MULTISENSOR DEAD RECKONING NAVIGATION ON A TRACKED VEHICLE USING KALMAN FILTER

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

BY

SERDAR KIRIMLIOĞLU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

SEPTEMBER 2012

Approval of the thesis:

MULTISENSOR DEAD RECKONING NAVIGATION ON A TRACKED VEHICLE USING KALMAN FILTER

submitted by **SERDAR KIRIMLIOĞLU** in partial fulfillment of the requirements for the degree of **Master of Science in Mechanical Engineering Department, Middle East Technical University** by,

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

Prof. Dr. Süha Oral
Head of Department, **Mechanical Engineering** _____

Asst. Prof. Dr. E. İlhan Konukseven
Supervisor, **Mechanical Engineering Dept., METU** _____

Prof. Dr. M. Kemal Özgören
Co-Supervisor, **Mechanical Engineering Dept., METU** _____

Examining Committee Members:

Prof. Dr. Tuna Balkan
Mechanical Engineering Dept., METU _____

Asst.Prof. Dr. E. İlhan Konukseven
Mechanical Engineering Dept., METU _____

Prof. Dr. M. Kemal Özgören
Mechanical Engineering Dept., METU _____

Prof. Dr. Ozan Tekinalp
Aerospace Engineering Dept., METU _____

Dr. Selçuk Himmetoğlu
Mechanical Engineering Dept., Hacettepe University _____

Date: 11.09.2012

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

Name, Last name : Serdar Kırımlıoğlu

Signature :

ABSTRACT

MULTISENSOR DEAD RECKONING NAVIGATION ON A TRACKED VEHICLE USING KALMAN FILTER

Kırımlıoğlu, Serdar
M.S., Department of Mechanical Engineering
Supervisor : Asst. Prof. Dr. E. İlhan Konukseven
Co-Supervisor: Prof. Dr. M. Kemal Özgören

September 2012, 120 pages

The aim of this thesis is to write a multisensor navigation algorithm and to design a test setup. After doing these, test the algorithm by using the test setup. In navigation, dead reckoning is a procedure to calculate the position from initial position with some measured inputs. These measurements do not include absolute position data. Using only an inertial measurement unit is an example for dead reckoning navigation. Calculating position and velocity with the inertial measurement unit is highly erroneous because, this calculation requires integration of acceleration data. Integration means accumulation of errors as time goes. For example, a constant acceleration error of $0.1\ m/s^2$ on $1\ m/s^2$ of acceleration will lead to 10% of position error in only 5 seconds. In addition to this, wrong calculation of attitude is going to blow the accumulated position errors. However, solving the navigation equations while knowing the initial position and the IMU readings is possible, the IMU is not used solely in practice. In literature, there are studies about this topic and in these studies; some other sensors aid the navigation calculations. The aiding or fusion of sensors is accomplished via Kalman filter.

In this thesis, a navigation algorithm and a sensor fusion algorithm were written. The sensor fusion algorithm is based on estimation of IMU errors by use of a Kalman filter. The design of Kalman filter is possible after deriving the mathematical model of error propagation of mechanization equations.

For the sensor fusion, an IMU, two incremental encoders and a digital compass were utilized. The digital compass outputs the orientation data directly (without integration). In order to find the position, encoder data is calculated in dead reckoning sense. The sensor triplet aids the IMU which calculates position data by integrations. In order to mount these four sensors, an unmanned tracked vehicle prototype was manufactured. For data acquisition, an xPC–Target system was set.

After planning the test procedure, the tests were performed. In the tests, different paths for different sensor fusion algorithms were experimented. The results were recorded in a computer and a number of figures were plotted in order to analyze the results. The results illustrate the benefit of sensor fusion and how much feedback sensor fusion is better than feed forward sensor fusion.

Keywords: Navigation, dead reckoning, inertial navigation system, Kalman filter, sensor fusion, unmanned vehicle, tracked vehicle, mechanization equations

ÖZ

KALMAN SÜZGECİ KULLANARAK BİR PALETLİ ARAÇ ÜZERİNDE ÇOK ALGILAYICILI KÖR ADIMI SEYRÜSEFER

Kırımlıoğlu, Serdar
Yüksek Lisans, Makine Mühendisliği Bölümü
Tez Yöneticisi : Yar. Doç. Dr. E. İlhan Konukseven
Ortak Tez Yöneticisi: Prof. Dr. M. Kemal Özgören

Eylül 2012, 120 sayfa

Bu tezde amaç, çok algılayıcılı seyrüsefer algoritması yazmak ve bu algoritmayı test etmek için bir test sistemi tertip etmektir. Seyrüseferde, ölü adımı veya havacılıkta kullanılan adıyla ölü uçuşu, bazı ölçümler yapmak kaydıyla ilk konum verilerinden son konumu hesaplamaya yarayan bir yöntemdir. Bu yöntemde mutlak konum girdileri kullanılmaz. Ataletsel ölçüm birimi (AÖB) kullanarak pozisyon ve hız hesabı yapmak kör adımına örnek olarak gösterilebilir. AÖB kullanılarak bu şekilde yapılan hesap, yüksek miktarda hatayı beraberinde getirir. Çünkü hesap, ivme verilerinin integralini gerektirir. İntegral almak, zamana bağlı olarak hata birikmesi demektir. Örneğin, $1\ m/sn^2$ ivme üzerindeki $0.1\ m/sn^2$ sabit ivme hatası 5 saniyede $\%10$ konum hatasına sebebiyet verecektir. Bununla beraber, yönelimin hatalı hesaplanması, birikmiş konum hatalarının daha da artmasına neden olacaktır. Her ne kadar, ilk konumun bilindiği varsayımıyla, seyrüsefer denklemlerinin çözümü AÖB çıktılarını kullanarak mümkün olsa da, AÖB pratikte tek başına kullanılmaz. Literatürde bu konuyla ilgili çalışmalar mevcuttur ve bu çalışmalarda AÖB'ye başka algılayıcıların destek olduğu

görülmektedir. Algılayıcı bileşimi Kalman süzgecini kullanarak mümkün olmaktadır.

Bu tezde bir seyrüsefer algoritması ve bir algılayıcı bileşim algoritması yazılmıştır. Algılayıcı bileşimi, Kalman süzgeci kullanarak AÖB hatalarının kestirimi temeline dayanmaktadır. Kalman süzgecinin tasarımı sistemin hata büyümesinin matematik modelinin çıkarılması ile mümkündür.

Algılayıcı bileşimi için bir AÖB, 2 artırımsal enkoder ve bir sayısal pusula kullanılmıştır. Sayısal pusula yönelim verisini doğrudan yani integral almadan verebilmektedir. Konumu bulabilmek için enkoder verileri ölü adım metodu temel alınarak kullanılır. 3 algılayıcı, çıktı verileri integraller alınarak konuma ulaşılan AÖB'ye destek olacak şekilde kullanılır. 4 algılayıcı bir insansız paletli araç prototipine monte edilmiştir. Veri toplamak için ise xPC–Target sistemi kullanılmıştır.

Test adımlarını planladıktan sonra testler icra edilmiştir. Testlerde sensör bileşim algoritmaları farklı hareket yörüngeleri için denenmiştir. Tüm çıktılar kayıt altına alınmış ve bu çıktılardan analizlere yardımcı olacak bazı grafikler çizdirilmiştir. Sonuçlarda algılayıcı bileşiminin faydası incelenmiştir ve ileri ile geri besleme algılayıcı bileşimi yöntemleri karşılaştırılmıştır.

Anahtar Kelimeler: Seyrüsefer, ölü adımı, ataletsel seyir sistemi, Kalman süzgeci, algılayıcı bileşimi, algılayıcı kaynaştırma, insansız araç, paletli araç, seyrüsefer denklemleri

Dedicated to my brother…

ACKNOWLEDGEMENTS

First, I would like to express my gratitude to my supervisor Asst. Prof. Dr. E. İlhan Konukseven and my co-supervisor Prof. Dr. M. Kemal Özgören for their guidance and support for my thesis studies.

I would like to express my sincere thanks to Ali Emrah Yarar for his guidance about dsPIC applications and serial data transfer and also for his friendship.

I am grateful to all the staff of the Department of Mechanical Engineering. I must thank my lab mate Gökhan Bayar for his supports.

Finally, I want to express my great thanks to my family for their all-round supports during my life.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

Symbol	Definition
ei	Earth centered and inertial reference frame
ef	Earth centered and fixed reference frame
n	Navigation reference frame
b	Body reference frame
$\hat{R}(n,\theta)$	Rotation matrix for θ much rotation about $\vec{n}$
$SSF(\quad)$	Skew symmetric form of
$SSF^{-1}(\quad)$	Inverse skew symmetric form of
$\vec{u}_1^{(0)}$	First basis vector of reference frame 0
$\hat{A}^T$	Transpose of matrix A
$\bar{n}$	Column matrix representation of $\vec{n}$
$\tilde{q}$	Skew symmetric form of $\bar{q}$
$\overrightarrow{OB}$	Position vector starting from point O and ending at point B
$\hat{C}^{(b,a)}$	Direction cosine matrix converting matrix expression in $\mathcal{F}_a$ into expression in $\mathcal{F}_b$
$\hat{C}_{ort}$	Orthogonalized direction cosine matrix
$\hat{C}_{nor}$	Normalized direction cosine matrix
D_a	Differentiation with respect to $\mathcal{F}_a$
$\vec{\omega}_{b/a}$	Angular velocity vector of $\mathcal{F}_b$ with respect to $\mathcal{F}_a$
$\vec{h}$	Acceleration vector caused by the motion of vehicle
$\vec{g}$	Plumb bob gravity
$\mathcal{F}_a$	Reference frame named as A
Ω	Earth's rotation rate
ψ	Longitude
θ	Latitude
y	Height above sea level
R_0	Earth's length of semi major axis
R_N	Meridian radius of curvature

R_E	Transverse radius of curvature
e	Eccentricity of the Earth
ω_1	Angular rate data for the first axis of the INS
a_1	Acceleration data for the first axis of the INS
$\|\|R\|\|$	Magnitude of column matrix R where R is matrix form of a vector
y	Matrix of IMU outputs
δy	Total error in IMU outputs
b	Matrix of bias in IMU outputs
w	Matrix of noise in IMU outputs
$E(\ \ )$	Expected value of
σ^2	Variance
W_x	Weighting factor for signal x.
A	State transition matrix
B	Control input
C, H	Output matrix
Q	Process noise covariance
P	Error covariance
R	Measurement noise covariance
K	Blending factor
$\vec{H}$	Angular momentum
$\vec{I}$	Angular impulse
$\vec{F}$	Force
m	Mass
c	Viscous friction coefficient
k	Spring constant
$F_A(\ \ )$	Cumulative distribution function
$f_A(a)$	Probability density function

Abbreviation	Definition
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
DCM	Direction Cosine Matrix
UTVP	Unmanned Tracked Vehicle Prototype
NES	Navigation Equations Solver
DOF	Degrees of Freedom
GPS	Global Positioning System

Through the text,

Numbers in brackets	DENOTE	References
Numbers in parenthesis		Equations

CHAPTER 1

INTRODUCTION

1.1 Dead Reckoning

Navigation, which is as old as the human being, is a method or collection of methods to find a way from one place to the other. The name is literally from the Latin “navigare” meaning to conduct a ship [2]. In order to find the way, the most crucial parameter is following a set of data or algorithm that describes that way. In ancient times, this data was the landmarks. Using landmarks to navigate is a practical and efficient method because landmarks indicate absolute position. This means, for the person navigating, there is chance to compare his position with the position of the landmark.

After the landmarks; coasts, mountains, shores seas etc. were noted and these instructions formed the maps. However landmarks may not be permanent. Instead of using landmarks, a global grid system to define the position of objects on Earth was admitted.

An alternative approach to using landmarks, maps, GPS or global coordinates is dead reckoning. According to [4], dead reckoning is the determination of present position from the initial position and the measurements of velocity, acceleration etc.

Dead reckoning is a method to appeal in navigation when the absolute position measurement is not available. In order to understand dead reckoning, following examples are supposed to be fancied.

Example 1:

Pedometer is a device that counts the footsteps. Footstep length can be stored in this device so that it can output the distance travelled. Many athletes use this device to calculate how much distance they run. For this application, the precision of the distance calculation is not so important. However the athlete may not aware, the calculation of the distance is highly related with the sensor (pedometer in this case) used. Also in reality, footstep length is not constant. As a result, precise calculation of the distance may not be possible with the pedometer.

Example 2:

A person with a rowing boat leaves a quay and puts out to the sea. But suddenly fog comes in. No landmark is visible. How can the boatman return to the quay? However he does not have any idea about where the quay is, he can just "estimate" the location. After estimating, he should also estimate the absolute orientation of the boat. After processing the orientation and the location of the quay, the best thing to do is rotating the boat 180° and to go straight to the estimated location of the quay. When arrived to the shore, it can be seen that there is difference between the arrived location and the quay. This is the "error". Some of the important sources of this error are:

1) Wrong estimation of quay location and orientation of the boat.
2) Effect of stream and wind
3) Impossibility of rowing along an exactly straight line.

Example 3:

Accelerometer is a device to measure the acceleration along its measurement axis. If accelerometer is moved straight and stopped, how can the final location be estimated? The answer is quite straight forward and easy for this "one dimensional

navigation" case. Collect the acceleration data and integrate twice. Add the initial position to this displacement.

But in this example there was only one dimension. What about 3 dimensional motion? For 3D case is it enough to measure 3 accelerometer readings? The answer is "No". This is because of the need of calculating orientation. There should be many inputs to estimate the final position.

In this thesis, a dead reckoning algorithm was written and tested on an unmanned tracked vehicle using a number of sensors (multisensor).

1.2 Multisensor Technology

Inertial navigation is a based on the fundamental laws of physics. The acceleration is created by a force and this force can stretch a spring whose end is mounted on a mass. Measuring the location of the mass will lead to measuring the acceleration. This is the working principle of a mechanical accelerometer.

The problem of navigation arises in a rotating mass. Assume that a mass is rotating on a table about a point. Attach an accelerometer to the body. The accelerometer will output some acceleration data. How can one know that is this data is created by a circular motion or rectilinear motion? It is impossible for this case. In order to solve this problem, about 1950's, gimbaled platforms were used. The accelerometers were mounted on these platforms. These stable platforms were isolating circular and rectilinear motions. In respect of [4], one of these platforms was 5ft in diameter and 2700lb.

Another solution to isolate rotation and linear motion is measuring the angular velocity by gyroscopes. From angular velocity, the centripetal acceleration can be subtracted from the acceleration readings and linear acceleration can be calculated. The development in sensor and microprocessor technology led the sensors be very

small. As a consequence, accelerometers and gyroscopes can be packed into small volumes. These kinds of systems are called strapdown systems.

According to [6], strap-down navigation systems are becoming increasingly important for a variety of reasons:

1) Increasingly powerful, smaller, and less-expensive computational equipment allow strap-down navigation algorithms to be implemented accurately and inexpensively in a small package.

2) Microminiaturization of inertial instruments allows for a reliable, low-cost, small-sized instrumentation package.

3) Global positioning technology and other aiding mechanisms allow on-line calibration of navigation systems at a reasonable cost.

4) The desire for higher levels of autonomy in many products necessitates on-board navigation.

5) The feasibility of navigation systems in some applications does not allow the size or the cost of mechanized-platform approaches.

In this thesis, not only a strapdown inertial measurement unit is used but also a digital compass and encoders aid to the navigation calculations. Navigation equations' solutions are redressed by the aiding sensors with Kalman filter algorithm in real time.

1.3 Scope of the Thesis

In this thesis, the aim is to collect real time acceleration and angular velocity data from an inertial measurement unit rigidly mounted on an unmanned tracked vehicle prototype. The data is sent to navigation equation solver algorithm and final position is calculated. Extra sensors improve the data by sensor fusion techniques. A number of sensor fusion methods are applied. Kalman filter was used in sensor fusion. As a result, an overall precise dead reckoning navigation algorithm was written and a test system (both the vehicle prototype and data

acquisition system) was built. The results of the tests are discussed at the end of the thesis.

1.4 Outline of the Thesis

The thesis is divided into 4 main parts. The sequence of parts is not only the timeline of the study done but also it is in a pedagogic order. The subjects was tried to be explained in an order so that, the reader can easily learn the matter.

In the first part of the thesis, a general topic is explained. For some main concepts in the thesis, definitions are made. For some concepts that are important and repeated through the thesis, quotation marks are used. Hereby, an interest was tried to be created.

In chapter 2, formulation of navigation equations were given. Some well-known theorems and equations were proved. In this section the core mathematical background to build navigation solver algorithm is made cleared.

Next chapter introduces the sensors and error modeling related to them. In order to understand the sensor fusion, the error models should be grasped.

The third section is all about the test setup built for this thesis. The unmanned tracked vehicle prototype is explained. The data acquisition system is expressed. In addition to these, the software form of algorithm in previous chapters is illustrated.

After the forth chapter where the test and test graphics were tabulated, comes the conclusion that is the end of the thesis.

CHAPTER 2

NAVIGATION EQUATIONS

2.1 Reference Frames

In navigation, a reference frame is the fundamental aspect from which the motion is observed.

A reference frame is defined by its origin and its coordinate axes. A basis vector is a unit vector whose direction is same as the direction of coordinate axes. The coordinate axes of the reference frame are orthogonal, right handed and in unit magnitude. Reference frame "a" having origin at A is expressed with its basis vectors as: $\mathcal{F}_a(A;\vec{\imath},\vec{\jmath},\vec{k})$. Mathematically, the basis vectors of a reference frame have the following properties:

1. Orthogonal: It means that each basis vector is orthogonal with respect to the others.
   $\vec{\imath}\cdot\vec{\jmath}=\vec{\jmath}\cdot\vec{k}=\vec{k}\cdot\vec{\imath}=0$
2. Right handed: The basis vectors' directions obey the right hand rule.
   $\vec{\imath}\times\vec{\jmath}=\vec{k}\quad\vec{\jmath}\times\vec{k}=\vec{\imath}\quad\vec{k}\times\vec{\imath}=\vec{\jmath}$
3. Unit basis vectors: All the three basis vectors have unit magnitude.
   $$\vec{\imath}\cdot\vec{\imath}=\vec{\jmath}\cdot\vec{\jmath}=\vec{k}\cdot\vec{k}=1$$

2.2 Definitions of Reference Frames in Navigation

In order to write the navigation equations, a number of reference frames is supposed to be defined. However, defining two reference frames is enough in writing the navigation equations for frame transformations, in literature there are intermediate frames to deal the calculations step by step.

2.2.1 Earth Centered and Inertial Reference Frame

This inertial frame is fixed to an inertial reference and its origin is Earth's center. A stationary point on with respect to ground has a velocity as much as the Earth's velocity with respect to the Earth centered and inertial reference frame. In this thesis, it is assumed that Earth centered and inertial reference frame is fixed with respect to any fixed point in space. Nonetheless, the Earth has a motion with respect to the Sun and the Sun is also not fixed, these terrestrial motions cannot to be sensed by the INS since they induce very little forces on the objects on Earth.

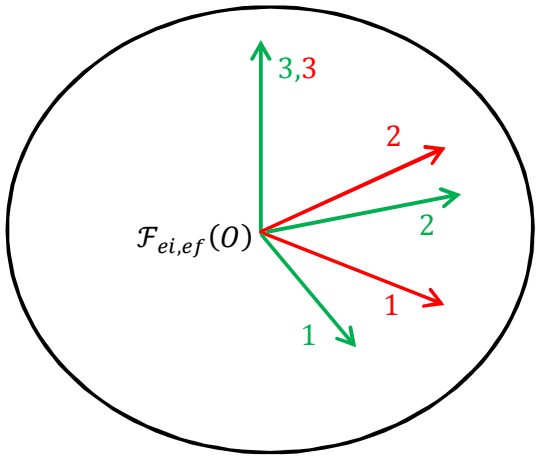


Figure 2.1 Earth Centered and Inertial Reference Frame and Earth Centered and Fixed Reference Frame

2.2.2 Earth Centered and Fixed Reference Frame

Earth centered and fixed reference frame is fixed to the Earth's origin and it rotates with respect to Earth. This means; a point that is stationary with respect to ground is also stationary with respect to Earth centered and fixed reference frame.

2.2.3 Navigation Reference Frame

Its orientation is dependent on the vehicle's location, onto which the INS is rigidly mounted, on Earth (latitude and longitude data). The frame's origin is attached to the vehicle on Earth. It points north, east and down directions by its basis vectors.

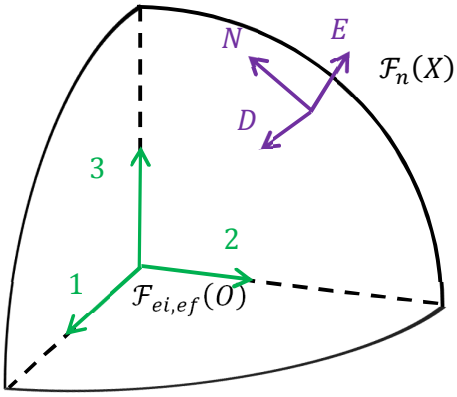


Figure 2.2 Navigation Reference Frame

2.2.4 Body Reference Frame

The body frame is rigidly attached to the vehicle carrying the INS. This frame has a motion with respect to navigation frame as the motion changes its attitude. In this thesis, attitude verbalizes that the orientation with respect to north east and down.

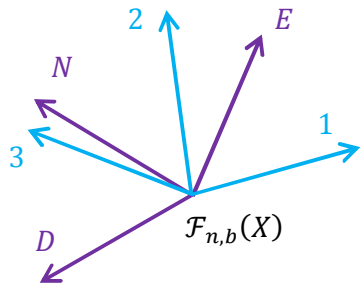


Figure 2.3 Body Reference Frame and Navigation Reference Frame

2.3 Reference Frame Transformations

2.3.1 Rodrigues Formula

Rodrigues formula is used in order to find the matrix representation of a vector that is obtained by the rotation of another vector.

Let $\vec{r} = \overrightarrow{OB}$ is obtained by θ much rotation of $\vec{p} = \overrightarrow{OA}$ about $\vec{n}$.

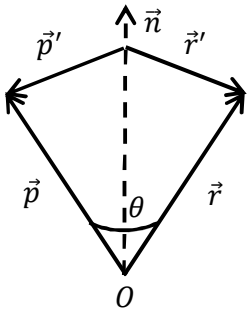


Figure 2.5 Front view

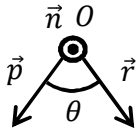


Figure 2.4 Top view

$\vec{p} = \vec{s} + \vec{p}'$ and $\vec{r} = \vec{s} + \vec{r}'$ Here, $\vec{s}$ is the vector component of $\vec{p}$ on $\vec{n}$.
$\vec{r}' = \vec{p}' cos\theta + \vec{q}' sin\theta$

$\vec{p}' = \vec{p} - \vec{s}$ where $\vec{s} = (\vec{p} \cdot \vec{n})\vec{n}$

$\vec{p}' = \vec{p} - (\vec{p} \cdot \vec{n})\vec{n}$ and

$$\vec{q}' = \vec{n} \times \vec{p}'$$

$$\vec{q}' = \vec{n} \times \vec{p} - (\vec{p} \cdot \vec{n})\vec{n} \times \vec{n} \text{ where } \vec{n} \times \vec{n} = \vec{0}$$

$$\therefore \ \vec{r} = \vec{s} + \vec{r}' = (\vec{p} \cdot \vec{n})\vec{n} + \vec{p}'cos\theta + \vec{q}'sin\theta$$

$$= (\vec{p} \cdot \vec{n})\vec{n} + (\vec{p} - (\vec{p} \cdot \vec{n})\vec{n})cos\theta + (\vec{n} \times \vec{p} - (\vec{p} \cdot \vec{n})\vec{n} \times \vec{n})sin\theta$$

If above equation is simplified:

$$\vec{r} = \vec{p}cos\theta + (\vec{n} \times \vec{p})sin\theta + (\vec{p} \cdot \vec{n})\vec{n}(1 - cos\theta) \quad (2.1)$$

If an observation frame is chosen and the vector expressions are expressed as matrix expressions in that reference frame:

$\bar{r} = \bar{p}cos\theta + \tilde{n}\bar{p}sin\theta + \bar{n}(\bar{n}^T\bar{p})(1 - cos\theta)$ where $\tilde{n}$ is the skew symmetric matrix of $\bar{n}$:

If $\bar{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}$; then its skew symmetric will be

$$\tilde{q} = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix} \quad (2.2)$$

If Rodrigues formula is divided by $\bar{p}$, the rotation matrix can be obtained as:

$$\hat{R}(n, \theta) = \hat{I}cos\theta + \tilde{n}sin\theta + \bar{n}\bar{n}^T(1 - cos\theta) \quad (2.3)$$

A vector rotation about a vector can be expressed as a single matrix. This matrix is called rotation matrix i.e. $\hat{R}(n, \theta)$. Successive rotations can be expressed as a unique rotation that is going to be the multiplication of each individual rotation. By the same way, a rotation matrix can be decomposed into at most 3 rotations about the basis vectors of a reference frame. This technique simplifies the use of Rodrigues formula. The 3 basic rotation matrices about basis vectors can be obtained as follows.

Rotation matrix about first axis or $\vec{u}_1^{(0)}$ (0 is the name of the reference frame in which all the vectors are expressed and 1 denotes the first axis.):

$$\hat{R}\left(\vec{u}_1^{(0)}, \psi\right) = \hat{I}cos\psi + \tilde{n}sin\psi + \bar{n}\bar{n}^T(1 - cos\psi) \tag{2.4}$$

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

$$+ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} [1 \quad 0 \quad 0](1 - cos\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\psi & -sin\psi \\ 0 & sin\psi & cos\psi \end{bmatrix}$$

Rotation matrix about 2nd axis or $\vec{u}_2^{(0)}$:

$$\hat{R}\left(\vec{u}_2^{(0)}, \theta\right) = \hat{I}cos\theta + \tilde{n}sin\theta + \bar{n}\bar{n}^T(1 - cos\theta) \tag{2.5}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} cos\theta + \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} sin\theta$$

$$+ \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} [0 \quad 1 \quad 0](1 - cos\theta) = \begin{bmatrix} cos\theta & 0 & sin\theta \\ 0 & 1 & 0 \\ -sin\theta & 0 & cos\theta \end{bmatrix}$$

Rotation matrix about 3rd axis or $\vec{u}_3^{(0)}$:

$$\hat{R}\left(\vec{u}_3^{(0)}, \phi\right) = \hat{I}cos\phi + \tilde{n}sin\phi + \bar{n}\bar{n}^T(1 - cos\phi) \tag{2.6}$$

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

$$+ \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} [0 \quad 0 \quad 1](1 - cos\phi) = \begin{bmatrix} cos\phi & -sin\phi & 0 \\ sin\phi & cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The most general rotation, meaning that, rotation at all three axes is the result of three basic rotations respectively. As a result, the general rotation matrix can be obtained as:

$$\hat{R}\left(\vec{u}_1^{(0)}, \psi\right)\hat{R}\left(\vec{u}_2^{(0)}, \theta\right)\hat{R}\left(\vec{u}_3^{(0)}, \phi\right)$$

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

(2.7)

2.3.2 Rotation Matrices (DCMs) between Reference Frames

The rotation matrix between the navigation frame and the Earth centered and fixed frame can be calculated by the latitude and longitude angles. Any point on the surface of Earth (at the sea level) can be located by latitude and longitude. In fact, these are the two rotations with respect the basis vectors of the Earth centered and fixed frame. Therefore the general rotation matrix can be rearranged to obtain the corresponding direction cosine matrix. Evaluating for $\psi = 0$ leads to:

$$\hat{R}(n,-\theta)\hat{R}(n,\phi) = \hat{C}^{(n,ef)} = \begin{bmatrix} cos\theta & -sin\phi sin\theta & cos\phi sin\theta \\ 0 & cos\phi & sin\phi \\ -sin\theta & -cos\theta sin\phi & cos\phi cos\theta \end{bmatrix} \tag{2.8}$$

In equation (2.8), θ is the latitude in radians (The sign convention of latitude is inverse of Earth centered and fixed frame hence it is negative.), ψ is the longitude in radians.

The rotation matrix between Earth centered and inertial frame and the Earth centered and fixed frame can be calculated by considering the Earth rotation rate. The rotation is about the 3rd axis only.

$$\hat{R}^T(n,\phi) = \hat{C}^{(ef,ei)} = \begin{bmatrix} cos\phi & -sin\phi & 0 \\ sin\phi & cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}^T \tag{2.9}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\Omega t) & sin(\Omega t) \\ 0 & -sin(\Omega t) & cos(\Omega t) \end{bmatrix}$$

Here t is the time. It can be thought as time of the day. This term is going to vanish by taking differentiations in the next sections.

2.3.3 Attitude Formulation

From INS, angular rate data from the gyroscope of INS is fed into the system. This data is used to calculate the attitude of the vehicle together with the accelerometer data.

By the definition of the orthonormality of the direction cosine matrix,

$\hat{C}^T \hat{C} = \hat{I}$

Differentiating this equation by applying product rule:

$$\frac{d(\hat{C}^T)}{dt}\hat{C} + \hat{C}^T \frac{d(\hat{C})}{dt} = \frac{d(\hat{I})}{dt} = \hat{0}$$

$$\frac{d(\hat{C}^T)}{dt}\hat{C} = -\hat{C}^T \frac{d(\hat{C})}{dt} = -\left[\frac{d(\hat{C}^T)}{dt}\hat{C}\right]^T$$

This shows that $\frac{d(\hat{C}^T)}{dt}\hat{C}$ is a skew symmetric matrix. Skew symmetric matrices of 3×3 that have only 3 nonzero elements [9]. These elements can be gathered together in a 3×1 column matrix. Let $\bar{\omega}$ be that column matrix and $\tilde{\omega}$ is a skew symmetric matrix formed by using the elements of this column matrix. Symbolically this operation is illustrated through the thesis as:

$SSF(\bar{\omega}) = \tilde{\omega}$

This can be applied to the reference frame rotation by defining $\hat{C} = \hat{C}^{(a,b)}$. Hence, $\hat{C}^T = \hat{C}^{(b,a)}$. a and b denote the reference frames.

$$\hat{C}^T \frac{d(\hat{C})}{dt} = \hat{C}^{(b,a)} \frac{d(\hat{C}^{(a,b)})}{dt} = \tilde{\omega}_{b/a} \tag{2.10}$$

As a result, above equation illustrates the relation between the angular velocity vector of a frame with respect to another frame and the corresponding direction cosine matrix.

2.3.4 Coriolis Theorem

The coriolis or transport theorem relates the differentiation of position vectors with respect to two different frames by using the knowledge of the angular velocity of one frame with respect to the other.

Let $\vec{r}$ is a position vector that is desired to be defined in two reference frames named as $\mathcal{F}_a$ and $\mathcal{F}_b$. $\{\vec{r}\}^{(a)}$ or $\bar{r}^{(a)}$ is the numeric column matrix representation of $\vec{r}$ which is expressed in $\mathcal{F}_a$.

$$\{\vec{r}\}^{(a)} = \hat{C}^{(a,b)}\{\vec{r}\}^{(b)} \tag{2.11}$$

Taking time derivative of above equation leads:

$$\frac{d\{\vec{r}\}^{(a)}}{dt} = \frac{d}{dt}\left[\hat{C}^{(a,b)}\{\vec{r}\}^{(b)}\right] = \frac{d\hat{C}^{(a,b)}}{dt}\{\vec{r}\}^{(b)} + \hat{C}^{(a,b)}\frac{d\{\vec{r}\}^{(b)}}{dt} \tag{2.12}$$

$$\frac{d\{\vec{r}\}^{(a)}}{dt} = \hat{C}^{(a,b)}\left[\hat{C}^{(b,a)}\frac{d\hat{C}^{(a,b)}}{dt}\{\vec{r}\}^{(b)} + \frac{d\{\vec{r}\}^{(b)}}{dt}\right] \tag{2.13}$$

$$\frac{d\{\vec{r}\}^{(a)}}{dt} = \hat{C}^{(a,b)}\left[\widetilde{\omega}_{b/a}^{(b)}\{\vec{r}\}^{(b)} + \frac{d\{\vec{r}\}^{(b)}}{dt}\right] \tag{2.14}$$

Converting all matrix representations to vectorial expressions turns the equation into coriolis theorem statement:

$$D_a\vec{r} = \vec{\omega}_{b/a} \times \vec{r} + D_b\vec{r} \tag{2.15}$$

2.3.5 Acceleration Vector Expression in Another Frame:

Assume that $\mathcal{F}_a(A)$ is a fixed frame and the relation between acceleration of point X with respect to $\mathcal{F}_a$ and $\mathcal{F}_b$is desired to be known.

$$\vec{r}_{X/A} = \vec{r}_{X/B} + \vec{r}_{B/A} \tag{2.16}$$

Taking double time derivative of both sides:

$$D_a^2\vec{r}_{X/A} = \vec{a}_{X/\mathcal{F}_a(A)} = D_a^2\vec{r}_{X/B} + D_a^2\vec{r}_{B/A}$$

$$\vec{a}_{X/\mathcal{F}_a(A)} = D_a^2\vec{r}_{X/B} + D_a^2\vec{r}_{B/A}$$

$$\vec{a}_{X/\mathcal{F}_a(A)} = D_a\left(D_a\vec{r}_{X/B}\right) + \vec{a}_{B/\mathcal{F}_a(A)}$$

Applying the coriolis theorem:

$$\vec{a}_{X/\mathcal{F}_a(A)} = D_a\left(\vec{v}_{X/\mathcal{F}_b(B)} + \vec{\omega}_{b/a} \times \vec{r}_{X/B}\right) + \vec{a}_{B/\mathcal{F}_a(A)}$$

$$\vec{a}_{X/\mathcal{F}_a(A)} = D_a\vec{v}_{X/\mathcal{F}_b(B)} + D_a\left(\vec{\omega}_{b/a} \times \vec{r}_{X/B}\right) + \vec{a}_{B/\mathcal{F}_a(A)}$$

$$\vec{a}_{X/\mathcal{F}_a(A)} = \vec{a}_{X/\mathcal{F}_b(B)} + \vec{\omega}_{b/a} \times \vec{v}_{X/\mathcal{F}_b(B)} + \vec{\alpha}_{b/a} \times \vec{r}_{X/B} + \vec{\omega}_{b/a} \times \left(\vec{v}_{X/\mathcal{F}_b(B)} + \vec{\omega}_{b/a} \times \vec{r}_{X/B}\right) + \vec{a}_{B/\mathcal{F}_a(A)}$$

$$\vec{a}_{X/\mathcal{F}_a(A)} = \vec{a}_{X/\mathcal{F}_b(B)} + \vec{\omega}_{b/a} \times \vec{v}_{X/\mathcal{F}_b(B)} + \vec{\alpha}_{b/a} \times \vec{r}_{X/B} + \vec{\omega}_{b/a} \times \vec{v}_{X/\mathcal{F}_b(B)} + \vec{\omega}_{b/a} \times \left(\vec{\omega}_{b/a} \times \vec{r}_{X/B}\right) + \vec{a}_{B/\mathcal{F}_a(A)}$$

$$\vec{a}_{X/\mathcal{F}_a(A)} = \vec{a}_{X/\mathcal{F}_b(B)} + 2\vec{\omega}_{b/a} \times \vec{v}_{X/\mathcal{F}_b(B)} + \vec{\alpha}_{b/a} \times \vec{r}_{X/B} + \vec{\omega}_{b/a} \times (\vec{\omega}_{b/a} \times \vec{r}_{X/B}) + \vec{a}_{B/\mathcal{F}_a(A)} \quad (2.17)$$

In this expression some terms have special names that are named as:

$2\vec{\omega}_{b/a} \times \vec{v}_{X/\mathcal{F}_b(B)}$: Coriolis acceleration term

$\vec{\omega}_{b/a} \times (\vec{\omega}_{b/a} \times \vec{r}_{X/B})$: Centripetal acceleration term

2.4 Position Formulation

Solution of navigation equations that is named as strapdown system mechanization can be accomplished by converting system components from one reference frame to the other.

Before going into the navigation equations, the acceleration sensed by the INS is supposed to be explained. The acceleration is measured absolutely in INS. In other words, acceleration data is sent in a fixed frame. Since Earth centered and inertial reference is assumed to be fixed in this thesis, measured acceleration is composed into two parts:

$$\vec{a}_{X/\mathcal{F}_{ei}(O)} = measured\ acceleration = \vec{h} + \vec{g} \quad (2.18)$$

$\vec{h}$: Acceleration vector caused by the motion of vehicle.

$\vec{g}$: Acceleration vector caused by Earth at the vehicle's location.

The $\vec{g}$ term also composed of two parts. The first one is the gravitational acceleration and the second part is centripetal acceleration caused by the rotation of Earth. For this reason, $\vec{g}$ term is known as plumb bob gravity [4].

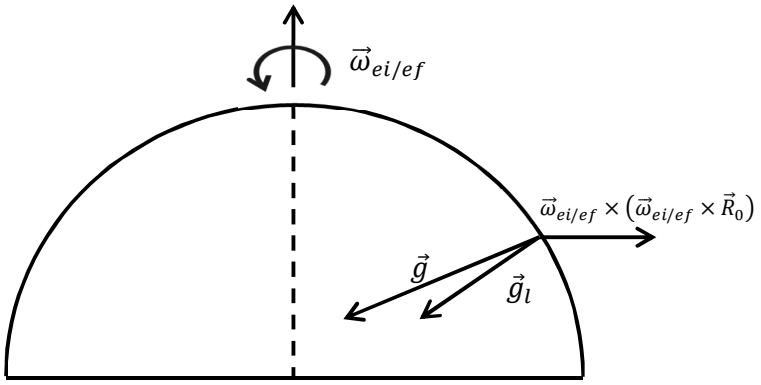


Figure 2.6 Plumb bob gravity vector and its components on Earth's surface

$$\vec{g}_l = \vec{g} - \vec{\omega}_{ei/ef} \times \left(\vec{\omega}_{ei/ef} \times \vec{R}_0\right) \tag{2.19}$$

$$\left.\frac{d\vec{v}_{ei/ef}}{dt}\right|_{ei} = \vec{h} - \vec{\omega}_{ei/ef} \times \vec{v}_{ei/ef} + \vec{g}_l \tag{2.20}$$

Velocity of point X which is the location of the INS should be differentiated in Earth centered and fixed reference frame:

$$D_n\vec{v}_{X/\mathcal{F}_{ef}(O)} = \vec{a}_{X/\mathcal{F}_{ef}(O)} + \vec{\omega}_{ef/n} \times \vec{v}_{X/\mathcal{F}_{ef}(O)} \tag{2.21}$$

Decomposing $\vec{a}_{X/\mathcal{F}_{ef}(O)}$ term according to equation (2.15):

$$\vec{a}_{X/\mathcal{F}_{ef}(O)} = \vec{a}_{X/\mathcal{F}_{ei}(O)} + 2\vec{\omega}_{ei/ef} \times \vec{v}_{X/\mathcal{F}_{ei}(O)} + \vec{\alpha}_{ei/ef} \times \vec{r}_{X/O} + \vec{\omega}_{ei/ef} \times (\vec{\omega}_{ei/ef} \times \vec{r}_{X/O}) + \vec{a}_{O/\mathcal{F}_{ef}(O)} \tag{2.22}$$

Since Earth's angular rate is constant, $\vec{\alpha}_{ei/ef} = \vec{0}$. This simplifies the equation to

$$\vec{a}_{X/\mathcal{F}_{ef}(O)} = \vec{a}_{X/\mathcal{F}_{ei}(O)} + 2\vec{\omega}_{ei/ef} \times \vec{v}_{X/\mathcal{F}_{ei}(O)} + \vec{\omega}_{ei/ef} \times (\vec{\omega}_{ei/ef} \times \vec{r}_{X/O}) + \vec{a}_{O/\mathcal{F}_{ef}(O)} \quad (2.23)$$

Inserting equation (2.23) to equation (2.21):

$$D_n\vec{v}_{X/\mathcal{F}_{ef}(O)} = \vec{a}_{X/\mathcal{F}_{ei}(O)} + 2\vec{\omega}_{ei/ef} \times \vec{v}_{X/\mathcal{F}_{ei}(O)} + \vec{\omega}_{ei/ef} \times (\vec{\omega}_{ei/ef} \times \vec{r}_{X/O}) + \vec{a}_{O/\mathcal{F}_{ef}(O)} + \vec{\omega}_{ef/n} \times \vec{v}_{X/\mathcal{F}_{ef}(O)} \quad (2.24)$$

By using the definitions in (2.18):

$$D_n\vec{v}_{X/\mathcal{F}_{ef}(O)} = \vec{h} + \vec{g} + 2\vec{\omega}_{ei/ef} \times \vec{v}_{X/\mathcal{F}_{ei}(O)} + \vec{g}_l - \vec{g} + \vec{a}_{O/\mathcal{F}_{ef}(O)} + \vec{\omega}_{ef/n} \times \vec{v}_{X/\mathcal{F}_{ef}(O)} \quad (2.25)$$

The coriolis acceleration term is equal to:

$$2\vec{\omega}_{ei/ef} \times \vec{v}_{X/\mathcal{F}_{ei}(O)} = 2\vec{\omega}_{ei/ef} \times \vec{v}_{X/\mathcal{F}_{ef}(O)} \quad (2.26)$$

Equation (2.26) can be rewritten as:

$$D_n\vec{v}_{X/\mathcal{F}_{ef}(O)} = \vec{h} + 2\vec{\omega}_{ei/ef} \times \vec{v}_{X/\mathcal{F}_{ef}(O)} + \vec{g}_l + \vec{a}_{O/\mathcal{F}_{ef}(O)} + \vec{\omega}_{ef/n} \times \vec{v}_{X/\mathcal{F}_{ef}(O)} \quad (2.27)$$

$$D_n\vec{v}_{X/\mathcal{F}_{ef}(O)} = \vec{h} - \left(2\vec{\omega}_{ef/ei} + \vec{\omega}_{n/ef}\right) \times \vec{v}_{X/\mathcal{F}_{ef}(O)} + \vec{g}_l \quad (2.28)$$

2.5 Navigation Equations in North-East-Down Velocity Form

In equation (2.28), the vectorial expressions have to be converted into numeric matrix forms:

$$\left\{D_n\vec{v}_{X/\mathcal{F}_{ef}(O)}\right\}^n = \bar{h}^n - SSF\left(2\bar{\omega}^n_{ef/ei} + \bar{\omega}^n_{n/ef}\right)\left\{\vec{v}_{X/\mathcal{F}_{ef}(O)}\right\}^n + \bar{g}^n_l \quad (2.29)$$

In order to go into the solution of equation (2.29), the terms should be written numerically. $\bar{h}^b$ is the acceleration data from INS in body frame. That is directly

equal to the accelerometer readings of INS that will be symbolized by a_1, a_2 and a_3. The subscripts express the axis number:

$$\bar{h}^n = \hat{C}^{(b,n)}\bar{h}^b \; and \; \bar{h}^b = [a_1 \quad a_2 \quad a_3]^T; \; \bar{h}^n = [a_N \quad a_E \quad a_D]^T \tag{2.30}$$

The velocity terms' matrix expressions are:

$$\left\{\vec{v}_{X/\mathcal{F}_{ef}(O)}\right\}^n = [v_N \quad v_E \quad v_D]^T \; and \; \left\{D_n\vec{v}_{X/\mathcal{F}_{ef}(O)}\right\}^n = [\dot{v}_N \quad \dot{v}_E \quad \dot{v}_D]^T \tag{2.31}$$

By using equations (2.8) and (2.10) following matrix forms can be written:

$$\bar{\omega}^n_{n/ef} = \bar{\omega}_{n/ef} = SSF^{-1}\left[\hat{C}^{(n,ef)}\frac{d\hat{C}^{(ef,n)}}{dt}\right] \tag{2.32}$$

$$\bar{\omega}^n_{n/ef} = SSF^{-1}\begin{bmatrix} 0 & -\dot{\phi}cos\theta & -\dot{\theta} \\ -\dot{\phi}sin\theta & 0 & \dot{\phi}sin\theta \\ \dot{\theta} & \dot{\phi}cos\theta & 0 \end{bmatrix} = \begin{bmatrix} \dot{\phi}cos\theta \\ -\dot{\theta} \\ -\dot{\phi}sin\theta \end{bmatrix} \tag{2.33}$$

The latitude and longitude rates can be written in velocity form as follows. Here, north and east radii of curvature are different because of the ellipsoidal shape of the Earth. This subject is explained in "Earth Reference Model".

$$\dot{\phi} = \frac{v_E}{(R_E + y)cos\theta} \; and \; \dot{\theta} = \frac{v_N}{(R_N + y)} \tag{2.34}$$

Inserting equation (2.34) into (2.33) leads

$$\bar{\omega}^n_{n/ef} = \begin{bmatrix} \frac{v_E}{(R_E + y)} & \frac{-v_N}{(R_N + y)} & \frac{v_E tan\theta}{(R_N + y)} \end{bmatrix}^T \tag{2.35}$$

By using equations (2.8) and (2.10) following matrix forms can be written:

$$\bar{\omega}^n_{ef/ei} = \hat{C}^{(ef,n)}\bar{\omega}_{ef/ei} = \hat{C}^{(ef,n)}SSF^{-1}\left[\hat{C}^{(ef,ei)}\frac{d\hat{C}^{(ei,ef)}}{dt}\right] \tag{2.36}$$

$$\bar{\omega}^n_{ef/ei} = \begin{bmatrix} cos\theta & -sin\phi sin\theta & cos\phi sin\theta \\ 0 & cos\phi & sin\phi \\ -sin\theta & -cos\theta sin\phi & cos\phi cos\theta \end{bmatrix}\begin{bmatrix} \Omega \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \Omega cos\theta \\ 0 \\ -\Omega sin\theta \end{bmatrix} \tag{2.37}$$

According to definition in (2.19) local gravity vector can be written in navigation frame as:

$$\bar{g}_l^n = \left\{\vec{g} - \vec{\omega}_{ei/ef} \times \left(\vec{\omega}_{ei/ef} \times \vec{R}_0\right)\right\}^n \tag{2.38}$$

$$\bar{g}_l^n = \bar{g}^n - \tilde{\omega}_{ei/ef}^n \tilde{\omega}_{ei/ef}^n [0 \quad 0 \quad -R_0 - y]^T$$

$$\bar{g}_l^n = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} - \begin{bmatrix} 0 & \Omega sin\theta & 0 \\ -\Omega sin\theta & 0 & \Omega cos\theta \\ 0 & \Omega cos\theta & 0 \end{bmatrix}^2 \begin{bmatrix} 0 \\ 0 \\ -R_0 - y \end{bmatrix} = \begin{bmatrix} sin2\theta \\ 0 \\ 1 + cos2\theta \end{bmatrix} \tag{2.39}$$

Inserting all equations from (2.30) to (2.39) into (2.29) gives navigation equations set.

$$\dot{v}_N = \frac{(R_0 + y)\Omega^2 sin2\theta}{2} - 2\Omega v_E sin\theta + a_N - \frac{v_E^2 tan\theta + v_N v_D}{R_N + y} \tag{2.40}$$

$$\dot{v}_E = 2(v_D cos\theta + v_N sin\theta)\Omega + a_E + \frac{v_E v_D}{R_E + y} + \frac{v_E v_N tan\theta}{R_N + y} \tag{2.41}$$

$$\dot{v}_D = \frac{(1 + cos2\theta)(R_0 + y)\Omega^2}{2} - 2\Omega v_E cos\theta + a_D - \frac{v_E^2}{R_E + y} - \frac{v_N^2}{R_N + y} \tag{2.42}$$

$$\begin{bmatrix} a_N \\ a_E \\ a_D \end{bmatrix} = \hat{C}^{(b,n)} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \tag{2.43}$$

2.6 Earth Reference Model

2.6.1 The Gravity

By solving the navigation equations, detection of gravity vector is crucial. This is because the gravity can be calculated as the part of acceleration causing motion. This results in high errors in position calculations. Therefore, a constant gravity assumption may not be enough for precise calculations.

Gravity is created due to the molten metals in the core of the Earth. The distribution of these metals may not be uniform at some levels and this causes little deflections in Earth's gravitation field. These deflections are so little; according to [14], a low cost INS cannot sense them.

The major variation of gravity in magnitude is due to latitude and altitude (height above sea level).

According to [13], the gravity as a function of latitude is formulated as:

$$g_{y=0} = 9.780318(1 + 0.0053024sin^2\theta - 5.9 \times 10^{-6}sin^2 2\theta) \quad (2.44)$$

In this formula, gravity is written for sea level. However, the gravity is also affected by distance to the Earth's center. The gravity at some height above sea level can be written from the law of universal gravitation as:

$$g_{y>0} = \frac{g_{y=0}}{\left(1 + \frac{y}{R_0}\right)^2} \quad (2.45)$$

2.6.2 Shape of the Earth

According to [1], the sphere is a close approximation of the true figure of the Earth and satisfactory for many purposes, precise measurements require that an ellipsoidal shape should be fit to the Earth's surface.

In this thesis, the Earth is assumed to be in the shape of an ellipsoid. This ellipsoid can be defined as a closed surface which is formed as the rotation of an ellipse 180° around its minor axis.

The eccentricity of an ellipsoid can be explained as the ratio of the distances between two foci (singular: focus) for a point *X* on that ellipse. Eccentricity is a constant for an ellipse.

By modeling the Earth in accordance with a reference ellipsoid as defined here, a meridian radius of curvature (R_N) and a transverse radius of curvature (R_E) can be derived in accordance with the following equations [4]:

$$R_N = \frac{R_0(1 - e^2)}{(1 - e^2 sin^2\theta)^{1.5}} \quad (2.46)$$

$$R_E = \frac{R_0}{\sqrt{1 - e^2 sin^2\theta}} \tag{2.47}$$

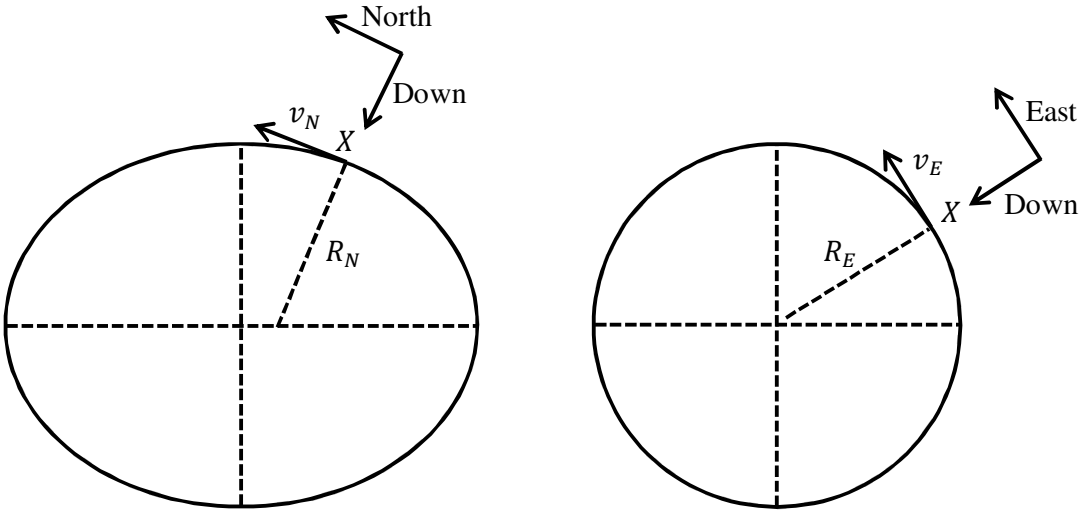


Figure 2.7 Meridian and transverse radius of curvature in side and top view of the Earth

The eccentricity of the Earth, the Earth's rotation rate, minor and major axes are taken from WGS-84 model. WGS-84 model is published by World Geodetic System Committee in 1984 [5].

Table 2.1 The WGS-84 model data used in this thesis

Length of semi major axis	R_0	$6378137.0\ m$
Length of semi minor axis	r	$6356752.3\ m$
Flattening of the ellipsoid	f	$^1/_{298.257}$
Eccentricity of the ellipsoid	e	0.0818191908426
Earth's rate	Ω	$7.292115 \times 10^{-5}\ rad/s$

Inserting (2.46) and (2.47) into the navigation equations leads to the detailed navigation equations.

2.7 Discretization of Navigation Equations

The matrix form of navigation equations was derived in (2.29) as

$$\left\{D_n \vec{v}_{X/\mathcal{F}_{ef}(O)}\right\}^n = \bar{h}^n - SSF\left(2\bar{\omega}^n_{ef/ei} + \bar{\omega}^n_{n/ef}\right)\left\{\vec{v}_{X/\mathcal{F}_{ef}(O)}\right\}^n + \bar{g}^n_l$$

If this is integrated following equation is obtained:

$$\left\{\vec{v}_{X/\mathcal{F}_{ef}(O)}\right\}^n = \int_0^t \bar{h}^n\, dt - \int_0^t \left[SSF\left(2\bar{\omega}^n_{ef/ei} + \bar{\omega}^n_{n/ef}\right)\left\{\vec{v}_{X/\mathcal{F}_{ef}(O)}\right\}^n\right] dt + \int_0^t \bar{g}^n_l\, dt \tag{2.48}$$

In this equation set, t represents the step time. Step time is the time elapsed between two data points. In order to integrate between two data points there are a number of numerical integration techniques. These techniques are explained in Appendix XX.

The initial conditions of equation (2.48) are known. Therefore solution is straight forward; at each time step, integrations are calculated and added to the initial condition. The result of each time step calculation is the initial condition of the next time step.

On the other hand, attitude of INS is supposed be calculated by the formula:

$$\bar{h}^n = \hat{C}^{(b,n)}\bar{h}^b$$

In this formulation, $\hat{C}^{(b,n)}$ is updated at each time step and the updated term can be calculated by the angular rate data from gyros of IMU.

From equation (2.10),

$$\hat{C}^{(b,n)} = \tilde{\omega}_{b/n} \frac{d\left(\hat{C}^{(b,n)}\right)}{dt} \tag{2.49}$$

This is a first order linear differential equation [7] and its solution is:

$$\hat{C}^{(b,n)}\big|_t = \hat{C}^{(b,n)}\big|_0 \, exp \int_0^t \tilde{\omega}_{b/n} dt \text{ where } t \text{ is the step time.} \tag{2.50}$$

Expanding $exp \int_0^t \tilde{\omega}_{b/n} dt$ in Taylor Series leads to:

$$exp \int_0^t \tilde{\omega}_{b/n} dt = \hat{I} + \frac{\left(\int_0^t \tilde{\omega}_{b/n} dt\right)^2}{2!} + \frac{\left(\int_0^t \tilde{\omega}_{b/n} dt\right)^3}{3!} + \frac{\left(\int_0^t \tilde{\omega}_{b/n} dt\right)^4}{4!} \ldots \tag{2.51}$$

This expression is collected as: (See Appendix XX for details.)

$$exp \int_0^t \tilde{\omega}_{b/n} dt = \hat{I} + \frac{sin\varphi}{\varphi} \int_0^t \tilde{\omega}_{b/n} dt + \frac{(1 - cos\varphi)}{\varphi^2} \left(\int_0^t \tilde{\omega}_{b/n} dt \right)^2 \tag{2.52}$$

$$\varphi = \sqrt{\omega_1^2 + \omega_2^2 + \omega_3^2} \tag{2.53}$$

Inserting equation (2.52) into (2.50):

$$\hat{C}^{(b,n)}\big|_t = \hat{C}^{(b,n)}\big|_0 \left[\hat{I} + \frac{sin\varphi}{\varphi} \int_0^t \tilde{\omega}_{b/n} dt + \frac{(1 - cos\varphi)}{\varphi^2} \left(\int_0^t \tilde{\omega}_{b/n} dt \right)^2 \right] \tag{2.54}$$

In order to solve this attitude formulation, initial condition (initial attitude of INS) is required. Like the position formulation, at each time step, integrations are calculated and added to the initial condition. The result of each time step calculation is the initial condition of the next time step.

2.8 Direction Cosine Matrix Corrections

The rows of the direction cosine matrix represent the projection of unit vectors which lie along each axis of the orthogonal reference coordinate in the body frame [4]. In computer because of some calculation errors, the orthogonality and

normality of direction cosine matrix may deteriorate. In order to solve this, DCM is corrected by two processes.

2.8.1 Orthogonalization by Gram-Schmidt Algorithm

An orthogonal matrix is a square matrix that its rows are orthonormal with respect to each other. This means, if the rows of direction cosine matrix are defined as vectors, these vectors have $90°$ angle between. According to [10], orthogonalization can be accomplished by Gram-Schmidt algorithm. Let the DCM has the following rows:

$$\hat{C} = \begin{bmatrix} \hat{R}_1 \\ \hat{R}_2 \\ \hat{R}_3 \end{bmatrix}$$

After orthogonalization the DCM takes the following form:

$$\hat{C}_{ort} = \begin{bmatrix} \hat{R}_1 \\ \hat{R}_2 - \dfrac{\hat{R}_2 \hat{R}_1^T}{\left\|\hat{R}_1\right\|^2} \hat{R}_1 \\ \hat{R}_3 - \dfrac{\hat{R}_3 \hat{R}_1^T}{\left\|\hat{R}_1\right\|^2} \hat{R}_1 - \dfrac{\hat{R}_3 \hat{R}_{ort,2}^T}{\left\|\hat{R}_{ort,2}\right\|^2} \hat{R}_{ort,2} \end{bmatrix}$$

2.8.2 Normalization

In normalization the magnitudes of the rows of the DCM (these rows are thought to be like vectors) is equated to one.

$$\hat{C}_{nor} = \begin{bmatrix} \dfrac{\hat{R}_{ort,1}}{\left\|\hat{R}_{ort,1}\right\|} \\ \dfrac{\hat{R}_2}{\left\|\hat{R}_{ort,2}\right\|} \\ \dfrac{\hat{R}_{ort,3}}{\left\|\hat{R}_{ort,3}\right\|} \end{bmatrix}$$

Normalization is necessary in discrete calculations of differential equations. In these calculations the DCM deforms and it requires orthogonalization and normalization procedures respectively.

CHAPTER 3

INERTIAL NAVIGATION SYSTEMS

3.1 Inertial Measurement Unit

An inertial navigation system is composed of three single axis gyroscopes and three single axis accelerometers and a computation facility, namely a computer, or a microprocessor where navigation equations are solved.

The gyroscopes and accelerometers are mounted in a casing so that they form a right handed triad at the body frame attached to this casing. The casing also includes electronic parts such as analog to digital converter and data transmission module etc. The 6 inertial sensors and the electronics make up the strapdown inertial measurement unit.

In 1950's IMU with stable platforms were in use. However these systems suffer from having high weights and occupying high volumes. They are still being used in ships. Strapdown IMU's became popular when sensors were manufactured as chips. This made the way for compacting 3 accelerometers and 3 gyroscopes in small volumes.

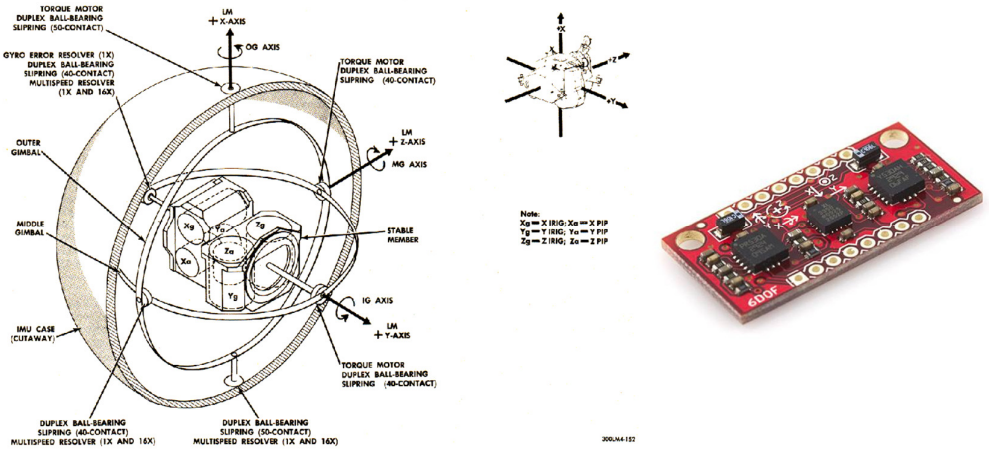


Figure 3.1 A gimbaled IMU schematic on the left and a photograph of strapdown modern IMU on the right [30, 31]

The gyroscopes and accelerometers of the IMU is mounted so that they form right handed triad. Thus, the angular rate and acceleration in all 3 axes are available. In order to read true angular velocities, IMU should be mounted on the mass center of the vehicle. If it is not possible, meaning that there is offset between the mass venter of vehicle and the IMU, the offset vector should be added into navigation equations as a corrector.

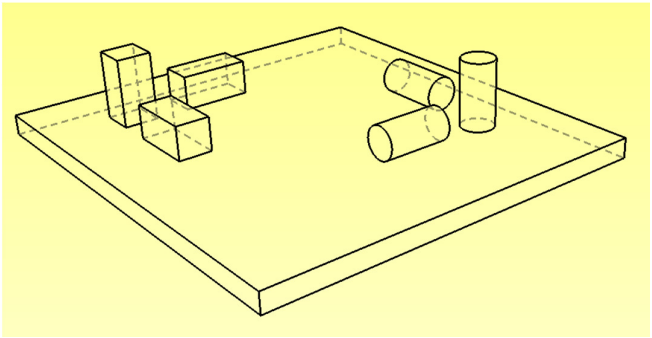


Figure 3.2 The representative orientation of sensors in an IMU. The cylinders are gyroscopes, the prisms are accelerometers

In order to understand the IMU, the accelerometers and the gyroscopes should be studied.

3.2 Gyroscope Technology

A gyroscope is a device to measure the angular velocity around a defined axis. There is also some other type of gyroscope that measures the angular displacement. In this thesis, the IMU includes gyroscopes for measuring the angular velocity.

According to [4], gyroscopes have variety of roles in:

- Stabilization
- Autopilot feedback
- Navigation
- Flight path sensor

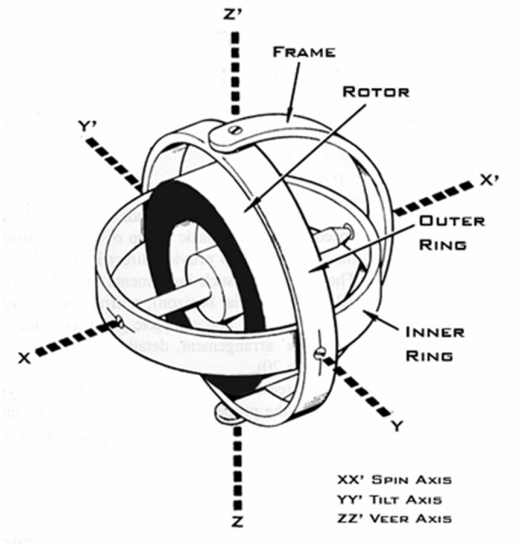


Figure 3.3 Simple mechanical gyroscope [29]

There are numerous types of gyroscopes designed to be used for specific reasons. Some important types of these gyroscopes are as follows:

- Rate integrated gyroscope
- Dynamically tuned gyroscope
- Silicon sensors
- Ring laser gyroscope
- Fiber optic gyroscope

In this decade, silicon low cost chips, ring laser and fiber optic gyroscopes are widely used.

The principle operation of a mechanical single axis gyroscope is as follows.

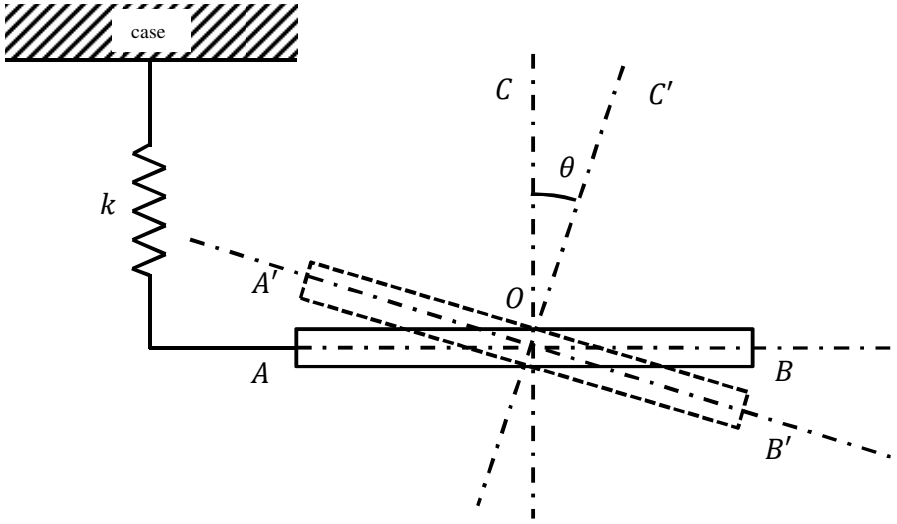


Figure 3.4 The simple single axis gyroscope; the disc on the gyroscope is slewed at constant velocity during operation

Assume that a disc is mounted with its gimbal into a case and it is free to rotate around axis OC. It has a constant angular velocity of $\vec{\omega}$ and therefore an angular

momentum of $\vec{H}$. A torque $\vec{T}(t)$ is applied on the disc by the casing and the disc rotates $\vec{\theta}(t)$ much. This angular deflection is called precession. The change in angular momentum of the disc in t_{end} much time is:
$\frac{d}{dt}\vec{H} = \frac{d}{dt}\vec{\theta} \times \vec{H} + \frac{dH}{dt}\overrightarrow{OC}$ where $\overrightarrow{OC}$ is an unit vector.

According to Newton's second law, the rate of change of angular momentum is equal to torque applied to the disc.

$$\vec{T} = \frac{d}{dt}\vec{\theta} \times \vec{H} + \frac{dH}{dt}\overrightarrow{OC} \tag{3.1}$$

Assume that gyroscope design is such that $\frac{dH}{dt}$ is extremely small and therefore negligible and $\vec{\omega}$ is still being kept constant. Neglecting this component leads to the law of gyroscopes:

$$\vec{T} = \frac{d}{dt}\vec{\theta} \times \vec{H} \tag{3.2}$$

Integrate from t_0 to t_{end} and obtain angular impulse:

$$\vec{I} = \vec{\theta} \times \vec{H} \tag{3.3}$$

By measuring the precession angle $\vec{\theta}$ by means of a spring with a very small spring constant k, total impulse can be calculated because $\vec{H}$ is known and constant. Since all inertial quantities are known from $\vec{I}$, the angular velocity can be calculated. After all these mathematical processing, the gyroscope outputs the angular velocity.

3.3 Accelerometer Technology

In order to solve the navigation equations, integration of accelerometer readings is required to reach translational velocity and position.

Accelerometers are used in:

- Navigation
- Vibration measurement
- Mechanical shock detection
- Gravity vector determination

For most of the applications, a 2nd order system model is enough for the study. In addition to that, a higher order mathematical model is going to need more information than the one presented in its datasheet.

A second order mathematical model can be developed by considering the accelerometer as a mechanical sensor as shown below.

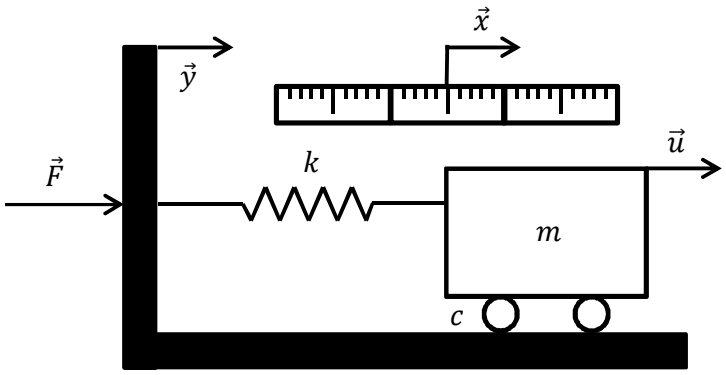


Figure 3.5 Simple mechanical one axis accelerometer

If a force $\vec{F}$ is exerted on the massless case of the accelerometer in, the equation of motion of the mass will be:

$$\vec{F} = m\vec{a} + c\vec{v} + k\vec{x} \text{ and } \vec{x} = \vec{u} + \vec{y} \quad (3.4)$$

Writing in Laplace domain leads to:

$$F = ms^2X(s) + csX(s) + kX(s) \quad (3.5)$$

According to the calculation steps in [15],

The natural frequency of accelerometer is $\sqrt{\frac{k}{m}}$ and the damping ratio is $\frac{c}{2\sqrt{km}}$. The natural frequency of an accelerometer is important for vibration readings. This quantity determines the frequency of mechanical oscillation that the accelerometer can read.

There are various types of accelerometers. Some accelerometer types are:

- Surface acoustic wave accelerometer
- MEMS accelerometer
- Fiber optic accelerometers
- Optical accelerometers

Further information about the types of accelerometers can be read from reference [4].

In this thesis, the IMU includes three axes of MEMS accelerometers. These sensors are based rugged, field proven silicon bulk micromachining technology. Each sensor is individually factory calibrated for temperature and non-linearity effects during Crossbow's manufacturing and test process using automated thermal chambers and rate tables [16].

3.4 Error model of IMU

In respect of [18], there are many error sources of an IMU. If gyroscope model in section 3.2 is considered, the case of the gyroscope can be mounted with an offset angle. This causes measurement of an amount of full angular velocity. There may be a problem in the reading of the length of the spring or there may be an electrical noise source making the outputs noisy or there may be problem in all etc. For an accelerometer, the reading may have offset that is used to find the location of accelerometer mass. There are many other error sensors. Reference [20] describes

more than ten error cases. The important principle to put the errors in mathematics is not grouping the error sources but finding some coefficients in error equation. Two error terms dominate the error quantity. One of them is the bias and the other is the constant linear coefficient. The same thing is valid for other sensors. That is to say, the error model of accelerometers and gyroscopes is assumed to be linear in this thesis as in the literature [4], [14] and [20].

Calibration is a process to relate the erroneous measurements with the reliable ones. In factories, the calibration is done for sensors and gauges. For precise measurement linear calibration is enough. In Roketsan, torque sensors, gyros, potentiometers etc. is linearly calibrated. The error equation can be thought to be same as calibration equation:

The error equation for the IMU is:

$$\delta y = b + w \tag{3.6}$$

3.4.1 Attitude Errors

Let estimated value of $\hat{C}^{(b,n)}$ be $\hat{C}_{est}^{(b,n)}$ and the relation between estimated direction cosine matrix and the true is expressed as:

$$\hat{C}_{est}^{(b,n)} = [I - \Gamma]\hat{C}^{(b,n)} \text{ where } \Gamma = \begin{bmatrix} 0 & -\delta\gamma & \delta\beta \\ \delta\gamma & 0 & -\delta\alpha \\ -\delta\beta & \delta\alpha & 0 \end{bmatrix}$$

In this equation, $\delta\alpha$, $\delta\beta$ and $\delta\gamma$ are the misalignment errors.

The equation can be rearranged as:

$$\Gamma = I - \hat{C}_{est}^{(b,n)}\left[\hat{C}^{(b,n)}\right]^T \tag{3.7}$$

Differentiate this equation:

$$\frac{d}{dt}\Gamma = -\frac{d}{dt}\hat{C}_{est}^{(b,n)}\left[\hat{C}^{(b,n)}\right]^T - \hat{C}_{est}^{(b,n)}\frac{d}{dt}\left[\hat{C}^{(b,n)}\right]^T \tag{3.8}$$

Solving this equation as in reference [4] leads to:

$$\frac{d}{dt}\Gamma = \Gamma\tilde{\omega}^n_{ei/n} - \tilde{\omega}^n_{ei/n}\Gamma + \delta\tilde{\omega}^n_{ei/n} - \hat{C}^{(b,n)}\delta\tilde{\omega}^b_{ei/b}\left[\hat{C}^{(b,n)}\right]^T \tag{3.9}$$

Remember that:

$$\bar{\omega}^n_{n/ef} = \left[\frac{v_E}{(R_E+y)} \quad \frac{-v_N}{(R_N+y)} \quad \frac{v_E tan\theta}{(R_N+y)}\right]^T \; and \; \bar{\omega}^n_{ef/ei} = \begin{bmatrix} \Omega cos\theta \\ 0 \\ -\Omega sin\theta \end{bmatrix}$$

Therefore;

$$\tilde{\omega}^n_{ei/n} = SSF\left(-\bar{\omega}^n_{ef/ei} - \bar{\omega}^n_{n/ef}\right)$$

$$= \begin{bmatrix} 0 & \frac{v_E}{(R_E+y)} + \Omega cos\theta & \frac{v_N}{(R_N+y)} \\ \frac{v_E tan\theta}{(R_N+y)} - \Omega sin\theta & 0 & \frac{-v_E tan\theta}{(R_N+y)} + \Omega sin\theta \\ \frac{-v_N}{(R_N+y)} & -\frac{v_E}{(R_E+y)} - \Omega cos\theta & 0 \end{bmatrix}$$

3.4.2 Position and Velocity Errors

The velocity equation is:

$$\left\{D_n\vec{v}_{X/\mathcal{F}_{ef}(O)}\right\}^n = \bar{h}^n - SSF\left(2\bar{\omega}^n_{ef/ei} + \bar{\omega}^n_{n/ef}\right)\left\{\vec{v}_{X/\mathcal{F}_{ef}(O)}\right\}^n + \bar{g}^n_l \tag{3.10}$$

The estimated velocity equation form of it is:

$$\left\{D_n\vec{v}_{X/\mathcal{F}_{ef}(O),est}\right\}^n \tag{3.11}$$

$$= \bar{h}^n{}_{est} - SSF\left(2\bar{\omega}^n_{ef/ei,est} + \bar{\omega}^n_{n/ef,est}\right)\left\{\vec{v}_{X/\mathcal{F}_{ef}(O),est}\right\}^n + \bar{g}^n_{l,est}$$

Difference these two equations:

$$\delta\left\{D_n\vec{v}_{X/\mathcal{F}_{ef}(O),est}\right\}^n = \left\{D_n\vec{v}_{X/\mathcal{F}_{ef}(O),est}\right\}^n - \left\{D_n\vec{v}_{X/\mathcal{F}_{ef}(O)}\right\}^n$$

Ignoring coriolis terms and changes in gravity vector reduces the difference equation to [4]:

$$\delta\left\{D_n\vec{v}_{X/\mathcal{F}_{ef}(O),est}\right\}^n = \bar{h}^n\Gamma + \hat{C}^{(b,n)}\delta\bar{h}^b$$

Position can be obtained by simply taking the integral of the velocity term:

$$\delta\dot{p} = \delta\vec{v}_{X/\mathcal{F}_{ef}(O)}$$

3.4.3 State Space Form of Error Propagation

Position, velocity and attitude error equations can be collected in a single equation:

$$\delta\dot{x} = F\delta x + Gu \tag{3.12}$$

In this equation;

$$\delta x = [\delta\alpha \quad \delta\beta \quad \delta\gamma \quad \delta v_N \quad \delta v_E \quad \delta v_D \quad \delta\theta \quad \delta\psi \quad \delta y]^T$$

$$u = [\delta\omega_1 \quad \delta\omega_2 \quad \delta\omega_3 \quad \delta a_1 \quad \delta a_2 \quad \delta a_3 \quad 0 \quad 0 \quad 0]^T$$

If one combines the attitude and the position, velocity error equations matrices F and G are obtained. Writing the equations in this form is crucial to build the Kalman filter.

$$F = \begin{bmatrix} 0 & -\Omega\sin\theta - \frac{\tan\theta\, v_E}{R_0} & \frac{v_N}{R_0} & 0 & \frac{1}{R_0} & 0 & -\Omega\sin\theta & 0 & \frac{-v_E}{R_0^2} \\ \Omega\sin\theta + \frac{\tan\theta\, v_E}{R_0} & 0 & \Omega\cos\theta + \frac{v_E}{R_0} & \frac{1}{R_0} & 0 & 0 & 0 & 0 & \frac{v_N}{R_0^2} \\ -\frac{v_N}{R_0} & -\Omega\cos\theta - \frac{v_E}{R_0} & 0 & 0 & -\frac{\tan\theta}{R_0} & 0 & -\Omega\cos\theta - \frac{v_E}{R_0\cos^2\theta} & 0 & \frac{v_E\tan\theta}{R_0^2} \\ 0 & -a_3 & a_2 & \frac{v_D}{R_0} & -2\Omega\sin\theta - \frac{2\tan\theta\, v_E}{R_0} & \frac{v_N}{R_0} & -v_E\left(2\Omega\cos\theta + \frac{v_E}{R_0\cos^2\theta}\right) & 0 & \frac{v_E^2\tan\theta - v_N v_D}{R_0^2} \\ a_3 & 0 & -a_1 & -2\Omega\sin\theta - \frac{2\tan\theta\, v_E}{R_0} & \frac{\tan\theta\, v_N + v_D}{R_0} & 2\Omega\cos\theta + \frac{v_E}{R_0} & 2\Omega(v_N\cos\theta - v_D\sin\theta) + \frac{v_N v_E}{R_0\cos^2\theta} & 0 & \frac{-v_E(\tan\theta\, v_N + v_D)}{R_0^2} \\ -a_2 & a_1 & 0 & -\frac{2v_N}{R_0} & -2\left(\Omega\cos\theta + \frac{v_E}{R_0}\right) & 0 & 2v_E\Omega\sin\theta & 0 & \frac{v_E^2 + v_N^2}{R_0^2} \\ 0 & 0 & 0 & \frac{1}{R_0} & 0 & 0 & 0 & 0 & \frac{-v_N}{R_0^2} \\ 0 & 0 & 0 & 0 & \frac{1}{R_0\cos\theta} & 0 & \frac{v_E\sin\theta}{R_0\cos^2\theta} & 0 & \frac{-v_E}{R_0^2\cos\theta} \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix}$$

$$G = \begin{bmatrix} & -\hat{C}^{(b,n)} & & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & & & & 0 & 0 & 0 \\ 0 & 0 & 0 & & \hat{C}^{(b,n)} & & 0 & 0 & 0 \\ 0 & 0 & 0 & & & & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

3.5 Digital Compass Technology

Digital compasses measure magnetic field by its 3 axis magnetometers and measures the gravity by 3 axis accelerometers. Blending the data of gravity vector and the magnetic field strength, it calculates the angles between the vector pointing the magnetic north and its own axes. The three angles all together give the orientation of the digital compass.

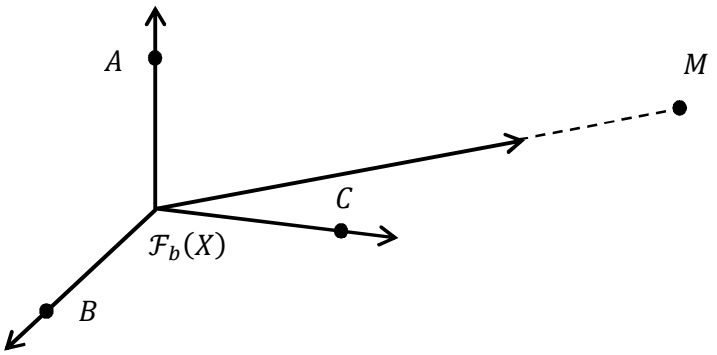


Figure 3.6 Digital compass' axes and the Magnetic North

In the figure, body reference frame is illustrated. Point X is the center of the compass. Point M is the Magnetic North of the Earth. The digital compass outputs three angles between the body frame and the frame formed by $\overrightarrow{XM}$ and gravity vector $\vec{g}$. These angles are called heading, roll and pitch.

Because the compass always changes its orientation with the vehicle; meaning that the vehicle is assumed to be rigid, the mounting location of the compass is not important. The location does not bring errors to the calculations. However, mounting location is important in some other aspect. There are some factors that disrupt or offset Earth' magnetic field. According to [17], these factors are ferrous

metal, electric currents and magnets. Distance from the sensor is critical to the compass' ability to keep an accurate heading.

A disturbed magnetic field is demonstrated in fig. The magnetic field lines are deformed by the ferrous materials and magnets inside the missile. For this missile, the best location to place the compass is the tip section of the missile since the lines are not deformed so much there.

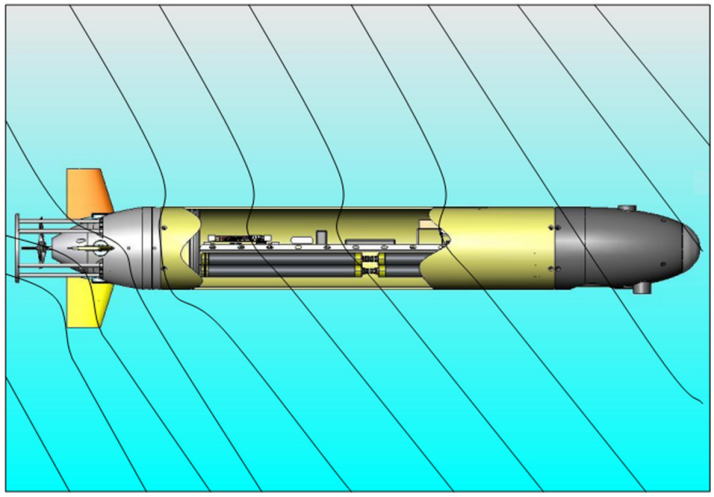


Figure 3.7 The deformation of Earth's magnetic field around a missile [17]

In the manufacturing and assembling of the unmanned tracked vehicle prototype, ferrous metal usage was avoided insofar as one is able. For this reason, the magnetic field disturbance is lessened.

The electric motors are the biggest source of disturbance in the prototype. According to [17], moving the compass more than 10 inches from magnets, removes all of the significant error introduced in the reading. As a consequence, in the design of the unmanned vehicle prototype, the location of the compass is kept

distant to the motors as much as possible. The compass' mounting location is illustrated in fig

3.6 Error Model of Digital Compass

The digital compasses, like all compasses, suffer from declination. According to [21], Magnetic declination, sometimes called magnetic variation, is the angle between magnetic north and true north. Declination is considered positive east of true north and negative when west. Magnetic declination is a function of the location of on Earth.

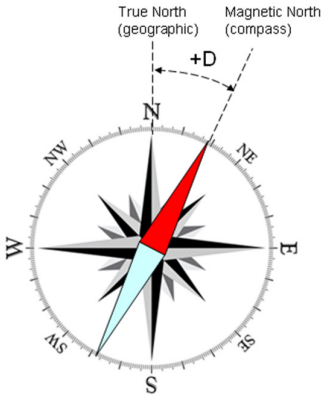


Figure 3.8 Magnetic declination on compass [32]

On the other hand, magnetic inclination which is also a function of the location is the angle between the needle of the compass and the Earth's surface. In order to calculate magnetic declination and inclination angles, there should be a function or a data set that relates the location with the geomagnetic field of the Earth. This data set is the "International Geomagnetic Reference Field" [21]. The model is updated 5 yearly intervals.

Luckily, this model is found in MATLAB/SIMULINK in aerospace block set as a block. This block is used in this thesis in order to find the magnetic inclination and declination angles of the digital compass which seen in navigation algorithm as digital compass bias errors.

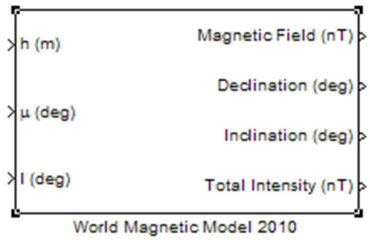


Figure 3.9 MATLAB World Magnetic Model block

3.7 Encoder Technology

Encoder is a device that converts angular displacement to electrical signals (ticks). Counting these signals and multiplying by tick per angle coefficient gives the angular displacement. Absolute encoders output the absolute position. On the other hand, incremental encoders measure the angular displacement from powering the encoder. This means an encoder defines the position as "zero" at the time instant of turning on the power. The encoders used in this thesis are of incremental type.

Encoders output the electrical signals in square wave form in fig. "Lines" or "ticks" per revolution of encoder is the number of period of square wave cycles in 360° of rotation. Resolution of an encoder is the minimum angle it can measure.

Some encoders have 2 electrical signal outputs. One of the square waves is delayed 90°. In this manner, the resolution of the encoder is increased 4 times. In fig, 2 delayed outputs of square waves are shown.

In the unmanned tracked vehicle prototype, 2 encoders are mounted at the end of wheel shafts. By this way, the linear motion of vehicle is measured.
Theoretically encoders do not give sensor noise.

3.8 Kalman Filter

Reference [19] defines the Kalman filter as an estimator for what is called the linear quadratic problem, which is the problem of estimating the instantaneous state of a linear dynamic system perturbed by white noise. The estimator designed for this problem is statistically optimal with respect to any quadratic function of estimation error. The output of a dynamic system has noise on it; because of sensors. However the output should be cleared from these noises. If the noise characteristics and the system model are known a Kalman filter can be deigned.

3.8.1 Impression of Kalman Filter on Technology

Kalman filtering is one of the greatest achievements in estimation theory of the twentieth century proposed by Rudolf Emil Kalman in 1960 [19]. In literature, there are many applications where Kalman filter plays a role in. According to [19], the efficient and precise spacecraft navigation is available with this. In international Space Station and in many missile navigation system algorithms, this filter is utilized. The principle uses of Kalman filtering have been in modern control systems, in the tracking and navigation of all sorts of vehicles and in predictive design of estimation and control systems. Although there are many filtering methods, Kalman filtering has some advantages:

1) The Kalman filter does not require deterministic dynamics and measurements.

2) Kalman filter is available and can easily be applied to digital systems. Not only the filter can be designed as software, but also it operates as an iterative discrete system.
3) It provides the availability of utilizing multiple sensors (multisensor) in order to correct an output.

3.8.2 Kalman Filter Theory

The probability calculation of a dice or a coin is quite straight forward. This kind of probability calculation is discrete. The set of total number of outcomes has finite elements. However think of the possibility of making score putting the ball through the basket in basketball. There are infinitely many possible outcomes. This kind of event is continuous event. For continuous random variables, the probability of any single discrete event is zero. Therefore talking about intervals rather than discrete points is the case. New definitions are supposed to be made for continuous events [22].

Cumulative distribution function: This function represents the cumulative probability of the continuous random variable a for all (uncountable) events up to and including.

$F_A(a) = p(-\infty, a]$

Probability density function: In order to define the probability of continuous events an assistant function. The visualization of this function on graph gives information about the distribution.

$$f_A(a) = \frac{d}{da} F_A(a)$$

The **probability** for an interval can be written as:

$$p_A[x, y] = \int_x^y f_A(a)\, da$$

The **expected value** of a random variable:

$$E(A) = \int_{-\infty}^{\infty} a f_A(a)\, da$$

Variance and **standard deviation**:

$Variance = E(A^2) - E(A)^2$ *and* $\sigma_A = \sqrt{Variance}$

3.8.3 Discrete Kalman Filter Algorithm

Kalman filter estimates the states of a linear system. In this thesis, the system is perturbation of errors. The states are the errors. State equation, in other words, process model for a discrete Kalman filter is:

$$x_{k+1} = Ax_k + Bu_k + w_k \tag{3.13}$$

The output equation or the measurement model is:

$$y_k = Cx_k + z_k \tag{3.14}$$

k is the time index, A is the state transition matrix, B is the control input, x is the state vector, w is the process noise, C is the output matrix and z is the measurement noise.

In section 3.4.3, the error propagation equation was illustrated as:

$$\delta\dot{x} = F\delta x + Gu \tag{3.15}$$

Solve the differential equation and write in discrete form as:

$$\delta x_{k+1} = \Phi_k \delta x_k + w_k \tag{3.16}$$

where $\Phi_k = exp[F(t_{k+1} - t_k)]$

The state transition matrix for the mechanization equations becomes:

$$A = exp[F(t_{k+1} - t_k)] \tag{3.17}$$

The state transition matrix is going to be updated at each time step in navigation equations solver algorithm. The system matrices are used in Kalman filter equations as:

Prediction session algorithm:

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k \quad (3.18)$$

$$P_k^- = Q + AP_{k-1}A^T + Q \quad (3.19)$$

Measurement correction session algorithm:

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \quad (3.20)$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \quad (3.21)$$

$$P_k = (1 - K_k H)P_k^- \quad (3.22)$$

The time update projects the current state estimate ahead in time. The measurement update renews the projected estimate by sensor measurements [22].

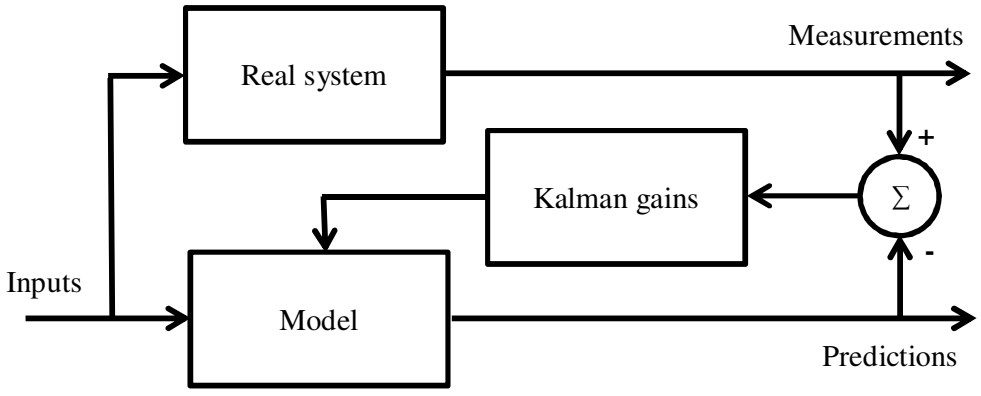


Figure 3.10 Kalman filter signal flow diagram

3.9 Sensor Fusion

In order to understand the sensor fusion, phenomenon combination of independent estimates is supposed to be comprehended.

Suppose that a dimension s is going to be measured by two different methods. The measurements by the methods are s_1 and s_2, on the other hand, the variances are σ_1^2 and σ_2^2. It is desired to unite the two measurements so that the variance of the resultant estimate will be minimum. The resultant estimate will be:

$$\hat{x} = W_1 s_1 + W_2 s_2 \tag{3.23}$$

The expected value comes out as:

$$E(\hat{x}) = W_1 E(s_1) + W_2 E(s_2) \tag{3.24}$$

Variance of x can be written as

$$\sigma^2 = E\left[\{W_1 s_1 + W_2 s_2 - W_1 E(s_1) - W_2 E(s_2)\}^2\right] \tag{3.25}$$

Since s_1 and s_2 are independent and $s_1 - E(s_1)$ and $s_2 - E(s_2)$ are uncorrelated, $E\{(s_1 - E(s_1))(s_2 - E(s_2))\} = 0$.

$$\therefore \sigma^2 = W_1^2 \sigma_1^2 + W_2^2 \sigma_2^2 \tag{3.26}$$

W is the weighting coefficient. Therefore, $W_1 = 1 - W_2$.

$$\sigma^2 = (1 - W)^2 \sigma_1^2 + W^2 \sigma_2^2 \tag{3.27}$$

Differentiate the equation above and equate to zero in order to solve for the minimum variance.

$$\sigma^2 = \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2} \tag{3.28}$$

As a result, it is obvious that, the resultant variance of fusion is smaller than any of the variance of the measurements. This means in sensor fusion the resultant measurement is more accurate than any of the measurements.

3.9.1 Sensor Fusion Algorithms

3.9.1.1 Feed Forward Sensor Fusion

In this method, Kalman filter estimations (Kalman outputs) are subtracted from the IMU mechanization equation outputs. This is because Kalman filter system equations are based on IMU error model.

For error accumulation of integrals of the mechanization equations, nothing is done. Since sensor data sent rates are different and the highest data sent frequency is the IMU's when there is no data change in compass and encoders, IMU navigation is done. When there is new data from compass and encoders, the Kalman filter is activated and sensor fusion is implemented at those time steps.

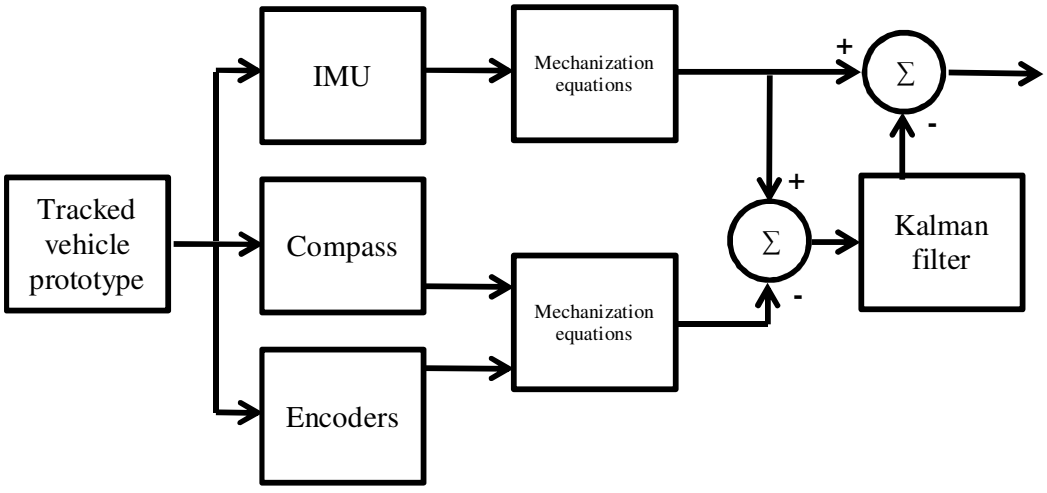


Figure 3.11 Kalman filter in feed forward sensor fusion

3.9.1.2 Feedback Sensor Fusion

For error accumulation of integrals of the mechanization equations, previous state variables are updated. Since sensor data sent rates are different and the highest data sent frequency is the IMU's when there is no data change in compass and encoders, IMU navigation is done. When there is new data from compass and encoders, the Kalman filter is activated and sensor fusion is implemented at those time steps.

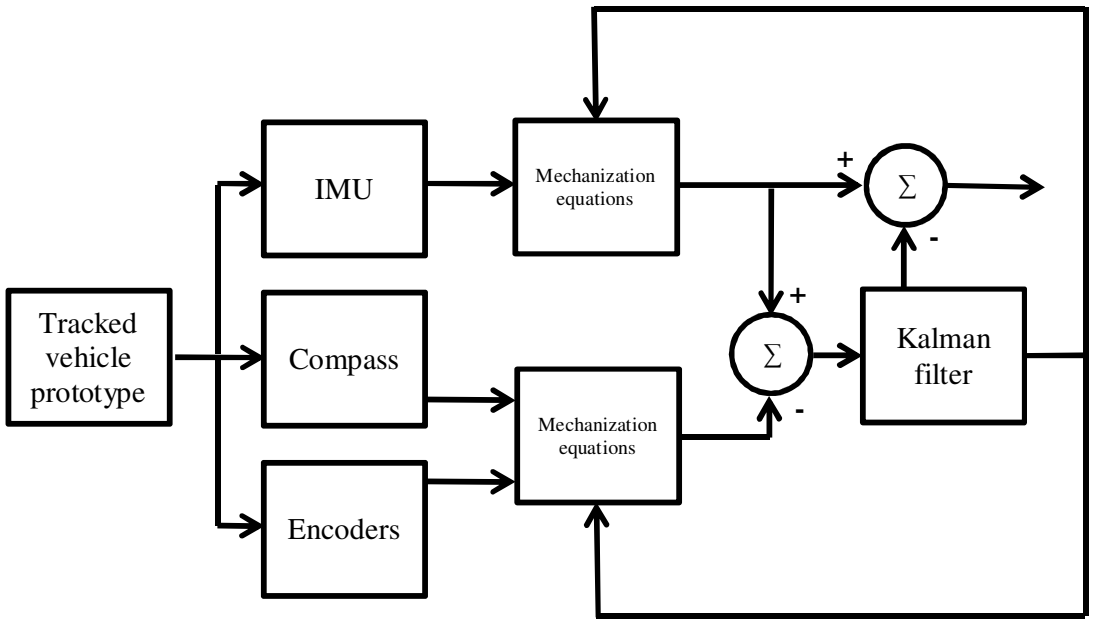


Figure 3.12 Kalman filter in feedback sensor fusion

CHAPTER 4

NAVIGATION EQUATIONS SOLVER

4.1 Simulation Model

Navigation equations solver is the simulation used for solving the dead reckoning navigation equations if IMU data is entered into simulation. The solver is a MATLAB/SIMULINK model and it is direct implementation of "discretization of navigation equations".

4.2 Test of the Simulation

The test of the simulation can be done by giving inputs by hand. This means, the data not from an IMU but a synthetic one is fed into the simulation. By this way the outputs of the simulation can be compared with the calculated ones.

4.3 The Synthetic Inputs

The first synthetic input to the simulation is constant acceleration on a single axis. The code for the input is below:

```
omegax = 0; % [rad/s]
omegay = 0; % [rad/s]
omegaz = 0; % [rad/s]
fx = 0.1; % [m/s^2]
fy = 0; % [m/s^2]
fz = -9.7987; % [m/s^2]
```

The simulation was run for 5 seconds and the simulation sampling frequency is 1000 Hz. Simply, the expected output is

$\Delta x = 0.5 f_x t^2 = 1.25m$ and $\Delta y = 0$.

The simulation outputs are as follows:

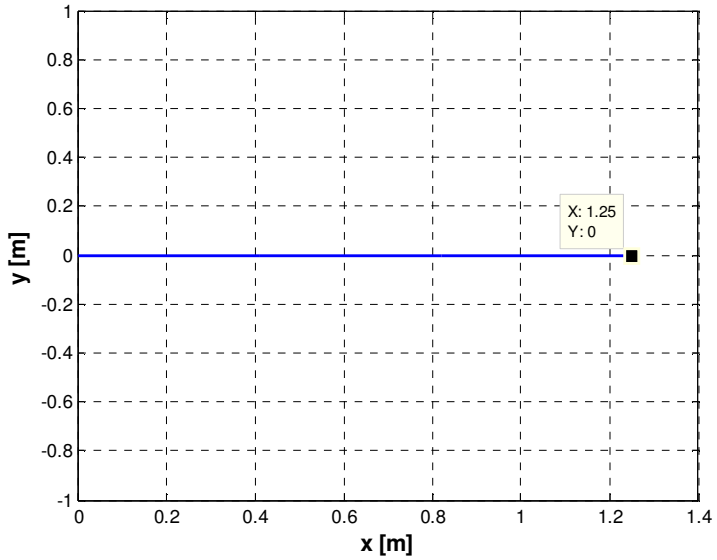


Figure 4.1 The navigation equation solver output for synthetic input 1

The second input to the system can be thought as a mission for the unmanned vehicle prototype. This mission requires going straight to point A with constant acceleration and start to turn about point O for 10 times. The result will be combination of inputs. This input tests the direction cosine matrix update, the orthogonalization and normalization algorithms. If there is something wrong about these algorithms the test result mapping is going to heavily diverge from the expected one.

Point A is located at ;

$\Delta y = 0.5 f_y t^2 = 0.5 \times \pi \times 1 = 1.5708m$

The mission requires to turn about point O which is 1m distant to point A on horizontal with a turn rate of $\frac{\pi}{2}$ rad/s. While doing this rotation, the orientation of the vehicle is always tangent to the path and the vehicle's front is on the velocity direction. The exact solution to the input where the vehicle arrives is point B located at $\left(0,\frac{\pi}{2}\right)$.

The code for the input is below:

```
omegax = 0; % [rad/s]
omegay = 0; % [rad/s]
omegaz = 0; % [rad/s]
fx = pi; % [m/s^2]
fy = 0; % [m/s^2]
if clock > 1
    fx = 0; % [m/s^2]
    fy = pi^2; % [m/s^2]
    omegaz = pi; % [rad/s]
end
fz = -9.7987; % [m/s^2]
```

The simulation outputs are as follows:

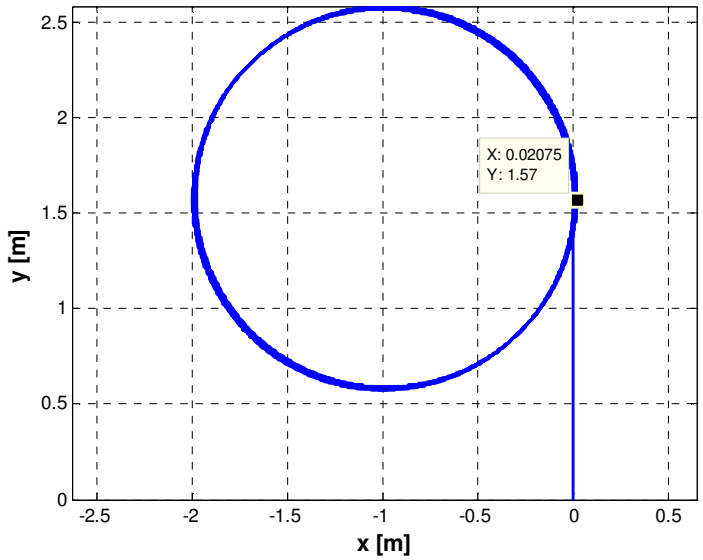


Figure 4.2 The navigation equation solver output for synthetic input 2

A mixture of inputs to the system is much more probable in an unmanned vehicle mission. For this reason, a mixed input is fed to the system. The input code is below:

```
omegax = 0; % [rad/s]
omegay = 0; % [rad/s]
omegaz = 0; % [rad/s]
fx = pi; % [m/s^2]
fy = 0; % [m/s^2]
if clock > 1
    fx = 0; % [m/s^2]
    fy = pi^2; % [m/s^2]
    omegaz = pi; % [rad/s]
end
fz = -9.7987; % [m/s^2]

omegax = 0; % [rad/s]
omegay = 0; % [rad/s]
omegaz = 0; % [rad/s]
fx = pi; % [m/s^2]
fy = 0; % [m/s^2]
fz = -9.7987; % [m/s^2]
if clock > 1
    omegaz = pi; % [rad/s]
    fx = 0; % [m/s^2]
    fy = pi^2; % [m/s^2]
end
if clock > 1.5
    omegaz = 0; % [rad/s]
    fy = 0; % [m/s^2]
end
if clock > 2.5
    omegaz = pi; % [rad/s]
    fy = pi^2; % [m/s^2]
end
if clock > 3
    omegaz = 0; % [rad/s]
    fy = 0; % [m/s^2]
```

```
end
if clock > 4
    omegaz = -pi; % [rad/s]
    fy = -pi^2; % [m/s^2]
end
if clock > 4.5
    omegaz = 0; % [rad/s]
    fy = 0; % [m/s^2]
end
```

The simulation output is:

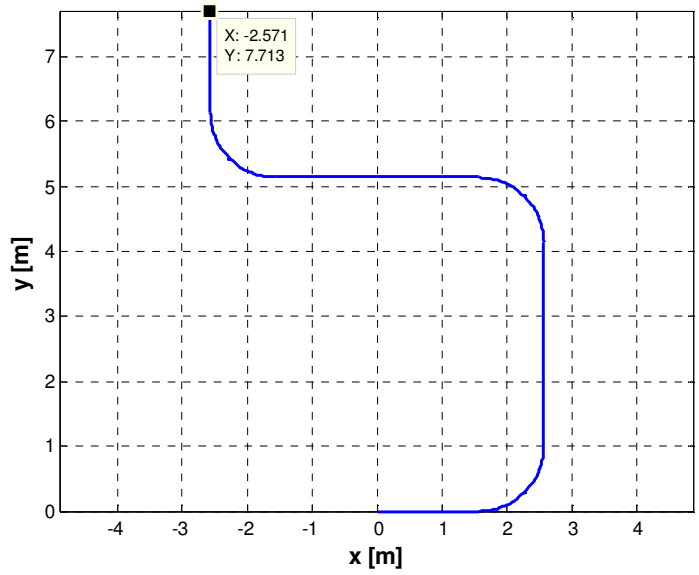


Figure 4.3 The navigation equation solver output for synthetic input 3

The exact solution is:

$\Delta x = -\frac{\pi}{2} - 1 = 2.5708m$

$\Delta y = 3 \times 1 + 1.5 \times \pi = 7.7124m$

4.4 Discussion of Results

The simulation outputs and the exact solutions are very close to each other in each case .In the first and the third case; the results nearly cannot be distinguished from the exact. However for the second synthetic input, the error is larger. This is because the simulation is not continuous in time but discrete. If sample frequency of the simulation is increased to 10000Hz, then the error becomes smaller. As a result, the navigation equations solver works and finds the mechanization equations' solution. However, like all discrete simulations, the sampling frequency is supposed to be maximized as much as possible.

CHAPTER 5

TEST SETUP AND SYSTEM MODEL

5.1 MATLAB/SIMULINK Model

5.1.1 Initialization Model

Before the test, a special designed simulation is run and the sensor outputs are stored in the computer. Because of mounting misalignments, the IMU has some bias values. For example, when the compass says that the vehicle is parallel to ground, from two accelerometers there should be zero readings, from the third accelerometer, there should be plumb bob gravity only. In practice this not the case. Because the compass' and IMU's orientation may not be the same. There may be small angular misalignments. Therefore this misalignment is supposed to be calculated and subtracted from the calculations as test continuous.

5.1.2 Real Time Simulation Model

The real time simulation model is the top level simulation model and there are some sub simulations in this model. The xPC to Target block is responsible for data acquisition. The encoders + compass block solves the mechanization equations for the outputs of compass and encoders. In Kalman block, the algorithm for the Kalman filter exists. Navigation equations solver includes the mechanization equations explained in previous chapters. Finally, after finding the errors, these errors are subtracted from the NES outputs as the corrected outputs.

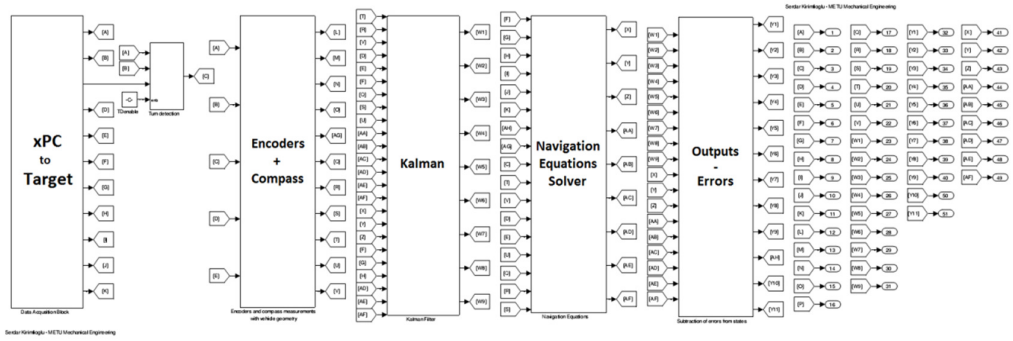


Figure 5.1 Real time simulation model

5.1.3 Data Acquisition

In this thesis, real time data acquisition and simulation run is accomplished. For this, MATLAB's xPC Target is utilized. [25] xPC Target is a solution for prototyping, testing, and deploying real-time systems using standard PC hardware. It is an environment that uses a target PC, separate from a host PC, for running real-time applications.

Real time windows target is another application of MATLAB utilized for similar purpose. Since this application uses, one computer that has an operating system, the simulation does not run real time. In addition to this, simulations consuming high computer performance cannot be run. Therefore RTWT is not applicable for this task.

In xPC Target application there is a target computer where the simulation is built in and booted with a boot disc. Since there is no operating system working on the background, the target computer can use most of its performance on running built in MATLAB simulations.

The host computer is capable of writing the codes for the simulation. After the simulation is written, it should be loaded to the target computer. In this thesis, host and target communicate with TCP/IP network connection. For host computer, standard mother board network card, for the target, Edimax EN-9130TX network card was used.

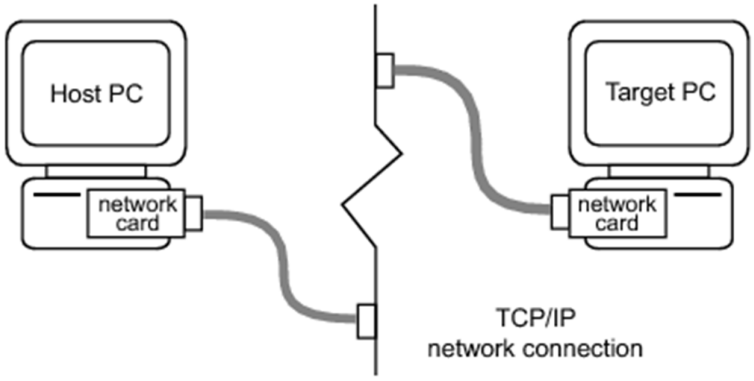


Figure 5.2 Host and target computers communicating with network cable [25]

The digital compass and the IMU send the measurement data by RS232 communication. The compass outputs the ticks as square waves. There is no communicating output. Therefore an encoder reading module and a RS232 communication module was designed. These two tasks were accomplished by DSPic. As a result, all sensors send their data by RS232 communication. The data is read with Quatech QSC100 serial card in target computer.

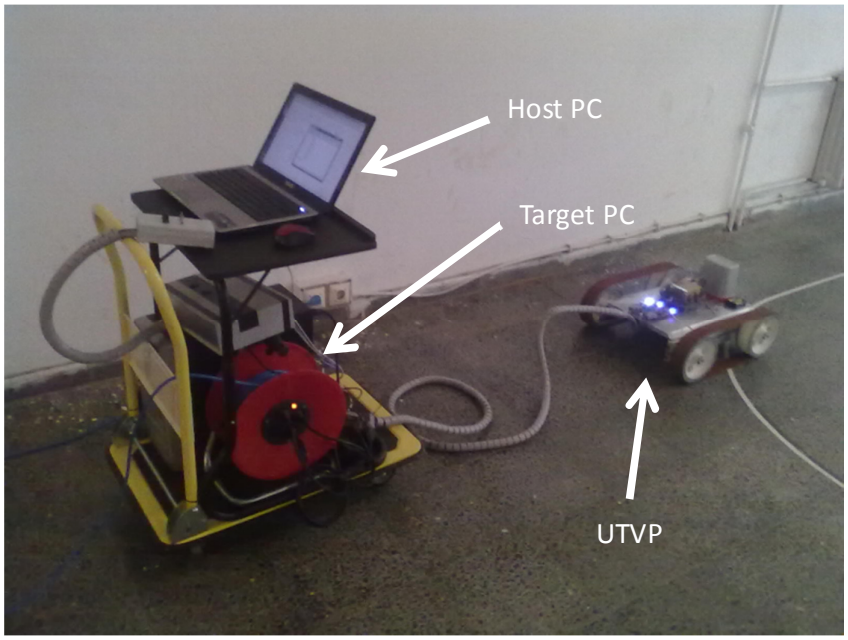


Figure 5.3 The overall test system

5.2 Unmanned Tracked Vehicle Prototype

The unmanned vehicle prototype is capable of carrying the sensors and its tracks angular velocity can be controlled by a manual controller. It allows manual control of DC motor voltage and its polarity by an H bridge.

The power unit is used for supplying the voltages to the vehicle .There are an electronic card on the vehicle. This card takes the sensor outputs and converts them into RS232 communication protocol signals. There is also a regulator to supply $5V$ to the dsPIC's on the electronic card.

The functioning of the electronic card, control unit and power unit are all illustrated in figure 5.9.

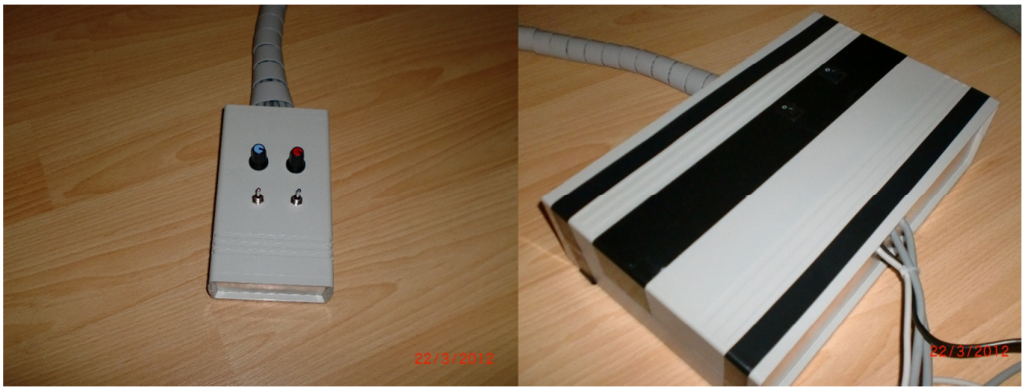


Figure 5.4 The control unit and the power unit of the vehicle

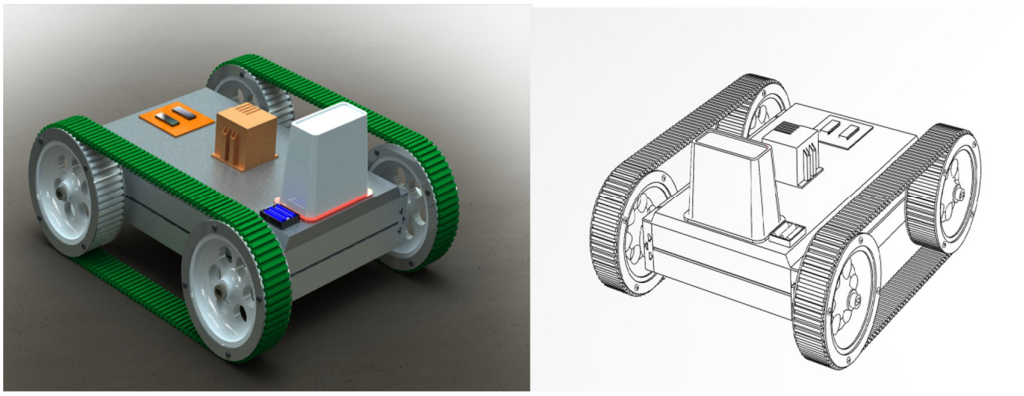


Figure 5.5 3D isometric view of the unmanned tracked vehicle prototype

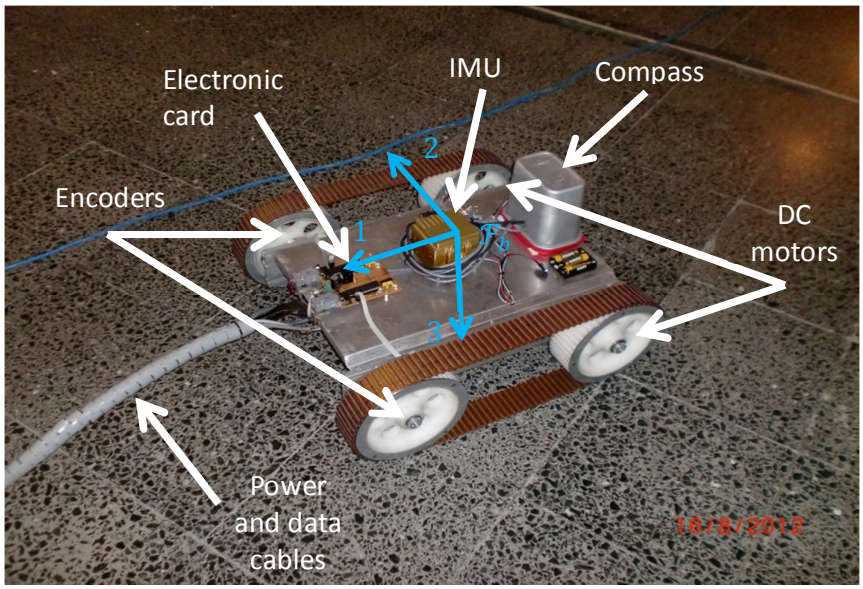


Figure 5.6 Tracked vehicle prototype in test

5.3 Sensors

5.3.1 Encoders

A 2 channel magnetic encoder was used in this thesis. The encoder is Faulhaber IE2-512. Its resolution is $2048 ticks/rev$ in quadrature mode. Total of 2 encoders measuring the displacements of each tracks were mounted on the rear wheels. Further information is given in appendix.

5.3.2 IMU

A strapdown IMU was used in this thesis. The IMU used is Crossbow 440 including 6 DOF MEMS inertial sensor cluster. Further information is given in appendix.

Figure 5.7 The IMU and the body frame attached to it [16]

5.3.3 Compass

A digital compass was used in this thesis. The compass used is Ocean Server OS5000. This is a digital compass including 3 magnetometers and 3 accelerometers and a 24 bit analog to digital converter. Accelerometers aid magnetometers to obtain precise measurements. The compass is powered by $4.5V$ and sends the data by RS232 interface. Further information is given in appendix.

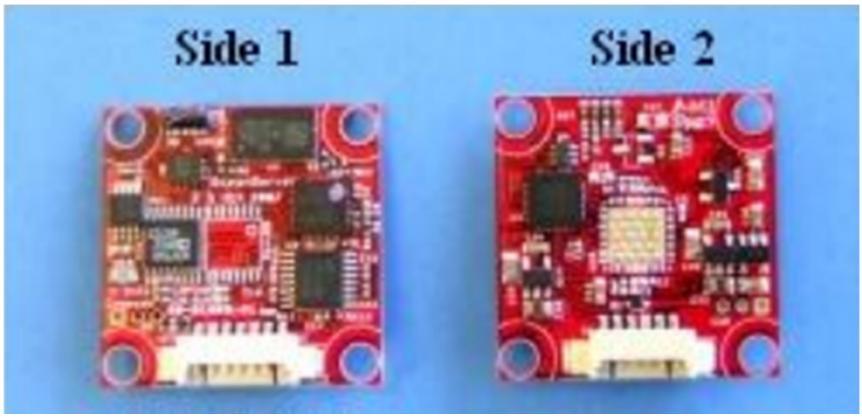


Figure 5.8 The Compass [17]

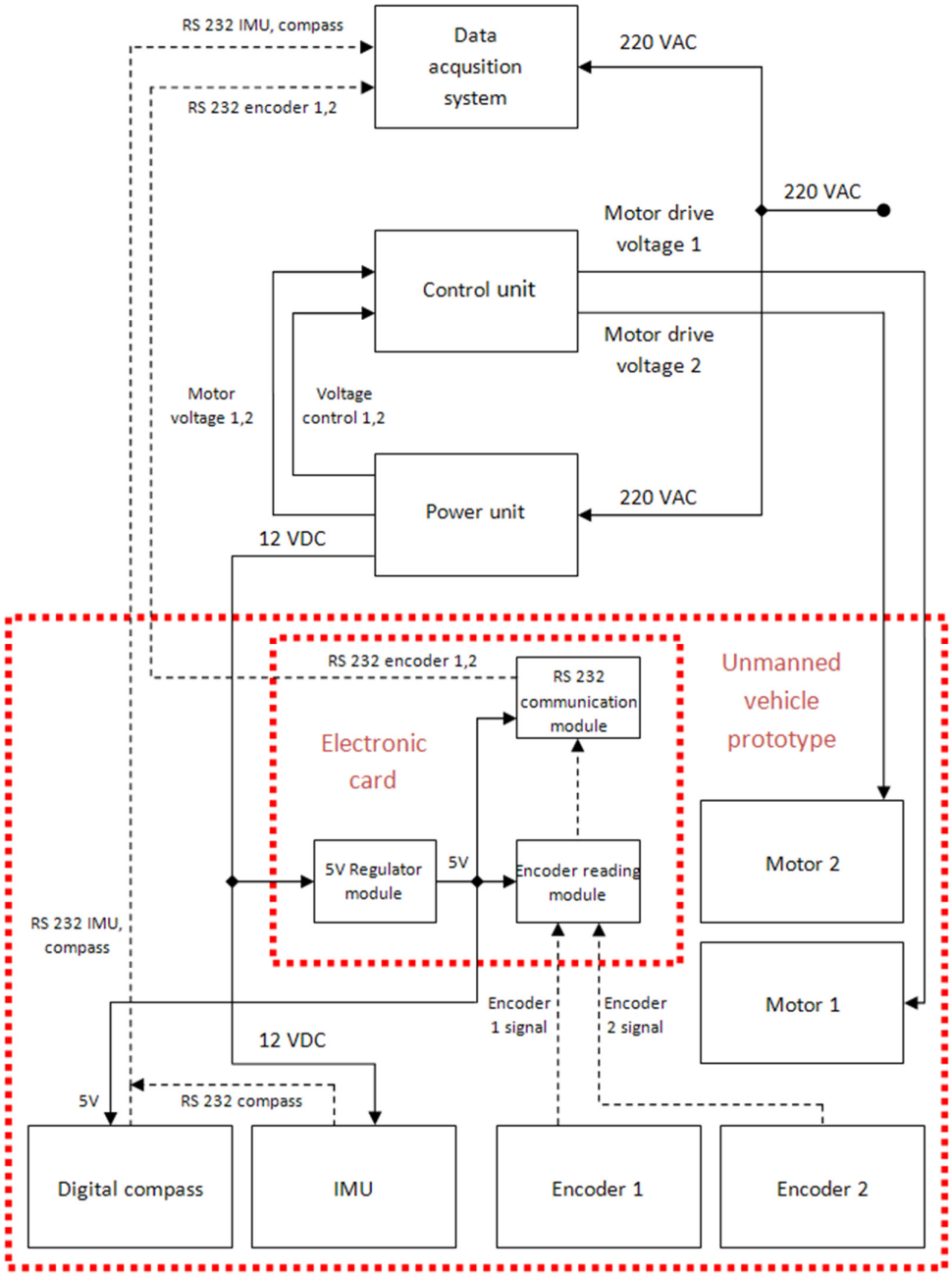


Figure 5.9 The functioning of the electronic card, control unit and power unit

CHAPTER 6

THE TESTS

6.1 Test Procedure

In this chapter, the results of the tests of the unmanned vehicle prototype are given. The tests cover a number of paths that has been mapped. These paths are:

- Staying at a stationary point
- Moving on a straight line: Moving straight between two points
- Sharp steering: The vehicle alters its orientation at a point.
- Smooth steering: The vehicle changes its orientation while moving on a curved path.
- Complicated motion: The vehicle moves on a map while making the motions described above in a planned order.

For each motion strategy, a sensor fusion algorithm was tested. In addition to this, a choice variable was added to the algorithm in order to test the navigation equations solver algorithm in the absence of digital compass and encoders (sole IMU). The navigation sensor fusion algorithms for which the test outputs were recorded are:

- The unaccompanied IMU case
- Feed forward multisensor fusion case
- Feedback multisensor fusion case

6.2 Test Initials

6.3 Sensor Outputs for the Paths

6.3.1 Sensor Outputs for Staying at a Stationary Point

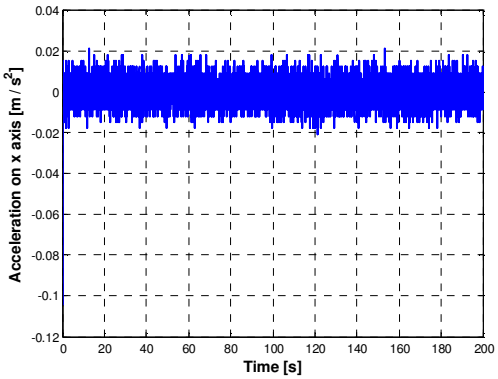


Figure 6.1 The acceleration data taken from x axis accelerometer of IMU with respect to time

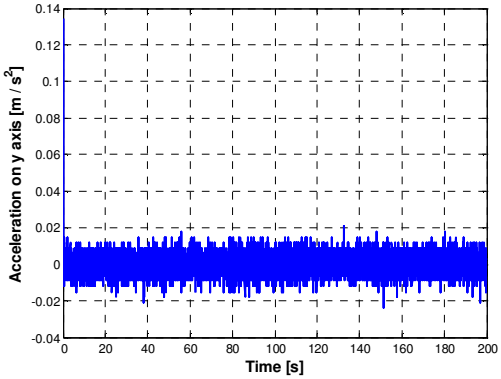


Figure 6.2 The acceleration data taken from y axis accelerometer of IMU with respect to time

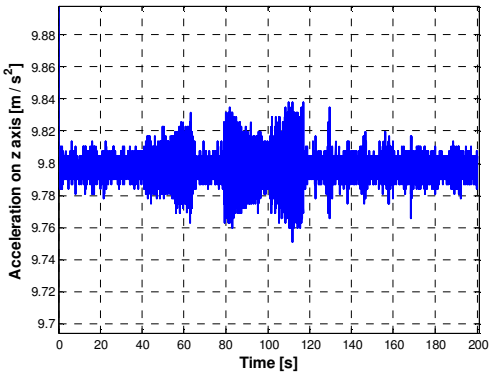


Figure 6.3 The acceleration data taken from z axis accelerometer of IMU with respect to time

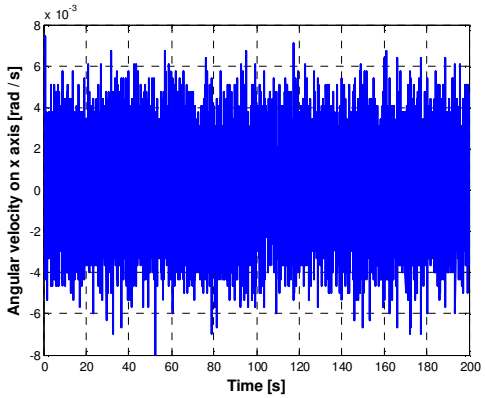


Figure 6.4 The angular velocity data taken from x axis gyroscope of IMU with respect to time

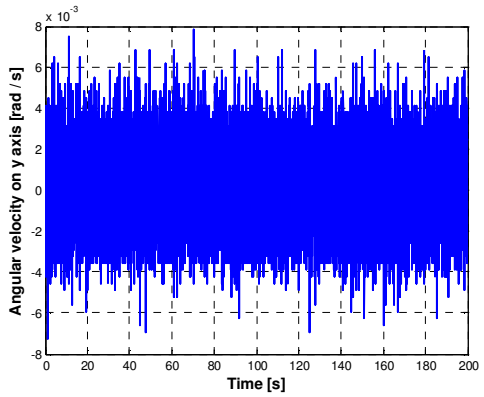


Figure 6.5 The angular velocity data taken from y axis gyroscope of IMU with respect to time

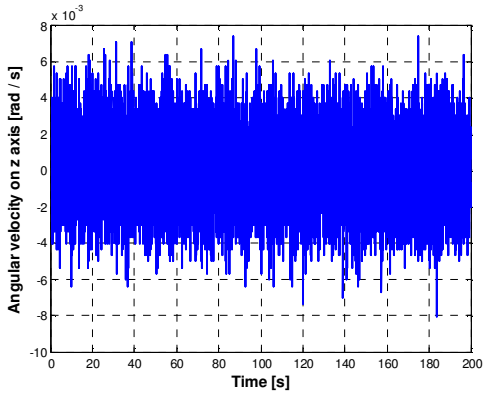


Figure 6.6 The angular velocity data taken from z axis gyroscope of IMU with respect to time

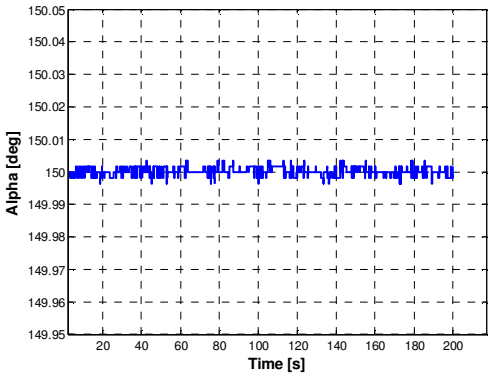


Figure 6.7 The orientation angle data α taken from digital compass with respect to time

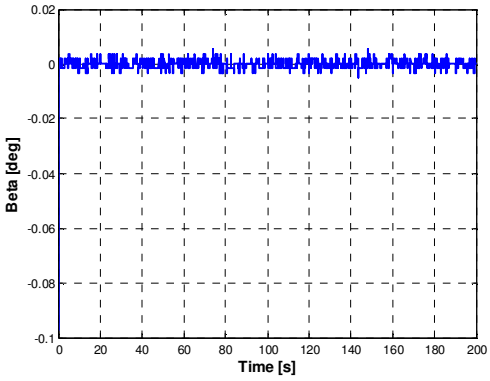


Figure 6.8 The orientation angle data β taken from digital compass with respect to time

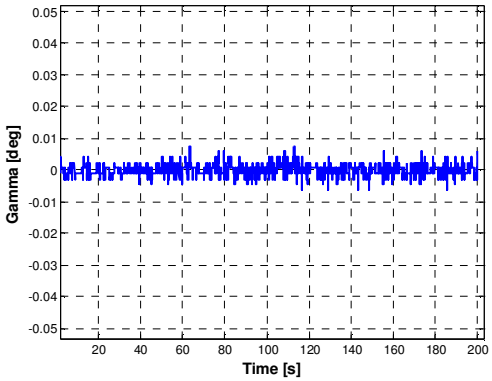


Figure 6.9 The orientation angle data γ taken from digital compass with respect to time

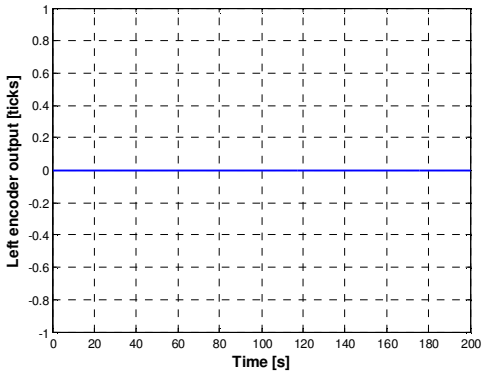


Figure 6.10 The left encoder output data with respect to time

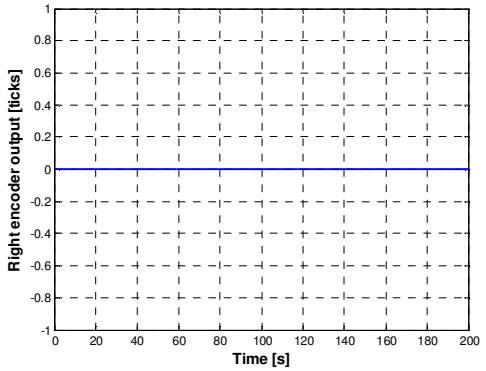


Figure 6.11 The right encoder output data with respect to time

6.3.2 Sensor Outputs for Moving on a Straight Line

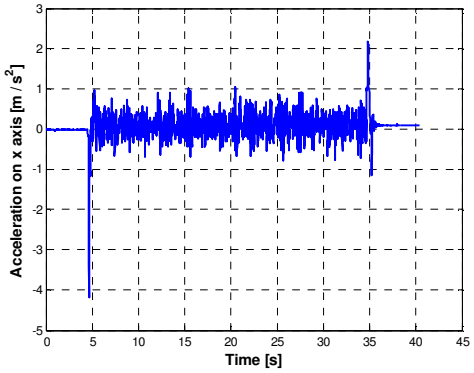


Figure 6.12 The acceleration data taken from x axis accelerometer of IMU with respect to time

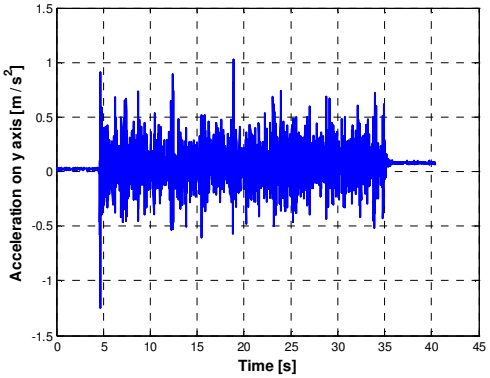


Figure 6.13 The acceleration data taken from y axis accelerometer of IMU with respect to time

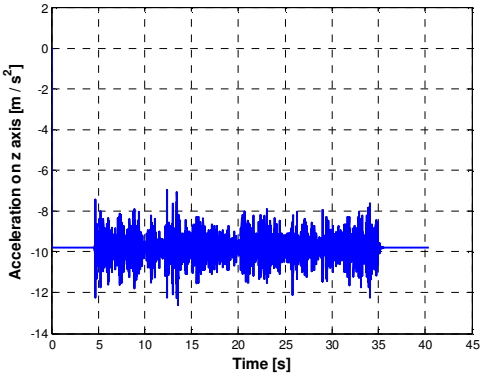


Figure 6.14 The acceleration data taken from z axis accelerometer of IMU with respect to time

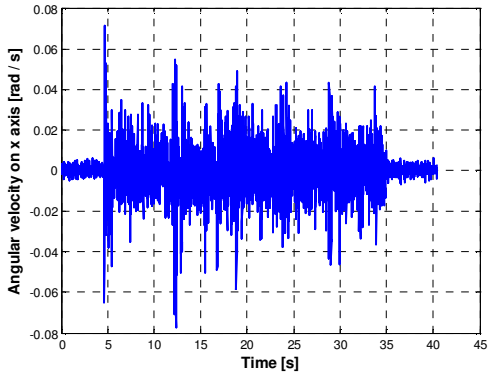


Figure 6.15 The angular velocity data taken from x axis gyroscope of IMU with respect to time

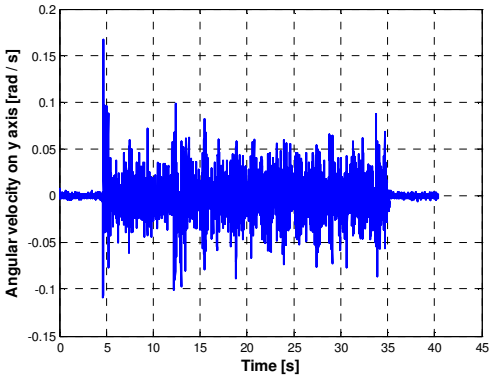


Figure 6.16 The angular velocity data taken from y axis gyroscope of IMU with respect to time

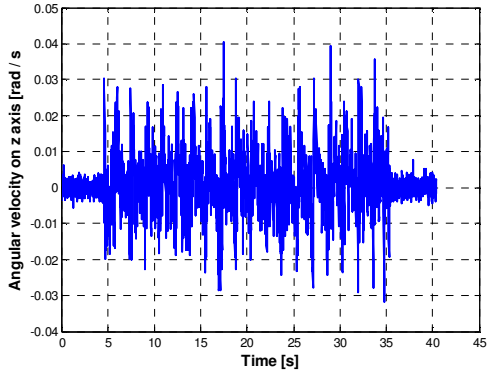


Figure 6.17 The angular velocity data taken from z axis gyroscope of IMU with respect to time

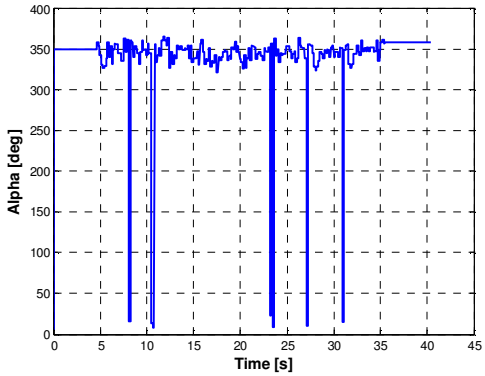


Figure 6.18 The orientation angle data α taken from digital compass with respect to time

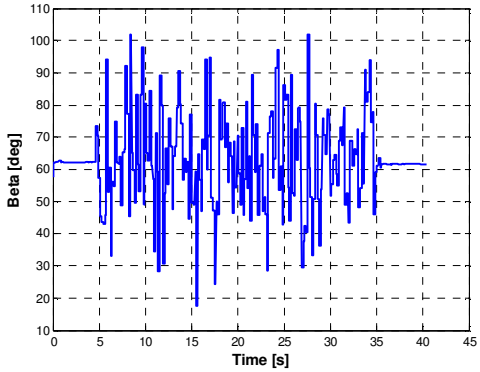


Figure 6.19 The orientation angle data β taken from digital compass with respect to time

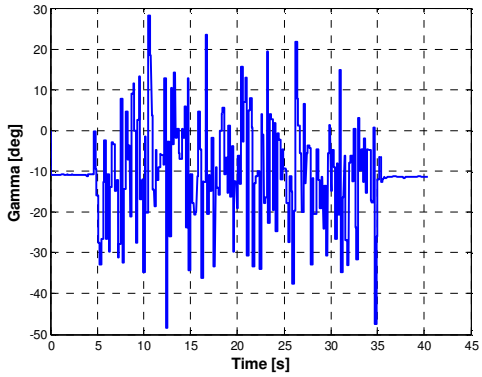


Figure 6.20 The orientation angle data γ taken from digital compass with respect to time

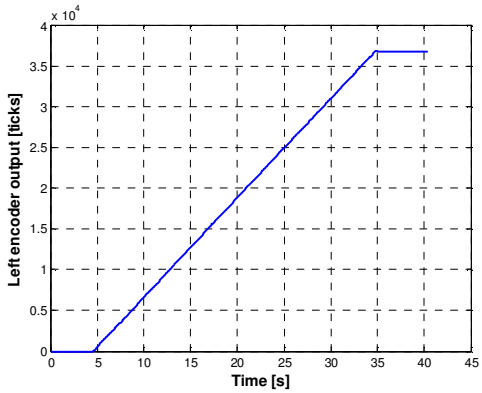


Figure 6.21 The left encoder output data with respect to time

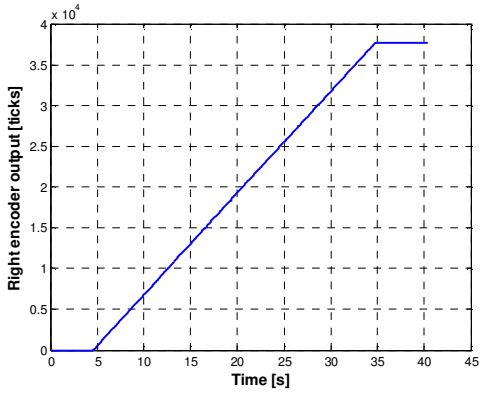


Figure 6.22 The right encoder output data with respect to time

6.3.3 Sensor Outputs for Sharp Steering

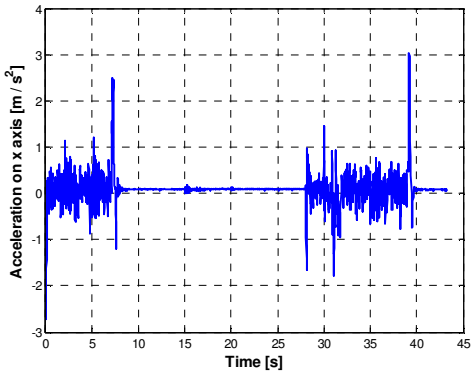


Figure 6.23 The acceleration data taken from x axis accelerometer of IMU with respect to time

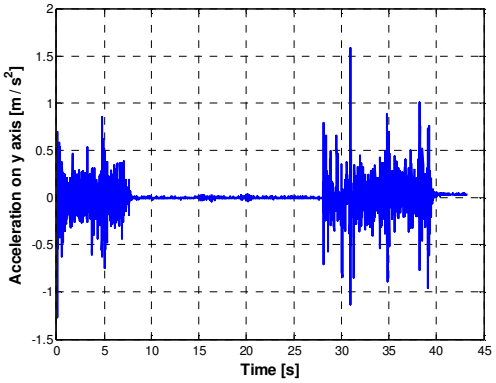


Figure 6.24 The acceleration data taken from y axis accelerometer of IMU with respect to time

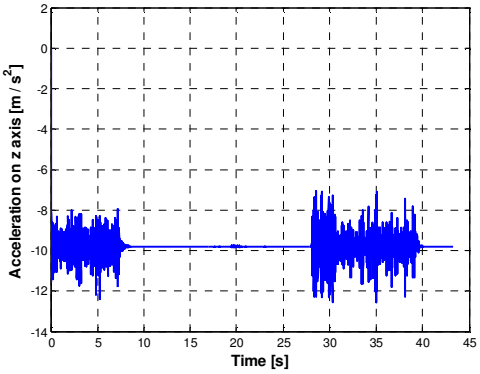


Figure 6.25 The acceleration data taken from z axis accelerometer of IMU with respect to time

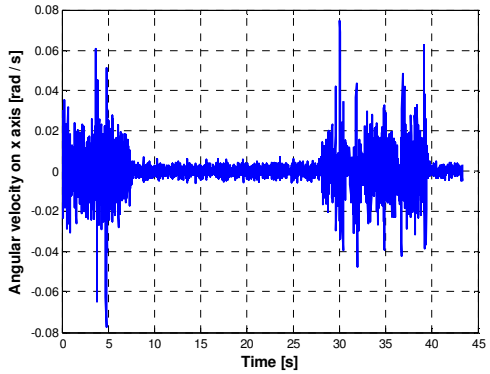


Figure 6.26 The angular velocity data taken from x axis gyroscope of IMU with respect to time

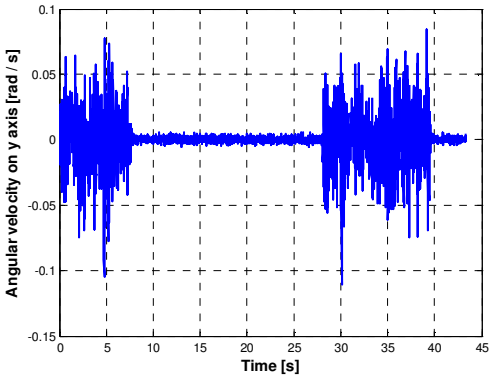


Figure 6.27 The angular velocity data taken from y axis gyroscope of IMU with respect to time

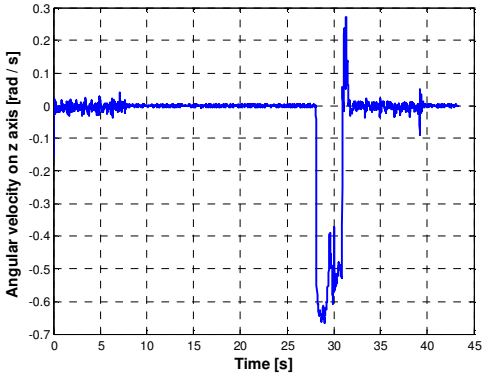


Figure 6.28 The angular velocity data taken from z axis gyroscope of IMU with respect to time

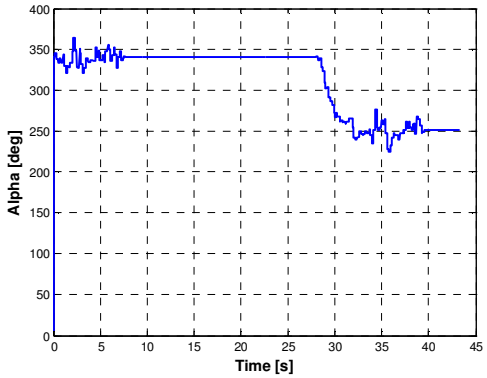


Figure 6.29 The orientation angle data α taken from digital compass with respect to time

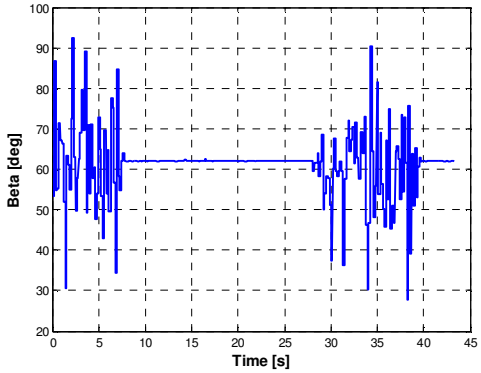


Figure 6.30 The orientation angle data β taken from digital compass with respect to time

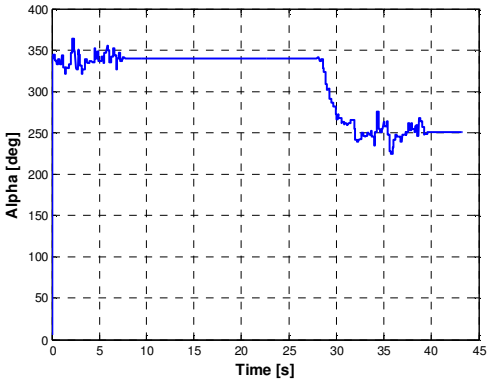


Figure 6.31 The orientation angle data γ taken from digital compass with respect to time

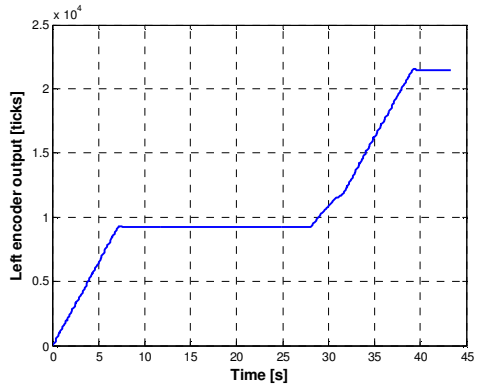


Figure 6.32 The left encoder output data with respect to time

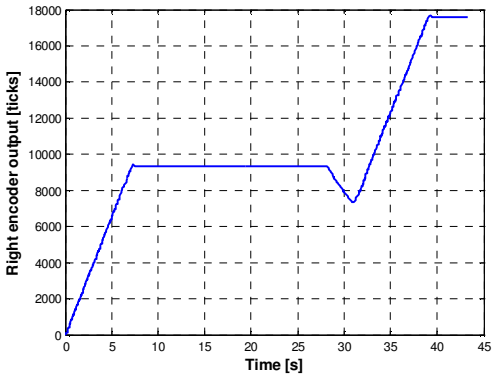


Figure 6.33 The right encoder output data with respect to time

6.3.4 Sensor Outputs for Smooth Steering

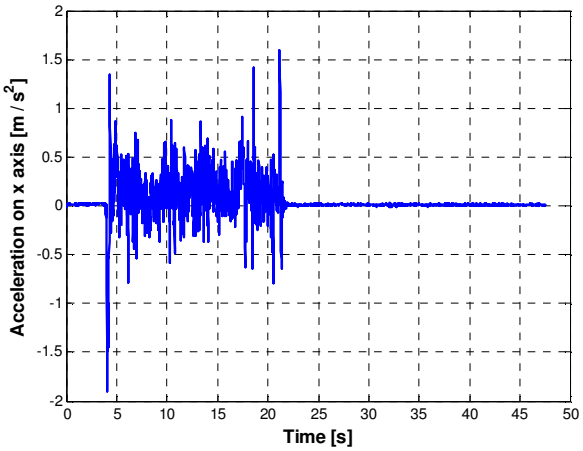


Figure 6.34 The acceleration data taken from x axis accelerometer of IMU with respect to time

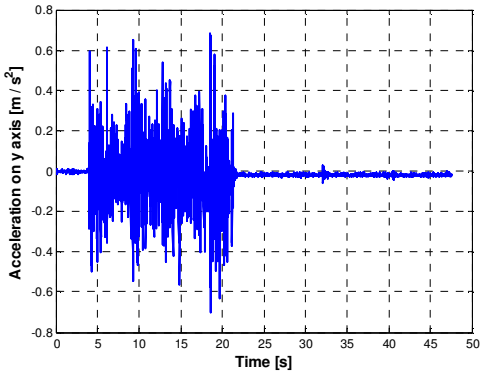


Figure 6.35 The acceleration data taken from y axis accelerometer of IMU with respect to time

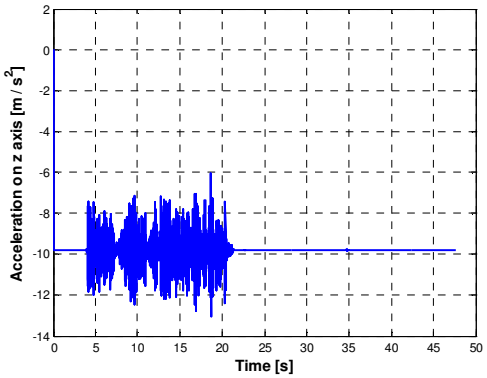


Figure 6.36 The acceleration data taken from z axis accelerometer of IMU with respect to time

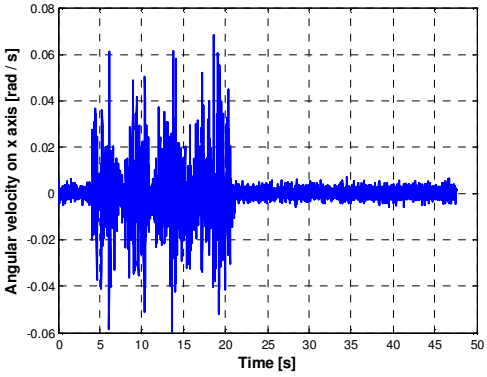


Figure 6.37 The angular velocity data taken from x axis gyroscope of IMU with respect to time

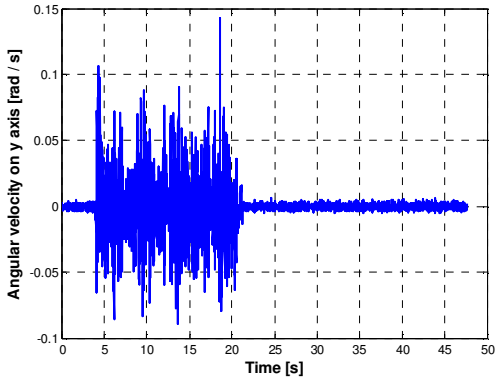


Figure 6.38 The angular velocity data taken from y axis gyroscope of IMU with respect to time

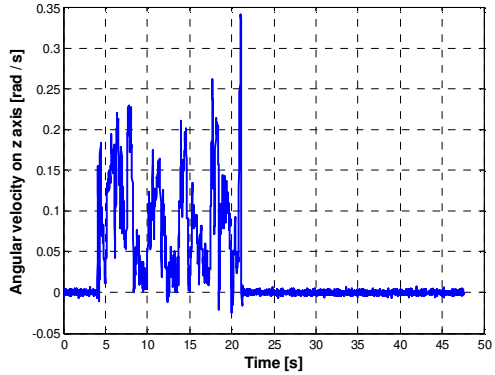


Figure 6.39 The angular velocity data taken from z axis gyroscope of IMU with respect to time

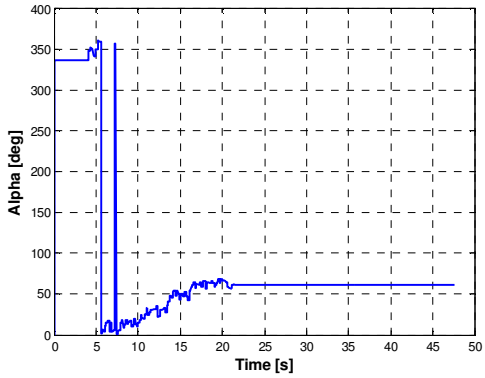


Figure 6.40 The orientation angle data α taken from digital compass with respect to time

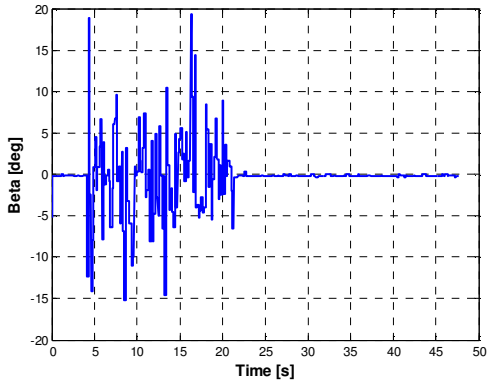


Figure 6.41 The orientation angle data β taken from digital compass with respect to time

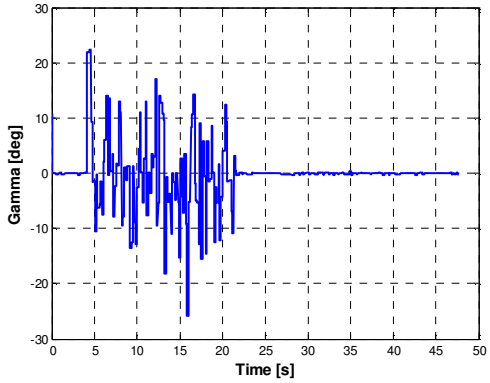


Figure 6.42 The orientation angle data γ taken from digital compass with respect to time

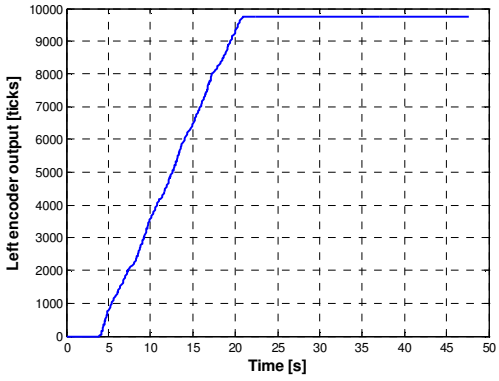


Figure 6.43 The left encoder output data with respect to time

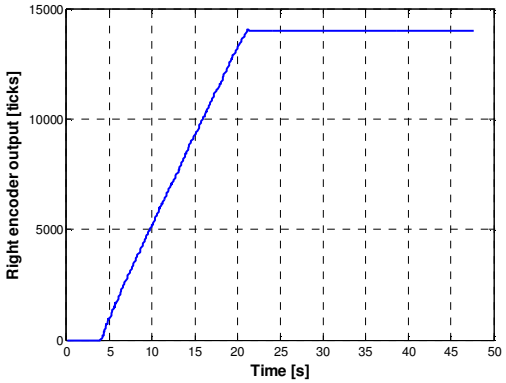


Figure 6.44 The right encoder output data with respect to time

6.3.5 Sensor Outputs for Making Complicated Motion

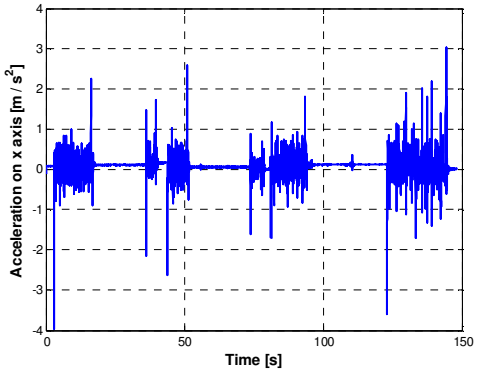


Figure 6.45 The acceleration data taken from x axis accelerometer of IMU with respect to time

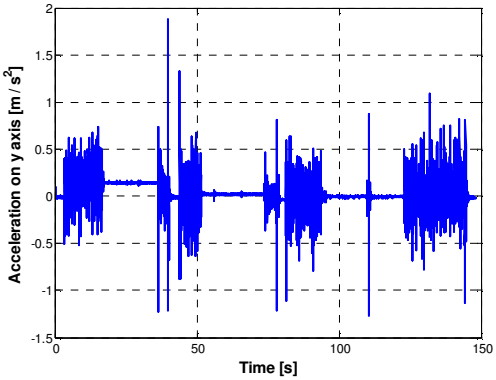


Figure 6.46 The acceleration data taken from y axis accelerometer of IMU with respect to time

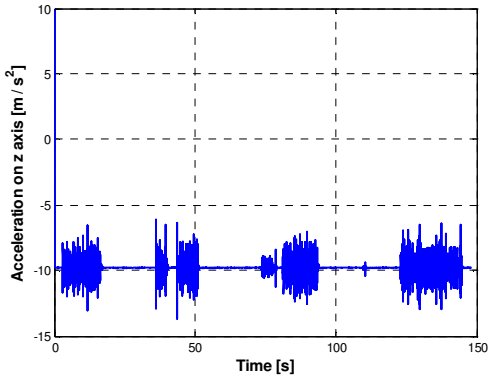


Figure 6.47 The acceleration data taken from z axis accelerometer of IMU with respect to time

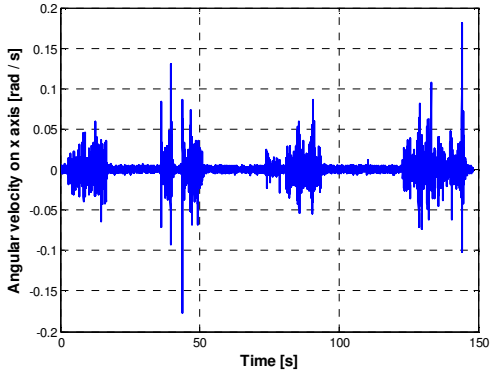


Figure 6.48 The angular velocity data taken from x axis gyroscope of IMU with respect to time

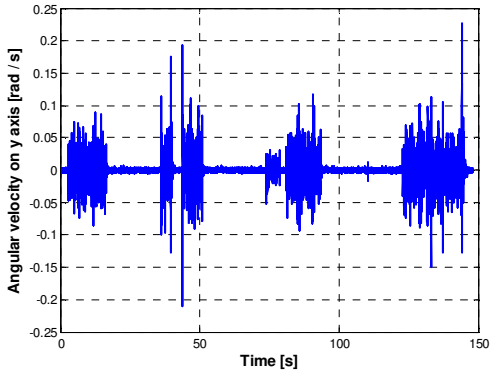


Figure 6.49 The angular velocity data taken from y axis gyroscope of IMU with respect to time

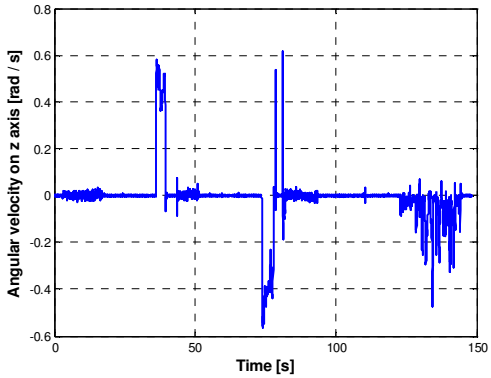


Figure 6.50 The angular velocity data taken from z axis gyroscope of IMU with respect to time

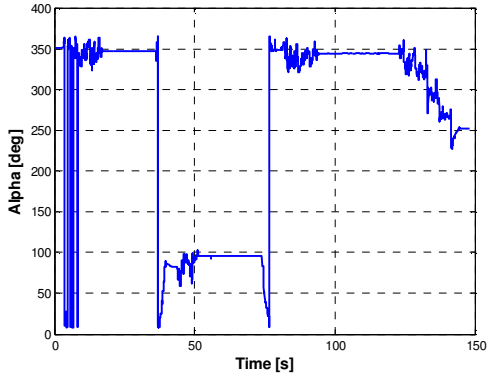


Figure 6.51 The orientation angle data α taken from digital compass with respect to time

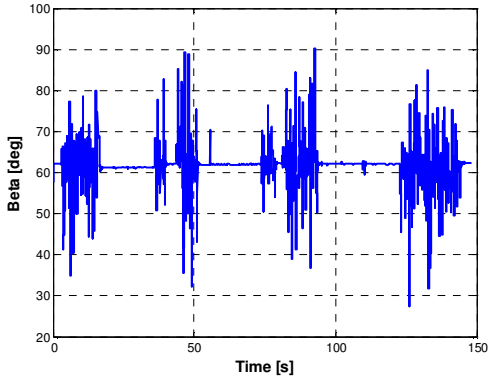


Figure 6.52 The orientation angle data β taken from digital compass with respect to time

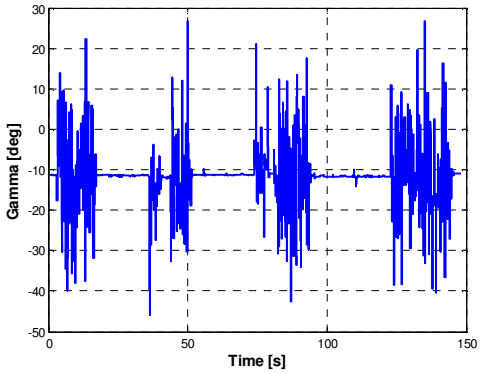


Figure 6.53 The orientation angle data γ taken from digital compass with respect to time

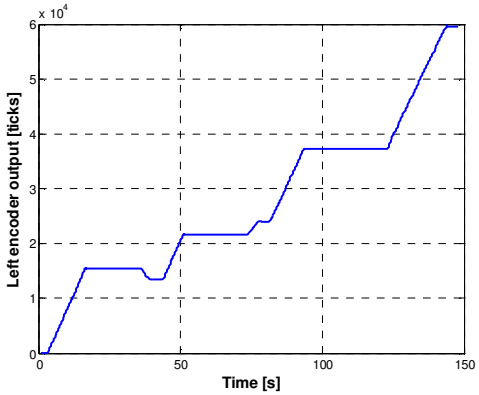


Figure 6.54 The left encoder output data with respect to time

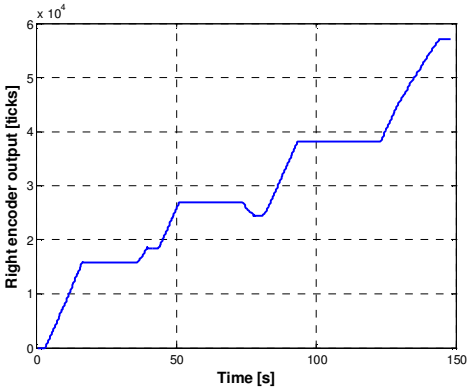


Figure 6.55 The right encoder output data with respect to time

6.4 Experimental Test Results for the Unaccompanied IMU Case

In this section, the test results for the unaccompanied IMU case are illustrated. However there are encoders and digital compass on the test system, the Kalman filter algorithms were not executed. Instead, navigation equations are solved only taking the IMU output data as the input.

6.4.1 Experimental Test Results for the Unaccompanied IMU Case while Staying at a Stationary Point

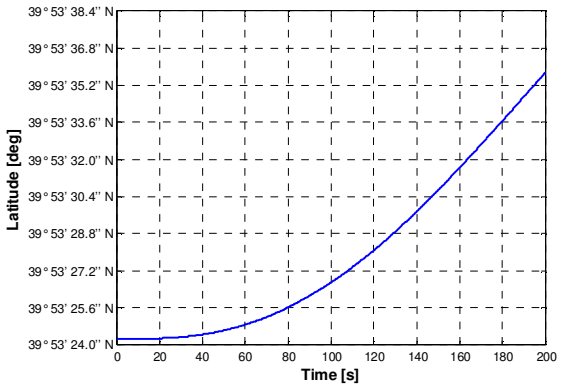


Figure 6.56 Change in latitude with respect to time

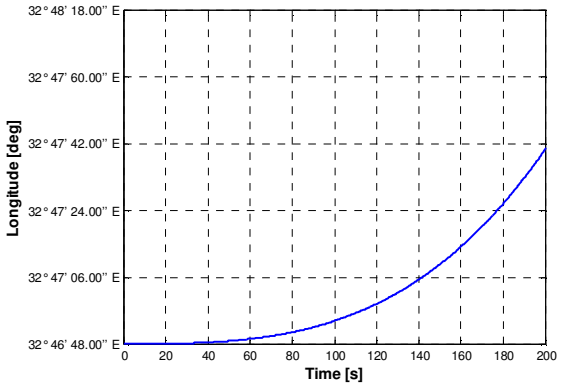


Figure 6.57 Change in longitude with respect to time

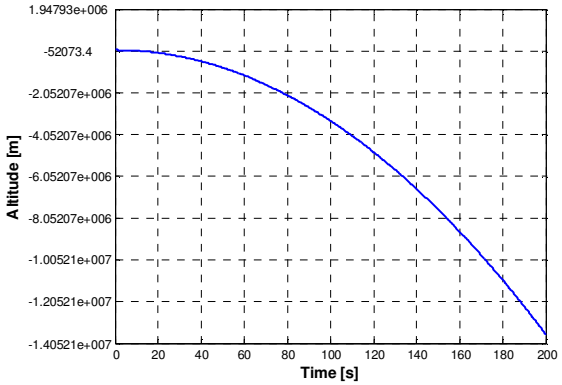


Figure 6.58 Change in altitude with respect to time

At staying position the unmanned vehicle does not move with respect to the Earth. However there is movement in the figures. This is because of the accumulation of position errors by double integrals as the test flows. The test continues for approximately 200 seconds which means very high error accumulation.

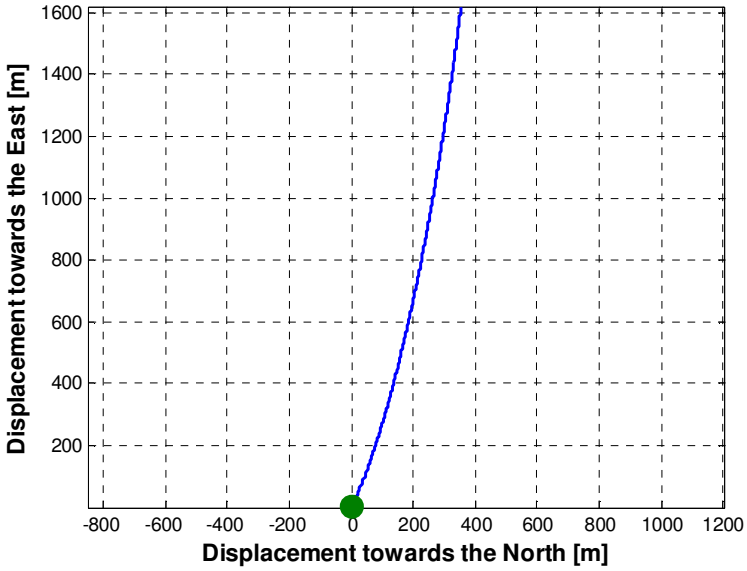


Figure 6.59 Navigation data in meters converted from changes in longitude and latitude outputs

The error in latitude and longitude is simply because of IMU sensor errors. The noise on measurements and also changes in bias result in such kind of errors.

6.4.2 Experimental Test Results for the Unaccompanied IMU Case while Moving on a Straight Line

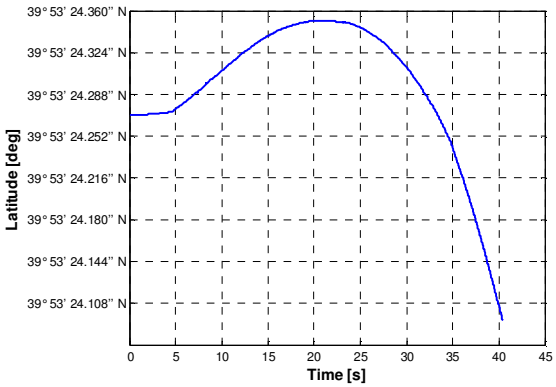


Figure 6.60 Change in latitude with respect to time

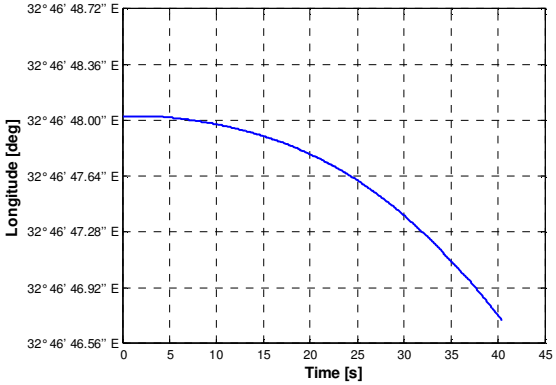


Figure 6.61 Change in longitude with respect to time

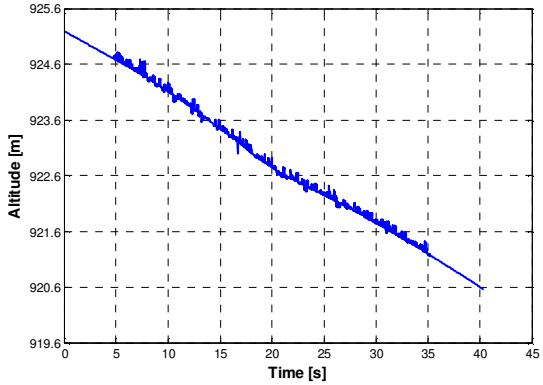


Figure 6.62 Change in altitude with respect to time

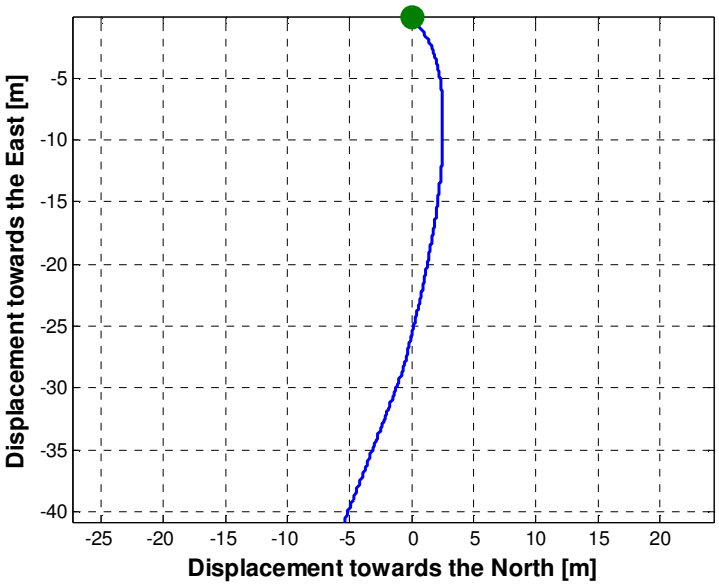


Figure 6.63 Navigation data in meters converted from changes in longitude and latitude outputs

This test is done by applying adjusted voltages to the DC motors of the vehicle so that the vehicle makes a rectilinear motion. However the trend on the motion can be seen on the figure, the distances travelled is 3 to 4 times larger.

6.4.3 Experimental Test Results for the Unaccompanied IMU Case while Sharp Steering

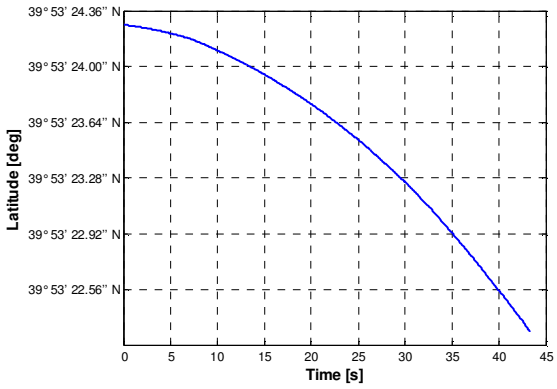


Figure 6.64 Change in latitude with respect to time

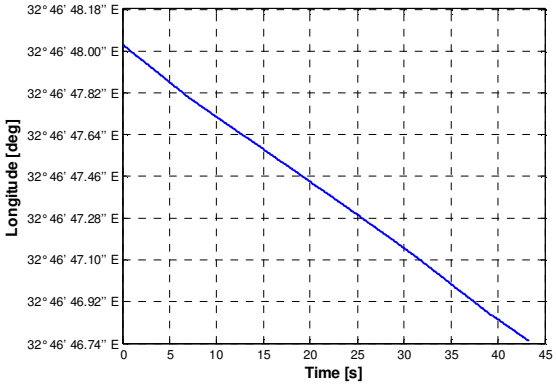


Figure 6.65 Change in longitude with respect to time

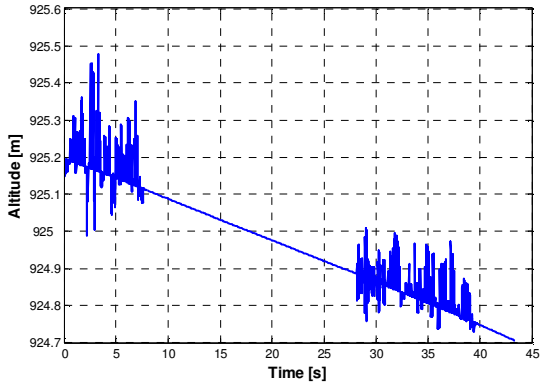


Figure 6.66 Change in altitude with respect to time

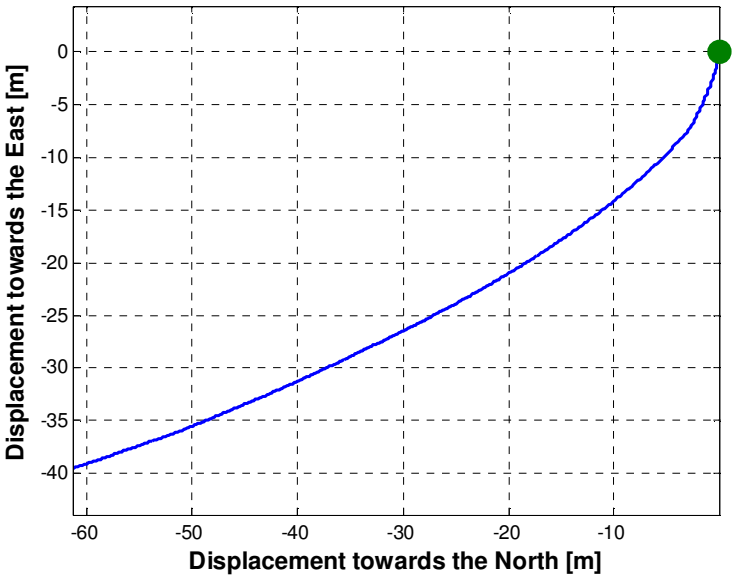


Figure 6.67 Navigation data in meters converted from changes in longitude and latitude outputs

In this figure, since the orientation data is also calculated by integrals, the orientation of the vehicle cannot be caught. This is because of gyroscope errors. However at the beginning of the test, in a few seconds, nearly successful orientation calculation is done. The noise and changes of bias and mechanical vibrations highly affective afterwards.

6.4.4 Experimental Test Results for the Unaccompanied IMU Case while Smooth Steering

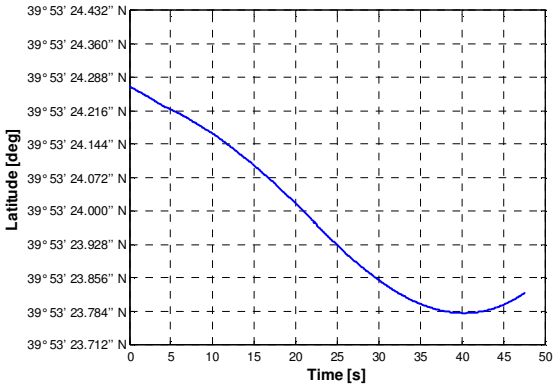


Figure 6.68 Change in latitude with respect to time

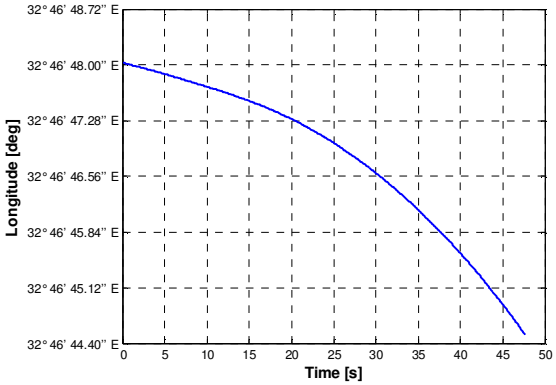


Figure 6.69 Change in longitude with respect to time

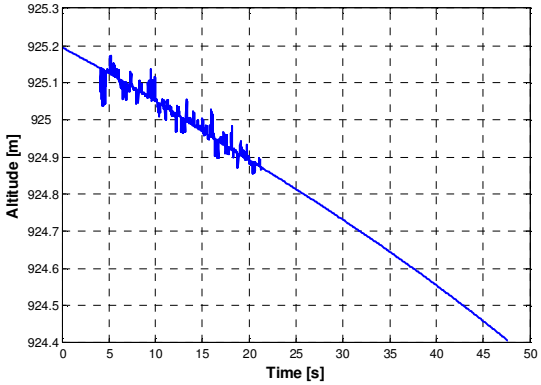


Figure 6.70 Change in altitude with respect to time

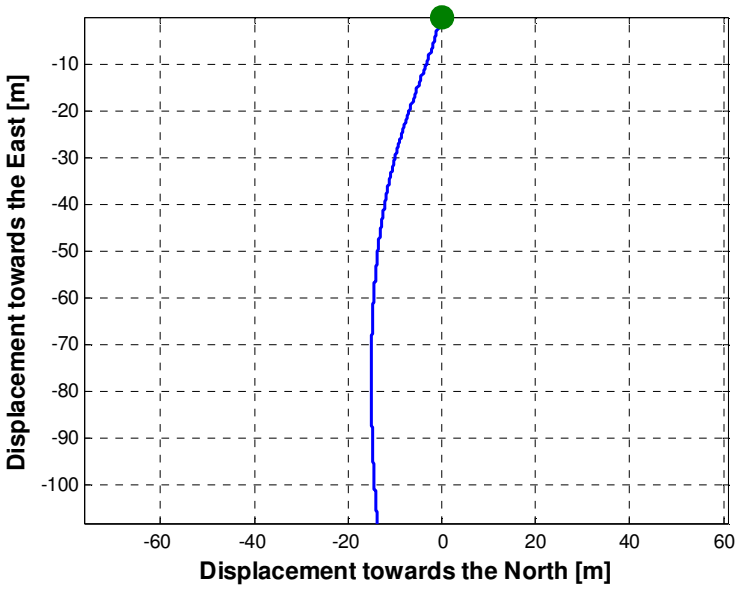


Figure 6.71 Navigation data in meters converted from changes in longitude and latitude outputs

The steering cannot be seen on the figure because steering was done at the middle of the test. Until that time error accumulation of integrals has already corrupted the navigation data.

6.4.5 Experimental Test Results for the Unaccompanied IMU Case while Making Complicated Motion

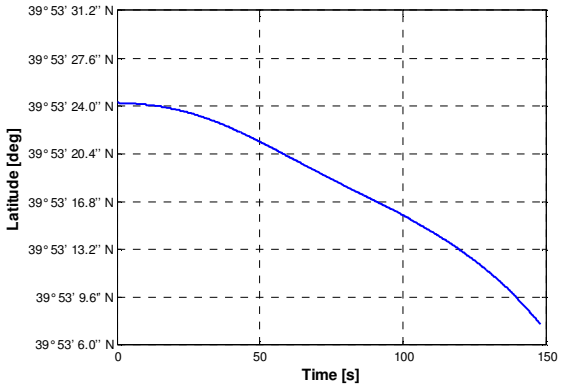


Figure 6.72 Change in latitude with respect to time

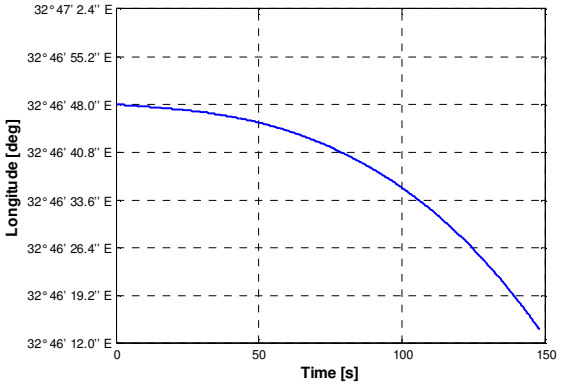


Figure 6.73 Change in longitude with respect to time

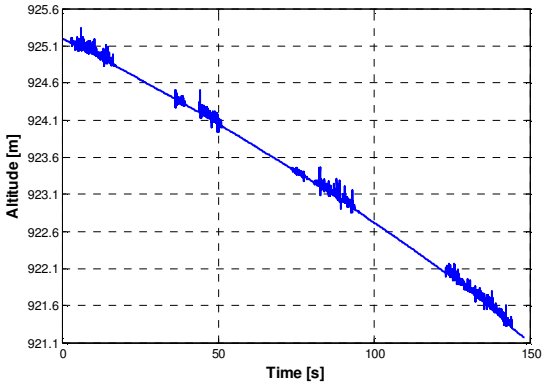


Figure 6.74 Change in altitude with respect to time

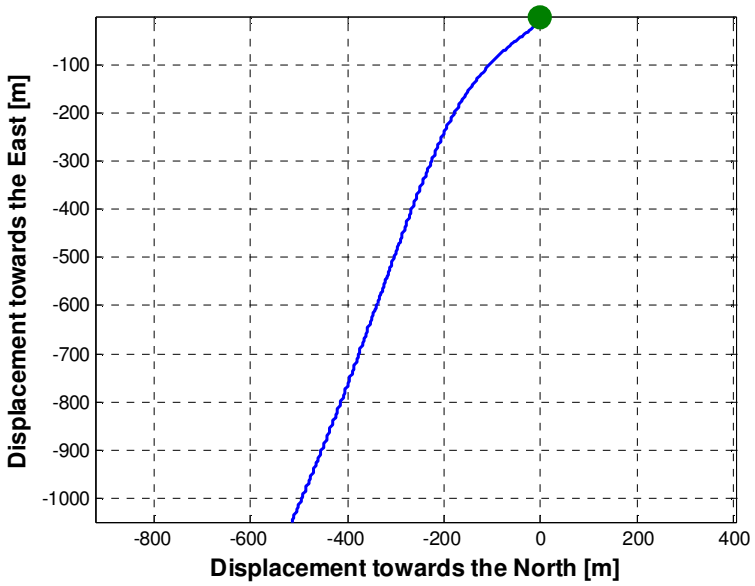


Figure 6.75 Navigation data in meters converted from changes in longitude and latitude outputs

This test was done for approximately 150 seconds. In the latitude versus longitude plot, the motion of the vehicle cannot be seen. As a result of unaccompanied IMU tests, it can be easily concluded that the IMU should not been used alone for long times. This is why IMU and GPS data are combined in literature. The data is renewed as the GPS signal is received. Some systems also reset all integrals when new GPS data is available. In this thesis, an approach similar to this was applied. Instead of an absolute position sensor, GPS, a dead reckoning sensor triplet (digital compass and encoders) were utilized. In the following sections, the results of the fusion of these sensors were displayed.

6.5 Experimental Test Results for the Feed Forward Multisensor Fusion Case

6.5.1 Experimental Test Results for the Feed Forward Multisensor Fusion Case while Staying at a Stationary Point

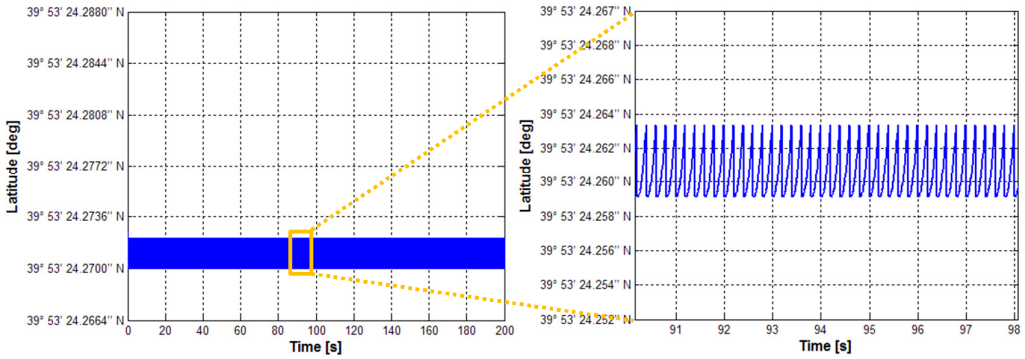


Figure 6.76 Change in latitude with respect to time

In the figures, since the latitude data has error accumulation, it is always been reset by the Kalman filter as sensor triplet data is available.

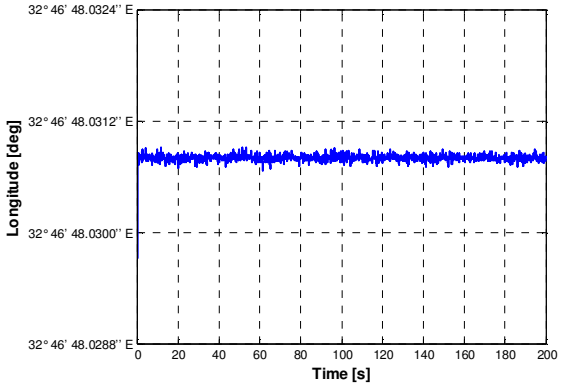


Figure 6.77 Change in longitude with respect to time

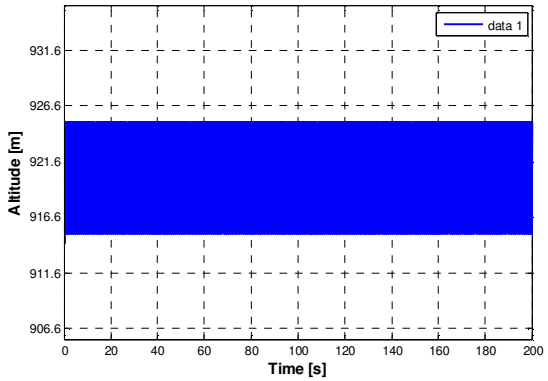


Figure 6.78 Change in altitude with respect to time

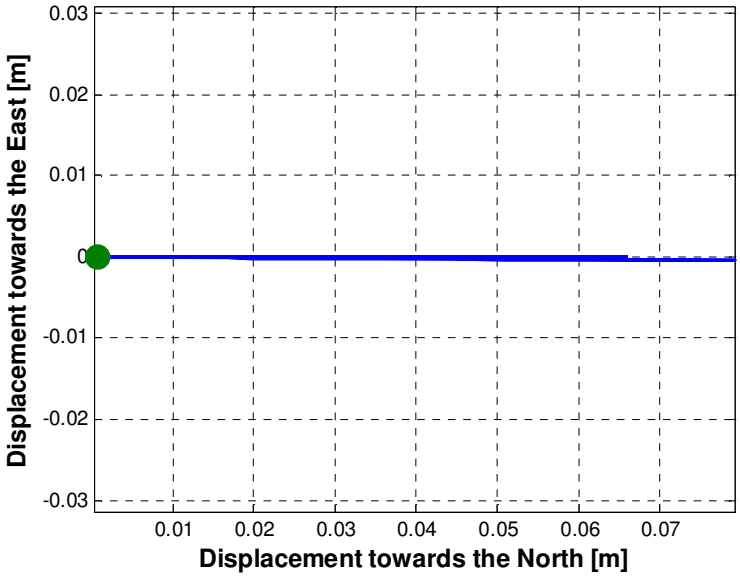


Figure 6.79 Navigation data in meters converted from changes in longitude and latitude outputs

In this figure, the vehicle seems it was moved about 8 centimeters to the North. This distance is the error of feed forward Kalman calculations. The result is highly improved by the fusion of sensors in comparison with sole IMU data. Remember that, the errors of un accompanied IMU calculations in the previous section was in the order of meters.

6.5.2 Experimental Test Results for the Feed Forward Multisensor Fusion Case while Moving on a Straight Line

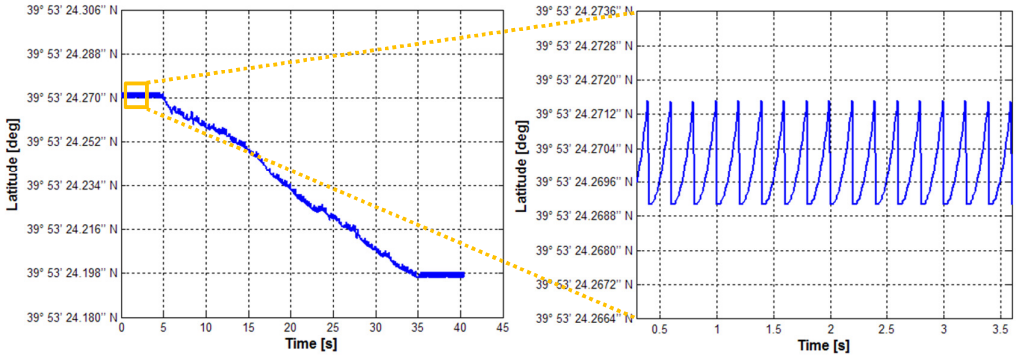


Figure 6.80 Change in latitude with respect to time

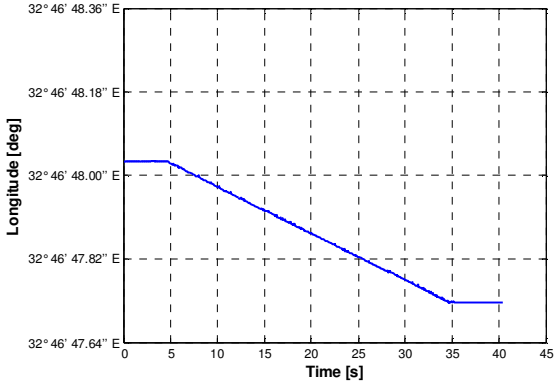


Figure 6.81 Change in longitude with respect to time

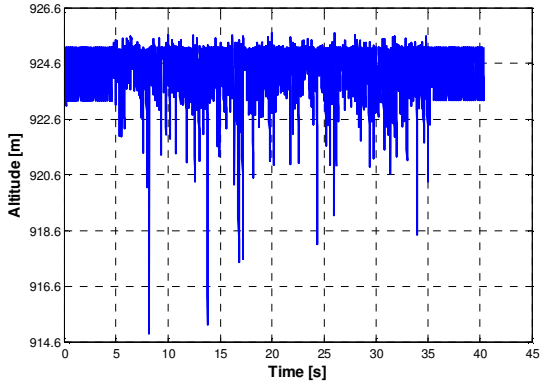


Figure 6.82 Change in altitude with respect to time

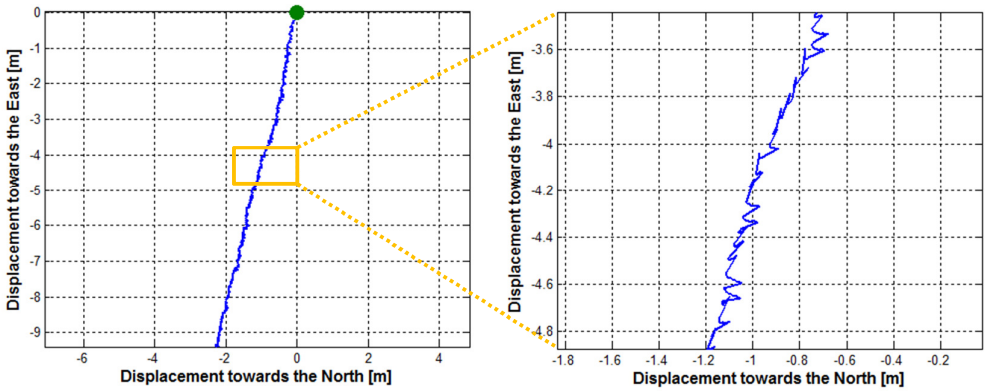


Figure 6.83 Navigation data in meters converted from changes in longitude and latitude outputs

The displacement plot shows the trend of motion well. Not only the trend but also the total displacement is highly improved by sensor fusion. The data connecting the Kalman sensor fusion outputs are ripple like and erroneous. In order to improve this data, a method better than the feed forward sensor fusion algorithm is required.

6.5.3 Experimental Test Results for the Feed Forward Multisensor Fusion Case while Sharp Steering

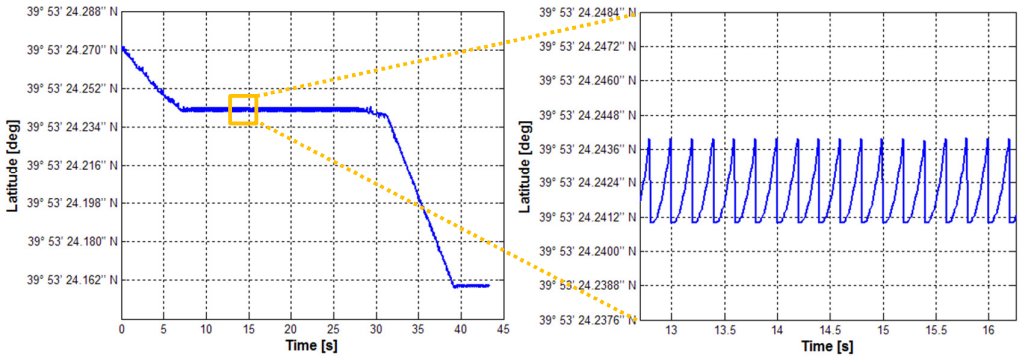


Figure 6.84 Change in latitude with respect to time

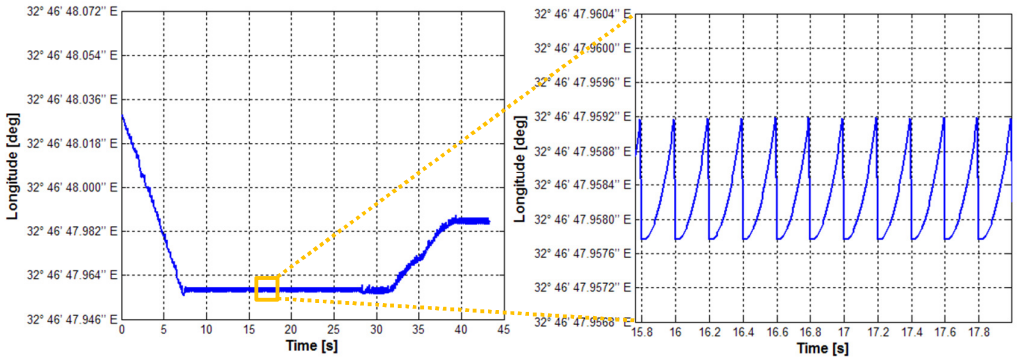


Figure 6.85 Change in longitude with respect to time

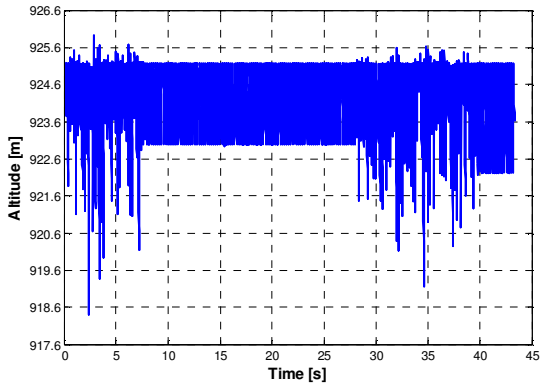


Figure 6.86 Change in altitude with respect to time

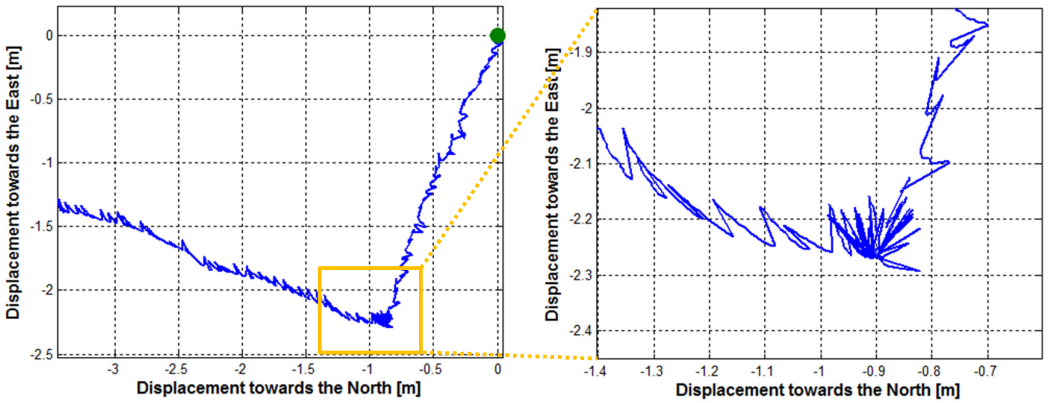


Figure 6.87 Navigation data in meters converted from changes in longitude and latitude outputs

This test is done by applying inverse and equal voltages to the DC motors of the vehicle by using the H bridge in the manual control unit. It results in rotation about the geometric center of the vehicle. A 90° orientation change was given and the vehicle continued to its rectilinear motion. The path travelled can be clearly seen on the figure. However the output figure suffers from ripples.

6.5.4 Experimental Test Results for the Feed Forward Multisensor Fusion Case while Smooth Steering

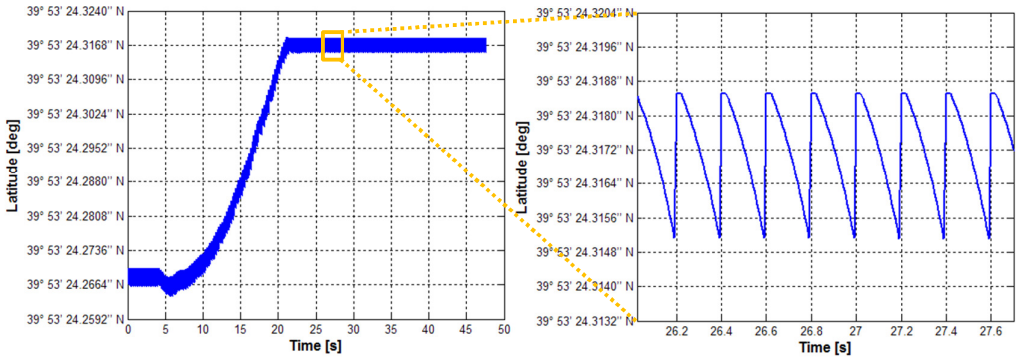


Figure 6.88 Change in latitude with respect to time

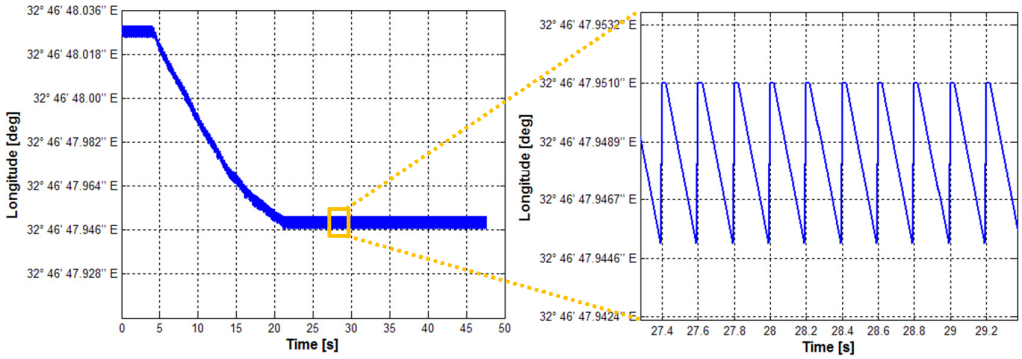


Figure 6.89 Change in longitude with respect to time

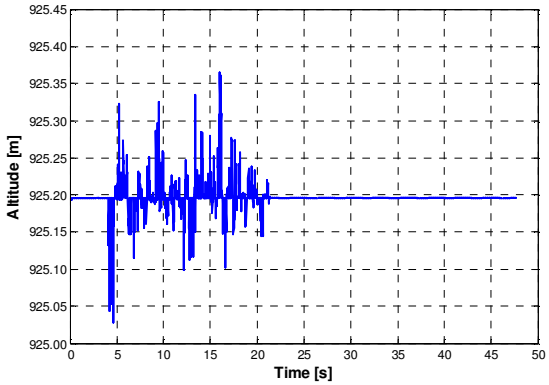


Figure 6.90 Change in altitude with respect to time

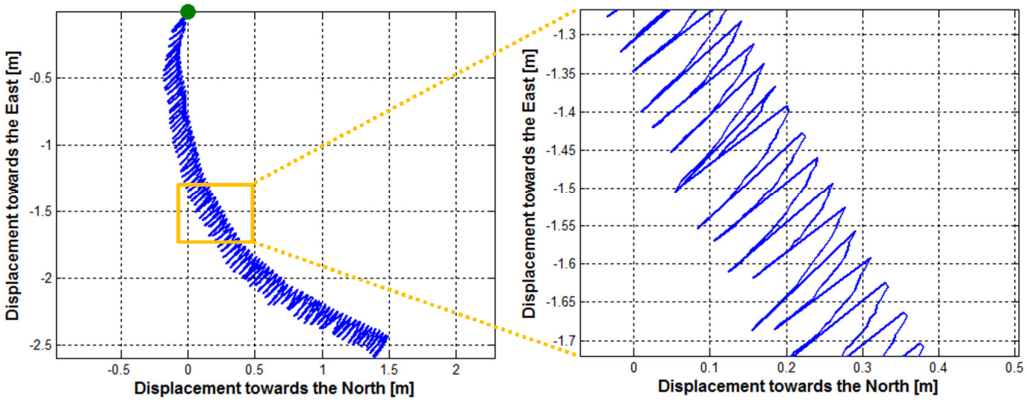


Figure 6.91 Navigation data in meters converted from changes in longitude and latitude outputs

Smooth steering is some kind of motion when voltages of same polarity but different in amount is applied. The angular velocity of motor shafts quickly settles to maximum speed and this speed is a function of the voltage. The difference in angular velocity of motor shafts, so the vehicle tracks causes the vehicle makes a rotation whose rotation axis location can be found by knowing the information of the speed of vehicle tracks. Ripple in this motion is higher due to corrupted direction cosine matrix data. The source of corruption is the accumulation of errors in integral calculations. It is obvious that, there is a need to reset the integral data as the sensor fusion Kalman outputs are available.

6.5.5 Experimental Test Results for the Feed Forward Multisensor Fusion Case while Making Complicated Motion

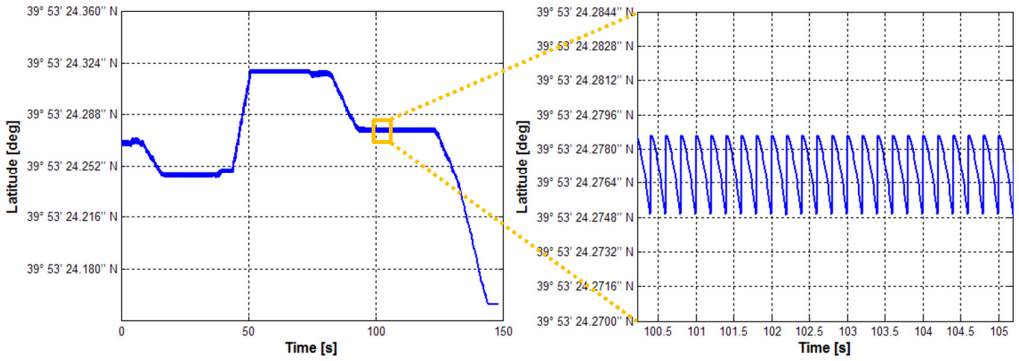


Figure 6.92 Change in latitude with respect to time

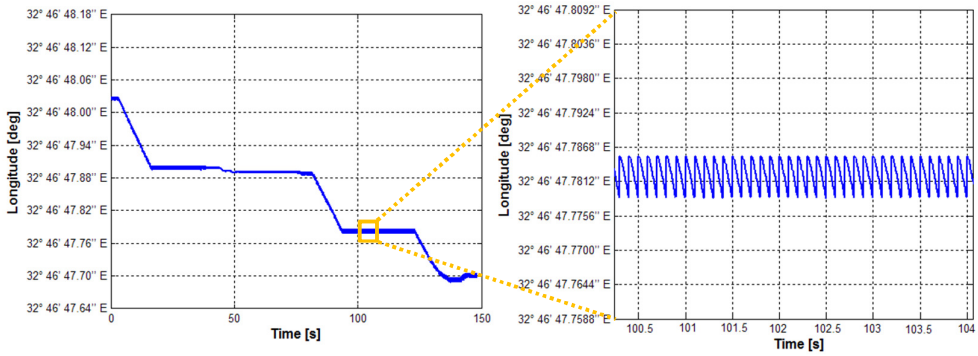


Figure 6.93 Change in longitude with respect to time

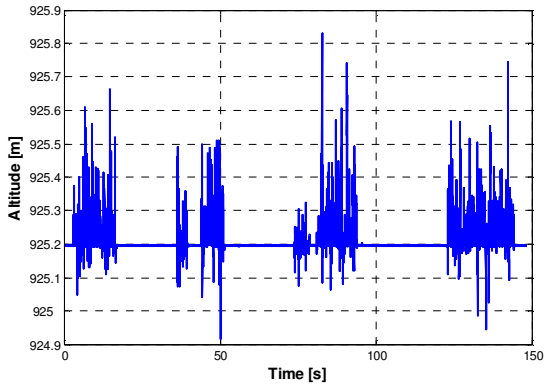


Figure 6.94 Change in altitude with respect to time

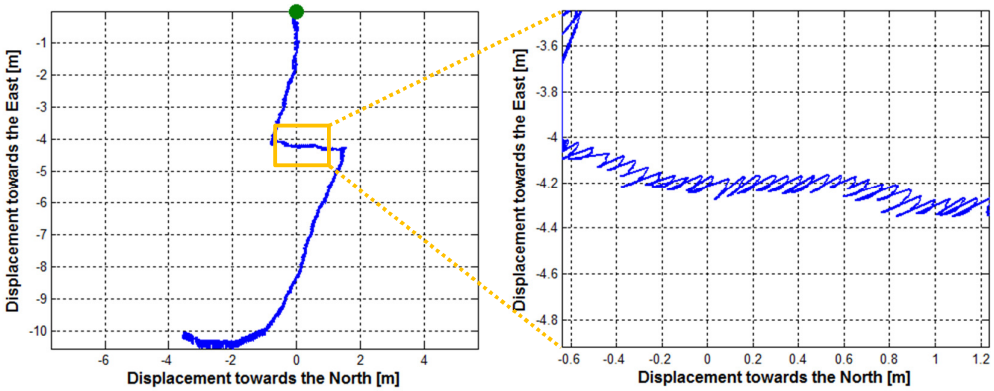


Figure 6.95 Navigation data in meters converted from changes in longitude and latitude outputs

Navigation path seems clear. The ripples should be corrected by feedback sensor fusion

6.6 Experimental Test Results for the Feedback Multisensor Fusion Case

6.6.1 Experimental Test Results for the Feedback Multisensor Fusion Case while Staying at a Stationary Point

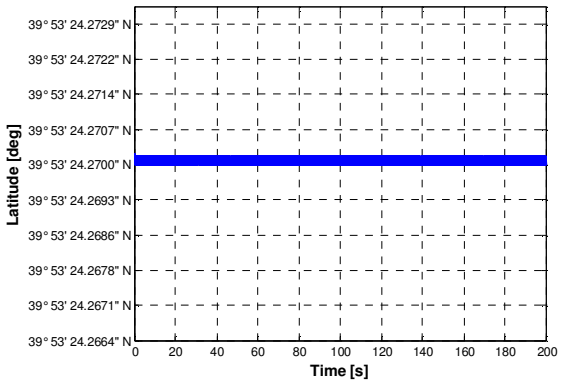


Figure 6.96 Change in latitude with respect to time

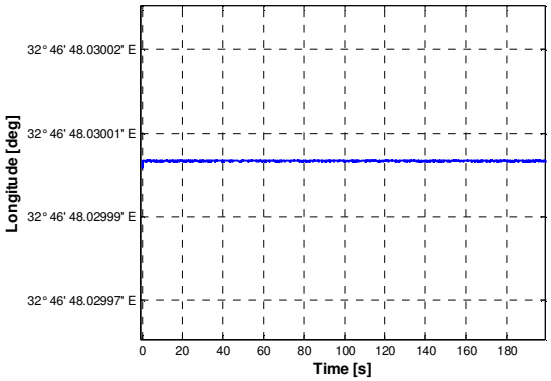


Figure 6.97 Change in longitude with respect to time

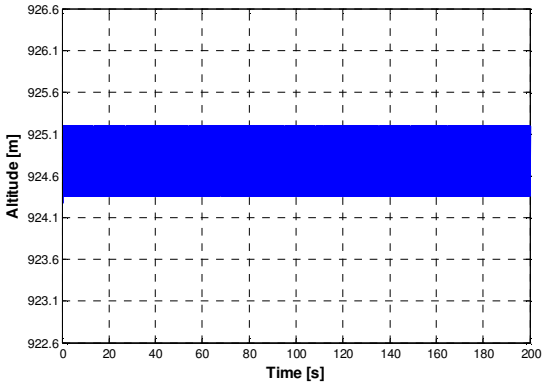


Figure 6.98 Change in altitude with respect to time

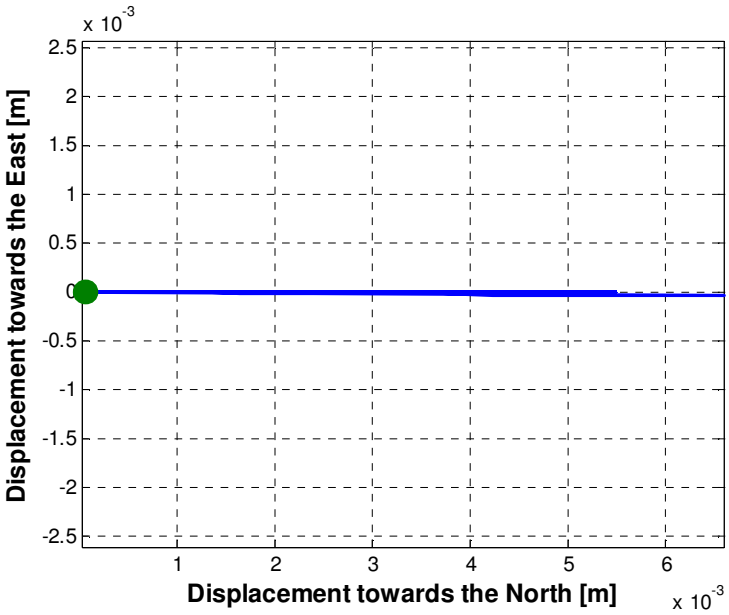


Figure 6.99 Navigation data in meters converted from changes in longitude and latitude outputs

In this figure, the vehicle seems it was moved about 7 millimeters to the North. The result is highly improved by the feedback fusion of sensors in comparison with feed forward fusion. The errors of feed forward calculations in the previous section were in the order of centimeters.

6.6.2 Experimental Test Results for the Feedback Multisensor Fusion Case while Moving on a Straight Line

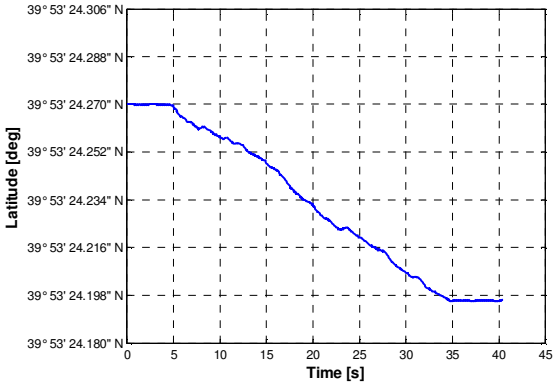


Figure 6.100 Change in latitude with respect to time

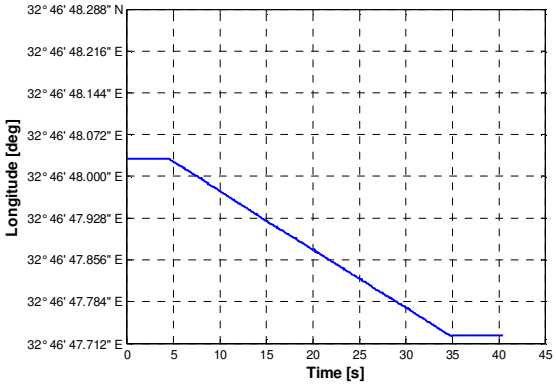


Figure 6.101 Change in longitude with respect to time

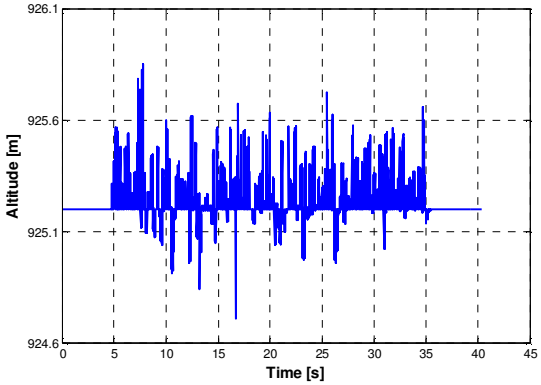


Figure 6.102 Change in altitude with respect to time

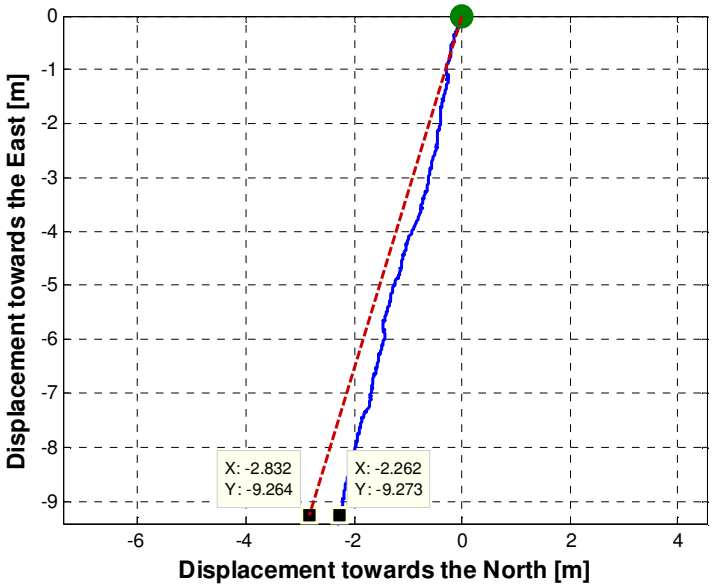


Figure 6.103 Navigation data in meters converted from changes in longitude and latitude outputs

(Dashed lines are the real path that the vehicle travelled during the test.)

The output data of feedback sensor fusion much smoother than feed forward data but it is not perfect. The vessel like view of the path is because of the noise in the sensor triplet. The noise of digital compass and the IMU causes the errors. In addition to this, the compass is affected from ferromagnetic materials. There may be iron or some other ferromagnetic material inside the floor or walls. This results in change in bias of compass. As the vehicle moves away from the disruption source, the bias reduces. This phenomenon is always a problem in compass use.

6.6.3 Experimental Test Results for the Feedback Multisensor Fusion Case while Sharp Steering

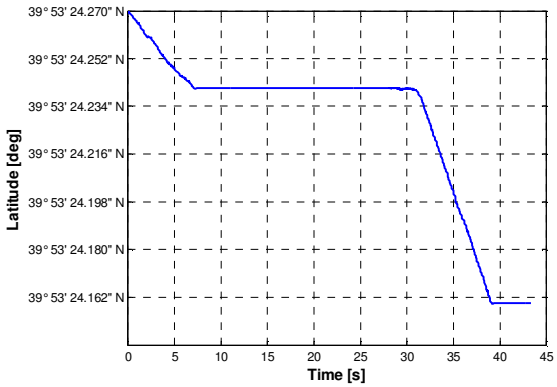


Figure 6.104 Change in latitude with respect to time

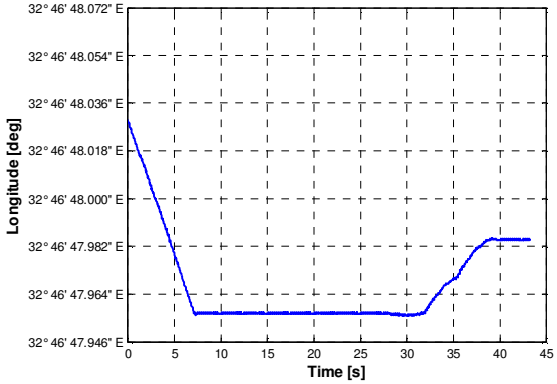


Figure 6.105 Change in longitude with respect to time

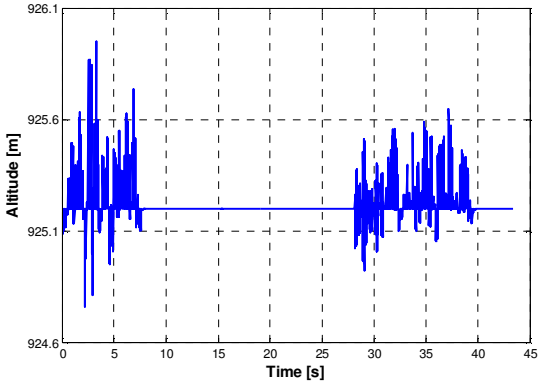


Figure 6.106 Change in altitude with respect to time

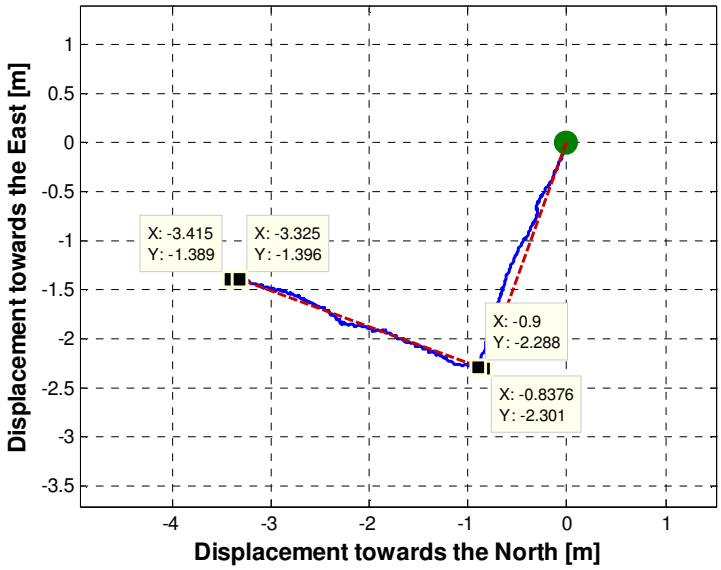


Figure 6.107 Navigation data in meters converted from changes in longitude and latitude outputs

(Dashed lines are the real path that the vehicle travelled during the test.)

The feedback output data is quite near to the real data in this test.

6.6.4 Experimental Test Results for the Feedback Multisensor Fusion Case while Smooth Steering

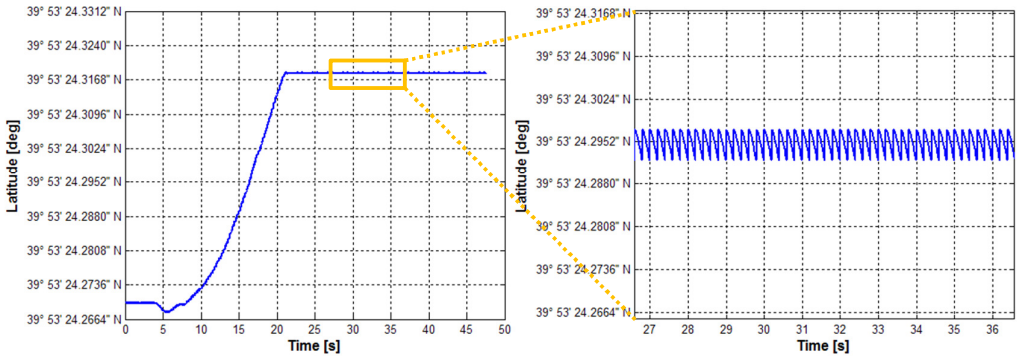


Figure 6.108 Change in latitude with respect to time

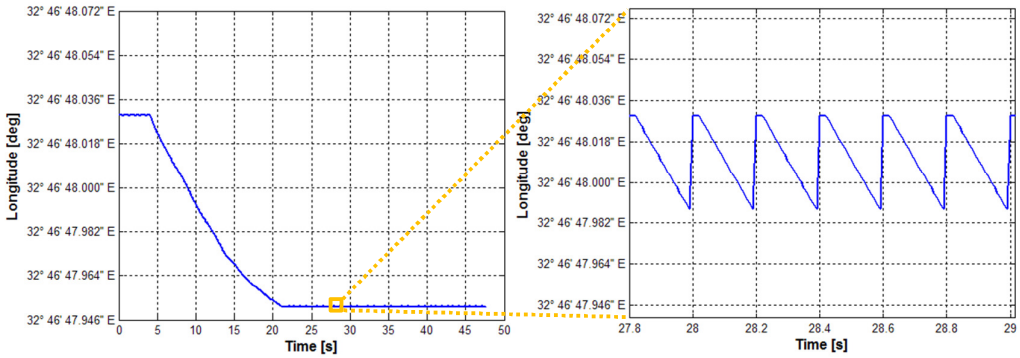


Figure 6.109 Change in longitude with respect to time

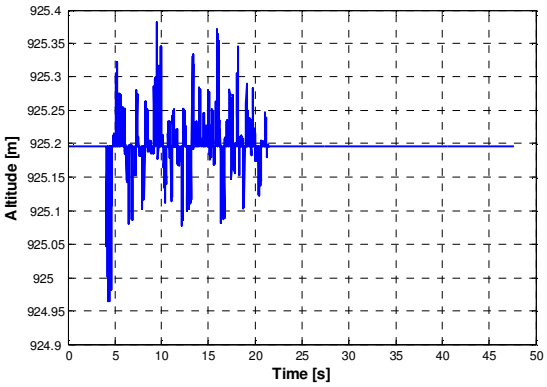


Figure 6.110 Change in altitude with respect to time

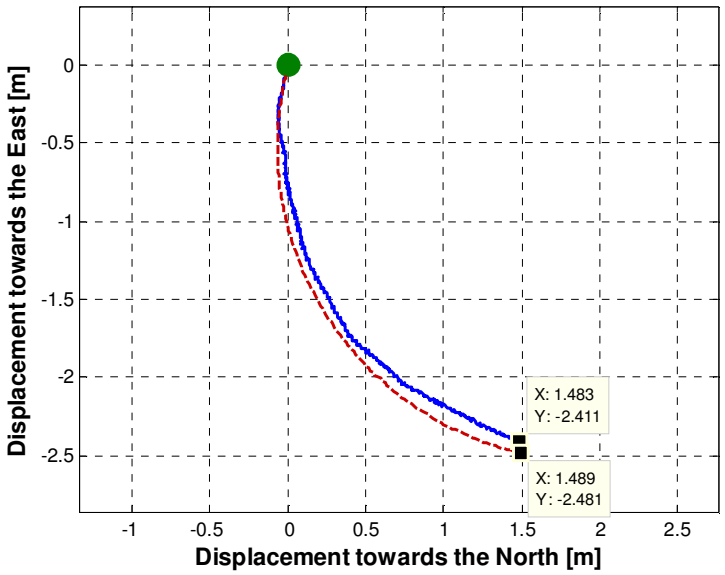


Figure 6.111 Navigation data in meters converted from changes in longitude and latitude outputs

(Dashed lines are the real path that the vehicle travelled during the test.)

In this test, the vehicle made a 90° smooth rotation. As the rotation continues the small position error gets larger but it is still in the order of centimeters at the end.

6.6.5 Experimental Test Results for the Feedback Multisensor Fusion Case while Making Complicated Motion

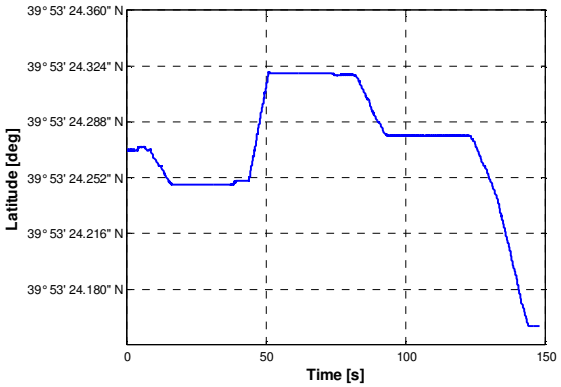


Figure 6.112 Change in latitude with respect to time

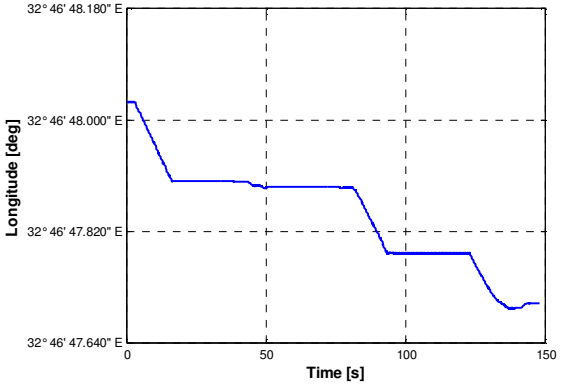


Figure 6.113 Change in longitude with respect to time

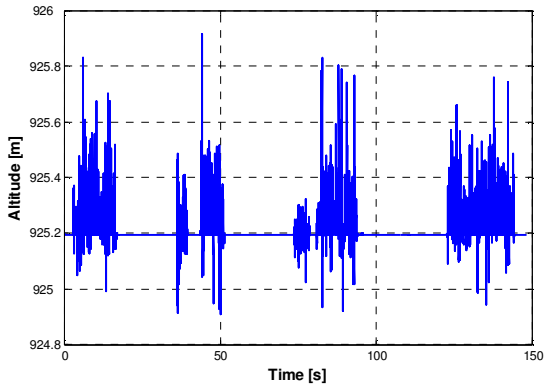


Figure 6.114 Change in altitude with respect to time

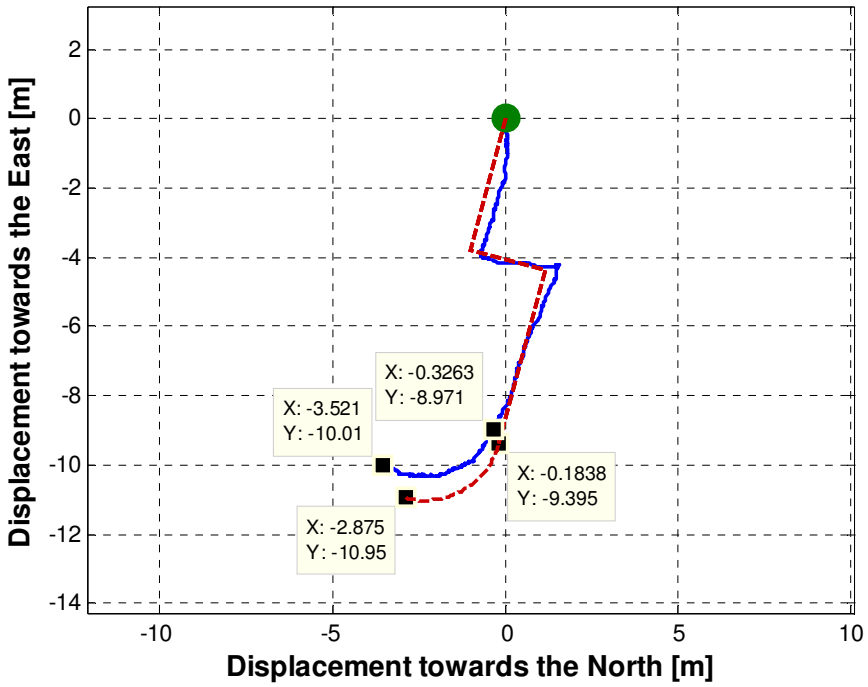


Figure 6.115 Navigation data in meters converted from changes in longitude and latitude outputs
(Dashed lines are the real path that the vehicle travelled during the test.)

This test is approximately 150 seconds. In this complicated motion, it can be easily seen that an orientation calculation error made in the past affects highly the final position calculation. The small orientation error at the start of the smooth steering caused a bigger position error at the final position. However the overall position error is still smaller than 1 meter. This is a problem of dead reckoning techniques. Since the calculations are cumulative, a small error influences the solution at the end.

One way to solve this problem is taking absolute position at some time instants. For example, if the vehicle goes on a road and the map of the road is in the

memory of the solver algorithm, than it can be estimated that the vehicle went to which road. This technique is called map matching [26], [27]. Another solution might be using a sensor that sends absolute position data. The sensor may be a GPS and/or a altimeter and/or a camera etc.

CHAPTER 7

CONCLUSIONS

7.1 Conclusions

In this thesis, 4 main tasks were accomplished.

The first one is writing an algorithm to solve mechanization equations if accelerometers and gyroscopes data are available. This algorithm was tested by synthetic input data. Therefore it was verified that the data output by the algorithm is agreement with the exact mathematical solution of the synthetic data.

Secondly, a big amount of data was collected from the sensors in order to find the process and measurement covariance matrices. Also a navigation system error matrix were found and entered into the algorithm of the Kalman filter. A feedback and a feed forward sensor fusion were added to the algorithm.

After this step, a unmanned tracked vehicle prototype was manufactured. A control unit was made in order to adjust the velocities of the tracks. An xPC – Target data acquisition system was set up. By doing all these, test of the algorithms were possible.

A test procedure was planned in order to see the benefits of sensor fusion and the differences of sensor fusion algorithms. The results were saved after each test for analyzing.

As a result of all these thesis work, it can be concluded sensor fusion should be used to reduce the signal noise of sensor. The resultant signal noise is lower in variance than the signals of the sensors that are combined. Therefore, instead of uncoupled algorithms coupled algorithms should be used. The worst case is using the IMU alone. IMU can only be utilized for a few seconds. After few seconds the integration errors accumulate and dominate the result. All these were illustrated in test results.

Among coupled algorithms, feedback sensor fusion gave better results than the feed forward sensor fusion. This is because in feedback algorithm; the previous states are fed back into the mechanization equations.

7.2 Future Work

An absolute position sensor can be added into the sensor fusion. This is going to make the position calculations more accurate. This sensor may be GPS, altimeter, distance sensor, camera etc.

The unmanned tracked vehicle prototype can be improved to take commands of position. The data communication can be done wireless and a battery mounted on it can power the vehicle.

If tests on a road is going to done, the map of the road and an estimator can be can be inserted into the algorithm. By this way, map match technique can be used.

One of the sources of errors is the digital compass' noise. This level of this noise is highly increased because of mechanical vibrations. In the vehicle the compass is mounted in a box and in this box a damper material covers the sensor. A suspension, maybe an active one can be designed in order to damp the mechanical vibrations.

By using 2 or more IMU's at different locations of the vehicle the accuracy of calculations can be increased.

REFERENCES

1. Biezad, J. D., Integrated Navigation and Guidance Systems, American Institute of Aeronautics and Astronautics & Institution of Electrical Engineers, 1999

2. Savant, C. F., Howard, R. C., Solloway, C. B., Savant, C. A., Principles of Inertial Navigation, McGraw-Hill Book Company Inc., 1961

3. Rogers, R. M., Applied Mathematics in Integrated Navigation Systems, American Institute of Aeronautics and Astronautics, 2003

4. Titterton, D. H., Weston, J. L., Strapdown Inertial Navigation Technology, American Institute of Aeronautics and Astronautics & Institution of Electrical Engineers, 2004

5. Farell, J. A., Aided Navigation GPS with High Rate Sensors, McGraw-Hill Book Company Inc., 2008

6. Farell, J. A., Barth, M., The Global Positioning System and Inertial Navigation, McGraw-Hill Book Company Inc., 1999

7. Greenberg, M. D., Advanced Engineering Mathematics, Prentice Hall, 1998

8. Goldstein, H., Poole, C., Safko, J., Classical Mechanics, Addison Wesley,

9. Özgören, M. K., Advanced Dynamics Lecture Notes, Middle East Technical University Mechanical Engineering Department, 2010

10. The Gram-Schmidt Algorithm
    http://www.math.hmc.edu/calculus/tutorials/gramschmidt/
    Last access date: 12.02.2012

11. Bekir, E., Introduction to Modern Navigation Systems, World Scientific, 2007

12. Ginsberg, J., Engineering Dynamics, Cambridge University Press, 2008

13. Jacoby, W., Smilde, P. L., Gravity Interpretation: Fundamentals and Application of Gravity Inversion and Geological Interpretation, Springer, 2009

14. Akçay, M. E., M. S. Thesis, Land Vehicle Navigation with GPS/INS Sensor Fusing Using Kalman Filter, METU, 2008

15. Ogata K., Modern Control Engineering, Prentice Hall, 2002

16. 440 Series Inertial Systems User Manual, Crossbow
www.xbow.com
Last access date: 12.02.2012

17. Digital Compass User's guide rev. 4, OS 5000 series, Ocean Server
www.ocean-server.com
Last access date: 12.02.2012

18. Savage, P., Strapdown Analytics: Part1-2, Strapdown Associates, Inc., Maple Plain, Minnesota, 2000

19. Greval, M., Andrews A., Kalman Filtering Theory and Practice using MATLAB, John Wiley and Sons, Inc., 2001

20. Britting, Kenneth R. Inertial Navigation Systems Analysis. Wiley-Interscience, 1971.

21. National Geophysical Data Center
http://www.ngdc.noaa.gov/
Last access date: 16.05.2012

22. Welch G., Bishop G., An Introduction to the Kalman Filter Course Notes, University of North Carolina at Chapel Hill, 2001

23. Maybeck P., Stochastic Models, Estimation and Control, Academic Press, 1979

24. Sana F., M. S. Thesis, Modeling and Estimation for Maneuvering Target Tracking with Inertial Systems using Interacting Multiple Models, Universitat Politecnica de Catalunya, Barcelona, Spain.

25. MATLAB Product Help documents

26. David McNeil Mayhew, Multi-rate Sensor Fusion for GPS Navigation Using Kalman Filtering, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, USA.

27. Walcho K. J., Nechyba M. C., Schwartz E., Arroyo A., Embedded Low Cost Inertial Navigation System, University of Florida, USA.

28. Park K., Chung D., Chung H., Lee J. G., Dead Reckoning Navigation of a Mobile Robot Using an Indirect Kalman Filter, Proceedings of IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems, 1996

29. Dutchops Gyroscopic Instruments
http://www.dutchops.com/Portfolio_Marcel/Articles/Instruments/Gyroscopic_Instruments/Theory_Gyroscopes.htm
Last access date: 11.09.2012

30. NASA Gimbal IMU
http://www.hackoh.com/blog/index.php/posts/Motion
Last access date: 08.07.2012

31. Sparkfun IMU
http://www.skpang.co.uk/catalog/sensors-imu-c-169_184.html
Last access date: 08.07.2012

32. Magnetic declination
http://magnetic-declination.com/what-is-magnetic-declination.php
Last access date: 16.05.2012

APPENDIX A

MOOG-CROSSBOW IMU SPECIFICATIONS

NAV440

GPS-Aided MEMS Inertial System

MOOG CROSSBOW

DESCRIPTION

The Moog Crossbow NAV440 is a fully-integrated combined GPS navigation and GPS-aided Attitude & Heading Reference System (AHRS) solution. The NAV440 provides full inertial data (angles, rates, accels) and GPS position, along with inertially computed velocity that provides significant improvement in stability and higher data rates compared with stand-alone GPS velocity measurements.

The NAV440 integrates highly reliable MEMS sensors (gyros and accelerometers), 3-axis magnetometer, and a WAAS/EGNOS-enabled GPS receiver all in a compact and rugged environmentally sealed enclosure. The NAV440 provides consistent performance in challenging operating environments, and is user-configurable for a wide variety of applications such as unmanned vehicle control, land vehicle guidance, avionics systems, and platform stabilization.

KEY FEATURES

- **Pitch and roll accuracy of <0.4°**
- **Output data rate up to 100 Hz**
- **High-range sensor options (400°/sec, 10g)**
- **WAAS and EGNOS enable GPS**
- **Low power < 4W at 28 VDC**
- **High reliability, MTBF >25,000 hours**
- **Rugged sealed enclosure**
- **Certified for DO-160D environments**

PERFORMANCE

Position/Velocity

Position Accuracy	<3.0 m CEP
1PPS Accuracy	±50 ns

Heading

Accuracy	<1.0° rms (magnetic) <0.75° rms (with GPS aiding)

Attitude

Range: Roll, Pitch	±180°, ±90°
Accuracy	<0.4°

Angular Rate

Range: Roll, Pitch, Yaw	±200°
Bias Stability in run	<10°/hr
Bias Stability over temp	<0.02°/sec

Acceleration

Input Range	±4 g or ±10 g
Bias Stability in run	<1 mg
Bias Stability over temp	<4 mg

SPECIFICATIONS

Environment

Operating Temperature	-40° to +71°C
Enclosure	IP66 compliant

Electrical

Input Voltage	9 to 42 VDC
Power Consumption	< 4 W
Digital Interface	RS-422

Physical

Size	3.0"w x 3.75"l x 3.0"h
Weight	1.3 lbs (0.58 kg)
Interface Connector	DB15, D-sub 15 pin Male
GPS Antenna Connector	SMA Male

ORDERING INFORMATION

Model	Description
NAV440CA-200-2	GPS-Aided MEMS Inertial System Includes NAV-VIEW software, User's manual, Quick start guide, GPS Antenna, and cable.

For more information

Phone: 1-408-965-3300
Email: sales@moog-crossbow.com
www.moog-crossbow.com

Moog-Crossbow
1421 McCarthy Blvd.
Milpitas, CA 95035

6020-0159-01 Rev B

APPENDIX B

OCEAN SERVER OS5000 COMPASS SPECIFICATIONS

Features:

- Compass accuracy, 0.5 degrees RMS heading while level, 1° RMS <±30° Tilt, 1.5° RMS <±60° Tilt, undisturbed field, .1 Degree resolution
- Roll & Pitch full rotation, typical 1° accuracy <±30° tilt
- Pitch Angles +/-90 degrees, Roll Angles +/- 180 degrees
- Tilt-compensated (electronically gimbaled)
- Tiny size, 1"x1"x0.3", less than 2 grams weight
- Low Power Consumption, <30ma @3.3V
- Hard and soft-iron compensation routines
- Optional support for a high resolution Depth or Altitude sensor (24 bit AD)
- Serial Interface:
  - RS232, USB or TTL
  - Baud rate programmable 4,800 to 115,000 baud
- Rugged design
  - 10,000 G shock survival
  - -40° C to 80° C operating temperature (Accuracy specified for 0° C to 50° C)
- ASCII sentence output, in several formats, NMEA checksum
- High Data Update Rate to 40HZ
- Support for True or Magnetic North Output
- Precision components
  - 3 Axis magnetic sensors from Honeywell
  - 3 Axis Accelerometers from ST Microelectronics
  - 24 bit differential Analog to Digital converters from Analog Devices

APPENDIX C

FAULHABER IE2-512 MAGNETIC ENCODER SPECIFICATIONS

FAULHABER

Encoders

Magnetic Encoders

Features:
64 to 1 024 Lines per revolution
2 Channels
Digital output

Series IE2 – 1024

		IE2 – 64	IE2 – 128	IE2 – 256	IE2 – 512	IE2 – 1 024	
Lines per revolution	N	64	128	256	512	1 024	
Signal output, square wave		2					channels
Supply voltage	V_{DD}	4,5 … 5,5					V DC
Current consumption, typical (V_{DD} = 5 V DC)	I_{DD}	typ. 6, max. 12				typ. 8,5	mA
Output current, max. [1]	I_{OUT}	5					mA
Phase shift, channel A to B	Φ	90 ± 45					°e
Signal rise/fall time, max. (C_{LOAD} = 50 pF)	tr/tf	0,1 / 0,1					µs
Frequency range [2], up to	f	20	40	80	160	300	kHz
Inertia of code disc [3]	J	0,09					gcm²
Operating temperature range		– 25 … + 85					°C

[1] V_{DD} = 5 V DC: Low logic level < 0,5 V, high logic level > 4,5 V: CMOS and TTL compatible
[2] Velocity (rpm) = f (Hz) x 60/N
[3] For the brushless DC-Servomotors the inertia of code disc is J = 0,14 gcm²

Ordering information

Encoder	number of channels	lines per revolution	in combination with:
IE2 – 64	2	64	DC-Micromotors
IE2 – 128	2	128	1336 … CXR,
IE2 – 256	2	256	1516 … SR, 1524 … SR,
IE2 – 512	2	512	1717 … SR, 1724 … SR,
IE2 – 1 024	2	1024	1727 … C, 1741 … CXR
			2224 … SR, 2232 … SR, 2342 … CR,
			2642 … CR, 2657 … CR,
			3242 … CR, 3257 … CR, 3272 … CR
			3863 … C, 3863 … CR
			Brushless DC-Servomotors
			1628 … B, 2036 … B, 2057 … B,
			2444 … B

Features

These incremental shaft encoders in combination with the FAULHABER DC-Micromotors and Brushless DC-Servomotors are used for the indication and control of both shaft velocity and direction of rotation as well as for positioning.

The encoder is integrated in the DC-Micromotors SR-Series and extends the overall length by only 1,4 mm. Built-on option for DC-Micromotors and Brushless DC-Servomotors.

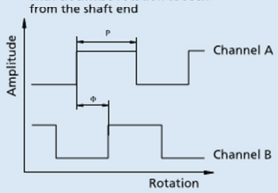
Hybrid circuits with sensors and a low inertia magnetic disc provide two channels with 90° phase shift.

The supply voltage for the encoder and the DC-Micromotor as well as the two channel output signals are interfaced through a ribbon cable with connector.

Details for the DC-Micromotors and suitable reduction gearheads are on separate catalogue pages.

Output signals / Circuit diagram / Connector information

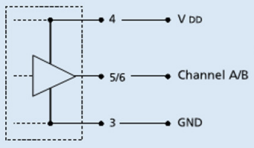
Output signals
with clockwise rotation as seen from the shaft end



Admissible deviation of phase shift:

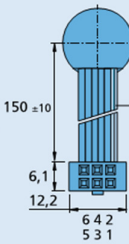
$$\Delta\Phi = \left| 90° - \frac{\Phi}{P} * 180° \right| \leq 45°$$

Output circuit



Pin Function

1 Motor – *
2 Motor + *
3 GND
4 V_{DD}
5 channel B
6 channel A



PVC-Ribbon cable
6 conductors 0,09 mm²

***Note:** The terminal resistance of all motors with precious metal commutation is increased by approx. 0.4 Ω, and the max. allowable motor current in combination is 1A. Motors with graphite commutation and brushless motors have separate motor leads and higher motor current is allowed.

Connector
DIN-41651
grid 2,54 mm