MEMS SENSOR BASED UNDERWATER AHRS (ATTITUDE AND HEADING REFERENCE SYSTEM) AIDED BY COMPASS AND PRESSURE SENSOR

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MEHMET ERÇİN ÖZGENECİ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2012

Approval of the thesis:

MEMS SENSOR BASED UNDERWATER AHRS (ATTITUDE AND HEADING REFERENCE SYSTEM) AIDED BY COMPASS AND PRESSURE SENSOR

submitted by **MEHMET ERÇİN ÖZGENECİ** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. İsmet Erkmen
Head of Department, **Electrical and Electronics Engineering** _____

Prof. Dr. Tolga Çiloğlu
Supervisor, **Electrical and Electronics Engineering Dept., METU** _____

Assoc. Prof. Dr. Çağatay Candan
Co-supervisor, **Electrical and Electronics Engineering Dept., METU** _____

Examining Committee Members:

Prof. Dr. Mübeccel Demirekler
Electrical and Electronics Engineering Dept., METU _____

Prof. Dr. Tolga Çiloğlu
Electrical and Electronics Engineering Dept., METU _____

Assoc. Prof. Dr. Çağatay Candan
Electrical and Electronics Engineering Dept., METU _____

Assist. Prof. Dr. Umut Orguner
Electrical and Electronics Engineering Dept., METU _____

Erdinç Oğuz, M.Sc.
Manager, ASELSAN _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: MEHMET ERÇİN ÖZGENECİ

Signature :

ABSTRACT

MEMS SENSOR BASED UNDERWATER AHRS (ATTITUDE AND HEADING REFERENCE SYSTEM) AIDED BY COMPASS AND PRESSURE SENSOR

Özgeneci, Mehmet Erçin

M.Sc., Department of Electrical and Electronics Engineering

Supervisor : Prof. Dr. Tolga Çiloğlu

Co-Supervisor : Assoc. Prof. Dr. Çağatay Candan

September 2012, 84 pages

Attitude and Heading angles are crucial parameters for navigation. Conventional navigation methods mostly uses IMU and GPS devices to calculate these angles. MEMS technology offers small sized, low cost IMU sensors with moderate performance. However, GPS cannot be used in underwater. Therefore, different aiding sensors are used in underwater vehicles in order to increase the accuracy. As the accuracy of devices increases, the cost of these devices also increases. In this thesis, rather than using GPS and high quality IMU sensors, low cost MEMS IMU sensor is used together with a magnetometer and a pressure sensor as aiding sensors. Considering the IMU error model and motion dynamics, two systems are designed and simulated using real data. The results seem to be satisfactory and using pressure sensor as an aiding sensor improves the attitude angles estimation.

Keywords: AHRS, MEMS IMU, underwater navigation, pressure sensor

ÖZ

PUSULA VE BASINÇ SENSÖRÜ YARDIMIYLA MEMS SENSÖRLERİ İLE SU ALTI DURUM VE YÖNELİM SİSTEMİ

Özgeneci, Mehmet Erçin
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Tolga Çiloğlu
Ortak Tez Yöneticisi : Doç. Dr. Çağatay Candan

Eylül 2012, 84 sayfa

Durum ve yönelim açıları navigasyon sistemleri için önemli parametrelerdir. Geleneksel navigasyon sistemleri, ataletsel ölçü birimi ve GPS kullanarak bu açıları hesaplar. Gelişen MEMS teknolojisi küçük boyutta, düşük maliyetli ve makul performansa sahip ataletsel ölçü birimleri sunmaktadır. Fakat, GPS cihazları su altında kullanılamamaktadır. Bu sebeple, su altı cihazlarında hassasiyeti ve doğruluğu arttırmak için farklı yardımcı sensörler kullanılmaktadır. Bu sensörlerin hassasiyeti arttıkça, maliyeti de artış göstermektedir. Bu çalışmada, GPS ve kaliteli ataletsel ölçü birimi kullanmak yerine, basınç sensörü ve manyetometre ile birlikte düşük fiyatlı MEMS ataletsel ölçü birimi kullanılmıştır. Ataletsel ölçü biriminin hata modeli ve hareket dinamikleri düşünülerek, iki farklı sistem tasarlanarak, gerçek verilerle simule edilmiştir. Sonuçların oldukça tatmin edici olduğu gözlemlenmiş olup, basınç sensörünün yardımcı sensör olarak kullanılması durum açılarının hesaplanmasında iyileştirme sağlamıştır.

Anahtar Kelimeler: Ataletsel ölçü birimi (AÖB), su altı navigasyon, basınç sensörü

To my family and and my dear Zeren

ACKNOWLEDGMENTS

I would like to express my sincere thanks to my supervisor Prof. Dr. Tolga Çiloğlu for his guidance, encouragements, criticism and insight throughout the research.

I would also like to thank my co-advisor Assoc. Prof. Dr. Çağatay Candan for his valuable ideas throughout this study.

Thanks to the examining committee members Prof. Dr. Mübeccel Demirekler, Assist. Prof. Dr. Umut Orguner and Erdinç Oğuz for evaluating my work.

I am grateful to The Scientific and Technological Research Council of Turkey (TÜBİTAK) for their financial support during my graduate study.

Deepest thanks to my family for their love, trust and every kind of support not only throughout my thesis but also throughout my life.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Inertial Navigation Systems (INS) are one of the most critical parts of moving vehicles and Attitude and Heading Reference System (AHRS) is a common part of INS. INS systems are first developed for navigating rockets during World War II and then its usage area spread out to marine, aerospace and land systems [8]. Direction of the vehicle is the basic information for navigation, because position depends on the movement direction of the vehicle. So attitude and heading angles must be determined with high accuracy. INS systems are mainly composed of gyroscopes and accelerometers which altogether are named as Inertial Measurement Unit (IMU). Attitude and Heading angles are also calculated using these sensors but aiding sensors are required in order to reduce the error.

Magnetometer can be interpreted as the digital version of a compass which shows the direction according to the magnetic north. Magnetometers are widely used in AHRS systems for heading calculation, but the materials around and changing environment can interfere magnetometer outputs easily. Therefore, magnetometer must be calibrated properly before using it for navigation purposes. Presure sensor is also one of the most generic sensors of the underwater vehicle. The depth of the water can be easily measured by a pressure sensor. It is prefered in most of the underwater vehicles because of its convenience.

IMU outputs suffer from different error sources which cause drift over time when their outputs are integrated. Thus, using only IMU sensor and mechanization equation formed by Newton's Law of Motion results as a drift free INS [9]. Aiding sensor is a generic solution to overcome the drift problem. Nevertheless, aiding sensors also have some deficiencies for certain type of motions which cause large amount of errors in the output. Because of these reasons, AHRS system must be designed by considering all of these problems.

1.1 Scope of Thesis

MEMS technology offers low cost IMU sensors with small sizes but these sensors include serious errors which makes them unsuitable for stand-alone operation. Therefore, MEMS IMU sensor needs to be integrated with other aiding sensors like magnetometer and Global Positioning System (GPS). However, the challenges of working in an underwater environment restrict the usage of certain sensors such as GPS. To overcome this deficiency, different methods are proposed and implemented. In the scope of this thesis, magnetometer and pressure sensor are used as aiding sensors to determine the attitude and heading angles of an underwater vehicle by using Linearized Error State and Extended Kalman Filtering algorithms. In order to measure the performance of the system, field data is used in simulations.

1.2 Outline of Thesis

Chapter 2 presents fundamentals of intertial navigation systems. Reference frames definitions and frame rotation matrix are given in this chapter. IMU sensors and their error models are introduced. In the last sections of Chapter 2, most commonly used navigation equations and their error models are derived.

Chapter 3 provides information about aiding sensors. Magnetometer is explained in detail because of its complexity. Possible error sources and calibration methods related to these errors are given in this chapter. At the end of the chapter, the usage of pressure sensor is desribed.

Chapter 4 expresses the background information about Kalman Filter. Linear and nonlinear state space equations are presented. Since the systems are implemented in digital environment, discretization of the state space equations are given. Finally, two different Kalman filtering methods are discussed in this chapter: Discrete Kalman Filter and Discrete Extended Kalman Filter. The algorithms for these two different implementations are given in related sections.

Chapter 5 focuses on the newly designed system in this thesis work. The results of Allan Variance algorithm to measure the error parameters of IMU sensors are given in the first section. Magnetometer calibration and tilt angle calculation using accelerometer are presented.

Proposed adaptive system is described in detail and the system model is represented by mechanization equations. State space model is constituted for the Kalman Filter. Then entire system structure of two different implementations are presented by figures. In the last section of this chapter, simulation results are prensented.

Chapter 6 discusses the simulations results and provides a conclusion about this thesis study. Future work related to this work are stated in the last section of this chapter.

CHAPTER 2

INERTIAL NAVIGATION SYSTEMS

This chapter provides the background information about inertial navigation systems. Basic principles, fundamental equations and crucial points are also clarified in this chapter. After making a brief introduction to inertial navigation systems, sensor units for the system are explained. Their error models are considered together with the identification of the stochastic error sources. Last part of this chapter includes the coordinate frame definitions and dynamic mechanization equations.

2.1 Introduction to Inertial Navigation Systems

Navigation is the most important concept of autonomous vehicle systems since they are not operated remotely by a human. Any complication in the navigation system could seriously affect other systems of the vehicle. The main information produced by a navigation system are position, velocity and attitude of the system. Inertial navigation is the most common type of navigation techniques for autonomous vehicles and Attitude and Heading Referance System (AHRS) is the most critical part of an Inertial Navigation System (INS).

Basic principles of INS are based on the laws of mechanics which are introduced by Isaac Newton. The fundamental law tells us that applied force yields an acceleration of the body. So, given the ability of measuring acceleration by an accelerometer, the change in movement of the vehicle can be detected. By a simple integration of measured acceleration with respect to time, velocity change can be calculated. Rotational motion of the vehicle can be sensed by gyroscopes so that the direction of the acceleration could be determined. Therefore, the movement of the vehicle can be observed for navigation purposes. Only an accelerometer and

a gyroscope are not sufficient (especially MEMS ones) for navigation due to the noise and error characteristics of the sensors [10]. Integration will sum up these errors and yields growing errors in the output of the navigation system. So additional aiding sensors are required for navigation systems.

There are two types of INS system configurations with different performances. The first one is named as gimbaled systems. In that configuration, the accelerometers and gyroscopes are mounted on a gimballed platform which is always kept aligned with the navigation frame. This configuration provides accurate navigation data but they are not desirable for most applications because of the complex mechanical structure and high cost. Also, maintenance of gimbaled system must be done in a clean room since it requires very sensitive calibration. The second configuration is called strapdown systems. The accelerometers and gyroscopes are attached rigidly to the vehicle body. This time, software solution is used in order to keep track of IMU's orientation. This method is more practical than gimballed systems since it reduces size, cost, power consumption and mechanical complexity [11].

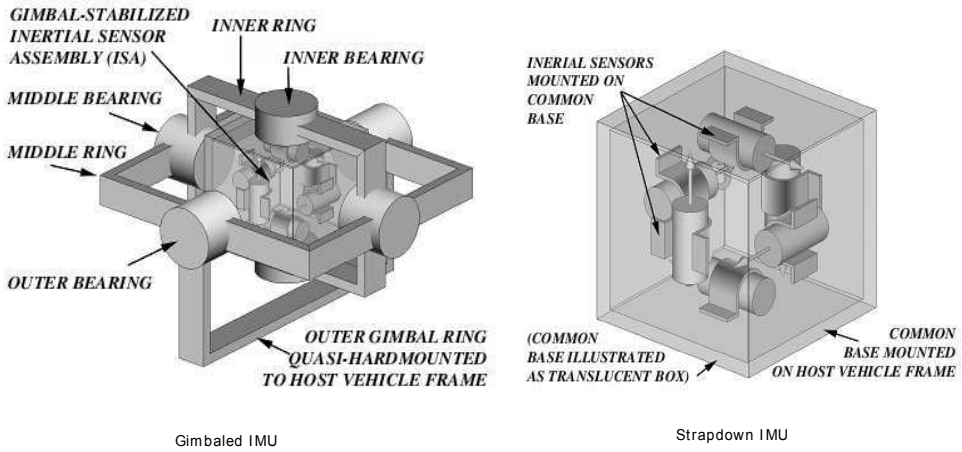

Figure 2.1: Gimbaled and strapdown inertial measurement units (taken from [2])

In this work, a strapdown INS is implemented considering the advantages which are mentinoned above. Our major interest is the effect of aiding sensor to the AHRS part of the INS system. Therefore, because of the proposed aiding sensors that are used in this thesis, INS system is implemented instead of implementing just AHRS system.

2.2 Reference Frames

Before going any further, it is important to define the reference frames, since mathematical representation of INS systems is based on reference frames. Later on this chapter, the mechnanization equations are derived considering these reference frames. All reference frames considered in this study form an orthogonal right handed basis set. Coordinate frames used throughout this thesis are explained in the subsequent sections.

2.2.1 Inertial Frame

It is hard to define perfect inertial frame, but an approximation of it is sufficient for the most of the navigation systems. Origin of the inertial frame is defined as Earth's center of mass and oriented with respect to fixed stars. Inertial frame is assumed not to accelerate and not to rotate with respect to universe [3]. Inertial frame is denoted by a symbol *i*.

2.2.2 Earth Frame

Earth frame has its origin at the center of Earth and all axes are fixed with respect to Earth's surface. Its x-axis points to Greenwich meridian, z-axis is the same as the Earth rotation axis which goes through Earth's polar axis and y-axis satisfies right handed orthogonal plane so that y-axis is defined along equatorial plane [12].

Unlike the inertial frame, Earth frame rotates with respect to inertial frame with a constant angular rate. This angular rate is calculated in Equation 2.1 [4].

$$w_{ie} \approx \left(\frac{1 + 365.25\ cycle}{(365.25)(24\ hour)}\right)\left(\frac{2\pi\ rad/cycle}{3600\ sec/hour}\right) = 7.292115 \times 10^{-5}\ \frac{rad}{sec} \tag{2.1}$$

In Equation 2.1, Earth's angular rate about its own axis relative to inertial frame is calculated and denoted by w_{ie}. Figure 2.2 shows the axes of the Earth frame.

2.2.3 Navigation Frame

Navigation frame is actually a local frame defined in the operation position of the vehicle. Its origin is at the location of vehicle (navigation system of the vehicle) and its axes are aligned

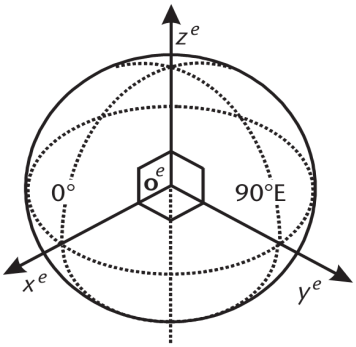

Figure 2.2: Axes of the Earth frame (taken from [3])

with north, east and down (local vertical down) direction. Because of the axes alignment, it also called as NED frame. Navigation frame axes may rotate as vehicle moves on Earth surface due to Earth's ellipsoidal shape . This rotation is named as transport rate. Since transport rate is angular rotation of navigation frame with respect to Earth frame, it is denoted by w_{en} in this study. Figure 2.3 clearly depicts navigation frame and its axes.

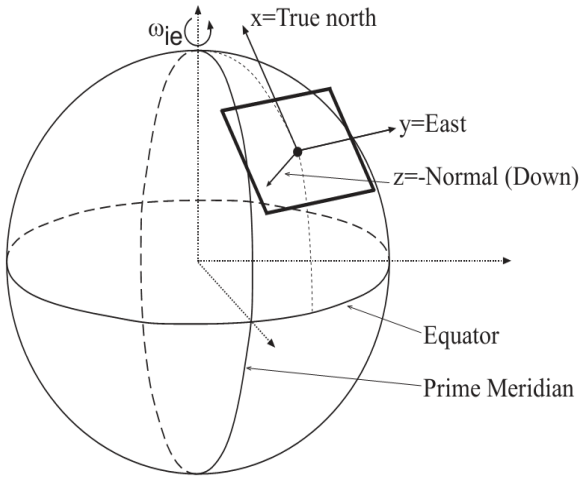

Figure 2.3: Axes of the navigation frame (taken from [4])

2.2.4 Body Frame

Body frame is rigidly attached to the vehicle and origin of the frame is the center of navigation system on the vehicle. Body frame axes are aligned with roll, pitch and yaw axes. In other words, x-axis is forward along the movement direction of the vehicle, z-axis is directed downward and y-axis is directed towards to right side of the vehicle so that orthogonal right handed rule is satisfied. Axes of body frame is shown in Figure 2.4. Body frame is denoted by symbol *b*.

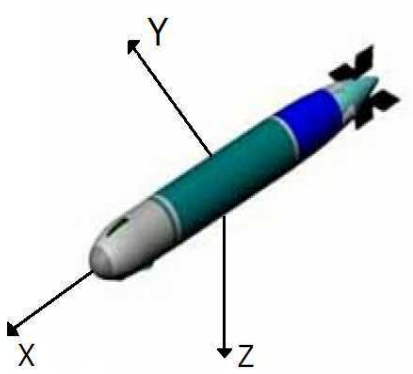

Figure 2.4: Axes of the body frame

2.3 Frame Transformation Matrix

In navigation systems, it is essential to work in different frames but the solution must be in a well defined frame. For strapdown navigation systems, the measurement of inertial sensors are defined in body frame which is not a global frame to use at the output of the whole system. So there has to be a transformation between two different frames. This transformation is perfomed by a matrix which is called Direction Cosine Matrix (DCM). In this study, the following notation is adapted when transforming one vector from one frame to another frame.

$$x^{to} = R^{to}_{from}\, x^{from} \tag{2.2}$$

Throughout this work, Euler angles are used to identify the attitude and heading of the vehicle. The Euler angles, roll(ϕ), pitch(θ), and yaw(ψ), are depicted in Figure 2.5.

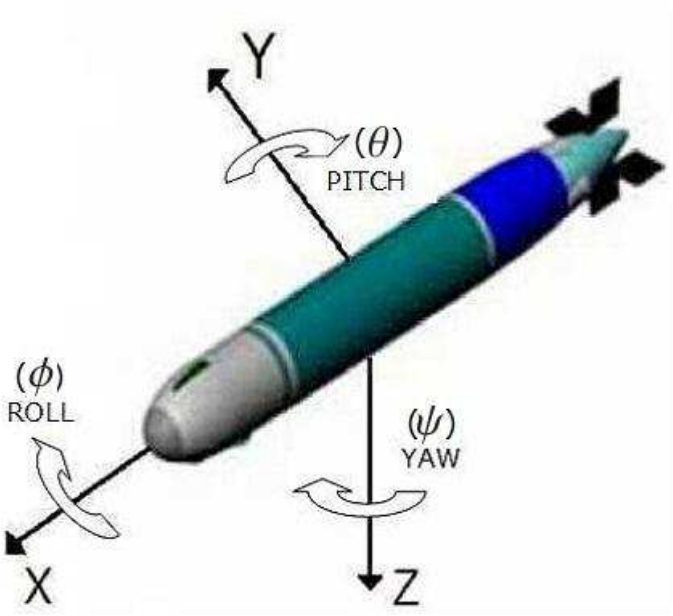

Figure 2.5: Euler angles

In this thesis work, a simple notation is adopted for the representation of specific forces and rotations. For example, ω_{ab}^{c} describes the rotation of frame-*a* relative to frame-*b*, and this rotation represented in frame-*c*. Similarly, f_{ab}^{c} means the specific force of frame-*a* relative to frame-*b*, and this force represented in frame-*c*.

Transformation between body and navigation frames is the most important transformation since Euler angles between these frames are highly dependent on vehicle dynamic motion. By using this transformantion matrix, linear velocities and angular velocities can be transformed from body frame to navigation frame or vice versa. In order to construct DCM matrix for body to navigation frame transformation, principal rotation matrices for each axes are required. Principal rotation matrices for x, y and z axis are given in Equations 2.3, 2.4 and 2.5 respectively.

$$R_{x,\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \tag{2.3}$$

$$R_{y,\theta} = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \tag{2.4}$$

$$R_{z,\psi} = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.5}$$

2.3.1 Linear Velocity Transformation Matrix

Transformation matrix for linear velocity transformation is described by principal rotation matrices about z, y and x axis. It is crucial that this order of axes is not arbitrary. In navigation systems, it is common to use this *zyx* convention for the transformation from body to navigation frame in terms of Euler angles [13]. Direction Cosine Matrix from body to navigation frame can be represented by $R_b^n(\Theta)$ where Θ is the Euler angles, $[\phi\ \theta\ \psi]^T$. Euler angles argument (Θ) is dropped to simplify notation, so that direction cosine matrix from body to navigation frame is denoted by R_b^n in this thesis work. Direction Cosine Matrix for linear velocity is formulated in Equations 2.6 - 2.7

$$\left(R_n^b\right)^{-1} = \left(R_n^b\right)^T = R_b^n = R_{z,\psi}\ R_{y,\theta}\ R_{x,\phi} \tag{2.6}$$

After, substituting the principle rotation matrices in Equations 2.3,2.4 and 2.5, final form of the DCM matrix is obtained in Equation 2.7.

$$R_b^n = \begin{bmatrix} \cos(\psi)\cos(\theta) & -\sin(\psi)\cos(\phi) + \cos(\psi)\sin(\theta)\sin(\phi) & \sin(\psi)\sin(\phi) + \cos(\psi)\sin(\theta)\cos(\phi) \\ \sin(\psi)\cos(\theta) & \cos(\psi)\cos(\phi) + \sin(\psi)\sin(\theta)\sin(\phi) & -\cos(\psi)\sin(\phi) + \sin(\psi)\sin(\theta)\cos(\phi) \\ -\sin(\theta) & \cos(\theta)\sin(\phi) & \cos(\theta)\cos(\phi) \end{bmatrix} \tag{2.7}$$

Note that direction cosine matrix for transformation between navigation and body frame is a unitary matrix.

2.3.2 Angular Velocity Transformation Matrix

The body fixed angular velocities and Euler's angle rates are related through Angular Velocity Transformation Matrix. This relation can be expressed as

$$\dot{\Theta} = T_b^n(\Theta)\ \omega_{nb}^b \tag{2.8}$$

where $T_b^n(\Theta)$ is the angular velocity transformation matrix from body frame to navigation frame and ω^b is the angular velocities (rates) of the vehicle measured in body frame. In this work, angular velocity transformation matrix is shortly symbolized by T_b^n. Angular velocity transformation matrix T_b^n is given in Equation 2.9 [13].

$$T_b^n = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)\sec(\theta) & \cos(\phi)\sec(\theta) \end{bmatrix} \tag{2.9}$$

2.4 Inertial Measurement Unit

Inertial Navigation System uses inertial data, acceleration and angular rate, to determine the position, velocity and attitude of the vehicle with respect to a known reference frame. Since INS uses acceleration and angular rate data, accelerometer and gyroscope are the main sensors of the INS systems. The combination of these two sensors is called Inertial Measurement Unit (IMU).

Different types of IMUs are available in the market for several applications. For critical applications, IMU's error must be minimum in order operate with high accuracy but low error IMUs are expensive. Thus, IMUs can be classified according to their error performances. This classification is presented with their specification limitations in Table 2.1. Strategic grades of

Table 2.1: IMU classification [1]

ERROR	**Tactical Grade** $< 20\ km/h$	**Navigation Grade** $< 1\ km/h$	**Strategic Grade** $< 30\ m/h$
Gyroscope Drift	$1-10\ deg/h$	$0.015\ deg/h$	$0.0001\ deg/h$
Accelerometer Bias	$100-1000\ \mu$g	$50-100\ \mu$g	$1\ \mu$g
COST	<$10000	$10000-70000	>$200000

inertial sensors are used in submarines and spacecraft to satisfy highest accuracy. Navigation grade IMUs are the main components of the aircraft's navigation areas and control systems. Finally, tactical grade IMUs have a wide range of utilization areas. They are prefered in guided weapons, unmanned aerial and underwater vehicles due to their lower cost and smaller size.

2.4.1 Accelerometer

Working principal of an accelerometer is based on spring-mass system. Figure 2.6 depicts simple accelerometer structure. A mass is free to move with respect to the accelerometer case. When any force is applied to the case, mass will move, compressing one spring and stretching the other. So the resultant position of the mass with respect to case is proportional to the acceleration applied to the case. The significant point is the acceleration which is caused by gravitational force. Gravitation acts on mass directly, not via the springs. So there is no relative motion of the mass with respect to case caused by gravitational force [3].

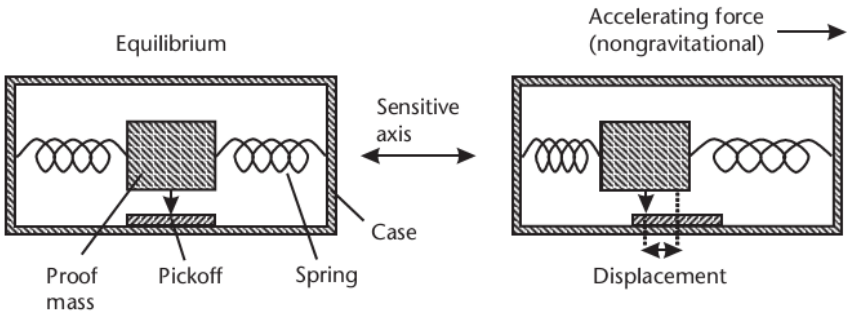

Figure 2.6: Simple accelerometer structure (taken from [3])

Accelerometer is a device that measures the specific force on the vehicle. This implies that accelerometer senses both inertial acceleration of the vehicle and gravitational acceleration. Therefore, when inertial forces are applied to the vehicle in the presence of Earth gravitational field, accelerometer will produce an output signal as in Equation 2.10.

$$f = \ddot{p} - G^b \tag{2.10}$$

In Equation 2.10, f represents specific force, p is the position (displacement) and G^b is the gravitational acceleration in body frame. To elaborate more about the accelerometer output, the above discussion clearly indicates that accelerometer measures specific force (or relative acceleration) between the case and the mass, not the accelerations that affect the case and mass identically [4]. It's worth considering some cases to understand the behaviour of the accelerometer. If an accelerometer is in free-fall with no applied external forces in a non-rotating frame, accelerometer output will be $f = 0$. If an accelerometer is fixed (not moving) in a non-rotating frame, then its output is $f = -G$. If a high quality accelerometer is at rest on

the Earth surface, then it measures the gravity and the force caused by Earth's rotation about its own axis. So the output will be $f = \widetilde{\omega}_{ie}\,\widetilde{\omega}_{ie}\,p - G$ where $\widetilde{\omega}_{ie}$ is the skew-symmetric matrix of the Earth angular rate.

MEMS technology is a crucial step for sensor applications since it gives a chance to produce sensors in small size with lower costs and large quatities. So development in the MEMS technology directly affects the inertial sensor technology, too. Even that MEMS inertial sensor production becomes one of the most popular subject of MEMS technology. Today, MEMS inertial sensors are used in a wide range of commercial applications. Accelerometers have also been advanced compared to old ones by means of MEMS technology. Now small, low cost, high range, high resolution MEMS accelerometers can be produced by many companies. Actually, MEMS accelerometer does not have different working principles. MEMS accelerometer may be built using either pendulous or vibrating-beam design like the old fashioned ones [3]. The difference in MEMS accelerometers comes from the production techniques that result as low power consumption, low cost and small size. Because of these advantageous features, MEMS accelerometer are prefered in most of the autonomous applications. So Analog Devices ADXL345 3-axis MEMS accelerometer is used for the implementation of this work.

2.4.2 Gyroscope

Gyroscope is a device that measures angular rate of the platform it is mounted on. Mechanically, gyroscopes are more complicated sensors than accelerometers. It is mostly used in marine applications. The first functional marine gyroscope is designed and used by Hermann Anschütz-Kaempfe [14]. These gyroscopes are mounted on a gimbaled platform so that they have a complicated structure to be built. Today, gimballed gyroscopes are still used in submarines, warships etc.

Gyroscopes can be categorized in three in terms of their working principles: spinning mass, optical and vibratory. Spinning mass gyroscopes operates on principle of conservation of angular momentum which is a part of Newton's second law of dynamics. There are different types of optical gyroscopes such as ring laser gyroscope (RLG), fiber optic gyroscope (FOG). They work on the principle that light travels at a constant speed in a given medium. Optical gyroscopes are high performance sensors, but they are highly priced equipments.

Modern rocketry and spacecraft technology are their popular usage areas. Vibratory gyroscopes' working princible is based on Coriolis effect. All MEMS gyroscopes are classified as vibratory gyroscope [3]. In the context of this thesis, InvenSense ITG-3200 3-axis MEMS gyroscope is modeled and used in the implementation.

MEMS gyroscopes are rigidly attached to the vehicle, so that they measure the angular rate of the body with respect to inertial frame in body frame. So it senses all of the rotation components: vehicle angular rate, angular rate as the vehicle moves about the spherical Earth (rotation of navigation frame with respect to Earth frame) and angular rate of the Earth itself [15]. The output of the gyroscope is given in Equation 2.11.

$$\omega_{ib}^{b} = \omega_{ie}^{b} + \omega_{en}^{b} + \omega_{nb}^{b} \tag{2.11}$$

In Equation 2.11, ω_{ie}^{b} represents Earth rotation about itself represented in body frame, ω_{en}^{b} is the angular rate of navigation frame with respect to Earth frame represented in body frame as the vehicle moves about the spherical Earth and ω_{nb}^{b} is the vehicle's angular rate about its axis represented again in body frame.

2.5 IMU Error Models

MEMS inertial sensors are favorable in today's navigation systems because of their low cost and small sizes but they suffer from the accuracy compared to the gimbaled and optical sensors. This low accuracy is caused by two different types of error sources which can be categorized as deterministic and stochastic errors. In early times of MEMS technology, deterministic errors affect the output of the system significantly unless they are calibrated. For example, in order to implement a three axes gyroscope system, there had to be three different single MEMS gyroscope mounted on the navigation system. This kind of a design comes up with a serious misalignment problem between axes. Today, almost all MEMS gyroscope produced with three axes in a single chip. Thus, misalignment error is minimized (approximately zero) at the production stage of the inertial sensor. So latest MEMS inertial sensors mainly suffer from stochastic error sources.

MEMS IMU measures the vehicle acceleration and angular velocity, then these measurements are integrated to obtain position, velocity and attitude of the vehicle. However, these measurements are not pure outcomes of the vehicle motion, they are disturbed by stochastic

error sources. Because of the integration, these errors are accumulated and result as drift in the position, velocity and attitude calculation. Therefore, MEMS sensors suffer from stochastic random noises severely so that the navigation system degrades if these variations in accelerometer and gyroscope outputs are not modelled and compensated properly [16].

There are different type of methods to identify the stochastic errors. Allan Variance technique is used in this thesis study which is the most commonly applied one for the error identification of MEMS inertial sensors. Allan Variance technique is actually a time domain series analysis which is developed in 1966 for studying the frequency stability of precision oscillators [17]. Allan variance method is simple and it is straightforward to interpret the results of it. Negative aspect of Allan Variance is that it needs long term static data to produce correct results. In this work, Allan variance method is applied offline and once it runs then its results are given as parameter to the navigation system. So this negative aspect does not affect the overall system.

2.5.1 Allan Variance

Allan Variance method requires short calculations so that they are easy to implement. The result of Allan Variance method gives information about five basic noise terms. These five basic noise terms are quantization noise, angle/velocity random walk, bias instability, rate/acceleration random walk, and ramp noise. Details of the method are given in this section [18]. Assume that, recorded static data has a length of N with a sampling time t_0. This data is divided into clusters and each cluster includes n samples($n < N/2$). So each cluster has a duration of T seconds, which equals to nt_0.

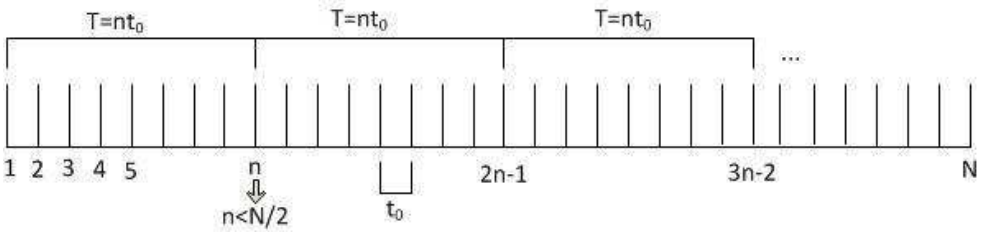

Figure 2.7: Data structure for Allan variance method

If the output of the sensor at time t is represented by $\Omega(t)$, cluster average is calculated as

$$\bar{\Omega}_k(T) = \frac{1}{T}\int_{t_k}^{t_k+T} \Omega(t)dt. \tag{2.12}$$

In Equation 2.12, $\bar{\Omega}_k(t)$ represents average of the k^{th} cluster out of N/n clusters. After obtaining averages of clusters, differences are calculated:

$$\xi_{k+1,k} = \bar{\Omega}_{k+1}(T) - \bar{\Omega}_k(T) \tag{2.13}$$

Allan variance aims to calculate the variance of these ξ which is calculated using the average of each consecutive clusters. Thus, Allan Variance of length T is calculated as follows:

$$\sigma^2(T) = \frac{1}{2(N-2n)}\sum_{k=1}^{N-2n}(\xi_{k+1,k})^2 \tag{2.14}$$

Allan Variance method is completed when these calculations are repeated for many different cluster times T. The analysis of the Allan Variance can be interpreted by an Allan Variance plot. A typical Allan Variance plot is given in Figure 2.8.

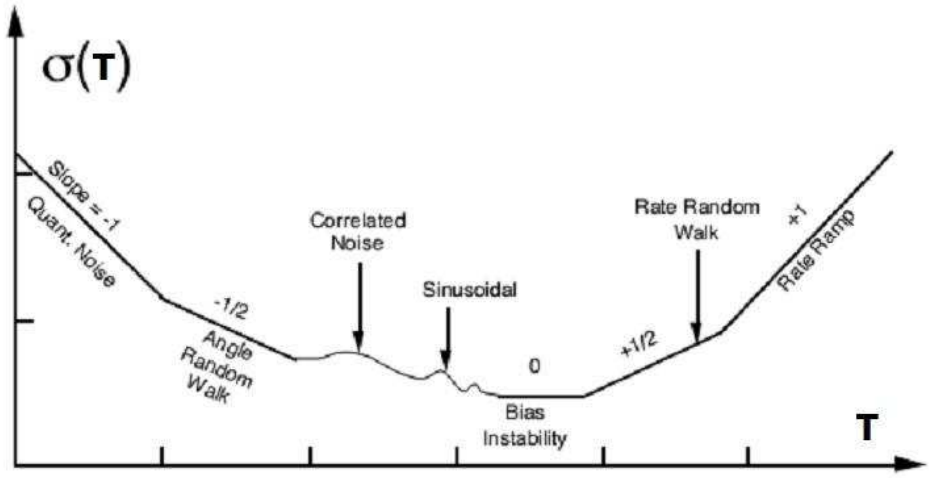

Figure 2.8: Typical Allan variance plot for MEMS inertial sensor (taken from [5])

The most significant feature of Allan variance is its ability to separate different types of noises by the slopes on the Allan Variance plot. Coefficients for five basic noise terms, their respective slopes and formulas in Allan Variance plot are given in Table 2.2.

Table 2.2: Allan variance noise analysis

Noise Types	AV $\sigma^2(\tau)$	Noise Coefficient	Slope
Quantization Noise	$\frac{3Q^2}{\tau^2}$	Q	-1
Angle/Velocity Random Walk	$\frac{N^2}{\tau}$	N	-1/2
Bias Instability	$\frac{2B^2 \ln(2)}{\pi}$	B	0
Rate/Acceleration Random Walk	$\frac{K^2\tau}{3}$	K	+1/2
Ramp Noise	$\frac{R^2}{2\tau^2}$	R	+1

2.5.2 Sensor Stochastic Model

In this section, stochastic sensor model is developed considering some noise terms available in the MEMS IMU sensors. Before going further, it is senseful to assume that the constant bias, misalignment error and scale factor error are calibrated by the manufacturer in the factory, so that these errors are not considered in the stochastic model of the sensors in this study.

The types of noise avaible in the MEMS inertial sensors are listed in Table 2.2. These noises are taken into consideration in the stochastic model of sensors. However, quantization noise is neglected in the model since Songlai Han and Knight points out that quantization noise cannot be modelled in Kalman filter equations due to the non-rational spectra depending on occurence of derivative of white noise [19]. So accelerometer and gyroscope have the following noise components in their outputs.

- **Angle / Velocity Random Walk (ARW)**

  Angle / Velocity Random Walk has rational spectra and it is modelled as white noise. Differential equation for ARW is given in Equation 2.15 [20].

  $$arw(t) = Nv(t) \tag{2.15}$$

  In Equation 2.15, N is the Allan Variance coefficient for ARW noise and $v(t)$ is unit white noise.

- **Bias Instability**

  Bias Instability is caused by variations in the bias term of the sensors. Many different

models are developed for bias instability noise. In this work, first order Gauss-Markov approach is used. The differential equation for bias instability noise is [20]

$$\dot{fn}(t) = -\beta fn(t) + \beta Bv(t) \tag{2.16}$$

where β is reciprocal of time constant which can be determined by autocorrelation analysis; B is Allan Variance bias instability noise parameter and $v(t)$ is unit white noise.

- **Rate / Acceleration Random Walk (RRW)**

  Rate / acceleration random walk has rational spectrum so that its model can be derived easily using its power spectral density. The differential equation for rate / acceleration random walk can be formulated as Equation 2.17 [20].

$$\dot{rrw}(t) = Kv(t) \tag{2.17}$$

  In Equation 2.17, K is the Allan Variance coefficient for RRW noise and $v(t)$ is unit white noise.

- **Ramp Noise**

  Ramp noise results as very slow monotonic change in the outputs of gyroscope and accelerometer. The effect of ramp noise could be observed over a long period of operation times (hours). This noise can be modelled approximately as second order Gauss-Markov process [20]. In this thesis study, the effect of ramp noise is disregarded because of its small impact.

Ultimately, our inertial sensor model is a combination of three different noise components plus true sensor measurements. Equations 2.18 and 2.19 show the formulation of accelerometer and gyroscope output models respectively.

$$a_{output} = a_{true} + a_{arw} + a_{fn} + a_{rrw} \tag{2.18}$$

$$w_{output} = w_{true} + w_{arw} + w_{fn} + w_{rrw} \tag{2.19}$$

2.6 Inertial Navigation Mechanization Equations

In this section, inertial navigation mechanization equations are derived considering Earth model, coriolis effect and gravity. The aim of these derivations is to express position, ve-

locity and attitude of the vehicle in terms of accelerometer, gyroscope and aiding sensors' outputs. Before the derivation, some notation and parameters need to be clarified.

In the equations, skew-symmetric matrices are used instead of cross product operations. Skew-symmetrix matrix form of a vector is represented in Equation 2.20.

$$a \times b = \widetilde{a}\, b = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \tag{2.20}$$

Earth rate is given in Equation 2.1 and it is symbolized as ω_{ie}^{e} which means angular rate of Earth frame with respect to inertial frame resresented in the Earth frame.

$$\omega_{ie}^{e} = \begin{bmatrix} 0 \\ 0 \\ 7.292115 \times 10^{-5} \end{bmatrix} rad/s \tag{2.21}$$

Equation 2.21 also implies that Earth rotational rate is constant such that $\dot{\omega}_{ie}^{e} = 0$. Moreover, since Earth rotational speed varies with different points on Earth, Earth rate equation in navigation frame is a function of geodetic latitude (L) of the vehicle. Then, Equation 2.22 shows the expression for Earth rate in navigation frame.

$$\omega_{ie}^{n} = \begin{bmatrix} 7.292115 \times 10^{-5}\ \cos(L) \\ 0 \\ -7.292115 \times 10^{-5}\ \sin(L) \end{bmatrix} \tag{2.22}$$

As it is stated in Section 2.2.3, navigation frame will move as vehicle changes its position. However, considering this rotation increases the computational load and the complexity of equations a lot. So it is assumed that navigation frame is fixed with respect to Earth frame such that $\omega_{in}^{n} = \omega_{ie}^{n}$.

For the gravity model, "plumb-bob gravity or local gravity" is used. This model comprises of gravitational acceleration plus centripetal acceleration caused by Earth rate. Therefore, the gravitational acceleration can be writen as in Equation 2.23.

$$G_{ib}^{n} = g_{ib}^{n} + \widetilde{\omega_{ie}^{n}}\, \widetilde{\omega_{ie}^{n}}\, p_{nb}^{n} \tag{2.23}$$

In Equation 2.23, $\widetilde{\omega_{ie}^{n}}$ (also equals to $\widetilde{\omega_{in}^{n}}$ because of fixed navgation frame) is the skew-symmetric form of Earth rate in navigation frame, p_{nb}^{n} is the position represented in navigation frame and g_{ib}^{n} is the gravitational acceleration in navigation frame.

Considering all the information given in the previous sections, generic inertial navigation mechanization equations can know be formulated. While the velocity is represented in body frame, position attitude angles are expressed in navigation frame in the equations. Mechanization equations are itemized as follows:

- **Position Equation**

  The derivative of position is velocity. So the derivative of position in navigation frame can be defined as

  $$\dot{p}^n_{nb} = R^n_b \, v^b_{nb} \tag{2.24}$$

  where v^b_{nb} is the velocity of the vehicle in body frame.

- **Velocity Equation**

  The derivative of velocity is acceleration. So the differential equation for velocity in navigation frame can be derived as follows by also taking into account the gravitational and coriolis forces [21]. Position of a vehicle with respect to inertial frame and its derivative can be written as

  $$p^i_{ib} = p^i_{in} + R^i_n \, p^n_{nb} \tag{2.25}$$

  $$\dot{p}^i_{ib} = \dot{p}^i_{in} + R^i_n[\widetilde{\omega^n_{in}} p^n_{nb} + \dot{p}^n_{nb}] \tag{2.26}$$

  where ω^n_{in} represents the rotation of navigation frame with respect to inertial frame and represented in navigation frame, p^i_{in} is the position of navigation frame relative to inertial frame, R^i_n is the transformation matrix from navigation frame to inertial frame and p^n_{nb} is the position of body frame with respect to navigation frame. Differentiating Equation 2.26 with respect to time and solving for acceleration of the body relative to navigation frame is given in Equations 2.27.

  $$\begin{aligned} \ddot{p}^n_{nb} &= f^n_{ib} + (G^n_{ib} - \widetilde{\omega^n_{in}}\widetilde{\omega^n_{in}} p^n_{nb}) - \ddot{p}^n_{in} - 2\widetilde{\omega^n_{in}} \, \dot{p}^n_{nb} - \dot{\widetilde{\omega^n_{in}}} \, p^n_{nb} \\ &= f^n_{ib} + g^n_{ib} - 2\widetilde{\omega^n_{in}} \, \dot{p}^n_{nb} \end{aligned} \tag{2.27}$$

  In Equation 2.27, f^n_{ib} is the specific force of body with respect to inertial frame represented in navigation frame. The first step of Equation 2.27 is simplified in the second step considering plumb-bob gravity, constant Earth rate and fixed navigation frame which are mentioned in this section before.

  Finally, using basic equalities $v^b_{nb} = \dot{p}^b_{nb}$, $\dot{p}^b_{nb} = R^b_n \, \dot{p}^n_{nb}$ and Equation 2.27, differential equation for the velocity of the body frame relative to navigation frame, represented in

body frame is calculated in Equations 2.28.

$$\begin{aligned}\ddot{p}^b_{nb} &= \dot{R}^b_n\, \dot{p}^n_{nb} + R^b_n \ddot{p}^n_{nb} \\ &= (R^b_n \widetilde{\omega^b_{nb}}) R^n_b \dot{p}^b_{nb} + R^b_n (f^n_{ib} + g^n_{ib} - 2\widetilde{\omega^n_{in}}\, \dot{p}^n_{nb}) \\ \dot{v}^b_{nb} &= f^b_{ib} + R^b_n g^n_{ib} - R^b_n \widetilde{\omega^n_{in}} R^n_b v^b_{nb} - \widetilde{\omega^b_{ib}} v^b_{nb}\end{aligned} \tag{2.28}$$

In the last part of Equation 2.28, ω^b_{ib} is also substituted for $\omega^b_{nb} + R^b_n \omega^n_{ie}$.

- **Attitude Equation**

  There are two different methods to define the attitude mechanization equation. The first one is already given in Equation 2.8. Another method is actually an indirect way of calculating attitude angles. The purpose of second method is to make use of the direction cosine matrix and ω^b_{nb}.Then differential equation for direction cosine matrix is

$$\dot{R}^n_b = R^n_b\, \widetilde{\omega^b_{nb}}. \tag{2.29}$$

  With the knowledge of initial conditions of the direction cosine matrix, R^n_b can be solved so that the attitude of the craft can be determined [15].

A summary of the mechanizations equations in terms of accelerometer output f^b_{ib} and gyroscope output ω^b_{nb} is given in Table 2.3.

Table 2.3: Inertial navigation mechanization equations

Position Equation	$\dot{p}^n_{nb} = R^n_b\, v^b_{nb}$
Velocity Equation	$\dot{v}^b_{nb} = f^b_{ib} + R^b_n g^n_{ib} - R^b_n \widetilde{\omega^n_{in}} R^n_b v^b_{nb} - \widetilde{\omega^b_{ib}} v^b_{nb}$
Attitude Equation	$\dot{\Theta} = T^n_b\, \omega^b_{nb}$ or $\dot{R}^n_b = R^n_b\, \widetilde{\omega^b_{nb}}$

2.7 Error Model of Inertial Navigation Mechanization Equations

For the integration of IMU sensors and aiding sensors, linear estimation filter namely Kalman Filter which is explained in Chapter 4, is applied. However, mechanization equations found

in Section 2.6 are non-linear equations. Therefore, these equations need to be linearized properly. For that purpose, this section describes the error model of previously defined navigation mechanization equations and their derivations.

For the linearization procedure, mechanization equations are perturbed by first order as in Equations 2.30, 2.31, 2.32.

$$p^n_{nb} = \hat{p}^n_{nb} + \delta p^n_{nb} \tag{2.30}$$

$$v^b_{nb} = \hat{v}^b_{nb} + \delta p^b_{nb} \tag{2.31}$$

$$R^n_b = \hat{R}^n_b + \delta R^n_b \tag{2.32}$$

Also IMU sensor outputs and gravitational acceleration are perturbed as in Equations 2.33, 2.34, 2.35. Actually, these perturbations in IMU sensors are the sum of stochastic IMU errors which are introduced in Section 2.5.2.

$$f^b_{ib} = \hat{f}^b_{ib} + \delta f^b_{ib} \tag{2.33}$$

$$w^b_{ib} = \hat{w}^b_{ib} + \delta w^b_{ib} \tag{2.34}$$

$$g^n_{ib} = \hat{g}^n_{ib} + \delta g^n_{ib} \tag{2.35}$$

Following relations are obtained by using small-angle approximation of transformation matrix and they are useful for further derivations [21]:

$$\hat{R}^n_b = (I - \widetilde{\delta\Theta})R^n_b \tag{2.36}$$

$$R^n_b = (I + \widetilde{\delta\Theta})\hat{R}^n_b \tag{2.37}$$

$$\hat{R}^b_n = R^b_n(I + \widetilde{\delta\Theta}) \tag{2.38}$$

$$R^b_n = \hat{R}^b_n(I - \widetilde{\delta\Theta}) \tag{2.39}$$

Then using the equations above and nonlinear mechanization equations, error model can be formulated for position, velocity and attitude.

- **Position Error Model Equation**

  By differentiating Equation 2.30 and using given relations, position error model is computed [21].

$$\begin{aligned}
\delta\dot{p}^n_{nb} &= \dot{p}^n_{nb} - \dot{\hat{p}}^n_{nb} \\
&= R^n_b v^b_{nb} - \hat{R}^n_b \hat{v}^b_{nb} \\
&= \left(\hat{R}^b_n(I - \widetilde{\delta\Theta})\right)(\delta v^b_{nb} + \hat{v}^b_{nb}) - \hat{R}^n_b \hat{v}^b_{nb} \\
&= \hat{R}^n_b \delta v^b_{nb} + \widetilde{\delta\Theta}\hat{R}^n_b \delta v^b_{nb} + \widetilde{\delta\Theta}\hat{R}^n_b \hat{v}^b_{nb}
\end{aligned} \tag{2.40}$$

In Equation 2.40, $\widetilde{\delta\Theta}\hat{R}_b^n\delta v_{nb}^b$ part could be neglected since both $\delta\Theta$ and δv_{nb}^b is small. Then, the final form of position error is rewritten in Equation 2.41 by using the fact that $\widetilde{a}\,b = -\widetilde{b}\,a$.

$$\delta\dot{p}_{nb}^n \approx \hat{R}_b^n\delta v_{nb}^b - \widetilde{\hat{R}_b^n\hat{v}_{nb}^b}\,\delta\Theta \tag{2.41}$$

- **Velocity Error Model Equation**

  The method in the derivation of position error model is adopted for velocity error model equation but this time the derivation is longer and contains more terms. So the final form of the velocity error model is given in Equation 2.42 [21].

$$\begin{aligned}\delta\dot{v}_{nb}^b &= \dot{v}_{nb}^b - \dot{\hat{v}}_{nb}^b \\ &\approx \hat{R}_n^b\left(\widetilde{\hat{g}_{ib}^n} + \hat{\omega}_{ie}^n(\hat{R}_b^n\hat{v}_{nb}^b)^T - (\hat{\omega}_{ie}^n\cdot\hat{R}_b^n\hat{v}_{nb}^b)I\right)\delta\Theta \\ &- \left(\widetilde{\hat{\omega}_{ib}^b} + \hat{R}_n^b\widetilde{\hat{\omega}_{in}^n}\hat{R}_b^n\right)\delta v_{nb}^b - \delta f_{ib}^b - \widetilde{\hat{v}_{nb}^b}\delta\omega_{ib}^b \\ &+ \hat{R}_n^b\delta g_{ib}^n + \hat{R}_n^b\widetilde{\hat{R}_b^n\hat{v}_{nb}^b}\delta\omega_{ie}^n\end{aligned} \tag{2.42}$$

  In Equation 2.42, $\delta\omega_{ie}^n$ represents the error of Earth rate in navigation frame. Since Equation 2.42 consists of detailed error parameters, INS system requires a comprehensive analyses and sensitive IMU sensors.

- **Attitude Error Model Equation**

  The attitude error model is derived by using Equation 2.29. In order to formulate the attitude error model, following steps must be computed.

$$\begin{aligned}\delta\dot{R}_b^n &= \dot{R}_b^n - \dot{\hat{R}}_b^n \\ &= R_b^n\widetilde{\omega_{nb}^b} - \hat{R}_b^n\widetilde{\hat{\omega}_{nb}^b} \\ &= (I + \widetilde{\delta\Theta})\hat{R}_b^n\widetilde{\omega_{nb}^b} - \hat{R}_b^n\widetilde{\hat{\omega}_{nb}^b}\end{aligned} \tag{2.43}$$

$$\delta\dot{R}_b^n = \hat{R}_b^n(\widetilde{\omega_{nb}^b} - \widetilde{\hat{\omega}_{nb}^b}) + \widetilde{\delta\Theta}\hat{R}_b^n\widetilde{\omega_{nb}^b} \tag{2.44}$$

Also

$$\begin{aligned}\delta R_b^n &= R_b^n - \hat{R}_b^n \\ &= (I + \widetilde{\delta\Theta})\hat{R}_b^n - \hat{R}_b^n \\ \delta R_b^n &= \widetilde{\delta\Theta}\hat{R}_b^n\end{aligned} \tag{2.45}$$

and by taking the derivative with respect to time of Equation 2.45 gives

$$\begin{aligned}\delta\dot{R}_b^n &= \dot{\widetilde{\delta\Theta}}\hat{R}_b^n + \widetilde{\delta\Theta}\dot{\hat{R}}_b^n \\ &= \dot{\widetilde{\delta\Theta}}\hat{R}_b^n + \widetilde{\delta\Theta}\hat{R}_b^n\widetilde{\hat{\omega}_{nb}^b}\end{aligned} \tag{2.46}$$

Combining Equation 2.44 and 2.46 results in

$$\hat{R}_b^n(\widetilde{\omega_{nb}^b} - \widetilde{\hat{\omega}_{nb}^b}) + \delta\Theta\hat{R}_b^n\widetilde{\omega_{nb}^b} = \dot{\widetilde{\delta\Theta}}\hat{R}_b^n + \widetilde{\delta\Theta}\hat{R}_b^n\widetilde{\hat{\omega}_{nb}^b}$$

$$\dot{\widetilde{\delta\Theta}}\hat{R}_b^n = -\widetilde{\delta\Theta}\hat{R}_b^n\widetilde{\hat{\omega}_{nb}^b} + \hat{R}_b^n(\widetilde{\omega_{nb}^b} - \widetilde{\hat{\omega}_{nb}^b}) + \widetilde{\delta\Theta}\hat{R}_b^n\widetilde{\omega_{nb}^b}$$

$$\dot{\widetilde{\delta\Theta}} = -\widetilde{\delta\Theta}\hat{R}_b^n\widetilde{\hat{\omega}_{nb}^b}\hat{R}_n^b + \hat{R}_b^n(\widetilde{\omega_{nb}^b} - \widetilde{\hat{\omega}_{nb}^b})\hat{R}_n^b + \widetilde{\delta\Theta}\hat{R}_b^n\widetilde{\omega_{nb}^b}\hat{R}_n^b$$

$$\dot{\widetilde{\delta\Theta}} = \widetilde{\delta\Theta}\hat{R}_b^n(\widetilde{\omega_{nb}^b} - \widetilde{\hat{\omega}_{nb}^b})\hat{R}_n^b + \hat{R}_b^n(\widetilde{\omega_{nb}^b} - \widetilde{\hat{\omega}_{nb}^b})\hat{R}_n^b \tag{2.47}$$

$$\dot{\widetilde{\delta\Theta}} \approx \hat{R}_b^n(\widetilde{\delta w_{ib}^b})\hat{R}_n^b \tag{2.48}$$

where Equation 2.47 is approximated as Equation 2.48 since $\widetilde{\delta\Theta}$ and $(\widetilde{\omega_{nb}^b} - \widetilde{\hat{\omega}_{nb}^b})$ are both small matrices.

Finally, Equation 2.48 can be written in vector form as

$$\dot{\delta\Theta} = \hat{R}_b^n\,\delta w_{ib}^b \tag{2.49}$$

So Equation 2.49 shows that attitude error equation depends on estimate of transformation matrix, $\hat{R}_b^n$ and the gyroscope error vector, δw_{ib}^b .

A summary of the linearized error model of mechanizations equations is given in Table 2.4.

Table 2.4: Error model of inertial navigation mechanization equations

Position Error Equation	$\delta\dot{p}_{nb}^n \approx \hat{R}_b^n\delta v_{nb}^b - \widetilde{\hat{R}_b^n\hat{v}_{nb}^b}\,\delta\Theta$
Velocity Error Equation	$\delta\dot{v}_{nb}^b \approx \hat{R}_n^b\left(\widetilde{\hat{g}_{ib}^n} + \hat{\omega}_{ie}^n(\hat{R}_b^n\hat{v}_{nb}^b)^T - (\hat{\omega}_{ie}^n\cdot\hat{R}_b^n\hat{v}_{nb}^b)I\right)\delta\Theta$ $-\left(\widetilde{\hat{\omega}_{ib}^b} + \hat{R}_n^b\widetilde{\hat{\omega}_{in}^n}\hat{R}_b^n\right)\delta v_{nb}^b - \delta f_{ib}^b - \widetilde{\hat{v}_{nb}^b}\delta\omega_{ib}^b$ $+\hat{R}_n^b\delta g_{ib}^n + \hat{R}_n^b\widetilde{\hat{R}_b^n\hat{v}_{nb}^b}\delta\omega_{ie}^n$
Attitude Error Equation	$\dot{\delta\Theta} = \hat{R}_b^n\,\delta w_{ib}^b$

CHAPTER 3

AIDING SENSORS

IMU sensors are not sufficient alone for the navigation system of an underwater vehicle. A simple time integration algorithm is an ineffective solution due to their error characteristics. Using an aiding sensor as a measurement could assist to correct these errors that grow with time. In practice, most commonly used aiding sensors are GPS (Global Positioning System), magnetometer, doppler navigation sensor, pressure sensor and acoustic navigation systems [22]. Doppler navigation sensors are called Doppler Velocity Log (DVL). This sensor measures the speed of a vehicle with respect to sea bed using sonar signals. DVL provides velocity of a vehicle with a high precision but they are very expensive devices. GPS is a satellite based system and communicates with radio signals. Using GPS is a very efficient way of determining the position and velocity of the vehicle in Earth frame. However, GPS signals cannot penetrate the water. Thus, GPS does not work when the vehicle plunges into water. Moreover, acoustic navigation systems are used to obtain position of the vehicle according to deployed set of transponders. There are three different types of acoustic positioning systems depending on their baseline length: Ultrashort Baseline (USBL), Short Baseline (SBL) and Long Baseline (LBL). The position of the vehicle(or responder) is determined by using phase difference of the detected signals from the transponders [23]. Although they provide very good position accuracy independent of water depth, they are complex systems that require expert operators, long times for setup and expensive equipments.

Because of the disadvantages of the systems decribed above, magnetometer and pressure sensor are chosen as aiding sensors in this work. Magnetometer and pressure sensor are more appropriate in terms of size, operation simplicity and cost efficiency. In this chapter, properties of magnetometer and pressure sensor are decribed. Moreover, their error characteristics and

calibration methods are examined throughout this chapter.

3.1 Magnetometer

North finding is one of the oldest method of navigation techniques. Looking at north pole star is the most popular north finding method. Another method is to make use of Earth's magnetic field. Magnetic compass aligns itself according to Earth's magnetic field. By assuming Earth's magnetic do not change over long time period, magnetic compasses are started to be used for navigation purposes, especially for direction finding.

Electronic compass (magnetometer) measures Earth's magnetic field in three axis, thus it provides low cost heading calculation with minor errors. Magnetoresistive, fluxgate, magnetoinductive types of electronic compasses are available. In this study, Honeywell HMC5883L series magnetoresistive sensor is used. The sensor converts any incident magnetic field in the sensitive axis directions to a differential voltage output. These magnetoresistive sensors are made of a nickel-iron (Permalloy) thin-film and patterned as a resistive strip element. In the presence of any magnetic field, a change in resistive elements causes a relative change in the output voltage [24]. Heading of vehicle can be calculated by Equation 3.1 [4]. Magnetometer has to be calibrated and tilt compensated according to its error analysis before calculating heading angle.

$$\psi_{magneto} = \tan^{-1}\left(-\frac{magneto_y}{magneto_x}\right) \tag{3.1}$$

3.1.1 Magnetometer Error Analysis and Calibration Procedures

Magnetometer's measurements contain different type of errors and some of these errors have a serious impact on heading calculation. These errors are [6]:

- Quantization Error
- Magnetic Sensor Error
- Temperature Effects
- Nearby Ferrous Materials

- Compass Tilt Errors
- Variation of the Earth's Magnetic Field

In this study, quantization error and temperature effects are not considered in the calculations of the heading. Also it is assumed that Earth's magnetics field do not change due to position, since the vehicle moves in a limited local area. So variation of the Earth's magnetic field is negligible.

3.1.1.1 Magnetic Sensor Error

Magnetometers have some errors due to their internal structure. These errors may also be due to production imperfections. These errors could be sensor noise, linearity, hysteresis, and repeatability errors [6]. It is hard to model and compensate all of these error. Therefore, the effect of these errors on heading calculation is modeled as additive white Gaussian noise.

3.1.1.2 Nearby Ferrous Materials

Earth's magnetic field could be distorted easily by using any ferrous materials since magnitude of Earth's magnetic field is not too strong. Therefore, it's expected that magnetometer's measurements are disturbed by nearby ferrous materials. If it is assumed that Earth's magnetic field is constant in a local operation area, then the effect of any ferrous object will be constant as well. This effect is illustrated in Figure 3.1.

When a magnetometer is rotated arbitrarily in a completely isolated environment, the plot of measurements in three axis must be a complete sphere centered at the origin. However, it is not the case in practice due to the metalic vehicle body, other electronic components, wiring etc. Disturbance of these materials can be categorized as hard iron and soft iron effects. The cause of hard iron distortion is the permanent magnets and ferromagnetic irons on the vehicle platform. Hard iron effect add a constant magnitude field component along each axes so that its effect does not change for different heading orientations. This addition of constant magnitude field appears as a shift in the origin of the sphere. The cause of the soft iron distortion is the interaction of the Earth's magnetic field and any magnetically soft materials around the magnetometer. The difference between soft and hard iron distortion is that soft

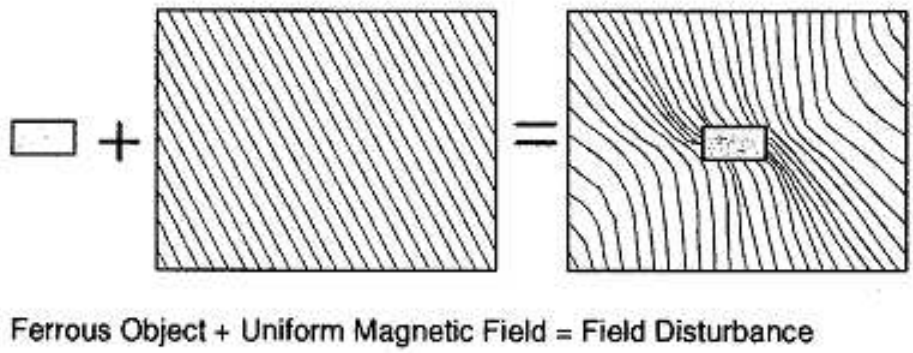

Figure 3.1: Disturbance of ferrous material in uniform magnetic field (taken from [6])

iron distortion depends on magnetometer orientation. When magnetometer experiences a soft iron distortion, the plot of measurements of arbitrary movements become ellipsoidal [6].

Nearby ferrous materials may cause serious errors in heading calculation if the magnetometer is not calibrated. In order to determine the calibration parameters, simple offline procedure can be applied [25, 26]:

- Magnetometer is mounted permenantly on the vehicle and then outputs of the magnetometer is recorded when the vehicle is rotated around at each three axis arbitrarily.
- By using recorded data, measurement ranges of each axis is calculated:

$$\begin{aligned} range_x &= max(mag_x^{raw}) - min(mag_x^{raw}) \\ range_y &= max(mag_y^{raw}) - min(mag_y^{raw}) \\ range_z &= max(mag_z^{raw}) - min(mag_z^{raw}) \end{aligned} \tag{3.2}$$

  In Equations 3.2, mag_x^{raw},mag_y^{raw} and mag_z^{raw} represent raw magnetometer data for x,y,z axes respectively.

- Scale factor coefficient for the soft iron distortion can be determined by using these range values.

  if $range_x > range_y$ **then**

  $S_y = \frac{range_x}{range_y}$

  $S_x = 1$

else

$S_x = \frac{range_y}{range_x}$

$S_y = 1$

end if

if $S_x == 1$ **then**

$S_z = \frac{range_x}{range_z}$

else

$S_z = \frac{range_y}{range_z}$

end if

- After the calculation of soft iron scale factors, constant offsets due to the hard iron effect for each axis are calculated:

$$\begin{aligned} Offset_x &= \left(\frac{max(mag_x^{raw}) - min(mag_x^{raw})}{2} - max(mag_x^{raw}) \right) \times S_x \\ Offset_y &= \left(\frac{max(mag_y^{raw}) - min(mag_y^{raw})}{2} - max(mag_y^{raw}) \right) \times S_y \\ Offset_z &= \left(\frac{max(mag_z^{raw}) - min(mag_z^{raw})}{2} - max(mag_z^{raw}) \right) \times S_z \end{aligned} \tag{3.3}$$

- Finally, these calculated scale factors and offset values are applied to all magnetometer data with respective axis.

$$\begin{aligned} mag_x^{calibrated} &= mag_x^{raw} \times S_x + Offset_x \\ mag_y^{calibrated} &= mag_y^{raw} \times S_y + Offset_y \\ mag_z^{calibrated} &= mag_z^{raw} \times S_z + Offset_z \end{aligned} \tag{3.4}$$

In the literature, only x and y axes are the subject of hard and soft iron calibration. In this thesis work, z axis is also considered for calibration, so that calibration method is modified in that way. This calibration procedure should be carried out for all different setups, since calibration parameter varies according to position and orientation of magnetometer in the platform.

3.1.1.3 Compass Tilt Errors

Magnetometer is attached to the platform in a fixed position. So, the measurement axes of the magnetometer depend on the vehicle orientation. However, Earth's magnetic field is assumed

to be constant in local area according to earth frame. Therefore, as the vehicle changes its roll and pitch angles, magnetometer will measure the Earth's magnetic field with a wrong axis configuration which concludes as erroneous heading calculation. These tilt error (roll and pitch) usually contributes the largest percentage error in heading calculation [6]. In order to compensate for the tilt error, it is required to know roll and pitch angles simultaneously. Therefore, many magnetometer manufacturers assembles tilt sensors (accelerometer) with their magnetometers. Tilt sensor calculates roll and pitch angles by taking advantage of gravitational acceleration. Because of this, if vehicle moves with a high acceleration, tilt sensor will not be able to calculate roll and pitch angles correctly.

For correct heading angle, magnetometer's axes have to be aligned back to the horizontal plane. For that reason, principal rotations for the NED frame have to be used. These rotation matrices were already given in Equations 2.3, 2.4 and 2.5. Equation 3.5 shows the back rotated magnetometer measurements.

$$\begin{bmatrix} mag_x^{rotated} \\ mag_y^{rotated} \\ mag_z^{rotated} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 1 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} mag_x^{calibrated} \\ mag_y^{calibrated} \\ mag_z^{calibrated} \end{bmatrix} \tag{3.5}$$

Since heading angle is specified as in Equation 3.1, only the x and y component of magnetometer (calibrated and tilt compensated) is required for the computation of heading angle by using mangetometer.

$$\begin{aligned} mag_x^{rotated} = {} & mag_x^{calibrated} \cos(\theta) + mag_y^{calibrated} \sin(\theta) \sin(\phi) \\ & + mag_z^{calibrated} \sin(\theta) \cos(\phi) \end{aligned} \tag{3.6}$$

$$mag_y^{rotated} = mag_y^{calibrated} \cos(\phi) - mag_z^{calibrated} \sin(\phi) \tag{3.7}$$

Finally, rotated and calibrated magnetometer measurements for x and y axis in Equation 3.6 and Equation 3.7 can be used for heading calculation. The result of Equation 3.1 gives heading angle with respect to magnetic north and this result later on will be used as a measurement for yaw angle.

3.2 Pressure Sensor

Pressure sensor measures the amount of water pressure on the underwater vehicle. By using this property, z-axis position of the vehicle can be calculated and this measurement could be

given as an auxiliary measurement for the navigation system. Pressure sensors are not complex sensors like magnetometers. Most manufacturers produce their items fully temperature compensated, calibrated and amplified before delivery. So it's easy to use for different applications. In this work, the depth of water (z-axis position) will be measured using pressure sensor.

Assuming that water density is constant and equals to distilled water under standard conditions, then depth of the water can be calculated as in Equation 3.8.

$$depth = P \times 10.1971 \tag{3.8}$$

In Equation 3.8, P refers to water pressure measured by pressure sensor in terms of bar. Since 1 bar corresponds to 10.1971 meter of water height, Equation 3.8 gives depth in terms of meter.

Pressure sensor can measure the water pressure with a high accuracy. This accuracy mostly depends on their full scale range; as the range of the sensor increases, their accuracy decreases. Honeywell MLH series pressure sensor was used in this work. It delivers as low as 2% total error of its full scale and its response time is smaller than 2 ms [27].

CHAPTER 4

KALMAN FILTER

Kalman filter is a recursive data processing algorithm which generates optimal estimate of the state variables using a set of measurements. Kalman filter works on a linear system model in which process noises and measurement noises are white and Gaussian. When these conditions satisfied, Kalman filter produces unique best estimate of the state, so that it is called the optimal filter [28]. In this work, Kalman filter is used for sensor fusion and error estimation. By using estimated error states, true state variables can be obtained.

Although Kalman filter works for linear problems, it can be also applied for non-linear problems. For non-linear problems, either dynamic system equations are linearized by pertubation technique or a different version of Kalman filter, Extended Kalman Filter (EKF) is used. This version of Kalman Filter uses partial derivatives as linear approximations of nonlinear equations.

This chapter of the thesis provides fundamental background for the Kalman filter. In the first section, continuous time state space representation of a generic system is given. In the second section, conversion from continuous to discrete time Kalman filter is explained with its algorithm and the third section decribes the implementation of discrete extended Kalman filter.

4.1 Continuous Time State Space Models

In real world, it is sometimes easy to formulate system dynamics in continuous time rather than discrete time. However, most of the system implementations depend on today's digital technology. Considering this situation, modelling the system in continuous time and convert-

ing this model from continuous time domain to discrete time domain seems to be the best option.

Systems can be classified as linear systems and nonlinear systems. State space models of linear and nonlinear systems have different representations.

4.1.1 Continuous Time Linear State Space Models

The dynamics of a linear system can be described as

$$\dot{x}(t) = A(t)x(t) + G(t)w(t) \tag{4.1}$$

$$z(t) = C(t)x(t) + v(t) \tag{4.2}$$

In equation 4.1, $x(t)$ is $n \times 1$ state vector, $A(t)$ is $n \times n$ state transition matrix, $G(t)$ is $n \times r$ process noise coupling matrix and $w(t)$ is $1 \times r$ zero mean process noise. In equation 4.2, $z(t)$ is $m \times 1$ measurement vector, $C(t)$ is $m \times n$ measurement sensitivity matrix and $v(t)$ is $m \times 1$ zero mean measurement noise vector. $v(t)$ can be expressed as zero mean additive Gaussian noise with

$$E\left\langle v(t)v^T(t+\tau)\right\rangle = R_c(t)\delta(\tau) \tag{4.3}$$

$w(t)$ can be expressed as zero mean additive Gaussian noise with

$$E\left\langle w(t)w^T(t+\tau)\right\rangle = Q_c(t)\delta(\tau) \tag{4.4}$$

4.1.2 Continuous Time Nonlinear State Space Models

The dynamics of a nonlinear system can be described as

$$\dot{x}(t) = f(x(t), t) + g(w(t), t) \tag{4.5}$$

$$z(t) = h(x(t), t) + v(t) \tag{4.6}$$

In Equation 4.5, $x(t)$ is state vector, $f(x(t), t)$ is nonlinear state transition function, $g(w(t), t)$ is process noise coupling fucntion and $w(t)$ is zero mean process noise. In Equation 4.6, $z(t)$ is measurement vector, $h(x(t), t)$ is nonlinear measurement sensitivity function and $v(t)$ is zero mean measurement noise vector. $v(t)$ and $w(t)$ can be expressed as zero mean additive Gaussian noise with same covariance given in Equations 4.3 - 4.4.

4.2 Discrete Time Kalman Filter

Since Kalman filter is implemented in digital platforms, continous time systems have to be converted to discrete time domain. This convertion is based on linear state space representation of a system since Kalman filter works for linear systems. A discrete time linear system can be represented as in the following form of Equations 4.7 - 4.8.

$$x_{k+1} = F_k x_k + w_k \tag{4.7}$$

$$z_{k+1} = H_k x_{k+1} + v_{k+1} \tag{4.8}$$

$A(t_k)$ and $G(t_k)$ are the state transition matrix and process noise coupling matrix at a sample time t_k respectively, assuming that continuous time state transition matrix $A(t)$ and process noise coupling matrix $G(t)$, which are defined in Equation 4.1, are constant over a sampling time period Δt. So Equation 4.7 can be written in terms of continuous time state space system parameters as follows [29].

$$x_{k+1} = e^{A(t_k)\,\Delta t} x_k + \int_t^{t+\Delta t} e^{A(t_k)\,(t+\Delta t-\tau)} G(t_k)\, w(\tau) d\tau \tag{4.9}$$

Equation 4.9 shows that state transition matrix F_k is a matrix exponential which equals to $e^{A(t_k)\,\Delta t}$. So it can be calculated easily by using Taylor series expansion. However, computation of an integral involving matrix exponential is more challenging than a matrix exponential. For the calculation of discrete time process noise matrix, Van Loan's method (1978) is used [30].

$$\Upsilon = \exp\left(\begin{bmatrix} -A & GQ_cG^T \\ 0 & A^T \end{bmatrix} \Delta t\right) \tag{4.10}$$

$$= \begin{bmatrix} \ldots & F^{-1}Q_d \\ 0 & F^T \end{bmatrix} \tag{4.11}$$

Then, by using upper right and the transpose of lower right part of the matrix Υ, discrete time equivalent process noise covariance matrix is obtained.

$$Q_d = (F^T)^T * F^{-1}Q_d \tag{4.12}$$

Although Van Loan's methods appears to be easy intuitively, it requires large amount of numerical calculation if state space matrices are big. Then, using first order approximation of these numerical calculations decreases computational load. Thus, algorithm will be able to

work faster. Continuous to discrete time conversion of state space representation is given in Equations 4.13 according to first order approximation by assuming again that system dynamics matrices are constant over a sampling time period [28, 29] .

$$\begin{aligned} F_k &= I + A(t_k)\Delta t \\ Q_d &= G(t_k)Q_c(t_k)G(t_k)^T\Delta t \\ H_k &= C(t_k) \\ R_d &= \frac{R_c(t_k)}{\Delta t} \end{aligned} \tag{4.13}$$

Discrete Time Kalman Filter estimates the state of a noisy system by using measurements which are also exposed to noise. These noises are assumed to be white Gaussian noise, so that Kalman filter predicts the best optimal state. Discrete Time Kalman filter contains two stages: time update and measurement update. In the time update stage, state is predicted using the system model and in measurement update stage, predicted state in time update stage is corrected by current measurements. The aim of the whole algorithm is to minimize error covariance matrix of the estimator. The Kalman Filter algorithm for discrete time system is summarized in Table 4.1 [28].

Table 4.1: Discrete time Kalman filter algorithm

Discrete Time Linear State Space Model	$x_{k+1} = F_k x_k + w_k$ $z_{k+1} = H_k x_{k+1} + v_{k+1}$
Initialization	$\hat{x}_0 = E(x_0)$ $P_0 = E\left((x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T\right)$
Time Update	$\hat{x}_{k+1\|k} = F_k \hat{x}_{k\|k}$ $P_{k+1\|k} = F_k P_{k\|k} F_k^T + Q_d$
Kalman Gain	$K_{k+1} = P_{k+1\|k} H_k^T \left(H_k P_{k+1\|k} H_k^T + R_d\right)^{-1}$
Measurement Update	$\hat{x}_{k+1\|k+1} = \hat{x}_{k+1\|k} + K_{k+1}(z_{k+1} - H_k \hat{x}_{k+1\|k})$ $P_{k+1\|k+1} = (I - K_{k+1} H_k) P_{k+1\|k}$

4.3 Discrete Extended Kalman Filter

Standard Kalman filter has been formulated for the linear systems. However, the reality is that most of the systems in engineering world are nonlinear. So, linearization procedure can

be used to extend the Kalman filter for nonlinear systems. Hence this filter is called Extended Kalman Filter. Extended Kalman filter is commonly used in navigation systems.

The main idea of behind the extended Kalman filter is to linearize the nonlinear state-space model of Equations 4.5 - 4.6 at each time instant around the most recent state estimate. Once a linear model is obtained, the standard Kalman filter equations are applied [31].

In order to implement extended Kalman filter, first of all, continous time nonlinear state space system is given in Equations 4.5 - 4.6 should be written in discrete form of equations with the input parameter u_k:

$$x_{k+1} = f_k(x_k, u_k) + w_k \tag{4.14}$$

$$z_{k+1} = h_k(x_{k+1}) + v_{k+1} \tag{4.15}$$

Linearization procedure is applied by calculating Jacobian matrices of f_k and h_k given in Equations 4.16

$$\begin{aligned} F_k &= \frac{\partial f_k(x_k, u_k)}{\partial x} \mid_{x=x_{k|k}, u=u_k} \\ H_k &= \frac{\partial h_k(x_k)}{\partial x} \mid_{x=\hat{x}_{k+1|k}} \end{aligned} \tag{4.16}$$

Using this linearization procedure, extended Kalman filter algorithm for discrete time system is described in Table 4.2 [29, 32].

Table 4.2: Discrete time extended Kalman filter algorithm

Discrete Time Nonlinear State Space Model	$x_{k+1} = f_k(x_k, u_k) + w_k$ $z_{k+1} = h_k(x_{k+1}) + v_{k+1}$
Initialization	$\hat{x}_0 = E(x_0)$ $P_0 = E\left((x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T\right)$
Jacobians for Linearization	$F_k = \frac{\partial f_k(x_k, u_k)}{\partial x} \mid_{x=x_k, u=u_k}$ $H_k = \frac{\partial h_k(x_k)}{\partial x} \mid_{x=x_{k+1\|k}}$
Time Update	$\hat{x}_{k+1\|k} = f_k(\hat{x}_{k\|k}, u_k)$ $P_{k+1\|k} = F_k P_{k\|k} F_k^T + Q_d$
Kalman Gain	$K_{k+1} = P_{k+1\|k} H_k^T \left(H_k P_{k+1\|k} H_k^T + R_d\right)^{-1}$
Measurement Update	$\hat{x}_{k+1\|k+1} = \hat{x}_{k+1\|k} + K_{k+1}\left(z_{k+1} - h_k(\hat{x}_{k+1\|k})\right)$ $P_{k+1\|k+1} = (I - K_{k+1} H_k) P_{k+1\|k}$

EKF has two different implementation types: direct (total-state) EKF and indirect (error-

state) EKF. Direct EKF uses original navigation variables in the states of the filter, that are available in system formulation in Section 2.6. Indirect EKF uses error states (also used for linearization) of the navigation variables which are depicted in Section 2.7 [33]. Indirect EKF corrects the navigation variable by updating these error states in the measurement update stage of Kalman filter.

If the indirect Kalman filter collapses because of any failures, the inertial navigation system estimator part will continue to integrate the IMU data since Kalman filter is out of the INS estimator. However, for direct Kalman filter configuration, INS estimator will be useless if the Kalman filter collapses since outputs of the INS estimator goes directly to Kalman filter. Therefore, entire navigation system will crash when any failures occurs in direct Kalman filter configuration [34].

CHAPTER 5

SYSTEM DESIGN AND SIMULATIONS

This chapter focuses on a strapdown inertial navigation system which is designed for a underwater vehicle using a low cost IMU, a magnetometer and a pressure sensor. The effects of the aiding sensors on the system outputs are observed in simulations. The designed system is different from the common ones mainly in terms of the way the pressure sensor is coupled and vehicle dynamic motion assumptions. Accelerometer is used also as an aiding sensor for measuring tilt angles (pitch and roll) and the filter for the sensor integration is adapted according to the total acceleration of the vehicle. Although the main point of this thesis is AHRS, velocities and positions are also taken into consideration because of the presssure sensor aid. Thus the system is implemented as an INS, however the observed states are attitude and heading angles. Considering all of these specific features, the implementation steps of the system is explained throughout this chapter.

For the system implementation, Sparkfun 9DOF Razor IMU sensor is used. This IMU includes ITG-3200 triple-axis digital output gyroscope, ADXL345 triple-axis accelerometer and HMC5883L triple-axis digital magnetometer [7]. This sensor has ATmega328 processor on board to process sensor outputs but this is a weak processor to implement high dimensional Kalman filters. In order to solve this problem, sensor outputs are recorded and all other processes are applied in MATLAB© environment.

For the integration of the IMU sensor and aiding sensors, linearized error state Kalman filter and indirect extended Kalman filter are implemented in discrete time whose details are given in Chapter 4. Thus, error states are the key point for both of the these filters. Moreover there are two types of implementations of the overall system based on the outputs of Kalman filter (error states), feedforward and feedback. Their main difference is the way that they handle

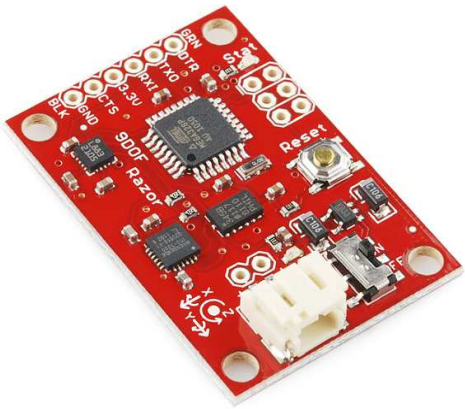

Figure 5.1: Sparkfun 9DOF Razor IMU (taken from [7])

updated error estimate which are the output of Kalman filters. In feedforward implementation, estimated error states are fed forward to currently estimated navigation variables that are calculated using IMU outputs and the inertial navigation equations. So no correction occurs when INS system produces new estimates of navigation variables before going into Kalman filter. In feedback implementation, new estimated error states is fed back to the INS to correct its estimated navigation variables. Therefore feedback system has the advantage of keeping the error states bounded, while unbounded error states could be observed in the feedforward implementation which may influence the linearization procedure negatively [35]. Because of these reasons, systems are designed in feedback configurations in this thesis. The feedback configuration for the linearized error state Kalman filter and extended Kalman filter version are depicted in Figure 5.3 and Figure 5.2 respectively.

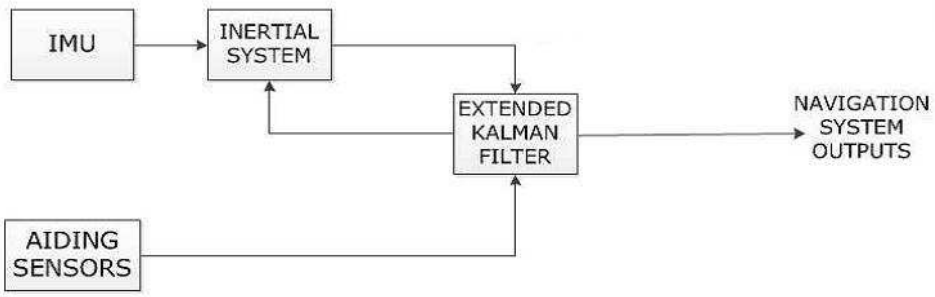

Figure 5.2: Feedback extended Kalman filter configuration

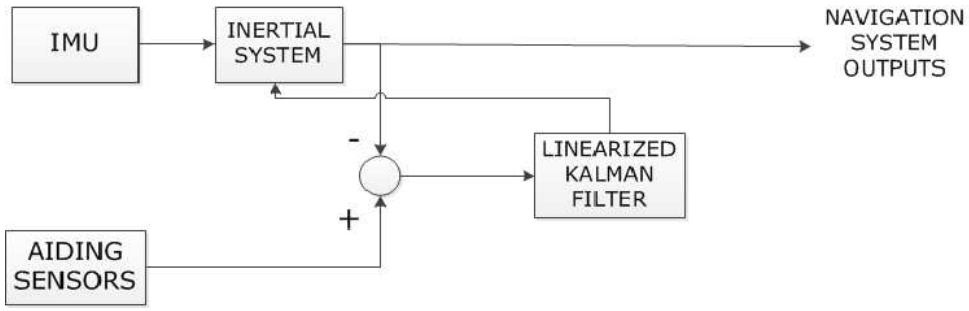

Figure 5.3: Feedback linearized error state Kalman filter configuration

The steps, that are performed in the design process, are described in the following sections. These steps are crucial to design a convenient navigation system. If any of these steps are misevaluated, significant errors could be observed in the system output.

5.1 Allan Variance Results

In order to work with a proper model of the IMU sensors, error coefficients must be identified using the Allan Variance method described in Section 2.5.1. In the error model of gyroscope and accelerometer, Angle / Velocity Random Walk, Bias Instability and Rate / Acceleration Random Walk errors are considered as it is stated in Equations 2.18 and 2.19. Therefore, the parts of Allan Variance plot related to these noise terms are evaluated.

For the implementation, approximately 20 hours data of stationary accelerometer and gyroscope at a sampling rate of 50Hz is recorded. Then the algorithm descriped in Section 2.5.1 is implemented in MATLAB. Since triple axis accelerometer and gyroscope are used, each axis must be evaluated seperately assuming that they are independent of each other. So there are different error coefficients for each axis of accelerometer and gyroscope. The resulting Allan Variance plots of gyroscope and accelerometer are given in Figure 5.4 and 5.5, respectively.

The error parameters can be found by using the formulas given in Table 2.2. Noise coefficient is determined by adjusting the coefficient such that a straight line with a fixed slope overlaps the Allan Variance plot. Figure 5.6, 5.7, 5.8, 5.9, 5.10 and 5.11 show the Allan Variance plots of x, y and z axes of gyroscope and accelerometer with three different error lines respectively when the coefficients of ARW, RRW and bias instability errors are correctly adjusted. The

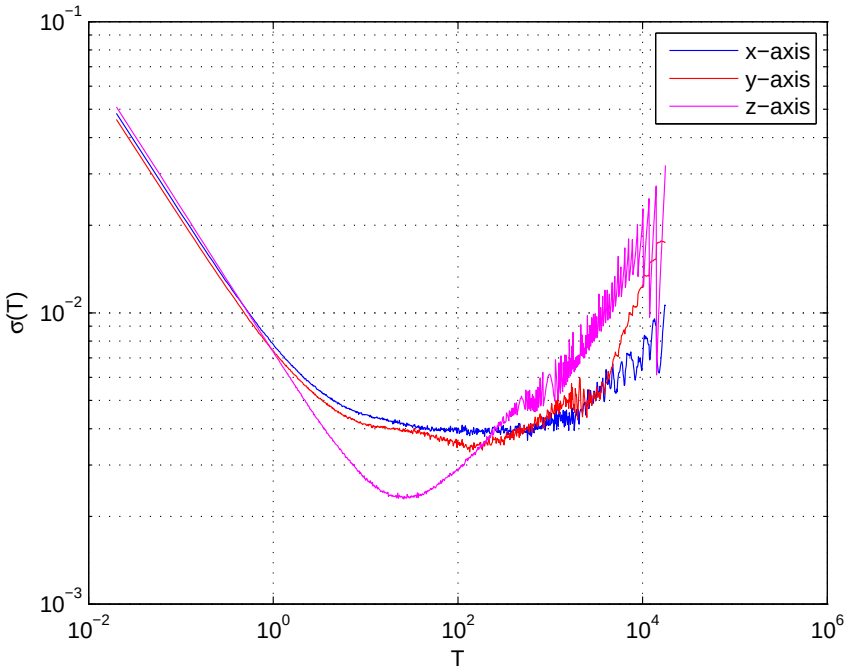

Figure 5.4: Allan variance plot of triple axis ITG-3200 Gyroscope

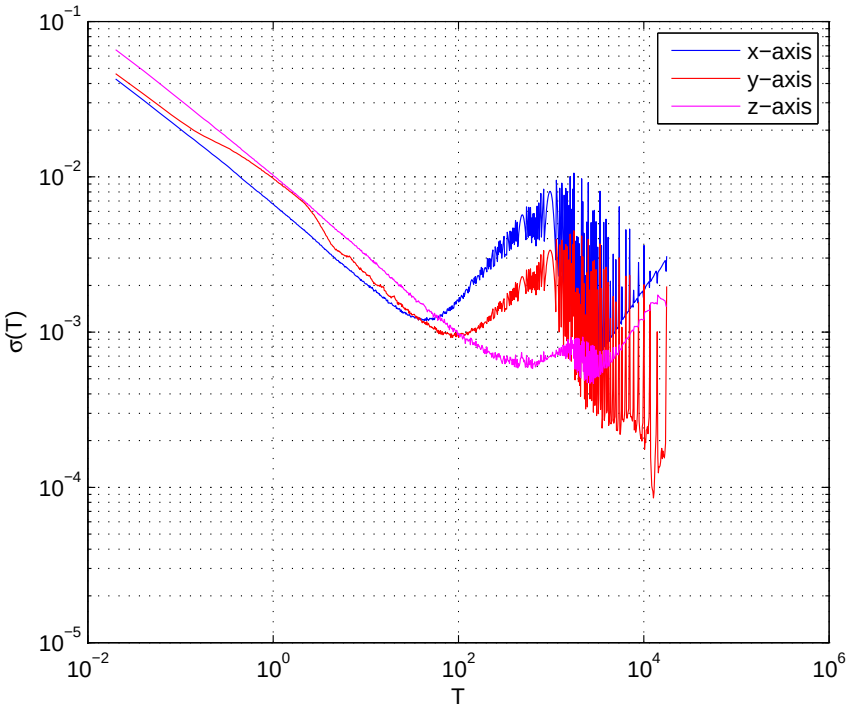

Figure 5.5: Allan variance plot of triple axis ADXL345 Accelerometer

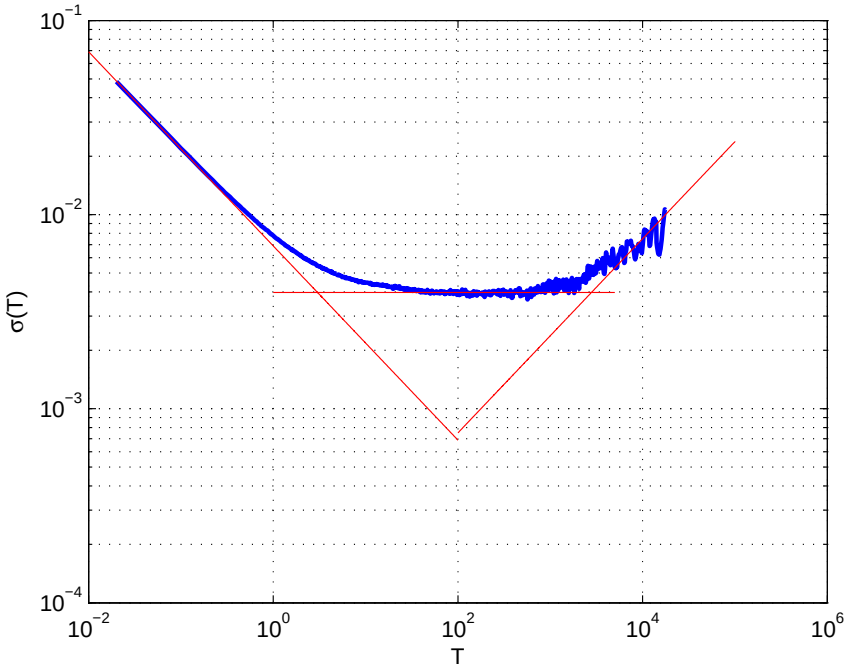

Figure 5.6: Allan variance plot of gyroscope X-axis with error lines

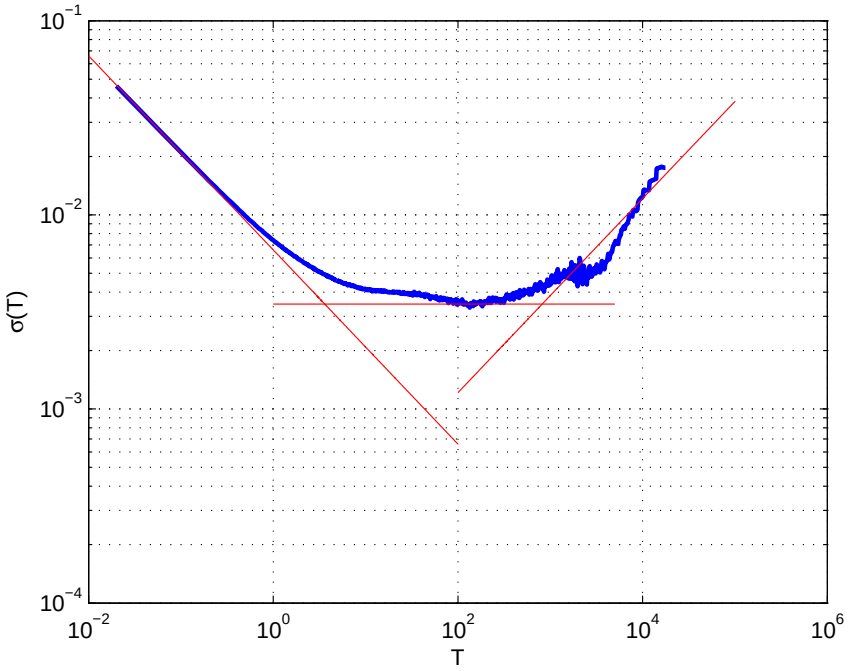

Figure 5.7: Allan variance plot of gyroscope Y-axis with error lines

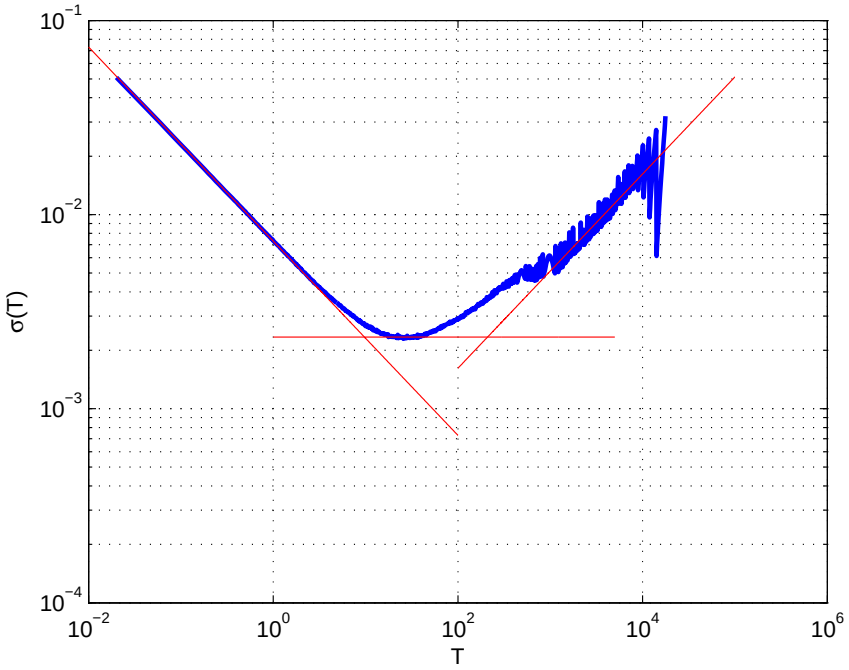

Figure 5.8: Allan variance plot of gyroscope Z-axis with error lines

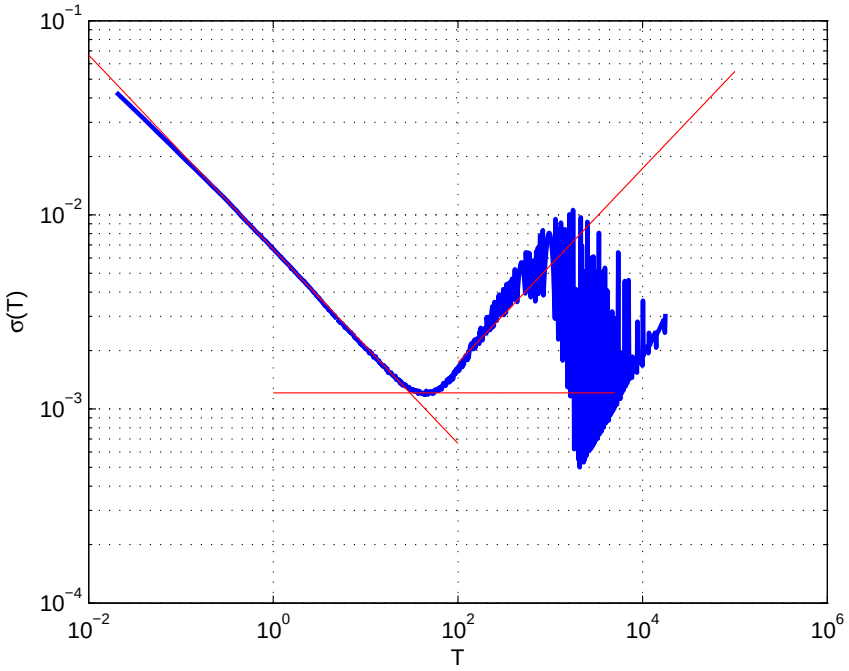

Figure 5.9: Allan variance plot of accelerometer X-axis with error lines

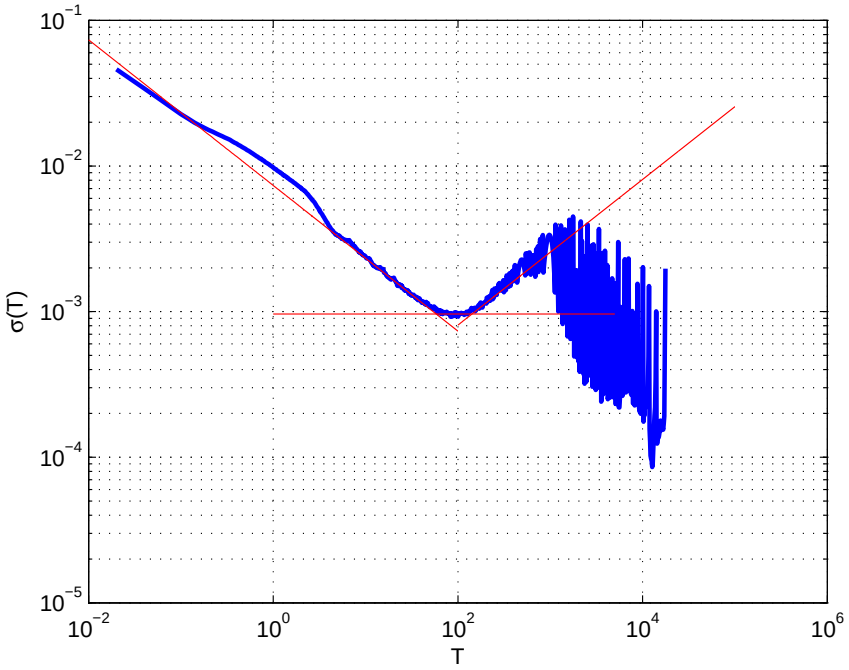

Figure 5.10: Allan variance plot of accelerometer Y-axis with error lines

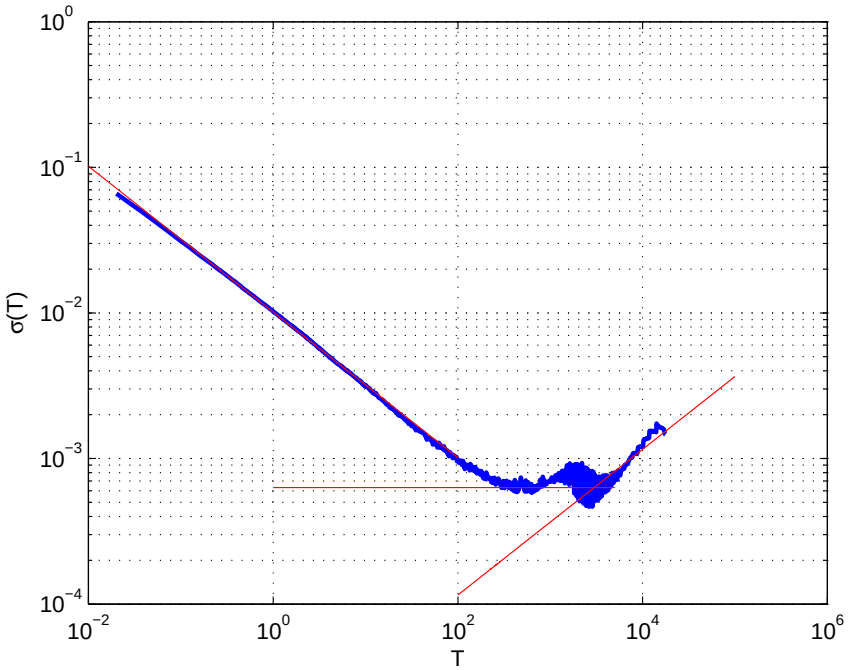

Figure 5.11: Allan variance plot of accelerometer Z-axis with error lines

error coefficients for gyroscope and accelerometer that will be used in the Kalman Filter is obtained by applying the method explained above. The list of these coefficients is given in Table 5.1.

Table 5.1: Error coefficients of IMU sensors

Sensor	**Angle/Velocity Random Walk - N**	**Bias Instability - B**	**Rate/Acceleration Random Walk - K**
Gyroscope X-axis	0.0069	0.00597	0.00013
Gyroscope Y-axis	0.0066	0.00522	0.00021
Gyroscope Z-axis	0.0073	0.00352	0.00028
Accelerometer X-axis	0.00666	0.00182	0.0003
Accelerometer Y-axis	0.00736	0.00145	0.00014
Accelerometer Z-axis	0.01014	0.00095	0.00002

These error coefficients must be identified reliably as far as possible. Because these coefficients form the error covariance matrix (process noise) of Kalman Filter which is significant for the operation of designed navigation systems.

5.2 Magnetometer Calibration

Magnetometer is the critical aiding sensor for the calculation of heading angle in the navigation systems but it is vulnerable to magnetic disturbances caused by different metals, electric currents etc. around itself. Effect of these disturbances can be eliminated by using the method discribed in Section 3.1.1, assuming that the placement of the magnetometer and the other items do not change during the operation time.

As it is described before, the calibration is handled at three axes instead of only X-Y axis. Therefore magnetometer is rotated arbitrarily about its three axes to collect data for different attitude and heading angles. This collected data is given as an input to the algorithm which is also implemented in MATLAB. This algorithm actually is not a part of the navigation

algorithm. It is only used to determine the calibration parameters of the magnetometer at a certain configuration. When it is run once, resulting calibration parameters are stored and applied to the rest of the magnetometer data according to formula given in Equation 3.4.

Uncalibrated and calibrated magnetometer data are shown in Figure 5.12 in three dimensional plot. It is expected that uncalibrated data forms an eliptical shape because of the soft iron effect. Also the center of the shape is away from the origin due to the hard iron effect. The algorithm produces the parameters that reforms this eliptical shape into a more smooth spherical one which is centered at the origin. Although the elipsoidal shape of uncalibrated magnetometer data is not obvious in Figure 5.12, the offset from the origin can be observed explicitly.

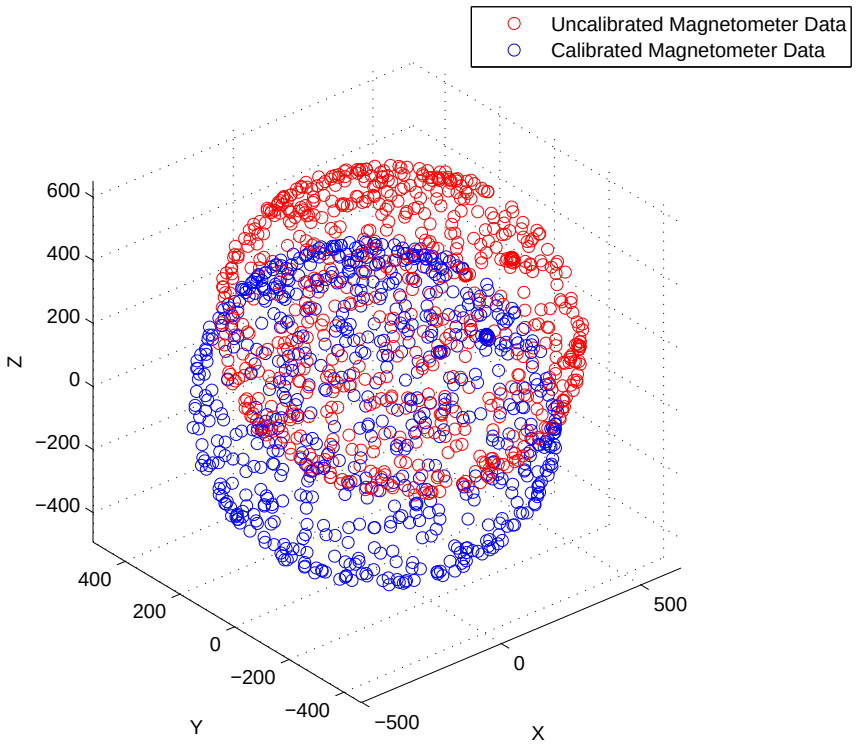

Figure 5.12: Uncalibrated and calibrated magnetometer data

After hard and soft iron calibrated data is obtained, tilt calibration procedure must be followed which is explained in Section 3.1.1.3. This calibration is performed online since the current attitude angles are required. Since Equations 3.6 and 3.7 contains trigonometric functions,

singularities may occur at certain angles. In order to get rid of these singularities, accelerometer outputs can be substituted for trigonometric functions. The equivalents of these trigonometric functions in terms of accelerometer output is explained in Section 5.3 in detail. So the final form of tilt calibration and heading calculation from hard and soft iron calibrated magnetometer data is given in Equation 5.1.

$$\psi_{magneto} = \text{atan2}\left(\frac{9.81(a_z\ mag_y^{calibrated} - a_y\ mag_z^{calibrated})}{(a_y^2 + a_z^2)\ mag_x^{calibrated} - a_x\ a_y\ mag_y^{calibrated} - a_x\ a_z\ mag_z^{calibrated}}\right) \quad (5.1)$$

In Equation 5.1, a_x, a_y and a_z are the accelerometer outputs at three different axes, $mag_x^{calibrated}$, $mag_y^{calibrated}$ and $mag_z^{calibrated}$ denote the hard and soft iron calibrated magnetometer outputs at three different axes. Also atan2 function is used for the inverse tangent in order to determine the correct quadrant.

5.3 Calculation of Tilt Angles Using Accelerometer Outputs

Tilt angles refer to roll and pitch angles of body frame with respect to navigation frame in the navigation systems. These angles are especially important for the frame transformations. In order to determine the amount of the these angles correctly, accelerometer could be used for measurement sensor. By combining the direction cosine matrix and gravitational acceleration roll and pitch angles can be derived just by using the low cost, MEMS accelerometers [36].

When vehicle rotates about its x or y axis, gravitational acceleration measured on z-axis of accelerometer is scattered also to x and y axis. Using the amount of this scattering on three axes of accelerometer, the roll and pitch angle can be calculated. Briefly, the projection of the gravitational acceleration on the accelerometer axes gives critical information about tilt angles of the vehicle. Figure 5.13 illustrates the projection of gravitational acceleration with red arrows on body axes for roll and pitch rotations.

When there is no external force applied on the vehicle, accelerometer only measures the plumb-bob gravity according to Equation 2.10. It is unfeasible to observe the coriolis effect of Earth rotation on accelerometer, since ADXL345 is a low cost MEMS accelerometer. Therefore, coriolis effect of Earth rotation can be neglected, i.e. $w_{ie}^n = 0$. Thus only the

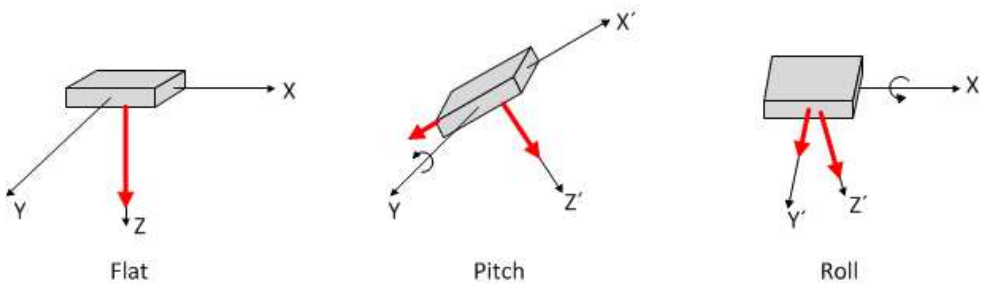

Figure 5.13: Projection of gravitational acceleration on body axes for roll and pitch rotations

gravitational component is left which is also given in Equation 5.2.

$$g_{ib}^{b} = R_{n}^{b}\, g_{ib}^{n} = \begin{bmatrix} -9.81\sin(\theta) \\ 9.81\sin(\phi)\cos(\theta) \\ 9.81\cos(\phi)\cos(\theta) \end{bmatrix} \tag{5.2}$$

Then when no external force applied, the output of the accelerometer become

$$a_{ib}^{b} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} 9.81\sin(\theta) \\ -9.81\sin(\phi)\cos(\theta) \\ -9.81\cos(\phi)\cos(\theta) \end{bmatrix} \tag{5.3}$$

Using basic algebra and trigonometric identities, sine and cosine function of the roll and pitch angles can be extracted from Equation 5.3. These functions are given in Equations 5.4, 5.5, 5.6 and 5.7.

$$\sin(\theta) = \frac{a_x}{9.81} \tag{5.4}$$

$$\cos(\theta) = \frac{\sqrt{a_y^2 + a_z^2}}{9.81} \tag{5.5}$$

$$\sin(\phi) = \frac{-a_y}{\sqrt{a_y^2 + a_z^2}} \tag{5.6}$$

$$\cos(\phi) = \frac{-a_z}{\sqrt{a_y^2 + a_z^2}} \tag{5.7}$$

Then by making use of the inverse tangent function, roll and pitch angle can be derived as in

Equations 5.8 and 5.9.

$$\phi = \tan^{-1}\left(\frac{a_y}{a_z}\right) \tag{5.8}$$

$$\theta = \tan^{-1}\left(\frac{a_x}{\sqrt{a_y^2 + a_z^2}}\right) \tag{5.9}$$

Instead of $\tan^{-}1$ function, $atan2(\frac{y}{x})$ function is used in the MATLAB© implementation of this method. $atan2(y, x)$ function takes into account the sign of y and x so that the quadrant of the angle is correctly obtained. Because of this property, Equation 5.8 is modified considering body frame axes and rotations depicted in Figure 2.5. This modification does not change the magnitude of the angle, it does only place the angle in correct quadrant. This modification is given in Equation 5.10.

$$\phi = \text{atan2}\left(\frac{-\text{a}_\text{y}}{-\text{a}_\text{z}}\right) \tag{5.10}$$

Final forms of tilt angles in terms of accelerometer output is summarized in Table 5.2.

Table 5.2: Calculation of roll and pitch angles in terms of accelerometer outputs

Roll	$\phi_{acc} = \text{atan2}\left(\frac{-\text{a}_\text{y}}{-\text{a}_\text{z}}\right)$
Pitch	$\theta_{acc} = \text{atan2}\left(\frac{\text{a}_\text{x}}{\sqrt{a_y^2+a_z^2}}\right)$

Up to now in this section, it is assumed that the vehicle is not exposed to any external forces that causes acceleration of the body. However, this assumption is not valid for all of the operation time. The vehicle may need to slow down or speed up depending on its mission. When the vehicle is making these kinds of motions, the accelerometer also measures these acceleration or deceleration activities plus the gravitational acceleration. Therefore, the calculation of tilt angles with the method presented in this section will not be correct. This problem is considered further in Section 5.4.

5.4 Adaptive Filter Gain

In this work, accelerometer is also used as an aiding sensor for roll and pitch angles calculation. However, this aid is not reliable when the vehicle makes accelerated motion. The

derivation of the tilt angles using accelerometer data in Section 5.3 shows that tilt angles depend directly on gravitational acceleration measured by the accelerometer. So any extra acceleration on vehicle motion causes miscalculation of the tilt angles. Actually, the gravitational acceleration($\sim 9.81 m/sn^2$) is far bigger than the acceleration of the vehicle that it could reach. This problem is nevertheless worth investigating in order to have a stable and accurate navigation system.

When the vehicle moves with positive or negative acceleration, it is challenging to get rid of the errors in the tilt angle calculation because of difficulty to distinguish between the gravitational acceleration and vehicle acceleration in the accelerometer output. Therefore the most feasible and simple method is to impose an adaptation depending on the total acceleration of the vehicle. The main idea of this method is to reduce the importance of the aiding sensor (accelerometer tilt measurements) when the vehicle accelerates or decelerates so that filter relies more on the dynamic model of the system. Not only the tilt angles are affected by accelerated motions, but also magnetometer heading angle calculation is affected due to tilt calibration given in Equation 5.1. Because of these reason, adaptive Kalman filter is used in this thesis work.

In the literature, different methods exist for the adaptive Kalman filter implementations. Among them, manipulating the error covariance matrices is the easiest way to handle it. So, adaptive Kalman filter can be implemented by applying a scale factor $S < 1$ to a priori estimate covariance matrix calculation to deliberately decrease the weight given to measurements from aiding sensors [37]. Adaptive a priori estimate covariance matrix calculation is carried out in the Kalman Filter algorithm using a scale factor S as given in Equation 5.11.

$$P_{k+1|k} = S\left(F_k P_{k|k} F_k^T + Q_d\right) \tag{5.11}$$

Instead of manipulating a priori estimate covariance matrix, it is more likely to scale the measurement error covariance matrix which is used in the Kalman gain calculation step since acceleration directly increases the amount of error in the measurements. Therefore, the scale factor can be applied to the measurement error covariance matrix in the innovation covariance calculation stage before the Kalman gain calculation as in Equation 5.12.

$$R_k = H_k P_{k+1|k} H_k^T + S\ R_d \tag{5.12}$$

This scaling is designed as a diagonal matrix since measurements could be affected in differ-

ent rates. Thus, the measurement error values are multiplied with different rates when any disturbing case occurs. S is determined by thresholding total acceleration of the vehicle, x-axis acceleration and high rotational speed. The threshold values are decided experimentally after experimenting over different cases of the application.

Any acceleration caused by water flow or other effects could disturb the measurements, especially roll and pitch angles. So, instead of observing the acceleration in y and z axes, it is more beneficial to check the total acceleration of the vehicle because both extra accelerations in y-z axes and the vehicle's true acceleration can be observed. Also it's not required to substract the projection of gravitational acceleration in y and z axes due to the pitch and roll angles. Total vehicle acceleration is simply calculated as in Equation 5.13.

$$\alpha = \sqrt{a_x^2 + a_y^2 + a_z^2} - 1 \text{ g} \tag{5.13}$$

The vehicle movement can be classified in three groups based on total vehicle acceleration given in Equation 5.13 [38]:

- **Non-acceleration mode**

  In this mode, acceleration of the vehicle is so small such that $|\alpha| < 0.05$ g where $|\alpha|$ is the absolute value of the vehicle acceleration. So the scale factor is small which means that there is no deliberate intervention in the Kalman Filter process, specifically in the a measurement covariance matrix. At this mode, system corrects its estimates based on incoming measurements from aiding sensors since aiding sensors works with a high precision. Generally this mode is valid when the vehicle has a constant speed.

- **Low-acceleration mode**

  In this mode, the vehicle moves with an acceptable acceleration so that there is a little uncertainty in the measurements of the aiding sensors. This acceptable threshold is defined as $0.5\text{g} > |\alpha| \geq 0.05\text{g}$. At this mode, scale factor is assigned a value larger than 1 so that importance of the measurements are slightly reduced by increasing the measurement error covariance matrix.

- **High-acceleration mode**

  In this mode, vehicle is in high dynamics and the aiding sensors are far from being accurate. The threshold for the vehicle acceleration is assigned as $|\alpha| \geq 0.5\text{g}$. When the vehicle acceleration is above the this threshold value, scale factor is too large which

implies that the measurements coming from aiding sensors are not taken into consideration due to the large measurement error covariance matrix. At this mode, the system works according to system model and gyroscope error model.

The accelerometer outputs are too noisy if the vehicle makes vibratory fast moves. So sometimes just observing the total acceleration does not give clear results. Therefore, it is worthwhile to check the x-axis acceleration also with the total vehicle acceleration. If the value of the x-axis acceleration is above the threshold value, then the measurements error covariance values will be scaled up so that measurement importance is reduced in the Kalman Filter.

The last condition is the high rotational speed. When the vehicle rotates at a high rate, it is far more accurate to detect the amount of this rotation using gyroscope. So this rotation is measured using system mechanization equations and gyroscope outputs. Also, high rate rotation could cause changes in the accelerometer measurement because of the coriolis effect. These changes reflect as an error in the measurements. Coriolis effect indeed can be calibrated but meausuring this coriolis effect is difficult with MEMS sensors. Because of these reasons, high rotational speed detection seems to be the best option for the correct system implementation.

Considering these three conditions, adaptive system could be described by nine scenarios which are summarized in Table 5.3.

Diagonal values in the scale matrix are determined experimentally and optimized for the lowest RMS error. Scale factor matrix has different values for each kind of measurements since the effect of each condition gives different responses in the measurements.

This method is implemented as a function which has inputs from accelerometer and gyroscope and produces the scale factor matrix, S as an output.

5.5 System Model

The navigation system in this thesis work is designed for an underwater vehicle. Because of the complexity of the underwater environment and lack of the use of some aiding sensors, system is modelled in a different structure based on three assumptions. These assumptions are zero buoyancy, x-axis body velocity and offgrade MEMS sensor properties. These assumption are represented in the mechanization equations in this section.

Table 5.3: Scale factor matrix for adaptive system scenarios

Total Vehicle Acceleration	**X-Axis Acceleration**	**High Rotational Speed**	**Scale Factor Matrix**
Non-acceleration	Below Threshold	Below Threshold	Identity Matrix
Non-acceleration	Below Threshold	Above Threshold	Large Diagonal Values
Non-acceleration	Above Threshold	Below Threshold	Large Diagonal Values
Non-acceleration	Above Threshold	Above Threshold	Large Diagonal Values
Low-acceleration	Below Threshold	Below Threshold	Low Diagonal Values(> 1)
Low-acceleration	Below Threshold	Above Threshold	Large Diagonal Values
Low-acceleration	Above Threshold	Below Threshold	Large Diagonal Values
Low-acceleration	Above Threshold	Above Threshold	Large Diagonal Values
High-acceleration	x	x	Largest Diagonal Values(>> 1)

The underwater vehicle does not sink or surface when no external force is applied to it. It stays where it is left in the underwater when its propulsion system does not work. This means that buoyant force equals to the weight of the vehicle, i.e. the buoyancy is zero. This assumption is called as zero buoyancy which makes it easier to model the movement of the vehicle in navigation frame. If it is buoyancy is not zero, it will always move in the z-axis. Since very limited number of sensors are available in this system, it is hard to model this kind of motion in z-axis. A detailed model of gravitational force is required for the correct analysis. Therefore, in this work, the vehicle buoyancy is assumed to be zero in order to simplify the system model.

X-axis velocity assumption is the most critical assumption for the movement of the vehicle in this system. It is assumed that propulsion system of the underwater vehicle only thrust into x-axis of the vehicle in body frame. Other motions in y-axis and z-axis are neglected so that vehicle moves forward only in the direction of its attitude and heading angle. This assumption also neglects the drift in the y-axis and z-axis because of the water course. X-axis velocity assumption is compatible with the zero buoyancy assumption such that the vehicle does not change its position if it has zero buoyancy. If the vehicle moves with a high speed, x-

axis velocity assumption becomes more realistic. Moreover, this kind of assumption makes it easier to model the motion of the vehicle since there is no aiding sensor to measure the velocity of the vehicle directly. Because of these reasons, mechanization equations in Sections 2.6 and 2.7 are modified according to x-axis velocity assumption.

The last assumption is actually adopted due to the properties of MEMS IMU sensors. In this work, a low cost MEMS IMU sensor is used and its accuracy is not enough to measure all of the terms available in the mechanization equations given in Tables 2.3 and 2.4. MEMS gyroscopes are not able to sense the Earth rate and it is hard to observe the coriolis effect in MEMS accelerometers. They can only measure the primary rotations and accelerations. Therefore, these terms should be neglected in the mechanization equations and reformulate according to MEMS sensor properties. That is also why only the gravitational component is considered and coriolis effect is neglected in the plumb-bob gravity model given in Equation 2.23 in the previous calculations.

Mechanization equations are derived assuming that navigation frame does not rotate. This assumption is based on that the underwater vehicle does not go far away from its starting point. Moreover, since MEMS sensor is not able to detect the Earth rate, it could be assumed that $w_{in}^{n} = w_{ie}^{n} = 0$. This means that the navigation frame does not rotate with respect to any frame. So navigation frame (NED) can be expressed as a local frame that the vehicle does not go out of it, i.e. navigation frame can be defined as the inertial frame of this system. Thus the attitude angles, heading angle and position are calculated in the navigation frame. However, it is easier to express velocity in the body frame because of the x-axis velocity assumption.

Considering all of the statements above, nonlinear mechanization equations given in Section 2.6 are rewritten in Equations 5.14, 5.15 and 5.16 which will be used in the extended Kalman filter implementation.

$$\dot{p}_{nb}^{n} = R_{b}^{n}\, v_{nb}^{b} \tag{5.14}$$

$$\dot{v}_{nb}^{b(x)} = f_{ib}^{b(x)} + R_{n}^{b} g_{ib}^{n(x)} \tag{5.15}$$

$$\dot{\Theta} = T_{b}^{n}\, \omega_{nb}^{b} \tag{5.16}$$

Actually, Equations 5.14, 5.15 and 5.16 are the simplified version of the Equations given in Table 2.3 based on the system assumptions.

Nonlinear mechanization equations can be linearized using the perturbation method and error

models of MEMS accelerometer and gyroscope given in Section 2.5. The linearized error state mechanization equations for newly designed system are derived as follows:

Position Error Model

Let's define the correct position as p^n_{nb} and estimated position as $\hat{p}^n_{nb}$. Then perturbation equation and its derivative can be written as

$$\delta p = p^n_{nb} - \hat{p}^n_{nb} \tag{5.17}$$

$$\dot{\delta} p = \dot{p}^n_{nb} - \dot{\hat{p}}^n_{nb} \tag{5.18}$$

Using Equation 5.14 and 2.37, Equation 5.18 can be rewritten as follows:

$$\dot{\delta} p = R^n_b \, v^b_{nb} - \hat{R}^n_b \, \hat{v}^b_{nb} \tag{5.19}$$

$$= (I + \widetilde{\delta\Theta})\hat{R}^n_b(\hat{v}^b_{nb} + \delta v^b_{nb}) - \hat{R}^n_b \, \hat{v}^b_{nb} \tag{5.20}$$

$$= \hat{R}^n_b \, \hat{v}^b_{nb} + \hat{R}^n_b \, \delta v^b_{nb} + \widetilde{\delta\Theta}\hat{R}^n_b \, \hat{v}^b_{nb} + \underbrace{\widetilde{\delta\Theta}\hat{R}^n_b \, \delta v^b_{nb}}_{\text{small}} - \hat{R}^n_b \, \hat{v}^b_{nb} \tag{5.21}$$

$$= \hat{R}^n_b \, \delta v^b_{nb} + \widetilde{\delta\Theta}\hat{R}^n_b \, \hat{v}^b_{nb} \tag{5.22}$$

$$\dot{\delta} p = \hat{R}^n_b \, \delta v^b_{nb} - \widetilde{\hat{R}^n_b \, \hat{v}^b_{nb}}\delta\Theta \tag{5.23}$$

Velocity Error Model

The error state of velocity is derived using 'derivative of velocity equals acceleration' relation. Accelerometer output can be fully rewritten as in Equation 5.24 using Equations 2.10 and 2.18.

$$f^b = \ddot{p} - g^b_{nb} + a_{arw} + a_{fn} + a_{rrw} \tag{5.24}$$

The derivatives of estimate and real velocities can be identified as

$$\dot{v}^b_{nb} = \ddot{p}^b \tag{5.25}$$

$$\dot{\hat{v}}^b_{nb} = f^b + \hat{g}^b_{nb} \tag{5.26}$$

Using the error definition, the linearized error state for velocity can be derived as follows:

$$\dot{\delta v}_{nb}^{b} = \dot{v}_{nb}^{b} - \dot{\hat{v}}_{nb}^{b} \tag{5.27}$$

$$= \ddot{p}^{b} - (f^{b} + \hat{g}_{nb}^{b}) \tag{5.28}$$

$$= \ddot{p}^{b} - (\ddot{p}^{b} - g_{nb}^{b} + a_{arw} + a_{fn} + a_{rrw}) - \hat{g}_{nb}^{b} \tag{5.29}$$

$$= -a_{arw} - a_{fn} - a_{rrw} + \hat{R}_{n}^{b}(I - \widetilde{\delta\Theta}) g_{nb}^{n} - \hat{R}_{n}^{b}\, g_{nb}^{n} \tag{5.30}$$

$$= -\hat{R}_{n}^{b}\, \widetilde{\delta\Theta}\, g_{nb}^{n} - a_{arw} - a_{fn} - a_{rrw} \tag{5.31}$$

$$\dot{\delta v}_{nb}^{b} = \hat{R}_{n}^{b}\, \widetilde{g_{nb}^{n}}\, \delta\Theta - a_{arw} - a_{fn} - a_{rrw} \tag{5.32}$$

According to the x-axis velocity assumption, only x-axis component of the vehicle velocity is calculated and other components are ingored in the body frame such that $v_{nb}^{b(y)} = v_{nb}^{b(z)} = 0$. So Equation 5.32 is converted for the x-axis velocity representation as in Equation 5.33.

$$\dot{\delta v}_{nb}^{b(x)} = \hat{R}_{n}^{b}\, \widetilde{g_{nb}^{n}}\, \delta\Theta\,(1,:) - a_{arw}^{x} - a_{fn}^{x} - a_{rrw}^{x} \tag{5.33}$$

In Equation 5.33, $\hat{R}_{n}^{b}\, \widetilde{g_{nb}^{n}}\, \delta\Theta\,(1,:)$ means the first row of the matrix $\hat{R}_{n}^{b}\, \widetilde{g_{nb}^{n}}\, \delta\Theta$.

Attitude Error Model

The attitude error state model is based on Equation 2.29. The derivatives of estimate and real direction cosine matrix can be identified as

$$\dot{R}_{b}^{n} = R_{b}^{n}\, \widetilde{\omega_{nb}^{b}} \tag{5.34}$$

$$\dot{\hat{R}}_{b}^{n} = \hat{R}_{b}^{n}\, \widetilde{\hat{\omega}_{nb}^{b}} \tag{5.35}$$

Then the derivative of the error state of direction cosine matrix is calculated as

$$\dot{\delta R}_{b}^{n} = \dot{R}_{b}^{n} - \dot{\hat{R}}_{b}^{n} \tag{5.36}$$

$$= R_{b}^{n}\, \widetilde{\omega_{nb}^{b}} - \hat{R}_{b}^{n}\, \widetilde{\hat{\omega}_{nb}^{b}} \tag{5.37}$$

$$= (I + \widetilde{\delta\Theta})\hat{R}_{b}^{n}\widetilde{\omega_{nb}^{b}} - \hat{R}_{b}^{n}\, \widetilde{\hat{\omega}_{nb}^{b}} \tag{5.38}$$

$$= \hat{R}_{b}^{n}(\widetilde{\omega_{nb}^{b}} - \widetilde{\hat{\omega}_{nb}^{b}}) + \delta\Theta\hat{R}_{b}^{n}\widetilde{\omega_{nb}^{b}} \tag{5.39}$$

Also the derivative of the error state of direction cosine matrix can be calculated directly as follows.

$$\delta R_{b}^{n} = R_{b}^{n} - \hat{R}_{b}^{n} \tag{5.40}$$

$$= (I + \widetilde{\delta\Theta})\hat{R}_{b}^{n} - \hat{R}_{b}^{n} \tag{5.41}$$

$$= \widetilde{\delta\Theta}\, \hat{R}_{b}^{n} \xrightarrow{derivative} \dot{\delta R}_{b}^{n} = \dot{\widetilde{\delta\Theta}}\hat{R}_{b}^{n} + \widetilde{\delta\Theta}\, \dot{\hat{R}}_{b}^{n} \tag{5.42}$$

$$\dot{\delta R}_{b}^{n} = \dot{\widetilde{\delta\Theta}}\hat{R}_{b}^{n} + \widetilde{\delta\Theta}\, \hat{R}_{b}^{n}\, \widetilde{\hat{\omega}_{nb}^{b}} \tag{5.43}$$

Combining Equation 5.39 and Equation 5.43 result as

$$\dot{\widetilde{\delta\Theta}}\hat{R}_b^n + \widetilde{\delta\Theta}\,\hat{R}_b^n\,\widetilde{\hat{\omega}_{nb}^b} = \hat{R}_b^n(\widetilde{\omega_{nb}^b} - \widetilde{\hat{\omega}_{nb}^b}) + \widetilde{\delta\Theta}\hat{R}_b^n\widetilde{\omega_{nb}^b} \tag{5.44}$$

$$\dot{\widetilde{\delta\Theta}}\hat{R}_b^n = -\widetilde{\delta\Theta}\,\hat{R}_b^n\,\widetilde{\hat{\omega}_{nb}^b} + \hat{R}_b^n(\widetilde{\omega_{nb}^b} - \widetilde{\hat{\omega}_{nb}^b}) + \widetilde{\delta\Theta}\hat{R}_b^n\widetilde{\omega_{nb}^b} \tag{5.45}$$

$$\dot{\widetilde{\delta\Theta}} = -\widetilde{\delta\Theta}\,\hat{R}_b^n\,\widetilde{\hat{\omega}_{nb}^b}\hat{R}_n^b + \hat{R}_b^n(\widetilde{\omega_{nb}^b} - \widetilde{\hat{\omega}_{nb}^b})\hat{R}_n^b + \widetilde{\delta\Theta}\hat{R}_b^n\widetilde{\omega_{nb}^b}\hat{R}_n^b \tag{5.46}$$

$$\dot{\widetilde{\delta\Theta}} = -\underbrace{\widetilde{\delta\Theta}}_{\text{small}}\,\hat{R}_b^n\,\underbrace{(\widetilde{\hat{\omega}_{nb}^b} - \widetilde{\omega_{nb}^b})}_{\text{small}}\hat{R}_n^b + \hat{R}_b^n(\widetilde{\omega_{nb}^b} - \widetilde{\hat{\omega}_{nb}^b})\hat{R}_n^b \tag{5.47}$$

$$\dot{\widetilde{\delta\Theta}} = \hat{R}_b^n(\widetilde{\omega_{nb}^b} - \widetilde{\hat{\omega}_{nb}^b})\hat{R}_n^b \xrightarrow{in\ vector\ form} \dot{\delta\Theta} = \hat{R}_b^n\delta\omega_{nb}^b \tag{5.48}$$

In Equation 5.48, $\delta\omega_{nb}^b$ implies the gyroscope error which can be formulated using Equation 2.19 as in Equation 5.49.

$$\begin{aligned}\delta\omega_{nb}^b &= \omega_{nb}^b - \hat{\omega}_{nb}^b \\ &= \omega_{nb}^b - (\omega_{nb}^b + w_{arw} + w_{fn} + w_{rrw}) \\ &= -w_{arw} - w_{fn} - w_{rrw}\end{aligned} \tag{5.49}$$

Finally, the attitude error equation is defined in Equation 5.50 by substituting Equation 5.49 into Equation 5.48.

$$\dot{\delta\Theta} = -\hat{R}_b^n\,w_{arw} - \hat{R}_b^n\,w_{fn} - \hat{R}_b^n\,w_{rrw} \tag{5.50}$$

Linearized mechanization error equations for this thesis work is summarized in Table 5.4. These equations will be used for the state space model of the linearized error state Kalman filter implementation.

Table 5.4: Linearized mechanization equations for newly designed system

Position Equation	$\dot{\delta}p = \hat{R}_b^n\,\delta v_{nb}^b - \widetilde{\hat{R}_b^n\,\hat{v}_{nb}^b}\delta\Theta$
Velocity Equation	$\dot{\delta v}_{nb}^{b(x)} = \hat{R}_n^b\,\widetilde{g_{nb}^n}\,\delta\Theta\,(1,:) - a_{arw}^x - a_{fn}^x - a_{rrw}^x$
Attitude Equation	$\dot{\delta\Theta} = -\hat{R}_b^n\,w_{arw} - \hat{R}_b^n\,w_{fn} - \hat{R}_b^n\,w_{rrw}$

5.6 Continuous Angle Generation

Attitude and heading angles are calculated using *atan*2 function in Equations 5.1, 5.10 and 5.9. This function calculates the inverse tangent between $[-\pi, \pi]$. Therefore full rotations

(360°) cannot be detected in any axes. Estimated attitude angles are calculated by integrating gyroscope outputs so that estimated angles are piecewise continous. Therefore, a simple algorithm is developed to make the results of the *atan*2 function in continuous form. Equation 5.51 and 5.52 describe this algorithm.

$$k_{opt} = \underset{k}{\operatorname{argmin}} |\Theta_{n-1} - (\Theta_n^{aid} + 2\pi k)| \qquad k : \text{integer}; (... -2, -1, 0, 1, 2, ...) \tag{5.51}$$

$$\Theta_n^{cont} = \Theta_n^{aid} + 2\pi k_{opt} \tag{5.52}$$

In Equation 5.51, Θ_{n-1} previous attitude or heading angle which is calculated at time $n-1$, Θ_n^{aid} is the output of the aiding sensor at time n and k is the set of positive and negative integers. By comparing these two angles, k value is calculated. Actually, the coefficient k implies the number of 180° rotations at respective axes for roll, pitch and heading angles. Using this optimal coefficient k_{opt}, outputs of aiding sensor generated by atan2 function become continuous in Equation 5.52.

The drawback of this method is chosing the size of the set k in Equation 5.51. If the set contains too many consecutive integers, more rotations can be detected. So, the interval of continuity will be that long but the computational time will be longer. Therefore, a proper search set should be determined. If the vehicle rotates too much due to its dynamic stability or mission definition, a large set search k is required. If the vehicle does not rotate frequently, a smaller set with a safety margin can be used.

Continuous angle generation block is crucial for the system in this work since it makes to observe the critical movements of the vehicle that could damage its stability. Thus, this algorithm is applied to all of the aiding sensor outputs.

5.7 State Space Model of Kalman Filters and Overall System Structure

In order to implement extended Kalman filter (EKF) and linearized error state Kalman filter, state space representation must be constructed according to mechanization equations given in Section 5.5. State space representation will be different for these two implementations since one of them uses linearized error state mechanization equations and the other one uses nonlinear mechanization equations. When state space model is constructed, Kalman filtering procedures can be applied as given in Chapter 4. State space models of these two different implementations are explained in the following subsections.

5.7.1 Linearized Error State Kalman Filter Implementation

State space model is formed according to the equations given in Table 5.4 and IMU error models are explained in Section 2.5. Position error, velocity error, attitude and heading error, rate random walk error for accelerometer and gyroscope and bias instability for accelerometer and gyroscope are chosen as the components of state. State space representation for these error states is given in Equations 5.53 and 5.54.

$$\begin{bmatrix} \dot{\delta p} \\ \dot{\delta v^b_{nb}} \\ \dot{\delta\Theta} \\ \dot{a_{rrw}} \\ \dot{w_{rrw}} \\ \dot{a_{fn}} \\ \dot{w_{fn}} \end{bmatrix} = \begin{bmatrix} 0_{3\times3} & \hat{R}^n_b & -\widetilde{\hat{R}^n_b \hat{v}^b_{nb}} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & A_{23} & A_{24} & 0_{3\times3} & A_{26} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & -\hat{R}^n_b & 0_{3\times3} & -\hat{R}^n_b \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & -\beta_a & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & -\beta_g \end{bmatrix} \begin{bmatrix} \delta p \\ \delta v^b_{nb} \\ \delta\Theta \\ a_{rrw} \\ w_{rrw} \\ a_{fn} \\ w_{fn} \end{bmatrix}$$
$$+ \begin{bmatrix} 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ N^{(x)}_a & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & \hat{R}^n_b N_g & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & K_a & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & K_g & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & \beta_a B_a & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & \beta_g B_g \end{bmatrix} \begin{bmatrix} v^{arw}_a \\ v^{arw}_g \\ v^{rrw}_a \\ v^{rrw}_g \\ v^{bias}_a \\ v^{bias}_g \end{bmatrix} \tag{5.53}$$

$$y = \begin{bmatrix} 0 & 0 & 1 & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} \\ 0_{1\times3} & 0_{1\times3} & 1 & 0 & 0 & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} \\ 0_{1\times3} & 0_{1\times3} & 0 & 1 & 0 & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} \\ 0_{1\times3} & 0_{1\times3} & 0 & 0 & 1 & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} \end{bmatrix} \begin{bmatrix} \delta p \\ \delta v^b_{nb} \\ \delta\Theta \\ a_{rrw} \\ w_{rrw} \\ a_{fn} \\ w_{fn} \end{bmatrix} + S_v \tag{5.54}$$

The open form of A_{23}, A_{24} and A_{26} are as follows:

$$A_{23} = \begin{bmatrix} & \hat{R}_n^b \, \widetilde{g_{nb}^n} \, \delta\Theta \, (1,:) & \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{5.55}$$

$$A_{24} = A_{26} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{5.56}$$

Also $N_a^{(x)}$ is a 3-by-3 matrix whose first element is the x component of angular random walk error of accelerometer and the other elements are all zero. Open form of $N_a^{(x)}$ is given in Equation 5.57. K_a, K_g, β_a, β_g, B_a and B_g are the diagonal matrices of rate random walk coefficients, reciprocals of time constants and bias instability coefficients respectively for accelerometer and gyroscope. These coefficients are determined before the operation as it is declared in Section 5.1.

$$N_a^{(x)} = \begin{bmatrix} N_a^x & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{5.57}$$

For each new data, state space matrices are recalculated and used in the Kalman filter in order to estimate the amount of error. Then these estimated error are given as a feedback for the correct position, velocity and heading calculations. Complete block diagram of the system with a linearized error state Kalman filter is shown in Figure 5.14.

5.7.2 Extended Kalman Filter Implementation

For the extended Kalman filter (EKF) implementation, Jacobian of nonlinear mechanization equations given in Equations 5.14, 5.15 and 5.16 are used for the state state representation. The reason of using the Jacobian matrix calculation is to linearize the mechanization equations in the most recent state. So the Jacobian of the equations is calculated first and then its discrete equivalent matrices are generated in the implementation. Although the nonlinear equations are written for the real states (position, velocity, attitude and heading), the Jacobian of them

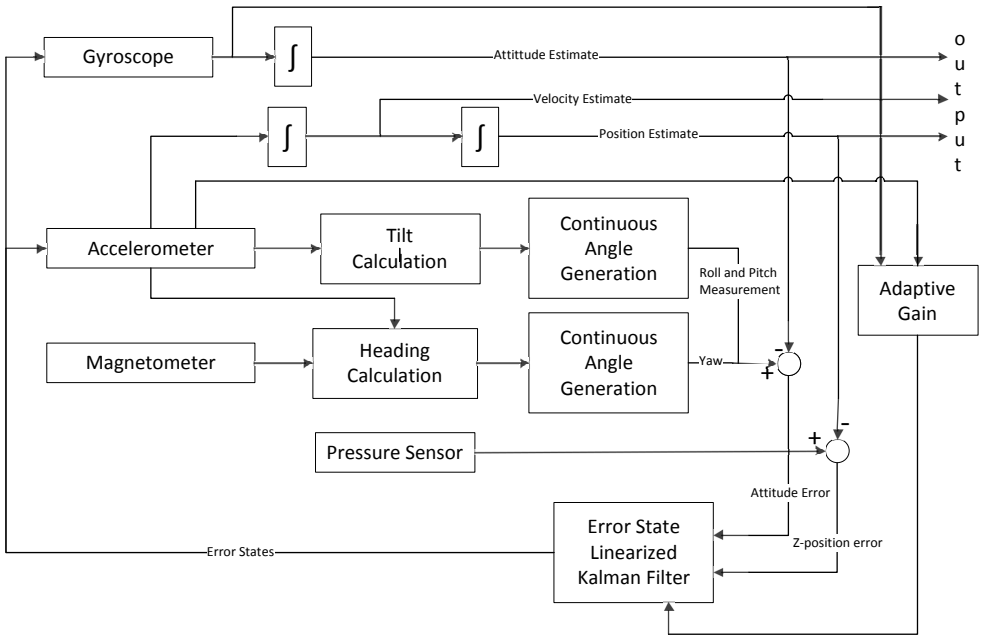

Figure 5.14: Block diagram of the system with linearized error state Kalman filter

are also valid for the error states. The proof is given below [39].

$$\dot{x}(t) = f(x(t), u, t) + w(t) \tag{5.58}$$

$$x(t) = \hat{x}(t) + \delta x(t) \tag{5.59}$$

Combine Equations 5.58 and 5.59

$$\dot{\hat{x}}(t) + \dot{\delta} x(t) = f(\hat{x}(t) + \delta x(t), u, t) + w(t) \tag{5.60}$$

Using Taylor Series expansion and omiting second and higher terms gives

$$\dot{\hat{x}}(t) + \dot{\delta} x(t) \approx f(\hat{x}(t), u, t) + \frac{\partial f}{\partial x} \mid_{x=\hat{x}} \delta x + w(t) \tag{5.61}$$

Since $\dot{\hat{x}}(t) = f(\hat{x}(t), u, t)$, it's concluded that

$$\dot{\delta} x(t) = \frac{\partial f}{\partial x} \mid_{x=\hat{x}} \delta x + w(t) \tag{5.62}$$

As Equation 5.62 indicates that error states equations can be extracted using Jacobian of non-linear equations. So, the indirect extended Kalman filter implementation is based on this property. Finally, the state space representation for the EKF is given in Equations 5.63 and 5.64 that involves the Jacobian matrix of nonlinear equations.

$$\begin{bmatrix} \dot{\delta p} \\ \dot{\delta v_{nb}^b} \\ \dot{\delta\Theta} \\ \dot{a_{rrw}} \\ \dot{w_{rrw}} \\ \dot{a_{fn}} \\ \dot{w_{fn}} \end{bmatrix} = \begin{bmatrix} 0_{3\times3} & R_b^n & A_{13} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & A_{23} & A_{24} & 0_{3\times3} & A_{26} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & A_{33} & 0_{3\times3} & -T_b^n & 0_{3\times3} & -T_b^n \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & -\beta_a & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & -\beta_g \end{bmatrix} \begin{bmatrix} \delta p \\ \delta v_{nb}^b \\ \delta\Theta \\ a_{rrw} \\ w_{rrw} \\ a_{fn} \\ w_{fn} \end{bmatrix}$$

$$+ \begin{bmatrix} 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ N_a^{(x)} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & T_b^n N_g & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & K_a & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & K_g & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & \beta_a B_a & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & \beta_g B_g \end{bmatrix} \begin{bmatrix} v_a^{arw} \\ v_g^{arw} \\ v_a^{rrw} \\ v_g^{rrw} \\ v_a^{bias} \\ v_g^{bias} \end{bmatrix} \tag{5.63}$$

$$y = \begin{bmatrix} 0 & 0 & 1 & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} \\ 0_{1\times3} & 0_{1\times3} & 1 & 0 & 0 & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} \\ 0_{1\times3} & 0_{1\times3} & 0 & 1 & 0 & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} \\ 0_{1\times3} & 0_{1\times3} & 0 & 0 & 1 & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} \end{bmatrix} \begin{bmatrix} \delta p \\ \delta v_{nb}^b \\ \delta\Theta \\ a_{rrw} \\ w_{rrw} \\ a_{fn} \\ w_{fn} \end{bmatrix} \tag{5.64}$$

The open form of A_{13}, A_{23}, A_{24}, A_{26} and A_{33} in Equation 5.63 is shown below:

$$A_{13} = \begin{bmatrix} 0 & -\sin(\theta)\cos(\psi)v_b^{(x)} & -\cos(\theta)\sin(\psi)v_b^{(x)} \\ 0 & -\sin(\theta)\sin(\psi)v_b^{(x)} & \cos(\theta)\cos(\psi)v_b^{(x)} \\ 0 & -\cos(\theta)v_b^{(x)} & 0 \end{bmatrix} \tag{5.65}$$

$$A_{23} = \begin{bmatrix} 0 & -9.81\cos(\theta) & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{5.66}$$

$$A_{24} = A_{26} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{5.67}$$

$$A_{33} = \begin{bmatrix} \cos(\phi)\tan(\theta)w_y - \sin(\phi)\tan(\theta)w_z & \sin(\phi)\sec^2(\theta)w_y + \cos(\phi)\sec^2(\theta)w_z & 0 \\ -\sin(\phi)w_y - \cos(\phi)w_z & 0 & 0 \\ \frac{\cos(\phi)w_y - \sin(\phi)w_z}{\cos(\theta)} & \frac{\sin(\phi)\sin(\theta)w_y + \cos(\phi)\sin(\theta)w_z}{\cos^2(\theta)} & 0 \end{bmatrix} \tag{5.68}$$

In Equation 5.65, $v_b^{(x)}$ is the recent x-axis velocity of the vehicle in body frame. Also in Equation 5.68, w_x, w_y and w_z represent the recent estimated angular rates of the vehicle when the error terms are extracted. Equation 5.64 is the same of Equation 5.54 since measurement equation is linear. Therefore, there is no need for the Jacobian of the observation equation $h(x(t), t)$. The T_b^n in the process noise matrix and state transition matrix is the angular velocity transformation matrix which is shown in Equation 2.9.

This matrices are formed according to new inputs and recent outputs at each step. Also the state space representation matrices are written above in continuous time. Therefore, they must be converted to discrete time using Equations 4.13. Complete block diagram of the system with an extended Kalman filter is shown in Figure 5.15.

Figure 5.15 shows that outputs of IMU sensors go directly through to the Kalman Filter different than that in linearized error state implementation. In extended Kalman filter block, first of all the navigation parameters are calculated using the nonlinear equations and the IMU sensor outputs. Then error state is estimated by the Kalman filter algorithm with the aid of measurements coming from pressure sensor and continous angle generation blocks. Finally, calculated navigation parameters are corrected with the estimated error states and IMU errors in the error state is given as a feedback to correct IMU outputs.

When the state space matrices for both systems are compared, the most significant difference is the $A33$ term. This term comes from the partial differentiation of nonlinear attitude equation with respect to attitude and heading (roll, pitch, yaw) angles. Also both of the systems are partially observable which means that all of the state parameters cannot be estimated correctly. This issue is verified in the simulations.

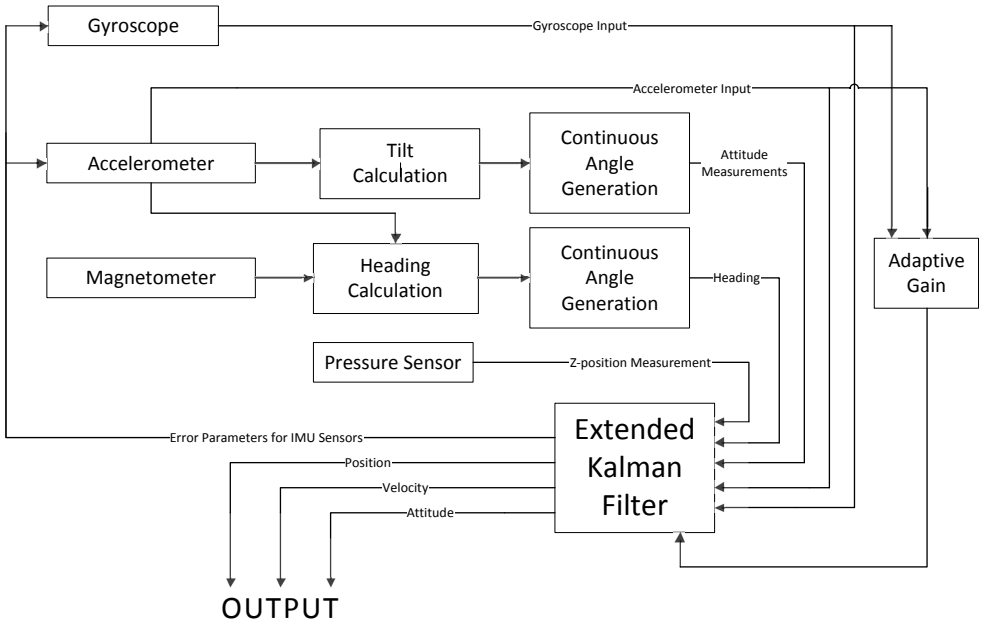

Figure 5.15: Block diagram of the system with extended Kalman filter

5.8 Simulations

To test the designed system, an inclined path is planned for the vehicle movement. This path also consists of different manoeuvres. The vehicle follows the reference path which is shown in Figure 5.16 for visualization of generated scenario, so that the system designed in this thesis work is compared with the reference values. As it is stated in Section 5.7, both of the designed systems are not fully observable. Therefore, some of the state variables are not estimated correctly. The main issue about these simulations is the integration of the pressure sensor and adaptive system. Thus, the effects of these two terms are evaluated with Root Mean Square Error (RMSE) values for attitude and heading angles.

In the simulations, both error state Kalman filter implementation and extended Kalman filter implementation are tested. For both systems, the effects of pressure sensor aid and adaptive system are analyzed. In order to observe whether there is an improvement or not, RMSE of the results are calculated using Equation 5.69

$$RMSE = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(x_n - x_n^{ref})^2} \tag{5.69}$$

RMSE is calculated for roll, pitch and yaw angles seperately. This error calculation is divided

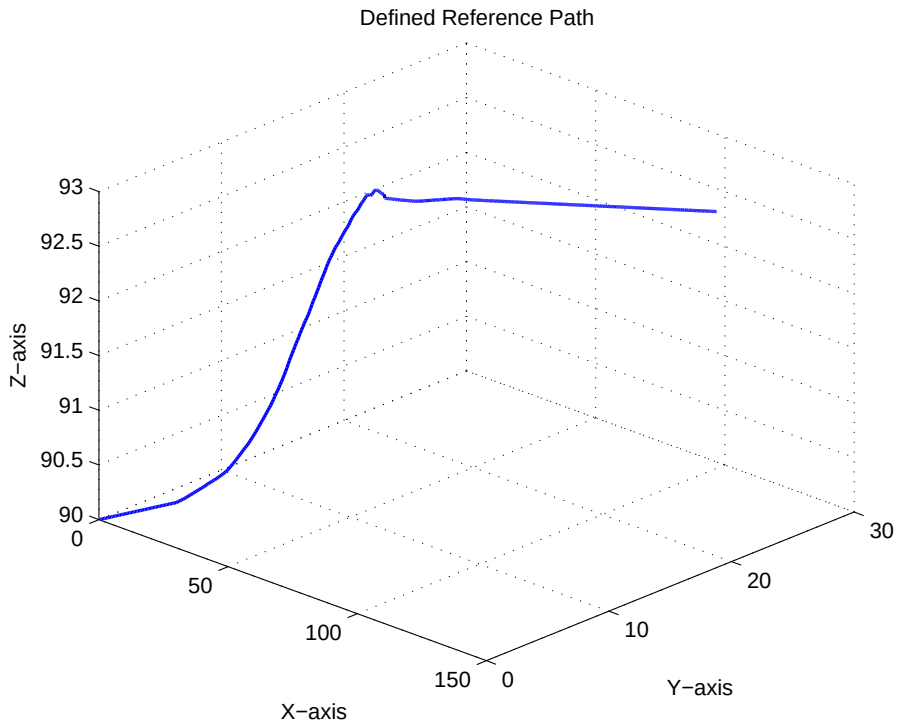

Figure 5.16: Defined reference path for simulations

into two groups for motion period and stationary period in order to have a better comparison. Moreover, output data of IMU sensor and pressure sensor are recorded when the vehicle moves in the defined path. Then same data set is used for every different run in MATLAB© implementations of the described system.

Figure 5.17 and Figure 5.18 are the results of direct implementation of linearized error state Kalman filter without adaptive system and pressure sensor for the attitude and heading angles, for motion and stationary periods, respectively. Position and velocity values are not given since they are not estimated correctly due to the partial observability. The errors of position and velocity diverge.

It is obvious from the given figures that when pressure sensor and adaptive system are not included, the navigation system is not practically useful because of the large amount of errors. Especially when the vehicle moves and rotates, outputs deviate from the reference curves significantly.

As it is stated in the scope of thesis, the integration of pressure sensor could reduce the amount

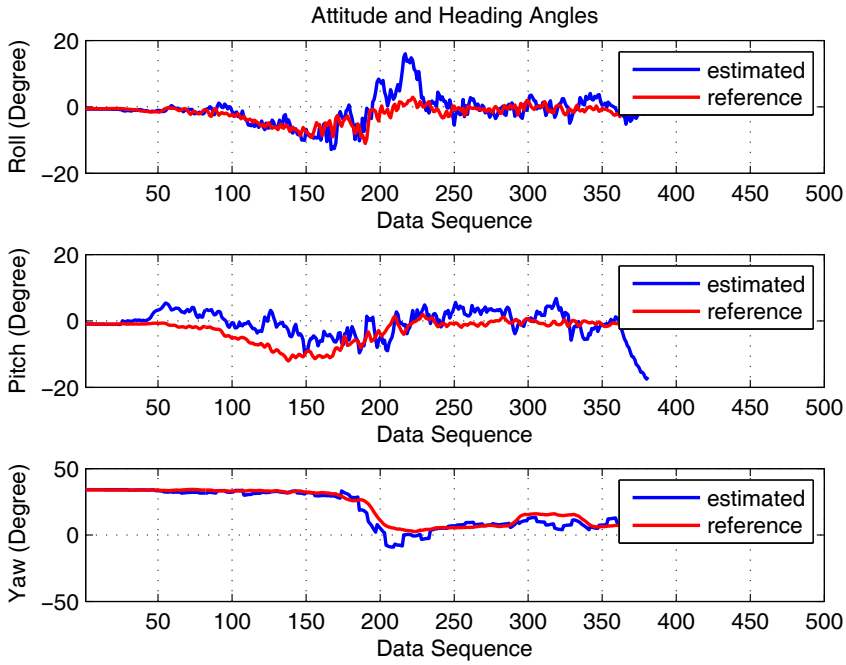

Figure 5.17: Roll, pitch and yaw angles outputs of linearized error state Kalman filter for motion period without adaptive system and pressure sensor

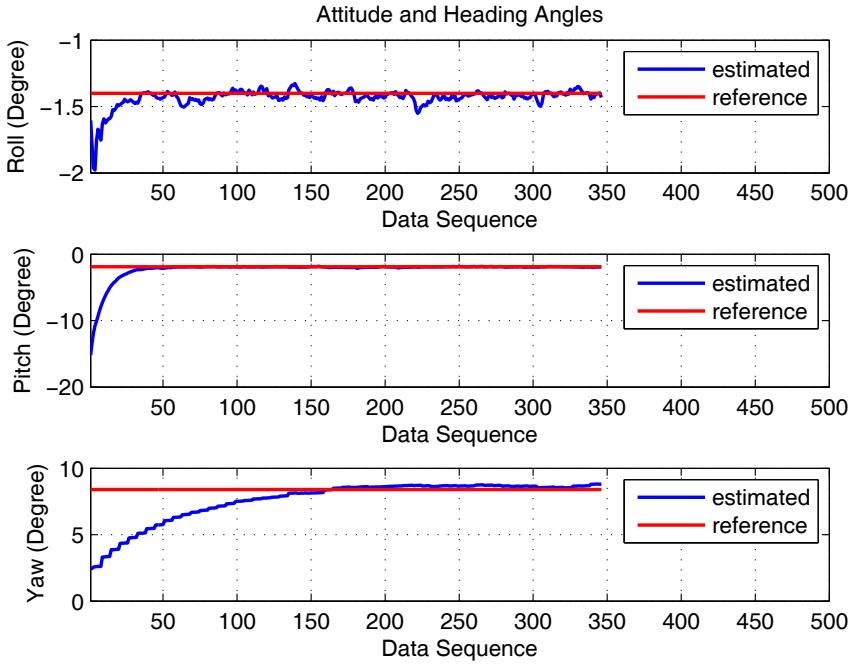

Figure 5.18: Roll, pitch and yaw angles outputs of linearized error state Kalman filter for stationary period without adaptive system and pressure sensor

of error. Therefore, pressure sensor aid is given as a measurement to the linearized error State Kalman filter implementation. The results are presented in Figure 5.19 and Figure 5.20 for motion period and stationary period, respectively.

If Figure 5.19 and Figure 5.20 are evaluated carefully, the slight improvement can be observed in the Pitch angle. Also it is obvious in stationary position plot that pressure sensor makes pitch angle output to converge faster to the real value.

Although, pressure sensor provide a little improvement in the attitude and heading angles, the amount of error is still too large. As it is mentioned in Section 5.4, the reason of this error is the sensitivity of the measurements to the highly dynamic vehicle motion. So adaptive system that is explained in Section 5.4 is integrated to the navigation system. In order to observe the effect of adaptive system only, the simulation is run first without pressure sensor aid. The resulting output graphs for attitude and heading angles are given in Figure 5.21 and Figure 5.22 for motion period and stationary period respectively.

The outputs graphs in Figure 5.21 and Figure 5.22 shows that, outputs are closer to the reference values so that adaptive system provides a considerable improvement in the attitude and heading angle.

For the lowest possible error, both adaptive system and pressure sensor is included in the navigation system and Figure 5.23 and Figure 5.24 represent the attitude and heading angle outputs of the simulation result for motion period and stationary period, respectively.

It is obvious when pressure sensor and adaptive system work together, best calculation results are achieved for linearized error state Kalman filter implementation. Velocity and position graphs are given for this system in Figure 5.25 and Figure 5.26, respectively.

As it is mentioned before, velocity and position are not observable in the system. Therefore velocity and position outputs contains large amount of error. Therefore, they are not considered in the comparison of the navigation system implementations.

Performances of linearized error state Kalman filter implementations are compared in terms of RMSE error. RMSE error for the above simulations are given for comparison in Table 5.5 for roll, pitch and yaw angles grouped by motion period and stationary period.
The improvements of the adaptive system and pressure sensor can be easily observed in Table

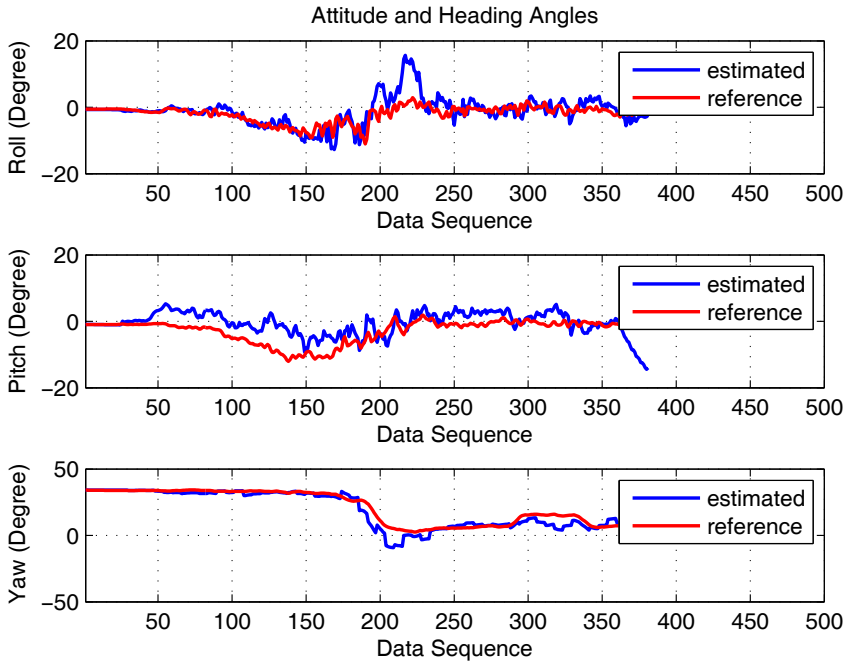

Figure 5.19: Roll, pitch and yaw angles outputs of linearized error state Kalman filter for motion period with pressure sensor

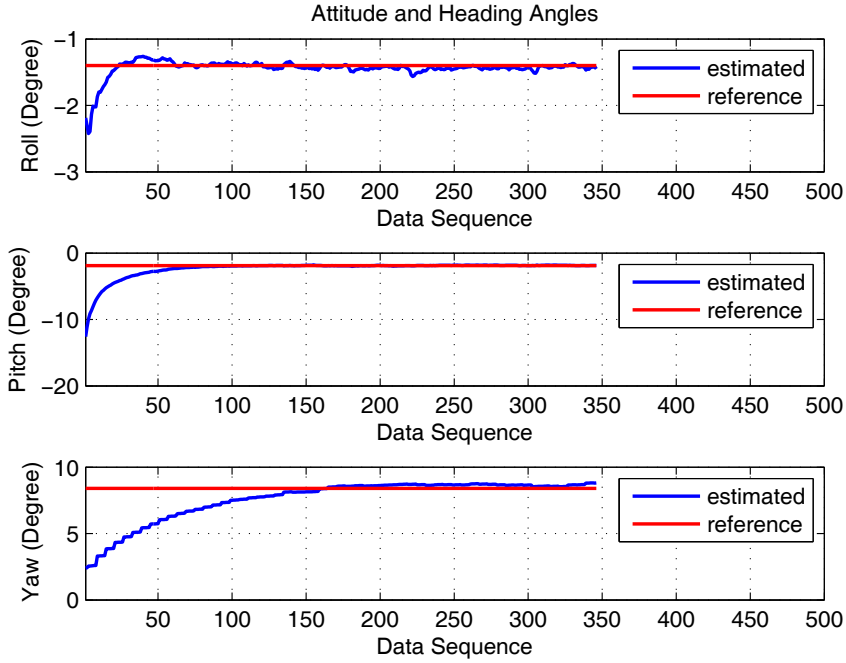

Figure 5.20: Roll, pitch and yaw angles outputs of linearized error state Kalman filter for stationary period with pressure sensor

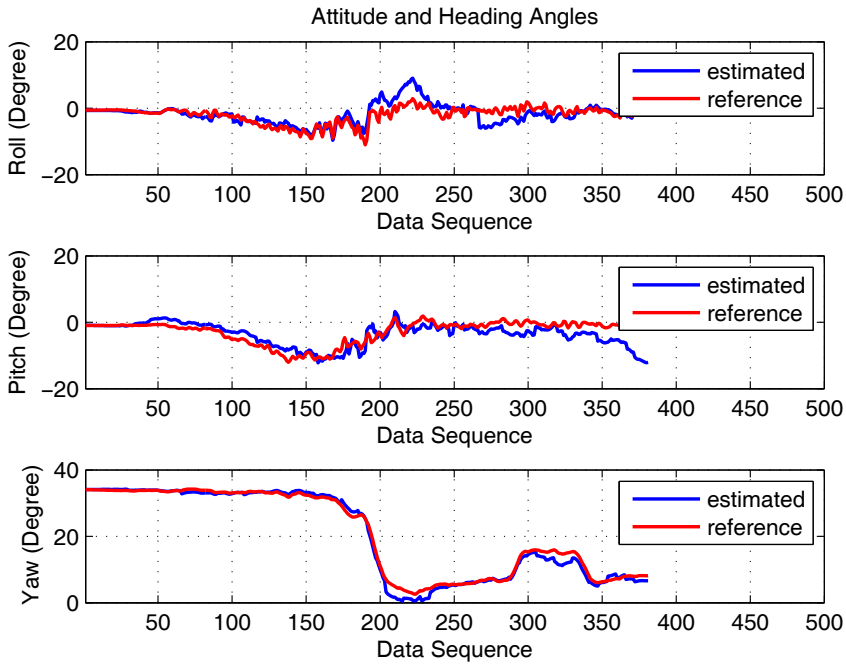

Figure 5.21: Roll, pitch and yaw angles outputs of linearized error state Kalman filter for motion period with adaptive system

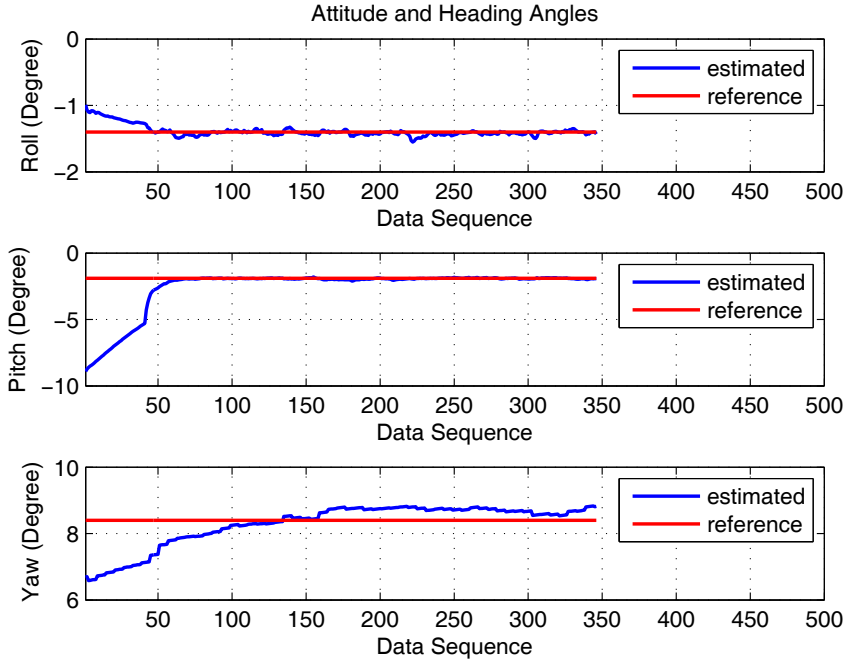

Figure 5.22: Roll, pitch and yaw angles outputs of linearized error state Kalman filter for stationary period with adaptive system

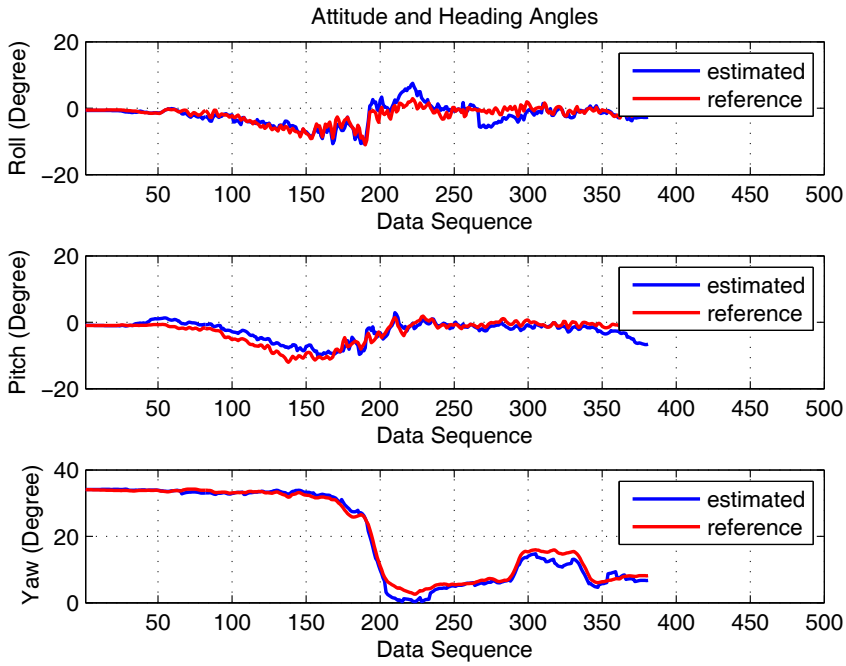

Figure 5.23: Roll, pitch and yaw angles outputs of linearized error state Kalman filter for motion period with adaptive system and pressure sensor

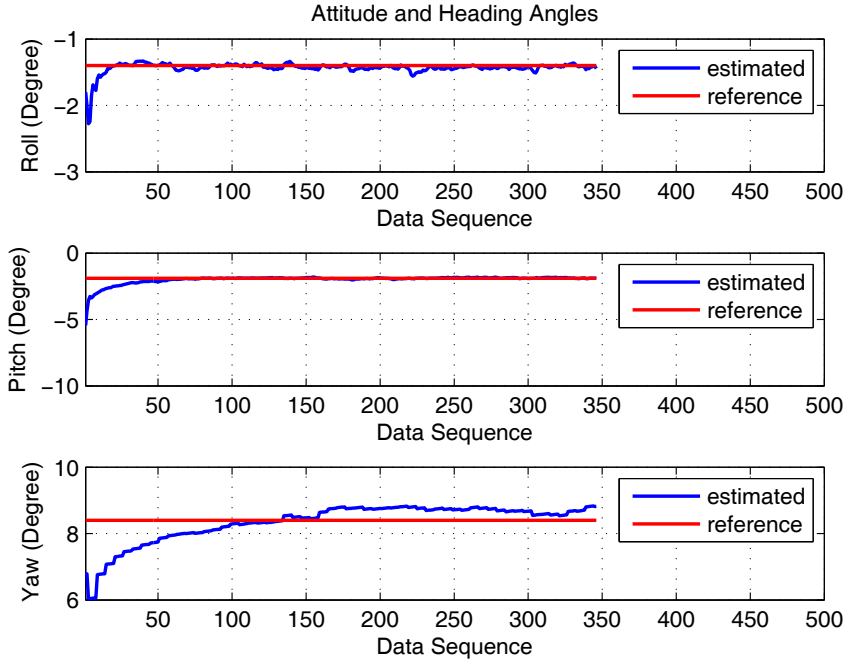

Figure 5.24: Roll, pitch and yaw angles outputs of linearized error state Kalman filter for Stationary period with adaptive system and pressure sensor

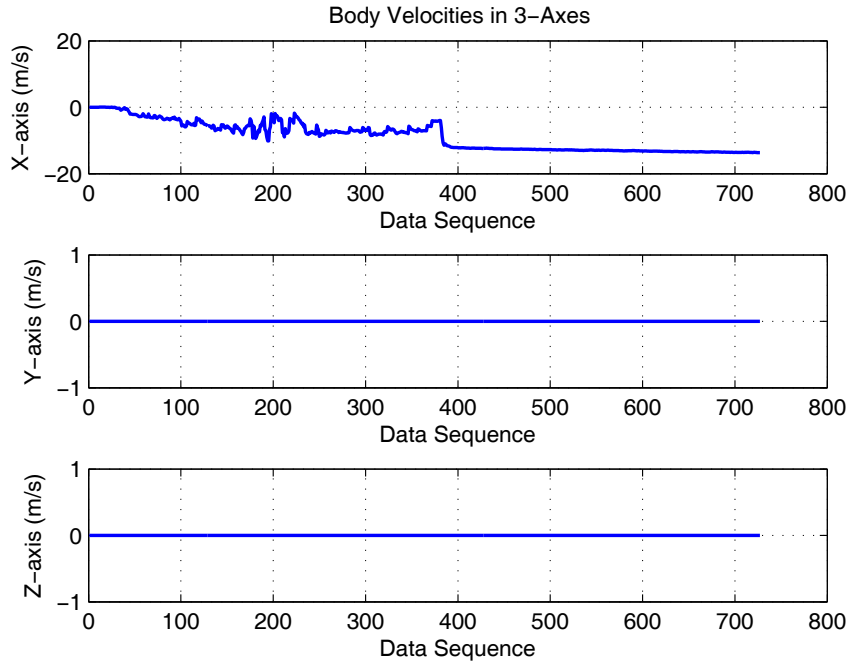

Figure 5.25: Velocity output of linearized error state Kalman filter with adaptive system and pressure sensor

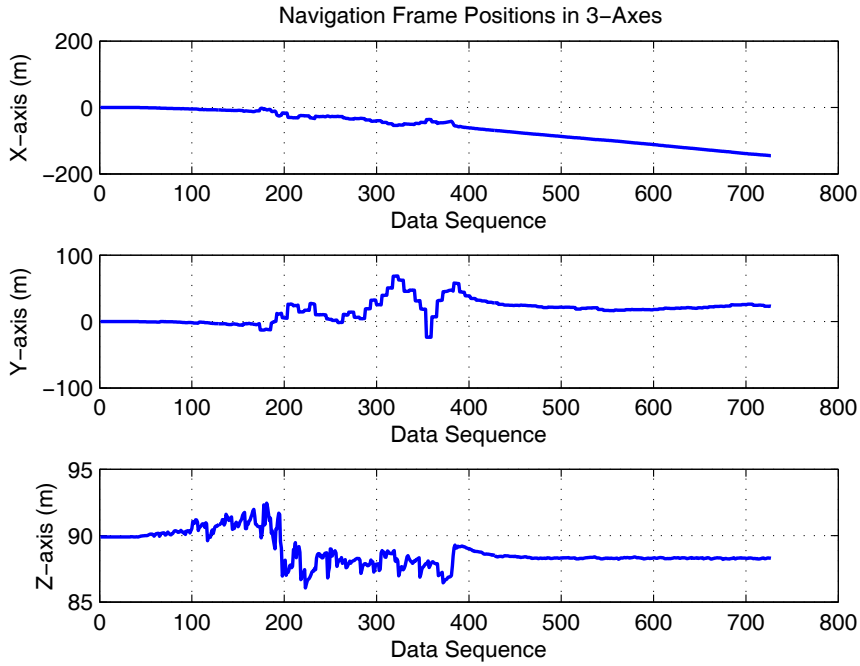

Figure 5.26: Position output of linearized error state Kalman filter with adaptive system and pressure sensor

Table 5.5: Linearized error state Kalman filter RMS errors

Linearized Error State Kalman Filter RMS Errors				
	Pressure OFF Adaptive OFF	**Pressure ON Adaptive OFF**	**Pressure OFF Adaptive ON**	**Pressure ON Adaptive ON**
Roll Error (Motion)	3.2935	3.1533	2.1707	1.7245
Pitch Error (Motion)	4.6746	4.2787	2.809	1.7926
Yaw Error (Motion)	4.4009	4.4154	1.3187	1.4973
Roll Error (Stationary)	0.07828	0.14215	0.08631	0.09674
Pitch Error (Stationary)	1.5611	1.4175	1.8026	0.35412
Yaw Error (Stationary)	1.7459	1.7542	0.6379	0.5758

5.5 for the linearized error state Kalman filter implementation. Integration of adaptive system to the Kalman filter makes an important improvement in the RMS errors. Pressure sensor reduces the pitch error for both motion period and stationary period.

Same course of experiments are applied for the extended Kalman filter implementation. In the first simulation, both pressure sensor and adaptive system are not available in the system and outputs of this system are given in Figure 5.27 and Figure 5.28. Then, Figure 5.29 and Figure 5.30 show the output of the system when only pressure sensor is included. After that, only adaptive system is included in the simulation and output graphs for attitude and heading angles are given in Figure 5.31 and Figure 5.32. Finally, both adaptive system and pressure sensor is integrated in the extended Kalman filter implementation. The results for the final simulation are given in Figure 5.33 and Figure 5.34.

Although, the EKF system is not completely observable, velocity and position outputs are given in Figure 5.35 and Figure 5.36 respectively. Position and velocity graphs show that they diverge as time goes by becaues of the error accumulation.

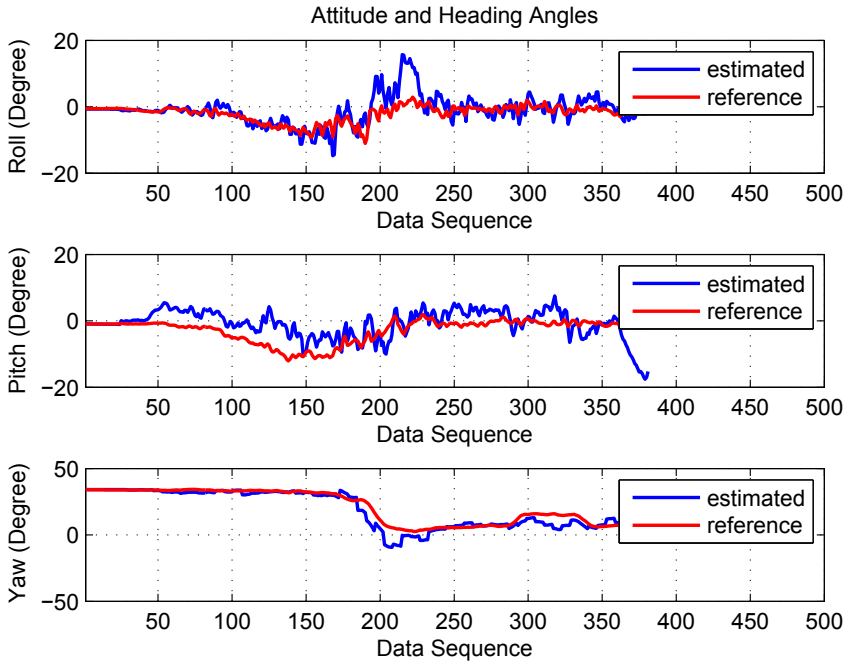

Figure 5.27: Roll, pitch and yaw angles outputs of extended Kalman filter for motion period without adaptive system and pressure sensor

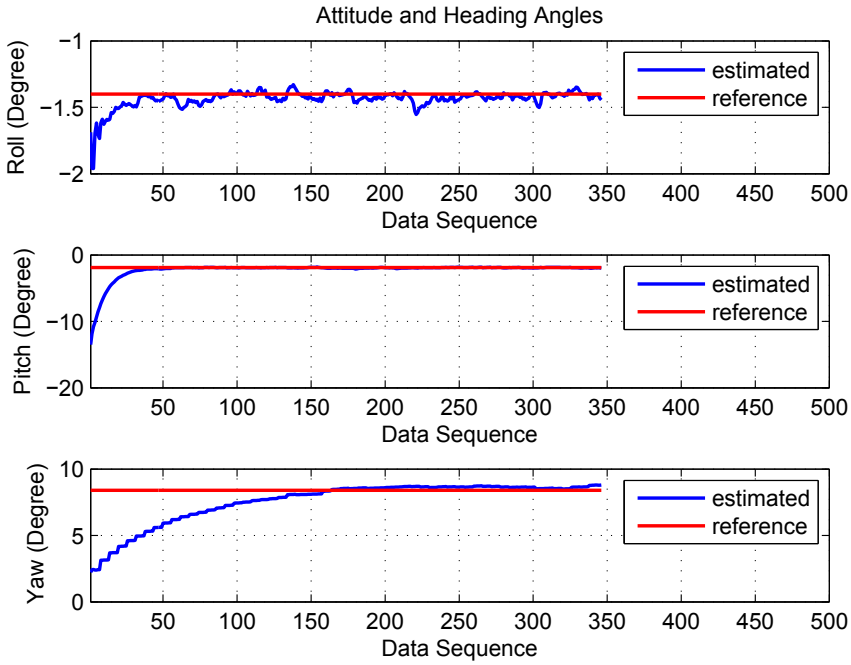

Figure 5.28: Roll, pitch and yaw angles outputs of extended Kalman filter for stationary period without adaptive system and pressure sensor

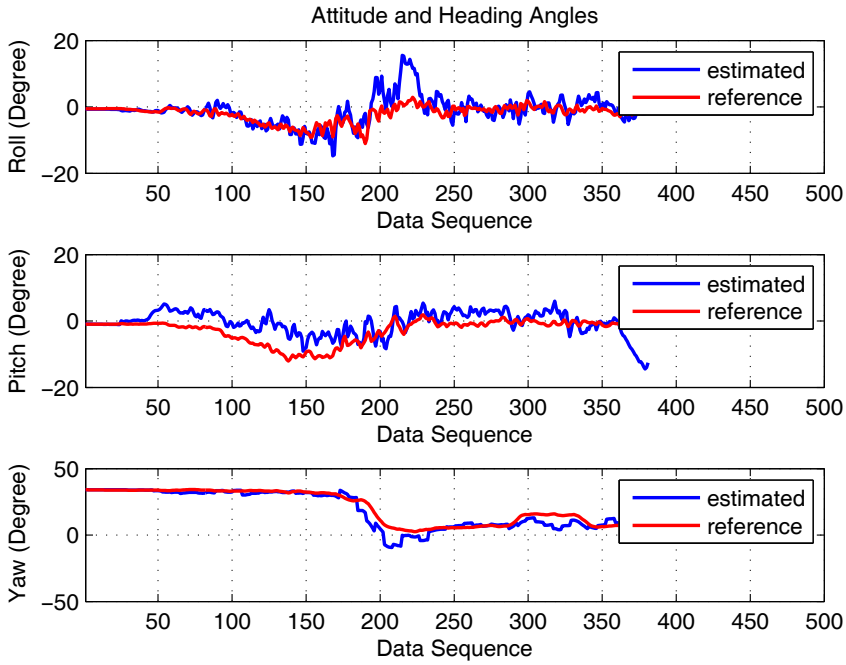

Figure 5.29: Roll, pitch and yaw angles outputs of extended Kalman filter for motion period with pressure sensor

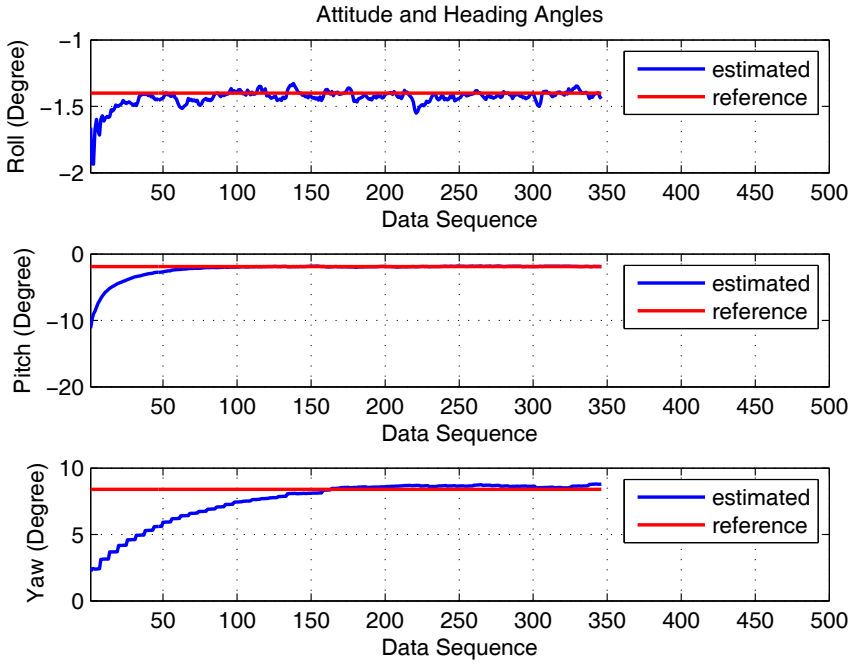

Figure 5.30: Roll, pitch and yaw angles outputs of extended Kalman filter for stationary period with pressure sensor

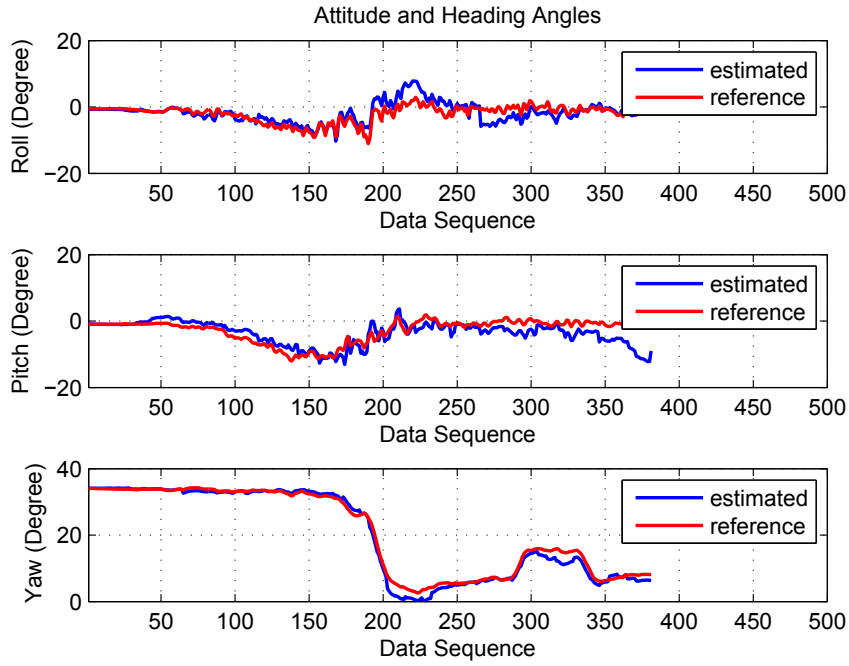

Figure 5.31: Roll, pitch and yaw angles outputs of extended Kalman filter for motion period with adaptive system

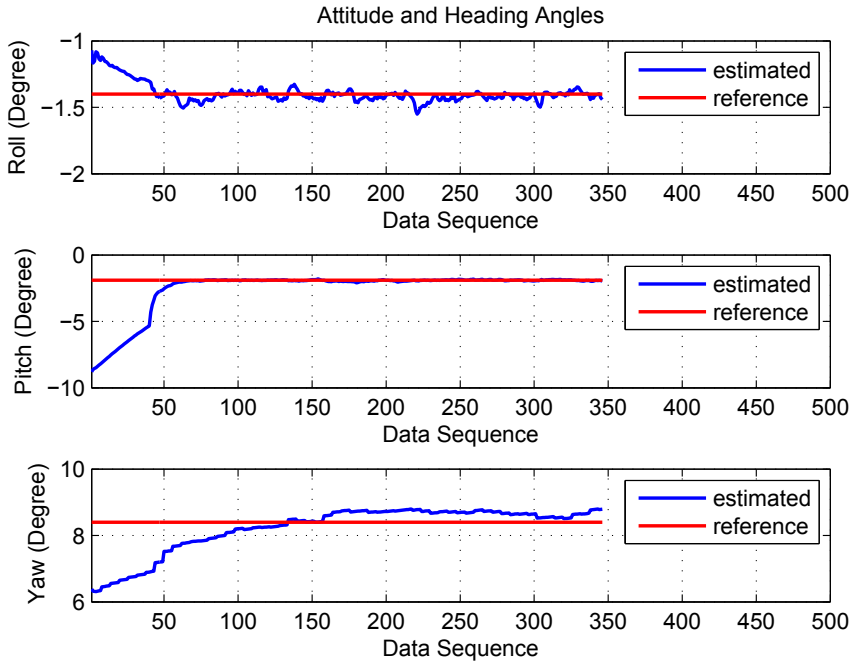

Figure 5.32: Roll, pitch and yaw angles outputs of extended Kalman filter for stationary period with adaptive system

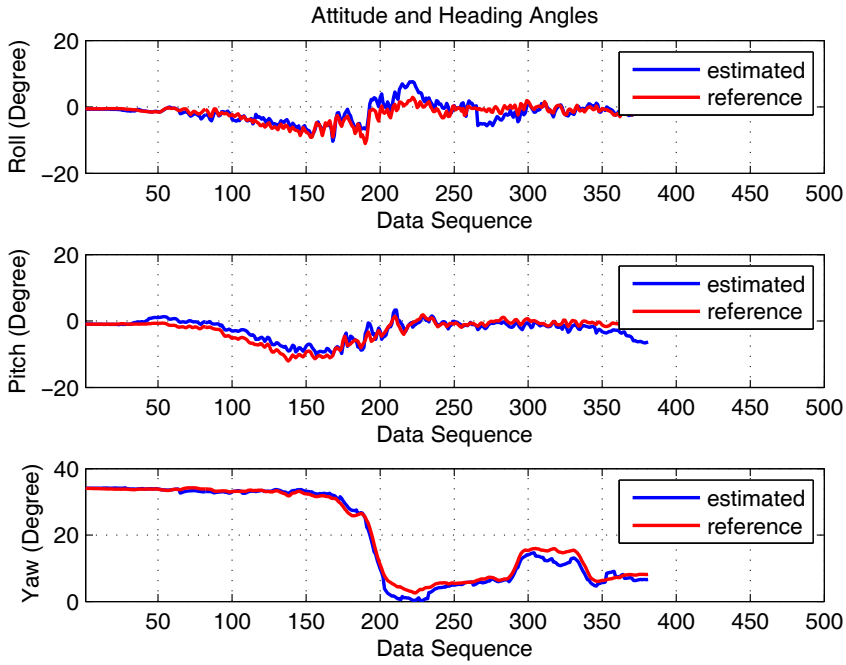

Figure 5.33: Roll, pitch and yaw angles outputs of extended Kalman filter for motion period with adaptive system and pressure sensor

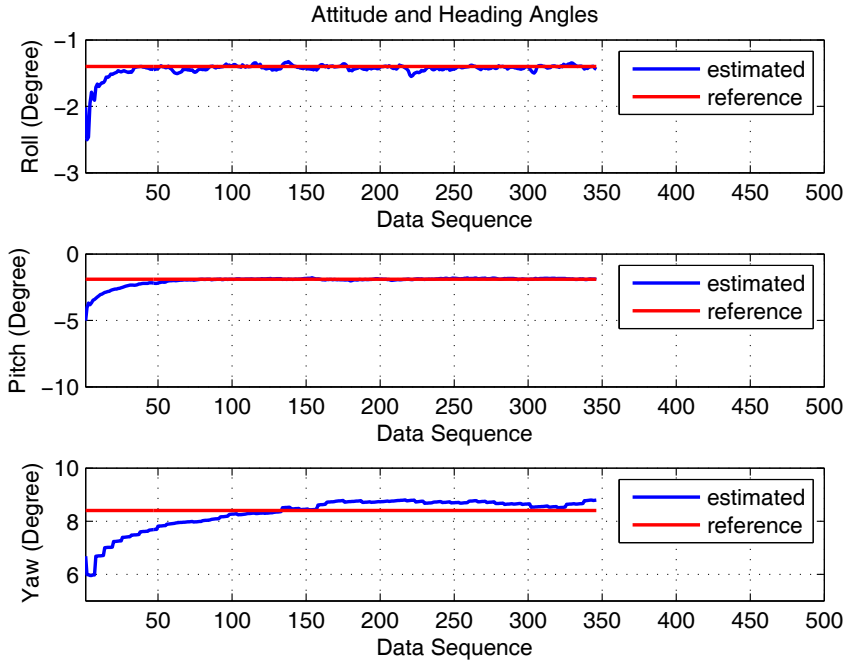

Figure 5.34: Roll, pitch and yaw angles outputs of extended Kalman filter for stationary period with adaptive system and pressure sensor

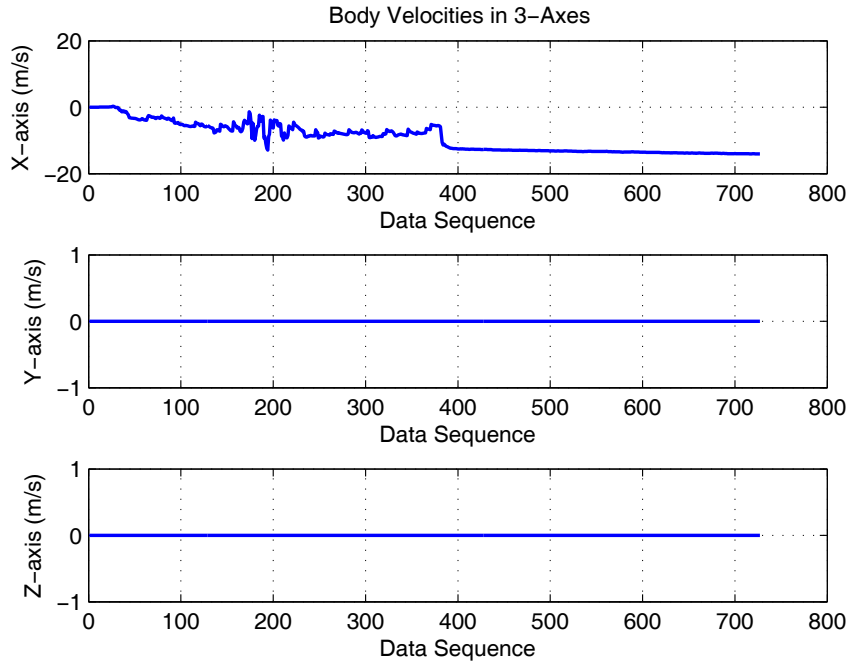

Figure 5.35: Velocity output of extended Kalman filter with adaptive system and pressure sensor

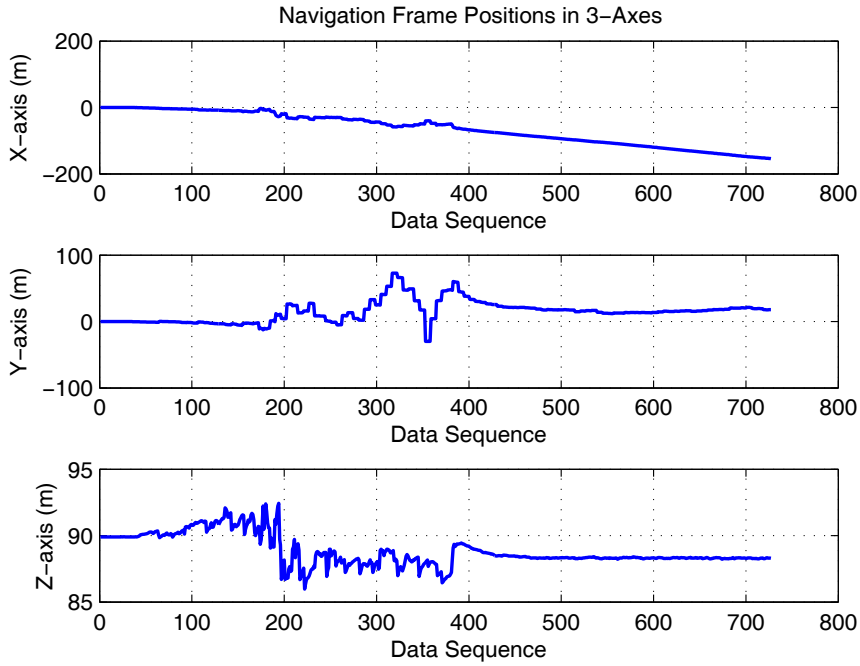

Figure 5.36: Position output of extended Kalman filter with adaptive system and pressure sensor

As in the linearized error state implementation, best results are obtained when adaptive system and pressure sensor are available. RMSE values for extended Kalman filter implementations are summarized in Table 5.6

Table 5.6: Extended Kalman filter RMS errors

Extended Kalman Filter RMS Errors				
	Pressure OFF Adaptive OFF	**Pressure ON Adaptive OFF**	**Pressure OFF Adaptive ON**	**Pressure ON Adaptive ON**
Roll Error (Motion)	3.2848	3.2267	2.0792	1.8358
Pitch Error (Motion)	4.7308	4.2872	2.8431	1.684
Yaw Error (Motion)	4.5616	4.5494	1.4367	1.5309
Roll Error (Stationary)	0.07655	0.07251	0.07591	0.1179
Pitch Error (Stationary)	1.39451	1.2828	1.7759	0.3835
Yaw Error (Stationary)	1.78960	1.7914	0.71776	0.5872

When RMSE errors in Table 5.5 and Table 5.6 are compared, linearized error state Kalman filter appears to have a better performance. Only pitch angle in motion period is estimated better by extended Kalman filter. Therefore, linearized error state Kalman filtering implementation should be prefered for the navigation system of the vehicle if the pitch angle is not the most critical element for the operation.

CHAPTER 6

CONCLUSION AND FUTURE WORKS

6.1 Conclusion

In this thesis, integration of pressure sensor and adaptive gain is examined for attitude and heading reference system of underwater vehicle. Attitude and Heading calculation is the most crucial part of the navigation of an underwater vehicle. Linearized error state Kalman filter is the most widely used algorithm for AHRS systems, but extended Kalman filter algorithm is also practiced in this thesis work. Since there is a lack of aiding sensor opportunities for underwater environment, only magnetometer, pressure sensor and accelerometer can be used. However these sensors are not accurate enough, so an adaptive system is developed in this thesis work. MEMS IMU sensor is used in order to reduce the cost of the system, however such sensors yield large amounts of error. For the system implementations, these errors are categorized and their parameters are measured. The designed system is verified in MATLAB© using the real field data which is collected by using a low cost MEMS IMU sensor.

In the literature, error state model of mechanization equations are mostly used in the navigation system. While working on mechanization equations for this thesis work, it is experienced that working with error states is easier. The derivation of the equations are simple and most of the terms in equations can be figured out intuitively. Furthermore using the error states provides an advantage in Kalman Filter implementation because of the simple terms and the basic form of equations. That is why the error state model of mechanization equations is preferred.

In the simulations, it is found that the designed system and pressure sensor provides a sig-

nificant improvement in the attitude and heading angles. Especially adaptive system reduces the RMS error in motion period more than 50%. On the other hand, pressure sensor mostly focuses on pitch angles as it is expected. Because of the x-axis velocity and zero buoyancy assumption, when the pitch angle of the vehicle changes, it's z-position also changes. Therefore, pressure sensor supports the pitch angle estimation of the navigation system.

Because of the limited number of the aiding sensors, the position and velocity of the vehicle cannot be observed in the simulations. So they are considered in the simulations in detail. The main concern is the attitude and heading angles with the new designed system model. Therefore, this system can be interpreted as an attitude and heading reference system.

When the RMS errors of extended and linearized error state Kalman filter are compared, linearized error state Kalman filter implementation has a better performance. Actually, it is expected that extended Kalman filter would work better, because the linearization is more accurate. However, because of the observability problem, some of the terms in the state transition matrix are not linearized perfectly. For example, A_{13} matrix contains velocity (x-axis) of the vehicle which is not estimated correctly. So this results in an increase in the error of the attitude and heading angles.

If the extended Kalman filter outputs are analyzed in short time periods, it is observed that outputs deviate from the reference values when the vehicle moves with swings. The underlying reason for this situation is the fact that the first order linearization of extended Kalman filter cannot handle this kind of motions. Thus, growing nonlinearity error is reflected as an error in the outputs of the Kalman Filter.

The simulation results show that adaptive system is the critical block of the whole system. So the output of the adaptive system block must be designed properly. It is observed that, the RMS errors are dependent on this output, named as scale factor matrix. The diagonal elements of this matrix must be optimized for the best performance.

6.2 Future Work

In order to further increase the performance of the navigation system, proper initialization of the state and covariance matrix could be further considered, so that the RMS error would be

reduced in the initialization period. Also a different aiding sensor can be added to the system. Thus, the observability problem could be solved. In order to measure the velocity, a pitot tube could be used. However, this kind of implementation will increase the cost and space requirement. Also an extra mechanical effort is needed.

This system can be compared with another IMU sensor. This sensor could be of better quality than the one used in this thesis work. So, the significance of the algorithm can be observed. Finally, a better adaptive system can be developed by considering all posible situations. This adaptation might be designed to achieve smooth crossing between different models of Kalman Filter.

Finally, the number of simulations can be increased along with scenarios as a future work. Increase in the number of simulations and their critical studies could help to improve the system performance.

REFERENCES

[1] A. Lawrence. *Modern Inertial Technology: Navigation, Guidance, and Control*. Mechanical Engineering Series. Springer, 1998.

[2] Mohinder S. Grewal, Lawrence R. Weill, and Angus P. Andrews. *Global Positioning Systems, Inertial Navigation, and Integration*. Wiley-Interscience, 2007.

[3] P.D. Groves. *Principles of GNSS, inertial, and multi-sensor integrated navigation systems*. GNSS technology and applications series. Artech House, 2008.

[4] Jay Farrell. *Aided Navigation: GPS with High Rate Sensors*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 2008.

[5] IEEE standard specification format guide and test procedure for single-axis interferometric fiber optic gyros. *IEEE Std 952-1997*, page i, 1998.

[6] M.J. Caruso. Applications of magnetic sensors for low cost compass systems. In *Position Location and Navigation Symposium, IEEE 2000*, pages 177 –184, 2000.

[7] SparkFun Electronics. 9 Degrees of Freedom - Razor IMU. `http://www.sparkfun.com/products/10736`, 2012.

[8] B. Hofmann-Wellenhof and K. Legat. *Navigation: Principles of Positioning and Guidance*. Springer, 2003.

[9] C. Jekeli. *Inertial Navigation Systems With Geodetic Applications*. Walter de Gruyter, 2000.

[10] Varadan Vijay K., Jose K. A., and Varadan Vasundara V. High sensitive and wide dynamic range navigation microsystem on a single chip. *Proceedings of SPIE, the International Society for Optical Engineering*, 4236:134–140, 2001. eng.

[11] K. J. Walchko and Paul Mason. Inertial navigation. *2002 Florida Conference on Recent Advances in Robotics*, May 2002. eng.

[12] D. Titterton and J. Weston. *Strapdown Inertial Navigation Technology*. The American Institute of Aeronautics and Astronautics, second edition, 2004.

[13] T.I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons, 2011.

[14] Hermann Anschütz-Kaempfe and Friedrich von Shirach. Kreiselapparat(gyroscope), 04 1907.

[15] E. Bekir. *Introduction to Modern Navigation Systems*. World Scientific, 2007.

[16] Wall J. Flenniken, W. and D. Bevly. Characterization of various imu error sources and the effect on navigation performance. *Proceedings of the 18th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2005)*, 1:967 –978, September 2005.

[17] D Allan. Statistics of atomic frequency standards. *Proceedings of the IEEE*, 54(2):221–230, 1966.

[18] N. El-Sheimy, Haiying Hou, and Xiaoji Niu. Analysis and modeling of inertial sensors using allan variance. *Instrumentation and Measurement, IEEE Transactions on*, 57(1):140 –149, jan. 2008.

[19] Songlai Han and Nathan Knight. Using allan variance to determine the calibration model of inertial sensors for gps / ins intergration. *6th International Symposium on Mobile Mapping Technology*, July 2009.

[20] Vaibhav Saini, S C Rana, and M M Kuber. Online estimation of state space error model for mems imu. *Journal of Modelling and Simulation of Systems*, 1:219–225, 2010.

[21] P.A. Miller, J.A. Farrell, Yuanyuan Zhao, and V. Djapic. Autonomous underwater vehicle navigation. *Oceanic Engineering, IEEE Journal of*, 35(3):663 –678, July 2010.

[22] James C. Kinsey, Ryan M. Eustice, and Louis L. Whitcomb. A survey of underwater vehicle navigation: Recent advances and new challenges. In *IFAC Conference of Manoeuvering and Control of Marine Craft*, Lisbon, Portugal, September 2006. Invited paper.

[23] K. Vickery. Acoustic positioning systems. a practical overview of current systems. In *Autonomous Underwater Vehicles, 1998. AUV'98. Proceedings Of The 1998 Workshop on*, pages 5 –17, aug 1998.

[24] 3-Axis Digital Compass IC HMC5883L Data Sheet. `http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Sensors/Magneto/HMC5883L-FDS.pdf`, March 2011.

[25] Michael J Caruso. Applications of magnetoresistive sensors in navigation systems. *Readings*, 72:15–21, 1997.

[26] Wei Dong, Kwang Yong Lim, Young Koon Goh, Kim Doang Nguyen, I-Ming Chen, Song Huat Yeo, and Been-Lirn Duh. A low-cost motion tracker and its error analysis. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 311 –316, may 2008.

[27] MLH Series All Metal Pressure Sensors Product Sheets. `http://sensing.honeywell.com/index.php?ci\_id=51394`, February 2011.

[28] P.S. Maybeck. *Stochastic models, estimation and control. Volume I.* 1979.

[29] Mohinder S. Grewal and Angus P. Andrews. *Kalman Filtering : Theory and Practice Using MATLAB.* Wiley-Interscience, 2 edition, January 2001.

[30] C. Van Loan. Computing integrals involving the matrix exponential. *Automatic Control, IEEE Transactions on*, 23(3):395 – 404, jun 1978.

[31] S.S. Haykin. *Kalman Filtering and Neural Networks*. Adaptive and learning systems for signal processing, communications, and control. Wiley, 2001.

[32] Greg Welch and Gary Bishop. An introduction to the kalman filter, 1995.

[33] M. Ahmadi, A. Khayatian, and P. Karimaghaee. Orientation estimation by error-state extended kalman filter in quaternion vector space. In *SICE, 2007 Annual Conference*, pages 60 –67, sept. 2007.

[34] E. Foxlin. Inertial head-tracker sensor fusion by a complementary separate-bias kalman filter. In *Virtual Reality Annual International Symposium, 1996., Proceedings of the IEEE 1996*, pages 185 –194, 267, April 1996.

[35] Stergios I. Roumeliotis, Gaurav S. Sukhatme, and George A. Bekey. Circumventing dynamic modeling: Evaluation of the error-state kalman filter applied to mobile robot localization. In *In Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, pages 10–15, 1999.

[36] Rong Zhu, Zhaoying Zhou, Sha Li, and Xuefeng Sun. A novel miniature azimuth-level detector based on mems. In *Microelectromechanical Systems Conference, 2001*, pages 50 –53, 2001.

[37] Yongqi Chen Congwei Hu, Wu Chen and Dajie Liu. Adaptive kalman filtering for vehicle navigation. *Journal of Global Positioning Systems (2003)*, 2(1):42–47, november 2003.

[38] Tae Suk Yoo, Sung Kyung Hong, Hyok Min Yoon, and Sungsu Park. Gain-scheduled complementary filter design for a mems based attitude and heading reference system. *Sensors*, 11(4):3816–3830, 2011.

[39] Panich S. Indirect kalman filter in mobile robot application. *Journal of Mathematics and Statistics*, 6(3):381–384, 2010.