

Comparison of Quaternion and Invariant Extended Kalman Filter

Jack Sonstegard

November 14, 2024

1 Introduction

1.1 Purpose

The purpose of this project is to explore advanced state estimation techniques that are applicable to the Aerospace industry. Specifically, I wanted to compare a Quaternion Extended Kalman Filter (QEKF) to a Invariant Extended Kalman Filter (InEKF). Both filters provide unique methods to tackle the problem of nonlinear state estimation in systems involving attitude. This report serves as a resource for background information on the formulation of each filter and provides a Monte Carlo simulation comparing each filter.

1.2 Outline

This document begins with a background on how IMU and GPS measurements can be modeled. The essential background needed to understand both the QEKF and InEKF filters is then provided including derivation steps for the error dynamics. Building on this foundation, the filters are applied in a simulation using drone data from the Mid-Air Dataset, which includes accelerometer and gyroscope measurements [FD19], along with GPS position data. A Monte Carlo analysis reveals that when using just a GPS position measurement both the InEKF and QEKF suffer from unobservability in the bias states. However, quicker convergence to a solution is seen across Monte Carlos in the InEKF.

1.3 Disclaimer

This document is not intended to be a comprehensive guide to each filter but is intended to provide enough background such that someone could start experimenting with each filter assuming they already have some background in Kalman Filtering. Throughout this document different references will be cited, however, there are two main papers that helped me learn the most about each filter. The first is the paper by Joan Solà titled "Quaternion kinematics for the error-state Kalman Filter" [Sol17]. This paper gives detailed explanation of quaternion definitions, conventions, and their use in filtering. The second paper is by Ross Hartley et al., titled "Contact-Aided Invariant Extended Kalman Filtering for Robot State Estimation" [HJEG19]. This paper was my original inspiration for starting this project. In this paper, the InEKF is derived in multiple forms and necessary background is provided for Lie groups and Lie algebras.

2 IMU GPS Modeling

2.1 Purpose

In order to compare the QEKF and InEKF a drone simulation from the Mid-Air Dataset [FD19] was used which simulated low altitude drone flights. Using the trajectory of the drone, an Inertial Measurement Unit (IMU) containing an accelerometer and gyroscope as well as a GPS receiver was simulated. These sensors provide measurements that are received by each filter and must be correctly modeled by each filter to estimate the drone pose.

This section explains how these sensors are modeled in the simulation. This model will then be subsequently used in each filter in order to estimate the state of the drone.

2.2 Continuous Time Model

The IMU consists of an model for the acceleration, angular rate, and random walk biases. These states are measured in the body frame of the drone. Overall, two coordinate frames will be used in this report. A body frame attached to the drone and a world frame with is the initial starting point of the drone trajectory. Both frames use the North, East, Down (NED) convention. The measured acceleration and angular rate of the drone in the body frame coordinates as well as the model of the biases are defined as

$$\tilde{a} = a + b^a + w^a, \quad w^a \sim \mathcal{N}(0_{3,1}, \sigma_a^2) \quad (1a)$$

$$\tilde{\omega} = \omega + b^\omega + w^\omega, \quad w^\omega \sim \mathcal{N}(0_{3,1}, \sigma_\omega^2) \quad (1b)$$

$$\dot{b}^a = w^{ba}, \quad w^{ba} \sim \mathcal{N}(0_{3,1}, \sigma_{ba}^2) \quad (1c)$$

$$\dot{b}^\omega = w^{b\omega}, \quad w^{b\omega} \sim \mathcal{N}(0_{3,1}, \sigma_{b\omega}^2) \quad (1d)$$

Each variable in the above equations are in \mathbb{R}^3 representing a component in the North, East, and Down directions for the respective coordinate frame. In equations (1a) and (1b), the measured values \tilde{a} and $\tilde{\omega}$ are the sum of the true values a and ω , the random walk biases b^a and b^ω , and white noises w^a and w^ω . The random walk biases are defined in equations (1c) and (1d) by integrating the white noises w^{ba} and $w^{b\omega}$ which simulates a slowly moving bias that drifts over time. The system dynamics of the drone can then be written as the following

$$\dot{q} = \frac{1}{2} q \otimes (\tilde{\omega} - b^\omega - w^\omega) \quad (2a)$$

$$\dot{R} = R(\tilde{\omega} - b^\omega - w^\omega)_\times \quad (2b)$$

$$\dot{v} = R(\tilde{a} - b^a - w^a) + g \quad (2c)$$

$$\dot{p} = v \quad (2d)$$

Note that two methods are used to rotate measurements in the body frame to the world frame. The first method is shown using a quaternion denoted as $q \in \mathbb{R}^4$. The use of quaternion multiplication \otimes is used in the definition and is defined further in (11). The quaternion q can also be written in this context as q_{WB} to signify its use in rotating from the body to world frame. The second method is the rotation matrix $R \in \mathbb{R}^{3 \times 3}$ also written as R_{WB} . Additionally note that in equation (2c), the acceleration is written with the inclusion of gravity g , which is sensed by the accelerometer.

The angular rate ω is measured in the body frame and can be written formally as ω_B^{BW} . This notation signifies that the angular rate is measured in the body frame and can be represented as a vector with a origin in the world frame pointing to the body frame. The skew operator $(\cdot)_\times$ used in equation (2b) is written as

$$\omega_\times = \begin{bmatrix} 1 & -\omega_3 & \omega_2 \\ \omega_3 & 1 & -\omega_1 \\ -\omega_2 & -\omega_1 & 1 \end{bmatrix} \quad (3)$$

2.3 Discrete Time Model

Each filter in this report is written in discrete time, therefore, it is necessary to discretized each continuous time equation. This can be accomplished by assuming a zero order hold (ZOH) of the measurements over a time interval $\Delta t = t_k - t_{k-1}$.

The bias dynamics in Equations (1c) and (1d) are equal to white Gaussian noise. Therefore in propagating these equations forward in time the biases are simply

$$b_k^a = b_{k-1}^a, \quad b_k^\omega = b_{k-1}^\omega \quad (4)$$

The orientation represented by the quaternion can be propagated using the Taylor series [Sol17]. The series can be expanded and each quaternion derivative can be written in terms of q and the

measured angular rates. Setting $\dot{\omega}$ to zero over the ZOH yields the exponential which can be related to quaternions complex and real components as shown in (17) and (18)

$$\begin{aligned}
q_k &= q_{k-1} + \dot{q}_{k-1}\Delta t + \frac{1}{2}\ddot{q}_{k-1}\Delta t^2 + \dots \\
&= q_{k-1} + \frac{1}{2}q_{k-1} \otimes (\tilde{\omega}_{k-1} - b_{k-1}^\omega)\Delta t + \frac{1}{2}\left(\frac{1}{4}q_{k-1} \otimes (\tilde{\omega}_{k-1} - b_{k-1}^\omega)^2\Delta t^2 + \frac{1}{2}q_{k-1} \otimes \frac{d}{dt}(\tilde{\omega}_{k-1} - b_{k-1}^\omega)\Delta t^2\right) + \dots \\
&= q_{k-1} \otimes \left(1 + \frac{1}{2}(\tilde{\omega}_{k-1} - b_{k-1}^\omega)\Delta t + \frac{1}{2}\left(\frac{1}{4}(\tilde{\omega}_{k-1} - b_{k-1}^\omega)\Delta t\right)^2 + \dots\right) \\
&= q_{k-1} \otimes \exp(\tilde{\omega}_{k-1} - b_{k-1}^\omega)\Delta t \\
&= q \{(\tilde{\omega}_{k-1} - b_{k-1}^\omega)\Delta t\} \\
&= \begin{bmatrix} \cos\left(\|\tilde{\omega}_{k-1} - b_{k-1}^\omega\|\frac{\Delta t}{2}\right) \\ \frac{(\tilde{\omega}_{k-1} - b_{k-1}^\omega)}{\|\tilde{\omega}_{k-1} - b_{k-1}^\omega\|} \sin\left(\|\tilde{\omega}_{k-1} - b_{k-1}^\omega\|\frac{\Delta t}{2}\right) \end{bmatrix}
\end{aligned} \tag{5}$$

Note that the white noise term in $\tilde{\omega}$ is dropped in (5) because its mean is zero. Note also that equation (5) essentially converts angular rates to a quaternion which can also be done using equation (22). The orientation represented by the rotation matrix is propagated by the following equation

$$R_k = \int_{t_{k-1}}^{t_k} R(\tilde{\omega} - b^\omega)_\times dt = R_{k-1} \exp((\tilde{\omega}_{k-1} - b_{k-1}^\omega)_\times \Delta t) \tag{6}$$

The velocity can then be written as

$$\begin{aligned}
v_k &= v_{k-1} + g\Delta t + (\tilde{a}_{k-1} - b_{k-1}^a) \int_{t_{k-1}}^{t_k} R dt \\
&= v_{k-1} + g\Delta t + R_{k-1}(\tilde{a}_{k-1} - b_{k-1}^a) \int_{t_{k-1}}^{t_k} \exp((\tilde{\omega} - b^\omega)t) dt \\
&\approx v_{k-1} + g\Delta t + R_{k-1}(\tilde{a}_{k-1} - b_{k-1}^a)(I\Delta t + \frac{1}{2}\Delta t^2(\tilde{\omega}_{k-1} - b_{k-1}^\omega)_\times)
\end{aligned} \tag{7}$$

Here the identity matrix is defined as $I \in \mathbb{R}^3$. Note that in (7), the Taylor Series first order approximation is used in integrating the exponential $\exp(\omega) \approx I + \omega$. Integrating equation (7) one more time yields the discrete position update

$$p_k \approx p_{k-1} + \frac{1}{2}g\Delta t^2 + R_{k-1}(\tilde{a}_{k-1} - b_{k-1}^a)(I\Delta t^2 + \frac{1}{2}\Delta t^3(\tilde{\omega}_{k-1} - b_{k-1}^\omega)_\times) \tag{8}$$

2.4 GPS Model

The GPS receiver is simply modeled to solely receive a position estimate of the drone in the north, east, and down coordinates. This measurement is considered to be fairly accurate but is received at a slower rate than the acceleration and angular rate measurements. The measurement is modeled as

$$\tilde{y}_p = y_p + \nu, \quad \nu \sim \mathcal{N}(0, \sigma_{y_p}^2) \tag{9}$$

Here $\tilde{y}_p \in \mathbb{R}^3$ is the received measurement from the GPS receiver. This measurement is equal to the true position plus white Gaussian noise $\nu \in \mathbb{R}^3$.

3 Quaternion Extended Kalman Filter

3.1 Purpose

Quaternions are one way to represent a rotation in 3D space. They are four-dimensional vectors that allow efficient and stable computation of orientation and rotation. Quaternions have many advantages over other representations such as Euler angles, especially when it comes to representing the attitude state of a system. One significant advantage is that quaternions avoid the singularities, or “gimbal

lock” [PTKXM16] that can arise with Euler angles, which leads to undefined attitudes. Furthermore, quaternions provide smoother interpolation between orientations and require fewer computational resources due to their compact form.

In this section, a Quaternion Extended Kalman Filter is detailed by first giving a background on quaternions and useful properties. The state model is then formulated considering the process dynamics and measurement model for the QEKF. An outline is then provided for running the algorithm.

3.2 Background

The quaternion q is represented by a four element vector made of a complex $q_w \in \mathbb{R}$ and real part $q_v \in \mathbb{R}^3$

$$q = \begin{bmatrix} q_w \\ q_v \end{bmatrix} = \begin{bmatrix} q_w \\ q_x \\ q_y \\ q_z \end{bmatrix} \quad (10)$$

Quaternion multiplication \otimes is used in the composition of two quaternions and can be defined for two quaternions p and q [Sol17] as

$$p \otimes q = \begin{bmatrix} p_w q_w - p_x q_x - p_y q_y - p_z q_z \\ p_w q_x + p_x q_w + p_y q_z - p_z q_y \\ p_w q_y - p_x q_z + p_y q_w + p_z q_x \\ p_w q_z + p_x q_y - p_y q_x + p_z q_w \end{bmatrix} \quad (11)$$

An alternate expression for quaternion multiplication that can be utilized is the following [Sol17]

$$p \otimes q = (p)_L q = (q)_R p \quad (12)$$

Where $(q)_L$ and $(q)_R$ are given by

$$(q)_L = q_w I + \begin{bmatrix} 0 & -q_v^T \\ q_v & (q_v)_{\times} \end{bmatrix} \quad (13)$$

$$(q)_R = q_w I + \begin{bmatrix} 0 & -q_v^T \\ q_v & -(q_v)_{\times} \end{bmatrix} \quad (14)$$

The conjugate of the quaternion is defined as follows and is utilized in the quaternion norm

$$q^* = \begin{bmatrix} q_w \\ -q_v \end{bmatrix} \quad (15)$$

$$\|q\| = \sqrt{q \otimes q^*} = \sqrt{q^* \otimes q} = \sqrt{q_w^2 + q_x^2 + q_y^2 + q_z^2} \quad (16)$$

A unit quaternion, with a norm equal to 1 ($\|q\| = 1$), is the type of quaternion used in this report and can be shown to be related to the exponential map. The relationship to the exponential map can be related to a rotation action about a angle $\psi \in \mathbb{R}$ and unit axis $u \in \mathbb{R}^3$. While further explained in [Sol17], the resulting equation is the following

$$q = \exp(\psi u) = q\{\psi u\} = \cos\left(\frac{\psi}{2}\right) + u \sin\left(\frac{\psi}{2}\right) = \begin{bmatrix} \cos\left(\frac{\psi}{2}\right) \\ u \sin\left(\frac{\psi}{2}\right) \end{bmatrix} \quad (17)$$

A incremental rotation can then be represented as $\Delta\theta = \omega\Delta t$ where $u = \frac{\omega\Delta t}{\|\omega\Delta t\|}$ and $\psi = \|\omega\Delta t\|$. Substituting into equation (17), an incremental quaternion can be defined as

$$\delta q = \exp(\omega\Delta t) = q\{\omega\Delta t\} = \cos\left(\frac{\|\omega\Delta t\|}{2}\right) + \frac{\omega\Delta t}{\|\omega\Delta t\|} \sin\left(\frac{\|\omega\Delta t\|}{2}\right) = \begin{bmatrix} \cos\frac{\|\omega\Delta t\|}{2} \\ \frac{\omega\Delta t}{\|\omega\Delta t\|} \sin\frac{\|\omega\Delta t\|}{2} \end{bmatrix} \quad (18)$$

This incremental quaternion δq can also simply be approximated by [Sol17]

$$\delta q = \begin{bmatrix} 1 \\ \frac{1}{2}\omega\Delta t \end{bmatrix} \quad (19)$$

The rotation of a given vector can be accomplished with the double quaternion product. As an example, the position vector in the body frame can be rotated into the world frame

$$p_W^{BW} = q_{WB} \otimes p_B^{BW} \otimes q_{WB}^* \quad (20)$$

A quaternion can also be converted into a rotation matrix, which can accomplish the same transformation as in (20), with the following formula [Sol17]

$$R = R\{q\} = \begin{bmatrix} q_w^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_x q_y - q_w q_z) & 2(q_x q_z + q_w q_y) \\ 2(q_x q_y + q_w q_z) & q_w^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_y q_z - q_w q_x) \\ 2(q_x q_z - q_w q_y) & 2(q_y q_z + q_w q_x) & q_w^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix} \quad (21)$$

Additionally, there exists conversions between quaternions and Euler angles which are necessary to initialize a quaternion about initial set of Euler angles and to convert a quaternion to a more understandable Euler angle representation. These conversions are defined for a 3-2-1 rotation sequence for the Euler angles $\theta = [\theta_1, \theta_2, \theta_3]^T$ [Hen77] [BC21]

$$q = q\{\theta\} = \begin{bmatrix} \cos(\theta_1/2) \cos(\theta_2/2) \cos(\theta_3/2) + \sin(\theta_1/2) \sin(\theta_2/2) \sin(\theta_3/2) \\ \sin(\theta_1/2) \cos(\theta_2/2) \cos(\theta_3/2) - \cos(\theta_1/2) \sin(\theta_2/2) \sin(\theta_3/2) \\ \cos(\theta_1/2) \sin(\theta_2/2) \cos(\theta_3/2) + \sin(\theta_1/2) \cos(\theta_2/2) \sin(\theta_3/2) \\ \cos(\theta_1/2) \cos(\theta_2/2) \sin(\theta_3/2) - \sin(\theta_1/2) \sin(\theta_2/2) \cos(\theta_3/2) \end{bmatrix} \quad (22)$$

$$\theta = \theta\{q\} = \begin{bmatrix} \text{atan2}(2(q_w q_x + q_y q_z), 1 - 2(q_x^2 + q_y^2)) \\ \text{asin}(2(q_w q_z - q_x q_y)) \\ \text{atan2}(2(q_w q_z + q_x q_y), 1 - 2(q_y^2 + q_z^2)) \end{bmatrix} \quad (23)$$

Note that equation (22), provides an alternate method to compute δq as done in equation (18). This alternate method will be the one used in this report.

3.3 Process Model Error States and Jacobians

The QEKF used in this report is a Error State Kalman Filter (ESKF). In this report, this means that the matrices used to update the state covariance for the Kalman Filter are derived from errors dynamics for each of the states in the filter. Error dynamics can better handle nonlinear state dynamics because the ESKF linearizes only the error between the estimated and true states as opposed to linearizing the entire state which can introduce larger linearization errors. Therefore, by using the error states to linearize the system, the covariance can be more accurately propagated.

The propagation of the states themselves is done using the discrete process model equation defined in equations (5), (7), and (8). Because of the use of the error states, the estimated state must be broken down into a error δx and nominal state \bar{x} . The nominal state is propagated by the high frequency IMU data not accounting for noise. The nominal states are defined as

$$\bar{x}_k = \begin{bmatrix} \bar{p}_k \\ \bar{v}_k \\ \bar{q}_k \\ \bar{b}_k^a \\ \bar{b}_k^\omega \end{bmatrix} \quad (24)$$

Additionally, the inputs from the IMU are defined as

$$u_k = \begin{bmatrix} \tilde{a}_k \\ \tilde{\omega}_k \end{bmatrix} \quad (25)$$

The process model $f_{\text{QEKF}}(\bar{x}_{k-1}, u_{k-1})$ for the nominal states is

$$\bar{p}_k = \bar{p}_{k-1} + \frac{1}{2}g\Delta t^2 + \bar{R}_{k-1}(\tilde{a}_{k-1} - \bar{b}_{k-1}^a) \left(I\Delta t^2 + \frac{1}{2}\Delta t^3(\tilde{\omega}_{k-1} - \bar{b}_{k-1}^\omega)_{\times} \right) \quad (26a)$$

$$\bar{v}_k = \bar{v}_{k-1} + g\Delta t + \bar{R}_{k-1}(\tilde{a}_{k-1} - \bar{b}_{k-1}^a) \left(I\Delta t + \frac{1}{2}\Delta t^2(\tilde{\omega}_{k-1} - \bar{b}_{k-1}^\omega)_{\times} \right) \quad (26b)$$

$$\bar{q}_k = \bar{q}_{k-1} \otimes \bar{q}_{k-1} \{ \Delta t(\tilde{\omega}_{k-1} - \bar{b}_{k-1}^\omega) \} \quad (26c)$$

$$\bar{b}_k^a = \bar{b}_{k-1}^a \quad (26d)$$

$$\bar{b}_k^\omega = \bar{b}_{k-1}^\omega \quad (26e)$$

The error state vector is similar to equation (24), however, the parameters for the quaternion are replaced with the Euler angles because while it is useful to update the representation of the attitude with the quaternion, the actual error in the attitude is better represented by the Euler angles themselves. The error states vector δx is

$$\delta x_k = \begin{bmatrix} \delta p_k \\ \delta v_k \\ \delta \theta_k \\ \delta b_k^a \\ \delta b_k^\omega \end{bmatrix} \quad (27)$$

To derive the error states, the true state can be broken down into the nominal state and error state

$$x = \bar{x} \oplus \delta x \quad (28)$$

Where the symbol \oplus is used to represent the addition of the nominal and error state and accounts for the quaternion multiplication \otimes required to update quaternion. A quaternion error in the world frame, also known as a right error for a world centric estimator, is equal to [Sol17]

$$\delta q = q \otimes \bar{q}^* \quad (29)$$

Therefore, to update a quaternion q , which can be formally written as q_{WB} , the error quaternion state can be multiplied by the nominal quaternion

$$q = \delta q \otimes \bar{q} \quad (30)$$

To derive the velocity error, equation (2c) can be written in terms of the true states then broken down into the nominal and error states. In deriving (32), it is useful to derive a alternate equation for acceleration error

$$\begin{aligned} a &= \tilde{a} - b^a - w^a \\ &= (\tilde{a} - \bar{b}^a) + (-\delta b^a - w^a) \\ &= \bar{a} + \delta a \end{aligned} \quad (31)$$

From (31), it is shown that $\delta a = -\delta b^a - w^a$. A first order approximation can also be used to update the rotation matrix from the error $\delta \theta$. Furthermore, using the proof that uncorrelated white noise is invariant under a rotation [Sol17] and dropping second order terms the continuous time error velocity equation is

$$\begin{aligned} \dot{v} &= Ra + g \\ \dot{\bar{v}} + \delta \dot{v} &\approx (I + (\delta \theta)_\times) \bar{R}(\bar{a} + \delta a) + g \\ \bar{R}\bar{a} + g + \delta \dot{v} &= \bar{R}\delta a + \bar{R}\delta a + (\delta \theta)_\times \bar{R}\bar{a} + (\delta \theta)_\times \bar{R}\delta a + g \\ \delta \dot{v} &= \bar{R}\delta a + (\delta \theta)_\times \bar{R}(\bar{a} + \delta a) \\ &\approx \bar{R}\delta a - (\bar{R}\bar{a})_\times \delta \theta \\ &= -(\bar{R}(\tilde{a} - \bar{b}^a))_\times \delta \theta - \bar{R}\delta b^a - \bar{R}w^a \\ &= -(\bar{R}(\tilde{a} - \bar{b}^a))_\times \delta \theta - \bar{R}\delta b^a - w^a \end{aligned} \quad (32)$$

To derive the attitude error $\delta \theta$, the continuous time equation for a quaternion (2a) can be broken down into nominal and error components using equation (30). Because ω in (2a) is in the body frame, equation (20) can be used as well to redefine the coordinate frame of ω . Using equation (14) allows $\delta q \otimes \omega$ to be rewritten as a matrix with δq being approximated by equation (19). Dropping second

order terms, the attitude error is

$$\begin{aligned}
\dot{q} &= \frac{1}{2}q \otimes (\omega) \\
\frac{d}{dt}(\delta q \otimes \bar{q}) &= \frac{1}{2}q \otimes (\omega) \\
\delta \dot{q} \otimes \bar{q} + \delta q \otimes (\frac{1}{2}\bar{q} \otimes \bar{\omega}) &= \frac{1}{2}q \otimes (\omega) \\
\delta \dot{q} \otimes \bar{q} &= \frac{1}{2}(-\delta q \otimes (\frac{1}{2}\bar{q} \otimes \bar{\omega}) + q \otimes \omega) \\
\delta \dot{q} \otimes \bar{q} &= \frac{1}{2}\delta q \otimes \bar{q} \otimes \delta \omega \\
\delta \dot{q} &= \frac{1}{2}\delta q \otimes \bar{q} \otimes \delta \omega \otimes \bar{q}^* \\
\delta \dot{q} &= \frac{1}{2}\delta q \otimes \delta \omega_W^{BW} \\
\begin{bmatrix} 0 \\ \delta \dot{\theta} \end{bmatrix} &= 2\delta \dot{q} = \delta q \otimes \delta \omega_W^{BW} \\
&\approx \begin{bmatrix} 0 & (-\delta \omega_W^{BW})^T \\ \delta \omega_W^{BW} & -(\omega_W^{BW})_{\times} \end{bmatrix} \begin{bmatrix} 1 \\ \frac{1}{2}\delta \theta \end{bmatrix} \\
\delta \dot{\theta} &= \delta \omega_W^{BW} - \frac{1}{2}(\delta \omega_W^{BW})_{\times} \delta \theta \\
&\approx \delta \omega_W^{BW} = R_{WB} \delta \omega \\
&= -R \delta b^{\omega} - R w^{\omega} \\
&= -R \delta b^{\omega} - w^{\omega}
\end{aligned} \tag{33}$$

The continuous time error states are then

$$\delta \dot{p} = \delta v \tag{34a}$$

$$\delta \dot{v} = -(\bar{R}(\tilde{a} - \bar{b}^a))_{\times} \delta \theta - \bar{R} \delta b^a - w^a \tag{34b}$$

$$\delta \dot{\theta} = -R \delta b^{\omega} - w^{\omega} \tag{34c}$$

$$\delta \dot{b}^a = w^{ba} \tag{34d}$$

$$\delta \dot{b}^{\omega} = w^{b\omega} \tag{34e}$$

These continuous time error states can be discretized user Euler integration. To calculate the rotation matrix, the nominal estimate of the quaternion can be converted into the rotation matrix using equation (21)

$$\delta p_k = \delta p_{k-1} + \delta v_{k-1} \Delta t \tag{35a}$$

$$\delta v_k = \delta v_{k-1} + (-(R\{\bar{q}_{k-1}\}(\tilde{a} - \bar{b}_{k-1}^a))_{\times} \delta \theta_{k-1} - R\{\bar{q}_{k-1}\} \delta b_{k-1}^a + w^a) \Delta t \tag{35b}$$

$$\delta \theta_k = \delta \theta_{k-1} - (R\{\bar{q}_{k-1}\} \delta b_{k-1}^{\omega} + w^{\omega}) \Delta t \tag{35c}$$

$$\delta b_k^a = \delta b_{k-1}^a + w^{ba} \Delta t \tag{35d}$$

$$\delta b_k^{\omega} = \delta b_{k-1}^{\omega} + w^{b\omega} \Delta t \tag{35e}$$

With the discrete time equations, the state transition matrix F_k and state covariance matrix Q for the Kalman Filter can be defined as follows

$$F_k = \left. \frac{\partial f}{\partial \delta x} \right|_{\bar{x}} = \begin{bmatrix} I & \Delta t I & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I & (-(R\{\bar{q}_{k-1}\}(\tilde{a} - \bar{b}_{k-1}^a))_{\times}) & -R\{\bar{q}_{k-1}\} \Delta t & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I & \mathbf{0} & -R\{\bar{q}_{k-1}\} \Delta t \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & I & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & I \end{bmatrix} \tag{36}$$

$$Q = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \sigma_{a,d}^2 I & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \sigma_{\omega,d}^2 I & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \sigma_{ba,d}^2 I & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \sigma_{b\omega,d}^2 I \end{bmatrix} \quad (37)$$

Here the bolded zeros $\mathbf{0}$ represents a 3 by 3 zero matrix $0_{3 \times 3}$. Additionally, the subscript d is used to represent the discretized noise based on table (1).

3.4 Measurement Jacobian

The measurement Jacobian H is defined with respect to the error state δx . However, for the GPS model, the estimated position is not a error state and is equal to \bar{p} . Therefore, the chain rule can be used to break down the Jacobian into derivatives of known states [Sol17]

$$H = \frac{\partial h}{\partial \delta x} \Big|_{\bar{x}} = \frac{\partial h}{\partial x} \Big|_{\bar{x}} \frac{\partial x}{\partial \delta x} \Big|_{\bar{x}} \quad (38)$$

With the GPS model being fairly simple the only partial in $\frac{\partial x}{\partial \delta x} \Big|_{\bar{x}}$ of relevance is $\frac{\partial(\bar{p} + \delta p)}{\partial \delta p}$ which is equal to the identity matrix. Therefore, the measurement Jacobian reduces to the typical one derived without error states equal to

$$H = \begin{bmatrix} I & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ & 0_{12 \times 15} & & & \end{bmatrix} \quad (39)$$

The full measurement $\tilde{y} \in \mathbb{R}^{15}$ and the measurement covariance $N \in \mathbb{R}^{15 \times 15}$ are defined as

$$\tilde{y} = \begin{bmatrix} \tilde{y}_p \\ 0_{12} \end{bmatrix} \quad (40)$$

$$N = \begin{bmatrix} \sigma_{y_p}^2 I & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ & 0_{12 \times 15} & & & \end{bmatrix} \quad (41)$$

3.5 ESEK Reset Jacobian

As described in [Sol17], after the nominal states is updated with the error state, during the correction step, the error δx gets reset to zero. Therefore, to make the ESKF update complete, the covariance of the error needs to reflect this reset. Calling the error reset function $g()$, it is defined as [Sol17]

$$\delta x \leftarrow g(\delta x) = \delta x \ominus \delta \hat{x} \quad (42)$$

Here \ominus is the composition inverse of \otimes . Additionally, the $(\hat{\cdot})$ is used to signify the estimated error variable. The ESKF reset is defined as [Sol17]

$$\delta \hat{x} \leftarrow 0 \quad (43a)$$

$$P \leftarrow GPG^T \quad (43b)$$

Here $P \in \mathbb{R}^{15 \times 15}$ is the error state covariance matrix and $G \in \mathbb{R}^{15 \times 15}$ is the reset Jacobian matrix defined as

$$G = \frac{\partial g}{\partial \delta x} \Big|_{\delta \hat{x}} \quad (44)$$

The proof for the matrix G can be found in [Sol17]. The resulting matrix affects only the error attitude states and is defined as

$$G = \begin{bmatrix} I_6 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I + (\frac{1}{2}\delta\hat{\theta})_{\times} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_9 \end{bmatrix} \quad (45)$$

3.6 Algorithm

In this section, the QEKF algorithm is outlined. The initial nominal state \bar{x}_0 and error state covariance P_0 are first initialized. The algorithm then has two optional steps given the current measurement. If IMU measurements are available, which are expected to be measured at a high rate, the propagation steps are followed. During propagation, the nominal state and error state covariance are updated. An additional check is included to ensure that the quaternion state maintains a norm close to one and if the norm has deviated the quaternion is normalized. If GPS measurements are available, which are collected at a slower rate, the state and covariance update steps are followed. The Kalman gain K is first calculated and used to estimate the error state δx_k . This is then used to update the true state x_k which is redefined to be the nominal state \bar{x}_k . The quaternion is again checked to ensure it is a unit quaternion. This is then followed by the update to the error state covariance which includes the error state reset matrix G . Note that the error state is always redefined as part of the correction step but no memory is given to the state which accomplishes the error reset. Also note that traditionally the state variables \bar{x} and δx are given the hat symbol ($\hat{\cdot}$) to indicate that they are estimated but this notation was dropped to reduce clutter in the following algorithm.

Algorithm 1: QEKF

```

 $P_0 = \Sigma_0;$ 
 $\bar{x}_0 = x_0;$ 
while receiving measurements do
  if IMU measurement then
     $\bar{x}_k = f_{\text{QEKF}}(\bar{x}_{k-1}, u_{k-1});$  // See equations in (26)
     $F_k = \left. \frac{\partial f}{\partial x} \right|_{\bar{x}_{k-1}};$ 
     $P_k = F_k P_{k-1} F_k^T + Q;$ 
    if  $|1 - \|\bar{q}_k\|| > 10^{-5}$  then
       $\bar{q}_k = \frac{\bar{q}_k}{\|\bar{q}_k\|};$ 
    end
  end
  if GPS measurement then
     $S = H P_k H^T + V;$ 
     $K = P_k H^T S^{-1};$ 
     $\delta x_k^+ = K(\tilde{y} - H \bar{x}_k);$ 
     $\bar{x}_k^+ = \delta x_k^+ \oplus \bar{x}_k$  // Estimated true state  $x_k$  redefined as nominal state  $\bar{x}_k$ 
    if  $|1 - \|\bar{q}_k\|| > 10^{-5}$  then
       $\bar{q}_k \leftarrow \frac{\bar{q}_k}{\|\bar{q}_k\|};$ 
    end
     $P_k^+ = (I - KH) P_k (I - KH)^T + K N K^T;$ 
     $P_k^{++} = G P_k^+ G^T$ 
  end
end

```

4 Invariant Extended Kalman Filter

4.1 Purpose

This section will provide the necessary background to implement the InEKF. The InEKF is a more recent filter design as of 2015 first purposed in “The Invariant Extended Kalman filter as a stable observer” [BB14]. The filter uses the theory of Lie groups and log-linear error dynamics to achieve a invariant Extended Kalman Filter. This formulation allows the error dynamics to be independent of the system trajectory, avoid linearization dependent on the current state estimate, and lead to improved convergence [HJEG19]. Furthermore, Lie groups are singularity free which like the QEKF

prevents “gimbal lock”. However, this invariant system is not always achievable and depends upon how the states are characterized using matrix Lie groups.

In this section, a background on matrix Lie groups and Lie algebra is first described. The specific formulation of a Right InEKF with a Left InEKF measurement model for this drone state estimation problem is then detailed which is followed by a outline of the InEKF algorithm.

4.2 Background

A Lie group \mathcal{G} is a set of elements that for a operation combine any two elements to form a third element that also is in the set. The set of elements satisfy four conditions named closure, associativity, identity, and inevitability detailed in [Bar17]. A widely used group is the Special Orthogonal Group $SO(3)$ which represents rotations defined as [Bar17]

$$SO(3) = \{R \in \mathbb{R}^{3 \times 3} \mid RR^T = I, \det(R) = 1\} \quad (46)$$

The representation of pose can also be defined through the Special Euclidean Group which in the case of this paper will be the $SE_2(3)$ group defining the rotation matrix, velocity, and position as the entire pose

$$SE_2(3) = \left\{ X = \begin{bmatrix} R & v & p \\ 0_{3 \times 1} & 1 & 0 \\ 0_{3 \times 1} & 0 & 1 \end{bmatrix} \in \mathbb{R}^{5 \times 5} \mid R \in SO(3); v, p \in \mathbb{R}^3 \right\} \quad (47)$$

Each Lie group has its associated Lie algebra \mathfrak{g} which defines the tangent space of \mathcal{G} at the identity element of the group. Both \mathcal{G} and \mathfrak{g} have the same dimensions $n \times n$. A linear map is used to map vector element into the Lie algebra \mathfrak{g} of the group \mathcal{G} [HJEG19]

$$(\cdot)^\wedge : \mathbb{R}^{dim \mathfrak{g}} \rightarrow \mathfrak{g} \quad (48)$$

The Lie algebra associated with the $SE_2(3)$ is given as

$$\mathfrak{se}_2(3) = \left\{ \xi^\wedge = \begin{bmatrix} \xi^R \\ \xi^v \\ \xi^p \end{bmatrix}^\wedge = \begin{bmatrix} (\xi^R)_\times & \xi^v & \xi^p \\ 0_{3 \times 1} & 1 & 0 \\ 0_{3 \times 1} & 0 & 1 \end{bmatrix} \in \mathbb{R}^{5 \times 5} \mid \xi^R, \xi^v, \xi^p \in \mathbb{R}^3 \right\} \quad (49)$$

A mapping to the Lie group is done through the exponential map [HJEG19]

$$\text{Exp} : \mathbb{R}^{dim \mathfrak{g}} \rightarrow \mathcal{G} \quad (50)$$

Where $\text{Exp}(\cdot)$ is defined in terms of the matrix exponential $\exp_m(\cdot)$ as follows

$$\text{Exp}(\xi) = \exp_m(\xi^\wedge) \quad (51)$$

Note that in equation (50), the exponential $\text{Exp}(\cdot)$ is the matrix exponential $\exp_m(\cdot)$ of the Lie algebra element $(\xi)_\times$ which differs from the previous uses of $\exp(\cdot)$ used to take the exponential of a vector or scalar. Another key concept in Lie group theory is the adjoint representation, which is a linear map used to move vectors of the Lie algebra between tangent spaces of two group elements [HJEG19]

$$Ad_X : \mathfrak{g} \rightarrow \mathfrak{g} \quad (52)$$

$$Ad_X(\xi^\wedge) = X\xi^\wedge X^{-1} = \begin{bmatrix} R & \mathbf{0} & \mathbf{0} \\ (v)_\times & R & \mathbf{0} \\ (p)_\times & \mathbf{0} & R \end{bmatrix} \quad (53)$$

The process model can evolve on a Lie group, excluding the bias states, with the Lie group defined in equation (47) and the process dynamics defined as [HJEG19]

$$\frac{d}{dt}X = f_u(X) \quad (54)$$

Here the deterministic system dynamics are rotation matrix, velocity, and position propagation in equations (2b), (2c), and (2d). Two definitions of the state estimated error are possible depending

on the right or left multiplication of X^{-1} on to the estimated state \hat{X} . The left and right errors are defined with L being a random element of the group [HJEG19]

$$\eta^r = \hat{X}X^{-1} = (\hat{X}L)(XL)^{-1} \quad (\text{Right Error}) \quad (55a)$$

$$\eta^l = X^{-1}\hat{X} = (LX)^{-1}(L\hat{X}) \quad (\text{Left Error}) \quad (55b)$$

In both the QEKF and the InEKF, the right error corresponds to the world frame, while left error corresponds to the body frame. This distinction arises from the world-centric system dynamic equations used to model the IMU dynamics. This definition of left and right error is also applied to quaternions. The left error formulation is used when updating the nominal quaternion because the angular rate is measured in the body frame. The right error is used during the state update as the estimated angular error is in the world frame.

With the background provided so far, the following two important theorems from [BB14] provide the essential guarantees for the InEKF.

Theorem 1 (State Independent Error Dynamics [BB14]) *A system is said to be group affine if $f_t(\cdot)$ satisfies*

$$f_u(X_1X_2) = f_u(X_1)X_2 + X_1f_u(X_2) - X_1f_u(I)X_2 \quad (56)$$

for all time $t > 0$ and $X_1, X_2 \in G$. If this condition is satisfied, the right and left invariant errors are trajectory independent and satisfy the following differential equations:

$$\frac{d}{dt}\eta^r(t) = g_u(\eta_r(t)) = f_t(\eta^r(t)) + \eta^r(t)f_t(I_d), \quad (57)$$

$$\frac{d}{dt}\eta^l(t) = g_u(\eta_l(t)) = f_t(\eta^l(t)) + f_t(I_d)\eta^l(t). \quad (58)$$

Here $I_d \in \mathcal{G}$ and is the group identity element. This theorem defines state independent differential equations for the left and right errors making the error dynamics invariant subject to satisfying condition (56). This invariance of the estimation error means there will exist no dependence on the current state estimate removing all possible linearization errors. The right or left error can be linearized by the following equation with $A \in \mathbb{R}^{dim\mathfrak{g} \times dim\mathfrak{g}}$ [HJEG19]

$$g_u(\text{Exp}(\xi)) = (A\xi)^\wedge + \mathcal{O}(\|\xi\|^2) \quad (59)$$

For all $t \geq 0$, let ξ be the solution of the linear differential equation [HJEG19]

$$\frac{d}{dt}\xi = A\xi \quad (60)$$

Theorem 2 (Log-Linear Property of Error [BB14]) *Consider the right or left invariant error, η , between any two trajectory. For arbitrary initial error $\xi_o \in \mathbb{R}^{dim\mathfrak{g}}$, if $\eta_0 = \text{Exp}(\xi_o)$, then for all $t \geq 0$*

$$\eta = \text{Exp}(\xi) \quad (61)$$

Thus, the nonlinear estimation error η can be precisely recovered from the the time-varying linear differential equation (60)

Therefore these error dynamics allow the left or right error invariant error on the Lie group to be exactly recovered. This allows for the covariance to be exactly propagated in the case without any noise.

4.3 Process Model

The InEKF is based upon the theoretical background described in the previous section. However, the noise and biases must be considered in the formulation of the system dynamics. This results in decoupling the states into a state tuple. This state tuple consists of a Lie group defined for $SE_2(3)$ (47) and parameter vector for the biases. The error state dynamics are then defined through a linearization of the group error η^r . The choice of a Right InEKF model for propagation is used such that the error is in the world frame.

As determined in [BB14], there is no Lie group that can include the bias states and meets the group affine property (56). Therefore, an inexact InEKF is designed so that the bias states can be included. A parameter vector Θ can be estimated as part of the Right InEKF state [HJEG19]

$$\Theta = \begin{bmatrix} b^\omega \\ b^a \end{bmatrix} \in \mathbb{R}^6 \quad (62)$$

The state tuple model can now be written as the Lie group and parameter vector [HJEG19]

$$e^r = (\eta^r, \zeta) = (\hat{X}X^{-1}, \hat{\Theta} - \Theta) \in \mathbb{R}^{5 \times 5} \times \mathbb{R}^6 \quad (63)$$

Where the right Lie group error η^r is defined as

$$\eta^r = \hat{X}X^{-1} = \begin{bmatrix} \hat{R} & \hat{v} & \hat{p} \\ 0_{3 \times 1} & 1 & 0 \\ 0_{3 \times 1} & 0 & 1 \end{bmatrix} \begin{bmatrix} R^T & -R^T v & -R^T p \\ 0_{3 \times 1} & 1 & 0 \\ 0_{3 \times 1} & 0 & 1 \end{bmatrix} = \begin{bmatrix} \hat{R}R^T & \hat{v} - \hat{R}R^T v & \hat{p} - \hat{R}R^T p \\ 0_{3 \times 1} & 1 & 0 \\ 0_{3 \times 1} & 0 & 1 \end{bmatrix} \quad (64)$$

and the parameter error vector is

$$\zeta = \begin{bmatrix} \zeta^{b\omega} \\ \zeta^{ba} \end{bmatrix} = \begin{bmatrix} \hat{b}^\omega - b^\omega \\ \hat{b}^a - b^a \end{bmatrix} \quad (65)$$

With noise, Theorem 2 [2] no longer holds and the log of the invariant error, ξ , approximately satisfies the linear system for the right error dynamics [HJEG19]

$$\frac{d}{dt} \begin{bmatrix} \xi \\ \zeta \end{bmatrix} = A^r \begin{bmatrix} \xi \\ \zeta \end{bmatrix} + \begin{bmatrix} Ad_{\hat{X}} & 0_{9 \times 6} \\ 0_{6 \times 9} & I_6 \end{bmatrix} w = A^r \begin{bmatrix} \xi \\ \zeta \end{bmatrix} + Ad_{(\hat{X}, \hat{\Theta})} w \quad (66a)$$

$$\eta^r = \text{Exp}(\xi) \quad (66b)$$

Here the noise w is defined as

$$w = \begin{bmatrix} w^\omega \\ w^a \\ 0_{3 \times 1} \\ w^{b\omega} \\ w^{ba} \end{bmatrix} \quad (67)$$

Note that when incorporating the noise, the adjoint is used in the right error model. This is done because the noise is defined for the Lie algebra in the body frame. Therefore, because the right error model is in the world frame, the adjoint is used to move the body noise into the world frame.

In order to determine the continuous time state transition matrix A^r (66a), the error dynamics e^r (63) must be differentiated

$$\frac{d}{dt} e^r = \left(\frac{d}{dt} \eta^r, \begin{bmatrix} w^{b\omega} \\ w^{ba} \end{bmatrix} \right) \quad (68)$$

Once differentiated, the error dynamics of the Lie group must be linearized which utilizes the following first order approximation

$$\eta^r = \text{Exp}(\xi) \approx I_d + (\xi)^\wedge \quad (69)$$

Here I_d is the identity matrix with dimension d from the error $\xi \in \mathbb{R}^d$. This linearization is done each of Lie group errors. Using this approximation, the true values for the rotation is

$$\begin{aligned} \eta^R &= \hat{R}R^T \\ R^T &= \hat{R}^T \eta^R \\ R^T &\approx \hat{R}^T (I + (\xi^R)_\times) \\ R &= (I + (\xi^R)_\times)^T \hat{R} \end{aligned} \quad (70)$$

The true velocity can be approximated as

$$\begin{aligned}
\eta^v &= \hat{v} - \hat{R}R^T v \\
\xi^v &= \hat{v} - \hat{R}R^T v \\
v &= (\hat{R}R^T)^T (-\xi^v + \hat{v}) \\
&\approx (I + (\xi^R)_\times)^T (-\xi^v + \hat{v}) \\
&\approx -\xi^v + \hat{v} - \xi_\times^R \hat{v} \\
&= -\xi^v + \hat{v} + \hat{v}_\times \xi^R
\end{aligned} \tag{71}$$

Additionally, the true position is approximated as

$$\begin{aligned}
\eta^p &= \hat{p} - \hat{R}R^T p \\
\xi^p &= \hat{p} - \hat{R}R^T p \\
p &= (\hat{R}R^T)^T (-\xi^p + \hat{p}) \\
&\approx (I + (\xi^R)_\times)^T (-\xi^p + \hat{p}) \\
&\approx -\xi^p + \hat{p} - \xi_\times^R \hat{p} \\
&= -\xi^p + \hat{p} + \hat{p}_\times \xi^R
\end{aligned} \tag{72}$$

Using these approximations and neglecting second-order error terms, the linearization of the differentiated right Lie group error is possible. The resulting rotational error is

$$\begin{aligned}
\frac{d}{dt}\eta^R &= \frac{d}{dt}(\hat{R}R^T) \\
\frac{d}{dt}(I + \xi_\times^R) &\approx \frac{d}{dt}(\hat{R}R^T) \\
\frac{d}{dt}(\xi^R)_\times &= \dot{\hat{R}}R^T + \hat{R}\dot{R}^T \\
&= \hat{R}(\tilde{\omega} - \hat{b}^\omega)_\times R^T + \hat{R}(R(\tilde{\omega} - b^\omega - w^\omega)_\times)^T \\
&\approx \hat{R}(\tilde{\omega} - \hat{b}^\omega)_\times \hat{R}^T (I + (\xi^R)_\times) + \hat{R}[(I + (\xi^R)_\times)^T \hat{R}(\tilde{\omega} - b^\omega - w^\omega)_\times]^T \\
&= \hat{R}(\tilde{\omega} - \hat{b}^\omega)_\times \hat{R}^T + \hat{R}(\tilde{\omega} - \hat{b}^\omega)_\times \hat{R}^T (\xi^R)_\times - \hat{R}[(\tilde{\omega} - b^\omega - w^\omega)_\times](\hat{R}^T (I + (\xi^R)_\times))] \\
&\approx \hat{R}(\tilde{\omega} - \hat{b}^\omega)_\times \hat{R}^T - \hat{R}(\tilde{\omega} - b^\omega - w^\omega)_\times \hat{R}^T \\
&= \hat{R}(-\zeta^\omega + w^\omega)_\times \hat{R}^T \\
&= (\hat{R}(-\zeta^\omega + w^\omega))_\times \\
\frac{d}{dt}\xi^R &= (\hat{R}(-\zeta^\omega + w^\omega))_\times
\end{aligned} \tag{73}$$

Note that the following property is used when taking the transpose of vector that is skewed

$$(V)_\times^T = -V_\times \tag{74}$$

Here the vector V is defined as $V \in \mathbb{R}^3$. Additionally, the following property is used to reorganize the terms

$$R(V)_\times R^T = (RV)_\times \tag{75}$$

The derived velocity approximations and second-order terms can also be used to linearize the velocity

error

$$\begin{aligned}
\frac{d}{dt}\eta^v &= \frac{d}{dt}(\hat{v} - \hat{R}R^T v) \\
\frac{d}{dt}(I + \xi^v) &\approx \frac{d}{dt}(\hat{v} - \hat{R}R^T v) \\
\frac{d}{dt}\xi^v &= \dot{\hat{v}} - \dot{\hat{R}}R^T v - \hat{R}\dot{R}^T v - \hat{R}R^T \dot{v} \\
&= \hat{R}(\tilde{a} - \hat{b}^a)_\times + g - (\dot{\hat{R}}R^T + \hat{R}\dot{R}^T)v - \hat{R}R^T(R(\tilde{a} - b^a - w^a) + g) \\
&\approx \hat{R}(\tilde{a} - \hat{b}^a)_\times + g - \hat{R}(-\zeta^\omega + w^\omega)_\times v - \hat{R}((\tilde{a} - b^a - w^a) + R^T g) \\
&= (v)_\times \hat{R}(-\zeta^\omega + w^\omega) + g + \hat{R}(-\zeta^a + w^a) - \hat{R}R^T g \\
&\approx (-\xi^v + \hat{v} + \hat{v}_\times \xi^R)_\times \hat{R}(-\zeta^\omega + w^\omega) + \hat{R}(-\zeta^a + w^a) + g - (I + (\xi^R)_\times)g \\
&\approx (\hat{v}_\times) \hat{R}(-\zeta^\omega + w^\omega) + \hat{R}(-\zeta^a + w^a) - (\xi^R)_\times g \\
&= (g)_\times \xi^R + (\hat{v}_\times) \hat{R}(-\zeta^\omega + w^\omega) + \hat{R}(-\zeta^a + w^a)
\end{aligned} \tag{76}$$

Lastly, the position error is derived as follows

$$\begin{aligned}
\frac{d}{dt}\eta^p &= \frac{d}{dt}(\hat{p} - \hat{R}R^T p) \\
\frac{d}{dt}(I + \xi^p) &\approx \frac{d}{dt}(\hat{p} - \hat{R}R^T p) \\
\frac{d}{dt}\xi^p &= \dot{\hat{p}} - \dot{\hat{R}}R^T p - \hat{R}\dot{R}^T p - \hat{R}R^T \dot{p} \\
&\approx \hat{v} - (\dot{\hat{R}}R^T + \hat{R}\dot{R}^T)p - (I + \xi_\times^R)v \\
&\approx \hat{v} + (-\xi^p + \hat{p} + \hat{p}_\times \xi^R)_\times (\hat{R}(-\zeta^\omega + w^\omega)) - (I + \xi_\times^R)(-\xi^v + \hat{v} + \hat{v}_\times \xi^R)_\times \\
&\approx \xi^v + (\hat{p})_\times (\hat{R}(-\zeta^\omega + w^\omega)_\times)
\end{aligned} \tag{77}$$

With these linearization the right continuous time matrix A^r for the error dynamics can be defined for the error propagation equation (66a)

$$A^r = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & -\hat{R} & \mathbf{0} \\ (g)_\times & \mathbf{0} & \mathbf{0} & -(\hat{v})_\times \hat{R} & -\hat{R} \\ \mathbf{0} & I & \mathbf{0} & -(\hat{p})_\times \hat{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{78}$$

Note that with the inclusion of the biases in A^r means the matrix must be linearized about the current state for each time step. Without the biases this is not required and A^r is invariant for error propagation. In order to discretize A^r , a Euler approximation can be used which was also done for the quaternion Kalman Filter

$$\begin{aligned}
F^r &= \exp(A^r) \approx I_{15} + \Delta t A^r \\
&= \begin{bmatrix} I & \mathbf{0} & \mathbf{0} & -\Delta t \hat{R} & \mathbf{0} \\ \Delta t (g)_\times & I & \mathbf{0} & -\Delta t (\hat{v})_\times \hat{R} & -\Delta t \hat{R} \\ \mathbf{0} & \Delta t I & I & -\Delta t (\hat{p})_\times \hat{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & I & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & I \end{bmatrix}
\end{aligned} \tag{79}$$

The process covariance matrix Q is defined using the updated adjoint definition in equation (66a) along with the previously specified noise (67), now discretized (indicated by d) from table (1)

$$Q^r = Ad_{(\hat{X}, \hat{\Theta})} \text{Cov}(w_d) (Ad_{(\hat{X}, \hat{\Theta})})^T \tag{80}$$

For the propagation of the state variables, the input vector is defined as follows

$$u_k = \begin{bmatrix} \tilde{\omega}_k \\ \tilde{a}_k \end{bmatrix} \tag{81}$$

The process model $f_{\text{InEKF}}(X_{k-1}, \Theta_{k-1}, u_{k-1})$ is then used to propagate the states from step $k-1$ to k and is defined by the following equations

$$R_k = R_{k-1} \exp((\tilde{\omega}_{k-1} - b_{k-1}^\omega) \times \Delta t) \quad (82a)$$

$$v_k = v_{k-1} + g\Delta t + R_{k-1}(\tilde{a}_{k-1} - b_{k-1}^a) \left(I\Delta t + \frac{1}{2}\Delta t^2(\tilde{\omega}_{k-1} - b_{k-1}^\omega) \times \right) \quad (82b)$$

$$p_k = \bar{p}_{k-1} + \frac{1}{2}g\Delta t^2 + R_{k-1}(\tilde{a}_{k-1} - b_{k-1}^a) \left(I\Delta t^2 + \frac{1}{2}\Delta t^3(\tilde{\omega}_{k-1} - b_{k-1}^\omega) \times \right) \quad (82c)$$

$$b_k^\omega = b_{k-1}^\omega \quad (82d)$$

$$b_k^a = b_{k-1}^a \quad (82e)$$

It should be noted that when updating the rotation matrix it should be inspected to make sure the constraint $RR^T = I$ is true. One way of doing this is checking if $\det(R) = 1$. If the determinant is not approximately one, the matrix can be renormalized with the following method described in [Gre52]. Note that this method assumes the determinant is greater than zero which will be assumed in this algorithm implementation because the time step is small

$$R^+ = (RR^T)^{-\frac{1}{2}} R \quad (83)$$

4.4 Measurement Model

The GPS measurement is collected relative to a world frame. This makes the update for the InEKF more difficult because the Right InEKF measurement update requires a measurement in the body frame while propagating the state covariance in the world frame. Therefore, the Left InEKF is required for the measurement update. This can be accomplished by switching the Right InEKF covariance to the Left InEKF using the adjoint. After performing the update to the left covariance, the left covariance can then be switched back to the right covariance for propagation. The derivation for switching between the left and right error is as follows [HJEG19] using the original definition of the adjoint defined in this report (53)

$$\begin{aligned} \eta^r &= \hat{X}X^{-1} = \hat{X}\eta^l\hat{X}^{-1} \\ \text{Exp}(\xi^r) &= \hat{X}\text{Exp}(\xi^l)\hat{X}^{-1} = \text{Exp}(Ad_{\hat{X}}\xi^l) \\ \xi^r &= Ad_{\hat{X}}\xi^l \end{aligned} \quad (84)$$

The right covariance matrix conversion is then

$$\begin{aligned} P^r &= E[\xi^r(\xi^r)^T] \\ &= Ad_{\hat{X}}E[\xi^l(\xi^l)^T]Ad_{\hat{X}}^T \\ &= Ad_{\hat{X}}P^lAd_{\hat{X}}^T \end{aligned} \quad (85)$$

$$\begin{aligned} P^r &= Ad_{\hat{X}}P^lAd_{\hat{X}}^T \\ P^l &= Ad_{\hat{X}^{-1}}P^r(Ad_{\hat{X}^{-1}})^T \end{aligned} \quad (86)$$

Because biases are used in the error, the switching of the covariances can be accomplished with the following equations with the updated version of the adjoint $Ad_{(\hat{X}, \Theta)}$ (66a)

$$P^l = Ad_{(\hat{X}^{-1}, \hat{\Theta})}P^rAd_{(\hat{X}^{-1}, \hat{\Theta})}^T \quad (87a)$$

$$P^r = Ad_{(\hat{X}, \hat{\Theta})}P^lAd_{(\hat{X}, \hat{\Theta})}^T \quad (87b)$$

Note that this conversion is imperfect but without the bias this conversion is exact.

The left measurement model takes the following form [HJEG19]

$$\tilde{y}^l = \hat{X}b + \nu \quad (88)$$

In the Left InEKF, the measurement \tilde{y} and the constant vector b used to relate the Lie group to the measurement are defined as

$$\tilde{y} = \begin{bmatrix} \tilde{y}_p \\ 0 \\ 1 \end{bmatrix} \quad (89)$$

$$b = \begin{bmatrix} 0_{4 \times 1} \\ 1 \end{bmatrix} \quad (90)$$

The Kalman filter update uses the measurement Jacobian H . To relate the H to b and the Lie group state X , the following approximation can be done by rearranging equation (89) for the measurement innovation ν

$$\begin{aligned} \nu &= \hat{X}^{-1} \tilde{y} - b \\ &= \hat{X}^{-1} (Xb + \nu) - b \\ &= (\eta^l)^{-1} b + \hat{X}^{-1} \nu - b \\ &\approx (I + (\xi^l)_\times)^T b + \hat{X}^{-1} \nu - b \\ &= -(\xi^l)_\times b + \hat{X}^{-1} \nu \\ &\Rightarrow (\xi^l)_\times b + \hat{X}^{-1} \nu \triangleq H \xi^l + \hat{X}^{-1} \nu \end{aligned} \quad (91)$$

Note that the innovation can be reduced in dimensions by dropping the last two rows with zeros

$$\begin{aligned} \nu &= \hat{X}^{-1} \tilde{y} - b \\ &= \begin{bmatrix} \hat{R}^T & -\hat{R}^T \hat{v} & -\hat{R}^T \hat{p} \end{bmatrix} \begin{bmatrix} \tilde{y}_p \\ 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 0_{3 \times 1} \\ 0 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} -\hat{R}^T \tilde{y}_p \\ 0 \\ 0 \end{bmatrix} \\ &\triangleq -\hat{R}^T \tilde{y}_p \end{aligned} \quad (92)$$

Therefore, the innovation can be updated simplified by including a selection matrix [BB14] $\Pi = \begin{bmatrix} I & 0_{3 \times 2} \end{bmatrix}$

$$\nu = \hat{X}^{-1} \tilde{y} - b \triangleq \Pi \hat{X}^{-1} \tilde{y} \quad (93)$$

Inspecting equation (91), H can be defined as follows and then reduced in a similar manner as done for the innovation

$$H = \begin{bmatrix} 0_{3 \times 6} & I & 0_{3 \times 6} \\ & 0_{4 \times 15} & \end{bmatrix} \triangleq \begin{bmatrix} 0_{3 \times 6} & I & 0_{3 \times 6} \end{bmatrix} \quad (94)$$

The dimensions of the measurement noise matrix N can be also be reduced. For the left invariant form of the measurement noise matrix can be simplified to be $\mathbb{R}^{3 \times 3}$ matrix. The left invariant error is defined as follows [HJEG19]

$$V^l = \hat{R}^{-1} \text{cov}(\nu) \hat{R} = \hat{R}^T \sigma_{\tilde{y}_p}^2 \hat{R} \quad (95)$$

During the correction step of the Kalman filter, distinctions must be made between the Lie group states X and the bias states Θ in the state tuple. Once the Kalman gain is calculated, the gain can be split up into gain for the Lie group $K^\xi \in \mathbb{R}^{9 \times 3}$ and bias states $K^\zeta \in \mathbb{R}^{6 \times 3}$

$$K = \begin{bmatrix} K^\xi \\ K^\zeta \end{bmatrix} \quad (96)$$

Using these gains the state tuple can be updated. The typical update for vectors can be used for the bias states. However, in order to update the Lie group X , the error state ξ must be mapped to the Lie group. This can be done using the exponential map and skew operator as shown in equations (50) and (51). Once mapped, the Lie group X can be updated through matrix multiplication

$$\left(\hat{X}_k^+, \hat{\Theta}_k^+ \right) = \left(\hat{X}_k \text{Exp} \left(K_k^\xi \left(\Pi \hat{X}_k^{-1} \tilde{y}_k \right) \right), \hat{\Theta}_k + K_k^\zeta \left(\Pi \hat{X}_k^{-1} \tilde{y}_k \right) \right) \quad (97)$$

4.5 Initialization of Covariance

The covariance for the QEKF and the InEKF filters do not represent the same underlying distribution because the error defined in the InEKF is defined for the Lie algebra. Therefore, a conversion between the covariances is required so that the initial covariances are at least approximately equal. Furthermore, this conversion can be used to compare the uncertainties estimated in the states between the filters, however, in this report this conversion will only be used for initialization.

The attitude error for the quaternion $\delta\theta$, defined in the world frame, is equivalent to the rotation error ξ^R for the Right InEKF. The velocity and position errors are not equivalent. The following first order conversions from the quaternion error to the Right InEKF error are as follows [HJEG19]

$$\begin{aligned}\eta^v &= \hat{v} - \hat{R}R^T v \\ \xi^v &\approx \hat{v} - (I + (\xi^R)_\times)v \\ \xi^v &= \hat{v} - v - (\xi^R)_\times v \\ \xi^v &= -\delta v + (v)_\times \xi^R\end{aligned}\tag{98}$$

$$\begin{aligned}\eta^p &= \hat{p} - \hat{R}R^T p \\ \xi^p &\approx \hat{p} - (I + (\xi^R)_\times)p \\ \xi^p &= \hat{p} - p - (\xi^R)_\times p \\ \xi^p &= -\delta p + (p)_\times \xi^R\end{aligned}\tag{99}$$

Using equation (98), the initial covariance for the QEKF can be used to approximate the initial covariance for the InEKF for the velocity

$$\begin{aligned}\mathbb{E}[\xi_0^v (\xi_0^v)^T] &\approx \mathbb{E}[(-\delta v_0 + (v_0)_\times \xi_0^R)(-\delta v_0 + (v_0)_\times \xi_0^R)^T] \\ &= \mathbb{E}[\delta v_0 \delta v_0^T - \delta v_0 (\xi_0^R)^T (v_0)_\times^T - (v_0)_\times \xi_0^R \delta v_0^T + (v_0)_\times \xi_0^R (\xi_0^R)^T (v_0)_\times^T] \\ &= \mathbb{E}[\delta v_0 \delta v_0^T] + (v_0)_\times \mathbb{E}[\xi_0^R (\xi_0^R)^T] (v_0)_\times^T \\ P_{\xi_0^v} &= P_{\delta v_0} + (v_0)_\times P_{\xi_0^R} (v_0)_\times^T\end{aligned}\tag{100}$$

Note that the errors are assumed to be independent and have a mean of zero. The proof is essentially the same for the initial position covariance. The conversion for position is the following

$$P_{\xi_0^p} = P_{\delta p_0} + (p_0)_\times P_{\xi_0^R} (p_0)_\times^T\tag{101}$$

4.6 Algorithm

The Invariant Extended Kalman filter is summarized with the following pseudo code in the algorithm below. The initial state and covariance are first defined where the covariance uses conversions in the previous section. If IMU measurements are available the state and covariance are propagated. If a GPS measurement is received, the algorithm uniquely has to switch from the right to left covariance. The update steps for the left measurement model are then followed ending with a switch back to the right covariance so that the uncertainty in the error states is in the world frame. Additionally, the rotation matrix R is checked to ensure the $SO(3)$ constraint is maintained.

Algorithm 2: Invariant EKF

```
 $P_0^r \xleftarrow{\text{Convert}} \Sigma_0; // \text{ See equations (100) and (101)}$ 
 $(\hat{X}_0, \hat{\Theta}_0) = x_0; // \text{ See equations (47) and (62)}$ 
while receiving measurements do
  if IMU measurement then
     $F_k = I_{15} + \Delta t A^r|_{\hat{X}_{k-1}};$ 
     $(\hat{X}_k, \hat{\Theta}_k) = f_{\text{InEKF}}(\hat{X}_{k-1}, \hat{\Theta}_{k-1}, u_{k-1}); // \text{ See equations in (82)}$ 
    if  $|1 - \det(R)| > 10^{-5}$  then
       $R_k \leftarrow (R_k R_k^T)^{-\frac{1}{2}} R_k;$ 
    end
     $Q_k^r = Ad_{(\hat{X}_k, \hat{\Theta}_k)} \text{Cov}(w_d) (Ad_{(\hat{X}_k, \hat{\Theta}_k)})^T;$ 
     $P_k^r = F_k^r P_{k-1}^r (F_k^r)^T + Q_k^r;$ 
  end
  if GPS measurement then
     $P_k^l = Ad_{(\hat{X}_k^{-1}, \hat{\Theta}_k)} P_k^r Ad_{(\hat{X}_k^{-1}, \hat{\Theta}_k)}^T;$ 
     $V_k^l = \hat{R}_k^T \sigma_{\tilde{y}_p}^2 \hat{R}_k;$ 
     $S = H^l P_k^l (H^l)^T + V_k^l;$ 
     $(K^\xi, K^\zeta) = K = P_k^l (H^l)^T S^{-1};$ 
     $\hat{X}_k^+ = \hat{X}_k \text{Exp}(K_k^\xi (\Pi \hat{X}_k^{-1} \tilde{y}_k));$ 
     $\hat{\Theta}_k^+ = \hat{\Theta}_k + K_k^\zeta (\Pi \hat{X}_k^{-1} \tilde{y}_k);$ 
    if  $|1 - \det(R)| > 10^{-5}$  then
       $R_k^+ \leftarrow (R_k^+ R_k^{+T})^{-\frac{1}{2}} R_k^+;$ 
    end
     $P_k^{l+} = (I - KH) P_k^l (I - KH)^T + K N^l K^T;$ 
     $P_k^{r+} = Ad_{(\hat{X}_k^+, \hat{\Theta}_k^+)} P_k^{l+} Ad_{(\hat{X}_k^+, \hat{\Theta}_k^+)}^T;$ 
  end
end
```

5 Simulation

5.1 Purpose

The purpose of this simulation is to compare the performance of the QEKF and InEKF filters in terms of error reduction between the true and estimated state. The primary difference between these two filters is in the approach of representing rotation between different coordinate frames. In the more traditional QEKF, rotation action between states is achieved through quaternions. The InEKF utilizes a Lie group structure to represent key states. By running simulations with both filters, it can be determined whether the new Lie group-based InEKF offers advantages as compared to the quaternion based QEKF.

5.2 Setup

To run this comparison simulation, python scripts were created and truth trajectory data from the Mid-Air Dataset was utilized [FD19]. Position, velocity, acceleration, attitude (defined as a quaternion), and angular velocity data were all sampled at a frequency of 100 Hz. For GPS measurements, the data was sampled at a rate of 1 Hz. Noisy IMU and GPS data were then simulated using the truth data, equations defined for the IMU model (1) and GPS model (9), and the standard deviations defined in table (1). Note that standard deviations in table (1) simulate a fairly noisy IMU. Note also that the random walk states are multiplied by an additional $\frac{1}{s}$ because in the error update equations the noise is multiplied by Δt .

With the measurement data generated, a Monte Carlo simulation of 50 trials was run for each filter using the same flight trajectory. In each trial, the noise statistics and initial covariance were

Measurement	Noise Standard Deviation	Units
Accelerometer White Noise	$\sigma_a = 5e - 2$	$\frac{m}{s^2} \frac{1}{\sqrt{Hz}}$
Accelerometer Random Walk	$\sigma_{ba} = 2e - 3$	$\frac{m}{s^3} \frac{1}{\sqrt{Hz}}$
Gyroscope White Noise	$\sigma_\omega = 8e - 3$	$\frac{rad}{s} \frac{1}{\sqrt{Hz}}$
Gyroscope Random Walk	$\sigma_{ba} = 1e - 4$	$\frac{rad}{s^2} \frac{1}{\sqrt{Hz}}$
GPS White Noise	$\sigma_{\tilde{y}_p} = 1$	m

Table 1: Measurement Noise Statistics

constant. The initial condition of each of the states were then randomized over a uniform distribution. The randomized initial states are defined below in table (2). Note that once the initial attitude is randomized it is converted into a initial quaternion for the QEKF and the quaternion is converted into a initial rotation matrix for the InEKF.

Initial State	Randomization	Units
Position	$\mathcal{U}(p_0, 10)$	m
Velocity	$\mathcal{U}(v_0, 2)$	$\frac{m}{s}$
Attitude	$\mathcal{U}(\theta_0, 15)$	deg
Accelerometer Bias	$\mathcal{U}(b_0^a, 1e - 5)$	$\frac{m}{s^3}$
Gyroscope Bias	$\mathcal{U}(b_0^\omega, 1e - 6)$	$\frac{rad}{s^2}$

Table 2: Initial States Randomization using the Uniform Distribution

5.3 Results

The position, velocity, and attitude states were tracked across each Monte Carlo for comparison between the QEKF and InEKF filters. The truth was plotted in black against the fifty different Monte Carlo runs. Additionally, the mean and max error across the Monte Carlo trials were computed and saved in table (3).

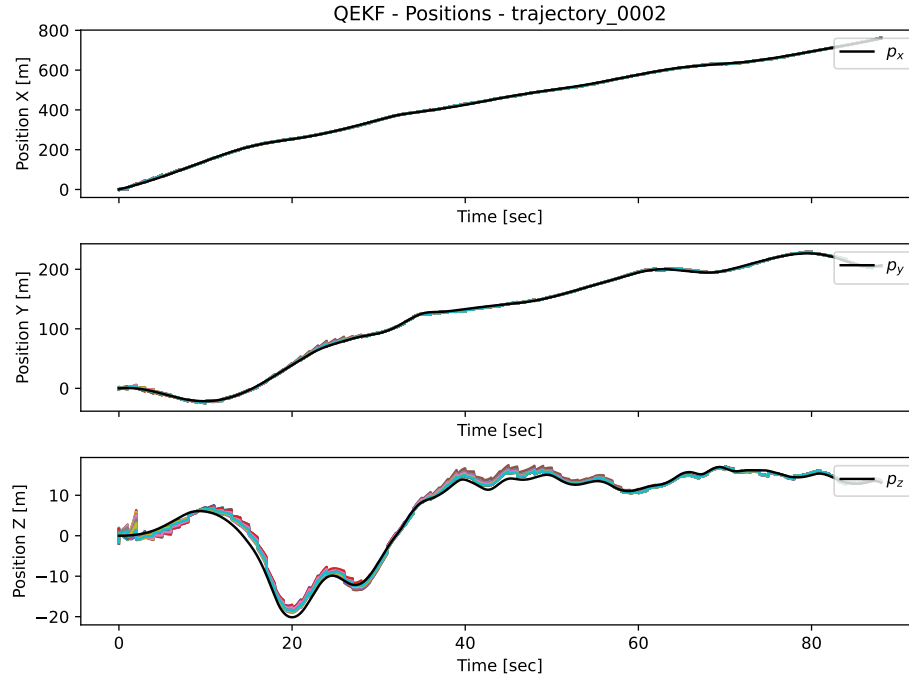


Figure 1: QEKF Position Monte Carlo Trials

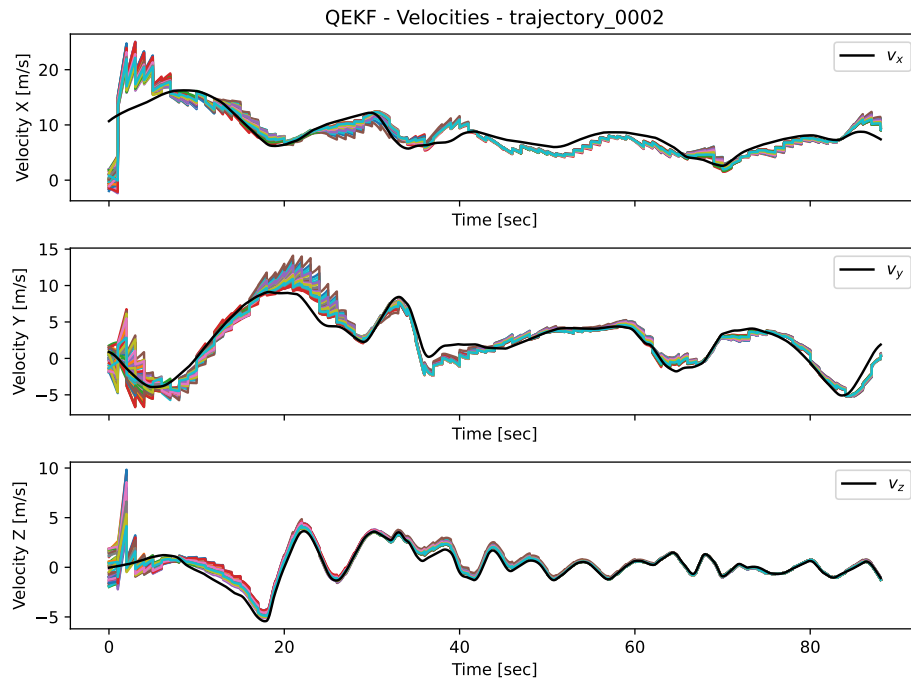


Figure 2: QEKF Velocity Monte Carlo Trials

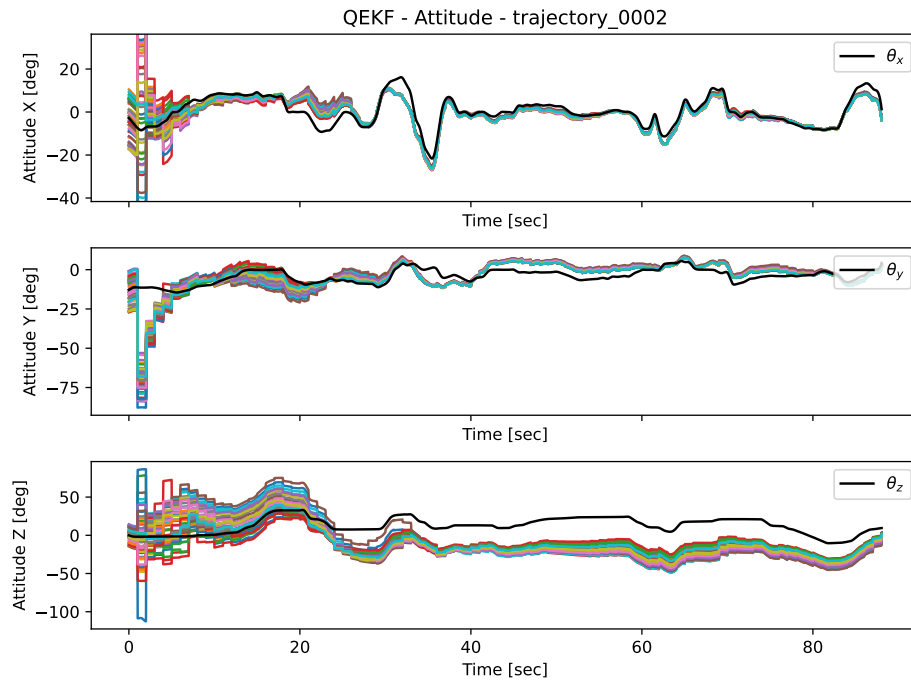


Figure 3: QEKF Attitude Monte Carlo Trials

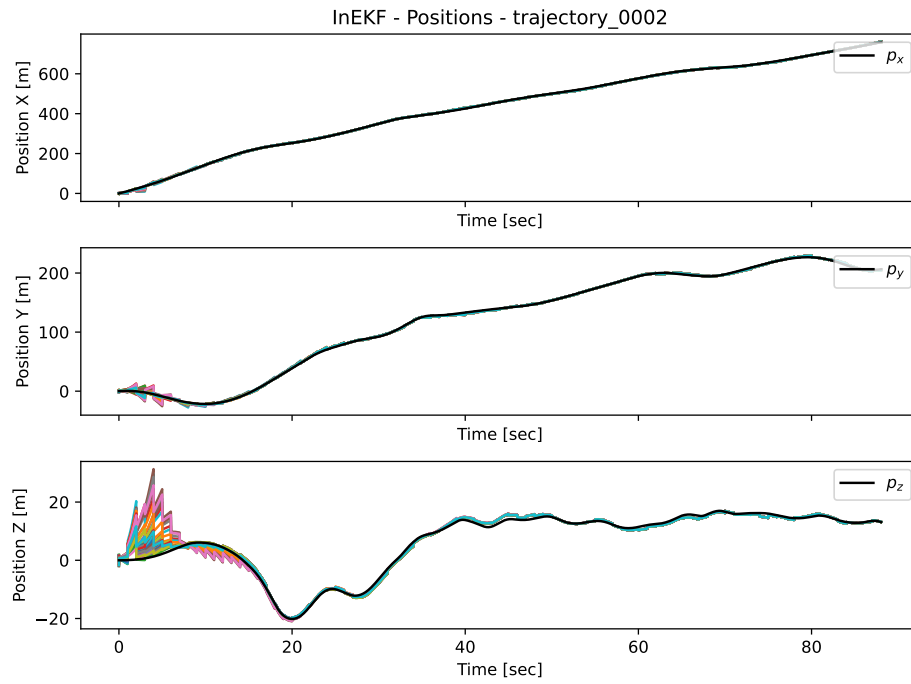


Figure 4: InEKF Position Monte Carlo Trials

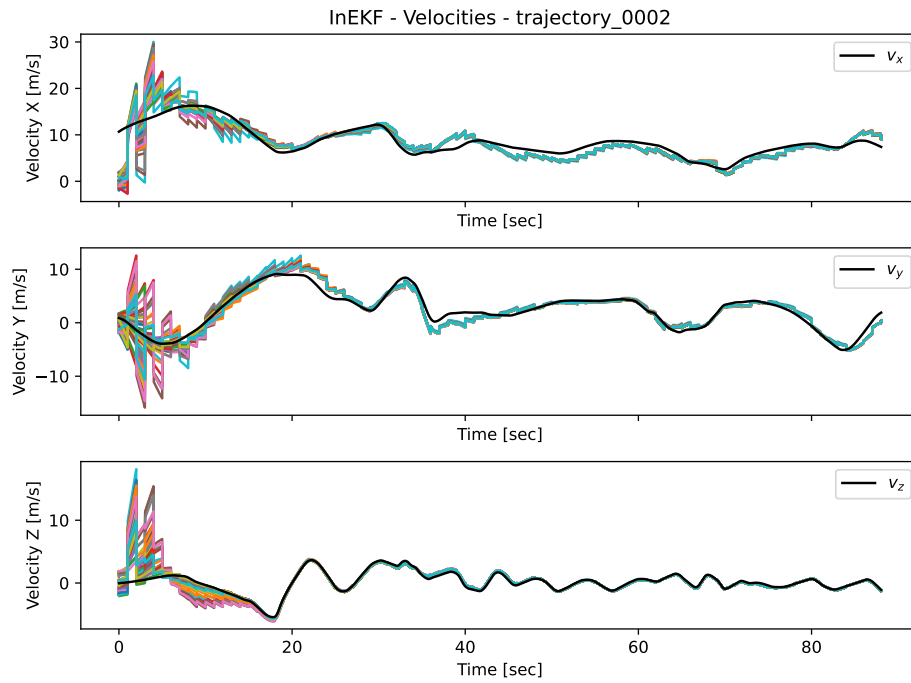


Figure 5: InEKF Velocity Monte Carlo Trials

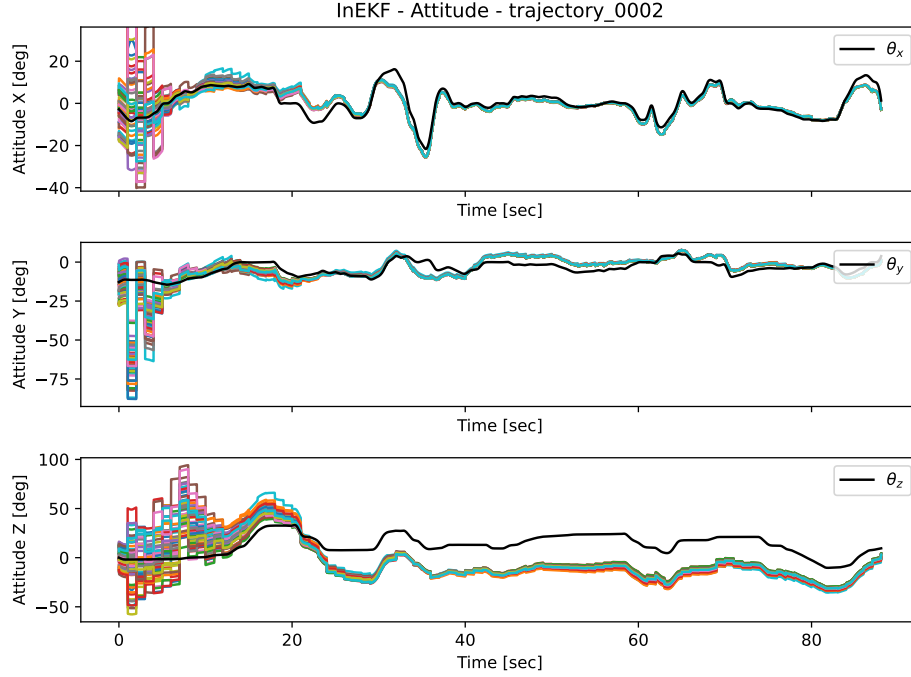


Figure 6: InEKF Attitude Monte Carlo Trials

States	Error Metrics				Units
	Mean QEKF	Mean InEKF	Max QEKF	Max InEKF	
Position X	1.2989	1.2205	9.2388	9.2800	m
Position Y	0.8957	0.8579	4.5489	3.8393	m
Position Z	0.8720	0.8396	3.6690	11.6386	m
Velocity X	1.4704	1.3089	10.8342	10.8531	$\frac{m}{s}$
Velocity Y	0.8594	0.7439	3.7589	3.1248	$\frac{m}{s}$
Velocity Z	0.3489	0.3536	5.8320	10.3547	$\frac{m}{s}$
Attitude X	2.6254	2.2281	19.5816	18.0167	deg
Attitude Y	4.5363	4.0555	59.7614	52.4789	deg
Attitude Z	26.9152	23.6016	43.5164	53.0185	deg

Table 3: Monte Carlo Error Metrics for QEKF and InEKF Filters

5.4 Discussion

Using just the GPS position measurement, it is clear that both QEKF and InEKF suffer from unobservability. This affects the bias states (not shown) and causes certain states such as the attitude in the Z direction to incur large errors. By adding more measurements, observability can be improved and this issues can be reduced.

Despite this issue, the QEKF and InEKF results can still be compared. In the InEKF plots, it is clear that quicker convergence of the states to a solution does occur which was also found in [HJEG19] and [PNM21]. This can be most clearly seen the in the velocity and attitude plots. This is due to the InEKF linearization being more accurate and being less sensitive to the initial errors as compared to the QEKF. Comparing the mean errors, the InEKF performs slightly better than the QEKF in all states besides in the Z velocity. This provides stronger evidence that the Lie group representation enhances performance by offering a more accurate linearization and by coupling rotation, velocity, and position. Unlike the QEKF, which treats these components independently, the InEKF's approach enables a more precise representation of the system's dynamics. Overall, the differences between the two filters were minor when using only GPS measurements. However, initial results suggest that the

InEKF may hold an advantage over the QEKF. Future work incorporating additional measurements into the system will help clarify the extent of this advantage.

6 Conclusion

6.1 Summary

This report explored how the Quaternion Extended Kalman Filter and Invariant Extended Kalman Filter compared in a drone state estimation simulation. Detailed background was provided for the IMU and GPS measurement models that were used to estimate the states of the drone. Background was then provided for both filters explaining the derivation of each filter for the estimation problem. A Monte Carlo simulation was then conducted for each filter. From the results, marginal improvements were seen in the InEKF but both filters suffered from unobservability in the bias states. Future work in adding additional measurements must be the next step in helping improve the observability of the system and the comparison between the two filters.

6.2 Future Work

There are multiple different ways this report could be improved in the future. The key improvements I would like to see worked are the following:

- Including a observability analysis for QEKF and InEKF to demonstrate which states are not observable.
- Adding new measurements to the current system. Currently, only GPS position is used to help correct the IMU measurements. Additional measurements could come from a magnetometer, barometer, and vision. For vision, the Mid-Air Dataset [FD19] provides various measurements such as stereo RGB pictures.
- Comparing body estimates of states could help give more insight into how the two filters compare. Currently, only the world frame estimate of the states are calculated.
- Compare uncertainty estimates for each of the filters.
- Apply these filters to an actual drone with a IMU and GPS receiver
- Perform additional Monte Carlo trials that include using different trajectories for the drone
- Give a example of the InEKF using a simple system to demonstrate the possible invariant properties and show what the covariance in the Lie group vs. Lie algebra looks like
- Include a tradition Kalman Filter that propagates the Euler angles without a quaternion or rotation matrix representation
- Create a C++ implementation of the filters
- Compare the computational time required to run each filter
- Include more background on the Lie groups and Lie algebra including visuals
- Add some of the missing proofs that are mentioned in the background sections for each filter

References

- [Bar17] Timothy Barfoot. *State Estimation for Robotics*. 08 2017.
- [BB14] Axel Barrau and Silvère Bonnabel. The invariant extended kalman filter as a stable observer. *CoRR*, abs/1410.1465, 2014.
- [BC21] José Luis Blanco-Claraco. A tutorial on se3 transformation parameterizations and on-manifold optimization. *arXiv preprint arXiv:2103.15980*, 2021.
- [FD19] Michael Fonder and Marc Van Droogenbroeck. Mid-air: A multi-modal dataset for extremely low altitude drone flights. In *Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, June 2019.
- [Gre52] Bert F. Green. The orthogonal approximation of an oblique structure in factor analysis. *Psychometrika*, 17(4):429–440, 1952.
- [Hen77] D. M. Henderson. Euler angles, quaternions, and transformation matrices for space shuttle analysis. *Legacy CDMS*, page 39, 1977.
- [HJEG19] Ross Hartley, Maani Ghaffari Jadidi, Ryan M. Eustice, and Jessy W. Grizzle. Contact-aided invariant extended kalman filtering for robot state estimation. *CoRR*, abs/1904.09251, 2019.
- [PNM21] Easton R. Potokar, Kalin Norman, and Joshua G. Mangelson. Invariant extended kalman filtering for underwater navigation. *IEEE Robotics and Automation Letters*, 6(3):5792–5799, 2021.
- [PTKXM16] Sayed Pouria Talebi, Sithan Kanna, Yili Xia, and Danilo P. Mandic. A distributed quaternion kalman filter with applications to fly-by-wire systems. In *2016 IEEE International Conference on Digital Signal Processing (DSP)*, pages 30–34, 2016.
- [Sol17] Joan Solà. Quaternion kinematics for the error-state kalman filter. *CoRR*, abs/1711.02508, 2017.