10^{-2}
DS1, G_z	2.76×10^{-2}	2.61×10^{-2}	2.17

The auxiliary figures which correspond to the data in the tables above are available in the appendices of this report.

Task #4: Repeat Analysis on Dataset #2

In this section, the previous (3) tasks will be repeated on dataset #2. This includes the inverse kinematics, the dynamics, and an assessment of the impact of adding noise and bias to the data.

Inverse Kinematics on Dataset #2

Figure 7 compares the calculated acceleration obtained from the inverse kinematics to the data provided on the outset of the project. Visually, the measurements obtained through the inverse kinematics closely follows the dataset. However, it is also clear that the computed data is smoother than the measured data. Once again, this is likely due to sensor dynamics not being included in the inverse kinematics. This is made more evident by comparing the RMSE values from dataset #1 to those from dataset #2. The RMSE values for the acceleration in dataset #1 was 0.0304 m/s^2. Meanwhile, the RMSE value for dataset #2 was 0.1928 m/s^2 . So, the overall match between the measurement and data for dataset # is worse than for dataset #1. This is consistent with the fact that the data in dataset #2 was obtained from a real sensor, whereas dataset #1 comes from a sensor model. The real sensor is likely to be even less idealized than the model, which itself is idealised when compared to the inverse kinematics.

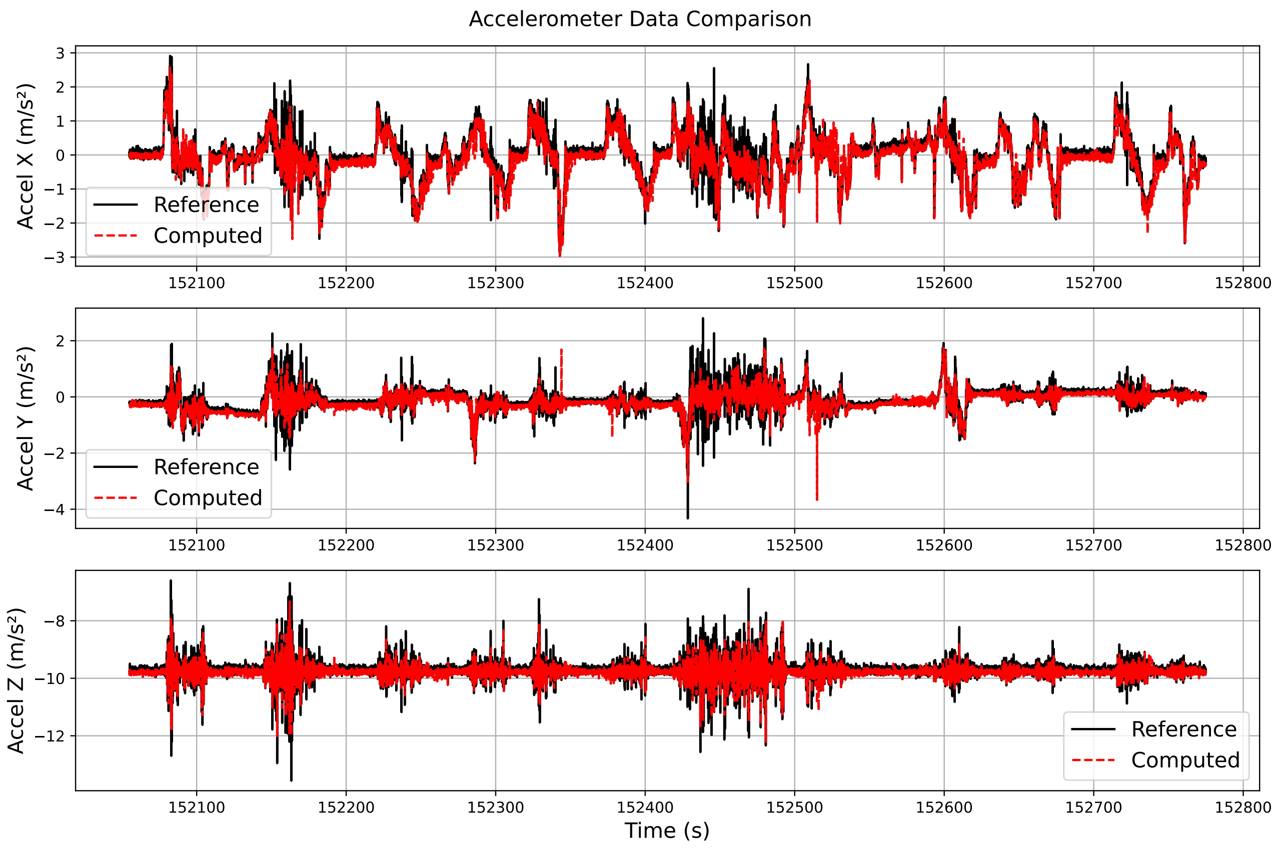

Figure 7: Comparison between the acceleration data provided in dataset #2 and the acceleration data calculated using the inverse kinematics.

Figure 8 compares the calculated acceleration obtained from the inverse kinematics to the data provided on the outset of the project. Similar to the acceleration, the overall visual match for the gyroscope data is good, but the RMSE is measurably higher for dataset #2, at 0.00742 rad/s compared to 0.00247 rad/s for dataset #1. This continues to support the conclusions drawn above.

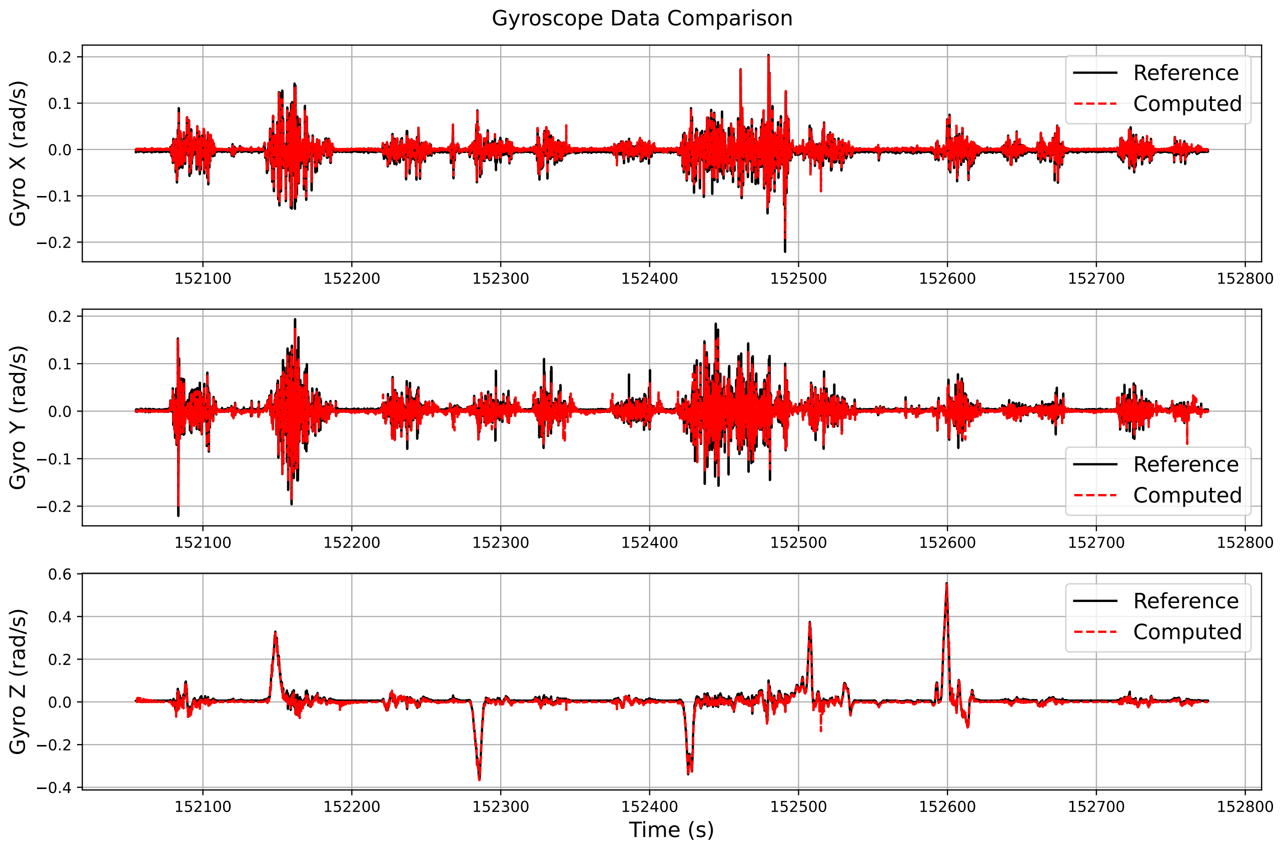

Figure 8: Comparison between the gyroscope data provided in dataset #2 and the gyroscope data calculated using the inverse kinematics.

Calculated Position, Velocity, and Orientation Data - Dataset #2

Having obtained the measurements for the gyroscope and accelerometer using inverse kinematics, the previously described INS dynamics model can be used to calculate the position, velocity, and orientation data for dataset #2. Figure 9 compares the position (latitute, longitude, and altitude) data from dataset #2 to the results obtained using the INS dynamics model. Similar to what was obtained for dataset #1, the overall match is quite reasonable, with a small amount of drift present in all states. The previous items identified as possible causes for this behavior stil apply, and won't be repeated.

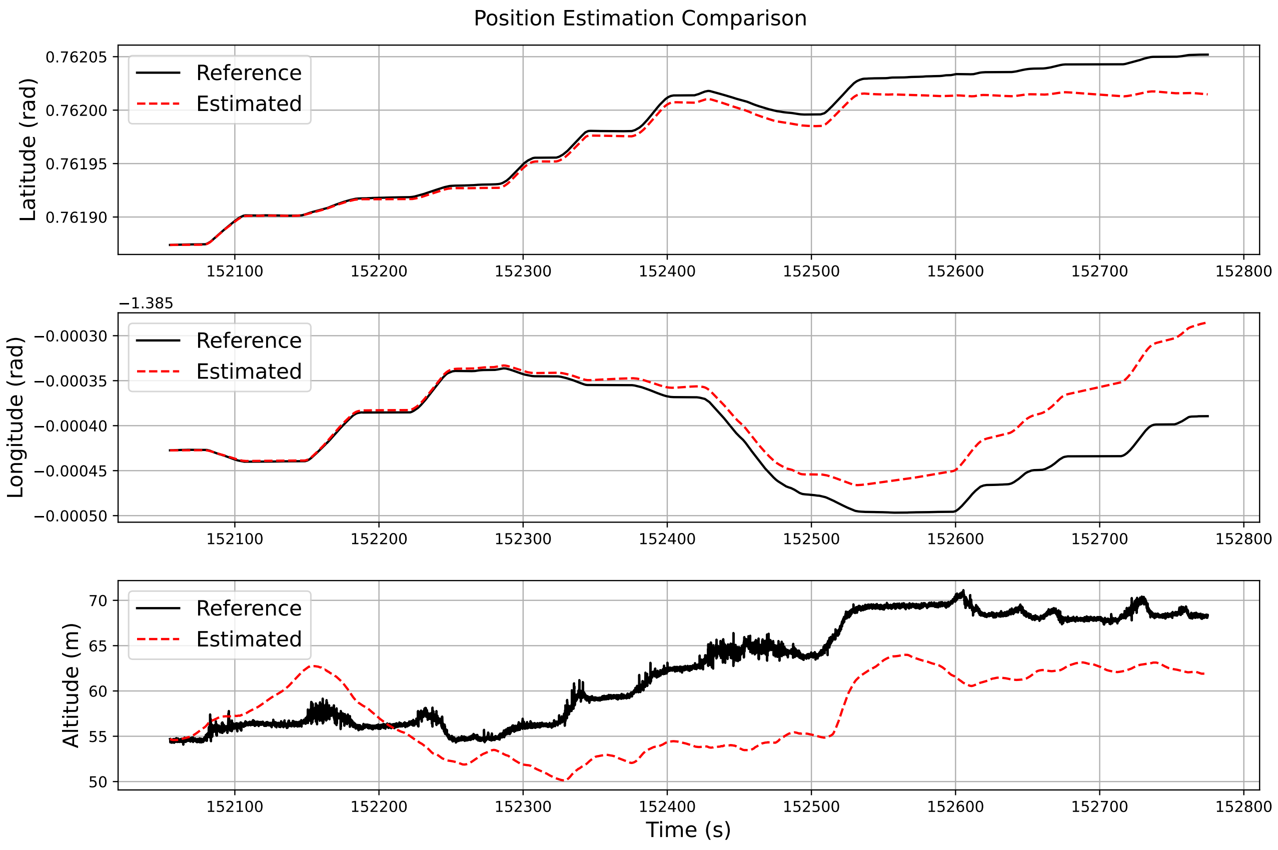

Figure 9: Comparison between the position data provided in dataset #2 and the position calculated using the INS dynamic model.

Figure 10 compares the velocity data from dataset #2 to the results obtained using the INS dynamics model. Once again, similar to dataset #1, the match is good. Despite some drift is still being present, the overall magnitude of the drift is smaller than it is for the position states. The likely cause is, once again, the two numerical integration steps required to calculate the velocity.

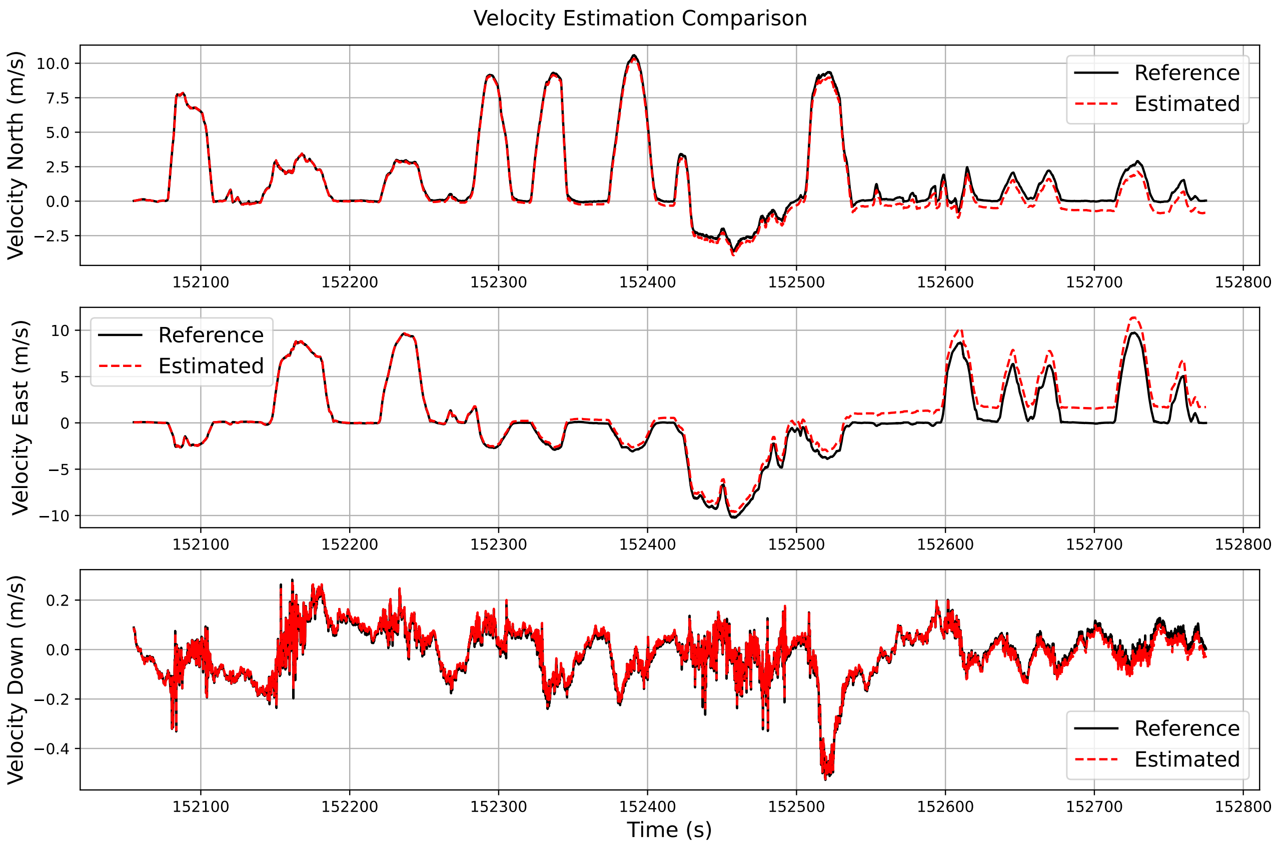

Figure 10: Comparison between the velocity data provided in dataset #2 and the velocity calculated using the INS dynamic model.

Lastly, Fig. 11 compares the orientation data from dataset #2 to the results obtained using the INS dynamics model. The overall match between the orientation measurements and the data is visually excellent, with no identifiable drift. This is, again, likely due to the single integration step required to calculate the orientation data.

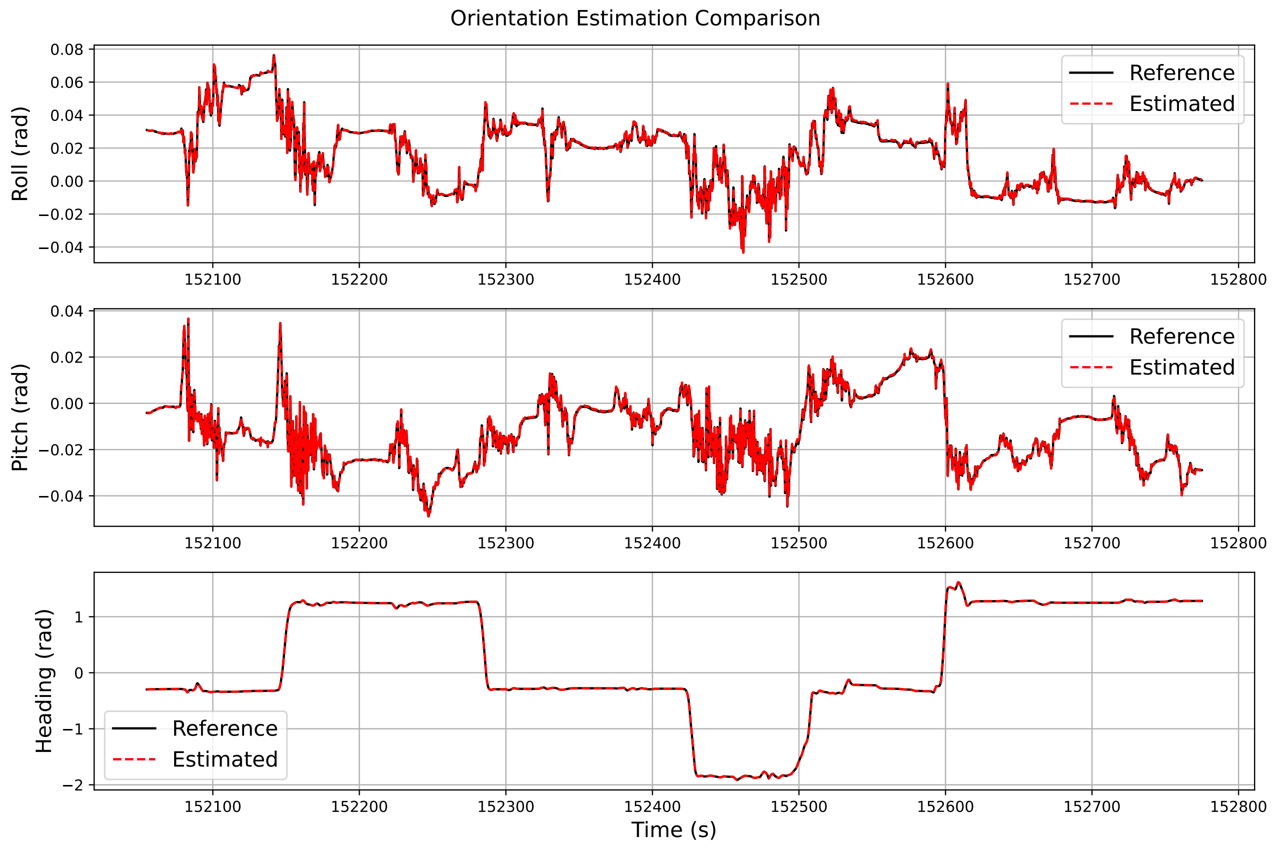

Figure 11: Comparison between the orientation data provided in dataset #2 and the orientation calculated using the INS dynamic model.

The RMSE values for the position, velocity, and orientation are summarized in Tables 5 through 7, which can be found in the following section. These values are overall small, though they are larger then their counterparts from dataset #1. In particular, the altitude error in dataset #2 is 10.4 m, compared to 1.84 m in dataset #1. Given that dataset #2 comes from a real sensor, this is expected. The real sensor is likely to have noise and bias factors which have not been accounted for in the dynamics.

Effect of Noise and Bias Factors on Dynamics - Dataset #2

Finally, the impact of noise and bias factors on the reconstructed position, velocity, and orientation data will be assessed for dataset #2. Similar to the analysis on dataset #1, all results are summarized in tables to reduce the number of figures required to convey the effect. However, to demonstrate the impact, Fig. 12 compares the position estimates to the data when noise and a bias factor are added to the z-axis of the accelerometer measurements. The impact is identical to what was observed for dataset #1, with the altitude estimation significantly diverging from the dataset as a result of the larger errors accumulating over time.

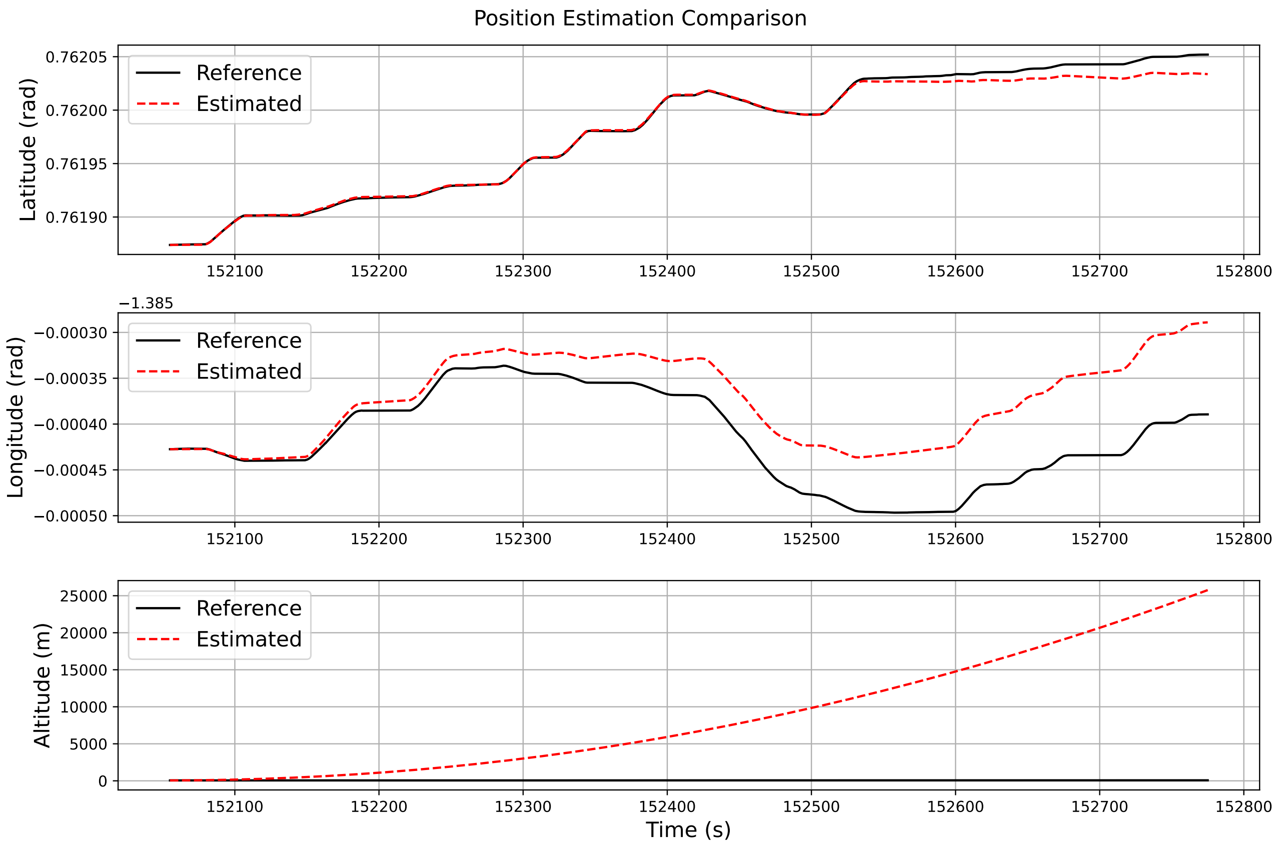

Figure 12: Comparison between the position data provided in dataset #2 and the position calculated using the INS dynamic model when noise and a bias factor are added to A_z.

Table 5 summarizes the RMSE values for the position states when noise and bias factors are applied to different axes of the accelerometer and gyroscope data. To avoid reiterating similar conclusions, the results are consistent with what was noted for dataset #1. Though, the overall magnitude of the RMSE values is much larger for the dataset #2, which is consistent with the previous discussion points.

Table 5: Root Mean Squared Error values for position states, with and without noise and bias factors (dataset #2).

Case	**Latitude (radian)**	**Longitude (radian)**	**Altitude (m)**
DS2, Default	1.52×10^{-5}	3.99×10^{-5}	1.04×10^{1}
DS2, A_x	1.13×10^{-3}	2.34×10^{-4}	1.14×10^{2}
DS2, A_y	1.80×10^{-4}	1.50×10^{-3}	1.32×10^{2}
DS2, A_z	6.14×10^{-6}	5.33×10^{-5}	1.15×10^{4}
DS2, G_x	1.87×10^{-2}	1.84×10^{-1}	8.08×10^{5}
DS2, G_y	1.32×10^{-1}	5.33×10^{-2}	7.90×10^{5}
DS2, G_z	1.74×10^{-3}	8.29×10^{-3}	2.59×10^{2}

Table 6 summarizes the RMSE values for velocity states. The overall conclusions are consistent with the results from dataset #1. The addition of noise and bias factors to the accelerometer data has a direct impact on the corresponding velocity components, while the impact of the gyroscope data is more evenly distributed across the velocities–though, it is interesting to note that adding noise and a bias factor to the z-axis of the gyroscope data has a smaller impact on the downward velocity error when compared to adding the same noise and bias to the other axes.

Table 6: Root Mean Squared Error values for velocity states, with and without noise and bias factors (dataset #2).

Case	**North Velocity (m/s)**	**East Velocity (m/s)**	**Down Velocity (m/s)**
DS2, Default	3.89×10^{-1}	9.15×10^{-1}	1.68×10^{-2}
DS2, A_x	2.33×10^{1}	5.37	4.01×10^{-1}
DS2, A_y	5.48	2.21×10^{1}	3.31×10^{-1}
DS2, A_z	2.54×10^{-1}	6.98×10^{-1}	4.13×10^{1}
DS2, G_x	7.48×10^{2}	2.82×10^{3}	3.59×10^{3}
DS2, G_y	2.93×10^{3}	7.06×10^{2}	3.48×10^{3}
DS2, G_z	3.18×10^{1}	1.34×10^{2}	2.18

Lastly, Table 7 summarizes the RMSE values for orientation states. The overall conclusions are consistent with the results from dataset #1. Adding noise and a bias factor to any axis of the gyroscope data has a significant impact on the orientation error. However, this same noise and bias, when added to the accelerometer data, has no significant impact on the orientation error.

Table 7: Root Mean Squared Error values for orientation states, with and without noise and bias factors (dataset #2).

Case	**Roll (radian)**	**Pitch (radian)**	**Azimuth (radian)**
DS2, Default	9.17×10^{-4}	9.52×10^{-4}	2.79×10^{-3}
DS2, A_x	1.25×10^{-3}	1.10×10^{-3}	2.82×10^{-3}
DS2, A_y	1.26×10^{-3}	1.43×10^{-3}	3.14×10^{-3}
DS2, A_z	9.15×10^{-4}	9.51×10^{-4}	2.79×10^{-3}
DS2, G_x	1.98	6.99×10^{-1}	2.10
DS2, G_y	2.07	6.32×10^{-1}	1.07
DS2, G_z	2.67×10^{-2}	2.43×10^{-2}	1.73

Conclusion

In this report, two different INS datasets were analysed. For both datasets, the inverse kinematics equations were used to calculate the acceleration and gyroscope measurements. In both datasets, the match between the estimates from the inverse kinematics and the data was good, with overall small Root-Mean-Square errors. However, the magntitudes of these errors was measurably larger for dataset #2, which was consistent with the fact that dataset #2 was known to have come from a real sensor (whereas dataset #1 was generated from a sensor model). As a result, larger errors were expexted from dataset #2 given the real-world noise and bias factors which were not considered in the analysis.

Following the calculation of the accelerometer and gyroscope data, an INS dynamics model was used to estimate the position, velocity, orientation for both datasets. In addition, these estimates were repeated for various noise and bias factors to determine their impact on the state estimates. With no noise or bias factors added, the estimated states compared favorably with their corresponding dataset values. A small amount of drift was observed, with the most significant drift occuring in the position state estimates. The orientation estimates on the other hand had no visible drift. It was noted that the likely cause is the greater number of integration steps required to calculate the position states, coupled with small errors present in the data that accumulate over time.

When noise and bias factors were added to the datasets, the impact was unsurprisingly significant. For the acceleration, noise added or a bias added to one axis appeared to primarily impact the corresponding estimate in line with that axis; in other words, adding a bias to A_z caused significantly more error in the altitude estimate and the downward velocity estimate. However, interestingly, increasing the noise and bias for the acceleration did not appear to have any effect on the orientation estimates. For the gyroscope data, the impact was generally more distributed when assessing the effect on the position and velocity estimates. However, even a small noise or bias factor on the gyroscope data caused signficant errors in the orientation data.

Appendix A: Software Instructions

All code was written in Python 3.13.2 on a Linux machine (Fedora 41). However, there is no reason this Python code should not work on a Windows computer, assuming the correct version of Python and the correct packages are used. The code was executed in Visual Studio Code. The following Python packages were used in the code:

- `numpy`
- `matplotlib`
- `scipy`

In addition to the main script, some functions written for the porject are contained within `project_utils.py`. No installation is required, as long as it is located in the same working directory as the main script. There is only one main script for this project:

- `project_2.py`: This script contains everything needed to generate the results discussed in this report.

The following booleans control which dataset is being used:

- `DATASET1`: If `True`, dataset #1 is used.
- `DATASET2`: If `True`, dataset #2 is used.

The following booleans control the plots that are generated.

- `NOISE_ACCEL_X`: If `True`, the impact of noise and bias on the x-axis of the accelerometer data is plotted.
- `NOISE_ACCEL_Y`: If `True`, the impact of noise and bias on the y-axis of the accelerometer data is plotted.
- `NOISE_ACCEL_Z`: If `True`, the impact of noise and bias on the z-axis of the accelerometer data is plotted.
- `NOISE_GYRO_X`: If `True`, the impact of noise and bias on the x-axis of the gyroscope data is plotted.
- `NOISE_GYRO_Y`: If `True`, the impact of noise and bias on the y-axis of the gyroscope data is plotted.
- `NOISE_GYRO_Z`: If `True`, the impact of noise and bias on the z-axis of the gyroscope data is plotted.