Attitude Estimation of Quadcopter Drone

Mike Timmerman
Aerospace Engineering
Stanford University
Stanford, USA
mtimmerm@stanford.edu

Irene Yeu
Aeronautics and Astronautics
Stanford University
Stanford, USA
iyeu@stanford.edu

Richard Qiu
Mechanical Engineering
Stanford University
Stanford, USA
rwqiu@stanford.edu

***Abstract*—This paper analyzes attitude estimation techniques for drones, focusing on the Multiplicative Kalman Filter (MKF), Unscented Kalman Filter (UKF), and Particle Filter (PF). Measurement data from a 9-axis Inertial Measurement Unit (IMU) with accelerometer, magnetometer, and gyroscope readings are modeled with Gaussian noise. The MKF uses an axis-angle representation for quaternion updates, addressing attitude dynamics nonlinearity. The UKF captures state distribution with sigma points for robust nonlinear estimation. The PF represents the state distribution with weighted particles, handling non-Gaussian distributions and complex dynamics. Simulations with waypoint-generated drone trajectories evaluate these filters' performance, demonstrating their effectiveness and highlighting their comparative performance in attitude estimation.**

I. Introduction

Drones have become indispensable across various domains due to their versatility and accessibility [8]. In agriculture, they aid in precision farming through crop monitoring and aerial spraying, optimizing resource use and enhancing yields. In construction and infrastructure inspection, drones provide cost-effective solutions for assessing structural integrity and conducting visual inspections, improving safety and efficiency in maintenance activities. They also play a crucial role in search and rescue operations by navigating hazardous environments and providing real-time situational awareness to rescue teams.

These domain applications rely on the knowledge of where in space the drone is located. This is the field of state estimation, where the problem is to estimate the state in which a dynamical system currently finds itself. For drones specifically, this consists of estimating their position and orientation with respect to an inertial frame. Drones are also increasingly being used autonomous operations in which case an accurate knowledge of the drone's current state is especially important for enabling drones to navigate safely without human intervention.

While drones rely on sensor measurements to gain knowledge about their current state, these alone are often not sufficient to provide an accurate estimate since sensor measurements, such as those from accelerometers, gyroscopes, and GPS receivers, are often corrupted by noise and biases inherent in the sensor hardware or environmental factors. Without filtering and processing, these noisy measurements can lead to inaccurate state estimates. Furthermore, drones typically rely on multiple sensors to estimate their state accurately. Each sensor provides different types of information, such as position, velocity, and acceleration, with varying levels of accuracy and precision. Sensor fusion techniques are necessary to integrate data from multiple sensors and extract meaningful information about the drone's state while compensating for the limitations of individual sensors.

Given these aspects, an efficient algorithm that filters and fuses the different data streams while also effectively being able to capture the inherently nonlinear dynamics of drone motion necessitates nonlinear state estimation techniques. This led to the development of the Extended Kalman Filter, Unscented Kalman Filter, Particle Filter, and many variations. This project focuses on the implementation of different filtering techniques for the purpose of estimating the drone's attitude.

The paper is structured as follows. A comprehensive literature review is provided in section II, with background on the models utilized in section III. The approach and different algorithms utilized are specified in section IV. The results are then presented in section V. In section VI, the results are analyzed, and future work is proposed.

II. Literature Review

The vast amount of literature on state estimation and filtering occurred after the publication of the original Kalman Filter paper [5]. The Kalman filter provides a powerful solution to the problem of state estimation in dynamic systems. Originally developed by Rudolf E. Kálmán in the 1960s, this recursive algorithm enables optimal estimation of the state of a linear dynamic system in the presence of noisy measurements. Its ability to fuse information from multiple sensors, account for process and measurement noise, and provide accurate and efficient state estimates has had a profound impact across diverse domains, including aerospace, robotics, navigation, finance, and signal processing. The Kalman Filter's simplicity, effectiveness, and versatility have made it a cornerstone of modern estimation theory, shaping the way researchers and engineers approach problems involving state estimation, control, and prediction.

While the Kalman Filter is a powerful tool for state estimation, it has limitations that can impact its performance in real-world applications. The main limitation is its reliance on linear dynamics and Gaussian noise assumptions. Consequently, there's been a drive to develop methods which can handle nonlinear dynamical systems. Given that drone dynamics inherently exhibit nonlinearity, state estimation for drones heavily relies on these advanced methods.

The Extended Kalman Filter (EKF) and the Unscented Kalman Filter (UKF) are two variants of the Kalman Filter that are better suited to deal with such nonlinearities. Unlike the KF, EKFs utilize nonlinear state transition and dynamics models to better represent the ground truth. Then, the functions are linearized around the estimate at each time step, and the traditional KF equations can be used [2]. This process can introduce errors in the posterior mean and covariance, which can in turn lead to poor performance and filter divergence. These issues were addressed with the development of the UKF, which utilizes deterministic sampling to calculate a more accurate posterior mean and covariance, and propagates them through a nonlinear system [9]. The UKF has a higher performance than the EKF, capturing the posterior mean and covariance accurately to the third order, compared to the first order offered by the EKF. In particular, UKFs outperform EKFs in terms of accuracy and consistency in highly nonlinear systems. However, this accuracy comes at the price of higher computational demands. Da Silva and Natássya [1] compare the performances of the UKF and EKF specifically in the context of sensor fusion for UAVs. They found that the UKF consistently outperformed the EKF, but was slightly more computationally expensive due to the propagation of multiple sigma points rather than one state vector.

The UKF operates as follows [6]. First, sigma points are chosen around the mean state estimated from the previous time step in order to more accurately capture the mean and covariance of the state distribution in the presence of nonlinearities. Then, each sigma point is propagated through the process model to obtain the predicted sigma points and the weight of the sigma points, which are then utilized to predict the state mean and covariance in the time update. In the update step, the cross-covariance and Kalman gain are computed to update the state estimate and covariance.

The particle filter (PF) is a Monte Carlo filtering method that represents a target probability distribution using a weighted set of random sample points. The PF algorithm iteratively updates particle weights based on measurements and resamples points to construct an unknown posterior distribution. As the PF does not make any assumptions about state space dynamics or underlying distributions, it can be utilized with nonlinear and non-Gaussian systems [7] which is especially relevant for nonlinear attitude estimation. A quaternion particle filter method for attitude estimation of inertial measurement units is presented in [11]. In particular, the authors introduce a discrete quaternion particle filter which uses a discrete deterministic sampling method to generate quaternion particles. They present simulations comparing results against a UKF and random sampling PF and conclude that the deterministic sampling PF has similar or better accuracy along with lower computational cost. A standard particle filter can also be combined with optimal control theory to produce a feedback particle filter where particles propagate based on a control law to avoid resampling. An application of the feedback particle filter for attitude estimation is presented in [10] along with numerical methods of finding control gains and a comparison with other nonlinear filters. The authors conclude that the feedback PF has superior performance compared to other filtering methods given a large initial uncertainty in the system.

The Multiplicative Kalman Filter (MKF) is another further development of the Kalman Filter. It operates by representing the evolution of states and measurement updates as multiplicative rather than additive processes. This feature renders it particularly effective in scenarios where states evolve multiplicatively, as seen in attitude dynamics. By capturing nonlinearities in the system through multiplicative transformations in both state transition and measurement updates, the MKF provides a more precise representation of system dynamics, resulting in enhanced estimation performance for nonlinear systems. [4] provides a comparative analysis between the Additive and Multiplicative Kalman filter for satellite attitude determination. The authors conclude that the estimation using the MKF has a smaller absolute error and lower uncertainty. This is due in part to state representations such as the quaternion and Euler angles not being in the vector group, hence are not closed under vector addition, i.e. adding a quaternion to another quaternion does not yield a quaternion. [3] presents a quaternion attitude estimation for a miniature air vehicle using the MKF. In their methodology, they use an Euler attitude error vector as the state vector. This rotation parameterization can be readily converted to a unity-norm error quaternion. In the time update, the attitude quaternion is simply propagated using the angular velocity measurements from the gyroscope, while the error vector is reset to zero. For the measurement update, they use the gravity vector measurement from the accelerometer. They provide a methodology to update the error vector from these measurements, after which the corresponding unity-norm error quaternion is obtained which is used to update the overall attitude quaternion. With their approach, they showed the effectiveness of estimating attitude through a full range of attitudes through loop maneuvers.

III. PRELIMINARIES

A. Drone Model

For the drone model, six-degree-of-freedom dynamics are considered, which consist of 12 states: position p, linear velocity v, attitude η, and angular velocity Ω, and 4 inputs: four rotational speeds of the propellers ω_i, $i \in \{1, \ldots, 4\}$. The angular velocity consists of the components roll rate p, pitch rate q and yaw rate r, and the attitude consists of the components roll ϕ, pitch θ and yaw ψ.

The drone's translational dynamics are derived from Newton's third law:

$$m\mathbf{a} = F_{ext} = D + T + G \tag{1}$$

where m is the drone's mass, **a** is the linear acceleration, and F_{ext} is the externally applied force. The linear acceleration is modeled to include contributions from atmospheric drag D, thrust force T generated by the propellers, and gravitational force G.

The rotational dynamics are derived from Euler's rotation equations:

$$I\dot{\Omega} + \Omega \times (I\Omega) = M_{ext} = M_{prop} \tag{2}$$

where I is the inertia tensor of the drone and M_{ext} is the externally applied moments. The moments are modeled as being induced by the rotating propellers.

Process noise is added to the dynamics through the propeller torques. A small noise term is added to the thrust produced by the propellers. The noise is taken to be Gaussian with a standard deviation of $1 \times 10^{-2}[N]$.

B. Measurement Model

For the purpose of attitude estimation, the drone relies on measurements from a 3-axis accelerometer, 3-axis magnetometer, and 3-axis gyroscope, which can be integrated into a single $9-axis$ Inertial Measurement Unit.

The accelerometer measurements are modeled as follows:

$$\tilde{\boldsymbol{f}}^b = \boldsymbol{a}^b + R_i^b \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + \boldsymbol{\nu}_{acc} \tag{3}$$

where $\boldsymbol{a}^b$ is the drone's acceleration in body axes frame. R_i^b is the rotation matrix from inertial to body frame, which rotates the gravitational acceleration to body frame. ν_{acc} is the accelerometer noise which is taken to be distributed as $\nu_{acc} \sim \mathcal{N}(0, R_{acc})$, with $R_{acc} = 1 \times 10^{-2}\mathbb{I}_3$. Hence, it is assumed that measurements have zero bias.

The magnetometer measurements are modeled as follows:

$$\tilde{\boldsymbol{r}}^b_{mag} = \hat{\boldsymbol{r}}^b_{mag} + \boldsymbol{\nu}_{mag}$$

where $\hat{\boldsymbol{r}}^b_{mag}$ is the unit vector in body axes frame indicating the magnetic field direction and ν_{mag} is the gyroscope noise which is taken to be distributed as $\nu_{mag} \sim \mathcal{N}(0, R_{mag})$, with $R_{mag} = 1 \times 10^{-2}\mathbb{I}_3$. Hence, it is assumed that measurements have zero bias.

The gyroscope measurements are modeled as follows:

$$\tilde{\boldsymbol{\omega}}^b = \boldsymbol{\omega}^b + \boldsymbol{\nu}_{gyro} \tag{4}$$

where $\boldsymbol{\omega}^b$ is the drone's true angular velocity in body axes frame. ν_{gyro} is the gyroscope noise which is taken to be distributed as $\nu_{gyro} \sim \mathcal{N}(\mu_{gyro}, R_{gyro})$, with $R_{gyro} = 1 \times 10^{-2}\mathbb{I}_3$ and $\mu_{gryo} = 1 \times 10^{-3}[rad]$. This bias is consistent with the fact that cheap gyroscopes used on drones are often inaccurate and not well calibrated, causing attitude estimation through integration of angular velocity measurements drift over time.

C. Kalman Filter

The state estimations presented in this paper are recursive filters to some extent based on the Kalman Filter. Therefore, this section briefly introduces the Kalman Filter state estimation process. The Kalman Filter consists of a prediction step in which the state is propagated through the system dynamics. This is followed by an update step in which measurement knowledge is integrated into the estimate. This leads to a trade-off between process noise and measurement noise. The algorithm is given in Algorithm 1.

Algorithm 1 Kalman Filter Algorithm

Input: At $t = 0 \quad \mathbf{X}_0 \sim \mathcal{N}(\mu_{0|0}, \Sigma_{0|0})$
for $t \in 1, 2, \ldots, n$ **do**
  Prediction Step:
  $\mu_{t|t-1} = A_{t-1}\mu_{t-1|t-1} + B_{t-1}u_{t-1}$
  $\Sigma_{t|t-1} = A_{t-1}\Sigma_{t-1|t-1}A_{t-1}^T + Q_{t-1}$
  Update Step:
  Observe measurement y_t
  $K_t = \Sigma_{t|t-1}C_t^T(C_t\Sigma_{t|t-1}C_t^T + R_t)^{-1}$
  $\mu_{t|t} = \mu_{t|t-1} + K_t(y_t - C_t\mu_{t|t-1})$
  $\Sigma_{t|t} = \Sigma_{t|t-1} - K_tC_t\Sigma_{t|t-1}$
end for
Output: State Estimate $\mathbf{X}_n \sim \mathcal{N}(\mu_{n|n}, \Sigma_{n|n})$

IV. Approach

A. Multiplicative Kalman Filter

The Multiplicative Kalman Filter (MKF) is an adaptation of the Extended Kalman Filter tailored for attitude estimation. Attitude estimation is naturally represented using quaternions or Euler angles, which are inherently nonlinear. Additionally, these representations do not form a vector space, meaning they are not closed under addition. Specifically, the sum of two quaternions does not result in a valid quaternion.

In the MKF, the measurement residual, error between actual and predicted measurements, is represented using an error axis-angle representation. Using generic notation, measured and predicted vectors are denoted as $\boldsymbol{v}_m$ and $\boldsymbol{v}_p$. The cross-product of normalized measurement and predicted vectors

$$\boldsymbol{n}_{err} = \frac{\boldsymbol{v_p}}{||\boldsymbol{v}_p||} \times \frac{\boldsymbol{v_m}}{||\boldsymbol{v}_m||} \tag{5}$$

yields the unit-vector axis around which the attitude estimate needs to be rotated to correct the error. The angle over which needs to be rotated is calculated as follows:

$$\theta_{err} = cos^{-1}\left(\frac{\boldsymbol{v_p}}{||\boldsymbol{v}_p||} \cdot \frac{\boldsymbol{v_m}}{||\boldsymbol{v}_m||}\right) \tag{6}$$

The error axis-angle is then simply $\boldsymbol{r}_{err} = \theta_{err}\boldsymbol{n}_{err}$. This quantity is used in the update step instead of the generic residual $\boldsymbol{v}_p - \boldsymbol{v}_m$ used in the standard Kalman Filter. The update equation then becomes

$$\boldsymbol{r}_{err_{t|t}} = K\boldsymbol{r}_{err} \tag{7}$$

which represented the attitude error between true and estimated attitude. K is the Kalman gain computed as in the standard KF. Hence, the attitude estimation is updated as follows:

$$q_{t|t} = \partial q(\boldsymbol{r}_{err_{t|t}}) \otimes q_{t|t-1} \tag{8}$$

where $\partial q(\boldsymbol{r}_{err_{t|t}})$ is the error quaternion obtained by converting from axis-angle to quaternion representation. The MKF is more effective for attitude estimation because its update step inherently respects the properties of rotations. The MKF algorithm, detailed in 2, is based on the methodology described in [3].

The update step of the MKF algorithm mirrors the KF steps: determining the measurement residual, calculating the Kalman gain, updating the state estimate, and updating the state covariance. The key difference is that the attitude error is now the state vector, represented using axis-angle parameterization. Thus, the output matrix C is the identity matrix, and the prediction step sets the state to zero, assuming zero measurement residual post-update. This algorithm estimates the attitude error, which is then used to update the actual attitude quaternion. The state transition matrix is given by [3]:

$$A = \begin{bmatrix} 0 & r & -q \\ -r & 0 & p \\ q & -p & 0 \end{bmatrix} \tag{9}$$

The algorithm relies on models of the measured quantities in the inertial frame, while the measurements are in the body frame. This allows the extraction of information about the drone's attitude. For the accelerometer, it's assumed the drone's acceleration is negligible, so the normalized measurement represents the gravity direction in the body frame, thus $\bar{r}^e_{acc} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$. The magnetometer measures the magnetic field direction in the body frame. A constant model of the magnetic field direction in the inertial frame is assumed due to the drone's small operating area. With the inertial frame chosen as the *NED*-frame, $\bar{r}^e_{mag} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$.

Algorithm 2 Multiplicative Kalman Filter Algorithm

Input: $\boldsymbol{q}_{0|0}$ and $\Sigma_{0|0}$ $\quad Q, R_{acc}$ and R_{mag}
for $t \in 1, 2, \ldots, n$ **do**
  Prediction Step:
  $\Sigma_{t|t-1} = A_{t-1}\Sigma_{t-1|t-1}A^T_{t-1} + Q$
  $\boldsymbol{r}_{err_{t|t-1}} = 0$
  $\bar{q} = q_{t-1|t-1} + \frac{dt}{2} q_{t-1|t-1} \otimes \begin{bmatrix} 0 \\ \tilde{\omega}^b \end{bmatrix}$
  $\hat{q}_{t|t-1} = \frac{\bar{q}}{||\bar{q}||_2}$

  Update Step:
  $R^b_e = quat2rotm(\hat{q}_{t|t-1})$
  Observe measurement $\tilde{r}^b_{mag}$
  $\boldsymbol{r}^{mag}_{err} \leftarrow \textit{Axis-Angle}\left(\tilde{\boldsymbol{r}}^b_{mag}, \bar{\boldsymbol{r}}^e_{mag}, R^b_e\right)$ (Eqns. 5, 6)
  $K = \Sigma_{t|t-1}\left(R_{mag} + \Sigma_{t|t-1}\right)^{-1}$
  $\boldsymbol{r}^{mag}_{err_{t|t}} = \boldsymbol{r}_{err_{t|t-1}} + K(\boldsymbol{r}^{mag}_{err} - \boldsymbol{r}_{err_{t|t-1}})$ (Equation 7)
  $\bar{\Sigma} = (I - K)\Sigma_{t|t-1}$

  Observe measurement $\tilde{f}^b$
  $\boldsymbol{r}^{acc}_{err} \leftarrow \textit{Axis-Angle}\left(\tilde{f}^b, \bar{r}^e_{acc}, R^b_e\right)$ (Eqns. 5, 6)
  $K = \Sigma_{t|t-1}\left(R_{acc} + \Sigma_{t|t-1}\right)^{-1}$
  $\boldsymbol{r}^{acc}_{err_{t|t}} = \boldsymbol{r}_{err_{t|t-1}} + K(\boldsymbol{r}^{acc}_{err} - \boldsymbol{r}_{err_{t|t-1}})$ (Equation 7)
  $\Sigma_{t|t} = (I - K)\bar{\Sigma}$

  $\hat{q}_{t|t} = \partial q(\boldsymbol{r}^{acc}_{err_{t|t}}) \otimes \partial q(\boldsymbol{r}^{mag}_{err_{t|t}}) \otimes q_{t|t-1}$
end for
Output: $\hat{q}_{n|n}$

B. Unscented Kalman Filter

The Unscented Kalman Filter (UKF) is another alternative to the Extended Kalman Filter for attitude estimation in highly non-linear dynamics. Like in the KF, in the UKF, the state distribution is represented by a Gaussian random variable (GRV), which is in turn represented by a number of sigma points. These sigma points are chosen to capture the true mean and covariance of the GRV, and when propagated through the non-linear model, capture the posterior mean and covariance more accurately compared to the EKF.

The filter utilizes an Unscented Transform (UT) to calculate the statistics of a random variable that undergoes a non-linear transformation. Assuming that we are propagating a random variable x with dimension n through a non-linear function $g(x)$, we form a matrix X of $2n+1$ sigma vectors X_i with corresponding weights W_i. Assuming x has mean $\bar{x}$ and covariance P_x the sigma vectors and weights are calculated as follows:

$$\begin{aligned} X_0 &= \bar{x} \\ X_i &= \bar{x} + \sqrt{(n+\lambda)P} \qquad i = 1, ..., n \\ X_{i+n} &= \bar{x} - \sqrt{(n+\lambda)P} \\ W_0 &= \frac{\lambda}{\lambda + n} \\ W_i &= \frac{1}{2(\lambda + n)} \qquad i = 1, ..., 2n \end{aligned} \tag{10}$$

Here λ is a tunable parameter. Once the sigma vectors are calculated, they are then fed into the non-linear function to produce

$$Y_i = g(X_i) \qquad i = 1, ..., 2n \tag{11}$$

The mean and covariance for y can then be approximated with the following equations.

$$\bar{y} \approx \sum_{i=0}^{2n} W_i Y_i \tag{12}$$

$$P_y \approx \sum_{i=0}^{2n} W_i (Y_i - \bar{y})(Y_i - \bar{y})^T$$

Given the dynamics model F, the measurement model H, the dynamics noise covariance matrix Q, and the measurement noise covariance matrix R, the UKF uses these sigma points very similarly to the KF, as seen in the algorithm below [9].

C. Particle Filter

The particle filter (PF) provides another suitable option for attitude estimation given the non-linear drone dynamics. Instead of using a parameterized distribution family, the particle filter represents the posterior state distribution using samples drawn from a large random set of state samples (particles) with associated weights. Each particle therefore becomes a hypothesis of the true attitude state with probability corresponding to the normalized particle weight.

Algorithm 3 Unscented Kalman Filter Algorithm

Input: $\bar{x}_{0|0}$ and $P_{0|0}$ Q and R
for $t \in 1, 2, 3, \cdots, t_f$ **do**
  Calculate Sigma Points:
  $X_{t-1|t-1} = \left[\bar{x}_{t-1|t-1}, \bar{x}_{t-1|t-1} \pm \sqrt{(n+\lambda)P_{t-1|t-1}}\right]$

  Prediction Step:
  $X_{t|t-1} = F[X_{t-1|t-1}]$
  $\bar{x}_{t|t-1} = \sum_{i=0}^{2n} W_i X_{i,t|t-1}$
  $P_{t|t-1} = \sum_{i=0}^{2n} W_i (X_{i,t|t-1} - \bar{x}_{t|t-1})(X_{i,t|t-1} - \bar{x}_{t|t-1})^T + Q$

  Update Step:
  $Y_{t|t-1} = H[X_{t|t-1}]$
  $\bar{y}_{t|t-1} = \sum_{i=0}^{2n} W_i Y_{i,t|t-1}$
  $P_y = \sum_{i=0}^{2n} W_i (Y_{i,t|t-1} - \bar{y}_{t|t-1})(Y_{i,t|t-1} - \bar{y}_{t|t-1})^T + R$
  $P_{xy} = \sum_{i=0}^{2n} W_i (X_{i,t|t-1} - \bar{x}_{t|t-1})(Y_{i,t|t-1} - \bar{y}_{t|t-1})^T$

  Observe measurement y_t
  $\bar{x}_{t|t} = \bar{x}_{t|t-1} + P_{xy} P_y^{-1} (y_t - \bar{y}_{t|t-1})$
  $P_{t|t} = P_{t|t-1} - P_{xy} P_y^{-1} P_{xy}^T$
end for
Output: $\bar{x}_{t_f|t_f}, P_{t_f|t_f}$

Algorithm 4 Particle Filter Algorithm

Input: $p(x_{0|0})$
Initialize:
$\bar{X}_{0|0} = \{(x_{0|0}^i, w_0^i = \frac{1}{N})\}^i$ for $i = 1, 2, \cdots, N$

for $t = 1, 2, \cdots, t_f$ **do**
  for $i = 1, 2, \cdots, N$ **do**
    Prediction Step:
    $x_{t|t-1}^i \sim p(x_t^i | u_t, x_{t-1}^i)$

    Update Step:
    Observe measurement y_t
    $\hat{w}_{t|t}^i = p(y_t | x_{t|t-1}^i)$
    $w_{t|t}^i = \frac{w_{t|t}^i}{\sum_{j=1}^N \hat{w}_{t|t}^j}$
  end for
  $\hat{x}_{t|t} = \sum_{j=1}^N w_{t|t}^j x_{t|t-1}^j$

  Importance Resampling:
  Draw index $m \in 1, 2, \cdots, N$ with probability $w_{t|t}^m$
  $\bar{X}_t^i = \{(\bar{X}_{t-1}^m, w_{t|t}^i = \frac{1}{N})\}$
end for
Output: $\hat{x}_{t|t}$

The PF algorithm initializes the particle set by sampling from an initial state belief, chosen to be Gaussian, with each particle equally weighted. Each particle is propagated through the nonlinear attitude dynamics model in the prediction step with different drawings of the attitude process noise.

In the update step, each particle's weight is calculated based on the conditional probability of seeing the attitude measurement given that particle and the attitude measurement model. This probability is obtained via a multivariate normal probability density function of the measurement using the measurement covariances. The weighted particles now represent our posterior belief and we can obtain our state estimate as the weighted mean of all particles.

The particle filter requires an importance resampling step which modifies the unweighted set of state particles to approximate the posterior belief. The resampling consists of a random sampling of each particle with a probability equal to its normalized weight. These new sampled points then become our particle set for the next timestep and we reset all particle weights to be equal.

It is worthwhile to note that while our drone attitude posterior distribution is modeled as Gaussian, the particle filter allows for an approximation of any parametric or non-parametric distributions (e.g. skew normal, multimodal, etc). This property makes the particle filter suitable for essentially any dynamics and measurement model.

V. Results

The three attitude estimation algorithms developed in this paper are evaluated through drone trajectory simulations. The trajectory is generated from random waypoints with timestamps based on waypoint distances. A cubic spline with clamped boundary conditions fits these waypoint-timestamp pairs, creating a reference trajectory. Figure 1 shows the 3D view of the trajectory, highlighting sharp turns and maneuvers. Figure 2 displays the reference position, velocity, and acceleration profiles, indicating a maximum acceleration of $5\,\mathrm{m/s^2}$. For trajectory tracking, the drone uses incremental nonlinear dynamic inversion.

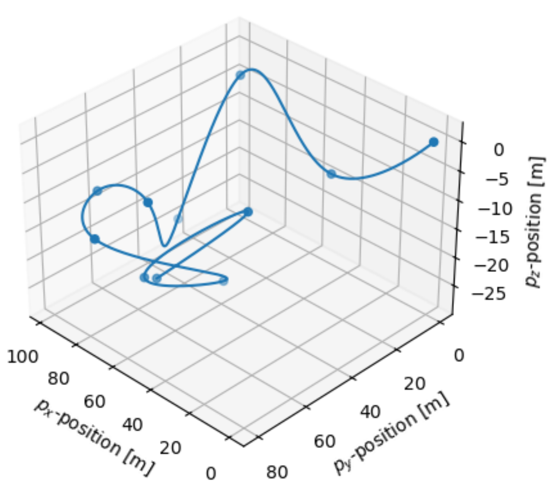

Fig. 1. 3d-view of the reference trajectory

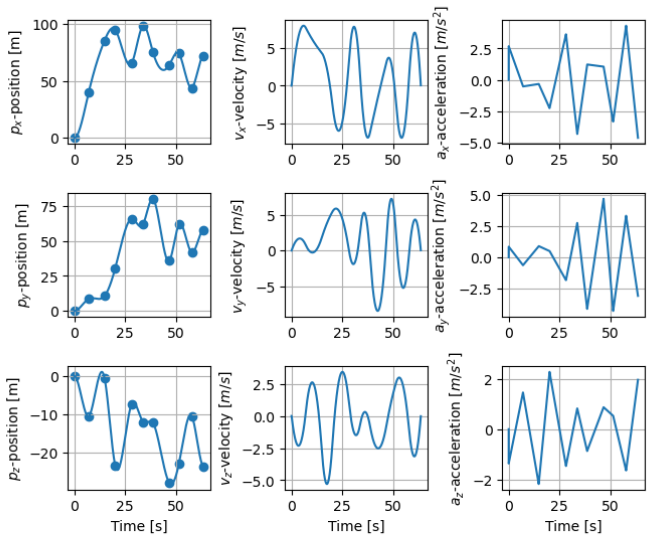

Fig. 2. Reference position, velocity, and acceleration over time. Position waypoints are indicated with blue circles.

A. Multiplicative Kalman Filter

This section presents the performance of the MKF for attitude estimation. Figure 3 shows the estimation error using Euler angle representation. These are the Euler angle rotations required to go from the attitude estimate to the true attitude. It can be seen that the error is contained to within $5[deg]$ along any Euler angle. However, the roll angle error seems to drift to a higher magnitude which is likely due to the gyroscope bias which is unaccounted for. Figure 4 shows the axis-angle representation of the estimation error, where the error is plotted. Additionally, the uncertainty in the estimate, extracted from the covariance, is also plotted. It can be seen that the maximum angle remains below $5[deg]$, but also drifts away from zero. The error does stay within the uncertainty bounds.

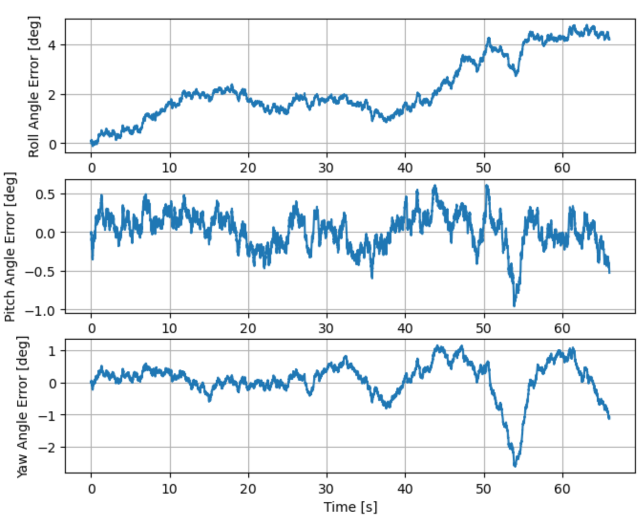

Fig. 3. Attitude estimation error using the MKF.

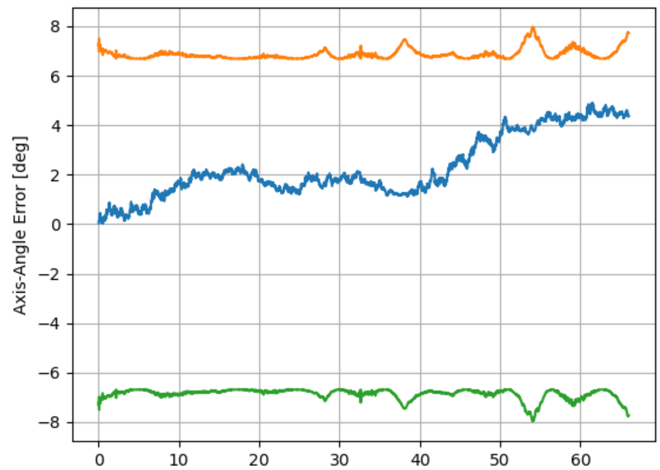

Fig. 4. Attitude estimation error using the MKF (blue), upper confidence level (orange) and lower confidence level (green).

B. Unscented Kalman Filter

This section contains the results from the UKF implementation for attitude estimation. Figure 6 provides an exact measure of the roll, pitch, and yaw angle estimation errors, which remain below 3 degrees after an initialization period.

Like the MKF, the angle errors seem to drift to a higher magnitude. This drift in the error values suggests that there may be compounding errors affecting the estimation process as the time steps advance. These compounding errors could be due to various factors, such as sensor noise, modeling inaccuracies, or numerical errors inherent in the computation process. The tendency of the errors to increase highlights the need for potential adjustments or compensatory mechanisms within the filter design to mitigate long-term drift and maintain estimation accuracy over extended periods.

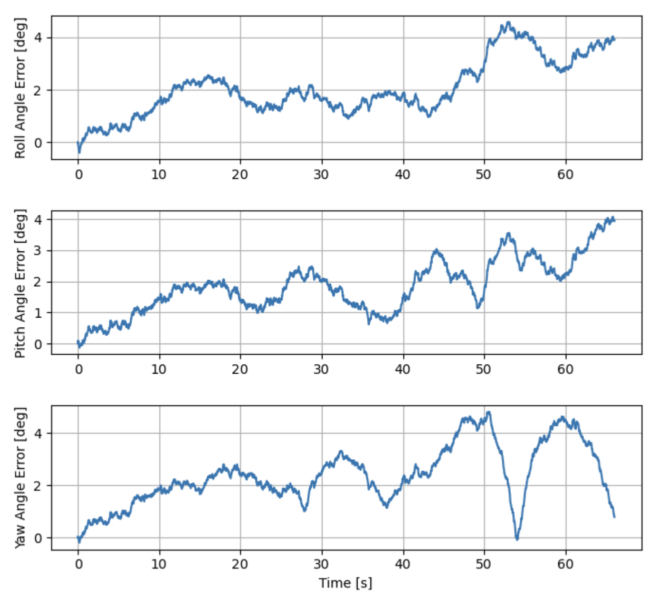

Fig. 5. Attitude estimation error using the UKF.

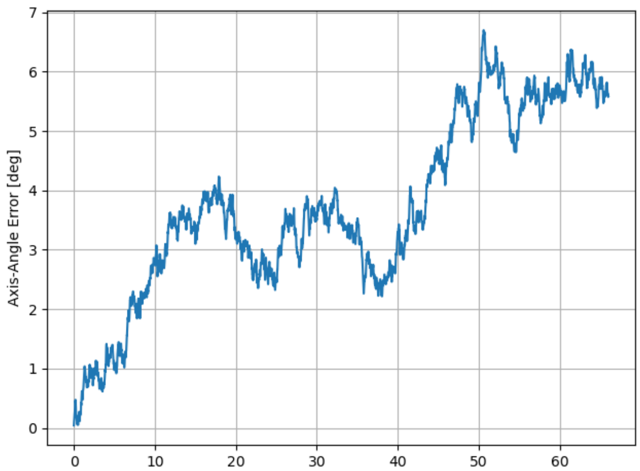

Fig. 6. Attitude estimation error using the UKF.

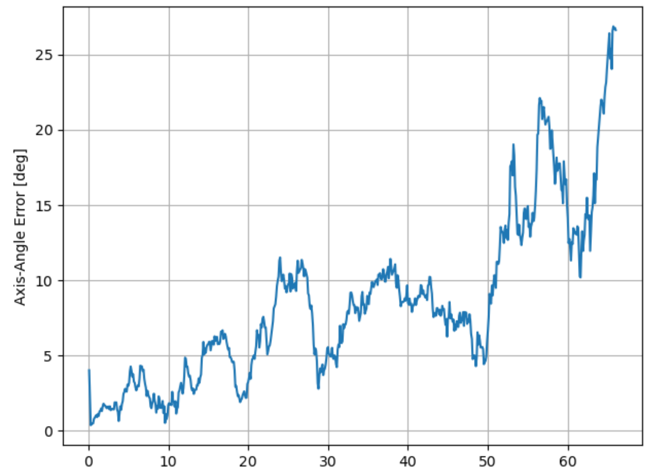

Fig. 8. Attitude estimation error using the PF.

C. Particle Filter

This section discusses the results of the Particle Filter (PF) attitude estimation implementation. Figure 7 illustrates the roll, pitch, and yaw angle attitude estimation errors. These errors generally remain under 10 degrees but tend to increase in magnitude over time, a pattern similar to what is observed with the Multiplicative Kalman Filter (MKF) and Unscented Kalman Filter (UKF) results. Additionally, the axis-angle error is depicted in Figure 8, showing a similar upward drift as time progresses. This drift behavior is likely caused by the gyroscope bias, which is currently ignored in the PF attitude estimation. Addressing this bias could potentially reduce the error drift and improve the accuracy of the attitude estimates.

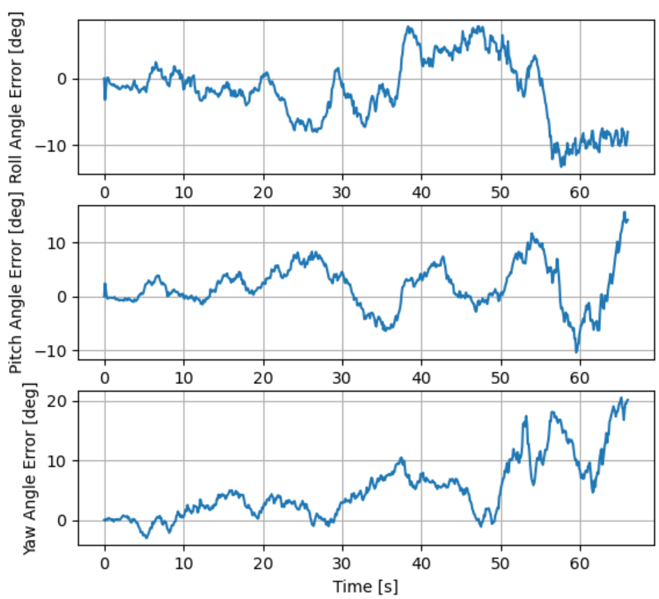

Fig. 7. Attitude estimation error using the PF.

D. Comparison of Results

Out of the three methods explored in this paper, the MKF achieves the highest attitude estimation accuracy. The mean estimation error is $3\times$ smaller than for the UKF and $4\times$ smaller than for the PF. While the maximum estimation error of the MKF is comparable to the UKF, it is $5\times$ smaller than the maximum estimation error of the PF. In terms of computation time, the UKF did have a shorter run time than the MKF, but only by a matter of seconds. As such, when weighing the performance of the filter with the runtime, the MKF was the superior filter. The PF runtime was significantly higher than either the MKF or UKF runtimes making it unsuitable for online estimation. The detailed comparisons can be found in Table 1.

TABLE I
COMPARISON OF ATTITUDE ESTIMATION METHODS.

Method	MKF	UKF	PF
Mean Estimation Error $[deg]$	2.627	8.611	10.07
Max. Estimation Error $[deg]$	4.885	6.62	26.88
Computation Time $[s]$	20.7	16.92	158.48

VI. CONCLUSION

In this project, we explored different nonlinear filtering algorithms for estimating the attitude state of a quadcopter drone. The filters implemented were the MKF, UKF, and the PF. We generally see acceptable accuracy across all filters but conclude the MKF to be the superior filter when considering filter performance and runtime. All filters exhibit a long-term error drift which is likely caused by an unaccounted gyroscope bias.

For future work, it is recommended to include estimation of the gyroscope bias to prevent estimation error drift, improving the performance of the estimation techniques. Additionally, other estimation techniques could be evaluated and compared.

Mike Timmerman was responsible for the quadcopter simulator and MKF attitude estimator. Irene Yeu was responsible for the UKF attitude estimator. Richard Qiu was responsible for the PF attitude estimator. The simulation and filter code can be found at https://github.com/MikeTimmerman-ae/Drone_State_Estimation.

REFERENCES

[1] Natássya B. F. da Silva, Daniel B. Wilson, and Kalinka R. L. J. Branco. Performance evaluation of the extended kalman filter and unscented kalman filter. In *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 733–741, 2015.

[2] G.A. Einicke and L.B. White. Robust extended kalman filtering. *IEEE Transactions on Signal Processing*, 47(9):2596–2599, 1999.

[3] James Hall, Nathan Knoebel, and Tim McLain. Quaternion attitude estimation for miniature air vehicles using a multiplicative extended kalman filter. pages 1230 – 1237, 06 2008.

[4] Hamza Hassan, William Tolstrup, Johanes Suriana, and Ibrahim Kiziloklu. A comparative analysis between the additive and the multiplicative extended kalman filter for satellite attitude determination, 07 2023.

[5] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[6] Rambabu Kandepu, Bjarne Foss, and Lars Imsland. Applying the unscented kalman filter for nonlinear state estimation. *Journal of Process Control*, 18(7):753–768, 2008.

[7] Jun Liu and Rong Chen. Sequential monte carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93, 04 1998.

[8] Mithra Sivakumar and Naga Tyj. A literature survey of unmanned aerial vehicle usage for civil applications. *Journal of Aerospace Technology and Management*, 13, 11 2021.

[9] E.A. Wan and R. Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, pages 153–158, 2000.

[10] Chi Zhang, Amirhossein Taghvaei, and Prashant G. Mehta. Attitude estimation with feedback particle filter. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 5440–5445, 2016.

[11] Zhaihe Zhou, Chuanwei Zeng, Xiangrui Tian, Qingxi Zeng, and Rui Yao. A discrete quaternion particle filter based on deterministic sampling for imu attitude estimation. *IEEE Sensors Journal*, 21(20):23266–23277, 2021.