Extended $\mathcal{H}_2$ Filtering for Attitude Estimation in Low Power Microprocessors

Sunsoo Kim∗
Texas A&M University, Electrical & Computer Engineering, College Station, TX, 77840

Vaishnav Tadiparthi † and Raktim Bhattacharya ‡
Texas A&M University, Aerospace Engineering, College Station, TX, 77840

Accurate state estimation using low-cost MEMS (Micro Electro-Mechanical Systems) sensors present on Commercial-off-the-shelf (COTS) drones is a challenging problem. Most UAV systems use a combination of a gyroscope, an accelerometer, and a magnetometer to obtain measurements and estimate attitude. Under this paradigm of sensor fusion, the Extended Kalman Filter (EKF) is the most popular algorithm for attitude estimation in UAVs. In this work, we propose a novel estimation technique called extended $\mathcal{H}_2$ filter that can overcome the limitations of the EKF, specifically with respect to computational speed, memory usage, and root mean squared error. We formulate our attitude-estimation algorithm using two distinct coordinate representations for the vehicle’s orientation: Euler angles and unit quaternions, each with its own sets of benefits and challenges. The $\mathcal{H}_2$ optimal filter gain is designed offline about a nominal operating point by solving a convex optimization problem, and the filter dynamics is implemented using the nonlinear system dynamics. This implementation of this $\mathcal{H}_2$ optimal estimator is referred as the extended $\mathcal{H}_2$ estimator. The proposed technique is tested on four cases corresponding to long time-scale motion, fast time-scale motion, transition from hover to forward flight for VTOL aircrafts, and an entire flight cycle (from take-off to landing). Its results are compared against that of the EKF in terms of the aforementioned performance metrics.

I. Introduction

In recent years, unmanned aerial vehicles (UAVs) have found applications in many diverse fields encompassing commercial, civil, and military sectors [1–3]. Moreover, as the sizes and costs associated with building UAVs keeps decreasing, the need for computationally efficient flight controllers is growing rapidly [4]. Attitude estimation algorithms that leverage the characteristics of the various low-cost sensors on board is key to developing such powerful control

∗Graduate Student and AIAA Student Member
†Graduate Student and AIAA Student Member
‡Associate Professor, and AIAA Associate Fellow
While translational state information can be recovered easily from sensor data, states pertaining to orientation cannot be directly obtained from the same. In that context, sensor fusing algorithms are typically employed to estimate the attitude/orientation of these vehicles. In the recent past, MEMS (Micro-Electro Mechanical Systems) sensors like MARG (Magnetic, Angular Rate, and Gravity) sensor and IMU (Inertial Measurement Unit) have become increasingly common because of their low costs and small sizes. A three-axis gyroscope, a three-axis accelerometer, and a three-axis magnetometer are used to measure the angular rate of the vehicle, the acceleration of the vehicle, and the magnetic force vector respectively. It is important to note that measurements from MEMS sensor are corrupted by noise and bias. Additionally, rapid movements and magnetic disturbances can temporarily influence the attitude calculations.

Attitude can be computed independently from each of the sensors present in MARG and IMU. Integrating the angular rate obtained from gyroscope outputs can help determine the attitude of the vehicle. However, due to the presence of noise and bias in gyroscope measurements, errors in estimation build up over long periods of time. However, passing accelerometer measurements through a low pass filter can improve the tilt angle estimated by comparing the measured gravitational acceleration against that in the inertial frame. Although this computation is quite accurate, we cannot determine the heading of the vehicle since the gravity vector is aligned to the z-axis of the inertial frame. In aircraft navigation systems, a magnetometer measuring the magnetic field can be used to estimate the orientation as well, but is susceptible to magnetic interference. Thus, a filtering framework that incorporates sensor fusion is essential to obtaining a reliably accurate estimate of the vehicle’s attitude.

To perform attitude estimation, we can choose which coordinates to use to represent the orientation of the vehicle. The most popular representation is quaternions, proven to be numerically stable and efficient. These are followed by Euler angles, arguably the most understandable to a human controller. Both representations have pros and cons in the context of attitude estimation, and will be discussed in further detail in §II. Nonlinear sensor models with these two different representations have been developed for estimation using the Allen variance model. The gyroscope dynamics model is developed for use as system equations while the accelerometer and magnetometer models are used as measurement equations under the filtering paradigm. However, the unique nature of the quaternion vector prohibits a direct application of the general estimation algorithm. Hence, an error dynamics model of sensors is derived to estimate the error quaternion.

A number of nonlinear sensor fusing algorithms have been proposed for attitude estimation, especially for the flying vehicle. Among these methods, computationally intensive algorithms like the unscented filter and particle filter etc. are not considered in this study, since one of our primary goals is to reduce computation time for low-cost embedded processors. The most widely used filter is the Extended Kalman Filter (EKF). EKF predicts attitude with the gyroscope data and updates the prediction with the measurement from accelerometer and magnetometer in a way that minimizes the mean square error. This estimation is very accurate and widely used in practical scenarios.
particularly on open-source autopilot softwares like Ardupilot and PX4. However, the Extended Kalman filter has a few limitations of its own. Firstly, it can be complicated to implement which is reflected by the numerous solutions [32][33]. Secondly, determining Kalman gain after every time interval requires two steps: prediction and update, thus requiring more computations to calculate mean and covariance, and larger memory to store the results. Finally, the EKF scheme also assumes Gaussian uncertainty in its modeling, which is reasonable for uncertainty propagation over short intervals of time, but requires the algorithm to run at a higher rate resulting in larger processor usage. These aspects make it difficult to implement EKF in low power microprocessors.

In this paper, we propose an extended $H_2$ optimal estimator that can tackle the aforementioned limitations of the extended Kalman filter, and is capable of using either coordinate representation, Euler angles or quaternions. Our primary contribution focuses on the update step during estimation. The optimal filter gain for the update is derived offline about a nominal point by solving a convex optimization problem. This offline process avoids the need to solve the associated Riccati equation in real-time, however, at the cost of reduced performance. For attitude estimation using an Euler angle representation, filtering occurs in a single step, since it is solved with equations that include both prediction and update steps at once. This is essentially why we observed in our simulations that estimation using Euler angles is superior in computational speed when compared to quaternions. For the case involving quaternions, extended $H_2$ estimation is applied on error state dynamics to compute the error between predicted quaternion from gyro dynamics and the unknown final quaternion. This error is now multiplied with the prediction from gyroscope dynamics in accordance with quaternion algebra to obtain the final estimate.

The paper is organized as follows. In section §II, we present the choices for attitude representation in various scenarios. This is followed by details of the measurement model for the three sensors, i.e., the gyroscope, the accelerometer, and the magnetometer. In section §IV we outline the estimation frameworks for both Euler Angles and quaternions. Next, we describe the conventional $H_2$ optimal attitude estimation algorithm. We elaborate upon our proposal for the extended $H_2$ attitude estimation algorithm in section §VI. The changes that need to be addressed for the proposed framework to allow a quaternion representation are also discussed in this section. Proceeding further, we apply the proposed filter on an illustrative example using a commercially available sensor and compare our performance with that of the popularly used EKF-based estimator. The paper concludes with a few final remarks and potential directions for further investigation. We have also attached an appendix with some preliminary background on the requisite kinematics involved and the derivation of the error measurement equation.

II. Attitude Representation

The Newton Euler approach is used to present the dynamics of a rigid body system in this paper [34]. Therefore, the dynamics of the vehicle is expressed in the inertial frame($I$) and Body frame($B$) [35]. The axes orientation of the vehicle’s Body Frame with $B_x$, $B_y$, and $B_z$ is specified with respect to the Inertia Frame with $I_x$, $I_y$, and $I_z$. In the body
frame, the $B_x$ and $B_y$ axes point to the forward directions (heading of vehicle) and rightward (starboard), respectively while the $B_z$ axis points downwards. The $I_x$ and $I_y$ axes point to the North and East parallel to the earth’s surface respectively. The $I_z$ axis points down to earth. This axes’ frame is commonly referred to as NED for North, East, and Down.

The attitude or the orientation of the vehicle can be expressed in one of several different coordinate representations \cite{36}. Euler angles and quaternions are two of many such popular choices \cite{37}. In our case, the Euler angles are composed of 3-2-1 rotation with $\phi, \theta, \psi$ angles, defining the transformation between the inertial frame ($I$) to body frame ($B$):

$$\Phi := \begin{pmatrix} \phi & \theta & \psi \end{pmatrix}^T \in \mathbb{R}^3$$

(1)

The rotation matrix $C_B^I (\Phi) \in SO(3)$ represents the orientation of the Body frame($B$) relative to a fixed inertial frame($I$). $C_B^I (\Phi)$ is the DCM (Direction Cosine Matrix) with $3 – 2 – 1$ sequence from the inertial frame ($I$) to body frame ($B$) as:

$$C_B^I (\Phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \times \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \times \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(2)

Quaternions represent another way of describing the orientation of an aerial vehicle. This type of representation is typically used as an alternative for modeling attitude dynamics whilst avoiding the singularities inherent in other parameter sets like the Euler angles and Rodrigues parameters. It is based on a four parameter representation that can be defined globally, i.e., it doesn’t face any. A unit-norm quaternion, which defines the rotation between the inertial frame ($I$) to body frame ($B$), is defined as:

$$\bar{q} =_I^B \bar{q} = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix}^T \in \mathbb{R}^4$$

(3)

where $q_4$ and $q$ are the scalar and the vector parts of the quaternion, respectively. The quaternion of rotation is a unit quaternion, i.e. norm of quaternion is 1 as

$$|\bar{q}| = \sqrt{\bar{q}^T \bar{q}} = \sqrt{q_1^2 + q_2^2 + q_3^2} = 1$$

(4)
In this notation, $k$ describes the unit vector along Euler’s principal axis and $\theta$ defines the principal angle of rotation about that axis. The rotation matrix $C^B_I(q_1, q_2, q_3, q_4) \in SO(3)$ represents the orientation of the body frame ($B$) relative to a fixed inertial frame ($I$). $C^B_I(\bar{q})$ is the DCM (Direction Cosine Matrix) with $3−2−1$ sequence from the inertial frame ($I$) to body frame ($B$) as:

$$C^B_I(\bar{q}) = \begin{bmatrix}
q_1^2-q_2^2-q_3^2+q_4^2 & 2(q_1q_2+q_3q_4) & 2(q_1q_3-q_2q_4) \\
2(q_1q_2-q_3q_4) & -q_1^2+q_2^2-q_3^2+q_4^2 & 2(q_2q_3+q_1q_4) \\
2(q_1q_3+q_2q_4) & 2(q_2q_3-q_1q_4) & -q_1^2-q_2^2+q_3^2+q_4^2
\end{bmatrix}$$

(6)

For more details about quaternion algebra, refer to [38].

Each of these representations has its own advantages and disadvantages. Euler angles have the well known gimbal-lock problem [36] but are fairly intuitive, i.e., they are easy to visualize and are a minimal representation of attitude. On the other hand, quaternions are not as naturally understood and involve over-parameterization by using 4 quantities to express three angles. However, more importantly, they avoid singularities. In this paper, we consider both representations for attitude estimation.

### III. Sensor Measurement Model

For small UAV systems, low-cost MEMS sensors are typically used. Data collected from cheap sensors on such drones tends to be corrupted by noise and bias.

Following the Allan variance analysis [23–25], sensor models are described in this section. During estimation, gyroscope model will be used for state prediction, whereas the accelerometer and magnetometer models will be used for state update.

#### A. Gyroscope Model

Gyroscope sensor measurements are modeled as:

$$^B\omega = \omega_m - b - n_\omega,$$

(7a)

$$\dot{b} = n_b,$$

(7b)

where $\omega$ is the true angular rate of the body, $\omega_m$ is angular rate measured by the gyroscope, $b$ is the bias of gyroscope, $n_\omega$ is gyroscope sensor noise, and $n_b$ represents gyroscope bias noise. In this paper, the gyroscope bias is non-static.
and we model it as a random walk process.

**B. Accelerometer Model**

Accelerometer sensor measurements can be formulated as:

\[ B_a = a_m - n_a \]  

(8)

with \( B_a \): the true sum of the gravity and external acceleration of the body, \( a_m \): the sum of the gravity and external acceleration of the body measured by accelerometer, and \( n_a \): accelerometer sensor noise. The external acceleration of the vehicle is derived from position estimation and subtracted from \( B_a \) to obtain acceleration due to gravity. In the context of attitude estimation, it will initially be assumed that an accelerometer will measure only gravity.

**C. Magnetometer Model**

Magnetometer sensor measurements can be formulated as:

\[ B_m = m_m - n_m \]  

(9)

with the true magnetic field \( B_m \), the magnetic field measured by the magnetometer \( m_m \), and magnetometer sensor noise \( n_m \).

**IV. Attitude Estimation System Structure for Sensor Fusion**

Attitude estimation comprises two sets of equations representing the system dynamics and measurement model. In this section, we present these equations using an Euler angle and quaternion representation. The system equation is derived from the gyroscope dynamics and the measurement equations are composed of accelerometer and magnetometer models. Additionally, error dynamics is derived for estimation when using quaternions.

**A. Euler Angle Attitude Estimation System**

1. **System Equations with Euler Angle**

   The attitude kinematics equation in terms of a 3 – 2 – 1 Euler angle sequence is given by:

   \[
   \begin{pmatrix}
   \dot{\phi} \\
   \dot{\theta} \\
   \dot{\psi}
   \end{pmatrix}
   = T \omega,
   \]  

   (10)
where

\[
T(\phi, \theta, \psi) := \begin{bmatrix}
1 & \tan \theta \sin \phi & \tan \theta \cos \phi \\
0 & \cos \phi & -\sin \phi \\
0 & \sin \phi \sec \theta & \cos \phi \sec \theta
\end{bmatrix},
\]

(11)

and $\omega$ is the angular rate of the body with respect to inertial frame. Refer to §Appendix.A for the complete derivation of the attitude influence matrix $T$.

With the gyroscope measurement model (7), the system equations with the Euler angle representation (1) is then given by:

\[
\begin{pmatrix}
\dot{\Phi} \\
\dot{b}
\end{pmatrix} =
\begin{bmatrix}
0 & -T(\Phi) \\
0_{3\times3} & 0_{3\times3}
\end{bmatrix}
\begin{pmatrix}
\Phi \\
b
\end{pmatrix} +
\begin{bmatrix}
0_{3\times3} & T(\Phi) \\
0_{3\times3} & I
\end{bmatrix}
\begin{pmatrix}
n_w \\
n_b
\end{pmatrix} +
\begin{pmatrix}
n_w \\
n_b
\end{pmatrix} \omega_m,
\]

(12)

The process noise covariance matrix for $\begin{pmatrix}
n_w \\
n_b
\end{pmatrix}$ is given by:

\[
Q = \begin{bmatrix}
N_w & 0_{3\times3} \\
0_{3\times3} & N_b
\end{bmatrix} =
\begin{bmatrix}
n_w^2 I_{3\times3} & 0_{3\times3} \\
0_{3\times3} & n_b^2 I_{3\times3}
\end{bmatrix},
\]

(13)

2. Measurement equations with Euler angle

The attitude of the vehicle with Euler angle is recovered from the DCM with Euler angle by comparing data between body frame and inertia frames with accelerometer and magnetometer model.

The relationship between the gravity vector $^Ig$ in the inertial frame and the acceleration vector $^B a$ in body frame from the accelerometer measurement (8) can be formulated as:

\[
^B a = C_{I \ acc}^B (\Phi) \ ^I g
\]

(14)

with $C_{I \ acc}^B$ is the DCM from inertial frame (I) to body frame (B) as defined in (2).

The relationship between the Earth’s magnetic vector $^I h$ and the local magnetic vector $^B m$ from the magnetometer
measurement (9) can be expressed as:

$$B^m = C^B_{I_{mag}}(\Phi)^t h$$  \hspace{1cm} (15)

Therefore, the final measurement equations combining two sensor models (8) and (9) are formulated as:

$$\begin{pmatrix} B^a_m \\ B^m_m \end{pmatrix} = \begin{bmatrix} C_{acc}(\Phi) & C_{mag}(\Phi) \end{bmatrix} \begin{pmatrix} g \\ h \end{pmatrix} + \begin{pmatrix} n_a \\ n_m \end{pmatrix}$$  \hspace{1cm} (16)

and, the sensor noise covariance matrix for $$\begin{pmatrix} n_a \\ n_m \end{pmatrix}$$ is given by:

$$Q = \begin{bmatrix} N_a & 0_{3x3} \\ 0_{3x3} & N_m \end{bmatrix} = \begin{bmatrix} n^2_a I_{3x3} & 0_{3x3} \\ 0_{3x3} & n^2_m I_{3x3} \end{bmatrix}$$  \hspace{1cm} (17)

B. Quaternion Attitude Estimation System

1. System Dynamics

a. System Equations with Quaternion Using the definition of the quaternion derivative \[39\] and the gyroscope sensor model (7), the system of differential equations is obtained as:

$$\frac{d}{dt} \tilde{q} = \frac{1}{2} \Omega(\omega) \tilde{q}.$$  \hspace{1cm} (18a)

$$\dot{b} = n_\omega$$  \hspace{1cm} (18b)

where:

$$\Omega(\omega) := \begin{bmatrix} 0 & w_z & -w_y & w_x \\ -w_z & 0 & w_x & -w_y \\ w_y & -w_x & 0 & w_z \\ -w_x & w_y & -w_z & 0 \end{bmatrix}, \quad \tilde{q} := \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix} = \begin{pmatrix} q \\ \dot{q} \end{pmatrix}$$  \hspace{1cm} (19)

with $$\omega$$ is the angular rate of the body frame with respect to inertial frame.

b. Measurement equations with Quaternion

The attitude of the vehicle in quaternions is recovered from the DCM using quaternions by comparing sensor values
in the body frame against gravity or Earth’s magnetic field vector in the inertial frame.

The relationship between the gravity vector $^Ig$ in the inertial frame and the acceleration vector $^Ba$ in body frame from the accelerometer measurement (8) can be formulated with quaternions as:

$$ ^Ba = C_{acc}^I(\bar{q})^Ig $$

with $C_{acc}^I$ is the DCM from inertial frame (I) to body frame (B).

The relationship between the Earth’s magnetic vector $^Ih$ and the local magnetic vector $^Bm$ from magnetometer measurement (9) can be expressed as:

$$ ^Bm = C_{mag}^I(\bar{q})^Ih $$

Therefore, the final measurement equations combining two sensor model (8) and (9) are formulated as:

$$ \begin{bmatrix} ^Ba_m \\ ^Bm_m \end{bmatrix} = \begin{bmatrix} C_{acc}(\bar{q}) & C_{mag}(\bar{q}) \end{bmatrix} \begin{bmatrix} g \\ h \end{bmatrix} + \begin{bmatrix} n_a \\ n_m \end{bmatrix} $$

and, the sensor noise covariance matrix for $\begin{bmatrix} n_a \\ n_m \end{bmatrix}$ is the same as in the Euler angles’ case and given by:

$$ R = \begin{bmatrix} N_a & 0_{3x3} \\ 0_{3x3} & N_m \end{bmatrix} = \begin{bmatrix} n_a^2I_{3x3} & 0_{3x3} \\ 0_{3x3} & n_m^2I_{3x3} \end{bmatrix} $$

2. Error Dynamics

a. Error System Equations with Quaternion

Instead of using the arithmetic difference between quaternion and quaternion estimate to define the error, we will introduce the error quaternion $\delta\hat{q}$; a small rotation between the estimated and the true orientation of the body frame of reference.

This error calculation is expressed as a multiplication in quaternion algebra [38] is:

$$ ^B\delta q = ^B\delta \hat{q} \otimes \hat{q} $$

$$ ^B\delta \hat{q} = ^B\hat{q} \otimes \hat{q}^{-1} $$
Note that here, ⊗ is used to indicate a product of two terms in quaternion algebra, and is NOT the Kronecker product.

We can apply the small angle approximation to \( \delta \hat{q} \) assuming the rotation associated with the error quaternion is very small. Consequently, the attitude error angle vector \( \delta \theta \) is calculated as:

\[
\delta \hat{q} = \begin{pmatrix} \delta q \\ \delta q_4 \end{pmatrix} = \begin{pmatrix} \delta \hat{k} \sin(\delta \theta/2) \\ \cos(\delta \theta/2) \end{pmatrix} \approx \begin{pmatrix} \frac{1}{2} \delta \theta \\ 1 \end{pmatrix}
\]  

This error angle vector \( \delta \theta \) is of dimension \( 3 \times 1 \) and will be used together with the bias error in the error state vector. The bias error is defined as:

\[
\Delta b = b - \hat{b}
\]  

From [29], the definition of the error quaternion [24] results in the following set of error system equations:

\[
\begin{aligned}
\delta \hat{\theta} &= -[\hat{\omega} \times] \delta \theta - \Delta b - n_w \\
\Delta \hat{b} &= \hat{b} - \hat{b} = b
\end{aligned}
\]  

or, in the state space form as:

\[
\begin{pmatrix} \delta \hat{\theta} \\ \Delta \hat{b} \end{pmatrix} = \begin{pmatrix} -\hat{\omega} \times & -I_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} \end{pmatrix} \begin{pmatrix} \delta \hat{\theta} \\ \Delta \hat{b} \end{pmatrix} + \begin{pmatrix} -I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_{3 \times 3} \end{pmatrix} \begin{pmatrix} n_w \\ n_b \end{pmatrix}.
\]

The process noise covariance matrix for \( n_w \) is also the same as in the Euler angles’ case and given by:

\[
Q = \begin{bmatrix} N_w & 0_{3 \times 3} \\ 0_{3 \times 3} & N_b \end{bmatrix} = \begin{bmatrix} n_w^2 I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & n_b^2 I_{3 \times 3} \end{bmatrix}
\]

b. Error Measurement Equations with Quaternion

Error dynamics is used to estimate the error quaternion \( \delta \hat{q} \), defined previously in the error system equation. Recall that the measurement equations with quaternion [22] are in terms of the DCM and hence, we obtain the error in the estimate in the following manner.
\[ \bar{y} = y - \hat{y} = (C_{I}^{B}(\hat{q}) - C_{I}^{B}(\tilde{q})) I \begin{bmatrix} g \\ h \end{bmatrix} \]  

(31)

where, \( y \) is actual measurement and \( \hat{y} \) is estimated measurement with \( \hat{q} \) from the result of original system equations (18). With this equation we can derive final error measurement equations, formulated in detail in §Appendix B. With that, the final error measurement equations with accelerometer and magnetometer model can be written using the DCM as:

\[ \bar{y} = C_{I}^{B}(\hat{q}) \begin{bmatrix} g \\ h \end{bmatrix} + D_{w}w(t) \]  

(32)

where,

\[ C_{I}^{B}(\hat{q}) = \begin{bmatrix} C_{acc}(\hat{q}) \\ C_{mag}(\hat{q}) \end{bmatrix}, \quad D_{w} = \begin{bmatrix} I_{3x3} & 0_{3x3} \\ 0_{3x3} & I_{3x3} \end{bmatrix} \]

V. \( \mathcal{H}_2 \) Optimal Estimation

We next present very briefly, the necessary background for \( \mathcal{H}_2 \) optimal estimation method for linear systems. We consider the following linear system,

\[ \dot{x}(t) = Ax(t) + Bu(t) + B_{w}w(t) \]  

(33a)

\[ y(t) = C_{y}x(t) + Du(t) + D_{w}w(t) \]  

(33b)

\[ z(t) = C_{z}x(t) \]  

(33c)

with \( x \in \mathbb{R}^{n}, y \in \mathbb{R}^{l}, z \in \mathbb{R}^{m} \) are respectively the state vector, the measured output vector, and the output vector of interest. Variables \( w \in \mathbb{R}^{p} \) and \( u \in \mathbb{R}^{r} \) are the disturbance and the control vectors, respectively.

With the above defined system, the \( \mathcal{H}_2 \) state estimator has the following form,

\[ \dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L(C_{y}\hat{x}(t) + Du(t) - y(t)) \]  

(34a)

\[ \hat{z}(t) = C_{z}\hat{x}(t) \]  

(34b)

where \( \hat{x} \) is the state estimate, \( L \) is the estimator gain, and \( \hat{z}(t) \) is the estimate of the output of interest. The error equations
are then given by:

\[
\dot{e}(t) = (A + LC_y)e(t) + (B_w + LD_w)w(t) \quad (35a)
\]

\[
\ddot{z}(t) = C_z\dot{e}(t) \quad (35b)
\]

The problem of $\mathcal{H}_2$ state estimation design is then, given a system (35) and a positive scalar $\gamma$, find a matrix $L$ such that,

\[
\|G_{\hat{z}w}(s)\|_2 < \gamma. \quad (36)
\]

where the transfer function $G_{\hat{z}w}(s)$ of the system is:

\[
G_{\hat{z}w}(s) = C_z[sI - (A + LC_y)]^{-1}(B_w + LD_w). \quad (37)
\]

The optimization formulation to obtain $L$ is given by:

**Theorem ($\mathcal{H}_2$ Optimal Estimation) [40][41]**: The following two statements are equivalent:

1) A solution $L$ to the $\mathcal{H}_2$ state estimator exists.

2) $\exists$ a matrix $W$, a symmetric matrix $Q$, and a symmetric matrix $X$ such that:

\[
\begin{bmatrix}
XA + WC_y + (XA + WC_y)^T & XB_w + WD_w \\
* & -I
\end{bmatrix} < 0
\]

\[
\begin{bmatrix}
-Q & C_z \\
* & -X
\end{bmatrix} < 0
\]

\[
\text{trace}(Q) < \gamma^2
\]

The $\mathcal{H}_2$ optimal estimator gain is recovered by $L = X^{-1}W$. This optimal gain ensures that:

\[
e(t) = x(t) - \hat{x} \to 0, \quad \text{as } t \to \infty, \quad (39)
\]

In other words, $\hat{x}(t)$ is an asymptotic estimate of $x(t)$.
VI. Extended $\mathcal{H}_2$ Optimal Estimation

A. Euler angle Estimation with Extended $\mathcal{H}_2$

The proposed extended $\mathcal{H}_2$ estimation framework is summarized in Fig. 1. To express the dynamics of the system using the state space notation, we can define a 6-element state vector with Euler angle and gyroscope bias as:

$$x(t) = \begin{bmatrix} \Phi \\ b \end{bmatrix}$$

Therefore, non-linear system of gyroscope dynamics with the Euler angle \( \Phi \) can be rewritten with states as:

$$\dot{x} = f(x, u, w, t), \quad (40)$$

where \( u(t) := \omega_m(t) \) and \( w(t) := [n_{\omega}(t) \quad n_b(t)]^T \)

The measurement equations with accelerometer and magnetometer model \([14][15]\) can be rewritten with states as the following nonlinear output equation:

$$y(t) = h(x, w, t) \quad (41)$$
where $w(t) := [n_u(t) \ n_m(t)]^T$.

Extended $\mathcal{H}_2$ estimation is $\mathcal{H}_2$ estimation extended to nonlinear system models, along the lines of the extended Kalman filter. In extended Kalman filtering, the uncertainty is propagated using the linear system along the state trajectory, and the Kalman gain is computed at every time step with the instantaneous linear system. In the extended $\mathcal{H}_2$ framework, in theory, we can solve for the optimal $\mathcal{H}_2$ gain along the trajectory, however this may be computationally prohibitive for cheap processors. Instead, we design the optimal $\mathcal{H}_2$ gain about the nominal operating point, but use the nonlinear system dynamics to evolve the estimator’s states.

A linear approximation is implemented at nominal operating point $(x_0, u_0, w_0) = 0$. The linear equations are:

$$\dot{x}(t) \approx Ax(t) + Bu(t) + B_w w(t)$$  \hspace{1cm} (42)

i.e.,

$$\begin{pmatrix} \Phi \\ \dot{b} \end{pmatrix} = \begin{pmatrix} 0 & -I_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} \end{pmatrix} \begin{pmatrix} \Phi \\ b \end{pmatrix} + \begin{pmatrix} -I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I \end{pmatrix} \begin{pmatrix} n_w \\ n_b \end{pmatrix} + \begin{pmatrix} I_{3 \times 3} \\ 0_{3 \times 3} \end{pmatrix} \omega_m.$$  \hspace{1cm} (43)

A linear approximation of (41) is derived about the previously used nominal operating point $(x_0, w_0) = 0$. The linear equations thus obtained are:

$$y(t) \approx C_y x(t) + D_w(t) w(t)$$  \hspace{1cm} (44)

i.e.,

$$y(t) = \begin{bmatrix} g \times \\ h \times \end{bmatrix} x(t) + \begin{bmatrix} I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix} w(t)$$  \hspace{1cm} (45)

where $g \times$ and $h \times$ are the skew-symmetric matrix forms of the vectors $g$ and $h$ respectively.

The linear system, about the nominal operating point, is therefore:

$$\dot{x} = Ax + Bu + B_w w$$  \hspace{1cm} (46a)

$$y = C_y x + D_u u + D_w w$$  \hspace{1cm} (46b)

The optimal $\mathcal{H}_2$ gain $L_0$ can then be determined by solving the optimization problem in (38), where the subscript 0 is used to indicate that it is determined about the nominal operating point.
Once the gain $L_0$ is determined it is used to implement the $\mathcal{H}_2$ filter for the nonlinear system. We present a new implementation, called the extended $\mathcal{H}_2$ filter, where the filter states are propagated using the nonlinear dynamics. In conventional $\mathcal{H}_2$ filters, the error propagation occurs using the linear system. The filter dynamics and output equation for the extended $\mathcal{H}_2$ filter are given by:

\[
\begin{align*}
\dot{x} &= f(\hat{x}(t), u(t), 0, t) + L_0(h(\hat{x}(t), 0, t) - y) \\
\hat{z} &= C_z \hat{x}(t)
\end{align*}
\]  

(47a)  

(47b)

B. Quaternion Estimation with Extended $\mathcal{H}_2$

The process for extended $\mathcal{H}_2$ estimation using quaternions is not too different from that developed for using Euler angles. The major distinction arises when solving for the error in the error dynamics model, since quaternion vector algebra requires special attention. The error state between the true quaternion and the predicted quaternion estimate expressed as $\delta \hat{q}$ is used for estimation, as stated in (26). Another distinguishing characteristic is the introduction of gain scheduling to adequately cover all possible orientations of the vehicle. We obtain these gains as a function of the linearization points, and apply them during the nonlinear update step. The proposed extended $\mathcal{H}_2$ estimation framework for quaternion is outlined in Fig.2. Like any estimation procedure, this is essentially composed of two steps: state prediction and state update.

**Fig. 2 Quaternion estimation algorithm**
C. First step : State Prediction

To express the dynamics of the quaternion system with state space notation, we define a 7-element state vector.

\[ x(t) = \begin{pmatrix} \vec{q} \\ \vec{b} \end{pmatrix} \]  

(48)

using the definition of the quaternion derivative \[38\]. The non-linear system of gyroscope dynamics with quaternion \[18\] can be rewritten with states as:

\[ \dot{x} = f(x, u, w, t), \]  

(49)

where \( u(t) := \omega_m(t) \) and \( w(t) := [n_\omega(t) \quad n_b(t)] \).

Taking the expectation of the above equation \[49\] derived from the quaternion gyroscope dynamics \[18\], we can get prediction equations as:

\[ \hat{\dot{x}} = f(\hat{x}, u, w, t), \]  

(50)

i.e.,

\[ \begin{align*}
\dot{\hat{\vec{q}}} & = \frac{1}{2} \Omega(\omega_m - \hat{\vec{b}}) \hat{\vec{q}}, \\
\hat{\vec{b}} & = 0_{3\times1}
\end{align*} \]  

(51a, 51b)

With this dynamic equation, we can get a predicted estimate of \( \hat{\vec{q}} \) in Fig\[2\] which will be updated with a proposed extended \( \mathcal{H}_2 \) estimation with error dynamics.

D. Second Step : State Update

1. Error State Equations:

We defined the error between the true quaternion and quaternion estimate as \( \delta \vec{q} \) and applied small angle approximation it as \( \delta \theta \) in \[26\] in previous section. We can now define a 6-element state vector for the estimation of the error dynamics as:

\[ \tilde{x} = \begin{pmatrix} \delta \theta \\ \Delta \vec{b} \end{pmatrix} \]  

(52)
With this state, error system equation with quaternion \(^{29}\) can be rewritten as a nonlinear system as:

\[
\dot{x} = f(\bar{x}, u, w, t).
\]

(53)

where \(u(t) := \omega_m(t)\) and \(w(t) := [n_{\omega}(t) \quad n_b(t)]\).

A linear approximation is implemented at the chosen nominal operating point \((x_0, u_0, w_0) = 0\). The linear equations are:

\[
\dot{x}(t) \approx Ax(t) + B_w w(t)
\]

(54)
i.e.,

\[
\begin{pmatrix}
\dot{\delta}\theta \\
\Delta b
\end{pmatrix} =
\begin{bmatrix}
-|\hat{\omega}| & -I_{3x3} \\
0_{3x3} & 0_{3x3}
\end{bmatrix}
\begin{pmatrix}
\delta\theta \\
\Delta b
\end{pmatrix} +
\begin{bmatrix}
-I_{3x3} & 0_{3x3} \\
0_{3x3} & I_{3x3}
\end{bmatrix}
\begin{pmatrix}
n_{w} \\
n_{b}
\end{pmatrix},
\]

(55)

where \(\hat{\omega} := \omega_m - \dot{b}\).

2. Error Measurement Equation:

The measurement equation of the error system with accelerometer and magnetometer can be written as the following nonlinear output equation:

\[
\bar{y} = h(\bar{x}, w, t) = C_I^B(\hat{q})
\begin{bmatrix}
g \\
h
\end{bmatrix} + D_w w(t)
\]

(56)

where,

\[
C_I^B(\hat{q}) =
\begin{bmatrix}
C_{acc}(\hat{q}) & C_{mag}(\hat{q})
\end{bmatrix},
\quad D_w =
\begin{bmatrix}
I_{3x3} & 0_{3x3} \\
0_{3x3} & I_{3x3}
\end{bmatrix}
\]

We apply a linear approximation to (56) about eight points, each of which is the combination of a point from each axis covering a range of \([\frac{\pi}{2}, \frac{\pi}{2}]\) and \([\frac{\pi}{2} - \frac{\pi}{2}]\), i.e., the following eight points: \((0, 0, 0), (0, 0, \pi), (0, \pi, 0), (\pi, 0, 0), (0, \pi, \pi), (\pi, 0, \pi), (\pi, \pi, 0), (\pi, \pi, \pi)\).
The linear system, about the nominal operating point, is therefore:

\[ \dot{x} = A_i x + B_{ui} u + B_{wi} w \]  
\[ y = C_{yi} x + D_{ui} u + D_{wi} w \]

(i.e.,

\[ y(t) = \begin{bmatrix} [g \times]_i & 0_{3x3} \\ [h \times]_i & 0_{3x3} \end{bmatrix} x(t) + \begin{bmatrix} I_{3x3} & 0_{3x3} \\ 0_{3x3} & I_{3x3} \end{bmatrix} w(t) \]

where \( g \times \) and \( h \times \) are the skew-symmetric matrix forms of the vectors \( g \) and \( h \) respectively and the signs vary with the linearization point in consideration. The optimal \( \mathcal{H}_2 \) gain \( L_i \) can then be determined by solving the optimization problem in (38), where the subscript \( i=1 \sim 8 \) is used to indicate that it is determined about the nominal operating point. Once the gain \( L_i \) is determined, it is used to implement the \( \mathcal{H}_2 \) filter for the nonlinear system in the same way as performed in Euler angle estimation. Note that here, gain scheduling is applied, depending on the system states. This is because signs of \( g \) and \( h \) in the measurement equation (58) are inverted while the vehicle is rotating. The filter dynamics and output equation, for the extended \( \mathcal{H}_2 \) filter, are given by:

\[ \dot{\hat{x}} = f(\hat{x}(t), u(t), 0, t) + L_i(h(\hat{x}(t), 0, t) - y) \]  
\[ \hat{z} = C_z \hat{x}(t) \]

The error state of quaternion \( (\hat{\delta}\theta) \) from the estimation result of previous step is recovered from equation (26) as:

\[ \begin{bmatrix} \delta \hat{q} \\ \Delta \hat{b} \end{bmatrix} = \begin{bmatrix} \hat{\delta}\theta/2 \\ \Delta \hat{b} \end{bmatrix} \]

We need to ensure that the unit norm constraint of the estimated quaternion is not violated. The full quaternion satisfying the unit norm constraint can be recovered from equation (4) as:

\[ \delta \hat{q} = \begin{bmatrix} \delta \hat{q} \\ \sqrt{1 - \delta \hat{q}^T \delta \hat{q}} \end{bmatrix} \quad \text{or} \quad \delta \hat{q} = \frac{1}{\sqrt{1 - \delta \hat{q}^T \delta \hat{q}}} \begin{bmatrix} \delta \hat{q} \\ 1 \end{bmatrix} (\text{if } \delta \hat{q}^T \delta \hat{q} > 1) \]

The final estimate is a quaternion multiplication of the two results from the prediction step \( \hat{\hat{q}} \) and the update step \( \hat{\hat{q}} \).
\( \dot{\hat{q}} = \delta \hat{q} \otimes \hat{q}^- \)

(62)

VII. Results

A. Simulation Set-Up

The implementation of the proposed extended \( H_2 \) filter for the attitude estimation problem is tested with the MATLAB simulation environment as shown in Fig. 3. Its performance is compared with that from the extended Kalman filter based implementation, simulated using identical sensor data. The comparison is done in terms of accuracy and computational time under multiple scenarios of flight stages like take-off, landing, hovering, and transition flight.

For our experiment, we choose to simulate using the MPU 9250, an affordable commercial sensor used most popularly in the Pixhawk flight computer. Sensor characteristics like noise levels, bias, etc. are imported to the MATLAB simulation from sensor data sheet [42] of MPU 9250. We use MATLAB’s simulation function, `imuSensor` to generate MARG data [43]. This raw data is shown in Fig. 5 with the trajectory shown in Fig. 4. Four scenarios covering major flight stages are used to verify the proposed extended \( H_2 \) estimation algorithm for both of the coordinate representations discussed so far. The first two cases are used for both the Euler angle and quaternion representation whereas the last two cases correspond only to the quaternion representation.

**Case I: Slow and Small Angular Movements** – Here we consider angular movements < 30° about all three axes of the vehicle independently. This case broadly covers forward/backward and left/right cruise flight of popular quad-rotor based UAVs. Simulation is run for a time duration of 50 seconds and with an angular rate of \( \pi/50 \) rad/s. The simulated true state trajectories are shown in Fig. 6.

**Case II: Fast and High Angular Movements** – Here we consider angular variation > 30° about all three axes of the vehicle simultaneously. It represents scenarios of rapid movements or motion in the presence of wind disturbance during flight or aggressive maneuvers. Simulation is run for a duration of 10 seconds and with an angular rate of \( \pi/3 \) rad/s. The simulated true state trajectories are shown in Fig. 8.

**Case III: Gimbal-lock test** – Here we consider angular movements > 90° about pitch axis of the vehicle. This case
broadly covers transition flight of the increasingly popular VTOL (Vertical Take Off and Land) UAVs. Simulation is run for a time duration of 10 seconds and with an angular rate $\pi/2$ rad/s. The simulated true state trajectories are shown in Fig. 12.

**Case IV: Movement from 3D flight simulation** – Here we consider flight trajectory generated by SITL (Simulation In The Loop) with the Ardupilot firmware and Gazebo. This represents scenarios that include taking off, cruise to three way points and then landing. The simulated true state trajectories are shown in Fig. 14.

B. Simulation Results for Euler Angle Estimation

Here we examine the performance of the extended $\mathcal{H}_2$ estimator for Euler angle with that of the standard EKF in terms of root mean squared (RMS) error, memory use, and computational time required.

**Case I:** The comparison of the two estimators for Case I is shown in Fig. 7. We observe that the error of extended $\mathcal{H}_2$ estimator is less than that of EKF. The RMS error, and the upper and lower bounds of the error, for both the filters are shown in Table 1. From the plots and the data in the tables, we can conclude that the performance of extended $\mathcal{H}_2$ estimator is better than that of EKF.

| Algorithm     | Roll angle(°) | Pitch angle(°) | Yaw angle(°) |
|---------------|---------------|----------------|--------------|
| Extended $\mathcal{H}_2$ | 0.0331        | 0.0538         | 0.1107       |
| EKF           | 0.0533        | 0.0988         | 0.2298       |

**Case II:** The results of the two estimators for case II are shown in Fig. 9. And, here, we observe that the extended $\mathcal{H}_2$ filter has lower error bounds. From Table 2 we observe that the RMS errors are comparable for both the filters.

| Algorithm     | Roll angle(°) | Pitch angle(°) | Yaw angle(°) |
|---------------|---------------|----------------|--------------|
| Extended $\mathcal{H}_2$ | 0.3045        | 0.3260         | 0.3121       |
| EKF           | 0.3073        | 0.2656         | 0.5275       |

Based on the estimation errors for Case I and Case II, we can conclude that both the filters perform equally well, with the extended $\mathcal{H}_2$ filter performing slightly better. The main advantage of the extended $\mathcal{H}_2$ filter is in the implementation efficiency. The results for the average execution time are shown in Table 3 which reveals that the extended $\mathcal{H}_2$ estimator requires 50% less computational time than EKF, making it $2 \times$ more efficient than EKF. Table 3 also shows the variability in the computational time, which is about $3 \times$ less. The reduced variability in the computational time increases the
reliability of the real-time tasks that will execute in the microprocessor. Typically, more time is allotted to real-time tasks with large variability in computational time. This further improves the computational efficiency of the proposed extended $H_2$ estimator.

| Algorithm       | Mean Time (ms) | Standard Deviation (ms) |
|-----------------|----------------|------------------------|
| Extended $H_2$  | 0.853          | 0.244                  |
| EKF             | 1.7            | 0.736                  |

### C. Simulation Results for Quaternion Estimation

**Case I and II:** The comparison of the two estimators for Case I and II is shown in Fig. [10] and [11] respectively. Just like in the Euler angle case, we can conclude that the performance of the extended $H_2$ is slightly better than that of the EKF.

**Case III:** The comparison of the two estimators for Case III is shown in Fig. [13]. We observe that Extended $H_2$ estimation is functional even when it encounters gimbal lock, a problem faced in an Euler angle-based implementation. Moreover, the error of extended $H_2$ estimator is comparable with that of EKF. The RMS error for both the filters are shown in Table [4]. From the plots and the data in the tables, we can conclude that the performance of extended $H_2$ estimator is comparable with that of EKF.

| Algorithm       | Pitch angle(°) | Standard Deviation(°) |
|-----------------|----------------|-----------------------|
| Extended $H_2$  | 0.1204         | 0.013                 |
| EKF             | 0.1569         | 0.0143                |

**Case IV:** The comparison of the two estimators for Case IV is shown in Fig. [15]. We observe that the error of extended $H_2$ estimator is comparable with that of EKF. The RMS error for both the filters are shown in Table [5]. From the plots and the data in the tables, we can conclude that the performance of extended $H_2$ estimator is better than that of EKF for case IV as well.

### VIII. Conclusions

This paper presents a new nonlinear estimation framework, based on $H_2$ optimal state estimation, for attitude estimation in low power microprocessors. This algorithm is presented using the two popular choices for attitude
Table 5  RMS error for Case IV.

| Algorithm | Roll angle(°) | Pitch angle(°) | Yaw angle(°) |
|-----------|--------------|---------------|--------------|
| Extended $\mathcal{H}_2$ | 0.0410 | 0.0567 | 0.1274 |
| EKF | 0.0956 | 0.1233 | 0.2761 |

Table 6  Computational Time Comparison on Quaternion Estimation

| Algorithm | Mean Time (ms) | Standard Deviation (ms) |
|-----------|---------------|-------------------------|
| Extended $\mathcal{H}_2$ | 0.9263 | 0.5217 |
| EKF | 2.7 | 1.5 |

representation, namely Euler angles and quaternions. This work showed that the performance of the proposed estimator is comparable, if not better, than that of the EKF algorithm which is typically used in the application space considered. The primary advantage of the proposed framework is the $2 \times$ computational efficiency, and the $3 \times$ robustness with respect to computational uncertainty. Both these factors make the proposed attitude estimation algorithm very attractive for small UAVs with low power microprocessors.

Appendix

A. Euler Angle Attitude Kinematics

The attitude matrix in terms of an Euler angle sequence is formed in a straightforward way \cite{44, 45} as:

$$[C_{ijk}(\psi, \theta, \phi)] = [C_{k}(\phi)][C_{j}(\theta)][C_{i}(\psi)]$$  (63)

Therefore, the attitude matrix in terms of an Euler angle sequence 3-2-1 is given by:

$$[C_{321}(\phi, \theta, \psi)] = [C_{1}(\phi)][C_{2}(\theta)][C_{3}(\psi)]$$  (64)

The kinematic expression can be obtained by projecting the angular rates along the axes of the reoriented frame as:

$$\omega = [\phi, 0, 0]^T + [C_{1}(\phi)][0, \dot{\theta}, 0]^T + [C_{1}(\phi)][C_{2}(\theta)][0, 0, \dot{\psi}]^T$$  (65)
Implementing the matrix multiplication and collecting the angular rates in a column array gives the results as:

\[
B = \begin{bmatrix}
1 & 0 & -\sin(\theta) \\
0 & \cos \phi & \cos \theta \sin \phi \\
0 & -\sin \phi & \cos \phi \cos \theta
\end{bmatrix}
\]  

(66)

This matrix \( B \) can be inverted to produce the attitude influence matrix \( T \) that satisfies \( [\dot{\phi}, \dot{\theta}, \dot{\psi}]^T = T \omega \) as:

\[
T = \begin{bmatrix}
1 & \tan \theta \sin \phi & \tan \theta \cos \phi \\
0 & \cos \phi & -\sin \phi \\
0 & \sin \phi \sec \theta & \cos \phi \sec \theta
\end{bmatrix}
\]  

(67)

This matrix \( T \) transforms the angular rate vector from body frame to the inertial frame. This will be used in gyroscope sensor modeling.

**B. Derivation of Error Measurement Equation with the accelerometer and magnetometer**

The measurement equations of quaternion is rewritten with (41) as

\[
y = h(\bar{q}) + w
\]  

(68)

with \( y \) is the actual measurement, \( w \) is the noise of the sensor and

\[
h(\bar{q}) = C_B^I(\bar{q}) \begin{bmatrix} g \\ h \end{bmatrix}^T
\]  

(69)

\[
= C_B^R(\delta \theta) C_I^B(\tilde{q}) \begin{bmatrix} g \\ h \end{bmatrix}^T
\]  

(70)

where \( \begin{bmatrix} g \\ h \end{bmatrix}^T \) in the inertial frame.

The error measurement equations are defined as the subtraction between true measurement and estimation estimation as:

\[
\hat{y} = y - \hat{y} = (C_B^R(\bar{q}) - C_B^R(\tilde{q})) \begin{bmatrix} g \\ h \end{bmatrix}^T
\]  

(71)
substituting the definition $\vec{q}$ as $\vec{q} = \delta \vec{q} \otimes \hat{\vec{q}}$ and the properties of the rotational matrix $[29]$ lead to:

$$C_B^I(\vec{q}) = C_B^I(\delta \vec{q}) \cdot C_B^I(\hat{\vec{q}})$$  \hspace{1cm} (72)

$$C_B^R(\delta \vec{q}) = I - |\delta \theta \times |.$$  \hspace{1cm} (73)

Substituting the result equations above in the part of the error measurement equation (56) lead to:

$$C_B^I(\delta \vec{q}) - C_B^I(\hat{\vec{q}}) - C_B^I(\hat{\vec{q}}) = (C_B^R(\delta \vec{q}) - I_{3\times3}) \cdot C_B^I(\hat{\vec{q}})$$  \hspace{1cm} (74)

$$= -|\delta \theta \times | \cdot C_B^I(\hat{\vec{q}})$$  \hspace{1cm} (75)

Then, the error measurement (56) is rewritten as:

$$\tilde{y} = -|\delta \theta \times | \cdot C_B^I(\hat{\vec{q}}) v_n + n_m$$  \hspace{1cm} (77)

$$= |C_B^I(\hat{\vec{q}}) v_n \times | \cdot \delta \theta$$  \hspace{1cm} (78)

$$= \begin{bmatrix} \delta \theta \\ \delta \theta \\ \delta \theta \end{bmatrix} + n_m$$  \hspace{1cm} (79)

Therefore, the final measurement equation of error system with accelerometer and magnetometer model can be written as the following nonlinear equation:

$$\tilde{y} = h(\tilde{x}, w, t) = C_B^I(\hat{\vec{q}}) \begin{bmatrix} g \\ h \end{bmatrix} + D_w w(t)$$  \hspace{1cm} (80)

where,

$$C_B^I(\hat{\vec{q}}) = \begin{bmatrix} C_{acc}(\hat{\vec{q}}) & C_{mag}(\hat{\vec{q}}) \end{bmatrix}, D_w = \begin{bmatrix} I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I_{3\times3} \end{bmatrix}$$

**Funding Sources**

This project has been supported by NSF grants #1762825 and IUSE/PFE: RED: REvolutionizing Diversity Of Engineering (REDO-E) Award Number:1730693.
References

[1] Mazur, M., Wisniewski, A., and McMillan, J., “PwC global report on the commercial applications of drone technology,” PricewaterhouseCoopers, tech. Rep., 2016.

[2] Canetta, L., Mattei, G., and Guanziroli, A., “Exploring commercial UAV market evolution from customer requirements elicitation to collaborative supply network management,” 2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC), IEEE, 2017, pp. 1016–1022.

[3] DRONEII, “The Drone Market Report 2019,”, 2019. URL https://www.droneii.com/project/drone-market-report [Accessed: 10-Nov-2019].

[4] Cai, G., Dias, J., and Seneviratne, L., “A survey of small-scale unmanned aerial vehicles: Recent advances and future development trends,” Unmanned Systems, Vol. 2, No. 02, 2014, pp. 175–199.

[5] Eure, K. W., Quach, C. C., Vazquez, S. L., Hogge, E. F., and Hill, B. L., “An application of UAV attitude estimation using a low-cost inertial navigation system,” 2013.

[6] Gebre-Egziabher, D., Hayward, R. C., and Powell, J. D., “Design of multi-sensor attitude determination systems,” IEEE Transactions on aerospace and electronic systems, Vol. 40, No. 2, 2004, pp. 627–649.

[7] Kada, B., Munawar, K., Shaikh, M., Hussaini, M., and Al-Saggaf, U., “UAV attitude estimation using nonlinear filtering and low-cost mems sensors,” IFAC-PapersOnLine, Vol. 49, No. 21, 2016, pp. 521–528.

[8] Weibel, D., Lawrence, D., and Palo, S., “Small unmanned aerial system attitude estimation for flight in wind,” Journal of Guidance, Control, and Dynamics, Vol. 38, No. 7, 2015, pp. 1300–1305.

[9] Yazdi, N., Ayazi, F., and Najafi, K., “Micromachined inertial sensors,” Proceedings of the IEEE, Vol. 86, No. 8, 1998, pp. 1640–1659.

[10] Chang, H., Xue, L., Qin, W., Yuan, G., and Yuan, W., “An integrated MEMS gyroscope array with higher accuracy output,” Sensors, Vol. 8, No. 4, 2008, pp. 2886–2899.

[11] Fan, B., Li, Q., and Liu, T., “How magnetic disturbance influences the attitude and heading in magnetic and inertial sensor-based orientation estimation,” Sensors, Vol. 18, No. 1, 2018, p. 76.

[12] Luinge, H. J., and Veltink, P. H., “Measuring orientation of human body segments using miniature gyroscopes and accelerometers,” Medical and Biological Engineering and computing, Vol. 43, No. 2, 2005, pp. 273–282.

[13] Tuck, K., “Tilt sensing using linear accelerometers,” Freescale semiconductor application note AN3107, 2007.

[14] Ozyagcilar, T., “Implementing a tilt-compensated eCompass using accelerometer and magnetometer sensors,” Freescale semiconductor, AN, Vol. 4248, 2012.
[15] Geiger, W., Bartholomeyczik, J., Breng, U., Gutmann, W., Hafen, M., Handrich, E., Huber, M., Jackle, A., Kempfer, U., Kopmann, H., et al., “MEMS IMU for ahrs applications,” 2008 IEEE/ION Position, Location and Navigation Symposium, IEEE, 2008, pp. 225–231.

[16] Pedley, M., “Tilt sensing using a three-axis accelerometer,” Freescale semiconductor application note, Vol. 1, 2013, pp. 2012–2013.

[17] Frosio, I., Pedersini, F., and Borghese, N. A., “Autocalibration of MEMS accelerometers,” IEEE Transactions on Instrumentation and Measurement, Vol. 58, No. 6, 2008, pp. 2034–2041.

[18] Eberly, D., “Rotation representations and performance issues,” Magic Software Inc., Chapel Hill, NC, 2002.

[19] Han, S., and Wang, J., “A novel method to integrate IMU and magnetometers in attitude and heading reference systems,” The Journal of Navigation, Vol. 64, No. 4, 2011, pp. 727–738.

[20] Earl, M. G., and D’Andrea, R., “Real-time attitude estimation techniques applied to a four rotor helicopter,” 2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601), Vol. 4, IEEE, 2004, pp. 3956–3961.

[21] Kang, C. W., and Park, C. G., “Euler angle based attitude estimation avoiding the singularity problem,” IFAC Proceedings Volumes, Vol. 44, No. 1, 2011, pp. 2096–2102.

[22] Janota, A., Šimáč, V., Nemec, D., and Hrbček, J., “Improving the precision and speed of Euler angles computation from low-cost rotation sensor data,” Sensors, Vol. 15, No. 3, 2015, pp. 7016–7039.

[23] Allen, R., and Chang, D., “Performance testing of the systron donner quartz gyro,” Jpl Engineering Memorandum, EM, 1993, pp. 343–1297.

[24] El-Sheimy, N., Hou, H., and Niu, X., “Analysis and modeling of inertial sensors using Allan variance,” IEEE Transactions on instrumentation and measurement, Vol. 57, No. 1, 2007, pp. 140–149.

[25] Lam, Q., Stamatakos, N., Woodruff, C., and Ashton, S., “Gyro modeling and estimation of its random noise sources,” AIAA guidance, navigation, and control conference and exhibit, 2003, p. 5562.

[26] Crassidis, J. L., Markley, F. L., and Cheng, Y., “Survey of nonlinear attitude estimation methods,” Journal of guidance, control, and dynamics, Vol. 30, No. 1, 2007, pp. 12–28.

[27] Ludwig, S. A., and Burnham, K. D., “Comparison of Euler Estimate using Extended Kalman Filter, Madgwick and Mahony on Quadcopter Flight Data,” 2018 International Conference on Unmanned Aircraft Systems (ICUAS), IEEE, 2018, pp. 1236–1241.

[28] Teague, H., “Comparison of Attitude Estimation Techniques for Low-cost Unmanned Aerial Vehicles,” arXiv preprint arXiv:1602.07733, 2016.

[29] Trawny, N., and Roumeliotis, S. I., “Indirect Kalman filter for 3D attitude estimation,” University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep, Vol. 2, 2005, p. 2005.
[30] Ko, N., Jeong, S., and Bae, Y., “Sine rotation vector method for attitude estimation of an underwater robot,” *Sensors*, Vol. 16, No. 8, 2016, p. 1213.

[31] Jing, X., Cui, J., He, H., Zhang, B., Ding, D., and Yang, Y., “Attitude estimation for UAV using extended Kalman filter,” *2017 29th Chinese Control And Decision Conference (CCDC)*, IEEE, 2017, pp. 3307–3312.

[32] Madgwick, S., “An efficient orientation filter for inertial and inertial/magnetic sensor arrays,” *Report x-io and University of Bristol (UK)*, Vol. 25, 2010, pp. 113–118.

[33] Simon, D., *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*, John Wiley & Sons, 2006.

[34] Goldstein, H., Poole, C., and Safko, J., “Classical mechanics,” , 2002.

[35] Carrillo, L. R. G., López, A. E. D., Lozano, R., and Pégard, C., “Modeling the quad-rotor mini-rotorcraft,” *Quad Rotorcraft Control*, Springer, 2013, pp. 23–34.

[36] Diebel, J., “Representing attitude: Euler angles, unit quaternions, and rotation vectors,” *Matrix*, Vol. 58, No. 15-16, 2006, pp. 1–35.

[37] Quan, Q., “Coordinate System and Attitude Representation,” *Introduction to Multicopter Design and Control*, Springer, 2017, pp. 99–120.

[38] Kuipers, J. B., et al., *Quaternions and rotation sequences*, Vol. 66, Princeton university press Princeton, 1999.

[39] Shuster, M. D., “A survey of attitude representations,” *Navigation*, Vol. 8, No. 9, 1993, pp. 439–517.

[40] Duan, G.-R., and Yu, H.-H., *LMIs in control systems: analysis, design and applications*, CRC press, 2013.

[41] Apkarian, P., Tuan, H. D., and Bernussou, J., “Continuous-time analysis, eigenstructure assignment, and H/sub 2/synthesis with enhanced linear matrix inequalities (LMI) characterizations,” *IEEE Transactions on Automatic Control*, Vol. 46, No. 12, 2001, pp. 1941–1946.

[42] InvenSense, T., “MPU-9250 Product Specification Revision 1.1,”, 2016. URL http://www.invensense.com/wp-content/uploads/2015/02/PS-MPU-9250A-01-v1.1.pdf [Accessed: 13-Nov-2019].

[43] MATLAB, “Sensor Fusion and Tracking Toolbox,” , 2018. URL https://www.mathworks.com/products/sensor-fusion-and-tracking.html [Accessed: 18-Sep-2019].

[44] Schaub, H., and Junkins, J. L., *Analytical mechanics of space systems*, American Institute of Aeronautics and Astronautics, 2005.

[45] Hurtado, J. E., *Kinematic and Kinetic Principles*, Lulu. com, 2012.
Fig. 4 True trajectories imported to MATLAB for simulation.

Fig. 5 Sensor Data from MATLAB function *imuSensor* with MPU- 9250.
Fig. 6 True trajectories for Euler angles’ estimation in Case I.

Fig. 7 Comparison of the extended $\mathcal{H}_2$ filter and the EKF for Case I.

29
Fig. 8 True trajectories for Euler angles’ estimation in Case II.

Fig. 9 Comparison of the extended $\mathcal{H}_2$ filter and the EKF for Case II.
Fig. 10  Comparison of the extended $H_2$ filter and the EKF with quaternions for Case I.

Fig. 11  Comparison of the extended $H_2$ filter and the EKF with quaternions for Case II.
Fig. 12 True trajectories for quaternion estimation in Case III.

Fig. 13 Comparison of the extended $H_2$ filter and the EKF with quaternions for Case III.
Fig. 14 True trajectories for quaternion estimation in Case IV.

Fig. 15 Comparison of the extended $\mathcal{H}_2$ filter and the EKF for Case IV.