ISTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

MULTI-SENSOR AIDED INS/GPS INTEGRATION FOR AN UGV

M.Sc. Thesis by
Ziya ERCAN

Department : Control and Automation Engineering

Programme : Control and Automation Engineering

JULY 2011

ISTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

MULTI-SENSOR AIDED INS/GPS INTEGRATION FOR AN UGV

M.Sc. Thesis by
Ziya ERCAN
(504081130)

Date of submission : 6 May 2011
Date of defense examination: 28 July 2011

Supervisor (Chairman) : Prof. Dr. Metin GOKASAN(ITU)
Members of the Examining Committee : Prof. Dr. Ata MUĞAN (ITU)
Prof. Dr. Hakan TEMELTAS (ITU)

JULY 2011

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

İNSANSIZ KARA ARACI İÇİN ÇOKLU ALGILAYICI YARDIMLI ANS/GKS ENTEGRASYONU

YÜKSEK LİSANS TEZİ
Ziya ERCAN
(504081130)

Tezin Enstitüye Verildiği Tarih : 6 Mayıs 2011
Tezin Savunulduğu Tarih : 28 Temmuz 2011

Tez Danışmanı : Prof. Dr. Metin GÖKAŞAN(İTÜ)
Diğer Jüri Üyeleri : Prof. Dr. Ata MUĞAN (İTÜ)
Prof. Dr. Hakan TEMELTAŞ (İTÜ)

TEMMUZ 2011

FOREWORD

I would like to express my deep appreciation and thanks for my family that they always support me in any condition and to my advisor Metin Gökaşan who gives me this opportunity to be a part of the project. Also I would like give my special thanks to Ata Mugan and Pinar Boyraz for their supports.

Thanks to all my friends, especially Volkan Sezer, Çağrı Dikilitaş, Hasan Heceoğlu, Emre Akça, Alper Öner, Ersin Bulgu, Okan Türkmen, Mortaza Aliasghary, Eray Çakıray and all MEAM family, who have helped throughout the study.

July 2011

Ziya ERCAN

(Electrical&Electronics Engineer)

TABLE OF CONTENTS

Page

ABBREVIATIONS

UGV : Unmanned Ground Vehicle
INS : Inertial Navigation System
GPS : Global Positioning System
DGPS : Differential Global Positioning System
RTK : Real Time Kinematics
DOP : Dilution of Precision
IMU : Inertial Measurement Unit
ISA : Inertial Sensor Assembly
KF : Kalman Filter
EKF : Extended Kalman Filter
UKF : Unscented Kalman Filter
DCM : Direction Cosine Matrix
LORAN : LOng RAnge Navigation
ECI : Earth Centered Inertial
ECEF : Earth Centered Earth Fixed

LIST OF TABLES

Page

LIST OF FIGURES

Page

MULTI-SENSOR AIDED INS/GPS INTEGRATION FOR AN UGV

SUMMARY

Research on Unmanned Ground Vehicle (UGV) is currently attracting a lot of interest. The ultimate objective is to increase safety by reducing traffic accidents caused by human faults. To reach this objective with desired reliability, these vehicles require the use of multiple sensors of various types and artificial intelligence.

A fundamental capability of a UGV is navigation. Using the information from various sensors, an UGV should be capable of determining vehicle's kinematic states, path planning and calculating the necessary maneuvers to move between desired locations. Estimation of the vehicle states is important to achieve other navigation tasks. The common way to estimate the vehicle states is the Inertial Navigation System (INS) and Global Positioning System (GPS) integration. Since both systems have complementary properties, they are well suited for data fusion using Kalman filters.

In this work, the accuracy of INS is improved by using a digital compass and a motor encoder. In orientation estimation, two Kalman filters are implemented in cascade to estimate the transformation matrix elements by using the measurements available in the system. An EKF is used to estimate the velocity of the vehicle using non-holonomic model with the motor encoder measurements. After INS stage is completed, an EKF is used to fuse the INS navigation results with GPS measurements.

The results show that the accuracy of INS is improved during long term GPS outages and the implemented system is much more accurate than the standart INS/GPS integration system.

INSANSIZ KARA ARACI İÇİN ÇOKLU ALGILAYICI YARDIMLI ANS/GKS ENTEGRASYONU

ÖZET

İnsansız Kara Araçları (İKA) hakkındaki araştırmalar şu anda çok ilgi çekici durumdadır. Bu araştırmaların en önemli hedefi insan hatalarından kaynaklana trafik kazalarını azaltarak kara yolu ulaşımında güvenliği arttırmaktır. Bu hedefe istenilen güvenilirlikte ulaşabilmek için, bu araçlar farklı görevlerde ve çeşitlerdeki çoklu algılayıcılara ve yapay zekaya ihtiyaç duyarlar.

Bir İKA'nın en temel yeteneklerinden biri navigasyondur. Bir çok farklı algılayıcılardan gelen bilgileri kullanarak, İKA kendi kinematik durumlarını belirleyebilme, güzergah planlama ve istenilen konumlara gidebilmek için gerekli manevraları hesaplayabilme kabiliyetlerine sahip olmalıdır. Aracın kinematik durumlarını kestirebilmek diğer navigasyon görevlerini başarıyla yerine getirebilmek için çok önemlidir. Aracın durumlarını kestirebilmek için kullanılan genel yöntem Ataletsel Navigasyon Sistemi (ANS) ve Global Konumlandırma Sistemi (GKS) entegrasyonudur. Bu iki sistem birbirlerinin tamamlayanı özellikler gösterdikleri için, Kalman Filtreleri kullanılarak bilgi birleştirmesine uygun durumdadırlar.

Bu çalışmada, ANS'nin hassasiyeti dijital bir pusula ve aracın motor enkoderi kullanılarak ilerletilmiştir. Oryantasyon kestiriminde, iki Kalman filtresi arka arkaya kullanılarak transformasyon matrisinin elemanları sistemde bulunan ölçümler yardımıyla kestirilmiştir. Genişletilmiş Kalman Filtresi (GKF) kullanılarak aracın hızının kestirimi holonomik olmayan kıstlar modeli ve motor enkoderi ölçümleri ile yapılmıştırANS kısmı bittiği zaman, GKF kullanılarak ANS navigasyon sonuçları GKS'den gelen ölçümlerle birlikte birleştirilmiştir.

Sonuçlar, uzun zamanlı GKS kesintileri sırasında ANS'nin hassasiyetinin geliştirlmiş olduğunu göstermektedir. Bu gerçeklenen sistem standart ANS/GKS birleştirici sistemlerinden çok daha fazla hassas olduğu gözlemlenmiştir.

Bu algoritma Labview kullanılarak gerçekleştirilmiştir ve bir IKA üzerinde yüklenip gerçek zamanlı test edilmiştir. Sonuçlar, uzun zamanlı GKS kesintileri sırasında ANS'nin hassasiyetinin geliştirlmiş olduğunu göstermektedir. Bu gerçeklenen sistem standart ANS/GKS birleştirici sistemlerinden çok daha fazla hassas olduğu gözlemlenmiştir.

1. INTRODUCTION

Unmanned Ground Vehicles (UGV) subject has become an important research area for both military and civilian use. UGV's are generally used for military missions, surveying in an unknown environment, landmine detection, mining and etc. In order to accomplish these tasks, reliable and high integrity navigation systems should be developed [1]. The term navigation encompasses two important subjects for autonomous vehicles. The first subject is the accurate determination of vehicle states (i.e. position, velocity, orientation) and the second is planning of the route and execution of necessary moves that is planned while avoiding obstacles and collisions [1, 2]. In this work, navigation is used to refer to the process of estimating the vehicle state in real-time as the vehicle maneuvers along a trajectory. A classical approach to vehicle state estimation is to equip the vehicle with inertial sensors capable of measuring the acceleration and angular rate of the vehicle. With proper calibration and initialization, integration of the angular rates provides an estimate of the attitude, while integration of acceleration provides estimates of velocity and position [1]. This navigation system is known as Inertial Navigation System (INS).

On the other hand, a navigation system may require an external infrastructure as well as user components, such as radio or satellite based navigation systems (i.e. LORAN, GPS and GLONASS). Recently, most vehicle navigation systems rely mainly on GPS as the main source of positioning. GPS provides reliable position information but a direct line of sight between the GPS antenna and GPS satellite is required. Also position accuracy depends on the number of visible satellites and the geometry of the satellites in space. For land vehicle navigation applications where the vehicle could go through a tunnel or near buildings, GPS outages could easily occur. So stand alone GPS navigation system is not reliable in this case.

1.1 Purpose of the Thesis

In order to achieve reliability and accuracy in navigation solutions, the integration of INS with GPS is proposed. The proposed integrated navigation system should provide reliable and accurate solutions all the time by overcoming any sensor errors, multipath effects and signal blockages. The characteristics of both systems show complementary properties. INS tends to have accumulated unbounded error growth but it has short time accuracy and has a high sampling rate. On the other hand, GPS tends to have small and non-accumulated errors but has a low sampling rate and is not reliable due to the outages. In order to overcome each system's weakness, these two systems are integrated into one system. Their integration is more accurate than a single solution [3].

The objective of this thesis is the development and implementation of an integrated navigation system for a driverless car application. The proposed system consists of various sensors to reach required reliability for navigation solutions in case of GPS outages or any sensor failures.

1.2 Literature Survey

In literature, several different types of INS/GPS integration techniques have been proposed and implemented. Recent research topics have been focused on improving the performance of low cost sensors by estimating the errors related to sensors. In this section, the previous works related to INS/GPS integration are given.

In [4], theoretical and practical development of low- cost, high integrity aided inertial navigation system for land vehicles is proposed. The fusion of INS/GPS is implemented using a Kalman Filter as the estimation algorithm. Information Filter approach is also proposed since there are a number of advantages associated with this approach. Fault detection techniques are developed to reach high integrity by providing a reliable navigation information, i.e. correcting any faults or rejecting bad data. Vehicle constraints model is also developed to be used as an extra aiding source with other aiding sensors to improve the accuracy and integrity of the overall navigation system.

In another study [5], a low-cost in-house constructed IMU and an off-the-shelf GPS receiver are used. A loosely coupled GPS aided INS approach is designed and tested with a data set obtained from a field test. IMU sensor errors are also modeled and a calibration method for the inertial sensors is provided. Also the algorithms are implemented in C++ for the use in a real time system. Unlike most INS algorithms implemented in navigation frame, ECEF frame approach is used in this work. The tuning of the covariance matrices and lever arm correction is presented to improve accuracy.

Different sensor fusion techniques are discussed and proposed in [6]. The main purpose of this work is to examine various possible configurations by which one might implement such a system using sensors like accelerometers, gyroscopes, GPS and odometer. On the other hand, this work also proposes several methods for improving the positionestimation capabilities of a system by incorporating other sensor and data technologies,including Kalman filtered inertial navigation systems, rule-based and fuzzy-based sensor fusion techniques, and a unique map-matching algorithm.

In one of study [7], the main contribution is the development of a new field calibration method for low cost IMU and GPS integration system. It is seen that almost half of the positioning error could be removed with the accelerometer calibration information and also non-holonomic constraints dramatically reduced the horizontal positioning error so that with the proposed calibration method, it is tested that low cost INSs can be used as a stand-alone positioning system during the GPS outages of over 10 minutes.

The Quaternion method in the computer frame approach is used to estimate orientation and an INS algorithm is derived for low cost IMU to solve the initial attitudes uncertainty using in-motion alignment in [8]. The main contribution of this paper is the development of an INS error model for large attitude errors in the computer frame approach using quaterions while most of the methods make small angle assumptions. In this quaternion method, the novelty is that quaternion errors are presented using the quaternions between the platform frame and the computer frame without small angle assumption. Also to avoid deriving the Jacobian matrices

of the EKF, distribution approximation filter (DAF) is used to implement non-linear data fusion algorithm.

In [9], a INS/GPS integration system for land vehicle application is developed and implemented. In order to have a full measurement of states of the error model kalman filter, a new orientation method is proposed. In this method, two cascaded Kalman filters are used to estimate orientation. A new stand alone roll-pitch estimation scheme is implemented using IMU data and yaw angle is obtained by switching between a magnetic compass and GPS based heading. With full measurement, it is shown that the integration system can be preserved from accumulated errors so it can run in a long time without restarting.

Some of the works are focused on using unscented Kalman filter (UKF) rather than extended Kalman filter (EKF) to improve accuracy of the algorithm. An IMU, GPS and a digital compass are combined by using an unscented Kalman filter to obtain an optimal state of the vehicle in [10]. The initial state of IMU is calibrated by using a digital compass. Only GPS position is used as a measurement in UKF. The dynamic equations are integrated by a fourth order Runge-Kutta approach instead of a first order Euler integration. The implemented system was evaluated by the experimentation of an equipped land vehicle.

In another study [11], a sigma point Kalman filter is derived for integrating GPS measurements with inertial measurements from IMU to determine both the position and the attitude of a moving vehicle. Sigma point filters use a carefully selected set of sample points to more accurately map the probability distribution than the linearization of the standard extended Kalman filter. Hence leading to a faster convergence. Simulation results are also shown to compare the performance of the sigma point filter with an extended Kalman Filter.

An architecture to fuse different data from onboard sensors to estimate the vehicle state using particle filter is proposed in [12]. An adaptive joint observation model has been developed to fuse different observations according to accuracy and reliability of the corresponding sensor. And also a navigation architecture has been proposed for fully autonomous driving with dynamic obstacles. Experiments with real vehicle

show the proposed method is able to estimate the vehicle state precisely when the individual observations fail to be enough accurate.

1.3 System Overview

In this work, a loosely coupled approach is used since there is no feedback/correction to the GPS (aiding system). This method is preferred because of its ease of use and less complexity. Also a feedback method is implemented to minimize the growth of observed error. An error state Kalman filter is designed as the fusion algorithm. This filter estimates the errors in the navigation solution with using the difference of INS solutions and GPS solutions as measurements. The estimated errors are used to correct the INS solution via feedback, which results in a closed loop system. The proposed method is tested with real-time data which was logged during a field test. Also this algorithm is developed in real-time so as to provide accurate and reliable navigation solution to an unmanned ground vehicle named as Otonobil.

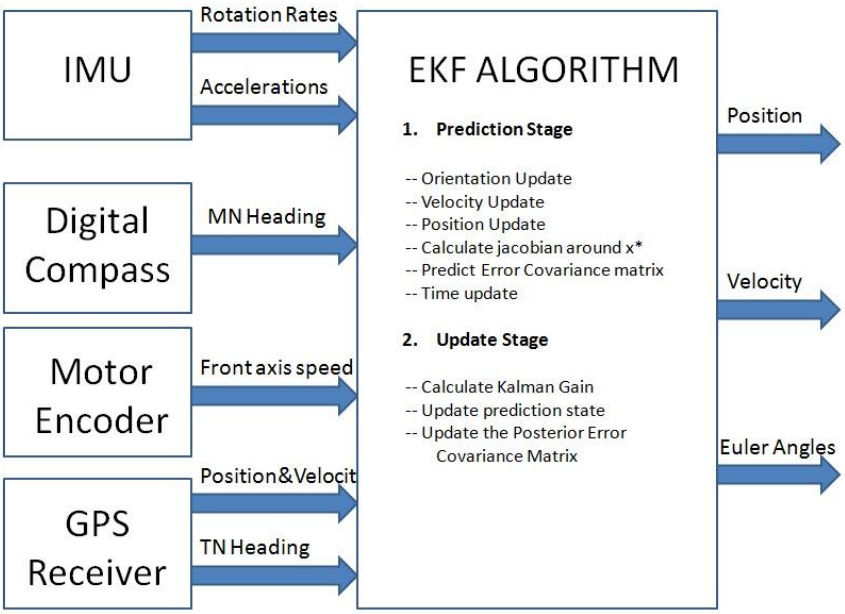


Figure 1.1: The overview of the integration system.

2. NAVIGATION MATHEMATICS

In this chapter, basic information about the mathematical foundations for explaining the principles of navigation is given. Since the some of the sensors measure their quantities in various frames, the navigation systems require the transformation of these quantities measured or computed in different frame of reference in order to use them in a common coordinate frame where the navigation process is performed. This chapter also introduces the concept of a coordinate frame and the main coordinate frames that are used in navigation systems. The objective of this chapter is to define various coordinate frames and the transformation techniques of information from one frame to another.

2.1 Concept of Coordinate Frame of Reference

A coordinate frame provides an origin and a set of axes in terms of which the motion of objects could be described. In other words, a coordinate frame provides a reference for the motion of objects. The position and orientation of an object is also defined by coordinate frames [2].

In navigation systems, unless otherwise stated, all the coordinate frames are assumed to have three axes, which are defined to be orthogonal and right handed, and have six degrees of freedom which are the position of the origin and the orientation of the axes. These must be expressed with respect to another frame in order to define them. So that makes a navigation system having at least two coordinate frames, the object frame and the reference frame. The object frame describes the body whose position and orientation is desired with respect to a known body whose position and orientation is described by a reference frame.

Some of the important notations are given below. These notations will be used throughout this work.

The reference frame in which a vector is represented is indicated by a superscript, i.e. $\mathbf{v^a}$ represents the vector **v** resolved in the reference frame **a**. Rotational transformation from one frame to another is represented by $\mathbf{R_a^b}$, which describes the rotation from origin frame of reference indicated by a subscript **a** to destination frame of reference which is indicated by a superscript **b**. For example a vector of quantity **v** represented with respect to the frame of reference **a** can be easily transformed to frame of reference **b** using the transformation below.

$$v^b = R_a^b v^a \quad \textbf{(2.1)}$$

Some kinematic quantities of an object or frame, i.e. angular rate or acceleration, are measured with respect to another frame in addition to being represented in a specific reference frame. For example, the notation $\mathbf{v_{ab}^c}$ describes the velocity vector of an object in frame **b** with respect to (relative to) frame **a**, resolved in frame **c**. It should be noticed that the object frame, the frame whose motion is described, i.e. notated as **a** in this case and the reference frame, the frame with which that motion is respect to, i.e. notated as **b** in this case must be different; otherwise there is no motion. Also with this notation, the given equation is always true.

$$v_{ab}^c = -v_{ba}^c \quad \textbf{(2.2)}$$

2.2 Coordinate Frames Used in Navigation Systems

In this section, five different coordinate frames which are commonly used in navigation systems are described. Each coordinate frame has different properties and some of them are rotating. The coordinate frames could be classified in three classes [5].

1. Earth Centered Systems
2. Local Coordinate Systems
3. Vehicle-centered Systems.

Detailed information about these coordinate frames is given subsequently in following sections.

2.2.1 Earth centered inertial frame (ECI frame)

An inertial coordinate frame is a non-accelerating and non-rotating reference frame in which Newton's laws of motion apply. For example all inertial sensors, like accelerometers and gyroscopes, output their measurements relative to an inertial frame resolving along their sensitive axes.

In navigation, a more specific form of the inertial frame should be defined with its origin and orthogonal axes. ECI frame is used for navigation purposes since it is convenient to have an origin coincided with ECEF origin, it is origin is chosen as Earth's center of mass. Anyone could easily notice that ECI frame is not an inertial frame strictly since Earth experiences acceleration in its orbit around the sun and its spin axis slowly moves and the Galaxy experiences rotation. But this frame is considered to be a sufficient approximation to an inertial frame [2].

The z-axis always points along the Earth spin axis (true North Pole) and x-axis points toward the vernal equinox and y-axis is defined to complete the right handed coordinate system. ECI frame is represented by a superscript i (i.e.v^i). Sometimes it is called *i*-frame.

2.2.2 Earth centered Earth fixed frame (ECEF frame)

The ECEF frame has its origin at the center of mass of Earth which is similar to ECI frame but its axes are fixed with respect to the Earth. Therefore all of its axes experience rotation with respect to the ECI frame at a rate of $\omega_{ie} = 7.292115\text{x}10^{-5}\ \text{rad/sec}$. This frame is denoted with a superscript $\boldsymbol{e}$. In figure 2.1, the relation between ECI frame and ECEF frame is shown. The vectors x_i, y_i, z_i define the axes of ECI frame and the vector $x_e$defines the x-axis of ECEF frame. From figure, it is seen that in a time instant the axes of both frames are coincident.

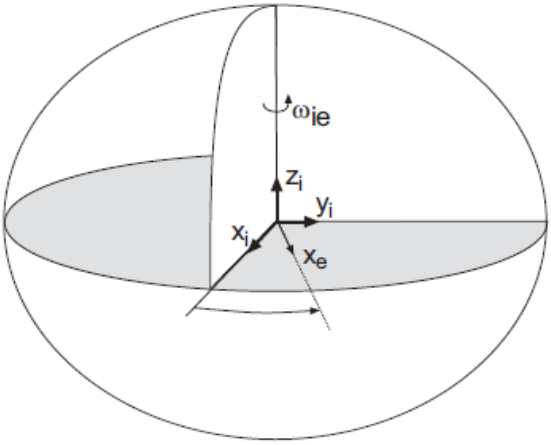


Figure 2.1: The rotation of ECEF frame with respect to ECI frame [1].

Figure 2.2 shows the orthogonal axes of ECEF frame. Its z-axis is coincident with spin axis and extends through the true North Pole and its x-axis points the intersection of the prime meridian (0° latitude) and the equator. The y-axis just completes the right-handed coordinate system that makes it extends through the intersection of 90° latitude and equator.

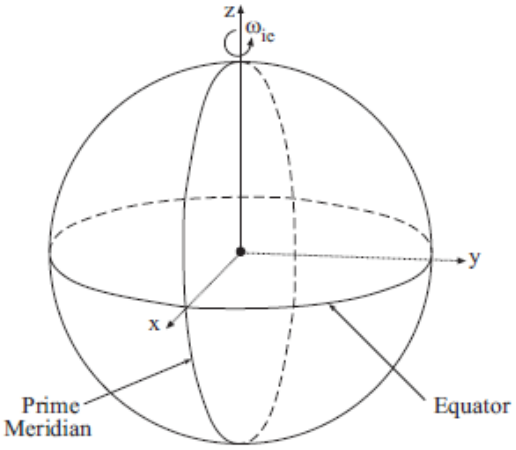


Figure 2.2: The origin and orthogonal axes of ECEF frame [1].

There are two different coordinate systems for describing the location of an object in the ECEF frame. The rectangular coordinates are defined as $(x, y, z)^e$ and are easy to obtain since they are directly determined from range measurements i.e. GPS range measurements. But in most of navigation applications, the geodetic coordinates $(\varphi, \lambda, h)^e$ are often desired since they are easy to understand. In geodetic coordinates,

latitude (φ) is the angle in the meridian plane from the equatorial plane to the ellipsoidal normal N. Longitude (λ) is the angle in the equatorial plane from the prime meridian to the projection of the point of interest onto the equatorial plane. Altitude (h) is the distance along the ellipsoidal normal between the surface of ellipsoid and the point of interest [1]. In figure 2.3, geodetic reference coordinate system is given. In both cross sections, the interpretation of the term longitude and latitude is shown.

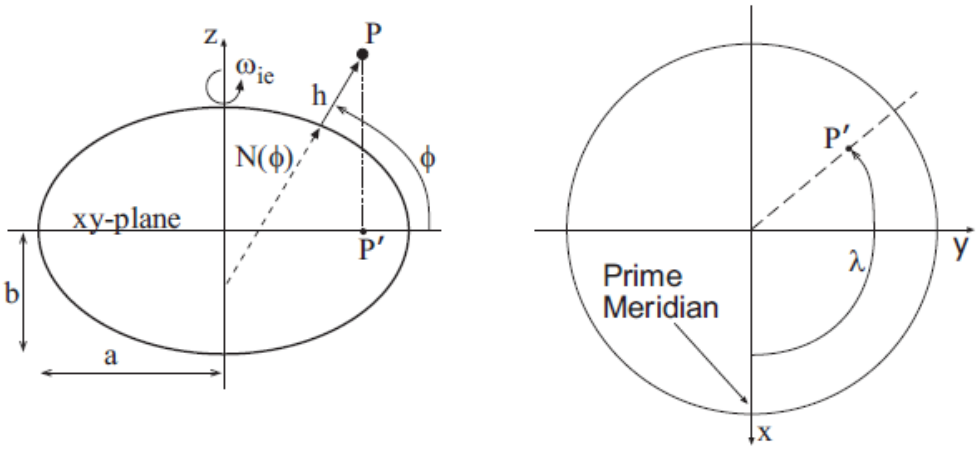


Figure 2.3: Geodetic coordinate system with interpretation of latitude and long. [1].

The ECEF frame is important in navigation since the users want to know their positions relative to the Earth.

2.2.3 Local navigation frame

Unlike ECEF frame, navigation frame has its origin coincident with the navigation system (i.e. the vehicle's center of mass). This frame is denoted with a superscript ***n***. The navigation frame is determined by fitting a tangent plane to the geodetic reference ellipse at a point of interest (i.e. center of mass of the object of interest).

There are different names of navigation frames and some of them have different properties. In this work, the navigation frame has origin coincident with the vehicle's center of gravity and its axes are defined as **n**orth, **e**ast and **d**own. The axes of navigation frame are given in the figure 2.4. The z-axis is defined as the down (D) axis and it is normal to the surface of the reference ellipsoid. The x-axis always

points to the true north thus defined as north (N) axis and the y-axis completes the orthogonal set, it always points east and hence it is known as east (E) axis.

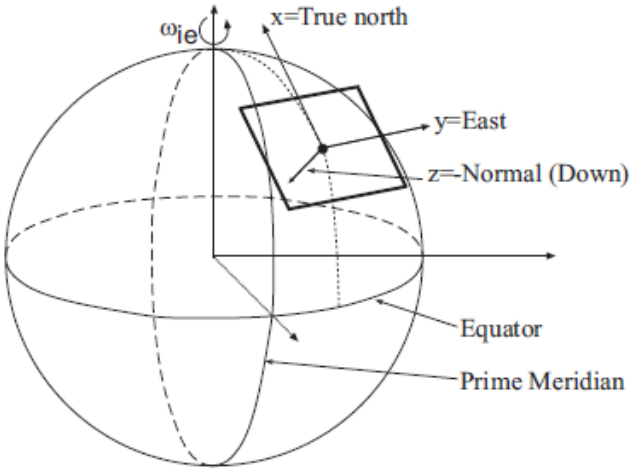


Figure 2.4: Navigation frame in relation to the ECEF frame [1].

In fact, navigation frame is frequently used in casual life because the user wants to know their attitude relative to north, east and down directions. In this way, navigation becomes easier to understand [2].

2.2.4 Vehicle body frame

The vehicle body frame has a great importance since it comprises the origin and the orientation of the object which is navigating. This coordinate frame has its origin at the center of gravity of the vehicle. In this way, the navigation frame's origin and the vehicle body frame's origin is coincident to ease the computation. The vehicle body frame is denoted with a superscript ***b***. The axes of this frame are fixed with respect to the vehicle where the x-axis is defined as forward which points the forward direction of the vehicle, the z-axis is defined as down which points the usual direction of gravity vector and the remaining y-axis is defined as transversal which completes the right handed orthogonal system. On the other hand if angular motion is considered, these axes are named differently. In this case, x-axis is defined as the roll axis, y-axis is defined as the pitch axis and z-axis is the yaw axis. The vehicle body system is shown in following figure 2.5.

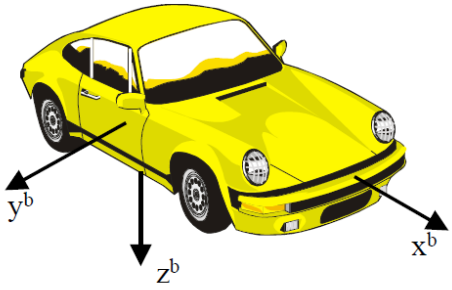


Figure 2.5: The body frame axes of a vehicle [7].

2.2.5 Instrumentation platform frame

Inertial sensors measure their quantities with respect to inertial frame resolved in their sensitive axes. When these sensors are placed and housed in a shell (i.e. inertial measurement unit), the alignment between the sensitive axes of sensors and the instrument platform axes is important. Ideally, these instruments are manufactured with aligned axes but in real conditions there may not be a perfect alignment. Therefore calibration and compensation algorithms are presented and programmed into the instrument. For low cost sensors, the user should design a calibration algorithm either at system initialization or during field operation [1].

In most navigation systems to ease the computation burden, the instrumentation platform frame axes are aligned with the vehicle body frame axes. In this case, there is no need for an extra transformation from instrumentation platform frame to vehicle body frame therefore reducing the computational cost. In this work, both frames are coincident.

2.3 Reference Frame Transformations

In this section, some important methods for transforming the points and vectors between rectangular coordinate frames are given.

2.3.1 Direction cosine matrix (DCM)

Consider a vector v_1 represented in a right-handed orthogonal frame a, from its origin $O_1$to an arbitrary chosen point P. The representation of this vector with respect

to frame *a*, using the orthogonal unit vectors I_1, J_1, K_1 **,** is given in (2.3) and the two dimensional illustration is given in figure 2.6.

$$v_1 = x_1 I_1 + y_1 J_1 + z_1 K_1 \quad \textbf{(2.3)}$$

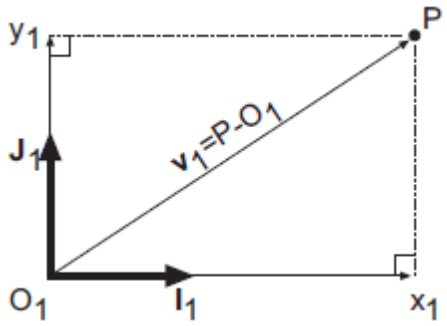


Figure 2.6: Vectors in 2D plane [1].

The vector $v_1^a = [x_1, y_1, z_1]^T$ is the representation of the vector $v_1$with respect to frame *a* and also the coordinates of the point *P* with respect to axes of frame *a*.

The same point can be easily represented in another reference frame *b*. This transformation requires two operations. These operations are translation and rotation. Let frame *b* has an origin at the point O_2 with the orthogonal unit vectors I_2, J_2, K_2 and let the vector v_2 represents a vector from O_2 to the point *P*. These vectors and frames are given in figure 2.6 which is a two dimensional illustration for easy understanding.

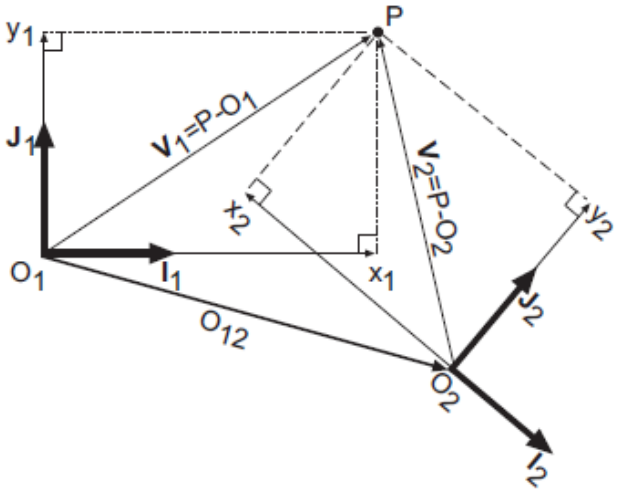


Figure 2.7: Illustration of two vectors in 2D plane [1].

Whether the vectors are represented in the coordinates of *a* frame or *b* frame, the following equation must hold.

$$v_1 = O_{12} + v_2 \quad \textbf{(2.4)}$$

If we assume that we have knowledge of the vector $v_2^b = [x_2, y_2, z_2]^T$ which is the representation of vector v_2 with respect to frame b and O_{12}^a which is the representation of the vector between two origin of the frames respect to frame *a*. In order to calculate the vector v_1^a, a transformation matrix R_b^a is needed to transform v_2^b into v_2^a. So that v_1^a is calculated as in (2.5). If the reference frames have the same origin then the calculation is simplified to (2.6).

$$v_1^a = O_{12}^1 + R_b^a v_2^b \quad \textbf{(2.5)}$$

$$v^a = R_b^a v^b \quad \textbf{(2.6)}$$

The transformation matrix R_b^a is a 3x3 matrix. It is also called as Direction Cosine Matrix. The name direction cosine is given since each element of R_b^a is the cosine of the angle between one of the unit vectors of I_1, J_1, K_1 and one of the unit vectors of I_2, J_2, K_2.

$$R_b^a = \begin{bmatrix} I_2.I_1 & J_2.I_1 & K_2.I_1 \\ I_2.J_1 & J_2.J_1 & K_2.J_1 \\ I_2.K_1 & J_2.K_1 & K_2.K_1 \end{bmatrix} = \begin{bmatrix} \cos\alpha_1 & \cos\beta_1 & \cos\gamma_1 \\ \cos\alpha_2 & \cos\beta_2 & \cos\gamma_2 \\ \cos\alpha_3 & \cos\beta_3 & \cos\gamma_3 \end{bmatrix} \quad \textbf{(2.7)}$$

Where $\alpha_i, \beta_i, \gamma_i$ (i=1, 2, 3) are the angles between the unit axes of the frames *a* and *b* (i.e. $\alpha_1$is the angle between I_1 of frame a and I_2 of frame b).

It should be noted that although DCM has nine elements, it has only three degrees of freedom due to the three normality constraints and the three orthogonality constraints [1].

2.3.2 Matrix transformation

In previous section, point transformation and vector transformation are given in (2.3) and (2.4) respectively. In this section, matrix transformation between frames of reference is discussed.

Let Φ^a be a 3x3 matrix defined with respect to frame a and let v_1^a and v_2^a be two vectors defined in frame a. If these two vectors are related by the matrix Φ^a as given below in (2.8), then the representation of the matrix in frame b is calculated as in (2.9, 2.10 and 2.11).

$$\mathrm{v}_1^{\mathrm{a}} = \Phi^{\mathrm{a}}\mathrm{v}_2^{\mathrm{a}} \quad \textbf{(2.8)}$$

$$\mathrm{R}_{\mathrm{b}}^{\mathrm{a}}\mathrm{v}_1^{\mathrm{b}} = \Phi^{\mathrm{a}}\mathrm{R}_{\mathrm{b}}^{\mathrm{a}}\mathrm{v}_2^{\mathrm{b}} \quad \textbf{(2.9)}$$

$$\mathrm{v}_1^{\mathrm{b}} = (\mathrm{R}_{\mathrm{b}}^{\mathrm{a}})^{-1}\Phi^{\mathrm{a}}\mathrm{R}_{\mathrm{b}}^{\mathrm{a}}\mathrm{v}_2^{\mathrm{b}} = \mathrm{R}_{\mathrm{a}}^{\mathrm{b}}\Phi^{\mathrm{a}}\mathrm{R}_{\mathrm{b}}^{\mathrm{a}}\mathrm{v}_2^{\mathrm{b}} = \Phi^{\mathrm{b}}\mathrm{v}_2^{\mathrm{b}} \quad \textbf{(2.10)}$$

$$\Phi^{\mathrm{b}} = \mathrm{R}_{\mathrm{a}}^{\mathrm{b}}\Phi^{\mathrm{a}}\mathrm{R}_{\mathrm{b}}^{\mathrm{a}} \quad \textbf{(2.11)}$$

2.4 Attitude Representations

In the previous section, the reference frame concept and transformations between frames were discussed. It was mentioned that the kinematic quantities such as acceleration and angular rates are described in three coordinate frames. These frames are the object frame, the reference frame and the resolving frame. In this section, three different forms of attitude representation will be described. These are Euler Attitude, Direction Cosine Matrix method and the Quaternion Attitude. Since there is no resolving frame for attitude, the object frame and the reference frame are involved. All these three representation methods describe the orientation of one coordinate frame with respect to other coordinate frame [2].

2.4.1 Euler attitude representation

Euler Attitude is the easiest way to describe an attitude with respect another frame (i.e. body frame with respect to navigation frame) because this method is easy to comprehend. It is known that the relationship between the vectors in the body frame and navigation frame can be described by a series of three plane rotations involving the Euler Angles which are roll angle(ϕ), pitch angle (θ) and yaw angle (ψ).

The transformation matrix R_n^b, is calculated by series of three plane rotations. It is assumed that the navigation frame axes are denoted as $(\mathrm{n}, \mathrm{e}, \mathrm{d})$ with the unit vectors

$(\mathrm{I}, \mathrm{J}, \mathrm{K})$ and the vehicle's body frame axes are denoted as $(\mathrm{u}, \mathrm{v}, \mathrm{w})$. The body frame has an origin located at the center of gravity of the vehicle and the navigation frame has an origin which is the projection of the vehicle's origin onto the ellipsoid. These two frames are shown in figure 2.8.

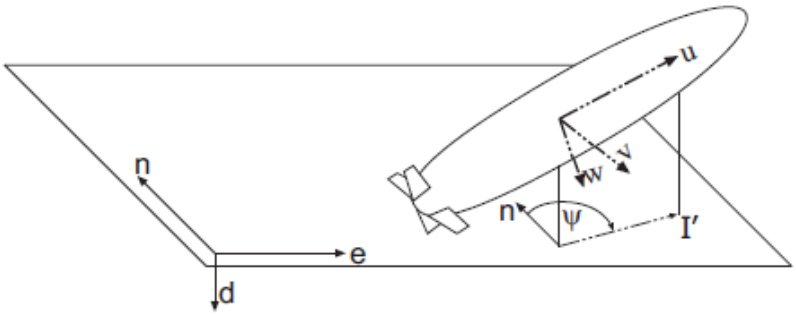


Figure 2.8: Navigation frame and vehicle body frame [1].

First rotation is yaw rotation which is performed along the navigation frame's d-axis, while n-axis and e-axis of the navigation frame is changed, d-axis of the navigation frame remains unchanged. Hence, yaw angle is the amount of rotation angle needed to align navigation frame's n-axis with the projection of body frame's u-axis onto the tangent plane to the ellipsoid as shown in figure 2.8. The resulting frame has axes (x', y', z') and unit vectors (I', J', K'). Yaw rotation is described as in (2.12) and figure 2.9.

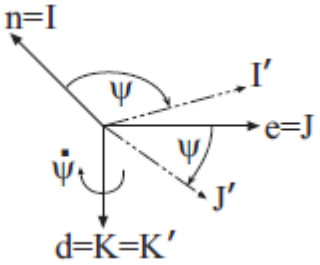


Figure 2.9: The yaw rotation around d axis [1].

$$\begin{bmatrix} \mathrm{x}' \\ \mathrm{y}' \\ \mathrm{z}' \end{bmatrix} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{n} \\ \mathrm{e} \\ \mathrm{d} \end{bmatrix} \quad \textbf{(2.12)}$$

Next, the pitch rotation is performed along the resulting frame's axis y', while x' and z' axes are changed and the rotation axis y' is unchanged. Pitch angle is the amount

of rotation angle needed to align resulting frame's x′ axis to align with the body frame's u-axis. The resulting frame has axes (x'', y'', z'') and unit vectors (I'', J'', K''). Pitch rotation is described as in (2.13) and figure 2.10.

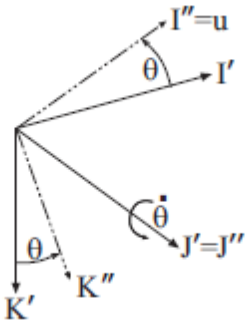


Figure 2.10 : Pitch rotation around y'-axis [1].

$$\begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \quad \textbf{(2.13)}$$

The last rotation is the roll rotation which is performed along the resulting frame's x″ axis, while y″ and z″ axes are changed and the rotation axis x″ remains unchanged. Roll is the amount of rotation angle needed to align y″ and z″ axes with the body frames v and w axes. So by the last rotation, all axes of navigation frame are aligned with the axes of body frame. Roll rotation is described as in (2.14) and figure 2.11.

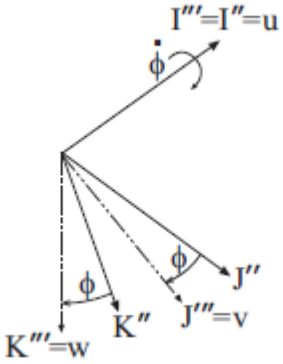


Figure 2.11: Roll rotation around x'' axis [1].

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} \quad \textbf{(2.14)}$$

These series of three rotations complete the transformation matrix R_n^b, which aligns the navigation frame axes with the body frame axes. The order of rotation is given in (2.15-16) to form the transformation matrix R_n^b and the resulting transformation matrix is given in (2.17).

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = [\phi][\theta][\psi] \begin{bmatrix} n \\ e \\ d \end{bmatrix} \quad \textbf{(2.15)}$$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n \\ e \\ d \end{bmatrix} \quad \textbf{(2.16)}$$

$$R_n^b = \begin{bmatrix} c(\psi)c(\theta) & s(\psi)c(\theta) & -s(\theta) \\ -s(\psi)c(\phi) + c(\psi)s(\theta)s(\phi) & c(\psi)c(\phi) + s(\psi)s(\theta)s(\phi) & c(\theta)s(\phi) \\ s(\psi)s(\phi) + c(\psi)s(\theta)c(\phi) & -c(\psi)s(\phi) + s(\psi)s(\theta)c(\phi) & c(\theta)c(\phi) \end{bmatrix} \quad \textbf{(2.17)}$$

As it was mentioned before the Euler angles describe the orientation of the object frame with respect to the reference frame. If the object frame is the vehicle body frame and the reference frame is the navigation frame, then the Euler angles have special names. In this case, roll rotation is known as bank, the pitch rotation is known as elevation. Sometimes, the term "attitude" is used for describing the bank and elevation, and sometimes bank and elevation is referred as tilt angles. And the yaw rotation is called as heading or azimuth [2].

The order of the rotation is important since the three Euler rotations do not commute, i.e. if roll rotation is performed first then different orientation will be resulted. Throughout this work, the rotation order will be denoted as $[\phi][\theta][\psi]$. In this rotation, first yaw rotation is performed followed by pitch and roll rotations

respectively. Sometimes, this order of rotation is called as zyx sequence. To reverse an Euler rotation, the original order of rotation should be reversed.

Another important property of Euler rotation is the rotation $[\phi + \pi][\pi - \theta][\psi + \pi]$ yields the same orientation with the rotation $[\phi][\theta][\psi]$. This causes singularity at $\theta = \pm\pi/2$ where the roll and yaw angle become indistinguishable. However, for land vehicle applications this singularity does not occur, since the pitch angle would not be equal to $\pm\pi/2$ in the field hopefully [1].

2.4.2 Direction cosine matrix method

In section 2.3.1, the concept of DCM was introduced and it was showed that the DCM elements are the angles between the unit vectors which describe the axes of object frame and the reference frame. DCM is also called as transformation matrix.

In previous section, the Euler angles were introduced and the relationship between the Euler angles and the DCM was shown by three consecutive plane rotations. DCM can be computed from a set of Euler angles by the equation (2.17).

Although, DCM has nine elements, it has only three independent elements due to the orthogonality and the normalization constraints. Thus, the DCM method has the same degree of freedom as Euler attitude and a reverse transformation of Euler angles from a given transformation matrix $\mathrm{R}_{\mathrm{b}}^{\mathrm{n}}$ can be calculated by the following equations given in (2.18-20) where atan2(y, x) is a four quadrant inverse tangent function and i.e. $\mathrm{R}_{\mathrm{b}}^{\mathrm{n}}[2,1]$ refers to the element of transformation matrix which is in second row and first column.

$$\psi = atan2(R_b^n[2,1], R_b^n[1,1]) \quad \textbf{(2.18)}$$

$$\phi = atan2(R_b^n[3,2], R_b^n[3,3]) \quad \textbf{(2.19)}$$

$$\theta = -\mathrm{atan}\left(\frac{R_b^n[3,1]}{\sqrt{1-(R_b^n[3,1])^2}}\right) \quad \textbf{(2.20)}$$

DCM method is easy to understand and the transformation matrix is easy to manipulate. The inverse of a transformation matrix yields the same result with its transpose. This property is given in (2.21). As given in (2.22), transformation

matrices are simply multiplied to perform successive rotations. Since the transformation matrices are orthogonal, the equation given in (2.23) always holds.

$$R_b^a = \left(R_a^b\right)^{-1} = \left(R_a^b\right)^T \quad \textbf{(2.21)}$$

$$R_b^a = R_c^a R_b^c \quad \textbf{(2.22)}$$

$$R_b^a R_a^b = I_{3x3} \quad \textbf{(2.23)}$$

2.4.3 Quaternion attitude representation

As mentioned in previous sections, the DCM method and the Euler Attitude suffers from singularity when the pitch angle is equal to $\pm\pi/2$. This is important because when singularity occurs, the yaw and roll angles are indistinguishable. But singularity is not a case for land vehicles assuming that pitch angle would not be equal to $\pm\pi/2$. This method presents singularity-free way to represent a transformation between two frames of reference. In this section a brief introduction to quaternion attitude is given. The reader should refer to [1-2] for details of this method.

A quaternion is a hyper-complex number which has four parameters i.e. $q = (q_0, q_1, q_2, q_3)$ and it can be represented by a four component complex number. In (2.24), a quaternion q is given where q_0 represents the magnitude of the rotation with basis 1 and (q_1, q_2, q_3) represent the axis where the rotation takes place with the basis of i, j, k .

$$q = q_0 + q_1 i + q_2 j + q_3 k \quad \textbf{(2.23)}$$

Quaternion attitude method was inspired by the idea that a transformation between two reference frames may be described as a single rotation about a vector E. Assume that n and b are the two frames of reference and if frame n is rotated by μ radians about the unit vector E, then the frame n will be aligned with frame b. The quaternion that represents the transformation from frame n to frame b is given as in (2.24) where i, j, k are components of rotation angle vector E [7].

$$q = \begin{pmatrix} \cos(\mu/2) \\ i\sin(\mu/2) \\ j\sin(\mu/2) \\ k\sin(\mu/2) \end{pmatrix} = \begin{pmatrix} \cos(\mu/2) \\ E\sin(\mu/2) \end{pmatrix} \quad \textbf{(2.24)}$$

It should be noted that the quaternion q has the normality property, i.e. $\|q\| = 1$ hence even if it has four parameters, it has only three degrees of freedom as the other attitude methods.

This method offers a computational efficiency, lack of trigonometric functions and singularities. But this idea of quaternion is not easy to comprehend and it makes navigation equations more difficult to follow. There are transformations to convert quaternion to DCM and quaternion to Euler angles easily. But they are not subject of this very brief introduction in this work.

2.5 Derivative Calculations In Rotating Reference Frames

In this section, kinematic properties of a reference frame with respect to another reference frame are discussed. The kinematic properties include the position, velocity, acceleration, orientation and angular velocity of a vector. As it was stated before, all three frames i.e. reference frame, object frame and resolving frame, should be defined to describe those kinematic properties.

In non-rotating frames, assuming no friction, the motion of an object depends only on applied force. But in rotating frames, the motion of an object also depends on virtual forces called Coriolis force and centrifugal force.

In section 2.5.1 the rate of change of direction cosine matrix for the frames experiencing relative rotation is introduced.

In next section, 2.5.2, the equations of motion of an object whose reference axes experience relative rotation are calculated and also Law of Coriolis is introduced basically.

2.5.1 Time derivative of direction cosine matrix

The time derivative of DCM is calculated using the small angle approximation which is the transformation between two reference frames which experience very

infinitesimal rotation in relative orientation. This transformation is convenient for calculating time derivative of DCM since there exists small angle differences valid at two infinitesimally different instants of time.

Consider a vector v in two reference of frames frame *a* and frame *b* where frame b is obtained from infinitesimal rotation of frame *a* by series of plane rotations. The infinitesimal rotation is given as $\delta\alpha = [\delta\alpha_1 \quad \delta\alpha_2 \quad \delta\alpha_3]^{\mathrm{T}}$ and the plane rotations, which were described in section 2.4.1, are given in (2.25).

$$v^b = [\delta\alpha_1][\delta\alpha_2][\delta\alpha_3]v^a \tag{2.25}$$

Under infinitesimal angle assumptions, let $\sin(\delta\alpha) \approx \delta\alpha$ and $\cos(\delta\alpha) \approx 1$, the small angle transformation matrix becomes as in (2.26).

$$\mathrm{v^b} = \begin{bmatrix} 1 & \delta\alpha_3 & -\delta\alpha_2 \\ -\delta\alpha_3 & 1 & \delta\alpha_1 \\ \delta\alpha_2 & -\delta\alpha_1 & 1 \end{bmatrix} \mathrm{v^a} \tag{2.26}$$

The small angle transformation matrix is 3x3 identity matrix minus the skew-symmetric representation of the infinitesimal rotation angle $\delta\alpha$ denoted as $\Omega_{\delta\alpha}$. So equation (2.26) could be written as in (2.27).

$$\mathrm{v^b} = (\mathrm{I_{3x3}} - \Omega_{\delta\alpha})\mathrm{v^a} \tag{2.27}$$

By the definition of time derivative, the derivative of the transformation matrix from frame *a* to frame *b* could be defined as in (2.28).

$$\dot{R}_a^b(t) = \lim_{\delta t \to 0} \frac{R_a^b(t+\delta t) - R_a^b(t)}{\delta t} \tag{2.28}$$

By using small angle assumption in equation (2.28) is simplified and the derivative of transformation matrix is defined as in (2.29) where $\omega_{\mathrm{ab}}^{\mathrm{b}}$ is the instantaneous angular rate of frame b with respect to frame *a* and skew-symmetric matrix of this angular rate is denoted as $\Omega_{\mathrm{ab}}^{\mathrm{b}} = \left[\omega_{\mathrm{ab}}^{\mathrm{b}}\mathrm{x}\right]$.

$$\dot{\mathrm{R}}_{\mathrm{a}}^{\mathrm{b}} = \mathrm{R}_{\mathrm{a}}^{\mathrm{b}}\Omega_{\mathrm{ab}}^{\mathrm{b}} \tag{2.29}$$

2.5.2 Equations of motion in rotating reference frames

Consider two frames of reference, frame *a* and frame *b* which have unit vectors of $(I^a \quad J^a \quad K^a)$ and $(I^b \quad J^b \quad K^b)$ respectively. Assume that those frames are experiencing relative rotation i.e. frame *a* is rotating with respect to frame *b* with an angular velocity of ω_{ba}^{a}.

Let P is a point or an object in these frames of reference illustrated in figure 2.12. The vector p represents the vector from the origin of frame b i.e. O_b to the point *P*. The vector r represents the vector from frame *a* origin i.e. O_a to the point *P* and also since these two frames have distinct origins, the vector ρ represents the vector from O_b to O_a. If the origins of reference frames are coincident then it is obvious that $\rho = \vec{0}$. All of these vectors and frames are illustrated in figure 2.12.

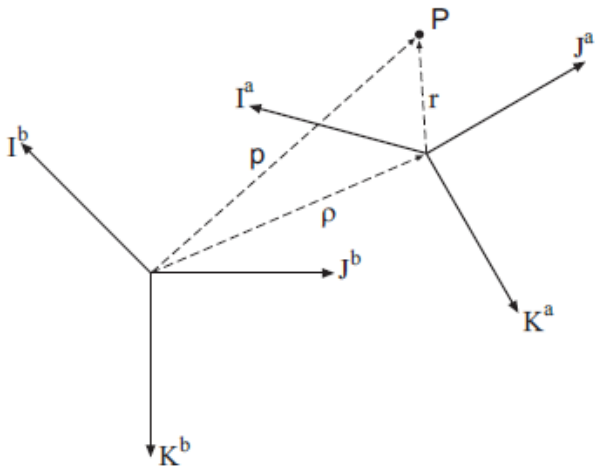


Figure 2.12: Two frames of references [1].

Assume that we have a priori knowledge of ρ^b,R_a^b and r^a where the superscripts denote the resolving frame. Then then the position of point *P* with respect to the frame b resolved in axes of frame b can be found by the following equation (2.30).

$$p^b = \rho^b + R_a^b r^a \qquad \textbf{(2.30)}$$

The velocity of an object is defined to be the instantaneous rate of change of the position of the object frame with respect to a reference frame where the resolving

frame does not affect thus it could be written as in (2.31). This definition only holds if the resolving frame is not rotating with respect to the reference frame.

$$\mathrm{v}_{\mathrm{ab}}^{\mathrm{c}} = \mathrm{R}_{\mathrm{b}}^{\mathrm{c}} \frac{\mathrm{dr}_{\mathrm{ab}}^{\mathrm{b}}}{\mathrm{dt}} = \mathrm{R}_{\mathrm{b}}^{\mathrm{c}} \dot{\mathrm{r}}_{\mathrm{ab}}^{\mathrm{b}} \tag{2.31}$$

When the example in figure 2.12 is considered and the time derivative of equation (2.30) is calculated, then the instantaneous velocity of p^b is found out to be equal to the equation (2.32).

$$\frac{dp^b}{dt} = \frac{d\rho^b}{dt} + R_a^b(\Omega_{ba}^a \mathrm{r}^{\mathrm{a}} + \frac{dr^a}{dt}) \tag{2.32}$$

The first term on the right hand side of the equation is the relative instantaneous velocity of the two frames. This term exists unless the origins of two frames are coincident. The second term ($\Omega_{ba}^a \mathrm{r}^{\mathrm{a}}$) represents the instantaneous velocity of point *P* with respect to frame a formed by the relative rotation of the frames. This is a virtual velocity caused by the rotating frames. The last term $\left(\frac{\mathrm{dr}^{\mathrm{a}}}{\mathrm{dt}}\right)$ is the instantaneous velocity of point *P* with respect to frame *a*. This term could be defined as in (2.33).

$$\frac{\mathrm{dr}^{\mathrm{a}}}{\mathrm{dt}} = \dot{\mathrm{r}}^{\mathrm{a}} = \mathrm{v}^{\mathrm{a}} \tag{2.33}$$

The acceleration of an object is defined to be the instantaneous rate of change of the velocity of the object frame with respect to a reference frame resolved in resolving frame. For non-rotating frames, the acceleration could be defined as given in (2.34).

$$a_{ab}^c = R_b^c \frac{d^2 r_{ab}^b}{dt^2} = R_b^c \ddot{r}_{ab}^b = R_b^c \dot{v}_{ab}^b \tag{2.34}$$

If the resolving frame and the reference frame are experiencing relative angular rotation, then equation (2.34) does not hold. The acceleration and the velocity depend on the rotation rate of resolving frame with respect to the reference frame. This effect is explained in the theorem given below named as *Law of Coriolis*.

Theorem 1: If the reference frame (frame *b*) and the resolving frame (frame *c*) are experiencing relative angular rotation $\omega_{\mathrm{ba}}^{\mathrm{c}}$, then the time derivative of vector $v^b$where $\mathrm{v}^{\mathrm{b}} = \mathrm{R}_{\mathrm{c}}^{\mathrm{b}} \mathrm{v}^{\mathrm{c}}$ and $\dot{\mathrm{R}}_{\mathrm{c}}^{\mathrm{b}} = \mathrm{R}_{\mathrm{c}}^{\mathrm{b}} \Omega_{\mathrm{ba}}^{\mathrm{c}}$, is given as in (2.35). This relationship

of the time derivative of the vector in the two coordinate systems is known as Law of Coriolis.

$$R_b^c \dot{v}^b = [\Omega_{ba}^c v^c + \dot{v}^a] \quad \textbf{(2.35)}$$

So if the derivative of equation (2.32) is taken, the acceleration of a point in rotating coordinate frames is found as in (2.36-37). Also the acceleration of vector *r* with respect to frame *b* is given in (2.38).

$$\frac{d^2 p^b}{dt^2} = \frac{d^2 \rho^b}{dt^2} + R_a^b \left[2\Omega_{ba}^a v^a + \Omega_{ba}^a \Omega_{ba}^a r^a + \dot{\Omega}_{ba}^a r^a + \frac{dv^a}{dt} \right] \quad \textbf{(2.36)}$$

$$\frac{d^2 p^b}{dt^2} = \frac{d^2 \rho^b}{dt^2} + \left[2\Omega_{ba}^a \frac{dr^b}{dt} + \Omega_{ba}^a \Omega_{ba}^a r^b + \dot{\Omega}_{ba}^a r^b + \frac{d^2 r^b}{dt^2} \right] \quad \textbf{(2.37)}$$

$$\frac{d^2 r^b}{dt^2} = \frac{d^2 p^b}{dt^2} - \frac{d^2 \rho^b}{dt^2} - 2\Omega_{ba}^a \frac{dr^b}{dt} - \Omega_{ba}^a \Omega_{ba}^a r^b - \dot{\Omega}_{ba}^a r^b \quad \textbf{(2.38)}$$

If the elements of equation (2.38) are discussed, it is seen that the motion of an object whose resolving axes experience relative angular rotation with reference frame, also depends on virtual forces as well as the applied force.

The first element on the right hand side of the equation is the acceleration of point P relative to origin of frame b. The second element is the acceleration of two frames. Again this term is zero if the origins of two frames are coincident. The third element is the Coriolis force and the fourth element is the centrifugal force. Last element accounts for the Euler acceleration.

3. NAVIGATION SYSTEMS

In this chapter, two important navigation systems and their components are introduced. In section 3.1, inertial navigation is briefly mentioned where in section 3.2, satellite navigation systems are introduced. For more details, the reader should refer to the references [1, 2].

3.1 Inertial Navigation

Inertial navigation is a method for calculating the pose of the vehicle i.e. position, velocity and orientation by using inertial sensors. The calculation of velocity, position and orientation of the vehicle is based on dead-reckoning since the change in velocity and the change in orientation, which are measured by the inertial sensors, are added to previous estimates of velocity and orientation to obtain current estimates. Also the step size is very important for accuracy of the current estimates. So the smaller step size yields a more accurate navigation solution. It is important to note that for two-dimensional navigation, only yaw (heading) angle is enough but for three- dimensional navigation, all Euler angles are needed.

The idea behind the inertial navigation is Newton's first and second laws. Newton's first law states that an object keeps its current state and uniform motion until force is applied. And Newton's second law states that, the acceleration of an object is proportional and in the same direction as the applied force and inversely proportional to its mass [2].

In section 3.1.1, a brief introduction about the inertial sensors and their working principles will be given. In section 3.1.2, inertial systems will be introduced and in the last section the errors associated with the inertial sensors will be discussed.

3.1.1 Inertial sensors

This section describes the basic principles of inertial sensors which are accelerometer and gyroscope. Inertial sensors measure their quantities with respect to an inertial frame. Sensors that measure acceleration or angular rate with respect to another frame or a feature in the environment are not inertial sensors. For land vehicle applications, inertial sensors make measurements of the internal states of the vehicle i.e. acceleration and angular rate.

It is important to note that current inertial sensor development is focused on MEMS technology which offers low-cost and smaller size but on the other hand they offer relatively poor performance. There are various kinds of inertial sensors depending on performance and price.

3.1.1.1 Accelerometers

Accelerometers are based on Newton's second law as stated in previous section and given in (3.1) where F denotes the physically applied forces, m denotes the mass of the object and $\ddot{r}^i$ denotes the acceleration of the object in inertial frame i.e. second time derivative of the object's position.

$$F = m\ddot{r}^i \quad \textbf{(3.1)}$$

A simple accelerometer is given in figure 3.1. A proof mass is mounted in the accelerometer's case along its sensitive axis. This mass is free to move but its movement is restrained by the springs. The position of the proof mass with respect to the case is measured by a pickoff. When an accelerating force along the sensitive axis is applied, the case will move with respect to the mass, one spring is compressed and the other is stretched. A displacement will occur. The pickoff measures this displacement which is proportional to the force that is applied. Besides the non-gravitational force, the gravitation also acts on the proof mass directly and it applies the same acceleration to all components so there is no relative motion with respect to the case [2].

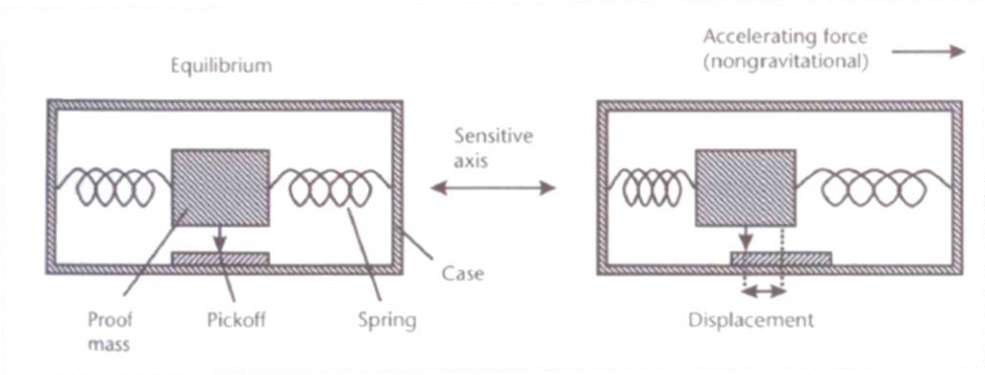


Figure 3.1: A simple accelerometer structure [2].

Accelerometers measure specific force about its sensitive axis with respect to an inertial frame which is non-accelerating, non-rotating and has no gravitational field. The specific force is the inertial force (i.e. spring forces, friction, lift etc.) per unit mass required to produce acceleration which is given in (3.1) as $\ddot{r}^i$, therefore if we denote specific force as f then $f = {}^{F}/_{m}$ [2].

But when the inertial frame has a gravitational field as Earth, then the acceleration of the object becomes as in (3.2) where G^i is the gravitational acceleration applied on the object and this equation is referred as the fundamental equation of the inertial navigation in inertial reference frame. So the accelerometers measure the specific force vector given in (3.3).

$$\ddot{r}^i = f^i + G^i \tag{3.2}$$

$$f^i = \ddot{r}^i - G^i \tag{3.3}$$

It could be important to consider some special cases associated with the measurement of specific force. Gravitational force is not sensed by the accelerometer because it acts all points equally however other forces are transmitted from point to point so they are sensed. The weight is sensed due to the forces opposing gravity. So if an accelerometer is in free-fall, the specific force will be zero because there is no opposing force to gravity. If the accelerometer is at rest on Earth's surface, it will measure the opposing force which is the reaction to gravity and the specific force is equal but opposite direction to gravity. For more detailed analysis and information on accelerometers, the reader should refer to [1, 2].

3.1.1.2 Gyroscopes

The gyroscopes often abbreviated as gyros, measure the angular rate of the sensitive axes with respect to the inertial frame. The orientation of the vehicle could be found by integrating the gyro measurements with respect to an initial orientation.

There are three main types of gyro technology. A spinning mass gyro operates on the principle of conversation of angular momentum. In spinning-mass gyroscopes also known as gyrocompass, a spinning mass which is free to rotate about the axes that are perpendicular to its spin axis is mounted in a case. This spinning mass will remain aligned with respect to inertial frame as the case is rotated and a pickoff provides measurements of the case's attitude about two axes and it measures the orientation of the spinning mass. In figure 3.2, a basic spinning mass gyro is given.

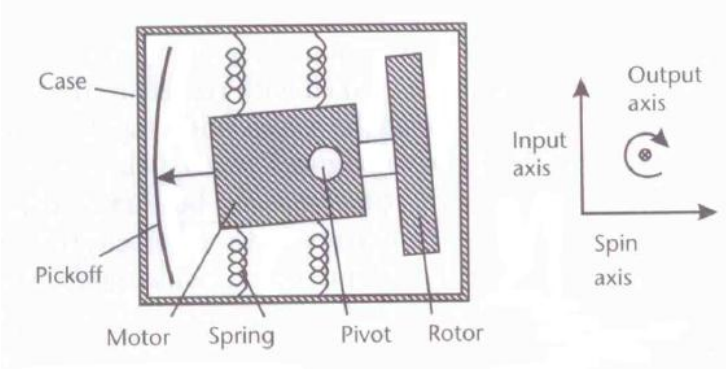


Figure 3.2: Basic spinning gyro structure [2].

Optical gyros work on the principle that light travels at a constant speed in a given medium. Using the Sagnac effect, the change in orientation, is determined by measuring the changes in path lengths [2].

Vibratory gyros operate on the principle to detect the Coriolis acceleration of a vibrating element i.e. string, beam, ring etc, when the gyro is rotated using the principles of harmonic motion. For more detailed analysis and information on accelerometers, the reader should refer to [1, 2].

3.1.2 Inertial systems

Generally, three accelerometers and three gyros are used in an orthogonal arrangement in three axes to measure the dynamics of the vehicle. Using this

implementation, the pose of this vehicle could be easily determined. The three-orthogonal installed accelerometers measure the specific force of the platform frame with respect to an inertial frame resolved in their sensitive axes. Also the three-orthogonal installed gyros measure the angular rate of the platform frame with respect to an inertial frame resolved in their sensitive axes. This type of inertial system is known as an "*inertial sensor assembly"* (ISA) which is illustrated in figure 3.3 [4].

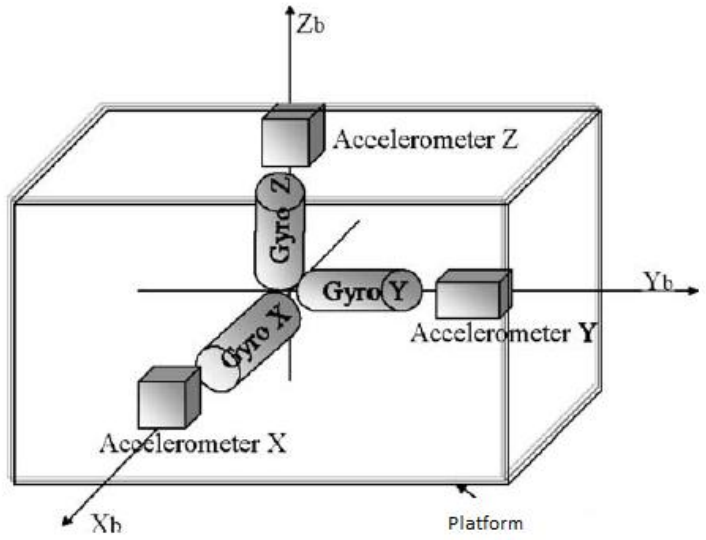


Figure 3.3: Inertial sensor assembly [8].

If the sensitive axes of the ISA are aligned with the axes of the vehicle's body frame then the accelerometers measure the acceleration of the vehicle in three axes while the gyros provide the rotation rates of body frame axes. This is illustrated in figure 3.4 where x, y and z axes are denoted as the axes of the vehicle which are forward, lateral and down direction respectively. The measurements of the accelerometers are denoted as in (3.4). The gyros provide the measurements of angular rates of the body frame denoted in (3.5) where $\dot{\theta}, \dot{\phi}$ and$\dot{\psi}$ represent the rotation rates in figure 3.4.

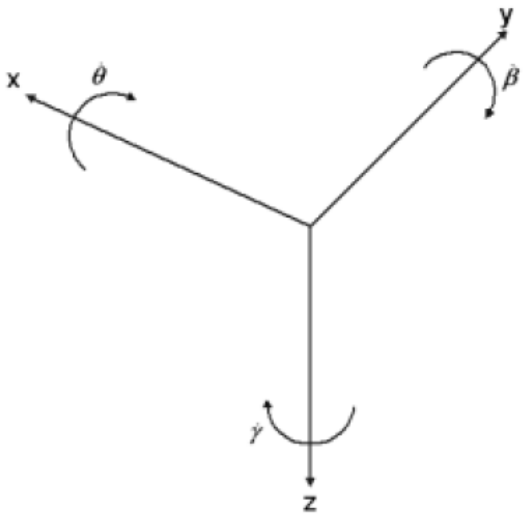


Figure 3.4 Three orthogonal accelerations and angular rates [4].

$$\tilde{f}_{ib}^{b} = \begin{bmatrix} \tilde{f}_{ib,x}^{b} \\ \tilde{f}_{ib,y}^{b} \\ \tilde{f}_{ib,z}^{b} \end{bmatrix} \quad \textbf{(3.4)}$$

$$\widetilde{w}_{ib}^{b} = \begin{bmatrix} \widetilde{w}_{ib,x}^{b} \\ \widetilde{w}_{ib,y}^{b} \\ \widetilde{w}_{ib,z}^{b} \end{bmatrix} \quad \textbf{(3.5)}$$

Besides three accelerometers and three single degree of freedom gyros, if a processor, a calibration-parameters store, a temperature sensor and etc. are also installed, then this inertial system is called "*inertial measurement unit"* (IMU). The outputs of inertial sensors are compensated for deterministic errors such as bias, scale factor and misalignment of axes and etc by using IMU's processor and calibration parameters stored within the unit. Also this processor performs unit conversions and range checks to detect any sensor failure. Internal architecture of an IMU is illustrated in the figure 3.5 [2].

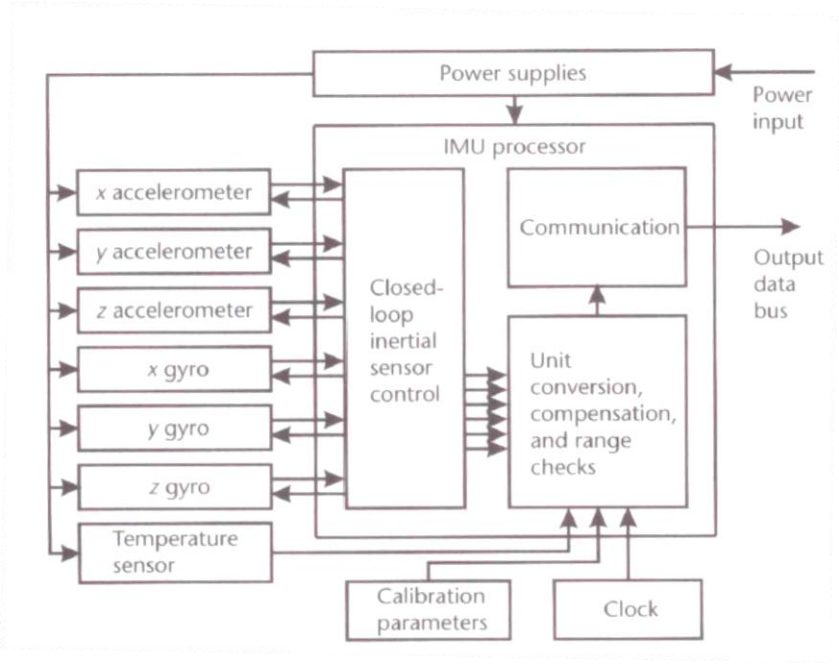


Figure 3.5: IMU inside architecture [2].

There are two ways of physical IMU implementation. If the inertial sensors are mounted on a gimballed platform where this platform is always aligned with the navigation frame then this implementation is called gimballed arrangement. Since the sensors are aligned with the navigation frame, there is no need for calculating the transformation matrix. But this mechanism is mechanical and suffers from its size and maintenance.

In a strapdown arrangement, the inertial sensors are mounted directly on a platform which is also mounted on the vehicle. In this configuration, transformation matrix has to be calculated by using measurements of gyros to align with the navigation frame. This introduces a computational burden and the sensors experience all effects of vehicle motion which results in noisy measurements and larger non-linearity errors. The advantage of this configuration over gimballed configuration is there is no mechanical system hence the size is small and it is not expensive. The system shown in figure 3.3 is also a strapdown configuration [2].

IMU's are used in navigation algorithm to calculate the position, velocity and the orientation of the vehicle by integrating the IMU's measurements. This inertial system is called "*inertial navigation system"* (INS). This system is a dead-reckoning

system using IMU and a navigation processor which performs navigation algorithms. Figure 3.6 illustrates an INS with three phases. The first phase is the input phase where the IMU is sampled and specific force measurement $\tilde{f}_{ib}^{b}$ and angular rate measurement $\tilde{\omega}_{ib}^{b}$ are outputted. In the second phase, the initial conditions of INS algorithm are determined. The last phase is the navigation phase which has four steps. At first step, the orientation is updated then the specific force measured from IMU is transformed to acceleration by a gravity model then this acceleration is transformed to navigation frame. First integral gives the velocity of this vehicle in navigation frame and the second integral gives the position of this vehicle in navigation frame [8].

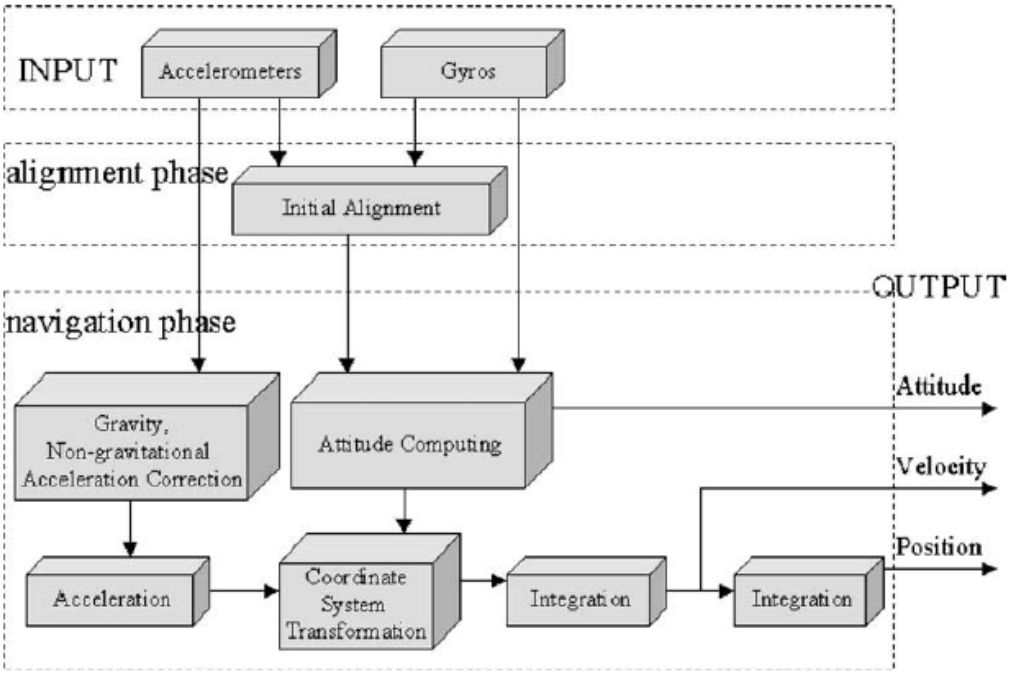


Figure 3.6: The components of INS algorithm [8].

3.1.3 Error characteristics of inertial sensors

In this section, the error characteristics of inertial sensors will be introduced. The error characteristics depend on the cost of the sensor. Low-cost sensors have relatively higher error characteristics than expensive sensors.

Nevertheless, all types of inertial sensors exhibit systematic errors like biases, scale factor, random walk and random noise. Since INS uses inertial sensors for

calculating position, velocity and orientation, any errors associated with inertial sensors will cause an accumulation in navigation errors dependent on time. INS is a dead reckoning system so any error in the previous solution will be carried to the next solution as time progresses and thus there will be larger errors with time.

Each systematic error consists of four components. A fixed part which is presented in measurements each time the sensor is used, but it is compensated by IMU processor. Temperature-dependent variation part can also be compensated by IMU and this error varies by temperature changes in the sensor. A run-to-run variation part changes in each time the sensor is used but it remains constant within any run. This type error is stochastic so it can't be corrected by IMU processor. And the last error associated with the inertial sensors is in-run variation part which contributes an error that is slowly changing during in each run. This error source is difficult to observe and it can't be corrected by IMU processor [2].

The measurements of inertial sensors, in this case accelerometers and gyros, could be modeled as in (3.6) and (3.7). The measurements of accelerometer and gyro are denoted as $\tilde{f}_{ib}^{b}$ and $\tilde{\omega}_{ib}^{b}$.

$$\tilde{\mathrm{f}}_{\mathrm{ib}}^{\mathrm{b}} = \mathrm{b_a} + (\mathrm{I_3} + \mathrm{M_a})\mathrm{f}_{\mathrm{ib}}^{\mathrm{b}} + \mathrm{w_a} \tag{3.6}$$

$$\tilde{\omega}_{\mathrm{ib}}^{\mathrm{b}} = \mathrm{b_g} + \left(\mathrm{I_3} + \mathrm{M_g}\right)\omega_{\mathrm{ib}}^{\mathrm{b}} + \mathrm{G_g}\mathrm{f}_{\mathrm{ib}}^{\mathrm{b}} + \mathrm{w_g} \tag{3.7}$$

The bias error source is denoted as b_a and b_g for accelerometer and gyro respectively. Both bias terms have two parts, static and dynamic. The static part of the bias consists of fixed bias and turn on bias where the dynamic part of the bias consists of in-run bias and run-to-run bias. It is important to note that static bias part has more effect on bias error since the dynamic part is only 10% of the static bias.

The accelerometer bias error is usually quoted in SI units (m/s^2) or in terms of acceleration of gravity (mg). The gyro bias error is quoted in SI units (rad/s) or in terms of degrees per hour ($°/hr$).

The scale factor error and cross-coupling errors are denoted as matrices M_a and M_g for accelerometers and gyros. These matrices have elements of scale factor errors and

cross-coupling errors of orthogonal tri-axial accelerometers and gyros. These matrices are given in (3.8) and (3.9).

The scale factor error is the error in input-output gradient of the sensor and this type of error is denoted as $s_{a,x}$ and $s_{g,x}$ which is the scale factor error of axis- x of accelerometer and gyro respectively and this error is illustrated in figure 8. The cross-coupling error is caused by the misalignment of the sensitive axes of the inertial sensors and also it is known as misalignment error. This type of error is denoted as $m_{a,yx}$ i.e. the cross-coupling coefficient of x-axis sensed by y-axis accelerometer. The cross-coupling error is illustrated in figure 3.7. So these errors affect the true quantities of specific force and angular rate as given in (3.10) and (3.11) [2].

$$\mathrm{M_a} = \begin{bmatrix} \mathrm{s_{a,x}} & \mathrm{m_{a,xy}} & \mathrm{m_{a,xz}} \\ \mathrm{m_{a,yx}} & \mathrm{s_{a,y}} & \mathrm{m_{a,yz}} \\ \mathrm{m_{a,zx}} & \mathrm{m_{a,zy}} & \mathrm{s_{a,z}} \end{bmatrix} \tag{3.8}$$

$$\mathrm{M_g} = \begin{bmatrix} \mathrm{s_{g,x}} & \mathrm{m_{g,xy}} & \mathrm{m_{g,xz}} \\ \mathrm{m_{g,yx}} & \mathrm{s_{g,y}} & \mathrm{m_{g,yz}} \\ \mathrm{m_{g,zx}} & \mathrm{m_{g,zy}} & \mathrm{s_{g,z}} \end{bmatrix} \tag{3.9}$$

$$\tilde{\mathrm{f}}_{\mathrm{ib}}^{\mathrm{b}} = (\mathrm{I_3} + \mathrm{M_a})\mathrm{f}_{\mathrm{ib}}^{\mathrm{b}} \tag{3.10}$$

$$\tilde{\omega}_{\mathrm{ib}}^{\mathrm{b}} = \left(\mathrm{I_3} + \mathrm{M_g}\right)\omega_{\mathrm{ib}}^{\mathrm{b}} \tag{3.11}$$

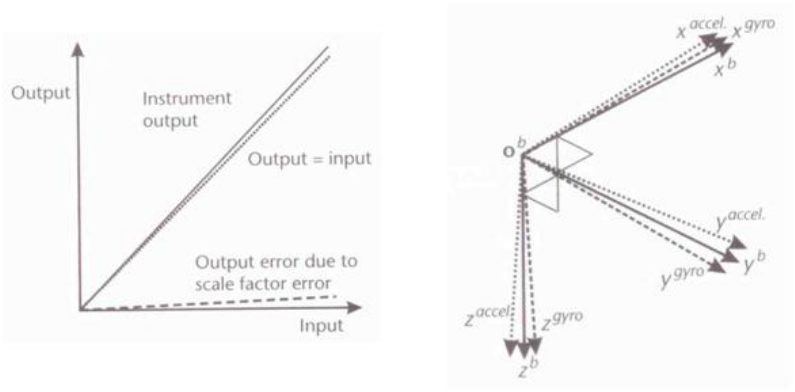


Figure 3.7: Scale factor and misalignment errors [2].

The random noise that is exhibited from inertial sensors are denoted as w_a and w_g. There are number of different sources for random noise especially MEMS sensors

exhibit a significant high frequency noise since the electrical noise limits the resolution of inertial sensors.

Also another error in gyro measurements is g-dependent bias due to mass unbalance which is denoted as G_g, a 3x3 matrix since gyros can be sensitive to accelerations all along three axes. In figure 3.8, all the inertial systems are illustrated.

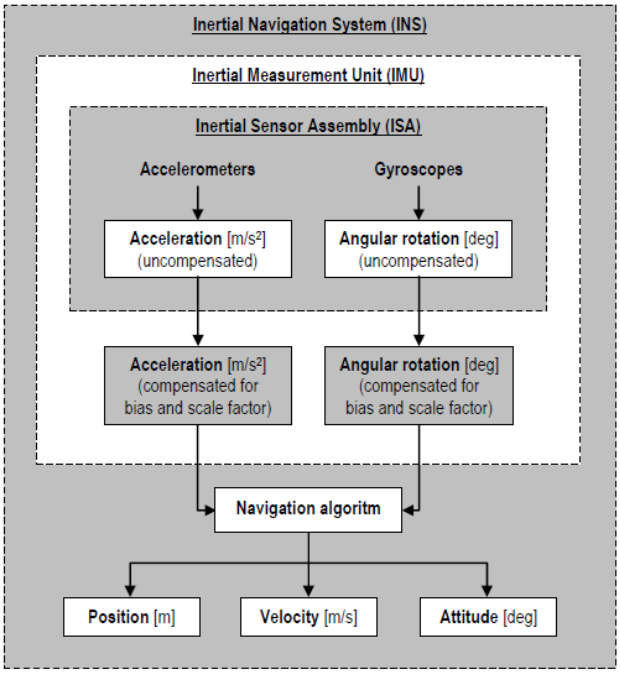


Figure 3.8: Illustration of inertial systems [13].

3.2 Navigation Equations for Land Vehicles

In previous section, a brief introduction to inertial sensors and inertial navigation system was given. INS is a real time algorithm for calculating the position, velocity and orientation of a vehicle by using inertial sensors. Since it's a dead-reckoning system, the navigation solution (position, velocity and orientation) should be initialized.

This section introduces the navigation equations that are performed in INS algorithm by navigation processor. This processor may be installed with an IMU as a complete INS or could be any processor apart from IMU. Usually, high grade inertial sensors are sold as a part of a complete navigation system. The form of the navigation

equations depends on the frame where the navigation solution is expressed i.e. the solution could be expressed in navigation frame or ECEF frame and etc. In figure 3.9, a general INS algorithm is given. At first step, initialization process should be done to determine the initial navigation solution. Then orientation update is done using the gyro measurements and the transformation matrix is formed followed by velocity and position update.

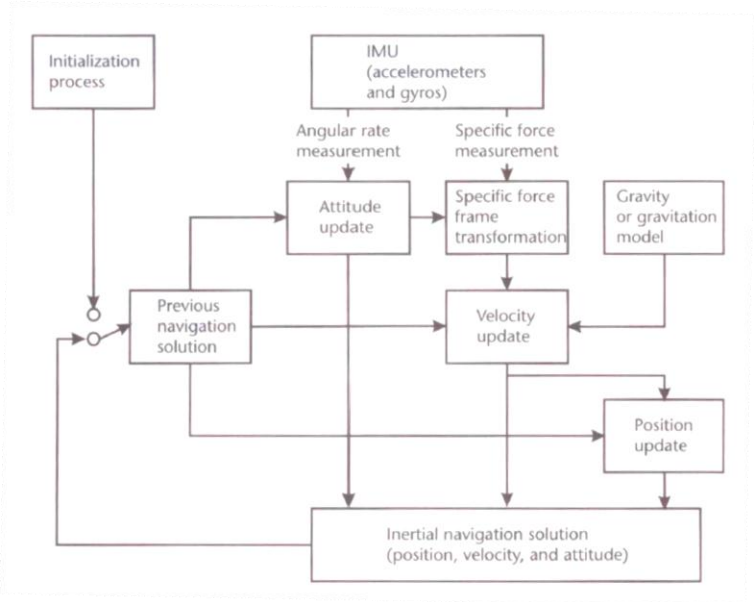


Figure 3.9: General steps of INS algorithm [2].

In this work, the terrestrial inertial navigation equations for navigation frame are described for a strapdown navigation system in section 3.2.1. Also the numerical integration of these equations, as known as mechanization, is discussed in section 3.2.2

3.2.1 Inertial navigation equations in navigation frame

In this section, the equation of motion of a vehicle navigating in local navigation frame is described. As in previous section, a navigation solution is defined to be the position, velocity and the orientation of a vehicle. These variables are time related so they change with time and they can be named as navigation states. The time derivative of navigation states of a vehicle in navigation frame can be denoted as in (3.12).

$$\dot{x}^n = \begin{bmatrix} \dot{r}^n \\ \dot{v}^n \\ \dot{R}_b^n \end{bmatrix} \quad \textbf{(3.12)}$$

In navigation frame, the origin of the frame is the projection of the vehicle's body frame origin onto the Earth reference ellipsoid and the origin of the navigation frame moves as the vehicle moves. So if the horizontal components of vehicle's position are considered, the horizontal components of vehicle's position are zero for example $r_{nb}^n = [0 \quad 0 \quad h]^T$since the projection of vehicle's body frame origin is coincident with navigation frame origin [1]. So in order to cope with this issue, Earth relative position is used to define the position of navigation frame origin which is the projection of vehicle body frame origin onto the reference ellipsoid. The Earth relative position of an object in navigation frame is denoted as r^n and it is expressed in curvilinear form as given in (3.13) where φ denotes the latitude of the vehicles, λ denotes the longitude of the vehicle and h denotes the height of the vehicle relative to Earth's ellipsoid.

Since the derivative of the position of navigation frame origin does not express vehicle's velocity in navigation frame i.e. $\dot{r}_{nb}^b = [0 \quad 0 \quad \dot{h}]^T$, the Earth-relative velocity vector represented in navigation frame is used to express the velocity of the vehicle which is denoted as $v_{eb}^n \triangleq v^n$. It should be considered that the velocity vector in navigation frame is not the derivative of the position vector in navigation frame i.e. $\dot{r}^n \neq v^n$ [1]. The velocity of the vehicle in navigation frame is given in (3.14), it is Earth-relative and resolved in navigation frame axes. The transformation matrix was introduced in section 2.1.

$$r^n = [\varphi \quad \lambda \quad h]^T \quad \textbf{(3.13)}$$

$$v_{eb}^n \triangleq v^n = [v_n \quad v_e \quad v_d]^T \quad \textbf{(3.14)}$$

The time-derivative of curvilinear position given in (3.15) is a function of the Earth-relative velocity which is resolved in navigation frame axes.

$$\dot{r}^n = \begin{bmatrix} \dot{\varphi} \\ \dot{\lambda} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} \frac{v_n}{R_M + h} & \frac{v_e}{\cos\varphi\,(R_N + h)} & -v_d \end{bmatrix}^T$$

$$= \begin{bmatrix} \frac{1}{R_M + h} & 0 & 0 \\ 0 & \frac{1}{\cos\varphi\,(R_N + h)} & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} v_n \\ v_e \\ v_d \end{bmatrix} \quad \textbf{(3.15)}$$

R_M is referred as meridian radius and R_N is referred as normal radius. They are given in (3.16) and function of the latitude φ , linear eccentricity e and equatorial radius a.

$$R_M = \frac{a(1-e^2)}{(1-e^2(\sin\varphi)^2)^{\frac{3}{2}}} \qquad R_N = \frac{a}{(1-e^2(\sin\varphi)^2)^{1/2}} \quad \textbf{(3.16)}$$

The time-derivative of the vehicle's velocity with respect to Earth resolved in navigation frame axes can be found by using Coriolis law which was represented in section 2.4.4.2. As it was stated in (3.14), the velocity of an object in navigation frame is the Earth-relative velocity resolved in navigation frame axes hence it could be written as $\mathrm{v}_{\mathrm{eb}}^{\mathrm{n}} \triangleq \mathrm{v}^{\mathrm{n}} = \mathrm{R}_{\mathrm{e}}^{\mathrm{n}}\mathrm{v}_{\mathrm{eb}}^{\mathrm{e}}$. So applying the Coriolis Law yields the result stated in (3.17).

$$\dot{\mathrm{v}}^{\mathrm{n}} = \dot{\mathrm{v}}_{\mathrm{eb}}^{\mathrm{n}} = \mathrm{R}_{\mathrm{e}}^{\mathrm{n}}(\Omega_{\mathrm{ne}}^{\mathrm{e}}\mathrm{v}_{\mathrm{eb}}^{\mathrm{e}} + \dot{\mathrm{v}}_{\mathrm{eb}}^{\mathrm{e}}) \quad \textbf{(3.17)}$$

The reader should refer to [1, 2] for more details about derivation of $\dot{v}^n$. The simplified form of $\dot{v}^n$ is given in (3.18).

$$\dot{\mathrm{v}}^{\mathrm{n}} = \mathrm{R}_{\mathrm{b}}^{\mathrm{n}}\mathrm{f}_{\mathrm{ib}}^{\mathrm{b}} + \mathrm{g}^{\mathrm{n}} - (\Omega_{\mathrm{en}}^{\mathrm{n}} + 2\Omega_{\mathrm{ie}}^{\mathrm{n}})\mathrm{v}^{\mathrm{n}} \quad \textbf{(3.18)}$$

This equation has three terms. In the first term, the accelerometer measurements, which are the specific force of the body frame with respect to inertial frame resolved in body frame axes, are transformed to navigation frame by the transformation matrix R_b^n. Since it is a strapdown implementation, a transformation matrix should be formed to align with the navigation frame using gyro measurements. The second term is the estimate of local gravity vector. As it was described in previous sections, the accelerometers only sense specific force not gravitational forces. So, to form the

kinematical acceleration of an object in an inertial frame (non-rotating and non-accelerating), the gravity vector should be added to specific force as given in (3.19).

$$\ddot{r}^{i} = f_{ib}^{i} + g^{i} \quad \textbf{(3.19)}$$

For a rotating frame like ECEF frame, the centripetal force should also be considered as well as gravity. For example, consider an accelerometer which is at rest on the surface of Earth. The Earth's gravitational field has an effect on the accelerometer as well as the Earth's rotation rate which causes the accelerometer to rotate around Earth at rate $\omega_{ie} = 7.292115\text{x}10^{-5}$ rad/sec.

The resulting accelerometer output is given in (3.20) where the skew-symmetrical representation of vector w_{ie}^{e} is Ω_{ie}^{e} and $G(r_{eb}^{e})$ is the gravitational acceleration at point r_{eb}^{e}.

$$f_{ib}^{e} = \Omega_{ie}^{e}\Omega_{ie}^{e}r_{eb}^{e} - G(r_{eb}^{e}) \quad \textbf{(3.20)}$$

The specific force given in (3.20) is the measurement of an accelerometer which is stationary with respect to ECEF frame. This specific force is the reaction to the acceleration due to gravity therefore the acceleration of gravity is the negative of sensed specific force which is defined in (3.21).

$$g^{e} = -f_{ib}^{e} = G(r_{eb}^{e}) - \Omega_{ie}^{e}\Omega_{ie}^{e}r_{eb}^{e} \quad \textbf{(3.21)}$$

The local gravity vector is the acceleration of gravity in navigation frame which is denoted as in (3.22) [2].

$$g^{n} = G(r_{eb}^{n}) - \Omega_{ie}^{n}\Omega_{ie}^{n}r_{eb}^{n} = G(r_{eb}^{n}) + \begin{bmatrix} \sin^2\varphi & 0 & \cos\lambda\sin\lambda \\ 0 & 1 & 0 \\ \cos\lambda\sin\lambda & 0 & \cos^2\lambda \end{bmatrix} r_{eb}^{n} \quad \textbf{(3.22)}$$

As defined above, the acceleration of gravity consists of two forces acting on an object which are the gravitational force and the centrifugal acceleration which is caused by Earth's rotation. This is illustrated in figure 3.10. There are several equations which provide models for acceleration of gravity. The Somigliana model is the one of simple acceleration of gravity model that is accurate on the surface of the

WGS84 ellipsoid and this model is as a function of latitude. It is presented in (3.23-24).

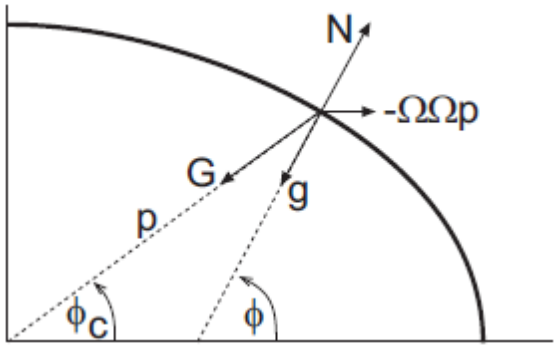


Figure 3.10: The local gravity vector [1].

$$g^n = \begin{bmatrix} 0 \\ 0 \\ \gamma(\varphi) \end{bmatrix} + \epsilon^n \quad \textbf{(3.23)}$$

$$\gamma(\varphi) = \gamma_e \frac{(1 + 0.002931851353\ \sin^2\varphi)}{(\sqrt{1 - 0.0066943800229 \sin^2\varphi})} \quad \textbf{(3.24)}$$

The equatorial effective gravity which is denoted by γ_e is equal to $9.7803267715\ m/s^2$ and ϵ^n represents local perturbations in the gravity vector relative to the ellipsoidal normal vector.

The third term , $-(\Omega^n_{en} + 2\Omega^n_{ie})v^n$ corresponds to the Coriolis acceleration which is the force formed due to the rotation of navigation frame with respect to the ECEF frame and the rotation of ECEF frame with respect to the inertial frame. This term consists of skew-symmetric representations of angular rate vectors ω^n_{en} and ω^n_{in} and could be written as below in (3.25).

$$(\Omega^n_{en} + 2\Omega^n_{ie})v^n = (\omega^n_{en} + 2\omega^n_{ie}) \times v^n \quad \textbf{(3.25)}$$

The vector ω^n_{en} is the transport rate which rotates the navigation frame axes as the vehicle travels. The transport rate is defined in (3.26).

$$\omega_{en}^{n} = \begin{bmatrix} \dot{\lambda}\cos(\varphi) \\ -\dot{\varphi} \\ -\dot{\lambda}\sin(\varphi) \end{bmatrix} = \begin{bmatrix} \frac{v_e}{R_N + h} & \frac{-v_n}{R_M + h} & \frac{-v_e \tan(\varphi)}{R_N + h} \end{bmatrix}^T \quad \textbf{(3.26)}$$

The vector ω_{ie}^{n} corresponds to the rotation rate of Earth with respect to inertial frame represented in navigation frame. The rotation rate of Earth is $\omega_{ie} = 7.292115x10^{-5}\, rad/sec$ along the spinning axis. Therefore the vector form is given as $\omega_{ie}^{e} = [0 \quad 0 \quad \omega_{ie}]^T$.

By using the rotation matrix R_e^n, the vector ω_{ie}^{n} could be easily calculated as in (3.27).

$$\omega_{ie}^{n} = R_e^n \omega_{ie}^{e} = \begin{bmatrix} -\sin\varphi\cos\lambda & -\sin\varphi\sin\lambda & \cos\varphi \\ -\sin\lambda & \cos\lambda & 0 \\ -\cos\varphi\cos\lambda & -\cos\varphi\sin\lambda & -\sin\varphi \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \omega_{ie} \end{bmatrix} \quad \textbf{(3.27)}$$

$$= \begin{bmatrix} \omega_{ie}\cos\varphi \\ 0 \\ -\omega_{ie}\sin\varphi \end{bmatrix}$$

The time derivative of direction cosine matrix was introduced in section 2.4. Since the body frame and the navigation frame experiences relative angular rotation, the derivative of direction cosine matrix is defined as given in (3.28-29).

$$\dot{\mathrm{R}}_{\mathrm{b}}^{\mathrm{n}} = \mathrm{R}_{\mathrm{b}}^{\mathrm{n}}\Omega_{\mathrm{nb}}^{\mathrm{b}} = \mathrm{R}_{\mathrm{b}}^{\mathrm{n}}\left(\Omega_{\mathrm{ib}}^{\mathrm{b}} - \Omega_{\mathrm{in}}^{\mathrm{b}}\right) \quad \textbf{(3.28)}$$

$$\dot{\mathrm{R}}_{\mathrm{b}}^{\mathrm{n}} = \mathrm{R}_{\mathrm{b}}^{\mathrm{n}}\Omega_{\mathrm{ib}}^{\mathrm{b}} - (\Omega_{\mathrm{ie}}^{\mathrm{n}} + \Omega_{\mathrm{en}}^{\mathrm{n}})\mathrm{R}_{\mathrm{b}}^{\mathrm{n}} \quad \textbf{(3.29)}$$

The first term of (3.29) is the measurement of gyros which is the skew-symmetric representation of angular rate vector of body frame i.e. $\Omega_{ib}^{b} = \left[\omega_{ib}^{b} x\right]$ with respect to inertial frame resolved in body frame.

The second term of (3.29) is the skew-symmetric representation of the rotation vector of the Earth with respect to inertial frame resolved in navigation frame axes i.e. $\Omega_{ie}^{n} = [\omega_{ie}^{n} x]$ which is given in (3.27).

The third term of (3.29) is the skew-symmetric representation of the transport rate vector which is represented in (3.26), i.e. $\Omega_{en}^{n} = [\omega_{en}^{n} x]$.

To sum up, the equations given in (3.15, 3.18 and 3.19) correspond to the motion of an object in navigation frame. The outputs of IMU are used in these equations to determine the position, velocity and the orientation of the vehicle represented in navigation frame. This INS process is illustrated in figure 3.11.

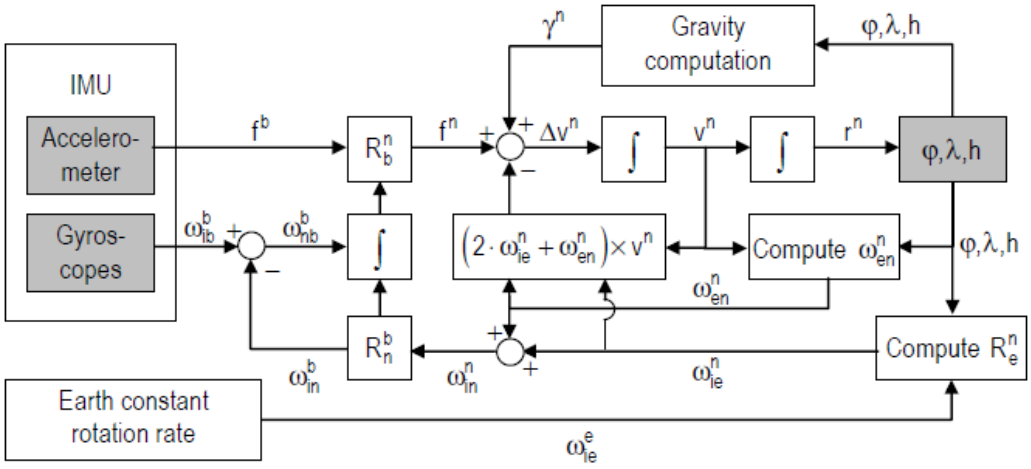


Figure 3.11: The INS algorithm [13].

3.2.2 Numerical integration for equations of motion

In equations (3.15, 3.18-19), the equations of motion of an object in navigation frame are given. Those differential equations should be numerically integrated in order to use in INS. This section introduces integration methods for differential equations.

Since INS is a dead-reckoning system, it depends on the previous values of position, velocity and orientation. In this section, subscripts k and k+1 denote values from previous step and after update respectively at each time interval.

If the measurements of IMU is assumed to be constant between each sampling period, then delta-v and delta-θ are denoted as in (3.30) and (3.31).

$$\Delta v = \int_t^{t+\tau} f_{ib}^b(\tau)d\tau = f_{ib}^b(t)\Delta t \qquad \textbf{(3.30)}$$

$$\Delta\theta = \int_t^{t+\tau} \omega_{ib}^b(\tau)d\tau = \omega_{ib}^b(t)\Delta t \qquad \textbf{(3.31)}$$

3.2.2.1 DCM update

After initialization for INS is done and the initial values of position, velocity and orientation are determined, the Direction Cosine Matrix is updated first. The time derivative of DCM is given in equation (3.29). This equation depends on previous position and velocity as well as the angular rate measurements of IMU. In [1], a closed form solution is derived for updating the DCM and the orthogonality of DCM is protected. The derived equation is given in (3.32). For detailed analysis, the reader should refer to [1].

$$R^{n}_{b,k+1} = \left(I - \frac{\sin(||v||)}{||v||}\Upsilon + \frac{1-\cos(||v||)}{||v||^{2}}\Upsilon^{2}\right)R^{n}_{b,k} \tag{3.32}$$

Where, $\Upsilon = [v(t_k)x]$ and $v_i(t_k) = \int_{t_{k-1}}^{t_k} \omega^{b}_{nb,i}(\tau)\, d\tau$, i=1, 2, 3.

3.2.2.2 Specific force transformation

After the DCM is updated, the specific force which is measured from IMU should be transformed to navigation frame by the following equation (3.33) assuming that f^{b}_{ib} is constant between each sampling period [2].

$$f^{n}_{ib}(t) = R^{n}_{b}(t) f^{b}_{ib}(t) \approx \frac{1}{2}\left(R^{n}_{b,k} + R^{n}_{b,k+1}\right) f^{b}_{ib} \tag{3.33}$$

3.2.2.3 Velocity update

The differential equation in (3.18) shows the rate of change of velocity in navigation frame. This differential equation could be integrated using Euler integration method as below in (3.34).

$$v^{n}_{k+1} = v^{n}_{k} + \left[f^{n}_{ib} + g^{n}_{k} - (\Omega^{n}_{en,k} + 2\Omega^{n}_{ie,k})v^{n}_{k}\right]\Delta t \tag{3.34}$$

3.2.2.4 Position update

The differential equation in (3.15) shows the rate of change of curvilinear position of an object in navigation frame. After updating velocity since the previous and the updated velocity is available i.e. v^{n}_{k} *and* v^{n}_{k+1} respectively, second order Runge-Kutta method could be used for update as in (3.35).

$$r_{k+1}^{n} = r_{k}^{n} + 0.5\begin{bmatrix} \frac{1}{R_M + h} & 0 & 0 \\ 0 & \frac{1}{(R_N + h)\cos\varphi} & 0 \\ 0 & 0 & -1 \end{bmatrix}(v_{k}^{n} + v_{k+1}^{n})\Delta t \quad \textbf{(3.35)}$$

3.3 Alignment and initialization

Initialization of INS is very important since the previous values of navigation states should be known to update these values with the outputs of IMU. In the previous section, equation of motion of an object in navigation frame is described. To use the navigation algorithm, the initial values of navigation states which are given in (3.36-38) should be determined through some processes called initialization and alignment. Euler angles, which are roll, pitch and yaw angle respectively, are represented by $\Theta(0)$.

$$r^{n}(0) = [\varphi(0) \quad \lambda(0) \quad h(0)]^{T} \quad \textbf{(3.36)}$$

$$v^{n}(0) = [v_n(0) \quad v_e(0) \quad v_d(0)]^{T} \quad \textbf{(3.37)}$$

$$\Theta(0) = [\phi(0) \quad \theta(0) \quad \psi(0)]^{T} \quad \textbf{(3.38)}$$

Initialization of position vector and velocity could be found by using GPS measurements. In this work, before the navigation equations are executed, an initialization algorithm is executed in order to find the initial values.

Initialization algorithm is executed when the vehicle is at rest and it is assumed that there will be no GPS outages. For example, 100 measurements of GPS are stored in memory; in 10 seconds and their arithmetic mean determine the initial values of position and velocity. The initialization of position is given in (3.39) where n is the number of measurements. The Earth-related velocity resolved in navigation frame is related to the rate of change of curvilinear position as in (3.40). If it is transformed to discrete form, then the equation (3.41) could be used in arithmetic mean given for $i \neq 1$, in (3.42).

$$r^n(0) = \frac{1}{n}\sum_{i=0}^{n} r_i^n \tag{3.39}$$

$$v^n = \begin{bmatrix} v_n \\ v_e \\ v_d \end{bmatrix} = \begin{bmatrix} (R_m + h) & 0 & 0 \\ 0 & (R_n + h)\cos\varphi & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \dot{\varphi} \\ \dot{\lambda} \\ \dot{h} \end{bmatrix} \tag{3.40}$$

$$v_k^n = \begin{bmatrix} v_{n,k} \\ v_{e,k} \\ v_{d,k} \end{bmatrix} = \begin{bmatrix} (R_{m,k} + h_k) & 0 & 0 \\ 0 & (R_{n,k} + h_k)\cos\varphi_k & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \frac{\varphi_k - \varphi_{k-1}}{\Delta t} \\ \frac{\lambda_k - \lambda_{k-1}}{\Delta t} \\ \frac{h_k - h_{k-1}}{\Delta t} \end{bmatrix} \tag{3.41}$$

$$v^n(0) = \frac{1}{n}\sum_{i=0}^{n} v_i^n \tag{3.42}$$

The determination of initial transformation matrix from body frame to navigation frame consists of determination of initial Euler angles. This process is called alignment and this alignment process could be done when the vehicle is at rest by using inertial sensors and could be done in motion using velocity measurements from GPS. Both these methods are described shortly in this section.

Stationary alignment has two steps to initialize the Euler angles when the vehicle is at rest. They are the accelerometer leveling and gyro compassing. Briefly, accelerometer leveling method uses accelerometer measurements with knowledge of local gravity vector, to determine the roll and pitch angles. Leveling corresponds to the alignment of navigation frame z-axis (down axis) and accelerometer triad's z-axis. Since if the vehicle is not leveled, the projection of gravity vector will appear in the axes of body frame as accelerations. In figure 3.12, leveled and misleveled axes of accelerometers are shown. If there is a misalignment between z-axis accelerometer and the down axis of navigation frame then this results a misleveling of X-Y plane where the roll angle represents misleveling of x-axis and pitch angle represents misleveling of y-axis. Using this measurements, the roll and pitch angles could be determined as in (3.43-44) [13].

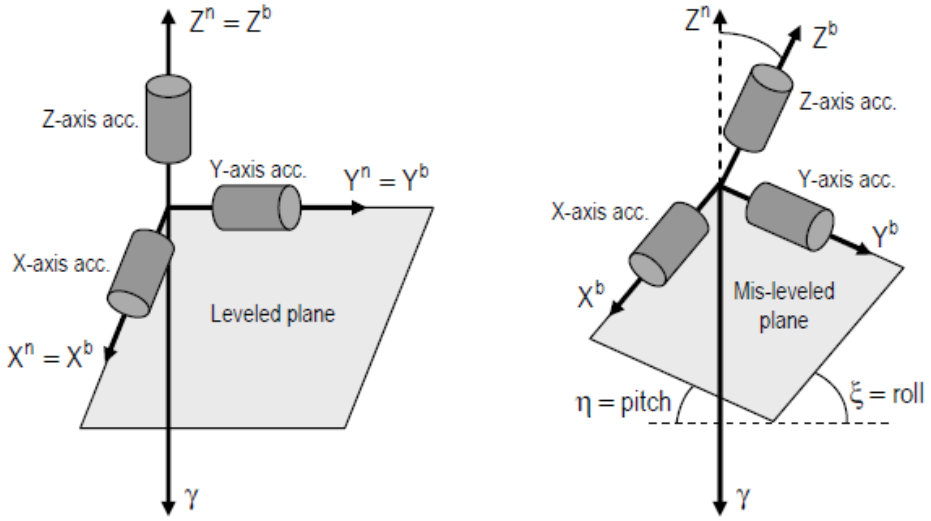


Figure 3.12 : The leveled and mis-leveled axes of IMU [13].

$$\theta = -\sin^{-1}\left(f^{b}_{ib,x}/g\right) \tag{3.43}$$

$$\phi = \sin^{-1}\left(f^{b}_{ib,y}/g\cos\theta\right) \tag{3.44}$$

One method for determination of heading is using gyro-compassing method where the gyro that is used should observe the Earth rotation rate. So for low-cost gyro sensors, gyro-compassing is not possible since the bias and noise exceeds the Earth rotation rate but on the other hand, for higher grade sensors heading is calculated using gyro-compassing method given in (3.45-46).

$$\omega^{h}_{ib} = [-\theta][-\phi]\omega^{b}_{ib} \tag{3.45}$$

$$\psi = -\tan^{-1}\left(\frac{\omega^{h}_{ib,x}}{\omega^{h}_{ib,y}}\right) \tag{3.46}$$

For low-cost sensors, the user should estimate the heading angle by using external sensors like digital compass and magnetometer. In this work, a digital compass is used when the vehicle is at rest, to determine the heading angle of the vehicle since the IMU that is used cannot observe the Earth rotation rate. That is given in (3.47).

$$\psi(0) = \frac{1}{n}\sum_{i=0}^{n}\psi_i \tag{3.47}$$

Alignment in motion could be done using GPS velocity. For most land vehicle applications, the Euler angles could be initialized and determined as given in (3.48-50). When the vehicle is at rest, the GPS measurements have higher deviation. So the vehicle speed should be more than 5m/s for accurate determinations [13].

$$\phi \approx 0 \tag{3.48}$$

$$\theta = \tan^{-1}\left(\frac{v_d^{GPS}}{\sqrt{(v_n^{GPS})^2 + (v_e^{GPS})^2}}\right) \tag{3.49}$$

$$\psi = \tan^{-1}\left(\frac{v_e^{GPS}}{v_n^{GPS}}\right) \tag{3.50}$$

3.4 Satellite Navigation Systems

In previous section, INS was introduced. This type of navigation system calculates navigation states by integrating inertial sensor measurements. Since there are errors associated with the inertial sensors as mentioned in section 3.1.3, their integration causes errors in navigation solutions that depend on time. So that makes INS system reliable in a short period of time if high accuracy is needed. The advent of satellite navigation systems is very important for high accuracy navigation since the errors associated with navigation solutions are not time dependent. In this section, a brief introduction about satellite navigation systems is given.

Satellite navigation systems provide a three dimensional position solution by using radio signals transmitted from satellites that are orbiting in space. Global Navigation Satellite Systems (GNSS) is the common term for those systems which provides three dimensional position solutions. The most popular system is Global Positioning System, which is the only system that provides global coverage, operated by the U.S government. In following section, GPS will be introduced in detail.

Besides GPS, there are several systems that are partially functioning and still developing. GLONASS is the Russian Satellite Navigation System which is the abbreviation of "**GLO**bal'naya **NA**vigatsionnaya **S**putnikovaya **S**istema" in russian and in English its long form is "**GLO**bal **NA**vigation **S**atellite **S**ystem". This project

began in the USSR in 1976 but due to collapse in Russian economy, the system fell into disrepair. At the time of writing (March 2011), the GLONASS is fully operational and has full coverage in Russia's territory and it is expected to have a global coverage during 2011 [14].

Galileo is the satellite navigation system which is currently being built and funded by European Union and European Space Agency. This project started in 2003. The main reason is to provide high accuracy position solutions to European nations independent from GLONASS and GPS since in times of war, both systems could be disabled to civilian use. At the time of writing, this project is supposed to be fully operational by 2014 [15].

COMPASS is a global navigation satellite system, which is also known as Beidou-2, is a project that is announced and still developing by China. The main aim of this project is to improve the quality of positioning by increasing the number of satellites. This system is planned to have 35 satellites that will offer complete coverage of globe [16].

In this work, the GPS is used to provide absolute position measurements. Hence GPS will be introduced in detail. The principle of positioning using satellites and the segments of GPS will also be mentioned. In another section, advanced GPS positioning techniques will be introduced.

3.4.1 Introduction to GPS

Global Positioning system is a big advent in navigation history since it provides very accurate position and time information in all weather conditions at any time if there is an unobstructed line of sight to four or more GPS satellites. Due to its global coverage and very reasonable cost of receivers, GPS is the most common and popular navigation system in the world [2].

GPS project was developed and realized by U.S government. It was started in 1973 and it became fully operational in 1994. GPS offers two kinds of navigation solutions. Standard Positioning Service is for civilian usage. Anyone who has a GPS receiver can benefit from this service. Since GPS is military based navigation system, Precise Positioning service provides high precision navigation solution to

users licensed by U.S government. This service has encrypted signals that are only available to licensed users.

GPS has three main segments, which are illustrated in figure 3.13, to operate. The first segment is the space segment. This segment is the heart of GPS and consists of GPS satellites orbiting in space given in figure 3.14. Nominally, GPS operates with a constellation of 24 satellites because it is designed to have six orbital planes in which each orbit plane, four satellites orbit Earth in nearly 12 hours. These six orbits are equally spaced around the equator so each orbit is separated by 60 degree of longitude and each inclined at 55 degrees to the equator. The satellites orbit at a radius of 26.600 km and approximately 20.100 km above the Earth Surface. By this way, this constellation guarantees that any user that is anywhere on Earth, has a direct line of sight to at least four satellites which is essential to get a 3-D position estimation and accurate time estimate.

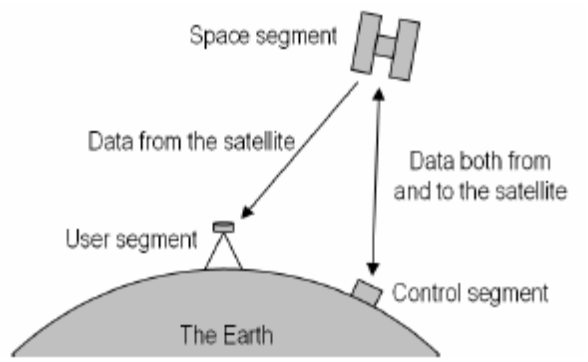


Figure 3.13 : Three segments of GPS [2].

GPS satellites in space continuously transmit the encoded data using three bands. Those data are encoded in 10 different GPS navigation signals. The carrier frequencies link-1 (L1), link-2 (L2) and link-5 (L5) are modulated by data and spread spectrum signals. These carrier frequencies are 1,575.42 MHz for L1, 1,227.60 MHz for L2 and 1,176.45 MHz for L5. More information about the signals that are transmitted by GPS satellites could be found in [2].

For navigation case, the carrier signals L1 and L2 are modulated by Code Division Multiple Access (CDMA) signals which are also Pseudo Random Codes (PRC). The PRC is a stream of values called chips which have a value of ± 1. The

coarse/acquisition code (C/A) which is modulated on L1 and L2, is 1023 samples (chips) long and repeats every 1ms (at 1.023 MHz) which is the tenth of fundamental frequency of L1 which is 1.023 MHz. This signal is named as coarse since it was designed to provide less accurate position estimate because of civilian use. The Precise (Encrypted Precise)-Code, has a notation of (P(Y)), is modulated only on L2 and transmitted at the fundamental frequency of 10.23 MHz. This notation, P(Y), is used to refer to the both P and Y codes. The P-code is encrypted by the Y-code, which is only available to licensed PPS users. This encryption is referred as anti-spoofing since it makes the code difficult for hostile forces to use spoofing. So the C/A code is used for both SPS and PPS but the P(Y) code is used only for PPS. That's the reason why PPS is more accurate than SPS.

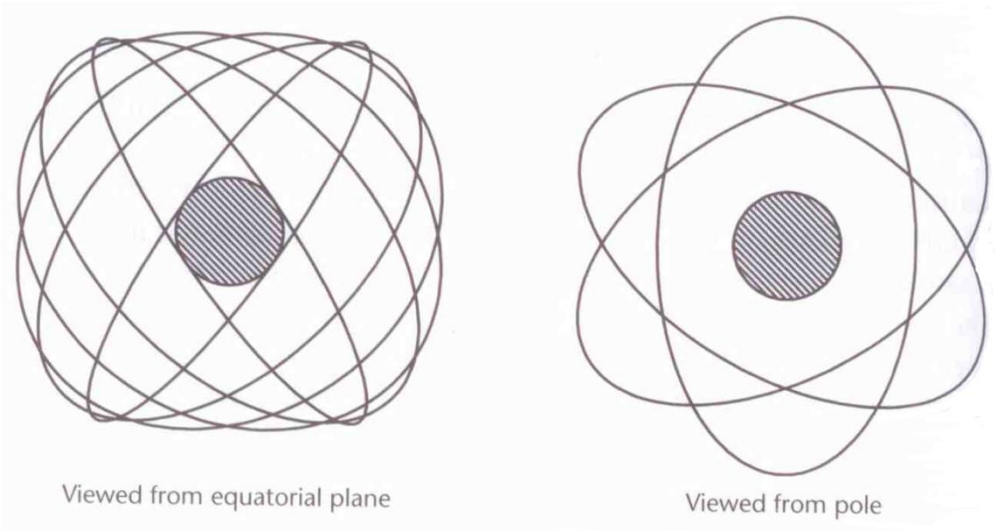


Figure 3.14: The view of orbits of GPS satellites in space [2].

The second important segment is the control segment which consists of six monitor stations located around the world, four uplink stations and a master control station near Colorado Springs, Colorado [2]. This segment is responsible for monitoring the status of the satellites in space segment whether they are healthy or not. Monitoring stations measure signals transmitted from satellites and then these signals are transmitted to master control station to determine the orbital model and clock corrections of each satellite [aided]. These parameters and corrections are transmitted to the satellites via uplink stations.

The last section is the user section. This section consists of GPS antenna, GPS receiver, ranging processor and navigation processor. The antenna converts the received GPS signals to electrical signals. The receiver is responsible for demodulation of the signals using a clock to provide a time reference. The receiver has three main stages. The input signal from the antenna is amplified, filtered and shifted by radio frequency front stage. In baseband stage, there are several channels which operate in parallel for tracking signal from each satellite. The navigation stage consists of ranging processor and navigation processor. The ranging processor performs acquisition and tracking algorithms to determine the range from antenna to each satellite. The navigation processor uses ranging measurements that is calculated by ranging processor to compute position, velocity and time estimate to user.

GPS receivers operate passively since they do not transmit any signals so by this way GPS space segment can provide service to unlimited number of users on Earth. The detailed illustration of the major segments of GPS is given in figure 3.15.

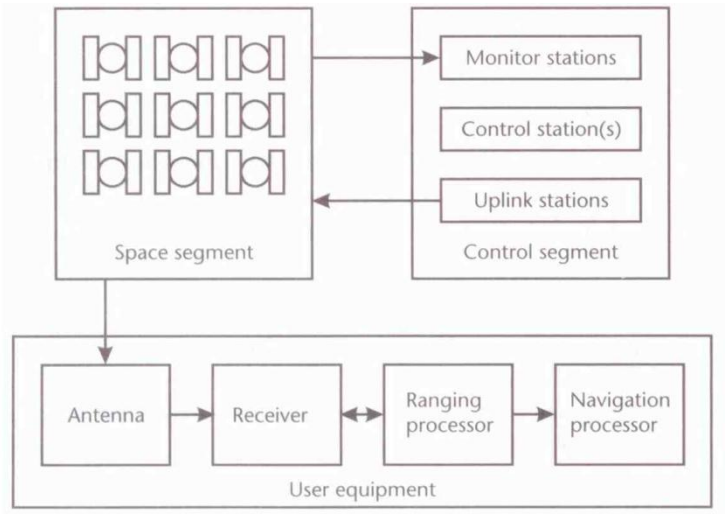


Figure 3.15: The parts in segments of GPS [2].

3.4.2 GPS positioning

The theory behind the positioning from GPS satellites actually has a fundamental concept. This concept is “distance is equal to velocity times travel time”. The electromagnetic signals that are transmitted from GPS satellites in space travels at

the speed of light. So, if the travel time could be measured accurately, the distance from satellite to antenna could be easily found which is given in (3.43). The subscript i denotes the satellite number, $t_{a,i}$ is the signal arrival time measured at the receiver and $t_{t,i}$ is the signal transmission time measured and coded in satellite processor. ρ_i is the ranging measurement from a satellite in space to user.

$$\rho_i = (t_{a,i} - t_{t,i})c \quad \textbf{(3.43)}$$

The timing is very important and it should be very accurate since a microsecond of timing error causes in a range error of 300 meters, therefore very accurate timing i.e. nanoseconds is required to achieve a range error of meter level. The satellites are equipped with very accurate atomic clocks for time measurements. But using atomic clock in GPS receiver makes the system very expensive. Instead of using atomic clocks, receivers use inexpensive clocks based on crystal oscillators. As a result, the satellite clock and the receiver clock are not synchronized during range measurements, a bias occurs because of the timing errors. Pseudo-range is the biased ranging measurement [2].

If we assume that the satellite and receiver are perfectly synchronized and the clock errors are zero, then pseudo-range is equal to actual range. If a single ranging measurement ρ_1 is considered then the user position could be anywhere on the surface of a sphere of radius ρ_1 centred on this satellite. If dual ranging measurements $\rho_1 and\ \rho_2$ are used, then the user position could be on the circle of the intersection of two spheres with the radii $\rho_1 and\ \rho_2$ centred on those two satellites. Moreover, if triple ranging measurements are considered, then the user position could be on two points which is the intersection of the circle and the sphere of radius ρ_3. In most cases, one of the solutions appears in space so that only one solution is viable. If the both solutions are viable then a fourth measurement should be used to overcome this ambiguity. So with zero receiver clock error, the position of a GPS antenna could be determined by at least three measurements. The satellites could be used as reference points in space since their position is obtained from the set of parameters known as ephemeris which is broadcast in the navigation data. Therefore, since the knowledge of satellites position and the range measurements, the

position of a user in anywhere on Earth could be calculated. These three measurements are illustrated in figure 3.16

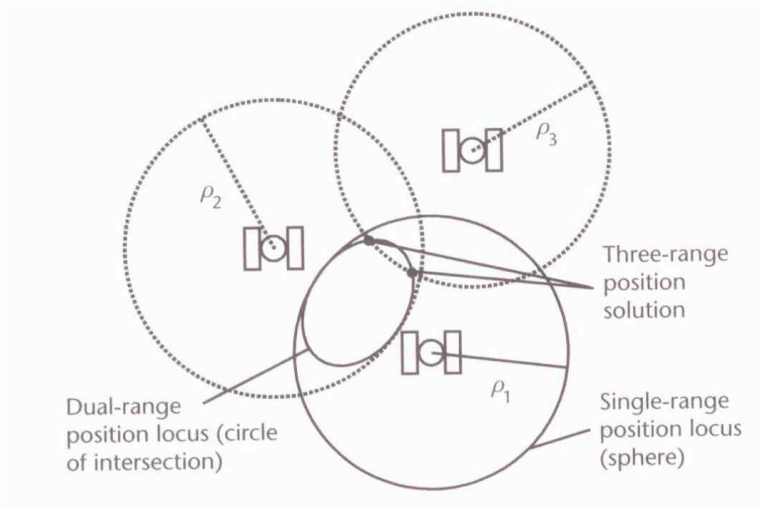


Figure 3.16 : The possible solutions resulting from three range measurements [2].

But in real time applications, the clock error is not zero, so at least four pseudo-range measurements are needed for obtaining a unique position solution. The determination of the user's position using four satellite range measurements is illustrated in figure 3.17.

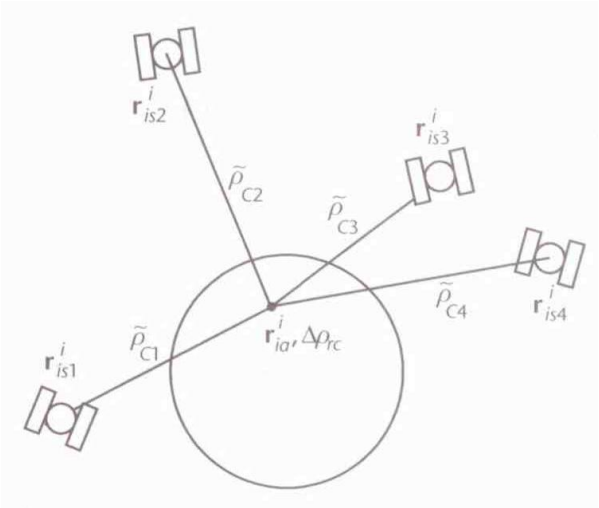


Figure 3.17: The only solution resulting from four range measurements [2].

3.4.3 GPS observations

There are three observations derived from GPS signals. The first observation is pseudo-range which is derived from PRN as stated in previous section. Since the

receiver and satellite clocks are not perfectly synchronized, there occurs a time shift needed to correlate between the received signal and receiver replicated signal. After the correlation of both signals, the travel time of code could be easily found. So pseudo-range represents the distance between the satellite and the user including the clock bias.

The second observation is the carrier phase measurement which is derived from the phase of incoming carrier frequencies before demodulation. By this method, better accuracy is obtained but the drawback of this method is the integer ambiguity. It represents the unknown number of the whole wavelengths between satellite and receiver and this ambiguity is needed to be solved. This ambiguity occurs because only a fraction of wavelength could be determined using phase measurement but the number of whole wavelengths between satellite and user remains unknown.

The third observation is the Doppler shift which could be derived from the movements of a receiver by measuring the rate of change of the carrier phase measurements. By using Doppler shift measurement, the velocity of the receiver could be calculated.

For more detailed analysis of these three variables the reader should refer to [1, 2, 12].

3.4.4 Advanced satellite navigation methods

In this section, brief introduction about advanced satellite navigation techniques is given. The aim of these methods is to improve the accuracy of navigation solutions, to improve robustness and reliability. Using these methods, some of the error sources could be significantly removed.

The most popular and common way to enhance the accuracy of GPS solution is Differential GPS method. The basic principle of DGPS method is to set up a reference (base) station where the location of this station is exactly known. Since the exact position of the station is known, the correlated range errors could be computed and transmitted to a mobile user (rover) via radio link using UHF radio modem. After the user receives the corrections from reference station, the correlated range errors could be compensated to improve the accuracy. If the user and the reference

station are relatively close to each other, then their pseudo-range measurements have same amount of error that is caused by atmospheric disturbances. So this method is effective if the rover and base station are relatively close.

The correction signals could be transmitted from a base station that is specific to the user or the user could receive correction signals from augmentation systems such as GBAS, GRAS, WAAS, LAAS and etc.

Also there are some companies that transmit correction signals via geostationary satellites. But a subscription must be bought to receive a subscription authorization.

The illustration of differential GPS method is shown in figure 3.18.

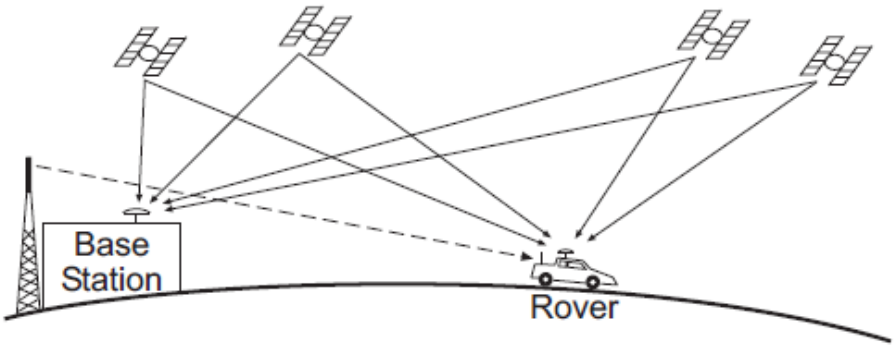


Figure 3.18 : The DGPS with its base station and rover [2].

Another method to enhance the accuracy, robustness and reliability of the navigation solution is the use of carrier phase measurements. This method is very similar to DGPS but instead of using the PRC message modulated in carrier signals, it uses the carrier signals as its signal. This method is referred as Carrier-Phase Enhancement or Real time kinematics.

In RTK system setup, there is a single base station where the phase of the carrier is measured and broadcasted via using a radio modem. Then the mobile users receive the measurement from the base station and they compare their carrier phase measurement with the received one and they correct their measurements. By using Virtual Reference Station method, RTK correction could be extended.

3.4.5 GPS error sources

This section represents the most important and effective error sources that affect the accuracy of range measurements of GPS. Some of the error sources are time-correlated and could be corrected by the receiver. But some of the errors could be caused by the environment and even if they could be caused by the position of the satellites in the space.

3.4.5.1 Selective availability

Selective Availability is the error source that was introduced by U.S government. It's a type of artificial clock error that affects satellite clocks to deteriorate the position determination for unauthorized users. This error could be reduced by DGPS method but now it is turned off. Before it was active, it was responsible of the largest error in the navigation solution for civilian users.

3.4.5.2 Clock errors and orbital errors

The clock errors are caused by the biases in the satellite clock and the receiver clock. The clocks in satellites should be synchronized with the clock in master station. Any error in the satellite clock will affect the pseudo-range measurements and the carrier phase measurements. Receiver clock bias also affects the error in measurements but when there are four satellite measurements available, it could be easily estimated. Therefore, it actually does not affect the navigation solution.

Any inaccuracy in ephemeris parameters, which are the parameters that describe the orbit during a given interval of time, causes errors in satellites position. The control segment monitors the satellite orbits. Then the ephemeris parameters are calculated and broadcast to the user by the satellite signals which allows user to estimate the position of the satellite [2].

3.4.5.3 The atmospheric errors

The atmospherics errors are caused by the delays in the section of atmosphere. The signals that are transmitted from a satellite travel a distance of 20.000- 26.000 km to reach the user on Earth. About 1000km of this distance is the atmosphere of Earth surrounded by gases. When the signals travel in atmosphere the surround medium

affects the travel speed and the path. All this time, it was assumed that the GPS signals travel at speed of light. When the signals travel through the atmosphere so time delay occurs.

The time delay occurs in two sections of atmosphere. In the troposphere section, the signals experience atmospheric temperature, pressure and water vapour, which cause delay in the arrival of the signal. In the ionosphere section, there are large amount of free electrons and positively charged molecules that causes delay of the signal.

These sections of atmosphere are modeled to estimate the possible time delays when the satellite signal is travelling through the sections. Also DGPS method reduces the atmospheric errors and improves the accuracy of the navigation solution [1].

3.4.5.4 The multipath error

Multipath errors are caused by the environment of the receiver since the satellite signal reaches the receiver in multiple reflected paths not a direct path. These multipath reflections affect the measurements in GPS receiver causing errors in the navigation solution. But most of the high grade GPS receivers have software installed to eliminate the multipath error. To decrease the effect of multipath, the user should avoid being near to the reflective surfaces.

4. STATE ESTIMATION WITH KALMAN FILTER

In chapter 3, the navigation systems were introduced. Those two systems, INS and GPS, have complementary properties and each navigation system has advantages and disadvantages over the other navigation system. INS provides position, velocity and orientation of the vehicle by integrating the measurements of IMU in high rate but it suffers from the integration process which leads unbounded error growth dependent on time. GPS provides accurate position and velocity information depending on some factors that was introduced in section 3.4 and the error within its measurements are independent of time. The disadvantages of GPS are the necessity of line of sight between the antenna and the satellite, the lost of sight causes GPS outages, and the low sampling rate. These properties are complementary and make the data fusion of both systems ideal.

The information about the vehicle may be provided from different sensors such as IMU, GPS, odometer, digital compass with different sampling frequencies and different advantages. Combining these sensor measurements makes it possible to provide more robust and accurate information overcoming the disadvantages of sensors. This method is known as data fusion and it is an important process for navigation to provide accurate and robust estimations of position, velocity and orientation of the vehicle. For this purpose, statistical estimation algorithms are used for estimation of vehicle states from noisy sensor measurements. In this work, Kalman filter is used therefore in this chapter, a brief introduction to Kalman Filtering is given. In section 4.1 basic concepts in recursive state estimation is introduced then in section 4.2, the continuous and discrete time stochastic process models are given and at the end in section 4.3 the Kalman Filtering Algorithm is briefly introduced.

The purpose of this chapter is to comprehend the idea behind the state estimation with Kalman Filtering which is one of the most useful methods for combining sensor information.

4.1 Basic Concepts

State estimation is the method for estimating the quantities from the sensor measurements which may not be directly observable. To comprehend this important subject, some basic definitions in probability should be known.

Random variable concept is very important in robotics since the states, sensor measurements and the control signals are all modeled as random variable where they could take on multiple values. If *X* is assumed to denote a random variable in a single dice tossing, then this random variable could take values of {1, 2, 3, 4, 5 and 6} with the probability of $\frac{1}{6}$ for each value that random variable X takes i.e. $p(X = 1) = \frac{1}{6}$, $p(X = 2) = \frac{1}{6}$ and etc.

The continuous random variables possess probability density function which describes the relative likelihood for the random variable to occur at given point. There are several probability density functions but in this section, the Gaussian (normal) distribution is introduced.

The Gaussian distribution is a continuous probability distribution used to describe real valued random variables that tend to cluster around a single mean value. The probability density function of a Gaussian is given in (4.1).

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{\frac{-(x-\mu)^2}{2\sigma^2}\right\} \quad \textbf{(4.1)}$$

The probability density function of several Gaussian random variables is given in figure 4.1. There are two important parameters that specify the random variable. The mean is denoted as μ and it specifies the mean value of the random variable throughout the distribution and the mean is the peak of bell shaped distribution in the figure 4.1. The variance is denoted by σ^2 and it specifies how far the distribution spread is. The variance is a measure of the width of the distribution. Standard

deviation is the square root of the variance and denoted as σ. The Gaussian distribution is denoted as $N(x;\ \mu\ ,\sigma^2)$.

In (4.1), the probability distribution of a scalar Gaussian random variable is given but the random variable x is a vector frequently. So the multivariate Gaussian distribution is given as in (4.2) where C is the covariance matrix. The covariance matrix is a positive semi definite and symmetric matrix corresponds to the variance in scalar case. The diagonal elements of this matrix are the variance values of each random variable and the non-diagonal elements are the mutual variances between the random variables. The argument in exponential is quadratic in terms of random variable vector $\bar{x}$.

$$p(\bar{x}) = \det(2\pi C)^{-1/2}\exp\left\{\frac{-1}{2}(\bar{x}-\bar{\mu})^{\mathrm{T}}\mathrm{C}^{-1}(\bar{x}-\bar{\mu})\right\} \tag{4.2}$$

In recursive state estimation, the Gaussian filters play an important role and they are the most popular technique for linear and nonlinear state estimation [23]. In this estimation technique, all the random variables are assumed to be represented by multivariate Gaussian distribution defined above. The most important property of this distribution is that Gaussians are unimodal and have a single maximum point (mean) in distribution. In next section, Kalman filter, which is a linear Gaussian filter, will be introduced.

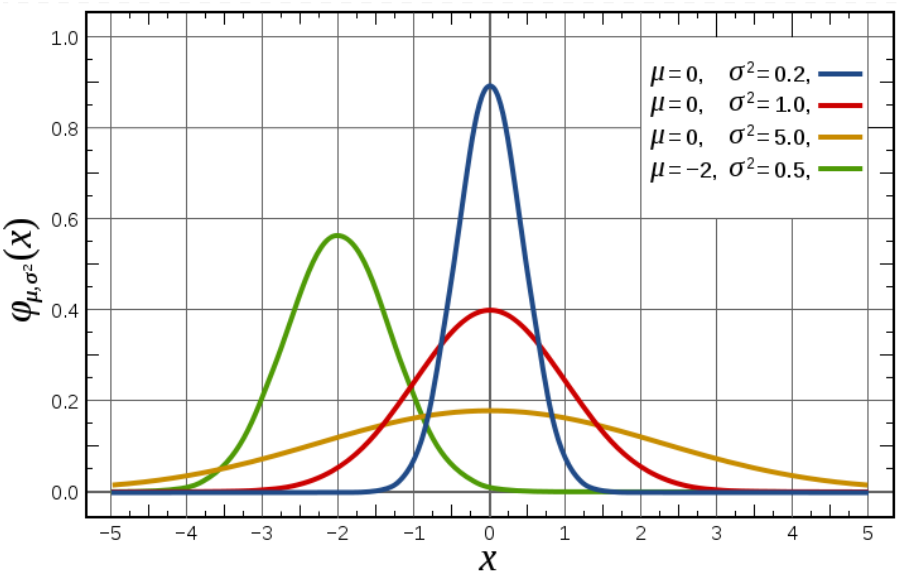


Figure 4.1 The Gaussian distributions with different parameters [23].

4.2 The Kalman Filter

The Kalman Filter is the most popular estimation technique for linear systems and it has been used successfully in many fields and particularly in autonomous navigation [21]. Also it is the basis of the majority of estimation algorithms used in navigation i.e. calibration of low-cost INS, integration of INS with GPS and other aiding sensors and also maintaining a smooth and optimal solutions in a noisy environment.

The KF is named from its inventor R.E. Kalman who published his idea for estimating and filtering technique for linear Gaussian systems in 1960 [2]. Basically, KF is a recursive Bayes Filter implementation for linear Gaussian systems where the random variables are represented as Gaussians [23]. This estimation algorithm uses the deterministic and statistical properties of the system parameters i.e. states and the measurements, which are subject to noise, to obtain optimal estimates from the given information available. The measurements should be functions of states of the system. Since this is a recursive algorithm, it carries information from iteration to iteration so it also provides a set of uncertainties in its estimates and the correlations between the errors in the estimates by specific matrices.

The KF has two basic steps as Bayesian Filter. Before the recursive algorithm starts, it should be initialized. After initialization, the algorithm could go on forever. In the first step, KF uses a dynamic model of the system for prediction of the system states between times of measurements and in the second step when the measurements arrives, KF uses the measurements that are related to states of the system to optimally correct the prediction in the first step. This algorithm is recursive, computationally efficient and optimal in the sense of minimum mean of squared errors [21].

4.2.1 Linear continuous time system model

The physical systems can be described mathematically by using a set of ordinary differential equations. These equations also can be linear or non-linear. In this section, the system is assumed to be finite dimensional linear continuous time and driven by a stochastic random variable. These models are important because it is necessary to know how the states vary with time. The dynamic model, which is also

named as system model, process model or the time propagation model, and the measurement model of a system could be given as in (4.3) and (4.4).

$$\dot{x}(t) = F(t)x(t) + G(t)w(t) \quad \textbf{(4.3)}$$

$$y(t) = H(t)x(t) + v(t) \quad \textbf{(4.4)}$$

In the equations above, some of KF elements are given. The state vector $x(t)$ is the set of parameters of the system which describes a property of the system. In a specific application like navigation, these states could be position, velocity and orientation of the vehicle or the biases of the sensors. These states are to be estimated by KF. The first time derivative of the states provides the rate of change of the states which is denoted as $\dot{x}(t)$. For continuous linear systems, if it is assumed that the number of the states is n then the system matrix $F(t)$ is a nxn square matrix which could be constant or time varying. The measurement vector $y(t)$ describes the outputs of the system where the measurement matrix $H(t)$ describes how the measurements depend on the states of the system. This matrix should be linear for linear systems. If the number of measurements is assumed to be m, then $H(t)$ is a mxn matrix. The stochastic variables $w(t)$ and $v(t)$ are the system noise and the measurement noise respectively. These random processes are accounting for the system uncertainties and the noise in the measurements. In KF approach, these random processes are assumed to white noise and have a Gaussian distribution with zero mean.

The white noise process is a random process where the samples taken at random time intervals of this random process are uncorrelated as in (4.5). For a scalar white noise process $w_i(t)$, the variance integrated over the time interval τ is given in (4.6). The power spectral density of $w_i(t)$ is the variance per unit bandwidth and denoted as η^2. The PSD is a function frequency generally but in continuous white noise process, it is constant [2].

$$E[\omega_i(t_1)\omega_i(t_2)] = 0\,, t_1 \neq t_2 \quad \textbf{(4.5)}$$

$$\sigma_\omega^2 = \eta^2\tau \quad \textbf{(4.6)}$$

For the given system model in (4.3) and (4.4), the statistical properties of white Gaussian processes are given below with the PSD of $w(t)$ and the covariance matrix of $v(t)$ in (4.7) and (4.8) respectively. The expectation operator is denoted as $E[..]$.

$$\mathrm{E}[\mathrm{w}(\mathrm{t})] = 0 \quad \text{and} \quad \mathrm{E}[\mathrm{w}(\mathrm{t})\mathrm{w}^{\mathrm{T}}(\tau)] = \mathrm{cov}\big(\mathrm{w}(\mathrm{t}), \mathrm{w}(\tau)\big) = \mathrm{Q}_{\mathrm{w}}(\mathrm{t})\delta(\mathrm{t}-\tau) \tag{4.7}$$

$$\mathrm{E}[\mathrm{v}(\mathrm{t})] = 0 \quad \text{and} \quad \mathrm{E}[\mathrm{v}(\mathrm{t})\mathrm{v}^{\mathrm{T}}(\tau)] = \mathrm{cov}(\mathrm{v}(\mathrm{t}), \mathrm{v}(\tau)) = \mathrm{R}(\mathrm{t})\delta(\mathrm{t}-\tau) \tag{4.8}$$

Also the system noise and the measurement noise are independent of the current and previous states given in (4.9) and (4.10) and as well as independent from each other as in (4.11).

$$\mathrm{cov}\big(\omega(\mathrm{t}), \mathrm{x}(\tau)\big) = 0 \quad \text{for } \mathrm{t} \geq \tau \tag{4.9}$$

$$\mathrm{cov}\big(\mathrm{v}(\mathrm{t}), \mathrm{x}(\tau)\big) = 0 \quad \text{for } t \geq \tau \tag{4.10}$$

$$\mathrm{cov}\big(\mathrm{v}(\mathrm{t}), \omega(\tau)\big) = 0 \quad \text{for all } t, \tau \geq 0 \tag{4.11}$$

The statistical properties of Gaussians are usually estimated by the designer in advance and their correctness plays an important role for optimality. If one of these random processes is not white, then it may be possible to append the linear dynamics to the process model to utilize the process model driven by a white noise.

4.2.2 Discreatization of linear continuous time system model

In most applications, the measurements occur at discrete time intervals with respect to the sampling frequency of the related measurement sensors. In this case, implementing a discrete KF is appropriate where the discrete KF needs a discrete system model to propagate the state estimates between measurement times and to relate the measurements with the states at the measurement time. The linear discrete form of the system model given in (4.3) and (4.4) is introduced in (4.12) and (4.13).

$$\mathrm{x}_{\mathrm{k}+1} = \Phi_{\mathrm{k}}\mathrm{x}_{\mathrm{k}} + \mathrm{w}_{\mathrm{k}} \tag{4.12}$$

$$\mathrm{y}_{\mathrm{k}} = \mathrm{H}_{\mathrm{k}}\mathrm{x}_{\mathrm{k}} + \mathrm{v}_{\mathrm{k}} \tag{4.13}$$

The discrete time intervals are denoted as k and the time is propagated at constant rate such as $t_{k+1} = t_k + \Delta t$.

The state transition matrix at time interval k, which is denoted as $\Phi_k = \Phi(t_{k+1}, t_k)$, is used for propagating the state from time interval t_k to t_{k+1}. This matrix can be calculated by the inverse Laplace transformation in (4.14) or a more practical way is to use truncated power series as in (4.15). For most cases, the higher order terms of (4.15) is omitted for ease of calculation.

$$\Phi_k = L^{-1}\{[sI - F]^{-1}\} \quad \textbf{(4.14)}$$

$$\Phi_k = e^{F\Delta t} = I + F\Delta t + \frac{F^2\Delta t^2}{2!} + \cdots \cong I + F\Delta t \quad \textbf{(4.15)}$$

The discrete form of measurement matrix in (4.4) is H_k and they are equal i.e. $H_k = H(t)$. The discrete process noise is denoted as w_k and it is defined as in (4.16).

$$w_k = \int_{t_k}^{t_{k+1}} \Phi(t_{k+1}, \tau)\, G(\tau) w(\tau) d\tau \quad \textbf{(4.16)}$$

The discrete process noise and the measurement noise are assumed to be Gaussian with zero-mean and with the covariance properties as given below in (4.17-19).

$E[w_k, w_j{}^T] = Q_k$, for k=j, otherwise it is equal to 0 **(4.17)**

$E[v_k, v_j{}^T] = R_k$, for k=j, otherwise it is equal to 0 **(4.18)**

$E[w_k, v_j{}^T] = 0$, for all values of k and j **(4.19)**

The computation of the covariance matrix Q_k, which is associated to w_k, is given in (4.20). It is calculated by using the first order approximation of the transition matrix [7]. The value of measurement noise covariance matrix is equal in each case.

$$Q_k \approx \Phi_k G Q_w G^T \Phi_k{}^T \Delta t \quad \textbf{(4.20)}$$

4.2.3 The discrete Kalman filter algorithm

In this section, the algorithm of DKF is given. For the derivations of this algorithm, the reader should refer to [24], and [23]. As it was stated before, DKF algorithm has two stages: Prediction and update. Since DKF is a recursive algorithm, initialization should be done. The algorithm steps are illustrated in figure 4.2.

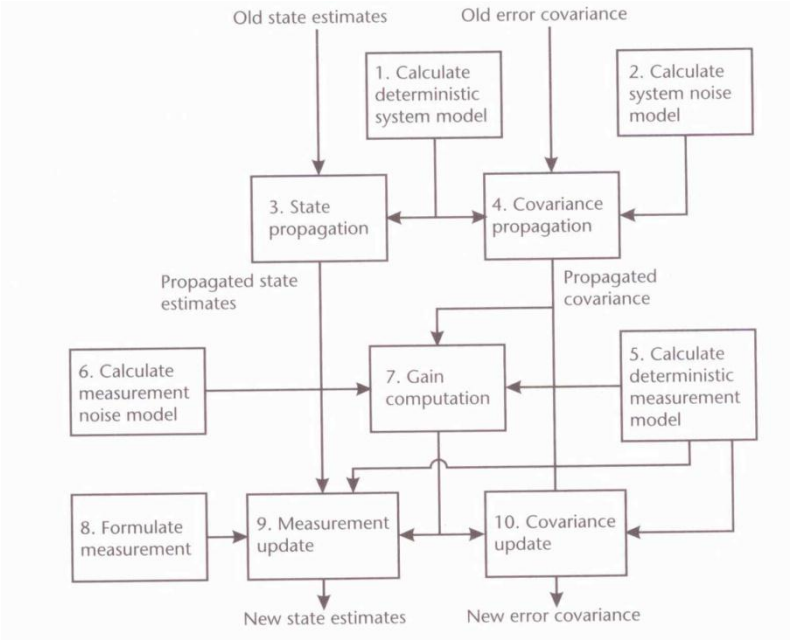


Figure 4.2 The KF algorithm [2].

In initialization stage, the initial values of the estimates of the states, the error covariance matrix and discrete time should be given. These values should be given as in (4.21), (4.22) and (4.23).

$$\hat{x}_{0|0} = E[x_0] \tag{4.21}$$

$$P_{0|0} = \mathrm{cov}(x_0) \tag{4.22}$$

$$k = 0 \tag{4.23}$$

The equations for the prediction stage are given in (4.24), (4.25) and (4.26). In this stage, the aim is to predict forward the states of the system and the error covariance matrix using system model from the time of the validity of the last measurement to the time of current measurement.

$$\hat{x}_{k+1|k} = \Phi_k \hat{x}_{k|k} \quad \textbf{(4.24)}$$

$$P_{k+1|k} = \Phi_k P_{k|k} {\Phi_k}^T + Q_k \quad \textbf{(4.25)}$$

$$t_{k+1} = t_k + \Delta t \quad \textbf{(4.26)}$$

In the update stage, the estimated states are updated using the measurements at that time. The Kalman gain is the optimal gain that minimizes the mean squared errors of the states and it can be thought as a blending factor between the dynamic model and the measurements. The equations for update stage are given in (4.27), (4.28) and (4.29).

$$K_k = P_{k|k-1} {H_k}^T \left[H_k P_{k|k-1} {H_k}^T + R_k \right]^{-1} \quad \textbf{(4.27)}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \left[z_k - H_k \hat{x}_{k|k-1} \right] \quad \textbf{(4.28)}$$

$$P_{k|k} = [I - K_k H_k] P_{k|k-1} \quad \textbf{(4.29)}$$

5. MULTISENSOR AIDED INS/GPS INTEGRATION SYSTEM

State estimation in unmanned ground vehicles (UGV) is very important subject since in order to accomplish such applications like planning, guidance and control applications, the kinematic states of the vehicle should be estimated accurately [1]. These kinematic states include the three dimensional position, velocity and orientation of the vehicle and they are named as navigation states.

In order to estimate the navigation states accurately with desired reliability, the use of multiple sensors with various types is essential. Generally, those vehicles are equipped with IMU and GPS receiver since the most popular state estimation method for UGV is the integration of INS and GPS navigation systems. Using their complementary properties, the fusion of their data gives the best result than the systems that are used alone. Since the equations of motion in INS (dynamic model) are nonlinear and GPS measurement model may be nonlinear, the Extended Kalman Filter (EKF) is generally used for INS/GPS fusion process.

Most of the applications, INS provides the estimation of navigation states at high rates than GPS measurements. Usually, the frequency of INS is 100 Hz on the other hand GPS provides measurements at 1Hz or 10 Hz. So, it could be stated that INS provides the navigation states between two GPS measurements. The EKF only propagates the state estimations and error covariance matrix between two GPS measurements. In this time interval, the uncertainty in the system grows due to errors in the IMU measurements, the errors in the system modeling and the errors in the linearization of nonlinear equations. This stage is the prediction part in EKF algorithm.

When the GPS measurement arrives, Kalman Gain is calculated with respect to the probabilistic properties of measurements and the dynamic model of the system which are carried through the error covariance matrix and the measurement noise matrix. After Kalman gain is calculated, the predicted state is updated and the error

covariance matrix is also updated. This stage is the update part in EKF.By this way, the difference in actual navigation states and the estimated navigation states are kept small. This is important for EKF which performs linearization around the estimated states. GPS acts like a reset system for INS since it corrects the errors in estimated navigation states.

The performance and accuracy of INS becomes a great issue when there is GPS outage and between the GPS measurement intervals. Without corrections of GPS, the errors in navigation state estimates accumulate with time since INS is an integrative process. The error in previous steps propagates to next state estimates. In this situation, INS is short term reliable because of its error characteristics. In order to have robust, accurate and reliable navigation state estimates in case of GPS outages, the accuracy of INS should be improved using aiding sensors like digital compass, wheel encoder, odometer, visual system and etc. In this work, the accuracy of INS is improved by using a digital compass and a wheel encoder.

In 5.1, The EKF structure that is implemented in this work is basically introduced. The algorithm is given in section 5.2 with details of aiding system in the prediction part. The linearization of the nonlinear navigation equations and forming the Jacobian matrix is also introduced.

5.1 The Extended Kalman Filter Structure for INS/GPS Integration

In this section, an EKF is implemented to fuse the GPS measurements with the estimated navigation states which are propagated using the nonlinear dynamic model. Since the equations of motion are nonlinear, linearization should be done about the estimated states using Taylor series expansion.

In some papers and works, the EKF is thought as the combination of INS which is the reference trajectory $x^*(t)$ and the Error State KF whose dynamic model is the error model of the navigation equations. The state matrix and the measurement matrix is calculated using the Jacobians of the nonlinear navigation equations. This KF estimates the error states using the error measurements which are formed by the difference of GPS measurements and the INS estimations. Then the estimated errors

are fedback to the INS system to correct the INS estimates. This structure is called loosely-coupled since there is correction to the navigation state estimates of INS. There is no correction to the aiding navigation system. In tightly-coupled integration systems, the KF also estimates the GPS errors and it corrects the GPS aiding system. In this work, only the errors of INS are corrected. In figure 5.1, the loosely-coupled integration system is illustrated.

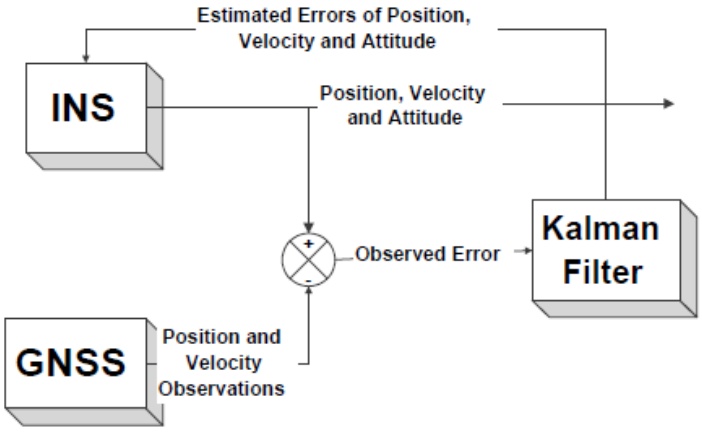


Figure 5.1 Loosely-coupled INS/GPS integration system [4].

For the navigation systems, the EKF is very popular and it is proved to be useful because the linearized dynamic model and the measurement models are accurate for the short period of time, i.e. between the GPS measurements [21]. The linearization is performed relative to a reference trajectory, which should be close to the true trajectory. The GPS measurement system keeps the linearization errors small so that the nominal trajectory and the true trajectory are close. During long GPS outages, the uncertainty in the prediction stage i.e. INS, grows high and the nominal trajectory becomes far away from the true trajectory. This leads inaccurate linearization and even it may lead to instabilities in the EKF [22].

In this work, the accuracy of the INS is improved by using aiding sensors and Kalman filters inside the INS algorithm which is given in 3.2.2. The algorithm, given in the section 3.2.2, propagates the navigation states only integrating the IMU measurements of rotation rates and accelerations. The errors in the IMU measurements are directly integrated every period that causes error growth in the navigation states depending on time. In order to get higher accuracy in navigation

states during GPS outages, aiding sensors are needed to keep the error between the predicted trajectory and the true trajectory. Seperate Kalman filters are used in orientation estimation and the velocity estimation in prediction stage of the EKF. The algorithm of the EKF is given in the figure 5.2.

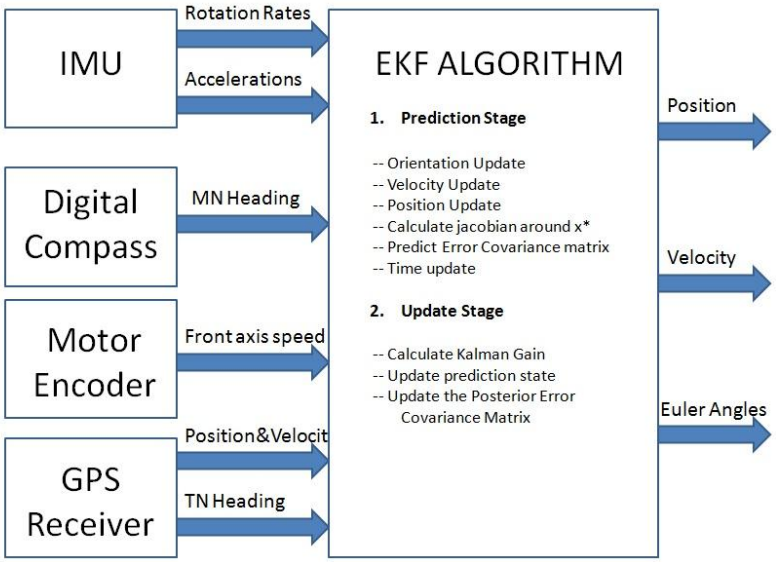


Figure 5.2 The EKF structure used in this work.

5.2 The EKF Algorithm

EKF algorithm is used when either the dynamic model of the system or the measurement model of the system is nonlinear. In navigation equations, the dynamic model of the system is nonlinear. EKF algorithm has the same prediction and update stages as KF, but in order to propagate its error covariance matrix and output estimates, EKF needs linearization of the nonlinear system model or measurement model around the reference trajectory $x^*(t)$ which is a solution of $\dot{x}^*(t)$=f(x^*, u, t).

There are three stages in EKF just like KF algorithm which is given in chapter 4. Actually the only difference in the algorithm is the finding the Jacobians of the nonlinear equations in order to propagate the error covariance matrix and the measurement model.

5.2.1 Initialization of EKF algorithm

The EKF is a recursive algorithm, so it should be initialized first as given in (5.1-4).

$$t_k = k = 0 \quad \textbf{(5.1)}$$

$$\hat{x}_{0|0} = E[x_0] \quad \textbf{(5.2)}$$

$$\delta\hat{x}_{0|0} = x_0 - \hat{x}_{0|0} = 0 \quad \textbf{(5.3)}$$

$$P_{0|0} = \mathrm{cov}(x_0) \quad \textbf{(5.4)}$$

The diagonal elements of initial error covariance matrix are chosen to be large values at initialization.

5.2.2 The prediction stage

In prediction stage the nonlinear dynamic equations are propagated using the estimated navigation states by integrating the equation $\dot{x}^*(t)$=f(x^*, u, t) over the time interval $t \in [t_k, t_{k+1}]$ with the initial condition $x^*(t_k) = x_{k|k}$ Actually, since GPS measurements occur at distant time intervals, a discrete time equivalent model of the navigation equations should be utilized.

$$\hat{x}_{k+1|k} = f(\hat{x}_{k|k}, u_k, t_k) \quad \textbf{(5.5)}$$

In figure 5.1, the block diagram named as INS corresponds to (5.5). After the estimated state is propagated, the reference trajectory becomes$x^*(t_{k+1}) = \hat{x}_{k+1|k}$.

The prediction stage is of INS is given in section 3.2.2. First orientation propagation is performed to build the transformation matrix R_b^n, then the velocity propagation is performed using the accelerometer measurements. After the velocity is predicted, the position could be easily calculated using 2^{nd} order Runge-Kutta integration method.

For aided INS, in the orientation estimation, two Kalman Filters are used in cascade in order to estimate the elements of Direction Cosine Matrix by using the measurements from IMU, GPS and digital compass. In order to select the best heading measurement in the system, a rule based decision algorithm is also implemented. The accuracy in the estimation of transformation matrix R_b^n is important since IMU and wheel encoder outputs their measurements in vehicle body frame. So in order to use these measurements in navigation frame, a transformation

should be done. Any errors in this transformation matrix cause the transformed measurements to be miscalculated. This leads errors in velocity and also in position. The orientation estimation is given in section 5.2.3. The second part is the velocity update. In this part, the velocity of the vehicle in navigation frame is updated by the IMU acceleration measurements in which the Coriolis acceleration, centripetal accelerations and the effect of gravity are compensated. So the errors in accelerometer measurements are integrated and thus these errors are accumulated with time. This leads errors in updated velocity as well as the position estimates. In this work, a wheel encoder is used in order to have a velocity measurement of vehicle forward axis. By using non-holonomic constraints acting on a vehicle travelling on surface of Earth, navigation frame velocity vector measurements could be formed. In order to fuse these measurements with the velocity estimates, an extended Kalman filter is implemented. By this way, the accuracy of the velocity estimates is improved and the errors in these estimates are bounded. The velocity estimation is given in 5.2.4.

For propagation of the error covariance matrix, the Jacobian matrix of the nonlinear navigation equations should be calculated. This calculation is given in section 5.2.5.

5.2.3 DCM based orientation estimation

Orientation is the term used for the pose of a vehicle on the surface of Earth. The orientation can be represented by three methods. These are the Direction Cosine Matrix, Euler angles and the quaternions. All three methods are just set of kinematic differential equations with different parameterization.

Therefore to estimate orientation accurately, multi-sensor stand-alone methods are proposed. In [17], a Direction Cosine Matrix (DCM) based orientation estimation algorithm is presented with two cascaded Kalman Filters. A low cost strapdown inertial measurement unit (IMU) and a magnetic compass are sampled at the same frequency to use Kalman Filter directly. In another study, the Quaternion method in the computer frame approach is used to estimate orientation and to derive INS algorithm for low cost IMU to solve the initial attitudes uncertainty using in-motion alignment. The distribution approximation filter (DAF) is used to implement non-

linear data fusion algorithm [8]. In [19], a fuzzy multi-sensor fusion of orientation estimation using Euler method is presented.

In this work, DCM method is used for orientation representation due to its linear model and also because of the fact that no singularity occurs with this method for land vehicle applications. Using this method, the transformation matrix, R_b^n, also named as direction cosine matrix, is directly estimated. One of the important advantages of this method arises in here since there is no extra computation for calculating the transformation matrix that is needed for navigation equations.

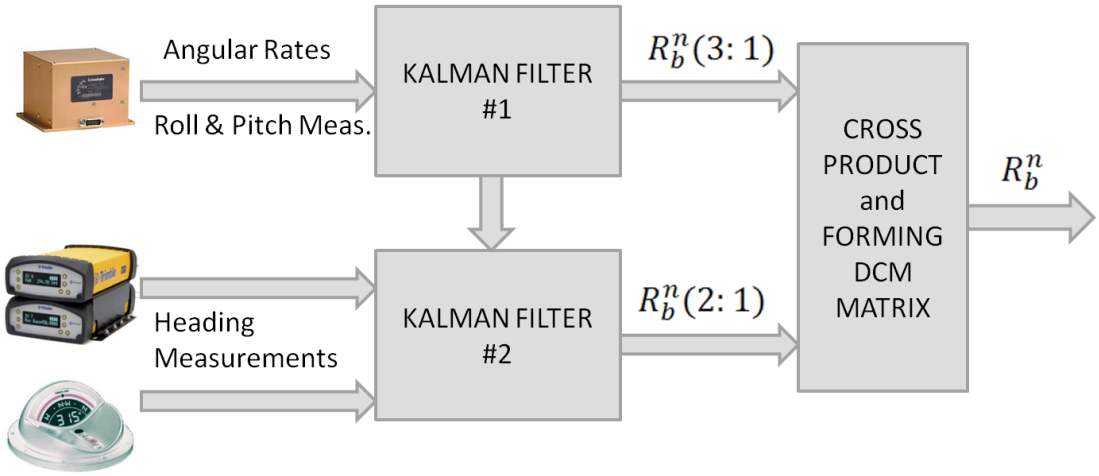


Figure 5.3 The overview of the orientation estimation algorithm.

5.2.3.1 Basics of DCM method

The relationships between vectorial quantities in the body frame and the navigation frame can be described by forming a rotation matrix R_n^b. It is formed by three consecutive plane rotations involving Euler angles, yaw (ψ), pitch (θ), and roll angle (ϕ), respectively. The DCM method is based on integrating the measured angular rate from gyros with the following relationship where R_b^n is body frame to navigation frame rotation matrix.

$$\dot{R}_b^n = R_b^n[\omega_{nb}^b x] \tag{5.6}$$

$$\omega_{nb}^b = \left(\omega_{ib}^b - \omega_{in}^b\right) \tag{5.7}$$

If the position and velocity information are unavailable then ω_{in}^{b} could be ignored because this term is dependent on position and velocity. ω_{ib}^{b} is the angular velocity of the body frame with respect to the inertial frame resolved in body frame (gyro measurements) and $[\omega_{nb}^{b}\mathrm{x}]$ is the skew-symmetric representation of ω_{nb}^{b}.

Using orthogonality property of the rotation matrix, it is sufficient to integrate only two rows separately. At first step, a Kalman filter is used for estimating the third row of R_b^n using gyro measurements, where roll and pitch angle could be easily calculated from the third row of R_b^n. This is followed by a second Kalman filter which fuses the heading information from GPS and a digital compass which has different sampling frequency from the algorithm to estimate the second row of R_b^n, through which the yaw angle is easily calculated.

5.2.3.2 The estimation of the third row of DCM

In this part, the third row of DCM is estimated by using angular rate and acceleration measurements of IMU. The roll and pitch angle could be easily calculated from the elements of the third row of the DCM. The Euler angles are easier to comprehend so the results will be discussed in Euler angles.

Generally, accelerometers (tilt sensors) are used to estimate roll and pitch angles using the projection of gravity vector in each axis. But to do this calculation accurately, the vehicle should be going at a constant speed or should be at rest. So the tilt sensors only measure the projection of gravity vector, not the sum of motional acceleration and the projection gravity vector. To use the tilt sensors as attitude angle measurements, vehicle stops are proposed to correct the roll and pitch angles periodically [19].

Some Inertial Measurement Units (IMU) provides stabilized pitch and roll angles by combining the angular rate sensors with accelerometers. The weighting between the accelerometers and gyros is controlled by a parameter named erection rate. In general for dynamic measurements low erection rate is appropriate since IMU depends more on gyros than the accelerometers to calculate the stabilized pitch and roll angles. In this work, high erection rate (100+) is set when the vehicle is at rest [20]. To avoid interference on the IMU for changing the erection rate any time whether the motion

is dynamic or slow, a Kalman filter could be designed to provide optimal estimates for both situations by simply changing the measurement covariance matrix. However, in this work the IMU is configured to output its data in scaled sensor mode so the roll and pitch angles should be calculated by using acceleration measurements. The state space representation of the process model is given as follows in equation (5.8).

$$\dot{\mathrm{R}}_{\mathrm{b}}^{\mathrm{n}}(3:) = -\left[\omega_{\mathrm{nb}}^{\mathrm{b}}\mathrm{x}\right]\mathrm{R}_{\mathrm{b}}^{\mathrm{n}}(3:) + \mathrm{w}_1(\mathrm{t}) \tag{5.8}$$

Where $R_b^n(3:)$ is the third row of the rotation matrix and the states of the system. Although the state matrix is time varying, it is constant between two sampling times. The process noise $w_1(t)$ is assumed to be zero-mean Gaussian white noise with the power spectral density (PSD) Q_w (assumed to be diagonal) . The measurement model is given as in equation (5.9).

$$z(t)^T = [\, f_{ib,x}^b \quad f_{ib,y}^b \quad f_{ib,z}^b] = \mathrm{g\ x\ I_{3x3}R_b^n}(3:) + \mathrm{v_1(t)} \tag{5.9}$$

Where the measurement $z(t)$ is formed by the specific force measurements of the IMU body frame. The measurement noise $v_1(t)$ is assumed to be zero-mean Gaussian white noise with the measurement noise covariance matrix of $R_1(k)$ (assumed to be diagonal). Φ_k is the state transition matrix and G is the input matrix. The term g is the gravitational acceleration which is assumed to be 9.8 m/s^2.

In order to use Kalman filter, the process model in (5.8) and the measurement model in (5.9) should be transformed to discrete time as given below.

$$\mathrm{x_{k+1}} = \Phi_{\mathrm{k}}\mathrm{x_k} + \mathrm{w_k} \tag{5.10}$$

$$\Phi_{\mathrm{k}} = \mathrm{e}^{\mathrm{F}\Delta\mathrm{t}} = \mathrm{I} + \mathrm{F}\Delta\mathrm{t} + \frac{\mathrm{F}^2\Delta\mathrm{t}^2}{2!} + \cdots \cong \mathrm{I} + \mathrm{F}\Delta\mathrm{t} \tag{5.11}$$

$$\omega_{\mathrm{k}} = \int_{\mathrm{t_k}}^{\mathrm{t_{k+1}}} \Phi(\mathrm{t_{k+1}}, \tau)\, \mathrm{G}(\tau)\omega(\tau)\mathrm{d}\tau \tag{5.12}$$

$$\mathrm{z_k} = \mathrm{H_k x_k} + \mathrm{v_k} \tag{5.13}$$

The noise covariance matrices $Q_1(k)$ and $R_1(k)$ are used as performance tuning parameters. The diagonal elements of the matrices $Q_1(k)$ and $R_1(k)$are selected to have a smooth response. The rate of motor encoder is calculated in order to find the motional acceleration of the vehicle in forward axis of the vehicle body frame and this rate is checked whether it is bigger than a threshold i.e. $1\ m/s^2$. If this condition is satisfied then it means the roll and pitch calculations using specific force measurements are not accurate since these measurements consist of both motional acceleration and projection of gravity. In this case, the diagonal elements of $R_1(k)$ is set to be bigger than $Q_1(k)$ to make Kalman Filter relying on dynamic model more than the measurements. Since new measurement is weighted less, the states converge slowly and vice versa [21]. However, if this condition is not satisfied, the diagonal elements of $R_1(k)$ is set to be smaller than $Q_1(k)$ to make Kalman filter trusting measurements more than the dynamic model.

5.2.3.3 Estimation of the second row of DCM

For heading measurement, generally GPS and digital compass are used. GPS heading information is more accurate especially for dynamic motion but an outage could be easily occurred dependent on the environment and the accuracy of the measurement could be changed according to the position of satellites in space. A digital compass is more reliable in a way, since there is no outage (unless sensor fails) but due to its slow response to dynamic motion, the heading information is time delayed and also it is sensitive to magnetic fields in the environment causing inaccurate measurements. In figure 5.4, the measurements of GPS and digital compass when the vehicle is at rest are shown. The GPS heading measurements have high variance when the vehicle steady. On the other hand, the digital compass heading measurement has a low variance but it is affected by the magnetic field change in the environment as seen in the figure. Blue dot are compass measurements and red dots are the GPS measurements.

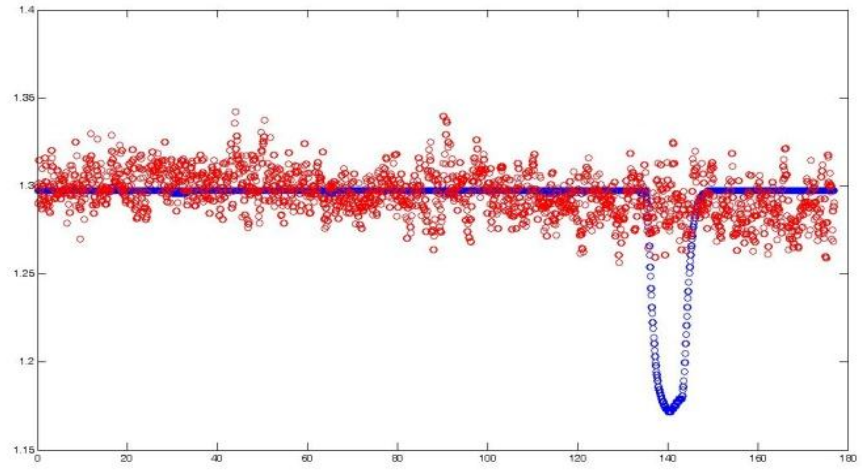


Figure 5.4: GPS and digital compass measurements when the vehicle is at rest.

Also in figure 5.5, the slow response of digital compass when the motion of the vehicle is dynamic could be seen.

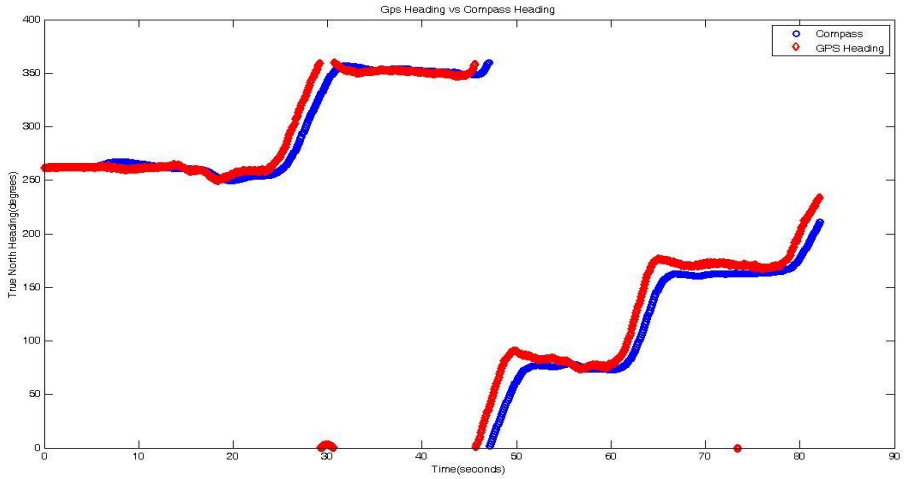


Figure 5.5: GPS and digital compass measurements when the vehicle is dynamic.

So a rule based decision structure should be constructed to select the best measurement depending on the environment and the motion of the vehicle. This structure is constructed using “if and else” logic given below in figure 5.6.

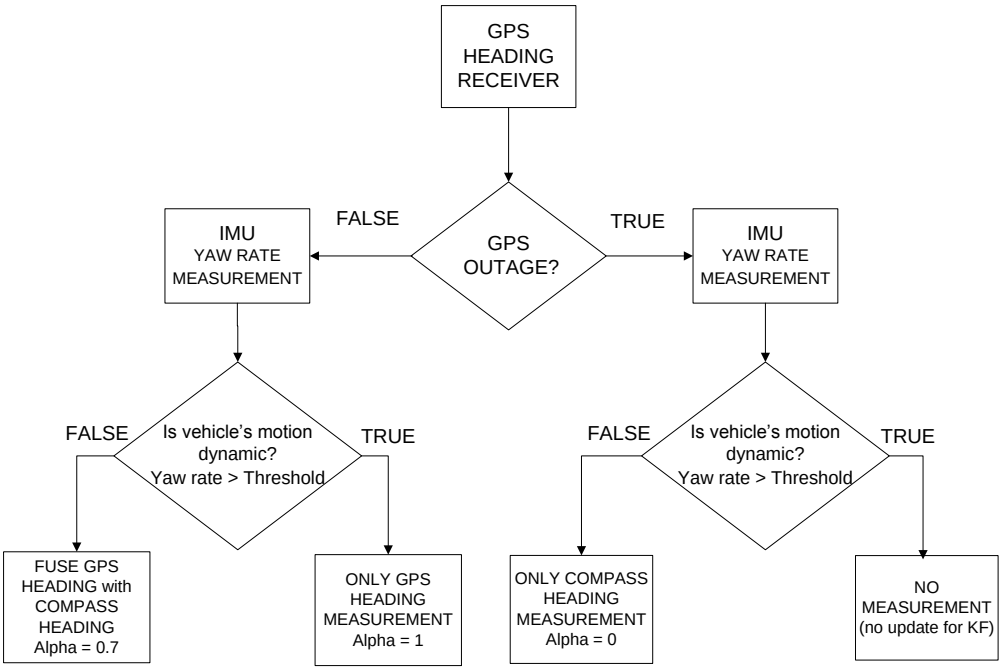


Figure 5.6: A rule based decision structure.

Whether there is GPS outage or not, the structure checks the yaw rate measurement from IMU whether the angular motion of the vehicle about z-axis is slow or dynamic. The threshold value is selected to be 0.1 radian / sec (5.73°/sec). This value is found experimentally.

Also, a digital compass measures the heading from the Magnetic North so it should be compensated with magnetic declination (δ) to align with the GPS heading measurement which is with respect to True North as given in (5.14). Magnetic declination is equal to + 4° 37' for Istanbul/TURKEY.

$$\psi_{\text{True North}} = \psi_{\text{Magnetic North}} + \delta \qquad \textbf{(5.14)}$$

For fusing the GPS heading (ψ_{GPS}) and the compensated compass heading measurement ($\psi_{compass}$), a simple weighting equation given in (5.15) is used where $0 \leq \alpha \leq 1$.

$$\psi_{\text{Fused}} = \alpha\, \psi_{\text{GPS}} + (1 - \alpha)\, \psi_{\text{compass}} \qquad \textbf{(5.15)}$$

The parameter α is selected according to the quality of the GPS heading measurement and Dilution of Precision (DOP) value. For example if the DOP is low

(i.e. 1-5) and the quality is Real Time Kinematics (RTK) then α is chosen to be 0.7 to weight GPS measurement more.

The state space representation of the process model is given as follows in (5.16).

$$\dot{\mathrm{R}}_{\mathrm{b}}^{\mathrm{n}}(2:) = -\left[\omega_{\mathrm{nb}}^{\mathrm{b}}\mathrm{x}\right]\mathrm{R}_{\mathrm{b}}^{\mathrm{n}}(2:) + \mathrm{w}_2(\mathrm{t}) \quad \textbf{(5.16)}$$

Where $R_b^n(2:)$ is the second row of the rotation matrix and the states of the system. Although the state matrix is time varying, it is constant between two sampling times.

The process noise $w_2(t)$ assumed to be zero-mean Gaussian white noise with the power spectral density (PSD) Q_{w2} (assumed to be diagonal). The measurement model is given as in (5.17) and (5.18).

$$\mathrm{z(t)}^{\mathrm{T}} = [\cos\hat{\theta}\sin\tilde{\psi}\ ; \cos\hat{\phi}\cos\tilde{\psi} + \sin\hat{\phi}\sin\hat{\theta}\sin\tilde{\psi}\ \ldots \quad \textbf{(5.17)}$$
$$;\ -\sin\hat{\phi}\cos\tilde{\psi} + \cos\hat{\phi}\sin\hat{\theta}\sin\tilde{\psi}]$$

$$\mathrm{z(t)} = \mathrm{I}_{3\mathrm{x}3}\mathrm{R}_{\mathrm{b}}^{\mathrm{n}}(2:) + \mathrm{v}_2(\mathrm{t}) \quad \textbf{(5.18)}$$

Where the measurement $z(t)$ is formed by the roll and pitch angle estimates of the first Kalman filter ($\hat{\phi}$ and $\hat{\theta}$ respectively) and the heading angle measurement ($\tilde{\psi}$), that is formed from the decision structure.

The measurement noise $v_2(t)$ is assumed to be zero-mean Gaussian white noise with the noise covariance matrix of $R_2(k)$ (assumed to be diagonal). Since there are two measurement sources, $R_{GPS}(k)\ and\ R_{compass}(k)$ are different. The related noise covariance matrix is selected with the related measurement used in Kalman filter in that time interval. $R_{GPS}(k)$ is determined according to DOP value and quality of the GPS solution experimentally. $R_{compass}(k)$ is selected relatively bigger than $R_{GPS}(k)$ according to trust degree.

Since measurements are made at 10 Hz and Kalman filter works at 100Hz, the conventional Kalman filter design cannot be used. In this case, prediction steps are computed every time but the update steps are computed each time the measurement arrives.

After normalization is done to form the whole rotation matrix R_b^n using orthogonality property, $R_b^n(1:)$ could be found by the equation given in (5.19).

$$\mathrm{R_b^n}(1:) = \mathrm{R_b^n}(2:)\mathrm{x}\mathrm{R_b^n}(3:) \quad \textbf{(5.19)}$$

5.2.4 Velocity estimation using a wheel encoder

Velocity estimation is very important since it is the most effective error source in position estimates of INS. Any error in velocity estimate propagates in position estimate since position is the integration of velocity.

During GPS outages, the velocity estimate error accumulates with time since there is no velocity measurement. In order to improve the accuracy in velocity estimates, using non-holonomic constraints for a land vehicle is proposed [4]. By using these constraints with the help of an aiding sensor that provides the forward speed, such as a wheel encoder, a measurement model can be implemented.

5.2.4.1 Three dimensional motion of a vehicle on surface

In this part, the motion of a vehicle travelling on Earth surface is constrained in order to bound the error growth of the velocity of the vehicle. If the three dimensional equations of motion of an object is considered, it is applied for both an object travelling in sky and an object travelling on surface. However, it can be stated that those two motions are clearly different and there are some constraints for a vehicle moving on surface.

In [4], two non-holonomic constraints are considered for the motion of a wheeled vehicle on a surface. Under ideal conditions for a wheeled vehicle travelling on surface, there is no side slip along the transversal direction i.e. the vehicle does not slide on the surface, and there is no motion perpendicular to the surface of motion i.e. the vehicle does not jump off the ground. So under ideal conditions, the constraints could be defined as given in (5.20). The transversal velocity and the normal velocity should be zero.

$$\mathrm{v_y} = \mathrm{v_z} = 0 \quad \textbf{(5.20)}$$

This assumption is not realistic for any practical situation. For example during cornering, the slide slip occurs and these constraints are violated due to some degree. So if these effects are considered the constraints could be modelled approximately as in (5.21-22).

$$v_y - \eta_y = 0 \quad \textbf{(5.21)}$$

$$v_z - \eta_z = 0 \quad \textbf{(5.22)}$$

In equation (5.21-22), the random variables η_y and η_z are selected as Gaussian white noise with zero mean and with variances ${\sigma_y}^2$ and ${\sigma_z}^2$ respectively. Those variance values can be selected to reflect the expected violations on the constraints [4]. For example high variance could be selected when the vehicle is turning to reflect the side slip.

By using these constraints given in (5.21-22), the three dimensional velocity of the vehicle's body frame could be given as in (5.23).

$$v_e^b = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} \tilde{v}_x \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \eta_x \\ \eta_y \\ \eta_z \end{bmatrix} \quad \textbf{(5.23)}$$

In this observation model, the forward velocity of the vehicle should be observed from a sensor such as a wheel encoder. In the given equation above, the measurement of forward velocity is denoted as $\tilde{v}_x$ and η_x denotes the Gaussian white noise, with zero mean and variance ${\sigma_x}^2$, of the measurements.

Since the navigation equations are performed in navigation frame, the observation of vehicle velocity in navigation frame is considered. So using the transformation matrix R_b^n, equation (5.23) could be easily transformed to navigation frame.

$$v_e^n = R_b^n \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = R_b^n \begin{bmatrix} \tilde{v}_x \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \tilde{v}_x \cos\theta \cos\psi \\ \tilde{v}_x \cos\theta \sin\psi \\ -\tilde{v}_x \sin\theta \end{bmatrix} \quad \textbf{(5.24)}$$

5.2.4.2 Extended Kalman filter structure

The velocity dynamic equation for a vehicle in navigation frame is written as in (5.25). This equation is the process model for velocity estimation. Since the term Ω_{ie}^{n} depends on velocity vector, this dynamic equation is nonlinear. Because of the nonlinearity of process model, an EKF should be used for velocity estimation. The states of the system are the velocity vector in navigation frame as given in (5.26) and the nonlinear stochastic system could be written as in (5.27-5.28).

$$\dot{v}^{n} = R_{b}^{n} f_{ib}^{b} + g^{n} - (\Omega_{en}^{n} + 2\Omega_{ie}^{n}) v^{n} \quad \textbf{(5.25)}$$

$$v^{n} = x = [v_{n} \quad v_{e} \quad v_{d}]^{T} \quad \textbf{(5.26)}$$

$$\dot{x}(t) = f(x, u, t) + w(t) \quad \textbf{(5.27)}$$

$$y(t) = h(x, t) + v(t) \quad \textbf{(5.28)}$$

For propagating its covariance matrix and measurement updates, the EKF requires linearization. The linearization is performed relative to a reference trajectory $x^{*}(t)$. Using Taylor series expansion, the nonlinear equation at (5.25) could be written as in (5.29). The high order terms (H.O.T) are generally ignored to ease the computation. The Jacobian should be computed and discreatization should be done in order to execute EKF algorithm. The Jacobian is given at (5.30).

$$\dot{x}(t) = f(x^{*}, u, t) + \frac{\partial f}{\partial x}\big(x(t) - x^{*}(t)\big) + H.O.T + w(t) \quad \textbf{(5.29)}$$

$$F(t) = \frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} \end{bmatrix}_{x(t)=x^{*}(t)} \quad \textbf{(5.30)}$$

The equivalent discrete time model of the model in (5.29) is given in (5.31).

$$x_{k} = f\big(x_{k-1|k-1}, u_{k-1}, t_{k-1}\big) + (I + F\Delta t)\big(x_{k} - x_{k-1|k-1}\big) + H.O.T + w_{k} \quad \textbf{(5.31)}$$

Since EKF is a nonlinear estimation process, $x_k - x_{k-1|k-1}$ should be kept small [21]. Because this approximation holds if the reference trajectory is near to the actual trajectory.

5.2.5 Linearization of navigation equations

The nonlinear navigation equations are given in (5.32-34). These equations are linearized using Taylor series expansion given in (5.29) around the estimated navigation states.

$$\dot{r}^n = \begin{bmatrix} \dot{\varphi} \\ \dot{\lambda} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} \frac{v_n}{R_M + h} & \frac{v_e}{\cos\varphi\,(R_N + h)} & -v_d \end{bmatrix}^T \quad \textbf{(5.32)}$$

$$\dot{v}^n = R_b^n f_{ib}^b + g^n - (\Omega_{en}^n + 2\Omega_{ie}^n)v^n \quad \textbf{(5.33)}$$

$$\dot{R}_b^n = R_b^n \Omega_{ib}^b - (\Omega_{ie}^n + \Omega_{en}^n)R_b^n \quad \textbf{(5.34)}$$

The error state is defined to be the difference between true value of the navigation states and estimated value of the navigation states. The error state and error state dynamics are given in (5.35-36).

$$\delta x = x - \hat{x} \quad \textbf{(5.35)}$$

$$\delta\dot{x} = \dot{x}(t) - f(\hat{x}, u, t) = \frac{\partial f}{\partial x}\delta x + w(t) \quad \textbf{(5.36)}$$

In navigation frame, the error vector of position, velocity and orientation is denoted as given in (5.37-43).

$$\delta x^n = [\delta r^n \quad \delta v^n \quad \epsilon^n]^T \quad \textbf{(5.37)}$$

$$\delta r^n = r^n - \hat{r}^n = [\delta\varphi \quad \delta\lambda \quad \delta h]^T \quad \textbf{(5.38)}$$

$$\delta v^n = v^n - \hat{v}^n = [\delta v_n \quad \delta v_e \quad \delta v_d]^T \quad \textbf{(5.39)}$$

$$R_b^n = (I + E^n)\hat{R}_b^n \quad \textbf{(5.40)}$$

$$E^n = [\epsilon^n x],\ \epsilon^n = [\epsilon_n \quad \epsilon_e \quad \epsilon_d]^T \quad \textbf{(5.41)}$$

$$\delta f_{ib}^{b} = f_{ib}^{b} - \hat{f}_{ib}^{b} \quad \textbf{(5.42)}$$

$$\delta \omega_{ib}^{b} = \omega_{ib}^{b} - \hat{\omega}_{ib}^{b} \quad \textbf{(5.43)}$$

The dynamic equation of error model in navigation frame and the state space error model is given in (5.44-45).

$$\dot{\delta x}^{n} = F\delta x^{n} + Gw \quad \textbf{(5.44)}$$

$$\begin{bmatrix} \dot{\delta r}^{n} \\ \dot{\delta v}^{n} \\ \dot{\epsilon}^{n} \end{bmatrix} = \begin{bmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{bmatrix} \begin{bmatrix} \delta r^{n} \\ \delta v^{n} \\ \rho^{n} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -\hat{R}_{b}^{n} & 0 \\ 0 & \hat{R}_{b}^{n} \end{bmatrix} \begin{bmatrix} \delta f_{ib}^{b} \\ \delta \omega_{ib}^{b} \end{bmatrix} \quad \textbf{(5.45)}$$

5.2.5.1 The error dynamics of position

The position dynamic equation given in (5.32) is perturbed to obtain linearized position error dynamics. The derivation of this equation is included in [1]. The Jacobian matrices are given in (5.47-49).

$$\dot{\delta r}^{n} = F_{11}\delta r^{n} + F_{12}\delta v^{n} \quad \textbf{(5.46)}$$

$$F_{11} = \begin{bmatrix} 0 & 0 & \dfrac{-\hat{v}_{n}}{\left(R_{m} + \hat{h}\right)^{2}} \\ \dfrac{\hat{v}_{e}\sin(\hat{\varphi})}{\left(\left(R_{N} + \hat{h}\right)\cos(\hat{\varphi})^{2}\right)} & 0 & \dfrac{-\hat{v}_{e}}{\left(\left(R_{N} + \hat{h}\right)^{2}\cos(\hat{\varphi})\right)} \\ 0 & 0 & 0 \end{bmatrix} \quad \textbf{(5.47)}$$

$$F_{12} = \begin{bmatrix} \dfrac{1}{\left(R_{m} + \hat{h}\right)} & 0 & 0 \\ 0 & \dfrac{1}{\left(\left(R_{N} + \hat{h}\right)\cos(\hat{\varphi})\right)} & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad \textbf{(5.48)}$$

$$F_{13} = 0_{3x3} \quad \textbf{(5.49)}$$

5.2.5.2 The error dynamics of velocity

The velocity dynamic equation given in (5.33) is perturbed to obtain linearized velocity error dynamics. The Jacobian matrices are given in (5.51-53).

$$\dot{\delta v}^n = F_{21}\delta r^n + F_{22}\delta v^n + F_{23}\epsilon^n - R_b^n \delta f_{ib}^b \tag{5.50}$$

$$F_{21} = \begin{bmatrix} -2\Omega_N v_e - \dfrac{\rho_n v_e}{\cos^2\varphi} & 0 & \rho_e k_d - \rho_n \rho_d \\ 2(\Omega_N v_n + \Omega_D v_d) + \dfrac{\rho_n v_n}{\cos^2\varphi} & 0 & -\rho_n k_d - \rho_e \rho_d \\ -2\Omega_D v_e & 0 & F_{63} \end{bmatrix} \tag{5.51}$$

$$F_{22} = \begin{bmatrix} k_D & 2\omega_D & -\rho_e \\ -(\omega_{\boldsymbol{d}} + \Omega_D) & (k_D - \rho_e \tan\varphi) & \omega_N + \Omega_N \\ 2\rho_e & 2\omega_N & 0 \end{bmatrix} \tag{5.52}$$

$$F_{23} = \begin{bmatrix} 0 & f_{ib,D}^n & -f_{ib,E}^n \\ -f_{ib,D}^n & 0 & f_{ib,N}^n \\ f_{ib,E}^n & -f_{ib,N}^n & 0 \end{bmatrix} = [f_{ib}^n \times] \tag{5.53}$$

5.2.5.3 The error dynamics of orientation

The orientation dynamic equation given in (5.34) is perturbed to obtain linearized velocity error dynamics. The Jacobian matrices are given in (5.55-57).

$$\dot{\epsilon} = F_{31}\delta r^n + F_{32}\delta v^n + F_{33}\epsilon^n + R_b^n \delta w_{ib}^b \tag{5.54}$$

$$F_{31} = \begin{bmatrix} w_{ie}\sin\varphi & 0 & \dfrac{v_e}{(R_N + h)^2} \\ 0 & 0 & \dfrac{-v_n}{(R_M + h)^2} \\ w_{ie}\cos\varphi + \dfrac{v_e}{(R_N + h)\cos^2\varphi} & 0 & \dfrac{-v_e \tan\varphi}{(R_N + h)^2} \end{bmatrix} \tag{5.55}$$

$$F_{32} = \begin{bmatrix} 0 & \dfrac{-1}{(R_N + h)} & 0 \\ \dfrac{1}{(R_M + h)} & 0 & 0 \\ 0 & \dfrac{\tan\varphi}{(R_N + h)} & 0 \end{bmatrix} \tag{5.56}$$

$$F_{33} = \begin{bmatrix} 0 & \omega_D & -\omega_E \\ -\omega_D & 0 & \omega_N \\ \omega_E & -\omega_N & 0 \end{bmatrix} = [\omega_{ib}^n \times] \tag{5.57}$$

After the Jacobian matrices are calculated, the EKF algorithm could be implemented. The error dynamics are given in (5.36). The discrete time equivalent of the error dynamics model given in (5.36) can be written as in (5.58-59).

$$\delta x_{k+1} = x(t_{k+1}) - \hat{x}_{k+1|k} = \Phi_k \delta x_k + w_k \quad \textbf{(5.58)}$$

$$\delta y_k = y(t_{k+1}) - h(\hat{x}_{k+1|k}, k) = H_k \delta x_k + v_k \quad \textbf{(5.59)}$$

The term $x(t_{k+1})$ corresponds to the true values of navigation states where the term $y(t_{k+1})$ corresponds to the measurements of GPS. The navigation states calculated at prediction part of EKF is denoted as $\hat{x}_{k+1|k}$ and the output states formed by using nonlinear equation h (.) are denoted as $h(\hat{x}_{k+1|k}, k)$.

In order to propagate the error covariance matrix, the state transition matrix and the state noise covariance matrix should be calculated as given in (5.60-61).

$$\Phi_k = e^{F\Delta t} = I + F\Delta t \quad \textbf{(5.60)}$$

$$Q_k \approx \Phi_k G Q_w G^T {\Phi_k}^T \Delta t \quad \textbf{(5.61)}$$

Since the initial value of the error states are set to zero, the propagation of the error states are not necessary. The important part in prediction stage is the propagation of error covariance matrix given in (5.62).

$$P_{k+1|k} = \Phi_k P_{k|k} {\Phi_k}^T + Q_k \quad \textbf{(5.62)}$$

5.2.6 The update stage

The update stage executes when the GPS measurements arrive. Between the GPS measurements, only the prediction stage executes and the navigation states and the error covariance matrix are propagated using the equations (5.5) and (5.60-62).

If the measurement model is nonlinear, it should be linearized around the estimate $\hat{x}_{k|k-1}$ and in order to calculate the EKF gain, the jacobian of the nonlinear matrix should be calculated. The nonlinear measurement model and its linearization is given in (5.63-64).

$$y(t_k) = h(x_k, k) \quad \textbf{(5.63)}$$

$$y(t_k) = h(\hat{x}_{k|k-1}, k) + H_k(x_k - \hat{x}_{k|k-1}) + H.O.T \quad \textbf{(5.64)}$$

In this work, the measurement model of GPS has a linear model, so there is no need for Jacobian calculation. The linear measurement model and the measurement error model are given in (5.65-68).

$$z_k = H_k x_k + v_k \quad \textbf{(5.65)}$$

$$\hat{y}_k = H_k \hat{x}_{k|k-1} \quad \textbf{(5.66)}$$

$$\delta y_k = z_k - \hat{y}_k = H_k\left(x_k - \hat{x}_{k|k-1}\right) + v_k \quad \textbf{(5.67)}$$

$$\delta y_k = H_k \delta x_k + v_k \quad \textbf{(5.68)}$$

Since GPS provides position and velocity measurements, the measurement matrix becomes 6x9 in size, given in (5.69-70).

$$z_k = H_k x_k + v_k \quad \textbf{(5.69)}$$

$$z_k = \begin{bmatrix} r^n_{GPS} \\ v^n_{GPS} \end{bmatrix} = \begin{bmatrix} I_{3x3} & 0_{3x3} & 0_{3x3} \\ 0_{3x3} & I_{3x3} & 0_{3x3} \end{bmatrix} \begin{bmatrix} r^n \\ v^n \\ EA \end{bmatrix} + v_k \quad \textbf{(5.70)}$$

The EKF gain could be calculated after H_k is calculated. The EKF gain is denoted as K_k and it is chosen as to minimize the conditional mean squared estimation errors [22]. The calculation of EKF gain is given at (5.71-72). The innovation covariance is denoted as S_k and R_k is the measurement covariance matrix.

$$K_k = P_{k|k-1} {H_k}^T S_k^{-1} \quad \textbf{(5.71)}$$

$$S_k = H_k P_{k|k-1} {H_k}^T + R_k \quad \textbf{(5.72)}$$

After EKF gain is computed, the predicted navigation states in prediction stage can be updated using the measurements. The innovation is defined as the difference between the actual measurement and the estimated measurement. The EKF basically fuses the predicted navigation states with the measurements by EKF gain. Therefore

EKF gain could be interpreted as a blending factor. The update stage is given in (5.73-74).

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\left(z_k - H_k\hat{x}_{k|k-1}\right) \quad \textbf{(5.73)}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\delta y_k \quad \textbf{(5.74)}$$

If the error dynamics are considered, then it could be written as in (5.75-78).

$$\delta\hat{x}_{k|k} = \delta\hat{x}_{k|k-1} + K_k\left(\delta y_k - H_k\delta\hat{x}_{k|k-1}\right) \text{ where } \delta\hat{x}_{k|k-1} = 0 \quad \textbf{(5.75)}$$

$$\delta\hat{x}_{k|k} = K_k\delta y_k \quad \textbf{(5.76)}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + \delta\hat{x}_{k|k} \quad \textbf{(5.77)}$$

$$\text{Set } \delta\hat{x}_{k|k} = 0 \quad \textbf{(5.78)}$$

In equation (5.77), it is seen that since the error is included in the state estimate, the error has been corrected. Therefore, the error state should be set to zero. This is called the feedback method. The equations (5.74) and (5.77) are equivalent. In figure 5.1, the KF structure estimates the error state vector by using the error dynamics and the residual measurement and using the feedback structure, the predicted navigation states (showed as INS in figure 5.1) are corrected by the error states that is estimated in KF structure (showed as KF in figure 5.1). Actually, those two structures (KF and INS) and the feedback complete the EKF structure [21].

After the navigation state is updated via the equation (5.74), the posterior error covariance matrix should also be updated by the equations (5.79).

$$P_{k|k} = [I - K_kH_k]P_{k|k-1} \quad \textbf{(5.79)}$$

6. THE SYSTEM ARCHITECTURE

In this chapter, the system architecture is introduced. At first, the unmanned ground vehicle project, named as Otonobil, is briefly introduced in 5.1. The hardware which is the sensors and the computers used in the vehicle is described in 5.2 and also the software that is implemented in this work is briefly described.

6.1 The Unmanned Ground Vehicle Testbed "Otonobil"

The Otonobil is an unmanned ground vehicle project conducted at the Istanbul Technical University (ITU) Mechatronics Education and Research Center (MEAM). It is basically an electric vehicle which has 6.5kW electrical motor and 130 Ah capacity Lead-Acid batteries. One of the objectives in this project is to implement an intelligent vehicle which has ability of perception of its environment, detects the obstacles and plans its trajectory to reach the goal points. The main aim is to implement such algorithms like path planning, mapping, obstacle avoidance, also semi-autonomous algorithms and active safety systems.

6.2 The Hardware

The hardware refers to the sensors and the computers that are used in Otonobil. In figure 6.1, 6.2 and 6.3 the hardware of Otonobil is shown. For mapping and obstacle detection algorithms, the Otonobil is surrounded with LIDAR sensors and ultrasonic sensors to perceive the environment. For state estimation algorithms, there are GPS antennas and a digital compass mounted on the roof of the car.

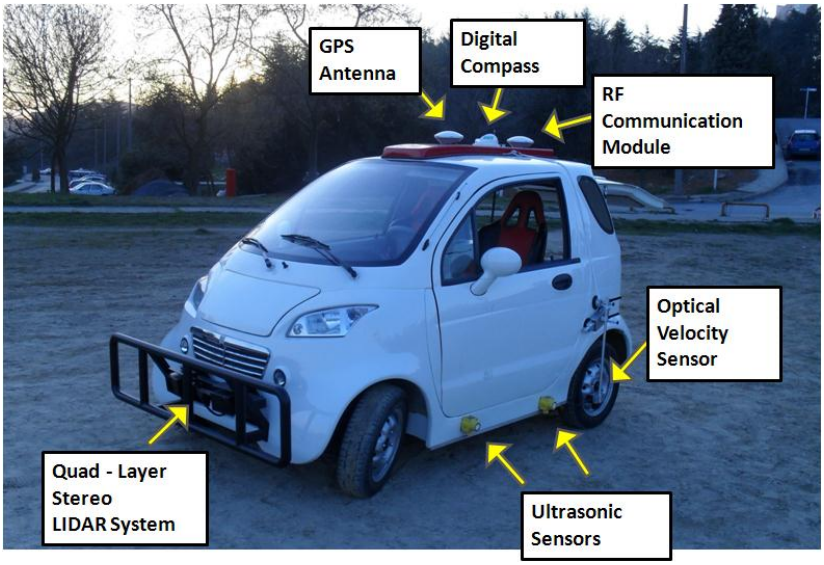


Figure 6.1: The sensors used in Otonobil (front view).

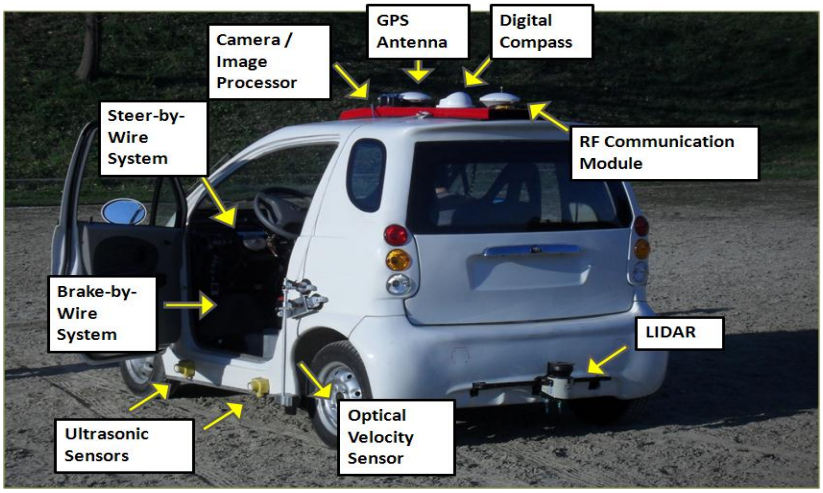


Figure 6.2: The sensors used in Otonobil (back view).

In figure 6.3, the computers that are used in Otonobil are shown. There are four main processors. The main processor is National Instruments PXI Real time system which has 2.26 GHz quad-core processor and a powerful FPGA module with several I/O cards. All the sensor are connected to this computer and all the sensor read, data acquisition and data logging processes are done by this computer. For low level control of the vehicle actuators i.e. brake motor and the steer motor, Dspace Microautobox is used. This device has 800 MHz processor and several I/O interfaces. For image processing, a Sony Camera is used. It is a smart camera, which can process images with its own processor and sends processed data via UDP/IP and

at last for LIDAR system, an IBEO ECU is used. It can give both object and raw data. It has tracking algorithms inside and can classify the objects around it. The table of the processors that are used is given in table 6.2 and they are illustrated in figure 6.3.

Table 6.1: The table of sensors

Sensor Type	Quantity	Brand/Model
Laser Scanner	2	Ibeo-lux
Laser Scanner	1	Sick lms 151
Camera	1	Sony-xci-sx100
Ultrasonic Sensor	4	Banner-qt50ulb
GPS	1	Trimble Sps851&Sps551H
Digital Compass	1	Kvh Azimuth 1000
IMU	1	Crossbow Vg700Ab-201
Optic Speed Sensor	1	Corsys-Datron LF II

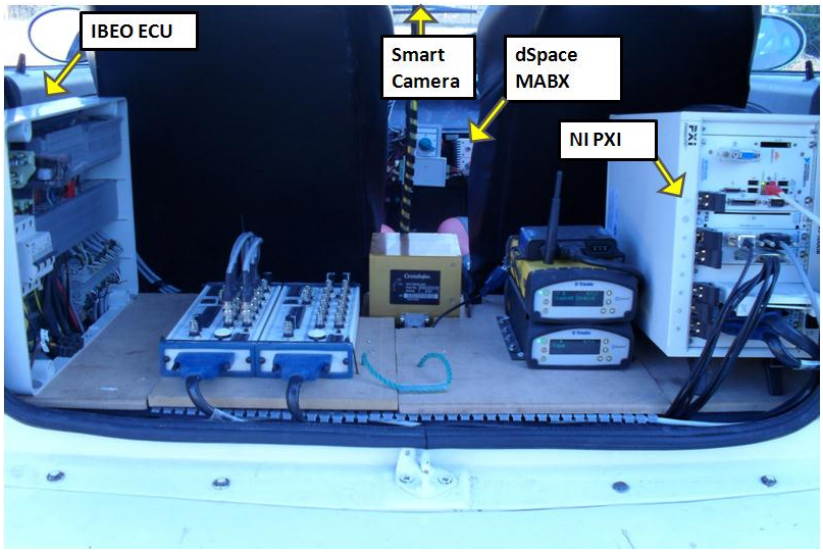


Figure 6.3: The processors that are used in Otonobil (trunk view).

Table 6.2: The processors used in Otonobil

Processor	Duty
NI PXI-8110RT processor	Sensor Read & Acquisition, Sensor Fusion
D-space Microautobox 1401	Low level control
SONY-XCI-SX100	Image Processing, Mapping
IBEO ECU	LIDAR Processing, Mapping

The communication between the sensors and the processors are maintained by several protocols. GPS receiver, IMU and digital compass use RS-232 protocol whereas the LIDARs and the camera use Ethernet protocol. The communication between the NI PXI and the Microautobox is governed by the Can bus. These communication protocols are illustrated in detail in figure 5.4.

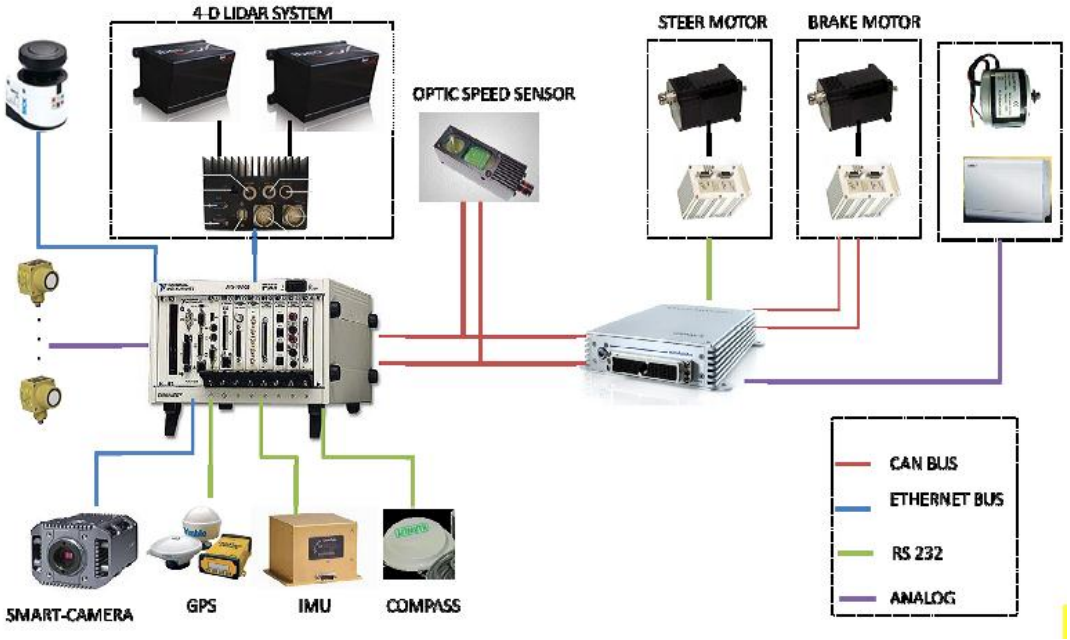


Figure 6.4: The communication in vehicle.

The mounting of the sensors are an important issue. Special care must be taken while mounting the sensors on the vehicle. The IMU is mounted at the center of gravity of the vehicle in order to minimize rotational accelerations affecting the accelerometer measurements. Also one important hint is aligning the axes of IMU body frame with the vehicle body frame to reduce the computational cost. The GPS antennas and the digital compass is mounted on a platform on a line to align the heading axis of GPS and the digital compass and also to align with the forward axis of the vehicle. The platform is made of carbon fiber since using ferromagnetic materials deteriorate the measurements of digital compass because of magnetic field. The optical velocity sensor is mounted on the left side of the vehicle and the sensor axis is aligned with the forward axis of the vehicle to measure the forward speed of the vehicle. The four layered LIDAR is mounted in front of the vehicle with a maximum line of sight,

giving information about the level surface and the objects up to 80m. There are four ultrasonic sensors on the right and left of the vehicle. These sensors are mostly used for parking applications and also mapping. Also one layered LIDAR is mounted on the back side of the vehicle. This configuration enables high level of perception of the environment for the vehicle.

6.3 The Software

In this work, the algorithms are implemented in NI Labview graphical programming language since the NI PXI hardware is compatible with this software. The acquisition, processing and logging of the sensors are performed in real-time module of Labview since these applications require deterministic real time performance that general purpose operating systems cannot guarantee [25]. By using real-time operating system, it is guaranteed that the calculations finish on time, without any time latency or any leads or lags. For this application, the acquisition and processing of information from the sensors and logging the information that is essential for algorithms should be performed in every sampling period of the related sensors.

6.3.1 Introduction to real-time concept

A real time system can be defined as a system which has to process and respond to externally generated stimuli within a finite time [26] .

A real-time operating system (RTOS) should be used for the applications where the timing is very important and where the calculations should finish on time in every period. Therefore the term real-time means that the responses are ensured to occur within a given time period [26]. The acquired data should be processed by software and it is used by algorithms in every period for this application hence using a DAQ device is not enough. A RTOS is required since the software and operating system should behave deterministically [26]. By using the Labview real-time module, it is possible for the users to combine Labview programming environment with real-time operating system to build deterministic real-time applications. The hardware and software architecture of Labview real-time module is given in the figure 6.5.

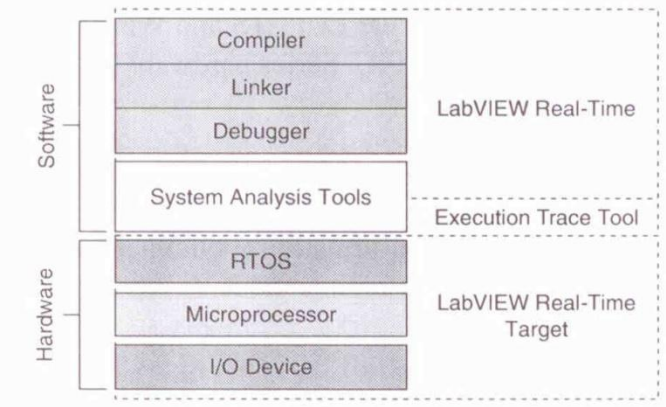


Figure 6.5 The Hardware and software architecture of RT system [25].

6.3.2 Real-time module components and architecture

There are four main components in a real-time system which consists of software and hardware components. For this application, the software components are real-time engine and Labview. The hardware components are the host computer and the real-time target. Labview and the RT module are installed in host computer to develop, deploy and run the code on RT target, a hardware that runs the RT engine, by using Labview. This engine provides deterministic real time performance since it runs on a RTOS and RT Target. Also virtual memory which can cause unpredictable performance is not used by RT Targets [25].

Real-time application architecture consists of two main parts. These are the host application and the target application. This basic architecture is illustrated in figure 6.6.

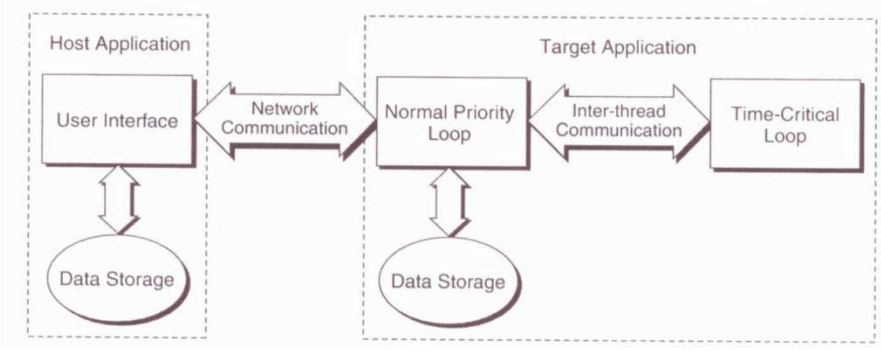


Figure 6.6 Real-time application architecture [25].

As seen in figure 6.6, the host application and the target application runs on different hardware. The host application runs on host computer and it provides a graphical user interface (GUI) by communicating the target application running on a RT Target via TCP/IP. A GUI is needed to initialize the software and monitor the running code that cannot be done in RT Target. On the other hand, target application which runs on RT Target hardware is divided into two tasks. The time critical loop consists of deterministic applications which are the tasks that should be completed on time in every period such as control loops, reading a data from a sensor and etc. These tasks must execute on time to assure accuracy since any lag can cause faults on the algorithms. The normal priority loop shown in figure 6.6 consists of the applications which are non-time critical codes such as the communication between the host and the target and also the data logging because an accurate time tag can easily identify the data when it is acquired.

6.3.3 Multi-threading and passing data between threads

A thread is refered as an independent flow of execution for an application. In multithreading methodology, the operating system brakes a single application into independent small tasks which can effectively execute in different execution system threads at the same time. Multithreaded applications increase the efficiency since any processor that is idle could perform any other threads that are ready to run. In this paper, the implemented software is divided into three threads which are the user interface thread, data acquisition and processing thread and data logging thread. To maintain determinisim, the code should be divided into time critical and non-time critical tasks. Also on multi-core symmetric multiprocessing (SMP) systems, Labview run tasks in paralel by assigning tasks to different cores[25]. Therefore, there are several advantages of multithreading for a real-time system.
Assigning priorities to the tasks are very important. Each thread is assigned different priority depending on they are time-critical or not. A higher priority should be assigned to time-critical tasks to ensure that they always finish on time. In order to assign the priority levels, timed-loop structure is used with different priorities to control the execution and timing of deterministic tasks. Each timed loop runs in its execution system that contains a single thread. In this work, each sensor has its own

timed loop for acquisition of the information. The sensors which has a higher sample rate is assigned a higher priority according to the ones which have lower sample rate in order to finish on time.

Also by using the timed loop structure, processors can be assigned manually for each timed loop in multi-core and multi-processor machines. Manually assigning of the processors could improve the determinisim since a dedicated processor is assigned to complete the task. However, this prevents Labview to assign an idle processor for other threads waiting to execute. So this decreases the effiency of the processors.

As it was mentioned in previous section, each sensor has a timed loop with different priorities in order to acquire and process the data within the specific sample period. After the data is processed, it should send to other threads on the RT Target such as the data logging thread which is a non-time critical thread or any deterministic algorithm. Passing data using wires does not work since each thread has a different timed loop. In order to communicate between the loops running on RT Target or host computer, shared variables are used. For the deterministic data transfer between loops running on RT Target, single process shared variables with FIFO structure is used. By enabling the real time FIFO of a shared variable, the data can be shared without affecting the determinism [26]. For the data transfer between the host and the target, network published shared variable is used.

6.3.4 The implemented software

The real-time software consists of two parts: The host computer side and the RT Target computer side. In the host computer side, you can develop the real-time code by using Labview and you can deploy the code into RT Target computer. Also host computer provides a GUI to monitor the acquisition and processing of the data from the sensors, to start, to end the process and also to initialize the system. The front panel of the program is illustrated in figure 6.7 As mentioned before, the host computer and the RT target communicates via Ethernet protocol. After initializing the system, the user can press the start button to run the real-time code in RT Target side.

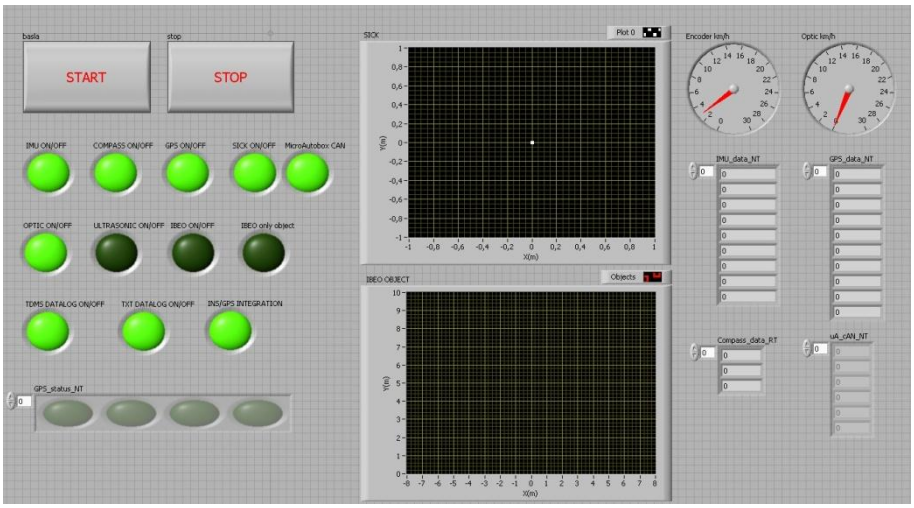


Figure 6.7 The GUI of the software (host computer side).

The real-time code has three stages illustrated in figure 6.8. In the initialization part, the specific settings for communication protocols are set i.e. for serial communication, the COM port and the baud rate is set according to the sensor also the buffer size is adjusted and the serial buffer is flushed before data is received. For TCP/IP communication only IP and port of the sensor are set. The acquisition part is the heart of the code which consists of different loops for each sensor to acquire the data and process it. In each data acquisition loop, the data is read from the communication port at constant rate and after the data is gathered, it should be processed i.e. dequantized or converted, to get useful information for data logging and to use in other algorithms. After this process, each data is tagged by using a tick clock timer and transferred to shared variable for logging process and other algorithms that uses this information. Timed loop structure is used in order to set the priority and the sampling rate of the loop. These loops run forever until the stop condition is satisfied, i.e. user clicks stop button in the host computer side. When this condition is satisfied, the loop terminates and the communication port is closed. Each data acquisition loop has a specific sample period depending on the sampling frequency of related sensor. Since all the data acquisition loops are time-critical, i.e. hardware I/O that has specified timing, they have a high priority than the data logging loop which is a non time-critical loop. In order to maintain the determinism, there should be no user interface like indicators and control inputs in the code running in RT Target side.

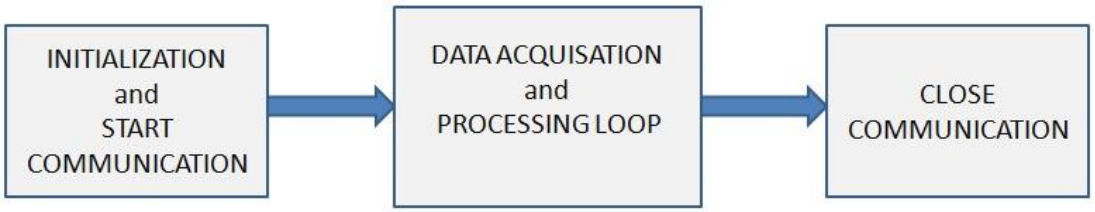


Figure 6.8 Real-time code (RT target side).

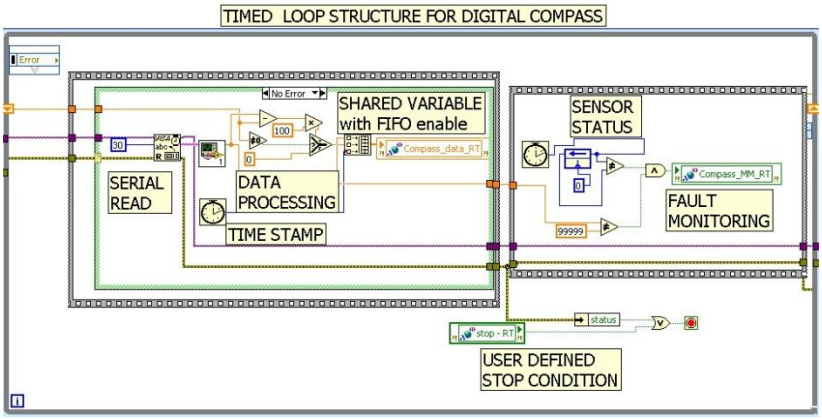


Figure 6.9 Data Acquisition loop for digital compass.

In the figure 6.9, the data acquisition and processing loop of digital compass is seen. After setting the COM port and baud rate in initialization part, the loop starts with reading data from serial port. After data is read, it is processed and time tagged and it is written to a shared variable in order to be used by another algorithms. The termination of the loop is controlled by a Boolean shared variable which is set as true when the stop button in host computer side is clicked by user. The stages performed in figure 6.9 is all the same for the other acquisition loops except some differences caused by different communication protocols.

7. THE TEST RESULTS

In order to test the performance of the algorithm that is implemented, real time data log of the whole sensors were performed. By this way, the algorithm could be tested in simulation using the real vehicle data. In section 7.1, the data set is introduced and in section 7.2 the results are given.

7.1 The Data Set

The software was tested in ITU Ayazaga campus in manual driving mode of Otonobil. The duration of the test was around 9 minutes with an average speed of 18 km/h and in this duration no sensor failures except GPS outages were occurred. Since the frequency of data logging software was 100Hz, 54000 data points for each sensor was logged in this test. The dataset consists of IMU, optical velocity sensor, motor encoder which are sampled at a frequency of 100 Hz where as GPS, digital compass, ultrasonic sensors are sampled at a frequency of 10 Hz. The data from these sensors first acquired then processed and logged in *tdms* format. The LIDARs are sampled at a rate of 12.5 Hz and their raw data is logged in *txt* format. The trajectory of the vehicle is illustrated in figure 7.1 using GPS measurements.

In the figure 7.2, 7.3 and 7.4, the latitude, longitude and height measurements of the vehicle from GPS are illustrated. There are six natural GPS outages occurred at the time t=61 ,158, 170, 208, 309 and 383 seconds. The longest outage took 4 seconds. It is a very short time to test the performance of the implemented integration system so artificial outages are created.

The GPS velocity in the navigation frame is calculated from the GPS position measurements since GPS receiver does not provide velocity measurements directly. GPS velocity vector is calculated using the equation in (3.39) and the figures are illustrated in 7.5, 7.6 and 7.7.

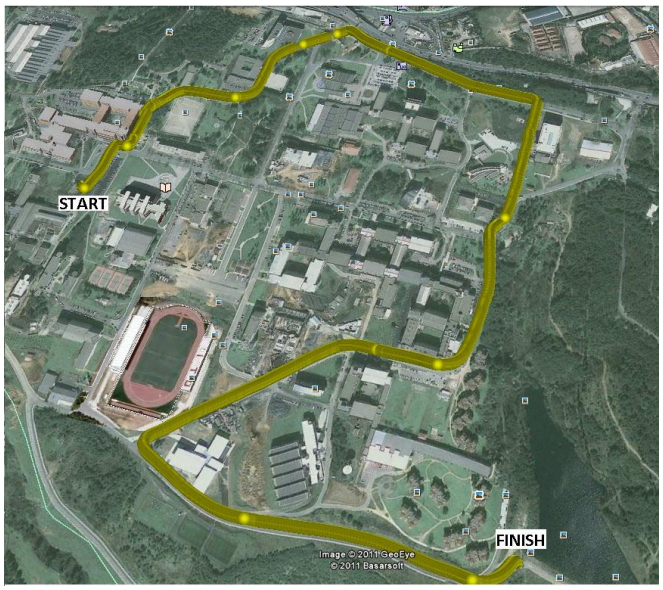


Figure 7.1 The trajectory of the vehicle during the test.

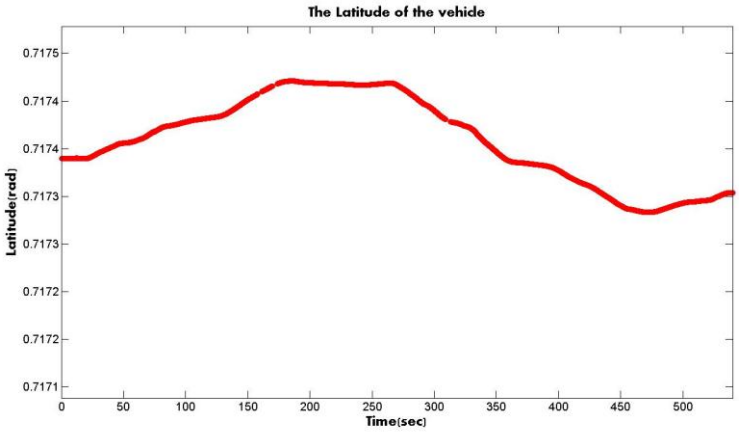


Figure 7.2 The Latitude measurement of the vehicle.

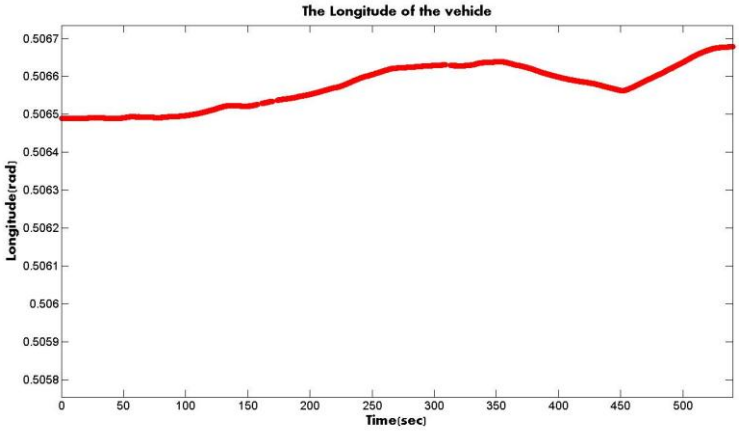


Figure 7.3 The Longitude measurement of the vehicle.

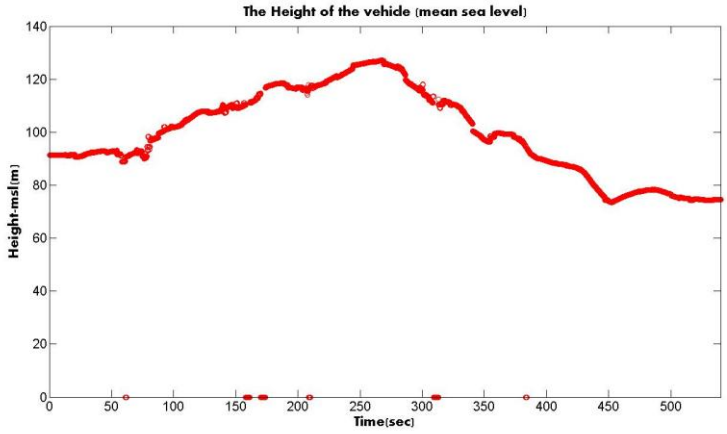


Figure 7.4 The mean sea level height measurement of the vehicle.

The measurements acquired from the sensors are given. The angular rates and the accelerations of the vehicle body frame are illustrated in figure 7.8 and 7.9.

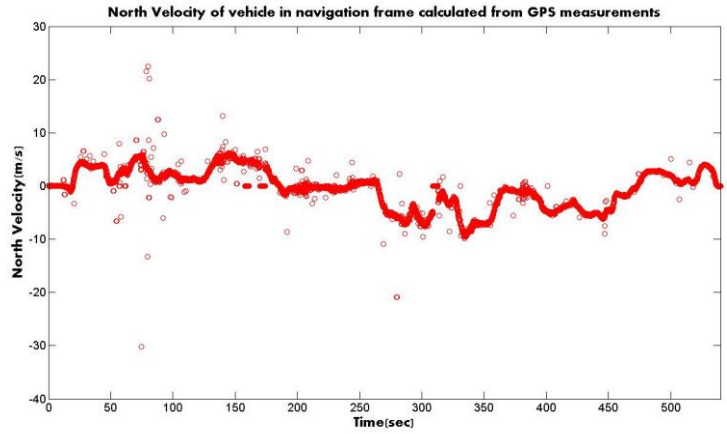


Figure 7.5 North Velocity of the vehicle.

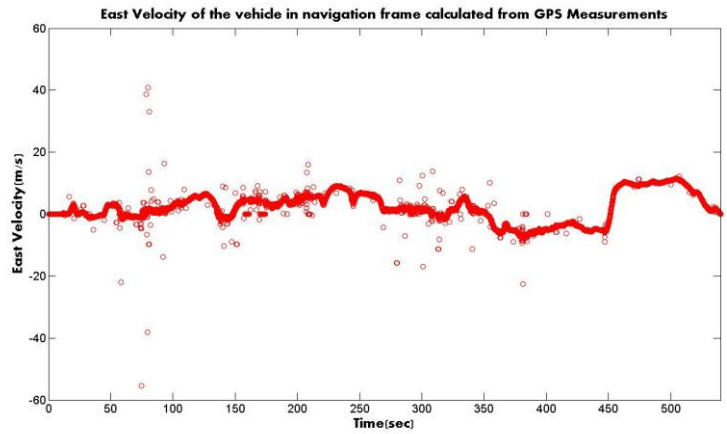


Figure 7.6 East Velocity of the vehicle.

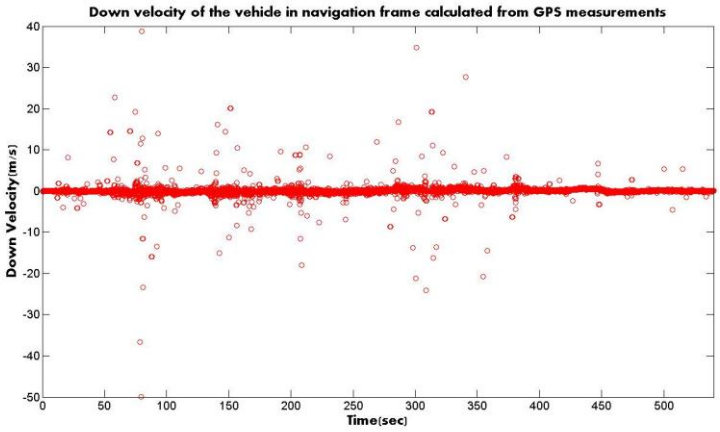


Figure 7.7 Down Velocity of the vehicle.

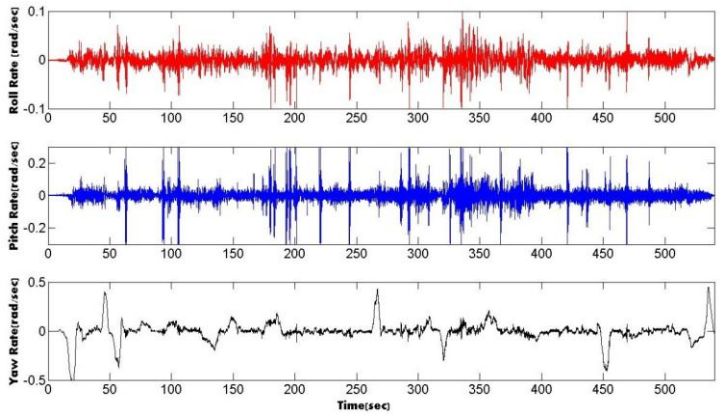


Figure 7.8 Angular rate measurements in three axes.

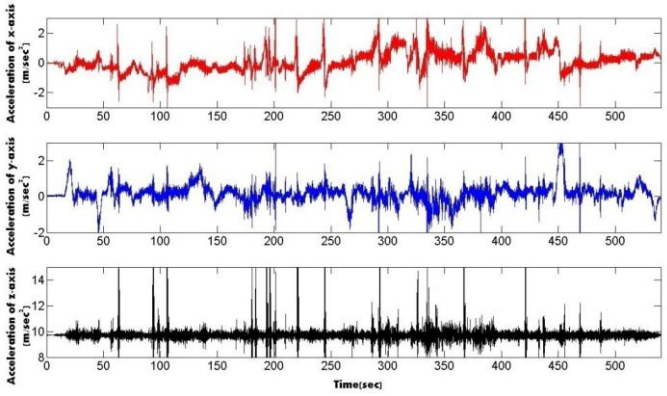


Figure 7.9 Linear acceleration measurements in three axes.

There are two heading measurement sources installed in the vehicle. By using DGPS with the help of two antennas, the GPS receiver outputs the heading from True North. The advantage of GPS heading over digital compass is that it is not affected

by the magnetic field in the environment and it gives quick responses. The disadvantage is that GPS signal could be easily lost when passing near a building or a road covered with tree branches. The digital compass is reliable in that case as there is no outage occurs but due to its working principle, it gives slow responses when the vehicle turns. The heading measurements are illustrated in figure 7.10.

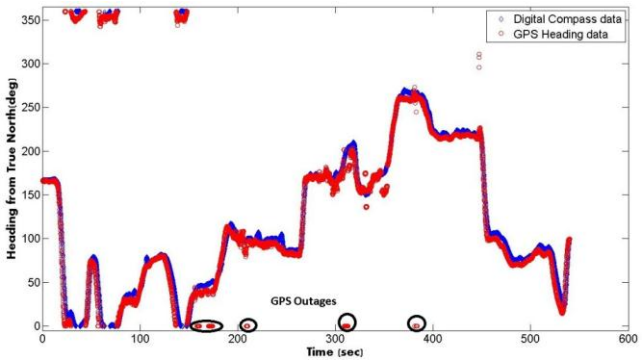


Figure 7.10 True North heading measurements.

There are two velocity sources in the vehicle. An optical velocity sensor is installed and also the motor encoder is used to calculate the speed of the vehicle. Using motor encoder is a cheap solution but it has noisy measurements in comparison with optical velocity sensor. The forward velocity of the vehicle is given in figure 7.11.

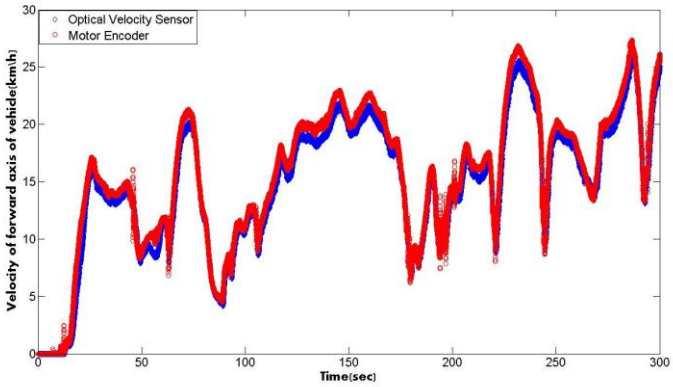


Figure 7.11 The velocity of the forward axis of the vehicle.

7.2 The Results

The data set was used in implemented INS/GPS integration algorithm as an input. Artificial GPS outages are created because the GPS outages that were occurred

naturally in the data set are at most four seconds. Long term GPS outages are needed in order to test the performance of the algorithm. Between time intervals, t={[30,40], [200,250], [350,380] and [450,500], artificial and long term GPS outages are created. The longest GPS outage is fifty seconds which is a very long time.

In section 7.2.1, the initialization of the algorithm is introduced.

7.2.1 The initialization of the algorithm

Since the aided INS/GPS integration system is a recursive algorithm, it should be initialized first and also the covariance matrices used in Kalman Filter should be calculated before the simulation.

The integration algorithm estimates the kinematic states of the vehicle which are the position, velocity and the Euler angles of the vehicle. So before running the code, these should be initialized.

The initialization of the position is maintained by calculating the average of GPS measurements for one minute when the vehicle is at rest. Since the simulation starts when the vehicle is at rest, the velocity vector is initialized to be zero. For initializing the heading angle, the average of digital compass measurements is used. The initialization of the roll and pitch angles are done by using the accelerometer measurements of IMU. The equation given in (3.40) is used for calculating the angles.

When the states of the Kalman filter is initialized, it is important to calculate the measurement noise covariance matrix (R_k) and the system noise covariance matrix (Q). The values of these matrices are actually the performance tuning parameters for the Kalman filter. These matrices are diagonal because cross correlations are assumed to be zero.

The system noise covariance matrix diagonal elements are the variances of the IMU measurements when the IMU is at rest. Since at rest, the IMU measurements are assumed to be the error measurements such as bias, random walk and random noise. The data is logged for one hour at 100 Hz and the variance values of three accelerometers and gyroscopes are found as in (7.1) and (7.2).

$$Q = \mathrm{diag}(\sigma^2_{a,x}, \sigma^2_{a,y}, \sigma^2_{a,z}, \sigma^2_{g,x}, \sigma^2_{g,y}, \sigma^2_{g,z}) \quad \textbf{(7.1)}$$

$$Q = \mathrm{diag}(6.8\text{x}10^{-7}, 6.13\text{x}10^{-7}, 7.34\text{x}10^{-7}, 9.71\text{x}10^{-8}, 3.34\text{x}10^{-8}, 7.31\text{x}10^{-8}) \quad \textbf{(7.2)}$$

The terms in (7.1) are the variances of accelerometers and the gyroscopes in three axes i.e. $\sigma^2_{a,x}$ is the variance of the accelerometer in x-axis and $\sigma^2_{g,x}$ is the variance of the gyroscope in x-axis.

The elements of measurement noise covariance matrix are calculated in the same way as system noise covariance matrix elements. The measurements of GPS is logged for an hour at 10 Hz when the vehicle is at rest and the variance values are found as in (7.3) and (7.4).

$$R_k = \mathrm{diag}(\sigma^2_{\varphi}, \sigma^2_{\lambda}, \sigma^2_{h}, \sigma^2_{vn}, \sigma^2_{ve}, \sigma^2_{vd}) \quad \textbf{(7.3)}$$

$$Q = \mathrm{diag}(2.147\text{x}10^{-18}, 1.212\text{x}10^{-18}, 3\text{x}10^{-5}, 0.05, 0.05, 0.05) \quad \textbf{(7.4)}$$

The terms in (7.3) are the variances of latitude, longitude, height and GPS velocity measurements.

7.2.2 Aided INS/GPS algorithm results

The algorithm is implemented in Labview and the dataset that is used in the algorithm is logged in real-time and introduced in previous section. In this part, the results of the aided INS/GPS integration algorithm are given and these results are compared with the standard INS/GPS algorithm results.

In the figures 7.12 and 7.13, the 2-D position of the vehicle in the navigation frame (north-east) is illustrated. The origin (0,0) is the start position of the vehicle and the finish position is (-260,910). During this simulation, artificial long term GPS outages are created in the time intervals t={[30,40], [200,250], [350,380] and [450,500]}. In the figure 7.12, the trajectory from start to finish is given. During GPS outages, only aided INS algorithm runs since there is no feedback from GPS. Any error in the position, velocity and orientation accumulates with time. This error is tried to be minimized by using aiding sensors.

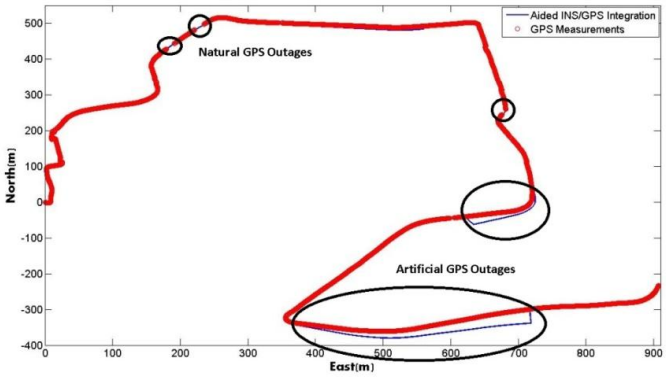


Figure 7.12 The position of the vehicle in local navigation frame.

In the figure 7.13, a zoomed view of the GPS outages between the time intervals [450-500] can be seen. The red dots are the GPS measurements that show the true trajectory. Actually 50 seconds of GPS outage is a big deal when the vehicle travels at 10 m/s average. In real time situations, it is advised to slow down or stop and wait till the GPS is online again. But in this simulation, the magnitude of error is considered to comprehend the performance of the algorithm. The errors in the position during outages are illustrated in the figure 7.14. The north error is 38 m and the east error is 2m at the end of GPS outage between [450,500] seconds.

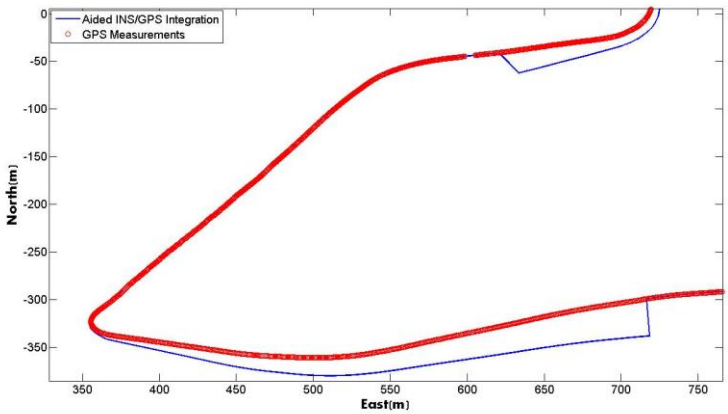


Figure 7.13 The position of the vehicle in local navigation frame (zoomed).

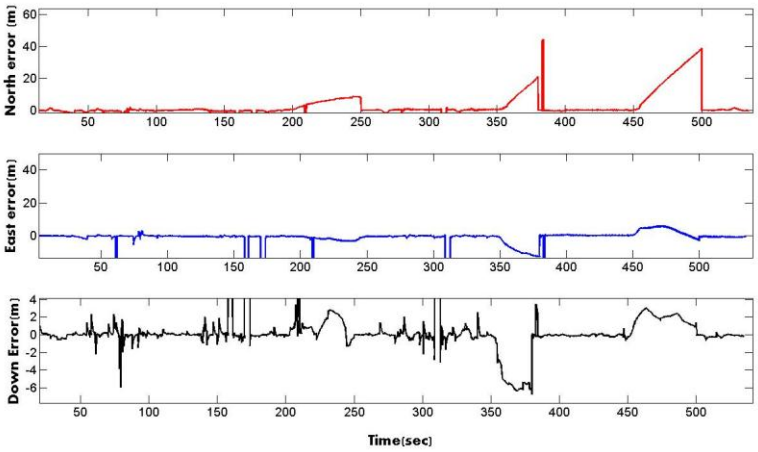


Figure 7.14 The position error in north,east and down directions.

The position errors are most caused by the errors in the velocity because the position is calculated by simply integration the velocity estimates. In this work, during the GPS outages, the main velocity measurement source is the motor encoder which outputs the motor shaft turning speed. By using the non-holonomic constraints mentioned in section 5.2.4, a velocity observation vector in navigation frame could be formed. But these velocity observations are highly dependent to heading vector since the speed of the forward axis of the vehicle is transformed to navigation frame velocity by using the heading angle. Also the measurement of the motor encoder is correct when the vehicle travels in forward direction at low speed because in high speed, slip of wheel can occur and when there is cornering, the measurement is not the forward speed anymore. Those drawbacks are tried to be eliminated by changing the measurement noise matrix of the velocity EKF by checking the steer angle and the speed of the vehicle. In figure 7.15-17, the velocity in the navigation frame is illustrated. For convenience, the true GPS velocity values during artificial GPS outages are shown in the figures. It can be seen from the figures 7.15 and 7.16 that the estimated velocity using aiding sensors are small in magnitude than the GPS velocity values. So that causes position errors in the North-East frame. If GPS velocity is taken as the reference velocity, the north velocity error during artificial GPS outage of 50 seconds is around 1m/s.

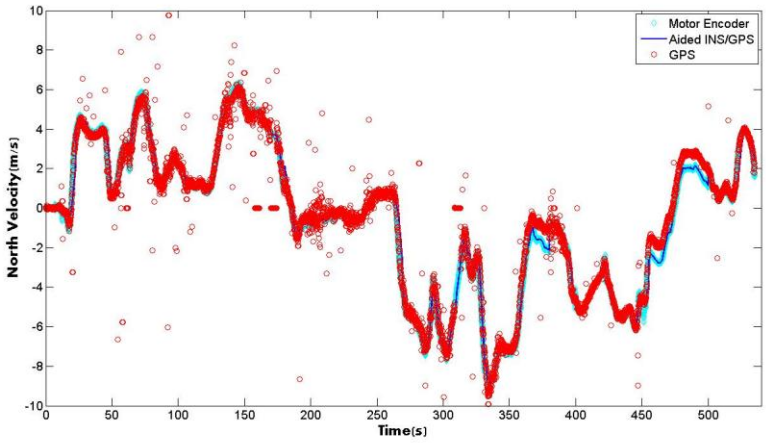


Figure 7.15 The velocity in the north axis.

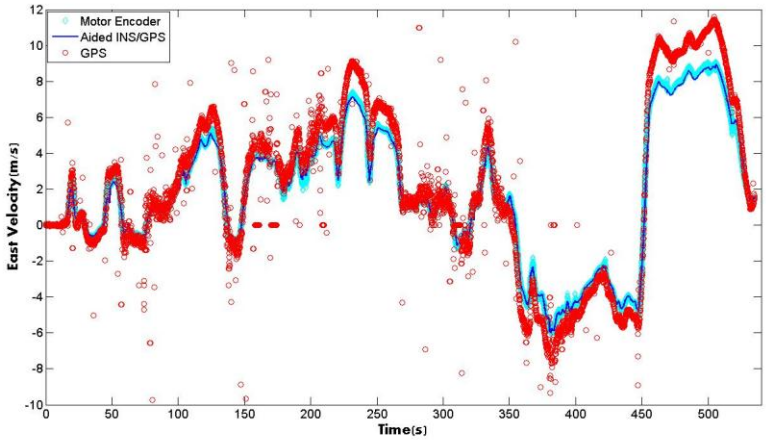


Figure 7.16 The velocity of the vehicle in east axis.

The figure given in 7.16, shows the east velocity of the vehicle during the simulation. Again the GPS east velocity during the artificial GPS outages are figured in order to show the error margin of the algorithm with respect to GPS velocity reference. The east velocity error during artificial GPS outage of 50 seconds is around 1.5 m/s.

In figure 7.17, the down velocity of the vehicle in local navigation frame is shown.

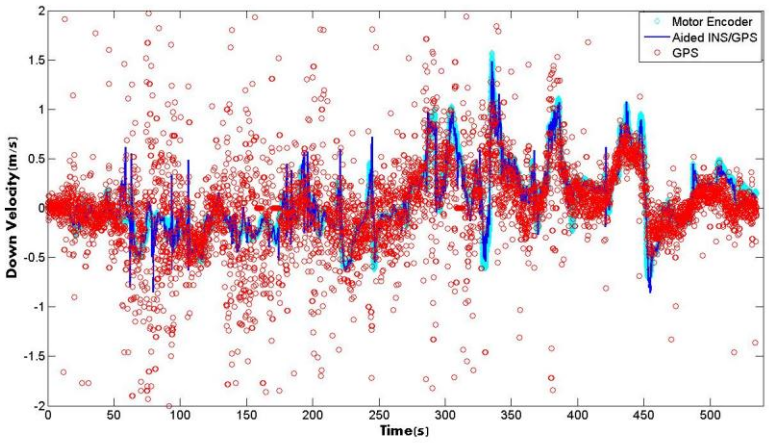


Figure 7.17 The velocity of the vehicle in down axis.

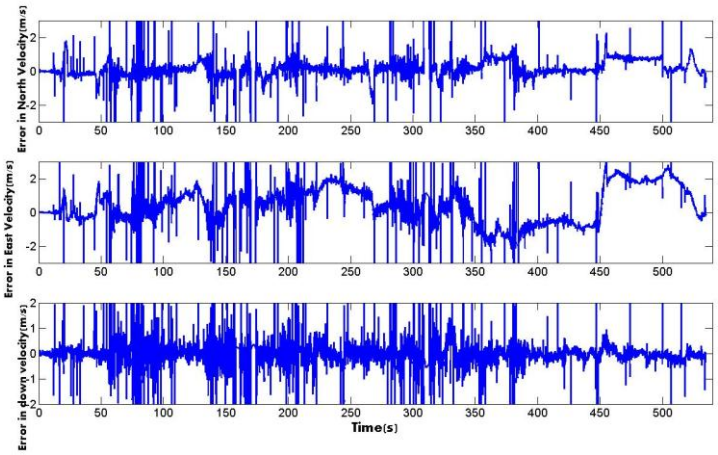


Figure 7.18 The velocity error in North-East-Down frame.

During GPS outages or GPS measurement intervals, the only velocity measurement source is using the motor encoder with non-holonomic constraints. There are two factors affecting the correctness of this calculation. One of the factors is that it is assumed that there is no cornering and slip. The vehicle travels only in its forward axis and travels at low speed. The second factor is the heading angle which is used to transform the speed in vehicle body frame to navigation frame. In order to estimate the Euler angles of the vehicle, digital compass is used as an aiding sensor. This algorithm is introduced in section 5.2.3.

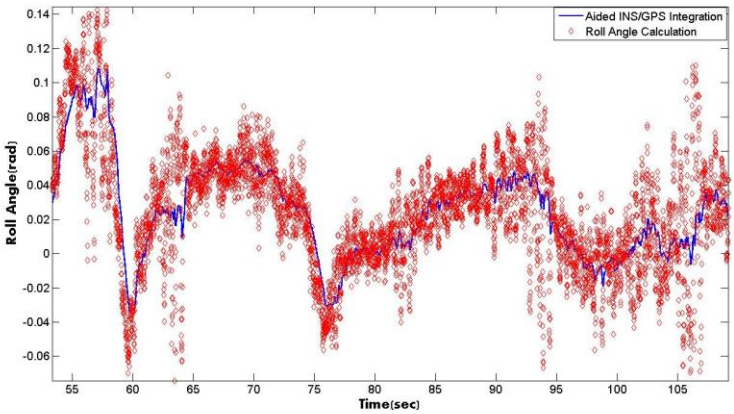


Figure 7.19 The roll angle of the vehicle with respect to ground.

In figure 7.19, the roll angle of the vehicle is shown. There is no measurement of roll angle in the vehicle but tilt sensors (accelerometers) of IMU is used with the formula given in 3.44 to represent the measurement of roll angle. But this measurement is only correct when there is no motional acceleration in the transversal axis of the body frame. The motional acceleration and the projection of gravity caused by the tilt angles (roll and pitch) cannot be distinguished since the IMU only senses the specific force which is the sum of those accelerations. Motor encoder velocity rate can be used for detecting acceleration in forward and transversal axes of the vehicle.

In figure 7.20, the pitch angle of the vehicle is shown. There is no measurement of pitch angle in the vehicle but tilt sensors (accelerometers) of IMU is used with the formula given in 3.43 to represent the measurement of pitch angle.

Usually the pitch and roll rates/angles are very small with respect to heading angle/rate. For navigation of land vehicles, it could be assumed that the navigation is in 2-D frame for simplicity. In that case, only heading angle is considered for transformation from vehicle body frame to navigation frame. So heading angle is the most important angle in land vehicle navigation. The estimated heading angle and the measurements are illustrated in figure 7.21.

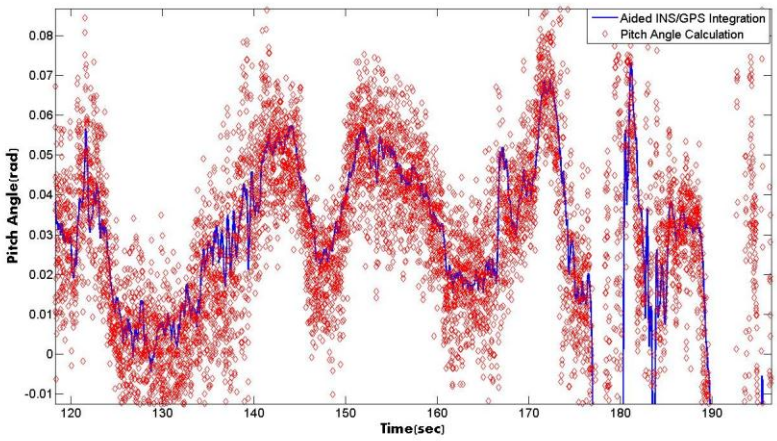


Figure 7.20 The Pitch angle of the vehicle with respect to ground.

The advantages and disadvantages of the heading sources are discussed in section 5.2.3.2. During the GPS outages, the only heading measurement source is the digital compass. But this measurement is not always used since a decision structure decides to use or not use the available measurement by comparing some kinematic properties. The decisions that are made by the decision structure is given in figure 7.22.

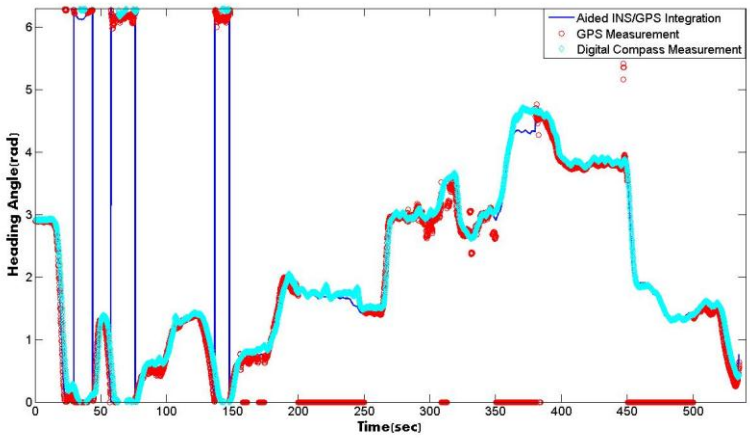


Figure 7.21 The heading angle of the vehicle with respect to True North.

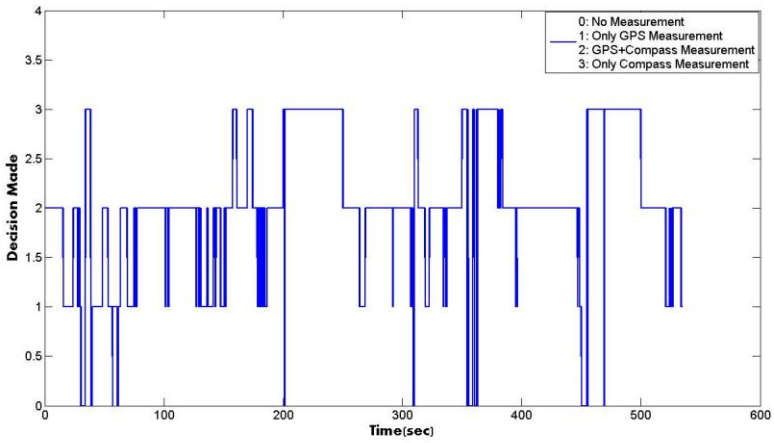


Figure 7.22 The decisions for heading source that is made during the simulations.

7.2.3 The comparison of the results with non-aiding algorithms

In this section, the results are compared with the results of other algorithms such as INS/GPS integration without aid, INS/GPS integration with velocity aid and INS/GPS integration with orientation aid. All the parameters of the Kalman Filters are the same in all algorithms. This results indicate the importance of both orientation and velocity aiding.

In the figure 7.23, the position result of both velocity and orientation aided algorithm and no aided algorithm is illustrated. It is seen that during long GPS outages, the error in the magnitude of north and east position are accumulating and the error is unbounded depending on time.

In the figure 7.24, the position results of partial aided algorithms are compared with aided algorithm is given. The velocity aiding is very important for position results during GPS outages since the position is calculated as the integration of the velocity. Without velocity aiding, the error in the position of the vehicle is growing depending on time. For a GPS outage of 50 seconds, the magnitude of error could be hundreds of meter depending on the velocity of the vehicle. Therefore it is suggested to slow down or stop during long GPS outages.

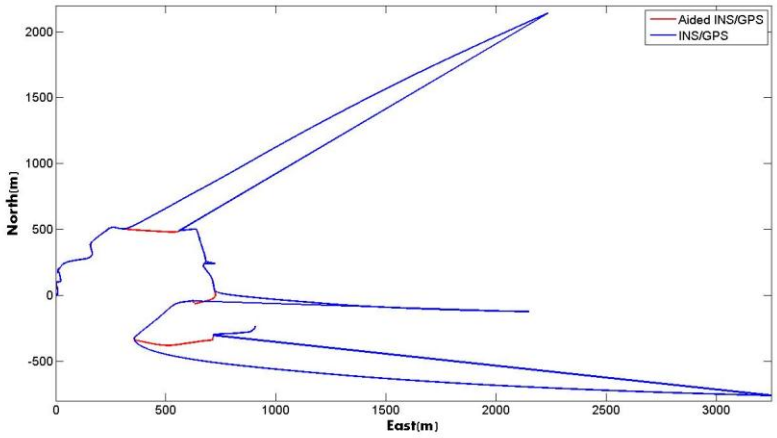


Figure 7.23 The position results of aided and non-aided algorithm.

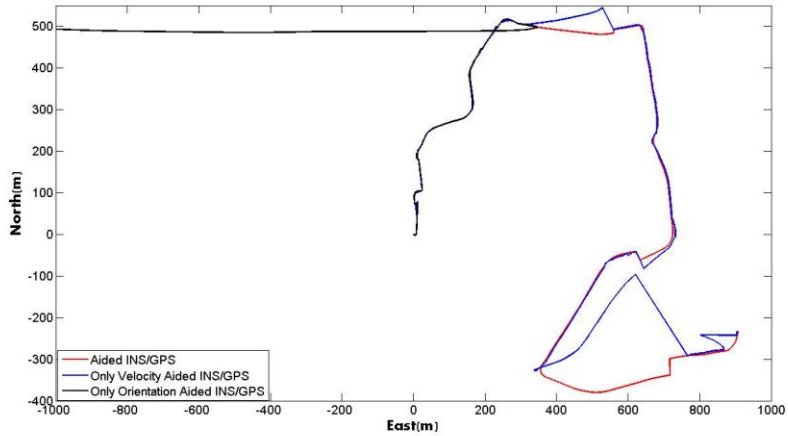


Figure 7.24 The position results of aided and partial-aided algorithm.

In the figure 7.25-27, the velocity results are given. There are four different algorithms that are tested. One of them is full-aided i.e. orientation and velocity, INS/GPS algorithm which is the red line in the figures. The blue line is the only velocity aided INS/GPS algorithm. There is no aiding sensor like digital compass, GPS heading receiver and Kalman Filters for attitude and heading estimation in this one. The black line represents the only orientation aided INS/GPS algorithm which does not use the motor encoder as a velocity source.

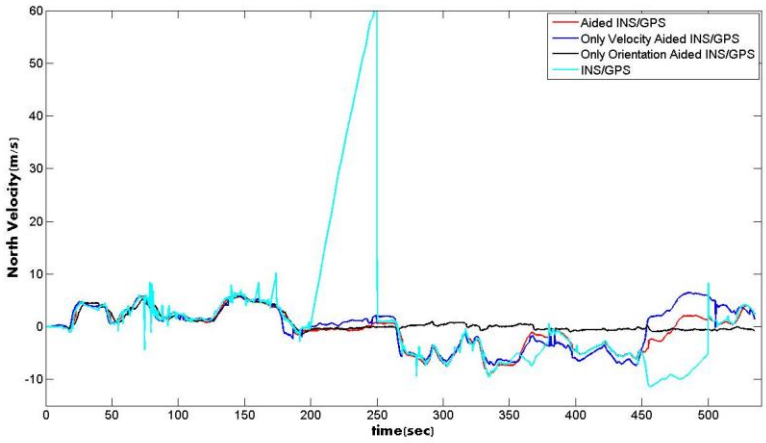
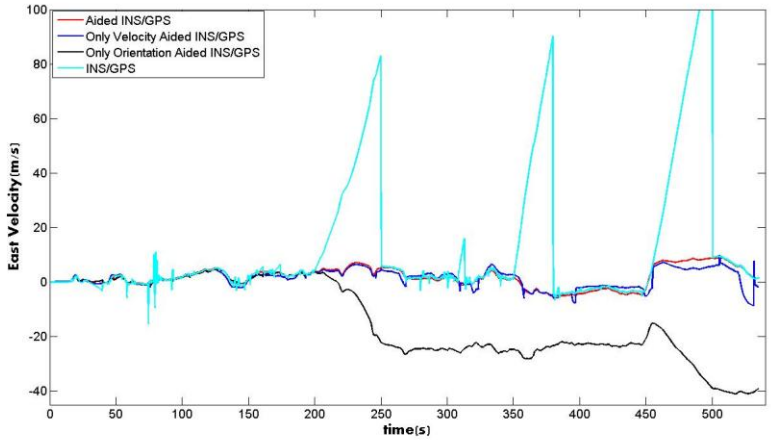


Figure 7.25 The north velocity results.



.

Figure 7.26 The east velocity results.

From the velocity results, it is understood that during GPS outages, using an aiding sensor like a wheel encoder, motor encoder or etc., is very important for keeping the velocity error minimum. But if accurate results are needed using a velocity sensor, the orientation should be estimated accurately. The transformation in vehicle body frame to navigation frame is important because these aiding sensors measure their quantities in their sensitive axes which are aligned with the vehicle body frame.

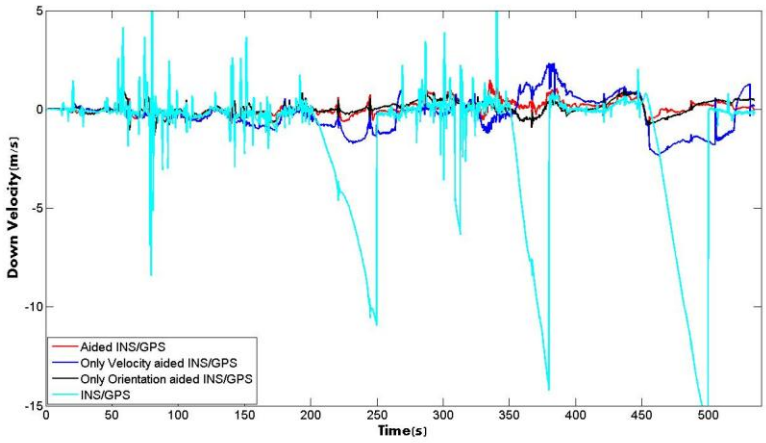


Figure 7.27 The down velocity results.

In the figures 7.28-30, the Euler angle estimation results are seen. The results of orientation aided INS/GPS algorithm are compared with the results of standard INS/GPS algorithm where there is no orientation aiding sensor. Without aiding sensor, the angular rate measurements of IMU are simply integrated to calculate the transformation matrix and the Euler angles. Since integration is an accumulative process, any error in the previous step accumulates to current stage. So for a long time without a reset system or an aiding sensor, the error grows without bounds.

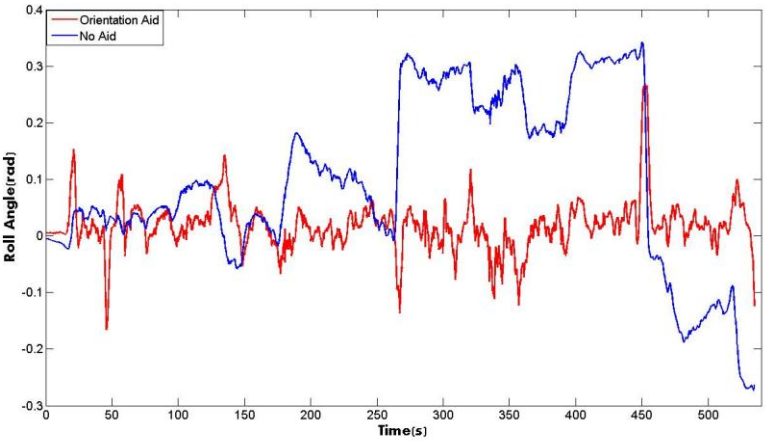


Figure 7.28 The roll angle results.

The acceleration measurements of IMU are used as measurements for estimating the roll and pitch angle. By using these measurements, the errors in roll and pitch angle

estimations are bounded. The results of roll angle are given in figure 7.28 and the result of pitch angle is given in figure 7.29.

The heading is the most important angle because the roll and pitch angles are small relative to heading angle and their rates are smaller than heading rate. The transformation between vehicle body frame and navigation frame mostly depends on heading angle in land vehicle applications.

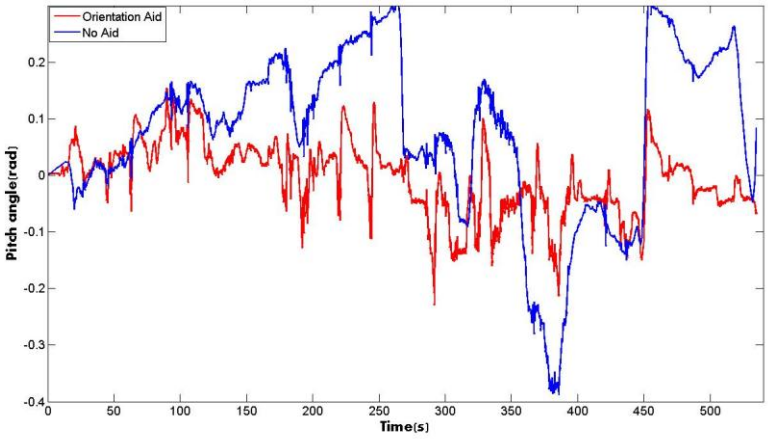


Figure 7.29 The pitch angle results.

For heading estimation, GPS heading receiver and a digital compass is used. Their measurements are fused using a Kalman Filter. During long time GPS outages, the accuracy in heading estimation important, because the accuracy in velocity aiding depends on accurate heading estimates.

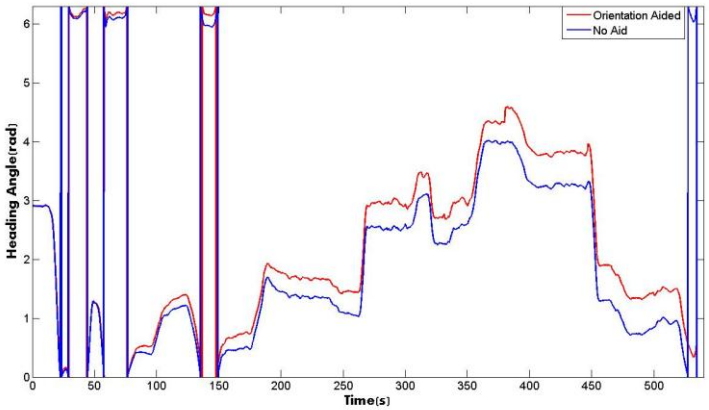


Figure 7.30 The heading angle results.

8. CONCLUSION AND FUTURE WORKS

Research and the studies about the unmanned ground vehicles (UGV) is attracting a lot of interest recently. These vehicles are equipped with various types of sensors and actuators to identify the environment and perform necessary maneuvers that are calculated by intelligent algorithms. One of the most important capabilities of UGV is state estimation. Those states are named as navigation states which are the kinematic states of the vehicles such as position, velocity and orientation.

A common way to estimate these states is the INS/GPS integration system. Only GPS system cannot be used because of signal blockages, interference and multipath, GPS reliability is degraded. During these outages or interferences the only INS functions and outputs the navigation states. But the reliability and accuracy of INS degrades according to time because INS is a dead-reckoning system so that any error in the previous step will be carried to current state. Therefore after a specific time, the error will be unbounded and grows with time.

In order to improve the accuracy of INS during GPS outages, some aiding sensor can be used. This is the main purpose of this thesis. The objective is to improve the accuracy of INS/GPS integration system, during GPS outages or not, with the help of aiding sensors like a digital compass and a motor encoder. In order to implement an estimation algorithm in INS structure, three distinct Kalman filters are used. Two Kalman filters are used for estimating the transformation matrix using DCM method. First one is used to estimate the third row of transformation matrix using the acceleration measurements. The second Kalman filter which is in cascade with the first one is used to estimate the second row of the transformation matrix using the estimates of the first Kalman filter and the heading measurements provided from a digital compass and GPS heading receiver.

After orientation update, velocity estimation is done by using an EKF structure since the dynamic model of the system is nonlinear. The motor encoder is used as a

forward axis speed measurement and two non-holonomic constraints are used because of the travelling of a vehicle in Earth surface. Using the transformation matrix, the velocity in navigation frame is calculated and used as the measurements in EKF algorithm. The position is calculated as using 2^{nd} order Runge-Kutta integration method. After aided INS stage is calculated and the measurements of GPS is arrived, the main EKF structure fuses the information from both systems and calculates the position and velocity states.

In the practical part, this algorithm is implemented in Labview and in order to test the performance algorithm, a real-time data acquisition and logging system is implemented. The algorithm is first tested with the synchronized and real-time data and after the algorithm is successful, it is deployed in the real-time system running in UGV.

The results show that the implemented algorithm is much more accurate than the standard INS/GPS system during the GPS outages. By the help of aiding sensors, the errors of navigation states are minimized according to the standard integration algorithm.

For future work, the same system can be used with low-cost sensors with additional error dynamics in the EKF. The bias errors and misalignment errors should be estimated and a calibration algorithm should be implemented. Besides the quality of the sensors, the INS/GPS algorithm can be implemented using quaternions and unscented Kalman filter. The UKF has some advantages over EKF in nonlinear estimation. Also a fuzzy-logic can be implemented to decide the measurement quality of the sensors i.e. GPS position measurement depends on DOP value and number of satellites and etc. a dynamic measurement covariance matrix can be found by checking the factors that decrease the quality of measurement.

REFERENCES

[1] **Farrell, Jay A.,** 2008. *Aided Navigation GPS with high rate sensors*, Mc Graw Hill, USA.

[2] **Groves, Paul D.,** 2008. *Principles of GNSS, Inertial and Multisensor Integrated Navigation Systems,* Artech House, Boston, USA.

[3] **Raol, Jitendra R.,** 2010. *Multi-sensor Data Fusion with MATLAB,* CRC Press, USA.

[4] **Sukkarieh, S.,** 2000. *Low Cost, High Integrity, Aided Inertial Navigation Systems for Autonomous Land Vehicles*, Unpublished Doctoral Dissertation, The University of Sydney Australian Center for Field Robotics Department of Mechanical and Mechatronic Engineering, Sydney, Australia.

[5] **Schumacher, A.,** 2006. *Integration of a GPS aided Strapdown Inertial navigation System for Land Vehicles*, Unpublished Master of Science Thesis, Kungliga Tekniska Högskolan Electrical Engineering Department, Stockholm, Sweden.

[6] **Mayhew, David M.** 1999. *Multi-rate Sensor Fusion for GPS Navigation Using Kalman Filtering,* Unpublished Master of Science Thesis, Virginia State University Electrical Engineering Department, Blacksbury, Virginia, USA

[7] **Shin, Eun-Hwan** 2001. *Accuracy Improvement of Low Cost INS/GPS for Land Applications,* Unpublished Master of Science Thesis, The University of Calgary, The Department of Geomatics Engineering, Calgary, Alberta, Canada.

[8] **Kong X.,** 2004. *INS Algorithm using Quaternion Model for Low Cost IMU,* Robotics and Autonomous Systems Vol. **46** pp.221-246

[9] **Nguyen, Ho.Q.P., Kang, H.J., Suh, Y.S., Ro, Y.S.,** 2009. *INS/GPS Integration System with DCM based Orientation Measurement,* ICIC 2009, LNCS 5754 pp.856-869

[10] **Zhang P., Gu, J. Milios, Evangelos E., Huynh, P.,** 2005. *Navigation with IMU/GPS/Digital Compass with Unscented Kalman Filter,* Proceedings of the IEEE International Conference on Mechatronics & Automation, Niagara Falls, Canada, July 2005.

[11] **Crassidis, John L.,** 2006. *Sigma-Point Kalman Filtering for Integrated GPS and Inertial Navigation*, IEEE Transactions on Aerospace and Electronic Systems, Vol.**42** pp.750-756

[12] **Hossein, Tehrani N.N, Mita, S., Long, H.,**, 2010. *Multi-Sensor Data Fusion for Autonomous Vehicle Navigation through Adaptive Particle Filter*, IEEE Intelligent Vehicles Symposium University of California, San Diego, CA, USA, June 21-24,2010

[13] **Schultz, Casper E.,** 2006. *INS and GPS Integration,* Unpublished Master of Science Thesis, Technical University of Denmark, Department of Informatics and Mathematical Modelling, Kopenhag, Denmark.

[14] **Url-1** < *http://en.wikipedia.org/wiki/GLONASS/*>, accessed at 05.05.2011.

[15] **Url-2** <*http://en.wikipedia.org/wiki/Galileo_positioning_system/*>, accessed at 05.05.2011.

[16] **Url-3** <*http://en.wikipedia.org/wiki/Compass_navigation_system/*>, accessed at 05.05.2011.

[17] **Nguyen, Ho.Q.P., Kang, H.J., Suh, Y.S., Ro, Y.S.,** 2009. *A DCM Based Orientation Estimation Algorithm with an Inertial Measurement Unit and a Magnetic Compass,* Journal of Universal Computer Science , vol 15,no.4(2009) pp 859-876

[18] **Van Der Merwe, R., Wan, Eric A.,** 2004. *Sigma-Point Kalman Filters for Integrated Navigation,* Proceedings of the 60th Annual Meeting of the Institute of Navigation (ION) pp.641-654

[19] **Wang, J.H., Yao,Y.,** 2005. *Multi-sensor Data Fusion for Land Vehicle Attitude Estimation using a Fuzzy Expert System,* Data Science Journal, Volume 4, pp 127-139 , 28 November 2005

[20] Crossbow *VG700AA/AB User's Manual*

[21] **Cheng, J., Lu, Y., Thomas, E.R., Farrell, J.A.,** 2007. *Data Fusion via Kalman Filter : GPS and INS* **,**Autonomous Mobile Robots Chapter 3 pp 99-147

[22] **Mutambara, A.G.O.,** 1998. *Decentralized Estimation and Control for Multisensor Systems,* CRC Press USA.

[23] **Thrun,S., Burgard,W., Fox,D.,** 2005. *Probabilistic Robotics*, The MIT Press, Cambridge ,Massachusetts

[24] **Ristic,B., Arulampalam,S., Gordon,N.** 2004. *Beyond the Kalman Filter Particle filters for tracking applications,* Artech House, Boston

[25] National Instruments, *Labview Real-Time Application Development Course manual*, February 2009 Edition

[26] **Burns, A., Wellings**, **A.,** 2009. *Real-Time Systems and Programming Languages*, Addison Wesley c2009.

CURRICULUM VITAE

Candidate's full name: Ziya ERCAN

Place and date of birth: Kuşadası, 25.02.1986

Permanent Address: Feriköy Mah. Bozkurt Cad. Sema Apt. No:75/4 Şişli/İstanbul

Universities Attended: TC Anadolu University, Electrical and Electronics Engineering (Undergraduate 2004-2008)