QUATERNION BASED ATTITUDE ESTIMATION TECHNIQUE INVOLVING THE EXTENDED KALMAN FILTER

A Thesis

Presented to

The Graduate Faculty of The University of Akron

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

Ishwor Gautam

May 2019

QUATERNION BASED ATTITUDE ESTIMATION TECHNIQUE

INVOLVING THE EXTENDED KALMAN FILTER

Ishwor Gautam

Thesis

Approved:	Accepted:
_____	_____
Advisor	Department Chair
Dr. Celal Batur	Dr. Sergio Felicelli
_____	_____
Committee Member	Dean of the College
Dr. Ajay Mahajan	Dr. Craig Menzemer
_____	_____
Committee Member	Dean of the Graduate School
Dr. Siamak Farhad	Dr. Chand Midha
	_____
	Date

ABSTRACT

This thesis illustrates the application of the Extended Kalman filter for online estimation of attitude of a body. The accuracy of controlled attitude largely depends on the performance of the estimation algorithm. In this thesis, the extended Kalman filter (EKF) algorithm consisting of quaternion based state representation is used. The EKF algorithm utilizes gyroscope reading for priori estimation and measurements reading from the accelerometer and the magnetometer to correct the states. In simple terms, the extended Filter is used as the estimation tool by fusing the data from the gyroscope, the accelerometer and the magnetometer. A device that combines the gyroscope, accelerometer and magnetometer is called inertial measurement unit (IMU). The non-accurate scaling, sensor misalignment and non-zero biases of IMU devices are eliminated by proper calibration. The sensors utilized in the estimation have noise and biases which results in propagation of error in time. The noise and biases should be eliminated to get the accurate estimates.

In this work, the EKF algorithm with some modification in state equation and in the Kalman filter gain is implemented for both the steady state and the body acceleration conditions. The estimation of the modified EKF is compared with the estimation technique used by VECTORNAV, a well-known commercial IMU. The modified EKF performed well compared to VECTORNAV in steady state condition. However, under body acceleration the modified EKF did not perform as well as what VECTORNAV did. The attitude estimation technique discussed in this thesis is less expensive and easy compared to those used in missile and aircraft guidance. The algorithm discussed in this thesis can be well implemented in navigation of robots and drones for home applications.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude and appreciation to my advisor, Dr. Celal Batur, for all his valuable suggestions, motivation and support that he has provided during my time as a Master's student at the department of Mechanical Engineering, The University of Akron. Whenever I was in a trouble or had a question related to my research work, it was easy for me to stop by his office. Dr. Batur is truly a mentor and it was a great pleasure working with him. I appreciate the contribution of Parker Hannifin Corporation for providing the Torsional Plant needed for project implementation.

I wish to extend my sincere thanks to the members of my thesis defense committee, Dr. Ajay Mahajan and Dr. Siamak Farhad, for their time and valuable advice on this study. I would like to acknowledge Mr. Clifford G Bailey, for his technical support for solving hardware and software related problems. Sincere appreciation is also given to Mrs. Stacy Meier and Ms. Shannon Skelton for their assistant to my graduate study. I would also like to acknowledge a number of friends at the University of Akron, Alexander Sorin, Hari Poudyal and Sudeep Adhikari for their support in my research work.

I also want to express my appreciation to my girlfriend for her love, support and always cheering me up. Last but not the least, I would like to give special thanks to my parents and sisters for their unconditional love, support and motivation for my inspiration.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

1.1 Introduction of Inertial Navigation Systems

In general, navigation is a process or technique to determine the position and orientation for controlling the motion of a body from one location to another location. Various traditional instruments such as the magnetic compass, clock, the chronometer etc. came to aid the navigator. As time progressed, other navigation system such as Pilotage, Dead reckoning, Celestial navigation, Radio navigation and Inertial navigation were used for navigation [1]. Among all of the above, the inertial navigation is used in a wide range of applications that include navigation of spacecraft, drones, missiles, submarines and ships. Also, recent development in the manufacture of MEMS devices have made it possible to use low-cost, small and light inertial navigation systems. MEMS inertial navigation are also used in areas such as robotics, human and animal motion capture, platform stabilization and camera stabilization [2]. The inertia is defined as a property of a body by virtue of which a body tends to maintain zero linear or angular acceleration unless disturbed by an external force or torque, respectively. Therefore, as the name suggests, inertial navigation systems are based on the principle of detecting linear acceleration using an accelerometer and detecting rotational rate using a gyroscope. Typically, inertial navigation consists of three orthogonal (*tri-axes*) accelerometers that read linear accelerations, three orthogonal gyroscopes that reads angular velocities and three orthogonal magnetometers that reads the components of the earth's magnetic field. Most of the inertial navigation systems have strap-down configuration. In strap-down configuration, the inertial sensors are

rigidly mounted on the device and the readings are based on the body frame rather than the reference frame that might be attached to the center of the earth or to a specific location on the surface of the earth.

1.1.1. Gyroscope

A conventional mechanical gyroscope consists of a rotating wheel on two gimbals allowing the rotation in all three axes. It is based on the principle of conservation of momentum i.e. the spinning wheel resists any changes in orientation. The conventional gyroscope shown in Figure 1.1 is mounted in an airplane. The gimbal makes an angle θ_1 with contact when the roll is zero. The gimbal makes an angle θ_2 with contact when there is a roll as shown in Figure 1.2. The difference between two contact angles $(\theta_1 - \theta_2)$ gives roll angle. However, mechanical gyroscope does not measure the angular rates. Instead it measures only angle. Mechanical gyroscope is expensive because it requires tight geometric dimensioning and tolerances and an intricate assembly.

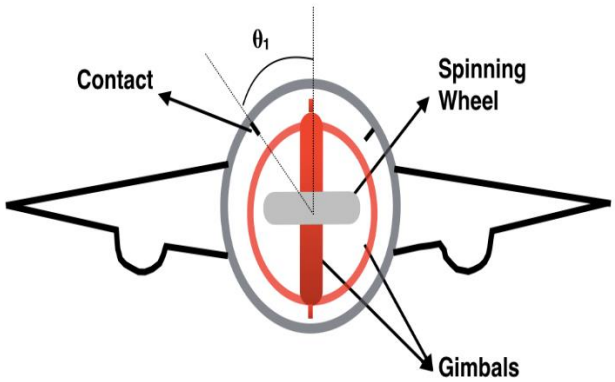

Figure 1.1.Conventional mechanical gyroscope mounted in an airplane (before roll)

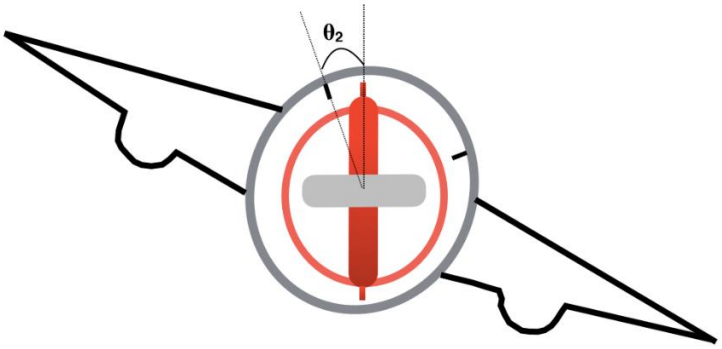

Figure 1.2.Conventional mechanical gyroscope mounted in an airplane (after roll)

In the last 20 years, MEMS gyroscopes have been introduced and rapid advancement have been made to produce low-cost and micro-sized gyroscopes by using silicon machining technique. MEMS gyroscope is based on Coriolis effect, which states that in a frame of reference rotating with angular velocity ω , a mass m moving with a translational velocity v experiences a force ($F_c = -2m(\omega \times v)$. This Coriolis force is measured using vibrating elements such as vibrating wheel, tuning fork or Piezoelectric (PZT) membrane [2], [3].

The MEMS gyroscope requires very low drive voltage and can be incorporated into other chips for the purpose of process control. The major disadvantage of MEMS gyroscope is that it is affected by noise, it may have bias and its axes may interfere with each other. The bias is an offset in the average readings when the device is stationary and the noise is an independent and uncorrelated random variations affecting the gyro. The bias and noise lead to accumulation of error if integrated i.e. the gyro output gets drifted over time period. The specification of gyroscope used in this thesis is given in Table 1.1. The low-cost gyroscope cannot be really used in an aircraft navigation since it will have high bias ($< 10°/hr$). The noise density that is shown in Table 1.1 refers to power spectral density which describe the distribution of signal over the frequency.

The noise of the gyroscope can be obtained by multiplying the noise density and square root of the band width. When the gyro signal is integrated to get an angle, the noise becomes gradually increasing which leads to inaccuracy in the results. The VECTORNAV gyroscope used in this study consists of three gyroscopes which measures angular velocity in *x*, *y* and *z axes*. These angular velocities are used as the input to the process equations for estimation of attitude.

Table 1.1. Specification of VECTORNAV gyroscope.

Range	$\pm 2000°/s$
In-Run Bias Stability	$< 10°/hr$
Linearity	$< 0.1\%\ FS$
Noise Density	$0.0035°/s/\sqrt{Hz}$
Bandwidth	256 Hz
Alignment Error	$\pm 0.05°$

1.1.2. Accelerometer

The accelerometer is a micro-electro mechanical device that measures linear acceleration when the device is subject to either static or dynamic force. MEMS accelerometer consist of proof mass or sensing element with attached comb fingers as shown in Figure (1.3). The proof mass moves freely in the direction where the force is applied. Due to the motion of proof mass, the displacement *x* changes which causes changes in the capacitance (C_0) between two parallel plates. The change in capacitance will lead to change in voltage (*V*). The relation between the capacitance of parallel plates and the displacement (*x)* is shown in equation (1.1). The change in displacement generates a spring force which is equivalent to force exerted on the proof mass as shown in (1.2).

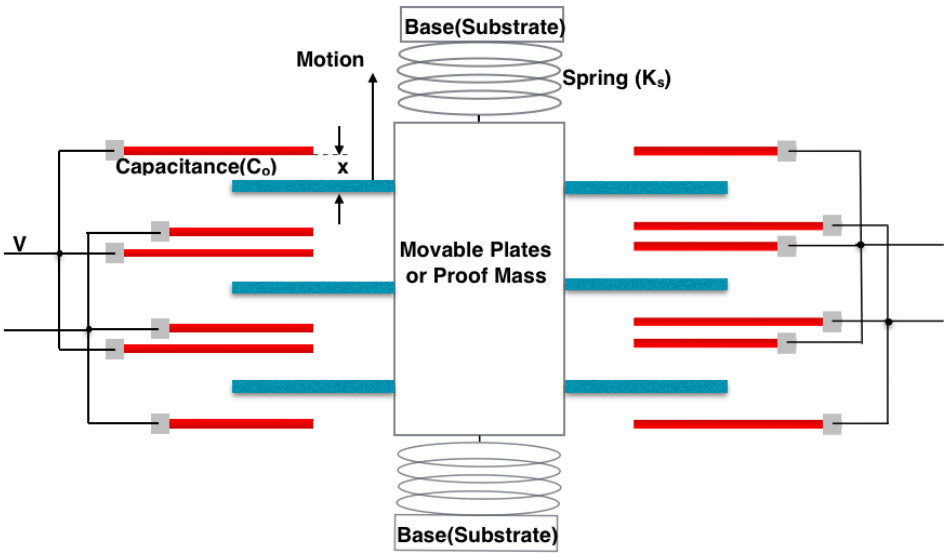

Figure 1.3. Working Principle of MEMS gyroscope.

$$C_0 = \varepsilon \frac{A}{x} \tag{1.1}$$

$$F_{spring} = F_m \quad | \quad m * a = -xK_s \tag{1.2}$$

Where, ε is permittivity of a material between two parallel plates, A is an effective area between the plates and 'a' is linear acceleration in the direction of the motion. Another type of MEMS accelerometer is the Piezo-resistive (PR) accelerometer in which piezo-resistive material such as germanium, polycrystalline silicon, silicon carbide and amorphous silicon is a sensing element. The PR accelerometer is the first micro-machined accelerometer that works by measuring the electrical resistance when the sensing element is subjected to mechanical stress [4].

When the axis of an accelerometer is placed vertically downward, the accelerometer reads gravitational acceleration (g) in that axis. Therefore, any motion in vertically in downward direction will have sum of linear acceleration and gravitational acceleration. This acceleration is known as Pseudo-acceleration. In few applications, user double integrates accelerometer reading to get a position. Since the accelerometer also reads gravitational acceleration, it needs to be subtracted if user is determining the

position from acceleration. The VECTORNAV device that is used in this study consist of three accelerometers that measures linear acceleration in *x, y* and *z directions.* These acceleration readings are used in measurement equation to update previously estimated states. Since these accelerations are measured in the body coordinate frame, it needs to be converted to the reference coordinate frame using a rotation matrix. The major disadvantage of MEMS accelerometer is that it is affected by bias and noise. The specifications of VECTORNAV accelerometer are listed in Table 1.2

Table 1.2. Specification of VECTORNAV accelerometer.

Range	$\pm 16g$
In-Run Bias Stability	$< 0.04mg$
Linearity	$< 0.5\%\ FS$
Noise Density	$0.14mg/\sqrt{Hz}$
Bandwidth	260 Hz
Alignment Error	$\pm 0.05°$

1.1.3. Magnetometer

The magnetometer is a device that measures magnetic strength of the local magnetic field. The local magnetic field is combination of earth's magnetic field and magnetic field due to nearby objects. For any location, the earth's magnetic field is represented by three terms: magnetic intensity, inclination and declination. If a tri-axis magnetometer is placed in any location free from external magnetic disturbance, the earth's magnetic intensity or earth's magnetic field is vector sum of magnetic field read by each magnetic sensor in respective axis as shown in (1.3). The inclination is an angle made by the earth's magnetic field with a horizontal plane for a given location. The

declination is the angle between the earth's magnetic pole and geographical pole for given location. The earth's magnetic pole and geographical pole is shown in Figure 1.3. The inclination, declination and magnetic intensity is shown in Figure 1.4. The MEMS magnetometer is affected by hard and soft iron biases. The magnetic bias due to hard iron effect refers to presence of permanent magnets or power supply wires attached to the sensor frame. The hard iron effect causes permanent bias in the sensor output. The magnetic bias due to soft iron effect refers to distortions due to induced magnetic field. This type of distortion is caused by ferromagnetic materials such as nickel and iron [5]. The specification of VECTORNAV accelerometer are listed in Table 1.4.

Let, h_X, h_y and h_Z be the magnetic field readings in *X-axis* (geographical North), *Y-axis* (geographical East) and *Z-axis* (down) respectively as measured by *tri-axis* magnetometer. The horizontal intensity (H_{XY}) is cosine component and h_Z is sine component of earth's magnetic intensity (*H*) and are given by (1.4). The magnetic reading in *X-axis* and *Y-axis* are given by (1.5).

$$H = \sqrt{{h_X}^2 + {h_Y}^2 + {h_Z}^2} \tag{1.3}$$

$$H_{XY} = H\cos\emptyset_I \quad | \quad h_z = H\sin\emptyset_I \tag{1.4}$$

$$h_X = H_{XY}\cos\emptyset_D \quad | \quad h_Y = H_{XY}\sin\emptyset_D \tag{1.5}$$

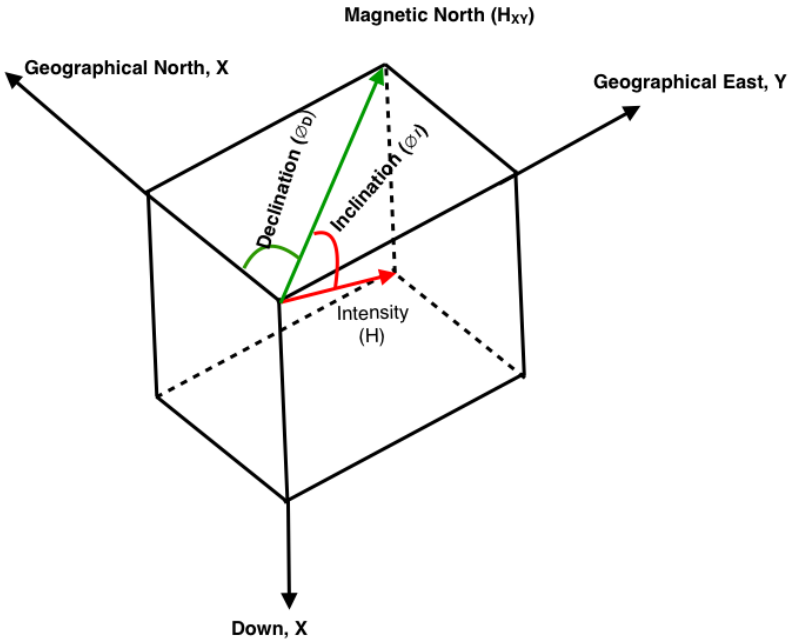

Figure 1.4. The inclination, declination and earth's magnetic intensity

Table 1.3. Earth's magnetic field at the University of Akron, OH using World Magnetic Model (WMM) 2015

Latitude	41.07591^0 North
Longitude	81.51380^0 West
Altitude	327 m
Declination (+East | -West)	-8.4034^0
Inclination (+Down | -Up)	67.9759^0
Horizontal intensity	0.19794 gauss
North component (+North | -South)	0.19581 gauss
East component (+East | -West)	-0.2892 gauss
Vertical component (+Down | -Up)	0.48933 gauss
Total Field	0.52785 gauss

The earth's magnetic field depends on latitude and longitude of a given location. For given location, if the tri-axis magnetometer is placed in North East Down frame (NED frame), the north, east and down component of magnetometer reading is given by Table 1.3. If the tri-axes magnetometer is placed in magnetic disturbance free region and in any orientation, the equation (1.4) and (1.5) can be solved to get the total magnetic field, inclination and declination. In practice, for attitude estimation the reference magnetic field vector is the local magnetic field vector. The local magnetic field vector changes with location. Therefore, it necessary to update the reference magnetic field with change in location. For example, an airplane flying from Cleveland, Ohio to Columbus, Ohio should update the reference magnetic field of each location on its route. The update is being done online using the magnetic readings. For any location, the reference magnetic field can be updated either by using World Magnetic Model chart or by determining the total magnetic field, inclination and declination from magnetometer reading.

The most of the MEMS magnetometer is based on the principle of Hall Effect. According to Hall Effect, the current carrying conductor will accumulate charges in the direction mutually perpendicular to applied magnetic field and direction of charge. The Figure 1.5 shows a current I flowing through the conductor. The Figure 1.6 shows deflection of negative charge $-q$ when the conductor is subjected to magnetic field B. The deflection of charge is due to the Lorentz force given by (1.6). The direction of force is determined using Fleming's left hand rule as shown in Figure 1.7. The voltage reading across the conductor varies depending on the strength of applied magnetic field and electric field.

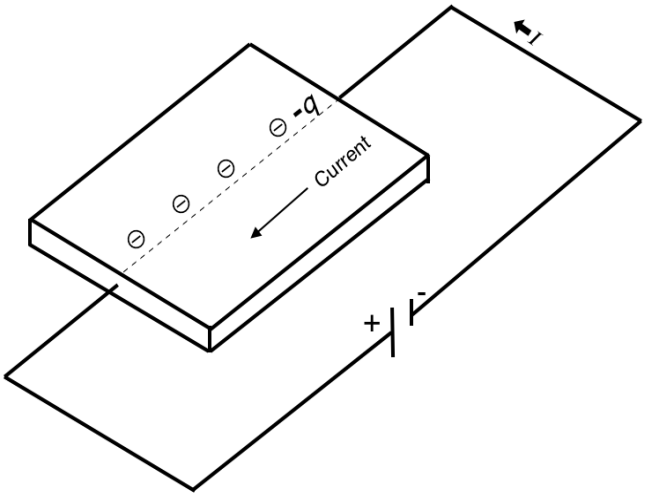

Figure 1.5. The charge flow in a conductor carrying current.

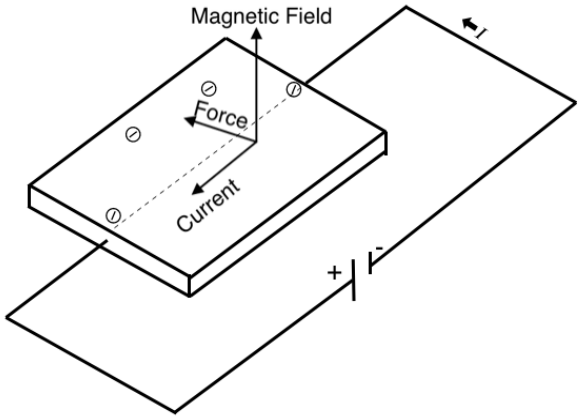

Figure 1.6. The charge flow in a conductor carrying current subjected to the magnetic field.

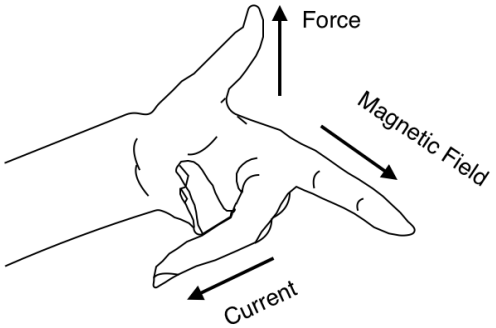

Figure 1.7. Fleming's left hand Rule

$$F = q(E + v \times B) \tag{1.6}$$

Where, *F* is Lorentz force, *q* is charge, *v* is velocity of charge, *E* and *B* are electric and magnetic fields respectively.

Table 1.4. Specification of VECTORNAV accelerometer.

Range	$\pm 2.5\ Gauss$
Linearity	$< 0.1\%$
Noise Density	$140 \mu Gauss/\sqrt{Hz}$
Bandwidth	200 Hz
Alignment Error	$\pm 0.05°$

1.2. Literature Review

The state estimation is an online and real time filtering technique that uses the process and the measurement equations shown below in order to provide reliable estimates of the states [6].

$$x_{k+1} = f_k x_k + w_k \tag{1.7}$$

$$z_k = h_k x_k + v_k \tag{1.8}$$

Where, *k* is a discrete time sample, $x \in \mathbb{R}^{n\times 1}$ is state variable, $z \in \mathbb{R}^{m\times 1}$ is measurement, $f \in \mathbb{R}^{n\times n}$ is non-linear state function, $h \in \mathbb{R}^{m\times n}$ is non-linear measurement function, $w \in \mathbb{R}^{n\times 1}$ is white process noise with covariance matrix $Q \in \mathbb{R}^{n\times n}$, $v \in \mathbb{R}^{m\times 1}$ is white measurement noise with covariance matrix $R \in \mathbb{R}^{m\times m}$.

These estimated states x of the equation (1.7) are typically used as elements of a feedback for control strategy to force the process output to follow what is desired. In this research field, Liang investigated application of *EKF* (extended Kalman filter) for estimating accelerations and yaw rates of a vehicle [7]. The Kalman filter is an iterative technique for linear system that predicts and correct the states based on the available measurements [8]. The prediction step includes prediction of states using (1.9) and

prediction of error covariance matrix using (1.10). The update step includes update of states and error covariance matrix using (1.11) and (1.12). The Kalman gain is determined using (1.13). The extended Kalman filter algorithm is explained detail in Chapter 4.

$$\hat{x}_{k|\mathrm{k}-1} = f(\hat{x}_{k-1|\mathrm{k}-1}, k) \tag{1.9}$$

$$P_{k|\mathrm{k}-1} = F\big(k, \hat{x}_{k-1|\mathrm{k}-1}\big)P_{k-1|\mathrm{k}-1}F^{T}\big(k, \hat{x}_{k-1|\mathrm{k}-1}\big) + Q_k \tag{1.10}$$

$$K_k = P_{k|\mathrm{k}-1}H^{T}\big(k, \hat{x}_{k|\mathrm{k}-1}\big)\big\{H\big(k, \hat{x}_{k|\mathrm{k}-1}\big)\, P_{k|\mathrm{k}-1}H^{T}\big(k, \hat{x}_{k|\mathrm{k}-1}\big) + R_k\big\}^{-1} \tag{1.11}$$

$$\hat{x}_{k|\mathrm{k}} = \hat{x}_{k|\mathrm{k}-1} + K_k\big(z_k - h(\hat{x}_{k|\mathrm{k}-1}, k)\big) \tag{1.12}$$

$$P_{k|\mathrm{k}} = P_{k|\mathrm{k}-1} - K_k H\big(\hat{x}_{k|\mathrm{k}-1}, k\big)P_{k|\mathrm{k}-1} \tag{1.13}$$

Where, $\hat{x}_{\mathrm{k\text{-}1/k\text{-}1}}$ is the estimate of the state variable from previous time instance, $P_{k1|k\text{-}1}$ is covariance error matrix from previous time instance, $\hat{x}_{k|k\text{-}1}$ is estimate of state form prediction step, $P_{k|k\text{-}1}$ is covariance error matrix from prediction step, $\hat{x}_{k|k}$ is state variable obtained from update state and passed on to the next instance, $P_{k|k}$ is covariance error matrix obtained from update step and passed on to next instance [9].

For the non-linear systems extended Kalman filter is used that linearizes the state or measurement equation around a given normal operating point. Another approach that is widely used for state estimation is *Particle Filtering.* John et al. presented attitude estimation using particle filter [10]. Particle filter is an estimation of states for non-linear systems corrupted with noise. This technique is based on probabilistic representation of states by a set of samples known as *particles.* In the particle filtering technique, at prediction step, the particles are estimated using the equation (1.14) such that their weights remain unchanged. The particles are updated based on the weights of the process noise calculated by using the equation (1.15).

$$x_{k+1}^{(i)} = f(x_k^{(i)}, w_k^{(i)}) \tag{1.14}$$

$$w_{k+1}^{(i)} = w_k^{(i)} . p_{k+1}(\bar{z}_{k+1}|x_{k+1}^{(i)}) \tag{1.15}$$

$$p_{k+1}\left(\bar{z}_{k+1}\middle|x_{k+1}^{(i)}\right) = p_{v_{k+1}}(\bar{z}_{k+1} - h_k{}^{(i)}(x_{k+1})) \tag{1.16}$$

$$\bar{z}_{k+1} = h_{k+1}(x_{k+1}) + v_{k+1} \tag{1.17}$$

Where, $x_k^{(i)}$ is state variable at time step *k* and *i* is number of samples, $\bar{z}_{k+1}$ is the measurement at time step *k+1* given by the equation (1.11), The $h_k$is function of current states and the readings from measurement devices. The process and measurement noise is denoted by w_k and v_k and their respective probability distribution function is denoted by p_{w_k} and p_{v_k}. In the prediction step the process noise is drawn according to p_{w_k} and in the update step the measurement noise is drawn according to p_{v_k}.

Oliver provided introduction to inertial navigation and investigated the error characteristics of inertial navigation system [2]. In his work, he identified the propagation of error caused by the gyro noise. Also, he showed that integrating twice the noise in accelerometer caused rapidly growing error in estimated position. He concluded that it is appropriate to fuse gyroscope with magnetometer to reduce drift in navigation. He successfully reduced the drift of 50 m to 5 m (for 60 seconds) after fusing the gyroscope with magnetometer. Dinuka et al. designed an EKF based algorithm to estimate the velocity, yaw angle and roll angle of quadrotor micro aerial vehicle by integrating tri-axis gyroscope and two-axis accelerometer [11]. In their estimation, they included gyro bias as a process equation and defined process noise as a function of current states, sampling time and biases.

Matthew et al. designed a new navigation tool that rely on inertial sensor and optical flow information [12]. The velocity and attitude state estimation were performed

using the Unscented Kalman filter. To avoid inaccuracy in attitude estimation due to body acceleration of an aircraft, additional state is used. The additional state is the ground velocity of an aircraft captured by laser range finder. This technique is effective in reducing the inertial sensors drift. Unscented information filter eliminates unwanted information by using minimum mean square error estimation[13].

Zhang et al. constructed an attitude measurement system based on an extended Kalman filter for multi-rotors by fusing *tri-axis* gyroscope, accelerometer and magnetometer [14]. They introduced quaternion and gyroscope bias as a state vector, the acceleration and the magnetic field as measurement vector. The state vector is minimum set of variables that represents system dynamics. The gyroscope bias state vector refers to the bias in three orthonormal gyroscopes. Quaternion refers to complex algebra that represents the orientation of a body in three dimensional space. To improve accuracy in attitude estimation, the gyroscope bias and drift are detected and corrected by measurement vector.

In their work they discussed covariance tuning. Covariance tuning is a procedure to determine the optimized process and measurement noise matrix to get better estimation performance. In the extended Kalman filter, the confidence of estimated state is based on error covariance matrix. The prediction of error covariance matrix relies on process noise covariance matrix and the update of error covariance matrix relies on measurement noise covariance matrix. In their work, they determined the least mean square error for different values of process noise covariance matrix and measurement noise covariance matrix by varying initial error covariance matrix. To start the extended Kalman filter algorithm, initial states should be defined and the error covariance matrix should be guessed and adjusted. It should be noted that the initial

guess for error covariance matrix should not be zero because the estimated states for all iterations will be zero for zero initial states.

1.3. Thesis Outline

Chapter 1 is an introduction to inertial navigation system. This chapter briefly describes the component of inertial navigation system and their working principle. In this chapter, the contributions made in attitude estimation is highlighted.

Chapter 2 is an introduction to orientation representation technique. This chapter describes Euler Angles, rotation matrix and quaternions. In this chapter, the advantages and disadvantages of Euler Angles and quaternions is discussed.

Chapter 3 introduces biases and noise in the IMU (Inertial Measurement Unit) units. This chapter explains the procedure for calibration of the gyroscope, the magnetometer and the accelerometer.

Chapter 4 introduces the extended Kalman filter technique. This chapter presents state and measurement equation for quaternion based attitude estimation. The algorithm for extended Kalman filter is elaborated in this chapter. The simulated results for steady state condition using this algorithm is studied.

Chapter 5 presents modified extended Kalman filter in which state equation and process noise covariance is modified. This chapter discuss the reason for these modifications. The algorithm is used for state estimation under the body acceleration. The simulated results are discussed here.

Chapter 6 summarizes the achievement of this dissertation and suggest future research direction.

CHAPTER II

EULER ANGLES, ROTATION MATRIX AND QUATERNION

2.1 Introduction

The orientation of a rigid body in three dimensional space can be represented using different techniques. Euler angles representation is the most common and an understandable way to represent the orientation of a rigid body. This technique uses three sets of angle of rotation about mutually orthogonal axes. However, Euler angles are less accurate in comparison to quaternions when used to integrate incremental change in orientation over time. Certain Euler angles function have singularities in a sense that, starting from the same attitude, two different Euler angles sequence may end up with the same attitude [15]. Another way to represent the orientation of rigid body is the quaternion. Quaternion is a vector that consists three imaginary number terms and one real number. It is rather difficult to give an intuitive physical meaning to quaternions. Most of the researchers suggest to use quaternion for orientation determination because their algebraic properties make them efficient, faster and computationally robust [16]. A body frame (B), fixed to a moving body, is known as the local co-ordinate frame or body co-ordinate frame. The reference frame (*N*) on the other hand is external frame which lies outside the body. To determine the orientation with respect to reference frame, the body coordinates is multiplied by the rotation matrix C_B^N.

2.2 Euler Angles

Euler angles are the rotational angles about three principal axis *X*, *Y* and *Z* which are orthogonal to each other. Euler angles have three rotations in the sequence. The Euler angles sequence selection can be done in twelve different ways depending on the order of chosen principal axis. Euler angle sets are categorized into two different types: symmetric sets and asymmetric sets. The symmetric sets or Proper Euler angles have the first and last rotation about same axis. The symmetric sets are *X-Y-X*, *X-Z-X*, *Y-X-Y*, *Y-Z-Y*, *Z-X-Z* and *Z-Y-Z*. The asymmetric sets or Tait-Bryan angles has rotations about all three distinct axes. The asymmetric sets are *X-Y-X*, *X-Z-Y*, *Y-X-Z*, *Y-Z-X*, *Z-X-Y* and *Z-Y-X*.

The most common and popular terminology used in aviation: *yaw*, *pitch* and *roll* are also known as Euler angles. In an airplane, the yaw angle is the rotation of nose left or right about an axis running from top to bottom. The pitch angle is the rotation of nose up or down about an axis running from wing to wing. The roll angle is the rotation of nose with respect to an axis running from nose to tail. The yaw, pitch and roll angles of an airplane is shown in Figure 2.1.

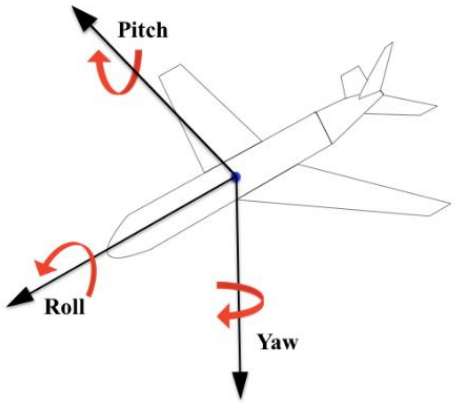

Figure 2.1 Yaw, pitch and roll in an airplane

According to Euler Angles convention, the X-axis, Y-axis and Z-axis are represented by numbers 1, 2 and 3 respectively. The orientation described by the rotation about Z-axis, Y-axis and X-axis in a sequence is '*Z-Y-X*' Euler angle notation or '321' Euler angle notation. The Euler angle set used in this thesis is '321' and it is shown in the Figure 2.2.

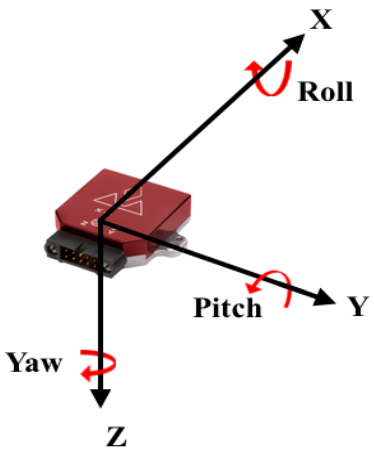

Figure 2.2 Example of 3-2-1 Euler angles set angle is rotation about the y-axis and the roll angle is rotation about the x-axis.

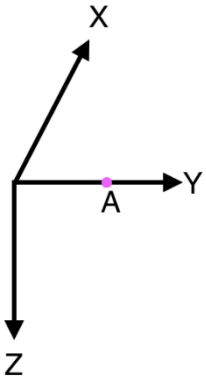

Figure 2.3.a. Reference frame with point A in Y-axis

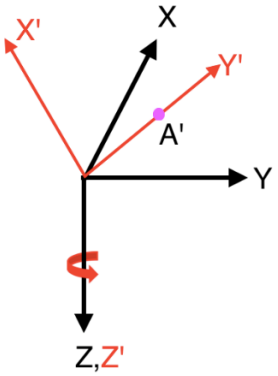

Figure 2.3.b. Rotation about Z-axis by -45^{o}

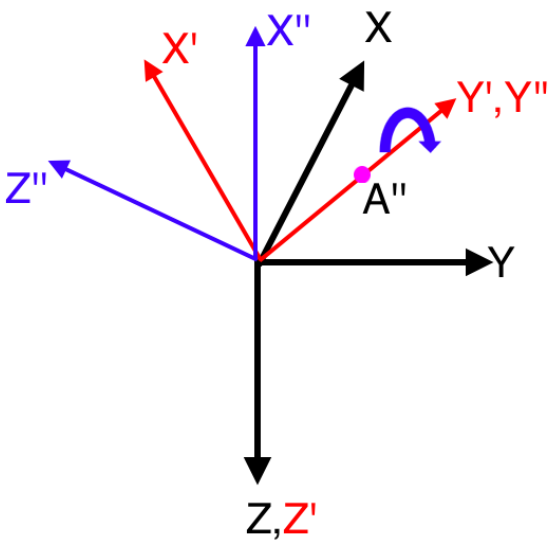

Figure 2.3.c. Rotation about Y-axis by +90^{o}

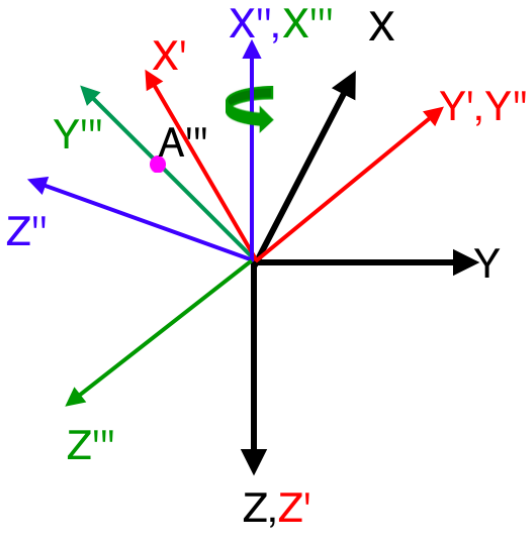

Figure 2.3.d. Rotation about X-axis by +90^{o}

Let A be a point in Y-axis of body frame XYZ as shown in figure 2.3.a. Considering '321' Euler angle set, rotate -45^{0} about Z-axis as shown in Figure 2.3.b. Euler angle rotation uses right hand thumb rule for sign convention i.e. if axis is pointed towards the direction of thumb, clockwise rotation is positive and anti-clockwise rotation is negative. The rotation about Z-axis orients the point A to point A' and the X, Y and Z axis is changed to X', Y' and Z' respectively as shown in Figure 2.3.b. Again, rotate +90^{0} about Y' axis as shown in Figure 2.3.c. Now, the point A' moves to point A'' and the X', Y' and Z' axis is changed to X'', Y'' and Z'' respectively as shown in Figure 2.3.c. Further, rotate +90^{0} about X'' axis as shown in Figure 2.3.d. Now, the point A'' moves to A''' and the X'', Y'' and Z'' axis is changed to X''', Y''' and Z''' respectively as shown in Figure 2.3.d. Therefore, the point A is moved to the point A''' by using three different rotations in '321' sequence. However, the point A can be moved to the point A''' by using another three different rotations in '321' sequence.

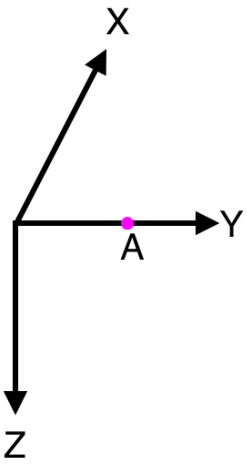

Figure 2.4.a. Reference frame with point A in Y-axis

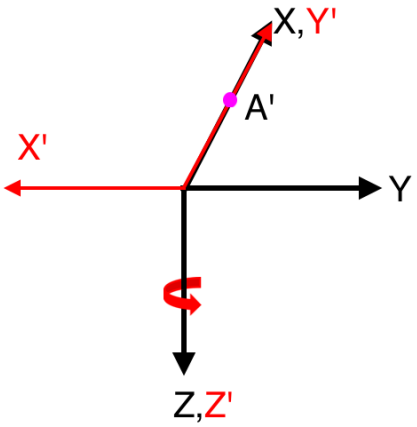

Figure 2.4.b. Rotation about Z-axis by -90^{o}

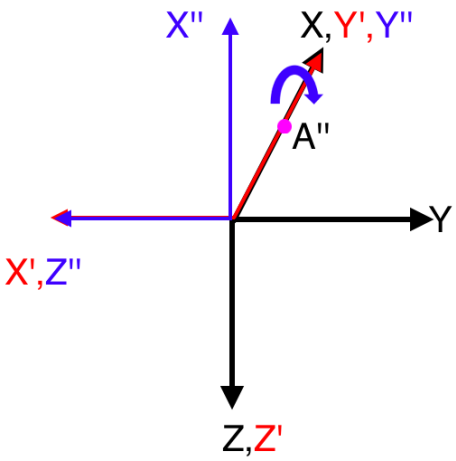

Figure 2.4.c. Rotation about Y-axis by +90^{o}

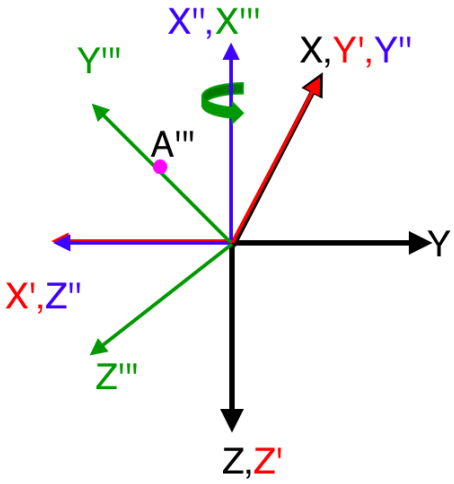

Figure 2.4.d. Rotation about X-axis by +45^{o}

Let A be the point in Y-axis as shown in figure 2.4.a. First, the point A is rotated -90^{0} about Z-axis as shown in Figure 2.4.b. Second, the point A' is rotated +90^{0} about Y'-axis as shown in Figure 2.4.c. Finally, the point A'' is rotated +45^{0} about X''-axis as shown in Figure 2.4.d. The Figure 2.3.a to Figure 2.3.d explains orientation of point A to point A''' using Euler angel set as: -45^{0} about Z-axis, +90^{0} about Y' axis and +90^{0} about X''-axis. Similarly, Figure 2.4.a. to Figure 2.4.d. explains orientation of same point A to same point A'' using Euler angle set as: -90^{0} about Z-axis, +90^{0} about Y' axis and +45^{0} about

X’’-axis. It is observed that the point A’’’ in Figure 2.3.c. and Figure 2.4.d is parameterized by different Euler angles although point the same point A is oriented to same point A’’’ in both Figures. Therefore, the orientation defined by the Euler angels is not unique. Also, the Euler angle lead to the singularity. The singularity occurs if determinant of rotation matrix is zero. The symmetric sets lead to singularity if the second rotation is equal to $\pm n\pi$ and the asymmetric sets lead to singularity if the second rotation is equal to $\pm n\frac{\pi}{2}$, n is any integer (n= 0, 1, 2,…)[17]. In figure 2.3.c and Figure 2.4.c, the rotation about Y’- axis by $+90^0$ leads to the singularity.

2.3. Rotation Matrix

The rotation matrix (*C*) is a real orthogonal matrix that operates on vector to produce a rotated vector, while co-ordinate axes are held fixed. The rotation matrix converts attitude measured in body co-ordinate to the attitude measured in reference co-ordinate. The rotational matrix (C_B^N) is orthogonal i.e. the determinant of C_B^N is unity. It means all the row and column vectors should be perpendicular to each other. However, the rotation matrix may deviate from orthogonality and this may result errors. To avoid these errors and to guarantee C_B^N to be pure rotational matrix, normalization is performed. The technique for normalization is mentioned in [18] .

The attitude of an airplane with respect to body co-ordinate is shown in Figure 2.5. The *X-axis, Y-axis* and *Z-axis* are the principal axes of reference frame located on the earth. The *X’-axis, Y’-axis* and *Z’-axis* are the principal axes of body co-ordinate frame located at the mass center of an airplane. In order to determine the attitude of an airplane with respect to the reference frame, the attitude determined with respect to body co-ordinate has to be multiplied with rotation matrix.

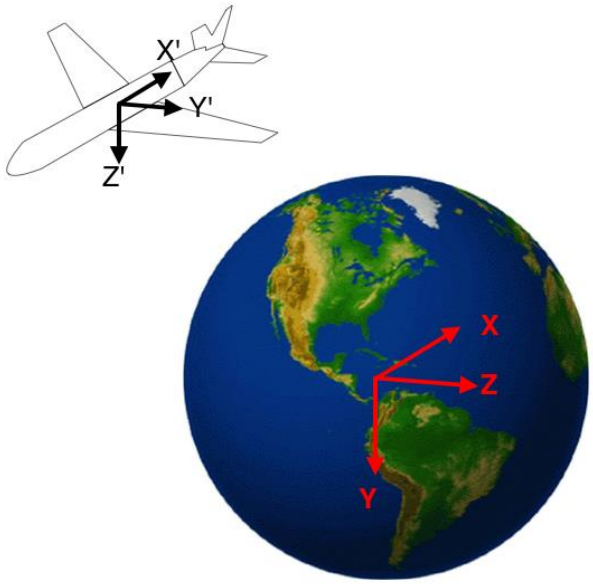

Figure 2.5. Body co-ordinate located on an airplane and reference co-ordinate on the earth

The rotation matrix can be derived from the Euler angles rotation. For '*Z-Y-X'* Euler angles set, assume ψ be the angle of rotation about *Z-axis*, θ be the angle of rotation about *Y'-axis* and ϕ the angle of rotation about *X''-axis*. Let A be a point in *X-Y* plane and A is rotated about *Z-axis* by an angle ψ as shown in Figure 2.6. Let, *r* be the magnitude of $\overrightarrow{OA}$. The rotation matrix to define a rotation of A (*x, y, z*) to A' (*x', y', z'*) using Euler anglesψ, θ and ϕ are given by (2.8), (2.16) and (2.24) respectively.

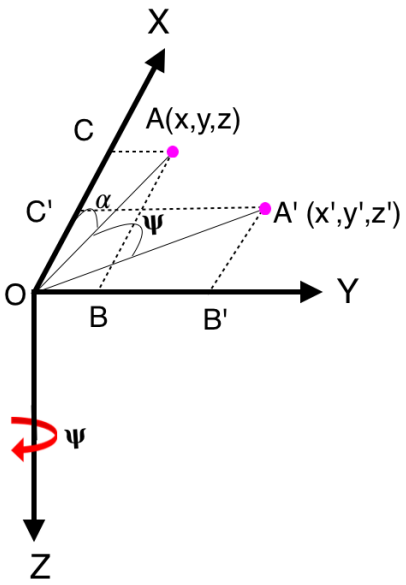

Figure 2.6. Rotation of point A by an angle ψ in Z-Y-X Euler angle set.

From Figure 2.6,

$$\text{OC}=\left|\overrightarrow{OA}\right| \sin\alpha \Rightarrow x = r\sin\alpha \tag{2.1}$$

$$\text{OB}=\left|\overrightarrow{OA}\right| \cos\alpha \Rightarrow y = r\cos\alpha \tag{2.2}$$

$$\begin{aligned}\text{OC}' = \left|\overrightarrow{OA'}\right| \sin(\psi+\alpha) \ &\Rightarrow x' \\ = r\sin(\psi+\alpha) &\Rightarrow x' = r\sin\alpha\cos\psi + r\cos\alpha\sin\psi\end{aligned} \tag{2.3}$$

$$\begin{aligned}\text{OB}' = \left|\overrightarrow{OA'}\right| \cos(\psi+\alpha) \ &\Rightarrow y' \\ = r\cos(\psi+\alpha) &\Rightarrow y' = r\cos\alpha\cos\psi - r\sin\alpha\sin\psi\end{aligned} \tag{2.4}$$

Substituting (2.1) and (2.2) in (2.3),

$$x' = x\cos\psi + y\sin\psi \tag{2.5}$$

Substituting (2.1) and (2.2) in (2.4),

$$y' = y\cos\psi - x\sin\psi \tag{2.6}$$

$$z' = z \tag{2.7}$$

From (2.5), (2.6) and (2.7),

$$\begin{bmatrix}x'\\y'\\z'\end{bmatrix} = \begin{bmatrix}\cos\psi & \sin\psi & 0\\ -\sin\psi & \cos\psi & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}x\\y\\z\end{bmatrix}$$

$$C_z = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.8}$$

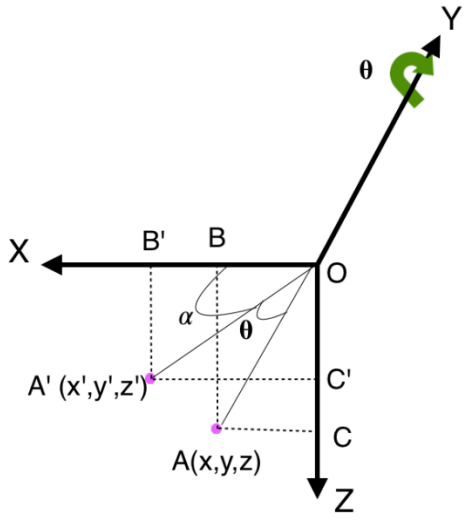

Figure 2.7. Rotation of point A by an angle θ in Z-Y-X Euler angle set.

From Figure 2.7,

$$\text{OB}=\left|\overrightarrow{OA}\right| \cos\alpha \Rightarrow x = r\cos\alpha \tag{2.9}$$

$$\text{OC}=\left|\overrightarrow{OA}\right| \sin\alpha \Rightarrow z = r\sin\alpha \tag{2.10}$$

$$\begin{aligned} \text{OB}' = \left|\overrightarrow{OA'}\right| \cos(\theta+\alpha) \ &\Rightarrow x' \\ = r\cos(\alpha+\theta) &\Rightarrow x' = r\cos\alpha\cos\theta - r\sin\alpha\sin\theta \end{aligned} \tag{2.11}$$

$$\begin{aligned} \text{OC}' = \left|\overrightarrow{OA'}\right| \cos(\theta+\alpha) \ &\Rightarrow z' \\ = r\sin(\alpha+\theta) &\Rightarrow z' = r\sin\alpha\cos\theta + r\cos\alpha\sin\theta \end{aligned} \tag{2.12}$$

Substituting (2.9) and (2.10) in (2.11),

$$x' = x\cos\theta - z\sin\theta \tag{2.13}$$

Substituting (2.9) and (2.10) in (2.12),

$$z' = z\cos\theta + x\sin\theta \tag{2.14}$$

$$y' = y \tag{2.15}$$

From (2.13), (2.14) and (2.15),

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$C_Y = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \tag{2.16}$$

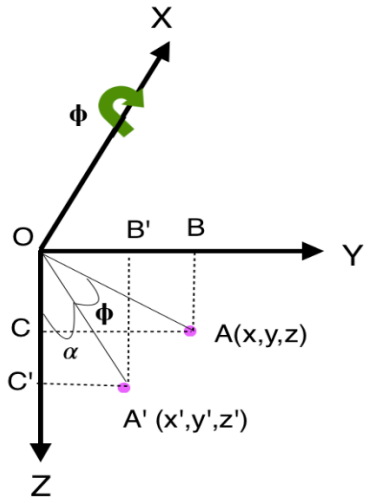

Figure 2.8. Rotation of point A by an angle ϕ in Z-Y-X Euler angle set.

From Figure 2.8,

$$\text{OB}=\left|\overrightarrow{OA}\right| \cos\alpha \Rightarrow y = r\sin\alpha \tag{2.17}$$

$$\text{OC}=\left|\overrightarrow{OA}\right| \sin\alpha \Rightarrow z = r\cos\alpha \tag{2.18}$$

$$\begin{aligned} \text{OB}' = \left|\overrightarrow{OA'}\right| \cos(\phi+\alpha) \ &\Rightarrow y' \\ = r\sin(\phi+\alpha) &\Rightarrow y' = r\sin\alpha\cos\phi + r\cos\alpha\sin\phi \end{aligned} \tag{2.19}$$

$$\begin{aligned} \text{OC}' = \left|\overrightarrow{OA'}\right| \cos(\phi+\alpha) \ &\Rightarrow z' \\ = r\cos(\phi+\alpha) &\Rightarrow z' = r\cos\alpha\cos\phi - r\sin\alpha\sin\phi \end{aligned} \tag{2.20}$$

Substituting (2.17) and (2.18) in (2.19),

$$y' = y\cos\phi + z\sin\phi \tag{2.21}$$

Substituting (2.17) and (2.18) in (2.20),

$$z' = z\cos\phi - y\sin\phi \tag{2.22}$$

$$x' = x \tag{2.23}$$

From (2.21), (2.22) and (2.23),

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\phi & sin\phi \\ 0 & -sin\phi & cos\phi \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$C_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\phi & sin\phi \\ 0 & -sin\phi & cos\phi \end{bmatrix} \tag{2.24}$$

According to 'Z-Y-X' Euler angle convention, any vector is first rotated by an angle ψ about *Z-axis* followed by rotation of an angle θ about *Y'-axis* and finally followed by rotation of an angle ϕ about *X''-axis*. If a vector is rotated in 'Z-Y-X' Euler angle set, the rotation matrix (C_N^B), reference frame to body frame, is product of rotation about *Z-axis* followed by *Y'-axis* and then followed by rotation about *X''-axis*. The multiplication of individual rotation matrix should be done in reverse way[19].

$$C_N^B = C_x C_y C_z$$

$$C_N^B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\phi & sin\phi \\ 0 & -sin\phi & cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$C_N^B = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ \sin\phi\sin\theta & \cos\phi & \cos\theta\sin\phi \\ \cos\phi\sin\theta & -\sin\phi & \cos\theta\cos\phi \end{bmatrix} \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$C_N^B = \begin{bmatrix} \cos\psi\cos\theta & \cos\theta\sin\psi & -\sin\theta \\ -\cos\phi\sin\psi + \cos\psi\sin\theta\sin\phi & \cos\psi\cos\phi + \sin\psi\sin\theta\sin\phi & \cos\theta\sin\phi \\ \cos\psi\cos\phi\sin\theta + \sin\psi\sin\phi & -\cos\psi\sin\phi + \cos\phi\sin\psi\sin\theta & \cos\theta\cos\phi \end{bmatrix} \tag{2.25}$$

The rotation matrix C_N^B in (2.25) represents rotation from a reference frame to a body frame. To get the rotation matrix from a body frame to reference frame, the inverse

of C_N^B should be determined. The inverse of a rotation matrix is also a transpose of it [20]. The rotation matrixC_B^N, a body to reference frame rotation is given by (2.26).

$$C_B^N = (C_N^B)^{-1} = (C_N^B)^T$$

$$C_B^N = \begin{bmatrix} \cos\psi\cos\theta & \cos\theta\sin\psi & -\sin\theta \\ -\cos\phi\sin\psi + \cos\psi\sin\theta\sin\phi & \cos\psi\cos\phi + \sin\psi\sin\theta\sin\phi & \cos\theta\sin\phi \\ \cos\psi\cos\phi\sin\theta + \sin\psi\sin\phi & -\cos\psi\sin\phi + \cos\phi\sin\psi\sin\theta & \cos\theta\cos\phi \end{bmatrix}^T$$

$$C_B^N = \begin{bmatrix} \cos\psi\cos\theta & -\cos\phi\sin\psi + \cos\psi\sin\theta\sin\phi & \cos\psi\cos\phi\sin\theta + \sin\psi\sin\phi \\ \cos\theta\sin\psi & \cos\psi\cos\phi + \sin\psi\sin\theta\sin\phi & -\cos\psi\sin\phi + \cos\phi\sin\psi\sin\theta \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix} \quad (2.26)$$

The rotational matrix C_B^N is also called Directional Cosine Matrix (DCM) because all the element in rotation matrix are cosine of angles between body frame and reference frame. For linearization of Euler angle functions, the Jacobian of Euler angle functions has to be determined. The cosine and sine terms are approximated to *1* and *0* respectively and setting higher order terms to *0*. Typically, this approximation is valid for small angles less than 10^0[15].

2.4. Gimbal Lock

The gimbal is a ring that can freely rotate about its axis of suspension. The gimbals are mounted together to produce independent rotations about their respective axes of suspensions. For Euler angle set, three sets of gimbals are used which represent *yaw, pitch* and *roll* allowing three degree of freedom. When the axes of two gimbals are parallel, there is loss of one degree of freedom, causing three-dimensional space to degenerate into two dimensional space. At this situation, the two gimbals become co-planar and control torque becomes ineffective to stabilize the platform. The rotation about an axis that is normal to

the plane of co-planar gimbals causes locking of platform. This is known as gimbal lock [21].

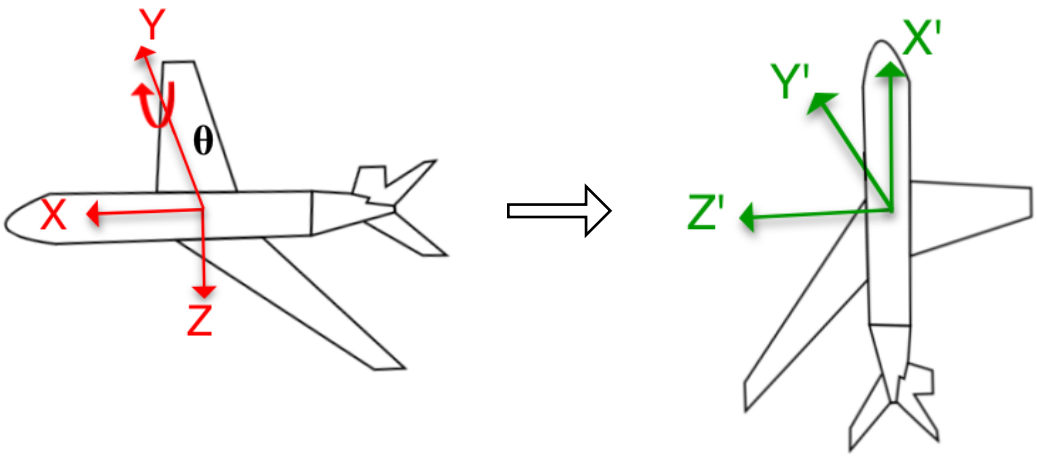

Figure.2.9 Rotation of an airplane plane in by pitch angle θ about *Y-axis*

Let, an airplane shown in Figure 2.9 is rotated about *Y-axis* by 90^o *pitch* angle(θ). It means an airplane flying in a horizontal plane rotates such that the nose of an airplane faces vertically upward. In this case, the rotation matrix given in (2.25) is re-written as in (2.27). From (2.27), it can be noted that the change in either *yaw* (ψ) or *roll* (ϕ) angle will not have any effects on the rotation of z-axis. In general, the z-axis is locked if the pitch angle is either 90^o or 270^o. This is also known as gimbal effect.

$$C_N^B = \begin{bmatrix} \cos\psi\cos 90^o & \cos 90^o \sin\psi & -\sin 90^o \\ -\cos\phi\sin\psi + \cos\psi\sin 90^o \sin\phi & \cos\psi\cos\phi + \sin\psi\sin 90^o\sin\phi & \cos 90^o\sin\phi \\ \cos\psi\cos\phi\sin 90^o + \sin\psi\sin\phi & -\cos\psi\sin\phi + \cos\phi\sin\psi\sin 90^o & \cos 90^o\cos\phi \end{bmatrix}$$

$$C_N^B = \begin{bmatrix} 0 & 0 & -1 \\ -\cos\phi\sin\psi + \cos\psi\sin\phi & \cos\psi\cos\phi + \sin\psi\sin\phi & 0 \\ \cos\psi\cos\phi + \sin\psi\sin\phi & -\cos\psi\sin\phi + \cos\phi\sin\psi & 0 \end{bmatrix}$$

$$C_N^B = \begin{bmatrix} 0 & 0 & -1 \\ \sin(\phi-\psi) & \cos(\phi-\psi) & 0 \\ \cos(\phi-\psi) & -\sin(\phi-\psi) & 0 \end{bmatrix} \quad (2.27)$$

2.5. Quaternions

Quaternions are complex algebra that represent the orientation of a point in three-dimensional space. The quaternion is composed of scalar part, q_0 and vector part q. The vector part in three- dimensional space, S^3; consists *i, j* and *k* as the standard orthonormal basis. The orthonormal basis is given in (2.28). Since quaternion consist of four scalars, it defines element in S^4. The quaternion can be represented as in (2.29) and (2.30).

$$i = (1,0,0) \qquad \text{j=(0,1,0)} \qquad \text{k=(0,0,1)} \tag{2.28}$$

$$q = q_0 + q = q_0 + q_1 i + q_2 j + q_3 k \tag{2.29}$$

$$q = (q_0, q_1, q_2, q_3) \tag{2.30}$$

2.5.1. Quaternion Algebra

In 1843, W.R. Hamilton came up with an idea of quaternion's multiplication while he was walking with his wife. He wrote the rules for multiplication on the stone wall of the bridge in Dublin. The quaternion multiplication rule that W. R, Hamilton wrote is given in (2.31) [22]. Also, the multiplication of quaternions is shown in Table 2.1.

$$i^2 = j^2 = k^2 = -1 \tag{2.31}$$

Table 2.1. Quaternion's Multiplication

	i	*j*	*k*
i	-1	*k*	-*j*
j	-*k*	-1	*i*
k	*j*	-*i*	-1

2.5.2. Quaternion Properties

- Complex Conjugates: The complex conjugates of quaternion is denoted by q^*. To determine complex conjugates, the sign of the vector part is changed as shown in (2.32). The properties of complex conjugates are shown in (2.33)

$$q^* = q_0 - \boldsymbol{q} = q_0 - q_1\boldsymbol{i} - q_2\boldsymbol{j} - q_3\boldsymbol{k} \tag{2.32}$$

$$(pq)^* = q^*p^* \quad \text{and} \quad (p^*q)^* = q^*p \tag{2.33}$$

- Quaternion Norm: The norm of the quaternion is denoted by *N*(q) and is given by

$$N(q) = |q| = \sqrt{q^*q} = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} \tag{2.34}$$

- Unit Quaternion: The unit quaternion has norm equal to one and it is given by (2.35). The product of unit quaternion is also a unit quaternion and it is given by (2.36)

$$|q| = |q^*| = 1 \tag{2.35}$$

$$q^*q = 1 \tag{2.36}$$

2.5.3. Rotation Matrix and Quaternion conversion

Every unit quaternion can be expressed as rotation operator. The rotation operator consists of two quaternions, the normal and its conjugate as shown in (2.37) . From (2.37), it is observed that the vector *v* is rotated from the reference frame to body frame using unit quaternion (q) and it's conjugate.

$$w = q \otimes \begin{bmatrix} 0 \\ v \end{bmatrix} \otimes q^* \tag{2.37}$$

Where, $q \in \mathbb{R}^{4\times1}\, is\, [q_0\ q_1\ q_2\ q_3]^T$, $q^* \in \mathbb{R}^{4\times1}\, is\, [q_0 - q_1\ -q_2\ -q_3]^T$ and $\otimes$ is Kronecker product operator. The equation (2.37) can be re-written by replacing vector *v* with orthonormal basis as mentioned in (2.28). This will give a rotation vector for the

respective axes and they are represented in terms of unit quaternion as shown in (2.38) to (2.40) [9]. The rotation vector $R_X(q)$, $R_Y(q)$ and $R_Z(q)$ given in equation (2.38) to (2.40) is combined as shown in (2.41) to get a reference frame to body frame rotation matrix (C_N^B).

$$R_X(q) = q \otimes \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \otimes q^* = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 \\ 2(q_1 q_2 + q_0 q_3) \\ 2(q_1 q_3 - q_0 q_2) \end{bmatrix} \tag{2.38}$$

$$R_Y(q) = q \otimes \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \otimes q^* = \begin{bmatrix} 2(q_1 q_2 - q_0 q_3) \\ q_0^2 - q_1^2 + q_2^2 - q_3^2 \\ 2(q_2 q_3 + q_0 q_1) \end{bmatrix} \tag{2.39}$$

$$R_Z(q) = q \otimes \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \otimes q^* = \begin{bmatrix} 2(q_0 q_2 + q_1 q_3) \\ 2(q_2 q_3 - q_0 q_1) \\ q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \tag{2.40}$$

$$R(q) = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_0 q_2 + q_1 q_3) \\ 2(q_1 q_2 + q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 + q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \tag{2.41}$$

2.5.4. Euler Angles and Quaternion conversion

The unit quaternions can be used to derive Euler angles set. Since, quaternion do not give physical intuitive meaning, quaternions are converted to yaw, pitch and roll as shown in equation (2.42), (2.43) and (2.44) respectively [6]. The quaternions are used as the state vector and the priori estimation is done based on quaternions. However, they are again converted back to Euler angles set to give physical sense.

$$yaw = \tan^{-1}\left(\frac{2(q_1 q_2 + q_0 q_3)}{q_0^2 + q_1^2 - q_2^2 - q_3^2}\right) \tag{2.42}$$

$$pitch = \sin^{-1}(2(q_1 q_3 - q_0 q_2)) \tag{2.43}$$

$$roll = \tan^{-1}\left(\frac{2(q_2 q_3 + q_0 q_1)}{q_0^2 - q_1^2 - q_2^2 + q_3^2}\right) \tag{2.24}$$

2.5.5. Advantages of quaternions.

The advantages of quaternion are listed below [23], [24]:

i. The quaternion representation does not create any worries about a certain sign or order convention of the rotation about the axes [23].

ii. The four partial derivatives $\frac{\partial R}{\partial q_0}, \frac{\partial R}{\partial q_1}, \frac{\partial R}{\partial q_2}$ and $\frac{\partial R}{\partial q_3}$ exist and they are linearly independent, which means quaternion are free from gimbal lock.

iii. Integration of ODEs that has quaternion as a state will move the quaternion off the unit quaternion which results to non-rotations. The integration will push the quaternion out of S^3. This problem can be addressed by re-normalizing the quaternion after every integration step. Re-normalizing the quaternion forces quaternion to maintain its unit length.

iv. The quaternion algebra is simple and computationally faster.

v. The quaternion can achieve a smooth interpolation using Beizer, Hermite or B-spline blending functions [24].

CHAPTER III

CALIBRATION

3.1. Introduction

Low cost inertial measurement units are affected by errors caused due to non-accurate scaling, sensor misalignment, cross-axis sensitivity and non-zero biases [25]. Therefore, calibration of these sensor is required to ensure accurate results. The calibration of each sensor should be done with respect to known standard values. The tri-axis accelerometer is calibrated using static approach in which accelerometer is mounted in six different positions. For calibration of accelerometer, the gravity vector is the known standard value. The tri-axis magnetometer is calibrated by rotating the device about all three principal axes and collecting the magnetic field readings in each axis. For calibration of magnetometer, the Earth magnetic field vector is known as the standard value. The tri-axis gyroscope is calibrated by using the rotating disk of a known angular velocity.

3.2. Calibration of tri-axis accelerometer

The readings in tri-axis accelerometer can be calibrated using (3.1). In (3.1), $A_i \in \mathbb{R}^{3\times1}$ is given as $\left[A_{x_{Pi}}\ A_{y_{Pi}}\ A_{z_{Pi}}\right]^T$. $A_{x_{Pi}}, A_{y_{Pi}}$and $A_{z_{Pi}}$ are compensated measurement of accelerometer in x, y and z direction respectively for a given positionPi, $K \in \mathbb{R}^{3\times3}$ is scale factor and it is given in (3.2). $T_a^p \in \mathbb{R}^{3\times3}$ is a transformation matrix that accounts for misalignment correction and it is shown in (3.2), $a_i^p \in \mathbb{R}^{4\times1}$ is given as $\left[a_{x_{Pi}}\ a_{y_{Pi}}\ a_{z_{Pi}}\right]^T$. $a_{x_{Pi}}, a_{y_{Pi}}$and $a_{z_{Pi}}$ are uncompensated measurements of accelerometer in x, y and z

directions respectively for a given position Pi, b $\in \mathbb{R}^{3\times1}$ is bias vector and it is given as $[b_x\ b_y\ b_z]^T$.

$$A_i = K\left[T_a^p\right]^{-1} a_i^p + b \tag{3.1}$$

$$\begin{bmatrix} A_{x_{Pi}} \\ A_{y_{Pi}} \\ A_{z_{Pi}} \end{bmatrix} = \begin{bmatrix} k_x & 0 & 0 \\ 0 & k_y & 0 \\ 0 & 0 & k_z \end{bmatrix} \begin{bmatrix} 1 & -\alpha_{yz} & -\alpha_{xy} \\ 0 & 1 & -\alpha_{zx} \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} a_{x_{Pi}} \\ a_{y_{Pi}} \\ a_{z_{Pi}} \end{bmatrix} + \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} \tag{3.2}$$

In (3.2), the product of K and $\left[T_a^p\right]^{-1}$ gives sacling and compensation matrix. This matrix is responsible for compensating both the misalignment and scale factor errors. The equation (3.2) can be re-written as

$$\begin{bmatrix} A_{x_{Pi}} \\ A_{y_{Pi}} \\ A_{z_{Pi}} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \begin{bmatrix} a_{x_{Pi}} \\ a_{y_{Pi}} \\ a_{z_{Pi}} \end{bmatrix} + \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} \tag{3.3}$$

Taking transpose on both side of (3.3)

$$\begin{bmatrix} A_{x_{Pi}} \\ A_{y_{Pi}} \\ A_{z_{Pi}} \end{bmatrix}^T = \left(\begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \begin{bmatrix} a_{x_{Pi}} \\ a_{y_{Pi}} \\ a_{z_{Pi}} \end{bmatrix} \right)^T + \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix}^T$$

$$\left[A_{x_{Pi}}\ A_{y_{Pi}}\ A_{z_{Pi}}\right] = \begin{bmatrix} a_{x_{Pi}} \\ a_{y_{Pi}} \\ a_{z_{Pi}} \end{bmatrix}^T \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}^T + [b_x \quad b_y \quad b_z]$$

$$\left[A_{x_{Pi}}\ A_{y_{Pi}}\ A_{z_{Pi}}\right] = \left[a_{x_{Pi}}\ a_{y_{Pi}}\ a_{z_{Pi}}\right] \begin{bmatrix} c_{11} & c_{21} & c_{31} \\ c_{12} & c_{22} & c_{32} \\ c_{13} & c_{23} & c_{33} \end{bmatrix} + [b_x \quad b_y \quad b_z]$$

$$\left[A_{x_{Pi}}\ A_{y_{Pi}}\ A_{z_{Pi}}\right] = \left[a_{x_{Pi}}\ a_{y_{Pi}}\ a_{z_{Pi}}\ 1\right] \begin{bmatrix} c_{11} & c_{21} & c_{31} \\ c_{12} & c_{22} & c_{32} \\ c_{13} & c_{23} & c_{33} \\ b_x & b_y & b_z \end{bmatrix}$$

$$A_{pi} = a_{pi}\, C \tag{3.4}$$

In C given in (3.4), there are 12 unknown parameters which are to be identified. To determine this parameters, six different sets of experiment is performed by mounting

accelerometer in six positions. For each position, $A_{pi} \in \mathbb{R}^{1\times3}$ is known and $a_{pi} \in \mathbb{R}^{1\times4}$ is readings coming from each accelerometer. The six different position and their known standard value is given in Table 3.1.

Table 3.1. Sign and Magnitude definition of accelerations for different positions in Akron, OH (Earth Gravitational Model EGM2008)

Position	Description	Known Accelerometer Reading (A)		
		$A_{x_{Pi}}$	$A_{y_{Pi}}$	$A_{z_{Pi}}$
$P1$	Z-axis Downward	0	0	-9.81
$P2$	Z-axis Upward	0	0	+9.81
$P3$	Y-axis Downward	0	-9.81	0
$P4$	Y-axis Upward	0	+9.81	0
$P5$	X-axis Downward	-9.81	0	0
$P6$	Y-axis Upward	+9.81	0	0

In this approach, for position $P1$ the accelerometer is mounted on a flat surface such that the z-axis faces vertically downward. Assume n_1 is the number of data samples collected from accelerometer for position $P1$ for approximately 5 to 10 seconds. For this position, A_{p1} and a_{p1} are given by (3.5) and (3.6) respectively.

$$A_{p1} = [\,0 \qquad 0 \qquad -9.81\,]_{n_1\times3} \tag{3.5}$$

$$a_{p1} = \left[a_{x_{P1}}\; a_{y_{P1}} a_{z_{P1}}\; 1\,\right]_{n_1\times3} \tag{3.6}$$

Similarly, the equation (6) and (7) can be rewritten for remaining five position $P2$ to $P6$ and the number of data samples collected is n_2 to n_6 respectively. For all six position,

the equation (3.5) and (3.6) can be combined and can be written as (3.7). In (3.7), each matrix has a suffix which represents the size of matrix. The size of each matrix is provided for easy and convenient understanding. In the equation (3.6), the first column represents reading from *x-axis*, second column represents reading from *y-axis* and third column represents reading from *z-axis*.

$$\begin{bmatrix} [\,0 & 0 & -9.81\,]_{n_1\times 3} \\ [\,0 & 0 & 9.81\,]_{n_2\times 3} \\ \vdots & \vdots & \vdots \\ [\,9.8 & 0 & 0\,]_{n_6\times 3} \end{bmatrix}_{n\times 3} = \begin{bmatrix} [\,a_{x_{P1}} & a_{y_{P1}} & a_{z_{P1}} & 1\,]_{n_1\times 4} \\ [\,a_{x_{P2}} & a_{y_{P2}} & a_{z_{P2}} & 1\,]_{n_2\times 4} \\ \vdots & \vdots & \vdots & \\ [\,a_{x_{P6}} & a_{y_{P6}} & a_{z_{P6}} & 1\,]_{n_6\times 4} \end{bmatrix}_{n\times 4} [C]_{4\times 3} \quad (3.7)$$

$$[A]_{n\times 3} = [a]_{n\times 4}[C]_{4\times 3} \quad (3.8)$$

Using least square method in (3.8) to solve for *C*, each element of C is determined by (3.9) [26].

$$C = [a^T \cdot a]^{-1} a^T A \quad (3.9)$$

Where, *a* is uncompensated readings from the tri-axis accelerometer, *A* is known or standard value and *C* is compensation matrix. The uncompensated or non-calibrated readings are multiplied by the compensation matrix *C* to get compensated or calibrated readings. The first three row of above matrix *C* take care for the misalignment and the last row takes care for the bias.

The Figure (3.1) shows the accelerometer readings of each sensor when z-axis is vertically downward and upward i.e. position *P1* and *P2* respectively. In position *P1*, the accelerometer reading should be *-9.81* $\frac{m}{s^2}$ in *z-axis*, zero in *y-axis* and zero in *x-axis*. In Figure (3.1), for positon *P1*, the accelerometer reading in *z-axis* is approximately *-9.3* $\frac{m}{s^2}$ and approximately zero in *x-axis* and *y-axis*. In position *P2*, the accelerometer reading

should be *9.81* $\frac{m}{s^2}$ in *z-axis*, zero in *y-axis* and zero in *x-axis*. In Figure (3.1), for position *P2*, the accelerometer reading in *z-axis* is approximately *10.3* $\frac{m}{s^2}$ and approximately zero in *x-axis* and *y-axis.*

The Figure (3.2) shows the accelerometer readings of each sensor when y-axis is vertically downward and upward i.e. position *P3* and *P4* respectively. In position *P3*, the accelerometer reading should be zero in *z-axis*, *-9.81* $\frac{m}{s^2}$ in *y-axis* and zero in *x-axis*. In Figure (3.1), for positon *P3*, the accelerometer reading in *y-axis* is approximately *-9.8* $\frac{m}{s^2}$, approximately *0.6* $\frac{m}{s^2}$ in *z-axis* and zero in *x-axis.* In position *P4*, the accelerometer reading should be *9.81* $\frac{m}{s^2}$ in *y-axis*, approximately zero in *z-axis* and zero in *x-axis*. In Figure (3.2), for position *P4*, the accelerometer reading in *y-axis* is approximately *9.8* $\frac{m}{s^2}$ and approximately *0.6* $\frac{m}{s^2}$ in *z-axis* and zero in *x-axis.*

The Figure (3.3) shows the accelerometer readings of each sensor when x-axis is vertically downward and upward i.e. position *P5* and *P6* respectively. In position *P5*, the accelerometer reading should be zero in *z-axis*, zero in *y-axis* and *-9.81* $\frac{m}{s^2}$ in *x-axis*. In Figure (3.3), for positon *P5*, the accelerometer reading in *z-axis* is approximately *0.6* $\frac{m}{s^2}$, approximately *zero* in *y-axis* and *-9.82* $\frac{m}{s^2}$ in *x-axis.* In position *P6*, the accelerometer reading should be *9.81* $\frac{m}{s^2}$ in *x-axis*, approximately zero in *y-axis* and zero in *z-axis.* In Figure (3.3), for position *P6*, the accelerometer reading in *x-axis* is approximately *9.8* $\frac{m}{s^2}$ and approximately *0.6* $\frac{m}{s^2}$ in *z-axis* and zero in *y-axis.*

After observing the accelerometer readings for six different position and comparing those results with known standard value as discussed above, it is clear that the readings

from accelerometer is not correct. It can be concluded that the accelerometer has some bias in its reading. Therefore, calibration of accelerometer is required. The compensation matrix needed for calibration of accelerometer is calculated and it is given in (3.10).

$$C = \begin{bmatrix} 1.0009 & -0.0029 & 0.0007 \\ 0.0082 & 1.0006 & -0.0001 \\ -0.0008 & 0.0012 & 1.0010 \\ 0.0179 & 0.0291 & -0.5266 \end{bmatrix} \tag{3.10}$$

The calibrated or compensated readings of accelerometer in six different position is shown in Figure (3.4) to (3.6). The axis that points vertically downward should read -*9.8* $\frac{m}{s^2}$ and the other two axis should read zero. Similarly, the axis that points vertically upward should read *9.8* $\frac{m}{s^2}$ and the other two axis should read zero. In Figure (3.4), for position *P1*, the downward pointing *z-axis* reads -*9.81* $\frac{m}{s^2}$ and for position *P2*, the upward pointing *z-axis* reads *9.81* $\frac{m}{s^2}$ approximately while other two axis reading being zero. In Figure (3.5), for position *P2*, the downward pointing *y-axis* reads -*9.81* $\frac{m}{s^2}$ and for position *P4*, the upward pointing *y-axis* reads *9.81* $\frac{m}{s^2}$ approximately while other two axis reading being zero. In Figure (3.6), for position *P5*, the downward pointing *x-axis* reads -*9.81* $\frac{m}{s^2}$ and for position *P6*, the upward pointing *x-axis* reads *9.81* $\frac{m}{s^2}$ approximately while other two axis reading being zero. In this work, each element of compensation matrix *C* given in (3.10) is provided in VECTORNAV software's compensation tool/tab to get the compensated or calibrated readings. Therefore, the compensated readings obtained from the device is directly fed as a measurement in the extended Kalman algorithm.

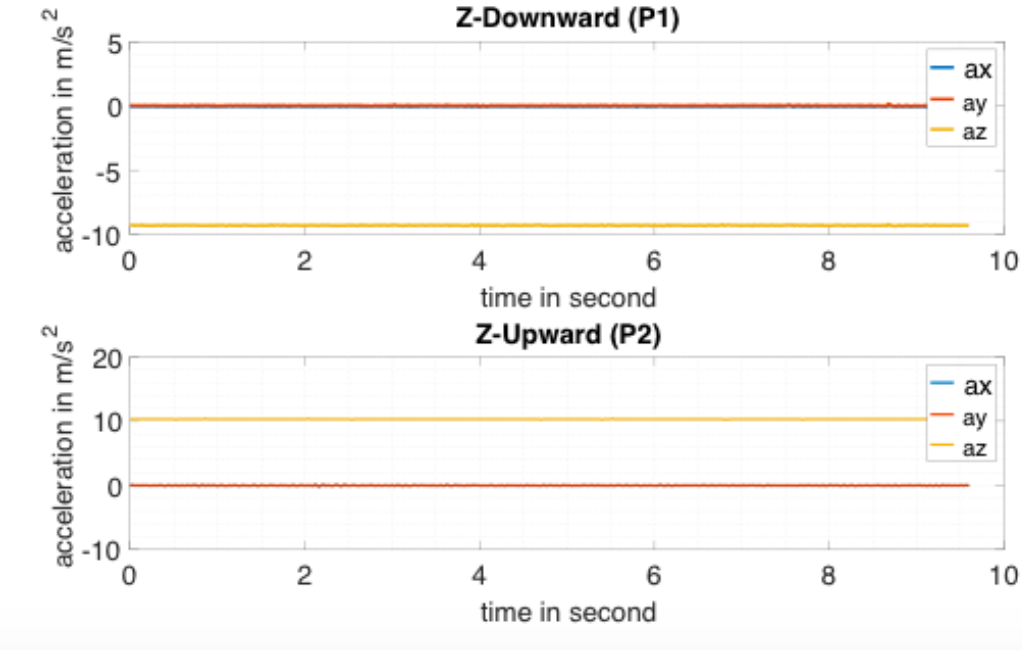

Figure 3.1. Uncompensated accelerometer reading for position P1 and P2.

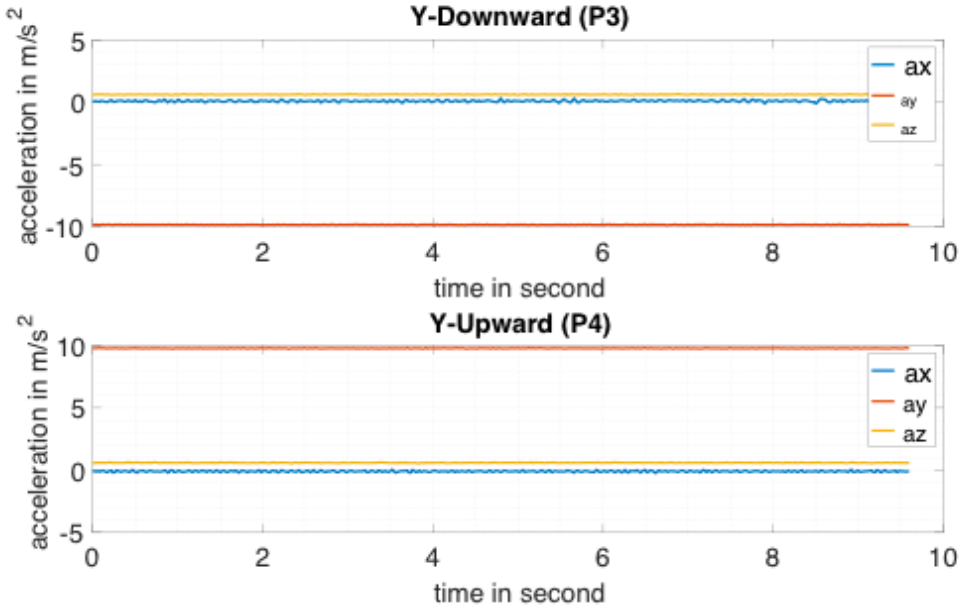

Figure 3.2. Uncompensated accelerometer reading for position P3 and P4.

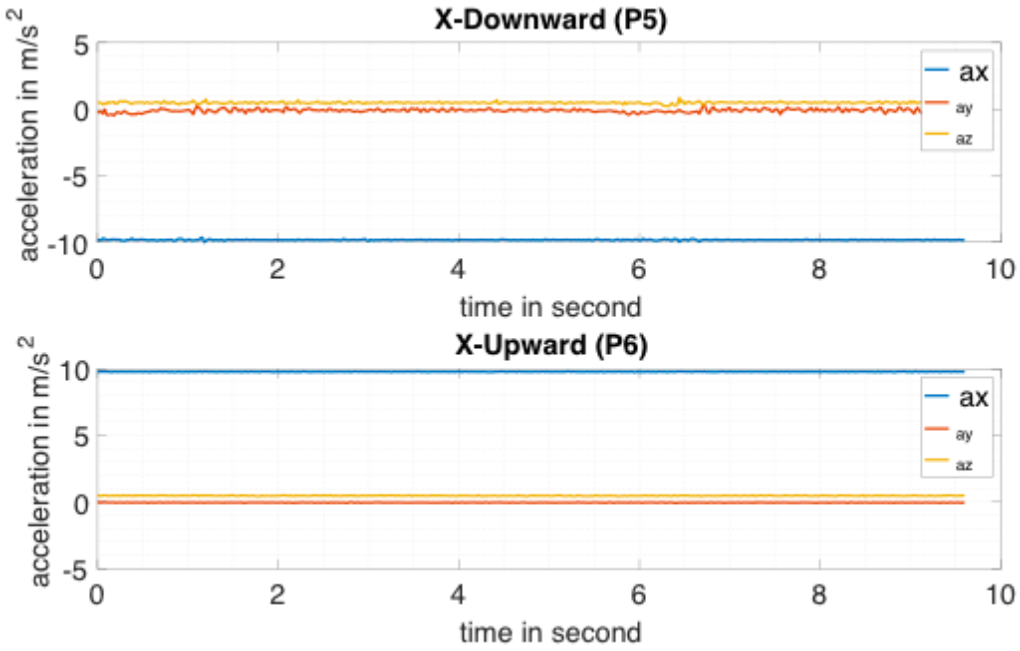

Figure 3.3. Uncompensated accelerometer reading for position P5 and P6.

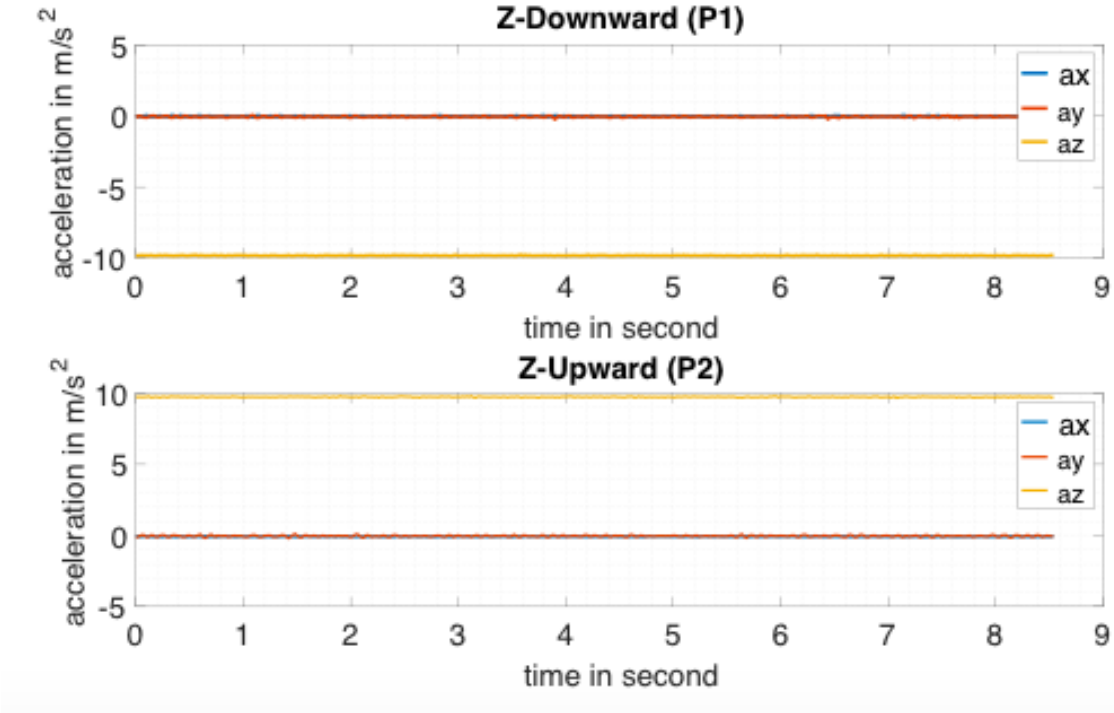

Figure 3.4. Compensated accelerometer reading for position P1 and P2.

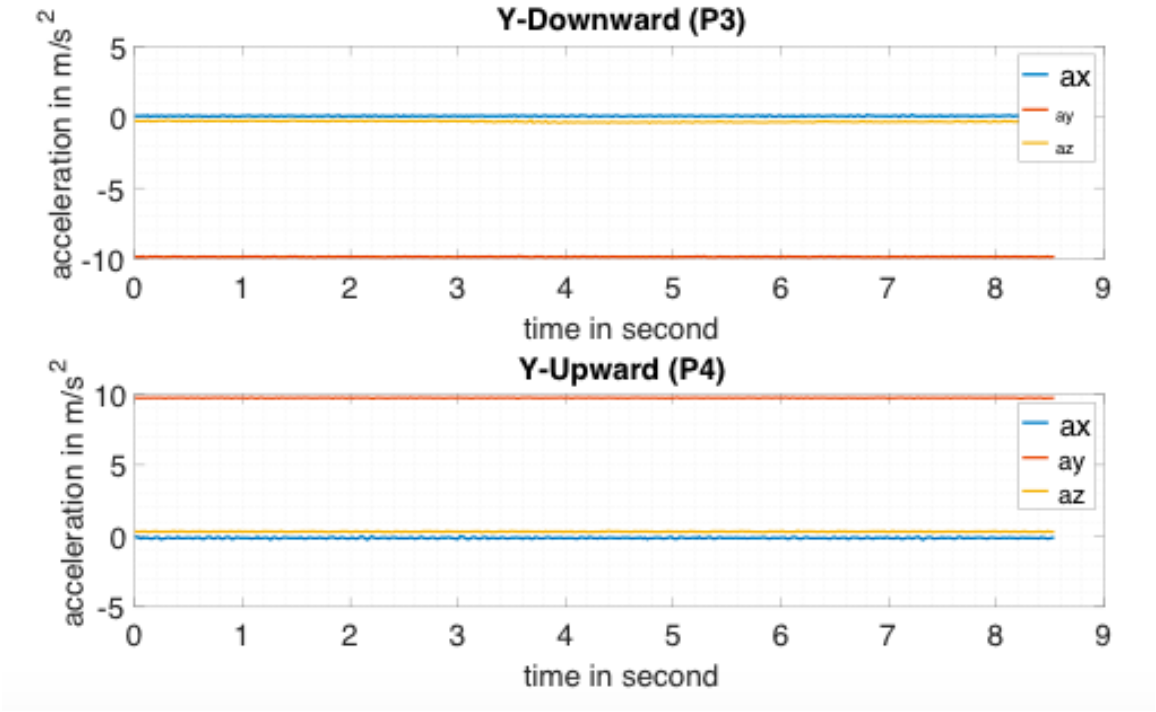

Figure 3.5. Compensated accelerometer reading for position P3 and P4.

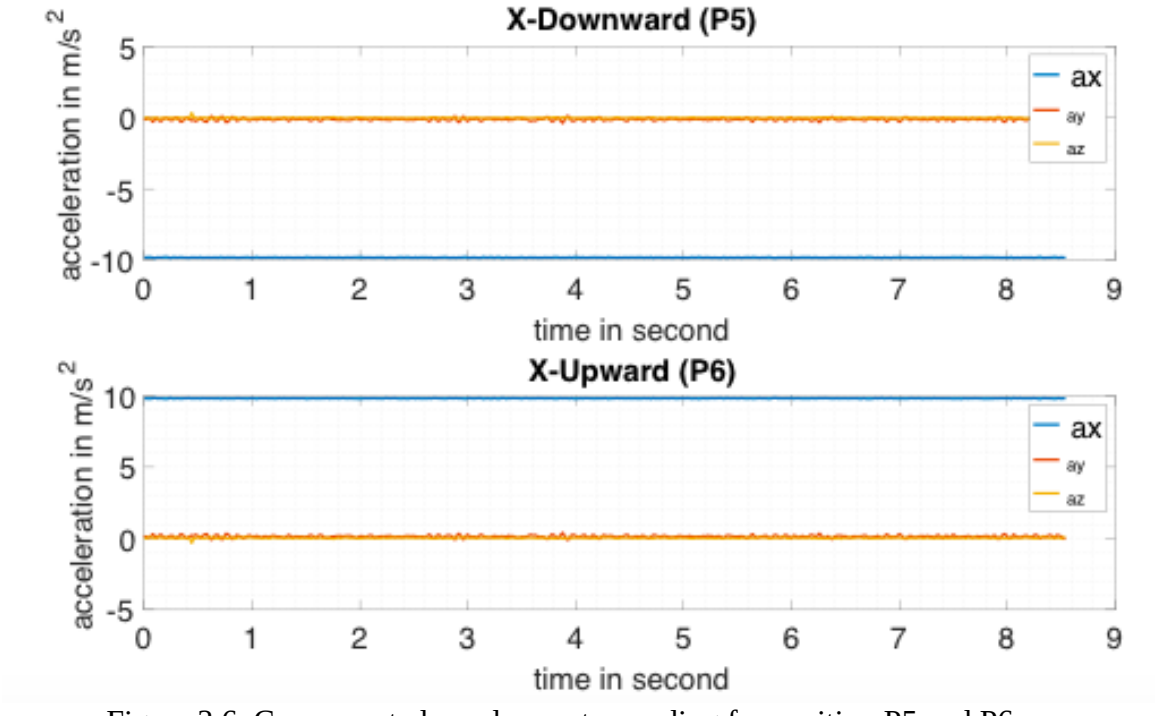

Figure 3.6. Compensated accelerometer reading for position P5 and P6.

3.3. Calibration of tri-axis magnetometer

The magnetometers are important unit for navigation systems. The tri-axis magnetometer measures magnetic field in three principal axes. The output of the magnetometers are corrupted due to the sensor imperfections, misalignment and interfering magnetic field. Sensor imperfections is related to micromachining and design. The misalignment is related to non-orthogonality. The errors due to both sensor imperfections and misalignment can be minimized by careful manufacturing and factory calibration. The interfering magnetic field can be broadly classified into two distinct groups. First group consists of constant or time varying magnetic field generated by the materials in the proximity of magnetometer. This is known as hard iron effect. The second group consists of materials that influence or distorts magnetic field. This is known as soft iron effect. The soft iron distortion is caused by ferromagnetic materials such as nickel and iron. The hard and soft iron effect in magnetic measurement is demonstrated by the equation (3.11) [27].

$$m = S_d(M_{Earth} + MI_{AC} + MI_{DC}) + H_l + MB_{AC} + MB_{DC} \qquad (3.11)$$

Where, m is readings from the magnetometer, S_d is distortion due to soft iron effect and M_{Earth} is the Earth's magnetic field. MI_{AC} and MI_{DC} are cyclic and constant magnetic field external to sensor frame and their respective biases are MB_{AC} and MB_{DC}. H_l is magnetic biases due to hard iron effects. By rearranging (3.11), the compensation of hard and soft iron components can be found using (3.12). It is assumed that no external fields are present except the earth's magnetic field i.e. magnetometer is placed in the magnetically clean environment. The term MI_{AC} and MB_{AC} can be rejected by low pass filtering. The

term MB_{DC} can be lumped with the H_l term. The term MI_{DC} can be eliminated by avoiding the proximity to the objects that cause deviation in Earth's magnetic field. For example, large structure, bars and rail roads.

$$m_{compensated} = {S_d}^{-1}(m - H_l) \tag{3.12}$$

The equation (3.12) can be solved online using an on board algorithm provided by VECTORNAV software. The term ${S_d}^{-1}$ and H_l can be downloaded in the software to get the compensated magnetic readings.

In this work, the tri-axis magnetometer is calibrated only in horizontal plane, i.e. *x-y* plane. The reason to do so is that there was high external magnetic field generated in z-direction The magnetic field in *z-axis* when it faces downward direction was measured in different location of lab . It was found that the reading was highly non-uniform. The magnetic reading was varying from 0.26 gauss to 1.5 gauss in different location. If there is not any external magnetic field, the z-axis sensor should read 0.489 gauss as given in Table 1.3.

In this calibration technique, the magnetometer is rotated in horizontal plane and magnetic readings is collected from each sensor. The magnetic reading in *x-axis* and *y-axis* is plotted in Figure (3.7). In Figure (3.7), the magnetic readings in *x-axis* and *y-axis* form a close circle. Ideally the radius of the circle shown in Figure (3.7) should be the horizontal magnetic intensity. The value of horizontal intensity in Akron, OH is 0.1979 gauss (From Table 1.3). The approximate radius of dotted blue circle in Figure (3.7) is 0.20 gauss. Since the plot of *x-axis* and *y-axis* magnetic readings forms an approximate circle i.e. red circle shown in Figure (3.7), it can be concluded that there is negligible hard and soft iron effect. In this case, calibration of magnetometer is not required.

If the plot is an ellipse, the major axis and minor axis are drawn as shown in Figure (3.8). The major axis is the longest line segment run through the center of an ellipse and minor axis is shortest line segment run through the center. The data points obtained from the magnetic reading are used to get best ellipse fit as shown in Figure (3.8). The point A is intersection of the major and minor axis. The point B is center of the best fitted ellipse. The point A may shift from point B due to hard iron effect. The offset in *x-axis* and *y-axis* is used to determine the hard iron biases[28].

To determine the distortion due to soft iron effect, the ellipse is converted to an approximate circle by using a scaling factor. Here, the scaling factor is ratio of length of major axis to that of minor axis. Before, hard and soft iron calibration, the tilting of an ellipse should be avoided. The tilting of ellipse occurs when the major axis makes some angle with *x-axis*. The tilting of an ellipse can be avoided using appropriate rotation matrix[28].

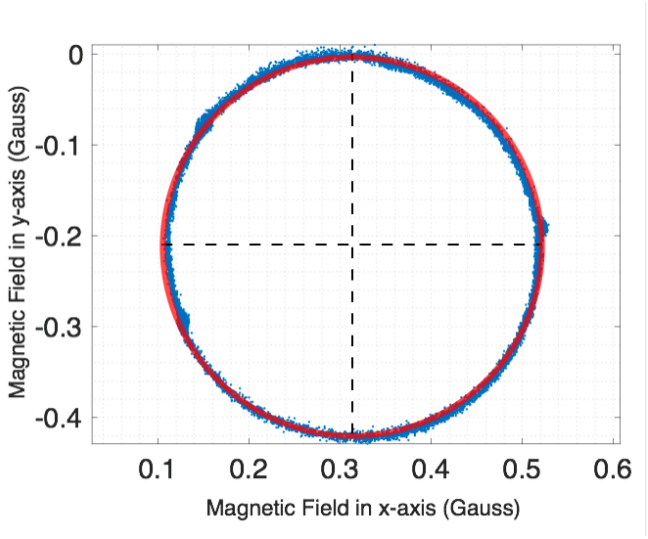

Figure 3.7. The magnetometer output under rotation in horizontal plane.

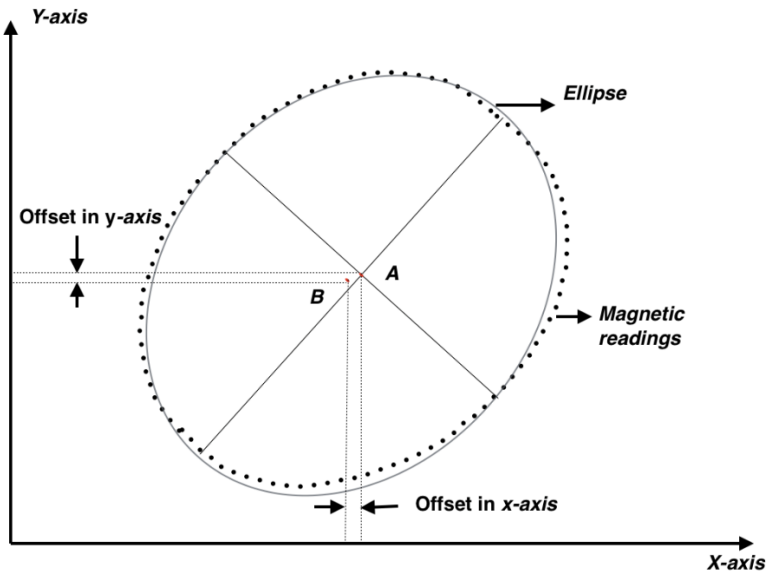

Figure 3.8. The shift in the center of an ellipse to show hard iron effect

3.4. Calibration of tri-axis gyroscope.

The biases in gyroscope have large impact in the performances of IMU systems. Since the readings coming from gyroscope are integrated to get an orientation, any bias errors present in the gyroscope will be accumulated and it will grow in time. The important fact about gyroscope bias is that it has a random walk component which cannot be calibrated out using standard calibration methods. Random walk is defined as process containing series of steps whose magnitude and direction is randomly determined. During random walk, the samples are identically distributed and has a finite variance and it can be minimized by statistical techniques. The random walk can be eliminated using real time online tool such as extended Kalman filter. The other bias due to misalignment and linear acceleration sensitivity can be eliminated by standard calibration technique. If the gyroscope is subjected to linear acceleration which is less than acceleration due to gravity, the gyroscope sensitivity can be ignored [29].

The misalignment in the gyroscope refers to non-orthogonality of three principal axes. The misalignment in the gyroscope can be eliminated by using device named “rate table”. In this device the gyroscope is first attached on the disk. Then, the disk is rotated with known angular speed about *x, y* and *z-axis* as well as both in positive and negative direction. Finally, the readings from the gyroscope and angular rate reading from rate table is obtained for six different scenarios. The least square method given in the equation (3.9), is applied to solve for misalignment and scale factor.

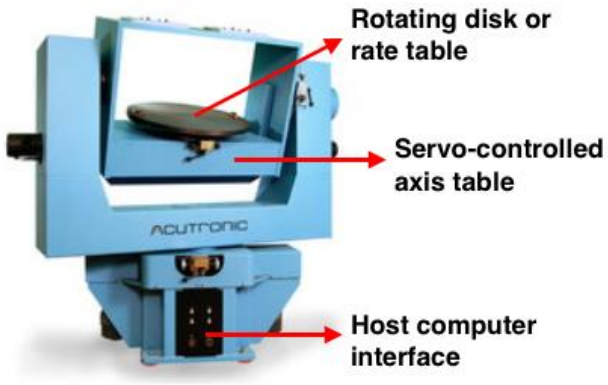

Figure 3.9. Three-axis rate table (Source:[30])

In this work, calibration technique discussed above is not applied to calibrate the tri-axis gyroscope. But, the gyroscope readings for different position is compared with the known angular rates. The known angular rates are generated by utilizing the torsional plant experimental setup. The controlled sinusoidal motion of *1 Hz* frequency and amplitude of 22.5^{0} is shown in Figure (3.10). The control algorithm and the procedure is explained in Appendix A1. Here, the IMU device is attached firmly on the disk and disk is rotated for an above mentioned sinusoidal input. The angular displacement shown in Figure (3.10) is measured by an encoder. The encoder is a feedback device in torsional plant that measures

angular displacement of the disk. The angular displacement is converted to angular velocity using Euler's first order approximation.

The test is being performed by positioning the IMU device in six different position i.e. rotating gyro about *x, y* and *z-axis* and about both positive and negative direction. When the *tri-axis* gyroscope is rotated about downward directed *z-axis*, the angular velocity readings from *z-axis gyro* should read the same as the encoder's angular velocity. In this case, *x* and *y-axis* gyro reading should ideally be zero. The Figure (3.11) represents the gyroscope reading and the encoder reading when the *z-axis* of gyroscope directs downward. Comparing these two reading in Figure (3.11), it can be said that there is not much significant difference in the both readings. It can be noticed that there is negligible difference at the peak of each cycle. It may be due to the power cable of IMU device being stretched at the peak points because of high inertia or may be due to error in Euler's approximation for converting angular displacement to angular velocities.

The Figures (3.12) to (3.16) show the gyroscope reading and encoder reading for remaining five different position. In all figures, the gyroscope reading and encoder reading are approximately the same. From this work, it can be concluded that, the misalignment or scale factor matrix can be neglected for calibration of the used VECTORNAV gyroscope. In simple, the reading from the MEMS gyroscope is being trusted and the same reading is directly fed into the extended Kalman filter algorithm.

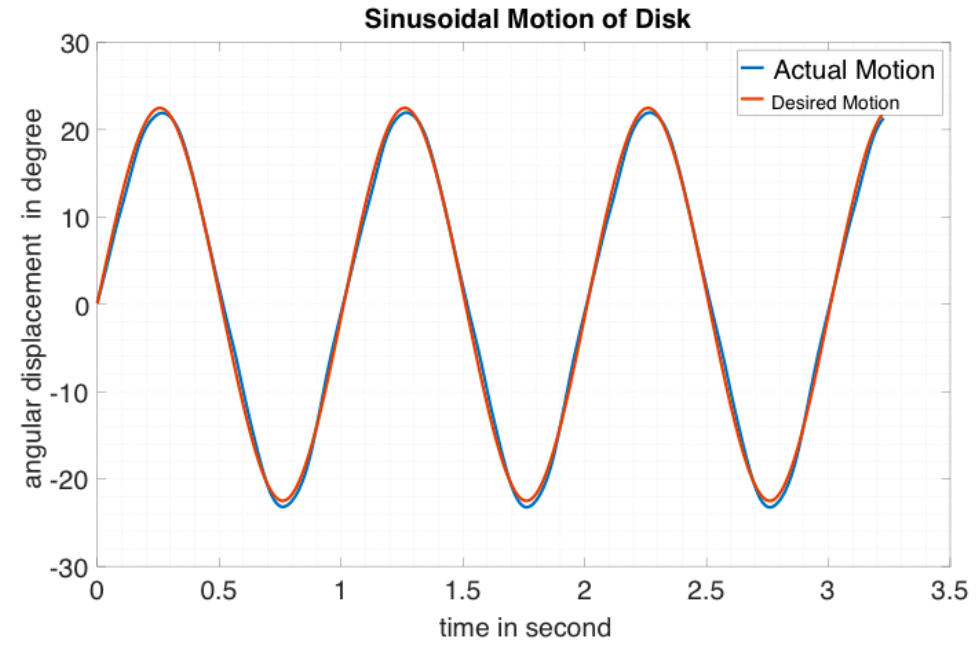

Figure 3.10. Sinusoidal disk rotation for 1 Hz frequency and 22.50 amplitude.

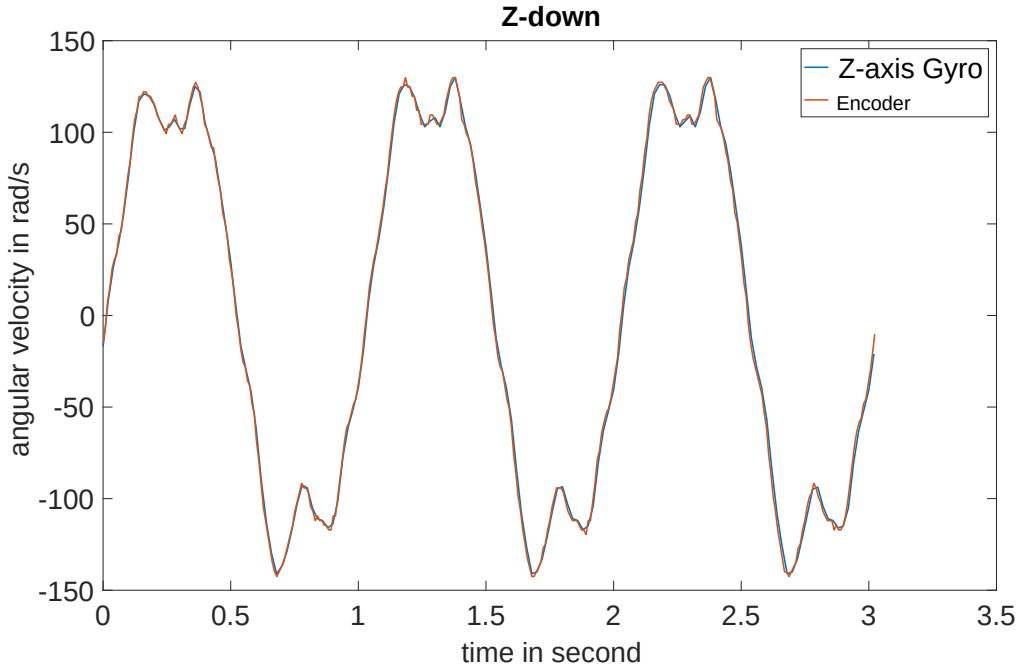

Figure 3.11. Gyroscope and Encoder angular rate reading when *z-axis* is downward.

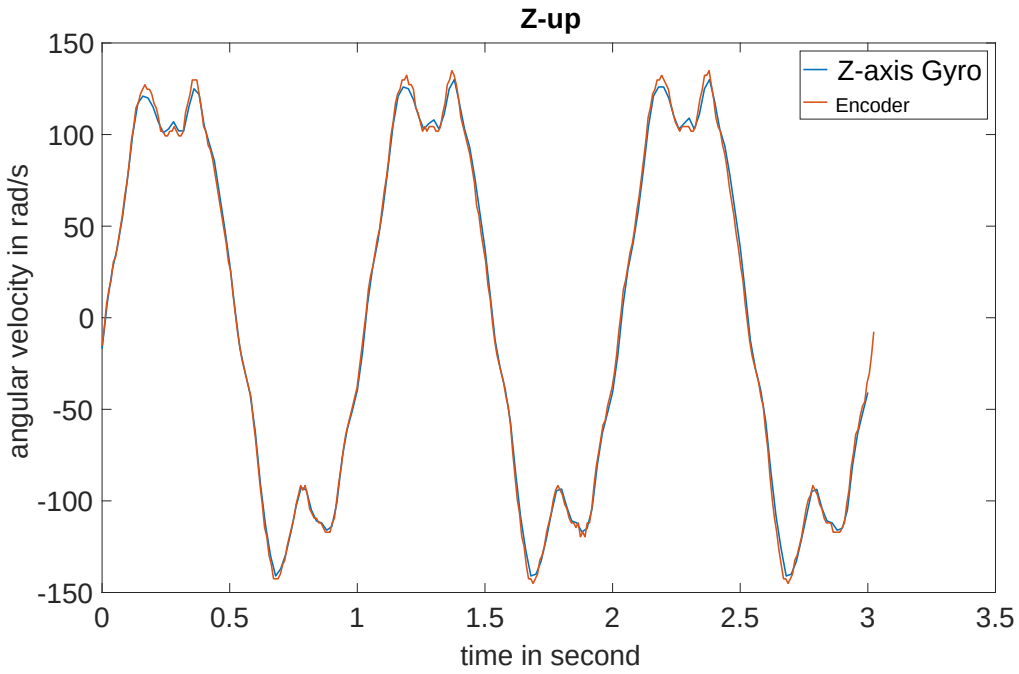

Figure 3.12. Gyroscope and Encoder angular rate reading when *z-axis* is upward.

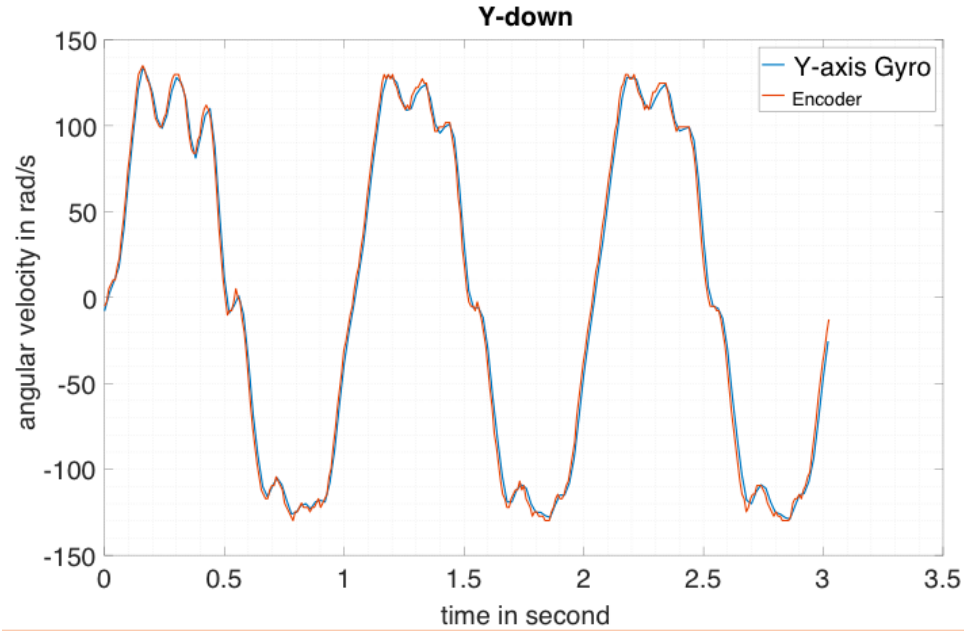

Figure 3.13. Gyroscope and Encoder angular rate reading when *y-axis* is downward.

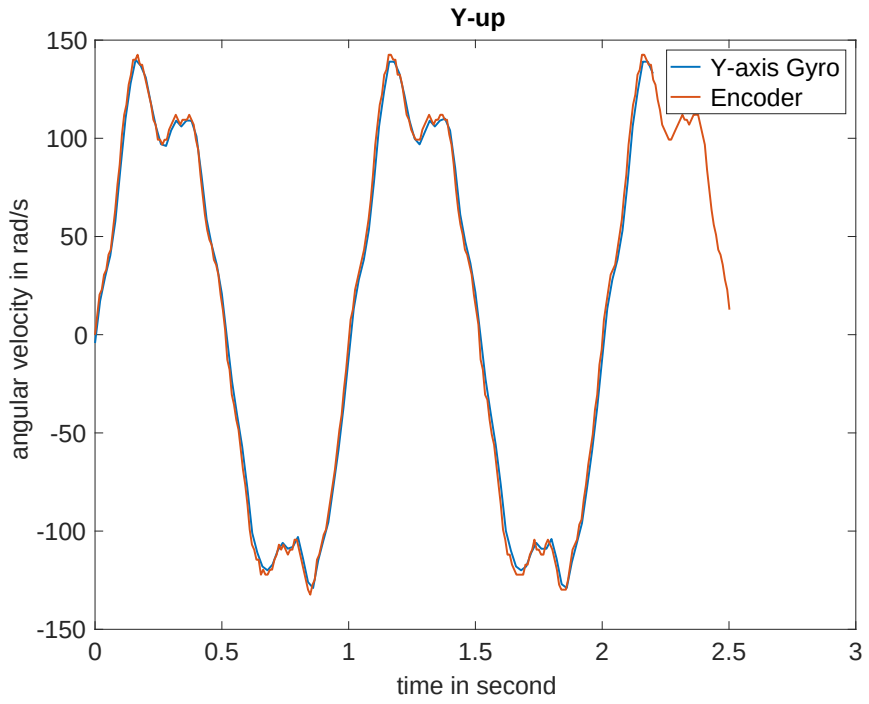

Figure 3.14. Gyroscope and Encoder angular rate reading when *y-axis* is downward.

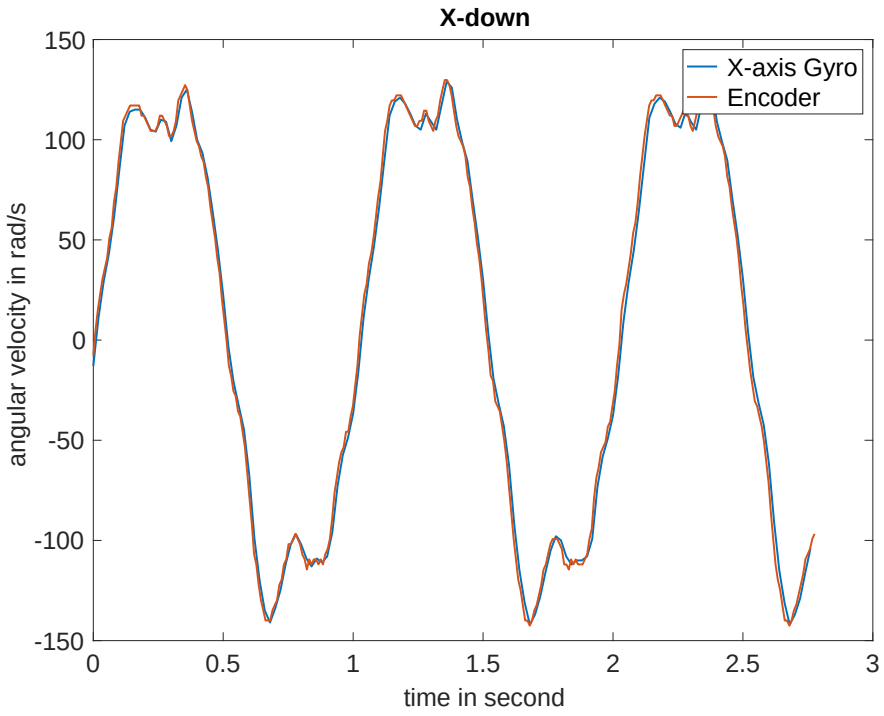

Figure 3.15. Gyroscope and Encoder angular rate reading when *y-axis* is downward.

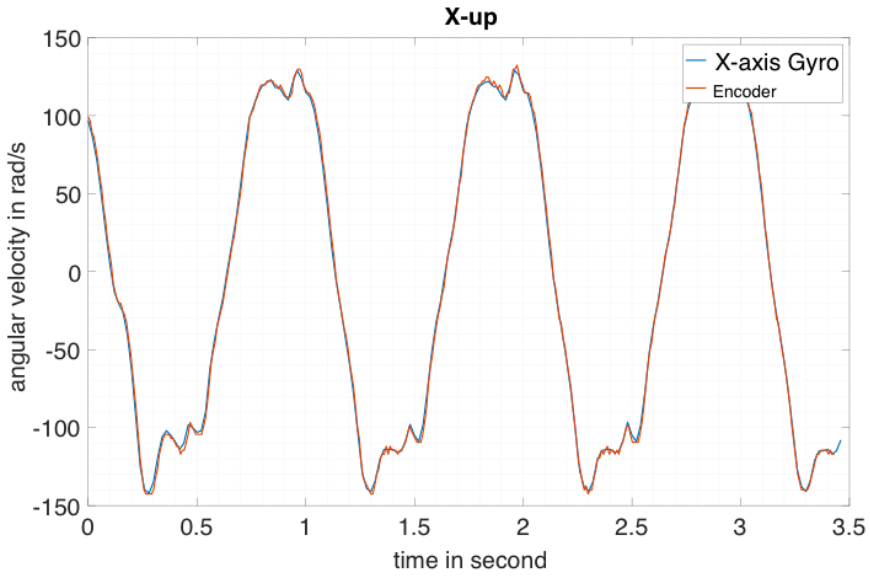

Figure 3.16. Gyroscope and Encoder angular rate reading when *y-axis* is downward.

3.5. Conclusion

In this chapter, the errors from different units such as the gyroscope, the accelerometer and the magnetometer are discussed. Most of the errors in those units are due to axis-misalignment and biases. To eliminate the errors in the reading of those units, quick and reliable calibration technique is applied. The calibration of the accelerometer and the magnetometer are based on position scheme i.e. collecting readings from different position such that reference or standard reading at each position is known. The least square approximation is used to identify the unknown parameters of the bias and scaling factor. After knowing those parameters, the compensated or calibrated measurements are generated and used in the estimation algorithm. The "rate table" gyroscopic calibration is suggested to eliminate the misalignment of readings from gyroscope. Here, the reliability on the gyroscope is checked by comparing the gyroscopic angular rates with known angular rates of disk.

CHAPTER IV

EXTENDED KALMAN FILTER (EKF)

4.1. Introduction

The Kalman filter is an iterative prediction/correction approach to estimate the state of a discrete-time process or measurement [31]. The Kalman Filter is used to make a reliable estimation of state variable in a real time to perform adequate control actions. The purpose of Kalman filter is to predict the state for next iteration given the current state and correct it if necessary. The correction is made based on the information read from the measurement sensor. Since every sensor have their own issues such as drift and noise, the Kalman filter combines those sensors to compensate their limitations. The Kalman filter judges the amount of confidence to be placed on the predictions and measurements [32]. The goal of Extended Kalman Filter is to determine the best estimate of the nonlinear system's state at a given point from the information collected at multiple observation points with respect to time. The following equations (4.1) and (4.2) represent state equation and measurement equation of the discrete-time approximation for non-linear state space model.

$$x_{k+1} = f_k x_k + w_k \tag{4.2}$$

$$z_k = h_k x_k + v_k \tag{4.3}$$

Where,

k is a discrete time sample,

$x \in \mathbb{R}^{n\times 1}$, is state variable,

$z \in \mathbb{R}^{m\times 1}$, is measurement,

$f \in \mathbb{R}^{n\times n}$, is non-linear state function,

$h \in \mathbb{R}^{m\times n}$, is non-linear measurement function,

$w \in \mathbb{R}^{n\times 1}$, is white process noise with covariance matrix $Q \in \mathbb{R}^{n\times n}$

$v \in \mathbb{R}^{m\times 1}$, is white measurement noise with covariance matrix $R \in \mathbb{R}^{m\times m}$

If *x* is a random vector of a Gaussian process and *p(x)* is probability density function, the covariance is defined as $E\{(x-\alpha)(x-\alpha)^T\}$, where α is mean of x and E{.} is expectation [33].

Prediction Step:

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, k) \tag{4.3}$$

$$P_{k|k-1} = F\left(k, \hat{x}_{k-1|k-1}\right)P_{k-1|k-1}F^T\left(k, \hat{x}_{k-1|k-1}\right) + Q_k \tag{4.4}$$

Kalman Filter Gain:

$$K_k = P_{k|k-1}H^T\left(k, \hat{x}_{k|k-1}\right)\left\{H\left(k, \hat{x}_{k|k-1}\right) P_{k|k-1}H^T\left(k, \hat{x}_{k|k-1}\right) + R_k\right\}^{-1} \tag{4.5}$$

Update Step:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\left(z_k - h(\hat{x}_{k|k-1}, k)\right) \tag{4.6}$$

$$P_{k|k} = P_{k|k-1} - K_k H\left(\hat{x}_{k|k-1}, k\right)P_{k|k-1} \tag{4.7}$$

Where,

$\hat{x}_{k-1/k-1}$ is the estimate of the state variable from previous time instance,

$P_{k-1|k-1}$ is covariance error matrix from previous time instance,

$\hat{x}_{k|k-1}$ is estimate of state form prediction step,

$P_{k|k-1}$ is covariance error matrix from prediction step,

$\hat{x}_{k|k}$ is state variable obtained from updated state and passed on to the next instance,

$P_{k|k}$ is covariance error matrix obtained from updated step and passed on to next instance,

The only available information before initializing the EKF is initial state (x_0) with known the initial covariance P_0. In prediction step, (4.3) calculates the states for time step (k|k-1) using states from previous time step (k-1|k-1). Also, (4.4) calculates the covariance error for the upcoming time step using the covariance error from previous time step.

In (4.3), *f(.)* is a state function and in (4.4), F is a Jacobian of state function. (4.5) calculates the Kalman filter gain for time step (k|k) using Jacobian of measurement function (h(.)) and measurement noise covariance matrix at time step (k|k-1). (4.6) updates state and (4.7) updates error covariance matrix for next iteration. The states obtained from (4.6) is the estimation of states for each time step and it is saved in memory.

4.2. The State Equation *(f(.))*

In system dynamics, any system can be expressed mathematically as a set of n-first order ordinary differential equations. This is known as state space representation. The set of variables required to refer a state of dynamic system is known as state variable or simply states. The knowledge of the state variables at the initial time is needed to predict the future states and outputs of a given system [32]. The equation (4.8) represents the state equation for quaternion based attitude system.

$$\dot{q}=\emptyset\, q \tag{4.8}$$

$$\text{or,} \qquad \frac{dq}{dt} = \emptyset\, q$$

$$\text{or approximately,} \qquad \frac{q_{k+1} - q_k}{T_s} = \emptyset\, q_k$$

$$q_{k+1} = q_k + \emptyset\, q_k\ T_s \tag{4.9}$$

$q \in \mathbb{R}^{4\times1}$ is a quaternion based state variable given as $[q_0\ q_1\ q_2\ q_3]^T$ and $\emptyset\ \in\ \mathbb{R}^{4\times4}$ is state function (f) given by (4.9). In prediction step of an extended Kalman filter, the state function is utilized to first predict the state when previous states are given.

$$\emptyset = \frac{1}{2}\begin{bmatrix} 0 & -w_x & -w_y & -w_z \\ w_x & 0 & -w_z & w_y \\ w_y & w_z & 0 & -w_x \\ w_z & -w_y & w_x & 0 \end{bmatrix} \tag{4.10}$$

w_x, w_y and w_z are the angular velocities in $\frac{rad}{s}$ measured by the gyroscope. Substituting (10) in (9).

$$q_{k+1} = q_k + \frac{1}{2}\begin{bmatrix} 0 & -w_x & -w_y & -w_z \\ w_x & 0 & -w_z & w_y \\ w_y & w_z & 0 & -w_x \\ w_z & -w_y & w_x & 0 \end{bmatrix} q_k\ T_s$$

$$q_{k+1} = q_k \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & -w_x\frac{T_s}{2} & -w_y\frac{T_s}{2} & -w_z\frac{T_s}{2} \\ w_x\frac{T_s}{2} & 0 & -w_z\frac{T_s}{2} & w_y\frac{T_s}{2} \\ w_y\frac{T_s}{2} & w_z\frac{T_s}{2} & 0 & -w_x\frac{T_s}{2} \\ w_z\frac{T_s}{2} & -w_y\frac{T_s}{2} & w_x\frac{T_s}{2} & 0 \end{bmatrix} q_k$$

$$q_{k+1} = \begin{bmatrix} 1 & -w_x\frac{T_s}{2} & -w_y\frac{T_s}{2} & -w_z\frac{T_s}{2} \\ w_x\frac{T_s}{2} & 1 & -w_z\frac{T_s}{2} & w_y\frac{T_s}{2} \\ w_y\frac{T_s}{2} & w_z\frac{T_s}{2} & 1 & -w_x\frac{T_s}{2} \\ w_z\frac{T_s}{2} & -w_y\frac{T_s}{2} & w_x\frac{T_s}{2} & 1 \end{bmatrix} q_k$$

$$q_{k+1} = \Omega q_k \tag{4.11}$$

Ω is known as state transition matrix. In discrete-time system, (4.11) shows that the product of state transition matrix (Ω) and states at previous time gives the states estimation for later time. In continuous-time system, the state transition matrix is the exponential of product of state function (f) and sampling time (T_s) and is given by (4.12). Integrating (4.8),

$$\int_{q_0}^{q} \dot{q}\, dt = \int_{t_0}^{t} \emptyset\, q\ dt$$

$$\int_{q_0}^{q} \frac{dq}{dt} = \int_{t_0}^{t} \emptyset\, q$$

$$\int_{q_0}^{q} \frac{dq}{q} = \int_{t_0}^{t} \emptyset\, dt$$

$$\ln q \Big\{_{q_0}^{q} = \emptyset\, t \Big\{_{t_0}^{t}$$

$$\ln q - \ln q_0 = \emptyset(t - t_0)$$

$$\ln\left(\frac{q}{q_0}\right) = e^{\emptyset(t-t_0)}$$

$$q = e^{\emptyset T_s}\, q_0 \tag{4.12}$$

To estimate the states in prediction step, a numerical integration approach is applied using Taylor series expansion method given by (4.13).

$$e^{\emptyset T_s} = \sum_{i=0}^{\infty} \frac{(\emptyset T_s)^i}{i!} = I + \emptyset T_s + \frac{\emptyset^2 {T_s}^2}{2!} + \frac{\emptyset^3 {T_s}^3}{3!} + \frac{\emptyset^4 {T_s}^4}{4!} + \cdots \quad (4.13)$$

4.3. Measurement Function (h(.))

In update step, the difference between the real measurement (z_k) obtained from time instant k+1 and measurement predicted from the predicted value of the state may give some error. This error is multiplied by a gain called the Kalman filter gain and added to predicted value of state to get new filtered or updated state. The Kalman filter gain is calculated using (4.5). The equation (4.14) represents the measurement function (h_k).

$$h_k = \begin{bmatrix} C_n^b & 0 \\ 0 & C_n^b \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ g \\ h_x \\ h_y \\ h_z \end{bmatrix} \quad (4.14)$$

C_n^b is body to reference frame rotation matrix given by (4.15). $[0\ 0\ g]^{\mathrm{T}}$ is the gravity vector and $[h_x, h_y, h_z]^{\mathrm{T}}$ is magnetic field vector.

4.4. Linearization

It can be noted that (4.14) represents non-linear measurement equation because the rotation matrix C_n^b consists of non-linear quaternion terms. Therefore, the distribution propagates in a non-linear fashion. Since an extended Kalman filter is a first-order filter, it is necessary to perform first order differentiation of the measurement equation [34]. Linearization can be formed by taking Jacobian of measurement equation. Substituting (4.15) in (4.14).

h_k

$$= \begin{bmatrix} q_0^2+q_1^2-q_2^2+q_3^2 & 2(q_1q_2+q_0q_3) & 2(q_1q_3-q_0q_2) & 0 & 0 & 0 \\ 2(q_1q_2-q_0q_3) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3+q_0q_1) & 0 & 0 & 0 \\ 2(q_0q_2+q_1q_3) & 2(q_2q_3-q_0q_1) & q_0^2-q_1^2+q_2^2-q_3^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & q_0^2+q_1^2-q_2^2+q_3^2 & 2(q_1q_2+q_0q_3) & 2(q_1q_3-q_0q_2) \\ 0 & 0 & 0 & 2(q_1q_2-q_0q_3) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3+q_0q_1) \\ 0 & 0 & 0 & 2(q_0q_2+q_1q_3) & 2(q_2q_3-q_0q_1) & q_0^2-q_1^2+q_2^2-q_3^2 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ g \\ h_x \\ h_y \\ h_z \end{bmatrix}$$

$$h_k = \begin{bmatrix} 2(q_1q_3-q_0q_2)g \\ 2(q_2q_3+q_0q_1)g \\ (q_0^2-q_1^2+q_2^2-q_3^2)g \\ (q_0^2+q_1^2-q_2^2+q_3^2)h_x + 2(q_1q_2+q_0q_3)h_y + 2(q_1q_3-q_0q_2)h_z \\ 2(q_1q_2-q_0q_3)h_x + (q_0^2-q_1^2+q_2^2-q_3^2)h_y + 2(q_2q_3+q_0q_1)h_z \\ 2(q_0q_2+q_1q_3)h_x + 2(q_2q_3-q_0q_1)h_y + (q_0^2-q_1^2+q_2^2-q_3^2)h_z \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix} \quad (4.16)$$

Taking Jacobian of (16),

$$H_k = \frac{\partial\left(f_{1,2,3,4,5,6}\right)}{\partial\left(q_{0,1,2,3}\right)} = \begin{bmatrix} \frac{\partial f_1}{\partial q_0} & \frac{\partial f_1}{\partial q_1} & \frac{\partial f_1}{\partial q_2} & \frac{\partial f_1}{\partial q_3} \\ \frac{\partial f_2}{\partial q_0} & \frac{\partial f_2}{\partial q_1} & \frac{\partial f_2}{\partial q_2} & \frac{\partial f_2}{\partial q_3} \\ \frac{\partial f_3}{\partial q_0} & \frac{\partial f_3}{\partial q_1} & \frac{\partial f_3}{\partial q_2} & \frac{\partial f_3}{\partial q_3} \\ \frac{\partial f_4}{\partial q_0} & \frac{\partial f_4}{\partial q_1} & \frac{\partial f_4}{\partial q_2} & \frac{\partial f_4}{\partial q_3} \\ \frac{\partial f_5}{\partial q_0} & \frac{\partial f_5}{\partial q_1} & \frac{\partial f_5}{\partial q_2} & \frac{\partial f_5}{\partial q_3} \\ \frac{\partial f_6}{\partial q_0} & \frac{\partial f_6}{\partial q_1} & \frac{\partial f_7}{\partial q_2} & \frac{\partial f_6}{\partial q_3} \end{bmatrix}$$

$$H_k = \begin{bmatrix} -2gq_2 & 2gq_3 & -2gq_0 & -2gq_1 \\ 2gq_1 & 2gq_0 & 2gq_3 & 2gq_2 \\ 2gq_0 & -2gq_1 & -2gq_2 & -2gq_3 \\ 2h_xq_0 + 2h_yq_3 - 2h_zq_2 & 2h_xq_1 + 2h_yq_2 + 2h_zq_3 & 2h_yq_1 + 2h_xq_0 + 2h_xq_0 & 2h_xq_0 + 2h_xq_0 + 2h_xq_0 \\ 2h_yq_0 - 2h_xq_3 + 2h_zq_1 & 2h_xq_2 - 2h_yq_1 + 2h_zq_0 & 2h_xq_1 + 2h_yq_2 + 2h_zq_3 & 2h_zq_2 - 2h_yq_3 - 2h_xq_0 \\ 2h_xq_2 - 2h_yq_0 + 2h_zq_1 & 2h_xq_3 - 2h_yq_0 - 2h_zq_1 & 2h_xq_0 + 2h_yq_3 - 2h_zq_2 & 2h_xq_1 + 2h_yq_2 + 2h_zq_3 \end{bmatrix} \quad (4.17)$$

4.5. Extended Kalman Filter Algorithm

The steps of the Modified Extended Kalman Filter Algorithm are as follows

a. Initialize the state $q_{k-1|k-1}$ and the error covariance matrix $P_{k-1|k-1}$.

b. Estimate the state $q_{k|k-1}$ using (4.11).

c. Estimate the error covariance matrix $P_{k|k-1}$ using (4.18).

$$P_{k|k-1} = F\left(k, \hat{x}_{k-1|k-1}\right)P_{k-1|k-1}F^T\left(k, \hat{x}_{k-1|k-1}\right) + Q_k \tag{4.18}$$

i. $F\left[k, \hat{x}_{k-1|k-1}\right]$ is a state transition matrix and it is calculated using (4.13).

ii. $P_{k-1|k-1}$is from above step a.

iii. Q_k is process noise covariance matrix. Here, each diagonal elements of Q_k are assigned as a standard deviation of noise of gyroscope readings.

d. Determine Kalman filter Gain, K_k using (4.19).

$$K_k = P_{k|k-1}H^T\left(k, \hat{x}_{k|k-1}\right)\left\{H\left(k, \hat{x}_{k|k-1}\right) P_{k|k-1}H^T\left(k, \hat{x}_{k|k-1}\right) + R_k\right\}^{-1} \tag{4.19}$$

i. $H\left[k, \hat{x}_{k|k-1}\right]$ is calculated using (4.17).

ii. $P_{k|k-1}$ is obtained from step c.

iii. R_k is measurement noise covariance matrix. Here, R_k is assigned as the square of standard deviation of measurement readings data at steady state condition.

e. Update $x_{k|k}$ and $P_{k|k}$ using (4.20) and (4.21) respectively

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\left(z_k - h(\hat{x}_{k|k-1}, k)\right) \tag{4.20}$$

$$P_{k|k} = P_{k|k-1} - K_k H\left(\hat{x}_{k|k-1}, k\right)P_{k|k-1} \tag{4.21}$$

i. $\hat{x}_{k|k-1}$ is obtained from step b.

ii. K_k is obtained from step d.

iii. z_k is measurements at time step k.

iv. $h(\hat{x}_{k|k-1}, k)$ is obtained from (4.16).

v. $P_{k|k-1}$ is obtained from step c.

vi. $H\left(\hat{x}_{k|k-1}, k\right)$ is calculated using (4.17).

4.6. Simulated performance of the Extended Kalman filter under steady state condition

4.6.1. Introduction

The algorithm described in section 4.5 is implemented to get the estimate of states at steady state condition i.e. tri-axis sensors are at rest on a flat surface. The readings from tri-axis gyroscope i.e. angular velocities in *x, y* and *z* directions should theoretically be zero when the gyroscope is place at rest. However, the readings were noisy as shown in Figure 1.a. The readings from tri-axis accelerometer should be 0, 0 and -9.8 m/sec^2 in *x, y* and *z* directions, respectively when it is at rest or moving with no acceleration. This is known as the local gravity vector and it is a function of latitude and altitude at a given location. Also, the tri-axis magnetometer should read 0.19566, 0.02889 and 0.49007 Gauss in *x, y* and *z* directions, respectively for a given location when it is placed in the North East Down (NED) frame. This is known as local magnetic vector and it is a function of latitude and longitude of a given location.

4.6.2. Process and Measurement noise covariance matrix.

The mis-modeled system and measurement dynamics, the Taylor approximation used for linearization and the existence of hidden states not modeled in EKF are captured as a noise and the effects of these noise will cause perturbations in the state estimation and

measurements [35]. In this chapter, to reduce the effect of the noise, the standard deviation of readings from each axis sensor is determined at steady state condition assuming that both process and measurement noise are uncorrelated white noise with zero mean. The square of calculated standard deviation of process noise and measurement noise are assigned in the main diagonal term of process noise covariance matrix (*Q*) and measurement noise covariance matrix (*R*) respectively. Here, for implementing EKF at steady state condition, both *Q* and *R* is assigned constant number obtained from the calculation of standard deviation.

$$K_k = P_{k|\mathrm{k}-1} H^T\left(k, \hat{x}_{k|\mathrm{k}-1}\right)\left\{H\left(k, \hat{x}_{k|\mathrm{k}-1}\right) P_{k|\mathrm{k}-1} H^T\left(k, \hat{x}_{k|\mathrm{k}-1}\right) + R_k\right\}^{-1} \qquad (4.22)$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\left(z_k - h(\hat{x}_{k|k-1}, k)\right) \qquad (4.23)$$

The good thing about the EKF is that user has a choice to vary *R* in the Kalman filter gain equation given in (4.22). If measurement noise is large, *R* can be assigned a large number which will make Kalman filter gain (*K*) small. If *K* is small, the credibility to the measurements won't be given when making correction in the priori estimated states. Similarly, if measurement noise is small, *R* can be assigned a small number which may be close to the covariance of measurement noise. As a result, K will be large. If *K* is large, the credibility to the measurements will be given a lot when making correction in the priori estimated states.

In the equation (4.23), if the difference between the quantity that is being measured by the measurement device (z_k) and the quantity that is obtained from the measurement equation ($h(\hat{x}_{k|k-1}, k)$) is large, the correction in the priori states may be screwed up. This problem can be eliminated by providing very low reliability in the readings from

measurement sensors. The reliability of the measurements depends on the value of R. If R is higher, lower will be the reliability on the measurements.

The readings from stationary tri-axis accelerometer and magnetometer are shown in Figure 1.b and 1.c respectively. The standard deviations of readings from each sensor at stationary condition are calculated. The standard deviations of readings from stationary tri-axis gyroscope, accelerometer and magnetometer are shown in Table 1. The square of standard deviation gives the covariance of readings for each sensor.

Table 4.1. The standard deviation of white noise of tri-axis gyroscope, accelerometer and magnetometer.

Gyroscope Reading $^{deg}/_{s}$			Accelerometer Reading $^{m}/_{s^2}$			Magnetometer Reading Gauss		
σ_{ω_x}	σ_{ω_y}	σ_{ω_z}	σ_{a_x}	σ_{a_y}	σ_{a_z}	σ_{m_x}	σ_{m_y}	σ_{m_z}
0.00056	0.00067	0.0007	0.0142	0.0374	0.0275	0.0016	0.0029	0.0023

The extended Kalman filter is implemented by feeding the angular velocity obtained from stationary tri-axis gyroscope to predict the state. The process noise standard deviation is assumed to be approximately 0.0007 $\frac{deg}{s}$ based on Table 4.1. The process noise covariance is obtained by squaring standard deviation of process noise. The states are updated by feeding measurements from tri-axis accelerometer and magnetometer. From Table 4.1, the measurement noise standard deviations are assumed to be 0.04 m/s^2 and 0.004 Gauss for tri-axis accelerometer and magnetometer respectively.

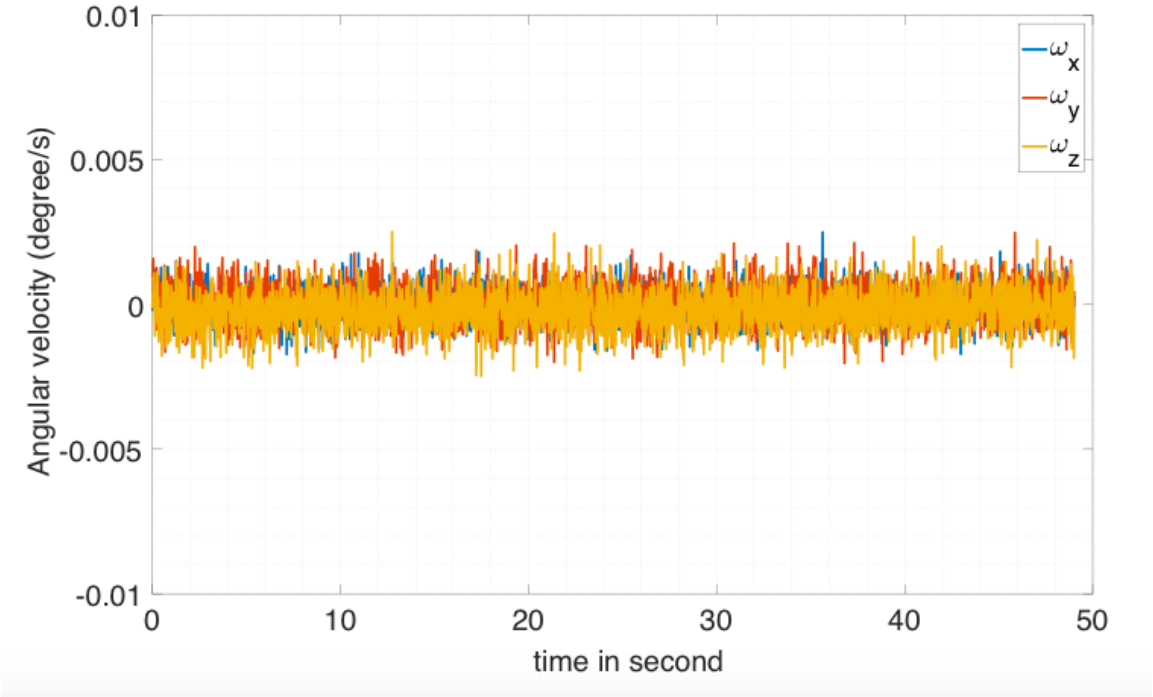

Figure 4.1. Angular velocity readings in x, y and z direction when device is at rest

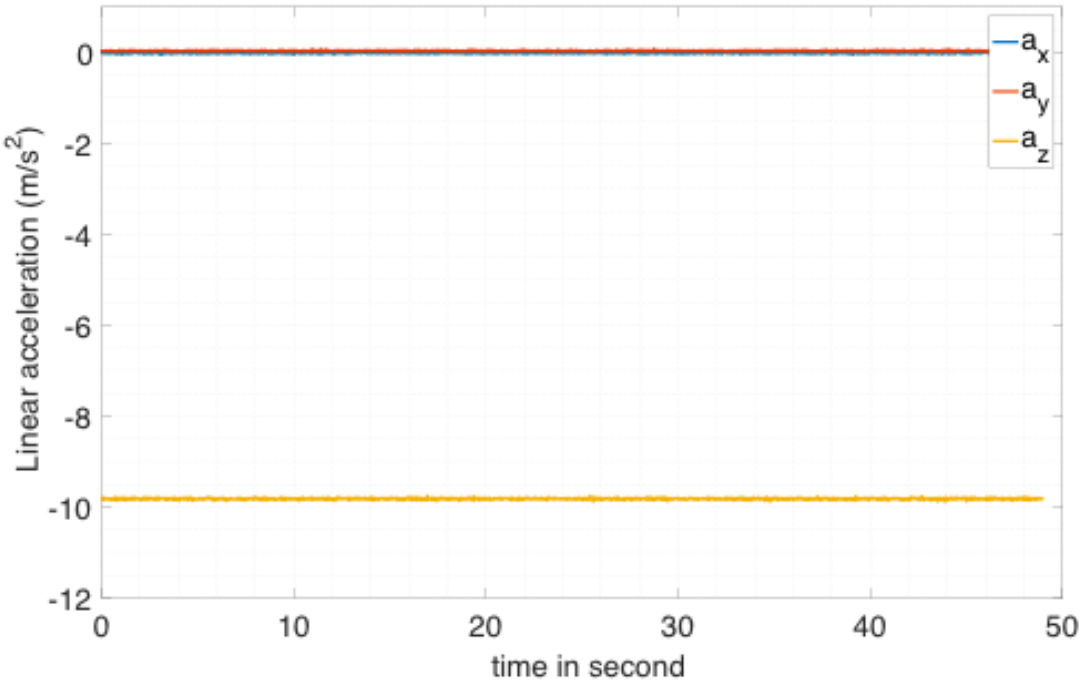

Figure 4.2. Accelerometer readings in x, y and z direction when device is at rest

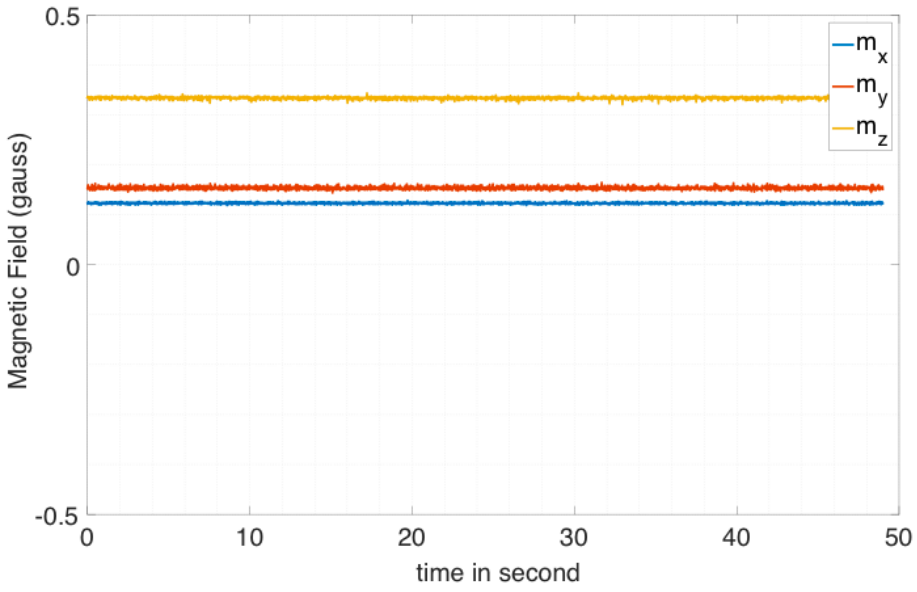

Figure 4.3. Magnetometer readings in x, y and z direction when device is at rest

4.6.3. Simulated Results

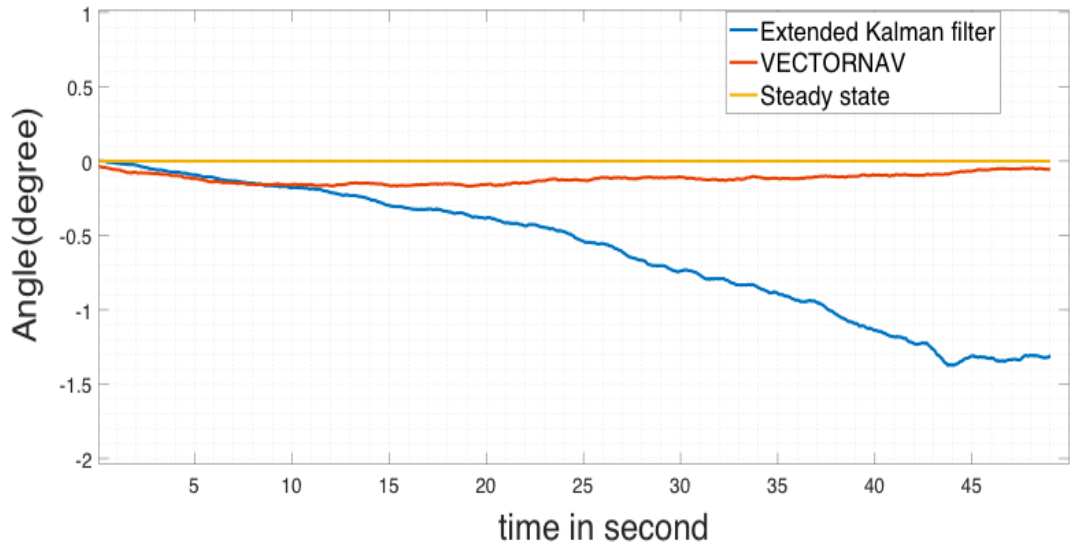

Figure 4.4. The yaw angle estimation at steady state condition using the extended Kalman filter.

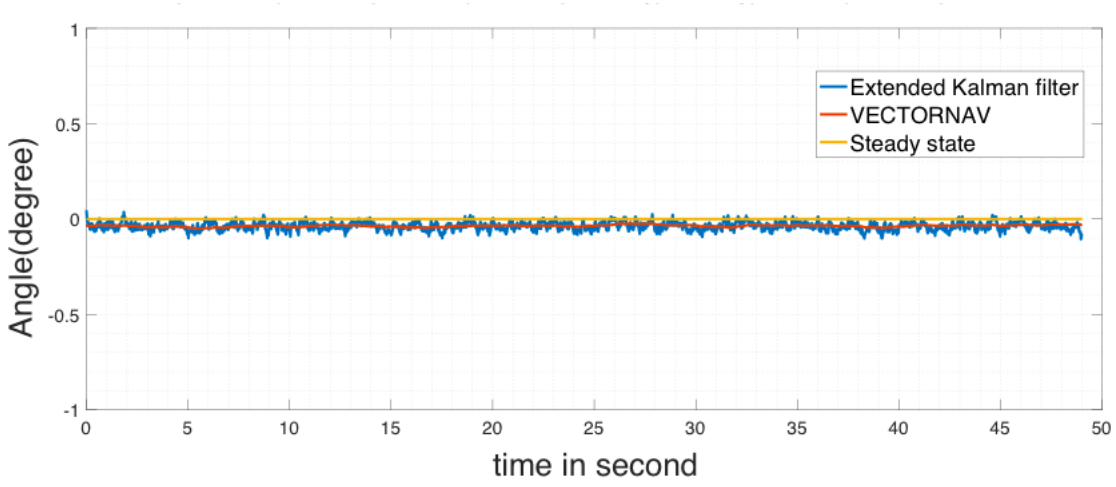

Figure 4.5. The pitch angle estimation at steady state condition using the extended Kalman filter.

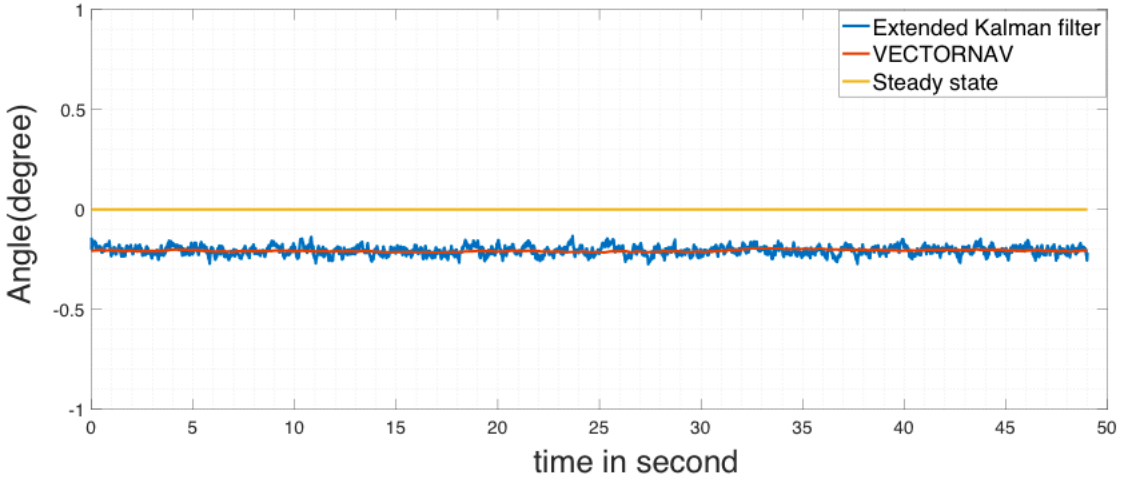

Figure 4.6. The roll angle estimation at steady state condition using the extended Kalman filter.

4.6.4 Discussion of Result

The estimated yaw, pitch and roll for steady state condition is shown in Figure (4.4), (4.5) and (4.6) respectively. The yaw pitch and roll should ideally be zero because the device is kept stationary on a flat surface; z-axis of device faces vertically downward.

However, it is expected to get noisy estimation of yaw, pitch and roll close to zero degree. This is due to the fact that the readings from the tri-axis gyroscope, accelerometer and magnetometer are affected by noise. For simplicity, the extended Kalman filter algorithm presented in section 4.5 is named as EKF. In the Figure (4.4), the yaw angle estimated by EKF is propagating in such a way that the error is increasing with respect to time. It is observed that the error in yaw angle estimation by EKF is approximately 1^o after 45 seconds. The error in yaw angle estimated by VECTORNAV initially increases by 0.2^0 and finally converges to zero. The EKF did not perform as well as whatever they used in VECTORNAV. The VECTORNAV refers to the company from which the IMU device is bought. The readings from the gyroscope, the accelerometer and the magnetometer of VECTORAV is used in the EKF estimation. In VECTORAV device, an onboard extended Kalman filter is installed for attitude estimation. The estimated results of VECTORNAV is compared with the estimated result of EKF algorithm. In the Figure (4.5), the pitch angle estimated by is propagating with an error of 0.05^o and the roll angle is propagating with an error of 0.2^o approximately. The pitch and roll angle estimation by VECTORNAV and EKF is approximately the same but the estimation by EKF is noisier. Also, it is expected that the pitch and roll angle estimation should be drifted. But, it is observed only noisy because the pitch and yaw angle estimation are relying more on measurements rather than process. This may be the result of assignment of same co-variance for three gyroscopes and same co-variance for three accelerometers. In practical sensor's covariance in each axis may differ.

4.7. Conclusion

In this chapter, the extended Kalman filter is introduced for estimation of attitude by fusing the readings from the tri-axis gyroscope, the tri-axis accelerometer and the tri-axis magnetometer. For implementation of extended Kalman filter, the quaternion-based state equation is formulated which utilizes angular velocities measured by tri-axis gyroscope to predict the states. The readings from the accelerometer and the magnetometer are utilized as the measurements. The measurement equation is function of the state variable and measurement readings. The update of predicted states relies on the Kalman gain. If the Kalman gain is high, the EKF places more weight on the measurement and if the Kalman gain is low, the EKF places more weight on the process. The standard deviation of process noise and measurement noise is calculated for steady state condition. The squared value of the standard deviation of process noise and measurement noise are respectively chosen as the process noise covariance and measurement noise covariance in the EKF algorithm. The yaw, pitch and roll angle are estimated for steady state condition. It is observed that the yaw angle propagated with increasing error. This is due to the fact that the bias term of the process was not taken into account in the state equation. Therefore, the bias was accumulated and caused gradual increase in the error with respect to time.

The performance of EKF under steady state condition is not good compared to the performance of the VECTORNAV. For better performance, state equation is modified in the next chapter by adding measurement biases in the process equation. Also, the process noise covariance matrix is assigned for each iterations as a function of sampling time, current states and biases.

CHAPTER V

MODIFIED EXTENDED KALMAN FILTER

5.1 Introduction

The algorithm used in previous chapter 4 has quaternions as states. The process and measurement noise covariance matrix were assigned as fix numbers, i.e. no adaptation to improve performance. If one uses the algorithm described in previous Chapter 4, the accumulation of bias causes drift in the Euler's angle estimation. In this chapter, to eliminate the effect of bias term, the state equation is modified by adding measurement biases as a process. Also, the process noise covariance is calculated for each iteration using formula given in equation (5.3). Since changes have been made in state equation and the process noise covariance, the algorithm for the extended Kalman filter in this chapter is defined as modified Extended Kalman Filter. The state equation shown in (5.1) consists of the quaternion based state equation and the bias terms for the accelerometer and magnetometer measurements. The measurement equation given in (4.14) is also repeated here for completeness.

$$x_{k+1} = \begin{bmatrix} q_{k+1} \\ a_{b\,k+1} \\ m_{b\,k+1} \end{bmatrix} = \begin{bmatrix} \exp(\emptyset_k T_s) & 0_{4\times3} & 0_{4\times3} \\ 0_{3\times4} & I_{3\times3} & 0_{3\times3} \\ 0_{3\times4} & 0_{3\times3} & I_{3\times3} \end{bmatrix}_{10\times10} \begin{bmatrix} q_k \\ a_{b\,k} \\ m_{b\,k} \end{bmatrix}_{10\times1} + \begin{bmatrix} w_{q\,k} \\ w_{a\,k} \\ w_{m\,k} \end{bmatrix}_{10\times1} \quad (5.1.a)$$

$$\emptyset = \frac{1}{2} \begin{bmatrix} 0 & -w_x & -w_y & -w_z \\ w_x & 0 & -w_z & w_y \\ w_y & w_z & 0 & -w_x \\ w_z & -w_y & w_x & 0 \end{bmatrix} \quad (5.1.c)$$

Where, $x_{k+1} \in \mathbb{R}^{10\times1}$ is the estimated state variable, $q_{k+1} \in \mathbb{R}^{4\times1}$ is the estimated quaternion state i.e. $[\,q_0\ q_1\ q_2\ q_3\,]^T$, $a_{b_{k+1}} \in \mathbb{R}^{3\times1}$ and $m_{b_{k+1}} \in \mathbb{R}^{3\times1}$ are estimated biases of accelerometer and magnetometer, respectively, $\emptyset_k \in \mathbb{R}^{4\times4}$ is quaternion based state function and it is given in (5.1.c), $w_{q_k} \in \mathbb{R}^{3\times1}$ is process noise coming from the gyroscope readings, w_{a_k} and w_{m_k} are the noise terms associated with the accelerometer and magnetometer bias. $[0\ 0\ g]^{\mathrm{T}}$ is the gravity vector and $[h_x, h_y, h_z]^{\mathrm{T}}$ is magnetic field vector.

The state transition matrix (Ω) for a state equation (5.1) is given in (5.2). Since the state equation is linear, the Jacobian matrix is the same as the state transition matrix. The process noise covariance is obtained using (5.3). The process noise covariance matrix (Q) is function of quaternion state variable, sampling time and standard deviation of the gyroscope, the accelerometer and the magnetometer measurements[36].

$F =$

$$\begin{bmatrix}
1 & -w_x\frac{T_s}{2} & -w_y\frac{T_s}{2} & -w_z\frac{T_s}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\
w_x\frac{T_s}{2} & 1 & -w_z\frac{T_s}{2} & w_y\frac{T_s}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\
w_y\frac{T_s}{2} & w_z\frac{T_s}{2} & 1 & -w_x\frac{T_s}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\
w_z\frac{T_s}{2} & -w_y\frac{T_s}{2} & w_x\frac{T_s}{2} & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{bmatrix} \quad (5.2.a)$$

0	0	0	0	0	0	0	0	0	1

$$x_{k+1} = \Omega x_k \tag{5.2.b}$$

$$Q_k = \begin{bmatrix} \left(\frac{T_s}{2}\right)^2 \Sigma_k \Sigma_g {\Sigma_k}^T & 0 & 0 \\ 0 & \Sigma_a & 0 \\ 0 & 0 & \Sigma_m \end{bmatrix} \tag{5.3}$$

Where, $\Sigma_g \in \mathbb{R}^{3\times3}$ is the covariance of gyroscope measurements, $\Sigma_a \in \mathbb{R}^{3\times3}$ is the covariance of accelerometer measurement, and $\Sigma_m \in \mathbb{R}^{3\times3}$ is the covariance of magnetometer measurement. $\Sigma_k \in \mathbb{R}^{4\times3}$ is given by (5.4).

$$\Sigma_k = \begin{bmatrix} [e_k \times] + q_{0k} I \\ -{e_k}^T \end{bmatrix} = \begin{bmatrix} q_1 + q_0 & 0 & 0 \\ 0 & q_1 + q_0 & 0 \\ 0 & 0 & q_3 + q_0 \\ -q_1 & -q_2 & -q_3 \end{bmatrix} \tag{5.4}$$

Where, $e_k \in \mathbb{R}^{3\times1}$ is $[\, q_1 \; q_2 \; q_3]^T$, q_{0k} is a scalar term of quaternion, *I* is an identity matrix and $[e_k \times]$ is a diagonal matrix containing vector part of quaternion as a diagonal element and it is given in (5.5).

$$[e_k \times] = \begin{bmatrix} q_1 & 0 & 0 \\ 0 & q_2 & 0 \\ 0 & 0 & q_3 \end{bmatrix} \tag{5.5}$$

The state equation given in (5.1) has four quaternion state variables, three bias variables as a state from *tri-axis* accelerometer measurement and three bias variables as a state from *tri-axis* magnetometer. Therefore, the measurement equation given in (5.1.b) should be linearized by performing first order differentiation using ten states. Actually linearization is only needed if you are using this particular extended Kalman filter. The Jacobian matrix (H_k) is given by (5.6).

$$H_k = [\, H \;\; 0_{6\times3} \;\; 0_{6\times3} \,] \tag{5.6}$$

Where, H $\in$ $\mathbb{R}^{6\times4}$ is given by (5.7).

$$h_k =$$

$$\begin{bmatrix} q_0^2+q_1^2-q_2^2+q_3^2 & 2(q_1q_2+q_0q_3) & 2(q_1q_3-q_0q_2) & 0 & 0 & 0 \\ 2(q_1q_2-q_0q_3) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3+q_0q_1) & 0 & 0 & 0 \\ 2(q_0q_2+q_1q_3) & 2(q_2q_3-q_0q_1) & q_0^2-q_1^2+q_2^2-q_3^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & q_0^2+q_1^2-q_2^2+q_3^2 & 2(q_1q_2+q_0q_3) & 2(q_1q_3-q_0q_2) \\ 0 & 0 & 0 & 2(q_1q_2-q_0q_3) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3+q_0q_1) \\ 0 & 0 & 0 & 2(q_0q_2+q_1q_3) & 2(q_2q_3-q_0q_1) & q_0^2-q_1^2+q_2^2-q_3^2 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ g \\ h_x \\ h_y \\ h_z \end{bmatrix} + v_k$$

(5.1.b)

$$H = \begin{bmatrix} -2gq_2 & 2gq_3 & -2gq_0 & -2gq_1 \\ 2gq_1 & 2gq_0 & 2gq_3 & 2gq_2 \\ 2gq_0 & -2gq_1 & -2gq_2 & -2gq_3 \\ 2h_xq_0+2h_yq_3-2h_zq_2 & 2h_xq_1+2h_yq_2+2h_zq_3 & 2h_yq_1+2h_xq_0+2h_xq_0 & 2h_xq_0+2h_xq_0+2h_xq_0 \\ 2h_yq_0-2h_xq_3+2h_zq_1 & 2h_xq_2-2h_yq_1+2h_zq_0 & 2h_xq_1+2h_yq_2+2h_zq_3 & 2h_zq_2-2h_yq_3-2h_xq_0 \\ 2h_xq_2-2h_yq_0+2h_zq_1 & 2h_xq_3-2h_yq_0-2h_zq_1 & 2h_xq_0+2h_yq_3-2h_zq_2 & 2h_xq_1+2h_yq_2+2h_zq_3 \end{bmatrix}$$

(5.7)

5.2. Modified Extended Kalman Filter Algorithm

The steps of the Modified Extended Kalman Filter Algorithm are as follows

a. Initialize the state $x_{k-1|k-1}$ and the error covariance matrix $P_{k-1|k-1}$.

b. Estimate the state $x_{k|k-1}$ using (5.2.b).

c. Estimate the error covariance matrix $P_{k|k-1}$ using (5.8).

$$P_{k|k-1} = F\left(k, \hat{x}_{k-1|k-1}\right)P_{k-1|k-1}F^T\left(k, \hat{x}_{k-1|k-1}\right) + Q_k \quad (5.8)$$

iv. $F\left[k, \hat{x}_{k-1|k-1}\right]$ is a state transition matrix and it is calculated using (5.2).

v. $P_{k-1|k-1}$is from above step a.

vi. Q_k is calculated using (5.3)

d. Determine Kalman filter Gain, K_k using (5.9).

$$K_k = P_{k|k-1}H^T\left(k, \hat{x}_{k|k-1}\right)\left\{H\left(k, \hat{x}_{k|k-1}\right) P_{k|k-1}H^T\left(k, \hat{x}_{k|k-1}\right) + R_k\right\}^{-1} \quad (5.9)$$

iv. $H\left[k, \hat{x}_{k|k-1}\right]$ is calculated using (5.6).

v. $P_{k|k-1}$ is obtained from step c.

vi. R_k is measurement noise covariance matrix. Here, R_k is obtained by calculating square of standard deviation of raw measurement data at steady state condition.

e. Update $x_{k|k}$ and $P_{k|k}$ using (5.10) and (5.11) respectively

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\left(z_k - h(\hat{x}_{k|k-1}, k)\right) \quad (5.10)$$

$$P_{k|k} = P_{k|k-1} - K_k H\left(\hat{x}_{k|k-1}, k\right)P_{k|k-1} \quad (5.11)$$

vii. $\hat{x}_{k|k-1}$ is obtained from step b.

viii. K_k is obtained from step d.

ix. z_k is measurements at time step k.

x. $h(\hat{x}_{k|k-1}, k)$ is obtained from (5.1.b).

xi. $P_{k|k-1}$ is obtained from step c.

xii. $H\left(\hat{x}_{k|k-1}, k\right)$ is calculated using (5.6)

5.3 Implementation of the Algorithm for the Steady State Conditions.

The algorithm described in section 5.2 is implemented to get the steady state response i.e. the IMU sensor is placed at rest on a flat table. The algorithm is initialized using initial quaternion, and initial measurement bias. The IMU device can be reset to get zero attitude and the initial quaternion is assumed to be zero attitude. The zero attitude refers to quaternion with scalar part as one and all vector part as zeros i.e. $[\, q_0\ q_1\ q_2\ q_3\,]^{\,T} = [\,1\ 0\ 0\ 0\,]^{\,T}$. The initial quaternion states is assumed to be [1 0 0 0]. Also, the initial measurement biases are assumed to be zero.

The readings from *tri-axis* gyroscopes are the input to the state equation to predict the state. The readings from tri-axis accelerometer and tri-axis magnetometer are input to measurement equation which is utilized to update the predicted states. The estimated yaw, pitch and roll for the steady state condition is shown in Figures 5.1, 5.2 and 5.3 respectively. The Figures 5.1 to 5.3 show estimation of attitude using the extended Kalman Filter algorithm described in section 5.2 and the attitude estimation generated by VECTORNAV software.

Since the IMU device is at rest, it is expected that the yaw, pitch and roll should ideally be zero. However, the negligible vibration and noise may cause error in estimation.

In Figure 5.1, the yaw angle obtained from the extended Kalman filter has an error less than 0.3^0 lasting approximately for 45 seconds. Similarly, the yaw angle obtained from the VECTORNAV software has error less than 0.2^0 lasting approximately for 45 seconds. The yaw estimations in both cases are trying to converge to zero after 45 seconds. It can be concluded that the performance of extended Kalman filter in the steady state condition is better if biases from measurements are accounted for. Also, in the algorithm described in 5.2, the process noise co-variance matrix is calculated using (5.3) for each iteration instead of providing a guess.

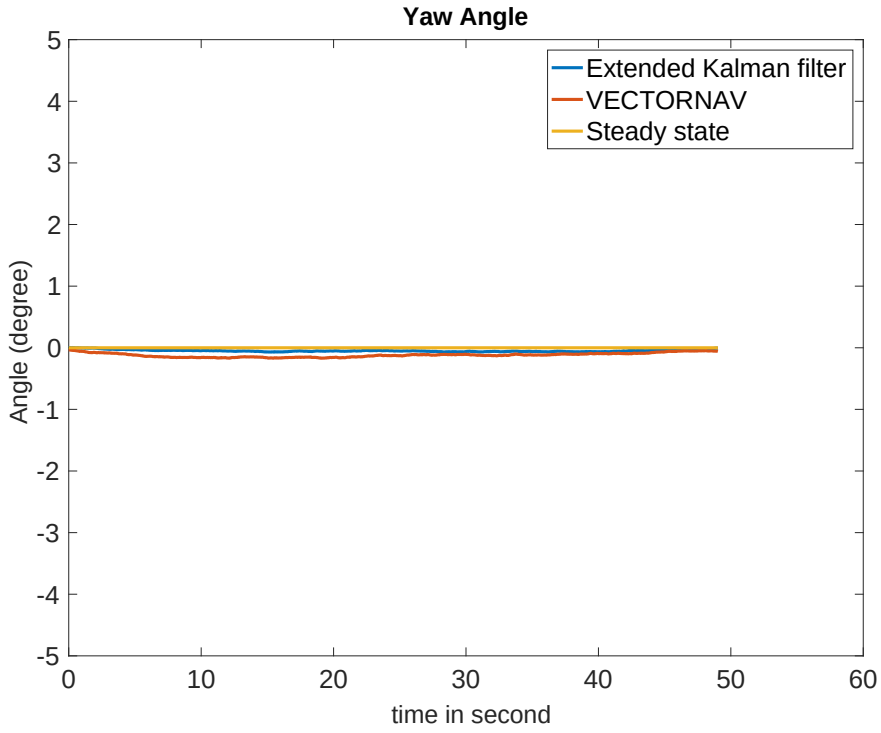

Figure 5.1. Estimated yaw angle using the modified extended Kalman filter and the VECTORNAV software.

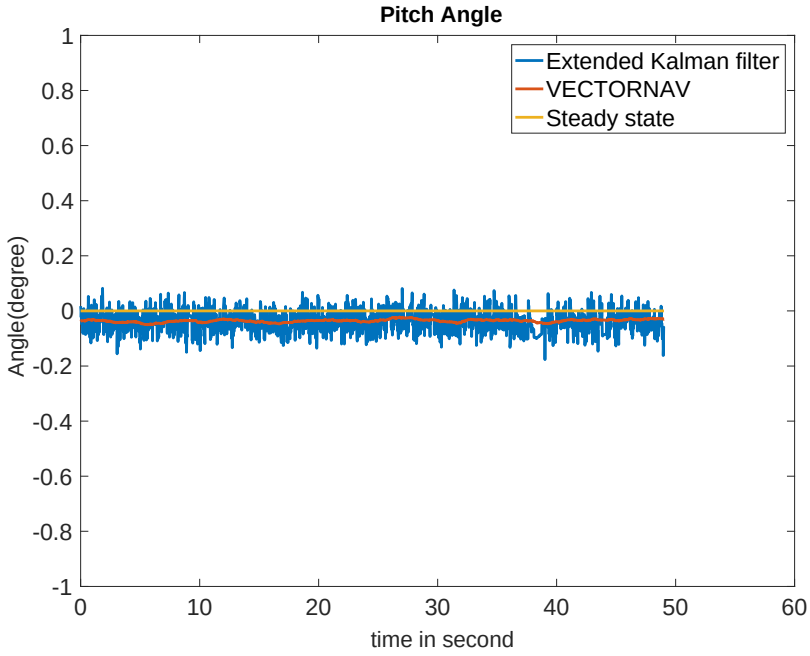

Figure 5.2. Estimated yaw angle using the modified extended Kalman filter and VECTORNAV software.

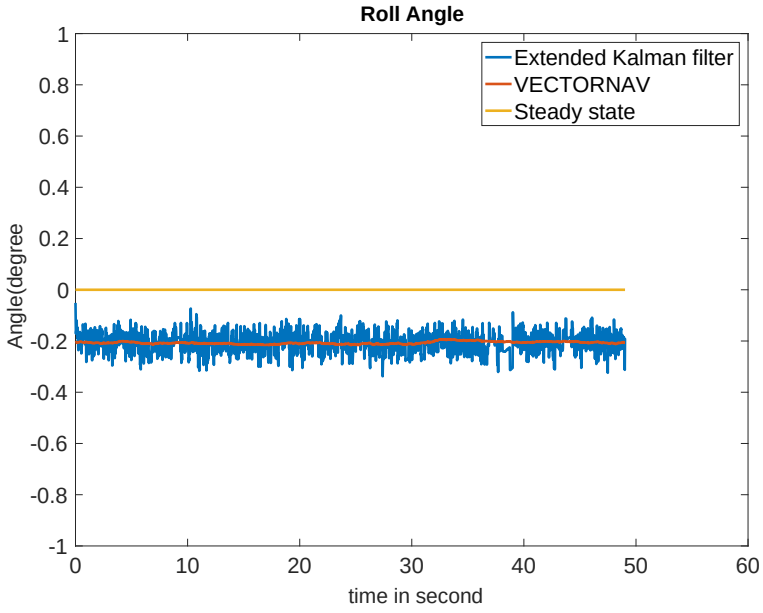

Figure 5.3. Estimated yaw angle using the modified extended Kalman filter and VECTORNAV software.

5.4. Simulated performance of the Modified Extended Kalman filter under body acceleration.

5.4.1. Introduction

An extended Kalman filter computes 3D orientation of a body utilizing measurements provided by the *tri-axis* gyroscope, accelerometer and magnetometer. The gyroscope measures short term relative change in the orientation, the accelerometer measures both the gravitational acceleration and the acceleration of body due to translational or rotational motion. The magnetometer measures the effect of the earth's magnetic field and the magnetic field created by the surrounding ferromagnetic materials. The attitude estimation encounters two signal processing difficulties. First, the accelerometer is sensitive to both gravity and body accelerations during dynamic conditions. The readings from accelerometer is a vector sum of gravitational acceleration and external acceleration. Therefore, the gravitational acceleration cannot be distinguished from external acceleration unless there is another device that can be used to calculate the body acceleration such as a GPS unit.

Second, the integration of gyroscope signal will cause drift errors since its bias and noise are also integrated. However, adding measurement biases to the state equation and changing process noise covariance matrix will reduce the drift errors. The change in process noise covariance matrix refers to calculating covariance matrix as a function of sampling time, current state and covariance of each sensor readings as explained in section 5.1. Also, the gyro drifts can be minimized by resetting the gyro after some time interval. But, this may lead to incorrect prediction of initial states which may cause inaccuracy in the attitude estimation.

If there is no significant body acceleration, the extended Kalman Filter algorithm has sufficient information for attitude estimation. This assumption works well for application where body does not experience long term acceleration such as when it used on large marine vehicles. But, this assumption is violated when the body is subjected to external acceleration such as airplane operating in banked turn. During a banked turn, an airplane is subjected to long term acceleration due to the centripetal force. The centripetal force is inward acting force on a body when traveling along a curved path.

Let us consider a body travelling on a curve path whose instantaneous radius is *r* and the center of rotation is *O*. The velocity vector $\vec{v}$ is a tangent to the circular path at point *A* as shown in Figure 5.4.a. This means, the velocity vector is perpendicular to the radius vector $\vec{r}$. Similarly, the acceleration vector $\overrightarrow{a_c}$ is a normal to $|\vec{v}|$ as shown in Figure 5.4.b. The acceleration vector is directed inward to the center of rotation. This acceleration is known as centripetal acceleration. The body that is moving in a circular path experiences inward force called centripetal force. The centripetal acceleration is given by (5.7.a) where, v velocity, r is instantaneous radius and ω is angular velocity. If the speed is not constant, body experiences acceleration in direction tangent to the path of motion. This acceleration is known as tangential acceleration (a_T) and it is given by (5.7.b), where a is angular acceleration.

$$a_c = \frac{v^2}{r} = \omega^2 r \tag{5.7.a}$$

$$a_T = a\, r \tag{5.7.b}$$

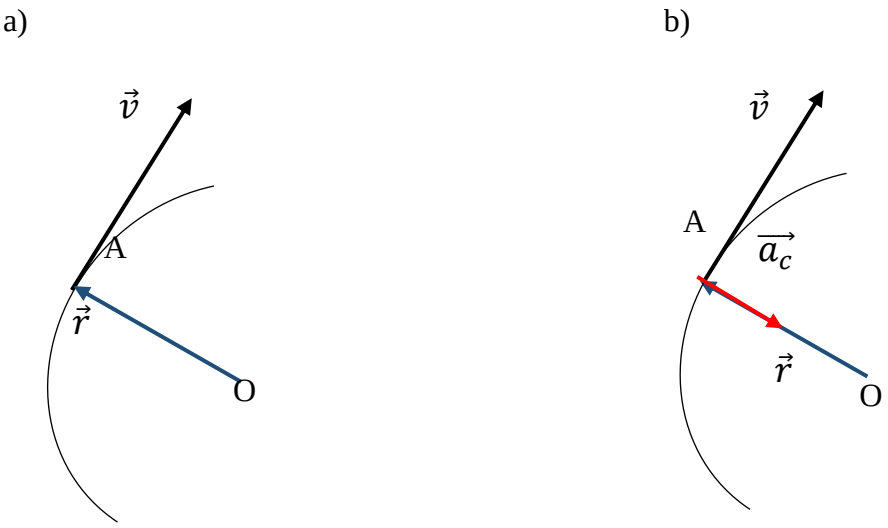

Figure 5.4. a) shows velocity vector in a curve path; b) shows direction of centripetal acceleration

If the body is traveling in a curve path, the acceleration measured by the tri-axis accelerometer is the sum of the centripetal acceleration and gravitational acceleration as shown in equation (5.8). In attitude estimation using extended Kalman Filter, difference between the norm of acceleration obtained from measurement equation and the norm of gravity vector should ideally be zero if the device is at steady state condition. In practical, this difference may be non-zero or close to zero due to the noise and biases in process and measurement equation. Also, the addition of body acceleration leads to large difference in accelerometer readings and gravity vector. Therefore, difference between the norm of accelerometer readings and the norm of gravitational acceleration should be detected. This difference is known as error and it is given in (5.9). The judgment on the reliance of measurement device depends on this error. If the error is large the measurement noise covariance matrix should be assigned large number so that a reduced confidence on the Kalman filter is imposed. Increasing the covariance matrix elements effectively lowers the

value of Kalman Filter gain and make the measurement equations less reliable. In this case, the update of the states solely depends on state equation prediction.

$$a_m = C_B^N(\vec{g} + a_{body}) \tag{5.8}$$

$$e_k = \left||z_k| - |\vec{g}|\right| \tag{5.9}$$

Where, a_m is measurement and z_k is a measured acceleration by tri-axis accelerometer, C_B^N is body to reference frame rotation matrix, $\vec{g}$ is gravity vector and a_{body} is external acceleration or body acceleration.

5.4.2. Low pass filter

When an IMU device is subjected to an unknown body acceleration, the measurements from accelerometer is not reliable since it generates error as given in (5.9). If the accelerometer readings are not reliable, the effect of this error can be eliminated by avoiding the readings from accelerometer. This can be done by providing large measurement noise covariance matrix. The measurement noise covariance matrix is assigned based on the error. Also, it is necessary to smooth the error by using a low pass filter. Smoothing of errors will eliminate the high fluctuation in assignment of measurement noise covariance matrix. The simple model for the low pass filter is given in (5.10).

$$y[k] = \alpha\, y[k-1] + (1-\alpha)\, x[k] \tag{5.10}$$

Where, $x[k]$ is signal at current time instant k that has to be filtered, $y[k]$ is a filtered signal at current time instant k, $y[k\text{-}1]$ is filtered signal at previous time instant $k\text{-}1$, α is smoothing factor such that $0 \leq \alpha \leq 1$.

The error before and after low pass filter is shown in Figure (5.5). Here, the IMU device is subjected to body acceleration by placing it firmly on the rotating disk of Torsional plant. The procedure to rotate the disk is given in Appendix I. The tri-axis accelerometer reading from IMU device is recorded. The error is calculated for each time step using (5.10). If the device is stationary, ideally the error should be zero. However, due to noises in accelerometer, small error can be observed. This is known as threshold error. The blue line indicates unfiltered error and the red line indicates filtered error. It is observed that the error gets smoother after applying low pass filter. The magnitude of unfiltered error signal is decreased and the measurement covariance matrix is assigned based on the magnitude of filtered error signal. In region *A,* as shown in Figure 5.5, the magnitude of the filtered error is large although the magnitude of unfiltered error is within its threshold. Therefore, the measurement covariance matrix is assigned a large number in region *A* although IMU device is not subjected to body acceleration. This is one of the disadvantage of using low pass filter. Also, the use of low pass filter will delay the estimation.

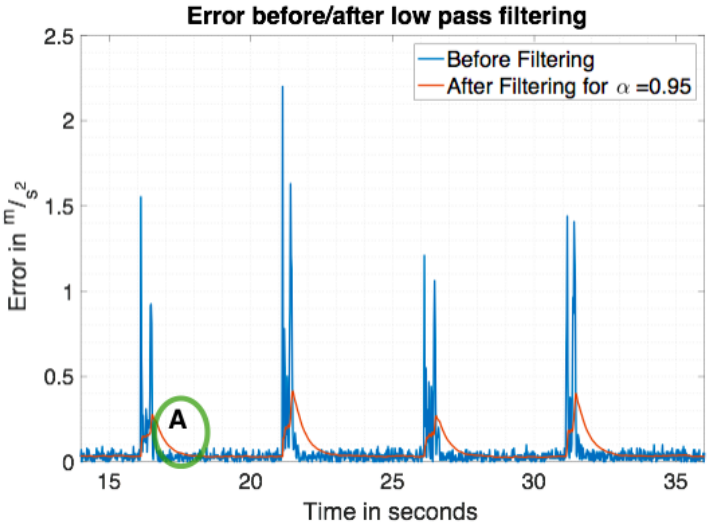

Figure 5.5. The error before and after low pass filtering when IMU device is subjected to body acceleration.

5.4.3. Assignment of measurement noise covariance (R) with respect to error.

If the device is subjected to body acceleration, measurement noise covariance matrix in the algorithm is assigned based on the error (e_k). During an attitude estimation, the error is calculated for each iteration and the measurement noise covariance matrix is chosen which corresponds to the given error. The measurement noise covariance is provided in series of steps with respect to error i.e. the measurement noise covariance is constant for given range of error and the step increases with increasing range of errors. The measurement noise covariance matrix for different range of error is shown in figure (5.6). The measurement noise covariance will be slightly different for increasing error and decreasing error. This will avoid the high fluctuation in assignment of measurement noise covariance at a point where the measurement covariance jumps or falls.

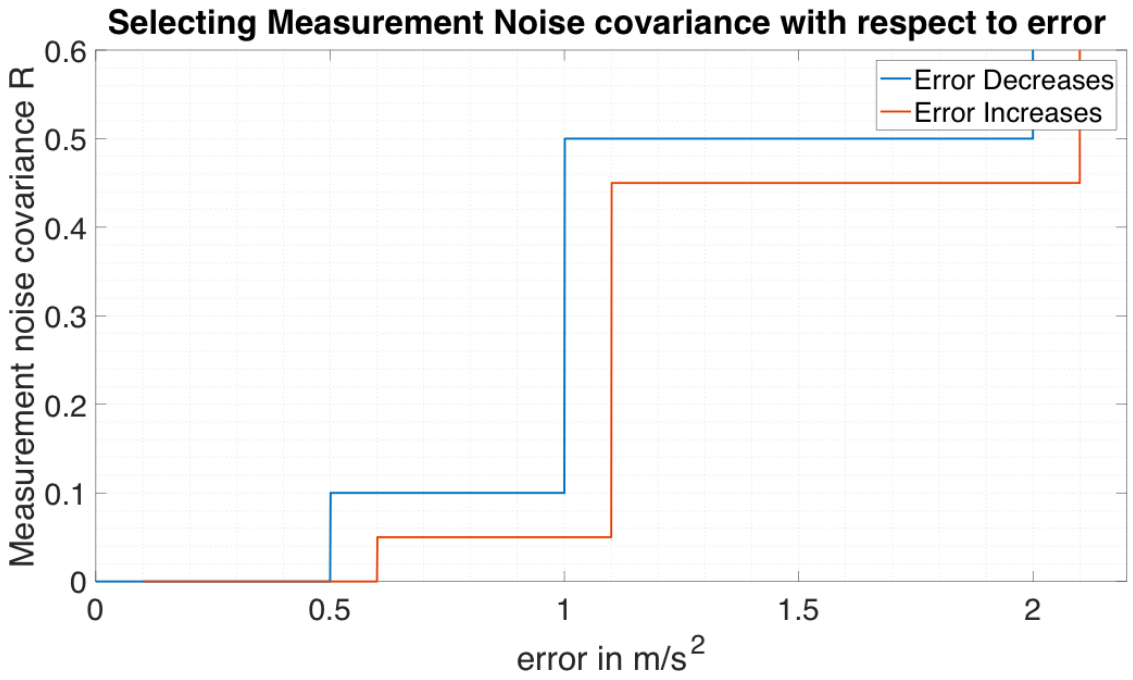

Figure 5.6. Selecting measurement noise covariance matrix for a given error value.

5.4.4. Simulated Results

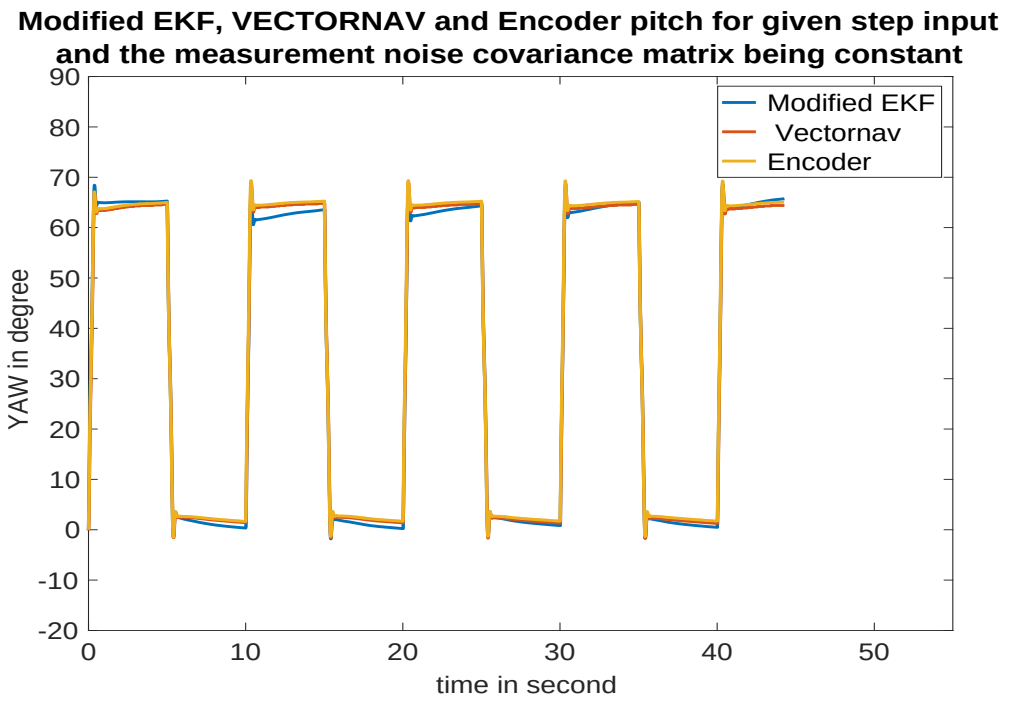

Figure 5.7. Yaw angle readings from modified EKF, VECTORNAV and Encoder when measurement noise is given a constant value.

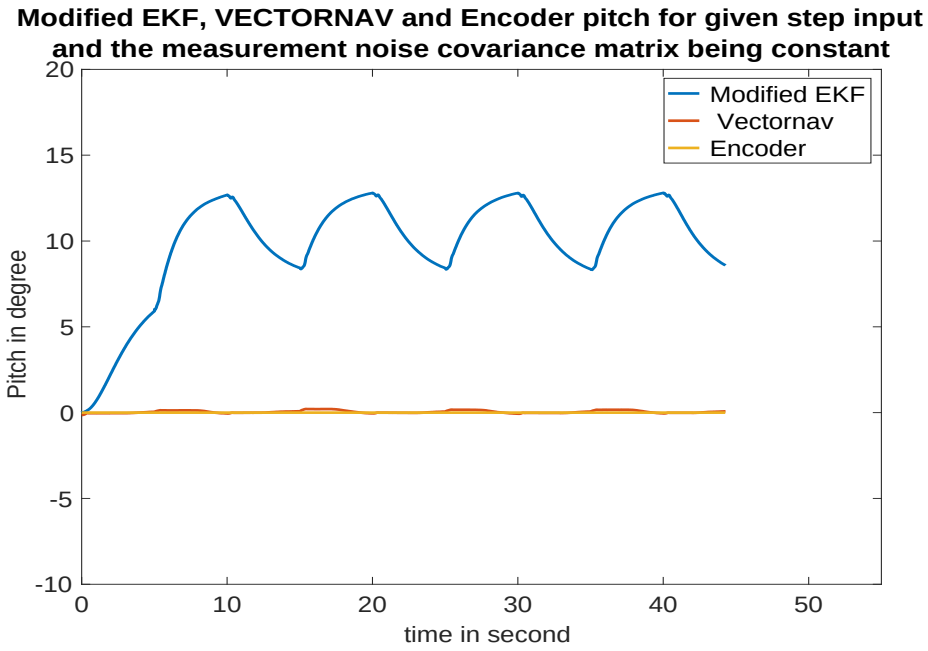

Figure 5.8. Pitch angle readings from modified EKF, VECTORNAV and Encoder when measurement noise is given a constant value.

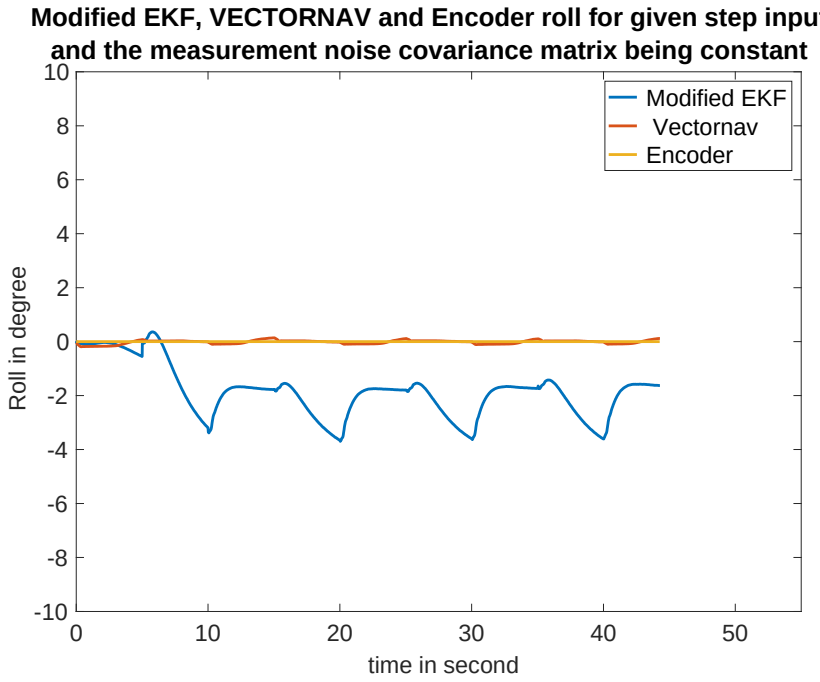

Figure 5.9. Roll angle readings from modified EKF, VECTORNAV and Encoder when measurement noise is given a constant value.

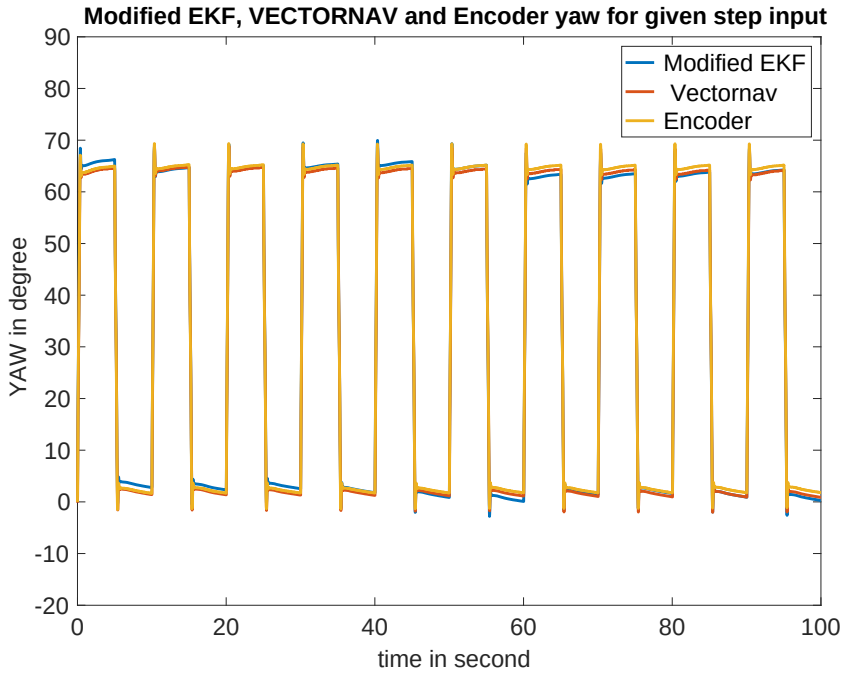

Figure 5.10. Yaw angle readings from modified EKF, VECTORNAV and Encoder when measurement noise is assigned a varying value after detecting the body acceleration.

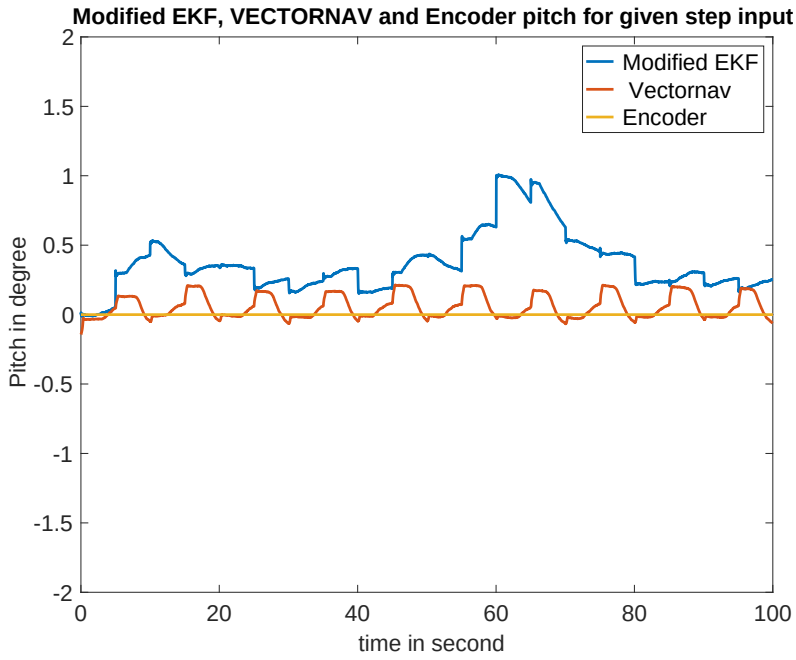

Figure 5.11. Pitch angle readings from modified EKF, VECTORNAV and Encoder when measurement noise is assigned a varying value after detecting the body acceleration.

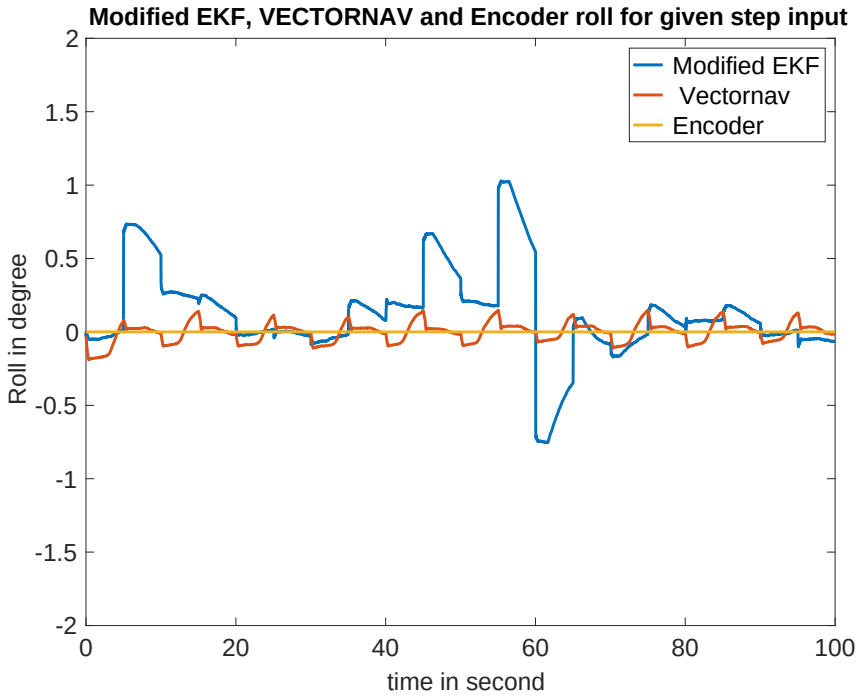

Figure 3.12. Roll angle readings from modified EKF, VECTORNAV and Encoder when measurement noise is assigned a varying value after detecting the body acceleration.

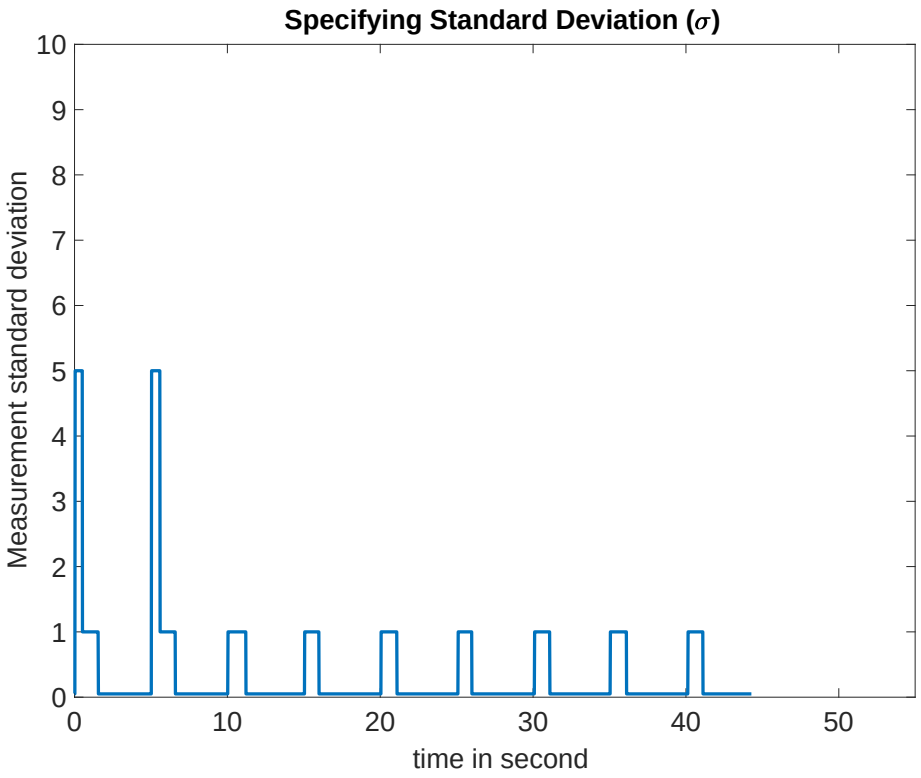

Figure 5.13. The variation of assigned measurement noise covariance when the IMU device is subjected to different body acceleration. The covariance is square of the standard deviation.

5.4.5. Discussion of simulated results

The modified extended Kalman filter algorithm described in this chapter is experimented by subjecting an IMU device to the body acceleration and collecting the readings from the gyroscope, the accelerometer and the magnetometer. In order to obtain body acceleration, an experiment on Torsional plant is conducted. In this experiment, the disk is rotated such that unidirectional step input of amplitude 67.5^0 and dwelling time of *5* seconds is achieved by using P-I controller. The procedure for this experiment is explained in Appendix A1and the extended Kalman filter MATLAB simulation is presented in Appendix A2. In this experiment the IMU device feels the body acceleration

when it moves from zero position to peak position or vice-versa. Also, the IMU device dwells for five second before each rotation.

In this work, the estimated results from the modified extended Kalman filter algorithm presented in this chapter is compared with the estimated results from the extended Kalman filter used by the VECTORNAV software. For simplicity, the modified extended Kalman explained in this work is named as modified EKF. The technical aspect for both cases is compared as follows:

1. In both cases, the raw data readings coming from the gyroscope, the accelerometer and the magnetometer are the same. The frequency of raw data is 50 Hz i.e. data are sampled at every 0.02 second in both cases.
2. The filtering technique to eliminate the effect of the body acceleration in the extended Kalman filter used by VECTORNAV software is not known to us. In modified EKF, the filtering technique is explained in Section (5.4. ii) and (5.4. iii)
3. In modified EKF, the gyro drift is minimized by assigning the process noise covariance matrix as given in the equation (5.3). The method used by the VECTORNAV software to overcome the gyro drift is unknown to us.
4. To validate the estimated results for both cases, the high resolution encoder is used. The high resolution encoder is an optical encoder that reads angular displacement.

If an IMU device is subjected to the body acceleration, there will be significant error in pitch and roll angle estimation. This can be shown in Figure (5.7), (5.8) and (5.9).

In the Figure (3.7), the yaw angle measured by the modified EKF, the VECTORNAV and the encoder is denoted by blue, red and yellow line. In the Figure (5.7), the estimated yaw angle by the VECTORNAV differs from the encoder by approximately 0.5^0 but the estimated yaw angle by modified EKF differs from the encoder by approximately 2^0 to 3^0. In the Figure (5.8), the estimated pitch angle by the VECTORNAV and the encoder differs by approximately 0.2^0 but the estimated pitch angle by modified EKF differs from the encoder by approximately 8^0 to 13^0. Similarly, in the Figure (5.9), the estimated yaw angle by the VECTORNAV and the encoder differs by approximately 0.1^0 but estimated pitch angle by modified EKF differs from encoder by approximately 2^0 to 4^0. In Figure (5.9), large errors of approximately 10^0 are obtained at time 4 to 5 second which is not expected. This may be caused due to the experimentation error. The above discussion made from Figure (5.7) to (5.9) is summarized in Table 5.1. From Table 5.1, it can be concluded that the IMU device under body acceleration will show significant error in pitch and roll angle estimation.

Table 5.1. Approximate error in yaw, pitch and roll angle when measurement covariance matrix is assigned a constant value.

Estimation Tool	Approximate error in degree		
	Yaw	Pitch	Roll
VECTORNAV	0.5	0.2	0.1
Modified EKF	2-3	8-13	2-4

To eliminate these errors, the body acceleration should be subtracted from the measured accelerations. The body acceleration can be determined by adding another device

such as GPS. The position determined from GPS can be utilized to get the body acceleration. This body acceleration can be fed into the measurement equation for its subtraction to nullify the effect of body acceleration. In this work, instead of adding another device, the effect of body acceleration is eliminated by assigning the large measurement noise covariance matrix as explained in Section (5.4. ii) and (5.4. iii). The Figure (5.13) shows the variation of measurement noise covariance matrix. The region where the measurement noise covariance matrix jumps to *1* is the region where the device is subjected to the body acceleration. In these regions, the assignment of high covariance will ignore the measurement readings and the estimation is done based on the process.

In the Figure (5.10), the estimated yaw angle by the VECTORNAV differs from the encoder by approximately 0.5^0 but the estimated yaw angle by modified EKF differs from the encoder by approximately 2^0 to 3^0. In the Figure (5.11), the estimated pitch angle by the VECTORNAV and the encoder differs by approximately 0.2^0 but the estimated pitch angle by modified EKF differs from the encoder by approximately 1^0. Similarly, in the Figure (5.9), the estimated yaw angle by the VECTORNAV and the encoder differs by approximately 0.1^0 but estimated pitch angle by modified EKF differs from encoder by approximately 1^0.

Table 5.2. Approximate error in yaw, pitch and roll angle when measurement covariance matrix is assigned a varying value.

Estimation Tool	Approximate error in degree		
	Yaw	Pitch	Roll
VECTORNAV	0.5	0.2	0.1
Modified EKF	2-3	1	1

5.5. Conclusion

In this chapter, the extended Kalman filter algorithm is modified by adding measurement biases as a process and the gyro drift is minimized by assigning process noise covariance as a function of current states, sampling time and biases. The poor performance of modified EKF is observed when the IMU device is under body acceleration. In order to increase the performance of modified EKF under body acceleration, the filter is designed. The filter assigns the varying values of measurement noise covariance matrix. The filter makes the judgement on reliance of measurement units based on the error from what it measures and reference measurement. The filter performed satisfactorily to eliminate the error caused by body acceleration. Comparing the estimated results of modified EKF and VECTORNAV, it can be concluded that the performance of VECTORNAV is better than Modified EKF.

CHAPTER VI

CONCLUSION AND RECOMMENDATIONS

6.1. Conclusion

This work starts with the understanding of the concept of online estimation of attitude and its importance in attitude control. It is very important to send the correct state information as a feedback in any control system that is expected to perform optimally. Since the measurement units that feed the data to control systems are affected by noise and biases, it is necessary to eliminate the state estimation errors caused by noise and biases. This can be done by using the proper estimation tool.

In this work, an extended Kalman filter is used as estimation tool for estimating the attitude. Before starting with this tool, it is necessary to design the process and measurement equation. In this work, the process equation consists of quaternion based states and the angular velocity information coming from the gyroscopes. The integration of process equation will determine the attitude of vehicle. However, these estimated attitudes will have drift which may grows in time. To eliminate the error due to drift, the states have to be updated. This update relies on measurement equation. The measurement equation consists of rotation matrix as a function of current quaternion states and the measurements from the accelerometer and the magnetometer. Date from gyroscope, accelerometer and the magnetometer are first calibrated before they are fed into Kalman filter estimation tool. The extended Kalman filter algorithm is coded in MATLAB and simulated results for two different cases are studied. In the first case, the performance of

the extended Kalman filter with constant process and measurement noise covariance matrix for steady state condition is studied. it is observed that the estimated states drift in time.. To overcome this problem, the extended Kalman filter is modified by adding biases from measurement units to process equation of the filter and assigning an additive process noise covariance as function of current states, sampling time ,and biases. The performance of the modified extended Kalman is obtained as desired. The drift in the estimated states are minimized.

In the second case, the performance of the modified extended Kalman filter in a case where significant body accelerations exist is studied. As expected, in this case the attitude estimation is inaccurate when constant measurement noise covariance matrix is assigned. However, after designing a filter that makes judgment on the reliability of measurements, the modified extended Kalman filter performed well with errors less than 3%.

Research efforts in the field of navigation and better performing IMU devices are growing rapidly. In our study, the estimation tool i.e. extended Kalman filter is designed with some modifications for estimation of orientations. The modified EKF performed well compared to VECTORNAV in steady state condition. However, under body acceleration the modified EKF did not perform as well as what VECTORNAV did because the first order Tayler series approximation used in the prediction step generated some error.

6.2. Recommendations

In this work, the online estimation of states using Extended Kalman filter consists the low pass filter to smooth the error between the measured quantities and reference quantities. The further work should be done to replace the low pass filter with new

technique that will eliminate the high cost calculations. Here, the body acceleration is avoided by assigning large covariance matrix. When large covariance matrix is assigned, the Kalman gain increases and the higher Kalman gain places more weights on process.

In the case of body acceleration for longer time, the estimation relies only on prediction step because the unreliability on the measurements will place more weights on process. This will cause propagation of biases which finally results inaccuracy in attitude estimation. It is suggested to measure the body acceleration and subtract it from the measured acceleration. This will help to reduce the error between measured acceleration and reference gravity vector. In the case of body acceleration caused due to the turning, the velocity and the instantaneous radius can be used to calculate the centripetal or body acceleration. The velocity and the instantaneous radius can be obtained from the positions and time data tracked by GPS unit. The compensation of body acceleration by aiding GPS unit is reliable if it is used in outside environment or free horizon i.e. not inside building, laboratory or large structure such has tunnel. Also, the accuracy of estimation depends on the sensitivity of the GPS unit.

For the best performance of the IMU device, it should be placed in the vibration free location. In actual practice it is not possible to get the vibration free environment. But, it should be made sure that there are not any significant vibrations that affect the IMU device readings. This can be done by using appropriate vibration analysis tool. In this work, the performance of the extended Kalman filter is tested in the lab environment which is prone to unknown vibrations. The IMU device is placed on the disk of the Torsional plant and any external vibrations on this system is not studied and eliminated in this work. It is assumed that external vibrations to the IMU device is negligible.

BIBLIOGRAPHY

[1] N. R. & N. D. N Dinesh Kumar, "On-board Navigation and Data Acquisition for High Dynamic Gps and Glonass Receiver," *Int. J. Electron. Commun. Instrum. Eng. Res. Dev.*, vol. 3, no. 2, pp. 123–136, 2013.

[2] O. J. Woodman, "An Introduction to Inertial Navigation," UCAM-CL-TR-696, 2007.

[3] M. . Burg, A., Meruani, A., Sandheinrich, B. and Wickmann, "MEMS Gyroscopes and their Applications," *clifton.mech.northwestern.edu*.

[4] T. K. Bhattacharyya and A. L. Roy, "MEMS Piezoresistive Accelerometers," 2014, pp. 19–34.

[5] D. Gebre-Egziabher, G. Elkaim, … J. P.-8th I. S., and undefined 2001, "A Non-linear, Two-step Estimation Algorithm for Calibrating Solid-state Strapdown Magnetometers," in *stanford.edu*, 2001.

[6] D. Simon, *Optimal state estimation : Kalman, H [infinity] and nonlinear approaches*. Wiley-Interscience, 2006.

[7] T. Xiao, L. Zhang, and S. Ma, Eds., *System Simulation and Scientific Computing*, vol. 326. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[8] S. S. Haykin, *Kalman Filtering and Neural Networks*. Wiley, 2001.

[9] M. Henriksson, "Estimation of Heading using Magnetometer and GPS," Uppsala Universitet, 2013.

[10] Y. Cheng and J. L. Crassidis, "Particle Filtering for Attitude Estimation Using a

Minimal Local-Error Representation," *J. Guid. Control. Dyn.*, vol. 33, no. 4, pp. 1305–1310, Jul. 2010.

[11] D. Abeywardena, S. Kodagoda, G. Dissanayake, and R. Munasinghe, "Improved State Estimation in Quadrotor MAVs: A Novel Drift-Free Velocity Estimator," *IEEE Robot. Autom. Mag.*, vol. 20, no. 4, pp. 32–39, Dec. 2013.

[12] M. B. Rhudy, Y. Gu, H. Chao, and J. N. Gross, "Unmanned Aerial Vehicle Navigation Using Wide-Field Optical Flow and Inertial Sensors," *J. Robot.*, vol. 2015, pp. 1–12, Oct. 2015.

[13] G. Liu, F. Wörgötter, and M. Irene, "The Square-root Unscented Information Filter for State Estimation and Sensor Fusion.," *fortknox.physik3.gwdg.de*, 2012.

[14] T. Zhang, Y. L.-C. and electronics in agriculture, and undefined 2017, "Attitude measure system based on extended Kalman filter for multi-rotors," *Elsevier*.

[15] J Diebel, "Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors," *Matrix 58, no. 15-16 1-35.*, 2006.

[16] E. Kraft, "A Quaternion-based Unscented Kalman Filter for Orientation Tracking," in *In Proceedings of the Sixth International Conference of Information Fusion, vol. 1*, , 2003, pp. 47–54.

[17] P. Singla, D. Mortari, … J. J. the A. 14th S., and undefined 2004, "How to Avoid Singularity When Using Euler Angles," in *In Proceedings of the AAS/AIAA 14th Space Flight Mechanics Meeting*, 2004, pp. 8–12.

[18] A. Janota, V. Šimák, D. Nemec, J. H.- Sensors, and undefined 2015, "Improving the Precision and Speed of Euler Angles Computation from Low-cost Rotation Sensor Data," *Sensors 15, no. 3, 7016-7039.*, 2015.

[19] N. Leonard, “Motion Control: Why Order Matters.” Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, pp. 1–9.

[20] P. R. Evans, “Biological Crystallography Rotations and Rotation Matrices,” *Acta Crystallogr. Sect. D*, vol. 57, pp. 1355–1359, 2001.

[21] E. G. Hemingway and O. M. O’Reilly, “Perspectives on Euler Angle Singularities, Gimbal Lock, and the Orthogonality of Applied Forces and Applied Moments,” *Multibody Syst. Dyn.*, pp. 1–26, 2017.

[22] W. R. Hamilton, “On Quaternions; or on a New System of Imaginaries in Algebra,” *London, Edinburgh, Dublin Philos. Mag. J. Sci.*, vol. 33, no. 219, pp. 58–60, Jul. 1848.

[23] Karsten G. and Yoon Z., “Introduction into Quaternions for Spacecraft Attitude Representation,” Berlin, Germany, 2012.

[24] E. B. Dam, M. Koch, and M. Lillholm, “Quaternions, Interpolation and Animation,” Denmark, 1998.

[25] D. Tedaldi, A. Pretto, E. M.-R. and Automation, and undefined 2014, “A Robust and Easy to Implement Method for IMU Calibration Without External Equipments,” in *Robotics and Automation (ICRA), 2014 IEEE International Conference on. IEEE*, 2014, pp. 3042–3049.

[26] “AN4508 Application Note Parameters and Calibration of a Low-g 3-axis accelerometer,” 2014.

[27] VectorNav Embedded Navigation Solutions, “VN-100 Hard and Soft Iron Calibration.”

[28] Christopher K, “Compensating for Tilt, Hard-Iron, and Soft-Iron Effects | Sensors

Magazine," *SENSORS | ONLINE* , 2009. [Online]. Available: https://www.sensorsmag.com/components/compensating-for-tilt-hard-iron-and-soft-iron-effects. [Accessed: 02-Aug-2018].

[29] VectorNav Embedded Navigation Solutions, "VectorNav Library." [Online]. Available: https://www.vectornav.com/support/library. [Accessed: 02-Aug-2018].

[30] Acutronic Carco Electronics, "3-Axis Motion Simulator Series BD357 - ACUTRONIC / US." [Online]. Available: http://www.acutronic.com [Accessed: 02-Aug-2018].

[31] N. Yadav, C. B.- Sensors, and 2014, "Accurate Orientation Estimation using AHRS under Conditions of Magnetic Distortion," *Sens. 14 20008-20024*, 2014.

[32] D. Rowell, "Analysis and Design of Feedback Control Systems," 2002.

[33] K. Fernando, "Covariance and Gramian Matrices in Control and Systems Theory.," The University of Sheffield, 1982.

[34] GA Terejanu, "Extended Kalman Filter Tutorial," *Department of Computer Science and Engineering, University at Buffalo*, 2008. [Online]. Available: http://homes.cs.washington.edu/~todorov/courses/cseP590/readings/tutorialEKF.pdf. [Accessed: 02-Aug-2018].

[35] P. Abbeel, A. Coates, M. Montemerlo, A. Y. Ng, and S. Thrun, "Discriminative Training of Kalman Filters."

[36] A. M. Sabatini, "Quaternion-Based Extended Kalman Filter for Determining Orientation by Inertial and Magnetic Sensing," *IEEE Trans. Biomed. Eng.*, vol. 53, no. 7, pp. 1346–1356, Jul. 2006.

APPENDICES

APPENDIX A: PROCEDURE TO GET STEP RESPONSE

The performance of the extended Kalman filter under a body acceleration is studied in this thesis. The body acceleration is generated on the Torsional Plant located in Parker Motion Control lab. The experimental setup consists of three subsystems as shown in Figure A.1, the first subsystem is an electromechanical system that consists of a torsion mechanism, an actuator, and three rotation sensors. The torsion mechanism can be transformed into variety of dynamic configurations by adding or removing disks as well as weights placed on disks. The second subsystem is the controller unit that has the Digital Signal Processor (DSP) whose functions are to support downloading various algorithms, specifying desired motion trajectories, selecting data to be acquired and specifying how the data should be plotted. The third subsystem is the executive program which runs on PC and supports controller specification, trajectory definition, data acquisition and plotting.

In this work, the IMU devices is attached on the lower disk as shown in Figure A.2. The encoder readings provide the angular displacement of the disk. Reset the IMU device so that the yaw, pitch and roll angle reading comes to zero at the start position. The program that execute the control motion and the VECTORNAV program that runs IMU data acquisition should be executed at the same time in order to synchronize the data.

The procedure to achieve the step response are as follows:

1. Put the mechanism in the configuration shown in Figure A.2.

2. Place the IMU device firmly on the disk at a center distance of 8 cm approximately. The farther the device from the center, the more body acceleration is experienced by IMU device.
3. Open ECP Control Executive from the desktop screen and turn the controller "ON" by pressing the black switch on the control box.
4. Go to Command Menu

   a. Enter trajectory and select step and click setup.

   b. Select close loop step and input size 3000 counts, a duration of 5000 ms and 5 repetitions. One count is equivalent to 0.0225^0. Click OK. (Note: For sinusoidal response used in Section 3.4, input trajectory is sinusoidal with amplitude 22.5^0, frequency 1 Hz and five repitions)

   c. Exit to the background screen by selecting OK.
5. Go to Set-up Menu

   a. Enter the control algorithm box and select DISCRETE time. Set T_s= 0.004420 seconds. This means the sampling period used by the controller will be 442 micro-second.

   b. Select PI with velocity feedback and click setup algorithm.

   c. Enter the proportional gain kp as 0.01 and the integral gain k_i as 0.0001. These numbers are obtained by hit and trial method. Click OK.

   d. Click implement algorithm and click OK.
6. Go to Command Menu

   a. Select execute.

   b. Select Normal Data Sampling from Execute Trajectory Dialogue Box.

c. Click RUN.

d. Select OK after data is uploaded.

7. Go to Data Menu to export the raw data from the encoder. The raw data from the IMU device is exported from VECTORNAV software.

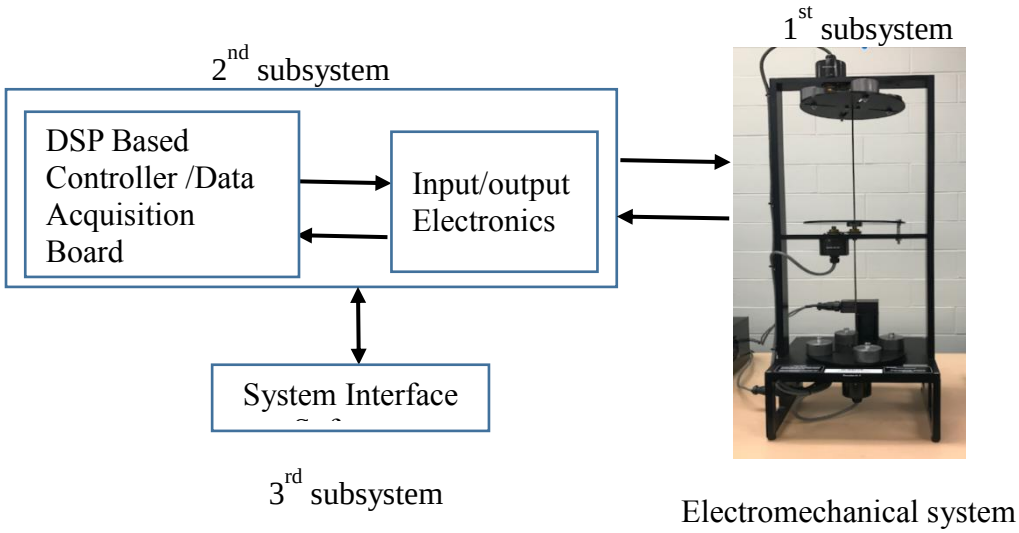

Figure A.1. Interaction of different subsystems in Torsional plant

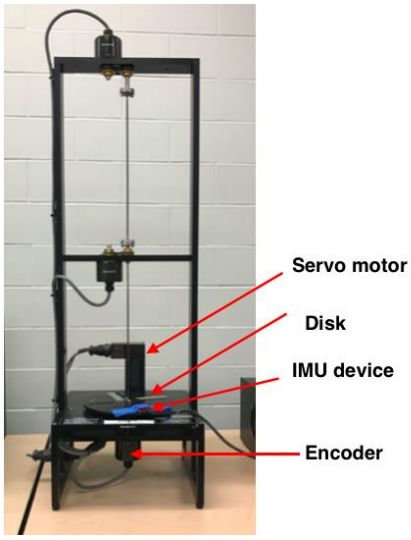

Figure A.2. Experimental setup that shows torsional plant and IMU device

APPENDIX B: MATLAB CODE FOR IMPLIMENTING THE EKF

The following is the MATLAB CODE written to implement the extended Kalman filter algorithm when IMU device is subjected to body acceleration.

```
%##########################################
%Importing Data from IMU device and Encoder
%##########################################


% Importing IMU device data as:
% Acceleration (ax, ay, az),
% Magnetic (ax, ay, az),
% Angular velocity (wx, wy, wz)
% Estimated Attitude from VECTORNAV (Yaw (Y), Pitch (P,
Roll(R))
% Encoder Reading (time and encoder position in count,
%1 count =0.0225 degree Encoder readings are in given in
counts.
load raw_Data.mat % raw_Data.mat contains data from IMU
device and Encoder


% Gravity vector
g=-9.81;
% For Magnetic Vector
hx=0.197;
```

```
hy=0.0289;
hz=1.50; % Magnetic reading in z-direction is constant with
value of 1.5.
%Sampling time for IMU readings
Ts=0.02;
%##########################################################
###
% Assigning measurement Covariance matrix by constructing
steps
%##########################################################
###
ss=0.001; %error step
E1=0.01; % first step limit
E2=0.1;  % second step limit
E3=1;    % third step limit
E4=5;    % fourth step limit
E5=10;   % fifth step limit
E6=100;  % sixth step limit

%  error of each step
e1=0:ss:E1;
e2=ss+E1:ss:E2;
e3=ss+E2:ss:E3;
e4=ss+E3:ss:E4;
```

```
e5=ss+E4:ss:E5;
e6=ss+E5:ss:E6;
% Creating x-axis
error1=horzcat(e1,e2,e3,e4,e5,e6);

% Measurement noise covariance for each error set
% r1 corresponds to covariance of error set e1
r1=0.0005*ones(1,length(e1));
r2=0.05*ones(1,length(e2));
r3=1*ones(1,length(e3));
r4=5*ones(1,length(e4));
r5=50*ones(1,length(e5));
r6=100*ones(1,length(e6));
 % Creating y-axis
R1=1*horzcat(r1,r2,r3,r4,r5,r6);
 % If error increases cho0se from R1
% If error decreases choose from R2
shift1=0.05*R1;
shift2=0.05*error1;
R2=R1-shift1;
error2=error1+shift2;

% Determining the error between measured and reference
quantity
```

```
l=length(ax); % length of data
y(1)=0.1; % initializing filter, y refers to filtered data
a=0.95; % filter gain
for k=1:l
norm_gravity_vector(k)=9.81; % reference vector
% norm of acceleration measurement
norm_acc(k)=abs(sqrt((ax(k))^2+(ay(k))^2+(az(k))^2));
% difference between norm of acceration and norm of reference
vector
error(k)=abs(norm_acc(k)-norm_gravity_vector(k));
% Filtered Error
y(k+1)=a*y(k)+(1-a)*error(k);
ee=round(y,3);
% Error sample
eee=round(ee/ss);
end
% Assigning covariance matrix that corresopnds to respective
error
% a_amnstd refers to standard deviation of measurement noise
for k=1:l-1;
          if ee(k+1)>= ee(k)
            a_mnstd=R2(eee);
        else
            a_mnstd=R1(eee);
```

```
        end
    end
% Initializing the EKF
% Initial measurement biases
aK1K1= [0;0;0]; % accelerometer bias
mK1K1= [0;0;0]; % magnetometer bias
% Initial error Covariance P
PK1K1= 0*eye (10);
%Initial State
QK1K1= [ 1   0   0   0]';
% Combining all states
xK1K1= [QK1K1; aK1K1; mK1K1];
% Creating zero and unit matrix
I=eye (3);
zero=zeros (3);
zero1=zeros (3,4);
zero2=zeros (4,3);
% Process noise standard deviation
sigma_g=0.0007;
% Measurement noise standard deviation for the accelerometer
sigma_a=0.04;
% Measurement noise standard deviation for the magnetometer
sigma_m=0.004;
for i=1:l-1
```

```
    % State Function
    phi=[0 -wx(i) -wy(i) -wz(i); wx(i) 0 -wz(i) wy(i); wy(i)
wz(i) 0 -wx(i);wz(i) -wy(i) wx(i) 0];
    % State Transition Matrix
    omega=[expm(1/2*phi*Ts)  zero1'  zero1'  ;zero1  I  zero;
zero1 zero I];

%     ###################                PREDICTION     STEP
########################
    xKK1=omega*xK1K1;
%  Quaternion states after prediction step
    q0=xKK1(1);
    q1=xKK1(2);
    q2=xKK1(3);
    q3=xKK1(4);
    QKK1=[q0 q1 q2 q3];

%Process Noise Covariance
    sum_g=sigma_g^2*I;
    sum_a= Ts*sigma_a^2*I;
    sum_m=Ts*sigma_m^2*I;
    e_k= [q1; q2; q3];
    sum_k= [diag(e_k)+q0*I;-e_k'];
% Process noise covariance matrix
```

```
    Qk=  [(Ts/2)^2*sum_k*sum_g*sum_k'  zero2  zero2;  zero2'
sum_a zero; zero2' zero sum_m];
    PKK1=omega*PK1K1*omega+Qk;
% Rotation Matrix
    C=[q0^2+q1^2-q2^2-q3^2     2*(q1*q2+q0*q3)     2*(q1*q3-
q0*q2);...
    2*(q1*q2-q0*q3) q0^2-q1^2+q2^2-q3^2 2*(q2*q3+q0*q1);...
    2*(q1*q3+q0*q2) 2*(q2*q3-q0*q1) q0^2-q1^2-q2^2+q3^2];

% Measurement Equation
    h=[C 0*eye(3); 0*eye(3) C]*[0 0 g hx hy hz]';

% Measurement Covariance Rk
    MNstd_a=[a_mnstd(i) a_mnstd(i) a_mnstd(i)];
    MNstd_m=[a_mnstd(i) a_mnstd(i) a_mnstd(i)];
    Rk= (diag([MNstd_a MNstd_m]))^2;

% LINEARIZATION
HHK= [            -2*g*QKK1(3)                    2*g*QKK1(4)
-2*g*QKK1(1)                          2*g*QKK1(2) ;
                    2*g*QKK1(2)
2*g*QKK1(1)                                       2*g*QKK1(4)
2*g*QKK1(3) ;
                    2*g*QKK1(1)                               -
```

```
2*g*QKK1(2)                                          -2*g*QKK1(3)
2*g*QKK1(4) ;
 2*hx*QKK1(1) + 2*hy*QKK1(4) - 2*hz*QKK1(3)  2*hx*QKK1(2) +
2*hy*QKK1(3) + 2*hz*QKK1(4)  2*hy*QKK1(2) - 2*hx*QKK1(3) -
2*hz*QKK1(1)  2*hy*QKK1(1) - 2*hx*QKK1(4) + 2*hz*QKK1(2) ;
 2*hy*QKK1(1) - 2*hx*QKK1(4) + 2*hz*QKK1(2)  2*hx*QKK1(3) -
2*hy*QKK1(2) + 2*hz*QKK1(1)  2*hx*QKK1(2) + 2*hy*QKK1(3) +
2*hz*QKK1(4)  2*hz*QKK1(3) - 2*hy*QKK1(4) - 2*hx*QKK1(1) ;
 2*hx*QKK1(3) - 2*hy*QKK1(2) + 2*hz*QKK1(1)  2*hx*QKK1(4) -
2*hy*QKK1(1) - 2*hz*QKK1(2)  2*hx*QKK1(1) + 2*hy*QKK1(4) -
2*hz*QKK1(3)  2*hx*QKK1(2) + 2*hy*QKK1(3) + 2*hz*QKK1(4)] ;
HK= [HHK zeros(6,6)];
% KALMAN GAIN
    KK=PKK1*HK'/(HK*PKK1*HK'+Rk);

%######################                    UPDATING     STEP
#########################
xKK=xKK1+ KK*([ax(i);ay(i);az(i);mx(i);my(i);mz(i)]-h);
PKK=(eye(10)-KK*HK)*PKK1;
% Normalization
denu= xKK(1)^2+xKK(2)^2+xKK(3)^2+xKK(4)^2;
QKKn=[xKK(1)/sqrt(denu);xKK(2)/sqrt(denu);xKK(3)/sqrt(denu)
;xKK(4)/sqrt(denu)];
%Saving estimated states in Memory
```

```
q0(i)=QKKn(1);
q1(i)=QKKn(2);
q2(i)=QKKn(3);
q3(i)=QKKn(4);
s(i).q=QKKn;
%INTITIALIZING QUATERNIONS FOR NEXT INTERATION
xK1K1=[QKKn;xKK(5);xKK(6);xKK(7);xKK(8);xKK(9);xKK(10)];
PK1K1=PKK;
end
 for j=1:length(wx)-1
%Separation of Estimated quaternion terms obtained from above
EKF algorithm
     q0(j)=s(j).q(1);
     q1(j)=s(j).q(2);
     q2(j)=s(j).q(3);
     q3(j)=s(j).q(4);
 % YAW,PITCH and ROLL calculation

yaw11(j)=180/pi*atan2(2*q1(j)*q2(j)+2*q0(j)*q3(j),q0(j)^2+q
1(j)^2-   q2(j)^2-q3(j)^2);
 pitch1(j)=180/pi*asin(2*q1(j)*q3(j)-2*q0(j)*q2(j));
 roll1(j)=180/pi*atan2(2*q2(j)*q3(j)+2*q0(j)*q1(j),q3(j)^2-
q2(j)^2- q1(j)^2+q0(j)^2);
 end
```

```
%############   Plotting Attitudes #################

l1=11296;
t_encoder=A(1:l1,2);

Y_encoder=A(1:l1,4);

l2=5000;

t=0:Ts:l2*Ts-2*Ts;

l=length(t)

figure(1)

plot(t,Y1(:,1:l),t,Y(:,1:l),t_encoder,0.0225*Y_encoder,'lin

ewidth',3)

xlim([0 55])

ylim([-20 90])

title('\fontsize{25}  Modified EKF, VECTORNAV and Encoder

yaw for given step input')

xlabel(' \fontsize{25} time in second')

ylabel('\fontsize{25} YAW in degree')

legend( '\fontsize{25} Modified EKF ',' \fontsize{25}

Vectornav', '\fontsize{25} Encoder')

set(gca,'fontsize',25)

grid minor


figure(2)

plot(t,pitch1(:,1:l),t,P(:,1:l),t_encoder,0*Y_encoder,'line

width',3)
```

```
xlim([0 55])
ylim([-2 2])
title('\fontsize{25}  Modified EKF, VECTORNAV and Encoder
pitch for given step input')
xlabel(' \fontsize{25} time in second')
ylabel('\fontsize{25} Pitch in degree')
legend( '\fontsize{25} Modified EKF ',' \fontsize{25}
Vectornav', '\fontsize{25} Encoder')
set(gca,'fontsize',25)
grid minor

figure(3)
plot(t,roll1(:,1:l),t,R(:,1:l),t_encoder,0*Y_encoder,'linew
idth',3)
xlim([0 55])
ylim([-10 10])
title('\fontsize{25}  Modified EKF, VECTORNAV and Encoder
roll for given step input')
xlabel(' \fontsize{25} time in second')
ylabel('\fontsize{25} Roll in degree')
legend( '\fontsize{25} Modified EKF ',' \fontsize{25}
Vectornav', '\fontsize{25} Encoder')
set(gca,'fontsize',25)
grid minor
```

```
 figure(4)
 plot(t,a_mnstd(1:l),'linewidth',3)
 xlim([0 55])
ylim([0 10])
 title('\fontsize{25}  Specifying Standard Deviation
(\sigma)')
xlabel(' \fontsize{25} time in second')
ylabel('\fontsize{25} Measurement standard deviation')
 set(gca,'fontsize',25)
grid minor
```