Real-Time Implementation of GPS aided Low Cost Strapdown Inertial Navigation System

A THESIS IN MECHATRONICS

Presented to the faculty of the American University of Sharjah
Collage of Engineering
in partial fulfillment of
the requirements for the degree of

MASTER OF SCIENCE

by

LAITH RASMI SAHAWNEH
B.S. 2002

Sharjah, UAE
April 2009

Real-Time Implementation of GPS aided Low Cost Strapdown Inertial Navigation System

Laith Rasmi Sahawneh, Candidate for Master of Science in Mechatronics Engineering

American University of Sharjah, 2009

ABSTRACT

As autonomous navigation becomes considerable and important because of the increased demand to their usage and benefits, therefore reliability and integrity issues become definite, specially when being implemented with commercially low-cost sensors. The objective of this thesis is to both develop and implement in real-time an INS/GPS integrated navigation system using the loosely-coupled linear Kalman filter. The importance of the implemented algorithms are to function appropriately and accurately using low cost inertial sensors where the rapid drift in sensors output requires a reliance on external and available aiding source as Global Positioning System (GPS).

This thesis describes the theoretical development and practical implementation in real-time of strapdown inertial navigation system (INS) using commercial of the shelf low cost inertial measurement unit (IMU) aided with the Global Positioning System (GPS). When describing the IMU as a "low cost", this term means that this unit is built using standard low grade accelerometers and gyros which cannot conduct self-alignment. Therefore the thesis provides a desirable calibration procedure where the requirements of a precisely controlled orientation of the IMU can be relaxed. To do so, the thesis describes in detail the design, construction, error modelling analysis and calibration approach of six-degree of freedom (6DOF) IMU. On the other hand, calibration of (IMU) unit is one of the most challenging issues in navigation field as it requires high accuracy measurements in order to maintain acceptable readings from sensors, and normally the cost of calibration platform often exceeds the cost of developing and constructing a MEMS sensor based IMU.

This thesis mainly discusses the development of inertial mechanization equations and algorithms that provides position, velocity and attitude of the host platform. Then, the data provided by inertial navigation mechanization is fused with GPS measurements using loosely-coupled linear. The accuracy of the estimation when utilizing a low-cost inertial navigation system (INS) is limited by the accuracy of the used sensors and the mathematical modelling of INS and the aiding sensors errors, however when fusing the INS data with GPS data, the errors can be bounded and the accuracy will increase. This thesis provides, both in practical and theoretical terms, the fusion processes adopted and real time implementation required for a high integrity aided inertial navigation system using state-of- art technology microcontroller MPC555 as the navigation computer. The theoretical work is verified by set of real-time experiments using our developed INS/GPS equipment -The developed IMU, standard GPS and MPC555-, the results have been compared with a top-notch INS/GPS Navigation device. The experimental results using our designed INS/GPS has shown that position and velocity accuracy can be archived using algorithms presented in this thesis work.

Contents

List of Figures

List of Tables

Nomenclature

Notation

$\mathbf{C}_x^y$	Direction cosine matrix transforming quantities from x frame to y frame
ω_{xy}^y	Angular rate of y frame relative to x frame expressed with components in y frame
$\mathbf{b}^x$	Vector $\mathbf{b}$ with components in x frame
$\underline{S}_{AB}$	True or actual value of quantity S measured in B axis of A sensor
$\delta\widehat{s}_{AB}$	The estimated error value of quantity S measured in B axis of A sensor
$\mathbf{\Omega}^x = [\omega^x \times]$	Skew symmetric matrix with components of ω in x frame

Axis System (Reference Frames), Angles and Transformations

i	Inertial reference frame
e	Earth-fixed reference frame
n	Navigation reference frame
b	Body reference frame
m	Mechanisation frame
h	Altitude
L	Latitude
l	Longitude
ϕ	Roll angle
θ	Pitch angle
ψ	Yaw angle

Earth Quantities (WGS-84)

g_n	Normal gravitational acceleration ($\varphi = 45^o$)
R	Equatorial radius of the Earth (semimajor axis) = 6378137.000 m
r	polar radius of the Earth (semiminor axis) = 6356752.3142 m
Ω	Earth turn rate with respect to i frame = 7.292116×10^{-5} rad/s
$\boldsymbol{g}_l$	Local gravity column matrix
f	Flattening (ellipticity) = 1/298.257223563 (0.00335281066474)
e	Major eccentricity of Earth = 0.0818191908426

μ	Earth's gravitational constant $= 3986005 \times 10^8 m^3/s^2$
M	Mass of Earth (including the atmosphere) $= 5.9733328 \times 10^{24}$ Kg

Dynamic Quantities

V_n	Normal gravitational acceleration ($\varphi = 45^o$)
$\boldsymbol{V}_e^n$	Kinematic velocity expressed in n frame
ω_{nb}^b	Angular rate of b frame relative to n frame expressed in b frame
ω_{en}^n	Angular rate of n frame relative to e frame expressed in n frame
ω_{ib}^b	Angular rate of b frame relative to n frame expressed in b frame
$\boldsymbol{f}^e$	Specific force in b frame
$\boldsymbol{f}^b$	Specific force in n frame
v_N, v_E, v_D	The north, east and down components of $\boldsymbol{V}_e^n$
f_N, f_E, f_D	The north, east and down components of $\boldsymbol{f}_e^n$

Subscripts

j, k, l	Indexes for high speed computer cycle (j-cycle), moderate computer speed cycle (k-cycle) and low computer speed cycle (l-cycle) respectively
N, E, D	North, East and Down components of n frame vector

Symbols

$(k\|k-1)$	Used to denote a quantity at time k immediately before the measurement update
$(k\|k)$	Used to denote a quantity at time k immediately after the measurement update
$N(k-1)$	A growing length measurement history vector consisting of all measurement vectors from 0 through $k-1$
N^k	Set of k observations

Abbreviations

6DOF	Six Degree-of-Freedom
AUS	American University of Sharjah
BDM	Background Debug Mode
CAN	Controllable Area Network
COTS	Commercial-Of-The-Shelf
DCM	Direction Cosine MAtrix
DGPS	Differential Global Positioning System
DoF	Degrees of Freedom

ECEF	Earth-Centered Earth-Fixed
ECI	Earth-Centered Inertial
GDOP	Geometric Dilution of Precision
GLONASS	Global Navigation Satellite System
GLR	Generalized Likelihood Ratio
GPS	Global Positioning System
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
KF	Kalman Filter
LKF	Linear Kalman Filter
MEMS	Micro-Electro-Mechanical Systems
MMSA	Minimal Mean Squared Error
NA	Navigation Algorithm
NASA	National Aeronautics and Space Administration
NED	North-East-Down
NEES	Normalized (state) Estimation Error Squared
NIS	Normalized Innovation Squared
PDF	Probability Distribution Function
PPS	Precision positioning System
PWM	Pulse Width Modulation
QADC	Quad Analog to Digital Convertor
SA	Selective Availability
SCT	Statistical Consistency Test
SDINS	Strapdown Inertial Navigation System
SLAM	Simultaneous Localization and Mapping
SV	Satellite Vehicle
UAV	Unmanned Aerial Vehicle

Acknowledgements

I would like to thank my supervisor Professor Mohammad-Ameen Al Jarrah for his continuous support, guidance and enthusiasm throughout my thesis work and all my master degree study. Professor Al Jarrah was always available and ready to give help whenever it was needed. I would also like to thank my co-advisor Dr Khalid Assaleh for his valuable advices and help during this research. Thanks as well goes to my co-advisor Dr. Mamoun Abdel-Hafez for his support and help during my thesis works.

I must give thanks to my colleagues and best friends in the Mechatronics Lab: Amer, Hussien and Younes; for the their help, all the discussions and the nights that we have spent running and experimenting the microcontrollers. For all my colleagues in the Mechatronics lab. We spent lots of days and nights on the test sites to make the system work, and finally watching it working which was the most exciting moment being here.

I owe special thanks to Tariq for his help and valuable advices and all the support he offers even while his short stay in UAE. Wish him the best luck to finish his PHD in ACFR.

I would like also to thank my cousin Rabea Sahawneh for the days and nights we spent studying in his office and in the camp, for his help in site tests. Special thanks goes to Mahir Rihani for his help and support during my stay in Sharjah (I am ready for ”Mansaf”).

Sincere thanks goes to Dirar Oweis for which his unlimited advocacy and help at my current job indirectly supports me to pursue my thesis works. Wish him and all of his family the success and prosperity.

Thanks to my roommates, work colleagues and friends in Abu Dhabi for their help and understanding, thanks goes to Tariq (Abu Ramez) and his wife Mary, Narcis, Amer, Fadi and Abed Al-Hay.

To my parents whom my words can't describe, for their endless love, support and enthusiasms. To my brothers Qais and Ehab and to my sister Diana. To my wife $Safa'a$ for her love, support and understanding.

To Knowledge.

Chapter 1

Introduction

1.1 Background

Nowadays, Inertial Navigation systems are becoming considerable specification of most vehicle's electronics systems, a motivation towards unmanned vehicles and autonomous driving life style! however, instead of being a luxury option; yet it is a crucial demand in designing the unmanned vehicles. Most navigation systems rely on Global Positing System (GPS) receivers as the main source of information to provide the position and velocity of the host vehicle, however GPS can provide a precise, long term and stable position information but within certain conditions i.e. open areas, on the other hand the received signal from satellites could be blocked or attenuated by obstacles, besides that the low-cost GPS receivers can be easily degraded in high maneuvering environments. Hence, in environments where the GPS signal propagation is attenuated or blocked the need for different sensors with complementary properties for robustness and redundancy becomes vital. One of available solutions is the INS/GPS integration which becomes a necessary, yet a crucial design requirement for unmanned navigation.

In addition to that navigation systems in many civilian and military applications require robust and exact position information, therefore the navigation system that based on the stand alone GPS is very sensitive to orbiting satellites, besides the Inertial Navigation Systems (INS)is a self-contained system, which means that the position of the vehicle is calculated using measurements of accelerations and angular velocities provided by accelerometers and gyros sensors, thus it can be argued that INS is a robust and jamming resistant navigation system but the error in the provided position data increases without bonds due to the integration of measurement error in the accelerometers and gyros. However high accuracy inertial navigation system specially the gimballed based structure system are expensive, restricted and unavailable commercially due to many different reasons. That makes the strapdown -Micro Electro Mechanical System (MEMS) based sensors- inertial navigation systems a considerable and affordable option to be chosen. Additionally, the GPS delivers position information at slower rate, typically a few times per second and the error in the position is bounded i.e. a lower rate compare to the INS system which make it a suitable to be integrated.

Consequently, and for these discussed reasons and because of these complementary properties for both systems, it can be concluded that by integrating both GPS and MEMS based INS; a compact navigation system of small size and weight can be achieved with modest power consumption and cost compare to the expensive and power consuming high advanced accuracy Inertial Navigation Systems. However, the commercial off-the shelf MEMS based INS are not ideal, the error characteristics are often high, non-linear and temperature dependant so as it is essentially to model and determine the most dominant errors that affect the desired accuracy of the navigation process.

1.2 Thesis Objectives

This thesis aims to design and develop a low-cost inertial navigation system (INS) and integrate it with the Global Positioning System (GPS). The design is intended for use by a low cost inertial measurement unit (IMU) to provide accurate navigation information, as position, velocity and attitude, of the host vehicle. The term ”low-cost IMU” is used to describe an IMU built with standard low grade gyros and accelerometers sensors. To achieve the design development, three issues are addressed in this thesis:

- Design, construct, calibrate and evaluate a MEMS based IMU unit.

- Development of INS error models appropriate for the designed low cost IMUs which determine the accuracy and behavior of the INS.
- Implement an INS/GPS integrated navigation system using the loosely-coupled linear Kalman filter. This requires designing and tuning the filter which implies log data in real-time using micro-controller as logged data will be used to run Kalman filter on desktop for the purpose of filter tuning.
- Construct the hardware and develop the software of the navigation system, and run the filter in real-time site experiments.

1.2.1 Contributions

The main contributions of this thesis are as follows:

- The design and development of Inertial measurement unit (IMU) using low grade MEMS based sensors and commercially available with affordable costs.
- Qualification and quantification of errors associated with Inertial measurement unit (IMU), model the errors and develop a low cost desirable calibration procedure where the requirements of a precisely controlled orientation of the IMU can be relaxed achieving an error model with an acceptable accuracy.
- Implementing and demonstration of real-time low cost GPS/Inertial navigation system using loosely coupled linear kalman filter algorithm using top-notch microcontroller MPC555. This compromise a detailed analysis of inertial navigation mechanization, Kalman filter operation, including filter consistency, tuning, discretization, and integrity. It also includes an investigation in real-time implementation issues including data latency and computing load.

1.2.2 Thesis Structure

The structure of this thesis is as follows:

- Chapter 2 discusses and compares between the strapdown inertial navigation and the gimballed construction. This chapter presents the adapted INS construction as corresponding common sensor errors and then moves to present different coordinate frames in which navigation process can be defined. This chapter is also presents navigation equations and developed the mathematical background for strapdown inertial navigation mechanization algorithm in the local geographical navigation frame (NED) navigation frame.

- Chapter 3 discusses different IMU units constructions presents the design specification and sensors selection criteria. This chapter formulate and developed the error model of the constructed IMU. Finally the calibration algorithm is provided with the corresponding results demonstrating the tools, equipment and mathematical approach.

- Chapter 4 discusses the Global Positioning System as a navigation aided source and presents the integration of GPS with inertial systems which become a subject of research in this field coupled with advances in estimation processing techniques and high-speed processors. This chapter presents the optimal linear estimator and discuses it in a single dimension, then it proposes the loosely coupled linear kalman filter with its mathematical structure then the chapter moves to provide the discrete-time filter formulation as it is the practical case which will be transferred into code and download it to the navigation computer.

- Chapter 5 concerns with the real time implementation of GPS aided INS navigation system. In this chapter the navigation equation will be developed in discrete form, this chapter formulate the attitude (Psi angle approach), position and velocity error equation in the discrete form. There equations are then expressed in a single error equation matrix using state space representation. Finally, the direct configuration is considered as the fusion scheme of the observed (GPS) measurements and estimated INS data, the filter arcitechture is formulated and implemeted with its different model stages, namely, process, prediction, update prediction and error correction stage. Filter initialization, filter tuning procedure and initial alignment and calibration are also discussed as an important real-time implementation aspects.

- Chapter 6 provides an overview of all sensors and equipment that have been utilized in this thesis work. This chapter provides INS/GPS navigation system schematics, and data flow diagrams and algorithm flow charts and at the end of this chapter the results of site-test are presented along with the conclusions and future work suggestions.

Chapter 2

Strapdown Inertial Navigation System

2.1 Introduction

Historically, people moved from place to another using their own sense or any other surrounded nature features as an aid for finding or 'knowing' their way; these skills have required some form of navigation. Then as the desire developed to explore more, civilization started to develop instruments for navigation applications. More recently, there have been significant developments in inertial sensors, and systems for inertial navigation for land, air, and marines as well as in the space to planets and beyond. Recently inertial navigation systems have been used in a wide range of civilian and military applications; however by the motivated industry a whole variety of low cost inertial instruments are becoming recently available in diverse applications.

Inertial sensors measure acceleration and rotation rate by accelerometers and gyroscopes, respectively. The input axis of an inertial sensor defines which vector component it measures and multi-axis sensors measure more than one component accordingly. Inertial navigation uses gyroscopes and accelerometers to maintain an estimate of the position, velocity and attitude rates of the vehicle in which the INS system is hosted. A complete INS system is consisted of inertial measurement unit (IMU) which contains accelerometers and gyroscopes triad sets and a navigation computer. The navigation computer is mainly used to read, fuse and integrate different sensors information in order to provide the needed position, velocity and attitude.

Literally, there are thousands designs for gyros and accelerometers sensors with many which are available in the market. However not all of them are used for inertial navigation, i.e. acceleration sensors are used for measuring vibrations, levelling and measuring gravity. In the same manner rate-gyros sensors are also used for steering and stabilizing missiles, submarines, ships and cameras etc.

Traditionally,the inertial navigation systems are designed in two main groups:

- The platform (or gimbaled) systems.
- The strapdown systems.

In a gimbaled system the accelerometer and gyros triad is rigidly mounted on the inner gimbal with pre-specific known direction as shown in figure 2.1, and it preserves this direction even the pose and orientation of the hosted vehicle is changing. In other words, the inner gimbal is isolated from the vehicle rotations, and its attitude remains constant in a desired orientation during the motion of the system. The gyroscopes which are constructed on a stable platform are used to sense any rotation of the platform, and their outputs are used in servo feedback loops with gimbal pivot torque actuators to control the gimbals such that the platform remains stable. These systems are very accurate, because the sensors can be designed for very precise measurements in a small measurement range, this complicated control process makes such a system accurate and highly expensive in the same time. In contrary, a strapdown inertial navigation system uses orthogonal accelerometers and gyro triads rigidly fixed to the axes of the moving vehicle as shown in figure 2.1. The angular motion of the system is continuously measured using the rate sensors. The accelerometers do not remain stable in space, but follow the motion of the vehicle and orientation is corrected using the navigation computers.

In a further categorization,inertial systems are mainly divided into three groups with respect to the free-running growth of their position error [1]:

- The strategic-grade instruments (performance . 100 ft/h).
- The navigation-grade instruments (performance . 1 nm/h).
- The tactical-grade instruments (performance . 10-20 nm/h).

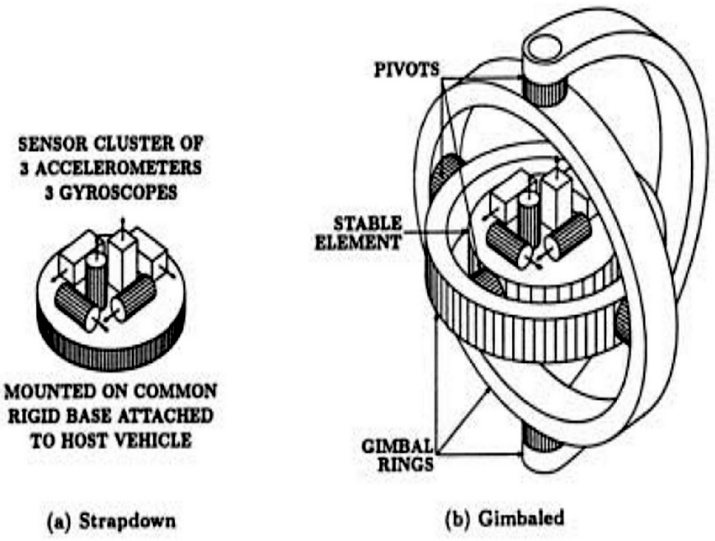


Figure 2.1: Inertial measurement units physical construction, Strapdown versus Gimbaled. (Grewal et al. 2001).

2.1.1 Common Sensor Error Models

There are many types of acceleration sensors which have several designs such as, strain-sensing accelerometers, pendulous accelerometers and gyroscopic accelerometers, etc. The main error sources for accelerometers are biases, input axis misalignments, centrifugal acceleration effects due to high rotation, vibrations rates and temperature drift etc.

The rate-gyros as well have many designs, which are used as attitude sensors in inertial navigation, such as laser gyroscopes, momentum wheels, and rotating multisensor, etc. Error models for gyroscopes are primarily used for two purposes:

- Predicting performance characteristics as function of gyroscope design parameters.
- Calibration and compensation of output errors.

The common error sources for gyroscopes are output bias, input axis misalignments, input/output non-linearity, vibrations, temperature drift and acceleration sensitivity,

2.1.2 Initialization and Alignment

Starpdown INS initialization and alignment will be discussed in the follwoing section as it is one of the most important issues concerns the starpdown navigation. INS navigation initialization is the process of determining initial values for system position, velocity and attitude defined in one or more of the different navigation coordinate frames, or in another words determining the initial values of the coordinate transformation from sensor coordinates to navigation coordinates (for strapdown systems). INS position initialization ordinarily relies on external sources such as GPS or manual entry, however the velocity initialization can be accomplished by starting when it is zero (i.e., the host vehicle is not moving) or by reference to the carrier velocity if any. There are four basic methods for INS alignment [2]:

- The optical alignment in which either optical line-of-sight reference to a ground based direction or an on board star tracker is used for aligning.
- For stationary vehicles, in which the sensed direction of acceleration to determine the local vertical and sensed direction of rotation to determine north is used. This is called gyrocompass alignment approach.
- Transfer alignment in a moving host vehicle, where velocity matching with an aligned and operating INS can be used for alignment.
- GPS-aided alignment, in which position matching with GPS in order to estimate the alignment variables.

2.1.3 Inertial Navigation System Error Models

Since there is no single, standard design for an INS, therefore the error sources of INS systems vary accordingly, generally, error sources can be classified into the following groups:

- Initialization errors, which comes from initial estimates of position and velocity.

- Alignment errors, that comes from attitude direction cosines used for strapdown systems navigation algorithms with respect to navigation axes.
- Sensor compensation errors, occur due to the change in the initial sensor calibration over the time;
- Gravity model errors, is the influence of the unknown gravity modeling errors on vehicle dynamics.

2.2 Coordinate Frames

Before moving to formulate the inertial navigation equations, it is crucial to present navigation coordinates as it is the most fundamentals to the process of inertial navigation. Each frame is an orthogonal right handed axis set. The following INS navigation coordinate frames are defined and used in this thesis [3] (*For more details refer to Appendix A*):

- *Inertial frame (i-frame)*: An inertial frame is a reference frame in which Newton's laws of motion holds. The origin is at the center of Earth and axes are non-rotating with respect to the fixed stars. All inertial measurement is relative to this frame.
- *Earth frame(e-frame)*: is an earth-fixed frame whose origin is at the Earth center, x axis points to the north pole, y axis points to Greenwich meridian in the equatorial plane and the z axis completes the system to a right-hand coordinate system.
- *Body frame (b-frame)*: is a frame fixed to the vehicle. The accelerations and angular rates generated by the strapdown accelerometers and gyros sensors are measured in the body frame. The axes are defined as forward, right and down. The objective of navigation is to determine the position and attitude of a vehicle based on measurements from accelerometers and gyros sensors attached to the vehicle.

- *Platform frame (p-frame)*: is the frame which the transformed accelerations from the accelerometers and angular rates from gyros are solved.
- *Navigation frame (n-frame)*: is a user defined frame for navigation output. Any frame defined above can be chosen as the navigation frame. The navigation frame is commonly used to describe the navigation of a vehicle in a local coordinate frame, to provide local directions north, east and down. Therefore the axes of the frame is defined in these directions (NED), another definition of the navigation frame is ENU. The down direction is defined from the down direction of a local tangent plane and therefore it does not always concede with the direction of earth's center.

The Global Positioning System (GPS) is another navigation source used in this thesis. The World Geodetic System WGS-84 is a coordinate system for GPS, hence WGS-84 (*For more details refer to Appendix A*) is an Earth-centered Earth-fixed reference frame defined as follows [4]:

Origin: Earth's center of mass.

x-axis: Intersection of the WGS-84 reference meridian plane and the plane of the mean astronomic equator, the reference meridian being parallel to the zero meridian defined by the Bureau International De L'HEURE on the basis of the longitudes adopted for the BIH stations.

y-axis: Completes a right-handed Earth-centered, Earth-fixed orthogonal coordinate system, measured in the plan of the mean astronomic equator 90 degrees east of the x-axis.

z-axis: Parallel to the the direction of the conventional international origin for polar motion, as defined by the Bureau International De L'HEURE (BIH) on the basis of the latitude adopted for the BIH stations.

2.3 Equations of Navigation (overview)

Back to the basic physics inertia as it is defined as the natural tendency of bodies to maintain constant translational and rotational velocity, unless disturbed by forces or torques, respectively (Newton's first law of motion). An inertial reference frame is a coordinate frame that is neither rotating nor translating and, therefore, at which Newton's laws of motion are valid.

$$F = \frac{d}{dt}(m\dot{X}) \tag{2.1}$$

Where F is the sum of applied forces, gravity is excluded, on the body with constant mass m and position x. Equation 2.1 needs to be modified to account for the earth's gravitational field where g is the gravitational acceleration vector.

$$m\ddot{X} = F + mg \tag{2.2}$$

The acceleration due to an applied force is

$$a = \frac{F}{m} \tag{2.3}$$

substituting that in equation 2.2 it becomes:

$$\ddot{X} = a + g \tag{2.4}$$

These equations only hold for the Newtonian inertial frame. Therefore if we look at the equation in the *i*-frame, 2.4 becomes:

$$\ddot{X}^i = g^i(X^i) + a^i \tag{2.5}$$

Where X^i is the position in the *i*-frame, g^i is the acceleration due to the gravitational field in the *i*-frame and depends on the position. a^i is the specific force sensed by the accelerometers. These equations are solved for the position X^i and velocity.

If this acceleration can be measured, then in principal if we integrate it the velocity and position of the object with respect to an initial condition can be determined [5]. Therefore inertial navigation is based on calculating position, velocity and orientation of a moving vehicle by using measurements from inertial sensors i.e accelerometers and gyros.

If we assume an arbitrary frame let us say n-frame that rotates with respect to the i-frame with an angular rate ω_a^i, hence a vector in the a-frame can be expressed in coordinates of the i-frame, by:

$$X^i = C_n^i X^n \tag{2.6}$$

Where $\mathbf{C}_n^i$ is the transformation matrix or the direction cosine matrix (DCM) transforming vectors from the a-frame to i-frame. The time derivative of this matrix is given by:

$$\dot{\mathbf{C}}_n^i = \mathbf{C}_n^i \mathbf{\Omega}_{in}^n \tag{2.7}$$

Where $\mathbf{\Omega}_{in}^n$ denotes a skew-symmetric matrix with elements from: $\omega_{in}^n = (\omega_1, \omega_2, \omega_3)$ according to:

$$\mathbf{\Omega}_{in}^n = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix} \tag{2.8}$$

Then the second time derivative becomes:

$$\ddot{C}_n^i = \mathbf{C}_n^i \dot{\mathbf{\Omega}}_{in}^n + \mathbf{C}_n^i \mathbf{\Omega}_{in}^n \mathbf{\Omega}_{in}^n \tag{2.9}$$

Using equations 2.7 and 2.9 and the second derivative of equation 2.6 with respect to time becomes:

$$\ddot{X}^i = \ddot{C}^i_n X^n + 2\dot{C}^i_a \dot{X}^n + \mathbf{C}^i_n \ddot{X}^n = \mathbf{C}^i_n \ddot{X}^n + 2\mathbf{C}^i_n \mathbf{\Omega}^n_{in} \dot{X}^n + \mathbf{C}^i_n (\dot{\mathbf{\Omega}}^n_{in} + \mathbf{\Omega}^n_{in} \mathbf{\Omega}^n_{in} X^n \tag{2.10}$$

Therefore solving for $\ddot{X}^n$ yields:

$$\ddot{X}^n = -2\mathbf{\Omega}^n_{in} \ddot{X}^n - \ddot{X}^n - (\dot{\mathbf{\Omega}}^n_{in} + \mathbf{\Omega}^n_{in} \mathbf{\Omega}^n_{in}) X^n + \mathbf{a}^n + \mathbf{g}^n \tag{2.11}$$

Where $\mathbf{a}^n = \mathbf{C}^n_i \mathbf{a}^i$, $\mathbf{g}^n = \mathbf{C}^n_i g^i(X^i)$. and $\dot{\mathbf{X}}^a = \frac{d}{dt}\mathbf{X}^a$. The equations 2.7- 2.11 define first order differential equations for the navigation states consisting of position, velocity and attitude driven by the IMU-signals $\mathbf{a}^b$ and ω^b_{ib}. if $\mathbf{X}$ denotes the total navigation state and $\mathbf{a}$ denotes the IMU-signals (both $\mathbf{a}^b$ and ω^b_{ib}) these differential equations may be written as:

$$\dot{\mathbf{X}}(t) = f(\mathbf{X}(t), \mathbf{a}(t)) \tag{2.12}$$

This equation will be referred as the *navigation equation.*

2.4 Strapdown Inertial Navigation Mechanization Equations

The navigation equation may be solved in any one of number of references frames. If the earth frame is chosen, for instance, then the solution of the navigation equation will provide estimates of velocity with respect to either inertial frame or the earth frame. The choice of mechanization is dependant on the application. The work in this thesis will be focused on inertial systems that will be used to navigate in the vicinity of earth. The estimates of position and velocity are derived by integrating a navigation equation of the form given in Equation. 2.5. In systems of the type described later, in which it is required to derive estimates of vehicles velocity and position with respect to an earth fixed frame, additional apparent forces will be acting which are functions of the reference frame motion. However should be noted that the variation in the mechanizations described here and elsewhere in the navigation literature are in the strapdown computational algorithms and not in the arrangement of the sensors or the mechanical layout of the system [3].

Navigation frame mechanization:

In order to navigate in the vicinity of earth, navigation information is most commonly required in the local geographic or navigation axis set described earlier. Position on the earth may be specified in terms of latitude (degrees north or south of a datum) and longitude (degrees east or west of a datum). Therefore navigation data are expressed in terms of north and east velocity components, latitude, longitude and height above the earth. For a terrestrial navigation system operating in the local geographical reference frame, the navigation equation can be expressed as follows [3]:

$$\dot{\boldsymbol{V}}_e^n = \mathbf{f}^n - [2\omega_{\mathbf{ie}}^{\mathbf{n}} + \omega_{\mathbf{en}}^{\mathbf{n}}] \times \boldsymbol{V}_e^n + \boldsymbol{g}_l^n \tag{2.13}$$

And the position can also be calculated by the following differential equation

$$\dot{\boldsymbol{P}}_e^n = \boldsymbol{V}_e^n \tag{2.14}$$

Where, $\boldsymbol{V}_e^n$ represents velocity with respect to the earth expressed in the local geographic frame defined by the direction of true north, east and the local vertical, in the component form:

$$\boldsymbol{V}_e^n = \begin{pmatrix} v_N & v_E & v_D \end{pmatrix}^T \tag{2.15}$$

$\mathbf{f}^n$ is the specific force measured by a triad of accelerometers and resolved into the local geographic reference frame $\boldsymbol{C}_b^n \boldsymbol{f}^b$ in component form:

$$\mathbf{f}^n = \begin{pmatrix} f_N & f_E & f_D \end{pmatrix}^T \tag{2.16}$$

and $\omega_{\mathbf{ie}}^{\mathbf{n}}$ is the turn rate of earth expressed in the local geographic frame, expressed in component form as follows:

$$\omega_{ie}^{n} = \begin{pmatrix} \Omega\cos(L) & 0 & -\Omega\sin(L) \end{pmatrix}^{T} \tag{2.17}$$

Where $\omega_{\mathbf{en}}^{\mathbf{n}}$ represents the turn rate of the local geographic frame with respect to the earth-fixed frame. As well this quantity can be expressed in terms of the rate change of latitude and longitude as follows:

$$\omega_{en}^{n} = \begin{pmatrix} \dot{l}\cos(L) & -\dot{L} & -\dot{l}\sin(L) \end{pmatrix}^{T} \tag{2.18}$$

Where $\dot{l} = v_E/(R_m + h)\cos(L)$ and $\dot{L} = v_N/R_m + h)$ yields:

$$\omega_{en}^{n} = \begin{pmatrix} \frac{v_E}{R_m+h} & -\frac{v_N}{R_m+h} & -\frac{v_E\tan(L)}{R_m+h} \end{pmatrix}^{T} \tag{2.19}$$

Where R_m is the radius of the earth and h is the height above the surface of earth. $\boldsymbol{g}_l^n$ is the local gravity vector which includes the combined effects of the mass attraction of the earth ($\mathbf{g}$) and the centripetal acceleration caused by the earth's rotation ($\omega_{ie}, \times\omega_{ie}, \times\mathbf{R}$) therefore the the local gravity vector can be written as:

$$\boldsymbol{g}_l^n = \mathbf{g} - \omega_{ie} \times \omega_{ie} \times \mathbf{R} = \mathbf{g} - \frac{\Omega^2(R_m + h)}{2} \begin{pmatrix} \sin(2L) \\ 0 \\ 1 + \cos(2L) \end{pmatrix} \tag{2.20}$$

In component form the navigation equation is as follows:

$$\dot{v_N} = f_N - v_E(2\Omega + \dot{l})\sin(L) + v_D\dot{L} + \xi g = f_N - 2\Omega v_E\sin(L) + \frac{v_N v_D - v_E^2\tan(L)}{R_m + h} + \xi g \tag{2.21}$$

$$\dot{v_E} = f_E + v_N(2\Omega + \dot{l})\sin(L) + \dot{v_D}(2\Omega + \dot{l})\cos(L) - \eta g \tag{2.22}$$

$$= f_E + 2\Omega(v_N\sin(L) + v_D\cos(L) + \frac{v_E}{R_m + h}(v_D + v_N\tan(L)) - \eta g \tag{2.23}$$

$$\dot{v_D} = f_D - v_E(2\Omega + \dot{l})\cos(L) - v_N\dot{L} + g \tag{2.24}$$

$$= f_D - 2\Omega v_E\cos(L) - \frac{v_E^2 + v_N^2}{R_m + h} + g \tag{2.25}$$

ξ and η represents angular deflections in the direction of the local gravity vector with respect to the local vertical owing to the gravity anomalies.(*For more details about the shape of the earth, datum reference models refer to appendix A*).

However the deflection of the local gravity vector from the vertical mat be expressed as angular deviations about the north and the east axes of the local geographic frame as follows :

$$\mathbf{g}_l^n = \begin{pmatrix} \xi g & -\eta g & g \end{pmatrix}^T \tag{2.26}$$

Latitude (L), longitude (l) and altitude (h) above the surface of the earth are given by:

$$\dot{L} = \frac{1}{(R_m + h)}v_N \tag{2.27}$$

$$\dot{l} = \frac{v_E\sec(L)}{(R_m + h)} \tag{2.28}$$

$$= \frac{v_E}{(R_M + h)\cos(L)} \tag{2.29}$$

$$\dot{h} = -V_D \tag{2.30}$$

A block diagram representation of the navigation frame mechanization is shown in Figure 2.2.

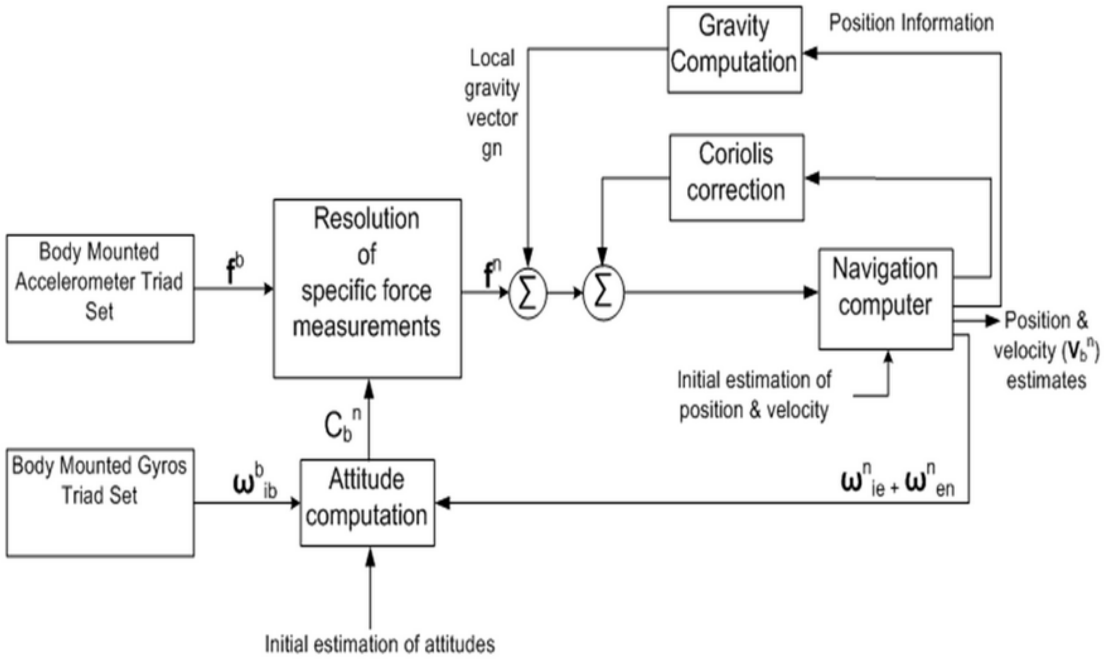

Gravity Computation
Position Information
Local gravity vector gn
Coriolis correction
Body Mounted Accelerometer Triad Set
f^b
Resolution of specific force measurements
f^n
Σ
Σ
Navigation computer
Position & velocity (V_b^n) estimates
Initial estimation of position & velocity
C_b^n
Body Mounted Gyros Triad Set
ω_{ib}^b
Attitude computation
$\omega_{ie}^n + \omega_{en}^n$
Initial estimation of attitudes

Figure 2.2: Strapdown inertial navigation system-local geographic navigation frame

2.4.1 Effect of Variation in the Earths Gravitational Field on The Navigation Equations

In the above equations it is assumed that the earth is perfectly spherical in shape and there are no variation in the earth's gravitational field with respect to the changes in position and altitude of the navigation system. The spherical model assumed so far is not sufficient representative for very accurate navigation, taking into account the slight faltering of the earth at the poles and approximating the earth as a reference ellipsoid which is more closely to the true geometry of the earth, therefore the above equation can be modified in order to take in account the errors introduced with the above assumptions [6] [7].

The rate of change of latitude and longitude can be expressed as follows:

$$\dot{L} = \frac{1}{(R_{Mr} + h)} v_N \tag{2.31}$$

$$\dot{l} = \frac{v_E \sec(L)}{(R_{Tr} + h)} \tag{2.32}$$

$$= \frac{v_E}{(R_{Tr} + h)\cos(L)} \tag{2.33}$$

Where R_{Mr} is the meridian radius of curvature and R_{Tr} is the transverse radius of curvature. Similarly the transport rate becomes:

$$\omega_{en}^{n} = [\frac{v_E}{R_{Mr} + h} - \frac{v_N}{R_{Tr} + h} - \frac{v_E \tan(L)}{R_{Tr} + h}]^T \tag{2.34}$$

And regarding the variation of gravity with position (latitude) and height there are several models given in the literature. The following expressions for the variation of the magnitude of the gravity with latitude at sea level ($h = 0$) and its rate of change with respect to the height above earth surface; are given by Steiler and Winter [7]:

$$g(0) = 9.780318 \left(1 + 5.3024 \times 10^{-3} \sin^2(L) - 5.9 \times 10^{-6} \sin^2(2L)\right) m/s^2 \tag{2.35}$$

$$\frac{dg(0)}{dh} = -3.0877 \times 10^{-6}(1 - 1.39 \times 10^{-3} \sin^2(L)) m/s^2/m \tag{2.36}$$

$$g(h) = \frac{g(0)}{(1 + h/R_m)^2} \tag{2.37}$$

The gravity component was evaluated for the AUS university area in Sharjah at latitude 25.310278° resulting in a gravity component of 9.7898 m/s^2.

Chapter 3

Design, Modelling and Calibration of A MEMS IMU

3.1 Introduction

The calibration of Inertial Measurement Unit (IMU) sensors is one of the most challenging issues in navigation field. It requires high accuracy measurements in order to maintain acceptable readings from sensors. Most of commercial low-grade IMUs have deviation of about six degrees per hour, which requires updating data every certain period of time not to lose the desired path. However, advances in MEMS technology combined with minimization of electronics, have made it possible to produce chip-based inertial sensor for use in measuring angular velocity and acceleration [8].

IMU calibration and error modelling has been studied extensively and wide research efforts were done in this field due to the importance of IMU as it is the core part of inertial navigation system. For instance [9], [10] proposed a calibration procedure for the IMU, where the requirements of a precise control of the IMUs orientation is relaxed. This calibration method utilizes the fact that ideally the norm of the measured output of the accelerometer and gyro cluster should be equal to the magnitude of the applied force and rotational velocity, respectively. However there is one major disadvantage with such a method; not all sensor parameters of the IMU are observable. This implies that these parameters (error sources) must be taken into account in the integration of the IMU with aided system, which implies that the complexity of the data fusion algorithm will increase accordingly [11]. [12]

develop an electro-mechanical system to calibrate the IMU using Global Positioning System (GPS) with four antennas. An alternate improved calibration system (Carpal Wrist) based on variable-geometry truss (VGT) was presented. The hardware part was designed for maximum pitch and roll angular and vertical displacements. Controller implementation with Cartesian kinematics has been performed with hardware response for pitch, roll, and yaw motions commanded on at a time.

3.2 Design Specifications and Sensors Selection

Inertial Measurement Unit (IMU) is the part of the inertial navigation system as it is responsible to determine the pose of the vehicle through the integration output readings provided of inertial sensors. The IMU is a six-degree of freedom inertial measurement unit consisting of three accelerometers and three orthogonal gyros to provide measurement of acceleration in three dimensions and rotation rates about three axes constructed in a strap-down configuration; which means that all accelerometers and gyros are fixed to a common platform and are not allowed to be controlled on gimbals; this design will eliminate the expensive moving parts mechanism required to align sensors in a pre-specified direction but on the other hand it requires a complex mathematical computation algorithm in order to find the true vehicle position, velocity and attitude.

The selection criteria of sensors mainly based on the application that sensors will utilized for, literally the application will determine the required accuracy and reliability etc of the IMU, accordingly the design parameters will determine the specification of the required sensors to be chosen from. The developed IMU will be a part of an avionics unit that can be implemented in a small unmanned vehicle in which it will not experience acceleration exceeds $\pm 2g$ as well as the angular rotation will not exceed $\pm 150^\circ/s$; these dynamics that the unmanned vehicle undertakes will be the aspects that sensors should cover. The availability in the market. The cost of the sensor is also an important factor taking into that this reduction in cost generally brings about reduction in reliability and accuracy. However the design, calibration and implementation of IMU using commercially of-the-shelf components including sensors in order fit the required application aspects with certain accuracy and reliability is really a challenging task to be accomplished.

The accelerometers are selected from a wide range of MEMS-based sensors produced by Analog Devices Inc. The accelerometer board is assembled from two orthogonal low cost ±2g dual axis accelerometer (ADXL202EB) [13] with high sensitivity ($12.5g$) that allows using a lower speed counter for its PWM output. The (ADXL202EB) is a simple evaluation board that simplifies the explantation of the ADXL202 performance [14]with only three additional through-hole passive components. The (ADXL202EB) is provided with a Simulink behavior model as a support tool to ease the understanding, simulation and evaluation of the sensor behavior.

As the (ADXL202EB) evaluation board provides dual axis ±2 g accelerometers; therefore we need two chips in order to measure the acceleration in three directions. From the same vendor three $\pm 150°/s$ (ADXRS150EB) [15] gyro sensors are selected to measure the angular rate about the accelerometers sensitive axes. The ADXRS150EB is a simple evaluation board as that allows to evaluate the performance of the ADXRS150 [16]. No additional external components are required for operation. The ADXRS150EB is still among the smallest gyros available today which has a 20-lead dual-in-line (0.3 inch width by 0.1 inch pin spacing) and can be easily mounted on PCB boards. Both selected accelerometer and rate-gyro sensors are shown in figure 3.1.

The IMU is designed to be stand-alone module in which it can be plugged and removed easily from the whole avionics unit, this will give the ability to perform a flexible calibration and troubleshooting -if any-. Figure 3.2 shows the developed Inertial Measurement Unit. The IMU unit equipped with a regulated power supply circuit that ensure a polarity protected input voltage for the different sensors boards and the internal temperature of the unit is also monitored in order to provide the operation temperature inside the unit, as this is an important factor affecting the

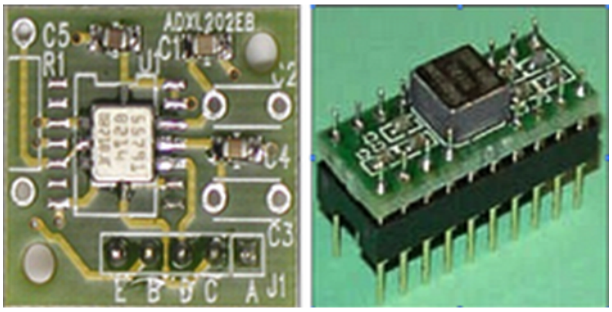


Figure 3.1: ADXL202EB accelerometer evaluation board sensor (Right) and ADXRS150EB rate-gyro sensor (Left),(*Analog Devices Inc.*)

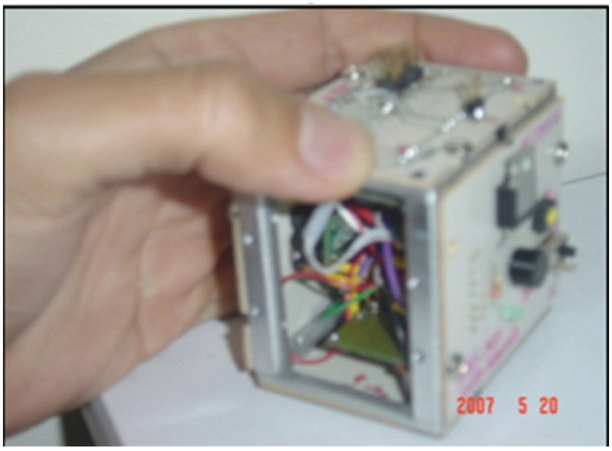
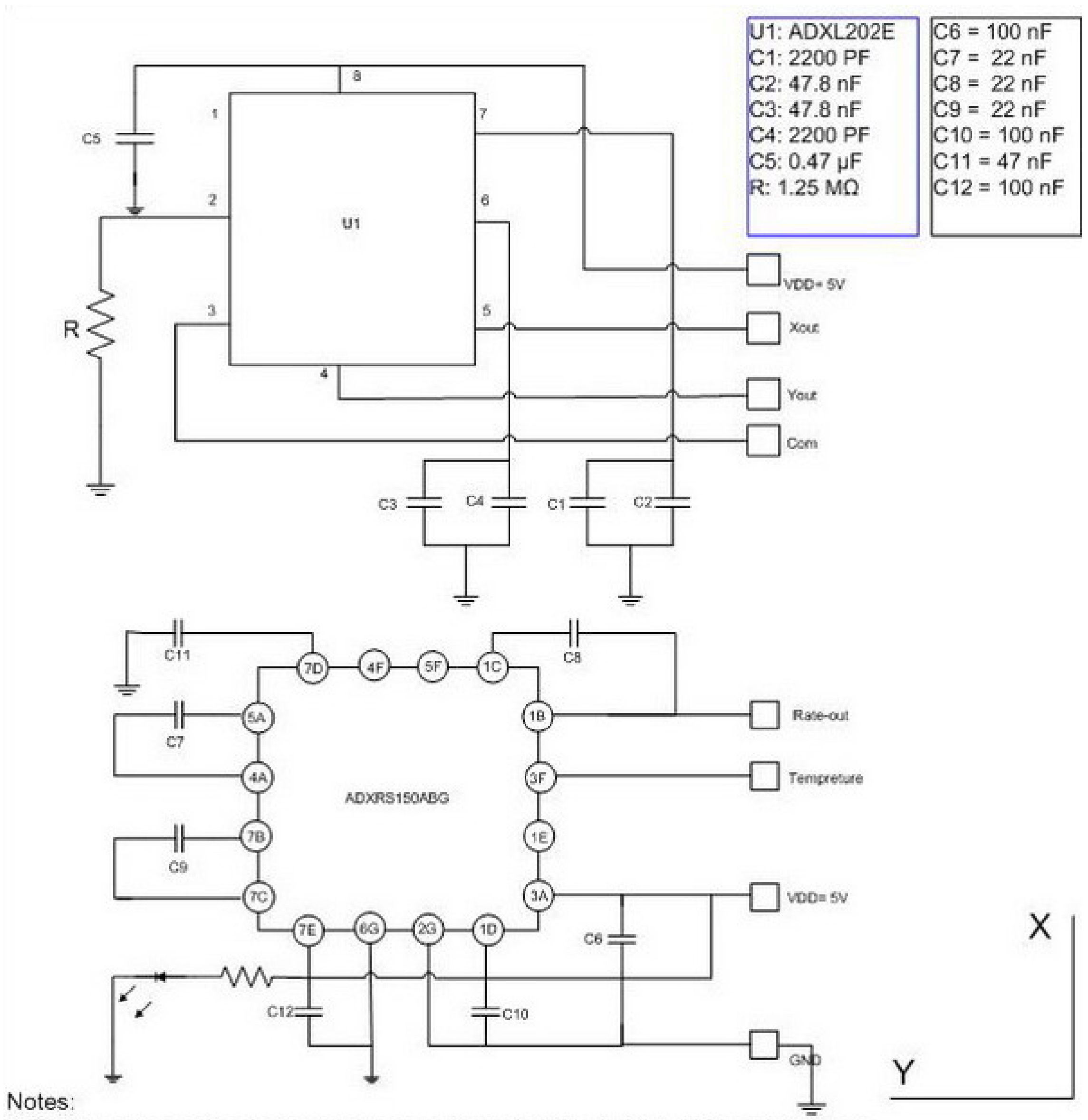


Figure 3.2: Constructed (6-DOF) IMU unit, which consists mainly from triad of gyros and accelerometers sensors. The unit dimensions are (5.9 × 4.6 × 5.8) cm.

sensors response, for instance sensors sensitivity drifts with temperature. Figures 3.3, 3.4, 3.5 3.6 and 3.22 shows the circuits schematics and PCB layout of different IMU boards.

3.3 Sensors Error Model

Based on Newton's second law of motion; the acceleration of an object is produced by net forces is directly proportional to the magnitude of the net forces acting on the

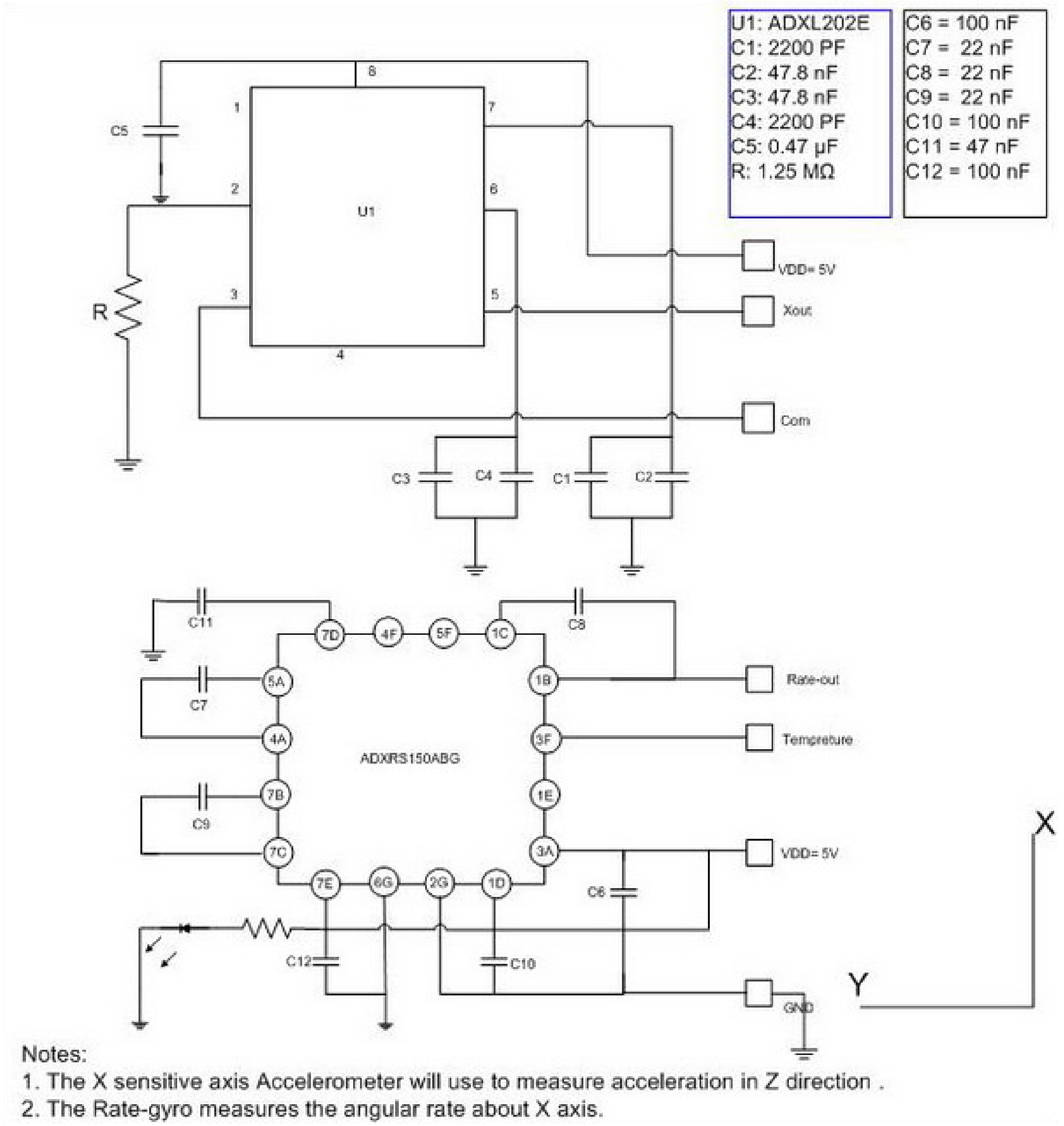


Notes:
1. The Accelerometer dual axes measures acceleration in X and Y sensitive axes.
2. The Rate-gyro measures the angular rate about Z axis.

Figure 3.3: Bottom board of IMU which consists of rate-gyro and 2-axis accelerometrs, this configuration will provide rotation around z-axis and acceleration in both x and y axes.

body, and inversely proportional to the mass of the object:

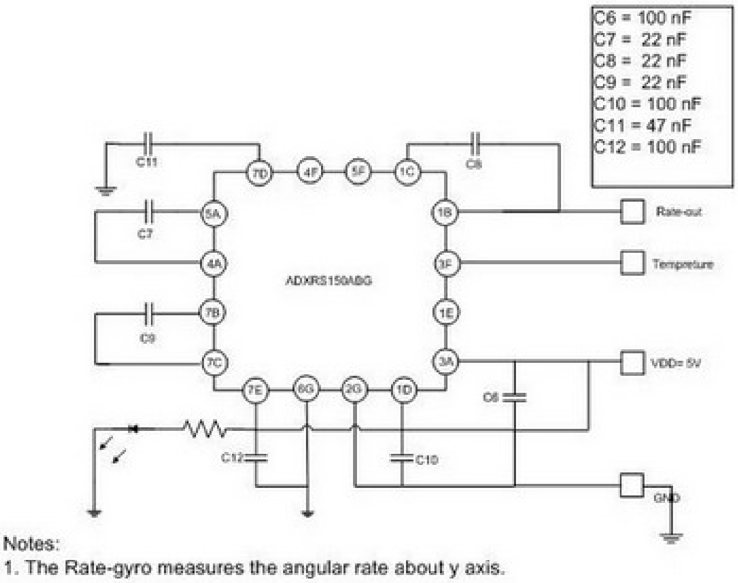
$$\frac{f_{net}}{m} = a = F_s \tag{3.1}$$



Notes:
1. The X sensitive axis Accelerometer will use to measure acceleration in Z direction .
2. The Rate-gyro measures the angular rate about X axis.

Figure 3.4: Back board of IMU which consists of rate-gyro and 2-axis accelerometrs, this configuration will provide rotation around x-axis and acceleration in z direction.

Where F_s is the specific force and a is the acceleration, which is independent on the mass. The accelerometer detect absolute accelerations due to forces applied on the body with respect to an inertial frame. The object here is to detect the translation acceleration; thus the calibration should compensate for other accelerations,



Notes:
1. The Rate-gyro measures the angular rate about y axis.

Figure 3.5: Side board of IMU which consists of rate-gyro sensor, this configuration will provide rotation around y-axis.

for example when inertial navigation system is fixed in aerial vehicles the unwanted acceleration which should be removed is the acceleration due to gravity. The acceleration of an arbitrary point i.e. p with respect to a space-fixed axis set, say i-frame and denoted by the subscript i is defined by:

$$\mathbf{a}_i = \frac{d^2\mathbf{r}}{dt^2}|_i \tag{3.2}$$

Where $\mathbf{r}$ represent the position vector of the point p with respect to the origin of the reference frame.

A triad of perfect accelerometers will provide a measure of the specific force ($\mathbf{f}_s$) acting at point P where:

$$\mathbf{f}_s = \frac{d^2\mathbf{r}}{dt^2}|_i - \mathbf{g} \tag{3.3}$$

Where $\mathbf{g}$ is the mass attraction gravitation vector. Rearranging equation 3.3 yields the following equation:

$$\frac{d^2\mathbf{r}}{dt^2}|_i = \mathbf{f}_s + \mathbf{g} \tag{3.4}$$

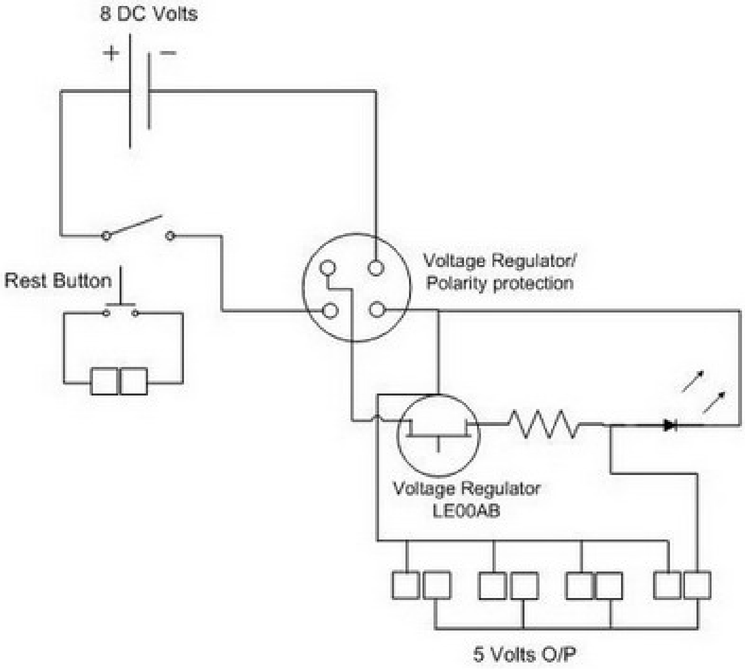


Figure 3.6: IMU Power board which consists of Input voltage (8-10) volts pin, regulated 5 volts outputs, polarity protection chip, 5 volts LE00AB voltage regulator, On/OFF and reset switches and LED indicator.

$$\begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} = \mathbf{f}_s + \mathbf{g} \tag{3.5}$$

Where a_x, a_y and a_z is the acceleration in x, y, z directions respectively, $\mathbf{f}_s$ is the specific translation force,and $\mathbf{g}$ is the earth gravity.

However in order to have an IMU of six-degree of freedom able to detect accelerations and angular rates in three dimensions, two accelerometer ADXL202EB each with dual axis sensitivity are mounted, then acceleration can be measured in x, y, z directions and three orthogonally mounted ADXRS150EB rate-gyros in order to measure angular rates ω_x, ω_y, ω_z around the accelerometer sensitive axes. However these sensors are not precisely mounted in orthogonal configuration due to imperfections during the construction process, fixing and soldering therefore there will be a small angle differences between inertial sensors sensitive axes and the IMU frame or platform as illustrated in figure 3.7.

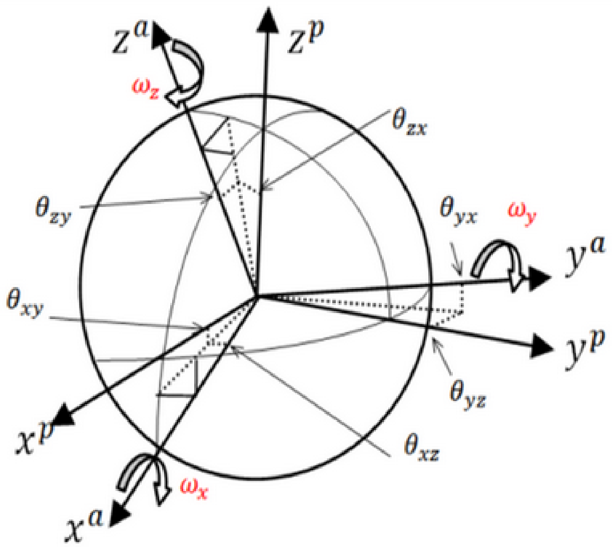


Figure 3.7: IMU platform coordinate axes, accelerometer and gyros sensitivity axes.

The stabilized platform (P) coordinates can be defined to be the orthogonalized accelerometer input axis coordinates or the orthogonolized accelerometer coordinates offset by a known set of three angles [10] which implies to define the transformation from the non-orthogonal accelerometer input axis cluster to the platform orthogonal coordinates with the same origin.

Because an ADXL202EB dual axis accelerometer is used to measure the acceleration in x, y directions therefore x, y sensitive axes could be considered as orthogonal; and by defining the platform coordinates system in which the platform coordinate axis x^p coincides with x^a accelerometer sensitivity axis. The x^p will be identical with x^a sensitivity axis, and the y^p axis will be in the $x^a y^p$ plane with this definition the angles θ_{xz} θ_{xy} θ_{yx} becomes zero, and because the dot product of two unit vectors is equal to the cosine of the angle between them and the platform frame is orthogonal, this can be shown as follows:

$$\begin{pmatrix} x^a \\ y^a \\ z^a \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ \hat{y^a} \cdot \hat{x^p} & \hat{y^a} \cdot \hat{y^p} & 0 \\ \hat{z^a} \cdot \hat{x^p} & \hat{z^a} \cdot \hat{y^p} & \hat{z^a} \cdot \hat{z^p} \end{pmatrix} \begin{pmatrix} x^p \\ y^p \\ z^p \end{pmatrix} \tag{3.6}$$

Where x^a, y^a and z^a denotes non-orthogonal accelerometers sensitivity axis in x, y, z directions respectively and x^p, y^p and z^p denotes the platform orthogonal coordinates in x, y, z directions respectively. If the accelerometer axes are very nearly orthogonal or accuracy requirements are not stringent then the specific force in accelerometer coordinate axes can be transformed to a specific force estimate in platform coordinate axes assuming that the non-orthogonal sensitivity axes of the accelerometers differs by three "small" angles from the platform coordinates axes therefore equation 3.6 can be given by as [10]:

$$\begin{pmatrix} x^a \\ y^a \\ z^a \end{pmatrix} \approx \begin{pmatrix} 1 & 0 & 0 \\ \theta_{yz} & 1 & 0 \\ -\theta_{xy} & \theta_{zx} & 1 \end{pmatrix} \begin{pmatrix} x^p \\ y^p \\ z^p \end{pmatrix} \tag{3.7}$$

Where θ_{ij} the rotation of i-th accelerometer sensitivity axis around j-th platform sensitivity axis. Thus the specific force in the accelerometer cluster coordinates can be transformed into specific force estimates in platform coordinates:

$$\mathbf{F}^p = \mathbf{T}_a^p \mathbf{F}^a \tag{3.8}$$

Where:

$$\mathbf{T}_a^p \approx \begin{pmatrix} 1 & 0 & 0 \\ \theta_{yz} & 1 & 0 \\ -\theta_{xy} & \theta_{zx} & 1 \end{pmatrix} \tag{3.9}$$

Where $\mathbf{M}_a^p$ the transformation matrix between the accelerometer triad input axis and the platform coordinate.

Because the angular rates are measured using three different rate-gyro sensors boards therefore six small rotations around the above defined platform are required to define the rotation from the gyro input axes to the platform axes; that is, three rotations are required to make the sensitivity axes of the three gyros orthogonal and once the gyros sensitivity axes are orthogonally aligned then the direction cosine matrix needed to align the orthogonal coordinate axes with the platform coordinate axes. Therefore the rotation in the rotation in the gyros coordinate axes can be transformed into rotation in the platform coordinates as [10]:

$$\omega^p = \mathbf{T}_G^p \omega^G \tag{3.10}$$

Where:

$$\mathbf{T}_G^p \approx \begin{pmatrix} 1 & -\varphi_{yz} & \varphi_{zy} \\ \varphi_{xz} & 1 & -\varphi_{zx} \\ -\varphi_{xy} & \varphi_{yx} & 1 \end{pmatrix} \tag{3.11}$$

And this may equivalently be written as:

$$\omega^p = \mathbf{R}_o^p \mathbf{T}_G^o \omega^G \tag{3.12}$$

Where:

$$\mathbf{T}_G^o \approx \begin{pmatrix} 1 & -\varphi_{yz} & \varphi_{zy} \\ 0 & 1 & -\varphi_{zx} \\ 0 & 0 & 1 \end{pmatrix} \tag{3.13}$$

Where φ_{ij} is the rotation of i-th gyros sensitivity axis around j-th platform sensitivity axis. The matrix $\mathbf{T}_G^o$ transforms the non-orthogonal gyro sensitivity axes into a set of orthogonal coordinate axes and $\mathbf{R}_o^p$ is the direction cosine matrix transforming the gyros rates in the orthogonal sensitivity axes coordinates into platform coordinates.

The measurement errors of inertial sensors are dependent on the physical operational principle of the sensor itself. And the error components:

- Internal misalignment of the sensor's input axis.
- The scale factor and the scale factor linearity can then be determined by comparing the output of the gyro/ accelerometer to the rotation rate/acceleration input.
- Accelerometer and rate gyro bias which is the non-zero output while sensor is in the stationary status that is there is no force acting onto the sensor.
- Linear acceleration sensitivity which affects the gyros stability output; i.e. for the used MEMS the linear acceleration sensitivity effect for any axis is in the order of $0.2/s/g$.

Other errors which also play a significant role in the stability of the output of low cost inertial sensors is the:

- Temperature drift and nonlinearities, i.e. the better the sensor, the smaller the bias variation over the temperature range; furthermore, the better the sensor the greater the linearity of this bias variation [17]. For the MEMS sensors used in our study the temperature drift is 0.5persent and 5mv for accelerometer and gyros respectively and the nonlinearity is in the order of 0.2persent and 0.1persent of FS of a best fit to a straight line for accelerometer and gyros respectively [14], [16]; therefore they may be neglected.
- Hysteresis effect is common on most inertial sensors. The drift rates and accelerometer biases tend to change each time the unit is switched on. This is due to the fact that measurements are noisy, and when random noise is filtered which typically most of systems used before measurements are processed and used in navigation equations that will produce what is called a random walk. The integration of this random walk will result in velocity and positions moving at different rates during different runs even the IMU (and vehicle) are in the same direction and experiencing the same acceleration during each run [3].

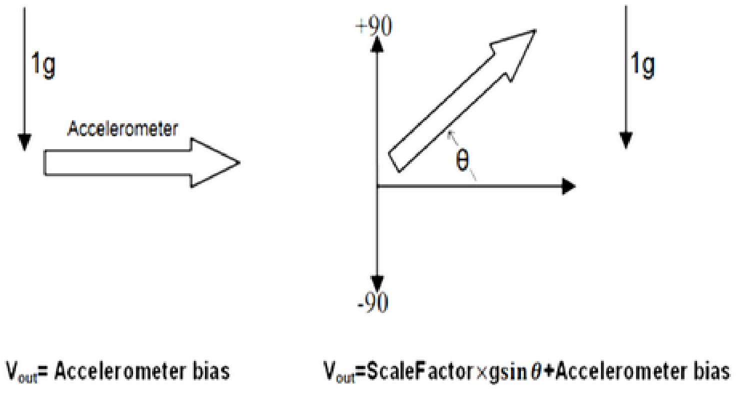


Figure 3.8: Accelerometer output voltage.

The measured output of a single accelerometer as shown in Figure 3.8 can be given as:

$$D_{out} = S_{factor} \times g\sin(\theta) + b_a \tag{3.14}$$

Where D_{out} is the accelerometer output duty cycle (refer to ADXL202EB data sheet), S_{factor} is the scale factor, g is the gravity acceleration, $g\sin(\theta)$ is the specific force along the accelerometer sensitive axis and b_a is the accelerometer nonzero bias output, this output can be measured even though there is no component of specific force acting along the input axis. Therefore the specific force of the accelerometer x, y or z-sensitive axis is given by:

$$g\sin(\theta) = \frac{D_{out} - b_a}{S_{factor}} \Longrightarrow \mathbf{F}^a = \frac{D_{out} - b_a}{S_{factor}} \tag{3.15}$$

Therefore the orthogonalized computed specific force in the platform coordinates $\mathbf{F}^p$ is equal to the actual measured force value in the accelerometer cluster $\mathbf{F}^a$, pre-multiplied by an adopted scale factor S_a which is in turn is pre-multiplied by the non-orthogonal transformation matrix $\mathbf{T}_a^p$ from the accelerometer to the platform which may be modelled as [10]:

$$\mathbf{F}^p = \mathbf{T}_a \mathbf{S}_a \mathbf{F}^a \tag{3.16}$$

In an expand form 3.16 becomes:

$$\begin{pmatrix} \mathbf{F}^p_x \\ \mathbf{F}^p_y \\ \mathbf{F}^p_z \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ \theta_{yz} & 1 & 0 \\ -\theta_{xy} & \theta_{zx} & 1 \end{pmatrix} \begin{pmatrix} S_{xa} & 0 & 0 \\ 0 & S_{ya} & 0 \\ 0 & 0 & S_{za} \end{pmatrix} \begin{pmatrix} \mathbf{F}^a_x \\ \mathbf{F}^a_y \\ \mathbf{F}^a_z \end{pmatrix} \tag{3.17}$$

Where F^p is the measured specific force vector of accelerometers, S_a the accelerometer scale factor matrix, b_a is the bias vector and v_a is a random noise on the sensor signal.

The expression for the F^p is derived by considering the actual specific force in the accelerometer coordinates to be equal to the actual output specific force premultiplied by the actual scale factor $\underline{S}_a$, the actual output of an accelerometer during calibration is not equal to the actual value of apparent gravity g^a_a at calibration site. When there is no component of specific force along the input axis, there is usually a small nonzero bias output b_a. Also, allowance must be made for scale factor nonlinearity S_{anl} because accelerometer calibration coefficients vary as a function of products of components of the specific force [10].

Therefore the actual specific force measurement of the accelerometer cluster becomes:

$$F^a = F^a_o + b_a + S_{anl} F^a_i F^a_i \tag{3.18}$$

Where $F^a_o = [F^a_{ox} F^a_{oy} F^a_{oz}]^T$ the actual raw output of accelerometers in accelerometer is input axis coordinates and: $b_a = [b_{ax} b_{ay} b_{az}]^T$, $\underline{S}_a = diag[\underline{S}_{ax} \underline{S}_{ay} \underline{S}_{az}]$, $F^a_i F^a_i = [F^2_{OX} F^2_{OY} F^2_{OZ}]^T$ and $S_{anl} = diag[S_{axnl} S_{aynl} S_{aznl}]$.

Solving 3.18 for F^a and combining the result with 3.16 yields to the following accelerometer observation equation:

$$F^p = T_a^p S_a \underline{S}_a^{-1}(F_o^a + b_a + S_{anl} F_i^a F_i^a) \tag{3.19}$$

The above model could be applied to the gyros; the measured output of the gyros may be modelled as:

$$\omega^p = T_G^p S_G (\underline{S}_G)^{-1}(\omega_o^G + b_G) \tag{3.20}$$

Where ω^p is the true rotational rate with respect to the inertial frame of reference, ω_o^G is the actual measured angular rate in gyro coordinates, T_G^p is the non-orthogonal transformation from gyro to platform coordinates, $S_G = diag[S_{Gx} S_{Gy} S_{Gz}]$ is the adopted scale factor matrix, and $\underline{S}_a = diag[\underline{S}_{Gx} \underline{S}_{Gy} \underline{S}_{Gz}]$ is the actual scale factor matrix and $b_G = [b_{Gx} b_{Gy} b_{Gz}]^T$ is the rates bias vector.

The actual measured angular rate for each gyro can be given as:

$$V_{out} = S_f \omega_{oi}^G + b_G \tag{3.21}$$

Using the opposite sense then we can get:

$$\omega_{oi}^G = \frac{(V_{out} - b_g)}{S_{fG}} \tag{3.22}$$

Where V_{out} is the gyro analog output voltage,ω_o^G is the angular rate acting along the gyros input sensitive axis; $i = x, y$ and z. S_{fG} is the gyro scale factor and b_a is the gyros nonzero bias output.

3.4 Calibration Procedure And Experiment Setup

Calibration of inertial instrument as discussed in the introduction of this chapter is necessary because the outputs are blend of accurate and erroneous. Modern calibration procedures utilize the benefits of Kalman filtering to obtain optimal estimates of the calibration coefficients. Obtaining analytical results for calibrations based on Kalman filtering is extremely difficult. However, analytical results can be derived for inverse and least-squares solutions of the calibration observation equations [10]. However calibration is usually performed in the field for platform-mounted inertial instruments and for body-mounted instruments at the factory. With the advent of GPS, the factory calibrations for body-mounted instruments can be updated to a certain degree during navigation [10].

Traditionally, a mechanical platform rotating the IMU into different precisely controlled orientations and turn rates have been used to calibrate IMU's. Then, observing the output and the pre-calculated specific force or angular velocity input acting upon the IMU for different orientations and rotation sequences, respectively, it is straightforward to estimate the misalignment, scaling and bias [10], [18] and [19]. The cost of such a platform often exceeds the cost of developing and constructing an IMU from a MEMS-based sensor. Therefore a calibration procedure is desirable where the requirements of a precisely controlled orientation of the IMU can be relaxed and cost will decreased dramatically [11].

The developed calibration procedure starts by comparing the output readings of rate-gyro (analogue Outputs) or oaccelerometer (digital output) when the set is excited with a known input motion i.e. the output signals from a gyroscope can be compared with the accurately known rotation rate and similarly, the output signals from a gyroscope can be compared with accurate standard gravity vector. There are several different levels at which error compensation can be applied. However, the fundamental idea is the same for all; to correct the effects of a predictable systematic error, or errors, on the accuracy of a sensor. Additionally, a basic requirement is that an error process can be represented by an equation and hence modelled mathematically, and that signal corresponding to the disturbing effect, such as temperature or acceleration, is available and can be measured to the required accuracy [3].

Predictable error components can be used in the opposite sense to correct, or compensate. Often this technique relies on the use of a constant coefficient in the error representation, but for more demanding applications, or complex error behavior, it is common to use polynomial representations.

The accuracy that may be achieved from the application of compensation techniques is dependent on how precisely the coefficients in the error equation represent the actual sensor errors. This representation can often vary as a function of time, temperature, the environment in which the sensor is used and how often it is used. For the more demanding applications, it may requires to recalibrate the IMU regularly, to ensure that the compensation routines are as effective as required by the particular application.

The method of calibration described here is for platform-mounted instruments and is based on performing a series of instrument cluster rotations from one orientation to the next; the constructed IMU is fixed on platform -Excel Dividing Heads Semi Universal table [20], which is manually rotated; the platform accuracy is 0.5°. Then the residuals are observed for a period of time at each orientation, after completing the series of observations, the data are used to compute the calibrations coefficients. The sensors outputs are sampled by 16-bit acquisition card at sampling time equals 0.0002 sec using real-time *dSpace*® Data acquisition unit. Control Desk is the software that interfaces with the dSpace hardware. It allows the user to interact with the hardware, design and build control or behaviors model, change input values and different gains or parameters and also measure the states of various components and log output data for further analysis in formats Thu can be easily handled by Matlab. *(For further technical information regarding dSpace product, see appendix A).*

Data sets are loged and processed using $Matlab^{®}$ then the sampled voltage signals are converted to angular rate and specific force using a *Simulink*® model - the *Simulink*® model is an implementation of equation 3.15 and equation 5.32. The calibration process is based on performing a series of manual rotations from $-90°$ to $+90°$ with a step of 5° along each accelerometer sensor sensitivity axis and a series of angular rotations starting from $-150°/sec$ to $150°/sec$ with a step of $30°/sec$ along each gyro sensor sensitivity axis. Data sets (1000 samples/each observation using real-time *dSpace*® are observed from 38 different static orientations and 11 different rotations rate for each accelerometer and gyro sensors, respectively.

The "Curve Fitting Toolbox" in *Matlab*® is used to estimate each accelerometer and gyro coefficients - Results shown in Figures 6, 7 and 8 and summarized in Table 1. Generally the calibration rotation is designed to provide measurement residuals that, together as a whole, reflect the effect of all of accelerometer and gyro calibration coefficients. During the accelerometer calibration, the specific force output is compared to the computed apparent gravity obtained from a gravity survey through the calibration rotation schedule; any difference is used to calculate accelerometer calibration coefficient values that force the computed specific force to agree with the extrapolated gravity survey value, the comparison is made by means of observation equations which derived earlier equation 3.19, then using least square method to determine the accelerometer calibration coefficients. Note that there are 12 unknowns in the observation equation. Therefore an 18-static observation schedule is designed; each accelerometer input axis is placed in an up and down orientation and at all possible halfway in between orientations as [10]. The measurement residual for each observation is $(g_a^p - S_o^a)$; it is the best to use the measurement residual composed of the magnitude of the apparent gravity vector at the calibration side minus the magnitude of the accelerometer triad; using the magnitude difference obviates the necessity to precisely control the platform orientation during the calibration rotations [10]. The data sets collected from the 18-different static orientations at a constant temperature $(25°c)$ and a general least squares algorithm is applied to the data to estimate accelerometer observation coefficients.

The least squares function for the accelerometer calibration as defined by [10] is:

$$X_a \beta_a = Y_a \tag{3.23}$$

Where $\beta_a = [\theta_{zx}\theta_{xy}\theta_{yz}b_{ax}b_{ay}b_{az}\delta\widehat{\underline{s}}_{ax}\delta\widehat{\underline{s}}_{ay}\delta\widehat{\underline{s}}_{az}S_{axnl}S_{aynl}S_{aznl}]^T$ is the estimated coefficient vector and $\delta\widehat{\underline{s}}_{ai} = \delta\underline{s}_{ai}/S_a i$ is the normalized scale factor perturbation, in which the scale factor perturbation is the difference between the actual and nominal values:

$\underline{S}_{ai} = S_{ai} + \delta\underline{S}_{ax}$, $i = x$, y and z. And Y_a to be the vector of measurement residuals, X_a is the first-order Taylor series expansion coefficient matrix of accelerometer observation equation 3.19. Then the calibration coefficients are evaluated [18]:

$$\beta_a = (X_a^T X_a)^{-1} X_a^T Y_a \tag{3.24}$$

In the same manner the sensed rotation rate of gyro cluster is compared to a reference rate, which is either the Earth rate or the sum of the rotation of the Earth and an accurately commanded rate; however using the Earth rate has two disadvantages. The Earth rotation rate is only $15°/h$, and the horizontal component is small at high latitudes [10].

The orthogonalized sensed angular rate is equal to the actual measure rate, pre-multiplied by the scale factor matrix and the orthogonalized transformation from gyro platform axes. The platform axes have been defined to be the orthogonalized accelerometer axes as shown in the observation equation 3.20. During the gyros calibration, the sensed rotation rate output is compared to the computed commanded rate obtained from a gravity survey through the calibration rotation schedule; any difference is used to calculate gyros calibration coefficient values that force the computed commanded rate to agree with the extrapolated rate survey value, the comparison is made by means of observation equations which derived earlier equation 3.20 and then using least square method to determine the gyros calibration coefficients. Note that there are 12 unknowns in the observation equation. An 18 observation schedule is designed. The measurement residual for each observation is $(\omega_r^p - \varphi_o^G)$. It is the best to use the measurement residual composed of the magnitude of the reference rate vector at the calibration side minus the magnitude of the gyro triad.

The least squares function for the gyros calibration as defined by [10] is:

$$X_G \beta_G = Y_G \tag{3.25}$$

Where $\beta_G = [\varphi_{xz}\varphi_{yz}\varphi_{zy}\varphi_{xy}\varphi_{yx}\varphi_{zx}b_{Gx}b_{Gy}b_{Gz}\delta\widehat{\underline{s}}_{Gx}\delta\widehat{\underline{s}}_{Gy}\delta\widehat{\underline{s}}_{Gz}]$ is the estimated coefficient vector and $\delta\widehat{\underline{s}}_{Gi} = \delta\delta\underline{S}_{Gi}/S_{Gi}$ is the normalized scale factor perturbation, in which the scale factor perturbation is the difference between the actual and nominal values:

$\underline{S}_{Gi} = S_{Gi} + \delta\underline{S}_{Gi}$, $i = x$, y and z. and Y_G to be the vector of measurement residuals, X_G is the first-order Taylor series expansion coefficient matrix of gyro observation equation 3.20. Then the calibration coefficients are evaluated [21]:

$$\beta_G = (X_G^T X_G)^{-1} X_G^T Y_G \tag{3.26}$$

A platform rotation schedule should be designed to provide measurement residuals that, together as a whole reflect the effect of all of the accelerometer and gyro calibration coefficients as shown in Figure 3.15.

3.5 Conclusions and Results

- The accuracy that may be achieved from the application of compensation techniques is dependent on precisely how the coefficients in the error equation represent the actual sensor errors.

- This representation can often vary as a function of time, the environment in which the sensor is used and how often it is used. For the more demanding applications, it may requires to recalibrate the sensor regularly, to ensure that the compensation routines are as effective as required by the particular application.

- The proposed method can be considered useful for many low-cost applications where the cost of constructing a mechanical platform many times exceeds the cost of developing the inertial measurement unit.

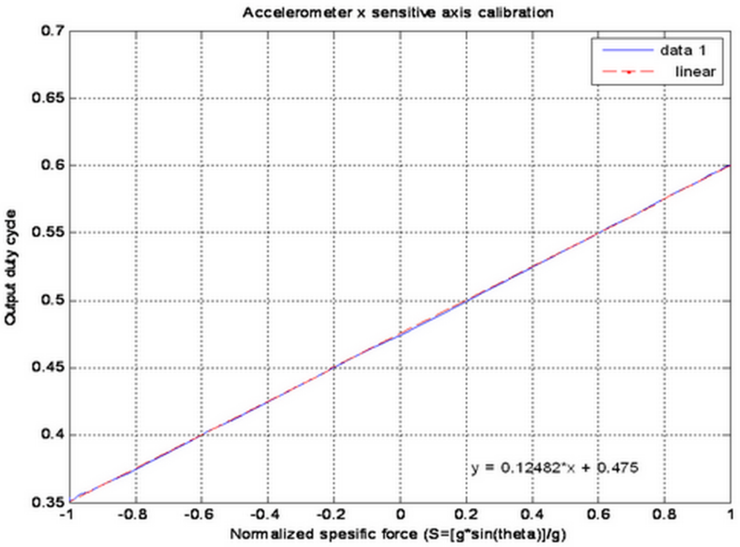


Figure 3.9: Output Duty Cycle versus normalized specific force acting along accelerometer sensitivity x-axis.

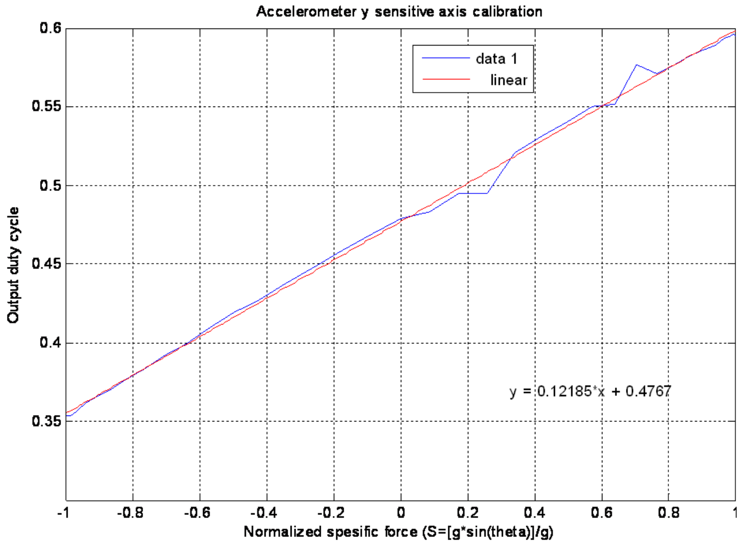


Figure 3.10: Output Duty Cycle versus normalized specific force acting along accelerometer sensitivity y-axis.

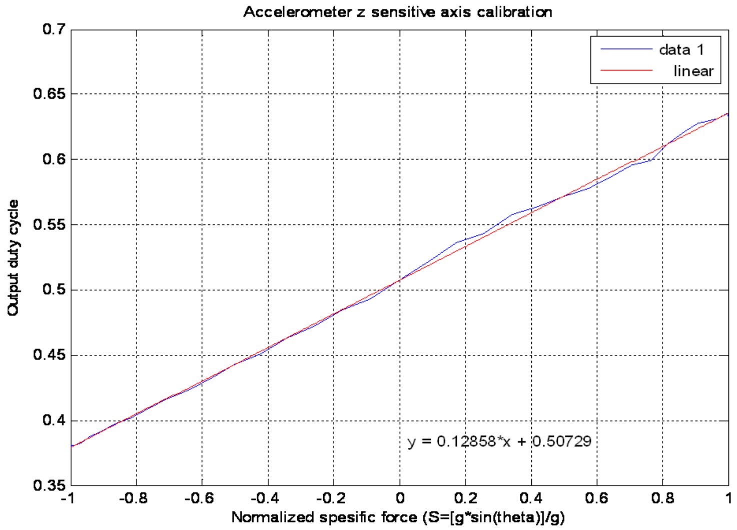


Figure 3.11: Output Duty Cycle versus normalized specific force acting along accelerometer sensitivity z-axis

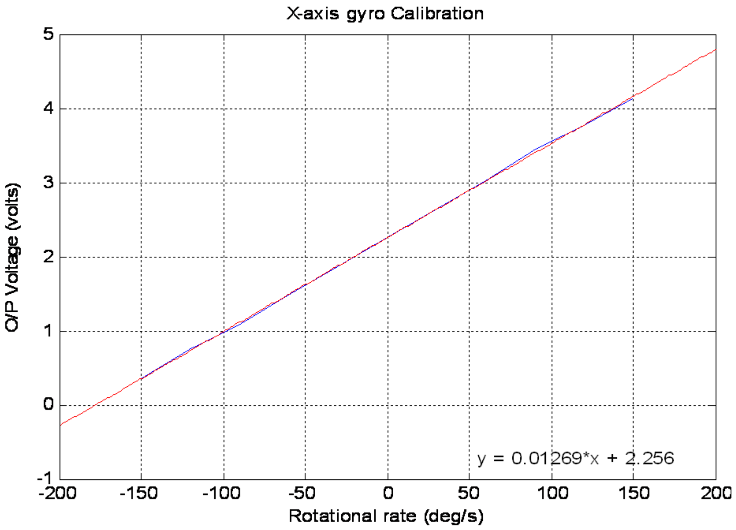


Figure 3.12: Output voltage versus commanded rotational rate acting along gyros sensitivity x-axis

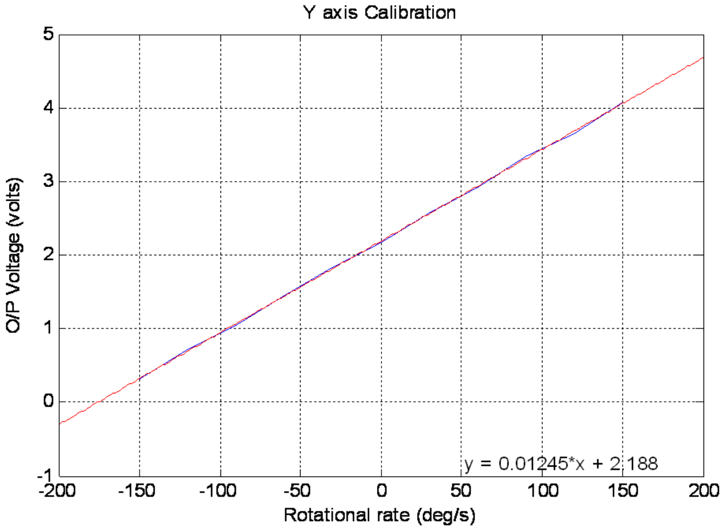


Figure 3.13: Output voltage versus commanded rotational rate acting along gyros sensitivity y-axis

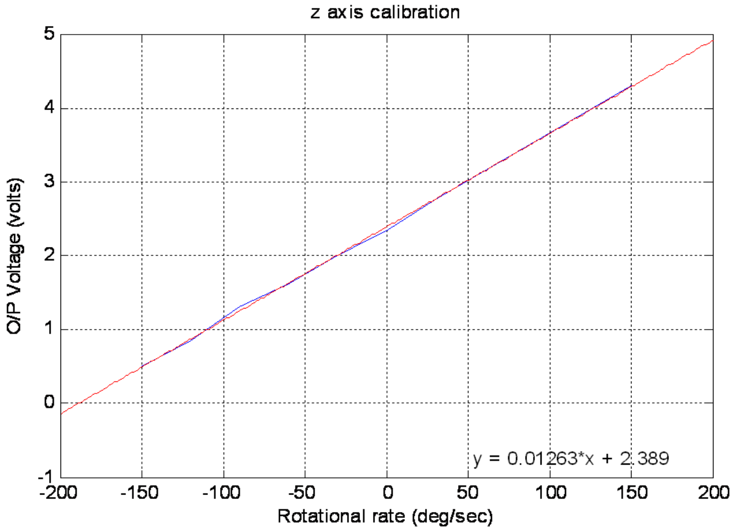


Figure 3.14: Output voltage versus commanded rotational rate acting along gyros sensitivity z-axis

Table 3.1: Accelerometers estimated scale factor and biases ($V_{out} = S_f \times g\sin(\theta) + b_a$). The platform was manually positioned into 38 different orientations, at each orientation the sensor's outputs were sampled by 16-bit acquisition card at sampling time of 0.0001 sec for a period of 1 sec using real-time *dSpace*®

Axis	**Scaling** (S_f)	**Bias** (b_a)	**Dutycycle**(percent)
x	0.12482	0.47500	47.500
y	0.12185	0.47670	7.670
Z	0.12858	0.50729	50.729

Table 3.2: IMU results,Accelerometrs triad observation equation coefficients using least squares estimate.

Axis	**Scaling**	**Bias** (m/s^2)	**Axis**	**Misalignment**	**Nonlinearity scaling**
x	-0.0032	-0.0350	θ_{zx}	-0.0199	-0.0037
y	0.0003	0.3976	θ_{xy}	0.0941	-0.0165
Z	-0.0789	0.7783	θ_{yz}	-0.0201	-0.1545

Table 3.3: Gyros estimated scale factor and biases ($V_{out} = S_f \times \omega + b_g$). Data was collected from 11 rotations, at each orientation the sensor's output voltages were sampled by 16-bit acquisition card at sampling time of 0.0001 sec for a period of 1 sec using real-time *dSpace*®.

Axis	**Scaling** (S_f)	**Duty Cycle**(b_g)
x	0.01269	2,256
y	0.01245	2.188
Z	0.12858	2.389

Table 3.4: IMU results, gyros triad observation equation coefficients using least squares estimate.

Axis	**Scaling**	**Bias**($\circ/s$)		**Misalignment**		**Misalignment**
x	0.0107	0.3658	φ_{xz}	-0.0294	φ_{xy}	0.0174
y	0.0078	1.3338	φ_{yz}	0.0369	φ_{yx}	0.0398
Z	0.0300	3.4827	φ_{zy}	0.0252	φ_{zx}	0.0117

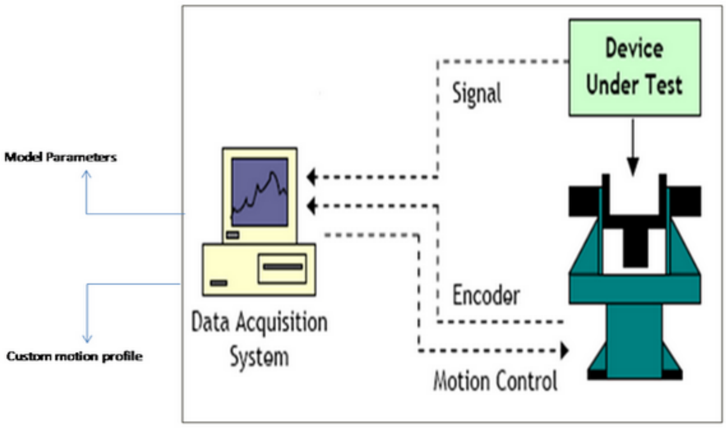


Figure 3.15: Experiment Methodology setup which is used to calibrate the IMU unit.

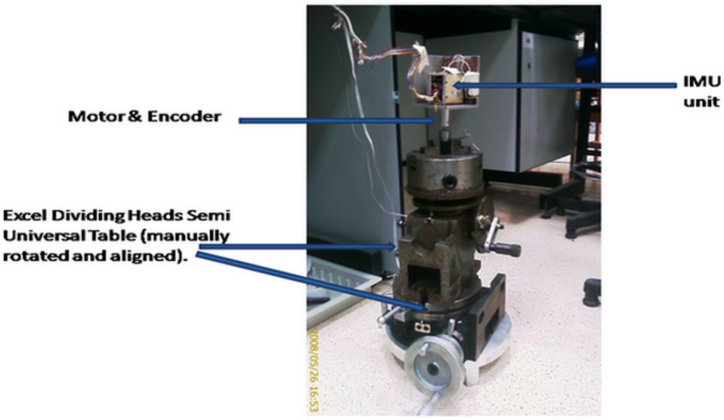

IMU unit
Motor & Encoder
Excel Dividing Heads Semi Universal Table (manually rotated and aligned).

Figure 3.16: Rate Table, developed in AUS-Mechatronics Lab for the purpose of IMU calibration.

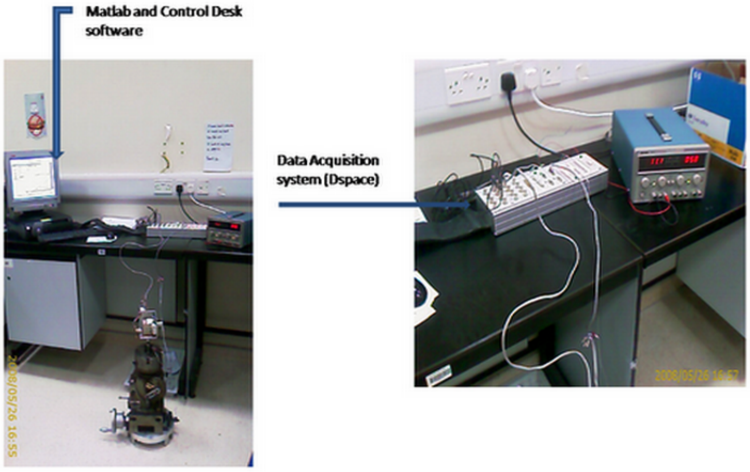

Matlab and Control Desk software
Data Acquisition system (Dspace)
2008/05/26 16:55
2008/05/26 16:57

Figure 3.17: Experiment Setup: IMU, IMU holder, motor with encoder, acquisition device (dSpace), software and PC.

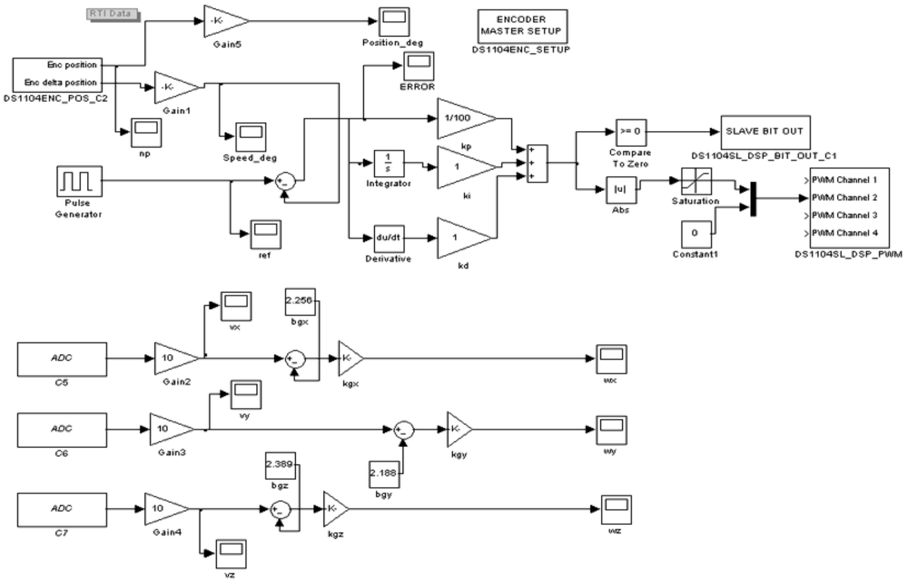


Figure 3.18: Experiment Gyros Simulink Models

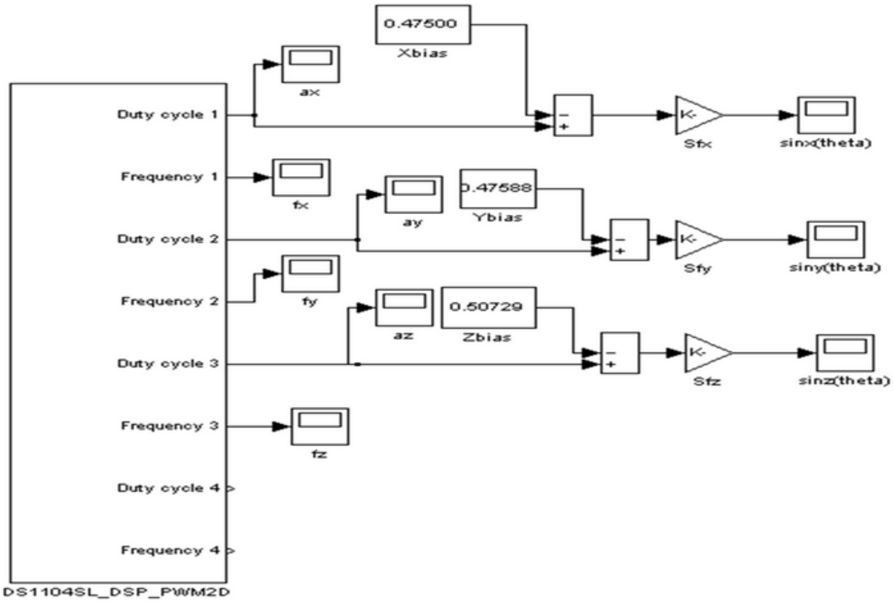


Figure 3.19: Experiment Accelerometers Simulink Model

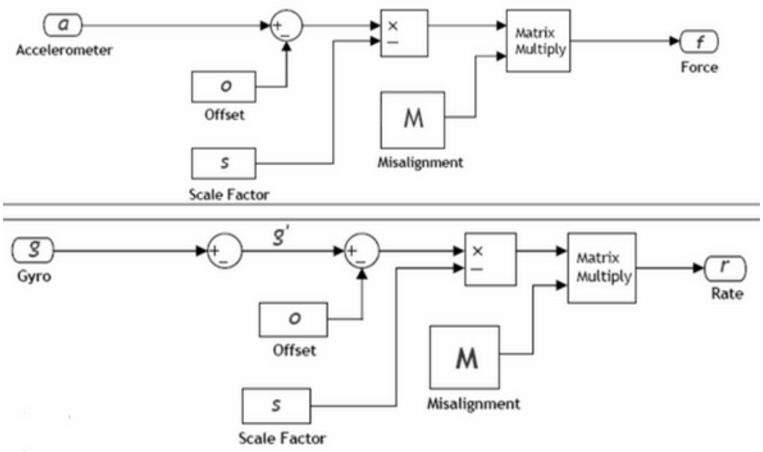


Figure 3.20: Accelerometer and Rate Gyro models used in the opposite sense.

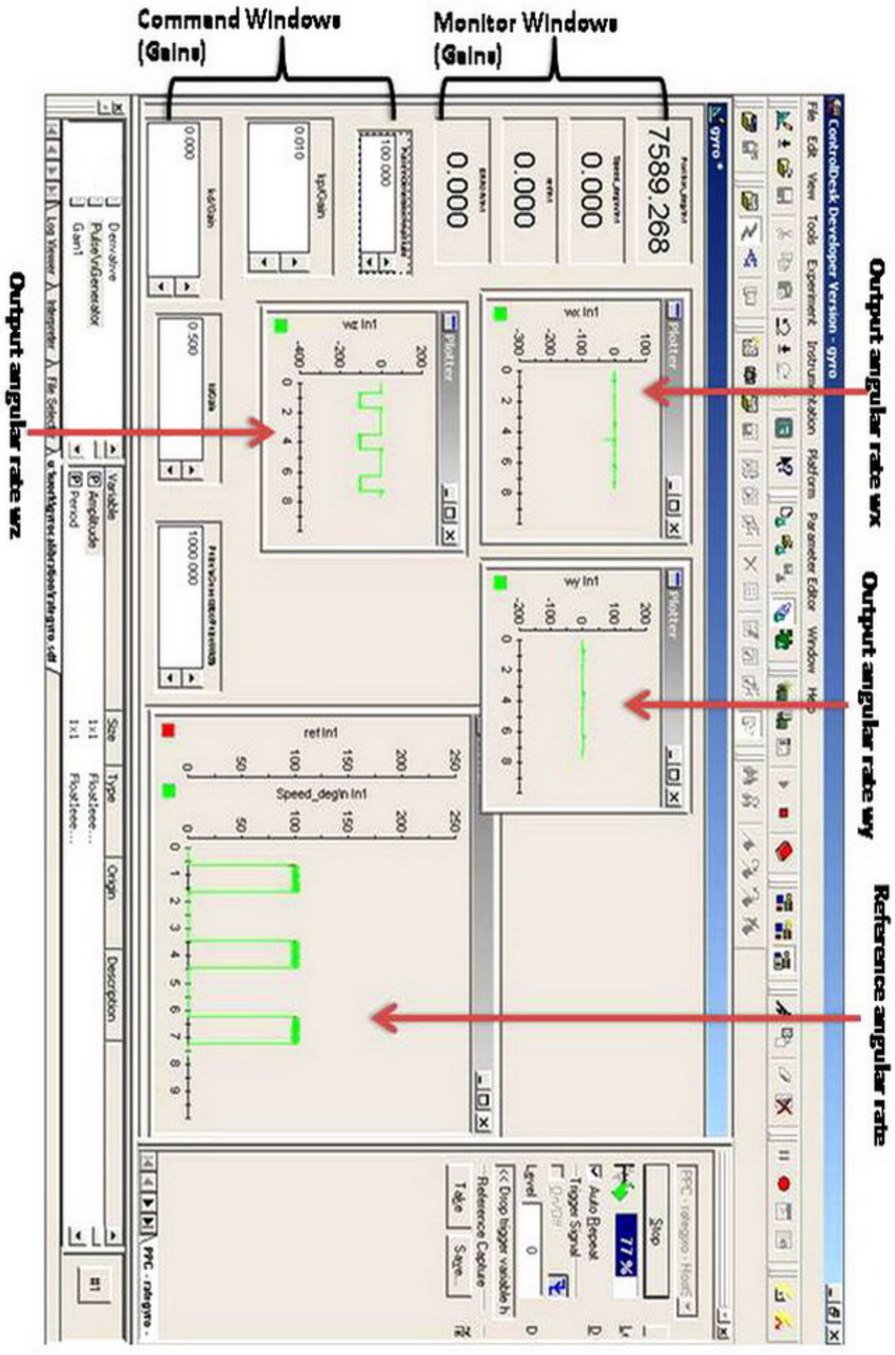


Figure 3.21: Experiment Acquisition Model.

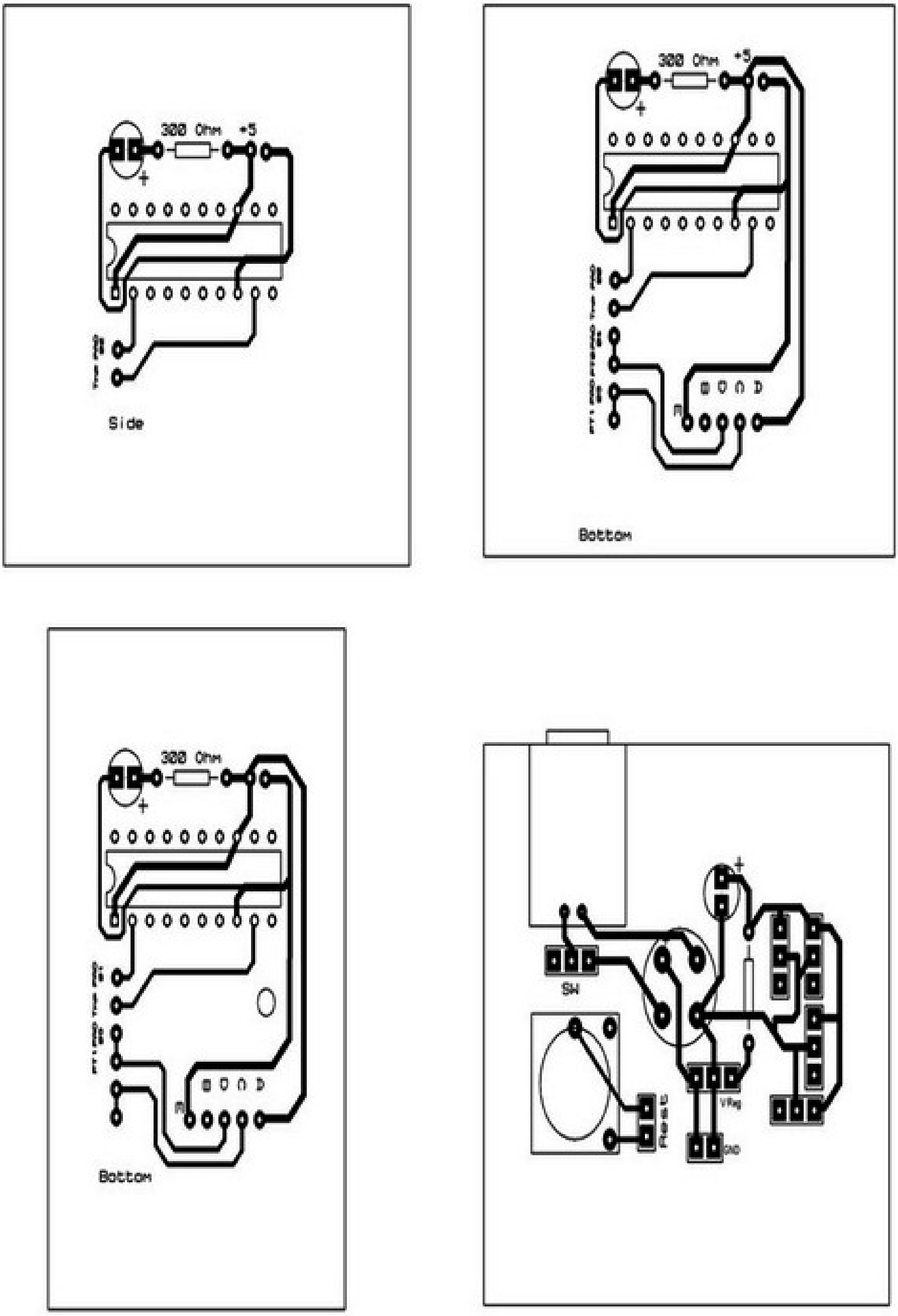


Figure 3.22: Printed Circuit Boards (PCB) layout of IMU. There are four layouts, one for the power and the other accelerometers and rate-gyros sensors are distributed on the remained layouts. PCB for IMU is designed using Proteus 6 Professional software.

Chapter 4

GPS aided Inertial Navigation System (INS)

4.1 Introduction

The integration of GPS and the inertial systems become a subject of research in this field. High accuracy and low cost is two basic but conflicting requirements to be considered for any vehicle requiring a navigation capability. The inertial navigation system is characterized by a time-dependent drift in accuracy of the position estimates it provides and the rate at navigation errors grow over long periods of time is governed predominately by the accuracy of the initial alignment, imperfections in the inertial sensors that the system uses and the dynamic of the trajectory followed by the host vehicle. Whilst high accuracy can be achieved through the use of high accurate inertial sensors but still there are limits to the performance that can reasonably be achieved before the cost of the inertial systems becomes s large, however expensive inertial system using high accurate expensive inertial sensor are incompatible with many applications except with strategic systems where there is no easy alternatives such as submarine and missiles. However high accuracy and low cost inertial navigation system can be satisfied by using integrated navigation systems, in which strapdown inertial navigation systems are used in conjunction with other navigation aids.

The variety modern navigation aids now available is extensive and coupled with advances in estimation processing techniques and high-speed processors, have resulted in greater application of integrated navigation systems in recent years [3].

Generally in an aided inertial system, one or more of the inertial navigation system output signals are compared with independent measurements of identical quantities derived from an external source,corrections to the inertial navigation system are then derived as function of these measurement differences. By judicious combination of this information, it is possible to achieve more accurate navigation than would be achieved using stand alone inertial navigation.

4.2 GPS Fundamentals

There are currently two Global Navigation Satellite Systems (GNSS), namely the widely used Global Positioning System (GPS) from the U.S. Department of Defense (DOD), and the Russian Global Orbit Navigation Satellite System (GLONASS). A third system named Galileo3, which is launched by the European Union and the European Space Agency, will be in commercial operation phase in year 2008. Because the GPS is well established and a broad range of receivers (also some relatively inexpensive receivers) are available, this system was chosen for this project.

The NAVSTAR (Navigation and Satellite Timing and Ranging)which the official abbreviation for the widely know GPS, is part of satellite -based navigation system developed by the U.S. Department of Defense. GPS belongs to a large class of radio navigation systems that allow the user to determine his range and/or direction from a known signal transmitting station by measuring the differential time of travel of the signal.

The GPS comprises a set of orbiting satellites at known locations in space and their signals can be observed on the Earth. Three distances to distinct satellites having known positions provide sufficient information to solve the observers three-dimensional position. The system is designed so that a minimum of four satellites is always in view anywhere in the world to provide continual positioning capability. This is accomplished with 24 satellites distributed unevenly in six symmetrically arranged orbital planes.

The applications of GPS range from military navigation, vehicle monitoring, to sporting activities. For geodetic applications, the precise measurement of baselines (relative positioning) in static mode of GPS is widely used. Static positioning involves placing the receiver at a fixed location on the Earth and determining the position of that point. Opposite to this, kinematic positioning refers to determining the position of a vehicle or a platform that is moving continually with respect to Earth. It is also known as real-time positioning. The term navigation is used for real-time processing of the positioning data. Differential GPS (DGPS) is a technique for reducing the error in GPS-derived positions by using additional data from a reference GPS receiver at a known position. The most common form of DGPS involves determining the combined effects of navigation message ephemeris and satellite clock errors at a reference station and transmitting the pseudo-range corrections in real time, to a users receiver [2].

The GPS is not without problems and limitations. It is not a self-contained, autonomous system. The user must be able to see the GPS satellites. Satellite visibility may be obstructed locally by intervening buildings, mountains, bridges, tunnels, etc. For kinematic applications, the effects of electronic interference or brief obstructions may cause the receiver to miss one or more cycles of the carrier wave. The frequency of the data output in most receivers is often 1 Hz. Most of the error in GPS positioning come from medium propagation effects that are unpredictable to model such as atmospheric effects.

4.2.1 Clocks and Time

Each GPS satellite carries an atomic clock to provide timing information for the signals transmitted by the satellites. The clocks are oscillating at a particular frequency. The relationship between the phase ϕ, frequency *f*, and the time is:

$$f(t) = \frac{d\phi}{dt} \tag{4.1}$$

where t represents true time. The phase of an oscillating signal is the angle it sweeps out over time ($0 \leq \phi \leq 2\pi$) and has the units of cycle. The frequency of the signal is the rate at which the phase changes in time and has the units of Hertz (cycles per second).

$$\phi(t) = \phi(t_0) + \int_{t_0}^{t} f(\grave{t}d\grave{t} \tag{4.2}$$

Where t_0 is some initial time. τ denote the indicated time related to the phase:

$$\tau(t) = \frac{\phi(t) - \phi_0}{f_0} \tag{4.3}$$

Where f_0 is some nominal (constant) frequency since the initial indicated time does not coincide with initial phase $(phi(t) \neq \phi_0)$.

The oscillator clock time (τ) and the true time (t) differ from each other both in scale and in origin. The true time reflects the atomic clock time in U.S., which also differs from Coordinated Universal Time (UTC) by 2000 with 13 seconds. However, the GPS true time is calibrated by U.S. atomic time. The true time reflects the fact that the times indicated on satellite and receiver clocks are not perfectly uniform and must be calibrated by master clocks on the Earth. The relationship between the phase-time,(τ), and the true time, t, is:

$$\tau(t) = t - \Delta\tau(t) \tag{4.4}$$

$$\tau(t) = t - t_0 + \tau(t_0) + \delta_\tau(t), \tag{4.5}$$

Where,

$$\delta(t) = \frac{1}{f_0} \int_{t_0}^{t} \delta f(\grave{t}d\grave{t} \tag{4.6}$$

4.2.2 GPS Signals

The signal is a carrier wave (sinusoidal wave) modulated in phase by binary codes that represent interpretable data. It can be represented mathematically by:

$$S(t) = AC(t)D(t)\cos(2\pi ft) \tag{4.7}$$

where is the frequency of the carrier wave, and A is the amplitude of the signal. The code sequence C(t) is a step function having values (1, -1), also known as chips or bits. D(t) represents a data message.

Each satellite actually transmits two different codes, the C/A (coarse acquisition) code and the P-code (precision code). The P-code has 10 times higher chipping rate and wavelength in comparison with C/A code. They are transmitted on two separate microwave regions, an L1 signal with carrier frequency $f_1 = 1545.72$ MHz and with wavelength $\lambda 1 = 0.1903m$; and an L2 signal with carrier frequency $f_2 = 1227.6$ MHz and with wavelength $\lambda 2 = 0.2442m$. The transmission on two frequencies allows approximate computation of the delay of the signal due to ionospheric refraction. The total signal transmitted by the satellite is given by the sum of three sinusoids, two for the two codes on the L_1 carrier and one for the P-code on the L_2 carrier. The total signal transmitted by a GPS satellite is given by:

$$S^p(t) = A_P P^p(t) W^p(t) D^p(t) \cos(2\pi f_1 t) + A_C C^p D^p(t) \sin(2\pi \tag{4.8}$$

$$f_1 t) + B_P P^p(t) W^p(t) D^p(t) \sin(2\pi f_2 t) \tag{4.9}$$

Where A_P, A_C, and B_P represent amplitudes of the corresponding codes, C and P represent C/A and P codes, D represents the data message, superscript P identifies a particular satellite, and W represents a special code which is used to decrypt a military code. The codes serve two operational purposes: determining the range between the satellite and receiver and to spread the signal over a large frequency bandwidth, thus permitting small antennas on the Earth to gather the transmitted signal. Both codes consist of unique sequences of binary states that are generated using a pseudo-random noise (PRN) algorithmic process. The PRN for C/A code is different for each satellite and repeats every millisecond. For P-code, it is much longer and repeats only after 38 weeks. Each satellite uses only one distinct weeks worth of the code. The satellites are distinguished by the codes rather than by frequency.

4.2.3 GPS Receiver

Before the signal is processed by the receiver, it is pre-amplified and filtered at the antenna, and subsequently down-shifted in frequency to a more manageable level for processing . The mixed signal is given by:

$$S_r{}^p(t)S^p(t) = A\cos(2\pi_{LO}t)\cos(2\pi_s t + \phi(t)) \tag{4.10}$$

$$= \frac{A}{2}\cos(2\pi(f_s - {}_{LO})t + \phi(t)) + \frac{A}{2}\cos(2\pi(f_s + {}_{LO})t + \phi(t)) \tag{4.11}$$

where $S_r(t)$ is the pure signal sinusoid generated by the receiver oscillator, ${}_{LO}$ is the local oscillator frequency, $S_P(t)$ is the satellite signal with frequency ${}_S$, and A is an amplitude factor. The satellite signal is then shifted to an intermediate frequency (IF), and appropriate filters are applied to control the amplitude of the signal for subsequent processing. The signal then passes to the main signal processing part of the receiver. To calculate the distance between the satellite and the receiver, the time tag of the signal at the time of transmission and the time of reception at the receiver is compared, and using the speed of the light, the delay is converted to the distance. It is not the true range if the satellite and the receiver clocks differ; and therefore, the calculated range is called pseudo-range.

4.2.4 GPS Observable and Errors

A broad overview of GPS errors is provided in table 4.1 The largest error is due to the receiver clock. The next significant error source is the medium in which the signal must travel. This includes the Ionosphere, which has an altitude between 50 km to 1000 km and has many free electrons; and the Troposphere, which is a non-dispersive medium and contains mostly electrically neutral particles.

Table 4.1: Error sources in GPS positioning (Jekeli,2000)

Error Source	**Typical Magnitude**
Receiver clock error (synchronized).	1 μ s (300 m).
Residual satellite clock error	20 ns (6 m).
Satellite synchronization to UTC	100 ns (30 m).
Selective Availability (cancelled by 2001)	100 m.
Orbit error (precise, IGS)	20 cm.
Tropospheric delay	$\leq$ 30 m.
Ionospheric delay	$\leq$ 150 m.
Multi-path	$\leq$5 m (P-code); $\leq$5 cm (phase)
Receiver Noise	1 m (C/A code); 0.1 m (P-code); 0.2 mm (L_1 phase)

4.2.5 Multi-path and Noise

Other errors in GPS observables include the multi-path error (the reflection of the GPS signal from nearby objects prior to entering the antenna), equipment delays and biases, antenna eccentricities (phase center variations), and the thermal noise of the receiver. In wireless radio links, one has to deal with multi-path reception. The best case is to have a direct line of sight between the transmitter and the receiver. Nonetheless, there are reflected signals that superimpose the direct signal; as shown in Figure 4.1. These multi-path reflections affect both the pseudo-range and phase measurements in a GPS receiver.At last, there is also noise generated in the receiver itself. The receiver noise is depending on the antenna gain, amplifiers, receiver dynamics, and the code correlation methods. It can mostly be reduced by a careful and thorough hardware design.

The pseudo-range is formulated as:

$$S_r^P(\tau_r) = \rho_r^P(\tau_r) + c(\Delta\tau_r(t) - \tau^P(t - \Delta\tau_r^P)) + \Delta\rho_{iono,r}^p + \epsilon_{p,r}^P \qquad (4.12)$$

Where $\rho_r^P(\tau_r)$ is the true range between the satellite and the receiver, $\Delta\tau_r^P)$ is the time of transit, $\Delta\rho_{iono,r}^p$ is the ionospheric error for each satellite, c is the speed

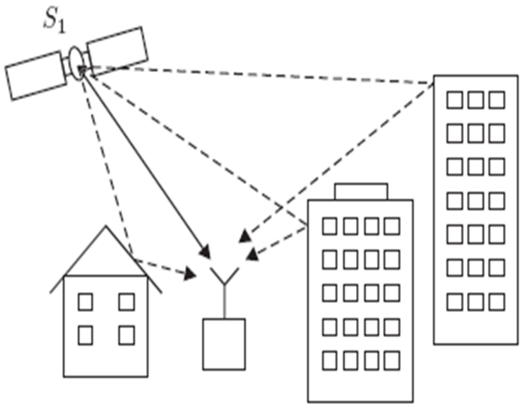


Figure 4.1: Multi-path propagation. *the solid arrow indicates the direct path, the dashed arrows indicate reflected path.*

of the light, and $\epsilon^P_{p,r}$ represents pseudo-range observation error (different for each satellite). For simplicity, the Tropospheric delay, the equipment and antenna offsets, the multi-path error, and the time registration error due to the receiver clock error are excluded. The phase observable can be expressed as follows:

$$\phi^P_r(\tau_r) = \frac{f_0}{c}\rho^P_r(\tau_r) + f_0(\Delta\tau_r(t) - \tau^P(t - \Delta\tau^P_r)) + \phi_{0,r} - \phi^P_0 N^P_r + \Delta\phi^P_{iono,r} + \epsilon^P_{\phi,r} \quad (4.13)$$

Where $\phi_{0,r}$ and ϕ^P_0 are the arbitrary phase offsets N^P_r is the integer representing the unknown number of full cycles and also called as carrier phase ambiguity, $\epsilon^P_{\phi,r}$ is the phase observation error.

4.3 GPS/INS Integration

The main idea of integrated navigation system is to take advantage of the complementary attributes of two or more navigation systems to get a system that provides greater precision than either of the system components operating stand alone. INS and GPS however have both advantages as well as disadvantages i.e. an INS exhibits relatively low noise, but drift over time and that because the position solution is based on the integration of accelerometers and gyros triad sensors set measurements, errors introduced in the measurements such as biases and noisy sensor data which will accumulate and result in an unbounded position error, on the other hand the GPS gives position with bonded error, which depends on the the quality of the GPS receiver, the disadvantage of GPS alone that it relies on a weak satellite signals, causing poor poor GPS-performance or outage in where the signals is blocked for instance in urban environments, in tunnels etc or most likely the great risk of jamming the GPS-receiver [22].

More specifically GPS provide pseudo-range pseudo-rate, while INS system measure specific force acceleration, which must be compensated for gravity and resolved into known coordinate reference frame before being integrated twice to yield estimate of position The accuracy of position measurements provided by GPS is limited as a result of low signal strength, the length of the pseudo-random code and errors in the code tracking loop. Further errors arise as a result of multi-path, variations in the satellite geometry , changes in the propagation conditions and user clock instability in addition the GPS velocity measurements are also noisy etc. Table 4.2 shows the main features between INS and GPS navigation systems.

We can conclude that Inertial and GPS measurements are complementary for the below two reasons [10]:

- There error characteristics are radically different.
- They measure different quantities.

Operating the two systems together yields benefits over operating ether system alone, it is possible an integrated system that yields low noise and low drift estimates of vehicle position, also the wide availability of GPS receivers, has provided much of the incites of the continuous development of techniques for the integration of GPS

Table 4.2: Features Comparison between INS and GPS Navigation

	Advantages	**Disadvantages**
INS (Inertial Navigation System)	1. High measurement output rate. 2. Provide both translation and rotational data. 3. Autonomous-Jamming incentives. 4. No signal outages.	1. Unbounded errors. 2. Knowledge of gravity model is required. 3.Accuracy decreasing with time
GPS (Global Positioning System)	1. Errors are bounded (uniform accuracy,independent of time). 2. Not sensitive to gravity	1. Low data rate. 2. No attitude information. 3. Sensitive to jamming.

and INS. Given the availability of uninterrupted GPS access, there is considerable scope to combine low accuracy inertial system with GPS in order to produce low cost precision navigation system capable of operating under wide range of conditions. The grade of the inertial sensors required for such system is determined to a large extent by the duration of the GPS interruption expected.

Figure 4.2 below illustrate the advantage of integrating GPS and INS as well as shows how GPS, INS and an integrated GPS/INS navigation system has complementary properties which makes an integration between them suitable.

As shown in the above figure INS/GPS system is robust and have high accuracy compare to stand alone GPS or INS. During the time before the satellite outage the system estimates sensors errors to be able to compensate for them, it is obvious that the position error grows during the satellite outage but slower than for the stand alone INS, however when the GPS signal is available again the INS/GPS recovers quickly after the outage.

Kalman Filter is an extremely effective and versatile procedure for combining noisy sensor outputs to estimate the state of a system with uncertain dynamics. In INS/GPS integration case, noisy sensors include GPS receivers and IMU components

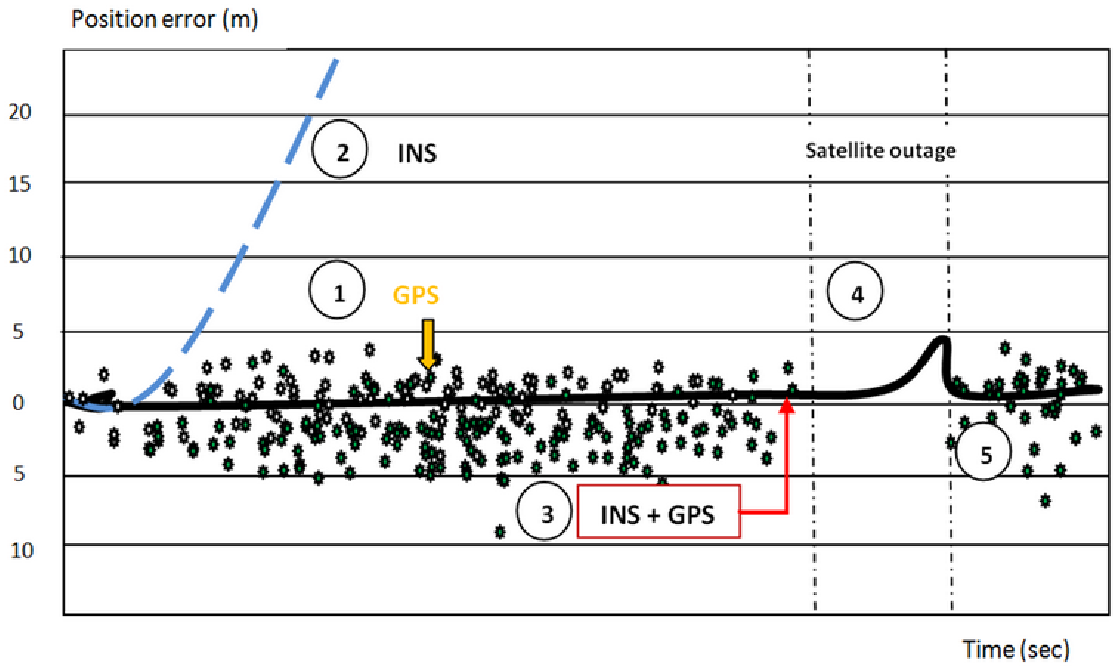


Figure 4.2: INS and GPS integration advantages

and the system state includes the position, velocity and acceleration, attitude rate of a vehicle . Uncertain dynamics unpredictable disturbances of the host vehicle and unpredictable changes in the sensor parameters [2]. Kalman filter optimally estimates position, velocity and attitude errors, as well as errors in inertial and GPS measurements [23].

The types of integration can be categorized by the extent to which data from each component aid the others function. First one is coupling of the systems and depends on the mechanization or the architecture of the system. The second categorization parameter is by the method of combining or fusing the data to obtain position coordinates. The system mechanization is generally understood in two ways, tight coupling and loosely coupling; where no coupling implies no data feedback from either instrument to the other for the purpose of improving its performance. Tightly coupled sensors are treated as belonging to a single system producing complementary types of data. The produced data are produced simultaneously and optimally, and used to enhance the function of of individual sensor components where possible. In a loosely coupled system, processed data from one instrument are feeded back in an aiding capacity to improve the utility of the others performance, but each instrument still has its own individual data processing algorithm.

There is a number of different integration architectures have been developed in the literature to allow INS and GPS to be combined; the level of integration depending in part on whether one is dealing with the creation of a new system, or the addition of GPS updates as a retro-fit to an existing system. A number of INS-GPS integration schemes that are in use or under development for the future are described below. Four main classes of integration architecture may be defined [24], [25]:

- *uncouples systems* in which GPS estimated position is used simply to reset the INS indicated position at regular intervals of time.
- *loosely coupled systems*in which the INS and GPS estimates of position and velocity are compared, resulting difference forming the measurement inputs to a Kalman filter.
- *Tightly coupled systems* in which GPS measurements of pseudo-range and pseudo-rate are compared with estimates of these quantities generated by inertial systems.
- *Deep or ultra-tightly couples systems* which combine the GPS signal tracking function and the INS/GPS integration into a single algorithm.

The real-time feedback of INS velocities to the GPS receiver enables an accurate prediction of GPS pseudo-range and phase at next epoch, thus allowing a smaller bandwidth of the receiver tracking loop in a high-dynamic environment with a subsequent increase in accuracy. Conversely, inertial navigation improves if the GPS solution functions as an update in a Kalman filter estimation of the systematic errors in the inertial sensors. Similarly, GPS positions and velocities may be used to aid the INS solution in a high-dynamic situation by providing a better reference for propagating error states based on the linear approximation.

There are two basic categories of processing algorithms that are centralized and de-centralized. In centralized processing, the raw sensor data is are combined optimally using one central processor to obtain a position solution. This kind of processing is usually associated with tight system integration. Decentralized processing is a sequential approach to processing, where processors of individual systems provide solutions that subsequently are combined with various degrees of optimality by a master processor. In principle, if the statistics of the errors are correctly propagated,the optimal decentralized and centralized methods should yield identical solutions. In some certain cases, such as system fault detection, isolation, and correction capability and the relative computational simplicity makes the decentralized approach more favorable. The centralized approach provides the best performance in navigation solutions that a single robust Kalman filter model. table 4.3shows the advantages and disadvantages of both Kalman filter implementation forms, loosely coupled (decentralized)and tightly-coupled centralized).

The performance of an integrated INS/GPS is a complex process depending on a variety of parameters including:

- Quality and type of inertial sensors.
- Operational aspects.
- The validity of error models.
- The estimation algorithm.

The improvements in trajectory determination are usually sought in the development of better models and estimation algorithms.The improvements in trajectory

Table 4.3: Different forms of Kalman filter implementation (Skaloud, 1999)

Filter Implementation	**Advantages**	**Disadvantages**
Loosely coupled (decentralized).	- Flexible, modular combination. - Small KF, faster processing. - Suitable for parallel processing.	- Sub-optimal performance. - Unrealistic covariance - Four satellites needed for a stable solution - INS data not used for ambiguity estimation
Tightly-coupled (centralized)	- One error state model. - Optimal solution. - GPS measurements can be used with less than 4 satellites. - Direct INS aiding throughout GPS outages. - Faster ambiguity estimation.	- Large size of error state model - More complex processing

determination are usually sought in the development of better models and estimation algorithms. With the rapid increase of computational power, the trend of finding the most suitable error model for a specific system and specific conditions is being replaced by using a multi-model approach in conjunction with some type of adaptive estimation. Another limiting factor band frequency. In the lower frequencies, the INS/GPS integration reduces the overall error; and in the high frequencies, the overall error is not reduced [1].

The vector state estimation can be implemented in open or closed, whether the estimated sensor errors are feeded back to correct the measurements. When properly designed, the closed-loop implementation generally has better performance and is therefore the preferred implementation when using a strapdown INS.

The loosely-coupled filtering approach has been highly popular due its modularity and smaller filter size. Although the arguments for choosing either form of the implementation have been very balanced, the tightly-coupled approach is currently gaining more weight mainly due to the rapid increase in computational power [1].

For the thesis work purpose the loosely-coupled filtering is only considered.

4.4 Loosely Coupled Integration

The loosely coupled integration allows GPS to function autonomously, while simultaneously providing measurement update to the inertial system. The two systems re effectively operated in cascade with the position and/or velocity estimates provided by the GPS navigation calculation forming measurement inputs to an INS/GPS integration Kalman filter. Figure 4.3 shows a simplified representation of loosely coupled INS/GPS integration architecture.

The main advantages of loosely coupled INS/GPS are:

- Simplicity.
- Redundancy.
- This approach can be used with any INS and any GPS receiver.

It is usual to provide a stand alone GPS solution, in addition to the integrated solution, the redundant solution can be used to monitor the integrity of the integrated solution.

In the loosely coupled configuration or decentralized filtering, the navigation solution estimated from the GPS receiver is blinded with the solution estimated from the INS. This configuration is sometimes referred to as cascaded configuration, since position and velocity measurements provided by the GPS receiver are in the first place estimates of a first stage filtering that is implemented within the receiver. This cascaded configuration results in degradation in accuracy when the number of viewed satellites drops bellow four or the satellites geometry becomes poor [26], [27] and [28].

Whilst GPS position updates alone may be used to aid the inertial system, it is more usual to use the velocity measurements for more robust solution. Because there are fewer integration steps between attitude errors and sensor biases, these errors propagate more rapidly as velocity errors. Therefore, velocity measurement allow more immediate estimates of sensor biases and attitude errors to be obtained. However, the use of velocity measurements alone reduces the observability of position errors in the INS. For these reasons, it is customary to use both GPS position and velocity updates to aid the inertial system in most algorithms of this type.

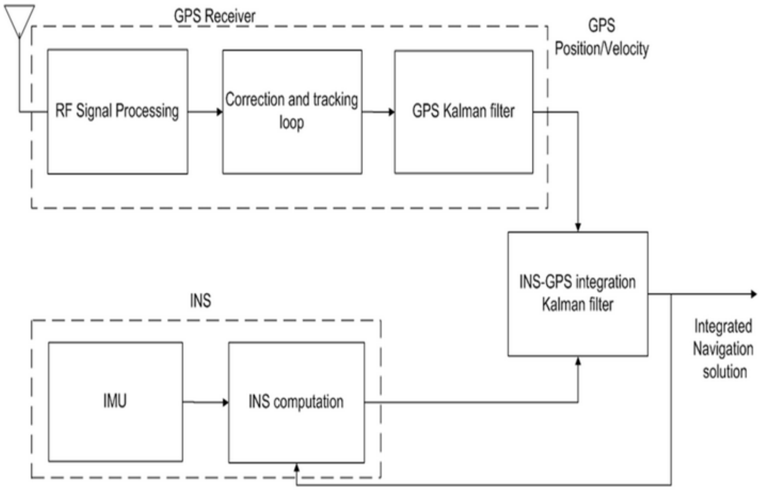

GPS Receiver
GPS Position/Velocity
RF Signal Processing
Correction and tracking loop
GPS Kalman filter
INS-GPS integration Kalman filter
Integrated Navigation solution
INS
IMU
INS computation

Figure 4.3: Basic concepts architecture of Loosely Coupled Integration. (Titterton, D.H, and Weston, 2004)

The main problem with loosely coupled INS/GPS integration stem from the use of the cascaded Kalman filter; the fact that the output of the GPS Kalman filter is used as an input to the integration filter. In formulating a Kalman filter the implicit assumption is made that the measurement errors are uncorrelated, that the measurement noise in white. For the system configuration considered here, such an assumption is not necessarily true; for instance, situation can arise where integration algorithm samples the GPS data faster than the tacking loops can supply independent measurements causing the Kalman filter measurement errors to be time correlated. Further time correlation can arise through multi-path effects, the process whereby several delayed copies of a signal reach the antenna following reflection from nearby surfaces. The correlation time of the GPS Kalamn filter output varies with the tracking loops bandwidths, and can be up to 100s on position and 20s on velocity and in a dynamic receiver can be 10s on position and 0.1-1.0 s on velocity, this is too short for the correlated errors to be estimated, but long enough to slow down the process of estimating the INS errors within the integration filter. As a result of these issues, the selection of Kalman filter measurement update interval becomes critical. Therefore to overcome this problem, the measurement update interval may be increased until the measurement errors are no longer are correlated.

Other main factors that should be taken into consideration when using loosely coupled integration are as follows:

- Signals from at least four separate satellites are required to form and maintain a GPS navigation solution, although degraded navigation can be maintained for short periods using only three satellites, therefore in situations where fewer satellites are in view GPS can not be used to aid the INS.
- The integration filter requires knowledge of the covariance of the GPS filter outputs. This varies with satellite geometry and availability, and for many GPS receivers, covariance data are unreliable or not available at all.

4.5 Mathematical Structure of Linear Kalman Filter

As discussed previously kalman filter is chosen in this thesis work as the data fusion algorithm and among different classes of filter integration architectures; loosely coupled integration is employed because of simplicity, redundancy etc as advantages among other integrations schemes. However this section aims to give a brief introduction of kalman filter and an insight of mathematical frame work of Linear Kalman filter. The mathematical derivation of kalman filter is beyond the intended scope of this thesis work, therefore the reader can refer to stochastic and estimation theory books which will be an excellent reference for him.

4.5.1 Optimal linear estimator, the Kalamn Filter

The kalamn filter was first introduced by R. Kalman in 1960, and an extensive research activities were done afterwards regarding the study, development, extension and applications. However it can be defined as an optimal recursive data processing algorithm, and optimal in the sense that it incorporates all available data information that can be provided with in order to estimate the current value of the variable of interest, and that is achieved with the knowledge of system and measurement sensors dynamics, the statistical description of uncertainty of system dynamics, measurement inaccuracies and noise and as well the initial conditions of the desire variable of interest. And it said to be recursive which means that kalman filter does not require all previous data to be store in memory and to be fused each time measurement is updated. A kalman filter fuses all measurement data in addition to the prior knowledge about system and measuring sensors to obtain an "*optimal*" estimate value of the desire variable in a way that the error is minimized statically [29].

4.5.2 Single-dimension estimator case

If x is considered as a variable of interest of a certain quantity, and if two estimates x_1 and x_2 are provided from two independent measuring devices with variances σ_1^2 and σ_2^2 respectively, and the aim is to fuse both estimates in order to find '*optimum*' weighted mean corresponding to the minimum variance of the estimate $\hat{x}$, this can be expressed mathematically as:

$$\hat{x} = w_1 x_1 + w_2 x_2 \tag{4.14}$$

$$w_1 + w_2 = 1 \tag{4.15}$$

where w_1 and w_2 are weighting factors. The expected value of $\hat{x}$ is given by:

$$\mathbf{E}(\hat{x}) = \mathbf{E}(w_1 x_1) + \mathbf{E}(w_2 x_2) \tag{4.16}$$

The variance of x is defined to be $\mathbf{E}[x - \mathbf{E}(x)^2]$. Therefore the variance of $\hat{x}$ may be written as:

$$\sigma^2 = w_1^2\sigma_1^2 + w_2^2\sigma_2^2 \tag{4.17}$$

Writing $w_2 = w$, and $w_1 = 1 - w$, the variance σ^2 may be expressed as:

$$\sigma^2 = (1 - w)^2\sigma_1^2 + w_2^2\sigma_2^2 \tag{4.18}$$

To find the value of w that minimizes σ^2, equation 4.18 is differentiated with respect to w:

$$\frac{d\sigma^2}{dw} = -2(1 - w)^2\sigma_1^2 + 2w_2^2\sigma_2^2 = 0 \tag{4.19}$$

Which will yields to:

$$w = \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} \tag{4.20}$$

Therefore the "*optimum*" estimated $\hat{x}$ and the corresponding variance can be found by substituting the value of w in equations 4.14 and 4.18. This process in way or another has yields upon the combination of both independent estimates to form a *weighted mean with minimum variance*, and thus, *maximum probability*.

In our navigation problem one estimate say x_1 is provided by updating a previous best estimate of INS according to the derived motion equations, whilst the another, x_2, can be obtained from an aided measurement device i.e. GPS, accordingly equations 4.14 and 4.18 can be re-written in the following form:

$$\hat{x} = x_1 + w(x_1 - x_2) \tag{4.21}$$

$$\sigma^2 = \sigma_1^2(1 - w) \tag{4.22}$$

From the above equations it can be concluded that the estimate x_1 and its corresponding variance σ^2 are improved by the measurement x_2, for instance estimate x_1 can be a vector of multi variables like position, velocity and attitude. This can be generalized to a multi-dimension form for a full kalman filter implementation.

4.6 Discrete-Time Linear Kalman Filter

Before moving to introduce the Kalamn filter equations, the intent in this thesis work is to implement these equations in real time, in which measurements will be provided in discrete intervals of time, consequently the continuous equations should be presented in the form of difference equations, to do so the continuous equations should be discretized.

If the above estimate quantities x_1 and x_2 are an n-elements vectors i.e. x_1 **and** x_2 representing a multi-dimensional vector quantity $\mathbf{x}$, and behavior of the system under consideration can be described by the following set of linear equations:

$$\dot{\mathbf{x}}(t) = \mathbf{F}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{G}\mathbf{w}(t) \tag{4.23}$$

And if system measurements are given by the following equation:

$$\mathbf{z}(t) = \mathbf{H}\mathbf{x}(t) + \mathbf{V}(t) \tag{4.24}$$

where:

1. $\mathbf{x}$(t) is an n-vector elements represents system states, $\mathbf{u}$(t)is am-vector of deterministic inputs and $\mathbf{w}$(t)is the system noise, which is normally distributed with a power spectral density of described by $\mathbf{Q}$ and has a zero mean.

2. $\mathbf{F}$ is system transition matrix with $n \times n$ matrix and $\mathbf{B}$ is $n \times m$ system input matrix.

3. $\mathbf{z}$(t) is l-vector describe the observation of the system and $\mathbf{V}$(t) is observation (measurements) noise.

4. $\mathbf{H}$ is $l \times n$ measurements matrix which as well has a zero-mean and normally distributed with power spectral density given by $\mathbf{R}$.

for discretization we may use the Jacobian matrix $\mathbf{J}_f(k)$ which can be calculated at time t_k in discrete time is as:

$$\begin{aligned} \mathbf{J}_{\mathbf{F}}(k) &= \frac{\partial \mathbf{F}_k(x_k)}{\partial x_k} \\ &\approx \mathbf{I} + \frac{\partial \mathbf{F}_k(t_k, x_{t_k})}{\partial x_k} dt \\ &= \mathbf{I} + \mathbf{J}_c(t_k) dt \end{aligned} \tag{4.25}$$

where $\mathbf{J}_c(t_k)$ is the continuous time jacobian matrix at time t_k. If however it is constant over the sampling interval $\triangle t$ and it follows from equation 4.25 that, the discrete time transition matrix is computed by

$$\begin{aligned} \mathbf{F}(k) &= exp(\mathbf{F}\triangle t) \\ &= \mathbf{I} + \mathbf{F}\triangle t + \frac{(\mathbf{F}\triangle t)^2}{2!} + \cdots \end{aligned} \tag{4.26}$$

where $\triangle t$ can be considered as the sampling time of the inertial unit. The equation discretisation can only be considered to the second term since any extra terms are of negligible value, thus equation 4.26 can be given as follows:

$$\hat{\mathbf{x}}(k+1|k) = \mathbf{F}(k+1)\hat{\mathbf{x}}(k|k) \tag{4.27}$$

The kalman filter algorithm mainly consists of two steps or two distinct sets of equations:

- Prediction step, it is concerned with the prediction of the current state based on the previous best estimate.
- Prediction or Observation update step, which concerned with updating the prediction states calculated in previous step by combining the prediction with the new observation.

4.6.1 Prediction stage

The best prediction of the state at time t_k is denoted by $\mathbf{x}(k|k)$ and given by:

$$\hat{\mathbf{x}}(k+1|k) = \mathbf{F}(k)\mathbf{x}(k|k) \tag{4.28}$$

And the expected value of the covariance matrix i.e. the prediction uncertainty $\mathbf{P}(k+1|k)$ is given by:

$$\mathbf{P}(k+1|k) = \mathbf{F}(k+1)\mathbf{P}(k|k)\mathbf{F}^T(k+1) + \mathbf{G}(k+1)\mathbf{Q}'(k+1)\mathbf{G}^T(k+1) \tag{4.29}$$

4.6.2 Observation Update

When a new observation arrives $\mathbf{z}(k+1)$ at time t_{k+1} by the aid instrument, it is compared with the prediction computed from the system model, then the correction stage starts by combining this new observation with the previous prediction to generate best estimate, thus the best estimate (correction) at time t_{k+1} is given by:

$$\hat{\mathbf{x}}(k+1|k+1) = \mathbf{x}(k+1|k) + \mathbf{W}(k+1)\mathbf{Innov}(k+1) \tag{4.30}$$

Where:

- $\mathbf{W}(k+1)$ is the Kalman gain and is given by:

$$\mathbf{W}(k+1) = \mathbf{P}(k+1|k)\mathbf{H}^T(k+1)[(\mathbf{H}(k+1)\mathbf{P}(k+1|k)\mathbf{H}^T(k+1) + \mathbf{R}(k+1))]^{-1} \tag{4.31}$$

- In real practice the true value is not existing thus it is replaced by the innovation, $\mathbf{Innov}(k+1)$ which is the difference between the observation given at time t_k+1 and the prediction at time t_k+1 and generally defined as:

$$\mathit{Innovation} = \mathit{Observation} - \mathit{Prediction} \tag{4.32}$$

$$\mathbf{Innov}(k+1) = \mathbf{z}(k+1) - \mathbf{H}(k+1)\hat{\mathbf{x}}(k+1|k) \tag{4.33}$$

The uncertainty in this stage is as well updated and computed as follows:

$$\mathbf{P}(k+1|k+1) = (\mathbf{I}-\mathbf{W}(k+1)\mathbf{H}(k+1))\mathbf{P}(k|k)(\mathbf{I}-\mathbf{W}(k+1)\mathbf{H}(k+1))^T+\mathbf{W}(k+1)\mathbf{R}(k+1)\mathbf{W}^T(k+1) \tag{4.34}$$

based on the above the system states may be updated at each time a new observation is arrived by implementing equations 4.30,4.31and 4.34.

Chapter 5

Real time INS/GPS Navigation

5.1 Introduction

This chapter of this thesis work concerns with the real time implementation of GPS aided INS navigation system. In this chapter the navigation equation will be developed, these equations which must be solved in order to extract position, velocity and attitude information provided by inertial sensors i.e. gyroscopes and accelerometers fixed on a starpdown platform that have been discussed in chapter 2. The most demanding of the computing tasks, in terms of computer loading, are the attitude computation and the specific force vector resolution however in the presence of the high frequency motion, the implementation of these tasks in real time creates a substantial computing burden for the strapdown system processor, even with modern ones.

The limitation of the computer technologies available at the time of the early attempts to produce a strapdown inertial navigation system was the main factor to delay the process of development. The lack of the computing speed was a major obstacle to achieve fast and accurate attitude computation, as well as the physical size of early computers and processors which delayed the development of strapdown systems, particulary for airborne applications, therefore the performance that could be achieved at the time of developing strapdown systems were limited specially under high frequency motion conditions. As a results of such difficulties much research efforts to be directed towards the development of efficient navigation computing algorithms.

5.2 Attitude Computations

As discussed in the introduction, attitude computation is one of the most critical computation tasks in strapdown systems. It can be said that the ability of the strapdown algorithm to keep track of body attitude accurately in a severe vibratory environment may be the critical factor in determining its performance, if accurate navigation to be achieved [3].

There are different approaches to represent the attitude namely Direction Cosine Matrix, Euler and Quaternion approach. Direction Cosine Matrix approach is used in this work as it is conventionally used in low-cost inertial navigation systems, because of the best representation, has minimal trigonometric calculations and it is considered the least expensive representation in terms of computation load [30].

One of the main steps in attitude computation algorithm is the transformation of body frame accelerations into the mechanization frame and the removal of the gravitation component in which body frame accelerations are integrated once to compute the velocity and twice to compute the position. In this work the navigation frame mechanization is considered and up-to-date starpdown mathematical frame is utilized in which velocity and position equations are computed using a Jordan-like attitude updating method [31],[32].

The attitude algorithm compute the direction cosine matrix (DCM) that will transform the measured specific forces into the desired navigation frame based on the Euler angles ϕ, θ and ψ which is given as follows:

$$\mathbf{C}_b^n = \begin{pmatrix} \cos\theta\cos\psi & -\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi & \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi \\ \cos\theta\sin\psi & \cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\theta\cos\psi \end{pmatrix} \tag{5.1}$$

Therefore if we consider that we are navigating with respect to the local geographic frame then in order to update the DCM, it is required to solve the matrix differential equation of the form:

$$\dot{\mathbf{C}} = \mathbf{C}\boldsymbol{\Omega} \tag{5.2}$$

where $\boldsymbol{\Omega}$ is a skew symmetric matrix formed from the elements of the turn rate values vector ω in the body frame with respect to the navigation reference frame.Thus, equation 5.2 takes the following form when navigating with respect to the local geographical frame:

$$\dot{\mathbf{C}}_b^n = \mathbf{C}_b^n \boldsymbol{\Omega}_{ib}^b - \boldsymbol{\Omega}_{in}^n \mathbf{C}_b^n \tag{5.3}$$

$\mathbf{C}_b^n \boldsymbol{\Omega}_{ib}^b$ is a function of the body rates sensed by gyroscopes and it is required to update the DCM. While the second term of equation 5.3, $-\boldsymbol{\Omega}_{in}^n \mathbf{C}_b^n$, takes into account updating the DCM for the lower navigation frame rotations rates.

The general updating computation algorithm of equation 5.3 is constructed by using direction cosine matrix product chain rule, using two steps of direction cosine rotations, the first takes care of body rotations which can be implemented at the higher rate computer cycle, and the second takes care of local navigation frame rotations which can be implemented at the lower rate l computer cycle)[33],[3]. The two rotation are described by equations 5.4 and 5.5

$$\mathbf{C}_{b(k+1)}^{n(l)} = \mathbf{C}_{b(k)}^{n(l)} \mathbf{C}_{b(k+1)}^{b(k)} \tag{5.4}$$

$$\mathbf{C}_{b(k+1)}^{n(l+1)} = \mathbf{C}_{n(l)}^{n(l+1)} \mathbf{C}_{b(k+1)}^{n(l)} \tag{5.5}$$

where:

k - high speed computer cycle index for body frame angular motion updates to $\mathbf{C}_b^n$.

l - low speed computer cycle index for navigation frame angular motion updates to $\mathbf{C}_b^n$.

$\mathbf{C}_{b(k+1)}^{n(l)}$ - DCM which transforms vectors from body frame coordinates at time t_{k+1} to navigation frame coordinates at time t_l.

$\mathbf{C}_{b(k)}^{n(l)}$ - DCM which transforms vectors from body frame coordinates at time t_k to navigation frame coordinates at time t_l.

$\mathbf{C}_{b(k+1)}^{b(k)}$ - DCM which transforms vectors from body frame coordinates at time t_{k+1} to body frame coordinates at the k.

$\mathbf{C}_{b(k+1)}^{n(l+1)}$ - DCM which transforms vectors from body frame coordinates at time t_{k+1} to navigation frame coordinates at time t_{l+1}.

$\mathbf{C}_{n(l)}^{n(l+1)}$ - DCM which transforms vectors from navigation frame coordinates at time t_l to navigation frame coordinates at time t_{l+1}.

Therefore the attitude computation should be split into two parts:

- Update the DCM for the rotations of body frames: to compute $\mathbf{C}_{b(k+1)}^{b(k)}$, equation 5.6 is used [3]:

$$\mathbf{C}_{b(k+1)}^{b(k)} = \mathbf{I} + \frac{\sin \bar{\sigma}}{\bar{\sigma}} [\sigma \times] + \frac{1 - \cos \bar{\sigma}}{\bar{\sigma}^2} [\sigma \times]^2 \tag{5.6}$$

The skew matrix $[\sigma \times]$ in equation 5.6 is computed by equation 5.7

$$[\sigma \times] = \begin{pmatrix} 0 & -\sigma_z & \sigma_y \\ \sigma_z & 0 & -\sigma_x \\ -\sigma_y & \sigma_x & 0 \end{pmatrix} \tag{5.7}$$

Where σ is defined as the angle vector with a magnitude and direction such that the rotation of body frame about it via an angle will be equal to the magnitude of σ and will rotate the body frame from its orientation at time t_k to its position at time t_{k+1} and$(\sigma_x, \sigma_y, \sigma_z)$ are the components of σ, computed as follows:

$$\begin{aligned} \sigma_x &= \omega_{ibx}^{b}(t_{k+1} - t_k) + \frac{1}{2}\omega_{ibx}^{b\,2}(t_{k+1} - t_k)^2 \\ \sigma_y &= \omega_{iby}^{b}(t_{k+1} - t_k) + \frac{1}{2}\omega_{iby}^{b\,2}(t_{k+1} - t_k)^2 \\ \sigma_z &= \omega_{ibz}^{b}(t_{k+1} - t_k) + \frac{1}{2}\omega_{ibz}^{b\,2}(t_{k+1} - t_k)^2 \end{aligned} \tag{5.8}$$

And $\bar{\sigma}$ is the magnitude of σ, which is given in equation 5.9

$$\bar{\sigma} = \sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2} \tag{5.9}$$

- Update the DCM for the rotations of navigation frame: Similar to equation 5.6 $\mathbf{C}_{n(l)}^{n(l+1)}$ can be computed using following equation 5.10:

$$\mathbf{C}_{n(l)}^{n(l+1)} = \mathbf{I} + \frac{\sin \bar{\varsigma}}{\bar{\varsigma}} [\varsigma \times] + \frac{1 - \cos \bar{\varsigma}}{\bar{\varsigma}^2} [\varsigma \times]^2 \tag{5.10}$$

Similarly, equations 5.7and 5.9are applied where $(\varsigma_x, \varsigma_y, \varsigma_z)$ are the components of ς, and are computed as follows:

$$\begin{aligned}
\varsigma_x &= \omega_{inx}^n (t_{k+1} - t_k) + \frac{1}{2} \omega_{inx}^{n\,2} (t_{k+1} - t_k)^2 \\
\varsigma_y &= \omega_{iny}^n (t_{k+1} - t_k) + \frac{1}{2} \omega_{iny}^{n\,2} (t_{k+1} - t_k)^2 \\
\varsigma_z &= \omega_{inz}^n (t_{k+1} - t_k) + \frac{1}{2} \omega_{inz}^{n\,2} (t_{k+1} - t_k)^2
\end{aligned} \tag{5.11}$$

And ω_{in}^n is computed using equation 5.12:

$$\omega_{in}^n = \omega_{ie}^n + \omega_{en}^n \tag{5.12}$$

5.3 Discrete form Of Velocity/Position Navigation Equations: Unified Mathematical Framework

The discrete form of strapdown inertial navigation algorithm which is implemented in the navigation computer (Microprocessor) to compute the velocity/position is based on the unified mathematical framework proposed by [31]. The update equations for position and velocity are excused to give the exact solution but under the assumption of constant angular rate and specific force, therefore the input for the algorithm is the direct integral of angular rate and specific force vector elements that are provided by rate gyros and accelerometers [31].

$$\boldsymbol{V}^n(k+1) = \boldsymbol{V}^n(k) + \boldsymbol{C}_b^n(k)\Delta\boldsymbol{V}_{sf}^n(k+1) + \Delta\boldsymbol{V}_g^n(k+1) \tag{5.13}$$

where $\Delta\boldsymbol{V}_{sf}^n(k+1)$ is the change in velocity due to specific force acceleration and $\Delta\boldsymbol{V}_g^n(k+1)$ is the change in velocity due to gravity. $\Delta\boldsymbol{V}_{sf}^n(k+1)$ is give by equation 5.14:

$$\Delta\boldsymbol{V}_{sf}^n(k+1) = \left[\mathbf{I} + f_2(\bar{\sigma})\,[\sigma\times] + f_3(\bar{\sigma})\,[\sigma\times]^2\right]\nu \tag{5.14}$$

Where ν and its components ν_x, ν_y and ν_z is the integral of specific force vector $\mathbf{F}^b$ in the body frame and computed as follows:

$$\nu = \int_t^{t+1} \mathbf{F}^b.dt \tag{5.15}$$

And $f_3(\bar{\sigma})$ is given as follows:

$$f_3(\bar{\sigma}) = \frac{1}{\bar{\sigma}^2}\left(1 - f_1(\bar{\sigma})\right) \tag{5.16}$$

and $f_1(\bar{\sigma})$ and $f_2(\bar{\sigma})$ are given by:

$$f_1(\bar{\sigma}) = \frac{\sin\bar{\sigma}}{\bar{\sigma}} \tag{5.17}$$

$$f_2(\bar{\sigma}) = \frac{1-\cos\bar{\sigma}}{\bar{\sigma}^2} \tag{5.18}$$

The position can be calculated by the following differential equation 5.19:

$$\dot{\boldsymbol{P}}_e^n = \boldsymbol{V}_e^n \tag{5.19}$$

However, the discrete form of the above equation is applied by using the simple rectangular integration which is given by the following form:

$$\boldsymbol{P}(k+1) = \boldsymbol{P}(k) + \boldsymbol{V}(k+1)(t_{k+1} - t_k) \tag{5.20}$$

The above framework approach that applied for velocity can be used as well to calculate position as shown in equation 5.21:

$$\boldsymbol{P}^n(k+1) = \boldsymbol{P}^n(k) + \boldsymbol{V}^n(k)\Delta t_k + \boldsymbol{C}_b^n(k)\Delta\boldsymbol{P}_{sf}^n(k+1) + \Delta\boldsymbol{P}_g^n(k+1) \tag{5.21}$$

Where $\Delta\boldsymbol{P}_{sf}^n(k+1)$ is the change in position due to specific force acceleration and $\Delta\boldsymbol{P}_g^n(k+1)$ is the change in position due to gravity, $\Delta\boldsymbol{P}_{sf}^n(k+1)$ is computed as follows:

$$\Delta\boldsymbol{P}_{sf}^n(k+1) = \left[\mathbf{I} + 2f_3(\bar{\sigma})\,[\sigma\times] + 2f_4(\bar{\sigma})\,[\sigma\times]^2\right]\mathbf{S}_\nu \tag{5.22}$$

Where $\mathbf{S}_\nu$ is the double integral of specific force since computer cycle is $k+1$,

$$\mathbf{S}_\nu = \int_t^{t+1} \nu.dt \tag{5.23}$$

And $f_2(\bar{\sigma})$ and $f_3(\bar{\sigma})$ are as in equations: 5.18 and 5.16, respectively, and $f_4(\bar{\sigma})$

$$f_4(\bar{\sigma}) = \frac{1}{\bar{\sigma}^2}\left(\frac{1}{2} - f_2(\bar{\sigma})\right) \tag{5.24}$$

5.4 INS Error Models

There have been wide research activities regarding INS error modelling specifically position, velocity and orientation. In the literature two main approaches have been adopted; namely the psi or computer approach and phi angle or true frame approach. The importance of developing error models is to derive set of error equations or to qualitatively evaluate the propagation and dynamics of the INS errors in order to understand the effect of these errors on INS system development as the accuracy of inertial navigation will depend on the accuracy of the sensors employed and on the computer implementation of the inertial algorithms. In addition, the developed inertial navigation error models equations will be used later on as the process model in the Kalman Filter implementation. Inertial navigation error models are archived by linearizing (perturbing) the nominal differential equations, and linearizing can be done into forms as mentioned before [34], [4][35]:

1. The ψ angle (Computer Approach) where the analysis occurs by using the local geographical reference frame as the computer frame and because the computer frame is known the perturbation of the angular position and rate are zero.
2. The ϕ angle (True Frame Approach) where the perturbation of ECEF frame occurs using the true position of the vehicle.

In this work the ψ angle approach is adopted as the implementation of the perturbation in the computer frame is more beneficial as the misalignment between the computer frame and the true frame is independent of the position of the computer frame. However, perturbation in the true frame is simpler, but the misalignment is coupled with the position. It is shown that developed error models through both approaches techniques are equivalent[4],[36][37].

5.4.1 Attitude Error Equation (Psi angle approach)

As discussed previously in chapter 2 the orientation of host vehicle can be given by direction cosine matrix $\boldsymbol{C}_b^n$. If the platform frame is defined as the frame in which the transformed accelerations and angular rates from the accelerometers and gyros triad cluster are resolved, and the computer frame as the local level frame located in the INS computer in which navigation equations are solved, then the misalignment between computer frame and platform frame is given by (ψ) angle and expressed as $\psi = [\psi_x, \psi_y, \psi_z]^T$. Therefore the attitude error equation is given by:

$$\delta\dot{\psi} \approx \delta\omega_{in}^n - \omega_{in}^n \delta\psi - \boldsymbol{C}_b^n \delta\omega_{ib}^b \tag{5.25}$$

Where $[\delta\psi\times]$ is a symmetrical skew matrix of attitude errors, and written as:

$$[\delta\psi\times] = \begin{pmatrix} 0 & -\delta\psi_z & \delta\psi_y \\ \delta\psi_z & 0 & -\delta\psi_x \\ -\delta\psi_y & \delta\psi_x & 0 \end{pmatrix} \tag{5.26}$$

And the angular rate of the body frame with respect to the navigation frame, ω_{nb}^n, is the angular rate of the body frame with respect to the inertial frame, ω_{ib}^n, minus the angular rate of the navigation frame with respect to the inertial frame,ω_{in}^n. Thus its error is expressed as:

$$\delta\omega_{nb}^n = \delta\omega_{ib}^n - \delta\omega_{in}^n \tag{5.27}$$

Therefore to update the direction cosine matrix the above misalignment angle vector ψ, can be used as follows:

$$\boldsymbol{C}_b^n = [\mathbf{I} + [\delta\psi\times]]\,\tilde{\boldsymbol{C}}_b^n \tag{5.28}$$

5.4.2 Velocity/Position Error Equations: General Motion

The general navigation equation if the navigation frame is considered as the mechanization frame is expressed as follows [3]:

$$\dot{\boldsymbol{V}}_e^n = \mathbf{f}^n - [2\omega_{\mathbf{ie}}^{\mathbf{n}} + \omega_{\mathbf{en}}^{\mathbf{n}}] \times \boldsymbol{V}_e^n + \boldsymbol{g}_l^n \tag{5.29}$$

The errors in the velocity/position are defined as the estimated quantity minus the true quantity as follows:

$$\delta v^n \triangleq \widehat{v}^n - v^n \tag{5.30}$$

$$\delta p^n \triangleq \widehat{p}^n - p^n \tag{5.31}$$

Thus the velocity error in the navigation frame may be found by linearizing (perturbing) the nominal equation of 5.29 and substitute in the above given equation, then it is shown that the rate of change of the velocity error in the navigation frame, $\delta\dot{\mathbf{v}}^n$, will be equal to the Coriolis acceleration caused by the velocity error, $2\omega_{in}^{n} \times \delta v^{n}$, together with the acceleration caused by the misalignment, $[\mathbf{f}^{n}\times]\delta\psi$, and the errors associated with the measurement of acceleration in body frame, $\boldsymbol{C}_{b}^{n}\delta\mathbf{f}^{b}$, transformed into the navigation frame. Starting from equation 5.29 and applying perturbation of the nominal state results into [3] [32]:

$$\begin{aligned}\delta\dot{\mathbf{v}}^{n} &= \dot{\tilde{\mathbf{v}}}^{n} - \dot{\mathbf{v}}^{n} \\ &= \tilde{\mathbf{C}}_{b}^{n}\tilde{\mathbf{f}}^{b} - \mathbf{C}_{b}^{n}\mathbf{f}^{b} - (\tilde{\omega}_{en}^{n} + 2\tilde{\omega}_{ie}^{n}) \times \tilde{\mathbf{v}}^{n} + (\omega_{en}^{n} + 2\Omega_{ie}^{n}) \times \mathbf{v}^{n} + \tilde{\mathbf{g}}_{l}^{n} - \mathbf{g}_{l}^{n}\end{aligned} \tag{5.32}$$

substituting for

$$\delta\dot{\mathbf{v}}^{n} = \dot{\tilde{\mathbf{v}}}^{n} - \dot{\mathbf{v}}^{n} \tag{5.33}$$

$$\delta\mathbf{f}^{b} = \tilde{\mathbf{f}}^{b} - \mathbf{f}^{b} \tag{5.34}$$

$$\delta\mathbf{v}^{n} = \tilde{\mathbf{v}}^{n} - \mathbf{v}^{n} \tag{5.35}$$

$$\delta\omega_{ie}^{n} = \tilde{\omega}_{ie}^{n} - \omega_{ie}^{n} \tag{5.36}$$

$$\delta\omega_{en}^{n} = \tilde{\omega}_{en}^{n} - \omega_{en}^{n} \tag{5.37}$$

$$\delta\mathbf{g}_{l}^{n} = \tilde{\mathbf{g}}_{l}^{n} - \mathbf{g}_{l}^{n} \tag{5.38}$$

gives the general velocity navigation equation:

$$\delta\dot{\mathbf{v}}^{n} = -[\mathbf{f}^{n}\times]\delta\psi + \boldsymbol{C}_{b}^{n}\delta\mathbf{f}^{b} - 2(\omega_{in}^{n} \times \delta\mathbf{v}^{n} + \delta\omega_{in}^{n} \times \mathbf{v}^{n}) + \delta\mathbf{g}_{l}^{n} \tag{5.39}$$

The velocity is defined by the derivative of the position, then perturbation of equation 2.14 results into position error equation as shown below:

$$\delta\dot{\mathbf{p}}^{n} = \delta\mathbf{v}^{n} \tag{5.40}$$

It can be noted that the propagation of attitude errors is independent of position and the attitude errors $\delta\dot{\boldsymbol{\Psi}}$ are simply associated with the rotation rate measurements ω_{ib}^{b}, transformed over to the local geographic frame via C_{b}^{n}. The errors which associated with the body rates measurements are purely an output of the gyros sensors, which makes gyros most critical components of the INS.

However, it is obvious from the the above position/velocity error equations are function of specific forces $\mathbf{f}^n$ in the navigation frame, body frame specific forces errors $\delta\mathbf{f}^n$ resulted from accelerometers measurements and misalignment angles $\delta\boldsymbol{\Psi}$.

It can also be noted that the result of the above perturbation is a linear error equations, which makes it an straight forward decision to implement a linear filter, i.e. Kalman Filter therefore the above developed error equations can be considered as the process model in the proposed Kalman Filter structure. For more details regarding the derivation of the attitude, velocity and position differential error equations refer to [3],[4] and [35].

5.4.3 Velocity/Position Error Equations: Planner Motion (x,y,ψ)

The same analysis can be employed for Velocity/Position in the general navigation equation but assuming that the navigation frame is fixed on the Earth, thus the error in the angular rate of the navigation frame becomes zero and assuming the gravity vector is known thus its error is also ignored, then the rate of change of the velocity error equation becomes:

$$\delta\dot{\mathbf{v}}^n = -2\omega_{in}^n \times \delta v^n + [\mathbf{f}^n\times]\delta\psi + \boldsymbol{C}_b^n\delta\mathbf{f}^b \qquad (5.41)$$

If the velocity error is maintained within small value, the Coriolis term due to this error can be ignored and if the gravity term is ignored then $\delta\dot{\mathbf{v}}^n$ reduces to:

$$\delta\dot{\mathbf{v}}^n = [\mathbf{f}^n\times]\delta\psi + \boldsymbol{C}_b^n\delta\mathbf{f}^b \qquad (5.42)$$

Similarly, the velocity is defined by the derivative of the position, then perturbation of equation 2.14 results into position error equation as shown below:

$$\delta\dot{\mathbf{p}}^n = \delta\mathbf{v}^n \qquad (5.43)$$

The attitude error equation 5.25 can be reduced to:

$$\delta\dot{\psi} = -\boldsymbol{C}_b^n \delta\omega_{ib}^b \tag{5.44}$$

In the same manner the unified mathematical frame work proposed by [31] can be applied to the 3DOF navigation equation, therefore equation 5.13 becomes:

$$\boldsymbol{V}^n(k+1) = \boldsymbol{V}^n(k) + \boldsymbol{C}_b^n(k)\Delta\boldsymbol{V}_{sf}^n(k+1) \tag{5.45}$$

where $\boldsymbol{V}^n(k)$ is:

$$\boldsymbol{V}^n(k+1) = \begin{pmatrix} Vx & Vy \end{pmatrix} \tag{5.46}$$

The direction cosine matrix DCM is given by:

$$\mathbf{C}_b^n = \begin{pmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{pmatrix} \tag{5.47}$$

And $\Delta\boldsymbol{V}_{sf}^n(k+1)$ is the change in velocity due to specific force acceleration. $\Delta\boldsymbol{V}_{sf}^n(k+1)$ is give by equation 5.14:

$$\Delta\boldsymbol{V}_{sf}^n(k+1) = \left[\mathbf{I} + f_2(\bar{\sigma})\left[\sigma\times\right] + f_3(\bar{\sigma})\left[\sigma\times\right]^2\right]\nu \tag{5.48}$$

Where ν and its components ν_x and ν_y is the integral of specific force vector $\mathbf{F}^b$ in the body frame and computed as follows:

$$\nu = \int_t^{t+1} \mathbf{F}^b.dt \tag{5.49}$$

And $\bar{\sigma}$ which is given by equation 5.9 becomes:

$$\begin{aligned} \bar{\sigma} &= \sqrt{\sigma_z^2} \\ &= \sigma_z \end{aligned} \tag{5.50}$$

And $f_3(\bar{\sigma})$ is given as follows:

$$f_3(\bar{\sigma}) = \frac{1}{\bar{\sigma}^2}\left(1 - f_1(\bar{\sigma})\right) \tag{5.51}$$

and $f_1(\bar{\sigma})$ and $f_2(\bar{\sigma})$ are given by:

$$f_1(\bar{\sigma}) = \frac{\sin\bar{\sigma}}{\bar{\sigma}} \tag{5.52}$$

$$f_2(\bar{\sigma}) = \frac{1 - \cos\bar{\sigma}}{\bar{\sigma}^2} \tag{5.53}$$

However, similarly the discrete form is applied to equation 5.19 by using the simple rectangular integration which is given by the following form:

$$\boldsymbol{P}(k+1) = \boldsymbol{P}(k) + \boldsymbol{V}(k+1)(t_{k+1} - t_k) \tag{5.54}$$

And as well the above framework approach that applied for velocity can be used as well to calculate position as shown in equation 5.55:

$$\boldsymbol{P}^n(k+1) = \boldsymbol{P}^n(k) + \boldsymbol{V}^n(k)\Delta t_k + \boldsymbol{C}_b^n(k)\Delta\boldsymbol{P}_{sf}^n(k+1) \tag{5.55}$$

Where $\Delta\boldsymbol{P}_{sf}^n(k+1)$ is the change in position due to specific force acceleration and $\Delta\boldsymbol{P}_{sf}^n(k+1)$ is computed as follows:

$$\Delta\boldsymbol{P}_{sf}^n(k+1) = \left[\mathbf{I} + 2f_3(\bar{\sigma})\left[\sigma\times\right] + 2f_4(\bar{\sigma})\left[\sigma\times\right]^2\right]\mathbf{S}_\nu \tag{5.56}$$

Where $f_2(\bar{\sigma})$ and $f_3(\bar{\sigma})$ are as in equations: 5.53 and 5.51, respectively, and $f_4(\bar{\sigma})$

$$f_4(\bar{\sigma}) = \frac{1}{\bar{\sigma}^2}\left(\frac{1}{2} - f_2(\bar{\sigma})\right) \tag{5.57}$$

5.5 State Space Representation

The above equations 5.26, 5.41 and 5.43 can be expressed in a single error equation matrix using state space representation:

$$\begin{aligned} \delta\mathbf{x}(t) &= \mathbf{F}(t)\delta\mathbf{x}(t-1) + \mathbf{G}(t)\mathbf{w}(t) \\ \delta\mathbf{z}(t) &= \mathbf{H}(t)\delta\mathbf{x}(t) + \mathbf{v}(t) \end{aligned} \tag{5.58}$$

where $F(k)$ is the system transition matrix, $G(k)$ is the system noise transition matrix and $w(k)$ is the system noise vector with noise strength $Q(k)$, $\delta\mathbf{z}(t)$ is the observation error, $\mathbf{H}(t)$ is the observation matrix and $\mathbf{v}(t)$ is the observation noise with with noise covariance matrix $\mathbf{R}$. $\delta\mathbf{x}(t)$ is given be the following:

$$\delta\mathbf{x} = \begin{pmatrix} \delta\dot{\mathbf{P}} & \delta\dot{\mathbf{V}} & \delta\dot{\mathbf{\Psi}} \end{pmatrix}^T \tag{5.59}$$

$$\delta\mathbf{w} = \begin{pmatrix} \delta\mathbf{f}_{bx} & \delta\mathbf{f}_{by} & \delta\mathbf{f}_{bz} & \delta\omega_x & \delta\omega_y & \delta\omega_z \end{pmatrix}^T \tag{5.60}$$

$$\delta\mathbf{z} = \begin{pmatrix} \delta\mathbf{P} \\ \delta\mathbf{V} \end{pmatrix} \tag{5.61}$$

Where $\delta\mathbf{P}$ and $\delta\mathbf{V}$ is the difference between the observed measurement (GPS) and the predicted observation (INS). The state transition matrix $\mathbf{F}$ is expressed as:

$$\mathbf{F} = \begin{pmatrix} \mathbf{F}_{ins} & \mathbf{F}_{coupled} \\ \mathbf{0} & \mathbf{F}_{biases} \end{pmatrix} \tag{5.62}$$

Where

$$\mathbf{F}_{ins} = \begin{pmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -2\omega_{in}^n\times & (\mathbf{C}_b^n\mathbf{f}^b)\times \\ \mathbf{0} & \mathbf{0} & -\omega_{in}^n\times \end{pmatrix} \tag{5.63}$$

$$\mathbf{F}_{coupled} = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{C}_b^n & \mathbf{0} \\ \mathbf{0} & -\mathbf{C}_b^n \end{pmatrix} \tag{5.64}$$

$$\mathbf{F}_{biases} = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \tag{5.65}$$

And noise transition matrix array G can be expressed as:

$$\mathbf{G} = \begin{pmatrix} \mathbf{G}_{ins} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_{biases} \end{pmatrix} \tag{5.66}$$

Where:

$$\mathbf{G}_{ins} = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{C}_b^n & \mathbf{0} \\ \mathbf{0} & -\mathbf{C}_b^n \end{pmatrix} \tag{5.67}$$

$$\mathbf{G}_{biases} = \mathbf{I}_{6\times6} \tag{5.68}$$

And observation transition matrix **H** is given by:

$$\mathbf{H} = \begin{pmatrix} \mathbf{I}_{6\times6} & \mathbf{0} \end{pmatrix} \tag{5.69}$$

5.6 Filter Architecture

Observed (GPS) and estimated (INS) data once are ready can be fused in two different integration schemes namely *direct* and *indirect* feedback. The decision that should be taken to go with either one of the filter integration schemes depends on the selected sensors and overall system complexity, however in direct configuration, filter *directly* estimates the states of interest (Position, Velocity and Attitude) but in the indirect architecture the filter estimates the errors in the states of interest and applies them to correct the desired states i.e. the filter indirectly estimate the states [38]. As our design selection is strapdown INS structure in which the accelerometers and rate-gyros sensors are rigidly attached to a fix platform and undergo the same dynamic as the platform, the direct structure has serious drawbacks; for instance the speed of the filter should be high in order to pick up the vehicle dynamics, in addition the dimensionality of the estimated states which is nine states (Three positions, Three Velocities and Three attitudes) plus sensors error states which could be at least six states, may cause a significant computational load bottleneck on the selected navigation processor that can not be met. Another serious drawback in direct filter structure configuration in which if the filter fails may cause the overall navigation system to fail. On the other hand the indirect filter structure which suits our strapdown INS system and real-time implementation as the filter is decoupled from the INS loop thus the sampling frequency of the filter can be relaxed, additionally the navigation system becomes more robustness and reliable to filter failure, as filter can run in the background mode and can be initialized and reconfigured when the faults are identified. The indirect filter structure is shown in figure 5.1. In this structure when a GPS observation occurs, innovation formed by subtracting the INS predicted observation from the measured observation. The filter estimate the error in position, velocity and attitude from innovation and knowledge of estimated and observed noise uncertainty, and then the estimated error is applied back INS navigation loop for correction.

The kalman filter is constructed in loosely coupled format as discussed in chapter 4. The mathematical models of kalman filter discussed in chapter 4 will be applied to our navigation problem formed into distinct stages: prediction of the states based on the previous best estimate and updating of the predicted best estimate by combining the prediction with a new measurement.

5.6.1 Process Model

As our objective to implement real time navigation system, thus a discrete form should be obtained from the continuous format obtained in equation 5.58 where the state vector, $\delta\mathbf{x}(t)$, is defined as the error in position, velocity, attitude of INS and

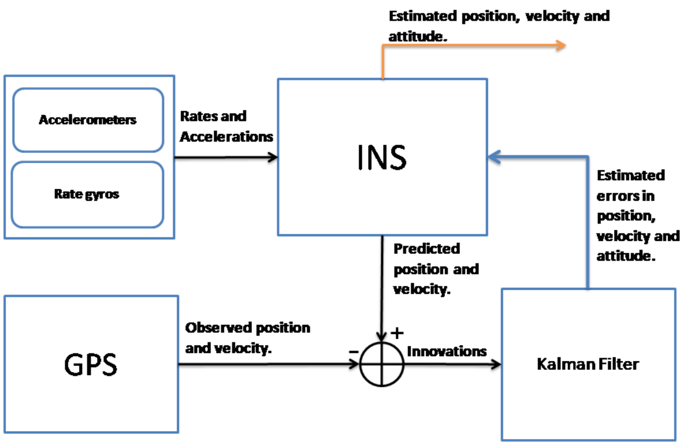

Estimated position, velocity and attitude.
Accelerometers
Rate gyros
Rates and Accelerations
INS
Estimated errors in position, velocity and attitude.
Predicted position and velocity.
GPS
Observed position and velocity.
−
+
Innovations
Kalman Filter

Figure 5.1: Indirect feedback implementation, filter estimates errors in position, velocity and attitude. The inertial block and GPS receiver behaves as a separate navigation systems, the filter can be decoupled from the main loop and can operate in a complementary fashion.

error in biases of accelerometers and rate-gyros:

$$\delta\mathbf{x}(k) = \begin{pmatrix} \delta\dot{\mathbf{P}}(\mathbf{k}) & \delta\dot{\mathbf{V}}(k) & \delta\dot{\mathbf{\Psi}}(\mathbf{k}) \end{pmatrix}^T \tag{5.70}$$

By using the same discretization procedure developed in equation 4.26, considering up to the second term, the discrete state transition matrix F(k) can be obtained as follows:

$$\mathbf{F}(k) = \mathbf{I} + \triangle t\mathbf{F} \tag{5.71}$$

Then the discrete state dynamic model from the continuous INS error equations developed in 5.29 is given by:

$$\delta\mathbf{x}(k) = \mathbf{F}(k)\delta\mathbf{x}(k-1) + \mathbf{G}(k)\mathbf{w}(k) \tag{5.72}$$

where F(k) is the system transition matrix, G(k) is the system noise input matrix and w(k) is the system noise vector with noise strength Q(k).

Recalling the error equations for position, velocity and attitude developed earillier, if the INS error equation in continuous form is represented as below:

$$\begin{pmatrix} \delta\mathbf{v}^n(k) \\ \delta\mathbf{p}^n(k) \\ \delta\psi(k) \end{pmatrix} = \begin{pmatrix} \mathbf{I} & \triangle t\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}-2\triangle t[\omega_{in}^n\times] & \triangle t[\mathbf{f}^n\times] \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \delta\mathbf{v}^n(k-1) \\ \delta\mathbf{p}^n(k-1) \\ \delta\psi(k-1) \end{pmatrix} \tag{5.73}$$

With the noise input vector $w(k) = [\delta f^b(k), \delta\omega^b(k)]^T$. And the discrete noise matrix is computed $Q_d(k)$ as follows:

$$\mathbf{Q}_d(k) = \mathbf{G}(k)\mathbf{Q}'(k)\mathbf{G}^T(k) \tag{5.74}$$

Where

$$\mathbf{G}(k) = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{C}_b^n(k) & \mathbf{0} \\ \mathbf{0} & -\mathbf{C}_b^n(k) \end{pmatrix} \tag{5.75}$$

$$\mathbf{Q}'(k) = \begin{pmatrix} \Delta t\sigma_{f^b}^2 & \mathbf{0} \\ 0 & \Delta t\sigma_{\omega^b}^2 \end{pmatrix} \tag{5.76}$$

5.6.2 Observation Model

We have defined the innovation by subtracting the measured observation from the INS predicted observation i.e. the observed error $\delta\mathbf{z}(k)$ which is defined as well as the filter input and given in discrete time as follows:

$$\delta\mathbf{z}(k) = \mathbf{H}(k)\delta\mathbf{x}(k) + \mathbf{v}(k) \tag{5.77}$$

Where $\mathbf{v}(k)$ is the observed noise (Position and Velocity)with observation error covariance matrix $\mathbf{R}(k)$, the $\delta\mathbf{z}(k)$ term is calculated for the position and velocity by subtracting the estimated INS from the GPS observation:

$$\delta\mathbf{z}(k) = \begin{pmatrix} \mathbf{P}(k) \\ \mathbf{V}(k) \end{pmatrix}_{GPS} - \begin{pmatrix} \mathbf{P}(k) \\ \mathbf{V}(k) \end{pmatrix}_{INS} \tag{5.78}$$

Considering that the estimated INS and the observed GPS measurements are noisy, thus the above equation can be represented as follows:

$$\delta\mathbf{z}(k) = \begin{pmatrix} \mathbf{P}(k) + \upsilon_p(k) \\ \mathbf{V}(k) + \upsilon_v(k) \end{pmatrix}_{GPS} - \begin{pmatrix} \mathbf{P}(k) - \delta\mathbf{P}(k) \\ \mathbf{V}(k) - \delta\mathbf{V}(k) \end{pmatrix}_{INS} \tag{5.79}$$

The above equation can be re-written as follows:

$$\delta\mathbf{z}(k) = \mathbf{H}(k) \begin{pmatrix} \delta\mathbf{P}(k) \\ \delta\mathbf{V}(k) \\ \delta\psi(k) \end{pmatrix} + \begin{pmatrix} \upsilon_p(k) \\ \upsilon_v(k) \end{pmatrix} \tag{5.80}$$

The $\mathbf{H}(k)$ is the observation error matrix and equals to:

$$\mathbf{H}(k) = \begin{pmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \end{pmatrix} \tag{5.81}$$

The Kalman filter algorithm operates recursively in three stages:

5.6.3 Prediction stage

The algorithm starts by generating a prediction of the error state $\delta\mathbf{x}(k|k-1)$ and covariance matrix (The prediction uncertainty) $\mathbf{P}(k|k-1)$ at time $t(k)$ is given by:

$$\delta\mathbf{x}(k|k-1) = \mathbf{F}(k)\delta\mathbf{x}(k-1|k-1) \tag{5.82}$$

$$\mathbf{P}(k|k-1) = \mathbf{F}(k)\mathbf{P}(k-1|k-1)\mathbf{F}(k) + \mathbf{Q}_d(k) \tag{5.83}$$

Assuming zero mean Gaussian noise, thus the predicted error state $\delta\mathbf{x}(k|k-1)$ equals zero therefore the error state will not be propagated in the prediction cycle.

5.6.4 Prediction update stage

The below set of equations are involved with the updating of the predicted best estimate by combining the prediction with the new coming measurement received from the aiding device (GPS), hence, the best estimate of the error state at time $t(k)$ is given by:

$$\delta\mathbf{x}(k|k) = \delta\mathbf{x}(k|k-1) + \mathbf{W}(k)\upsilon(k) \tag{5.84}$$

Where $W(k)$ is the Kalman gain and is given by equation 4.31 and $\upsilon(k)$ is the innovation vector that is described by equation 4.32 and 4.33. However since the error state prediction $\delta\mathbf{x}(k|k-1)$ is always set to zero, then the error state prediction is simply is the weighted sum of innovation (observations) as follows:

$$\begin{aligned}\upsilon(k) &= \delta\mathbf{z}(k) - \mathbf{H}\delta\mathbf{x}(k|k-1)\\ &= \delta\mathbf{z}(k)\end{aligned} \tag{5.85}$$

And in the same way the update prediction uncertainty represented by covariance matrix is updated $\mathbf{P}(k|k)$ which is given by equation 4.34.

5.6.5 Error Correction stage

When the prediction of the best estimate is updated with the observation, then these estimated errors are used to correct the position and velocity as follows:

$$\mathbf{p}(k|k) = \delta\mathbf{p}(k|k) + \mathbf{p}(k|k-1) \tag{5.86}$$

$$\mathbf{v}(k|k) = \delta\mathbf{v}(k|k) + \mathbf{v}(k|k-1) \tag{5.87}$$

Where the hat symbol denotes an estimated value and $\mathbf{p}k, \mathbf{v}k$ are the corrected position and velocity, respectively. However, the attitude can be corrected by updating the direction cosine matrix DCM , when the errors in the attitudes are estimated $\delta\boldsymbol{\Psi}(k)$ then they are used to update the DCM as follows:

$$\mathbf{C}_b^n(k|k) = (\mathbf{I}_{3\times 3}) + [\delta\boldsymbol{\Psi}(k)\times]^{-1}\hat{\mathbf{C}}_b^n(k|k-1) \tag{5.88}$$

Where $\delta\boldsymbol{\Psi}(k)\times$ is the skew matrix. The updated DCM will correct the attitude angles in the next navigation loop.

5.7 Filter Initialization

Normally the initial values of the state and state covariance matrix diminishes with time and do not critically affect the filter's steady state performance, i.e. the filter will converge to the same steady-state values regardless of the chosen initial values, on the other hand the reasonable good choice of the initialization will improve the filter convergence. The initialization is the first step in Kalman filter algorithm, in which an initial values should be provided for the state and its correspondence covariance matrix $\mathbf{X}(0|0)$ and $\mathbf{P}(0|0)$.

The simplest way to provide a reasonable initial 'guess' is by choosing a constant value as ς in which $\mathbf{P}(0|0)$ equals to:

$$\mathbf{P}(0|0) = \varsigma^2\mathbf{Q}(k) \tag{5.89}$$

Where in real time applications $\varsigma = 10$ is a typical reasonable start guess, provided as rule-of-thumb any initial value of chosen covariance matrix $\mathbf{P}(0|0)$ should be large enough in order to encompass any uncertainty in the initial state matrix $\mathbf{X}(0|0$, [39].

5.8 Filter Tuning

Filter tuning could be considered as the most critical issue as the filter steady-state performance is uniquely affected by the noise level values chosen for both process and observation covariance matrices, $\mathbf{Q}(k)$ and $\mathbf{R}(k)$ respectively, as well as the most time and effort consuming procedure compares to other filter real-time implantation processes. As discussed in chapter 4, to achieve an optimal filter solution a prior knowledge of the statistical description of uncertainty of system dynamics, measurement inaccuracies and noise - modelled as process $\mathbf{Q}(k)$ and observation $\mathbf{R}(k)$ covariance matrices, respectively - is required. Consequently, filter tuning process is defined as the appropriate selection of noise values of $\mathbf{Q}(k)$ and $\mathbf{R}(k)$ matrices which will lead to an optimal filtration solution that minimizes the conditional estimated squared error.

In most real time applications the noise levels of both process and observation sensors are unknown or difficult to identify, however as a rule-of-thumb a large $\mathbf{Q}(k)$ will imply highly confidence of the inertial system's sensors, the prediction uncertainty of the INS solution will reach a level in which the correction will happen even if it is the first fix observation regardless of its accuracy, on the other hand small $\mathbf{R}(k)$ implies greater reliance on GPS observation and fusion with inertial solution will happen even if the observation is in error.

A simple procedure is used to tune the filter by implementing the consistency test as an indication of filter performance. The filter consistency test is implemented by checking the following steps of requirements:

- ***Magnitude of the innovation*** to check the noise levels that have been selected for process $\mathbf{Q}(k)$ and observation $\mathbf{R}(k)$ covariance matrices, this test is implemented by direct looking to the plot of innovation sequence in which approx. 95-percent of innovations should fall within a 2σ gate. If the innovations are all well within the 2σ gate, this indicates that the noise levels have been set are too high. Conversely, if too many fall outside of this gate, this indicates that the noise levels set in the filter are generally lower than the true noise levels.

- χ^2 ***test of the Normalized Innovation Square (NIS)*** to check the unbiasdness for the innovation sequence. The innovation sequence should be zero-mean (unbiased) for a consistent filter.

- ***The autocorrelation of innovation sequence (Whiteness test)*** which is implemented by observing both the innovation sequence and its covariance i.e. the innovation sequence should be white noise and shows no periodic-like behavior and that can be checked by plotting the innovation sequence with the corresponding standard deviation $\pm\sigma$ (which is the square root of the corresponding diagonal element of $\mathbf{S}_k$) and $\pm 2\sigma$ bounds, at least 95% of the innovation sequence should fall within the $\pm 2\sigma$ bound. [40], [39], [32].

5.9 Data Latency

Latency generally, is defined as a measure of time delay experienced in a system, and the precise definition varies upon on the system and the time being measured. Data latency is a very important issue in real time applications, however in GPS/INS navigation system the use of a GPS receiver which provide an average velocity observation, while the INS provides an instantaneous velocity quantity will enlarge the possibility of velocity error mismatch. Data delay or latency can happen due to different factors that mainly depends of the GPS internal processing time and data transmission as well as the approach used by GPS to compute the velocity whether it is calculated from position solution or from Doppler shift measurement.The importance of velocity latency arise as it also correlated to the attitude error i.e. incorrect observed velocity error can results in a incorrect estimated attitude error [35]. To overcome this problem careful attention should be considered in real-time implementation; one approach is achieved by saving the inertial data and process the estimates at the time that the GPS observation should have occurred, and then propagate the inertial solution through the backlog of data [41]. Another approach is implemented by utilizing the fact that many GPS receivers provide a hardware signal (1 pulse per second) which is aligned with GPS time, thus using an interrupt routine that wakes up at that event and get the INS solution which corresponds to the GPS observation.

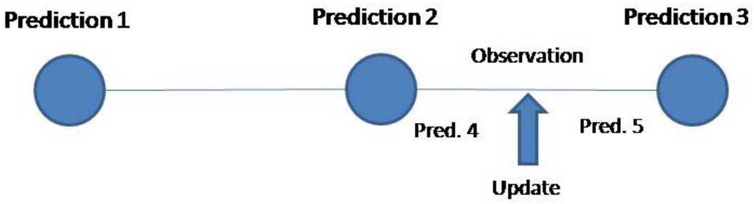


Figure 5.2: Data Latency, the INS data is saved and process the estimates at the time that the GPS observation should have occurred, and then propagate the inertial solution through the backlog of data.(*Nebot, E. M., Navigation System Design*)

5.10 Initial Alignment and Calibration

The most important step the precedes the start of navigation is to determine the initial conditions of state vector $\mathbf{x}(0|0)$ and the system noise vector $\mathbf{w}(0|0)$ which is accelerometers and rate gyros biases. If navigation in local coordinate frame like NED frame is considered then the initial position can be set to zero as a start point, however the initial position could be set to a known value if the start point is geographically known or described by a landmark. Assuming vehicle starts from stationary then initial velocities are sets to zeros. The remaining element of the state vector is the orientation of the vehicle which is important to initialize the INS which is represented by direction cosine matrix $\mathbf{C}_b^n(0|0)$.

The following algorithm describes the method to obtain the initial direction cosine matrix $\mathbf{C}_b^n(0|0)$ that is defined by euler angles $\theta, \beta and \gamma$. $\mathbf{f}^n$ is the specific force given be equation 2.16 measured by a triad of accelerometers and resolved into the local geographic reference frame $\boldsymbol{C}_b^n f^b$, assuming the specific force in the body frame as the true accelerations which can be read from the accelerometers then f^b can be written as follows:

$$\mathbf{f}^b = \mathbf{C}_b^n \mathbf{f}^n \tag{5.90}$$

The alignment is done while the vehicle is in stationary condition therefore the only measured acceleration is due to the gravity which implies that $\mathbf{f}^n$ equals to

$[00g]$. Resolving the $\mathbf{f}^b$ then:

$$f^{bx} = -g\sin(\beta) \tag{5.91}$$

$$f^{by} = g\sin(\theta)\cos(\beta) \tag{5.92}$$

$$f^{bz} = g\cos(\theta)\cos(\beta) \tag{5.93}$$

Arranging the above equations, roll and pitch angles can be determined, the heading of vehicle (yaw angle) can be obtained by using the gyros and knowledge of initial position, however due to the inaccuracies inherited in the acceleromenters and gyros measurements, low resolution and high noise digital compass is used in this work to evaluate the initial orientation angles. A digital compass provides bank, elevation and heading. A positive inclination in elevation and bank angles can cause the accelerometer x-axis and y-axis, to measure the gravity component in x-axis and y-axis respectively:

$$f^{bx} = -g\sin(Elevation) \tag{5.94}$$

$$f^{by} = g\sin(Bank) \tag{5.95}$$

Equating equations 5.91 and 5.92 with the above equations 5.94 and 5.95 respectively:

$$-g\sin(Elevation) = -g\sin(\beta) \tag{5.96}$$

$$g\sin(Bank) = g\sin(\theta)\cos(\beta) \tag{5.97}$$

It follows from the above:

$$\beta = Elevation \tag{5.98}$$

$$\theta = \sin^{-1}(\frac{\sin(Bank)}{\cos(-Elevation)}) \tag{5.99}$$

And the yaw angle equals the heading angle:

$$\gamma = Heading \tag{5.100}$$

The initial direction cosine matrix can be computed with equation 5.1 using the above calculated euler angles that result from solving equations 5.98, 5.99 and 5.100.

The biases of both accelerometers and gyros can be determined by measuring the reading of the sensors while the vehicle in stationary position. Therefore for gyros while vehicle is in stationary position in which velocity is zero thus the measured reading of gyros is noting but biases. For accelerometrs the biases are computed using the following equation:

$$Bias_{sensitive-axis} = Measured(f_{sensitive-axis}) - Expected(f_{sensitive-axis}) \tag{5.101}$$

The expected accelerations $Expected(f_{sensitive-axis})$ can be computed from the above attitude equations 5.91, 5.92 and 5.93.

Chapter 6

Real-time Implementation and Experimental Results

6.1 Introduction

This chapter provides the hardware and software development details and real-time implementation, experimental tests and corresponding results of the INS/GPS navigation system to validate the reliability and effectiveness of the implemented algorithms and performance of all over navigation system design.

An overview of hardware and software development will be presented. This development is mainly accomplished in three stages, the first stage is the design, construction, calibration and error modelling of IMU using commercial of the shelf (COTS) components which is discussed in chapter 3. The second stage is to validate, tune and verify the performance of the developed Kalman Filter software algorithm before to download on Microcontroller for final on-board testing. This step was accomplished by using backlogged real-time data collected at the same time frame, from both equipment: the designed IMU (AUSIMU)/GPS, IMU (MIDG II) from Microbotic, Inc. The collected data was used to compare the results of Microbotic navigation solution with our own designed IMU, and GPS solution. All are constructed on one setup and fixed in the host vehicle.

The final stage is to cross-compile and download the developed Kalaman filter software to the Microcontroller and log the data of the INS/GPS navigation setup for final verification of navigation system performance. The over all INS/GPS navigation setup is shown in figure 6.1.

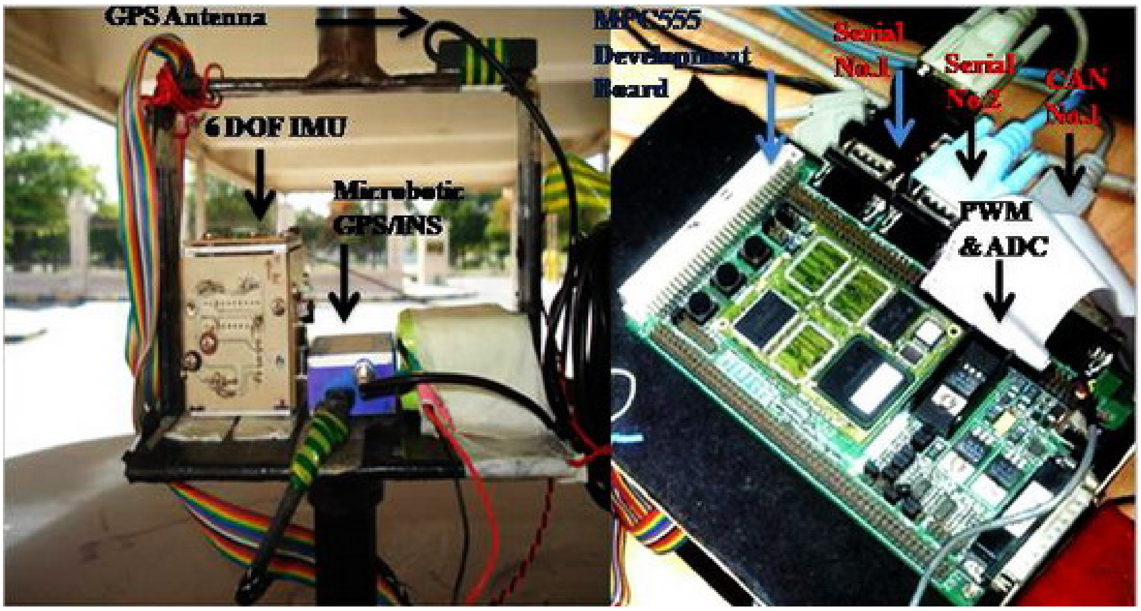


Figure 6.1: Strapdown inertial navigation system setup consist of 6 DOF IMU connected to PWM and Queued Analog to Digital Convertor (QADC) port. GPS is connected to the serial port of MPC555 Microcontroller.

6.2 Hardware Development

6.2.1 Navigation Computer

The $phyCORE - MPC555^{©}$ microcontroller $(72 \times 57)mm$ a product of Freescale Semiconductors, Inc., is chosen as the navigation computer, a view of MPC555 is shown in figure 6.2. And $PHYTEC^{®}$ Development Board $(160 \times 100)mm$ is used to support software development, debugging and programming of the phyCORE-MPC555, the Development Board is fully equipped with all mechanical and electrical components necessary for the speedy and secure insertion and subsequent programming. The MPC555 can be plugged like a "big chip" onto the Development Board and once programmed, it can be removed from the Development Board and inserted in a target hardware application. PhyCORE-MPC555 is featured with 40 MHz CPU speed, a 32-bit processor architecture with a 64-bit Floating-Point-Unit. It is equipped with two serial RS232 port, dual CAN 2.0B, eight channel dedicated

PWM output, 32 analog inputs and BDM (Background Debug Mode) test/debug port which communicates with the host PC via the parallel port as shown in figure 6.3.

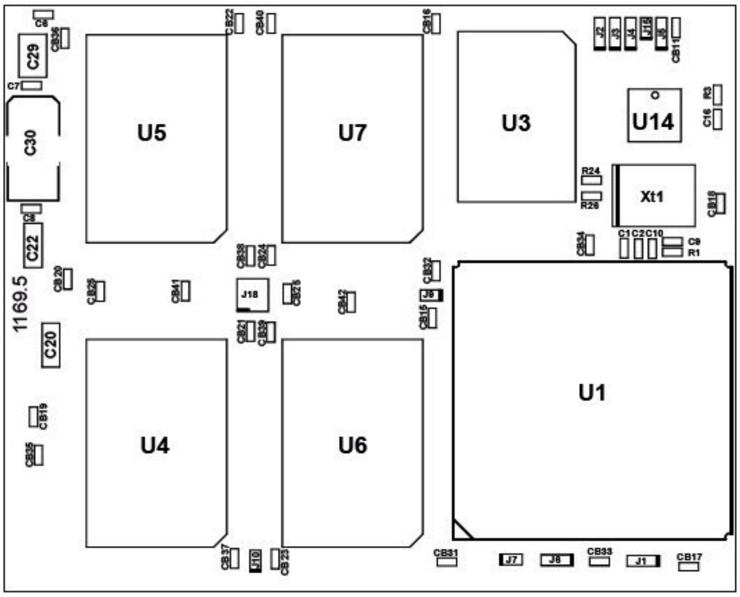


Figure 6.2: The pin layout of $phyCORE - MPC555^{\copyright}$ microcontroller $(72 \times 57)mm.$ *(PHYTEC Technology Holding Company)*

The internal (26 kByte internal SRAM, 448 kByte internal Flash EEPROM) and external Flash (optionally external 512 kByte, 1/ 2/ 4 MB or 8 MB) supports direct on-board programming without additional programming voltages [42].

The serial ports are used to interface the GPS, and the second port will be utilized for direct connection to laptop for data logging in order to be used for further analysis. CAN port will be used to download developed programm onto the microcontrolloer. As well the second CAN port is used to interface the PC through CAN drive card that can be installed, this is used for off-line Kalaman filter results demonstration, evaluation, debugging and testing.

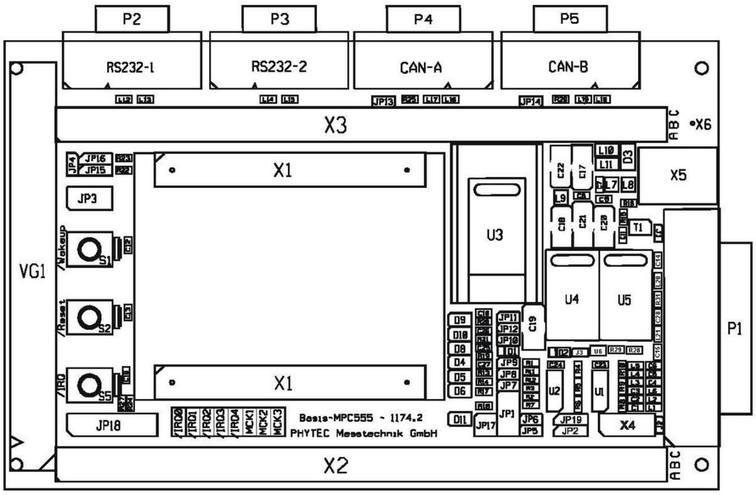


Figure 6.3: Development Board PHYTEC overview (component side). X1 is receptacle to install the phyCORE-MPC555 SBC module. X2 is a 2.54*mm* standard width pin header rows for extension boards, providing all signals of the phyCORE-connector. P1 is DB-25 plug of the on-board BDM interface. *(PHYTEC Technology Holding Company)*

6.2.2 MPC555 Memory Organization

When writing down the algorithm code, it is critical to ensure that the size of the generated code does not exceed the physical limitations of RAM or Flash memory spaces provided by processor. The MPC555 has a total of 448K of on-chip flash EEPROM memory. The flash memory is organized into 14 banks of 32K each as shown in figure 6.4, the first bank is always used to store the boot code and the remaining 416K is available for application code. The boot code is necessarily to be downloaded to flash memory in order to load and start application code when the board is powered-on or reset. The boot code performs basic hardware initialization and then branches to the application code, and also the boot code also acts as a download agent that downloads generated code into flash memory via CAN. It also manages the exception handling for the MPC555 or idle status.

When using the Embedded Target for Motorola MPC555, the on-chip flash memory is located at absolute address 0x0000 in the MPC555 address space [43].

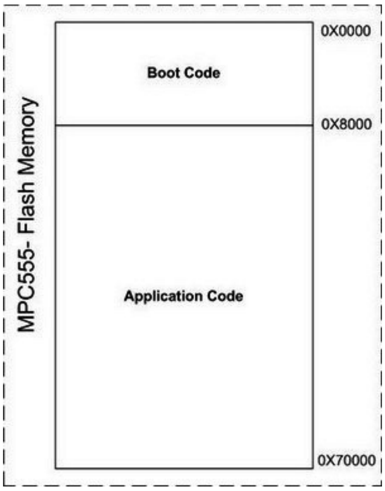


Figure 6.4: Flash memory arrangement in MPC555 microcontroller, it is necessarily to program the boot code into the first bank of flash memory, before starting to download any other application program (bootcode.s19 (for CAN download) or bootcode.elf (for BDM download)).

When using RAM or flash memory there are a few points should be taken into consideration, however generally the final code that will be used in site must be programmed into the nonvolatile "*flash memory*" so it will be run each time the microcontroller is powered on or rested. On the other hand, it is initially preferred for code testing and debugging to download the code into RAM memory as it lengthens the usable lifetime of flash memory. Advantages and disadvantages are list below in table 6.1.

6.2.3 Sensors

The Navigation system is equipped with IMU (AUSIMU) which is developed, designed and calibrated in Mechatronics Lab AUS, GPS, digital compass (HRM3000) and ion Inertial Navigation aided with GPS (MIDG II). The following section provides a brief summary of mentioned sensors.

Table 6.1: Advantages and disadvantages: RAM vs. Flash Memory

Memory Description	Advantages	Disadvantages
RAM	Loading the application code into RAM is faster than loading it into flash memory.	To program applications into RAM, your target hardware must have additional RAM external to the MPC555 on-chip RAM which cannot be archived due to size limitations.
	using RAM helps to avoid using up the programming cycles of the flash memory; this lengthens the usable lifetime of the flash memory.	The Embedded Target for Motorola MPC555 does not support downloading of code to the MPC555 on-chip RAM, because the MPC555 has only 26K of on-chip RAM.
	Running the application from RAM is a preferable option for initial testing and code debugging of the application.	
FLASH	416K of flash memory is available for application code.	Flash memory has a life time which decreases each time the code is downloaded into it.
	Loading the application code into flash memory is persistent and restarts when the board is powered on.	Loading time is higher than loading it to RAM.

6.2.4 IMU (AUSIMU)

The design procedure, calibration analysis and error modelling of IMU (AUSIMU) has been presented in details in chapter 3.

6.2.5 Digital Compass

The digital compass Honeywell Inc., HMR3000 shown in figure 6.5, the purpose of using the digital compass is to get the heading, roll and pitch angles which is necessary for navigation process initialization as discussed in chapter 5. This compass has an accuracy of 0.5° with 0.1° resolution for critical positioning applications. The tilt range of this compass is 40° for roll and pitch, and 0° to 360° for heading. It can be interfaced to the MPC555 via the RS232 port.

Figure 6.5: Digital compass Honeywell Inc., HMR3000

6.2.6 GPS

The MIDG II, Microbotic, Inc., is an inertial navigation system (INS) aided with GPS. The device is equipped with An internal GPS receiver which measures position and velocity and passes it to the data fusion processor to be combined with the inertial data to generate an optimal solution. The MIDG II can operate in three distinct modes of operation: IMU, VG (vertical gyro), and INS. The modes are mutually exclusive and the active mode is determined based on user configuration and internal operating criteria. What concerns our work is IMU mode which represents the most basic operation,in this mode, the MIDG II provides calibrated values for angular rate, acceleration. Measurements from the GPS receiver are also available. However, none of the position/velocity/attitude estimation algorithms are executed. As a result, attitude is not available, and position and velocity are available directly from the GPS receiver at up to 5Hz. Thus, GPS raw data will be used as our source of GPS's position and velocity observations which will be fused with inertial estimated ones, table 6.2 summarizes the technical specification of MIDG II.

Table 6.2: MIDG II technical specification

Item description	
Physical	
Dimension	$(3.81 \times 2.057 \times 4.382)cm$
Weight	55 grams
Power Requirement	
Input Voltage	(10-32) VDC
Power	1.2 watt max
Output	
Electrical	RS422 async., 115200 baud (configurable), 8-N-1
Data Format	Microbotics Binary Protocol

However the second helpful mode is the INS mode as it will be compared with our developed GPS/INS solution. In INS mode, the MIDG II provides estimates of position, velocity, and attitude at up to 50Hz using a state estimation filter. In order to enter INS mode, the MIDG II must be configured for INS operation (which is the default) and GPS must be available. MIDG II is shown in figure 6.6 Moreover MIDG II is provided with the a Graphical User Interface (GUI) Program, the MIDG II Display and Configuration Program which facilitates immediate access to the its data for evaluation and testing without requiring the user to write software, the Display utility program window is shown in figure 6.7.The start MIDG II Display and Configuration Program screen is the Nav/GPS screen, which is divided into seven areas, the first one is consisting of an Artificial Horizon and six Message Windows. Each message window corresponds to a message from the MIDG II, and the second screen is the IMU raw measurement data [44]. MIDG II can be connected to a PC or Laptop via

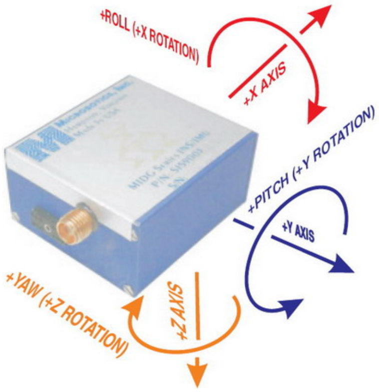
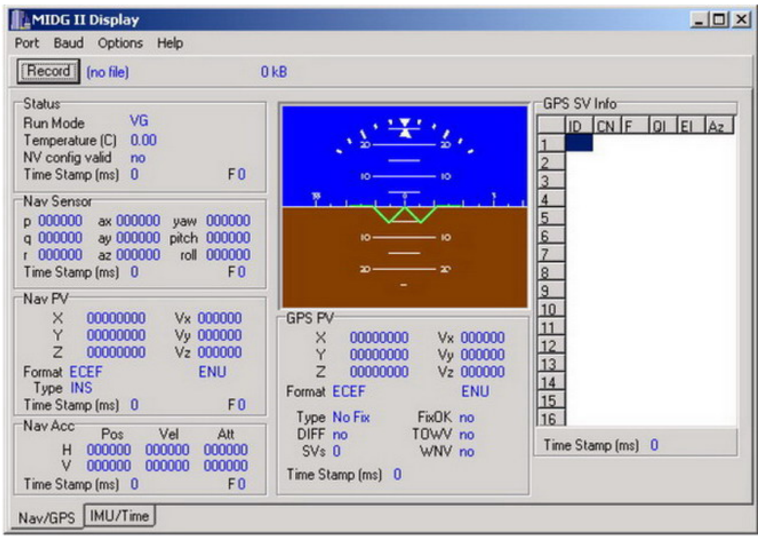


Figure 6.6: Inertial navigation system (INS) aided with GPS (MIDG II), the equipment shown with its sensors sensitive axises *(Microbotic, Inc.)*



(no file) 0 kB

```
Status
Run Mode          VG
Temperature (C)   0.00
NV config valid   no
Time Stamp (ms)   0              F 0

Nav Sensor
p 000000   ax 000000   yaw   000000
q 000000   ay 000000   pitch 000000
r 000000   az 000000   roll  000000
Time Stamp (ms)   0              F 0

Nav PV
    X   00000000     Vx 000000
    Y   00000000     Vy 000000
    Z   00000000     Vz 000000
Format ECEF             ENU
  Type INS
Time Stamp (ms)   0              F 0

Nav Acc
        Pos       Vel       Att
    H   000000    000000    000000
    V   000000    000000    000000
Time Stamp (ms)   0              F 0

GPS PV
    X   00000000     Vx 000000
    Y   00000000     Vy 000000
    Z   00000000     Vz 000000
Format ECEF             ENU
  Type No Fix        FixOK no
  DIFF no            TOWV  no
  SVs 0              WNV   no
Time Stamp (ms)   0
```

GPS SV Info

	ID	CN	F	QI	El	Az
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

Time Stamp (ms) 0

Figure 6.7: MIDG II Display and Configuration Program. All MIDG II messages and configuration options are supported by the program.*(Microbotic, Inc.)*

RS232 as shown in the below figure 6.8. MIDG II communication occurs over the RS232 communication port using the Microbotics binary protocol shown below in figure 6.9. Each message consists of two synchronization bytes, one bye for ID i.e. each message will have a unique ID. Fourth byte is the Count byte, i.e. if the value of Count is N then there are N data bytes which is called here the payload bytes. The payload is composed of a sequence of bytes that represent values within a message.

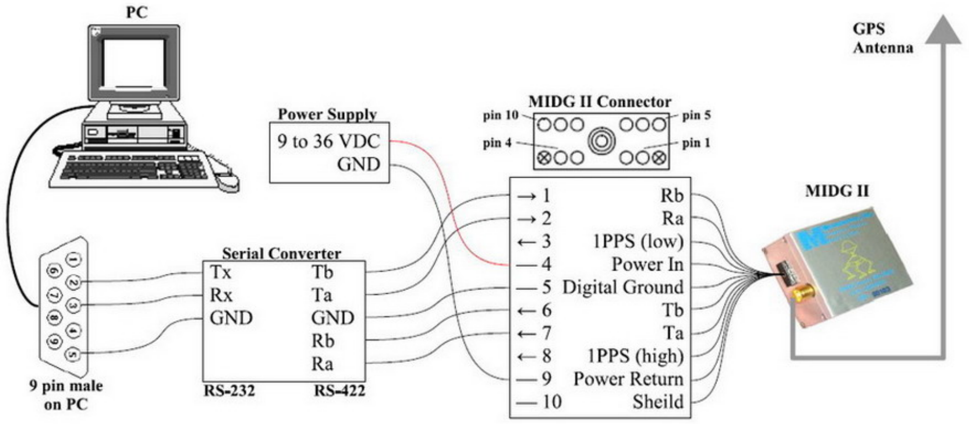


Figure 6.8: MIDG II typical connection to a PC via RS232 port. A cable with appropriate connectors and an RS-422 to RS-232 Converter (such as the Microbotics' SLC22232), to connect the MIDG II to the PC. *(Microbotic, Inc.)*

At the end of the message there are two byte called checksum namely CKSUM0 and CKSUM1. The checksum is a Fletcher checksum as defined in internet RFC 1145. It is computed over the bytes between the head and checksum, i.e. it includes the message ID, Count byte, and the payload bytes [44]. Basically the checksum can be computed as follows:

$$\begin{aligned} cksum0 &= 0, < -unsignedcharacter \\ cksum1 &= 0, < -unsignedcharacter \end{aligned} \tag{6.1}$$

for each byte from ID to $Payload_N$ (inclusive):

$$\begin{aligned} cksum0 &= cksum0 + byte < -only8bitspreserved \\ cksum1 &= cksum1 + cksum0 < -only8bitspreserved \end{aligned} \tag{6.2}$$

The same software program is used as the interface environment with our developed GPS/INS navigation system. Once the state vector estimate are updated, then output data of Kalman Filter namely (*position, velocity and orientation angles*) are arranged in a packet frame similar to MIDG II binary protocol format; this has been done to ease our apple to apple comparison and analysis with the results of the origin devise.

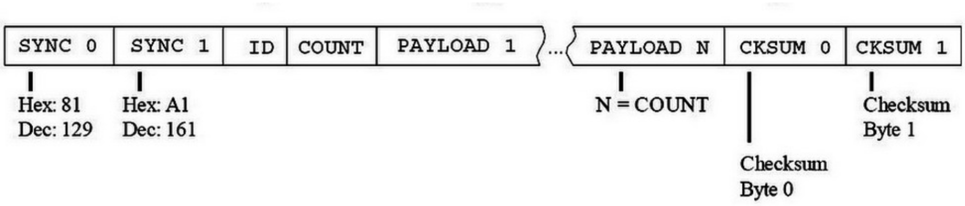


Figure 6.9: Microbotics binary protocol, herein referred to as mBin. The mBin protocol is a standard binary packet format that has the shown structure. *(Microbotic, Inc.)*

The following table 6.3 shows the measurements ranges of the MIDG II:

Table 6.3: MIDG II measurement specification

Measurement	
Data Rate Output	Position, velocity and attitude: 50Hz Raw GPS measurements: 5Hz.
Angular Rate Range Non-Linearity	 $\pm 300 \circ /sec$ VDC $0.1 percentofFS$
Acceleration Rate Range Non-Linearity	 $\pm 6g$ $0.3 percentofFS$

6.3 Software Environment Development

The real-time INS/GPS algorithm programm is developed on the PC desktop and cross-compiled and generated code download to the target navigation computer, hence this process accomplished in two stages:

- Software build-up, development, simulation and performance evaluation using Real Time Workshop embedded with *Simulink*® and *Matlab*®.

- Real-time execution, compile, download, run and debug generated code on the target navigation computer hardware using Embedded Target for $Freescale^{TM}$ MPC5xx in conjunction with Real-Time $Workshop$® and *Simulink*®.

6.3.1 Embedded Target for Freescale MPC5xx

Target for Freescale MPC5xx is a product which is used in conjunction with the Real-Time Workshop, Simulink and Matlab. It provides a complete and unified set of tools for developing embedded applications for the MPC555 microcontrollers. The features of using Target for MPC5xx is as follows:

- It allows to design and model the algorithm, compile, run and download the generated code via CAN or BDM ports.
- The Target MPC5xx generates a real-time, stand-alone code for MPC5xx.
- It provides a rapid prototyping techniques to evaluate performance and validate results.
- The Target MPC5xx Library provides device driver blocks that allows code applications to access on-chip I/O resources; i.e. Pulse width modulation (PWM),Queued Analog-to-Digital Converter (QADC64), Transmit or receive Controller Area Network (CAN) and Serial transmit and receive RS232 etc.

When program is completely built the Real-Time Workshop generate a code file in Motorola S-record format (*.s19), which is suitable for downloading and execution in RAM or FLash memory, this executable file is the one which will be downloaded to the target microcontroller, this file is for CAN download. Real-Time Workshop can generate the program in *.elf format, this file is for BDM download. This format contains debugging symbols which are suitable for use with a symbolic debugger such as Metrowerks CodeWarrior.

6.3.2 Navigation Algorithm

The navigation algorithm program is mainly divided into three parts; the first part is interface and communication with sensors which comprises of receiving data from IMU and GPS sensors. The AD and PWM blocks are used to receive the IMU data then pass them to another block to average and model the received data and converted them into acceleration and angular rates using the error models developed in chapter 3. Once the data are ready a median filter is used to eliminate the extremes. The GPS data are received through RS232 port, buffered and an S-function is to extract the required data i.e. position, velocity and GPS-time from the received packet frame message which is consist normally of other information. The second part is the Kalman Filter which is the core of the algorithm, the kalman filter code is developed using Embedded Function as it allows to add MATLAB functions to Simulink models for deployment to embedded processors. Embedded Function works with a subset of the MATLAB language called the Embedded MATLAB subset, which provides optimizations for generating efficient, production-quality C code for embedded applications. Third part is preparing the Kalman filter outputs in a packet frame message that can be viewed by the GUI software program installed in the host PC or laptop, the data are transmitted through the RS232 port. Figure 6.17 shows different stages of navigation algorithm interfaces peripherals and data flow of system.

The navigation algorithm flow is shown in figure 6.16 and summarized as follows:

1. (*Reading and Averaging*) When the Navigation Computer (MPC555 microcontroller) is powered on and while the hosted vehicle in a stationary position, the software read the IMU (Accelerometrs and rate-gyros), Digital Compass and GPS data, average, model, extract and filter the readings. In real-time application the vehicle should stay immovable for certain time in order to allow the GPS to warm-up and the reading to be available with its most accurate solution accepted numbers of satellites. Experimentally with the use of MIDG-II this

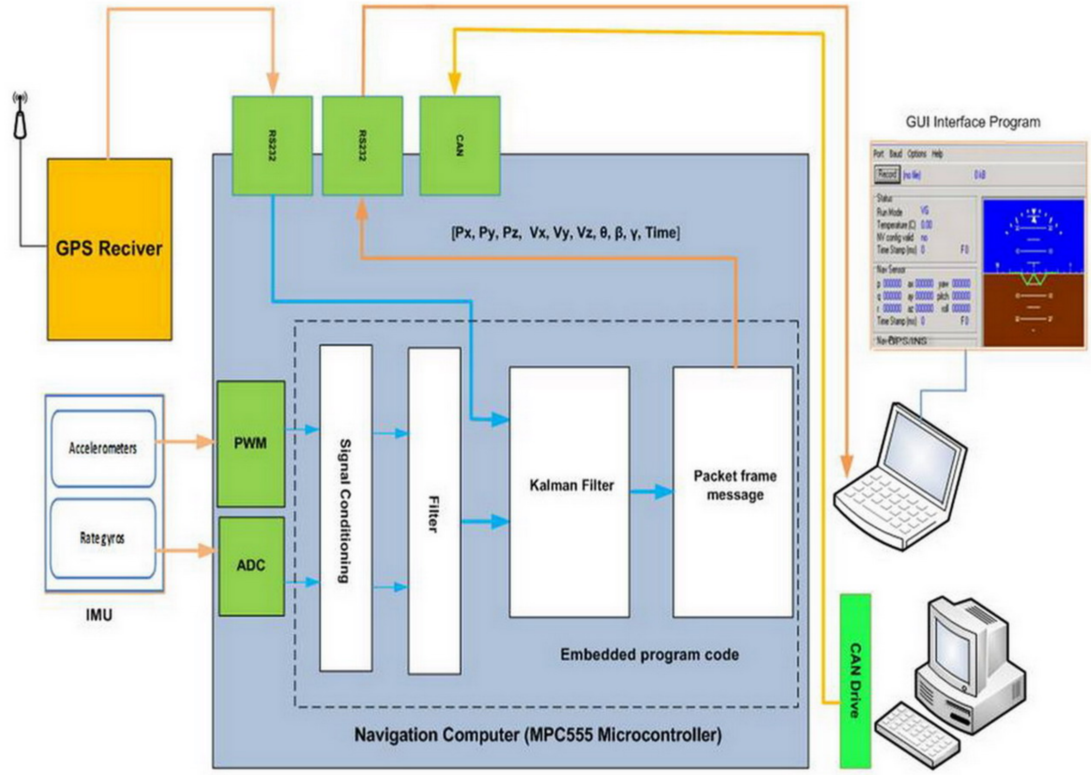


Figure 6.10: Navigation algorithm data flow which consists of different data receiving blocks, signal conditioning, filtration, Kalaman filter, data packaging and transmission. The algorithm blocks are built in Simulink in conjunction with Real-time Workshop using S-functions and Matlab embedded functions. The program is developed on the PC desktop and cross-compiled and generated code download to the target microcontroller via CAN port.

time is about 5 seconds.

2. (*Calibration and Initialization*) Once the readings are ready calibration is accomplished, initial position is determined by GPS, initial velocity is acquired. Initial position and velocity will be transformed into the different coordinate frames. Initial orientation angles are read from digital compass to initialize the direction cosine matrix $\mathbf{C}_b^n(0|0)$, so the initial attitude is determined. Kalman filter matrices; initial state vector $\mathbf{x}(0|0)$, error state covariance $\mathbf{P}(0|0)$ are initialized.

3. As the vehicle starts to move the acceleration and rates values are read in and the biases removed accordingly, $\mathbf{C}_b^n(0|0)$ is updated, and the acceleration in the navigation frame is computed, these values are then integrated to provide the position and heading of the vehicle.

4. The algorithm checks for GPS observations, if no GPS readings are validate then no correction will happen, return back to step three and system will continue navigation based on INS mechanization only. On the other hand if a GPS fix is available it will be fused with the inertial estimate.

5. The corrected INS/GPS solution is arranged in a packet frame massage and transmitted via the RS232 to be viewed and logged by the GUI software in host PC or laptop.

6. The discretization time Δt algorithm is the sampling time of the inertial unit.

This structure and algorithms have been successful in providing the sole means of navigation. Experimental results of the INS/GPS implementation are presented in the next section.

6.4 Real-time Experimental Methodology and Results

The implementation of real-time INS/GPS navigation system is accomplished in two main stages:

1. The first stage aims to verify the performance, reliability of developed INS/GPS unit in site by collecting real-time data from both designed IMU and GPS and a known IMU like the one constructed in Microbotic MIDG II navigation system which is a GPS aided inertial navigation system (INS) using (*MPC555 microcintroller*), then the collected data from both systems; i.e. raw measurements from our designed IMU (*accelerations and angular rates*), GPS (*Position,*

Velocity and time) and raw data; is fused off-line, integrated by Kalman Filter algorithm on a desktop PC and results are compared with the Microbotic MIDG II navigation solution. And the second important purpose is to use the collected data to tune the Kalamn filter, and select the proper noise level values of $\mathbf{Q}(k)$ and $\mathbf{R}(k)$ matrices.

2. Once the performance, functionality of the designed IMU is verified and Kalman filter is checked, code debugged and filter parameters are tuned, then the Kalamn filter generated application code is downloaded into the target *MPC555 microcintroller* and a real-time test is conducted and INS/GPS navigation solution results are collected to be statistically analyzed i.e. by implementing the consistency test as an indication of filter performance as discussed in chapter 5.

6.4.1 IMU Verification and Tuning Test

The aim of this test is to collect real-time data from the designed IMU (AUSIMU), a known IMU (Microbotic MIDG II) and GPS at the same time, using one microcontroller. All devices are constructed on the same setup and the setup is fixed on the roof of a land vehicle to minimizes any chances of GPS signal blockage. The AUSIMU is connected to the microcontroller via PWM to read the accelerometers signals and ADC ports to get the rate-gyros outputs. However the Microbotic MIDG II can be configured so that the output messages will contain the raw measurements (accelerations and angular rates) of its IMU and one another message will contain the GPS raw data (position, velocity) and time. The Microbotic MIDG II is connected to serial port (RS232) of MPC555 microcontroller and the microcontroller transmit the data frames to a laptop through the second serial port (RS232).

The program algorithm for this test is divided into three parts as shown in figure 6.11:

- The first part is to configure the selected ports to be able to read the sensors and MIDG II outputs, filter noisy readings and convert whatever readings to the

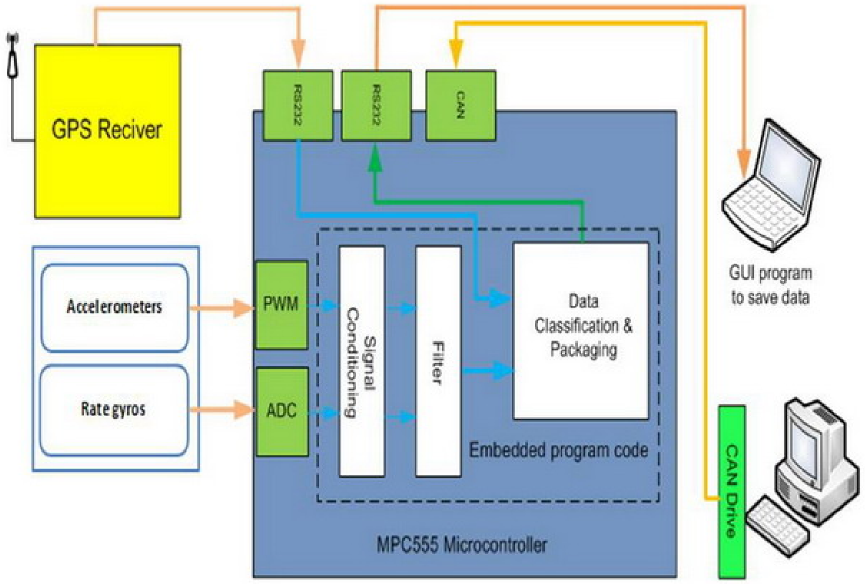
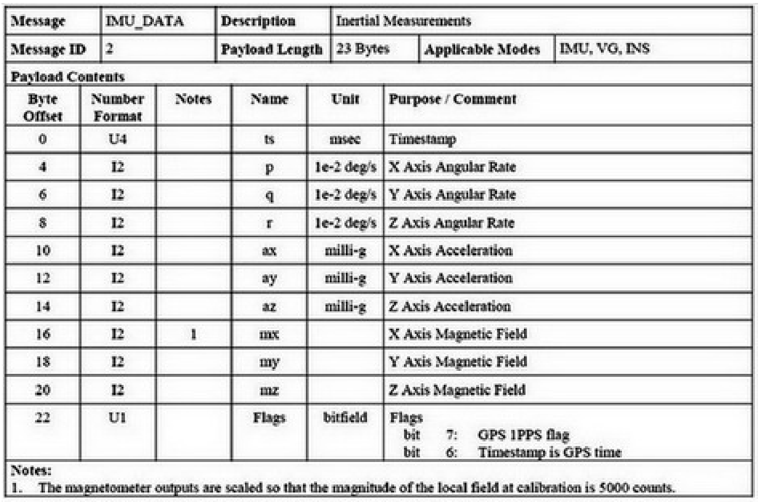


Figure 6.11: The data flow of algorithm used to verify the performance, reliability of developed INS/GPS unit. The CAN port is used to program the MPC555.

proper required quantities as the case with the AUSIMU, for instance the output of accelerometers is a pulse modulation signal and the outputs of the rate-gyros is analog voltages which need in both cases to be converted to accelerations and angular rates using the error model developed in chapter 3.

- Extract the needed GPS measurements, accelerations and rates measurements from the MIDG II. The MIDG II transmits the data within packet frames, each message of data distinguished by a unique ID. This part of code is responsible for checking whether the received message contents are valid, correct and do not contain noise. This is accomplished by verifying the heading bytes of the message SYNC0 and SYNC1, then to check the checksum bytes by reverse the algorithm steps 6.1 and 6.2. In our case two messages are configured namely message $ID = 2$ and message $ID = 20$. Message contents are shown in figure 6.12 and 6.13. Finally the data collected from AUSIMU is converted to binary format and prepared in one packet frame with an ID in a reverse manner of



Message	IMU_DATA	**Description**	Inertial Measurements		
Message ID	2	**Payload Length**	23 Bytes	**Applicable Modes**	IMU, VG, INS
Payload Contents					
Byte Offset	**Number Format**	**Notes**	**Name**	**Unit**	**Purpose / Comment**
0	U4		ts	msec	Timestamp
4	I2		p	1e-2 deg/s	X Axis Angular Rate
6	I2		q	1e-2 deg/s	Y Axis Angular Rate
8	I2		r	1e-2 deg/s	Z Axis Angular Rate
10	I2		ax	milli-g	X Axis Acceleration
12	I2		ay	milli-g	Y Axis Acceleration
14	I2		az	milli-g	Z Axis Acceleration
16	I2	1	mx		X Axis Magnetic Field
18	I2		my		Y Axis Magnetic Field
20	I2		mz		Z Axis Magnetic Field
22	U1		**Flags**	bitfield	Flags bit 7: GPS 1PPS flag bit 6: Timestamp is GPS time

Notes:
1. The magnetometer outputs are scaled so that the magnitude of the local field at calibration is 5000 counts.

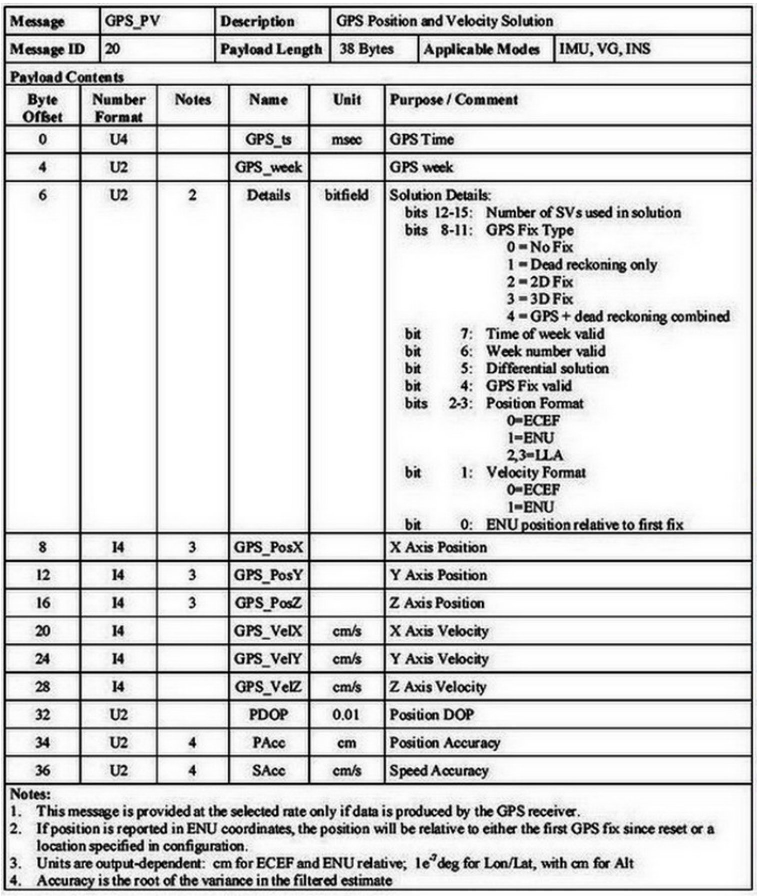
Figure 6.12: MIDG II Message $ID = 2$ which mainly consist of IMU raw data accelerations ax, ay and az and rates p, q and r. Total length of message is 29 bytes.I2 indicates signed , 16 bit integer and U4 indicates unsigned, 32 bit integer

what the MIDG II generates so as the MIDG II interface software can view and log the data.

- The last part of code is to configure the second serial port (RS232) of MPC555 to re-transmit the data frame message to the host Laptop or PC.

This program algorithm code is built with the use of Simulink in conjunction with Real-Time Workshop is shown in figure 6.15. Once the the collected data is saved, then it can be used to run the kalman filter algorithm. For this purpose the algorithm transferred to m file in Matlab and the data is simulated and results are analyzed by implementing the consistency test as an indication of filter performance as discussed in chapter 5 which is achieved by selecting the proper noise level values of **Q** and **R** matrices as shown in figure 6.14.

When the innovation sequence is unbiasdness, white noise and shows no periodic-like behavior satisfying our requirements then the code is updated with the final achieved **Q** and **R** matrices, then application code is downloaded to target MPC555 to be then tested on site for final performance and functionality verification test.



Message	GPS_PV	**Description**	GPS Position and Velocity Solution		
Message ID	20	**Payload Length**	38 Bytes	**Applicable Modes**	IMU, VG, INS
Payload Contents					
Byte Offset	**Number Format**	**Notes**	**Name**	**Unit**	**Purpose / Comment**
0	U4		GPS_ts	msec	GPS Time
4	U2		GPS_week		GPS week
6	U2	2	Details	bitfield	Solution Details: bits 12-15: Number of SVs used in solution bits 8-11: GPS Fix Type 0 = No Fix 1 = Dead reckoning only 2 = 2D Fix 3 = 3D Fix 4 = GPS + dead reckoning combined bit 7: Time of week valid bit 6: Week number valid bit 5: Differential solution bit 4: GPS Fix valid bits 2-3: Position Format 0=ECEF 1=ENU 2,3=LLA bit 1: Velocity Format 0=ECEF 1=ENU bit 0: ENU position relative to first fix
8	I4	3	GPS_PosX		X Axis Position
12	I4	3	GPS_PosY		Y Axis Position
16	I4	3	GPS_PosZ		Z Axis Position
20	I4		GPS_VelX	cm/s	X Axis Velocity
24	I4		GPS_VelY	cm/s	Y Axis Velocity
28	I4		GPS_VelZ	cm/s	Z Axis Velocity
32	U2		PDOP	0.01	Position DOP
34	U2	4	PAcc	cm	Position Accuracy
36	U2	4	SAcc	cm/s	Speed Accuracy

Notes:

1. This message is provided at the selected rate only if data is produced by the GPS receiver.
2. If position is reported in ENU coordinates, the position will be relative to either the first GPS fix since reset or a location specified in configuration.
3. Units are output-dependent: cm for ECEF and ENU relative; 1e^{-7}deg for Lon/Lat, with cm for Alt
4. Accuracy is the root of the variance in the filtered estimate

Figure 6.13: MIDG II Message $ID = 20$ which mainly consist of GPS raw measurements Position, velocity and time. Position and velocity are available in ECEF or ENU format. Total length of message is 44 bytes.

6.4.2 Real-Time Test

Once the MPC555 is receiving, processing and transmitting the data as designed, and the kalman filter is tuned by choosing the proper values of $\mathbf{Q}$ and $\mathbf{R}$, then the application code is downloaded to the target MPC555 to be tested in site. The below figure 6.17 shows the algorithm data flow of INS/GPS navigation system. As mentioned previously the INS/GPS navigation setup is fixed on the roof of a ground vehicle as shown below in figure 6.18 in odder to minimizes any chance of GPS signal blockage. The test was conducted in a open area around AUS campus Sharjah-UAE, the conducted test trip took 17.413 minutes. The average velocity of the vehicle was 34.615 km/h

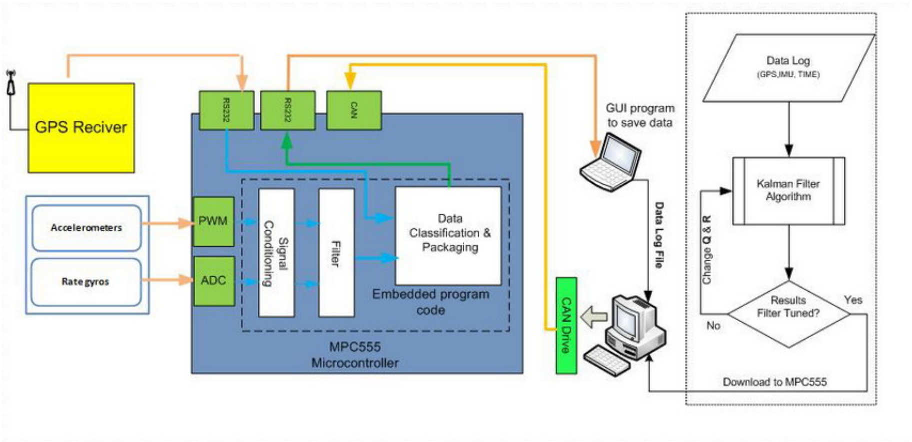


Figure 6.14: The figure shows a flow-chart of the off-line Kalman filter tuning procedure that is implemented using the logged data saved during a ground test conducted around the university campus. Kalman filter algorithm code is running on the desktop PC using Matlab. During the test the setup was fixed on the top of ground vehicle.

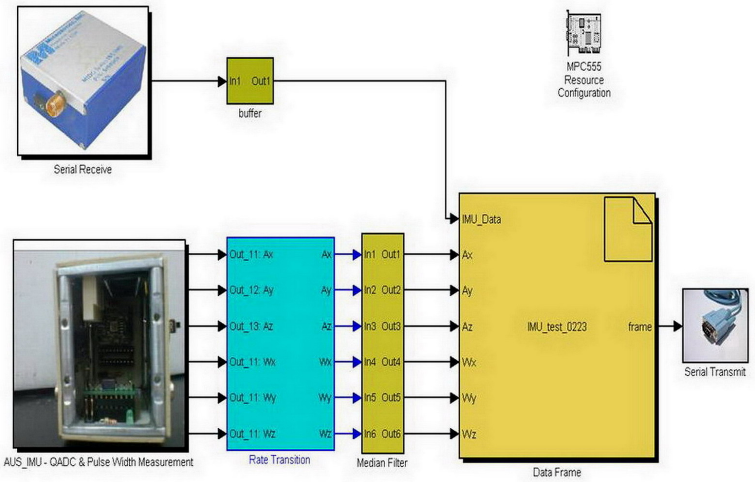


Figure 6.15: Real-Time data collection programm using MPC555. The program is built in Simulink in conjunction with Real-Time Workshop. The generated code is downloaded to the target MPC555 through CAN drive. The program model consists of I/O blocks, filtration and data frame blocks which is an S-function blocks that can be programmed in C-code.

(9.6153 m/s) where the velocity component in north direction is $V_N = 6.883m/s$ and velocity component in east direction is $V_E = 6.714m/s$. Figure 6.20 and 6.22 shows the IMUAUS three accelerometers and rate-gyros raw measurements, however figure 6.21 and 6.23 shows a zoom in view of the collected measurements: Figure 6.25 shows real-time test trajectory represented by GPS observations and INS/GPS estimations. Figure 6.24 shows the actual trajectory of the test, the trajectory is plotted using Google Earth software. The INS/GPS navigation solution results i.e. position estimates are converted from NED coordinate frame to LLH (Longitude, latitude and altitude) coordinate frame in decimal degrees which can be used to generate a *.kml file that can be viewed on actual scale by Google Earth. Figure 6.26 highlights the performance of the filter during strait and rotational motion, it can be noticed that loosely-coupled linear filter shows inefficient performance at corners, this is due to the nature of the filter where attitude and bias estimates are zeros during prediction, and are only available when there is a GPS observation that correct the estimate, the filter estimates deviates and this becomes significant at corners. Figure 6.27 shows a comparison between the navigation solution of MIDG II and our developed INS/GPS navigation system. The figure shows that our developed navigation system follows the same trajectory of the MIDG II except at corners where the navigation solution deviates and that is due to the nature of the filter as discussed before. Figure 6.28 and 6.29 shows a further zoom in view, the figure shows the developed INS/GPS position along with the MIDG II navigation solution in addition to the GPS observation.

For a comprehensive analysis the position and velocity innovations with its 2σ uncertainty for the developed INS/GPS navigation system are shown in figure 6.30 and 6.31, respectively. However the position and velocity innovations with its 2σ uncertainty for the MIDG II is shown in figure 6.32 and 6.33, respectively.

6.5 Conclusion and Future Work

6.5.1 Summary of Conclusions

The Inertial system/GPS was successfully implemented with an accurate and reliable performance. This thesis has provided a mean of Real-time GPS data aided INS using COTS components. In doing so, the thesis has discussed principles of inertial navigation. It has derived the equations required for the local geographic (NED) frame mechanizations as the implementation in the NED coordinates is more physically intuitive than other frames. The direction cosine matrix approach was selected to compute orientation angles and the unified mathematical frame work has been chosen for position/velocity algorithm computations, which resulted into significant reduction in mechanization errors.

This thesis work presented the MEMS-based AUSIMU IMU unit which is developed and constructed during this work research using commercial of-shelf-components. Traditionally, a mechanical platform rotating the IMU into different precisely controlled orientations and turn rates have been used to calibrate IMU's. Then, observing the output and the pre-calculated specific force or angular velocity input acting upon the IMU for different orientations and rotation sequences, respectively, it is straightforward to estimate the misalignment, scaling and bias [10], [18] and [19]. The cost of such a platform often exceeds the cost of developing and constructing an IMU. The thesis discussed the IMU error modelling and calibration procedure, introducing a handy and desirable calibration procedure where the requirements of a precisely controlled orientation calibration procedure of the IMU can be relaxed and thus cost decreased dramatically.

The thesis also discussed the INS/GPS integration and data fusion as the best method to achieve an accurate and reliable navigation solution when low-cost and degraded strap down inertial sensors are used. The Kalman filter was introduced as the best optimal recursive data processing algorithm, and optimal in the sense that it incorporates all available data information that can be provided with in order to estimate the current value of the variable of interest, and that was achieved with the knowledge of system and measurement sensors dynamics, the statistical description of uncertainty of system dynamics, measurement inaccuracies and noise and as well the initial conditions of the desire variable of interest, GPS system and its most parameters, errors and aspects relating and affecting the fusion process has been discussed.

Among different classes of filter integration architectures; loosely coupled integration linear filter was employed because of simplicity, redundancy etc as advantages among other integrations schemes. The single-dimension estimator case was discussed as an introductory to full dimension Kalman filter. The mathematical structure and filter state space representation in discrete form was also introduced with its two mainly algorithm stages the prediction and observation update step. Different real-time implementations issues were discussed as attitude computation as it is one of the most critical computation tasks in strapdown systems. Filter tuning and its consistency, initialization and data latency. It is stated that for the filter to be consistent and operating correctly the innovations should be zero mean, white and uncorrelated, based on filter innovations behavior, filter tuning was developed and a simplified tuning algorithm was implemented. Real time results were used to run the algorithm off-line on desktop PC where the results were used to chose the proper noise levels of error covariance matrices Q(k) and R(k) and properly tune the filter.

Finally, the thesis provided the hardware and software development details and real-time implementation, experimental tests and corresponding results of the INS/GPS navigation system to validate the reliability and effectiveness of the implemented algorithms and performance of all over navigation system design. An overview of hardware and software development is presented. This development was mainly accomplished in three stages, the first stage was the design, construction, calibration and error modelling of IMU using commercial of the shelf (COTS) components. The second stage was to validate, tune and verify the performance of the developed Kalman Filter software algorithm before to download on Microcontroller for final on-board testing. This step was accomplished by using backlogged real-time data collected from both at the same time frame, the designed IMU (AUSIMU)/GPS, IMU (MIDG II) from Microbotic, Inc. The collected data was used to compare the results of Microbotic navigation solution with our own designed IMU, and GPS solution. All are constructed on one setup and fixed in the host vehicle. The final stage is to cross-compile and download the developed Kalaman filter software to the Microcontroller and log the data of the GPS/INS navigation setup for final verification of navigation system performance.

6.5.2 Future Work

As a continuation from this thesis work, the future focus should be on the improvement of the algorithm performance and code optimization. Since the algorithm program is developed in Simulink in conjunction with Real-Time workshop and Embedded Target for Freescale MPC5xx, thus this environment does not support all features of MPC555 as well does not ese the use of all microcontroller's capabilities as sampling time which is only limited to 0.0001 sec. Therefore the Real-Time workshop blocks must be developed or/and upgraded in order to allow the usage of MPC555 optimum capabilities.

The observability of the attitude states should be thoroughly re-investigated. Unlike position and velocity errors, the uncertainty of attitude errors does not represent the state accurately. During real-time experiments it was found that the uncertainty increase while the vehicle is travelling in straight line and constant acceleration, this is more obvious in the heading case, and it is updated once the vehicle is turning. This leads to a conclusion that, even though attitude and bias estimates are not directly observed by the filter, they can become observable by maneuvering. Therefore the effect of vehicle maneuvering and dynamics on the Kalman filter states observability can be investigated, or an external orientation aided sensor like the digital compass should be added to provide observations that can be used to correct the estimated attitudes.

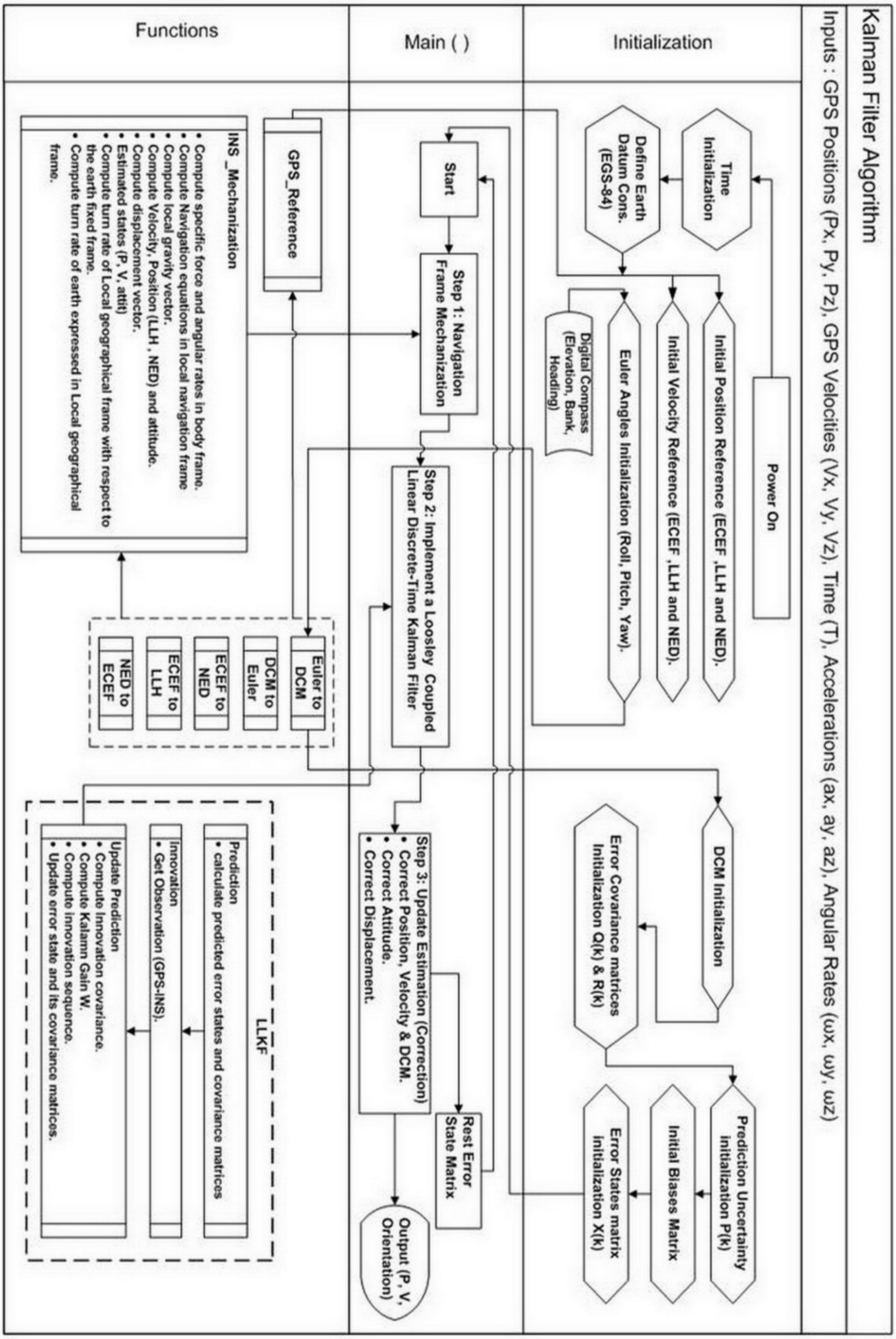

Kalman Filter Algorithm
Inputs : GPS Positions (Px, Py, Pz), GPS Velocities (Vx, Vy, Vz), Time (T), Accelerations (ax, ay, az), Angular Rates (ωx, ωy, ωz)
Initialization
Main ( )
Functions
Time Initialization
Power On
Define Earth Datum Cons. (EGS-84)
Initial Position Reference (ECEF ,LLH and NED).
Initial Velocity Reference (ECEF ,LLH and NED).
Euler Angles Initialization (Roll, Pitch, Yaw).
Digital Compass (Elevation, Bank, Heading)
DCM Initialization
Error Covariance matrices Initialization Q(k) & R(k)
Prediction Uncertainty initialization P(k)
Initial Biases Matrix
Error States matrix initialization X(k)
Start
Step 1: Navigation Frame Mechanization
Step 2: Implement a Loosley Coupled Linear Discrete-Time Kalman Filter
Step 3: Update Estimation (Correction)
• Correct Position, Velocity & DCM.
• Correct Attitude.
• Correct Displacement.
Rest Error State Matrix
Output (P, V, Orientation)
GPS_Reference
INS _Mechanization
• Compute specific force and angular rates in body frame.
• Compute Navigation equations in local navigation frame
• Compute local gravity vector.
• Compute Velocity, Position (LLH , NED) and attitude.
• Compute displacement vector.
• Estimated states (P, V, attit)
• Compute turn rate of Local geographical frame with respect to the earth fixed frame.
• Compute turn rate of earth expressed in Local geographical frame.
Euler to DCM
DCM to Euler
ECEF to NED
ECEF to LLH
NED to ECEF
LLKF
Prediction
• calculate predicted error states and covariance matrices
Innovation
• Get Observation (GPS-INS).
Update Prediction
• Compute Innovation covariance.
• Compute Kalamn Gain W.
• Compute innovation sequence.
• Update error state and its covariance matrices.

Figure 6.16: GPS aided inertial navigation system Kalman filter algorithm data flow. The discretization time Δt used in the algorithm is the sampling time of the inertial unit which is 0.02 *seconds*.

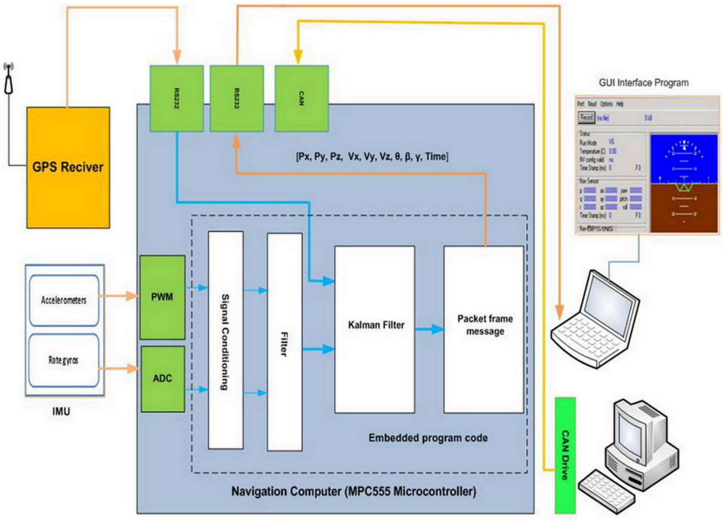


Figure 6.17: The figure shows a flow-chart of the Real-Time algorithm data-flow of INS/GPS navigation system. Kalman filter algorithm is implemented in the MPC555. The program is built in Simulink in conjunction with Real-Time workshop and Embedded Target for Freescale MPC5xx. The generated code is downloaded to the target MPC555 through CAN drive. The program model consists of I/O blocks, filtration, Kalman filter and data frame blocks which is an S-function blocks that can be programmed in C-code.

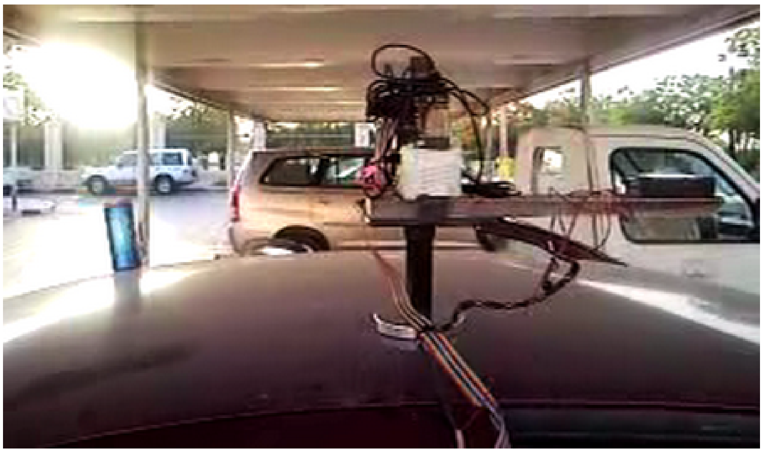


Figure 6.18: A view of INS/GPS navigation setup fixed on vehicle's roof as close as possible to the center, since the vehicle is considered as a small size vehicle, then the lever arm between the location of the setup and the center of the vehicle gravity is neglected.

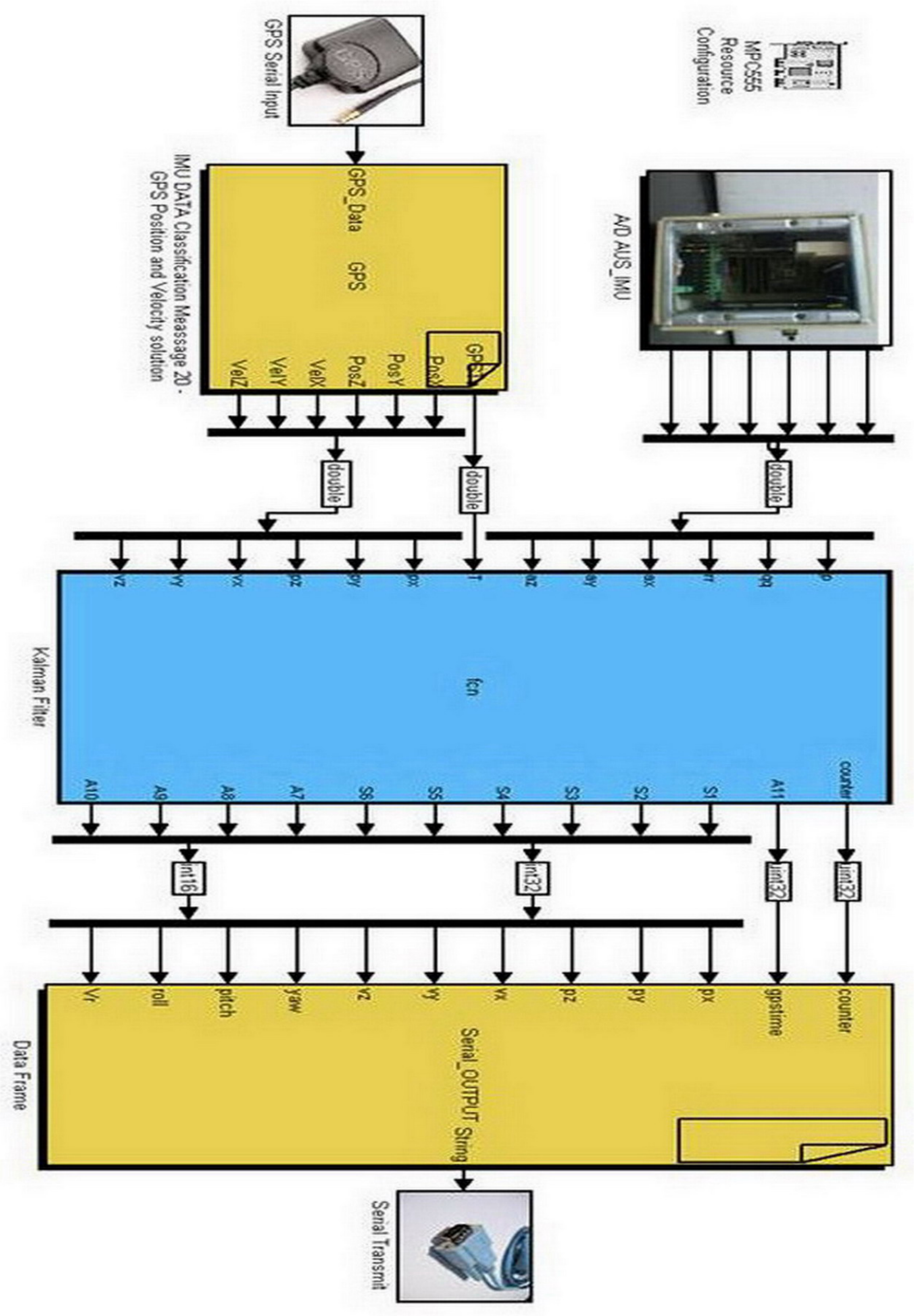


Figure 6.19: Real-Time INS/GPS navigation programm using MPC555. The program is built in Simulink in conjunction with Real-Time Workshop. The generated code is downloaded to the target MPC555 through CAN drive. The program model consists of I/O blocks, KF and data frame blocks which is an S-function blocks that can be programmed in C-code.

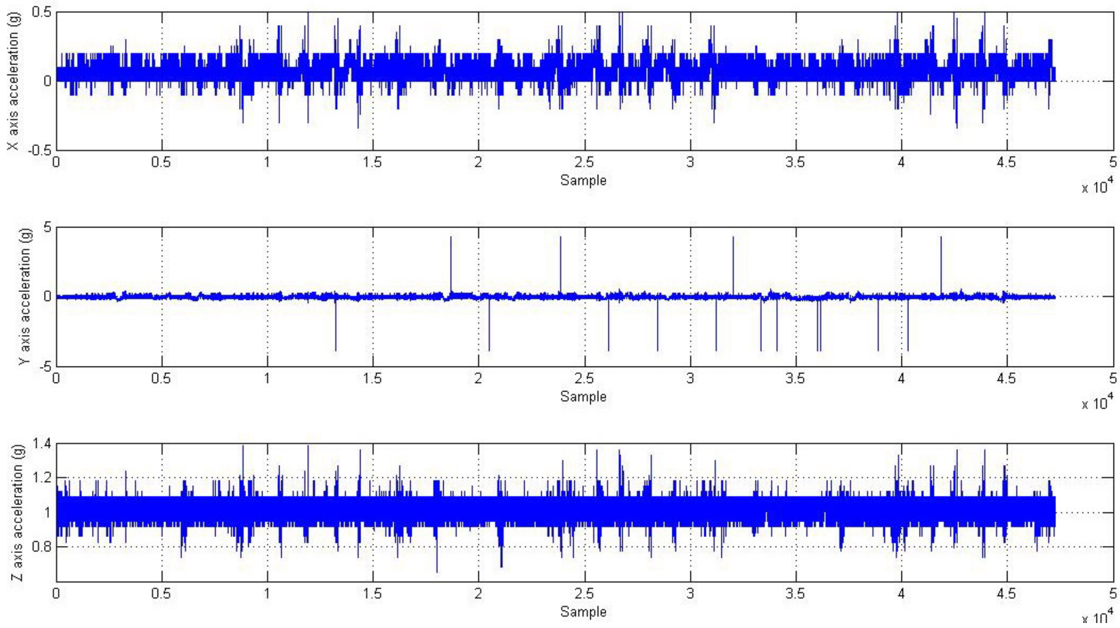


Figure 6.20: Accelerations measurements of three orthogonal accelerometers in the AUSIMU.

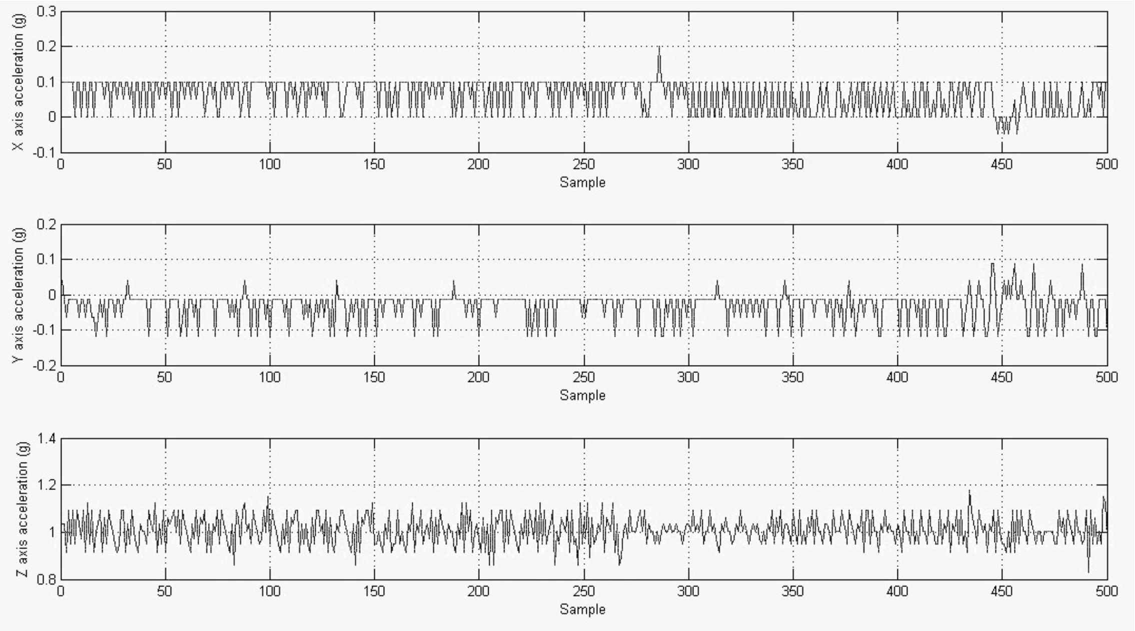


Figure 6.21: Accelerations measurements of three orthogonal accelerometers in the AUSIMU (The first 500 samples).

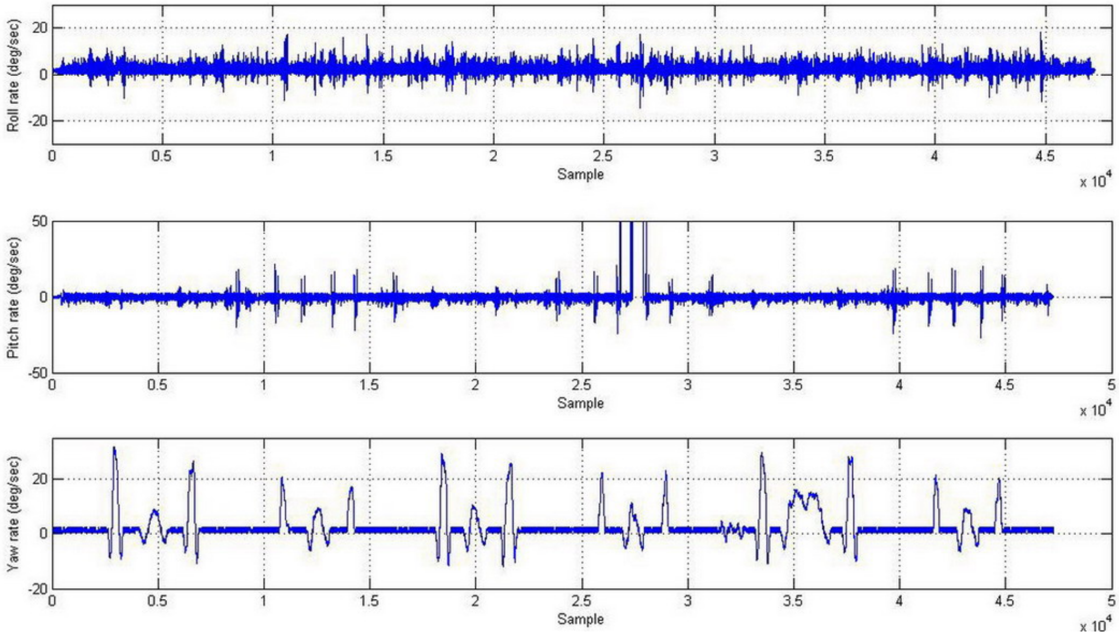


Figure 6.22: Angular rotations measurements of three orthogonal rate-gyros in the AUSIMU.

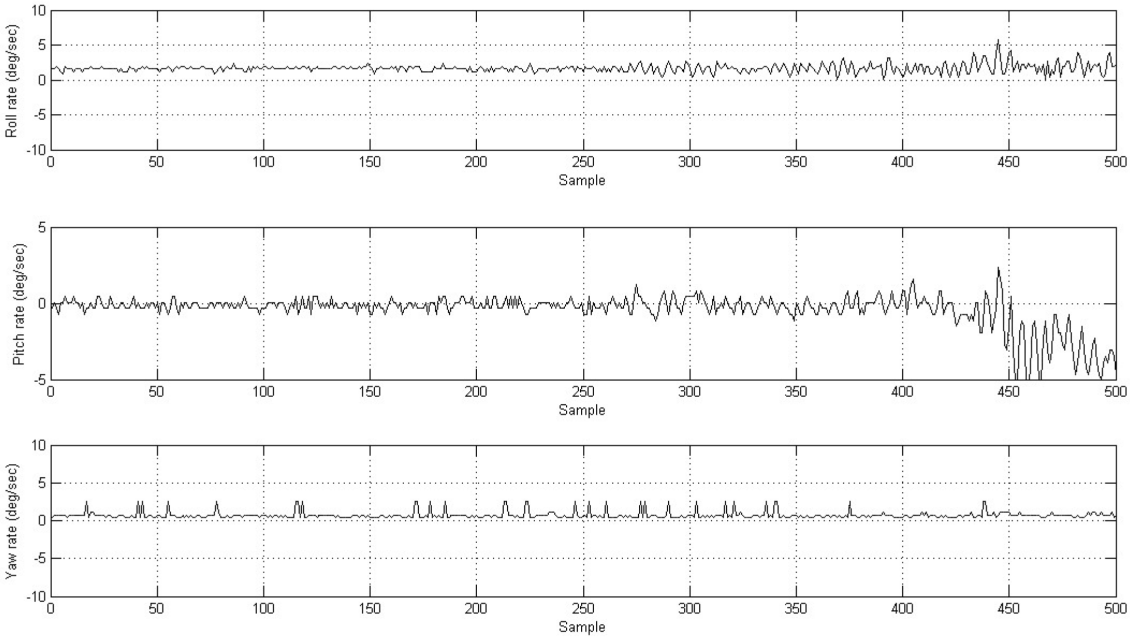


Figure 6.23: Angular rotations measurements of three orthogonal rate-gyros in the AUSIMU The first 500 samples.

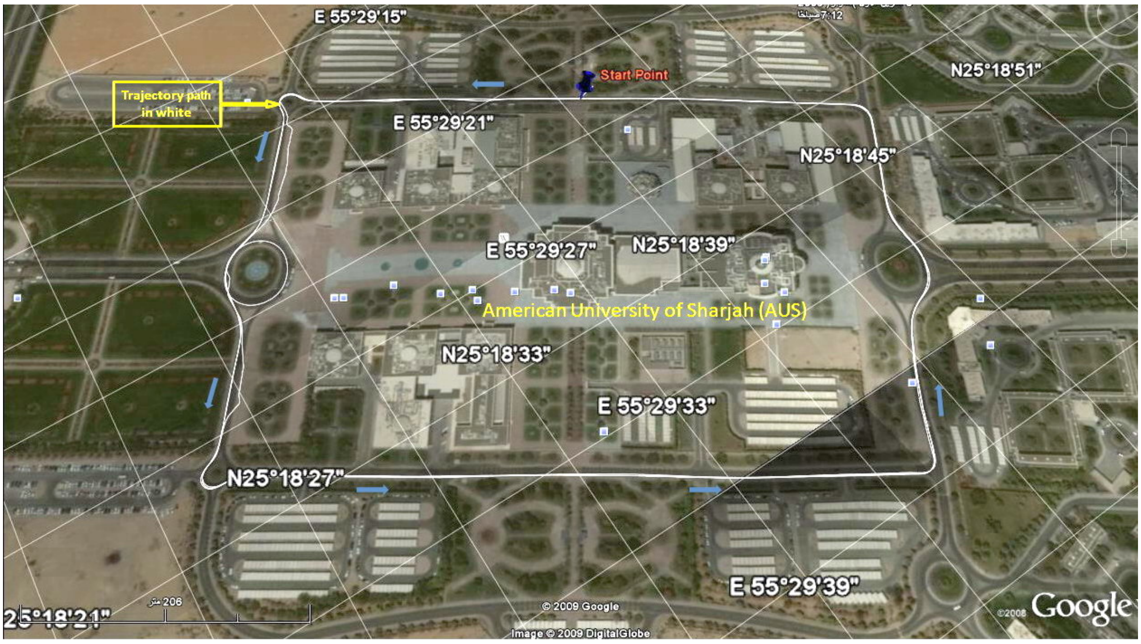


Figure 6.24: Actual trajectory of the trip test around AUS campus, the trajectory is plotted using Google Earth software.

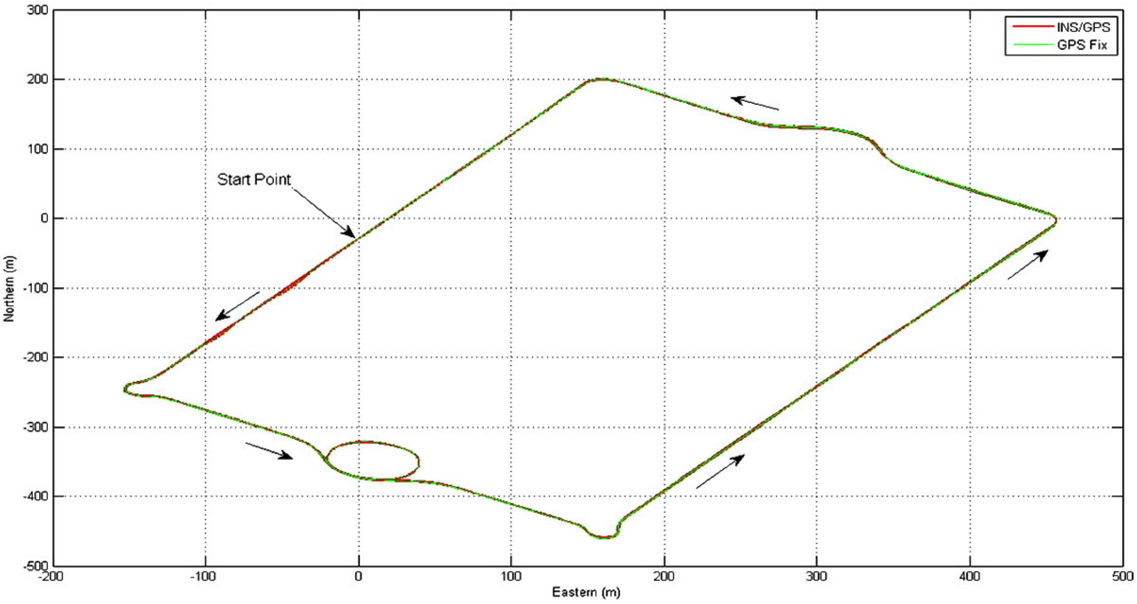


Figure 6.25: Real test trajectory representing GPS position (Blue Line) and INS/GPS solution (Black Line).

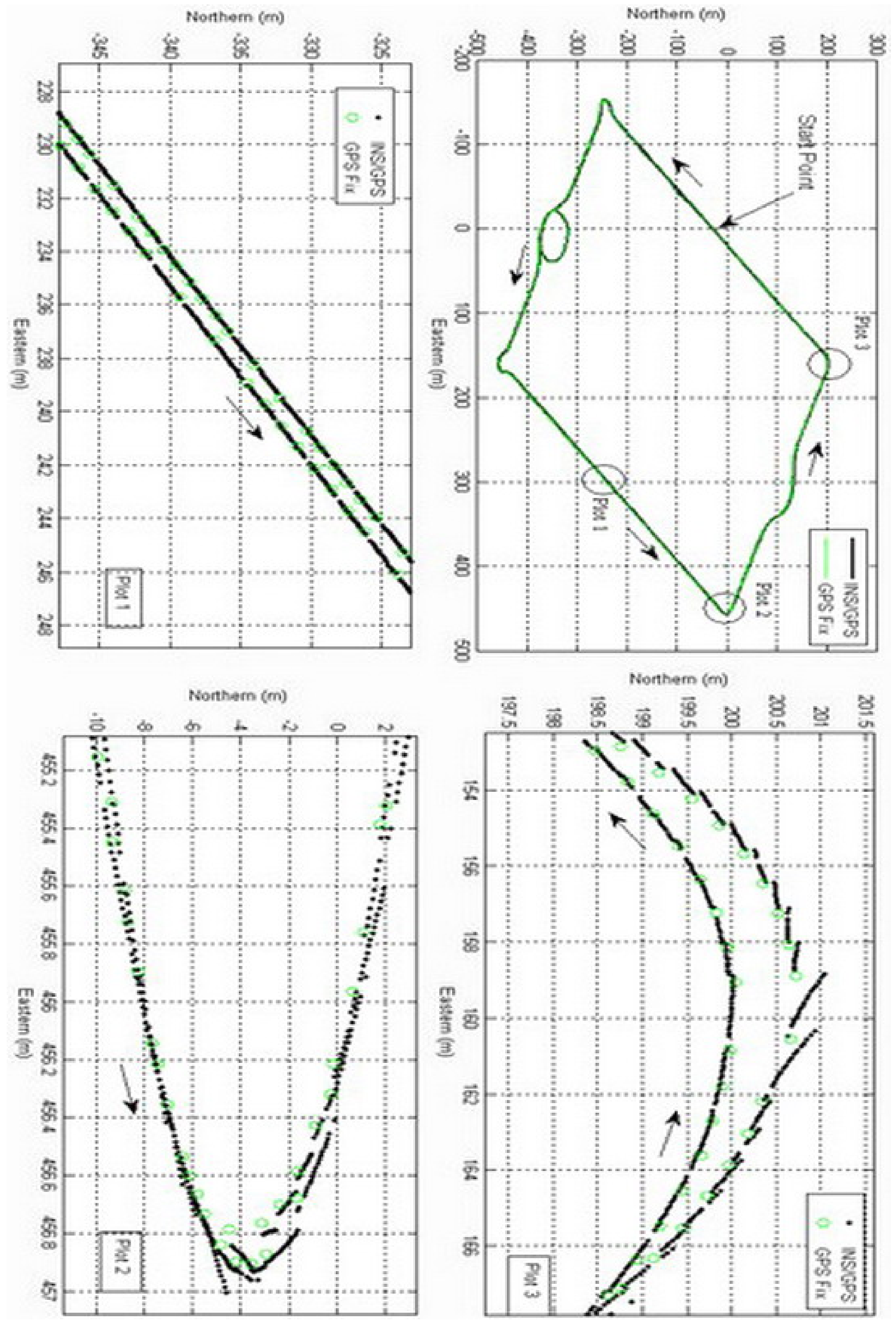


Figure 6.26: Real test trajectory representing GPS position (circle) and INS/GPS solution (solid). loosely-coupled linear filter shows inefficient performance at corners, this is due to the nature of the filter where attitude and bias estimates are zeros during prediction, and are only available when there is a GPS observation that correct the estimate, the filter estimates deviates and this becomes significant at corners.

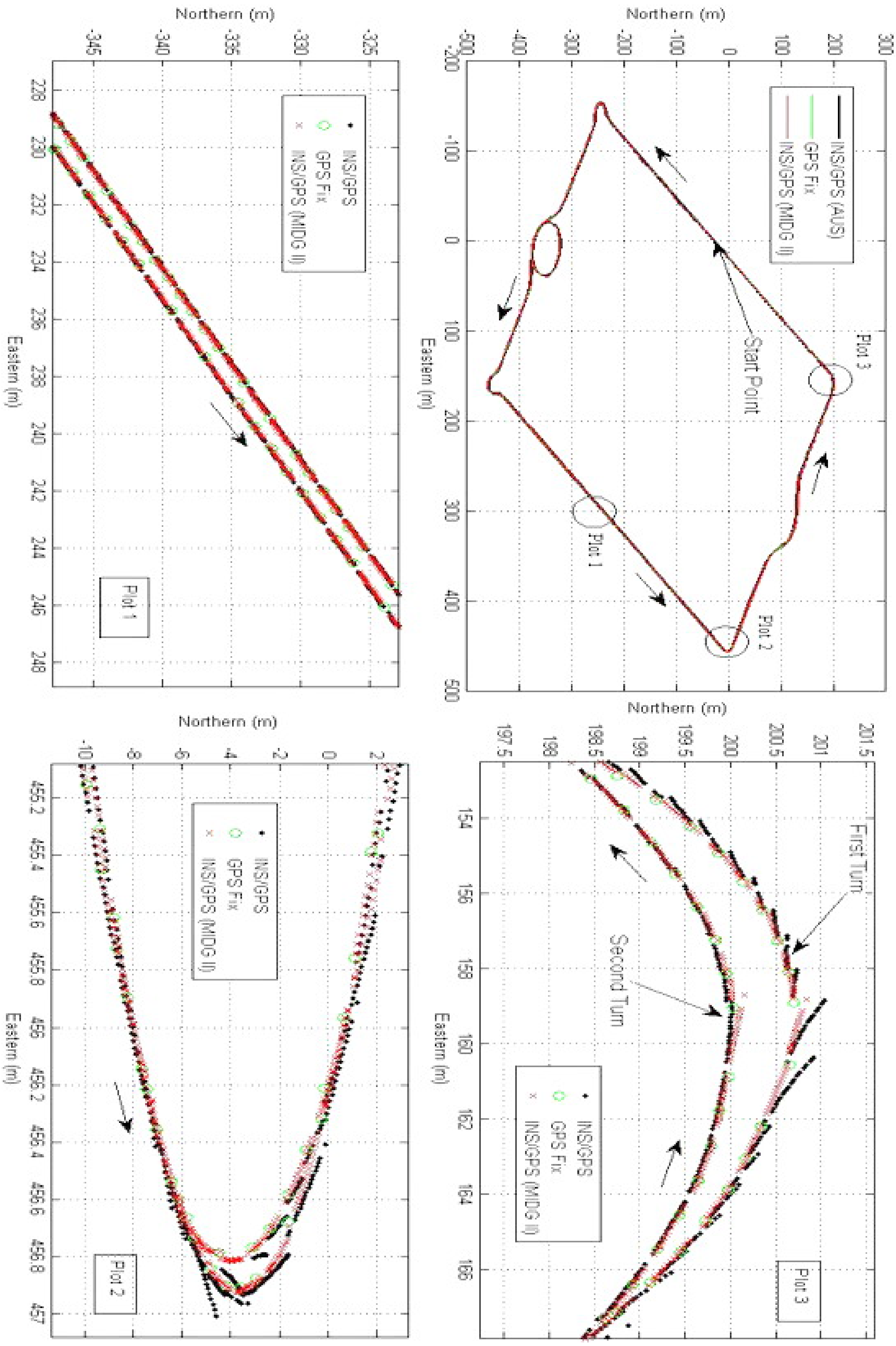


Figure 6.27: Real-time trajectory comparison between the navigation solution of MIDG II (x) and our developed INS/GPS navigation system (solid) and GPS observations (circle).

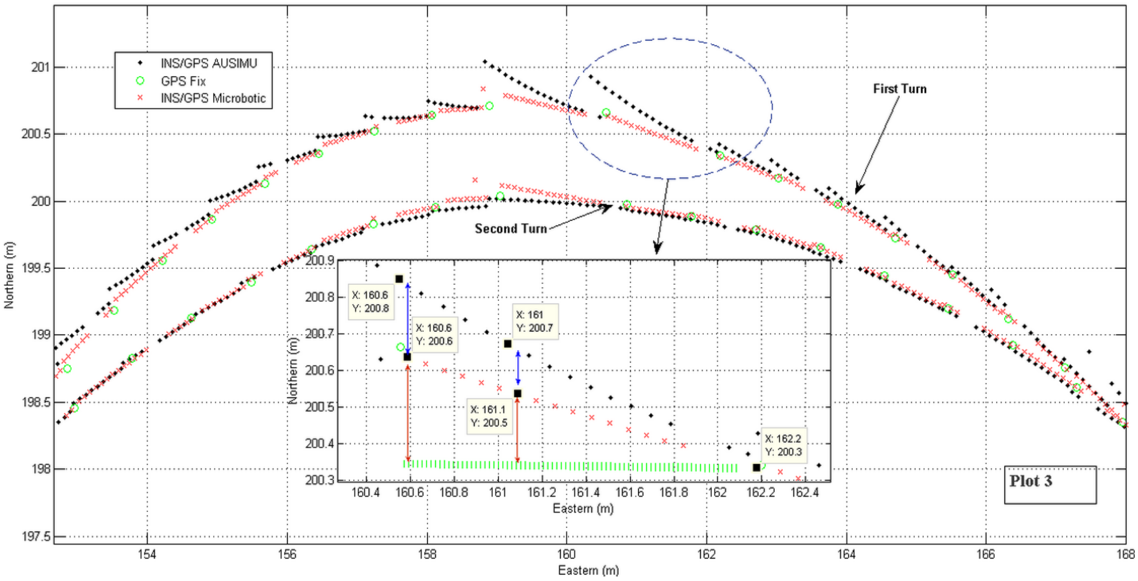


Figure 6.28: Real-time trajectory comparison between the navigation solution of MIDG II (x) and our developed INS/GPS navigation system (solid) and GPS observations (circle). We notic that in first turn that even when error in AUSINS solution starts to grow remains less than error when rely on the last GPS observation, assuming both solutions compared to Microbotic INS solution as shown in the small figure.

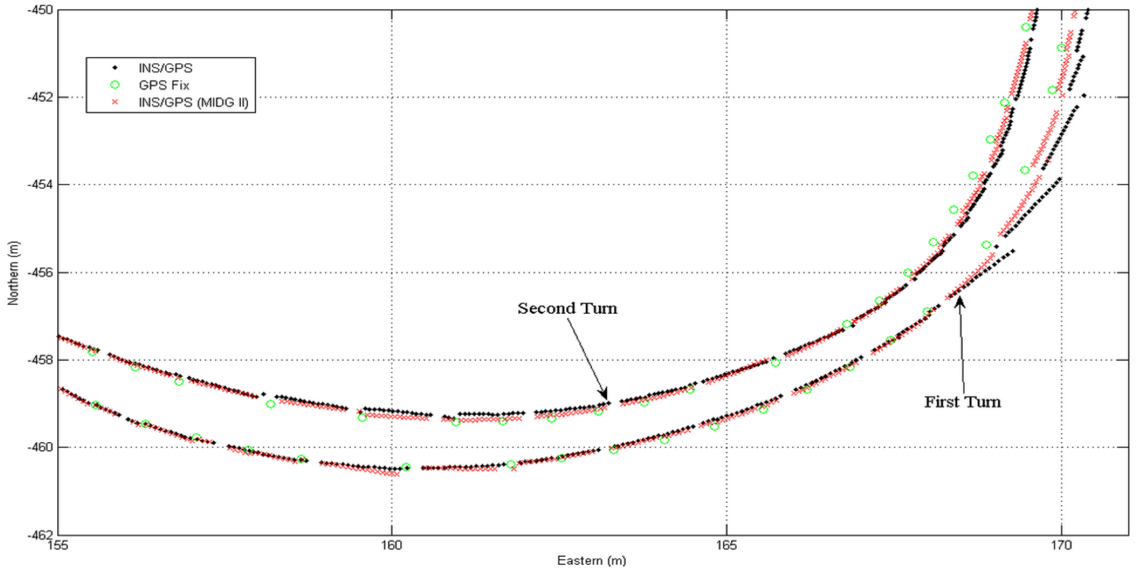


Figure 6.29: Real-time trajectory comparison between the navigation solution of MIDG II (x) and our developed INS/GPS navigation system (solid) and GPS observations (circle).

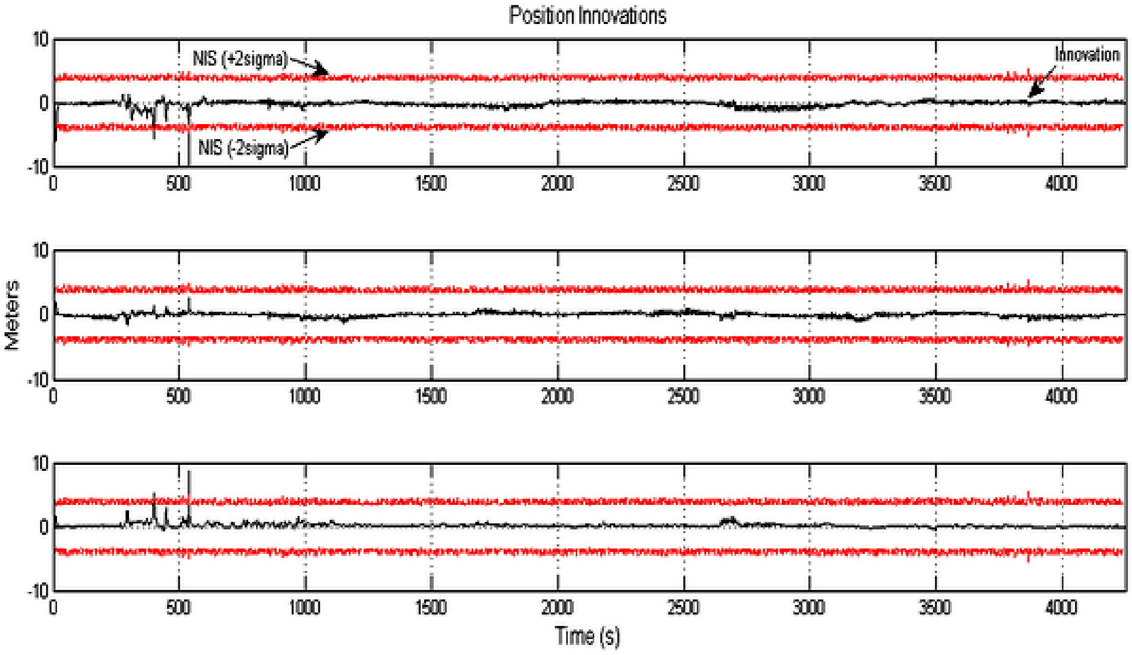


Figure 6.30: Position innovation with its 2σ uncertainty for developed INS/GPS navigation system. The position innovations resemble white noise behavior.

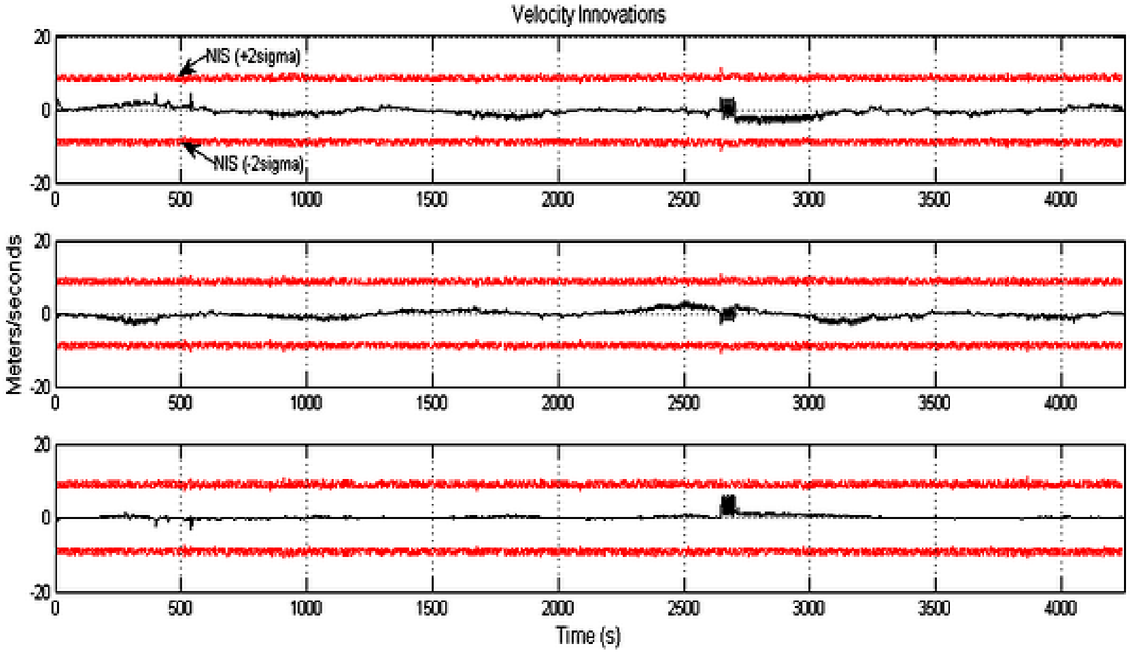


Figure 6.31: Velocity innovation with its 2σ uncertainty for developed INS/GPS navigation system. The position innovations resemble white noise behavior.

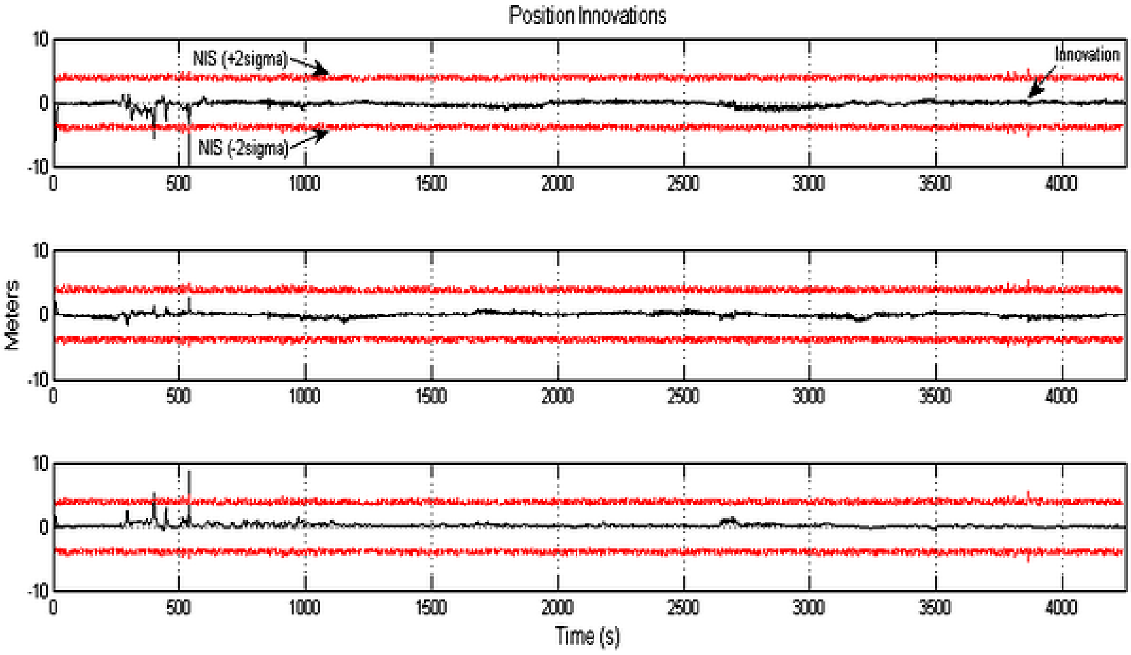


Figure 6.32: Position innovation with its 2σ uncertainty for Microbotic MIDG II. The position innovations resemble white noise behavior.

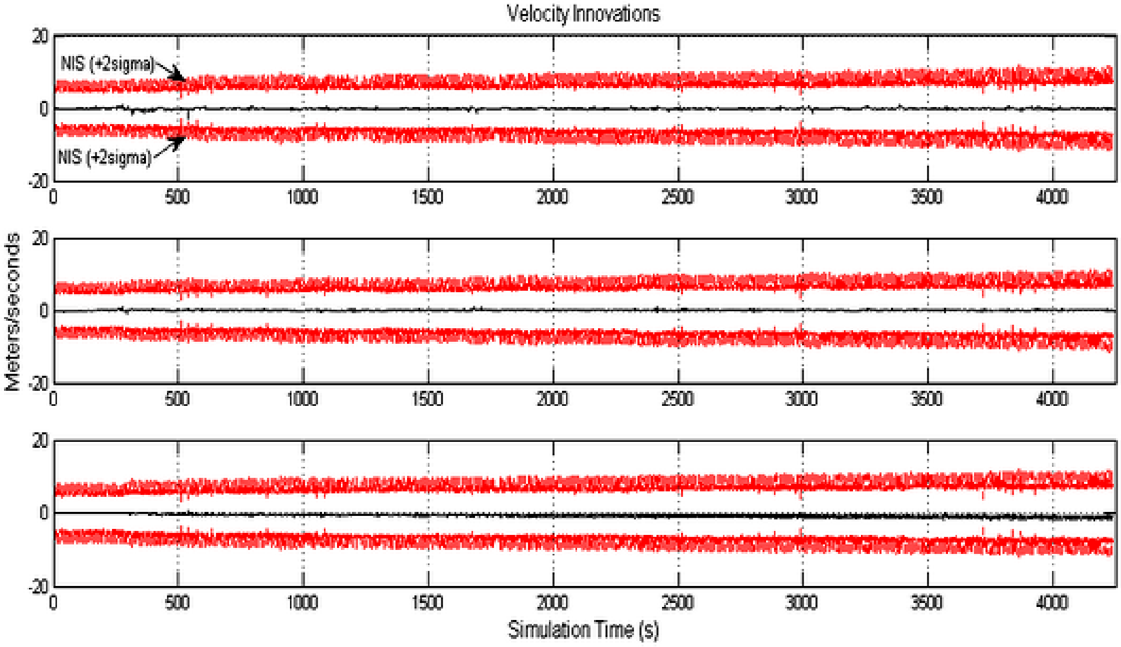


Figure 6.33: Velocity innovation with its 2σ uncertainty for Microbotic MIDG II. The position innovations resemble white noise behavior.

Appendix A

Navigation Reference Frames and Coordinate Systems

For the purpose of operative and efficient navigation a pre defined number of frame references is used through out this thesis works in which dynamics laws are formulated and measurements made, and there should be a defined coordinate system where equations of navigation are mechanized. The reference frames are real entities and part of physics, and the second is a arbitrarily created coordinates for several reason i.e. visualization and reduction of computational complexity [28].
This appendix describes the most common reference frames and coordinate systems found in the navigation field.

A.1 Earth Centered Inertial Reference Frame (ECI)

The Earth Centered Inertial (ECI) reference frame in which Newton's laws of motion holds. The frame axes are non-accelerating and non-rotating with respect to the fixed stars.

- **Origin**: The origin of (ECI) frame is the Earth's geometric center
- **x-axis**: The x-axis pointing to the distant star known as vernal equinox.
- **y-axis**: The y-axis spans the equational plane with x-axis completing the orthogonal right-handed set.
- **z-axis**: The z-axis pointing up through the geometric north pole

A.2 World Geodetic System 1984 (WGS-84)

The earth is actually approximated by an ellipsoids (*oblate spheroid*) rather than a sphere. These approximations can be modelled in different ways namely *The topographic models* which represents the real-physical shape of the Earth and the mean level of oceans or *The geodetic models.* The geodetic datum approximates the shape of the earth over the entire globe and defines a 3-D geographic coordinate system. The WGS 84 is Earth-fixed orthogonal, right-handed coordinate system where the geodetic latitude, longitude and the ellipsoidal height being the dimensions of this coordinate system [3]. Figure A.1 shows the Geoid Ellipsoidal Orthometric height and WGS-84 reference frame. The current standard model for the reference ellipsoid is the WGS-84 system defined by the Defense Mapping Agency, which is a state-of-the-art system based on the use of data, techniques, and technology available in early 1984[45]. The values which are given in table A.1 were defined for geodetic datum by the World Geodetic System Committee in 1984.

Table A.1: WGS-84 ellipsoid parameters

Parameter	**Value**	**Unit**
Length of semi-major axis R	6378137	m
Length of semi-minor axis r	6356752.3142	m
Flattening of the ellipsoid (Ellipticity) f, $f = \frac{R-r}{R}$	0.00335281066474	
Eccentricity of the Earth ellipsoid e, $e = \sqrt{f(2-f)}$	0.0818191908426	
Earth's angular velocity Ω_{ie}	7.292115×10^{-5}	rads
Earth's Gravitational Constant μ	$3986004.418 \times 10^{8}$	m^3s^2
Mass of the Earth M	5.9733328×10^{24}	Kg
Pi π	3.1415926535898	

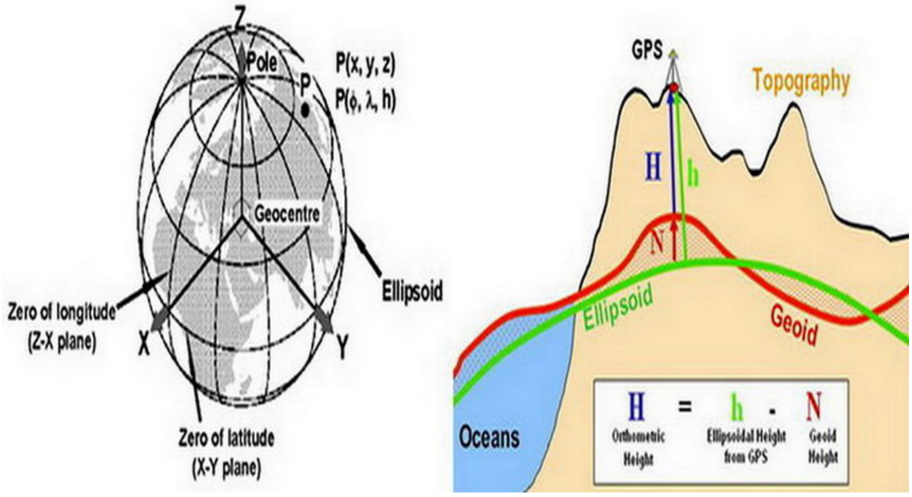


Figure A.1: The Geoid Ellipsoidal Orthometric height and WGS-84 reference frame

A.3 Coordinate Systems

The coordinate systems considered in this thesis are Earth stabilized systems in which they maintain orientation referenced in variance ways to the Earth. These coordinate systems are Earth Centered Earth Fixed Coordinate System (ECEF), North-East-Down, Geographic Coordinate System (NED) and latitude, longitude and altitude, Geodetic Coordinate System (LLH):

A.3.1 Earth Centered Earth Fixed (ECEF)

The Earth-center, Earth-fixed (ECEF) system is a non-inertial system that rotates with the respect to the Earth and its origin is fixed at the center of the Earth as shown in figure A.2. GPS uses this frame as defined by the WGS-84 to represent the velocity and position of a host vehicle relative to the Earth. This frame can be defined as follows:

- **Origin**: Earth's center of mass or Earth geometric center.
- **x axis**: This x axis pointing through the intersection of Greenwich meridian $longitude = 0$ and equator.

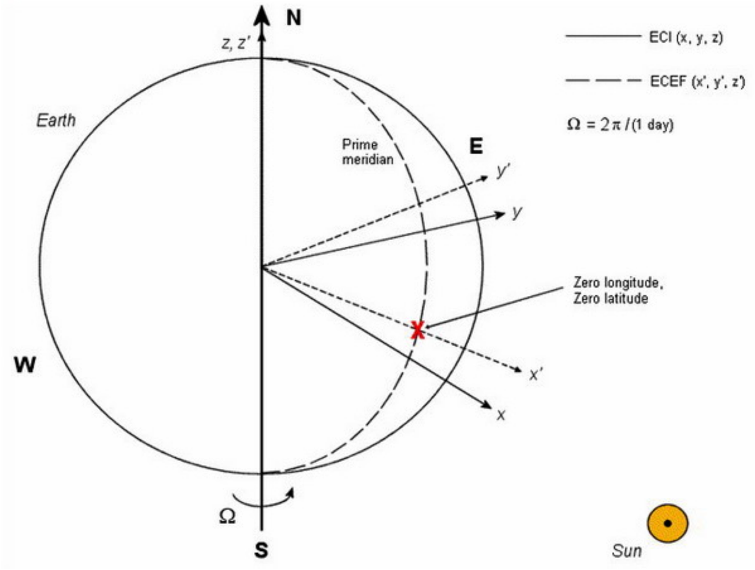


Figure A.2: The Earth Centered Earth Fixed ECEF and Earth Centered Inertial reference.

- **y axis**: Complete the right hand ECEF orthogonal coordinate system.
- **z axis**: The z axis points up from the origin to the north pole.

A.3.2 Geographic Coordinate System (NED)

The NED frame is also known as Earth surface navigation frame, this frame is considered as local level frame defined at the vehicle's current position. The NED coordinate system is shown in figure A.3. In this thesis the navigation equations are developed in this frame and all other measurements or observations are transferred to NED using the transformation equations. This frame is defined as follows:

- **x-axis**: This axis tangent to Earth's surface and points to the north.
- **y-axis**: Tangent to Earth's surface and points to the east.
- **z-axis**: Completes the set and positive to down.

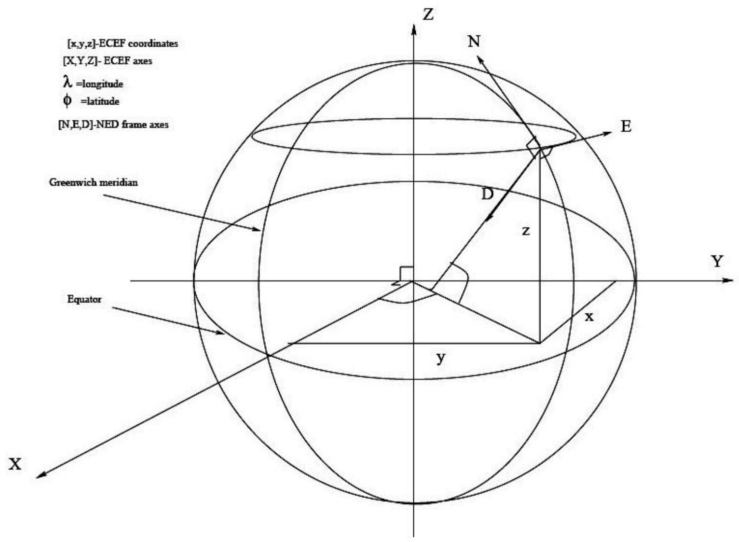


Figure A.3: The Geographic Coordinate System (NED) with respect to The Earth Centered Earth Fixed (ECEF).

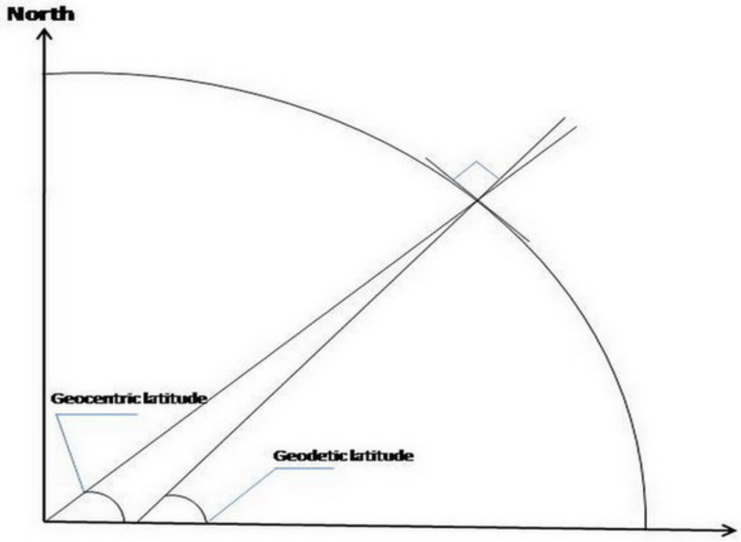


Figure A.4: The difference between the Geocentric latitude and Geodetic latitude.

A.3.3 Geodetic Coordinate System (LLH)

The Geodetic Coordinate System is a fixed coordinate with respect to rotating Earth, and the origin of this coordinate system is considered as Earth's geometric center. It can be described as follows:

- **Latitude**: Two distinct definitions of latitude can be find in the literature namely the geocentric latitude and geodetic latitude. The geodetic latitude is defined at a point as the angle between equatorial plane and a line perpendicular to the reference ellipsoid. However the Geocentric latitude is the angle between equatorial plane with respect to the Earth's center A.4. The distinction is drawn in figure A.4.

- **longitude**: The angle relative to Greenwich.

- **Height**: The minimum distance between the point of interest and the reference ellipsoid.

Appendix B

dSpace Data acquisition system

B.1 dSpace Data acquisition board

The dSpace DS1104 RD (RD stands for research and development) Controller Board is a piece of hardware that upgrades your PC to a powerful development system for rapid control prototyping see figure B.1and B.2. The real-time hardware based on PowerPC technology and its set of I/O interfaces makes the board an ideal solution for developing controllers in various industrial fields. The DS1104 RD Controller Board is impressive proof that power does not necessarily have to be expensive.

Yet it still gives you all the benefits of a dSpace Prototyper system: full graphical configuration, programming in Simulink/Stateflow from The MathWorks and experiment control with state-of-the-art software tools. The board can be installed in virtually any PC with a free PCI slot. Below some technical details about dSpace is listed to show the power of this hardware.

B.1.1 Technical Details

Main Processor:

- MPC8240, PowerPC 603e core, 250 MHz.
- 32 kB internal cache.

Timers:

- 1 sample rate timer

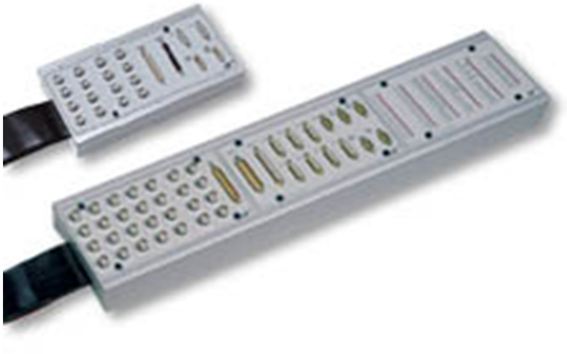


Figure B.1: dSpace terminal box.

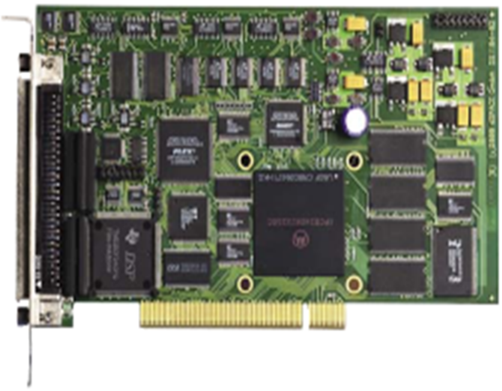
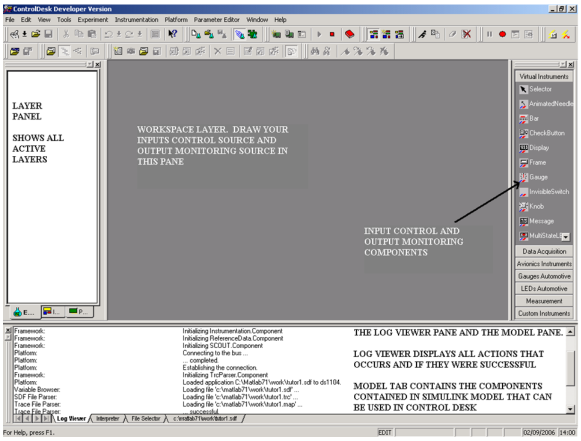


Figure B.2: dSpace DSP Card MPC8240, PowerPC 603e core, 250 MHz.



LAYER PANEL

SHOWS ALL ACTIVE LAYERS

WORKSPACE LAYER. DRAW YOUR INPUTS CONTROL SOURCE AND OUTPUT MONITORING SOURCE IN THIS PANE

INPUT CONTROL AND OUTPUT MONITORING COMPONENTS

```
Framework:           Initializing Instrumentation.Component
Framework:           Initializing ReferenceData.Component
Framework:           Initializing SCOUT.Component
Platform:            Connecting to the bus ...
Platform:            ... completed.
Platform:            Establishing the connection.
Framework:           Initializing TrcParser.Component
Platform:            Loaded application C:\Matlab71\work\tutor1.sdf to ds1104.
Variable Browser:    Loading file 'c:\matlab71\work\tutor1.sdf' ...
SDF File Parser:     Loading file 'c:\matlab71\work\tutor1.trc' ...
Trace File Parser:   Loading file 'c:\matlab71\work\tutor1.map' ...
Trace File Parser:   ... successful.
```

THE LOG VIEWER PANE AND THE MODEL PANE.

LOG VIEWER DISPLAYS ALL ACTIONS THAT OCCURS AND IF THEY WERE SUCCESSFUL

MODEL TAB CONTAINS THE COMPONENTS CONTAINED IN SIMULINK MODEL THAT CAN BE USED IN CONTROL DESK

Figure B.3: Control Desk Developer version software.

- 32-bit down counter.
- 4 general purpose timers, 32 bit.
- 64-bit time base for time measurement.

Memory.

- 32 MB synchronous DRAM (SDRAM).
- 8 MB boot flash for applications.

Interrupt Control Unit:

- Interrupts by timers, serial interface,slave DSP, incremental encoders, ADC, host PC and 4 external inputs.
- PWM synchronous interrupt.

Analog Input:

- 4 ADC inputs with one ADC unit, 16 bit, multiplexed.
- 10 V input voltage range.
- 2 s sampling time.
- > 80 dB signal-to-noise ratio.
- 4 ADC channels, 12 bit.
- 10 V input voltage range.
- 800 ns sampling time.
- > 65 dB signal-to-noise ratio.

Analog Output:

- 8 channels, 16 bit, 10 s max settling time.
- 10 V output voltage range.

Incremental Encoder Interface:

- Two digital inputs,TTL or RS422.
- 24-bit digital incremental encoders.
- Max. 1.65 MHz input frequency, i.e. fourfold pulse counts up to 6.6MHz.
- 5 V / 0.5 A sensor supply voltage.

Digital I/O:

- 20-bit digital I/O (bit-selectable.
- 5 mA output current.

Serial Interface:

- Serial UART (RS232, RS485 or RS422).
- Slave DSP Subsystem
- Texas Instruments' DSP TMS320F240.
- 4 kWord of dual-port RAM.
- Three-phase PWM outputs plus 4 single PWM outputs.
- Frequency measurement (F/D) and generation (D/F),4 channels each.
- Frequency measurement (F/D) and generation (D/F),4 channels each.
- 14 bits of digital I/O (TTL).

Physical Characteristics:

- Power supply 5 V, 2.5 A / −12 V, 0.2 A / 12 V, 0.3 A.
- Operating temperature 0 to 55 C (32 to 131 F).
- Requires one 33 MHz / 32-bit 5-V PCI slot.
- The I/O connector can be linked to two 50-pin, female Sub-D connectors using the adapter cable supplied.

B.2 Control Desk

Control Desk is the software that interfaces with the dSpace hardware. It allows the user to interact with the hardware, and change input values and also measure the states of various components. see figure B.3

Bibliography

[1] J. Skaloud and K.P. Schwarz, "Accurate Orientation for Airborne Mapping Systems", Photogrammetric Engineering and Remote Sensing, vol.66, No. 4,pp. 393-401, April, 2000.

[2] M.S. Grewal, L.R. Weill, and A.P. Andrews, *Global Positioning Systems, Inertial Navigation, and Integration.* 2nd ed. NJ:Wiley-Interscience, 2001.

[3] D. Titterton and W. Weston, *Strapdown Inertial Navigation Technology.* 2nd ed. Virginia: The Institution of Electrical Engineers (IEE), 2004.

[4] X. Kong, "Inertial Navigation System Algorithms for Low cost IMU," Ph.D. dissertation, Aerospace, Mechanical and Mechatronic Engineering, Australian Centre for Field Robotics, The University of Sydney, Sydney, Australia, August, 2000.

[5] S. Sukkarieh, "Intertial Navigation Workshop," ISMA07 Navigation Workshop, AUS, Sharjah, UAE, 2007.

[6] k. Britting, *Inertial Navigation System Analysis.* New York: Wiely Interscience, 1979.

[7] B. Steiler and H. Winter, *Gyroscopic Instruments and Their Application to Flight Testing*, vol.15, AGARD Flight Test Instrumentation Series, September, 2002.

[8] R. Schelling, *A Low-Cost Angular Rate Sensor for Automotive Applications in Surface Micromachining Technology*, Third Annual International Conference on Advanced Microsystems for Automotive Applications Proceedings, March, 1999.

[9] Z.C. Wu, Z.F. Wang and Y. Ge, "Gravity based online calibration of monolithic tri-axial accelerometers gain and offset drift," in Proceeding 4-th World Congress on Intelligent Control and Automation,1014 June. 2002.

[10] A.B. Chatfield, *Fundamentals of High Accuracy Inertial Navigation.* Reston, Virginia: American Institute of Aeronautics and Astronautics, 1997.

[11] I. Skog and P. Handel, *Calibration of MEMS Inertial Measurement Unit.* XVII IMEKO WORLD CONGRESS, Metrology for a Sustainable Development, 2006.

[12] J. Hall, II R. Williams, and F.Graas, *Cartesian Control for the Inertial Measurement Unit Calibration Platfor.* September, 2000.

[13] Analog Devices Inc., *ADXL202EB Dual Axis Accelerometer Evaluation Board Data Sheet(Rev. A),[Online].http://www.analog.com.*

[14] Analog Devices Inc., *ADXL202/210 Data sheet,[Online].http://www.analog.com.*

[15] Analog Devices Inc., *ADXRS150EB Single Chip Rate Gyro Evaluation Board,[Online].http://www.analog.com.*

[16] Analog Devices Inc., *ADXRS150 data sheet,[Online].http://www.analog.com.*

[17] E.M. Nebot and H.F. Durrant-Whyte, *Initial Calibration and Alignment of Low Cost Inertial NAvigation Units for Land Vehicle Applications.* Journal of Robotics Systems, vol.16(2),Feb. 1999,pp.81-92.

[18] J.C. Hung, J.R. Thacher and H.V. White, "Calibration of accelerometer triad of an IMU with drifting Z -accelerometer bias", in Proceeding NAECON 1989, IEEE Aerospace and Electronics Conference, 2226 May, 1989,vol.1,pp.153158.

[19] R M. Rogers, *Applied Mathematics In integrated Navigation Systems.* 2nd ed. AIAA Education Series, 2003.

[20] Excelmachinetools Inc., *Excel Dividing Heads Semi Universal and HV6 Excel Precision Rotary Tables, 1995-2008.*

[21] D. Montgomery and G. Runger, *Applied Statistics and Probability for Engineers.* 2nd ed. New York: Wiley, 1999.

[22] M. Jhon, "Robust navigation with GPS/INS and adaptive beam forming," Swedish Defence Research Agency,Stocholm,Sweden,Tech.Rep. FOI-R-0848-SE, 2003.

[23] D. Grejner-Brzezinska and C. Toth, *Precision Mapping Highway Linear Features.* Geoinformation for all, Proceedings, XIXth ISPRS congress, July 16-23, 2000, Amesterdam, Netherlands,pp.233-240.

[24] P.D. Groves, "Principles of Integrated Navigation," Issu 1.1 of a training course prepared by the Telmatic Solutions Group of QinertiQ Ltd, Farnborough, October 2003.

[25] R.E. Phillips and G.T. Schmidt, *GPS/INS integration architectures.* in Advisory Group for Aerospace Research and Development (AGARD),Lecture Series:9.1-9.10, 2003.

[26] J. Weiss, and D.S. Kee, "A Direct Performance Comparison Between Loosely Coupled and Tightly Coupled GPS/INS Integration Techniques", IEEE Sensors Journal, pp.537-544, 2001.

[27] M.G. George and S. Sukkarieh, *Tightly Coupled INS/GPS with Bias Estimation for UAV Applications.* in Proceedings of the Australasian Conference on Robotics and Automation (ACRA), 2005.

[28] M.G. George, "All-Source Navigation for an Unmanned Aircract," M.S. thesis, School of Engineering, Aerospace, Mechanical and Mechatronic Department, Australian Centre for Field Robotics, The University of Sydney,Sydney,Australia,March, 2007.

[29] P.S. Maybeck, *Stochastic models, estimation, and control. Vol.1*, Academic Press, 1979.

[30] A. Kim and M.F. Golnaraghi, *Initial calibration of an inertial measurement unit using an optical position tracking system.* in Proceeding IEEE Position Location and Navigation Symposium, April, 2004, pp. 96101.

[31] P.G. Savage, "A Unified Mathematical Framework for Strapdown Algorithm Design". Jornal of Guidance and Dynamics, vol.29,No.2,March-April, 2006, pp.237-249.

[32] S.A. Tariq, "Improving INS/GPS Integration for Mobile Robotics Applications," M.S. thesis, School of Engineering,Mechatronics Department, School of Engineering, The American University of Sharjah, Sharjah, UAE, April, 2008.

[33] M.S. Ahmed and D.V. Cuk, "Comparison of Different Computtion Methods for Strapdown Inertial Navigation Systems," Faculty of Mechanical Engineering, Belgrad, Scientific-Technical Review. vol.LV,No.2,2005.

[34] S. Sukkarieh, "Low Cost, High Integrity, Aided Inertial Navigation Systems for Autonomous Land Vehicles," Ph.D. dissertation, Mechanical and Mechatronic Engineering, Australian Centre for Field Robotics, The University of Sydney, Sydney, Australia, March, 2000.

[35] J. Kim, "Autonomous Navigation for Airborn Applications," Ph.D. dissertation, Aerospace, Mechanical and Mechatronic Engineering, Australian Centre for Field Robotics, The University of Sydney, Sydney, Australia,May, 2004.

[36] D.G. Meskin and I.Y. Bar-Ithack, *Unified Approach to Inertial Navigation System Error Modeling.* Journal of Guidance, Control, and Dynamics, vol.15(3),pp.648-653, 1992.

[37] I.Y. Bar-Itzhack, *Identity Between INS Position and Velocity Error Models.* Journal of Guidance and Control, September, 1981, pp.568-570.

[38] P.S. Maybeck, *Stochastic models, estimation, and control. Vol.1.2*, Academic Press, 1982.

[39] H. F. Durrant-Whyte, *Introduction to Estimation and the Kalman Filter*. Australia: Australian Centre for Field Robotics, The University if Sydney, NSW,January, 2001.

[40] Y. Bar-Shalom, X.R. Li and T. Kirubarajan,. *Estimation with Applications to Tracking and Navigation : Theory Algorithms and Software*. New York: John Wiley & Sons, Ltd., 2001.

[41] E.M. Nebot, *Navigation System Design*. Centre of Excellence for Autonomous Systems, Australian Centre for Field Robotics, The University of Sydney NSW, Australia, May 2005.

[42] PHYTEC Technology Holding Company, *phyCORE-MPC555 Hardware Manual*, PHYTEC MeBtechnik GmbH, 2005.

[43] The MathWorks, Inc., *Embedded Target for Motorola MPC555 Users Guide*, Dec., 2002.

[44] Microbotic Inc., *MIDG II Display and Configuration Program*, March, 2007.

[45] S. H. Stovel, *Basic Inertial Navigation*. China Lake, California: AIAA, 1997.