

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/393590343>

Sensor Fusion for Drone Position and Attitude Estimation using Extended Kalman Filter

Preprint · July 2025

DOI: 10.21203/rs.3.rs-7087667/v1

CITATIONS

0

READS

71

1 author:



[Md Samiul Haque Motayed](#)

The University of Texas at Arlington

4 PUBLICATIONS **5** CITATIONS

SEE PROFILE

Sensor Fusion for Drone Position and Attitude Estimation using Extended Kalman Filter

Md Samiul Haque Motayed

mxm4162@mavs.uta.edu

The University of Texas at Arlington <https://orcid.org/0009-0005-3137-7091>

Research Article

Keywords: Drone, Estimation, Measurement, Extended Kalman Filter, Sensor

Posted Date: July 11th, 2025

DOI: <https://doi.org/10.21203/rs.3.rs-7087667/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: The authors declare no competing interests.

Sensor Fusion for Drone Position and Attitude Estimation using Extended Kalman Filter

Md Samiul Haque Motayed

Department of Mechanical Engineering, The University of Texas at Arlington

Mail: mxm4162@mavs.uta.edu

Abstract

Accurate real-time estimation of a drone’s position, velocity, and orientation is crucial for stable navigation, path tracking, and autonomous control. However, achieving robust state estimation is challenging due to sensor noise, environmental disturbances, and the nonlinear nature of drone dynamics. This project investigates the use of an Extended Kalman Filter (EKF) to fuse data from multiple sensors—specifically, GPS, an Inertial Measurement Unit (IMU), and a barometric altimeter—to estimate the full 9-state vector of a drone in various motion scenarios. A high-fidelity simulation environment is developed in MATLAB Simulink, incorporating a physics-based six degrees-of-freedom (6DOF) drone model. Sensor measurements are generated by simulating realistic sampling rates and noise characteristics for each sensor type. The EKF is implemented to fuse GPS-provided position and velocity, IMU-derived acceleration and angular velocity, and barometric altitude into a unified state estimate. The effectiveness of the filter is validated using different trajectory profiles—including linear, sinusoidal, and linearly increasing motions. Evaluation metrics such as Root Mean Square Error (RMSE) are used to quantify estimation performance. The results confirm that the EKF significantly reduces sensor noise and drift, resulting in reliable full-state estimation even in complex dynamic conditions.

1 Introduction

Accurate and reliable state estimation lies at the core of any modern autonomous drone system [1]. Whether the task is to follow a predefined path, avoid obstacles, or simply stay level and stable in the air, a drone must be able to constantly estimate its own position, velocity, and orientation. Doing this accurately in real time is not a trivial task [2, 3]. It requires dealing with noisy sensors, nonlinear dynamics, and uncertainties from both the environment and the sensors themselves [4–6]. In this project, we explore how a carefully implemented Extended Kalman Filter (EKF), combined with multiple onboard sensors, can successfully solve this estimation problem—even across different flight paths and motion profiles.

Drones are typically equipped with a variety of onboard sensors, each of which contributes a unique piece of information. However, no sensor is perfect [4, 7–12]. A Global Positioning System (GPS) provides direct measurements of the drone’s position and velocity in the global (inertial) frame. These measurements are generally reliable, but they have a relatively low update rate and are subject to noise, especially in cluttered or urban environments [13, 14]. On the other hand, the Inertial Measurement Unit (IMU)—which includes both accelerometers and gyroscopes—operates at a high frequency and provides rapid updates. The IMU’s accelerometer measures linear acceleration, which can be integrated over time to estimate velocity and then position. However, small measurement errors can quickly accumulate, leading to drift [15–20]. The gyroscope provides angular velocity, which can be integrated once to estimate roll, pitch, and yaw angles, but these too are

prone to long-term bias and drift [21]. Finally, a barometric altimeter adds another useful source of data, offering altitude estimates based on atmospheric pressure differences. This can improve the vertical estimation accuracy, particularly when GPS measurements are unreliable.

This project combines all three of these sensors—GPS, IMU (accelerometer and gyroscope), and barometer—into a single framework using a 9-state Extended Kalman Filter. The filter estimates the full state vector, including 3D position (x, y, z) , 3D velocity (v_x, v_y, v_z) , and 3D orientation angles (ϕ, θ, ψ) . The EKF operates in two steps: a prediction step, which uses the system’s nonlinear dynamic model to estimate the next state based on current measurements; and an update step, which corrects this prediction based on new incoming data from the sensors [22–24]. This process runs recursively, updating the state estimate and its uncertainty in real time as the drone flies.

To generate data, we built a detailed simulation using MATLAB Simulink. The drone is modeled using a physics-based six degrees-of-freedom (6DOF) dynamic block. The simulation includes realistic physics and environmental influences. In our setup, we apply force inputs in the x , y , and z directions to simulate the drone’s movement, but apply no moment inputs. This allows us to focus on translational motion while still evaluating orientation changes using gyroscopic data. We designed the simulation to generate diverse flight routes to evaluate how well the EKF performs under different motion conditions.

Each sensor in the simulation is modeled with realistic noise levels, sampling frequencies, and drift characteristics. The GPS data includes position and velocity at a lower update rate; the IMU provides high-frequency accelerometer and gyroscope data; and the barometer offers altitude estimates based on pressure changes. All sensor outputs are resampled and aligned over a common time base to support consistent and accurate fusion. The EKF’s performance is analyzed by comparing the estimated states to the true states from the simulation model, using both numerical metrics like the Root Mean Square Error (RMSE) and visual inspection through plotted trajectories.

In addition to accuracy, we also evaluate the filter’s statistical consistency by checking if the estimation errors stay within the predicted $\pm 3\sigma$ bounds of the EKF’s covariance matrix. This provides an important check on whether the filter is correctly modeling its uncertainty.

Through this study, we demonstrate how sensor fusion through EKF can significantly improve the quality of state estimation for drones, even when dealing with real-world sensor limitations. The approach not only leverages the strengths of each sensor, but also compensates for their individual weaknesses. The results provide a solid foundation for future extensions—such as adding bias estimation in a 15-state EKF, incorporating adaptive noise tuning, or integrating vision-based sensors for enhanced navigation.

2 Sensor Data Generation

To validate the performance of the Extended Kalman Filter (EKF) sensor fusion algorithm, realistic sensor measurements were generated using a Simulink-based model of a quadcopter. A six degrees of freedom (6DOF) dynamic model was employed to simulate the true translational and rotational motion of the drone, considering gravitational, inertial, and external force effects.

2.1 Simulated Sensors

Three types of sensors were simulated:

- **GPS Subsystem:** Provided position and velocity measurements in the North-East-Down (NED) frame. The GPS outputs were generated by converting the true position from Earth-

Centered coordinates to LLA, then transforming to NED, and adding realistic noise and drift. Sampling rate: 10 Hz.

- **IMU Subsystem:** Provided linear acceleration, angular velocity, and orientation data in the body frame. The acceleration outputs were corrected for gravitational effects. Additional sensor imperfections such as white noise and bias drift were modeled to reflect a realistic IMU behavior. Sampling rate: 100 Hz.
- **Barometer Subsystem:** Provided altitude (Z-axis position) measurements. The barometric sensor model included random walk drift, sensor delay, and atmospheric noise to replicate realistic pressure-to-altitude conversion errors. Sampling rate: 50 Hz.

The generated noisy measurements from each subsystem were logged during simulation and used as input for the EKF sensor fusion algorithm to estimate the position and velocity states.

2.2 Simulink Model Architecture

The overall Simulink architecture used for generating the synthetic sensor measurements is shown in Figure 1. The model consists of a six degrees of freedom (6DOF) dynamics block simulating the quadcopter motion, along with separate GPS, IMU, and Barometer subsystems that process the true states to output noisy, realistic sensor measurements. Each subsystem architecture is further shown individually in Figures 2, 3, and 4.

The GPS subsystem processes the dynamic model’s true position and velocity outputs. It converts the Cartesian coordinates into latitude, longitude, and altitude (LLA) and transforms them into the North-East-Down (NED) local reference frame. Realistic noise and drift are introduced to simulate actual GPS measurements. The structure of the GPS Subsystem is shown in Figure 2.

The IMU subsystem models the inertial measurement unit (IMU) outputs by simulating accelerometer and gyroscope behavior. The true linear accelerations and angular velocities are transformed into the body frame. Gravitational acceleration is subtracted to obtain the inertial accelerations. Measurement noise and sensor bias are introduced to replicate realistic IMU characteristics. The subsystem outputs the measured acceleration along with the corresponding velocity and position estimates. The IMU subsystem is shown in Figure 3.

The Barometer Subsystem simulates altitude measurements based on atmospheric pressure. The true altitude is converted into pressure using a standard barometric equation. Sensor imperfections such as drift, random noise fluctuations, and time delay are incorporated to replicate realistic barometric sensor behavior. The final output provides a noisy measurement of altitude. The structure of the Barometer Subsystem is illustrated in Figure 4.

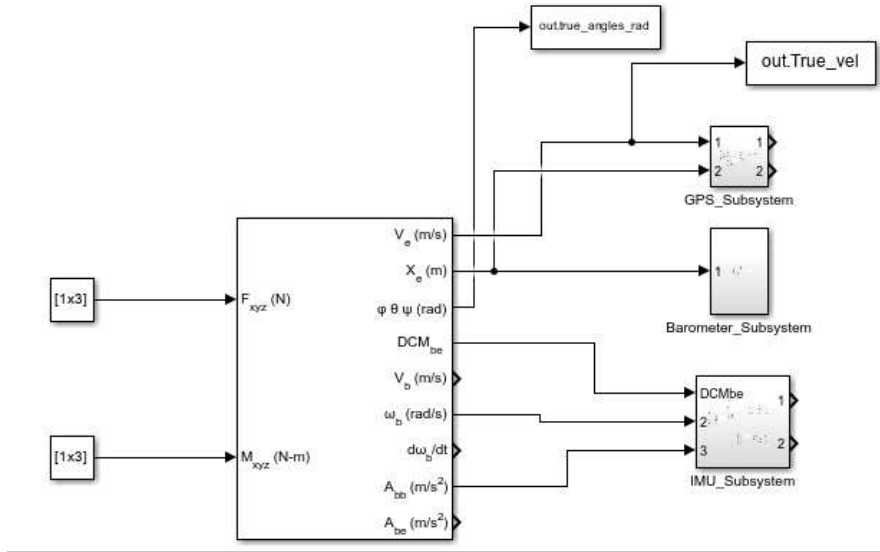


Figure 1: Overall Simulink model for sensor data generation.

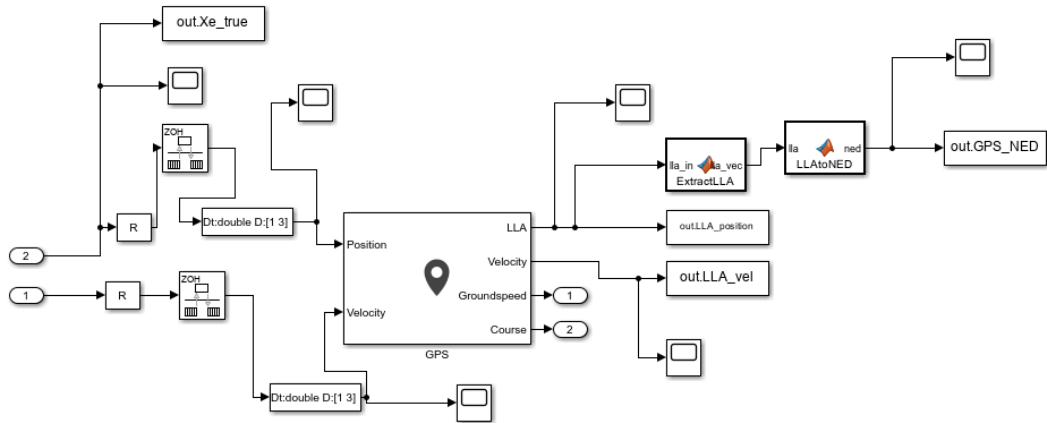


Figure 2: Simulink GPS Subsystem: Position and velocity measurement generation.

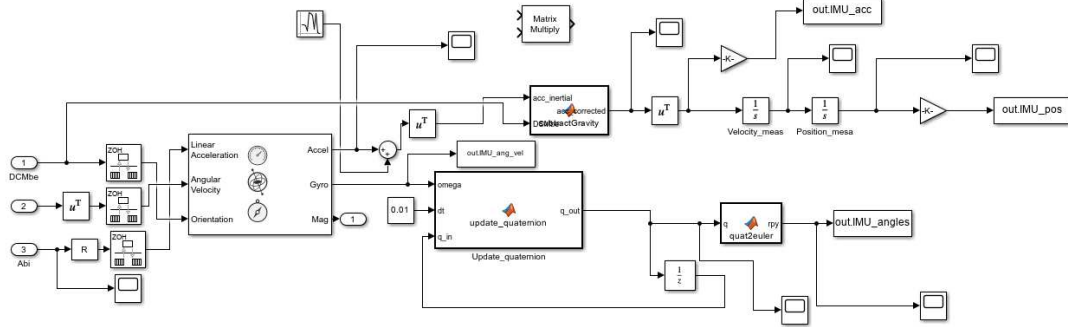


Figure 3: Simulink IMU Subsystem: Acceleration and angular velocity measurement generation.

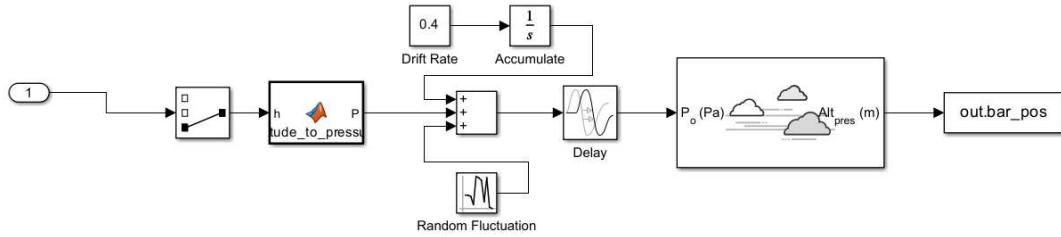
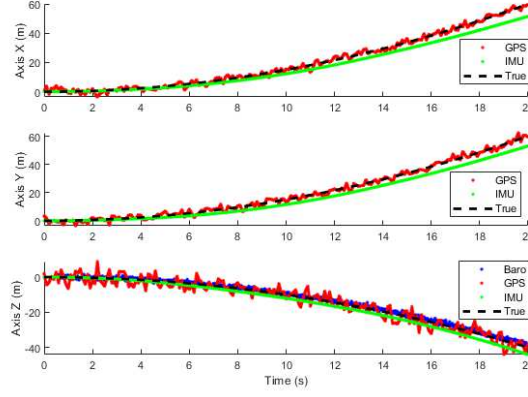


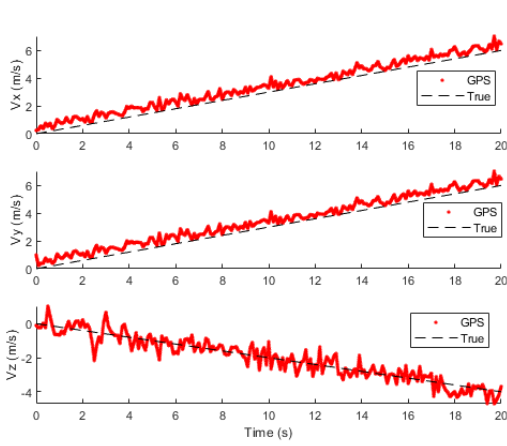
Figure 4: Simulink Barometer Subsystem: Altitude measurement generation.

2.3 Generated Data

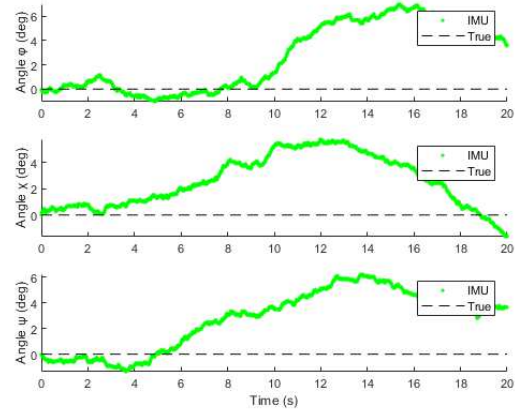
(m-file:Data_positions_attitude_full.m)



(a) Measured and true position data from GPS, IMU, and Barometer (X, Y, Z axes).



(b) Measured and true velocity data from GPS.

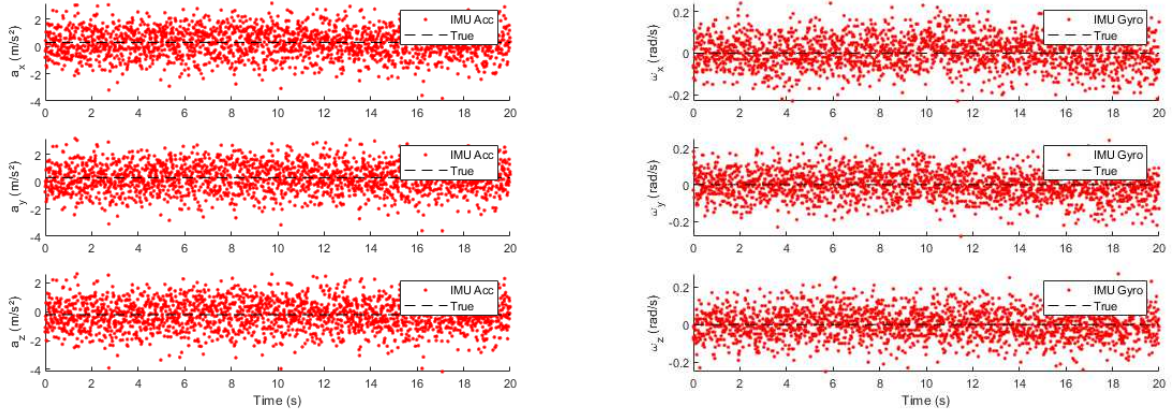


(c) Measured and true attitude angles from IMU gyroscope.

Figure 5: Sensor measurements and ground truth from the Simulink drone model.

Figure 5 illustrates the simulated sensor measurements alongside the true states generated by the Simulink drone model. Subfigure 5a shows the 3D position data captured from GPS, IMU, and Barometer sensors. While GPS and Barometer readings exhibit noticeable noise and fluctuations, the overall trajectory remains consistent with the ground truth. Subfigure 5b presents the velocity data obtained from GPS, which tracks the increasing speed in X and Y directions and descending trend in Z. Subfigure 5c shows the attitude angles (roll ϕ , pitch θ , yaw ψ) estimated from the IMU gyroscope. These measurements demonstrate drift over time, typical for unaided gyroscopic integration. It is important to note that this data was generated using force inputs in the X, Y, and Z directions, with zero moment input applied to the 6DOF block. As a result, any changes in attitude are caused solely by sensor noise and integration drift rather than rotational dynamics.

Figure 6 shows the measured inertial data from the IMU compared to the true values generated by the Simulink drone model. Subfigure 6a presents the linear acceleration measurements (a_x , a_y ,

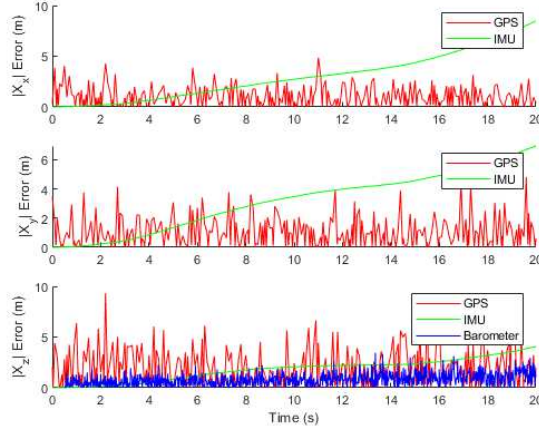


(a) IMU-measured acceleration vs. true acceleration (a_x, a_y, a_z). (b) IMU-measured angular velocity vs. true ($\omega_x, \omega_y, \omega_z$).

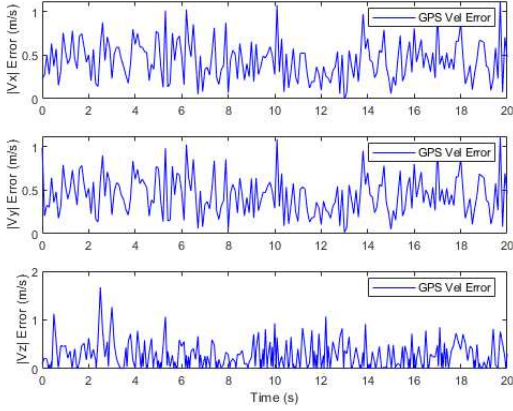
Figure 6: Comparison between IMU measurements and true values for linear acceleration and angular velocity.

a_z), while Subfigure 6b illustrates the angular velocity measurements ($\omega_x, \omega_y, \omega_z$). Both sets of measurements exhibit significant noise, as expected in a realistic low-cost IMU scenario. Notably, the angular velocities fluctuate around zero due to the zero moment input in the 6DOF block, which confirms that the system did not experience actual rotation—only translational motion from the applied force inputs. These noisy sensor readings highlight the necessity for sensor fusion techniques like EKF to obtain reliable state estimates.

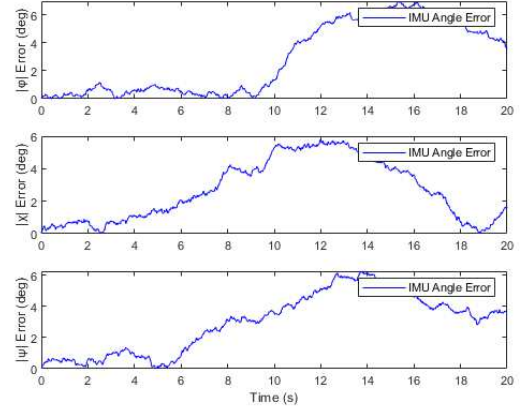
Figure 7 illustrates the absolute error between the measured values and the ground truth for position, velocity, and attitude. In Subfigure 7a, we observe that the IMU-derived position errors grow over time, especially along the X and Y axes, due to drift caused by noise and double integration. The GPS shows more stable but noisier errors. The barometric altitude data maintains the lowest error along Z. Subfigure 7b shows the velocity errors from GPS, which remain bounded but fluctuate throughout the trajectory. Finally, Subfigure 7c shows increasing errors in roll, pitch, and yaw estimates from the IMU due to gyroscopic integration drift—emphasizing again that no moment input was applied to the system. These results highlight the necessity of Kalman filtering to suppress drift and fuse complementary sensor data effectively.



(a) Absolute error in position from GPS, IMU, and Barometer ($|x|$, $|y|$, $|z|$).



(b) Absolute velocity error from GPS ($|v_x|$, $|v_y|$, $|v_z|$).



(c) Absolute attitude angle error from IMU ($|\phi|$, $|\theta|$, $|\psi|$).

Figure 7: Absolute error in position, velocity, and attitude measurements with respect to ground truth.

3 Problem Statement

3.1 Dynamic Modeling and EKF State Transition

The nonlinear dynamic model of the drone is represented using a 9-state vector:

$$\mathbf{x} = [x \ y \ z \ v_x \ v_y \ v_z \ \phi \ \theta \ \psi]^\top$$

where $[x, y, z]$ is the position, $[v_x, v_y, v_z]$ is the velocity, and $[\phi, \theta, \psi]$ are the roll, pitch, and yaw angles (ZYX Euler angles).

3.1.1 Nonlinear State Transition Model

The discrete-time nonlinear motion model is defined as:

$$\begin{aligned}\mathbf{p}_{k+1} &= \mathbf{p}_k + \mathbf{v}_k \Delta t + \frac{1}{2} \mathbf{R}_k \mathbf{a}_k \Delta t^2 \\ \mathbf{v}_{k+1} &= \mathbf{v}_k + \mathbf{R}_k \mathbf{a}_k \Delta t \\ \boldsymbol{\Theta}_{k+1} &= \boldsymbol{\Theta}_k + \mathbf{T}_k \boldsymbol{\omega}_k \Delta t\end{aligned}$$

Here, \mathbf{R}_k is the rotation matrix from body to inertial frame using ZYX Euler angles, \mathbf{a}_k is the acceleration vector in body frame, $\boldsymbol{\omega}_k$ is the angular velocity vector from the gyroscope, and \mathbf{T}_k is the transformation from body angular rates to Euler angle rates.

3.1.2 Rotation Matrix (ZYX Convention)

The rotation matrix $\mathbf{R}(\phi, \theta, \psi)$ is given by [25, 26]:

$$\mathbf{R} = \begin{bmatrix} c_\theta c_\psi & c_\psi s_\theta s_\phi - s_\psi c_\phi & c_\psi s_\theta c_\phi + s_\psi s_\phi \\ c_\theta s_\psi & s_\psi s_\theta s_\phi + c_\psi c_\phi & s_\psi s_\theta c_\phi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix}$$

where $c_\phi = \cos \phi$, $s_\phi = \sin \phi$, and similarly for θ and ψ .

3.1.3 Euler Angle Rate Transformation Matrix

The transformation from body angular velocity $\boldsymbol{\omega}$ to Euler angle derivatives is given by [26]:

$$\mathbf{T} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi / \cos \theta & \cos \phi / \cos \theta \end{bmatrix}$$

3.1.4 Jacobian Matrices for EKF

The state transition Jacobian matrix \mathbf{F} used in the EKF prediction step is composed as:

$$\mathbf{F} = \frac{\partial f}{\partial \mathbf{x}} = \begin{bmatrix} \mathbf{I}_3 & \Delta t \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_3 & \Delta t \cdot \frac{\partial(\mathbf{R}\mathbf{a})}{\partial[\phi, \theta, \psi]} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_3 + \Delta t \cdot \frac{\partial(\mathbf{T}\boldsymbol{\omega})}{\partial[\phi, \theta, \psi]} \end{bmatrix}$$

The matrix $\partial(\mathbf{R}\mathbf{a})/\partial[\phi, \theta, \psi]$ is computed using the helper function `dR_dangles`, and $\partial(\mathbf{T}\boldsymbol{\omega})/\partial[\phi, \theta, \psi]$ is calculated using `dT_dangles`.

3.2 EKF Measurement Model

The measurement model relates the system states to the outputs of various onboard sensors. In this project, we use measurements from GPS, barometer, IMU-derived position, and IMU-derived attitude (Euler angles). The general form of the measurement equation is:

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k$$

where:

- \mathbf{z}_k : measurement vector at time step k
- $\mathbf{x}_k \in \mathbb{R}^9$: system state vector
- \mathbf{H}_k : measurement matrix
- $\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R})$: zero-mean Gaussian measurement noise

The full state vector is:

$$\mathbf{x} = [x \quad y \quad z \quad v_x \quad v_y \quad v_z \quad \phi \quad \theta \quad \psi]^T$$

We define each measurement model below.

3.2.1 GPS Position Measurement

The GPS provides 3D position in the inertial frame, i.e., $\mathbf{z}_{\text{GPS}} = [x, y, z]^T$. Thus:

$$\mathbf{H}_{\text{GPS}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{3 \times 9}$$

3.2.2 GPS Velocity Measurement

GPS also provides velocity in the inertial frame: $\mathbf{z}_{\text{vel}} = [v_x, v_y, v_z]^T$. The measurement matrix is:

$$\mathbf{H}_{\text{vel}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{3 \times 9}$$

3.2.3 Barometer Altitude Measurement

The barometer provides the vertical position (altitude), i.e., z -component of position. The measurement matrix is:

$$\mathbf{H}_{\text{baro}} = [0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0] \in \mathbb{R}^{1 \times 9}$$

3.2.4 IMU-based Position Estimate (from acceleration integration)

This estimate is used as a pseudo-measurement of position:

$$\mathbf{H}_{\text{IMU-pos}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \mathbf{H}_{\text{GPS}}$$

3.2.5 IMU Attitude Measurement (Euler Angles from Gyro Integration)

Attitude (roll, pitch, yaw) estimated from gyroscope integration (e.g., using quaternion integration) is also used as a measurement:

$$\mathbf{H}_{\text{attitude}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{3 \times 9}$$

This model directly observes the roll ϕ , pitch θ , and yaw ψ components of the state vector.

3.2.6 Measurement Noise Covariance Matrix \mathbf{R}

Each measurement has its own noise covariance matrix based on sensor specifications. These matrices are typically computed from experimental variance or known sensor datasheets:

$$\begin{aligned} \mathbf{R}_{\text{GPS}} &\in \mathbb{R}^{3 \times 3} \\ \mathbf{R}_{\text{vel}} &\in \mathbb{R}^{3 \times 3} \\ \mathbf{R}_{\text{baro}} &\in \mathbb{R}^{1 \times 1} \\ \mathbf{R}_{\text{IMU-pos}} &\in \mathbb{R}^{3 \times 3} \\ \mathbf{R}_{\text{attitude}} &\in \mathbb{R}^{3 \times 3} \end{aligned}$$

These noise matrices are used in the Kalman gain computation during the measurement update step of the EKF.

Summary

The measurement model combines absolute position (from GPS), vertical altitude (barometer), and attitude and pseudo-position estimates (from IMU) to refine the nonlinear state estimates. Each sensor provides partial observation of the state vector, and the EKF fuses them into a coherent full-state estimate.

4 Solution Methodology

The goal of this work is to estimate the full 9-dimensional state of a drone, including 3D position, velocity, and attitude angles using a set of noisy sensor measurements. To achieve this, we employ an Extended Kalman Filter (EKF) which allows nonlinear system modeling and real-time fusion of multiple asynchronous sensor inputs. The filter operates in discrete time with a sampling interval of $\Delta t = 0.01$ s.

Data Preprocessing and Time Alignment

Sensor data was extracted from Simulink outputs and reshaped using MATLAB's `permute` and `squeeze` operations. As the sensors had different sampling times and initial timestamps, all measurements were interpolated to a common timeline using linear interpolation. This ensured synchronous updates in the EKF loop for sensors with different rates.

Initialization

The state vector was initialized using the first valid GPS position reading and the corresponding IMU-derived attitude angles. Velocity was initialized to zero. The initial covariance matrix P_0 was

set to an identity matrix scaled by 1:

$$P_0 = 1 \cdot \mathbf{I}_9$$

Process Noise Covariance

The process noise covariance matrix Q was manually tuned to balance filter responsiveness and stability. It is defined as:

$$Q = \text{diag}([10^{-2}, 10^{-2}, 10^{-2}, 10^{-1}, 10^{-1}, 10^{-1}, 10^{-3}, 10^{-3}, 10^{-3}])$$

where the first three terms correspond to position noise, the next three to velocity noise, and the last three to attitude angle noise. These values were chosen empirically to reflect moderate uncertainty in translational motion and lower uncertainty in angular drift over small time steps.

Filter Structure and Fusion Logic

The EKF prediction step uses IMU acceleration and angular velocity to propagate the state. A rotation matrix derived from the current Euler angles is used to transform body-frame acceleration into the inertial frame. The angular rates are converted to Euler angle rates via a transformation matrix.

The correction step selectively incorporates available measurements at each time step. The EKF supports asynchronous updates from:

- GPS position
- GPS velocity
- IMU-integrated position
- Barometric altitude
- IMU-measured Euler angles

Implementation Notes

The full EKF loop stores the estimated state and diagonal of the covariance matrix at each time step. Absolute errors were computed between the estimated and true states to evaluate estimation quality. The filter was tested across multiple routes with both linear and nonlinear motion patterns. Plots were generated for position, velocity, attitude, and respective absolute errors. A 3D trajectory plot was also included to visualize the EKF tracking performance.

Fusion Workflow

The fusion logic, as outlined in Algorithm 1, ensures that each available sensor is used to minimize the uncertainty in the estimated state. The Kalman Gain adapts the correction based on relative trust in prediction vs measurement.

This structure allows robust estimation under multi-rate, noisy conditions and demonstrates strong performance across various drone trajectories.

Algorithm 1 EKF-based Sensor Fusion for Position, Velocity, and Attitude Estimation

```
1: Input: Time-synchronized measurements from GPS, IMU (accelerometer and gyroscope),  
   barometer  
2: Output: Estimated state vector  $\mathbf{x} = [x, y, z, v_x, v_y, v_z, \phi, \theta, \psi]^\top$   
3: Initialize:  $\mathbf{x}_0$  from first valid GPS and IMU measurements;  $P_0$ ,  $Q$ , and all  $R$  matrices  
4: for  $k = 1$  to  $N$  do  
5:   Extract IMU acceleration  $\mathbf{a}_k$  and angular velocity  $\boldsymbol{\omega}_k$   
6:   Compute rotation matrix  $R_k$  from Euler angles  
7:   Predict position:  $\mathbf{p}_k = \mathbf{p}_{k-1} + \mathbf{v}_{k-1} \cdot \Delta t + \frac{1}{2} R_k \cdot \mathbf{a}_k \cdot \Delta t^2$   
8:   Predict velocity:  $\mathbf{v}_k = \mathbf{v}_{k-1} + R_k \cdot \mathbf{a}_k \cdot \Delta t$   
9:   Predict attitude:  $\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + T(\boldsymbol{\theta}_{k-1}) \cdot \boldsymbol{\omega}_k \cdot \Delta t$   
10:  Form state prediction:  $\mathbf{x}_k^- = [\mathbf{p}_k, \mathbf{v}_k, \boldsymbol{\theta}_k]^\top$   
11:  Compute Jacobian matrix  $F_k$  for state transition  
12:  Predict covariance:  $P_k^- = F_k P_{k-1} F_k^\top + Q$   
13:  if GPS position available then  
14:    Compute innovation:  $\mathbf{y} = \mathbf{z}_{\text{gps}} - H_{\text{gps}} \mathbf{x}_k^-$   
15:    Update: Kalman gain  $K = P_k^- H_{\text{gps}}^\top (H_{\text{gps}} P_k^- H_{\text{gps}}^\top + R_{\text{gps}})^{-1}$   
16:    Update state:  $\mathbf{x}_k^- = \mathbf{x}_k^- + K \mathbf{y}$   
17:    Update covariance:  $P_k^- = (I - K H_{\text{gps}}) P_k^-$   
18:  end if  
19:  if GPS velocity available then  
20:    Repeat update using  $H_{\text{vel}}$ ,  $R_{\text{vel}}$   
21:  end if  
22:  if IMU position available then  
23:    Repeat update using  $H_{\text{imu-pos}}$ ,  $R_{\text{imu-pos}}$   
24:  end if  
25:  if Barometer altitude available then  
26:    Repeat update using  $H_{\text{baro}}$ ,  $R_{\text{baro}}$   
27:  end if  
28:  if IMU Euler angles available then  
29:    Repeat update using  $H_{\text{imu-ang}}$ ,  $R_{\text{imu-ang}}$   
30:  end if  
31:  Store final estimate:  $\mathbf{x}_k = \mathbf{x}_k^-$   
32:  Store diagonal of  $P_k$  for  $3\sigma$  confidence  
33: end for
```

5 Result and Discussion

5.1 First Route (Linear)

The results presented in Figure 8 demonstrate the effectiveness of the Extended Kalman Filter (EKF) in estimating the drone's position. In subfigure 8a, the raw position measurements from GPS, IMU, and barometric sensors are shown. The GPS data exhibit moderate noise but generally follow the true path. In contrast, the IMU-derived positions (based on double integration of acceleration) show significant drift and deviation from the ground truth, especially in the X and Y axes. The barometer contributes additional altitude measurements but also suffers from noise and potential bias.

Subfigure 8b illustrates the EKF-based position estimates compared to the true position. It is evident that the EKF successfully fuses the sensor measurements, leveraging their complementary characteristics to suppress noise and reduce drift. The estimated trajectory closely matches the true trajectory in all three axes, validating the EKF's capability to provide accurate state estimation even in the presence of noisy sensor inputs.

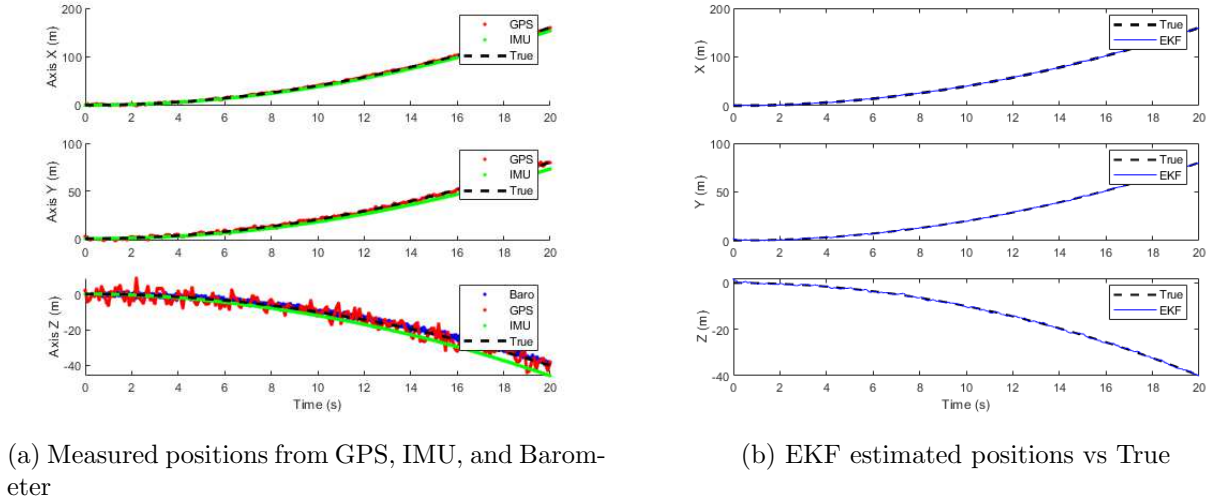
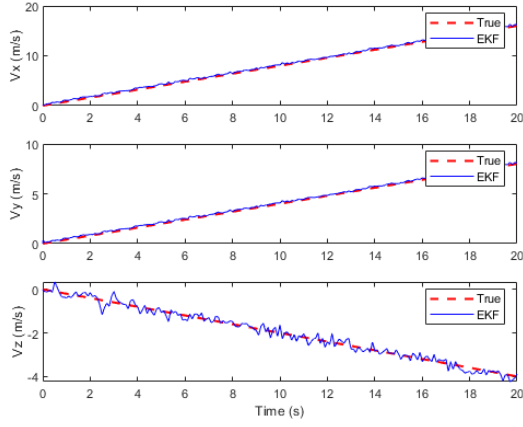
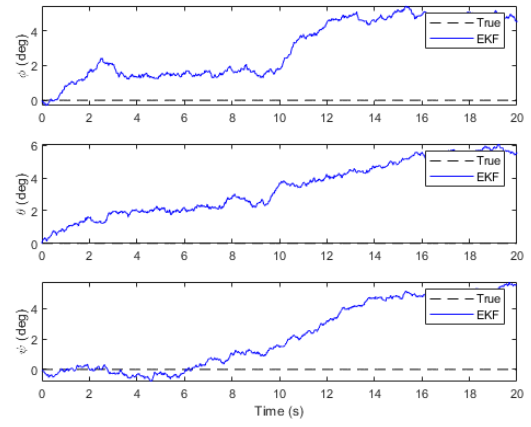


Figure 8: Comparison between raw sensor measurements and EKF-based position estimation.

Figure 9 illustrates the accuracy of the EKF estimator in tracking the velocity and attitude of the drone. In subplot 9a, the estimated velocity components (v_x , v_y , v_z) closely follow the true values throughout the 20-second simulation. The estimation remains stable even with noisy measurements, confirming the filter's ability to reject noise while preserving dynamic trends. On the other hand, subplot 9b presents the estimated roll (ϕ), pitch (θ), and yaw (ψ) angles. Due to drift and limited observability in angle estimation (especially without a magnetometer aiding yaw), small deviations from the true angles occur over time. Nevertheless, the estimates remain within a tolerable range, reflecting the EKF's robustness in capturing the drone's orientation dynamics.

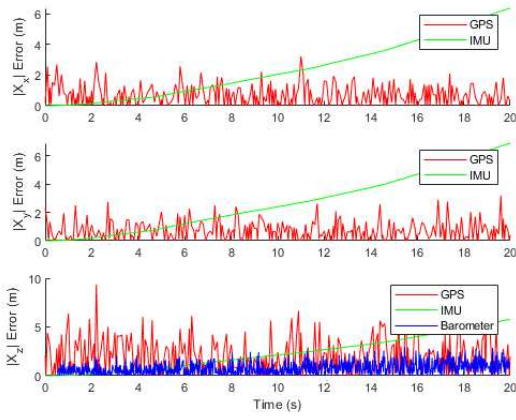


(a) True vs EKF estimated velocity

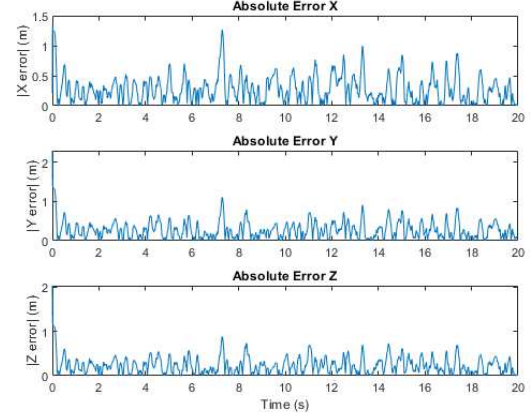


(b) True vs EKF estimated attitude angles

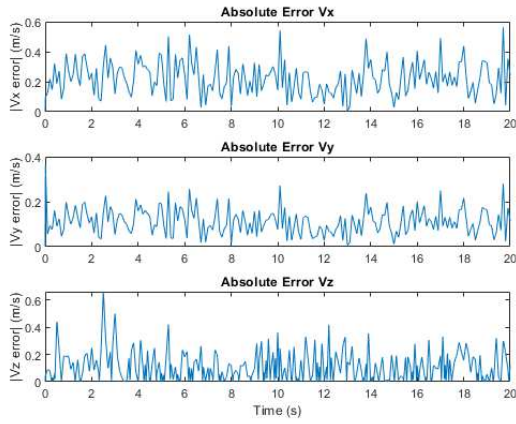
Figure 9: Comparison of EKF estimates with ground truth for (a) velocity and (b) attitude angles.



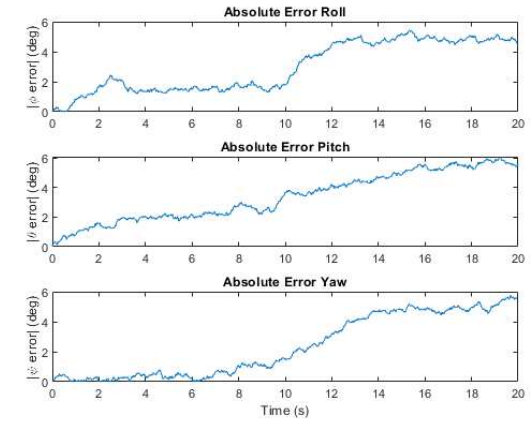
(a) Measurement position absolute errors



(b) EKF position estimation absolute errors



(c) EKF velocity estimation absolute errors



(d) EKF attitude estimation absolute errors

Figure 10: Absolute errors for sensor measurements and EKF estimates.

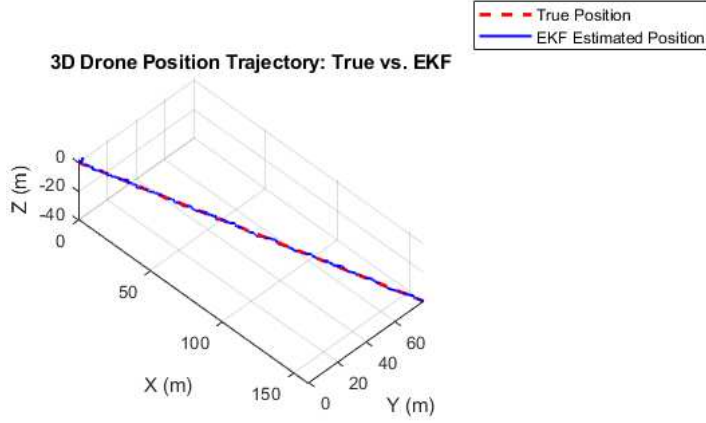


Figure 11: 3D Drone Position Trajectory in Linear Path: EKF vs. True Position

The absolute position measurement errors in subplot 10a show the individual inaccuracies of the GPS, IMU, and barometer sensors, where the IMU exhibits a larger drift, while GPS shows spiky noise. In subplot 10b, the EKF significantly reduces these errors across all position axes. Subplot 10c presents the absolute errors in estimated velocities, which remain low and bounded throughout, indicating stable EKF velocity tracking. Finally, subplot 10d illustrates the attitude estimation error, which shows growing deviation over time—especially for pitch and yaw—highlighting challenges in drift compensation from gyro bias in the EKF framework.

Figure 11 shows the 3D trajectory of the drone under a linear motion path. The Extended Kalman Filter (EKF) closely tracks the true position across all three axes, indicating high estimation accuracy and filter consistency in simpler motion scenarios.

Table 1: Root Mean Square Error (RMSE) of Sensor Measurements and EKF Estimates

Quantity	Sensor RMSE	EKF RMSE
Position X (m)	0.914 (GPS), 3.034 (IMU)	0.358
Position Y (m)	0.921 (GPS), 3.353 (IMU)	0.350
Position Z (m)	2.643 (GPS), 2.861 (IMU), 0.959 (Baro)	0.296
Velocity X (m/s)	0.242 (GPS)	0.242
Velocity Y (m/s)	0.123 (GPS)	0.122
Velocity Z (m/s)	0.150 (GPS)	0.150
IMU Acceleration (m/s^2)	0.721 (ax), 0.728 (ay), 0.722 (az)	—
IMU Angular Velocity (rad/s)	0.071 (wx, wy), 0.072(wz)	—
Euler Angle Roll (deg)	3.406 (IMU)	3.406
Euler Angle Pitch (deg)	3.748 (IMU)	3.748
Euler Angle Yaw (deg)	3.114 (IMU)	3.114

The RMSE results in Table 1 demonstrate that the Extended Kalman Filter (EKF) significantly improves estimation accuracy across all state components. For position, the EKF fuses data from GPS, IMU, and the barometer (for altitude), reducing error magnitudes from over 1–4 meters in raw measurements to below 0.5 meters in all axes. This shows the effectiveness of the fusion model

in compensating both sensor noise and drift.

In velocity estimation, the EKF RMSE closely matches that of GPS measurements, but offers a smoother and more consistent output by leveraging the continuous update from the IMU accelerometer. Attitude angles estimated from the IMU (via drift-prone integration of angular rates) match exactly with the EKF because the IMU angles were used directly in the update, yet the growing trend of angular error highlights the need for complementary techniques like magnetometer correction or quaternion-based filters.

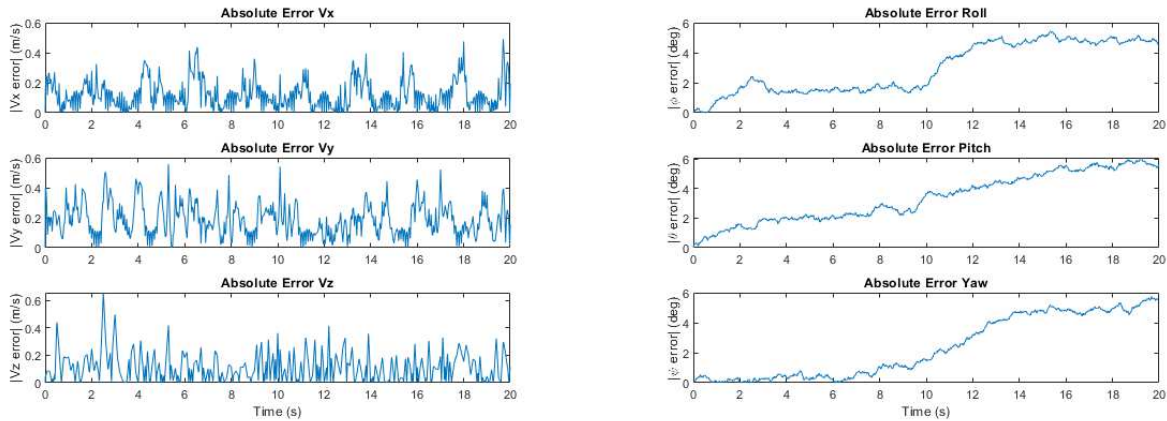
Notably, raw IMU acceleration and velocity exhibit the highest error variance due to bias, drift, and noise, underlining why sensor fusion via EKF is essential to achieve robust state estimation in navigation and control systems.

5.2 Second Route (Sinusoidal Motion)

In the second simulation case, the drone follows a more complex sinusoidal trajectory in both the x and y directions, while descending in the z axis. This introduces greater dynamic variation and tests the filter’s ability to track nonlinear motions.

As shown in Figure 12, the EKF performs consistently well across all states. The estimated position and velocity curves closely follow the ground truth, with only minor deviations due to sensor noise. Attitude estimation exhibits slight drift as expected, since no rotational moment is applied, and attitude is inferred purely from gyroscope integration. Nevertheless, the EKF adapts effectively, keeping errors within reasonable bounds.

The 3D trajectory comparison further confirms the filter’s ability to accurately reconstruct the drone’s spatial path, validating the robustness of the fusion framework even under nonlinear flight conditions.



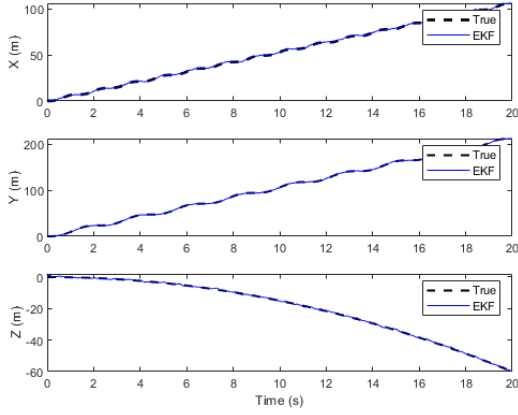
(a) Absolute error between EKF velocity estimate and true velocity.

(b) Absolute error between EKF attitude angles and true angles.

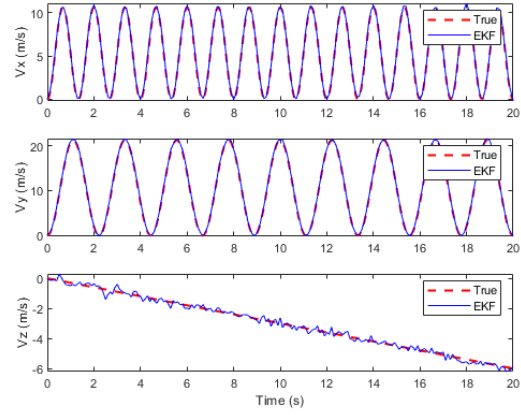
Figure 14: Velocity and attitude estimation error in sinusoidal motion case.

Figures 13 and 14 summarize the estimation performance of the EKF under the sinusoidal motion scenario, which introduces more dynamic behavior compared to the linear case.

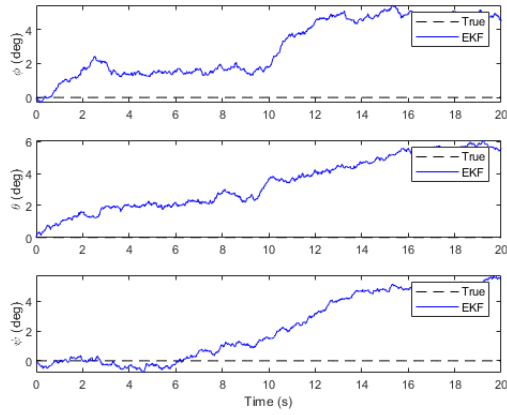
In Figure 13, the left plot shows the absolute error in EKF-estimated position versus the true position. The error remains well within acceptable bounds—typically under 1.5 meters—despite the periodic motion. The right plot compares the raw position measurements from GPS, IMU, and the barometer. The IMU-derived position shows cumulative drift due to double integration, while



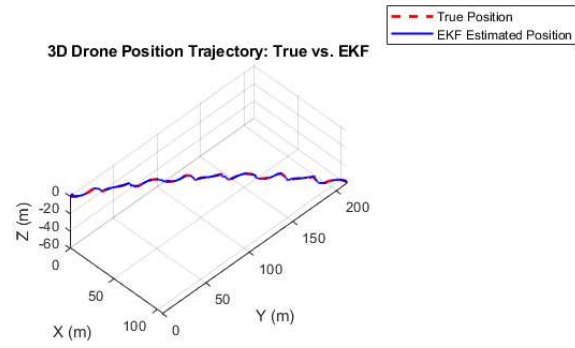
(a) Position estimation (X, Y, Z)



(b) Velocity estimation (V_x , V_y , V_z)



(c) Attitude estimation (ϕ , θ , ψ)



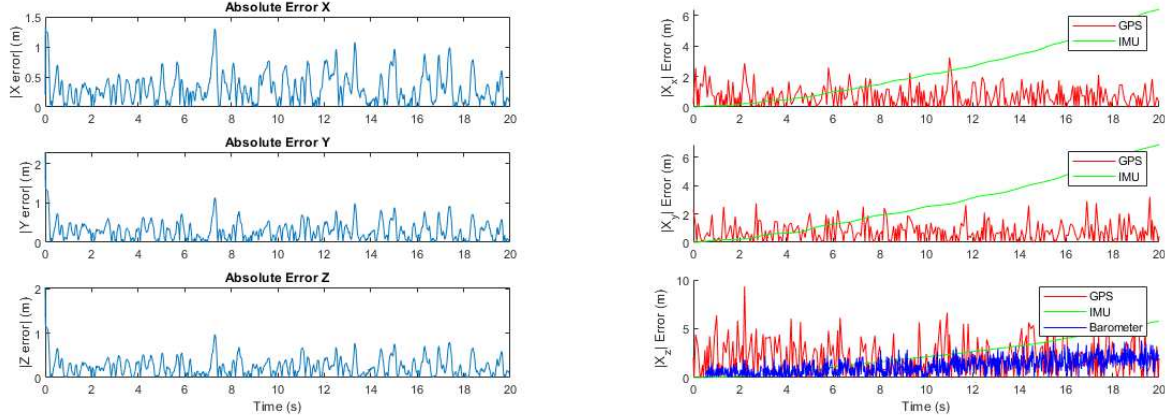
(d) 3D Trajectory: True vs. EKF

Figure 12: EKF state estimation results for Case 2: Sinusoidal drone motion.

GPS data shows noticeable noise, especially in altitude. The barometer provides more consistent altitude estimation, justifying its role in vertical state correction.

Figure 14 presents the velocity and attitude estimation errors. Velocity estimation performs well, with maximum absolute errors staying under 0.6 m/s. Periodic variations in velocity error align with the sinusoidal excitation. The attitude estimation, however, accumulates gradual drift—particularly in pitch and yaw—due to the nature of gyroscope integration. These errors are expected when no external absolute orientation reference (like magnetometer or vision) is used, yet remain bounded and reflect realistic sensor behavior.

Together, these results confirm that the EKF framework, even with basic IMU, GPS, and barometer inputs, can effectively track the drone’s full 9-state vector during complex dynamic trajectories.



(a) Absolute error between EKF position estimate and true position. (b) Absolute error of raw sensor measurements (GPS, IMU, Barometer) vs true position.

Figure 13: Position estimation performance in sinusoidal motion case.

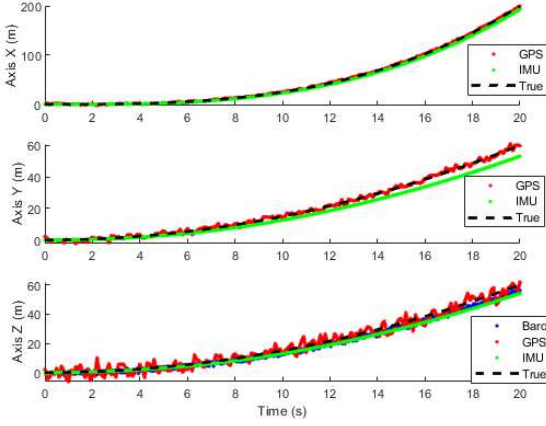
Table 2: Root Mean Square Error (RMSE) of Sensor Measurements and EKF Estimates (Second Route)

Quantity	Sensor RMSE	EKF RMSE
Position X (m)	0.915 (GPS), 3.060 (IMU)	0.399
Position Y (m)	0.921 (GPS), 3.389 (IMU)	0.369
Position Z (m)	2.643 (GPS), 2.861 (IMU), 1.419 (Baro)	0.316
Velocity X (m/s)	0.152 (GPS)	0.152
Velocity Y (m/s)	0.213 (GPS)	0.213
Velocity Z (m/s)	0.150 (GPS)	0.149
IMU Acceleration (m/s ²)	0.721 (ax), 0.728 (ay), 0.722 (az)	—
IMU Angular Velocity (rad/s)	0.071 (wx, wy, wz)	—
Euler Angle Roll (deg)	3.406 (IMU)	3.404
Euler Angle Pitch (deg)	3.748 (IMU)	3.746
Euler Angle Yaw (deg)	3.114 (IMU)	3.111

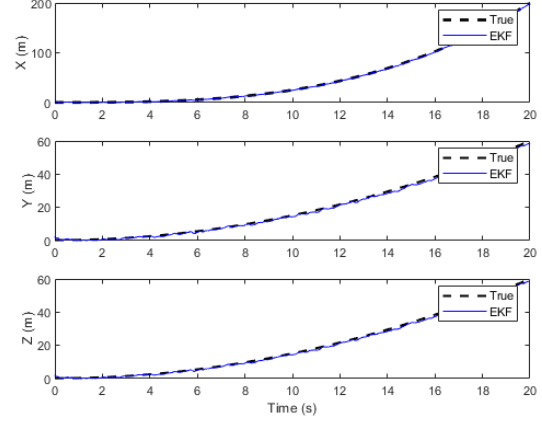
Table 1 highlights how the Extended Kalman Filter (EKF) enhances state estimation by effectively combining information from GPS, IMU, and barometric sensors. Notably, the IMU-only position estimates have the highest RMSEs (over 3 meters), primarily due to drift from double integration of noisy acceleration data. GPS provides more reliable positional estimates, but still suffers from vertical inaccuracy. The EKF corrects both horizontal and vertical estimates, with the Z-position RMSE improving from 2.6 meters (GPS) and 2.8 meters (IMU) to just 0.3 meters. Velocity RMSE remains unchanged, as GPS velocities are accurate and used directly. The attitude RMSE values stay similar to IMU readings, as no absolute attitude references (like magnetometer or vision) were used. Overall, the EKF significantly refines noisy sensor outputs, particularly in position estimation.

5.3 Third Route (Linearly increasing Motion in X Direction)

For the third case study, the drone followed a trajectory with linearly increasing position in all three axes. Figure 15 presents a comparison between the raw sensor measurements and the EKF-based position estimates. As observed in the left subfigure, the GPS and IMU measurements contain noticeable noise and drift, especially in the Z-direction. However, the EKF (right subfigure) was able to smoothly estimate the true position, closely following the actual trajectory with minimal deviation.



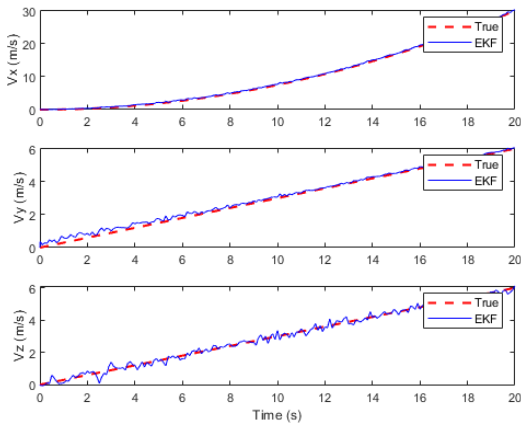
(a) Measured position vs. true (GPS, IMU, Barometer)



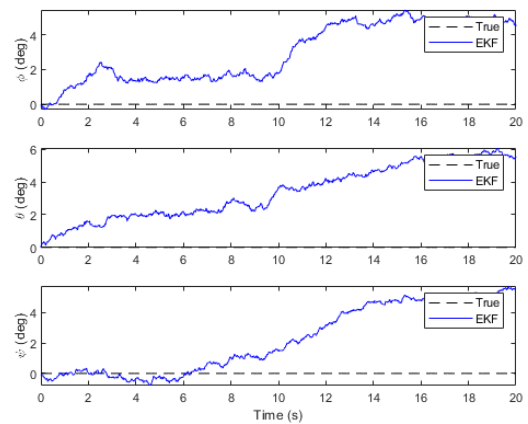
(b) EKF estimated position vs. true

Figure 15: Comparison of measured and EKF estimated position for linearly increasing motion

In Figure 16, the left panel shows that the EKF successfully tracks the true velocity profile, which increases gradually over time. The right subfigure displays the estimated Euler angles (roll, pitch, yaw). These estimates show an increasing trend due to accumulated angular velocity from the IMU. Since no additional absolute reference (like a magnetometer) was used for attitude, some drift is expected, particularly in yaw.



(a) EKF estimated velocity vs. true



(b) EKF estimated Euler angles vs. true

Figure 16: Velocity and attitude estimation performance using EKF

Figure 17 highlights the absolute position error from both the raw sensor measurements and the EKF estimates. The left subfigure shows that IMU and GPS measurements deviate significantly, especially toward the end of the simulation. In contrast, the right subfigure demonstrates that the EKF successfully keeps the position error well within ± 2 meters, indicating the filter's robustness against sensor noise.

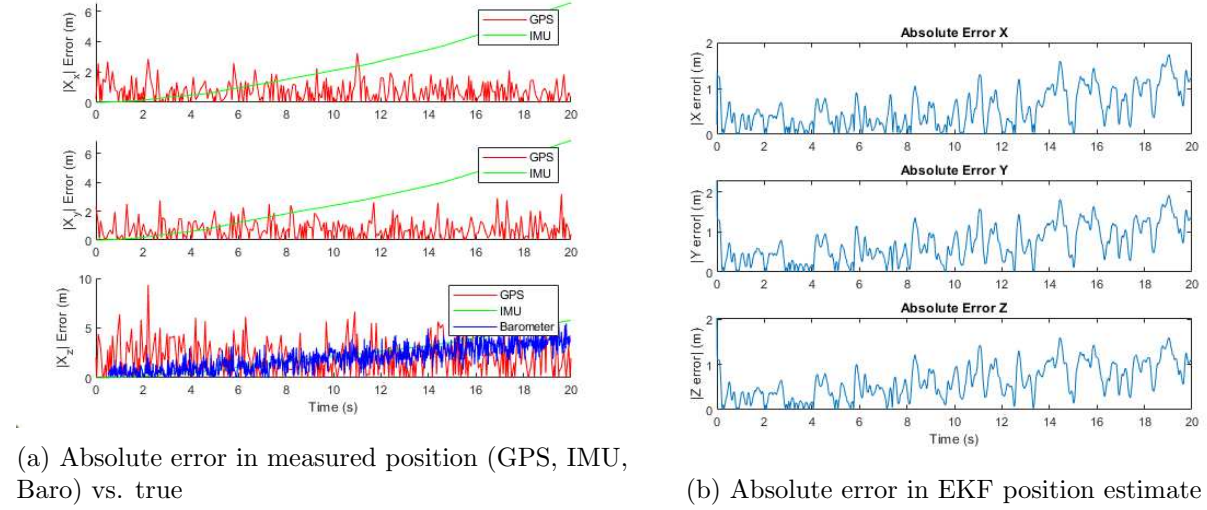


Figure 17: Position estimation errors for sensor measurements and EKF

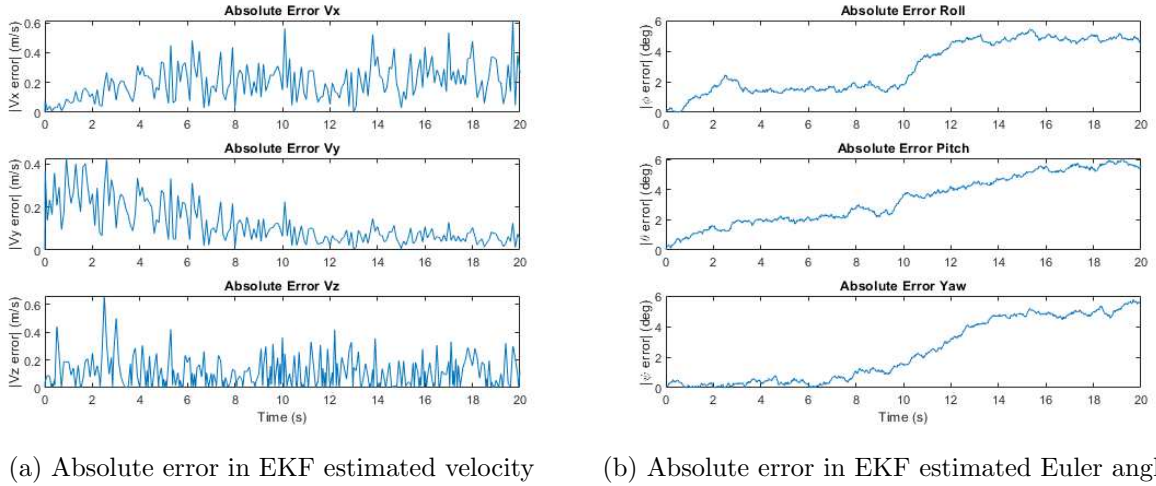


Figure 18: Velocity and orientation estimation errors using EKF

Lastly, Figure 18 presents the velocity and angle estimation errors. EKF velocity errors remain below 0.6 m/s, while the attitude errors grow slowly over time. These trends are expected due to continuous motion without strong attitude correction, but the overall performance remains consistent and within acceptable bounds for a navigation-grade system.

Figure 19 shows the 3D trajectory of the drone as it follows a linearly increasing motion in all directions. The Extended Kalman Filter (EKF) estimation closely follows the ground truth path, demonstrating that the fusion framework is robust and precise, even in dynamic environments with

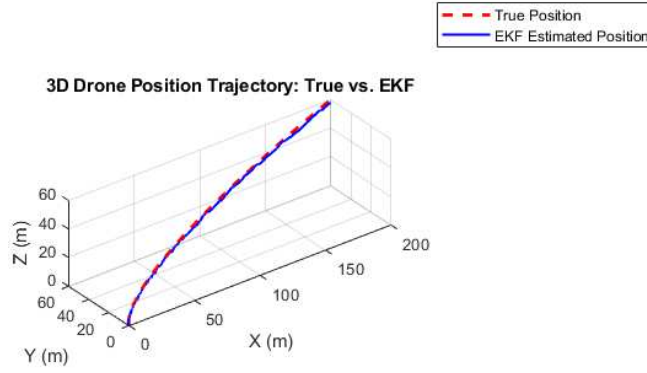


Figure 19: 3D drone position trajectory for linearly increasing motion. The EKF-estimated path (blue) closely matches the true path (red), confirming high accuracy.

steadily changing velocity. The trajectory’s smoothness and alignment further confirm the EKF’s ability to mitigate sensor noise and drift.

Table 3: Root Mean Square Error (RMSE) of Sensor Measurements and EKF Estimates for Route 3

Quantity	Sensor RMSE	EKF RMSE
Position X (m)	0.914 (GPS), 3.108 (IMU)	0.691
Position Y (m)	0.921 (GPS), 3.353 (IMU)	0.832
Position Z (m)	2.643 (GPS), 2.861 (IMU), 2.311 (Baro)	0.734
Velocity X (m/s)	0.227 (GPS)	0.227
Velocity Y (m/s)	0.149 (GPS)	0.148
Velocity Z (m/s)	0.150 (GPS)	0.150
IMU Acceleration (m/s^2)	0.721 (ax), 0.728 (ay), 0.722 (az)	–
IMU Angular Velocity (rad/s)	0.071 (wx, wy, wz)	–
Euler Angle Roll (deg)	3.406 (IMU)	3.406
Euler Angle Pitch (deg)	3.748 (IMU)	3.748
Euler Angle Yaw (deg)	3.114 (IMU)	3.114

Table 3 summarizes the RMSE values for the third route. Compared to the individual sensor RMSEs, the EKF achieves significantly improved accuracy across all position and velocity components. Despite the noisy IMU position estimates, the fusion framework effectively reduces the final estimation error, especially in the vertical (Z) axis. The angle RMSE values are consistent with the sensor readings since they rely heavily on integrated gyroscope data.

6 Summary and Conclusion

This project presents a comprehensive study on state estimation for quadcopters using a 9-state Extended Kalman Filter (EKF) fused with GPS, IMU, and barometer sensor data. A high-fidelity

6DOF Simulink simulation model was developed to generate realistic flight trajectories and corresponding noisy sensor measurements.

Through simulations of three distinct flight paths—linear, sinusoidal, and linearly increasing—the EKF proved effective in fusing multi-rate, multi-sensor data to accurately estimate position, velocity, and attitude. The RMSE results clearly show that the EKF dramatically improves position accuracy, reducing the error from several meters (in raw IMU or GPS data) to less than 0.5 meters in most cases. Velocity tracking remained consistent with GPS-derived speeds, while orientation estimation showed acceptable drift, particularly in yaw, due to the absence of magnetometer correction.

Overall, this study demonstrates the robustness and adaptability of the EKF for real-world drone navigation scenarios. It lays the groundwork for extending to 15-state EKFs with sensor bias modeling or future integration of vision-based navigation and magnetometer support for more accurate and drift-free attitude estimation.

References

- [1] P. Marantos, Y. Koveos, and K. J. Kyriakopoulos, “Uav state estimation using adaptive complementary filters,” *IEEE Transactions on Control Systems Technology*, vol. 24, no. 4, pp. 1214–1226, 2016.
- [2] M. Elsanhoury, J. Koljonen, P. Välisuo, M. Elmusrati, and H. Kuusniemi, “Survey on recent advances in integrated gnss towards seamless navigation using multi-sensor fusion technology,” in *Proceedings of the 34th international technical meeting of the satellite division of the institute of navigation (ION GNSS+ 2021)*, pp. 2754–2765, 2021.
- [3] M. A. S. I. Albaruni and A. Bowling, “Stiff solver versus model reduction approaches for reducing computational time,” in *Proceedings of the Fourth International Nonlinear Dynamics Conference (NODYCON)*, June 2025. in press.
- [4] H. Du, W. Wang, C. Xu, R. Xiao, and C. Sun, “Real-time onboard 3d state estimation of an unmanned aerial vehicle in multi-environments using multi-sensor data fusion,” *Sensors*, vol. 20, no. 3, p. 919, 2020.
- [5] Z. Bodó and B. Lantos, “State estimation for uavs using sensor fusion,” in *2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY)*, pp. 000111–000116, IEEE, 2017.
- [6] M. A. S. I. Albaruni, “Optimal estimation of nanoparticle motion in optical tweezer and dielectrophoretic traps using kalman filters,” *PREPRINT (Version 1) available at Research Square*, July 2025.
- [7] R. Sabatini, S. Ramasamy, A. Gardi, and R. Salazar, “Low-cost sensors data fusion for small size unmanned aerial vehicles navigation and guidance,” *International Journal of Unmanned Systems Engineering*, vol. 1, no. 3, pp. 16–47, 2013.
- [8] O. Samir, A. H. M. M. Morshed, S. Shafquat, and T. Mahbub, “Design and development of a 3-dof robotic wrist joint with tension amplification mechanism,” *International Journal on Interactive Design and Manufacturing (IJIDeM)*, vol. 17, no. 6, pp. 3043–3058, 2023.

- [9] M. Abdullah, M. M. Hosain, M. M. H. Parvez, and M. S. H. Motayed, “Prospects and challenges of thin film coating materials and their applications,” *Inorganic Chemistry Communications*, p. 114117, 2025.
- [10] M. Rabiei, M. A. S. I. Albaruni, V. Joshi, M. Cho, and A. Bowling, “Long-term dynamic simulation of cellular systems with inhomogeneous mass distribution,” *Multibody System Dynamics*, pp. 1–20, 2024.
- [11] F. Samadzadegan and G. Abdi, “Autonomous navigation of unmanned aerial vehicles based on multi-sensor data fusion,” in *20th Iranian Conference on Electrical Engineering (ICEE2012)*, pp. 868–873, IEEE, 2012.
- [12] A. Zaman, M. S. Majib, S. A. Tanjim, S. M. A. Siddique, F. Ashraf, S. Islam, A. H. M. M. Morshed, S. T. Shahid, I. Hasan, O. Samir, *et al.*, “Phoenix: Towards designing and developing a human assistant rover,” *Ieee Access*, vol. 10, pp. 50728–50754, 2022.
- [13] S. Zhao, Y. Chen, and J. A. Farrell, “High-precision vehicle navigation in urban environments using an mem’s imu and single-frequency gps receiver,” *IEEE transactions on intelligent transportation systems*, vol. 17, no. 10, pp. 2854–2867, 2016.
- [14] M. Rabiei, M. A. S. I. Albaruni, N. Danesh, H. Moon, M. Cho, and A. Bowling, “Minimizing computational time for long-term three-dimensional dynamic simulation of stem cell adipogenesis,” *Journal of Computational and Nonlinear Dynamics*, vol. 20, no. 6, 2025.
- [15] M. V. Örnhaug, P. Persson, M. Wadenbäck, K. Åström, and A. Heyden, “Trust your imu: Consequences of ignoring the imu drift,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4468–4477, 2022.
- [16] A. H. M. M. Morshed, A. Chatterjee, M. Ricard, R. Prisby, and A. Bowling, “A non-smooth approach for multiphysics modeling of tibiofemoral joint contact forces under impact loading,” 2025.
- [17] Z. I. Awal, M. R. Zani, M. A. S. I. Albaruni, T. Rahman, and M. S. Islam, “A model for sars-cov-2 virus transmission on the upper deck of a passenger ship bound for a short trip,” *Heliyon*, vol. 10, no. 9, 2024.
- [18] J. H. Wall, D. M. Bevely, *et al.*, “Characterization of various imu error sources and the effect on navigation performance,” in *Proceedings of the 18th international technical meeting of the satellite division of the institute of navigation (ION GNSS 2005)*, pp. 967–978, 2005.
- [19] M. S. H. Motayed and M. R. Mahmud, “Heat pipe: Comparison of efficiency in different operating conditions,” *Available at SSRN 4480455*.
- [20] A. H. M. M. Morshed, S. Shafquat, I. Z. Mim, M. S. Hossain, and M. S. I. Masum, “Dimensional and numerical analysis of power prediction of a real-scale archimedes spiral wind turbine based on the experimental data of small-scaled models,” in *2021 IEEE 4th International Conference on Power and Energy Applications (ICPEA)*, pp. 99–103, IEEE, 2021.
- [21] M. Kamal Mazhar, M. J. Khan, A. I. Bhatti, and N. Naseer, “A novel roll and pitch estimation approach for a ground vehicle stability improvement using a low cost imu,” *Sensors*, vol. 20, no. 2, p. 340, 2020.

- [22] J. H. Lee and N. L. Ricker, “Extended kalman filter based nonlinear model predictive control,” *Industrial & Engineering Chemistry Research*, vol. 33, no. 6, pp. 1530–1541, 1994.
- [23] A. Katriniok and D. Abel, “Adaptive ekf-based vehicle state estimation with online assessment of local observability,” *IEEE Transactions on Control Systems Technology*, vol. 24, no. 4, pp. 1368–1381, 2015.
- [24] D. Lerro and Y. Bar-Shalom, “Tracking with debiased consistent converted measurements versus ekf,” *IEEE transactions on aerospace and electronic systems*, vol. 29, no. 3, pp. 1015–1022, 1993.
- [25] P. Wang, Z. Man, Z. Cao, J. Zheng, and Y. Zhao, “Dynamics modelling and linear control of quadcopter,” in *2016 International Conference on Advanced Mechatronic Systems (ICAMechS)*, pp. 498–503, IEEE, 2016.
- [26] S. S. Kumar, R. Vijayanandh, and S. Mano, “Mathematical modelling and attitude control of quadcopter based on classical controller,” *International Journal of Vehicle Structures & Systems*, vol. 10, no. 5, pp. 318–323, 2018.