UNIVERSITI MALAYSIA PAHANG

BORANG PENGESAHAN STATUS TESIS

JUDUL: DATA FILTERING OF 5-AXIS INERTIAL MEASUREMENT UNIT USING KALMAN FILTER

SESI PENGAJIAN: 2012/2013

Saya, **NUR SYAZWANI BINTI SAMSUDIN (870704-10-5126)**
(HURUF BESAR)

mengaku membenarkan tesis (Sarjana Muda/~~Sarjana/Doktor Falsafah~~)* ini disimpan di perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Tesis ini adalah hakmilik Universiti Malaysia Pahang (UMP).
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. **Sila tandakan (√)

☐ **SULIT** (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

☐ **TERHAD** (Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi / badan di mana penyelidikan dijalankan)

☑ **TIDAK TERHAD**

Disahkan oleh:

___________________ ___________________

(TANDATANGAN PENULIS) (TANDATANGAN PENYELIA)

Alamat Tetap:

No 88, Jalan SJ 1A,
Taman Sijangkang Jaya,
42500 Telok Panglima Garang, Selangor
Tarikh:

WAN HASBULLAH BIN MOHD ISA
(NamaPenyelia)

Tarikh:

CATATAN: * Potong yang tidak berkenaan.

** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh tesis ini perlu dikelaskan sebagai SULIT atau TERHAD.

DATA FILTERING OF 5-AXIS INERTIAL MEASUREMENT UNIT USING KALMAN FILTER

NUR SYAZWANI BINTI SAMSUDIN

Report submitted in partial fulfilment of the requirements
for the award of the degree of
Bachelor of Mechatronics Engineering

Faculty of Manufacturing Engineering
UNIVERSITY MALAYSIA PAHANG

JUNE 2013

SUPERVISOR’S DECLARATION

We hereby declare that we have checked this project and in our opinion this project is adequate in terms of scope and quality for the award of the degree of Bachelor of Mechatronics Engineering.

Signature
Name of Supervisor: MR. WAN HASBULLAH BIN MOHD ISA
Position: HEAD OF MECHATRONICS ENGINEERING PROGRAMME
Date: 19 JUNE 2013

STUDENT’S DECLARATION

I hereby declare that the work in this project is my own except for quotations and summaries which have been duly acknowledged. The project has not been accepted for any degree and is not concurrently submitted for award of other degree.

Signature:
Name: NUR SYAZWANI BINTI SAMSUDIN
ID Number: FB09051
Date: 19 JUNE 2013

To my parents Samsudin Ahmad and Niyun Hj Mohd Ali
Thank you for your infinite and unfading love, sacrifice, patience, encouragement and best wishes, and to those who made it possible

ACKNOWLEDGEMENTS

All praises and thanks to Allah (S.W.T), who has guided us to this, never could we have found guidance, were it not that Allah had guided us! (Q7:43). First and foremost I offer my sincerest gratitude to my project supervisor, Mr Wan Hasbullah bin Mohd Isa for his supervision, germinal ideas, invaluable guidance, continuous encouragement, patience and support in making this project possible.

My deepest gratitude goes to my beloved parents and family for their endless love, prayers, encouragement and willingness in helping me completed this project. Without them, I could not have finished the study.

Last but not least, sincere thanks to all my friends for their kindness and moral support during my study. A special thanks to those who always give a hand till the very end of this project. Thanks for the friendship and memories, may Allah bless all of you.

ABSTRACT

This thesis has the purpose to design and develop data filtering of 5-axis Inertial Measurement Unit (IMU) using Kalman Filter. This project endeavour to verify that the data from 5DOF IMU can be filtered using Kalman Filter method so that it can be used as an algorithm in motion alignment. The IMU consists of 2-axis of gyroscopes and 3-axis of accelerometer. The Kalman filter is a set of mathematical equations that provides an efficient computational (recursive) means to estimate the state of a process, in a way that minimizes the mean of the squared error. The main contribution of these algorithms is the in-motion alignment approach with unknown initial conditions. This study explores the use of Kalman filtering of measurements from an inertial measurement unit (IMU) to provide information on the orientation. The performances of each filter are evaluated in terms of the roll, pitch, and yaw angles. In this thesis, I had made an entire required analysis, design circuit, output and input data measurement and other important parameters to develop the data filtering of 5-axis IMU that can be implemented by using Kalman filter method. Simulation with constructed data has been done to verify the algorithm. Also the sensor errors and their effects are discussed. Furthermore the strategy for calibration, initialization and alignment for the system is proposed. On the other hand, this thesis is aim to provide objective and scope of the research, the literature review study, research methodology, and fabrication process with result analysis and conclusion as part requirement in submitted the thesis to FYP supervisor.

ABSTRAK

Tesis ini bertujuan untuk merekabentuk dan membangunkan data penapisan Unit Pengukuran inersia 5-paksi (IMU) menggunakan kaedah Kalman Filter. Projek ini berusaha untuk mengesahkan bahawa data dari 5DOF IMU boleh ditapis menggunakan kaedah Kalman Filter supaya ia boleh digunakan sebagai satu algoritma dalam jajaran gerakan. IMU ini terdiri daripada 2-paksi giroskop dan 3-paksi pecutan. Penapis Kalman adalah satu set persamaan matematik yang menyediakan pengiraan cekap (rekursi) bermaksud untuk menganggarkan keadaan proses, dengan cara yang meminimumkan min ralat kuasa dua. Sumbangan utama algoritma ini adalah sejajar dengan pendekatan dalam gerakan dengan syarat awal tidak diketahui. Kajian ini meneroka penggunaan Kalman penapisan ukuran dari satu unit ukuran inersia (IMU) untuk memberi maklumat mengenai orientasi. Prestasi setiap penapis dinilai dari segi daftar, padang, dan sudut rewang. Dalam tesis ini, saya telah membuat keseluruhan analisis yang diperlukan, reka bentuk litar, output dan pengukuran data input dan parameter lain yang penting untuk membangunkan data penapisan IMU 5-paksi yang boleh dilaksanakan dengan menggunakan kaedah penapis Kalman. Simulasi dengan data yang dibina telah dilakukan untuk mengesahkan algoritma. Kesilapan sensor dan kesannya juga turut dibincangkan. Tambahan pula strategi untuk penentukuran, pengawalan dan penyelarasan untuk sistem turut dicadangkan. Selain daripada itu, karya ini adalah bertujuan untuk menyediakan objektif dan skop penyelidikan, kajian kajian literatur, kaedah penyelidikan, dan proses fabrikasi dengan hasil analisis dan kesimpulan sebagai keperluan bahagian dalam mengemukakan tesis kepada penyelia projek sarjana muda.

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION

CHAPTER 2 LITERATURE REVIEW

CHAPTER 3 RESEARCH AND METHODOLOGY

CHAPTER 4 RESULT AND ANALYSIS

CHAPTER 5 CONCLUSION AND RECOMMENDATION

LIST OF TABLE

LIST OF FIGURES

LIST OF SYMBOLS

h	Depth
p	Pressure
ρ	Water density
g	Gravitational acceleration
x	Roll
y	Pitch
z	Yaw
v	Velocity
X_w	World coordinates system
X_b	Body-fixed coordinates system
φ	Phi
ψ	Psi
θ	Theta
F	Force
m	Mass
a	Acceleration
q	Unit of quaternion
Φ	Magnitude of rotation
U	Normalized axis of rotation
H	Measurement sensitivity matrix or observation matrix
$H\hat{x}_k(-)$	Predicted measurement
$\overline{K}_k$	Kalman gain
$P_k(-)$	Predicted or a priori value of estimation covariance
$P_k(+)$	Corrected or a posteriori value of estimation covariance
Q_k	Covariance of dynamic disturbance noise
R	Covariance of sensor noise or measurement uncertainty

$\hat{x}_k\ (-)$	Predicted or a priori value of the estimated state vector
$\hat{x}_k\ (+)$	Corrected or a posterioroi value of the estimated state vector
Z	Measurement vector or observation vector

LIST OF ABBREVIATION

IMU	Inertial Measurement Unit
5DOF	5 Degree of Freedom
KF	Kalman Filter
INS	Inertial Navigation System
Nav	Navigation
AHRS	Attitude and heading reference system

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The effectiveness of navigation and controls of an air vehicle are highly dependent on the degree of precision of the on-board inertial measurement unit (IMU) (Shiau, Huang and Chang, 2012). The IMU is a single unit in the electronics module which collects angular velocity and linear acceleration data which is sent to the main processor (IMU, http://www.ssl.umd.edu/projects/RangerNBV/thesis/2-4-1.htm, 2013). The sensors in the IMU are two rate gyros and three accelerometers. With perfect gyro measurements, the estimate of the orientation could be determined quite accurately; however, using real sensors, the error in the estimate grows with time due to quantization, integration, and sensor errors (Kim and Golnaraghit, 2004).

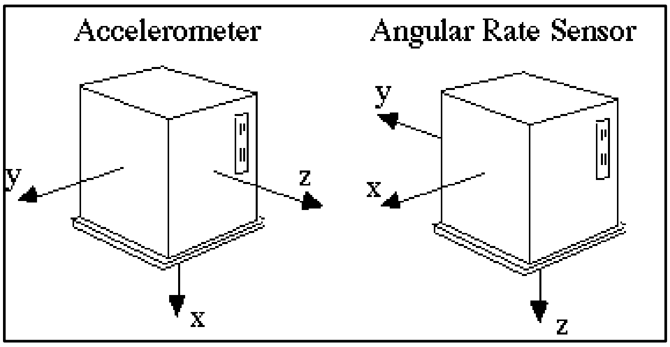


Figure 1.1: Initial IMU sensor coordinate axes [7]

An IMU works by detecting the current rate of acceleration, as well as it changes in rotational attributes, including pitch, roll and yaw. This data is then fed into a computer, which calculates the current speed and position, given a known initial speed and position (Hazry, Sofian, and Zul Azfar, 2012).

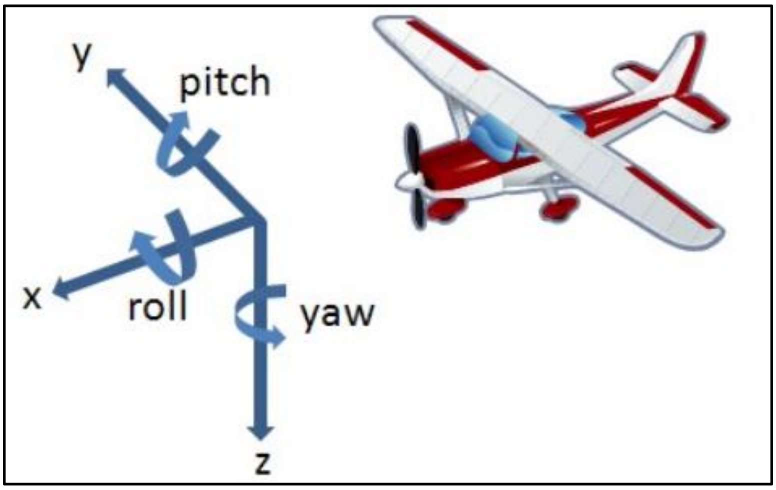


Figure 1.2: An IMU sensor measures linear acceleration and orientation [26]

IMU available in market now are in various types and shape. So, user can select what type, size and shape. The IMU can be selected from its degrees of freedom (DOF) that being developed by manufacturer. For five DOF, the sensors configurations are three accelerometers and two gyroscopes that measure pitch and roll.

1.2 PROBLEM STATEMENT

An inertial measurement unit, or IMU, is the main component of inertial guidance systems used in air space, and watercraft, including guided missiles. Driven by their low cost and small size, MEMS inertial sensors have been used to produce low cost INS that can be widely adopted in several navigation applications (Elkhidir, Shuhimi, Musa and Satti, 2011). An IMU works by sensing motion including the type, rate, and direction of that motion using a combination of accelerometers and gyroscopes (Hazry, Sofian and Zul Azfar, 2012). Therefore, the noise level at the output of MEMS-based inertial sensors must be reduced and the sensor errors separate from motion dynamics prior to processing their measurements by the KF module. One possible way of reducing high-frequency sensor noise is to use a low-pass filter. However, discrete low-pass filters result in inherent time delays when applied to system in real time, which could reduce the stability of the system (Angelosanto, 2008).

Low-pass filters also do not address the issues of inaccurate or unavailable measurements. In inertial navigation systems, two to three integrations are required to get from sensor outputs to a position solution. This is the major disadvantage of dead reckoning and inertial navigation; the fact that one or more integrations are required to convert sensor outputs to a position solution means that errors in sensor outputs lead to position errors that grow with time (Demoz Gebre-Egziabher, 2010). In this project, an inertial navigation system based on low-cost IMU sensor will be developed.

1.3 PROJECT OBJECTIVES

This project is intended to design and develop a data filtering of 5-axis IMU using Kalman Filter. The IMU use is low cost, high integrity and their board enables to easily incorporate roll, pitch, and tilt measurements for navigation application. The 5DOF-IMU data filtering is based on the Kalman filter method which is from a mathematical concept with a deep relationship with the foundations of algebra and number theory. In summary, the objectives of this project:

1. The main task of this project is to design and develop data filtering of 5-axis Inertial Measurement Unit (IMU) using Kalman Filter.

2. From all of the method used, verify the 5-axis inertial measurement unit can be filtered using Kalman Filter method.

1.4 SCOPE OF THE PROJECT

The main task of this work is to analyse several method in filtering data of 5-axis IMU but focusing on Kalman Filter method. A comparison between the results obtained will be performed on the basis of accuracy, complexity of the algorithm and ease of implementation on the embedded system. SIMULINK and MATLAB also will be used in this task to promote an understanding of the real system. In summary, the scope of this project is:

1. Design and modelling data filtering of 5-axis IMU using Kalman Filter via MATLAB.

2. In this project, programming and real circuit for sensor setup also been developed.

CHAPTER 2

LITERATURE REVIEW

2.0 INTRODUCTION

This chapter will focused on the historical background for the data filtering of 5-axis Inertial Measurement Unit (IMU) using Kalman Filter and other method involved in this study. This chapter deal with the previous work done which are reviewed from journal, website, books and other related resources. In this chapter, the basics of inertial navigation system and comparison between gimballed system and an underwater strap down inertial navigation is been covered.

2.1 NAVIGATION

Navigation can be defined as determination of a physical body's position and velocity relative to some reference coordinate frame (Anwar, 2010). A simple one dimensional example of navigation is determining the position and speed of a train between two points on a track. The basic concept of navigation is latitude and longitude. All navigation techniques can be viewed as processes in which the mathematical operation of integration is performed on the sensor outputs to yield position (Gebre-Egziabher, 2004). Navigation systems need different kinds of information: position from global or relative coordinates, as well as direction which is an orientation angles (Damian, 2011). This information is provided by the sensors whose data must be fused.

2.2 INERTIAL NAVIGATION SYSTEM

An inertial navigation system (INS) continuously calculates the position, orientation, and velocity (direction and speed of movement) of a moving object without the need for external references (Anwar, 2010). The calculation can be performing by using a computer, motion sensors (accelerometers) and rotation sensors (gyroscope) by using computer.

A block diagram in figure 2.1 shows an inertial measurement unit (IMU) combined with navigation equations to form an inertial navigation system (INS). A handful of useful tricks have been used to align blocks and arrows nicely. Hard coding coordinates has been avoided as much as possible.

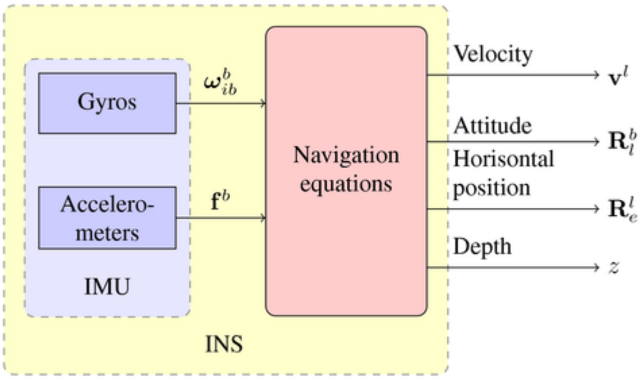


Figure 2.1: Inertial Navigation System [20]

The INS consists of 3-axis gyroscopes which give the system computer the roll, pitch and yaw rates about the body axes. Inertial navigation is a self-contained navigation technique in which measurements provided by accelerometers and gyroscopes are used to track the position and orientation of an object relative to a known starting point, orientation and velocity (Woodman, 2007).

An INS consists of the following:

- An IMU
- Instrument support electronics
- Navigation computers (one or more) calculate the gravitational acceleration (not measured by accelerometer) and doubly integrate the net acceleration to maintain an estimate of the position of the host vehicle

2.3 TYPES OF INERTIAL NAVIGATION SYSTEM

There are many different designs of INS with different performance, characteristics, but there are two types of inertial navigation system that can be compared. Their difference in mechanical design gives result in different requirements of the electronic and software. Inertial navigation is a self-contained navigation technique in which measurements provided by accelerometers and gyroscopes are used to track the position and orientation of an object relative to a known starting point, orientation and velocity (Woodman, 2007). Inertial measurement units (IMUs) typically contain three orthogonal rate-gyroscopes and three orthogonal accelerometers, measuring angular velocity and linear acceleration respectively.

2.3.1 Strapdown Inertial Navigation System

Traditionally, the general displacement of a rigid body in a strap down inertial navigation system (INS) are modeled and analyzed separately, i.e., the direction cosine matrix or quaternion for the analysis of rotation and translation analysis for vector. The strap down INS emerged about half a century ago as an alternative in which the gyro and accelerometer triads are directly mounted on the vehicle (Wu, Hu, Wu and Hu, 2004). The gyro outputs are used to maintain a digital reference frame, into which the specific force measurements from accelerometer triads are resolved and then double integrated to acquire velocity and position. The choice of the reference frame usually depends on the convenience of application.

Inertial sensors for strap down systems experience much higher rotation as compared to their gimballed counterparts. Rotation introduces error mechanisms that require attitude rate-dependent error compensation (Woodman, 2007).

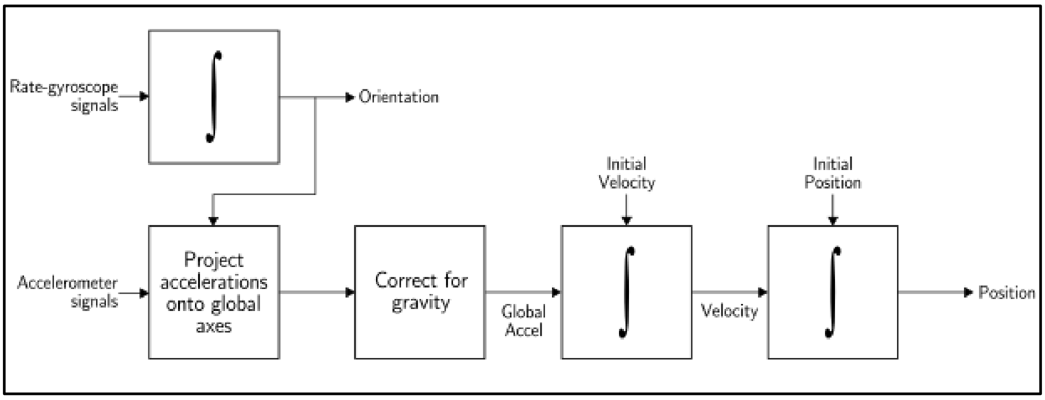


Figure 2.2: Strapdown inertial navigation algorithm [21]

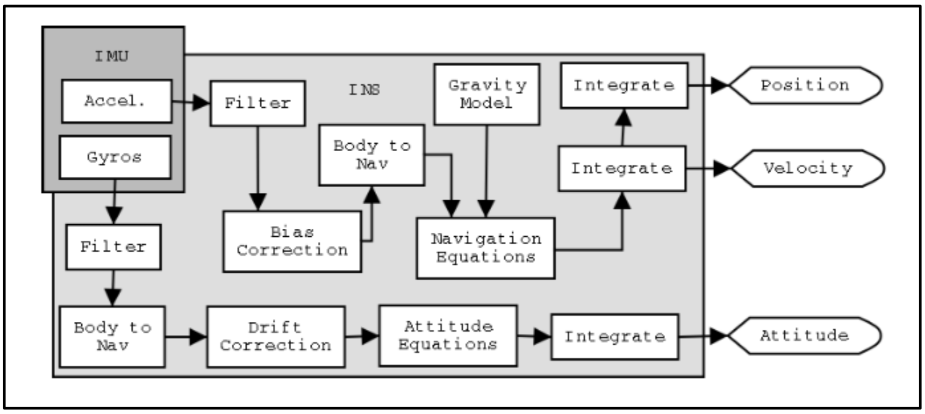


Figure 2.3: A flow chart of a strap-down INS which takes acceleration and rotation rates from the IMU and procedures position, velocity, and attitude of the system [24]

In strap down systems the inertial sensors are mounted rigidly onto the device, and therefore output quantities measured in the body frame rather than the global frame. To keep track of orientation the signals from the rate gyroscopes are ‘integrated’. To track position the three accelerometer signals are resolved into global coordinates using the known orientation, as determined by the integration of the gyro signals. The global acceleration signals are then integrated as in the stable platform algorithm.

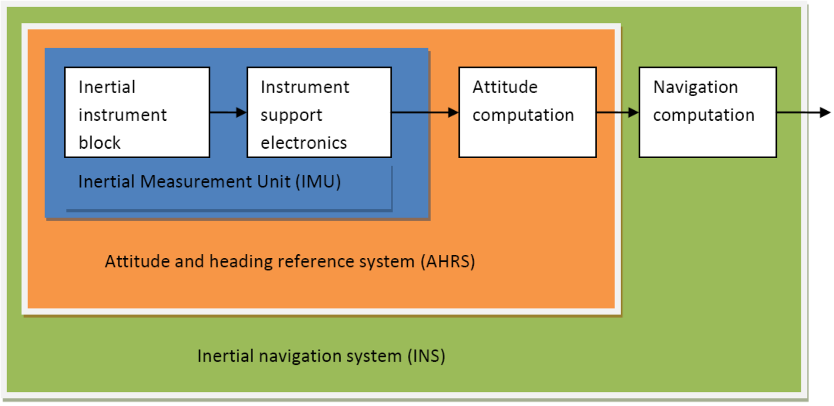


Figure 2.4: Strap down INS building blocks [16]

The advantage of a strap down system is the cost. Compared to gimbal, the cost of replicating software is relatively small. However, cost for calibrating and testing requires a precision rate table. Since there is no isolation from the body's rotation, to limit the accuracy, gyros with high range have to be chosen.

2.3.2 Gimballed System

The first type of INS developed was a gimballed system. The accelerometers are mounted on a motorized gimballed platform which was always kept aligned with the navigation frame (Woodman, 2007) (Walchko and Dr Mason, 2002). Pickups are located on the outer and inner gimbals which keep track of the attitude of the stabilized platform relative to the vehicle on which the INS is mounted.

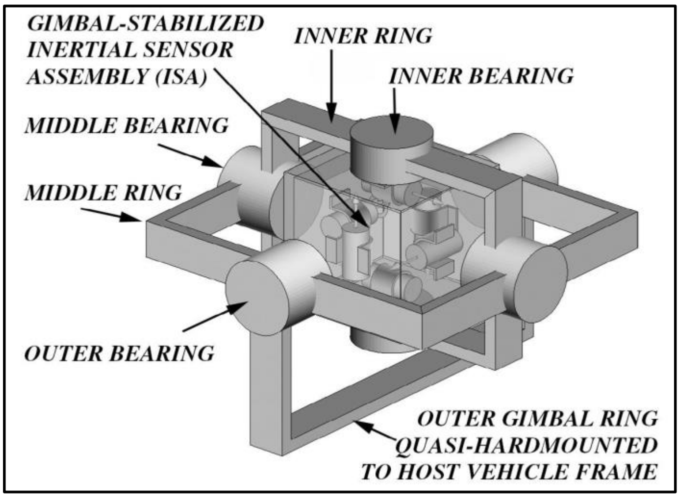


Figure 2.5: A gimballed IMU [12]

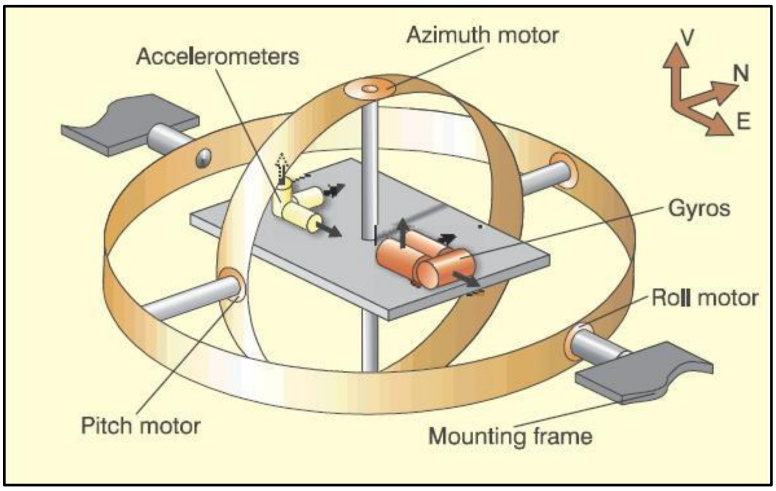


Figure 2.6: Gimballed inertial platform [17]

There is an advantage by using a gimballed system compared to a strap down system. Gimballed system eliminates many rate-dependant sensor errors and allows for higher accuracy sensors (Anwar, 2010). This might be happen from the isolation of the inertial sensors from high rates of. On the other hand, the complexity of the mechanical design which is bounded with problem like friction and reliability become a disadvantage of using this system. Bigger in size and higher in cost of the needs of motors and control electronics is disadvantages of gimballed system use.

2.4 COORDINATE SYSTEM

The relationships between data expressed can be considered in two different coordinate systems:

- The world coordinate system is fixed in inertial space. The origin of this coordinate system is denoted X_w.
- The body-fixed coordinate system is rigidly attached to the object whose attitude we would like to describe. The origin of this coordinate system is denoted X_b.

Depending on application, the coordinate system could be defined in a different way. Some application uses left hand coordinate system.

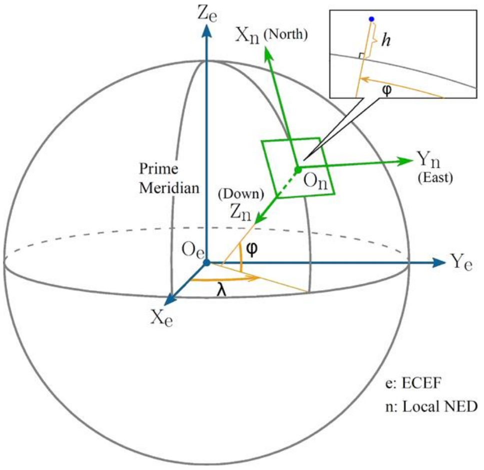


Figure 2.7: The Geodetic Coordinate System [5]

2.5 EULER ANGLE TRANSFORMATION

A rotation matrix can be built from three matrices representing rotation around the axes of the local coordinate system, where each rotation is defined by an angle (Anwar, 2010). Some authors refer the angles as yaw, pitch and roll. The order of multiplication of the matrices is important since matrix multiplication is not commutative. A rotational coordinate transformation delivers different column matrices for the same vector **x**.

Rotation about the x-axis from CS: $x_2 = X_{21}x_1$

Rotation about the y-axis from CS: $x_2 = Y_{21}x_1$

Rotation about the z-axis from CS: $x_2 = Z_{21}x_1$

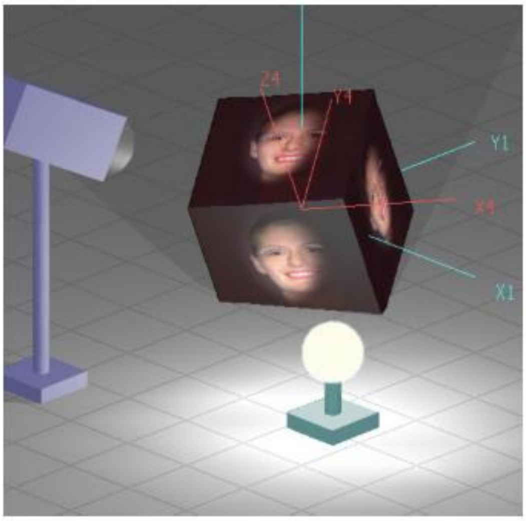


Figure 2.8: Euler Angle Coordinate Rotations [33]

$$X_{21} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \tag{2-1}$$

$$Y_{21} = \begin{bmatrix} \cos(\beta) & 0 & -\sin(\beta) \\ 0 & 1 & 0 \\ \sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \tag{2-2}$$

$$Z_{21} = \begin{bmatrix} \cos(\gamma) & \sin(\gamma) & 0 \\ -\sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2-3}$$

The rotation about the y-axis has a different sign pattern. Compound matrix rotations about three axes depend on the sequence.

2.6 QUATERNIONS

Quaternion was first introduced by Sir W. R. Hamilton in 1843. Quaternion is a relationship with a foundation of algebra and number of theory. A quaternion can be thought of as a vector with four components, as a composite of a scalar and an ordinary vector, or as a complex number with three different imaginary parts (Berthold, 1986). Quaternions (as a mathematical method) is an extension, or improvement, of Cartesian geometry, in which the artifices of coordinate axes, &c., are got rid of, all directions in space being treated on precisely the same terms (Peter Guthrie Tait, 1886). Furthermore, it is easily to construct the corresponding quaternion from a given axis and angle, and conversely read off the axis and the angle from a given quaternion.

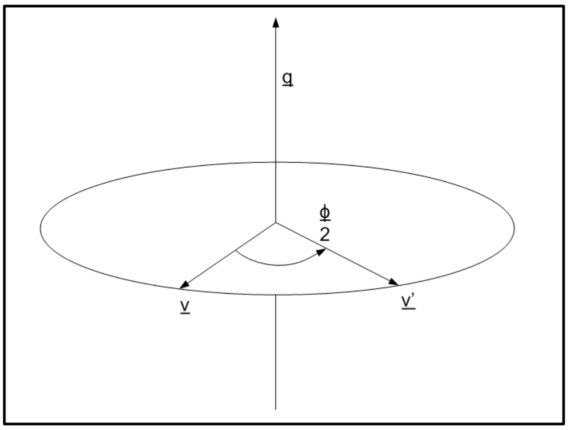


Figure 2.9: Rotation with quaternion [16]

Basically quaternion has 4 components, where it can be represented as follow:

$$q = [q_0 \;\; q_1 \;\; q_2 \; q_3] \tag{2-4}$$

Each of these imaginary dimensions has a unit value of the square root of -1, but they are different square roots of -1 all mutually perpendicular to each other, known as i, j and k. Of the 4 components, one is real scalar number and the other three forms a vector in imaginary ijk space. Multiplication of quaternions can be defined in terms of the products of their components:

$$q = q_0 + iq_1 + \; jq_2 + kq_3 \tag{2-5}$$

$$i^2 = \; j^2 = k^2 = ijk = -1 \tag{2-6}$$

$$i = \; jk = -kj \tag{2-7}$$

$$j = \; ki = -ik \tag{2-8}$$

$$k = \; ij = -ji \tag{2-9}$$

[In an article by Prof. F. Klein (Math. Ann. LI. 1898) a claim is somewhat obscurely made for Gauss to a share, at least, in the invention of Quaternions. Full information on the subject is postponed till the publication of Gauss' Nachlass, in Vol. VIII. of his Gesammelte Werke. From the article mentioned above, and from a "Digression on Quaternions" in Klein und Sommerfeld Ueber die Theorie des Kreisels (p. 58), this claim appears to rest on some singular misapprehension of the nature of a Quaternion:—whereby it is identified with a totally different kind of concept, a certain very restricted form of linear and vector Operator. 1899.]

A quaternion, $q \in H$ can be represented by a vector,

$$q = [q_0, \; q_1, \; q_2, \; q_3]^T = \begin{bmatrix} q_0 \\ q_{1:3} \end{bmatrix} \tag{2-10}$$

Along with a set of additional definitions and operations that may be applied to it. The ad joint, norm, and inverse of the quaternion, q are

$$\bar{q} = \begin{bmatrix} q_0 \\ -q_{1:3} \end{bmatrix} \tag{2-11}$$

$$\|q\| = \sqrt{{q_0}^2 + {q_1}^2 + {q_2}^2 + {q_3}^2} \tag{2-12}$$

$$q^{-1} = \frac{\bar{q}}{\|q\|} \tag{2-13}$$

With q4 being the real (scalar) parameter of the quaternion and q_1, q_2, q_3 being the imaginary (vectorial) parameters of the quaternion, the parameters are defined as;

$$q_1 = \sin\left(\frac{\Phi}{2}\right) \tag{2-14}$$

$$q_2 = \sin\left(\frac{\Phi}{2}\right) \tag{2-15}$$

$$q_3 = \sin\left(\frac{\Phi}{2}\right) \tag{2-16}$$

$$q_4 = \cos\left(\frac{\Phi}{2}\right) \tag{2-17}$$

Rotation not the angle between vectors does not change the length of a vector. When add a scalar and a 3D vector, quaternion can be built. The 3D vector represents the normal vector to a rotation plane without angle of $\frac{\Phi}{2}$.

The normal vector to a rotation plane with rotation angle of the normal vector or the axis of rotation is given by:

$$q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{2-18}$$

Vector v is rotated about $\frac{\Phi}{2}$ to v'.

Quaternion also has a conjugate quaternion, q^* just like complex numbers;

$$q^* = \begin{bmatrix} -q_1 \\ -q_2 \\ -q_3 \\ -q_4 \end{bmatrix} \tag{2-19}$$

To find the inversed quaternion, this formula is used:

$$q^{-1} = \frac{q^*}{|q|} \tag{2-20}$$

So in case of a unit quaternion, the inversed is just the conjugate quaternion,

$$q^{-1} = q^* \tag{2-21}$$

This is very convenient since unit quaternions (quaternions with length of 1) will always be used in dealing with rotations in 3D. The multiplication of two quaternions is associative but non commutative.

2.6.1 Rotation Operation with Quaternion

To rotate a vector using quaternion in 3D space requires a special formula;

$$v' = q.v.q^* \tag{2-22}$$

This formula is called sandwich multiplication. It is quite unusual that the multiply the vector with the quaternion need to be done first, and then multiply the result with the quaternion conjugate.

2.6.2 Quaternion Calculation from Gyro Data

Once have the rotation angles, by integrating the gyro signal.

$$Rot.Angle = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{2-23}$$

Then the magnitude of rotation, Φ will be given as

$$\Phi = \sqrt{X^2 + Y^2 + Z^2} \tag{2-24}$$

The normalized axis of rotation, Uwill be given as

$$U = \frac{(iX+jY+kZ)}{\Phi} \tag{2-25}$$

The quaternion, q will be:

$$q = \cos\left(\frac{\Phi}{2}\right) + i\sin\left(\frac{\Phi}{2}\right).\left(\frac{X}{\Phi}\right) + j\sin\left(\frac{\Phi}{2}\right).\left(\frac{Y}{\Phi}\right) + k\sin\left(\frac{\Phi}{2}\right).\left(\frac{Z}{\Phi}\right) \qquad (2\text{-}26)$$

If the sampling rate is high, the approximation for small angles,

$$q = 1 + i\left(\frac{X}{2}\right) + j\left(\frac{Y}{2}\right) + k\left(\frac{Z}{2}\right) \qquad (2\text{-}27)$$

The X, Y and Z must be in radian.

2.6.3 Orientation Tracking Using Quaternion Multiplication

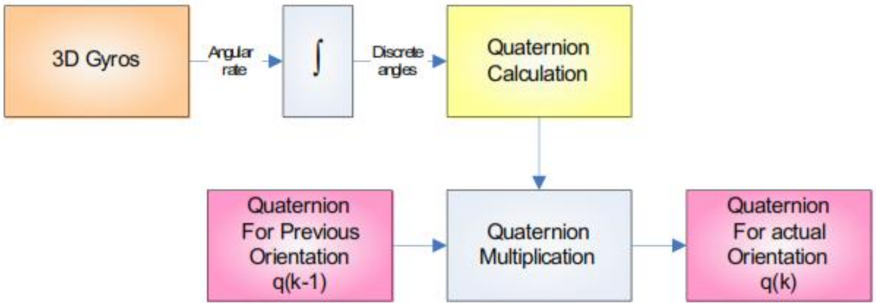


Figure 2.10: Orientation tracking using quaternion multiplication [16]

If q_{rot1} is the first rotation quaternion, the first rotation result q_1 is equal to:

$$q_1 = q_0 . q_{rot1} \qquad (2\text{-}28)$$

The second rotation result q_2 is

$$q_2 = q_0 . q_{rot1} . q_{rot2} \tag{2-29}$$

The $n - th$ rotation result q_n is

$$q_n = q_0 . q_{rot1} . q_{rot2} \cdots q_{rotn} \tag{2-30}$$

In the case where the initial orientation is aligned to the Global coordinate system,

$$q_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{2-31}$$

If the initial orientation is not aligned to Global, q_0 must be determined using self-alignment method.

CHAPTER 3

RESEARCH AND METHODOLOGY

3.1 INTRODUCTION

This chapter will discuss on the method used in order to produce an orientation algorithm of the digital compass. In this chapter, there are flow charts which show the flow of the project. There is also a block diagram for an orientation algorithm of digital compass using quaternion method included in this chapter with a corresponding process. The project flow chart is shown in figure 3.1 while block diagram is shown in figure 3.2 on the next page. The methods and algorithms to be tested come from the research mentioned above with some improvements made to make them fit into the embedded system or to enhance performance.

3.2 FLOW CHART

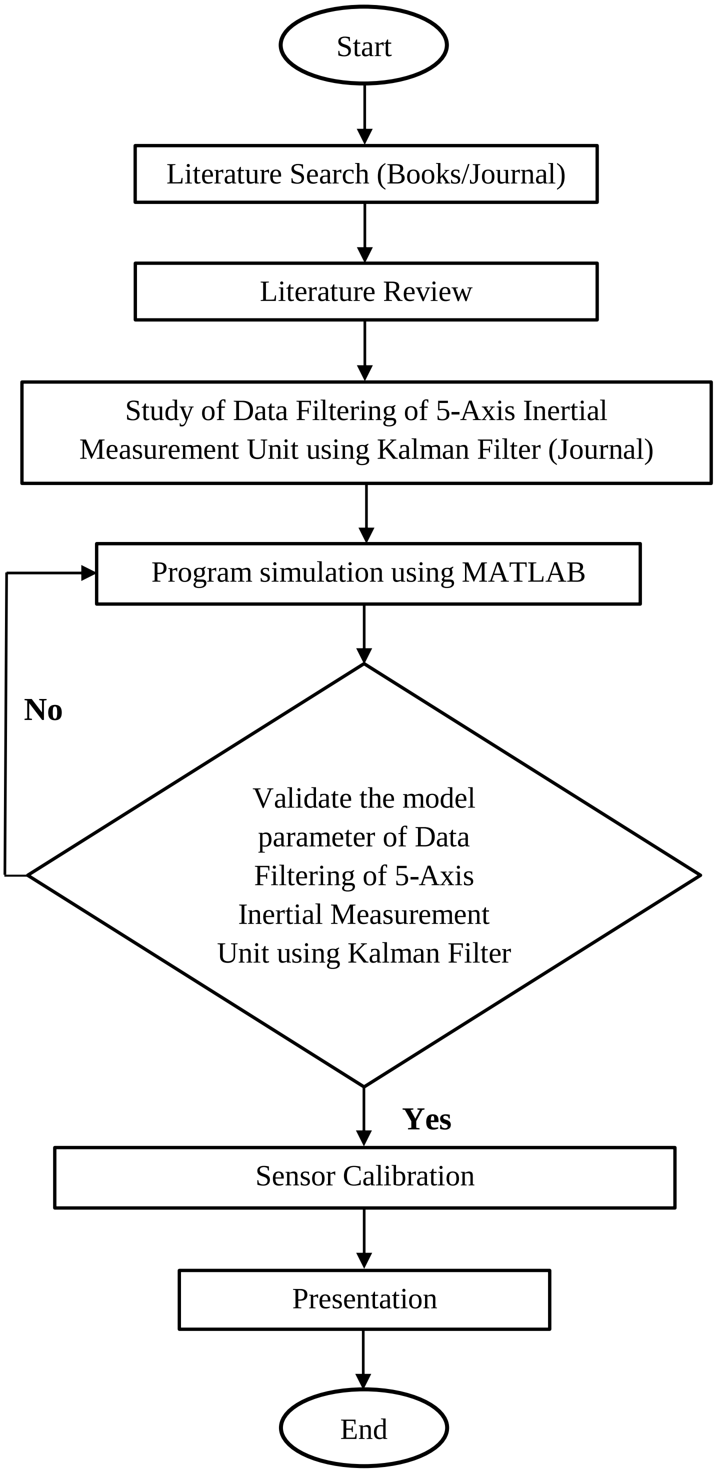


Figure 3.1: Project Flow Chart

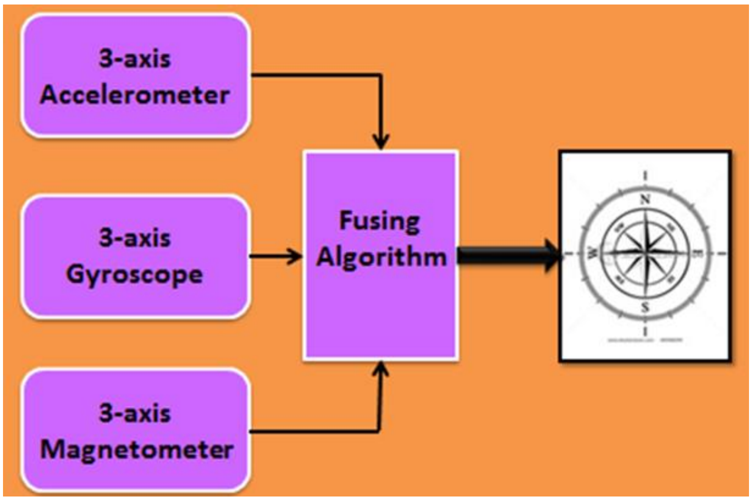


Figure 3.2: Block diagram of a digital compass developing

3.3 SIMULATION OF THE CORRECTION ALGORITHM

In order to get information of system behaviour as in reality, the simulation will be used as it could be defined as a manipulation of a system in such a way that operates on time or space. At some particular point in time or space, a model is defined as a simplified representation of a system intended to promote understanding of the real system.

Generally simulation has three purposes:

- Describe the behaviour of systems
- Construct theories or hypotheses that account for the observed behaviour
- Use these theories to predict future behaviour, that is, the effects that will be produced by changes in the system or in its method of operation.

In this project, the simulation of the algorithm plays an important role over the other tasks. Some of the purposes are, to prove the functionality of the algorithm

- as part of the iterative process in optimizing the algorithm (simulate -> analysis -> optimize)
- to detect any possible error in the system
- to give information on hardware requirements such as sensor range, resolution and sampling rate

3.4 TOOLS FOR SIMULATION AND DATA ANALYSIS

In this project, MATLAB and SIMULINK will be used in order to create a simulation on the algorithm. MATLAB provides an interactive environment with hundreds of reliable and accurate built-in mathematical functions (Saadat, 2002). These functions provide solutions to a broad range of mathematical problems including matrix algebra, complex arithmetic, linear systems, differential equations, signal processing, optimization, nonlinear systems, and many other types of scientific computations. The most important feature of MATLAB is its programming capability, which is very easy to learn and to use, and which allows user-developed functions.

MATLAB has been enhanced by the very powerful SIMULINK program. SIMULINK is a graphical mouse-driven program for the simulation of dynamic systems (Saadat, 2002). SIMULINK enables students to simulate linear, as well as nonlinear, systems easily and efficiently.

3.5 ORIENTATION DETERMINATIONS IN 3D SPACE

For a gimballed system, the orientation of the IMU is always equal to the Global coordinate system so that there is no need to trace the orientation of the moving body (Anwar, 2010). However in a strap down system, it is necessary to find the actual orientation of the body and transfer the accelerations to the Global coordinate system.

3.6 KALMAN FILTERING

The Kalman filter is widely used in aeronautics and engineering for two main purposes which is for combining measurements of the same variables but from different sensors, and also for combining an inexact forecast of a system's state with an inexact measurement of the state (Rojas). The Kalman filter has also applications in statistics and function approximation. The Kalman Filter is a set of mathematical equations that provides an efficient computational (recursive) means to estimate the state of a process, in a way that minimizes the mean of the squared error (Welch and Bishop, 2006). Foxlin has developed a Kalman Filter estimator for a head- mounted inertial orientation tracker (Foxlin, 1996) using rate gyros in triad formation, a fluid-filled inclinometer, and a magnetic compass. The differential equation for an Euler angle sequence was integrated into the process model:

$$\begin{bmatrix} \dot{\Psi} \\ \dot{\theta} \\ \dot{\emptyset} \end{bmatrix} = \begin{bmatrix} 1 & \tan(\theta)\sin(\Psi) & \tan(\theta)\cos(\Psi) \\ 0 & \cos(\Psi) & -\sin(\Psi) \\ 0 & \sec(\theta)\sin(\Psi) & \sec(\theta)\cos(\Psi) \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{3-1}$$

The body-fixed sequence ***z, y,*** **x** was used, to represent yaw, pitch, and roll, respectively. This model was used to develop a nonlinear complementary Kalman Filter, a Kalman Filter designed to estimate the state errors, rather than the states themselves (Kim and Golnaraghit, 2004).

A Kalman filter for an INS containing accelerometers and rate gyroscopes might include accelerations and rotation rates to which these instruments respond. The equations used for the variables and parameters of the Kalman filter are shown in the Table 3.1 on the next page.

Predictor (Time or Temporal Updates) Predicted state vector	$\hat{x}_k(-1) = \Phi_k\hat{x}_{k-1}(+)$
Predicted covariance matrix	$P_k(-) = \Phi_k P_{k-1}(+)\Phi_k{}^T + Q_{k-1}$
Corrector (Measurement or Observational Updates) Kalman Gain:	$\overline{K}_k = P_k(-)H_k{}^T(H_kP_k(-)H_k{}^T + R_k)^{-1}$
Corrected state estimate	$\hat{x}_k(+) = \hat{x}_k(-) + \overline{K}_k(z_k - H_k\hat{x}_k(-))$
Corrected covariance matrix	$P_k(+) = P_k(-) - \overline{K}_kH_kP_k(-)$

Table 3.1: Essential Kalman Filter Equations [32]

3.7 SIGNAL MODELING

The sensor is interest to calculate heading of 3-axis accelerometer, and 2 axis gyroscopes. The desired output of the fusion algorithms is an azimuth angle, which must be derived from the digital readings of the sensors (Titterton and Weston, 2004). However, the ranges, resolutions and binary format are not homologized among these sensors.

3.8 SENSOR FUSION

Sensor fusion is a combination of different types of sensors to reach a better performance than possible with a single sensor. The multi-sensor data fusion methods are usually based in Bayesian inference formulation, upon which data fusion algorithms can be created and analysed. Data fusion is encountered when dealing with state estimation. Fusing accelerometer and magnetometer data allows for attitude calculation (Kesten, 2010).

Depending on the manner in which the information is combined, the fusion algorithms are classified in two general types. In weakly coupled algorithms the operation of the different sensory processing modules is not affected by the fusion process; in strongly coupled algorithms the output of a model interact with other modules and affects their operation.

3.9 MICROELECTROMECHANICAL SYSTEMS (MEMS)

MEMS technology is of particular interest at the current time since it offers rugged, low cost, small and lightweight inertial sensors relative to the other available technologies. The performance of MEMS inertial devices is also improving rapidly. Recent improvements in the performance of small and lightweight micro machined electromechanical systems (MEMS) inertial sensors have made the application of inertial techniques to such problems possible.

3.10 INERTIAL MEASUREMENT UNIT (IMU)

An inertial measurement unit (IMU) or inertia reference unit (IRU) contains a cluster of sensors which is accelerometers (three or more, but usually three) and gyroscopes (three or more, but usually three) (Han and Wang). These sensors are rigidly mounted to a common base to maintain the same relative orientation.

Figure 3.3: IMU 5 Degrees of Freedom [34]

The Inertia Measurement Unit used in this project is a combining of the IDG500 and ADXL335 sensors. The IMU board enables to easily incorporate roll, pitch, and tilt measurements into this project. It allows an unheard of 5 axis of sensing (Roll, Pitch, X, Y, Z) in less than 1 square inch, and under 2 grams. The IMU board uses a standard 0.1" footprint and includes all outputs from both the IDG500 Gyro and ADXL335 Accelerometer ICs. The dimension for this IMU 5 Degrees of Freedom is 0.75"x0.9" (20x23mm).

3.10.1 Gyroscope

A gyroscope is a device for measuring or maintaining orientation, based on the principles of angular momentum. It is one of the primary sensors that comprise the IMU. Mechanically, a gyroscope is a spinning wheel or disk in which the axle is free to assume any orientation (Woodman, 2007). Traditional units, such as gimballed gyroscopes, laser gyroscopes, and fibre-optic gyroscopes provide high-precision angular rate information for navigation and control systems (Shiau, Huang and Chang, 2012). For low cost gyros, such as MEMS gyros, the drift rate can reach the level of several degrees per second; even for the best gyros, the accumulated errors cannot be ignored (Han and Wang).

Although this orientation does not remain fixed, it changes in response to an external torque much less and in a different direction than it would without the large angular momentum associated with the disk's high rate of spin and moment of inertia (Wikipedia, 2012). To distinguish between rotation and translation, gyroscopes are often used. A traditional gyroscope, or gyro for short, consists of a rotor that spins freely to maintain angular momentum and therefore its original orientation (Tsai, Tu, Bae and Chou).

3.10.2 Accelerometer

Accelerometers are sensors for measuring inertial acceleration, also called specific force to distinguish it from what we call "gravitational acceleration" (Grewal, Weill and Andrews, A John Wiley & Sons, Inc., Publication). Accelerometers do not measure gravitational acceleration, which is perhaps more accurately modelled as a warping of the space time continuum in a gravitational field. An accelerometer in free fall (e.g., in orbit) in a gravitational field has no detectable input.

What accelerometers measure is modelled by Newton's second law as a = F/m, where F is the physically applied force (not including gravity), m is the mass it is applied to, and specific force is the ratio F/m. A triaxial accelerometer alone can measure three degrees of freedom. If an object can be assumed to have either zero translation or zero rotation, then a single triaxial accelerometer suffices for measuring either rotation or translation, respectively (Tsai, Tu, Bae and Chou).

3.11 ARDUINO UNO R3

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started (Aduino Uno, http://www.arduino.cc/en/Main/ArduinoBoardUno. March, 2013).

Figure 3.4: Arduino Uno R3 Front [36]

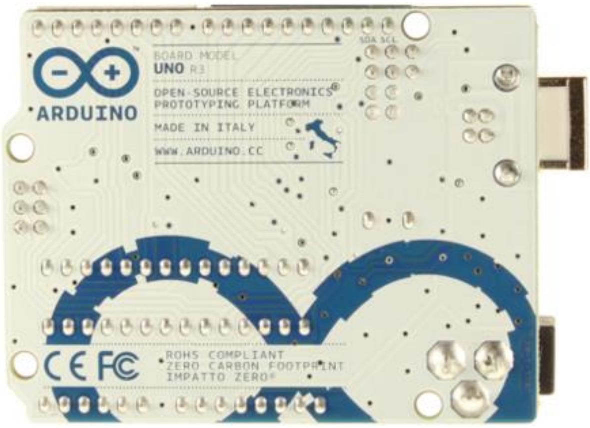


Figure 3.5: Arduino Uno R3 Back [36]

CHAPTER 4

RESULT AND ANALYSIS

4.1 INTRODUCTION

After completed all of the method from the previous chapter, this project will be analysed so the objective of this project can be achieved. In this chapter, the 5-Axis Inertial Measurement Unit (IMU) has been move or in other word it can be move, so analysis process can be made. The main objective of this chapter is to analyse the data filtering of 5-axis IMU using Kalman Filter. In the next section, I will discuss about the results obtained from this project.

4.2 TEMPERATURE EFFECTS ON GYROSCOPE

In Figure 4.1 until Figure 4.3 below show the standard temperature effect on the gyroscope’s bias measurement and linear sensitivity’s error of the sensor ADXL335. Rather than using additional temperature compensation circuitry, innovative design techniques ensure that high performance is built in to the ADXL335. As a result, there is no quantization error or non-monotonic behaviour, and temperature hysteresis is very low (typically less than 3 mg over the -25°C to +70°C temperature range).

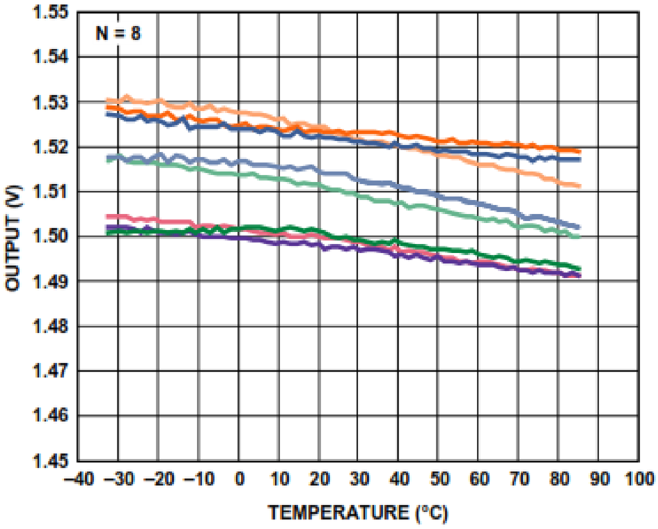


Figure 4.1: X-Axis Zero g Bias vs. Temperature – Eight Parts Soldered to PCB

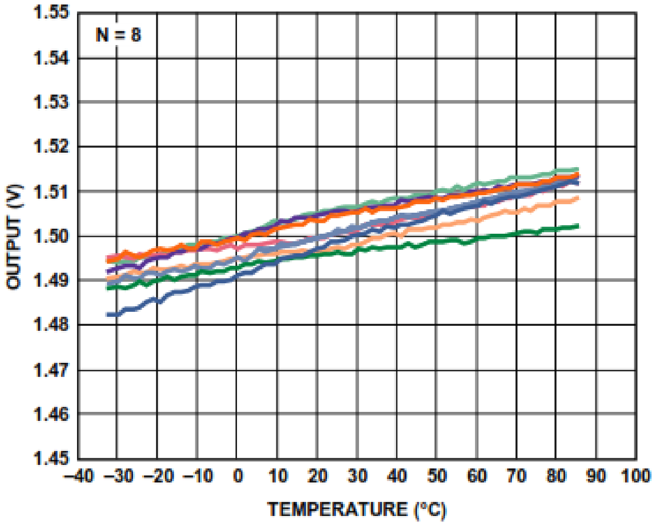


Figure 4.2: Y-Axis Zero g Bias vs. Temperature – Eight Parts Soldered to PCB

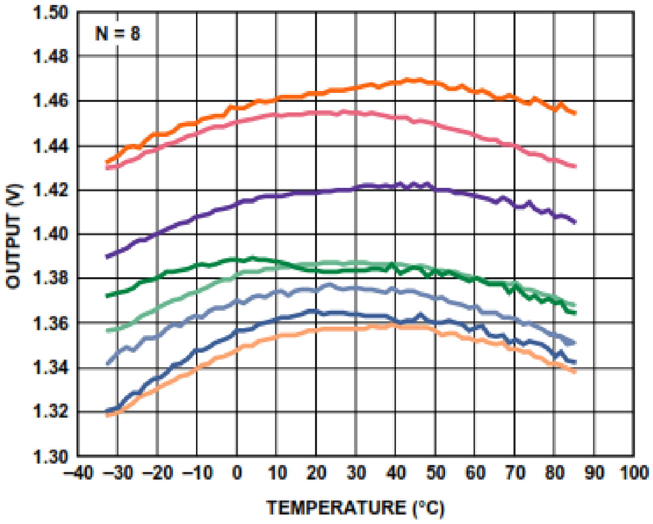


Figure 4.3: Z-Axis Zero g Bias vs. Temperature – Eight Parts Soldered to PCB

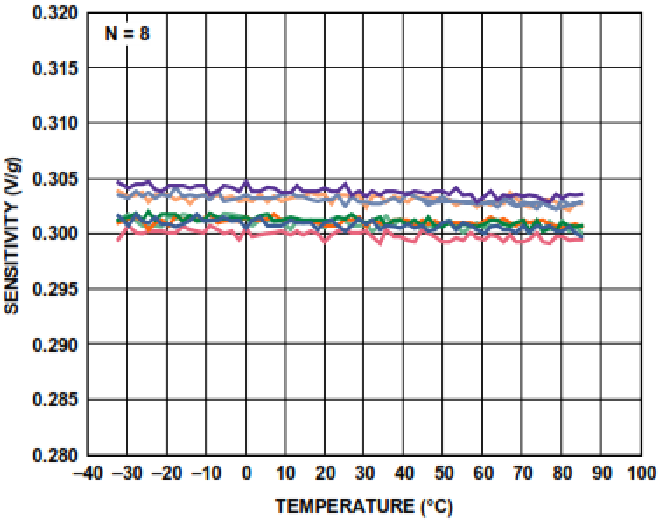


Figure 4.4: X-Axis Sensitivity vs. Temperature – Eight Parts Soldered to PCB, Vs = 3V

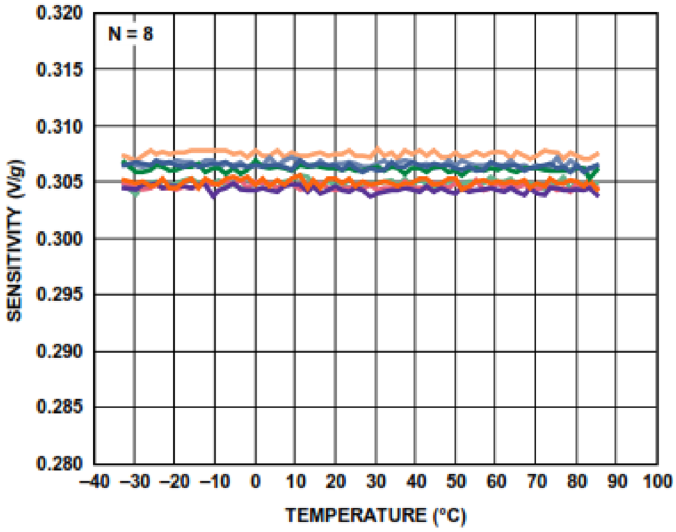


Figure 4.5: Y-Axis Sensitivity vs. Temperature – Eight Parts Soldered to PCB, Vs = 3V

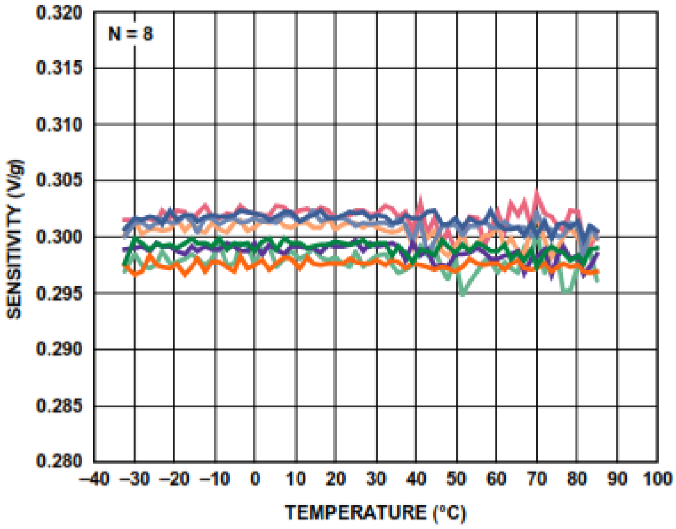


Figure 4.6: Z-Axis Sensitivity vs. Temperature – Eight Parts Soldered to PCB, Vs = 3V

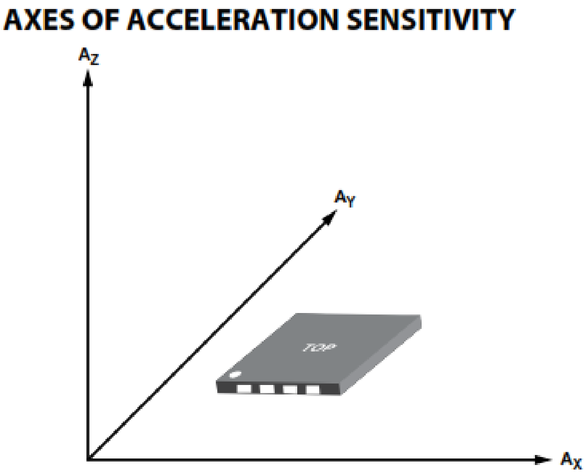


Figure 4.7: Axes of Acceleration Sensitivity; Corresponding Output Voltage Increases When Accelerated Along the Sensitive Axis

4.3 DATA FILTERING OF 5-AXIS IMU SENSOR

The raw data from the accelerometers cannot be plugged into the equations without any filtering and resetting. Due to various noises or packet loss, sometimes data can be out of range and thus need to be filtered out, and a relatively simple filter has been implemented for the experiments here.

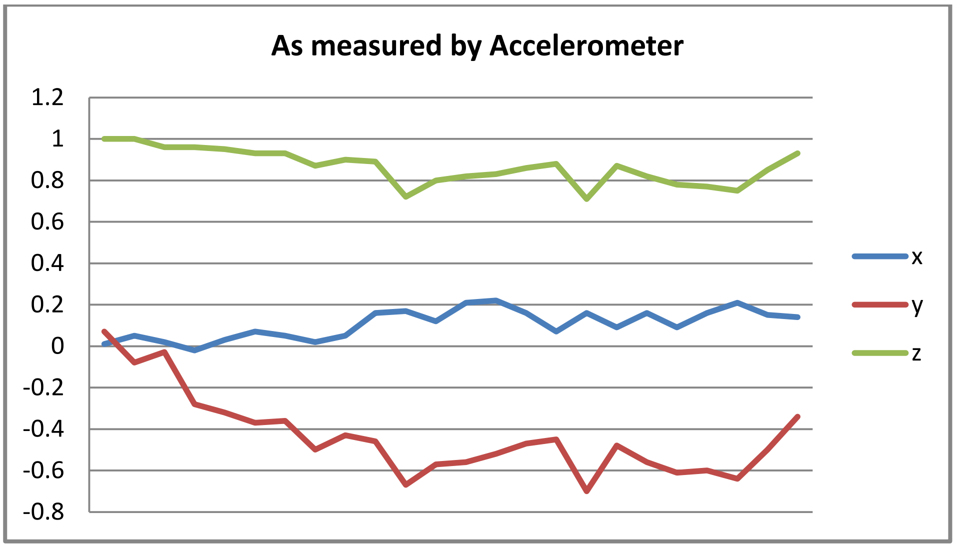


Figure 4.8: Sensor output before filter

IMU consists of gyroscopes and accelerometers to report on the orientation, velocity and gravity. The data from the IMU is filtered and fused using Kalman filter.

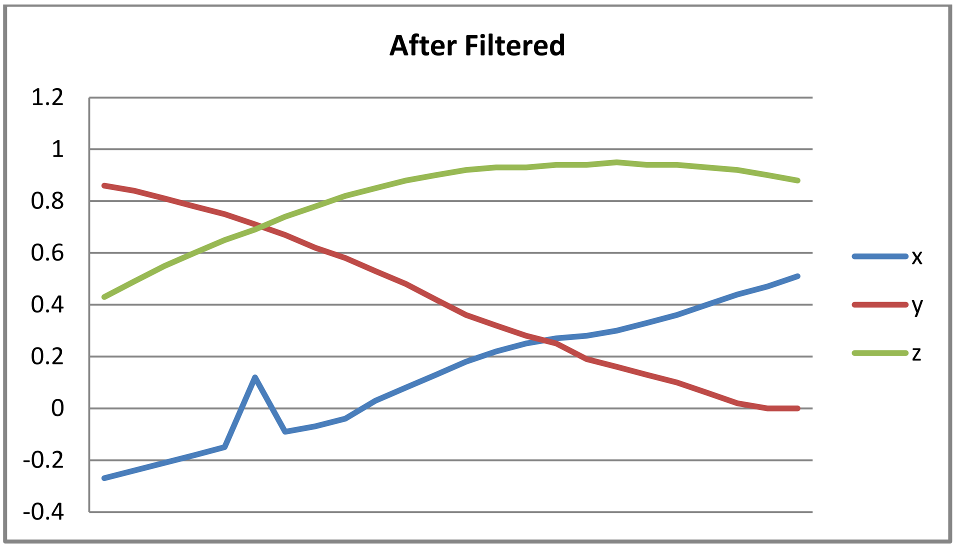


Figure 4.9: Sensor output after filtered using Kalman filter

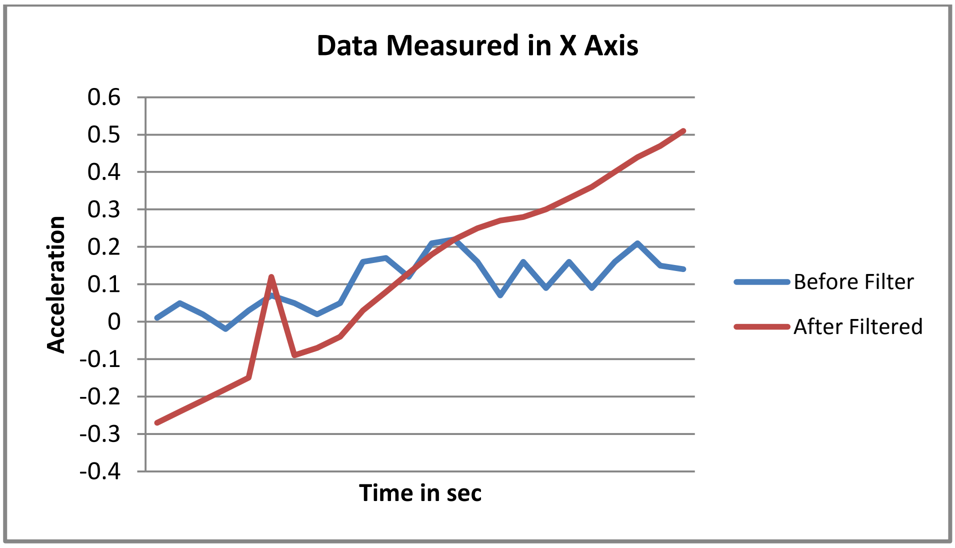


Figure 4.10: IMU Sensor output before and after filtered in x-axis

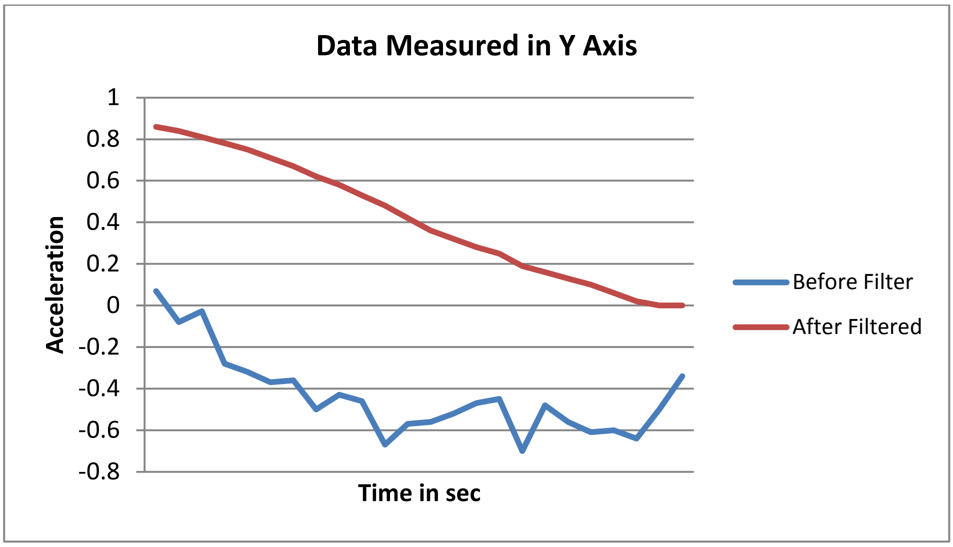


Figure 4.11: IMU Sensor output before and after filtered in y-axis

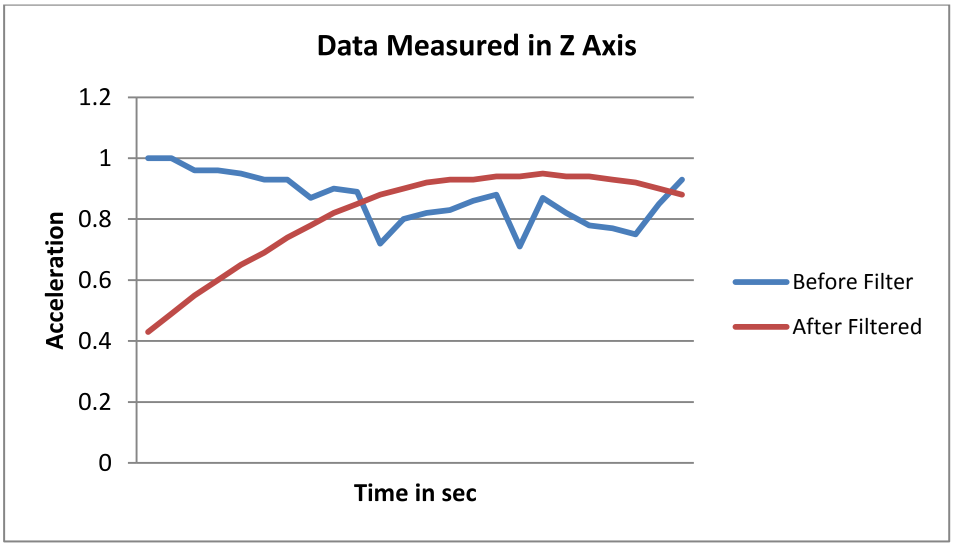


Figure 4.12: IMU Sensor output before and after filtered in z-axis

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 CONCLUSION

Measured sensor data usually consists of noise which reduces accuracy. In order to get accurate estimates of the true value, the sensor data needs to be filtered. A good filtering algorithm can eliminate noise from the data and retain useful information. A test was created to characterize the random noise and errors inherent to orientation sensing in the SN-IMU5D-LC for static cases as well as after experiencing an impact force. When not all states are measurable, the Kalman filter can increase the system stability compared to other methods of obtaining the unmeasurable states, such as differentiating or integrating sensor data from the measurable states.

The ability of the Kalman filter to incorporate both a model of the system and inaccurate measurements, along with the availability of fast and inexpensive computers make the Kalman filter an excellent choice to filter noise and estimate states. Rotational motion of the body with respect to inertial reference frame may be sensed using gyroscopic sensors that are used to determine the orientation of the accelerometers at all times. Given this information it is possible to resolve the accelerations into the reference frame before the integration process takes place.

In order to navigate with respect to our inertial reference frame, it is necessary to keep track of the direction in which the accelerometers are pointing. Kalman Filter combines measured data as well as previous knowledge about the system and measuring devices to produce an optimal estimate of the desired variables. Compared to other filters, Kalman filter minimizes the error significantly.

5.2 RECOMMENDATION

From the study, IMU seems can be used in many application. For future project, this study of IMU wanted to be applied to digital compass using quaternion method which will be used as orientation algorithm to the compass. The function of IMU that can measure the pitch, roll and yaw is the main type sensor that must be used for that application. IMU sensor also can be combined with other sensor such as GPS for accurate navigation, guidance and controlling system.

REFERENCES

[1] Kong, X. 2000. *Inertial Navigation System Algorithms for Low Cost IMU. Department of Mechanical and Mechatronic Engineering.* The University of Sydney.

[2] Welch, G. and Bishop, G. 2006. *An Introduction to the Kalman Filter*. University of North Carolina.

[3] Simon, D. 2001. *Kalman Filtering*. Embedded Systems Programming, Kalman Filters.

[4] Kim, A. and Golnaraghit, M.F. 2004. *A Quaternion-Based Orientation Estimation Algorithm Using an Inertial Measurement Unit.* University of Waterloo.

[5] Damian, J.F.V. 2011. *Kalman Filter for Sensor Data Fusion in an Underwater Inertial Navigation System*. Fachbereich Maschinenbau und Mechatronik.

[6] Caron, F., Duflos, E., Pomorski, D. and Vanheeghe, P. 2004. *GPS/IMU data fusion using multisensor Kalman filtering: introduction of contextual aspects*. F. Caron et al. / Information Fusion 7 (2006) 221–230.

[7] Space Systems Laboratory. June 2013. *Inertial Measurement Unit (IMU).* <http://www.ssl.umd.edu/projects/RangerNBV/thesis/2-4-1.htm>

[8] Kesten, R. 2010. *Data Fusion of Accelerometric, Gyroscopic, Magnetometric and Barometric Information for Attitude and Altitude Estimation of an Unmanned Quadrocopter.* Institut fur Zuverlassigkeitstechnik, Technische Universitat Hamburg.

[9] Shiau, J.K., Huang, C.X. and Chang, M.Y. 2012. *Noise Characteristics of MEMS Gyro's Null Drift and Temperature Compensation.* Department of Aerospace Engineering, Tamkang University.

[10] Physics Kenyon. December 2012. *Gyroscope*. <http://physics.kenyon.edu/EarlyApparatus/Mechanics/Gyroscope/Gyroscope.html>

[11] Han, S. and Wang, J. *A Novel Method to Integrate IMU and Magnetometers in Attitude and Heading Reference Systems.* National University of Defense Technology and University of New South Wales, Australia.

[12] Titterton, D.H. and Weston, J.L. 2004. *Strapdown Inertial Navigation Technology- 2nd Edition.* The Institution of Electrical Engineers.

[13] Foxlin, E. *Inertial head-tracker sensor fusion by a complementary separate-bias Kalman filter.* Proc. of The IEEE Vrnual Reolrry Annual InternarionnlSym~~sium, Mar. 30-Apr. 3, 1996, pp. 185.194.

[14] Elkhidir T.Y., Shuhimi M., Musa T.A. and Satti, A. 2011. *Pre-Filtering Low-Cost Inertial Measuring Unit using a Wavelet Thresholding De-noising for IMU/GPS Integration.* UTM Aeronautic Lab, Faculty of Mechanical Engineering, Universiti Teknologi Malaysia.

[15] Angelosanto, G. 2008. *Kalman Filtering of IMU Sensor for Robot Balance Control.* Department of Mechanical Engineer, Massachusetts Institute of Technology.

[16] Anwar, N.S.N. 2010. *Investigations on an Algorithm for Underwater Inertial Navigation*, Fachhochschule Aachen.

[17] Mau, S. 2005. *What is the Kalman Filter and how can it be used for Data Fusion*? Robotics Math 16-811.

[18] Wikipedia, the free encyclopaedia. December 2012. *Navigation.* <http://en.wikipedia.org/wiki/Navigation>

[19] Saadat, H. 2002. *Introduction to Matlab and Simu*link. Milwaukee School of Engineering.

[20] Kumar, V.N. 2004. *Integration of Inertial Navigation System and Global Positioning System Using Kalman Filtering.* Indian Institute of Technology, Bombay.

[21] Woodman, O.J. 2007. *An introduction to inertial navigation*. University of Cambridge.

[22] Gebre-Egziabher, D. 2004. *Design and Performance Analysis of a Low-Cost Aided Dead Reckoning Navigator*. Department of Aeronautics and Astronautics, Stanford University.

[23] Wu, Y., Hu, X., Wu, M. and Hu, D. 2004. *Strap down Inertial Navigation using Dual Quaternion Algebra: Error Analysis*. National University of Defense Technology.

[24] Walchko, K.J., Dr. Mason, P.A.C. 2002. *Inertial Navigation*. University of Florida, Gainesville, FL 32611-6200, NASA Goddard Space Flight Center, Greenbelt, MD.

[25] Rojas, R. *The Kalman Filter*. Freie Universitat Berlin, Institut fur Informatik.

[26] Hazry, D., Sofian, M. and Zul Azfar, A. 2012. *Study of Inertial Measurement Unit Sensor*. Autonomous System and Machine Vision Laboratory, School of Mechatronic, Universiti Malaysia Perlis, Malaysia.

[27] Zarchan, P.M. 2003. *Fundamentals of Kalman Filtering: A Practical Approach*. Lexington, MA: American Institute of Aeronautics and Astronautics, Inc.

[28] Berthold K. P. Horn. 1986. *Closed-Form Solution of Absolute Orientation Using Unit Quaternions*. Department of Electrical Engineering, University of Hawaii. Journal of the Optical Society of America A, Vol. 4, page 629, April 1987.

[29] Tait, P.G. 1886. *Quaternions*. M.A. Sec. R.S.E. Encyclopædia Britannica, Ninth Edition, 1886, Vol. XX, pp. 160-164.

[30] Tsai, Y.L., Tu, T.T., Bae, H. and Chou. P.H. *EcoIMU: A Dual Triaxial-Accelerometer Inertial Measurement Unit for Wearable Applications*. Department of Computer Science National Tsing Hua University Hsinchu, Taiwan and Center for Embedded Computer Systems University of California, Irvine.

[31] Kang, C.W. and Park, C.G. 2009. *Attitude Estimation with Accelerometers and Gyros Using Fuzzy Tuned Kalman Filter*. Seoul National University, Seoul.

[32] Grewal, M.S., Weill, L.R. and Andrews, A.P. *Global Positioning Systems, Inertial Navigation, and Integration*. Second edition. A John Wiley & Sons, Inc., Publication

[33] Hoffmann, G. 2003. Euler Angles and Projections. <http://www.fho-emden.de/~hoffmann/index.html>. June, 2013.

[34] Cytron Technologies. October 2012. *IMU 5 Degrees of Freedom*. <http://www.cytron.com.my/viewProduct.php?pcode=SN-IMU5D&name=IMU 5 Degrees of Freedom>

[35] Cytron Technologies. October 2012. *Compass Module*. <http://www.cytron.com.my/viewProduct.php?pcode=SN-HMC6352&name=Compass Module>

[36] Arduino. March 2013. *Arduino Uno*. <http://www.arduino.cc/en/Main/ArduinoBoardUno>

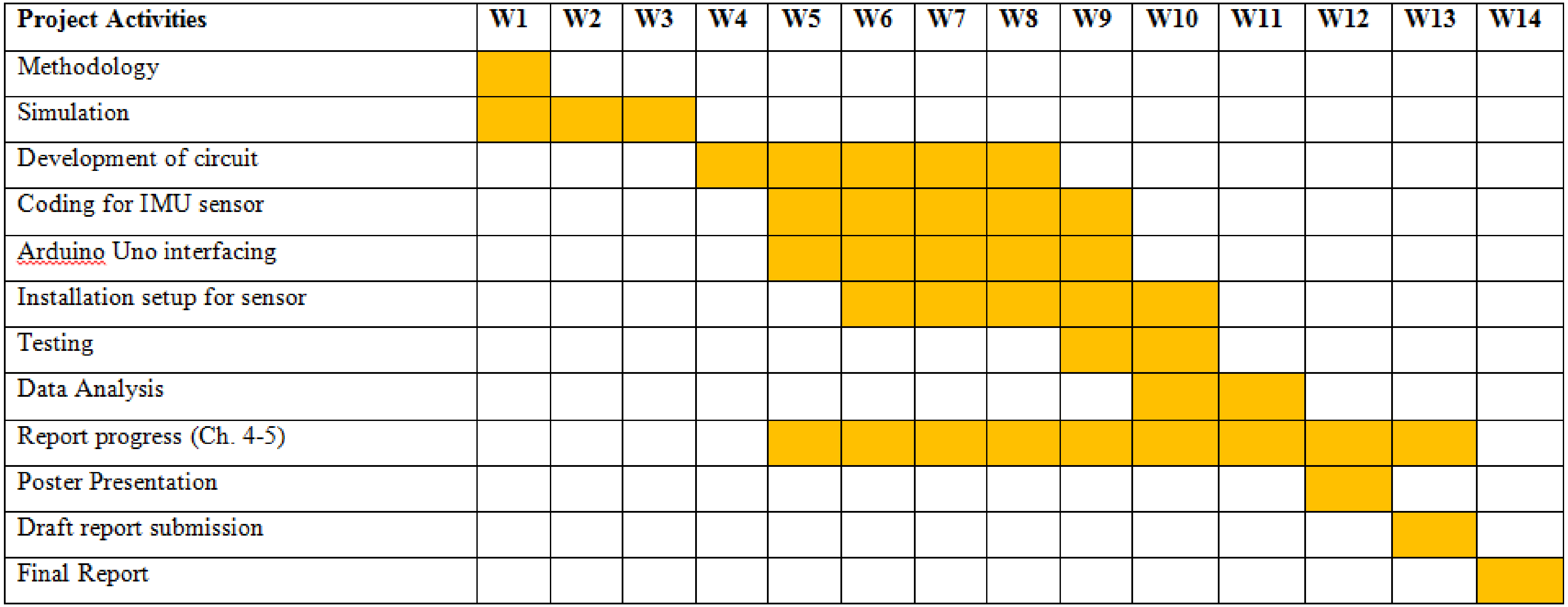
[37] Mohd Isa, W.H. 2012. *Orientation Algorithm for Dive Navigation with Data Fusion of Turning Rate and Magnetic Sensors*. Fachhochschule Aachen.

APPENDIX A

GANTT CHART

Project Activities	**W1**	**W2**	**W3**	**W4**	**W5**	**W6**	**W7**	**W8**	**W9**	**W10**	**W11**	**W12**	**W13**	**W14**
Verify the project title	■													
Verify the project objectives and scopes	■	■												
Literature search (books/journal)		■	■	■										
Study of Data Filtering of 5-Axis Inertial Measurement Unit using Kalman Filter (Journal)				■										
Simulation study using MATLAB				■	■	■	■	■						
Equipment required (software)					■	■	■	■	■	■	■	■		
Validate the modal parameter								■	■	■	■	■	■	
Improvement of MATLAB parameter									■	■	■	■	■	
Methodology										■	■	■	■	
Report progress (Ch. 1-3)												■	■	
Prepared for presentation												■	■	
Presentation														■

Figure A1: Gantt chart of FYP 1



Project Activities	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14
Methodology	🟨													
Simulation	🟨	🟨	🟨											
Development of circuit				🟨	🟨	🟨	🟨	🟨						
Coding for IMU sensor					🟨	🟨	🟨	🟨	🟨					
Arduino Uno interfacing					🟨	🟨	🟨	🟨	🟨					
Installation setup for sensor						🟨	🟨	🟨	🟨	🟨				
Testing									🟨	🟨				
Data Analysis										🟨	🟨			
Report progress (Ch. 4-5)					🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	
Poster Presentation												🟨		
Draft report submission													🟨	
Final Report														🟨

Figure A2: Gantt chart of FYP 2

APPENDIX B

ARDUINO CODING FOR IMU 5DOF

```
/*
Hardware Setup:
Acc_Gyro      <---> Arduino
5V            <---> 5V
GND           <---> GND
AX            <---> AN0
AY            <---> AN1
AZ            <---> AN2
GX4           <---> AN3
GY4           <---> AN4

*/
#define INPUT_COUNT 5              //number of analogue inputs
#define VDD 5000.0f                //Analogue reference voltage in millivolts
#define PI 3.14159265358979f

int an[INPUT_COUNT];               //analogue inputs
char firstSample;                  //marks first sample

struct {
   char inpInvert[INPUT_COUNT];    // bits 0..5 invert input
   int zeroLevel[INPUT_COUNT];     // 0...2 accelerometer zero level (mV) @ 0 G
                                   // 3...5 gyro zero level (mV) @ 0 deg/s
   int inpSens[INPUT_COUNT];       // 0...2 accelerometer input sensitivity (mv/g)
                                   // 3...5 gyro input sensitivity (mV/deg/ms)
```

```
   float wGyro;                              // gyro weight/smoothing factor
         } config;


//RwAcc[0], RwAcc[1], RwAcc[2] means RxAcc,  RyAcc, RzAcc
float RwEst[3];              //Rw estimated from combining RwAcc and RwGyro unsigned
                               long lastMicros;


unsigned long interval;  //interval since previous analogue samples
float RwAcc[3];        //projection of normalized gravitation force vector on x/y/z axis, as
                        measured by accelerometer
float RwGyro[3];     //Rw obtained from last estimated value and gyro movement
float Awz[2];           //angles between projection of R on XZ/YZ plane and Z axis (deg)


void setup()
{
   static int i;
   Serial.begin(9600);
   for(i=0;i<=2;i++)    {                  // X,Y,Z axis
       config.zeroLevel[i] = 1650;      // Accelerometer zero level (mV) @ 0 G
       config.inpSens[i] = 478;          // Accelerometer Sensisitivity mV/g
                              }
   for(i=3;i<=4;i++)    {
       config.inpSens[i] = 2000;         // Gyro Sensitivity mV/deg/ms
       config.zeroLevel[i] = 1230;      // Gyro Zero Level (mV) @ 0 deg/s
                              }
   config.inpInvert[0] = 1;              //Acc X
   config.inpInvert[1] = 1;              //Acc Y
   config.inpInvert[2] = 1;              //Acc Z
   config.inpInvert[3] = 1;              //Gyro X
   config.inpInvert[4] = 1;              //Gyro Y
```

```
   config.wGyro = 10;
   firstSample = 1;
}
void loop()
{
   getEstimatedInclination();
   Serial.print(" , ");
   Serial.print(RwAcc[0]);     //Inclination X axis (as measured by accelerometer)
   Serial.print(" , ");
   Serial.print(RwEst[0]);     //Inclination X axis (estimated / filtered)
   Serial.print(" , ");
   Serial.print(RwAcc[1]);     //Inclination Y axis (as measured by accelerometer)
   Serial.print(" , ");
   Serial.print(RwEst[1]);     //Inclination Y axis (estimated / filtered)
   Serial.print(" , ");
   Serial.print(RwAcc[2]);     //Inclination Z axis (as measured by accelerometer)
   Serial.print(" , ");
   Serial.print(RwEst[2]);     //Inclination Z axis (estimated / filtered)
   Serial.println("");
}

void getEstimatedInclination()
{
   static int i,w;
   static float tmpf,tmpf2;
   static unsigned long newMicros;          //new timestamp
   static char signRzGyro;

//get raw adc readings
```

```
  newMicros = micros();                    //save the time when sample is taken
  for(i=0;i<INPUT_COUNT;i++) an[i]= analogRead(i);

//compute interval since last sampling time
  interval = newMicros - lastMicros;
  lastMicros = newMicros;

//get accelerometer readings in g, gives us RwAcc vector
  for(w=0;w<=2;w++) RwAcc[w] = getInput(w);

//normalize vector (convert to a vector with same direction and with length 1)
  normalize3DVector(RwAcc);

  if (firstSample)
     {
     for(w=0;w<=2;w++)  RwEst[w]  =  RwAcc[w];     //initialize  with  accelerometer
                                                     readings
      } else {
//evaluate RwGyro vector
            if(abs(RwEst[2]) < 0.1)
           {
//Rz is too small and because it is used as reference for computing Axz, Ayz its error
  fluctuations will amplify leading to bad results
//in this case skips the gyro data and just use previous estimate
           for (w=0;w<=2;w++) RwGyro[w] = RwEst[w];
           } else {

//get angles between projection of R on ZX/ZY plane and Z axis, based on last RwEst
           for (w=0;w<=1;w++)
           {
```

```
        tmpf = getInput(3 + w);           //get current gyro rate in deg/ms

        tmpf *= interval / 1000.0f;        //get angle change in deg

        Awz[w] = atan2(RwEst[w],RwEst[2]) * 180 / PI;  //get angle and convert to
                                                         degrees

        Awz[w] += tmpf;                   //get updated angle according to gyro movement

        }


//estimate sign of RzGyro by looking in what quadrant the angle Axz is,

//RzGyro is positive if Axz in range -90...90 => cos(Awz) >= 0

  signRzGyro = ( cos(Awz[0] * PI / 180) >=0 ) ? 1: -1;

  //reverse calculation of RwGyro from Awz angles,

    for (w=0;w<=1;w++)

{

RwGyro[0] = sin(Awz[0] * PI / 180);

RwGyro[0] /= sqrt( 1 + squared(cos(Awz[0] * PI / 180)) * squared(tan(Awz[1] * PI /
180)) );

RwGyro[1] = sin(Awz[1] * PI / 180);

RwGyro[1] /= sqrt( 1 + squared(cos(Awz[1] * PI / 180)) * squared(tan(Awz[0] * PI /
180)) );

}

RwGyro[2] = signRzGyro * sqrt(1 - squared(RwGyro[0]) - squared(RwGyro[1]));

}


//combine Accelerometer and gyro readings

     for (w=0;w<=2;w++) RwEst[w] = (RwAcc[w] + config.wGyro* RwGyro[w]) / (1 +
    config.wGyro);


    normalize3DVector(RwEst);

}

    firstSample = 0;
```

```
}

void normalize3DVector(float* vector)
        {
            static float R;
            R = sqrt(vector[0]*vector[0] + vector[1]*vector[1] + vector[2]*vector[2]);
            vector[0] /= R;
            vector[1] /= R;
            vector[2] /= R;
        }
  float squared(float x)
        {
  return x*x;
        }

//For accelerometer it will return g (acceleration), applies when xyz = 0...2
//For gyro it will return deg/ms (rate of rotation), applies when xyz = 3...5
float getInput(char i)
{
  static float tmpf;                  //temporary variable
  tmpf = an[i] * VDD / 1023.0f;   //voltage (mV)
  tmpf -= config.zeroLevel[i];      //voltage relative to zero level (mV)
  tmpf /= config.inpSens[i];          //input sensitivity in mV/G(acc) or mV/deg/ms(gyro)
  tmpf *= config.inpInvert[i];       //invert axis value according to configuration
  return tmpf;
}
```

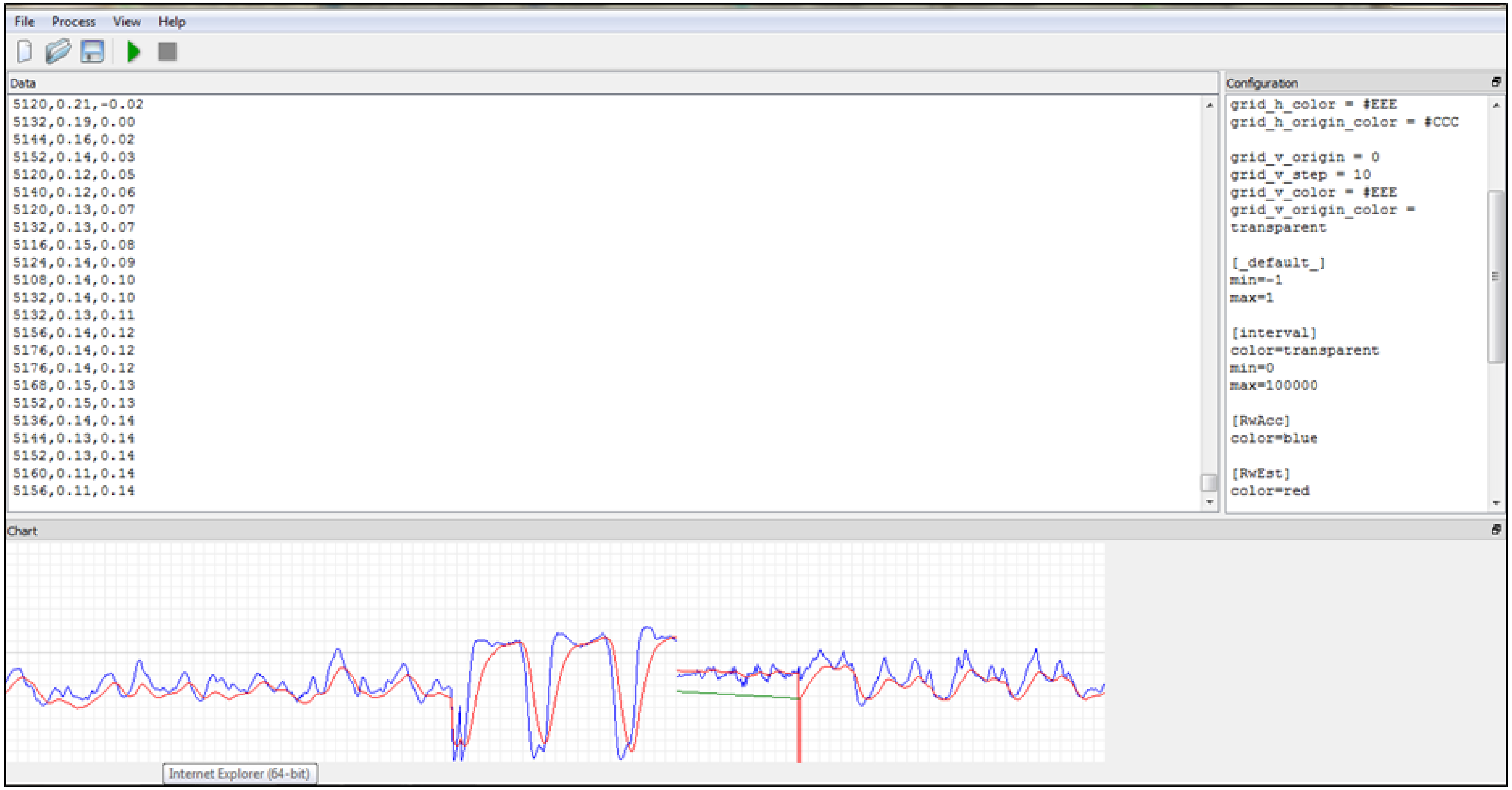
APPENDIX C

SERIAL CHART PROGRAMMING CODE

```
[_setup_]
port=COM3
baudrate=9600
width=1000
height=201
background_color = white

grid_h_origin = 100
grid_h_step = 10
grid_h_color = #EEE
grid_h_origin_color = #CCC
grid_v_origin = 0
grid_v_step = 10
grid_v_color = #EEE
grid_v_origin_color = transparent

[_default_]
min=-1
max=1

[RxAcc]
color=blue
[RxEst]
color=red
[RyAcc]
color=green
[RyEst]
color=magenta
[RzAcc]
color=black
[RzEst]
color=yellow

[RwGyro]
color=transparent
[GyroRate]
color=transparent
min=-10
max=10

[an3]
color=transparent
min=0
max=1023
```

APPENDIX D

SERIAL CHART OUTPUT



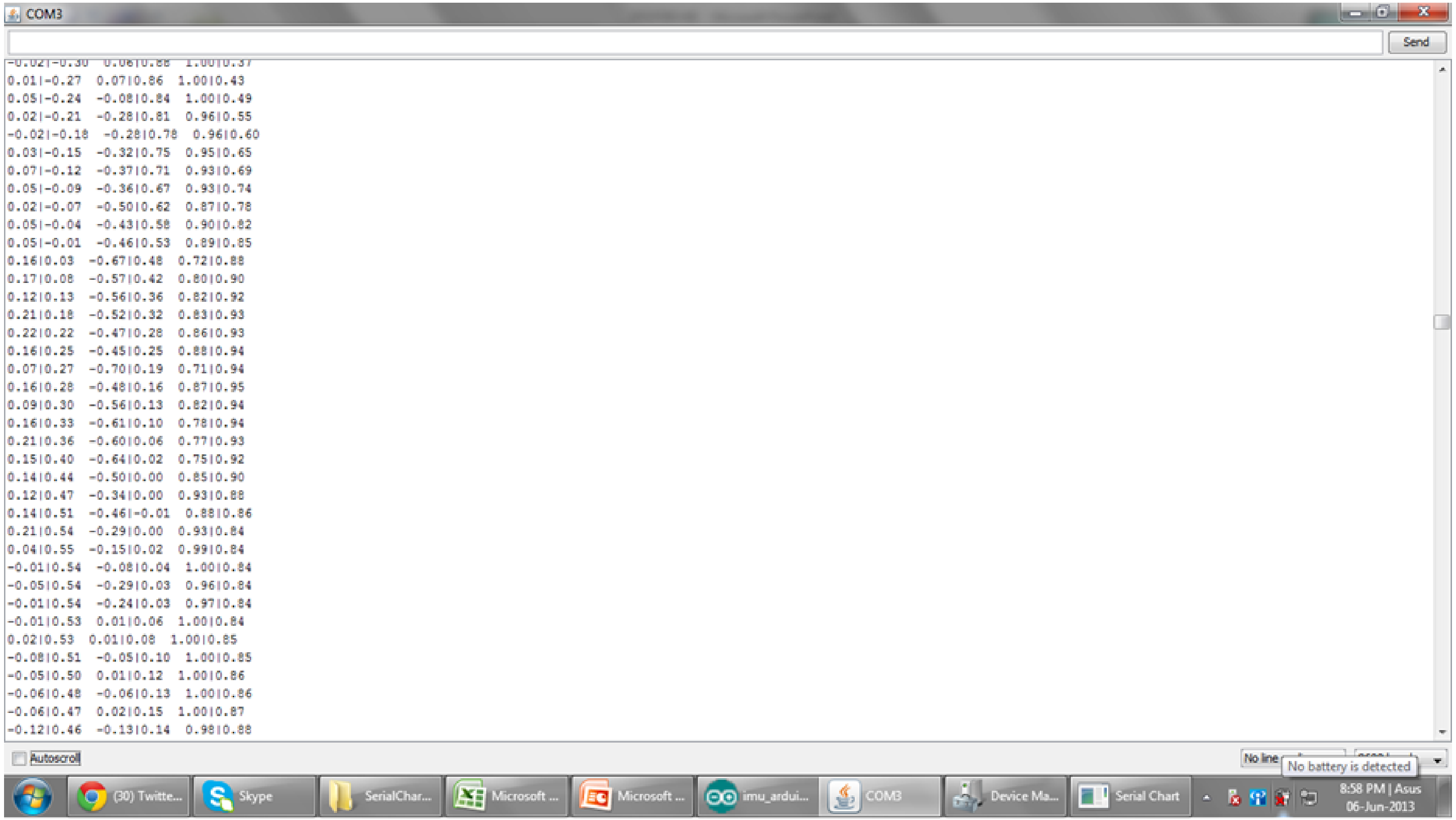
```
5120,0.21,-0.02
5132,0.19,0.00
5144,0.16,0.02
5152,0.14,0.03
5120,0.12,0.05
5140,0.12,0.06
5120,0.13,0.07
5132,0.13,0.07
5116,0.15,0.08
5124,0.14,0.09
5108,0.14,0.10
5132,0.14,0.10
5132,0.13,0.11
5156,0.14,0.12
5176,0.14,0.12
5176,0.14,0.12
5168,0.15,0.13
5152,0.15,0.13
5136,0.14,0.14
5144,0.13,0.14
5152,0.13,0.14
5160,0.11,0.14
5156,0.11,0.14
```

```
grid_h_color = #EEE
grid_h_origin_color = #CCC

grid_v_origin = 0
grid_v_step = 10
grid_v_color = #EEE
grid_v_origin_color =
transparent

[_default_]
min=-1
max=1

[interval]
color=transparent
min=0
max=100000

[RwAcc]
color=blue

[RwEst]
color=red
```

Figure D: Serial chart output when sensor moves in X-axis

APPENDIX E

OUTPUT FROM ARDUINO

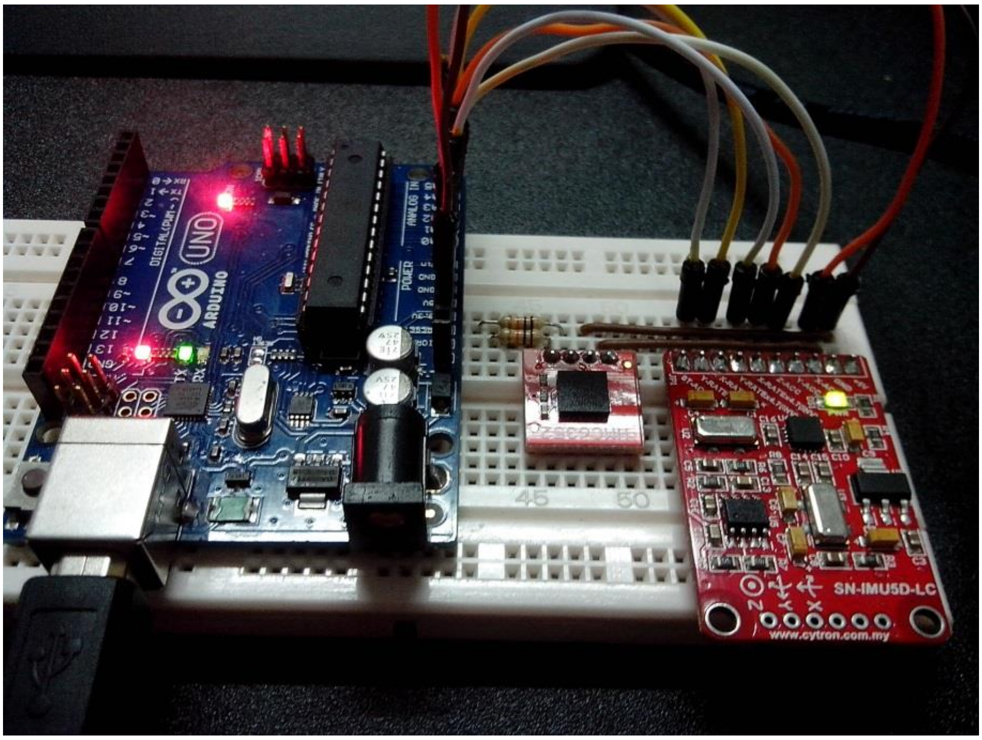


```
-0.02|-0.30  0.06|0.88  1.00|0.37
0.01|-0.27  0.07|0.86  1.00|0.43
0.05|-0.24  -0.08|0.84  1.00|0.49
0.02|-0.21  -0.28|0.81  0.96|0.55
-0.02|-0.18  -0.28|0.78  0.96|0.60
0.03|-0.15  -0.32|0.75  0.95|0.65
0.07|-0.12  -0.37|0.71  0.93|0.69
0.05|-0.09  -0.36|0.67  0.93|0.74
0.02|-0.07  -0.50|0.62  0.87|0.78
0.05|-0.04  -0.43|0.58  0.90|0.82
0.05|-0.01  -0.46|0.53  0.89|0.85
0.16|0.03  -0.67|0.48  0.72|0.88
0.17|0.08  -0.57|0.42  0.80|0.90
0.12|0.13  -0.56|0.36  0.82|0.92
0.21|0.18  -0.52|0.32  0.83|0.93
0.22|0.22  -0.47|0.28  0.86|0.93
0.16|0.25  -0.45|0.25  0.88|0.94
0.07|0.27  -0.70|0.19  0.71|0.94
0.16|0.28  -0.48|0.16  0.87|0.95
0.09|0.30  -0.56|0.13  0.82|0.94
0.16|0.33  -0.61|0.10  0.78|0.94
0.21|0.36  -0.60|0.06  0.77|0.93
0.15|0.40  -0.64|0.02  0.75|0.92
0.14|0.44  -0.50|0.00  0.85|0.90
0.12|0.47  -0.34|0.00  0.93|0.88
0.14|0.51  -0.46|-0.01  0.88|0.86
0.21|0.54  -0.29|0.00  0.93|0.84
0.04|0.55  -0.15|0.02  0.99|0.84
-0.01|0.54  -0.08|0.04  1.00|0.84
-0.05|0.54  -0.29|0.03  0.96|0.84
-0.01|0.54  -0.24|0.03  0.97|0.84
-0.01|0.53  0.01|0.06  1.00|0.84
0.02|0.53  0.01|0.08  1.00|0.85
-0.08|0.51  -0.05|0.10  1.00|0.85
-0.05|0.50  0.01|0.12  1.00|0.86
-0.06|0.48  -0.06|0.13  1.00|0.86
-0.06|0.47  0.02|0.15  1.00|0.87
-0.12|0.46  -0.13|0.14  0.98|0.88
```

Figure E: Arduino serial monitor output

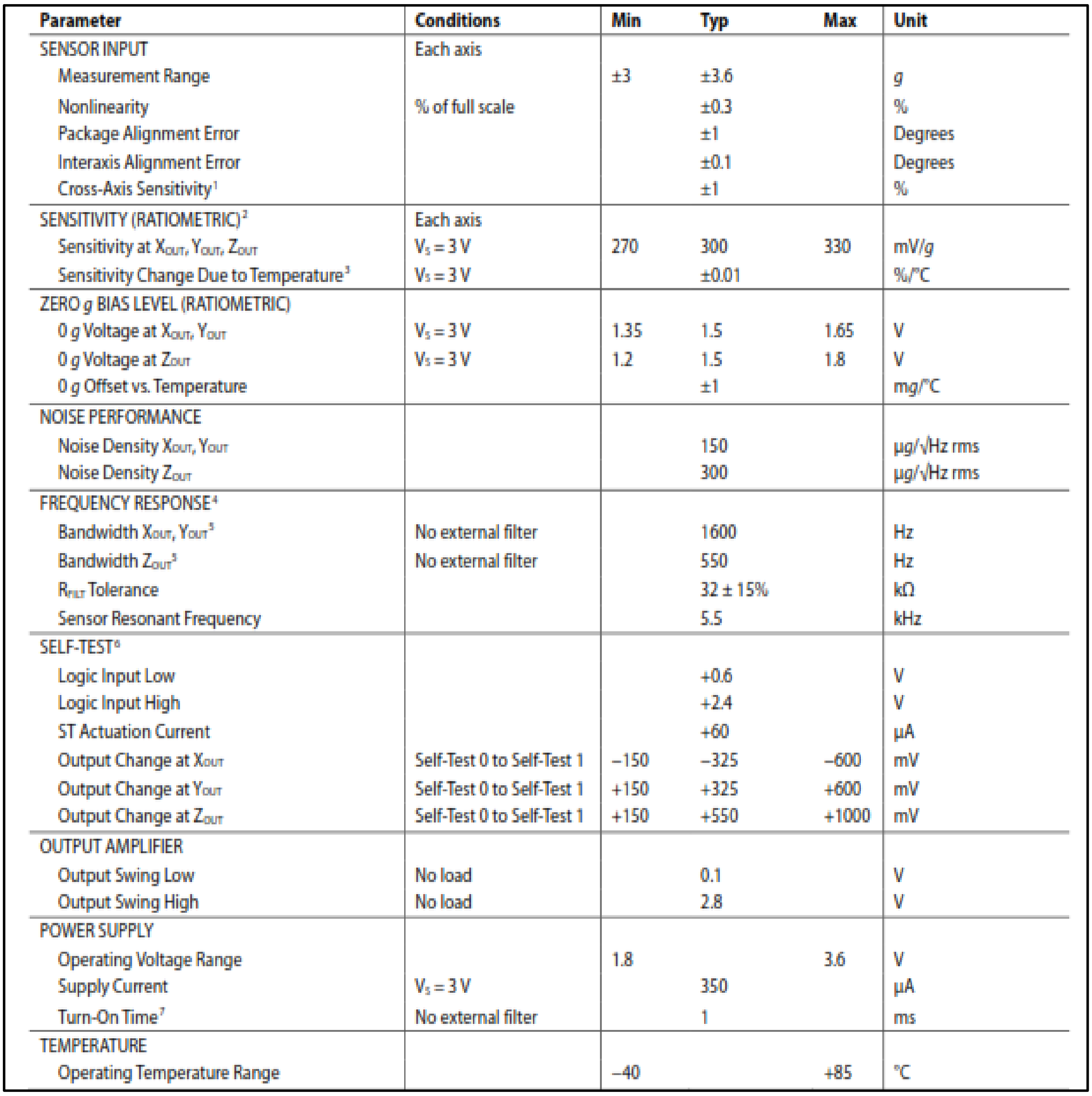
APPENDIX F

ELECTRICAL PART



APPENDIX G

ADXL335 PARAMETER SPECIFICATION



Parameter	Conditions	Min	Typ	Max	Unit
SENSOR INPUT	Each axis				
Measurement Range		±3	±3.6		*g*
Nonlinearity	% of full scale		±0.3		%
Package Alignment Error			±1		Degrees
Interaxis Alignment Error			±0.1		Degrees
Cross-Axis Sensitivity[1]			±1		%
SENSITIVITY (RATIOMETRIC)[2]	Each axis				
Sensitivity at X_{OUT}, Y_{OUT}, Z_{OUT}	V_S = 3 V	270	300	330	mV/*g*
Sensitivity Change Due to Temperature[3]	V_S = 3 V		±0.01		%/°C
ZERO *g* BIAS LEVEL (RATIOMETRIC)					
0 *g* Voltage at X_{OUT}, Y_{OUT}	V_S = 3 V	1.35	1.5	1.65	V
0 *g* Voltage at Z_{OUT}	V_S = 3 V	1.2	1.5	1.8	V
0 *g* Offset vs. Temperature			±1		m*g*/°C
NOISE PERFORMANCE					
Noise Density X_{OUT}, Y_{OUT}			150		µ*g*/√Hz rms
Noise Density Z_{OUT}			300		µ*g*/√Hz rms
FREQUENCY RESPONSE[4]					
Bandwidth X_{OUT}, Y_{OUT}[5]	No external filter		1600		Hz
Bandwidth Z_{OUT}[5]	No external filter		550		Hz
R_{FILT} Tolerance			32 ± 15%		kΩ
Sensor Resonant Frequency			5.5		kHz
SELF-TEST[6]					
Logic Input Low			+0.6		V
Logic Input High			+2.4		V
ST Actuation Current			+60		µA
Output Change at X_{OUT}	Self-Test 0 to Self-Test 1	−150	−325	−600	mV
Output Change at Y_{OUT}	Self-Test 0 to Self-Test 1	+150	+325	+600	mV
Output Change at Z_{OUT}	Self-Test 0 to Self-Test 1	+150	+550	+1000	mV
OUTPUT AMPLIFIER					
Output Swing Low	No load		0.1		V
Output Swing High	No load		2.8		V
POWER SUPPLY					
Operating Voltage Range		1.8		3.6	V
Supply Current	V_S = 3 V		350		µA
Turn-On Time[7]	No external filter		1		ms
TEMPERATURE					
Operating Temperature Range		−40		+85	°C