Calibration of an Accelerometer Using GPS Measurements

Sanskar Anil Nalkande

December 10, 2024

1 Abstract

The project focuses on the precise calibration of an accelerometer to enhance its ability to accurately track position and velocity. By integrating measurements from a GPS, we establish a framework for correcting discrepancies in accelerometer data. A Kalman Filter is employed as the core estimation technique, leveraging its ability to dynamically predict and correct deviations in position and velocity from true values. Additionally, the filter identifies and compensates for the time-invariant bias present in the accelerometer's acceleration signals. This calibration methodology improves the reliability and accuracy of accelerometer data, making it suitable for applications requiring precise motion tracking.

2 Introduction

Accurate motion tracking is crucial for navigation systems, but accelerometers often face challenges like noise and bias, while GPS systems operate at lower sampling rates. For a system characterized by a dynamic state transition model, Gaussian process noise, and sensors with Gaussian noise, a Kalman Filter can be designed to effectively estimate the system's state. In this case, by integrating data from both sensors through a Kalman Filter, errors in position, velocity, and accelerometer bias are minimized, enabling more reliable motion tracking.. The Kalman Filter is configured such that its model does not depend on the specific acceleration values.

3 Theory

We are considering a vehicle that accelerates in one dimension in an inertial frame. The acceleration is a harmonic of the form:

$$a(t) = 10\sin(\omega t),\ \omega = 0.2rad/s \tag{1}$$

3.1 Accelerometer Model

The acceleration is measured by an accelerometer with a sample rate of 200 Hz at sample times t_j. The accelerometer is modeled with additive white Gaussian noise w with zero mean and variance $V = 0.0004\ \mathrm{m}^2/\mathrm{s}^4$. The accelerometer has a bias b_a with *a priori* statistics $b_a \sim N(0,\, 0.01\ \mathrm{m}^2/\mathrm{s}^4)$. Considering the bias and Gaussian noise, the accelerometer $\mathrm{a_c}$ is modeled as:

$$a_c(t_j) = a(t_j) + b_a + w(t_j) \tag{2}$$

Using laws of motion, we have a simple model for computing the position and velocity of the vehicle which is given by the equations:

$$v_c(t_{j+1}) = v_c(t_j) + a_c(t_j)\Delta t \tag{3}$$

$$p_c(t_{j+1}) = p_c(t_j) + v_c(t_j)\Delta t + \frac{1}{2}a_c(t_j)\Delta t^2 \tag{4}$$

With the given initial conditions:

$$p_c(0) = \bar{p}_0 = 0 \tag{5}$$

$$v_c(0) = \bar{v}_0 = 100 \text{ m/s} \tag{6}$$

Here, $v_c(t_j)$ and $p_c(t_j)$ are the velocity and position of the vehicle at time t_j computed using the acceleration measurements from the accelerometer. Also, $\Delta t = t_j - t_{j-1} = \frac{1}{200Hz} = 0.005$s.

3.2 GPS Model

The GPS receiver gives position and velocity estimates in an inertial space at sample times t_i with a frequency of 5 Hz (synchronized with the accelerometer). The GPS measurements are available in the following form:

$$z_i = \begin{cases} z_{1i} = x_i + \eta_{1i} \\ z_{2i} = v_i + \eta_{2i} \end{cases} \tag{7}$$

The additive measurement noises are assumed to be white noise sequences and independent of each other with statistics:

$$\begin{bmatrix} \eta_{1i} \\ \eta_{2i} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 1 \text{ m}^2 & 0 \\ 0 & 0.0016 \text{ m}^2/\text{s}^2 \end{bmatrix}\right) \tag{8}$$

3.3 Kalman Filter

To derive the system model for the Kalman filter, we assume that the true acceleration is also obtained by using the same Euler integration formula that was employed in the accelerometer model in order to develop the model for vehicle motion that is given by the equation:

$$v_E(t_{j+1}) = v_E(t_j) + a(t_j)\Delta t \tag{9}$$

$$p_E(t_{j+1}) = p_E(t_j) + v_E(t_j)\Delta t + \frac{1}{2}a(t_j)\Delta t^2 \tag{10}$$

Here, $a(t_j)$ is the true acceleration which is formulated in equation (1). The given initial conditions are:

$$p(0) = p_E(0) \sim N(0, 100 \text{ m}^2) \tag{11}$$

$$v(0) = v_E(0) \sim N(100 \text{ m/s}, 1 \text{ m}^2/\text{s}^2) \tag{12}$$

Hence the true velocity and position can be calculated using the formula:

$$v(t) = v(0) + \frac{a}{\omega}(1 - \cos(\omega t)) = v(0) + 50(1 - \cos(0.2t)) \tag{13}$$

$$p(t) = p(0) + (v(0) + \frac{a}{\omega})t - \frac{a}{\omega^2}\sin(\omega t) = p(0) + (v(0) + 50)t - 250(\sin(0.2t)) \tag{14}$$

We can construct a stochastic discrete time system using the two sets of Euler integration formulas. This system will be approximately independent of the acceleration profile and is given by:

$$\delta x(t_{j+1}) = \Phi \delta x(t_j) + \Gamma w(t_j) \tag{15}$$

$$\begin{bmatrix} \delta p_E(t_{j+1}) \\ \delta v_E(t_{j+1}) \\ b(t_{j+1}) \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & \frac{1}{2}\Delta t^2 \\ 0 & 1 & -\Delta t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \delta p_E(t_j) \\ \delta v_E(t_j) \\ b(t_j) \end{bmatrix} + \begin{bmatrix} -\frac{1}{2}\Delta t^2 \\ -\Delta t \\ 0 \end{bmatrix} w(t_j) \tag{16}$$

Here the third state $b(t_{j+1})$ is the time invariant bias of the accelerometer, and this model describes the interaction between the bias and the motion of the vehicle The above stochastic discrete time system is used to construct the Kalman Filter with the state:

$$\hat{\delta} x(t_j) = \begin{bmatrix} \hat{\delta} p(t_j) \\ \hat{\delta} v(t_j) \\ \hat{b}(t_j) \end{bmatrix} = \begin{bmatrix} \hat{p}(t_j) - p_c(t_j) \\ \hat{v}(t_j) - v_c(t_j) \\ \hat{b}(t_j) \end{bmatrix} \tag{17}$$

4 Algorithm

The Kalman Filter has a given initial state estimate and associated covariance matrix as below:

$$\bar{\delta} x(0) = \begin{bmatrix} \bar{\delta} p(t_0) \\ \bar{\delta} v(t_0) \\ \bar{b}(t_0) \end{bmatrix} = \begin{bmatrix} \hat{p}(t_0) - p_c(t_0) \\ \hat{v}(t_0) - v_c(t_0) \\ \hat{b}(t_0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{18}$$

$$M_0 = \begin{bmatrix} 100\,\mathrm{m}^2 & 0 & 0 \\ 0 & 1\,\mathrm{m}^2/\mathrm{s}^2 & 0 \\ 0 & 0 & 0.01\,\mathrm{m}^2/\mathrm{s}^4 \end{bmatrix} \tag{19}$$

We already have the *a priori* statistics x_0, v_0 and b_a. Arranging equation 7 to be of the form:

$$z_i = Hx_i + \eta \tag{20}$$

We get:

$$\begin{bmatrix} z_{1i} \\ z_{2i} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_i \\ v_i \end{bmatrix} + \begin{bmatrix} \eta_{1i} \\ \eta_{2i} \end{bmatrix} \tag{21}$$

We already have the statistics for η from equation (8). This measurement is likely to conflict with the current state estimate and should be addressed through the *a posteriori* update process given by:

$$K_0 = M_0 H^T (H M_0 H^T + V)^{-1} \tag{22}$$

$$\hat{\delta} x(0) = \bar{\delta} x(0) + M_0 H^T V^{-1} (z_0 - H\bar{x}(0)) \tag{23}$$

$$P_0 = (I - K_0 H) M_0 (I - K_0 H)^T + K_0 V K_0^T \tag{24}$$

At each subsequent time step, an *a priori* estimate is first generated, followed by receiving a new measurement from the GPS sensor. The discrepancy between the *a priori* estimate and the sensor measurement is then resolved using the *a posteriori* posteriori update equations. This *a priori* estimate is determined by propagating the state estimate's mean and covariance as follows:

$$\bar{x}_1 = \Phi \hat{x}_0 \tag{25}$$

$$M_1 = \Phi P_0 \Phi^T + \Gamma W \Gamma^T \tag{26}$$

When we have the measurement z_1, we run the *a posteriori* update which is defined by the following equations:

$$K_1 = M_1 H^T (H M_1 H^T + V)^{-1} \tag{27}$$

$$\hat{\delta} x(t_1) = \bar{\delta} x(t_1) + M_1 H^T V^{-1} (z_1 - H\bar{x}(t_1)) \tag{28}$$

$$P_1 = (I - K_1 H) M_1 (I - K_1 H)^T + K_1 V K_1^T \tag{29}$$

The whole process can be repeated for the duration required. We can formulate the discrete time sequential Kalman Filter as follows. The *a priori* update process can be represented by equations:

$$\bar{x}_k = \Phi \hat{x}_{k-1} \tag{30}$$

$$M_k = \Phi P_{k-1} \Phi^T + \Gamma W \Gamma^T \tag{31}$$

And the *a posteriori* update process can be represented by equations:

$$K_k = M_k H^T (H M_k H^T + V)^{-1} \tag{32}$$

$$\hat{\delta} x(t_k) = \bar{\delta} x(t_k) + M_k H^T V^{-1} (z_k - H\bar{x}(t_k)) \tag{33}$$

$$P_k = (I - K_k H) M_k (I - K_k H)^T + K_k V K_k^T \tag{34}$$

We can calculate the estimated position and velocity for all sampling times from the state estimate using the equations:

$$\hat{p}(t_j) = p_c(t_j) + \hat{\delta} p(t_j) \tag{35}$$

$$\hat{v}(t_j) = v_c(t_j) + \hat{\delta} v(t_j) \tag{36}$$

Here, p_c and v_c are calculated using the equations (3) and (4).

5 Simulation Results

Now, we take a look at the results of the 30 second simulation by plotting various quantities that are of interest to us.

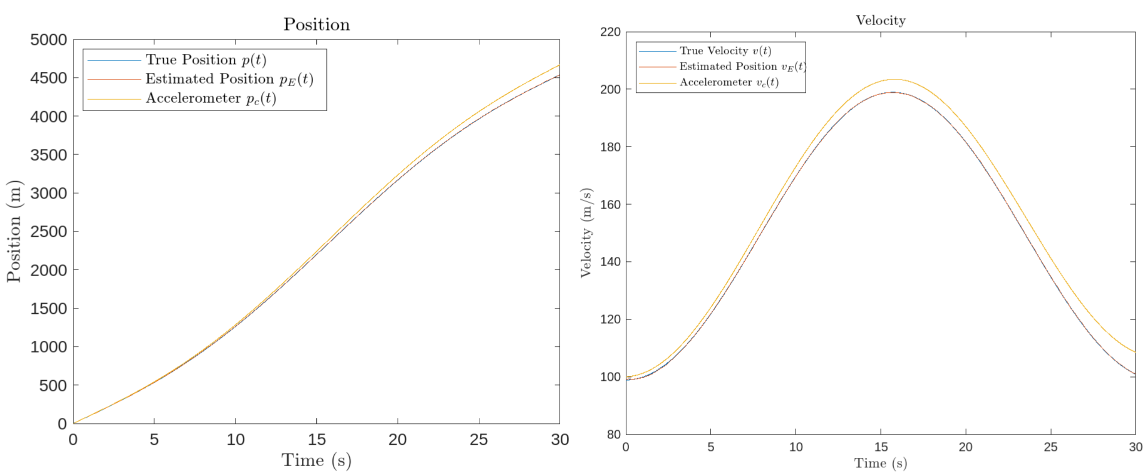

Figure 1: True, Estimated and Accelerometer values of Position and Velocity vs Time

In the plots above, we can see that the True and Estimated values of Position and Velocity are much closer to each other than to the values calculated using Acclerometer readings. This indicates that our Kalman filter has indeed improved the accuracy of the measurement.

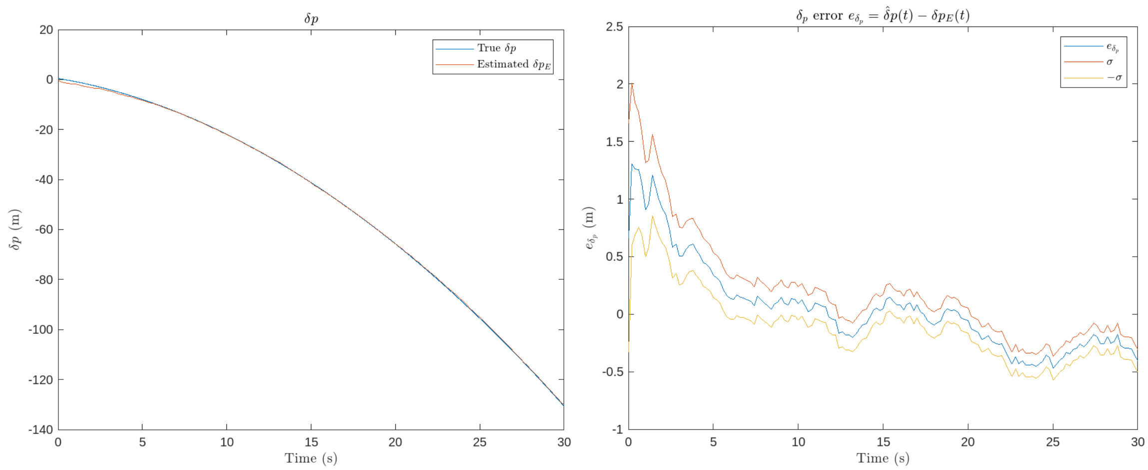

Figure 2: $\delta p(t)$ and Error $e_{\delta p} = \hat{\delta} p(t) - \delta p_E(t)$ vs Time

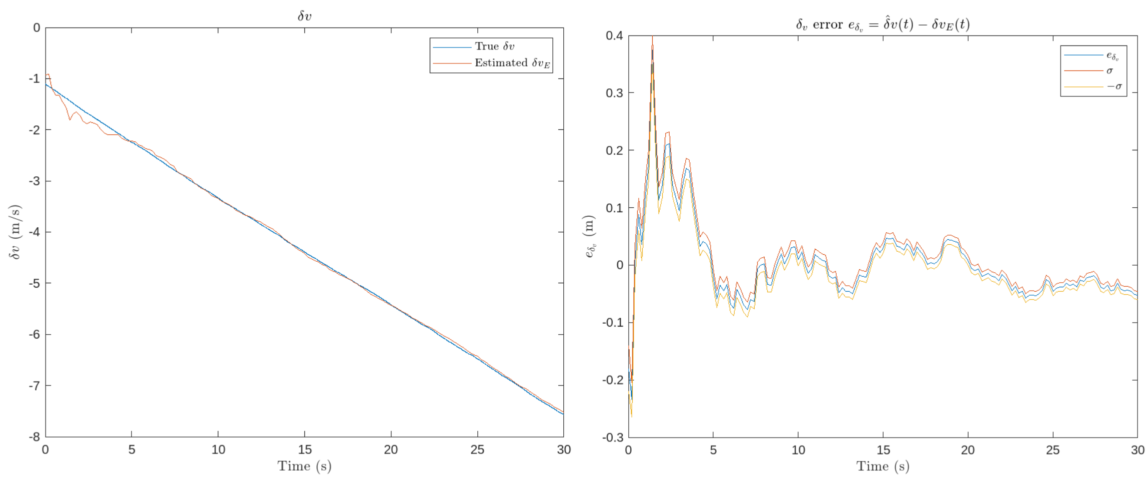

Figure 3: $\delta v(t)$ and Error $e_{\delta v} = \hat{\delta} v(t) - \delta v_E(t)$ vs Time

The plots above, in figures 2 and 3 compare the values of $\delta p(t)$ and $\delta v(t)$ with their estimated values $\delta p_E(t)$ and $\delta v_E(t)$, respectively which were obtained using the Kalman Filter. We notice that the true and the estimated values are very close to each other. The errors calculated by subtracting the estimated value from the true value also converges to zero in each case. This shows that the Kalman filter causes the estimated values to eventually converge with the true values given enough time.

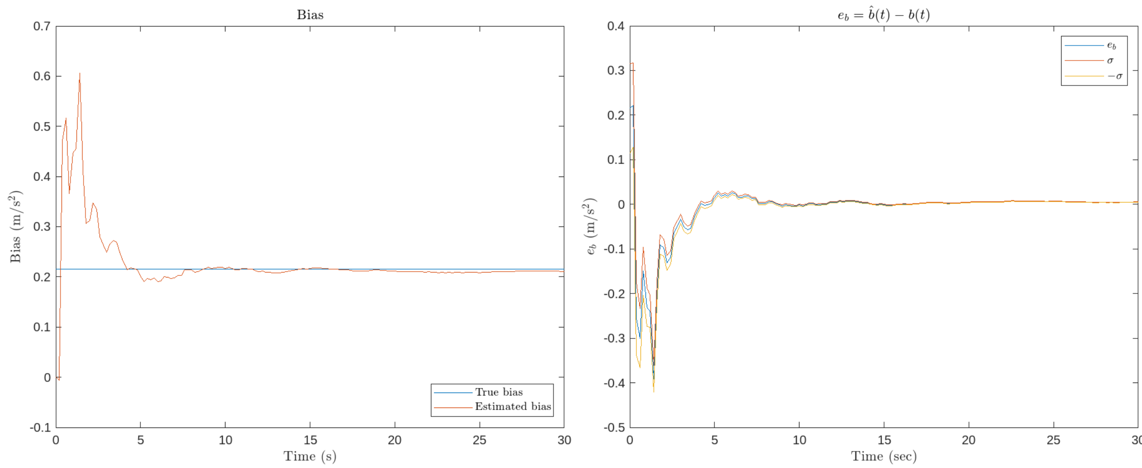

Figure 4: Bias and Error in estimated bias values vs Time

In the plot above, in figure 4, we see that the model also converges to the correct value of the accelerometer bias. With this information we can also offset the values of the estimated states in order to get closer to their true values.

6 Performance Analysis

To evaluate the performance of the Kalman Filter, the covariance matrix produced by the filter is compared to the covariance matrix obtained from 1,000 Monte Carlo simulations. The data covariance is determined by calculating the average error over time using the formula:

$$e_{\text{ave}}(t_i) = \frac{1}{N_{\text{ave}}} \sum_{l=1}^{N_{\text{ave}}} e^l(t_i) \tag{37}$$

And the covariance of the Monte Carlo simulation is calculated using the formula:

$$P_{\text{ave}}(t_i) = \frac{1}{N_{\text{ave}} - 1} \sum_{l=1}^{N_{\text{ave}}} [e^l(t_i) - e_{\text{ave}}(t_i)][e^l(t_i) - e_{\text{ave}}(t_i)]^T \tag{38}$$

Using the above equation, we will get the covariance matrix which will be a 3×3 square matrix. If our implementation of the Kalman filter is correct, each element of this matrix should eventually converge to zero. Figure 5 plots these elements of the covariance matrix P_{ave} versus time for the simulation period. We see that all the values are either zero or very close to zero. These elements have also been plotted along with the elements of the matrices M and P the values of which were calculated using the equations (31) and (34) respectively.

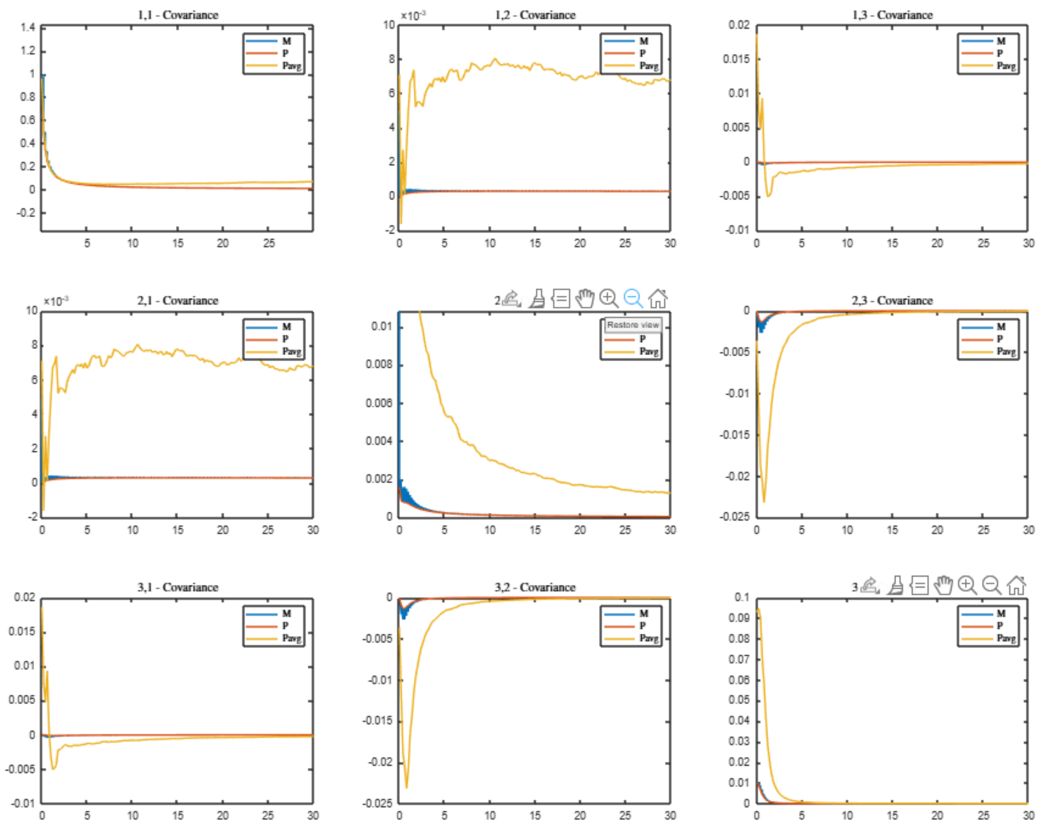

Figure 5: Elements of the average covariance matrix P_{ave} compared with the elements of the matrices M and P

Another way of assessing the performance of the Kalman filter is by checking the orthogonality of the error in estimated values. This condition is given by:

$$\frac{1}{N_{\text{ave}}}\sum_{l=1}^{N_{\text{ave}}}[e^l(t_i) - e_{\text{ave}}(t_i)]\hat{x}(t_i)^T \approx 0\,\forall\, t_i \tag{39}$$

Figure 6 illustrates that the elements of this orthogonality matrix goes to zero as time progresses. Finally, we check the independence of the residuals using the equation:

$$\frac{1}{N_{\text{ave}}}\sum_{l=1}^{N_{\text{ave}}} r^l(t_i) r^l(t_m)^T \approx 0\,\forall\, t_m < t_i \tag{40}$$

Using equation (40) to calculate the ensemble average to check the independence of the residuals at $t_m = 20.0$s and $t_i = 20.2$s we get the value 0.017396. This value satisfies the check on the independence of the residuals.

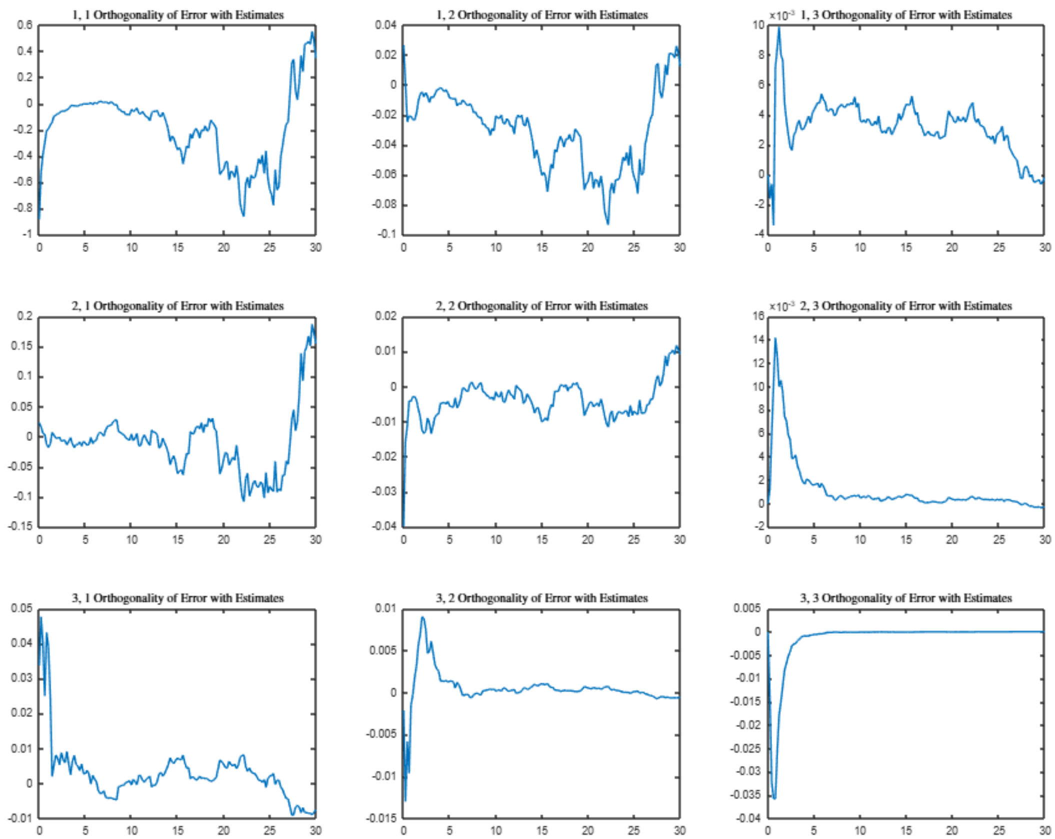

Figure 6: Elements of the average orthogonality matrix given by equation (39)

7 Conclusion

The Kalman Filter effectively estimates the accelerometer bias, as well as the position and velocity of the vehicle, by combining noisy accelerometer and GPS measurements. This sensor fusion capability enhances the accuracy of state estimation, which is crucial for systems operating in noisy environments. The filter's performance is rigorously validated through a Monte Carlo simulation with 1,000 realizations, demonstrating its robustness and consistent accuracy across diverse scenarios. Additionally, the Kalman Filter enables precise calibration of the accelerometer bias, ensuring long-term reliability in inertial navigation systems.

Furthermore, the orthogonality properties of the Kalman Filter are verified by confirming the independence of errors from posterior estimates and the independence of residuals. This verification reinforces the theoretical foundation of the filter and its practical reliability for real-world applications. These results collectively establish the Kalman Filter as a robust tool for sensor fusion and state estimation, capable of addressing challenges in noisy and dynamic environments.