2015-02-09

MEMS-based Downhole Inertial Navigation Systems for Directional Drilling Applications

Wang, Zhen Hua

UNIVERSITY OF CALGARY

MEMS-based Downhole Inertial Navigation Systems for Directional Drilling Applications

by

Zhen Hua Wang

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

GRADUATE PROGRAM IN

ELECTRICAL AND COMPUTER ENGINEERING

CALGARY, ALBERTA

January, 2015

Abstract

Conventional methods in horizontal drilling processes incorporate magnetic surveying techniques for determining the position and the attitude of the bottomhole assembly. Magnetic surveying results in higher costs and relatively large surveying errors. Micro-Electromechanical Systems (MEMS)-based inertial navigation has been proposed as an alternative to magnetometer-based downhole surveying due to its light weight, small size and low power consumption. The present study explores the feasibility of utilizing MEMS-based inertial measurement unit as a surveying sensor, in conjunction with a Rotary In-Drilling Alignment (R-IDA) method for measurement-while-drilling (MWD) processes. A novel, downhole-mountable, autonomous and cost-effective apparatus to practically implement R-IDA has been proposed and its capabilities to reduce the azimuth error have been assessed. Furthermore, this study discusses the concept of wireless data transmission within drill pipes downhole. It is shown that wireless telemetry inside the drill pipe is potentially capable of transmitting at a relatively high rate with low power consumption.

Acknowledgements

First and foremost, I would like to express deep appreciation to my supervisor Dr. Martin P. Mintchev for his practical advice, professional guidance, and continuous support in the development of this research. The things I learned from him will be an important part of my future life. I am also grateful to Dr. Kartikeya Murari, who has provided me with helpful advice and support for my research.

I would like to thank NSERC and PetroJet Canada Inc. for financially supporting this research. In particular, I would like to thank Myles McDougall and Dan McCormack at PetroJet Canada Inc. for providing me with an opportunity to learn and work in the oil industry, and I wish the best of luck for their company. Paul Perrault at Analog Devices Inc. is also thanked for providing experimental material.

I would like to thank all of my colleagues in the Low Frequency Instrumentation Lab, who helped me in various parts of my research: Xue-Xin Gao, Dobromir Filip, Michael Poscente, Gang Wang, Qian Lu, Roger Lu, Alex Jurkov and Leticia Angulo. I would also like to extend my gratitude to the technologists and colleagues at the Schulich School of Engineering: Frank Hickly, Rob Thomson, Tao Li, Hao Xie, Xiao-Bin Yang, Warren Flaman, Christopher Simon, and John Shelley. I am grateful to Dr. Claire Huot, who has given me tremendous help and advice on life and work. It was my great pleasure to work with her.

Last but not least, I would like to thank my parents Wang Yu-Nan and Xue Yun-Hua for their continuous and unconditional support and encouragement throughout my education. The research would not have been successfully completed without their endless love.

Dedication

For all the missed summers and sleepless nights

Table of Contents

List of Tables

List of Figures and Illustrations

List of Symbols

Symbol	Definition
A	Azimuth angle
D^{-1}	Velocity transform matrix from rectangular coordinates into curvilinear coordinates
$\mathbf{f}^{b}$	Specific forces measured by the accelerometers in the b-frame
f^{b}	Output of the accelerometers in m/sec^2
g	Gravity
G	Noise coefficient matrix
$\mathbf{g}^{l}$	Gravity vector in l-frame
h	Height
$\dot{h}$	Time derivative of height
H	Design matrix
Hz	Hertz
K	Kalman gain
m	Meter
M_x	Magnetometer-calibrated measurement along the x-axis
M_y	Magnetometer-calibrated measurement along the y-axis
M_x^{level}	Leveled magnetic components of the earth along the x-axis
M_y^{level}	Leveled magnetic components of the earth along the y-axis
M_y^{Max}	The maximum values of the measured magnetic field along the y-axis of the IMU
M_x^{Max}	The maximum values of the measured magnetic field along the x-axis of the IMU
M_x^{Min}	The minimum values of the measured magnetic field along the x-axis of the IMU

M_y^{Min}	The minimum values of the measured magnetic field along the y-axis of the IMU
p	Pitch angle
$P(-)$	Priori/predicted covariance matrix of the state estimation uncertainty
$P(+)$	Posteriori/corrected covariance matrix of the state estimation uncertainty
r	Roll angle
R_M	Median radius of the ellipsoid of the Earth
R_N	Normal radius of the ellipsoid of the Earth
R_b^l	Transformation matrix from the *b*-frame to the *l*-frame
$\dot{R}_b^l$	First derivative of the transformation matrix from the *b*-frame to the *l*-frame
R_b^s	Transformation matrix from b-frame to s-frame
s	Second
$S_{Accelerometer}$	Scale factor of accelerometer
$S_{Gyroscope}$	Scale factor of gyroscope
v_e	Velocity in the East direction
v_n	Velocity in the North direction
v_u	Velocity in the up direction
W	Watt
w	Noise vector of the process
X	State vector
$x(-)$	Priori/predicted value of the estimated state vector of a linear system
$x(+)$	Posteriori/corrected value of the estimated state vector of a linear system
Z	Measurement vector of the system

$\boldsymbol{\eta}$	Measurement noise vector
Ω_{ie}^{b}	Earth rotation rate
Ω_{ie}^{l}	Skew symmetric matrix of the angular velocity in the *e*-frame with respect to the *i*-frame
Ω_{el}^{l}	Skew symmetric matrix of the angular velocity in the *l*-frame with respect to the *e*-frame
Ω_{il}^{b}	Skew symmetric matrix of the angular velocity in the *l*-frame with respect to the *i*-frame
Ω_{ib}^{b}	Skew symmetric matrix of the angular velocity in the *l*-frame with respect to the *b*-frame
$\dot{\varphi}$	Time derivative of longitude
$\dot{\lambda}$	Time derivative of latitude
Ω_{lb}^{b}	Skew symmetric matrix of the angular velocity in the *b*-frame with respect to the *i*-frame
$\delta\boldsymbol{\omega}^{l}$	Gyroscope drifts vector in l-frame
$\delta\mathbf{f}^{l}$	Accelerometer biases vector in l-frame
$\delta\boldsymbol{\omega}^{s}$	Gyroscope drifts vector in s-frame
$\delta\omega_{x}^{s}$	Gyroscope drift along x-axis in the s-frame
$\delta\omega_{y}^{s}$	Gyroscope drift along y-axis in the s-frame
$\delta\omega_{z}^{s}$	Gyroscope drift along z-axis in the s-frame
δf_{x}^{s}	Accelerometer bias along x-axis in the s-frame
δf_{y}^{s}	Accelerometer bias along y-axis in the s-frame
δf_{z}^{s}	Accelerometer bias along z-axis in the s-frame
ψ	Local heading
$\tilde{\omega}_{ib}^{b}$	Output of the gyroscopes in radian/sec
$\tilde{\Delta v}^{b}$	Incremental velocity in m/s

$\Delta\theta_{ib}^{b}$	Change in angular rate during time interval
Δt	Sampling interval
δA	Azimuth error
δp	Pitch error
δr	Roll error
δV^{e}	Velocity error in the East
δV^{n}	Velocity error in the North
δV^{u}	Velocity error in the up direction
σ	Standard deviation
ω	Rotation rate of R-IDA setup
λ	Longitude
φ	Latitude
Φ	State transition matrix

List of Abbreviations

ARW	Angle Random Walk
BHA	Bottom Hole Assembly
CW	Clockwise
CCW	Counter Clockwise
CPR	Cycles Per Revolution
CTP	Conventional Terrestrial Pole
CVG	Coriolis Vibratory Gyroscope
DAQ	Data Acquisition
EKF	Extended Kalman Filtering
EMT	Electromagnetic Telemetry
FOG	Fiber Optic Gyroscope
GM	Gauss-Markov
GPS	Global Positioning System
HD	Horizontal Drilling
IDA	In-Drilling Alignment
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
KF	Kalman Filtering
LOD	Levels of Decomposition
LWD	Logging While Drilling
MCU	Microcontroller Unit
MEMS	Micro-electromechanical System

MPT	Mud Pulse Telemetry
MWD	Measurement-While-Drilling
PCB	Printed Circuits Board
R-IDA	Rotary In-Drilling Alignment
SDINS	Strapdown Inertial Navigation System
SPI	Serial Peripheral Interface
STD	Standard Deviation
UBD	Underbalanced Drilling
UART	Universal Asynchronous Receiver/Transmitter
WMM	World Magnetic Model
XBP	XBee Pro S1
ZUPT	Zero Velocity Update

Chapter One: **Introduction**

1.1 Directional Drilling

1.1.1 Horizontal Drilling

Horizontal Drilling (HD) (Fig.1.1) [1] has rapidly matured as a viable alternative to conventional oil exploration and production techniques. The method has proven its worth, since it allows for unprecedented increases in production of resources that might not otherwise be extracted. HD offers access to shallow, spread out, and fractured reservoirs which can be situated far from the drilling location or under riverbeds, mountains, railways, and commercial developments [2]. Horizontal wells are also very productive and offer higher longevity in comparison to vertical wells, due to the fact that there is a larger area of contact with the reservoir and a smaller pressure drop per unit of length in the production section [3]. This provides greater efficiency in the pumping and treatment process [4].

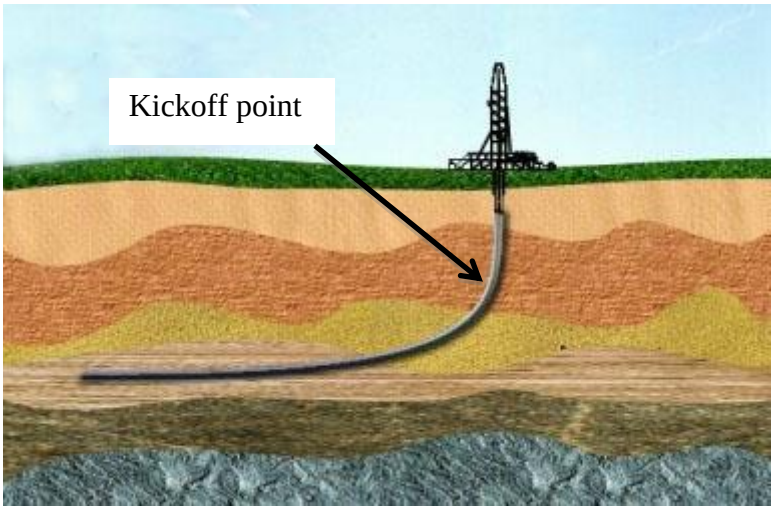


Figure 1.1 Horizontal well and drilling. Modified from [1].

The horizontal well formation process begins by drilling vertically from the surface to a predetermined subsurface location just above the target oil or gas reservoir called the kickoff

point (Fig. 1.1). The wellbore is then intentionally deviated from the vertical plane around a curve to intersect the reservoir at an entry point with a near-horizontal inclination, while remaining within the reservoir until the desired bottomhole location is reached. This complex drilling trajectory requires sophisticated data management, accurate measurement sensors, reliable data transmission methods and most importantly, extensive measurement-while-drilling (MWD) surveying methods. The traditional drilling assembly utilizes a diamond bit and turbo-drill mud motor installed at the front of a trajectory control sub, non-magnetic drill collars which include magnetic surveying sensors, and the drill pipe (Fig.1.2) [5]. MWD and steerable setup, equipped with sophisticated surveying sensors, are crucial elements in horizontal drilling processes.

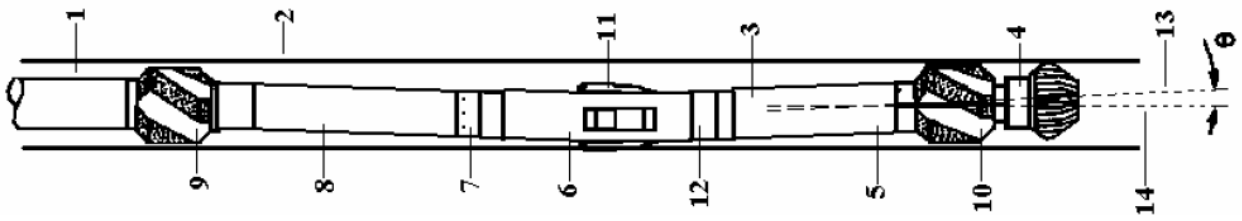
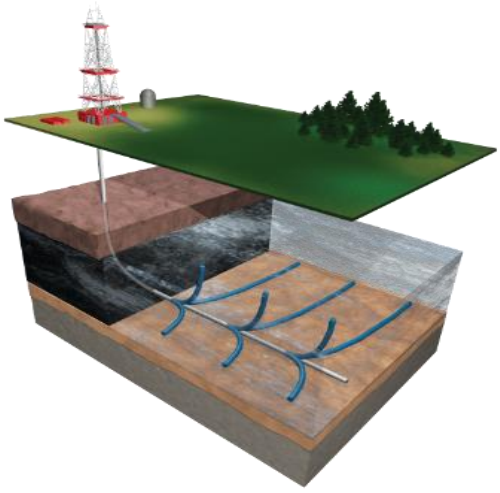


Figure 1.2 Drilling assembly. 1. Drill string. 2. Borehole. 3. Bottom hole assembly (BHA). 4. Drill bit. 5. Drilling motor. 6. Trajectory control sub. 7. Bypass sub. 8. MWD tool included in nonmagnetic collars. 9. Upper stabilizer for centering the drilling assembly in the borehole. 10. Lower stabilizers for centering the drilling assembly in the borehole. 11. Stabilizer blades. 12. Induced bend to provide angular offset (θ) between the (13) drill bit axis and the (14) center line. Modified from [5].

1.1.2 Multilateral Drilling

Multilateral drilling is a recent development in directional well exploration in which several smaller wellbore branches deviate orthogonally from their parent borehole [6]. Although

HD has a better chance of intersecting more fractures than a vertical well, there is a limit to how far horizontal wells can be drilled. Horizontal multilateral wells (Fig.1.3a) [7] feature multiple wells that radiate from the horizontal main borehole and penetrate into the sidetrack. The other type of multilateral wells are vertical multilaterals, which are additional wells sunk down from a main wellbore (Fig. 1.3b) [7] and stretched laterally up to 100 meters. Multilaterals can be as simple as an open hole sidetrack, or they can be more sophisticated with a junction that is cased and has pressure isolation and re-entry capabilities [8].



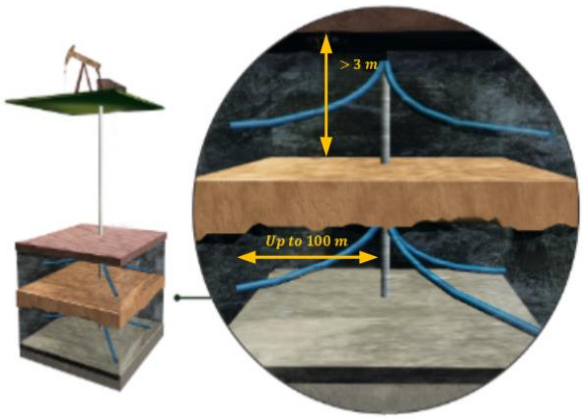


Figure 1.3a Horizontal multilateral drilling **Figure 1.3b Vertical multilateral drilling**

Modified from [7].

The goals of these types of multilaterals are to:

1) Increase reservoir exposure and production rate;

2) Improve production and ultimate recovery;

3) Reduce capital expenditures and lower production costs since more wells are allowed per slot;

4) Reduce dramatically the environmental impact.

1.2 Contemporary downhole navigation in measurement-while-drilling

1.2.1 Magnetic surveying

MWD surveys have been used for several decades to orient the borehole path and determine stability, thereby enhancing drilling efficiency. Conventional MWD surveying determines the position and the orientation of the bottom-hole assembly during the drilling operation by employing three-axis accelerometers and magnetometers, which are affixed in three mutually orthogonal directions. The magnetometers and the accelerometers provide the necessary measurements for computing BHA position and attitude (Fig. 1.4). However, because the azimuth measurements are referenced to the magnetic north [9], randomly located ore deposits and metallic materials in the vicinity of the drilling, as well as geomagnetic influences, degrade the ability of the magnetic triad to accurately navigate the drill bit [10]. In addition, magnetometers are susceptible to internal interferences such as the rotation of the drill string and the flow of drilling fluid [11], [12].

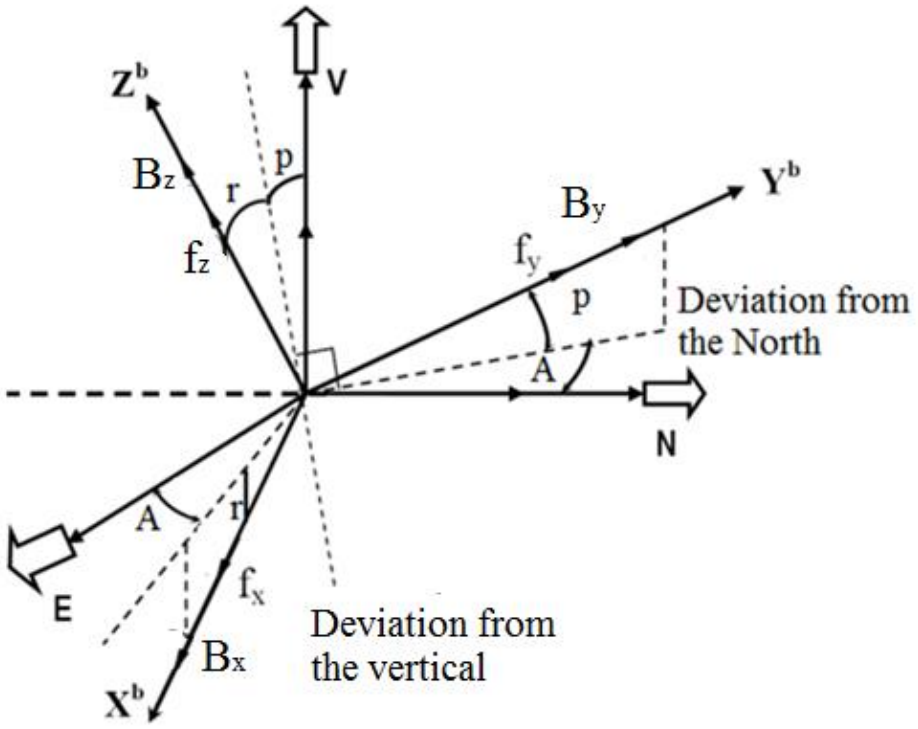


Figure 1.4 Orientation of the MWD magneto-surveying system with respect to North, East, and Vertical directions: the pitch (p), the roll (r), and the azimuth (A). In the drawing, X^b, Y^b and Z^b form the body frame, with its axes coinciding with the axes of the accelerometers and magnetometers. E, N, and V denote East, North, and Vertical and form the reference frame. The measured accelerations along the axes x, y and z of the body frame are respectively f_x, f_y, and f_z. The measured magnetic field in the body frame about the x, y and z axes are respectively B_x, B_y, and B_z. Modified from [5].

1.2.2 Gyroscope-based surveying

As an alternative, gyroscope-based MWD surveying systems have been proposed to provide surveying and orientation data [13], [14]. However, due to inaccurate aiding measurements, these systems are often limited to short-term navigation and cannot provide

continuous attitude and position measurements. These constraints limit the current applicability of gyroscope-based MWD surveying to reservoirs where strong geomagnetic interferences are present and magnetic navigation is impossible. In the quest to improve MWD accuracy, some companies have attempted to integrate both the magnetometers and the gyroscopes within a single unit [15], [16]. However, these wireline gyroscope-based systems share the same drawbacks of short-term navigation, since they are time-consuming and expensive [14].

1.3 MEMS-based inertial navigation system

1.3.1 Micro-electromechanical Systems (MEMS)

MEMS are miniature devices that integrate various mechanical, sensing, actuating, and microelectronic systems into a common silicon substrate utilizing micromachining technologies [17]. MEMS sensors collect information from the environment by measuring mechanical, thermal, chemical, optical and biological phenomena. Electronic components then process the information from the MEMS sensors and a microprocessor can direct actuators to respond by changing position, thereby controlling and influencing the conditions for expected outcomes.

Key benefits of MEMS are miniaturization, low power consumption and low cost, which make them popular in a wide variety of industrial applications including vehicle navigation, agriculture, patient monitoring, and wildlife/livestock tracking [17]. In navigational applications, these features allow the replacement of conventional bulky mechanical components with low cost and enhanced durability components with mass-production capabilities [17].

1.3.2 MEMS Inertial Sensors

1.3.2.1 Accelerometers

An inertial sensor that measures the linear acceleration while moving along its sensitive axis is called an accelerometer. An accelerometer provides the measurement of specific force in

an inertial reference frame, which is then being used to estimate the acceleration of the moving object. The working principle for a MEMS accelerometer is illustrated in Fig. 1.5 [18]. An accelerometer is normally made of three components: a small known mass, a suspension to hold the mass, and a pickoff, which is related to the output signal. In Fig.1.5, the accelerometer measures the exact force including the gravity. The output of it is called a specific force which is given by:

$$mf = a - g \tag{1.1}$$

where f is the specific force; g is the gravitational acceleration; a is an acceleration with respect to the inertial frame; and m is the known mass.

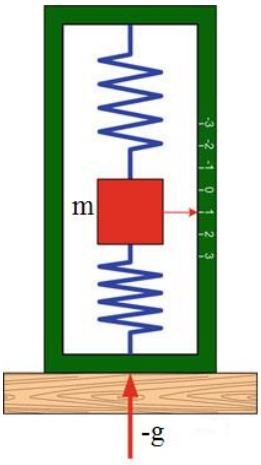


Figure 1.5 An accelerometer resting on a bench with gravity acting on it. Modified from [18].

1.3.2.2 Gyroscopes

A sensor that measures the rate of change of angular velocity along its sensitive axis in an inertial frame is called a gyroscope. In order to fully describe the motion of an object in 3D space, acceleration in conjunction with rotational rate of the body have to be measured and determined. Most of the gyroscopes manufactured with MEMS technology consist of vibrating

structural elements, which do not have any rotating parts that require bearings, and therefore, have dramatically reduced size and overall cost [17].

The vibrating structure gyroscope, standardized by the IEEE as Coriolis vibratory gyroscope (CVG) [19], is based on the principle of energy conversion between two vibration modes of a structure caused by Coriolis force as shown in Fig. 1.6 [17], where X^e and Y^e are the x-axis and the y-axis of the Earth Centered Earth Fixed reference frame, respectively- the reference frame in which we all live.

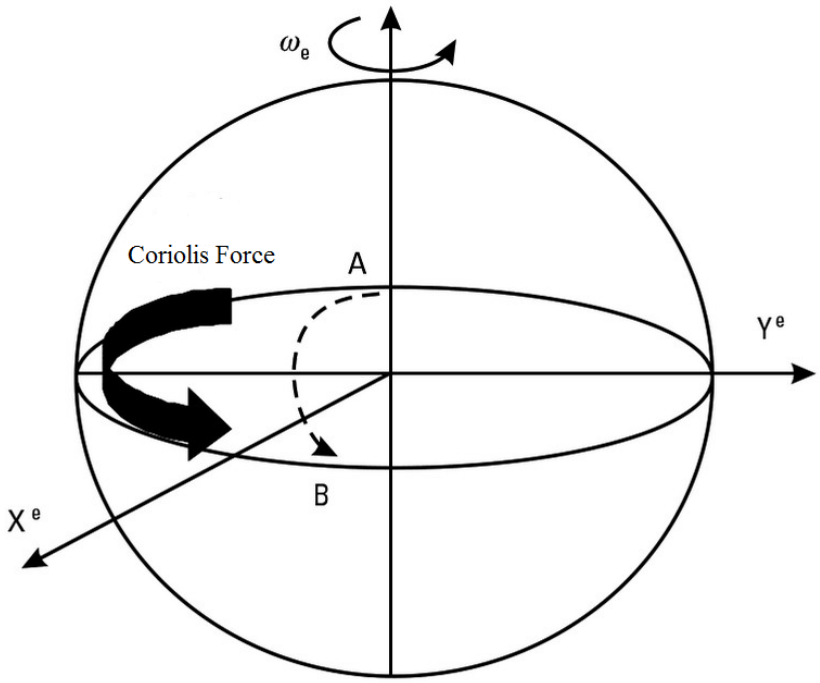


Figure 1.6 Coriolis force. Modified from [17].

Coriolis force, also called a fictitious force or pseudo force, manifests itself as a deflection of a moving object when it is viewed in a rotating reference frame [17]. Although it has been recognized previously by others, it was first explained mathematically by the French engineer and scientist G. G. Coriolis [17]. At a given rotation rate of the observer, the magnitude

of the Coriolis acceleration is proportional to the velocity of the object and to the angle between the direction of the movement and the axis of rotation, as given by:

$$\vec{a} = -2\vec{\omega} \times \vec{v} \tag{1.2}$$

where $\vec{a}$ is the acceleration of the object in the rotating reference frame, $\vec{v}$ is the body velocity in the rotating reference frame and $\vec{\omega}$ is the angular velocity of the rotating system. As the oscillation plane is rotated or moved by an external force, the response detected by the transducer is a direct result from the Coriolis term in the equation.

1.3.3 Inertial Sensor Errors

All inertial sensors are subject to errors which reduce their accuracy, since smaller measurements must overcome an inherent noise floor in order to be recognized. MEMS inertial sensors are particularly prone to various errors which are more complex when compared with their mechanical predecessors. Cost-efficient MEMS inertial sensors exhibit high bias offset, drift, scale factor and other noise characteristics [20]. These errors can be classified in two categories: (a) deterministic (or systematic) and (b) stochastic (or random) errors.

1.3.3.1 Deterministic errors

A. Bias offset

The bias in MEMS inertial sensors generally consists of two parts. The deterministic part is called bias offset, which refers to the offset in the inertial sensors measurements. It is defined as the output of the sensor when the input is zero, and is depicted in Fig. 1.7 [18]. The stochastic component of the bias is superimposed on the bias offset and can be regarded as white noise.

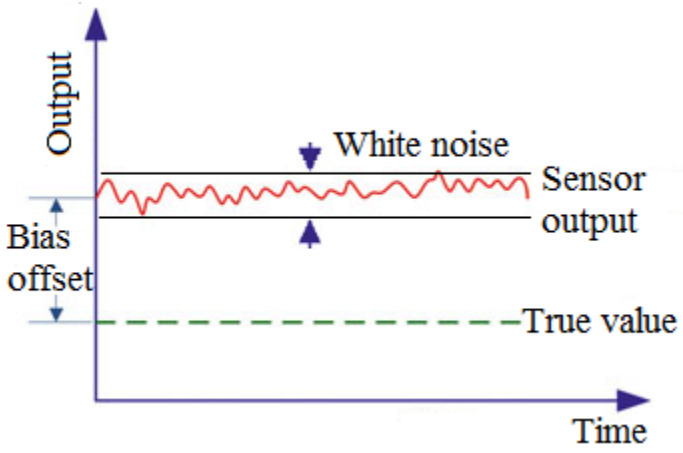


Figure 1.7 Inertial sensors bias offset error [18].

B. Scale factor error

The scale factor represents the relationship between the real physical quantities being measured and the output of the inertial sensors [20]. The stability of the scale factor reflects the capability of the system to accurately sense acceleration or rotation rate at different levels. The stability of the scale factor is presumed to quantify its variability with change in temperature and, respectively, its repeatability, and is usually expressed in parts per million (ppm). Fig.1.8 illustrates the effect of different scale factors, assuming their infinite stability (expressed as straight lines).

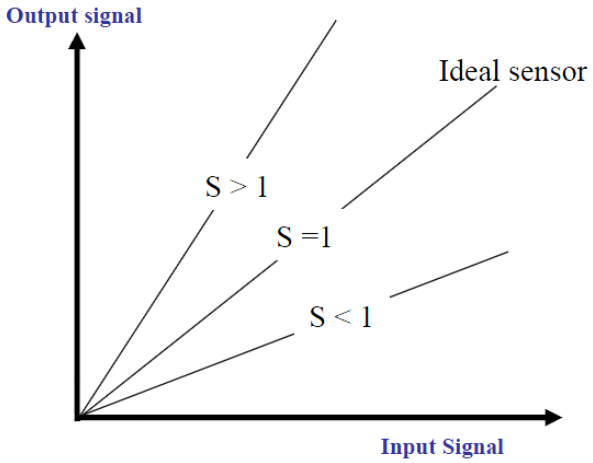


Figure 1.8 Inertial sensor scale factor error. Modified from [18].

C. Non-orthogonality errors

Non-orthogonality is the result of the imperfection in the mounting of the inertial sensors triad, the axes of which are departing from ideal mutual orthogonality. It usually happens at the time of manufacturing, but could occur at any point along the lifetime of the device. In either case the same problem arises in which each axis is seriously affected by the measurements of the other two axes. Fig. 1.9 illustrates a situation in which the z axis is deviated by an angle of θ_{zx} from the XZ plane and θ_{zy} from the YZ plane.

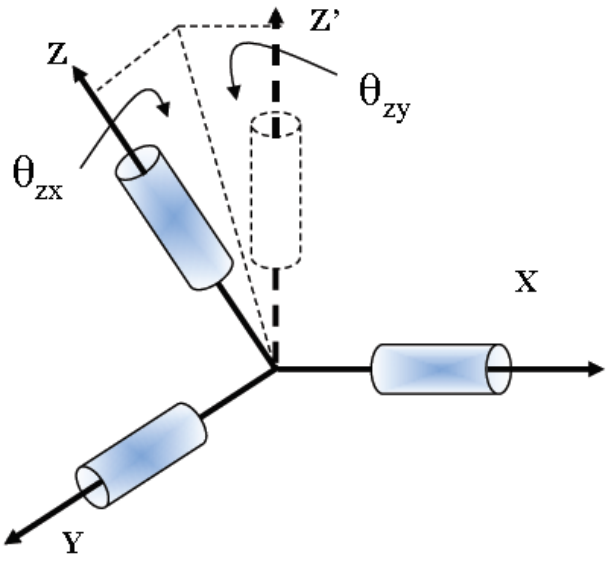


Figure 1.9 Inertial sensors axes non-orthogonality error. Modified from [18].

1.3.3.2 Stochastic errors

Low-cost MEMS inertial sensors suffer from wide variety of stochastic errors, which can only be modeled as stochastic processes to mitigate the interference [18].

A. Bias drift

The term drift is often used to describe the change that occurs during one run. Bias drift is a random change in the bias during a run over time as shown on Fig.1.10 [18]. Different from the deterministic bias offset and its superimposed white noise, bias drift is stochastic and has a direct impact on the stability of the output.

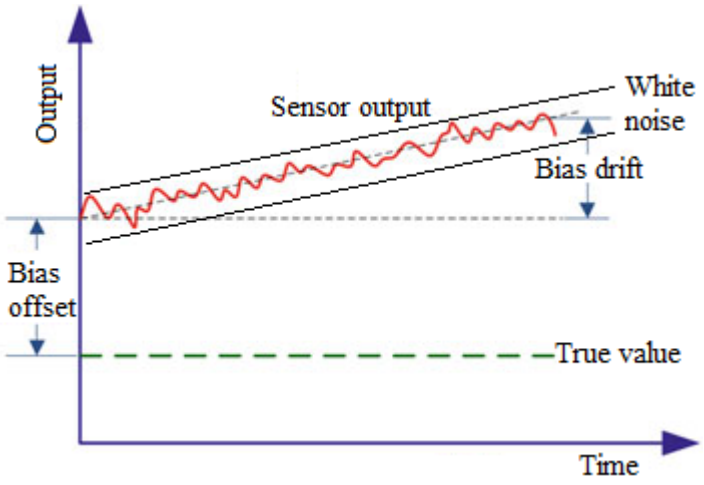


Figure 1.10 Bias drift in inertial sensors output. Modified from [18].

B. Scale factor drift

The scale factor can randomly change during a single run or from run to run due to temperature fluctuations or other external factors. This demonstrates the limited repeatability of the inertial sensors.

C. Stochastic noise

Unlike the deterministic noise, the random noise cannot be removed and it can only be modeled in order to mitigate its effect. The residual biases and drifts in the inertial sensors are usually correlated in time. Gauss process is a stochastic process consisting of random values describing the dynamics of a random variable x having normal distribution. If the Gaussian process is stationary and it has an exponential autocorrelation, it is called a Gauss-Markov (GM) process [21, 22]. The first order GM model is a decaying exponential autocorrelation sequence of the random variable x.

$$R_{xx}(\tau) = \sigma^2 e^{-\beta|\tau|} \tag{1.3}$$

where σ^2, τ and β are the variance, the time and the inverse of the correlation time of the process, respectively. Generally in order to approximately model the random errors of low-cost MEMS inertial sensors, a first order Gauss-Markov model is utilized [20]. Fig. 1.11 is a sketch of the autocorrelation function of a first-order Gauss-Markov process.

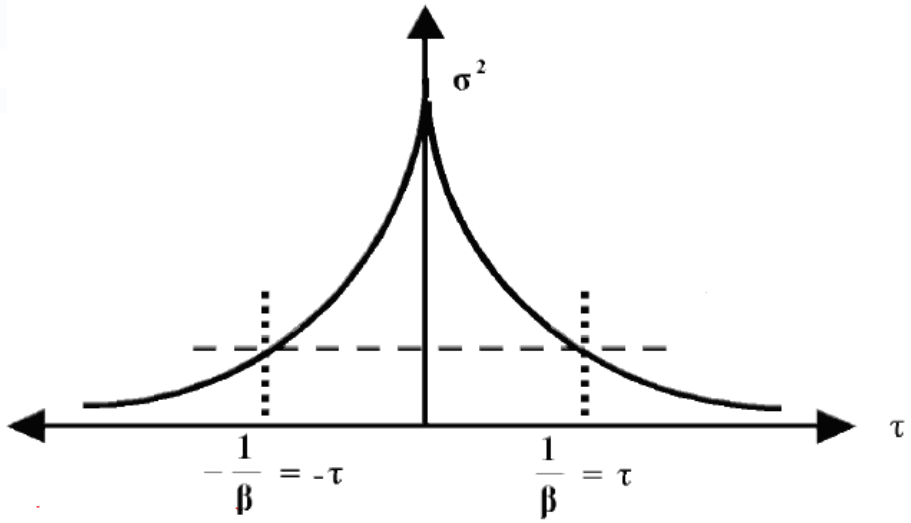


Figure 1.11 Sketch of an autocorrelation function for a first-order Gauss-Markov process.

The capability to precisely model the stochastic noise is critical as it allows further estimation techniques to work properly. Without utilizing estimation techniques such as adaptive filtering, the inherent stochastic noise will dramatically degrade the system performance in long-term navigation applications [18]. These estimation techniques will be discussed in details in Section 1.4.1.

1.3.4 Navigation equations

1.3.4.1 Coordinate Frames

Coordinate frames are used to precisely describe the position of the object in relation to a reference frame [18]. They are very useful as they are relevant to navigation and their mutual transformations are required for mechanization.

A. Body Frame (b-frame)

In most applications, the inertial sensors' axes are made to coincide with the axes of a moving platform in which the sensors are mounted. Measurement information from the inertial sensors is in the body frame (b-frame), which is a right-hand Cartesian coordinates-based frame. The b-frame is shown in Fig. 1.12, and it is defined as:

- The origin is in the center of gravity of the body;
- Y-axis is towards the forward direction;
- X-axis is towards the transverse direction.
- Z-axis is orthogonal to the other axes to complete the right-handed frame.

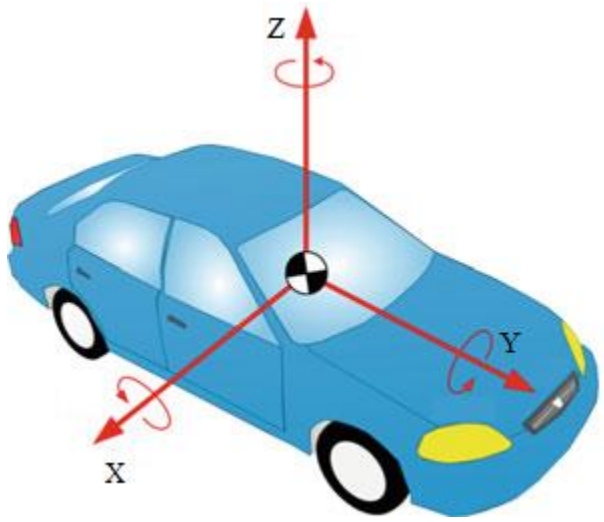


Figure 1.12 b-frame of a moving platform. Modified from [18].

Accelerations measured in the b-frame are converted to other coordinate frames by using the initial attitudes and the angular rate measurements.

B. Earth-Centered Inertial Frame (i-frame)

An inertial frame (i-frame) is considered to be either stationary or moving at constant velocity relative to distant galaxies [23]. Given the practical limitations of this definition for

navigation applications, the origin of the inertial frame is defined as the centre of the earth as shown in Fig. 1.13 [24]:

- X^i-axis is pointing towards the vernal equinox;
- Z^i-axis is identical to the direction of the earth's rotation axis defined by the Conventional Terrestrial Pole (CTP);
- Y^i-axis is orthogonal to the other axes to complete the right-handed frame.

All inertial sensors outputs are relative to an inertial frame resolved along the system's sensitive axes.

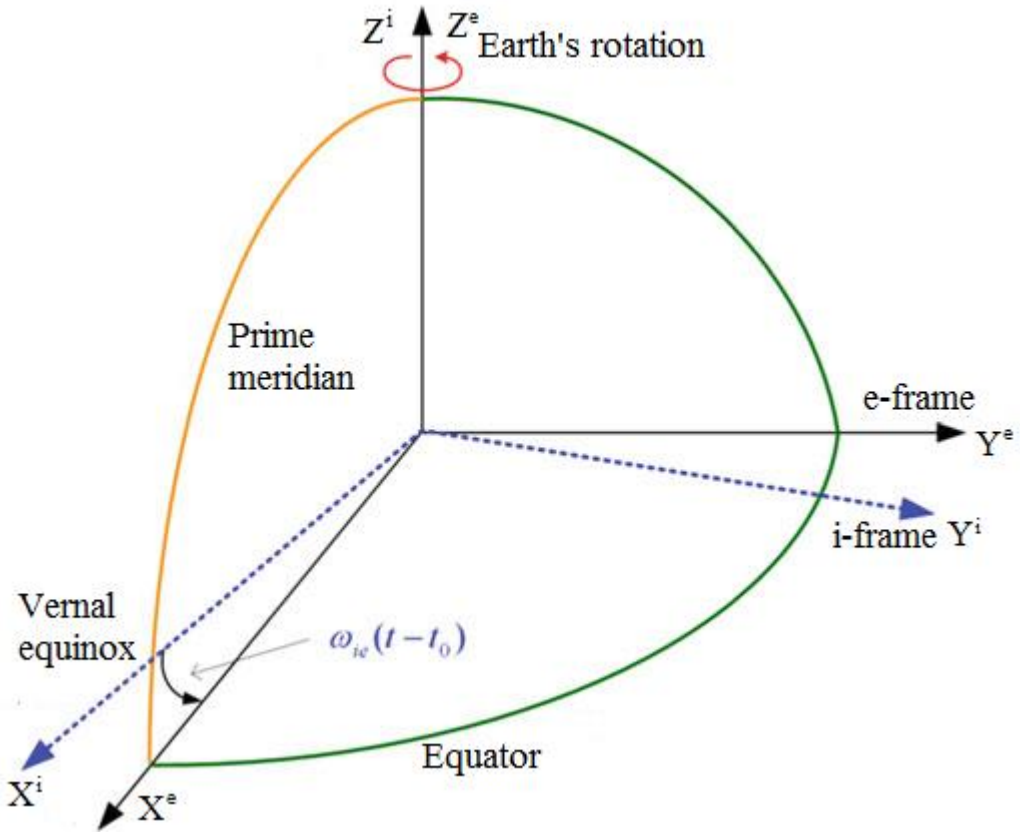


Figure 1.13 Illustration of the i-fram and the e-frame. Modified from [18].

C. Local Level Frame (l-frame)

The local level frame (l-frame), sometimes referred as navigation frame (n-frame), is utilized to represent an object's attitude and velocity when it is on the Earth [25]. A commonly used l-frame is defined as follow:

- The origin is located at the center of the inertial sensors
- Y-axis is pointing to true north (parallel to the tangent of the meridian).
- X-axis is pointing to east, perpendicular to North.
- Z-axis completes the right hand coordinate system by pointing up

This is the most common frame in subsurface navigation, since it is used with conventional magnetic based systems.

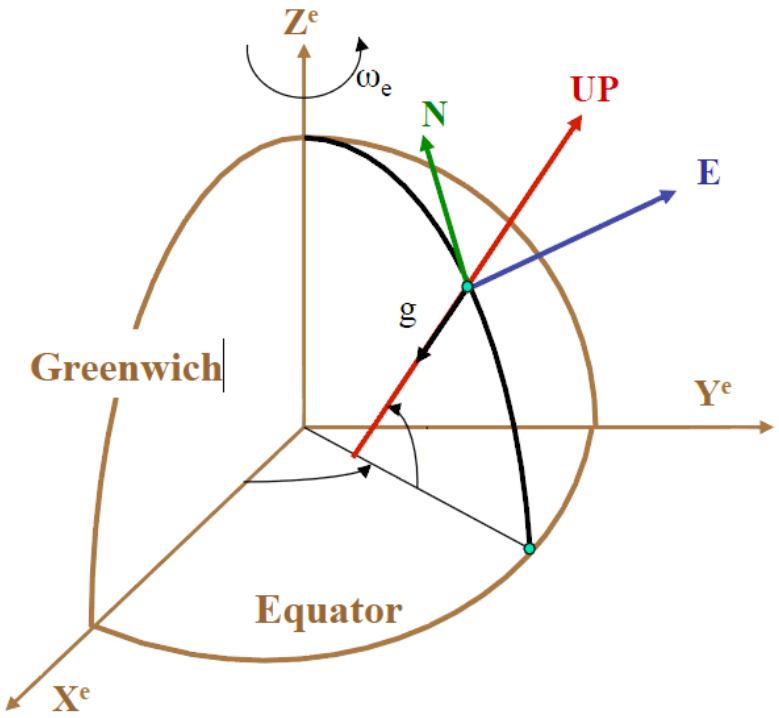


Figure 1.14 Illustration of the l-frame. Modified from [18].

D. Earth-Centered Earth-Fixed Frame (e-frame)

The Earth-Centered-Earth-Fixed frame or e-frame, is similar to the i-frame since it has the same origin located at the centre of mass of the Earth and a z-axis oriented along the spin of the Earth as shown in Fig. 1.13, but it rotates with the Earth as shown in the name Earth-fixed [25]:

- X-axis is in the equatorial plane and passing through the mean Greenwich meridian.
- Z-axis is parallel to the spin of the Earth and passing though the CTP.
- Y-axis: Orthogonal to the other axes to complete the right-handed frame.

Since the origin of the e-frame is the centre of Earth, comparatively, an object will have motion in this frame with respect to physical locations.

1.3.4.2 Inertial Navigation Algorithm Overview

The basic principle of inertial guidance is the utilization of Newton's 1st and 2nd laws of motion.

Newton's 1st law states that a body will continue its state of rest, or uniform motion in a straight line, unless it is compelled to change that state by forces impressed on it. For this law to apply, the body must be in the inertial reference frame, which is a non-rotating frame in which there are no inherent forces including gravity.

Newton's 2nd law states that the acceleration is proportional to the resultant force and is in the same direction as this force. The velocity and displacement can be derived from acceleration by performing one or more integrations given a known mass that is accelerating. Vice versa, velocity and acceleration can be calculated by differentiating the object's displacement.

Inertial navigation is based on the principle that differences in position can be determined by the double integration of acceleration, sensed as a function of time, in a specific stable

coordinate frame. In order to orient the acceleration to the same frame as the coordinate system, it is important to measure the object's acceleration in an i-frame before transforming it to the l-frame and integrating it with respect to time.

1.3.4.3 INS Mechanization in the local level frame

Mechanization is the process of computing useful information such as attitude, velocity and position by integrating the measurements from an IMU. The output of the inertial sensors includes 3D angular velocities and accelerations, all of which are in the i-frame. For INS applications the mechanization equations are desired in the l-frame for two important reasons:

1. The axes of the l-frame are aligned with local north, east and vertical directions, the attitude angles can be computed directly at the output of the mechanization equations. Thus, it is more intuitive to a user located on or near the Earth's surface;
2. The computational errors in the horizontal plane are bound by the Schuler effect, which stimulates the errors in the horizontal plane to produce the so-called Schuler loop [20]. This Schuler effect stimulates inertial system errors in the horizontal plane components that are coupled together to produce cyclic dynamics (Schuler loop), which results in error oscillations around certain frequency.

1.3.4.4 Position Mechanization Equations Overview

The position vector is described as $\mathbf{r}^l = [\varphi \quad \lambda \quad h]^T$

where φ is the latitude, λ is the longitude and h is the height. As the object moves on the surface of the Earth, the changing rate of its position is expressed in terms of its velocity in the north, east and vertical directions.

$$\begin{bmatrix} \dot{\varphi} \\ \dot{\lambda} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{R_M + h} & 0 \\ \frac{1}{(R_N + h)\cos\varphi} & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_e \\ v_n \\ v_u \end{bmatrix}^l \tag{1.4}$$

where R_M is the median radius of the ellipsoid of the Earth; R_N is the normal radius of the ellipsoid of the Earth; v_e is the component of the velocity in the East direction; v_n is the component of the velocity in the North direction; and v_u is the component of the velocity in the up direction.

$$\dot{\mathbf{r}} = D^{-1}\mathbf{v}^l \tag{1.5}$$

in which D^{-1} transforms the velocity vector from rectangular coordinates into curvilinear coordinates in the Earth-Centered Earth-Fixed frame.

1.3.4.5 Velocity Mechanization Equations Overview

The outputs of the three-axis accelerometers are measurements known as specific force given in the b-frame expressed as:

$$\mathbf{f}^b = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} \tag{1.6}$$

The transformation matrix R_b^l can be used to transform $\mathbf{f}^b$ to the l-frame:

$$\mathbf{f}^l = \begin{bmatrix} f^e \\ f^n \\ f^u \end{bmatrix} = R_b^l \mathbf{f}^b = R_b^l \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} \tag{1.7}$$

However, due to three important reasons, the accelerations in the l-frame cannot directly output the velocity components of the moving object [20]:

a. The Earth's gravity is also included in the measurement of the accelerometers;

b. The rotation of the Earth is around 15 °/hr;

c. Change of orientation of the l-frame while the object is moving

The three factors have to be taken into account in the mechanization equations:

$$\dot{\mathbf{v}} = R_b^l \mathbf{f}^b - (2\Omega_{ie}^l + \Omega_{el}^l)\mathbf{v}^l + \mathbf{g}^l \tag{1.8}$$

where $\mathbf{f}^b$ is the specific force measured by the accelerometers in the b-frame, $\mathbf{g}^l$ is the gravity vector in the l-frame. Ω_{ie}^l and Ω_{el}^l are skew-symmetric matrices corresponding to ω_{ie}^l and ω_{el}^l $(2\Omega_{ie}^l + \Omega_{el}^l)\mathbf{v}^l$ is also called Coriolis part [18].

1.3.4.6 Attitude Mechanization Equations Overview

The attitude angles of a moving object in the l-frame are determined by solving the time derivative of the transformation matrix R_b^l:

$$\dot{R}_b^l = R_b^l \Omega_{lb}^b \tag{1.9}$$

Moreover, the skew-symmetric matrix Ω_{lb}^b can be transformed into the l-frame as follows:

$$\Omega_{lb}^b = \Omega_{li}^b + \Omega_{ib}^b = -\Omega_{il}^b + \Omega_{ib}^b \tag{1.10}$$

Substitute Ω_{lb}^b in equation (1.9), and $\dot{R}_b^l$ can be obtained by solving the differential equations:

$$\dot{R}_b^l = R_b^l(\Omega_{ib}^b - \Omega_{il}^b) \tag{1.11}$$

The output of the gyroscopes contains the Earth rotation, the change in the orientation of the l-frame, and the rotation rate of the moving object. It is important to note that $\Omega_{il}^b = \Omega_{ie}^b + \Omega_{el}^b$, where Ω_{ie}^b represents the Earth rotation rate, and Ω_{el}^b represents the orientation of the l-frame

when object is moving. Therefore, the angular velocity Ω_{il}^{b} is subtracted from the Ω_{ib}^{b} to remove these effects.

1.3.4.7 Summary of INS mechanization in the local-level frame

The inputs of these mechanization equations are the measurements of the accelerometers and gyroscopes, while the outputs are curvilinear coordinates, velocity components, and attitude angles. The result can be summarized below [20]:

$$\begin{bmatrix} \dot{\mathbf{r}}^{l} \\ \dot{\mathbf{v}}^{l} \\ \dot{R}_{b}^{l} \end{bmatrix} = \begin{bmatrix} D^{-1}\mathbf{v}^{l} \\ R_{b}^{l}\mathbf{f}^{b} - \left(2\Omega_{ie}^{l} + \Omega_{el}^{l}\right)\mathbf{v}^{l} + \mathbf{g}^{l} \\ R_{b}^{l}\left(\Omega_{lb}^{b} - \Omega_{il}^{b}\right) \end{bmatrix} \tag{1.12}$$

1.3.5 Inertial navigation problems and limitations downhole

Many avenues to improve directional navigation performance downhole and to replace the magnetometers have been suggested [14], [26]. INSs are dead-reckoning autonomous systems that provide the position, velocity, and attitude of the platform on which they are mounted. They have been widely used in advanced ground and air navigation. On a stable platform, six inertial sensors (three orthogonal accelerometers and three orthogonal gyroscopes) are mounted on a set of gimbals, so that the platform (the b-frame) always remains aligned with the local level frame (the l-frame). Despite its superior performance, the gimballed INS is bulky and is a very costly technique. Therefore, the industry has adopted a strapdown INS (SDINS) approach which has been discussed in the previous sections. The main advantage of this method is its ability to provide navigation without the use of a mechanically stabilized platform. Conventionally, before the navigation starts, an initialization is performed to compute the initial attitude and the transformation matrix, which relates the b-frame to the l-frame. Previous studies

have theoretically and experimentally demonstrated that the strapdown INS-based surveying technique is an excellent candidate to replace magnetometer-based MWD [27], [28].

Despite its promise, the SDINS technique suffers greatly from the drift of the inertial sensors, and any errors within the measurements are significantly amplified due to the necessary INS integration mechanization procedures. Therefore, very small errors from the sensor measurements results in significant errors that increase rapidly with time [29]. To reduce this unlimited error growth in the INS, fine alignment is usually implemented by a Kalman Filter (KF) and must be performed repeatedly at certain intervals.

1.4 INS error reduction methods

1.4.1 Kalman filtering

The output of the INS mechanization drifts increasingly with time due to the implicit mathematical integration and the significant errors in the inertial sensors [18]. In order to minimize the INS errors, external measurements at regular time intervals should be utilized to determine these errors. Kalman filtering (KF) methodologies have been widely utilized to optimally estimate and compensate for the error states including position, velocity and attitude [30].

The first practical utilization of KF was in the Apollo program in 1960s for navigating the space shuttle to the moon. The KF is an algorithm to optimally estimate the error states of a specific system utilizing measurements corrupted by various noises. It is essentially a sequential recursive algorithm for an optimal least mean variance estimation of the error states [22]. Moreover, KF has the capability of providing real-time statistical data related to the estimation accuracy of the error states, which is useful for quantitative error analysis.

The most important KF characteristic is that it takes advantage of all available measurements, which are proportioned according to their reliability and precision.

1.4.1.1 Discrete time KF

The application of KF requires both the process and the measurement models of the entire system to be known and to be linear. The process model can be described as:

$$\mathbf{X}_k = \Phi_{k-1}\mathbf{X}_{k-1} + G_{k-1}\mathbf{w}_{k-1} \tag{1.13}$$

where $\mathbf{X}_k$ is the state vector, Φ_{k-1} is a state transition matrix; G_{k-1} is a noise coupling matrix; $\mathbf{w}_{k-1}$ is the noise vector of the process; k is the measurement epoch (interval).

The measurement model of the system is:

$$\mathbf{Z}_k = H_k\mathbf{X}_k + \boldsymbol{\eta}_k \tag{1.14}$$

where $\mathbf{Z}_k$ is the measurement vector of the system; H_k is the design matrix; $\boldsymbol{\eta}_k$ is the measurement noise.

Before implementation, Kalman filtering requires three assumptions [21]:

1) The system to be estimated can be described by a linear system.

2) The process and measurement noises are uncorrelated, zero-mean and white Gaussian random processes with known auto covariance functions.

3) The initial system state is a Gaussian random vector that is uncorrelated to measurement noise and the process.

$$E\left[\mathbf{w}_k\boldsymbol{\eta}_j^T\right] = 0, \forall k, j \tag{1.15}$$

$$E\left[\mathbf{w}_k\mathbf{w}_j^T\right] = Q_k, k = j;\ E\left[\mathbf{w}_k\mathbf{w}_j^T\right] = 0, k \neq j \tag{1.16}$$

$$E\left[\boldsymbol{\eta}_k\boldsymbol{\eta}_j^T\right] = R_k, k = j;\ E\left[\boldsymbol{\eta}_k\boldsymbol{\eta}_j^T\right] = 0, k \neq j \tag{1.17}$$

$$E\left[\mathbf{x}_0\boldsymbol{\eta}_j^T\right]=0; E\left[\mathbf{x}_0\mathbf{w}_j^T\right]=0, \forall j \tag{1.18}$$

where E represents the expectation; Q_k represents the covariance matrix of the system noise; R_k represents the covariance matrix of the measurement noise; $\mathbf{w}_k$ is the noise vector of the process at epoch k; $\mathbf{w}_j$ is the noise vector of the process at epoch j; $\boldsymbol{\eta}_k$ is the measurement noise at epoch k; $\boldsymbol{\eta}_j$ is the measurement noise at epoch j.

1.4.2 Zero-velocity Update (ZUPT)

The most widely used KF-based fine alignment technique for INS is the so-called zero-velocity update (ZUPT) because it utilizes velocity of 0 m/s as a reference measurement when the system is at rest [31]. During the routine resting periods of the drilling process, proper calibration and alignment associated with the system state vectors which describe the velocity and the attitude are achieved. It has been previously shown that the ZUPT alignment is very beneficial in reducing INS drift problems [32, 33]. However, it offers only limited benefit in eliminating azimuth errors which are important during horizontal drilling [20].

1.4.3 In-Drilling Alignment (IDA)

It has been shown theoretically that during ZUPT alignment, the velocities in the east and the north direction defining the azimuth plane are very weakly related to the misalignment in the vertical direction [27, 28]. Therefore, in a state space system, dynamic observability is very important for practically implementing low-error INS, as not all the error states can be measured directly and some states must be estimated [20].

A previously reported observability test demonstrated that when there is no motion induced on the IMU (i.e., during ZUPT), there are three unobservable error states, including the azimuth error state δA. When motion is induced on the IMU, all states are fully observable [34].

A nonstationary fine alignment method called In-Drilling Alignment (IDA) has already been proposed to address this insufficiency [34]. During the horizontal drilling process, IDA requires the induction of precisely controlled and observable motion on the IMU in the horizontal plane while the BHA is at rest. The dynamic observation can then be used as a reference measurement to align the IMU if the measurements from the induced motion have smaller variance than the routine IMU measurements. It has been shown that this method has theoretically and experimentally minimized the azimuth error growth when combined with KF-based optimal estimation [28], [34]. However, the IDA has high power requirements, utilizes complicated downhole enclosure, and most importantly, needs a large space to implement, which is prohibitive for real oil drilling.

1.5 Downhole data telemetry

Harsh downhole environments encountered in drilling applications present particular challenges, especially for reliable data transmission between the surface and the drilling devices due to the high pressure, high vibrations, extreme temperatures and huge distances downhole. At the bottom of the drill string is the BHA, which includes the drill bit along with electronic components such as sensors, controls and associated circuitry [35]. The sensors in the BHA typically include monitoring equipment for various properties of the formation and the fluid within it as well as navigation measurements. The processes that require information from such downhole sensors include MWD and Logging-While-Drilling (LWD) systems. Once relevant MWD/LWD downhole parameters are measured, there are three existing methods to bring the information to the surface currently used in industry.

1.5.1 Mud-pulse telemetry (MPT)

Mud-pulse telemetry (MPT) is the most common and standard method of data transmission downhole and is used by MWD/LWD tools developed in the 1970's [36]. The conventional MWD/LWD tool incorporates an electronic sensors package and a mudflow wellbore telemetry device. The drilling fluid called mud is pumped from the surface to the downhole BHA along the drill string. The mud serves as a cooling and lubricating circulation fluid and continuously carries the derbies back up to the ground during the drilling process. Mudflow wellbore telemetry device can selectively restrict the passage of the mud through the drill string to control and manipulate the pressure in the mud lines by operating a valve. These manipulations create pressure fluctuations which serially transmit information that is being encoded in binary format and propagated within the mud towards the surface, where it is registered by pressure transducers [36]. The problems with this type of data transmission are two: (1) low speed of the serial mud-based interface; and (2) high power consumption.

1.5.2 Electromagnetic telemetry (EMT)

Electromagnetic telemetry (EMT) employs a downhole current source to emit an electromagnetic signal into the formation. The signal can be detected and received at the surface as a small voltage drop between the top part of the BHA (the main drill string) and the bottom part of the BHA. Typically, the EM tool generates voltage differences between the drill string sections at a very low frequency, below 30 Hz [37]. The information is then converted into modulated EM waves by digital modulation. The typical transmission rate of the EMT tool is around 10 bits/sec. Compared to the MPT, the EMT method does not require changes in the major drilling parameters such as the rotation of the drill pipe and the mud flow rate, to send information to the surface. It also does not depend on the composition and the dynamics of the

mud flow. However, the EMT signal is severely attenuated during the data transmission that severely handicaps EMT's communication capability. Therefore, EMT tools are usually employed for certain specific applications such as underbalanced drilling (UBD) because of its immunity to drill fluid dynamics. Nevertheless, EMT can also attenuate dramatically in some types of formations, becoming unpredictable at depth of several thousand feet [37]. Its cost is also significant.

1.5.3 Wired cable data transmission

Since the beginning of the 21th century, some research has been focusing on the development of wired drill pipe system [38], [39]. It is based on the concept that composite drill pipes can also facilitate high-speed data transfer rates via special materials such as fiber optic cables embedded within the pipes during construction. A great benefit of such system is its superior data transmission rate which makes the real-time monitoring during downhole drilling completely feasible. The IntelliServ wired pipe, offering data rates upwards of 1 million bits/sec became commercially available in 2006 [38]. However, the pipe-embedded cables can cause reliability problems attributed to installation, connections and drilling fluid migration. In addition, the high capital costs and the lack of advanced technology in the present-day drill pipe manufacturing process is still limiting the wider applicability of this technique.

1.5.4 Summary

Since the data transmission rate of MPT and EMT is less than 100 bits/sec [39], most of the useful information provided by the MWD and LWD has to be stored in a memory logger associated with the downhole instrumentation near the drill bit and transmitted later. Therefore, the so-called "real-time MWD" can only be applied in a very limited scope, and only the wired

cable data transmission can be considered a real-time MWD method. Comparison of the commercial downhole transmission methods are given below:

Table 1.1 Comparison between different downhole data transmission methods

Transmission Method	**MPT**	**EMT**	**Wired Cable Drill Pipe**
Data Bandwidth[Bits/sec]	1-40 [36]	10-100 [39]	1,000,000 [38]
Reliability	Low	High	Low
Applications	Limited to UBD	Susceptible to formation	Wide
Working Depth[Meters]	Up to 12,000	Up to 3,000 [40]	Up to 15,000
Frequency [Hz]	< 100	< 30	N/A
Cost	Low	Medium	Very High

1.6 Aim of the thesis

The present thesis has two major aims:

(1) To introduce a miniaturized MEMS-based rotary In-Drilling Alignment (R-IDA) setup for directional drilling applications: and

(2) To propose a relay-based downhole wireless communication method based on multiple wireless transmission stations mounted along the pipelines, incorporating real-time data integrity testing and subsequent power consumption and/or communication optimization.

Chapter Two: **Methods**

2.1 Rotary In-Drilling Alignment (R-IDA)

2.1.1 Rotary Strapdown INS

Rotary Strapdown INS is an alignment technique to reduce the errors in the inertial sensors without utilizing other external information such as the one that might be provided by a satellite-based global positioning system (GPS). It was initially proposed by Levinson and Giovanni in 1980s [41], [42], and has seen successfully utilized in several military projects including MARLIN and MK39 [42] which have been integrated into naval worships and submarines. In rotary INS, an IMU with triad accelerometers and gyroscopes is installed on a rotary platform, and a rotary encoder is attached to it. The entire platform then rotates in a predesigned rotational configuration as shown in Fig. 2.1. The inertial sensors' errors can be modulated to periodic signals and removed during complete rotation cycles.

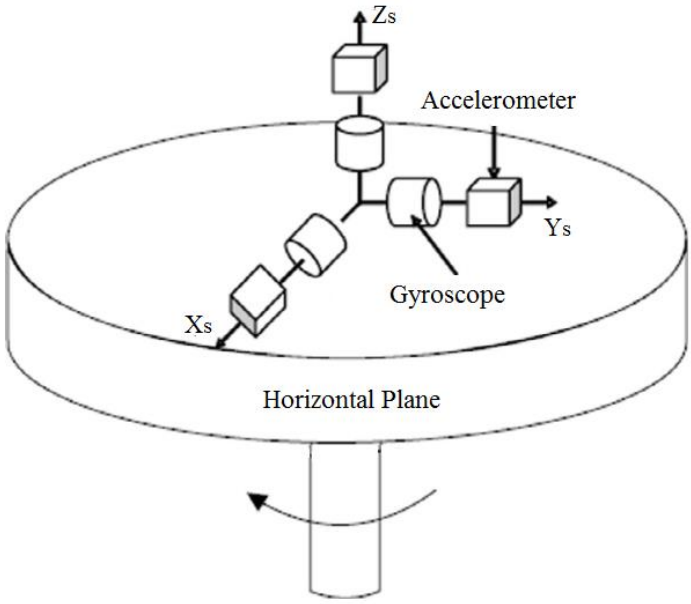


Figure 2.1 IMU rotation schematic

The MWD downhole process is quite similar to a submarine under the sea since both of them lack external information such as conventional GPS as a reference measurement. However, the submarine application utilizes either large mechanically stabilized gyroscopes or tactical grade FOGs-based IMUs, which are not applicable to downhole drilling due to the severely restricted space. Therefore we aimed at applying partial IDA [43] based on the rotary INS concept, to MEMS-based IMUs in downhole oil drilling applications.

2.1.2 R-IDA error modulation

R-IDA employs the same conventional INS mechanization except for a small modification, which is the utilization of a new IMU frame (also referred to as the inertial sensors frame, or s-frame). The new s-frame shares the same origin with the b-frame and changes simultaneously with the positions of the inertial sensors. Therefore, the data collected for the mechanization is in the s-frame. Since the IMU is not aligned with the b-frame due to its rotation, it requires a direction cosine matrix to transform IMU data from the s-frame to the b-frame.

Fig.2.2 demonstrates a rotation scheme in which the IMU is rotating counter clockwise (CCW) about the Z axis at constant rate ω in the b-frame in order to align the s-frame with the b-frame at time t, where t is the time since the reference epoch (Fig. 2.2). The transformation matrix from b-frame to s-frame, R_b^s, can be described as:

$$R_b^s = \begin{bmatrix} cos(\omega t) & sin(\omega t) & 0 \\ -sin(\omega t) & cos(\omega t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.1}$$

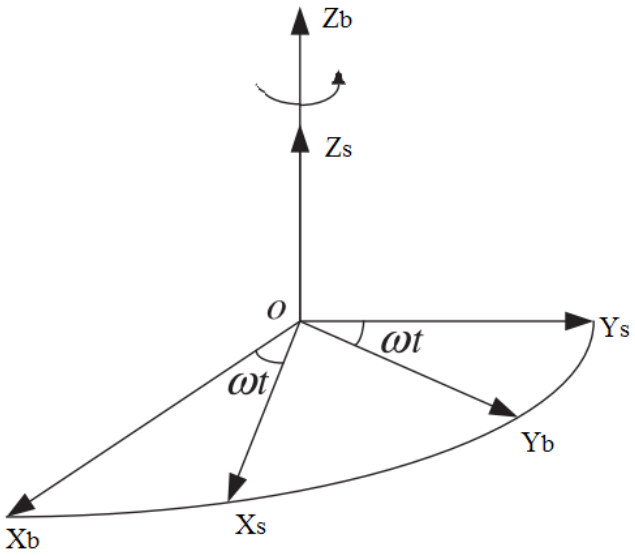


Figure 2.2 IMU CCW rotation scheme.

where ωt is the angle between the two pairs of axes (X_b, Y_b) and (X_s, Y_s) in the horizontal plane. Z_b, Y_b and X_b are the coordinates in the b-frame; Z_s, Y_s and X_s are the coordinates in the inertial sensor frame. In the horizontal plane the axes Y_b, X_b are shifted by an angle of ωt from Y_s, X_s.

Assume the b-frame is aligned with the l-frame for simplicity, and then the R_b^l is a 3 × 3 identity matrix. Transformation from the b-frame to the s-frame can be achieved through R_b^s, the inverse of R_s^b. Since rotation matrices are orthogonal,

$$R_b^s = \left(R_s^b\right)^{-1} = \left(R_s^b\right)^T \tag{2.2}$$

The inertial sensors' errors including gyroscope drift $\delta\boldsymbol{\omega}^l$ and accelerometer bias $\delta\mathbf{f}^l$ in the local level frame at time t can be expressed as:

$$\delta\boldsymbol{\omega}^l = R_s^n\delta\boldsymbol{\omega}^s = R_b^n R_s^b\delta\boldsymbol{\omega}^s = \left(R_b^s\right)^{-1}\delta\boldsymbol{\omega}^s = \begin{bmatrix} \delta\omega_x^s cos(\omega t) - \delta\omega_y^s sin(\omega t) \\ \delta\omega_x^s sin(\omega t) + \delta\omega_y^s cos(\omega t) \\ \delta\omega_z^s \end{bmatrix} \tag{2.3}$$

$$\delta \mathbf{f}^{l} = R_{s}^{n}\delta \mathbf{f}^{s} = R_{b}^{n}R_{s}^{b}\delta \mathbf{f}^{s} = \left(R_{b}^{s}\right)^{-1}\delta \mathbf{f}^{s} = \begin{bmatrix} \delta f_{x}^{s}cos(\omega t) - \delta f_{y}^{s}sin(\omega t) \\ \delta f_{x}^{s}sin(\omega t) + \delta f_{y}^{s}cos(\omega t) \\ \delta f_{z}^{s} \end{bmatrix} \quad (2.4)$$

Where $\delta \boldsymbol{\omega}^{s} = \begin{bmatrix} \delta\omega_{x}^{s} & \delta\omega_{y}^{s} & \delta\omega_{z}^{s} \end{bmatrix}^{T}$ and $\delta \mathbf{f}^{s} = \begin{bmatrix} \delta f_{x}^{s} & \delta f_{y}^{s} & \delta f_{z}^{s} \end{bmatrix}^{T}$ are the gyroscope drifts and accelerometer biases in the sensors frame, respectively;

$\delta \boldsymbol{\omega}^{l} = \begin{bmatrix} \delta\omega_{N}^{l} & \delta\omega_{E}^{l} & \delta\omega_{U}^{l} \end{bmatrix}^{T}$ and $\delta \mathbf{f}^{l} = \begin{bmatrix} \delta f_{N}^{l} & \delta f_{E}^{l} & \delta f_{U}^{l} \end{bmatrix}^{T}$ are represented in the local level frame.

According to equations 2.3 and 2.4, the gyroscope drift and accelerometer bias in the horizontal axes are modulated to periodic cosinusoidal signals in the local level frame, which limits the divergence of the calculated errors.

2.1.3 R-IDA Setup

The designed R-IDA experimental apparatus consisted of a stepper motor system, MEMS-based wireless IMU, horizontal drilling simulation device, and custom-built software that collected data and precisely controlled the induced rotation. These components are discussed individually.

2.1.3.1 Wireless IMU

A wireless MEMS-based IMU manufactured by Memsense LLC (Rapid City, SD, USA) was selected for the proposed laboratory design. Although its measurement and error-handling capabilities are less than modest, and its temperature range is completely insufficient for actual downhole drilling, it is ideal for laboratory testing. This IMU provides wireless real-time serial digital outputs of 3D accelerations, 3D angular rates of rotation and 3D magnetic field

measurements. The IMU communicates with a wide variety of devices including standard PCs and embedded systems with Bluetooth capability (Fig. 2.3).

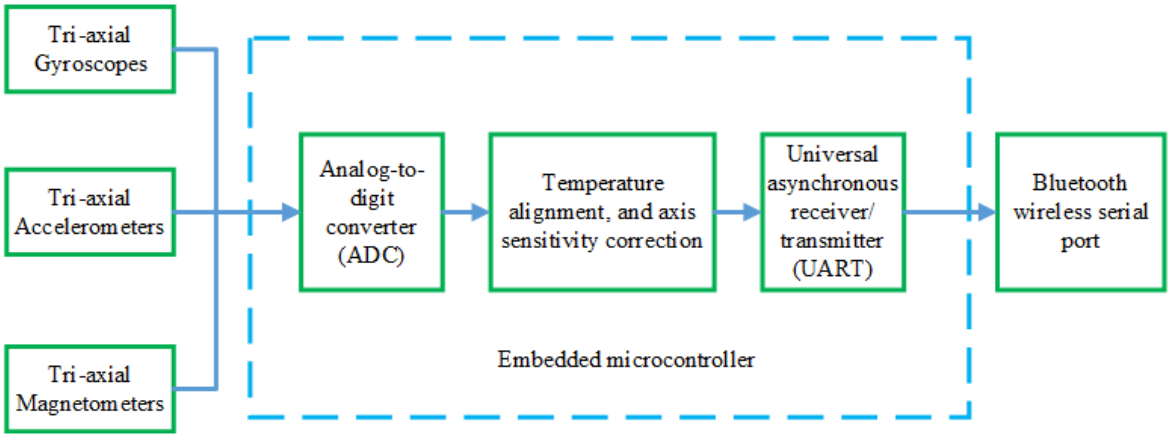


Figure 2.3 Functional block diagram of Memsense wireless IMU

The range of wireless transmission is typically up to 30 meters but can be affected by the environment. The angular velocity range is 600 °/s with a gyroscope bias instability of 30°/h and angle random walk (ARW) of $2.2°/\sqrt{h}$. The acceleration range is 5 g (g = 9.8 m/s^2) with a maximum bias of 30 mg. The system is very compact at 4.3 × 5.4 × 1.1 cm, weight of 5 grams, and uses 0.72 W. The power supply of this unit can be provided by three commercially available 9 V batteries from Energizer Holding Inc. St. Louis, MO, USA. With these features, this MEMS-based IMU is autonomous and is readily mountable for laboratory downhole applications.

2.1.3.2 Stepper motor system

A high resolution stepper motor (CMK245MBPA, Oriental-motor Ltd., Tokyo, Japan) equipped with a microstep driver (CMD2112P, Oriental-motor Ltd., Tokyo, Japan) and a controller (SCX-11, Oriental-motor Ltd., Tokyo, Japan) was selected for this design. The purpose of the stepper motor system was to induce a precisely controlled and accurately measurable reference rotation on the wireless IMU. Although the dual-axis rotary IDA offers

improved navigation performance, my research only focused on the single axis IMU rotary system due to its implementation simplicity, cost-effectiveness and high reliability. To improve the alignment of the azimuth error, the IMU was rotated in the horizontal north-east plane associated with the azimuth angle.

The MEMS wireless IMU was securely attached to a round table, which was rigidly connected to the forward shaft of the motor by a set of screws. Four pillars were used to support the motor and to level it in the horizontal plane. Three commercial 9-V batteries supplied power to the stepper motor controller and the microstep driver. The stepper motor generated strong vibrations and high heat due to internal friction during the operation. The IMU was very sensitive to vibrations and the stepper motor could not run for long durations at high temperatures, and to compensate for this, a microstep driver was used. It provided microsteps as small as 0.05625°, which significantly decreased vibrations and heat generation during operation. Using the microstep driver coupled to a controller, a variety of motion profiles could be designed by manipulating the number of input pulses and the frequency of the input signal to the controller. This offered a significant level of flexibility when various rotary patterns were needed. The microstep driver was compact (5.05 $\times$ 6.5 $\times$ 3.3 cm) and lightweight.

2.1.3.3 Optical rotary encoder

To provide an accurate reference measurement for the KF, a rotary optical encoder, E5 optical kit (US Digital Inc. Vancouver, WA, USA) with 1250 cycles per revolution (CPR), was mounted at the back of the stepper motor. The rotary encoder provided shaft rotation feedback by measuring the motion of the back shaft, which closed the controlling loop. The minimum measurable angle step was 0.072° with high precision. It measured the frequency and the total number of counts which were required for obtaining reference values for the angular velocity and

the rotational position. Fig. 2.4 illustrates the entire R-IDA apparatus. Please refer to Appendix A for related list of materials and details on the setup. Fig. 2.5 lists the dimensions of the designed R-IDA unit for downhole mounting.

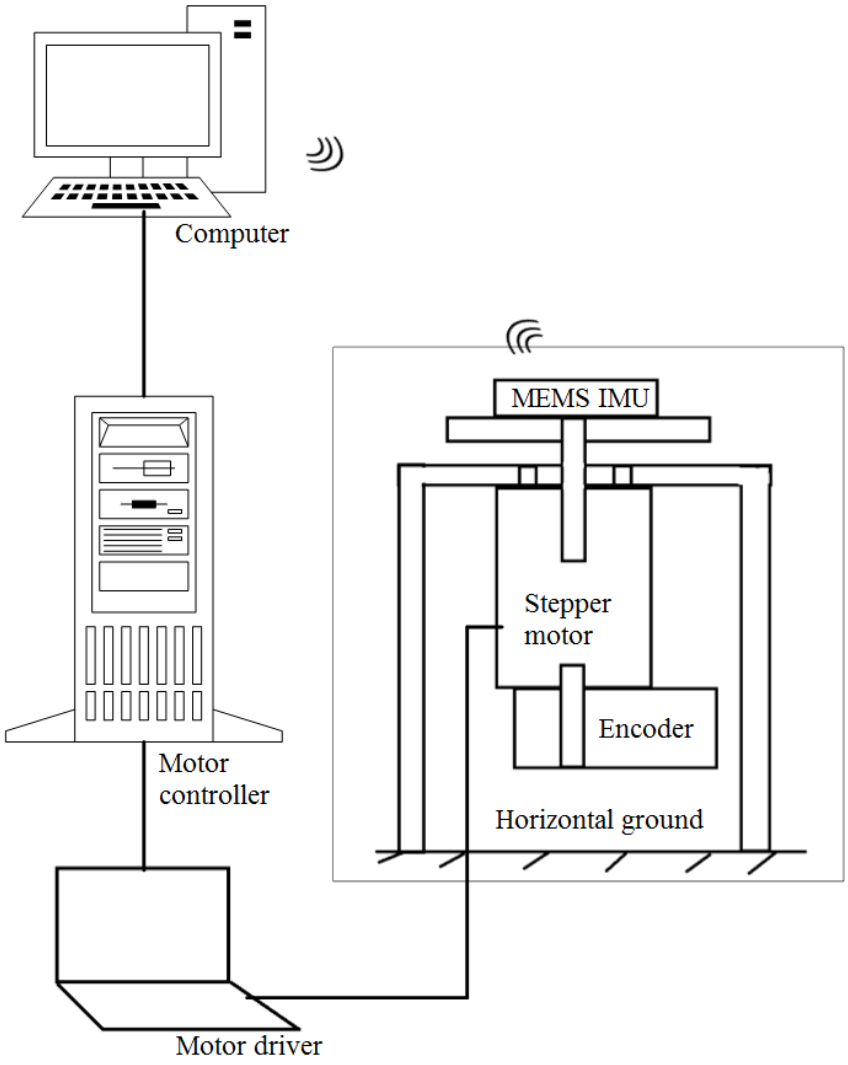


Figure 2.4 Block diagram of the designed stepper motor system.

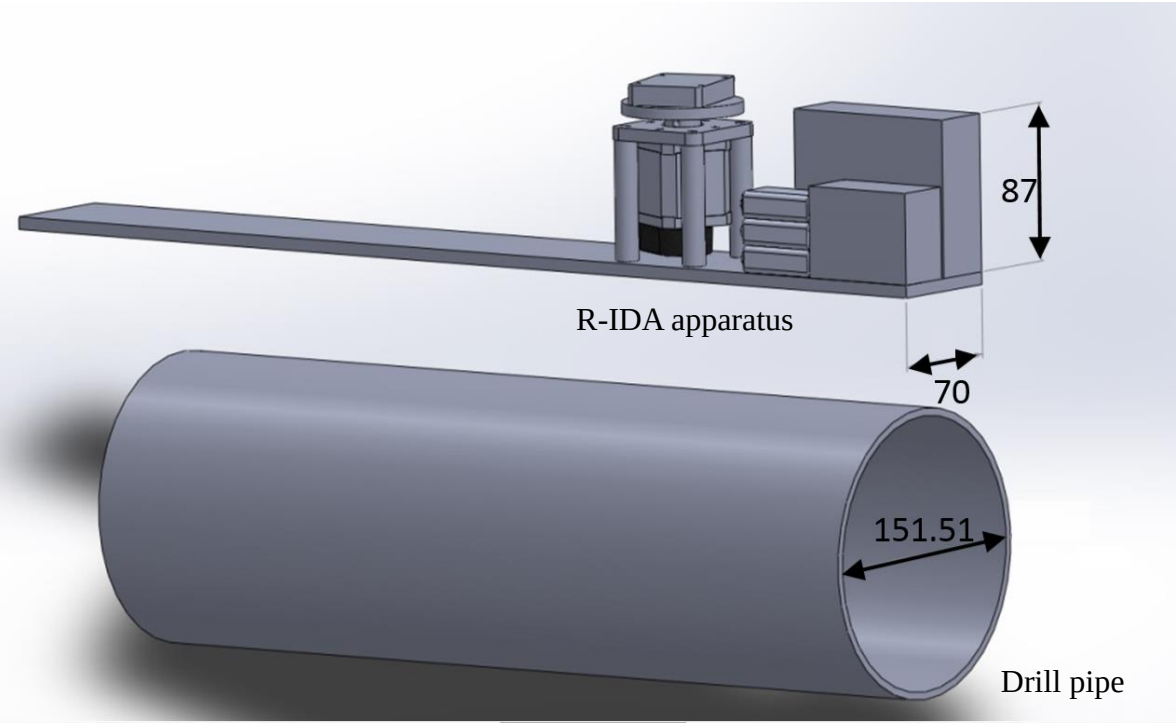


Figure 2.5 The R-IDA apparatus and a drill pipe. All dimensions are in mm.

2.1.3.4 Data acquisition (DAQ)

DAQ 1 is responsible for the control signal to the stepper motor and the feedback reference measurement from the rotary encoder and is in communication with PC1. Any accelerations and rotational rates measured by the wireless IMU were recorded by the IMU computer (PC2) via the DAQ 2 device at a sampling rate of 150 Hz. The logged IMU data were acquired in a Coarse alignment file and in a Navigation file. The Coarse alignment file included data from the wireless IMU when it remained static without any induced motion. The purpose of coarse alignment was to compute the initial attitude angles of the IMU combined with the built-in magnetometers. The Navigation file included dynamic wireless IMU measurements obtained as a result of the induced rotation by the stepper motor and utilized in the fine alignment process.

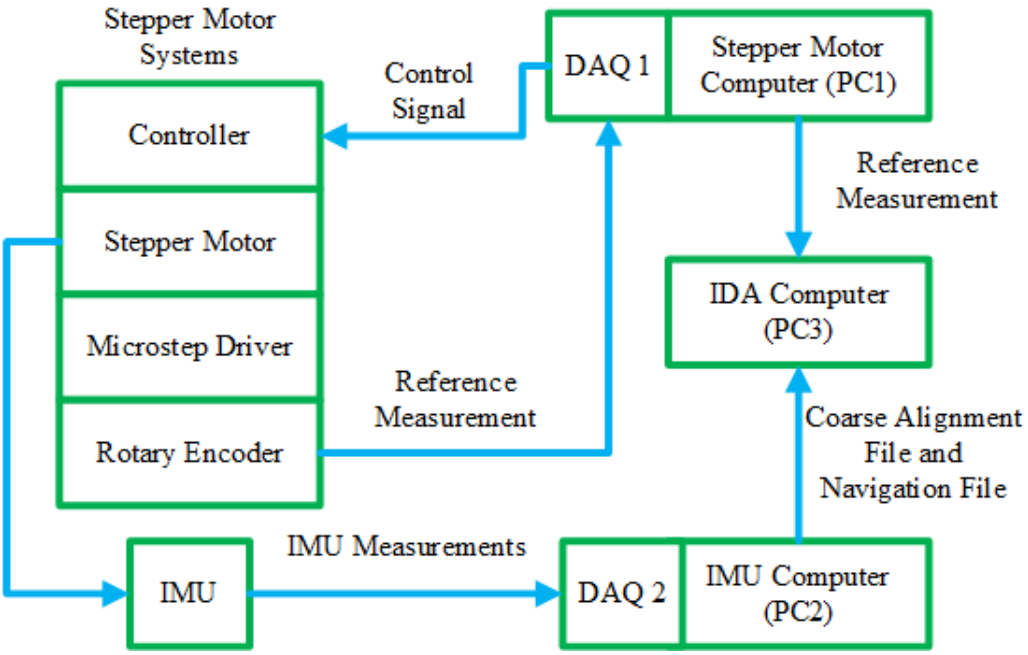


Figure 2.6 Block diagram of the experimental apparatus

Fig.2.6 is a block diagram of the experimental setup. The DAQ system consisted of three main parts: stepper motor control, acquisition of the reference signal, and acquisition of the wireless IMU measurements.

Custom software was developed for the SCX-11 controller to control the stepper motor system and the data acquisition of the reference measurements. The uncertainty of the reference measurements derived from the optical rotary sensor was calculated to be 0.07°/s, which was better than the manufacturer-specified Memsense wireless IMU uncertainty of 0.65°/s.

2.1.3.5 Horizontal directional simulation device

To validate the proposed R-IDA technique, a horizontal drilling simulation device [14] was utilized. The drilling simulation device was implemented with a flat turntable, on the top of which the pitch, roll and azimuth could be precisely controlled, and as such was able to provide controlled azimuth change in the horizontal plane. The pitch and roll angles were adjusted by

stepper motors that leveled the turntable as shown in Fig. 2.7. After the horizontal adjustment by setting the pitch and roll, the leveled plane was clamped and compactly fixed in order to provide very accurate reference measurements from the rotary encoder. The Memsense IMU was attached to the center of a round table, which was parallel to the flat turntable of the drilling simulation device.

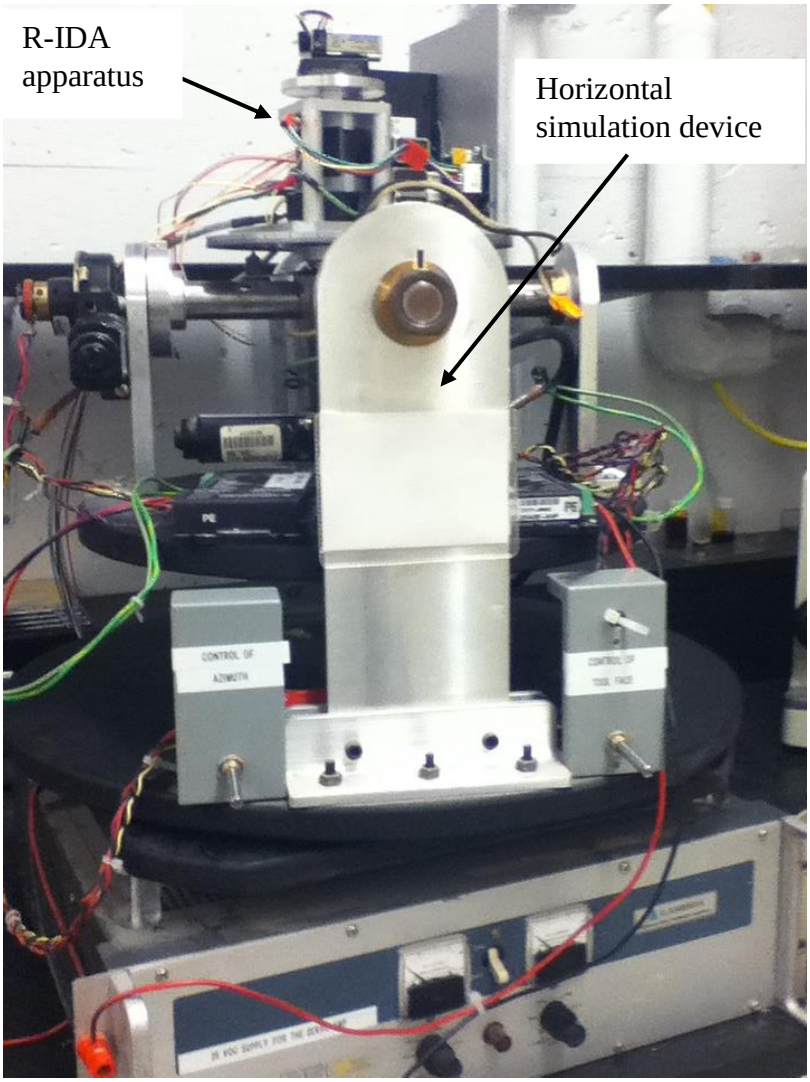


Figure 2.7 R-IDA setup on the platform of the horizontal directional simulation device.

2.1.4 Inertial sensors calibration

Precise calibration of the inertial sensors is very critical for navigation performance, especially for the MEMS-based IMU, since it has much larger deterministic errors when compared to tactical-grade IMUs. In this study, the traditional six-position test [44] was utilized for estimating the deterministic errors of the inertial sensors, including the bias and the scale factors of the gyroscopes and the accelerometers. The six-position method required the wireless IMU to be mounted on a levelled table with each sensitive axis pointing alternately up and down. The time to log the wireless IMU measurement data was 3 minutes, since the MEMS inertial sensors were very noisy and tended to drift as time passed.

The bias and scale factors can be calculated using the following equations:

$$Bias_{Accelerometer} = \frac{Up_f + Down_f}{2} \tag{2.5}$$

$$S_{Accelerometer} = \frac{Up_f - Down_f - 2 \times gravity}{2 \times gravity} \tag{2.6}$$

where $Down_f$ is the measurement when the sensitive axis is pointing downwards, and Up_f is the sensor measurement when the sensitive axis is pointing upwards.

Since low-cost MEMS gyroscopes suffer from bias instability and noise levels that can prevent measuring the earth's rotation rate, it was not feasible to use this measurement as a reference as the FOGs-based IMU commonly utilized. Instead, the table was rotated at a constant rate above the detection threshold of the gyroscopes clockwise (CW) and counterclockwise (CCW), and the outputs of the IMU were compared to these references. The bias factors of the gyroscopes were estimated using the same methodology applied to the accelerometers,

$$Bias_{Gyroscope} = \frac{W_{ref}(CW) + W_{ref}(CCW)}{2} \tag{2.7}$$

$$S_{Gyroscope} = \frac{W_{ref}(CW) - W_{ref}(CCW)}{2 \times W_{ref}} \tag{2.8}$$

where $W_{ref}(CW)$ represents the gyroscope output when rotating the turntable by W_{ref} in the CW direction; $W_{ref}(CCW)$ represents the gyroscope output when rotating the turntable by W_{ref} in the CCW direction; and $S_{Gyroscope}$ is the gyroscope scale factor.

To determine the initial azimuth angle the gyro-compassing method, which uses the Earth's rotation rate as reference, was not applicable for the MEMS-based IMU, and thus the magnetometers built in the wireless IMU needed to be calibrated. The quality of the magnetometer heading strongly depends on the tilt compensation (if the device is not in the horizontal plane) and the calibration procedure. In this study, the auto-calibration method [45] was utilized, based on the fact that the loci of the error-free magnetometer measurements form a circle if the magnetometers move around a circle in the local horizontal plane. Implementation of the auto-calibration method does not require any reference, but strongly depends on the surrounding environment and the effect of any ferromagnetic materials in the surrounding proximity, which are commonly known as hard iron and soft iron. This calibration requires rotating the wireless IMU in the horizontal azimuth plane in a circular pattern multiple times. The magnetometer-calibrated measurements M_x and M_y for the horizontal components are given with:

$$M_x = S_x M_x^{level} + B_x \tag{2.9}$$

$$M_y = S_y M_y^{level} + B_y \tag{2.10}$$

where S_x, S_y are scale factors, B_x, B_y are biases, and M_x^{level}, M_y^{level} are the leveled magnetic components of the earth.

$$S_x = Max\left(1, \frac{M_y^{Max} - M_y^{Min}}{M_x^{Max} - M_x^{Min}}\right) \tag{2.11}$$

$$S_y = Max\left(1, \frac{M_x^{Max} - M_x^{Min}}{M_y^{Max} - M_y^{Min}}\right) \tag{2.12}$$

$$B_x = \left(\frac{M_x^{Max} - M_x^{Min}}{2} - M_y^{Max}\right) S_x \tag{2.13}$$

$$B_y = \left(\frac{M_y^{Max} - M_y^{Min}}{2} - M_x^{Max}\right) S_y \tag{2.14}$$

where M_y^{Max}, M_x^{Max}, M_x^{Min}, M_y^{Min} are the maximum and minimum values of the measured magnetic field along the x- and y-axis of the IMU. It should be noted that the IMU axes in the body frame are directly related to the local-level frame axes by a transformation matrix [20].

2.1.5 Software

The R-IDA was implemented via custom-designed MATLAB (MathWorks Inc., Novi, MI, USA) software offline to simplify the data processing and KF tuning. Fig. 2.8 depicts the software implementation block diagram of the employed data processing procedure.

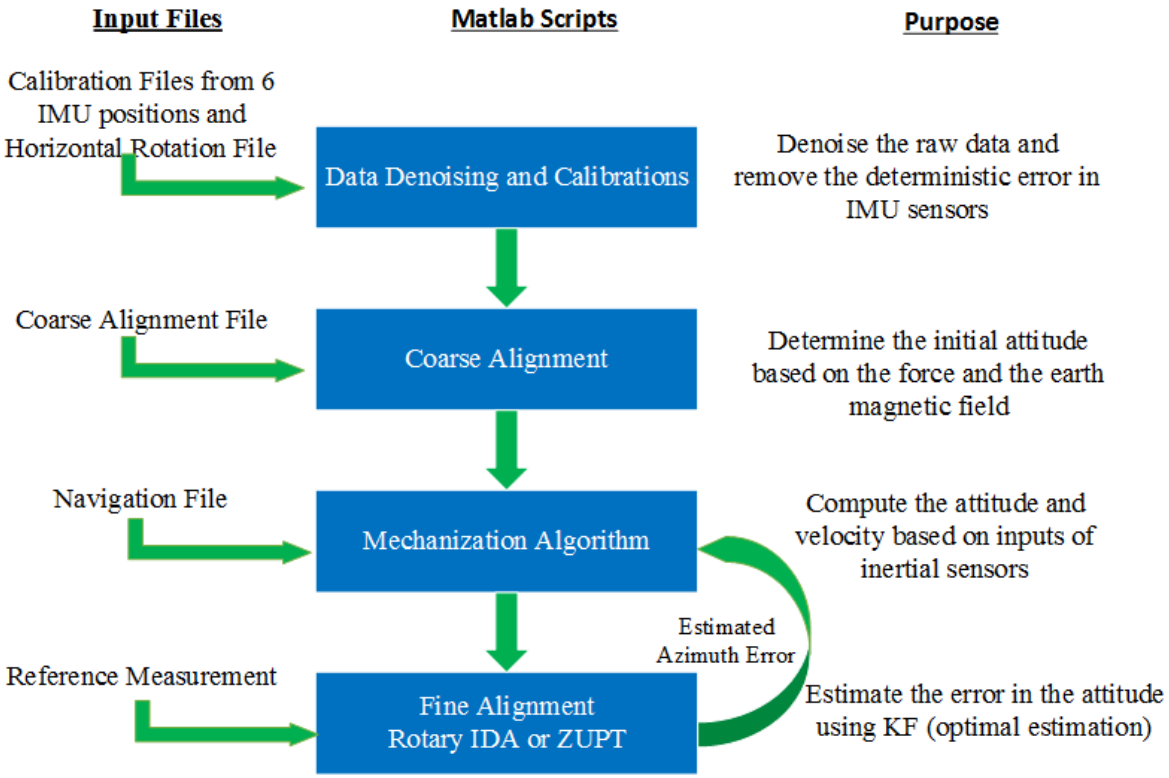


Figure 2.8 Block diagram for the software implementation of the data processing procedure

2.1.5.1 Sensor data denosing

Differing from a tactical grade FOGs-based IMU, MEMS inertial sensors always contain large amounts of measurement noise. At a sensor level, this noise is normally defined separately as low frequency components and high frequency components [46]. The high frequency components are associated with short-term noise, which consists of white noise and short-term sensor errors. The low frequency components are correlated with long-term noise, which includes white noise and object motion dynamics [47]. In this study, wavelet denoising analysis was selected to improve the signal to noise ratio (SNR) of the inertial sensors' raw measurements.

Wavelet analysis is performed for a number of decomposition levels that are chosen based on the sampling frequency of IMU. The number of the decomposition level should be chosen appropriately such that the process stops when its frequency band is reached. Since the wireless IMU sampling rate was 150 Hz, and most of the motion dynamics of objects ranged from 0 to 5 Hz [48], five decomposition levels were selected for this study.

This denoising process was carried out using the MATLAB Wavelet Toolbox. The *rigrsure* thresholding criterion, *soft* thresholding, and wavelet function *db5* were selected to perform the wavelet decomposition [47]. Fig.2.9 shows the considerable decrease in the noise level of one of the gyroscopes from the wireless IMU.

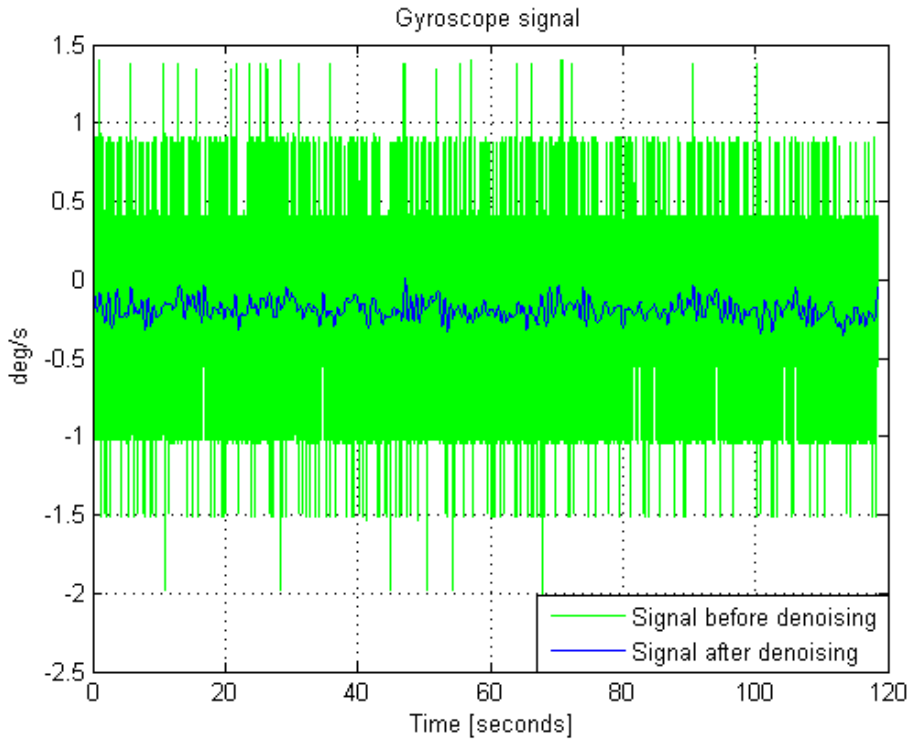


Figure 2.9 Wavelet denoising on the gyroscope signal

2.1.5.2 Coarse alignment

The purpose of coarse alignment is to compute the initial attitude of the wireless IMU in the body frame with respect to a reference frame. In this study, the outputs of the accelerometers were utilized to calculate the pitch and the roll angles, known also as the accelerometer leveling method, while the data from the magnetometers were used to calculate the azimuth angle. In the stepper motor-based experimental apparatus, the wireless IMU was mounted on the round table without inducing any rotation, while it recorded the measurements for a 2-min time period simultaneously. The logged data formed the Coarse alignment file.

MEMS accelerometers have better characteristics including noise level and stability compared to MEMS gyroscopes. The pitch p and the roll r angles can be derived from the measurements from the accelerometers using the gravitational force as:

$$p = \sin^{-1}\left(\frac{mean(f_x)}{f_{gravity}}\right) \tag{2.15}$$

$$r = \sin^{-1}\left(\frac{mean(f_y)}{f_{gravity}}\right) \tag{2.16}$$

where, $mean(f_x)$ and $mean(f_y)$ are the average forces in the x and y axis of the IMU body frame, respectively.

Because satellite-based global positioning system (GPS) signals are unavailable underground, the built-in magnetometers in the wireless IMU can be used to determine the absolute heading with respect to the local magnetic north and this can be done without any external equipment. Magnetic declination is the deviation between the local magnetic north and true north and can be calculated as a function of latitude, longitude and time using the global World Magnetic Model (WMM) [49]. The accuracy of the global model is 0.5°. This calculation

was carried out using free software from the website of the British Geological Survey [50]. The position of the experimental setup was latitude 51.07942°, longitude -114.13285°, and altitude 1107 m. The magnetic declination of the setup location is 14.76666 °E. If the magnetometers are in a leveled plane, the heading ψ can be calculated as:

$$\psi = \arctan 2(M_x, M_y) \tag{2.17}$$

where, M_x and M_y are the calibrated measurements of the horizontal components.

2.1.5.3 Mechanization Algorithm

The mechanization algorithm is performed in the local level frame due to its computational simplicity. It is based on a state space model that determines the wireless IMU's current position and attitude [18]. Dynamic inputs into the state space model include the change in the force measured by the accelerometers and the rotation rate measured by the gyroscopes during the sampling period. The change in position is added to the previously calculated position, and the initial position and attitude are obtained from the coarse alignment calculations.

Since the output of a wireless IMU is the real accelerations and angular rates in the i-frame, the raw measurements from the inertial sensors must be changed to their incremental components described as:

$$\Delta\tilde{v}^b = \tilde{f}^b \Delta t \tag{2.18}$$

$$\Delta\tilde{\theta}_{ib}^b = \tilde{\omega}_{ib}^b \Delta t \tag{2.19}$$

where, $\tilde{f}^b$ is the output of the accelerometers in m/sec^2; $\tilde{\omega}_{ib}^b$ is the output of the gyroscopes in radian/sec; $\Delta\tilde{v}^b$ is the velocity in m/s; $\Delta\tilde{\theta}_{ib}^b$ is the change in angular rate during time Δt; Δt is the sampling interval 1/150, which is the reciprocal of the sampling frequency.

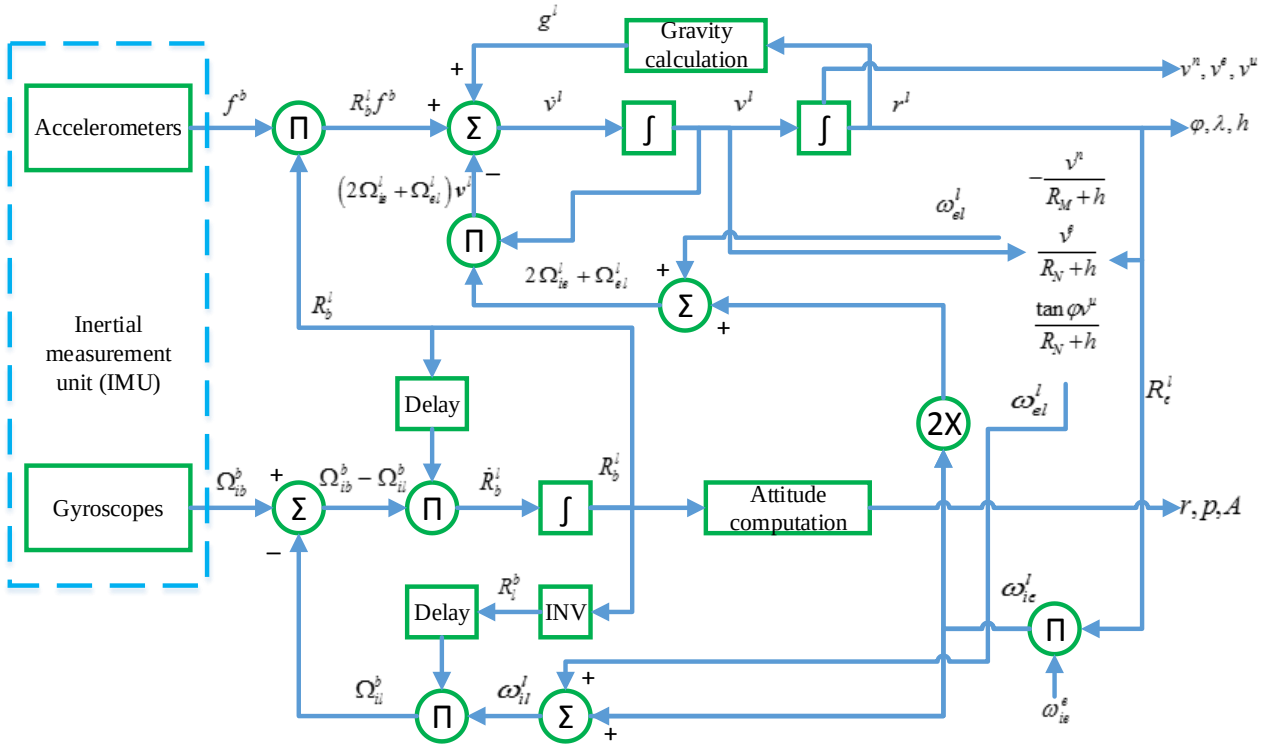


Figure 2.10 Detailed mechanization block diagram

Fig. 2.10 illustrates the detailed mechanization in the l-frame. After the wireless IMU is calibrated and denoised in laboratory conditions, the initial attitude angles of pitch, roll and azimuth in addition to the first transformation matrix R_b^l are computed through the coarse alignment process. The initial R_b^l, in conjunction with the gyroscope's measurements to compute and update of R_b^l. R_b^l is used to transform the specific forces in the b-frame to the l-frame. Meanwhile acceleration measurements are required to be corrected by removing the gravity and Coriolis forces. Next, the east, north and up velocities can be calculated by integrating the transformed specific forces. Finally, geodetic coordinates of the position are calculated by integrating the velocities. The mechanization equations in the l-frame, which were discussed in the 1.3.4, are reproduced here for reference.

$$\begin{bmatrix} \dot{\mathbf{r}}^l \\ \dot{\mathbf{v}}^l \\ \dot{R}_b^l \end{bmatrix} = \begin{bmatrix} D^{-1}\mathbf{v}^l \\ R_b^l \mathbf{f}^b - \left(2\Omega_{ie}^l + \Omega_{el}^l\right)\mathbf{v}^l + \mathbf{g}^l \\ R_b^l \left(\Omega_{lb}^b - \Omega_{il}^b\right) \end{bmatrix} \tag{2.20}$$

2.1.5.4 Fine alignment

Low-cost MEMS inertial sensors exhibit tremendous inherent systematic and random noise in their measurements. The purpose of the fine alignment was to dramatically minimize the accumulated error growth by optimal estimation. As discussed previously in section 1.4.1, a linear KF was used to optimally estimate these errors in the R-IDA method.

KF estimates the states of a system by operating as a recursive algorithm in a feedback loop. Since the model of the filter is known, the KF estimates the system states at an epoch k and then then received the reference feedback from external measurements. KF requires two phases for the implementation: prediction and measurement update.

A. Prediction.

The estimation of the state vector **x** at the epoch k only relies on time epoch k-1, is called a prediction $x_k(-)$, which is also called a priori estimation. Therefore, the best prediction of the state vector at epoch k is

$$\mathbf{x}_k(-) = \Phi_{k|k-1}\mathbf{x}_{k-1}(+) \tag{2.21}$$

where $\mathbf{x}_{k-1}(+)$ is the poteriori estimate, which is based on the dynamic behavior of the system, represented by the state transition matrix $\Phi_{k|k-1}$. At the same time, KF propagates the uncertainty of the system from the epoch k-1 to k. The error covariance $\mathrm{P}_k(-)$ is the expected value of the

variance of the error in the states, which represents the propagated uncertainty. $\mathrm{P}_k(-)$ is also called a priori covariance matrix.

$$\mathrm{P}_k(-) = \Phi_{k|k-1}\,\mathrm{P}_{k-1}(+)\Phi_{k|k-1}^T + G_{k-1}Q_{k-1}G_{k-1}^T \tag{2.22}$$

where $\mathrm{P}_{k-1}(+)$ is a posteriori estimation of covariance. $\mathrm{P}_k(-)$ is related with the posteriori covariance $\mathrm{P}_{k-1}(+)$ and the noise of the system. According to equation 2.21 and 2.22, the predicted state vector and predicted covariance matrix can be computed recursively.

B. Measurement update

After optimal prediction, KF will correct the state vector when the reference measurement is available. The crucial weighting parameter which is called the Kalman gain K is utilized.

$$K_k = P_k(-)H_k^T[H_k P_k(-)H_k^T + R_k]^{-1} \tag{2.23}$$

As shown in equation 2.23, K depends on a priori covariance matrix $P_k(-)$ and measurement noise R_k. When the measurement noise R_k is larger or the process noise $\mathrm{P}_k(-)$ is smaller, the K becomes smaller, which means the measurement is relatively less trustable and process prediction is more reliable. Once the Kalman gain is calculated, the state vector can be corrected as:

$$\mathbf{x}_k(+) = \mathbf{x}_k(-) + \mathrm{K}_k[Z_k - H_k\mathbf{x}_k(-)] \tag{2.24}$$

The covariance matrix representing the can be also updated by the KF, this results in improvement of its new prediction $\mathbf{x}_k(+)$.

$$\mathrm{P}_k(+) = [I - K_k H_k]P_k(-) \tag{2.25}$$

In directional drilling processes, there are periodic operational stops when a new drill pipe extension is added to the drill string. These stops provide approximately 2 minutes time

interval to perform KF-based IDA error compensation [51]. Twelve error states of the process were utilized in the present study for the simulated horizontal drilling process.

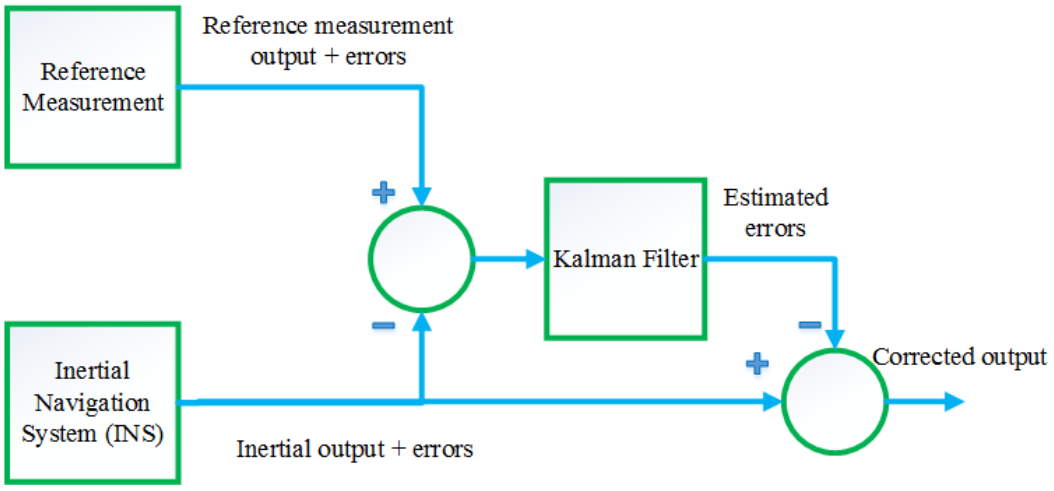


Figure 2.11 Block diagram of the experimental apparatus

As shown in Fig. 2.11, the input required to the KF is the difference between the reference measurement and the actual IMU measurement. The reference measurement is the feedback from the rotary encoder, which can provide the angular rate by calculating the frequency of the counts, while the actual IMU measurement is a result of the mechanization algorithm. Initializing the KF with reasonable values can prevent it from diverging [20]. Considering the characteristics of the utilized IMU and the geographic location, the initialization values for the KF are summarized in Table 2.1.

Table 2.1 Initial values for the Kalman filter

Estimated Variables	Initial Values
Velocity errors ($\delta V^e, \delta V^n, \delta V^u$)	0.2 m/s
Pitch and roll errors ($\delta p, \delta r$)	1.0°
Azimuth error (δA)	2.0°
Accelerometer biases ($\delta f^e\ \delta f^n\ \delta f^u$)	30 mg
Gyroscope drifts ($\delta b^e\ \delta b^n\ \delta b^u$)	30°/h

2.1.6 R-IDA experimental procedure for one run

The following experimental procedure was utilized for collecting each data set. Since the calibration of the wireless IMU was carried out only once at the beginning, it was not part of the procedure.

- The wireless IMU was fastened to the round table of the stepper motor, and the rotational IDA apparatus was placed on the horizontal drilling simulation device. Water level calibration was applied to check if the IMU and the azimuth plane were leveled.
- The round table and the IMU were aligned to a predetermined azimuth angle.
- Stationary IMU measurements were logged via DAQ 2 on PC2 for a period of 2 minutes. These measurements comprised the Coarse Alignment File.
- IMU dynamic measurements were logged on PC2 via DAQ 2. These measurements comprised the Navigation File.
- Motion was induced on the IMU for 2 minutes at a certain stable angular rate by PC1 and the reference rotation rate was recorded by the DAQ1 through PC1.
- Data logging of the navigation and the motion of the stepper motor was terminated.

The coarse alignment, navigation and induced angular rate data were transferred to PC3. Custom-developed Matlab software was applied to optimally estimate the azimuth error.

2.2 ZigBee-based wireless data transmission

2.2.1 Downhole wireless transmission options

A wireless communication system driven by a downhole battery could allow intelligent transmission sensors to be placed anywhere, avoiding the need for cables to supply power or to transmit data. Currently, the oil industry does not employ mature wireless communication systems downhole. However, recent advances in this area demonstrate its huge potential benefits. The concept of acoustic wireless communication has existed for several decades and it came to the verge of being commercialized. Wireless acoustic telemetry, based on the propagation of

stress waves along the drill pipe, requires less power than conventional systems such as the EMT and MPT [52]. It also shows a potential at higher data transmission rate capability of 50-100 bits/sec through the drill pipe channel [37]. However, it is still of relatively low frequency (400-2000Hz) and slow transmission rate, which can hamper real time monitoring. Furthermore, it is very susceptible to drill string interferences, since it is an acoustic transmission channel, where many passbands and stopbands occur [37]. As a result, reflected and transmitted acoustic signals interfere to a point where they are totally scrambled.

Other commonly used standard wireless RF communication methods including WiFi (IEEE 802.11) and Bluetooth (IEEE 802.15.1) protocols have unprecedented transmission rates operating at the 2.4 GHz frequency. However, they are not applicable downhole due to power consumption, reliability, limitations, complexity etc. [53].

2.2.2 Proposed downhole wireless transmission

ZigBee protocol station-based data transmission is proposed for downhole wireless communication. ZigBee was introduced in 1998, standardized in 2003, and revised in 2006, aiming at creating a highly reliable, cost-effective, secure, low power global wireless data transmission standard [54]. To meet the low-power and low-cost criteria, a relatively low data transmission rate is a constraint. Nevertheless, it is still very large at 250,000 bits/sec compared to MPT or EMT with maximum of 100 bits/sec. Not only the modern and simple specifications of IEEE protocol 802.15.4 make it reliable, but the mesh networking feature further enhances its reliability. With mesh networking, data from the first node is able to reach any other node within the allowable distance in the ZigBee network. The market price for low-volume sales is around a few dollars, which is low enough for the oil industry. Moreover, as the scope of the 802.15.4 market grows, some have predicted that this radio market will hit the bottom (below $ 1 in

quantity) in the next 3-5 years [54]. ZigBee can operate for years on a pair of AA batteries due to its low power consumption. One of the reasons for this low power feature is that radios and microcontrollers can sleep, since a node on the ZigBee network does not require a constant contact with the network to remain part of it. ZigBee-based wireless modules are disposed in the hollow interior of a drill pipe, and separated by some desired distances. Downhole data is transmitted by an initial transmitter at the BHA, relayed though numerous closed autonomous wireless modules, all powered by local batteries, and eventually received by the receiver on the surface [55].

Another key benefit of such a system is that it allows the entire drill string to act as a local network within the BHA, and many downhole tools located somewhere inside the drill string can be individually addressed and/or turned on and off.

2.2.3 R-IDA design for multilateral drilling navigation

For lateral drilling applications, a capsule-based, high-performance IMU wireless module has been proposed in the R-IDA context. The capsule includes a microcontroller unit (MCU), wireless radio frequency (RF) module and high performance IMU, all of which are finally integrated on a printed circuit board (PCB) to be packaged for downhole mounting. The stepper motor-based system [56] rotates the capsule freely at a certain speed during the scheduled operational stops. The data from the wireless MEMS IMU module are transmitted to an embedded wireless RF module residing at an appropriate distance. The RF module then automatically sends the received data to another module. By setting a number of wireless RF modules within the entire drill string at predetermined distances from each other, the wireless IMU measurements are transmitted along the drill string towards the surface. A computer equipped with a receiver at the surface captures and decodes downhole data and runs the Kalman

filtering-based navigation algorithms to compute attitude and position of the BHA in real-time. Preliminary test has been conducted in a field testing facility located in Crossfield, Alberta to determine ZigBee-based XBP's capability of communicating downhole.

2.2.4 Wireless MEMS-based IMU module design on a breadboard

In order to improve the bias, and drift performance of the MEMS IMU sensors, as well as to enhance temperature range applicability, high performance Analog Devices ADIS16488A (Analog Devices Inc., Norwood, MA, USA) has been used instead of the Memsense wireless IMU. This IMU provides tactical grade precision of the gyroscope measurements. The angular velocity range is 450°/s with a gyroscope in-run bias instability of 5.1°/h and angle random walk (ARW) of $0.26°/\sqrt{h}$. The system is very compact at $47 \times 44 \times 14$ mm, 48 g weight, and uses 0.8 W [57]. The temperature resistance standalone is as high as 105℃, which in conjunction with its low power consumption and small size, presents it as a very desirable candidate for multilateral drilling applications. However, the ADIS16488A has a wired cable connection interface based on the Serial Peripheral Interface (SPI) protocol, which is prohibitive for R-IDA applications. At the IMU end, it is proposed to employ an XBee Pro Series 1 (Digi International Inc., MN, USA) RF wireless module. The selected wireless module is ZigBee-based protocol with an asynchronous interface while the IMU's interface is synchronous. To allow these two different interfaces to communicate in real time, a low-power MBED microcontroller LPC11U24 (ARM Holdings plc, England) with an ARM-Cortex M0 processor embedded is incorporated. The role of the microcontroller is to perform an SPI-to-ZigBee interface translation operation since the ADIS16488A's interface is SPI-based while the XBee Pro S1 wireless module is Universal Asynchronous Receiver/Transmitter (UART)-based. At the computer end, free open source Tera Term software (Tera Term Project, Japan) was installed to display and log the IMU

measurements in real time through the wired connection with the XBee Explorer USB (Sparkfun Electronics Inc., CO, USA), which acted as a receiver. The block diagram in Fig.2.12 [58] illustrates the high-level wireless design suggested in this study. Fig. 2.13 is the actual view of the designed wireless navigation module downhole.

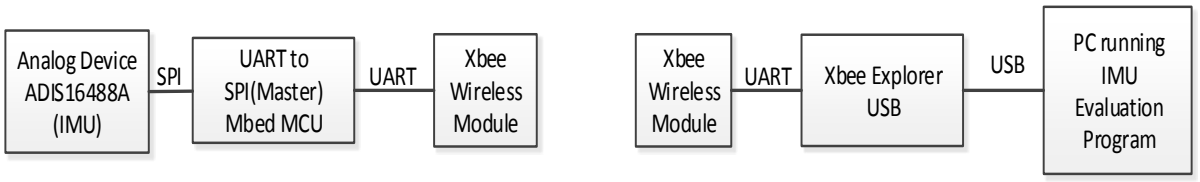


Figure 2.12 IMU-to-evaluation system wireless communication block diagram

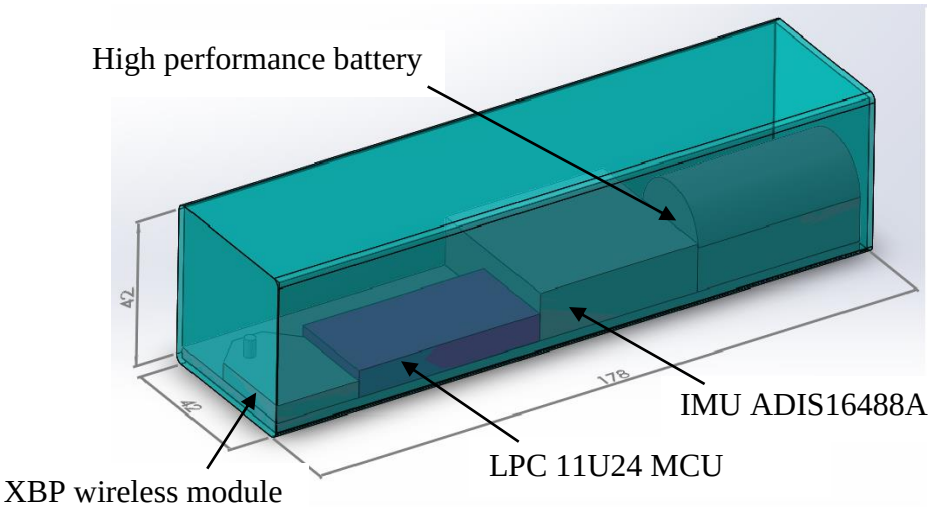


Figure 2.13 Design of wireless MEMS-based IMU navigation module. All dimensions are in mm. Modified from [58].

2.2.4.1 Analog Device ADIS16488A IMU signals and connections

The illustration in Fig. 2.14 is the diagram of a typical microcontroller connection. The microcontroller selected must have the SPI master interface in order to control and to communicate with the IMU. The IMU is an SPI slave device which is required to interface with a SPI master.

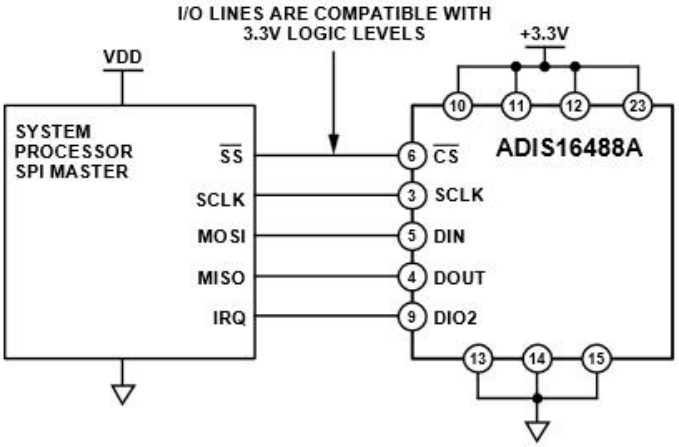


Figure 2.14 Microcontroller to ADIS16488A connection diagram [57]

Additionally, to allow for wireless data transmission, UART-to-SPI master module and SPI slave-to-UART modules are required. These two modules can be implemented using embedded C programming language and can be executed on a microcontroller which allows the two wireless transceivers to communicate.

2.2.4.2 XBee Pro Series 1 Wireless Transceivers

The XBee Pro Series 1 (XBP) is a wireless unit based on the ZigBee stack. It requires 3.3V with maximum current draw of 215mA, which makes it ideal for this autonomously powered solution. The output data of ADIS16488A includes 3 gyroscope, 3 accelerometer and 3 magnetometer registers. Since each register is 32-bits based, the total amount of data rate is 32 bits $\times$ 9 = 288 bits of data per sample. For any RF link, the loss will be around 10% to 20% of the data bandwidth due to overhead, protocol, etc. Assuming the worst-case scenario, to be conservative, the top end of 250 kbps along with a 20% protocol is used. This gives us 200 kbps of usable bandwidth in the 2.4 GHz band. With 288 bits per sample and a 200 kilo sample/sec data bandwidth, this is still 200 kbps/288 = 694 Hz data. Since the multilateral penetration rate is relatively low, 5 Hz sample rate for MEMS IMU is adequate for accurate downhole navigation.

Therefore, the date rate of 200 kbps is far more than the output data measurement rates of ADIS16488A. Two identical units are required; one to be used at the IMU end and the other to be used at the evaluation system end. Fig. 2.15 is a photo of the XBP module:

Figure 2.15 Xbee Pro S1 wireless module

To wire up the XBee Pro S1 module to the mbed microcontroller, only 5-pin connection is required. These pins are: VDD, GND, TX, RX and Reset. Because the XBP has 2 mm pitch sockets, a breakout board is required to increase the spacing to the standard 0.1" pitch. The XBP radio transmission requires a good stable power source for reliable transmission. To power up the XBP, two AA batteries were used, since the operating voltage for the XBP module is between 2.8-3.4V (VDD). The VDD was connected to XBP pin 1. The GND is connected to XBP pin 10. The reset line is connected to pin 5 of the XBP and the other end of the reset line can be connected to any *digital out* pin of the mbed MCU. The *receive data in* (RX) and *transmit data out* (TX) lines of the XBP module are connected to one of the serial ports on the mbed MCU. Please refer to Appendix B for configuration and setup.

2.2.4.3 Low-power microcontroller and interface connections

In 2012 ARM released a second MBED platform, the MBED MCU LPC11U24 based on the ARM Cortex-M0 microcontroller architecture. This microcontroller has the lowest power consumption in the mbed family which makes it ideal for battery-powered portable applications.

Recent tests showed the device drawing 16 mA while awake and less than 2 mA while asleep [59]. The purpose of the mbed MCU is to act as a SPI-to-UART bridge, which bridges the IMU and the XBP module. Pins 9 and 10 are the two main interface pins for the XBP transceiver (Fig. 2.16).

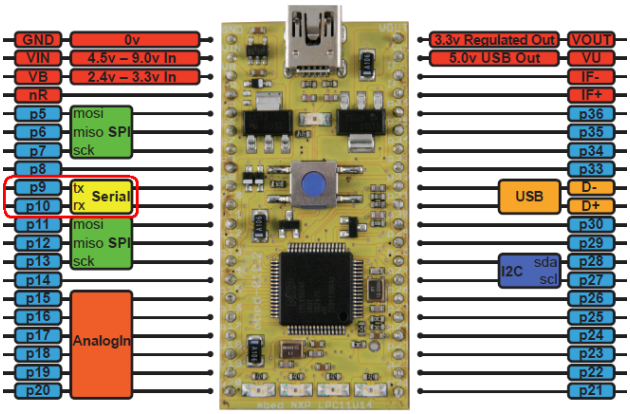


Figure 2.16 LPC11U24 low power mbed pin-outs

For the IMU SPI connections, the mbed microcontroller comes with two built-in SPI interfaces. The pins used are: p5, p6 and p7 and the general purpose I/O pin as the SPI Slave selection signal pin for the use of IMU interface. Alternatively, SPI ports with pin numbers: p11, p12, p13 can be utilized normally, which are reserved for the SPI-based Micro SD card interface. The general-purpose pin p14 is used for the slave select signal. This microcontroller resides at the IMU end because it is the lowest power version from the mbed microcontroller family. Please refer to Appendix B for the setup. The real wireless navigation IMU module system as shown in Fig. 2.17 will also incorporate a downhole battery on a PCB in a casing package. Table 2.2 is a summary of the components for the wireless navigation system.

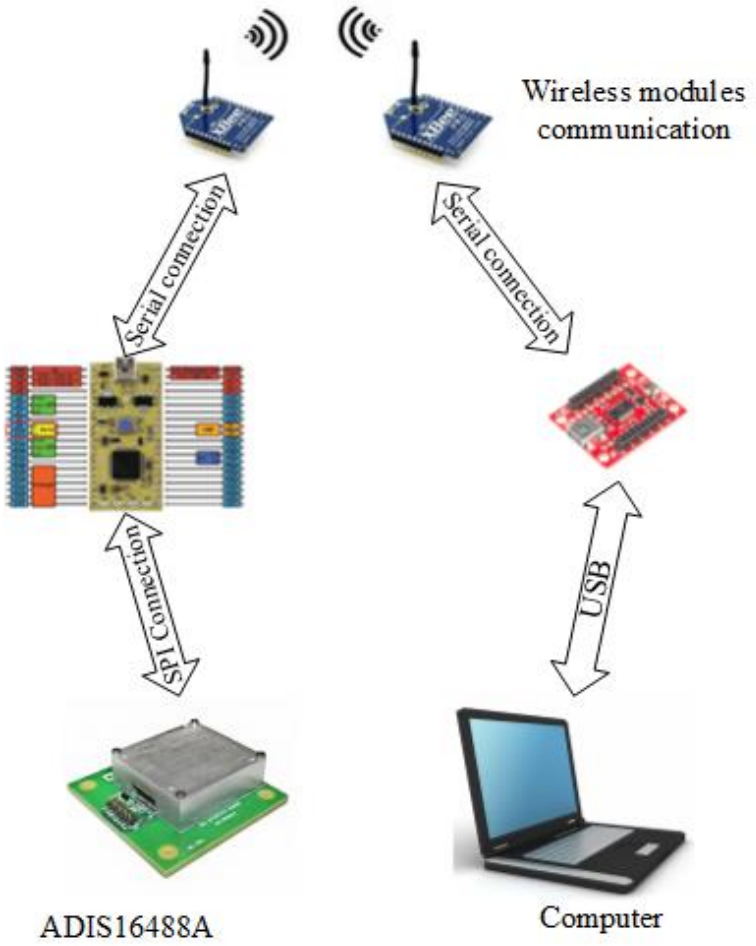


Figure 2.17 Complete connection diagram with MCU, wireless XBP, PC and MEMS IMU

Table 2.2 List of components, their manufacturer, and temperature ranges

Name	**Manufacturer**	**Temperature Ranges(℃)**	**Price(USD)**
ADIS16488A	Analog Devices Inc.	-55 to 105	1200-1500
XBee Pro Series 1	Digi International Inc.	-40 to 85	40
NXP LPC11U24	MBED	-40 to 85	60
Breakout board for IMU	Analog Devices Inc.	-40 to 85	70
Breakout board for XBP	Sparkfun Electronics Inc.	-40 to 85	5
Xbee Explorer USB	Sparkfun Electronics Inc.	-40 to 85	30

EVAL-ADIS	Analog Devices Inc.	-40 to 85	470

2.2.5 Printed circuit board design of Wireless MEMS-based IMU Module

The high performance downhole cell Lithium 33-127-165MR Series, provided by Excell Battery Company (Calgary, AB, Canada), are selected for the PCB module's power supply due to its small dimension, high performance downhole especially in capacity and operating temperature range as listed in Table 2.3. Fig. 2.18 is the schematic design for the wireless IMU system. A standard header was used for the connection of the downhole battery.

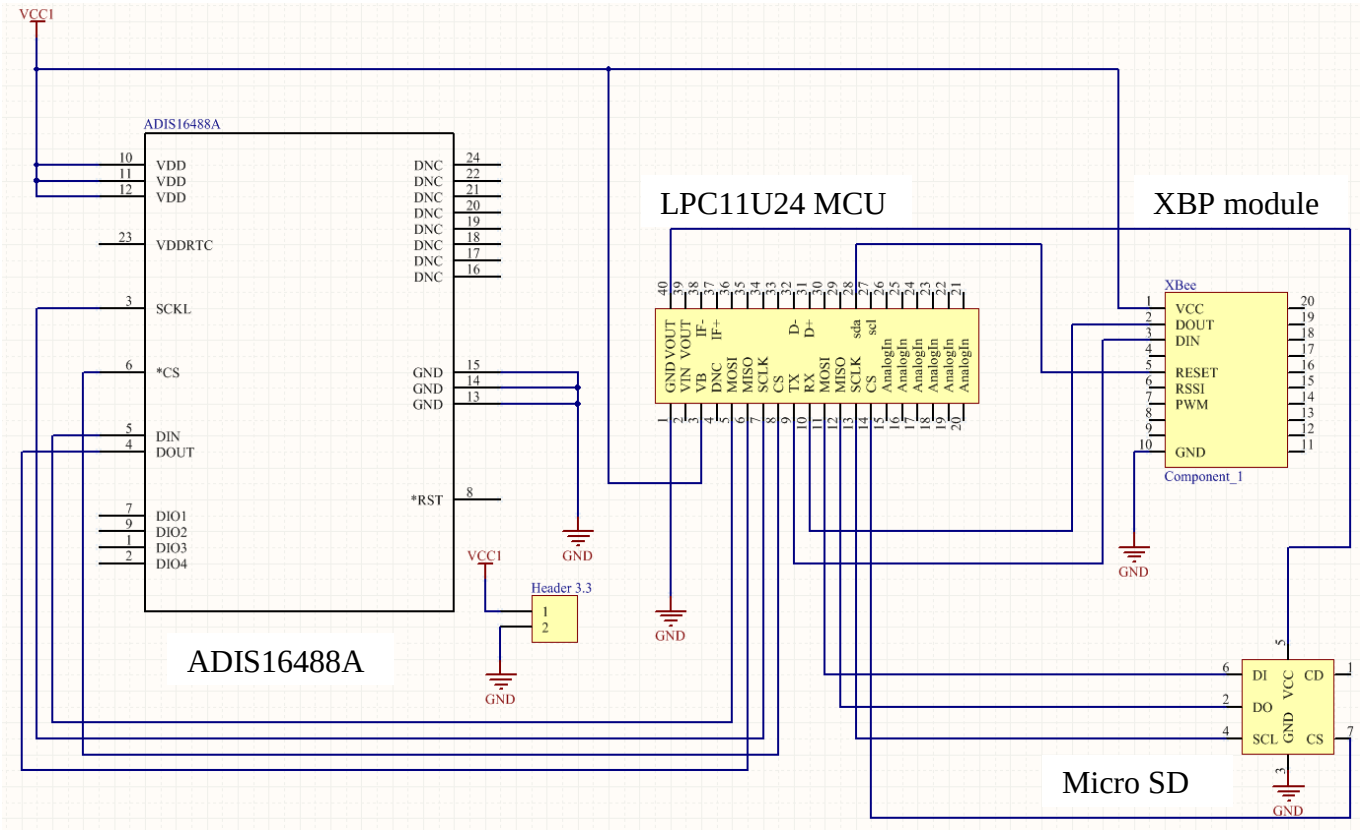


Figure 2.18 Schematic of the wireless ADIS16488A IMU system

Table 2.3 Technical overview of the downhole battery 33-127-165MR

Open Circuit Voltage (25℃)	3.67V
Rated Capacity	25 Ah
Max current to achieve 100% capacity	550 mA
Cell diameter	31.75 mm
Cell length	125.1 mm
Operating temperature	-40 to 165℃

Manufacturing data including Gerber files for each layer and drill statics were generated in Altium Designer. These files were submitted to the Alberta Printed Circuits Inc (Calgary, AB, Canada) for the manufacture. Fig. 2.19 depicts the design of the PCB's different layers and Fig. 2.20 shows the real version of the board prototype. The final design includes the casing of the downhole battery with the wireless navigation module prototype on PCB.

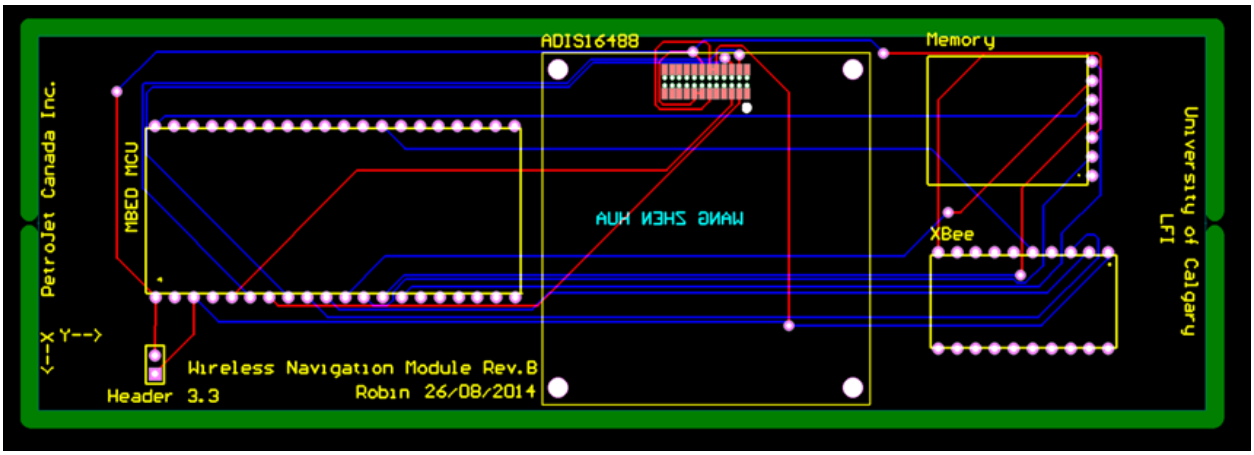


Figure 2.19 Layers design of the PCB

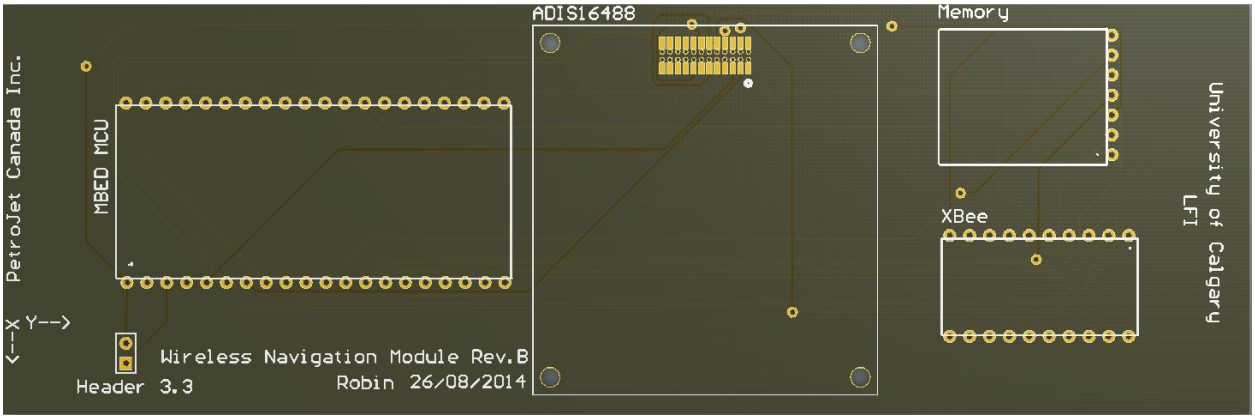


Figure 2.20 Actual view of the PCB design

Chapter Three: **Results**

3.1 MEMS-based R-IDA results

3.1.1 Output data denoising

The noise characteristics of an MEMS-based IMU can make accurate measurements impossible without first reducing noise levels. The most common methods, including the low pass filtering, are not capable of removing high frequency noise levels, which contain short-term inertial sensors errors and white noise. However, wavelet-based filtering has been shown to be effective for this task [47].

The majority of the motion dynamics of an object ranges from 0 to 5 Hz [48]. Data from the Memsense wireless IMU is delivered at a sampling frequency of 150 Hz, and in order to separate and reduce high and low frequency noise components, five levels of decomposition (LOD) are adequate. Increasing the LOD past this range will lead to the loss of the desired information, which results in poor navigational performance. Analysis of the x-axis gyroscope and accelerometer signals was done in order to demonstrate the effect of wavelet denoising. As shown in Fig. 3.1 and Fig. 3.2, noise levels were significantly reduced in both the x-axis gyroscope and accelerometer. This is of greater importance for the gyroscope, as it generally has higher noise levels than accelerometers and its fluctuation in the designed experimental setup was from - 1.7° to 1°, which could potentially deteriorate navigation accuracy. Standard deviation of the x-axis gyroscope signal before wavelet filtering was $\sigma = 0.4131$, however, it decreased to $\sigma = 0.0817$ after wavelet-based denoising.

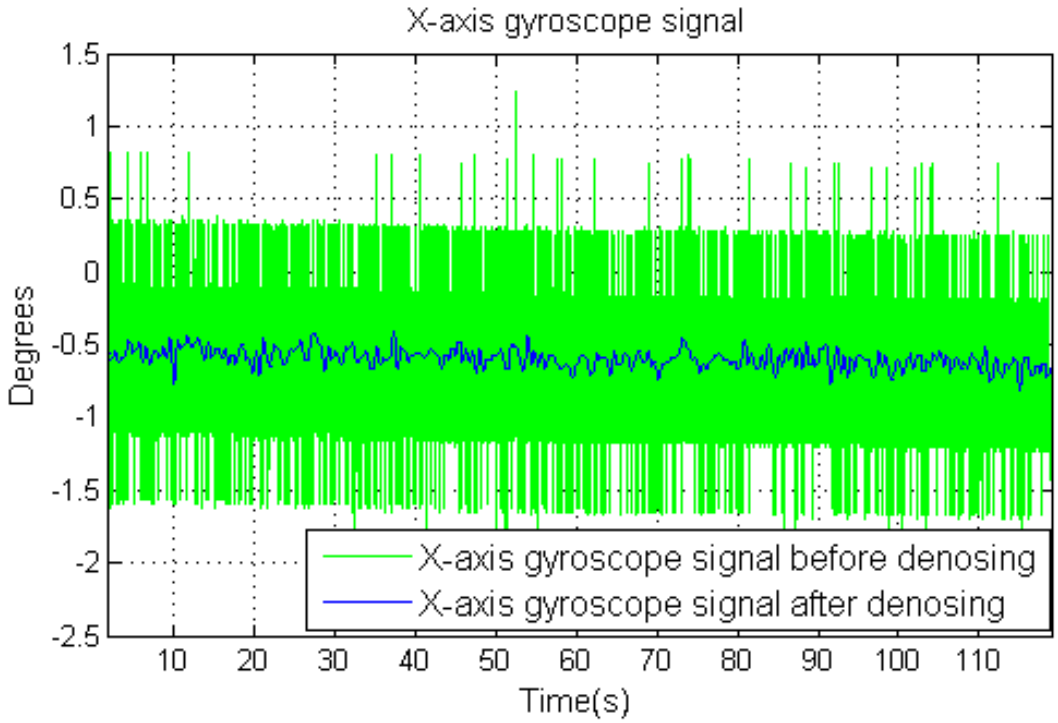


Figure 3.1 Memsense X-axis gyroscope signal analysis using wavelet denoising method

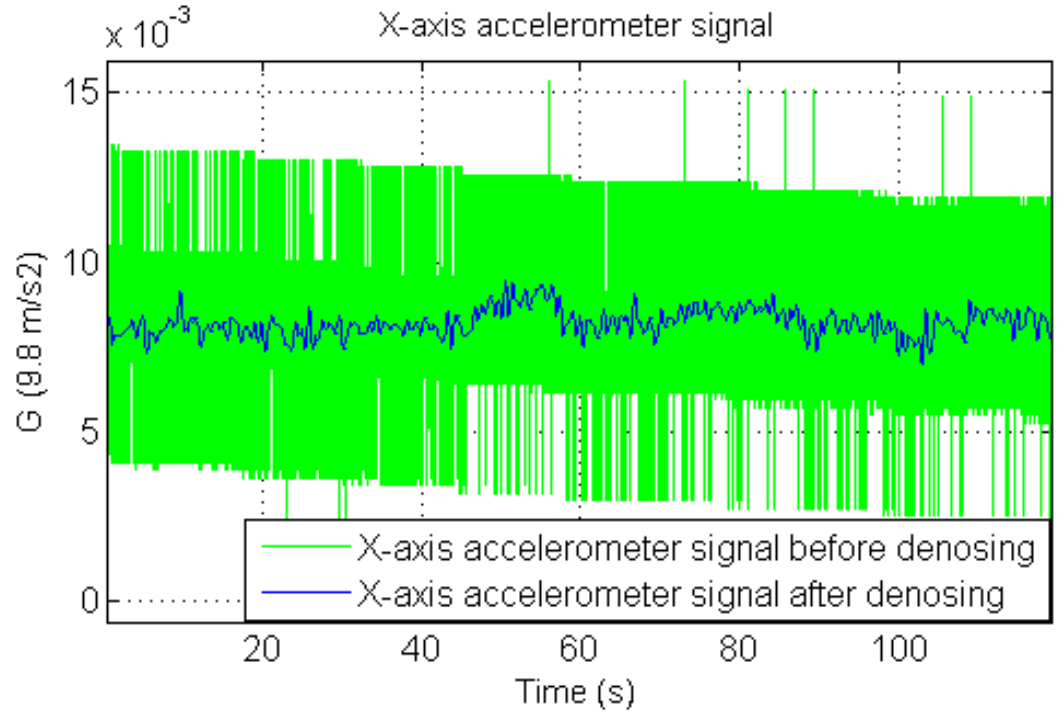


Figure 3.2 Memsense X-axis accelerometer signal using the wavelet denoising

3.1.2 Output data autocorrelation

Stochastic noise is embedded in inertial sensors measurements, and it is more prominent in gyroscopes. A first order GM model [18] with a decaying exponential autocorrelation sequence was used to model the error from the inertial sensors. Using a stationary run of more than 2 hours, the normalized autocorrelation sequences of two Memsense wireless IMU sensors were calculated as shown in Fig.3.3 and Fig.3.4.

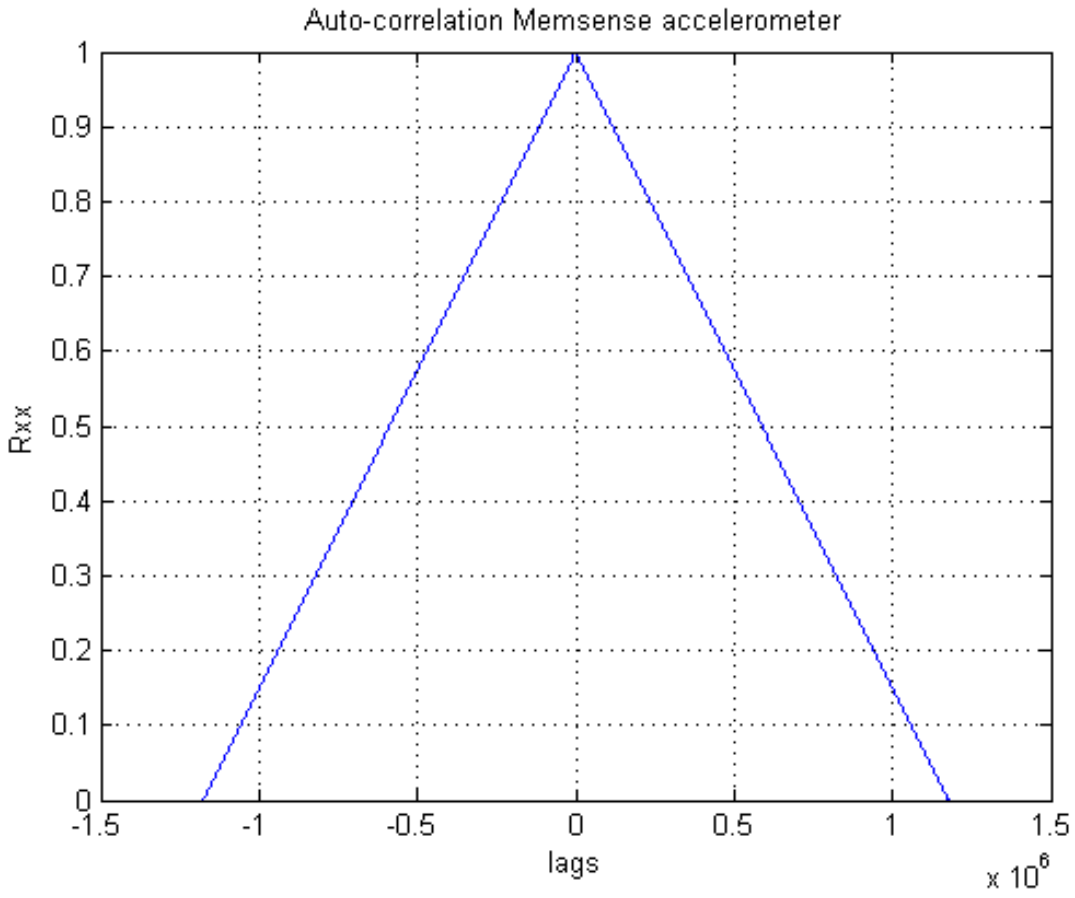


Figure 3.3 Autocorrelation sequence of a Memsense wireless IMU accelerometer

It was shown that the random errors associated with these inertial sensors were similar to a first order GM process, which was used in the KF implementation. The autocorrelation sequence of the gyroscope had a sharper peak at the center and was noisier, which was slightly

different from the ordinary GM model. However, the latter was considered adequate to use in further KF computations.

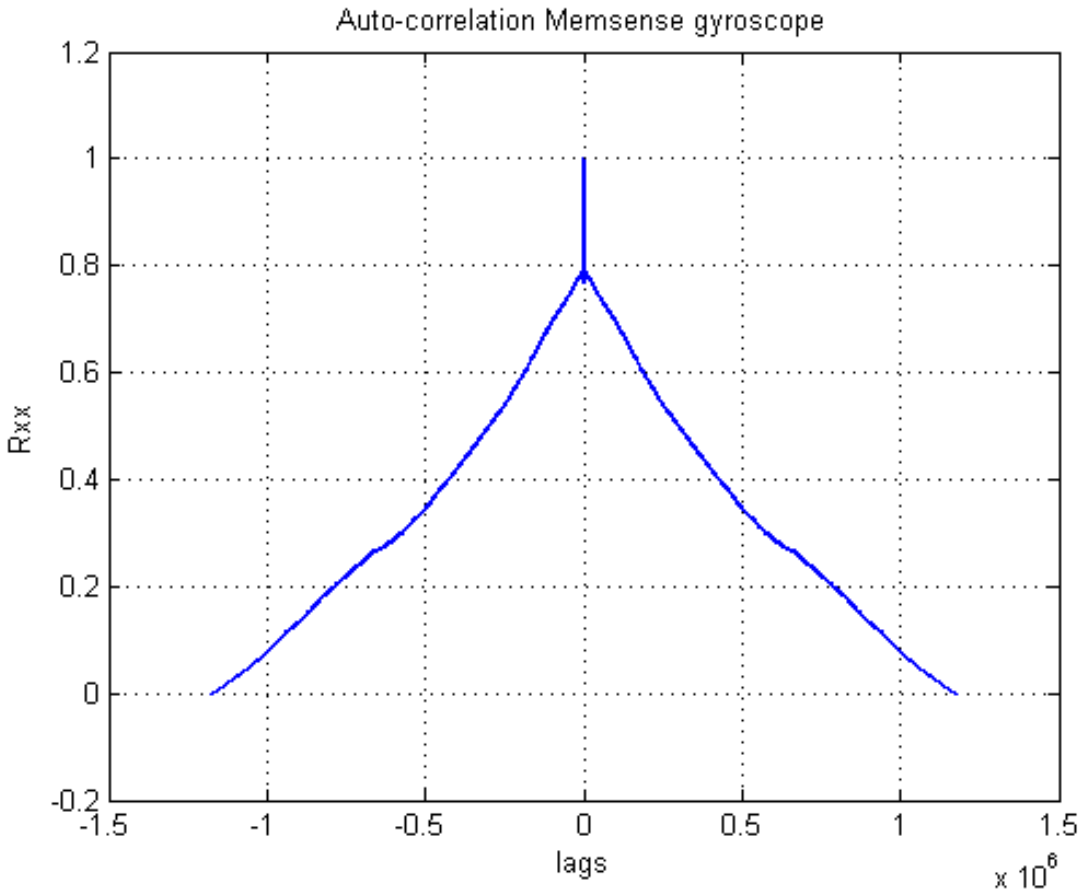


Figure 3.4 Autocorrelation sequences of a Memsense wireless IMU gyroscope

3.1.3 Calibration

Any inertial sensor needs to be calibrated before it can be used for navigation even though it has been calibrated during manufacture. As described in Chapter 2, the calibration procedures for the accelerometers are based on taking stationary measurements on a specific axis for a certain period of time (in this case, 10 minutes) in different orientations. From this the deterministic errors of the accelerometers including the bias and scale factor can be obtained.

Table 3.1 Average measurement values of the three accelerometers

Accelerometers	Orientation	Mean (g)	Standard Deviation (m/s^2)
X	Upward	-0.9834	0.0015
	Downward	1.0153	0.0014
Y	Upward	-0.9922	0.0015
	Downward	1.0029	0.0015
Z	Upward	-0.9854	0.0012
	Downward	1.0123	0.0014

Table 3.2 Accelerometers bias and scale factors

Accelerometer	Bias (m/s^2)	Scale factor (%)
X	0.0159	-0.8981
Y	0.0053	-0.8983
Z	0.0136	-0.8982

Since the MEMS-based gyroscopes are not capable of measuring the earth ration as a reference the applied calibration procedures consist of rotating the IMU on the stepper motor based R-IDA system at a certain rotational velocity for a period of time in different orientations. Due to the fact that MEMS-based gyroscopes are more prone to drift with time comparing to the accelerometers, the period of time for calibration was reduced to 4 minutes.

Table 3.3 Average measurement values of the three gyroscopes

Gyroscope	Orientation	Mean (°/s)	Standard Deviation
X	CCW	-562.8767	1.1889
	CW	562.1342	1.2743
Y	CCW	-563.4325	0.9817
	CW	563.1882	0.9847
Z	CCW	-561.7135	1.2642
	CW	561.6991	1.6047

Table 3.4 Gyroscope bias and scale factors

Gyroscope	Bias (°/s)	Scale factor (%)
X	-0.3702	-1.0704
Y	-0.1213	-1.0720
Z	-0.0070	-1.0689

In order to calibrate the embedded magnetometers in the IMU using the methods mentioned in Chapter 2, the IMU was rotated 360 degrees in the horizontal north-east plane several times on the R-IDA setup, the scale factors were determined from the ratio of the major and minor axes, in order to correct the circle to an ellipse. And the biases were calculated as the offset from center of the ellipse. Using the now calibrated magnetometer measurements the horizontal components and the local heading can be estimated. The azimuth angle can be calculated in conjunction with the declination from the WMM software.

The results are shown in Fig. 3.5. Here the magnetic field strength of the Earth is defined in the arbitrary units (a.u.), normalized to the Earth's field strength.

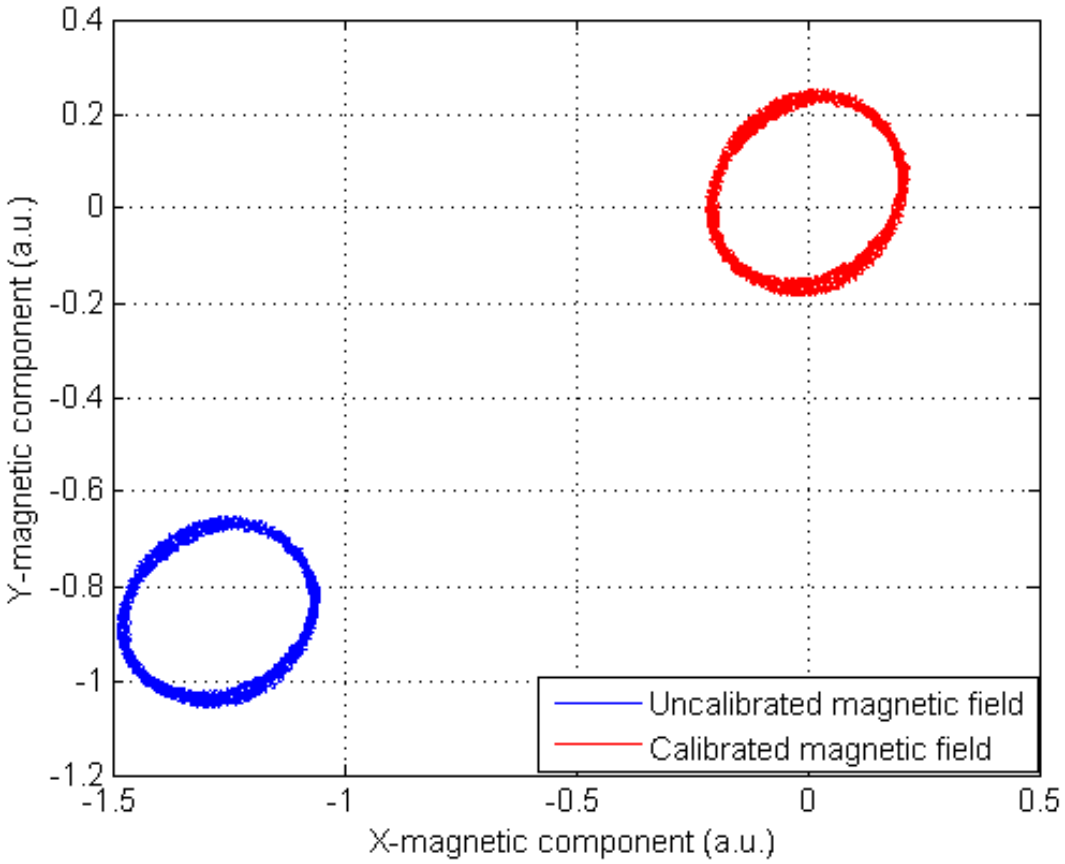


Figure 3.5 Performance of implemented auto-calibration procedure

Table 3.5 Magnetometers bias and scale factors

Magnetometers	Bias (a.u.)	Scale factor (%)
X	1.3348	1.0570
Y	0.9538	1.0000

3.1.4 Initial alignment

After applying the wavelet denoising technique and removing the deterministic errors of the IMU sensors by calibration, 2 minutes of static alignment was applied to the IMU (Fig. 3.6), in order to provide the initial attitude angles before navigation performs.

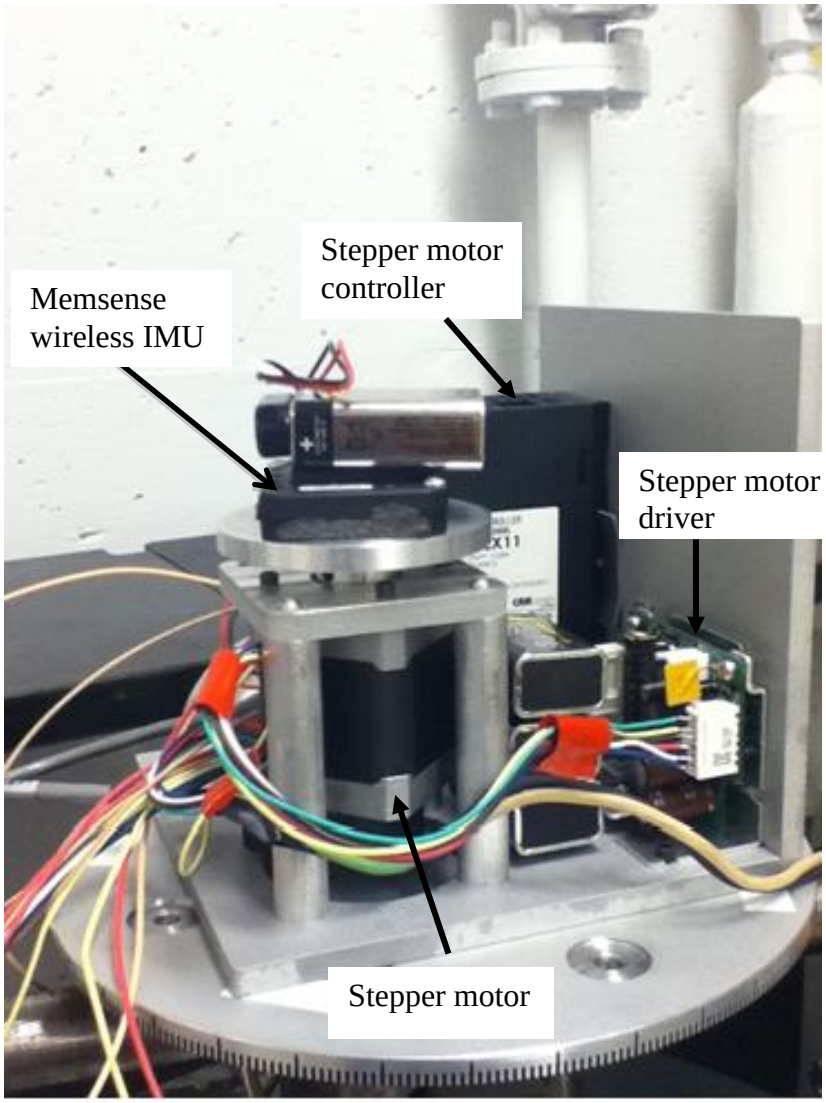


Figure 3.6 Stepper motor-based R-IDA setup during the coarse alignment

The discussed equations in Chapter 2 can be used to calculate the pitch and roll, which should be approximately of zero degrees since the IMU was placed on the levelled plane by the HD simulation device and levelled by water calibration. The azimuth angle was calculated using the measurements from the magnetometers, in conjunction with the known magnetic declination of 14.7666° E at the initial testing locations (Low Frequency Laboratory, University of Calgary). The azimuth angles were set at 70.0° ±1.0° by adjusting and rotating the stepper motors of the

horizontal drilling simulation. The initial azimuth uncertainty depended on the noise characteristics of the magnetometers and the accuracy of the WMM.

3.1.5 Azimuth error reduction in comparison with ZUPT

The initial angles were determined from the coarse alignment process, for each angle the stepper motor controller was used to accurately position the wireless IMU to the same azimuth angle, and 30 runs were conducted. Within each experimental run, a measurement outage was extended from 0 seconds to 15 and 30 seconds. During the horizontal drilling process, there are minutes during rotary drilling mode where the drill bit and the drill pipe are rotated at certain speed and reference measurement outages occur while the drill bit is active [60]. These outages provide a comparison between the R-IDA and the ZUPT methods by analyzing their response to an outage.

Kalman filtering is very sensitive to input values and cannot function properly if the model is not accurate enough. Although the inertial sensors models have been examined to fit the first order GM model, most of the system are non-linear and had to be linearized for the implementation of the KF, which resulted in further approximations on the system model. This deterioration of the KF performance was mitigated by artificially introducing noise through increasing process noise covariance and decreasing the measurement noise covariance. During the implementation, the Q and R were adjusted based on prior experience and tuned in order to optimize the performance of the KF [18]. Linear Kalman filtering works as an optimal predictor when the outage of the reference measurement is added. During the optimal prediction stage, the estimated error of the azimuth increased, since the low cost Memsense IMU has a very high noise level when compared to a tactical grade IMU.

When the rotation of the wireless IMU starts, the wireless IMU is at the center of the round table, and it only rotates at the center but it does not move. The velocity in the north and east were used to form the measurement equations since it is associated directly with horizontal plane. The rotation rate was set at ω = 281.25 °/s to avoid the severe vibration of the stepper motor system at lower rate for 2 minutes. In the R-IDA method, the rotation of the IMU modulates the inertial sensors errors as a cosinoidal signal which limits the divergence of the calculated errors as discussed in Chapter 2. According to the equations 2.3 and 2.4, the integration of these periodic signals in the local level frame for a complete rotation cycle is zero as shown below.

$$\int_0^T \delta\boldsymbol{\omega}^l = \begin{bmatrix} 0 \\ 0 \\ T\delta\omega_z^l \end{bmatrix} \tag{3.1}$$

$$\int_0^T \delta\mathbf{f}^l = \begin{bmatrix} 0 \\ 0 \\ T\delta f_z^l \end{bmatrix} \tag{3.2}$$

where T is the time for one complete rotation cycle.

After rotating the IMU with the round table of the stepper motor, only inertial sensors biases and drifts are considered in this study. The gyroscope drift can be modulated as [58]:

$$\delta\dot{\omega}^u = \left(\delta\omega^u cos\omega t\right)' = -\omega\delta\omega^u sin\omega t \tag{3.3}$$

The system equation is given as:

$$\dot{\boldsymbol{X}} = F\boldsymbol{X} + G\boldsymbol{w} \tag{3.4}$$

where F is the dynamic matrix, $\boldsymbol{w}$ is the process noise, $\mathbf{X}$ is the state vector consists of twelve components:

$$\boldsymbol{X} = \left[\delta v_e, \delta v_n, \delta v_u, \delta p, \delta r, \delta A, \delta\omega_x, \delta\omega_y, \delta\omega_z, \delta f_x, \delta f_y, \delta f_z\right]^T \tag{3.5}$$

where $\delta v_e, \delta v_n, \delta v_u$ are velocity errors, $\delta p, \delta r, \delta A$ are attitude errors, $\delta\omega_x, \delta\omega_y, \delta\omega_z$ are gyroscope drifts, and $\delta f_x, \delta f_y, \delta f_z$ are accelerometer biases.

The expended system model is given below, showing all the state vectors and how they coupled with the dynamic matrix F:

$$\begin{bmatrix} \delta\dot{v}_e \\ \delta\dot{v}_n \\ \delta\dot{v}_u \\ \delta\dot{p} \\ \delta\dot{A} \\ \delta\dot{r} \\ \delta\dot{\omega}_x \\ \delta\dot{\omega}_y \\ \delta\dot{\omega}_z \\ \delta\dot{f}_x \\ \delta\dot{f}_y \\ \delta\dot{f}_z \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & -f_u & -f_n & 0 & 0 & 0 & R_{11} & R_{12} & R_{13} \\ 0 & 0 & 0 & -f_u & 0 & -f_e & 0 & 0 & 0 & R_{21} & R_{22} & R_{23} \\ 0 & 0 & 0 & f_u & -f_e & 0 & 0 & 0 & 0 & R_{31} & R_{32} & R_{33} \\ 0 & \frac{1}{R_M + h} & 0 & 0 & 0 & 0 & R_{11} & R_{12} & R_{13} & 0 & 0 & 0 \\ -\frac{1}{R_N + h} & 0 & 0 & 0 & 0 & 0 & R_{21} & R_{22} & R_{23} & 0 & 0 & 0 \\ -\frac{\tan\varphi}{R_N + h} & 0 & 0 & 0 & 0 & 0 & R_{31} & R_{32} & R_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\omega\sin(\omega t) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\beta_{fz} \end{bmatrix} + \begin{bmatrix} \delta v_e \\ \delta v_n \\ \delta v_u \\ \delta p \\ \delta r \\ \delta A \\ \delta\omega_x \\ \delta\omega_y \\ \delta\omega_z \\ \delta f_x \\ \delta f_y \\ \delta f_z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \sqrt{2\beta_{\omega x}\sigma_{\omega x}^2} \\ \sqrt{2\beta_{\omega y}\sigma_{\omega y}^2} \\ \sqrt{2\beta_{\omega z}\sigma_{\omega z}^2} \\ \sqrt{2\beta_{fx}\sigma_{fx}^2} \\ \sqrt{2\beta_{fy}\sigma_{fy}^2} \\ \sqrt{2\beta_{fz}\sigma_{fz}^2} \end{bmatrix} \mathbf{w} \tag{3.6}$$

Where R_M is the meridian radius of curvature and is the radius of the ellipse; R_N is the normal radium which is the radius of curvature of the prime vertical; h is the height, R_{11} to R_{33} are components of the transformation matrix R_b^l.

The measurement equation utilized by the KF is given by:

$$\mathbf{Z} = \begin{bmatrix} \delta v_n \\ \delta v_e \\ \delta v_u \end{bmatrix} = \begin{bmatrix} v_n^{ref} - v_n^{IMU} \\ v_e^{ref} - v_e^{IMU} \\ v_u^{ref} - v_u^{IMU} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{X} + \mathbf{v} \tag{3.7}$$

where Z is the velocity errors, KF will estimate the error states **X** recursively in the closed-loop system.

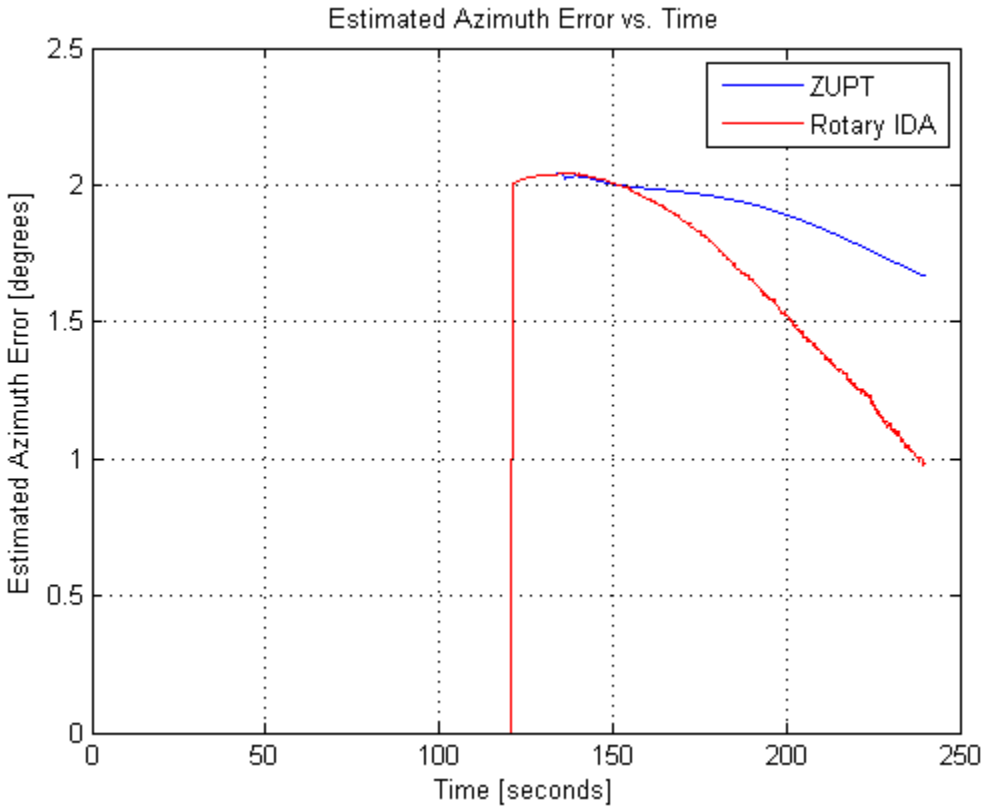


Figure 3.7 Estimated azimuth errors (70° azimuth angle, Trial 1) from ZUPT and R-IDA methods. The measurement outage occurred from t = 120 seconds to t = 135 seconds (15 seconds duration)

The R-IDA method reduced the azimuth errors from 2° to a mean value of 0.9802°, while the ZUPT methods decreased the azimuth errors to a mean value of 1.6720° in the same time.

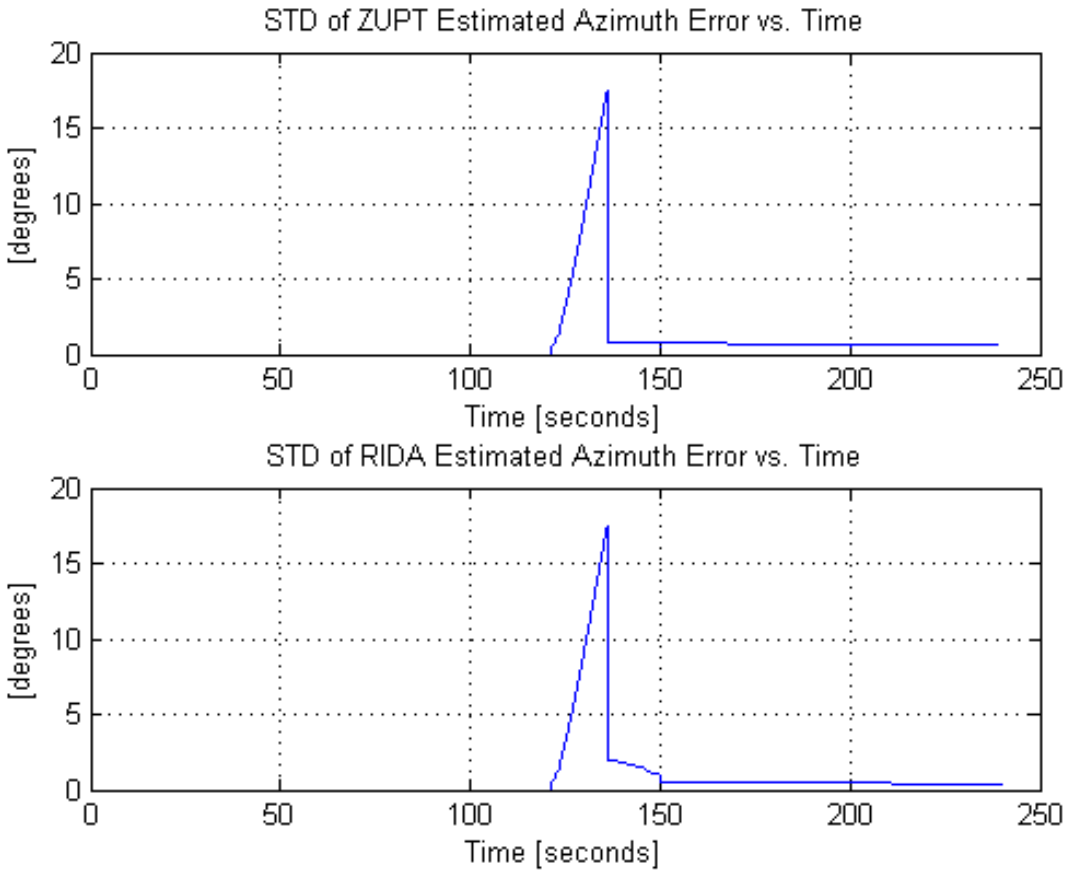


Figure 3.8 Comparison of STD of the ZUPT and R-IDA

This comparison is favorable to ZUPT as the standard deviation (STD) converged faster than the R-IDA on Fig. 3.8, because the R-IDA method is susceptible to vibrations of the stepper motor system during the dynamic alignment which exposes the gyroscopes to erroneous measurements. Nevertheless, the standard deviation of the azimuth error for the R-IDA also converged to a lower value compared to the ZUPT method (Fig. 3.8). This result was consistently repeatable in all the experiments conducted.

3.2 Wireless MEMS-based IMU module

3.2.1 Wireless navigation module testing on a breadboard

The proposed wireless navigation module design has been implemented and tested on a solderless breadboard as shown in Fig. 3.9, from left to right: MBED LPC11u24 MCU, Analog Devices ADIS16488A IMU, and XBP RF transmitter. They have been wired up according to the schematic diagram as described in Chapter 2, and powered by a high performance downhole battery.

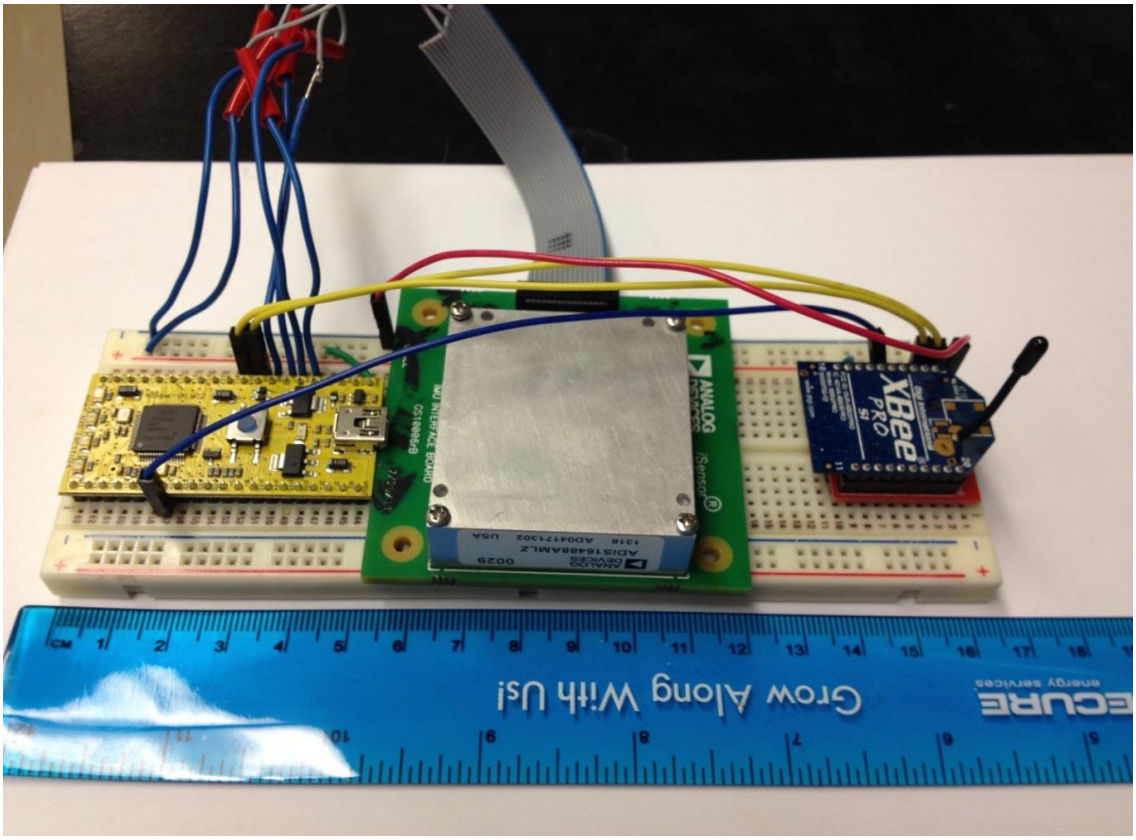


Figure 3.9 Wireless navigation module prototype

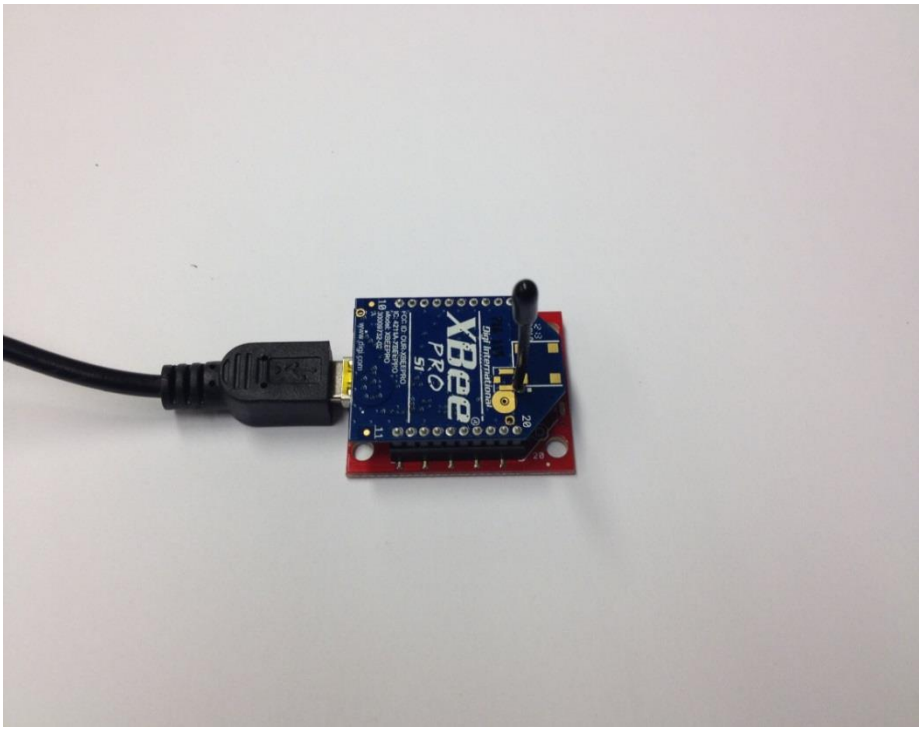


Figure 3.10 XBP RF receiver with XBee Explorer USB adapter

3.2.2 Laboratory testing

The above mentioned wireless navigation module was used for laboratory transmission testing. 2 AA commercial batteries in the battery holder powered the ADIS16488A and the XBP, while the LPC11U24 was wired to a 9 V battery for demonstration purposes.

The wireless navigation module was carried by a person, and the receiver on the XBee explorer USB adapter was connected to a PC in ENA 003, Schulich School of Engineering, University of Calgary. The person walked into room ENA 005 and then left the room and headed north to the end of the corridor in the Engineering building block A. During this process, there were several concrete walls between the module and receiver. And the testing displacement of the wireless IMU module was up to 150 meters. The XBP connected with the PC (Fig. 3.10) continuously and received inertial sensors measurements at the pre-set sample rate of 5 Hz, which could be

considered adequate for the very slow drilling processes, and displayed and logged the accelerometer and gyroscope measurements through the terminal of the Tera Term open access software.

3.2.3 Preliminary field testing

Since 2013 PetroJet Canada Inc. (Calgary, Alberta, Canada) has been working with the University of Calgary on a downhole positioning device to be used to orientate their whipstock product. The positioning device will be located on the bottom of the whipstock close to the BHA, for which a critical issue is how to determine its tool face (roll) before the multilateral goes out, and bring the positional information from the bottom of the whipstock to the surface.

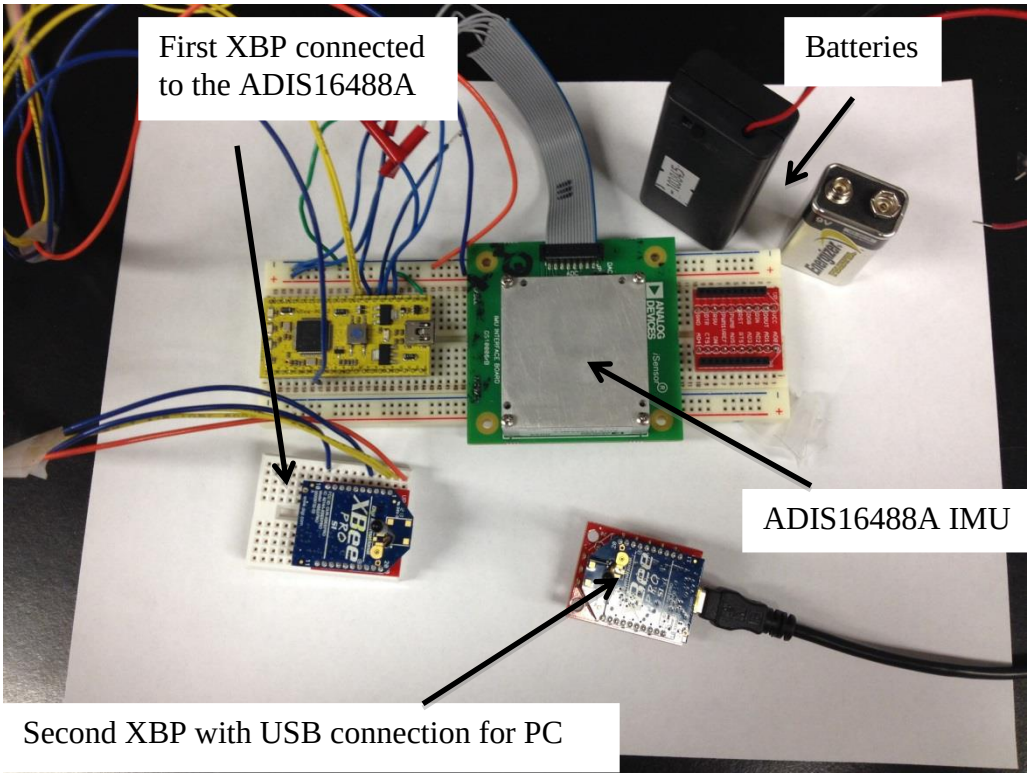


Figure 3.11 Wireless navigation module for a field testing

On May 8, 2014 a preliminary field test was successfully performed to determine the ability to use ZigBee-based wireless communication within a length of pipe and through

PetroJet's whipstock assembly. In order to do the test the following equipment (Fig. 3.11) was brought to PetroJet's testing facility located in Crossfield, Alberta.

- Wireless navigation module, of which the XBP has been separated onto a small solderless breadboard to fit into drill pipe;
- Second wireless XBP receiver connected to a computer.

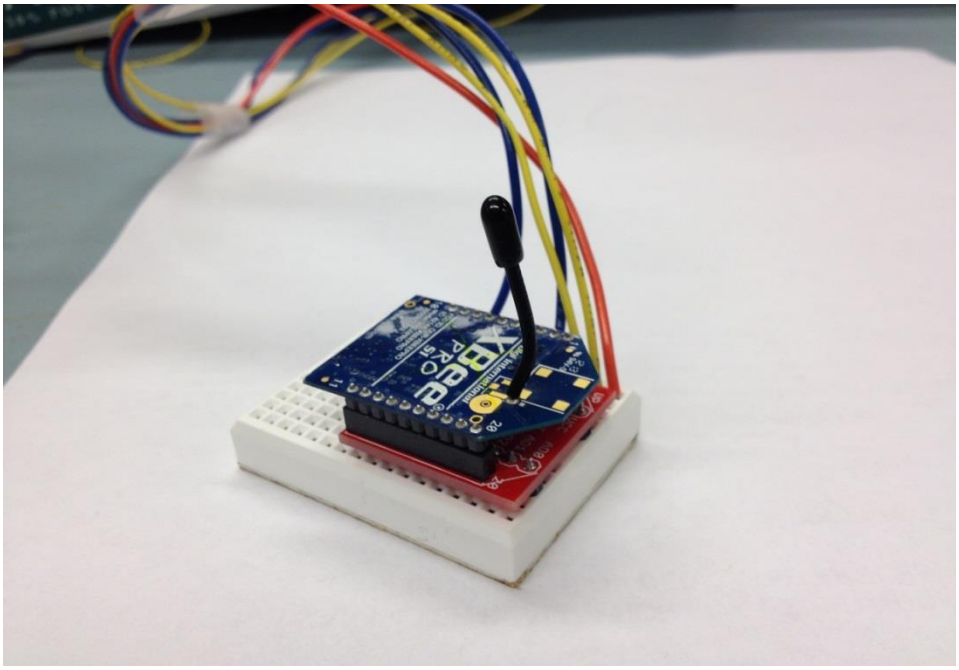


Figure 3.12 Zoom-in of the XBP transmitter put within the drill pipe

Test One consisted of placing the IMU XBP as a transmitter into one end of a length of 3000 psi 7.7 cm line pipe laid horizontally on the ground and the second XBP as a receiver into the other end of the same pipe and determining if the signal could be sent between the two devices. Pipe lengths were increased in 6 meters increments. Wireless XBP communications were achieved in all lengths of 6, 12, 18, 24 and 30 meters.

Figure 3.13 The test apparatus for Field Test One

Test Two consisted of placing the IMU wireless XBP as transmitter into the threaded end of the bottom of the PetroJet whipstock located in PetroJet's test tank and transmitting IMU positional information to the second wireless XBP located within the end of a piece of 7.7 cm, 15000 psi frac iron approximately 9 meters from the bottom of the whipstock. Successful wireless ZigBee-based communication was achieved.

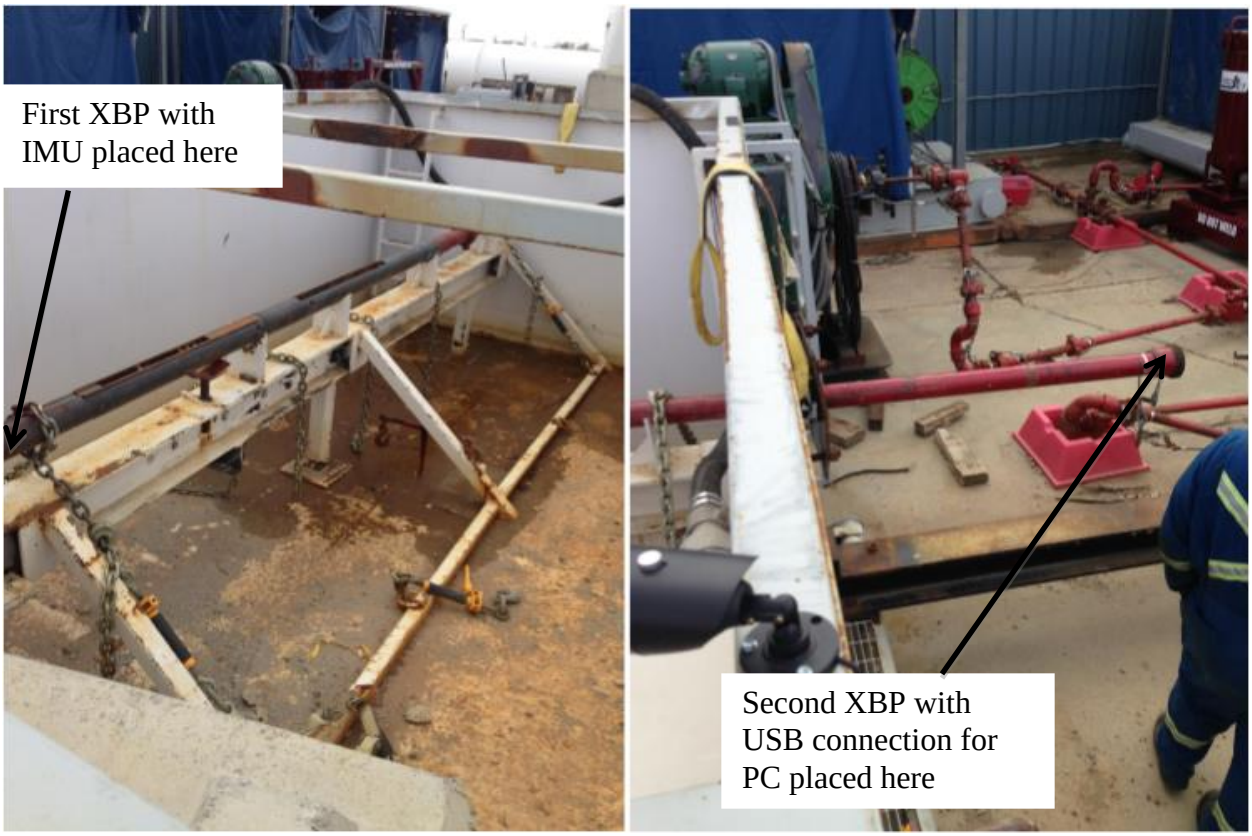


Figure 3.14 The test apparatus for Field Test Two

Two wireless XBP transmitter/receivers were used to transmit positional information from a MEMS IMU. Wireless communication was successfully sent through 9 meters lengths of 3000 psi, 7.7 cm line pipe up to a total length of 30 meters. Also a wireless signal was successfully sent from the bottom of the PetroJet whipstock to the outlet of a 15000 psi frac iron over a distance of approximately 9 meters.

This was the initial test to determine the feasibility of wireless communications downhole, further field testing is required to determine the suitability of downhole wireless communication.

3.2.4 PCB prototype for multilateral drilling

The PCB prototype includes the design of micro SD card (Fig. 3.15), which would store the IMU measurements directly through an SPI interface. By adding this feature, the integrity of the data transmission will be determined by comparison and more useful data such as pressure and temperature data downhole can be stored into micro SD card for a post drilling analysis.

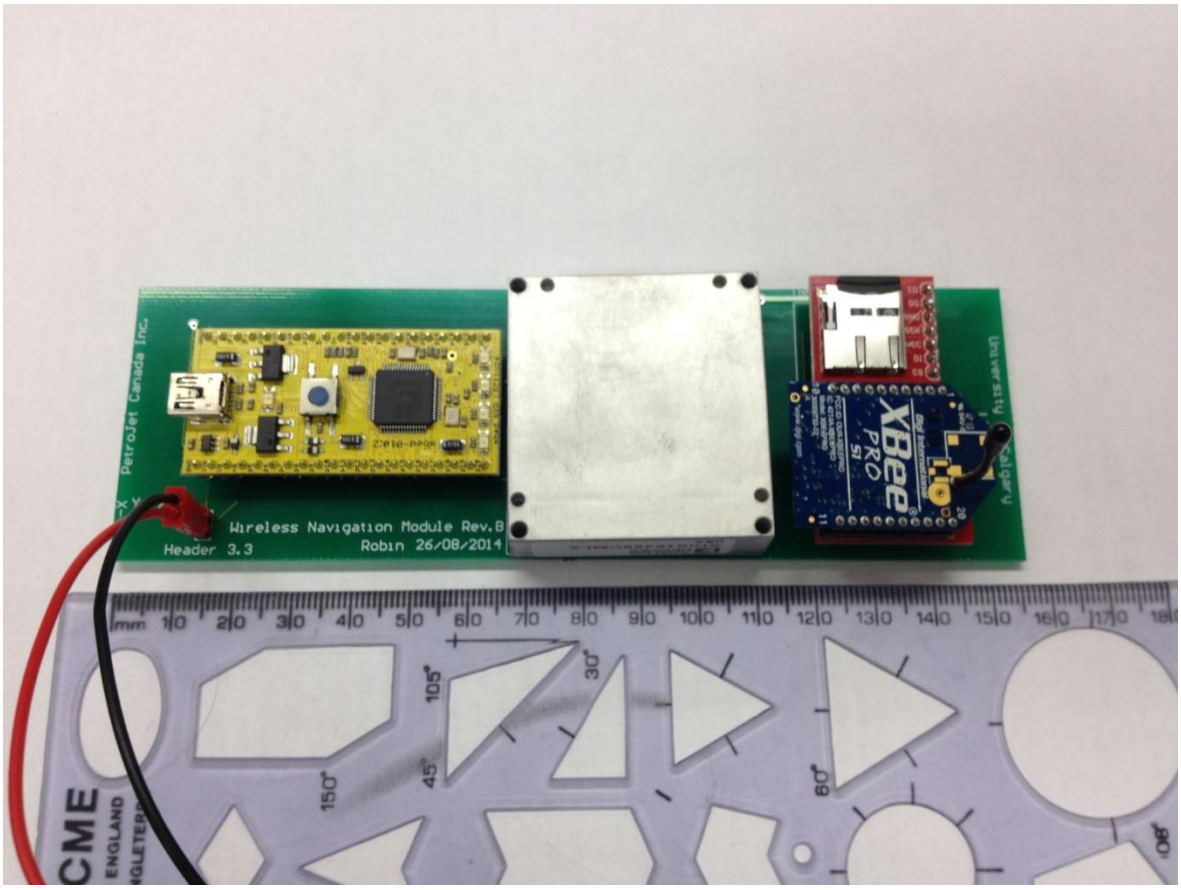


Figure 3.15 The PCB prototype of wireless navigation module

As shown in Fig. 3.15, a standard header was placed on the corner of the board for downhole battery connection. LPC11U24 was put at the edge of the board for convenient USB connection and the wireless XBP transmitter was placed on the other edge due to the fact that RF antenna is less effected by the electronic components at the corner of the board. It is quite small and compact of 15 cm long and 5 cm width.

Chapter Four: **Discussion and Conclusion**

4.1 Discussion

In the quest to replace the magnetometers in horizontal drilling processes, particular attention has been dedicated to strapdown INS that incorporate inertial sensors containing three-axis accelerometers and gyroscopes. This focus is because GPS signals are not applicable downhole, and other nonmagnetic methods for navigation have not produced any promising results. Replacing the magnetic-surveying system with FOG-based inertial sensors in horizontal drilling has been proposed by using a dual-FOG system combined with three orthogonal accelerometers in 2002 [14]. More work has been focused on the tactical grade FOGs-based INS. Although the FOGs-based IMU such as LN-200 (Northrop Grumman, USA) has excellent navigation performance, the dimension of 8.89 cm in diameter and 13.21 cm in height is too large for the smaller size of contemporary horizontal drill pipes. The In-Drilling Alignment (IDA) apparatus [28] in the Low Frequency Laboratory utilized a pneumatic piston based system that was not mountable downhole on account of its larger dimension and high power consumption. These two critical parameter restrictions make the implementation of previous proposed IDA apparatus downhole not feasible.

The rapid development of Micro-Electromechanical System technology makes MEMS-based inertial sensors the emerging edge for the downhole drilling technology. In fact, MEMS-based INS by itself cannot deliver desirable results either, due to three important limiting factors: inadequate temperature range, high susceptibility of the inertial sensors to downhole vibration, and more importantly high cumulative errors, particularly in assessing the azimuth. On the other hand, magnetic-based navigation is becoming more and more obsolete, particularly with the depletion of the easy-to-access downhole resources and the need to search for new ones in

remote or difficult to reach locations that require higher accuracy. The appropriate alignment methods for the INS have been developed for many years and the KF has become the benchmark for fine alignment implementation. Although the ZUPT offers high simplicity and acceptable results for some applications, the lack of the updated underground observations greatly handicaps the overall navigation performance, especially for estimation of azimuth error. The R-IDA methods supplement simplicity, reliability and high performance, in conjunction with tactical grade MEMS-based IMU, which has low weight, cost, and power consumption, while its small size makes the replacement of magnetometer-based MWD systems possible.

In the first study, I presented clear evidence theoretically and experimentally that proposed that the R-IDA technique is superior to the ZUPT in minimizing dynamic positioning errors. Moreover, its implementation is inexpensive, miniature and quite straightforward. In the second study, I improved the performance of the MEMS-based IMU by adding wireless data transmission function to the ADIS16488A, which gives it more flexibility for dynamic alignment downhole. By adding the wireless transmission function, the novel wireless network concept has been applied for the entire drill system by using a low power, reliable and relatively high transmission rate RF module. Another beauty of such a system is that it allows the entire drilling string to act as a local network within the BHA, and other downhole tools located somewhere inside the drill string can be individually addressed and/or turned on and off.

Additional effort is needed to fully implement the dynamic rotational IDA method downhole. Moreover, the R-IDA and IMU transmission experiments that were conducted provided not only positive results, but also information could be used to improve the design of the R-IDA apparatus and downhole data transmission.

4.2 Future work

4.2.1 R-IDA setup

4.2.1.1 Hardware improvement of the R-IDA setup

First and foremost, the Memsense IMU utilized in the initial laboratory and feasibility study should be replaced by a higher-performance unit, e.g. by the recently developed MEMS ADIS16488A by Analog Devices, which was utilized in the wireless IMU project. The latest STIM 300 IMU with a gyro bias instability of 0.5°/h produced by Sensonor (Horten, Norway) has achieved even better performance than some tactical grade FOG-based IMUs. All these sensors modules are equipped with standard digital protocol such as serial or SPI interface, which simplifies the wireless interface design.

Second, the vibration of the stepper motor system was significantly adverse, particularly when the rotation rate was below 200 °/s, which made implementation for R-IDA inapplicable since inertial sensors measurements deteriorated dramatically with noise. The vibration of the stepper motor itself affected the general performance of the R-IDA as the convergence time was extended and some KF could not converge at certain rotational velocities. In addition, a great amount of harmful heat would be generated by the motor in a short time during the operation, which limited a long period operation of the setup. The higher resolution, lower heat generation downhole-mountable brush/stepper motor is needed to replace the current stepper motor. It should be mechanically packaged and cased to reduce severe internal and external vibrations.

Thirdly, the round table needs to be light-weight, preferably made by high-temperature resistance plastic, to avoid inertial effects of different rotation schemes and to increase the control precision. In addition, the round table of the stepper motor system should be more precisely manufactured to fix the origin s-frame of IMU at the origin of its b-frame. In order to

achieve this, the IMU needs to be accurately calibrated and its structural information from the manufacturer regarding its s-frame should be determined. It has to be adjusted to align the center of the table with the IMU's origin, in order to minimize undesirable measurements from other inertial sensors. This misalignment is similar with non-orthogonal off axis errors that will deteriorate the performance of mechanization and KF.

Fourthly, only single axis rotation was implemented in the R-IDA system due to its simplicity, low cost and high reliability. However, a dual-axis rotary system will offer improved navigation performance when compared with the single axis system, since it modulates more inertial sensors errors into a periodic signal. One more axis of rotation was suggested along the y axis, which is in the same direction of the horizontal drilling process. This requires the design of a sophisticated rotation platform consisting of two sets of precisely controlled motors.

Fifthly, due to the limitation of the stepper motor in our experiments, continuous rotation at low rate was not applicable. Different rotation rates especially at lower rates should be tested. Moreover, induced rotation at different planes is also recommended. The different rotation schemes of the R-IDA system should be explored extensively. One special scheme called reciprocating rotation has a complete reciprocating rotation cycle including a rotation cycle in the CCW direction and then a rotation cycle in the CW direction.

Sixthly, the possibility exists to include a rotary actuator for fully implementing the IDA methods. The classic conversion from the rotation to linear motion has been widely used in the industry simply by several rods, and the previously proposed pneumatic piston-based IDA system can be simplified and minimized through this device, which makes it highly mountable downhole.

Seventhly, temperature gradients produce a significant drift and change in the MEMS inertial sensors, the amplitude of which is related to the particular temperature changes themselves. In real drilling processes the drilling fluid circulates along the drilling trajectory which keeps the temperature at the surface of navigation module relatively stable, and therefore it alleviates the fluctuation of the temperature. Moreover, the temperature range is particularly limited by the MEMS inertial sensors. The presently offered range (-55 to 105℃) of ADIS16488A for some shadow downhole navigation and guidance applications is acceptable. The oil drilling industry will require an increased temperature range for most of the drilling applications in the world, thus the temperature limitation should be increased to 150℃. If this extended temperature range is achieved, utilizing the recently proposed approach suggested by [61], the oil drilling industry is more likely to employ this technology, and further enhance the development and improvement of the MEMS IMUs.

Eighthly, the circulation of drilling fluid not only relieves equipment from the temperature gradient, but also can provide extra energy downhole harnessed from the motion of the fluids. The proposed device on Fig. 4.1 is a windmill design to utilize the fluid flow for energy. It allows the rotation of the MEMS sensors for a long period while only using a small amount of power for the MEMS sensors and rotary encoder. The unit also needs to be appropriately packaged to withstand broad temperature and high levels of vibrations, and finally be securely attached to the drill pipe.

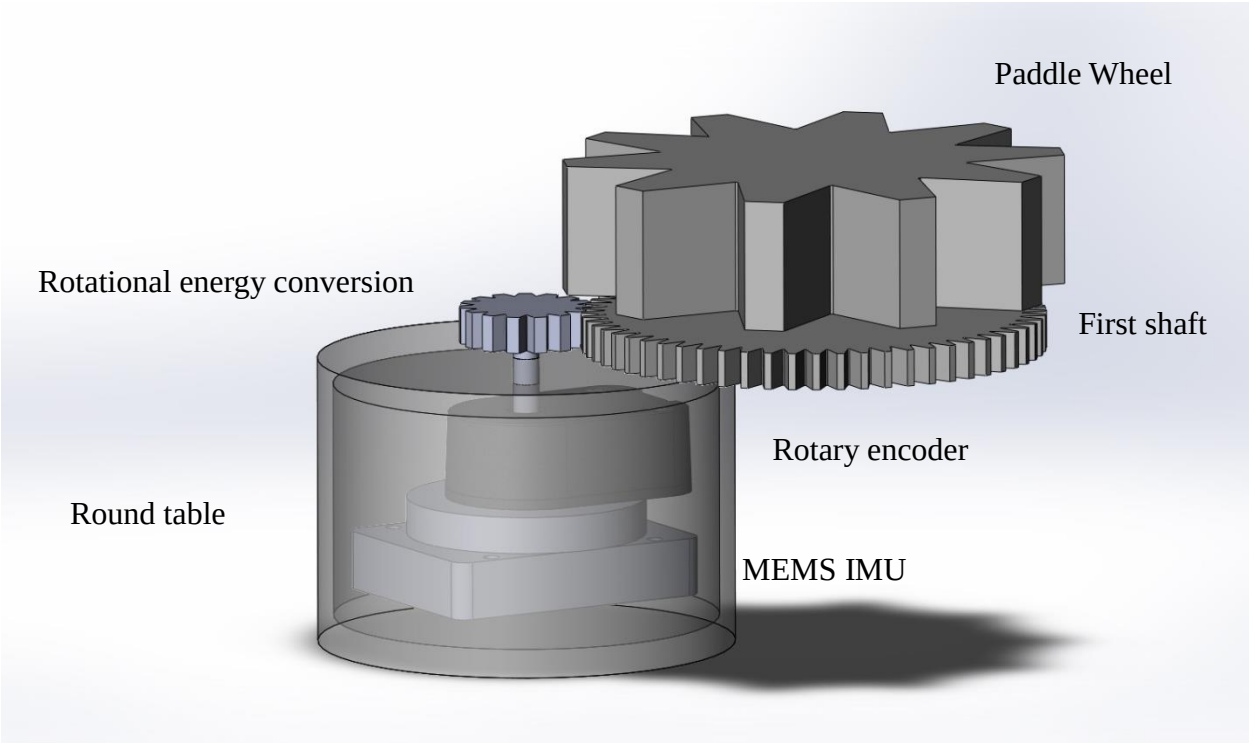


Figure 4.1 The proposed concept of windmill design

Ninthly, the goal of this study is to prove the feasibility of an R-IDA method, and the signals were processed offline. However, the algorithms are recursive in nature and it should be implemented in a real-time system for the drilling processes. The IMU's interface needs to be designed to intuitively present the real-time position and orientation. This can be achieved by programing a low-power consumption microprocessor either by the included LPC11U24 or implementing a more sophisticated software solution using the PC on the surface.

4.2.1.2 The Software improvement

The main engine for estimating the azimuth error is the linear Kalman filter, which makes a lot of assumptions and approximations. The linear system is an idealized system which, strictly speaking does not exist. However, because many systems such as tactical grade FOGs-based INS are very close to linear, these estimation approaches give satisfactory results. The Extended

Kalman filtering is the non-linear equivalent, which operates on the error states rather than the total states. The error states can be computed by observing the difference between the aiding source states and the INS states.

For the MEMS-based inertial sensors, the first order GM is not as good as tactical grade FOG-based IMUs as shown in the autocorrelation sequence. More accurate sensor noise models such as higher order AR modeling suggested by [47] are required for the implementation KF.

The embedded magnetometers in the MEMS-based IMU may be utilized for the implementation of a KF for the local heading observable update since the KF performance is strongly related to the aiding sources. Although the performance of the magnetometers changes with the environment, the calibration of magnetometers can be beneficial for the horizontal rotational alignment, from which the weakest coupled azimuth states can be updated.

4.2.2 Wireless MEMS-based IMU module

The ZigBee-based wireless transmission proved its feasibility to work downhole due to its various advantages. However more work is required in order to make it downhole applicable. A reliable high performance downhole battery should be selected and incorporated into the entire package with the PCB wireless module. The packaging should be able to resist the harsh downhole environment. The PCB prototype will be mounted close to the BHA and real drilling processes for several field tests, and would be required to check the data integrity and transmission reliability. The interface of the ZigBee protocol with a conventional MWD system as an alternative first step is also needed to implement a navigation algorithm in order to make it commercially feasible.

4.3 Conclusion

This thesis presents clear evidence that the proposed R-IDA technique is superior to ZUPT in minimizing dynamic positioning errors. Moreover, its implementation is inexpensive, miniature and quite straightforward. The objectives to further improve and implement the IDA-based method and make alignment apparatus downhole mountable have been achieved. R-IDA is a viable tool to minimize cumulative navigation errors in MEMS-based IMUs, which makes the replacement of magnetic surveying in horizontal drilling with modern INS possible.

The proposed wireless data transmission for a downhole IMU-based R-IDA setup uses precisely controlled rotation during R-IDA error compensation procedures, since rotating an IMU with wired cables would be impractical and would result in constant forces and torques applied to the wires, leading to uncontrolled noise induction. Preliminary field testing within the drill pipes showed a successful wireless transmission, suggesting that the method could be potentially applicable to for real drilling applications.

The approach outlined in the present thesis makes the practical implementation of the IDA method downhole feasible, which can further lead to the replacement of traditional magnetometer or compass-based navigation with modern inertial navigation systems.

4.4 Scholarly contributions related to the thesis

This work has already resulted in two journal papers in international journals, and a patent application that is about to submit to Innovate Calgary, the intellectual property company of the University of Calgary. Further information on these achievements is described below:

1) **Z. H. Wang**, M. Poscente, D. Filip, M. Dimanchev and M. P. Mintchev, “Rotary in-drilling alignment using an autonomous MEMS-based inertial measurement unit for measurement-while-

drilling processes," *IEEE Magazine of Instrumentation and Measurement*, no. 6 pp. 26-34, Dec. 2013

2) **Z. H. Wang**, T. Li, M. McDougall, D. McCormack, and M. P. Mintchev, "Wireless data transmission options in rotary in-drilling alignment setups for multilateral oil drilling applications," *Journal of information theories and applications*, vol. 21, pp. 154-161, 2014

3) **Z. H. Wang, et al**. Zigbee-based wireless data transmission for measurement while drilling processes. Patent application, draft to be submitted to Innovate Calgary. Calgary, AB, Canada.

References

[1] Amerex Corporation. “Oil and Gas 101: Directional Drilling.” Amerex Corporation (2014). Internet: [Online]. Available: http://www.amerexco.com/DirectionalDrilling.html.

[2] J. Burkmann and N. Nickels, “Directional, navigational and horizontal drilling techniques,” *Geothermal Resour. Council Bull.*, vol. 19, no. 4, pp. 106–112, 1990.

[3] S. D. Joshi and W. Ding, “The cost benefits of horizontal drilling techniques,” in *Proc. Amer. Gas Assoc.*, 1991, pp. 679–684.

[4] S. D. Joshi, *Horizontal well technology*: PennWell Books, 1991.

[5] A. S. Djurkov, J. Cloutier, and M. P. Mintchev, “Mathematical model and simulation of a pneumatic apparatus for inertial navigation unit during horizontal well drilling,” *Int. J. Inf. Theories Appl.*, vol. 2, pp. 366–376, Jun. 2007.

[6] A. Pasicznyk. "Evolution toward simpler, less risky multilateral wells." In *Proc. SPE/IADC drilling conf.*, Mar. 2001.

[7] PetroJet Canada Inc. “Technology.” PetroJet Canada Inc. (2013). Internet: [Online]. Available: http://www.petrojet.ca/technology.php.

[8] S. Bosworth, H. S. El-Sayed, G. Ismail, H. Ohmer, M. Stracke, C. West, and A. Retnanto. "Key issues in multilateral technology." Oilfield Review, vol 10, no. 4, pp. 14-28, 1998.

[9] P. Ripka. *Magnetic Sensors and Magnetometers*. Norwood, MA, USA: Artech House, 2001, pp. 516.

[10] J. L Thorogood, and D.R. Knott. “Surveying techniques with a solid state magnetic multi-shot device.” *SPE Drill. Eng.*, vol. 5, pp. 209-214. Sep. 1990.

[11] B.A. Shelkholeslami, B.W. Schlottman, F.A. Siedel, and D.M. Button, “Drilling and production aspects of horizontal wells in the Austin Chalk,” *J. Petroleum Technol.,* vol. 43, no. 7, pp. 773–779, Jul. 1991.

[12] T. Torkildsen, I. Edvardsen, A. Fjogstad, A. Saasen, and T. Omland, “Drilling fluid affects MWD magnetic azimuth and wellbore position,” in *Proc. SPE/IADC Drilling Conf., 2004*, pp. 595–602, 2004.

[13] R. A. Estes and D. M. Epplin, “Development of a robust gyroscopic orientation tool for MWD operations,” in *SPE Annu. Tech. Conf. Exhib.,* 2000, 63274-MS.

[14] A. Noureldin, “New measurement-while-drilling surveying technique utilizing a set of fiber-optic rotation sensors,” Ph.D. dissertation, Univ. Calgary, Calgary, Canada, 2002.

[15] J. Maehs, D. MacAfee, S. Renne, G. Cellos, and A. Srinivasan, “Successful relief well drilling utilizing gyroscopic MWD (GMWD) for reentry into an existing cased hole,” in *SPE Annu. Tech. Conf. Exhib.,* 2008, 116274-MS.

[16] Schlumberger. “GyroPulse MWD.” Schlumberger (2013). Internet: [Online]. Available: www.slb.com/services/drilling/mwd_lwd/mwd/gyropulse.aspx.

[17] P. Aggarwal, Z. Syed, N. El-Sheimy. *MEMS-Based Integrated Navigation*. Norwood, MA, USA: Artech House, 2010, pp. 1-22.

[18] A. Noureldin. K.B. Tashfeen. and G. Jacques, *Fundamentals of Inertial Navigation, Satellite-based Positioning and their Integration*. New York, NY, USA: Springer, 2013.

[19] IEEE Std 1431-2004, “IEEE Standard Specification Format Guide and Test Procedure for Coriolis Vibratory Gyros,", pp.1,78, Dec. 2004.

[20] A. Noureldin. ENGO 623. Class Lecture, Topic: “Inertial Surveying and INS/GPS Integration.” Faculty of Engineering, Univ. Calgary, Calgary, AB., 2013.

[21] R. Grover and P. Hwang, "*Introduction to random signals and applied Kalman filtering*," Willey, New York, 1992.

[22] A. Gelb, *Applied optimal estimation.* Cambridge, MA, USA: MIT press, 1974.

[23] Petovello M.G.: " Real-Time Integration of Tactical-Grade IMU and GPS for High-Accuracy Positioning and Navigation," Ph.D. Dissertation, Univ. of Calgary, 2003.

[24] E.-H. Shin, "Estimation techniques for low-cost inertial navigation," *Ph.D. Dissertation, Univ. of Calgary, 2005.*

[25] C. Jekeli, *Inertial navigation systems with geodetic applications*. Berlin, Germany: Walter de Gruyter, 2001.

[26]] A. Noureldin, H. Tabler, D. Irvine-Halliday, and M. P. Mintchev, "Quantitative study of the applicability of fiber-optic gyroscopes for MWD borehole surveying," *SPE J.*, vol. 5, no. 4, pp. 363–370, Dec. 2000.

[27] E. Pecht and M. P. Mintchev, "Modeling of oberservability during in-drilling alignment for horizontal directional drilling,'' *IEEE Trans. Instrum. Meas.*, vol. 56, no. 6, pp. 1935-1945, Oct. 2007.

[28] A. S. Jurkov, J. Cloutier, E. Pecht, and M. P. Mintchev, "Experimental feasibility of the indrilling alignment method for inertial navigation in measurement-whiledrilling," *IEEE Trans. Instrum. Meas.*, vol. 60, no. 3, pp. 1080–1090, Mar. 2011.

[29] A. Noureldin, D. Irvine-Halliday, and M. P. Mintchev, "Accuracy limitations of FOG-based continuous measurement-while-drilling surveying instruments for horizontal wells," *IEEE Trans. Instrum. Meas.*, vol. 51, no. 6, pp. 1177–1191, Dec. 2002.

[30] M. S. Grewal, L. R. Weill, and A. P. Andrews, *Global positioning systems, inertial navigation, and integration*: John Wiley & Sons, 2007.

[31] Grejner-Brzezinska D.A., Yi Y. and Toth C.K.: “Bridging GPS gaps in urban canyons: the benefits of ZUPTs,” *Journal of Navigation,* vol. 48, no. 4, pp. 217-226, 2001.

[32] Y.-Y. Ben, F. Sun, W. Gao, and M.-H. Chen, “Study of zero velocity update for inertial navigation,” *J. Syst. Simul.*, vol. 20, no. 17, pp. 4639–4642, Sep. 2008.

[33] L. Ojeda and J. Borenstein, “Personal dead-reckoning system for GPS denied environments,” in *Proc. IEEE Int. Workshop Safety Security Rescue Robot.*, pp. 1-6, 2007.

[34] E. Pecht, “In-drilling alignment,” Ph.D. dissertation, Univ. Calgary, Calgary, Canada, 2006

[35] PetroWiki, “Measurement while drilling (MWD).”(2014) Internet: [Online]. Available: http://petrowiki.org/Measurement_while_drilling_%28MWD%29.

[36] I. Wasserman, D. Hahn, D. H. Nguyen, H. Reckmann, and J. Macpherson. "Mud-pulse telemetry sees step-change improvement with oscillating shear valves," *Oil & Gas Journal*, vol. 106, no. 24 pp. 39-40, 2008.

[37] L. Gao, D. Finley, W. Gardener, C. Robbins, E. Linyaev, and J. Moore. "Acoustic telemetry delivers more real-time downhole data in underbalanced drilling operations," in *Proc. SPE/IADC Drilling Conf.*, Feb. 2006.

[38] National Oilwell Varco, “IntelliServ.”(2014) Internet: [Online]. Available: http://www.nov.com/intelliserv/.

[39] M. Jellison, D. R. Hall, D. C. Howard, H. T. Hall, R. C. Long, R. B. Chandler, and D. S. Pixton. "Telemetry drill pipe: Enabling technology for the downhole internet," in *Proc. SPE/IADC Drilling Conf.*, 2003.

[40] Baker Hughes, “E-MTrak Electromagnetic Telemetry.” (2014) Internet: [Online]. Available: http://www.bakerhughes.com/products-and-services/drilling/drilling-services/measurement while-drilling/e-mtrak-electromagnetic-telemetry.

[41] E. Levinson, J. Ter Horst, and M. Willcocks, "The next generation marine inertial navigator is here now," *in Position Location and Navigation Symposium*, IEEE, 1994, pp. 121-127, 1994.

[42] E. Levinson and R. Majure, "Accuracy enhancement techniques applied to the marine ring laser inertial navigator (MARLIN)," *Navigation*, vol. 34, pp. 64-86, 1987.

[43] A. S. Jurkov, J. Cloutier, E. Pecht, and M. P. Mintchev, "In-drilling Alignment," U.S. Patent 7,823,661, Jun 24, 2008.

[44] P. Aggarwal, Z. Syed, X. Niu, and N. El-Sheimy, "A standard testing and calibration procedure for low cost MEMS inertial sensors and units," *Journal of Navigation*, vol. 61, pp. 323-336, Apr. 2008.

[45] C. Cristina, B. Ludovico, and C. Alessandro, "Design of a low-cost GPS/magnetometer system for land-based navigation: integration and autocalibration algorithms," in *TS07F—Mobile and Asset Mapping Systems,* pp.1-15, 2011.

[46] W. Abd-Elhamid, A. Osman, and A. Noureldin, "Wavelet multi-resolution analysis for enhancing the performance of integrated GPS and MEMS- based navigation systems," *Journal of Geomatica*, vol. 59, no. 1, pp. 61–72, Mar. 2005.

[47] A. Noureldin, T. B. Karamat, M. D. Eberts, and A. El-Shafie, "Performance enhancement of MEMS-based INS/GPS integration for low-cost navigation applications," *Vehicular Technology, IEEE Transactions on*, vol. 58, pp. 1077-1096, 2009.

[48] M. D. Eberts, "Performance enhancement of MEMS based INS/GPS integration for low cost navigation applications," M.S. thesis, Roy. Mil. College Canada, Kingston, ON, Canada, 2007.

[49] V. Bistrov, "Performance Analysis of Alignment Process of MEMS IMU," *International Journal of Navigation and Observation*, vol. 2012, pp. 1-11, Sep. 2012.

[50] “World Magnetic Model 2010 Calculator”, British Geological Survey. (2013). [Online]. Available: Internet: www.geomag.bgs.ac.uk/data_service/models_compass/wmm_calc.html.,

[51] A. T. Bourgoyne, K. K. Millheim, and M. E. Chenever, *Applied Drilling Engineering.* Richardson, TX, USA: Soc. Petroleum Eng., 1986.

[52] D. Kyle, M. Fripp, and K. Fink. "Acoustic telemetry for oilfield operations," *IEEE Mag. Instrum. Meas.*, no. 6 pp. 18-24, Dec. 2013.

[53] S. Farahani, *ZigBee Wireless Networks and Transceivers*. Oxford, UK: Newnes, 2011.

[54] D. Gislason, *ZigBee Wireless Networking*. Oxford, UK, Newnes, 2008.

[55] V. Varveropoulos, R. Taherian, “Wireless telemetry through drill pipe”, US Patent Application 20110018734, Jan. 27, 2011.

[56] Z. H. Wang, M. Poscente, D. Filip, M. Dimanchev and M. P. Mintchev, “Rotary in-drilling alignment using an autonomous MEMS-based inertial measurement unit for measurement-while-drilling processes,” *IEEE Mag. Instrum. Meas.,* no. 6 pp. 26-34, Dec. 2013.

[57] Analog Devices. “ADIS16488A: Ten Degrees of Freedom MEMS Inertial Sensor with Dynamic Orientation Outputs.” (2014) Internet [Online]. Available: http://www.analog.com/en/mems-sensors/mems inertial-measurement-units/adis16488a/products/product.html.

[58] Z. H. Wang, T. Li, M. McDougall, D. McCormack, and M. P. Mintchev, "Wireless data transmission options in rotary in-drilling alignment setups for multilateral oil drilling applications," *Journal of Information Theories and Applications,* vol. 21, pp. 154-161, 2014.

[59] R. Toulson and T. Wilmshurst, *Fast and effective embedded systems design: applying the ARM mbed. Oxford, UK: Newnes,* 2012.

[60] L. Rong, Z. Jianhui, and L. Fan, “Accuracy improvements of gyro-based measurement-while-drilling surveying instruments by a laser-testing method,” in *Proc. Int. Soc. Opt. Eng.*, pp.738, 2009.

[61] J. Kähler, A. Stranz, A. Waag, E. Peiner, “Packaging of MEMS and MOEMS for harsh environments,” *J. Micro/Nanolith. MEMS MOEMS*, vol. 11, no. (2), pp. 021202 1-7, May 2012.

[62] Oriental Motor. “0.9° Stepper Motors.” . Oriental Motor (2013). Internet: [Online]. Available: http://www.orientalmotor.com/products/stepper-motors/stepper-motor-only-0-9.html

[63] US Digital. “E5 Optical Kit Encoder”. US Digital (2013). Internet: [Online]. Available: http://www.usdigital.com/products/e5#description

[64] Digi International Inc. “Product Manual: XBee / XBee-PRO 802.15.4 RF Modules”. Internet: [Online]. Available: http://ftp1.digi.com/support/documentation/90000982_R.pdf

Appendix A: R-IDA apparatus materials

Most of the electronic components for the stepper motor systems were purchased from Oriental Motor Inc. (Tokyo, Japan), in order to maintain the operational consistency and compatibility. The high resolution optical rotary encoder was purchased from US Digital Inc. (Vancouver, WA, USA) mounted at the back shaft of the stepper motor.

Materials and steps required to build the R-IDA apparatus are described in details below.

1. Stepper motor

The PK245M-01BA 2 phase motor with a back shaft was used. The stepper motor is driven by the number of input pulses and controlled by the frequency of the input pulse signals. A basic step angle is 0.9°.

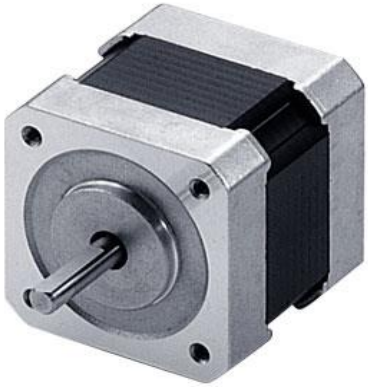


Figure A.1 PK245M-01BA motor [62]

2. Motor driver

The CMD 2112P microstep driver was used, which can minimize the basic step to 0.05625°, while at the same time, reducing the motor's vibration dramatically.

Figure A.2 CMD 2112P Micro step Driver [62]

3. Motor controller

The SCX-11 Controller was used, which came in a compact package. It connects to the computer using a micro USB cable and its GUI interference is very intuitive. The GUI contains and implements the rotation schemes, the rotary encoders feedback, and the rotation control panel. The function of program editor requires simple API commands to achieve different rotational schemes for the stepper motor. This helped us find the most efficient motion in the smallest time interval.

Figure A.3 SCX-11 Controller [62]

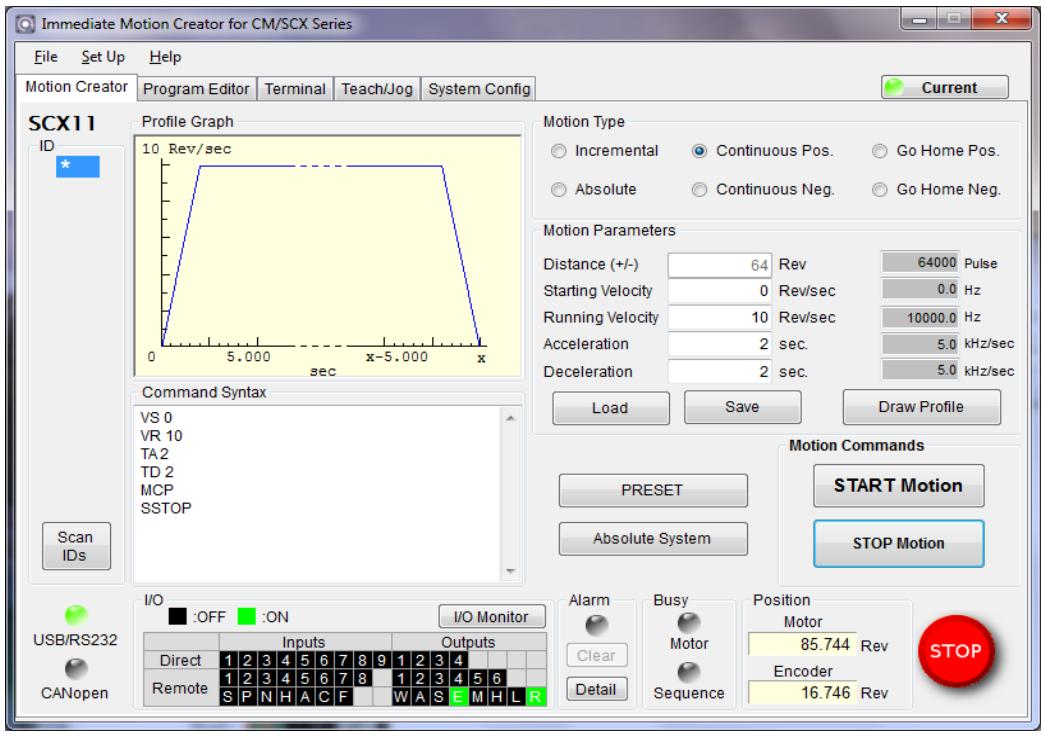


Figure A.4 GUI Interference of the controller

4. Optical rotary encoder

Figure A.5 E5 Rotary Encoder [63]

The E5 Optical Kit Rotary Encoder from the US Digital Inc. was used, with 1250 cycles per revolution (C/R). It measures the rotations as small as 0.288°. Each cycle can provide 1, 2 or 4 counts depending on how many signal edges the quadrature encoder counts.

The two output waveforms A and B are 90 degrees out of phase (Fig. A.6). These signals are decoded to produce a count up pulse or a count down pulse. The direction of the movement can be calculated based on which of the two signals rises first, as shown in Fig. A.6. To decode, the A and B outputs are read by software, via an interrupt on any edge or polling.

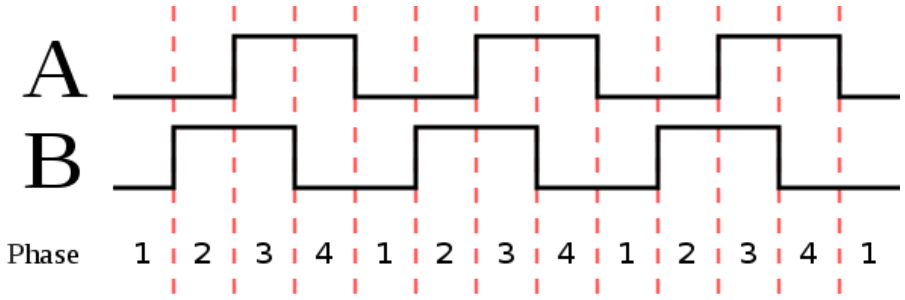


Figure A.6 A and B Output Square Waves in Quadrature [63]

Appendix B: X-CTU and XBee configuration setup

The XBee is a Wireless Microcontroller made by a company called Digi. It uses IEEE 802.14.5 Zigbee communication. The RF data rate is 250kbps transmitting at the ISM band of 2.4GHZ frequency. The indoor operating range is 40 meters and line of sight range is 120 meters. The XBee network must consist of one Coordinator, which is in charge of setting up the network and is required to be operational at all time (Fig. B.1). Multiple routers can be present in the network, if required, and function as a signal relaying mechanism between different routers or End Points. Multiple End Points may be present in the network, however, these End Points cannot relay signals. The Routers and End Point can be put into sleep mode to save power. For this application, the Pair topology was used. The Coordinator will be the module that connects to the PC end and the Router will be the XBP module that connects to the ADIS IMU end.

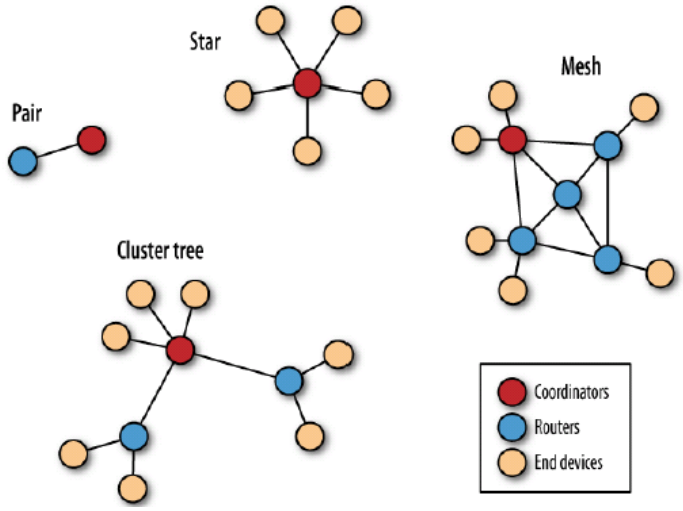


Figure B.1 Illustration of different possible XBee Topologies and the type of endpoints [64]

There are two modes of operation for XBee devices: transparent and command modes. In this application, the XBee is configured to be in transparent mode. In transparent mode, the communication is sent through the XBee. To enable the transparent mode, the XBee should be

set to AT command. The command mode, which is setup as the API mode on the XBee, is for handing the IMU's generated data. For example, the XBee with analog inputs reading data from a sensor and providing this sensed data to the network will be in the API mode. In this application transparent mode is sufficed. To setup the XBee the XCTU software is required on a PC to communicate with the XBee serially at 9600 baud, with 8-data bits without parity and 1-stop bit. This software is used to configure the XBee to operate at different modes (AT or API), data rate, personal area network (PAN) identification (ID), and destination serial number, pin setting, etc. The PAN ID is a network ID, which is an agreed upon network for which they will join and communicate in. The Destination serial number is used to send to a specific XBee with a unique serial number.

The power requirement for the module is 40 mA at 3.3V with an operating voltage range of 2.1 to 3.6V. The XBee module uses a smaller pin pitch then breadboards and proto boards. Because of this, an interface board is often used. It is a 20 pin device with 11 general purpose I/O pin GPIO and 4 analog input pins with maximum analog input reading of 1.2V. The Digital Input pins can be configured as analog input pins only in API mode. The following diagram in Fig. B.2 shows the pin-outs of the XBP device.

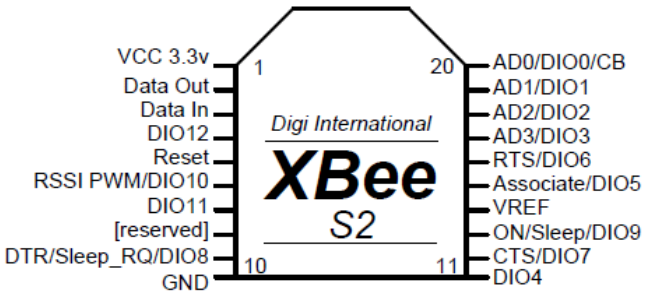


Figure B.2 XBP module pin-outs and signal names [64]

2. XBee Hardware Setup

In order for the PC to talk to the XBee device, a serial connection must be initiated. The X-CTU and the XBee require a serial connection. To overcome these limitations, a USB to serial converter is required.

The following are required for this test procedure:

1) XBIB-U-DEV interface board with USB cable (Digi International Inc, MN, USA). The XBIB-U-DEV board comes with a XBee and XBee Pro 802.15.4 OEM RF Module's development kit

2) The X-CTU Software (Digi International Inc, MN, USA)

a. Download the latest X-CTU Software from www.digi.com.

b. When at the webpage, hover the mouse over the Support at the top menu bar,

c. Select the Diagnostics, Utilities and MIBs

d. Search for XCTU and download the latest version of the XCTU installer and install it.

e. Digi's windows driver downloaded from www.digi.com.

3. XBP Programming

1) Install a new XBee Pro onto XBIB-U-DEV board as shown in Fig. B.3.

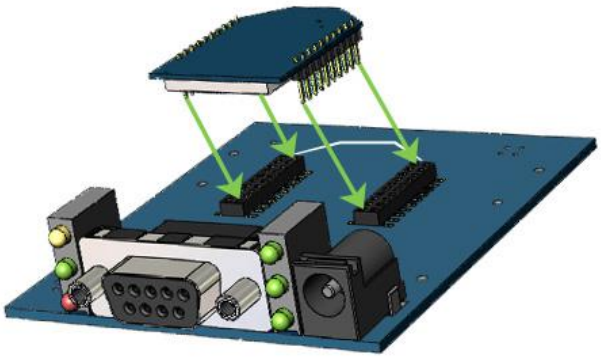


Figure B.3 Location of XBee Pro on XBIB-U-DEV board [64].

2) Plug USB cable from PC to XBIB-U-DEV board. At the first time of connecting the hardware connection, windows will ask for XBee Pro driver.

3) Launch Digi's X-CTU software from start menu -> Digi-MaxStream -> X-CTU

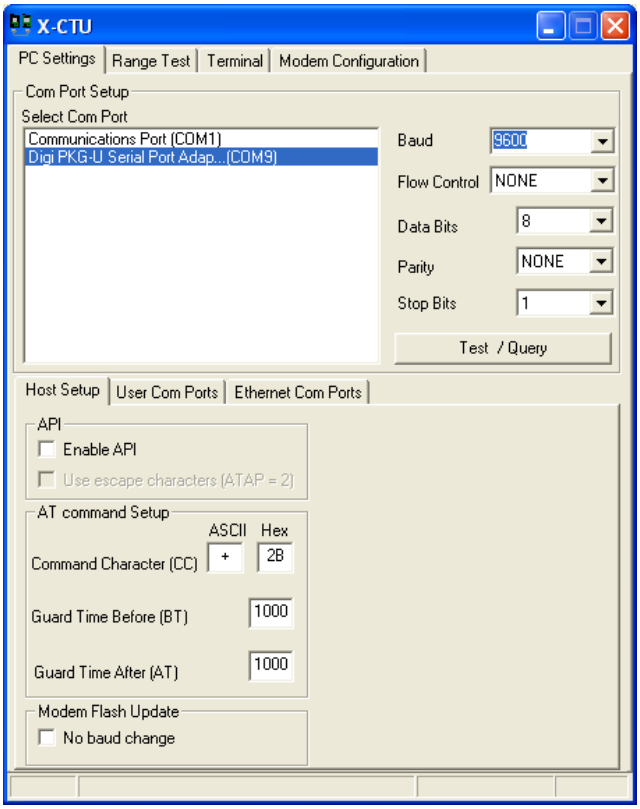


Figure B.4 X-CTU program

4) Under PC Settings tab, select COM Port "Digi PKG-U Serial Port Adap… (COMX)" and select Baud "9600". Click "Test / Query" button and the following window will pop up (See Fig. B.4). If it is a brand new XBee Pro, the Modem type = XBP24 and firmware version. This verifies that communication with the XBee Pro is successful.

5) Click "OK" to close window and Click on the Modem Configuration tab and select Modem XBP24.

Router Setup

1) Select the Modem to be XBEE-PRO XBP24ZB to function as ZIGBEE END DEVICE AT and Version number to be the latest version.

2) Set the PAN ID to an arbitrary number 12345.

3) Set the JV – Channel Verification to 1 so that it will more be more join the network more readily.

4) Set the Destination Address to 0 so that this Router will communicate to the Coordinator only.

5) Once XBP24 is selected, the following setting will appear.

6) Click on “PC Settings” tab and select Baud to “38400”.

Coordinator Setup

1) Go to the PC Setting again and select the COM port that is connected to the Coordinator XBee module.

2) Leave the COM port setting to 9600-8-N-1 setting

3) Select Modem Configuration (See Fig. B.5).

4) Make sure the PAN ID is the same on the Coordinator and the Router. Set the same number for receiver and transmitter.

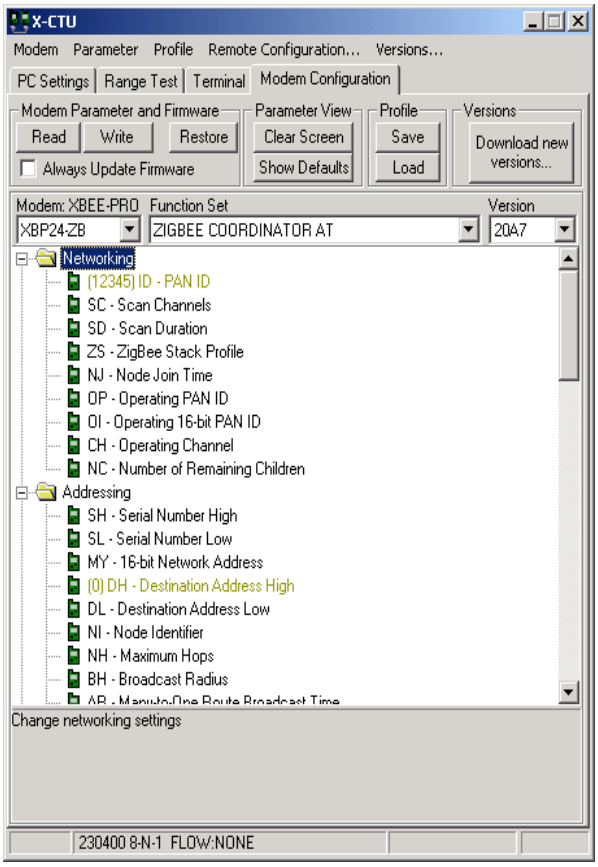


Figure B.5 Coordinator Setup

In the end, click Write to write the configuration onto the router.

Appendix C: Matlab code

```
%Navigation algorithms in Matlab
%70 degrees and 15 seconds outage azimuth assessment in ZUPT&R-IDA
%%%%%%%%%%%%%%%
clear all;
clc;
%Load Memsense IMUs calibration files from the six-position test
load zup5min.txt;
load zdown4min.txt;
load xup4min.txt;
load xdown4min.txt;
load yup4min.txt;
load ydown4min.txt;
load xCCW3min.txt;
load xCW3min.txt;
load yCCW3min.txt;
load yCW3min.txt;
load zCCW3min.txt;
load zCW3min.txt;
L= length(zCW3min);
%Set calibration variables
fzn= zup5min(1:L, 6);
fzn2= zdown4min(1:L, 6);
```

```
fxn= xup4min(1:L, 4);

fxn2= xdown4min(1:L, 4);

fyn= yup4min(1:L, 5);

fyn2= ydown4min(1:L, 5);

wx= xCCW3min(1:L, 1);

wxn= xCW3min(1:L, 1);

wy= yCCW3min(1:L, 2);

wyn= yCW3min(1:L, 2);

wz= zCCW3min(1:L, 3);

wzn= zCW3min(1:L, 3);

%Declare normal gravity constants

%Reference: D.H Titterton and J.L. Weston, Strapdown inertial navigation Tech.

a1=9.7803267715;  a4=-1*0.000003087691089;

a2=0.0052790414;  a5=0.000000004397731;

a3=0.0000232718;  a6=0.000000000000721;

lab_latitude = 51.0794278*pi/180; % Convert to [radians]

lab_longitude = -1*114.132858*pi/180; % Convert to [radians]

lab_altitude = 1106.6; % [meters]

SemiMajorAxis=6378137; % [meters]

SemiMinorAxis=6356752.3141;  % [meters]

EarthMeanRadius=(SemiMajorAxis+SemiMinorAxis)/2;

EarthRotation = 15.041067*pi/(180*3600); %convert to [radians]

e2 = 1 - SemiMinorAxis^2/SemiMajorAxis^2; %not sure what is e2
```

```
%Calculate gravity of the lab
%Reference: ENGO623 Course notes
NormalGravity = a1*(1 + a2*(sin(lab_latitude)^2) + a3*(sin(lab_latitude)^4)) +...
    (a4 + a5*(sin(lab_latitude)^2))*lab_altitude + a6*(lab_altitude^2);
AnomalousGravity = 0; %Change in gravity
%Calculate bias for the accelerometer
Bias_accl_X = (mean(fxn2)+mean(fxn))/2;
Bias_accl_Y = (mean(fyn2)+mean(fyn))/2;
Bias_accl_Z = (mean(fzn2)+mean(fzn))/2;
%Calculate Scale factor for the accelerometer
Sca_fact_accl_x = (mean(fxn2)-mean(fxn)-2*NormalGravity)/(2*NormalGravity);
Sca_fact_accl_y = (mean(fyn2)-mean(fyn)-2*NormalGravity)/(2*NormalGravity);
Sca_fact_accl_z = (mean(fzn2)-mean(fzn)-2*NormalGravity)/(2*NormalGravity);
%Calculate bias for the gyroscope
Bias_gyro_X = (mean(wx)+mean(wxn))/2;
Bias_gyro_Y = (mean(wy)+mean(wyn))/2;
Bias_gyro_Z = (mean(wz)+mean(wzn))/2;
%Calculate Scale factor for the gyroscope
%Reference rotation rate is 525.5 deg/s
Wref = 525.5; %[deg/s]
Sca_fact_gyro_x = (mean(wx) - mean(wxn))/(2*Wref);
Sca_fact_gyro_y = (mean(wy) - mean(wyn))/(2*Wref);
Sca_fact_gyro_z = (mean(wz) - mean(wzn))/(2*Wref);
```

```
% Six Position Test can not calibrate for the following
%non-orthogonalities of accelerometer and gyroscope.
% The misalignment parameters are set to be 0 to avoid "non-declared variables"
Accel_Misalign_xy = 0;
Accel_Misalign_xz = 0;
Accel_Misalign_yx = 0;
Accel_Misalign_yz = 0;
Accel_Misalign_zy = 0;
Accel_Misalign_zx = 0;
Force_Proj_x = 0; %There is no alignment thus no projection on each axis
Force_Proj_y = 0;
Force_Proj_z = 0;
Gyro_Misalign_xy = 0;
Gyro_Misalign_xz = 0;
Gyro_Misalign_yx = 0;
Gyro_Misalign_yz = 0;
Gyro_Misalign_zx = 0;
Gyro_Misalign_zy = 0;
Rot_Proj_x = 0;
Rot_Proj_y = 0;
Rot_Proj_z = 0
%based on navigational algorithm outlined by
%Oleg Salychev, 1998, Inertial systems in navigation and
```

```
%Geophysics, gyroscope and accelerometer error model
% Accelerometers
% Stochastic
Stoch_Noise_fx = 0; % 0 b/c stochastic noise be calibrated
Stoch_Noise_fy = 0; % 0 b/c stochastic noise be calibrated
Stoch_Noise_fz = 0; % 0 b/c stochastic noise be calibrated
% Deterministic
Accel_Error_x = Bias_accl_X + Sca_fact_accl_x*Force_Proj_x +
Accel_Misalign_xy*Force_Proj_y+ Accel_Misalign_xz*Force_Proj_z;
Accel_Error_y = Bias_accl_Y + Sca_fact_accl_y*Force_Proj_y +
Accel_Misalign_yx*Force_Proj_x+ Accel_Misalign_yz*Force_Proj_z;
Accel_Error_z = Bias_accl_Z + Sca_fact_accl_z*Force_Proj_z +
Accel_Misalign_zy*Force_Proj_y+ Accel_Misalign_zx*Force_Proj_x;
% Gyroscopes
% Stochastic
Stoch_Noise_wx = 0; % 0 b/c stochastic noise be calibrated
Stoch_Noise_wy = 0; % 0 b/c stochastic noise be calibrated
Stoch_Noise_wz = 0; % 0 b/c stochastic noise be calibrated
% Deterministic
Gyro_Error_x = Bias_gyro_X + Sca_fact_gyro_x*Rot_Proj_x + Gyro_Misalign_xy*Rot_Proj_y
+ Gyro_Misalign_xz*Rot_Proj_z;
Gyro_Error_y = Bias_gyro_Y + Sca_fact_gyro_y*Rot_Proj_y+ Gyro_Misalign_yx*Rot_Proj_x
+ Gyro_Misalign_yz*Rot_Proj_z;
```

```
Gyro_Error_z = Bias_gyro_Z + Sca_fact_gyro_z*Rot_Proj_z+ Gyro_Misalign_zx*Rot_Proj_x
+ Gyro_Misalign_zy*Rot_Proj_y;
%Load alignment files
load ZUPT_2_2_15sec.txt; %file name is "Coarse_Alignment_file"
Alignment = (ZUPT_2_2_15sec);
Raw_data = (ZUPT_2_2_15sec);
Xgyro = Raw_data(:,1);
Ygyro = Raw_data(:,2);
Zgyro = Raw_data(:,3);
Xaccel = Raw_data(:,4);
Yaccel = Raw_data(:,5);
Zaccel = Raw_data(:,6);
Xmagnetometer = Raw_data(:,7);
Ymagnetometer = Raw_data(:,8);
Zmagnetometer = Raw_data(:,9);
changdu = length(Xgyro);
%Wavelet denoising level 5
scal = 'one';
Xgyro_new = wden(Xgyro, 'rigrsure', 's', scal, 5, 'db5');
Ygyro_new = wden(Ygyro, 'rigrsure', 's', scal, 5, 'db5');
Zgyro_new = wden(Zgyro, 'rigrsure', 's', scal, 5, 'db5');
Xaccel_new = wden(Xaccel, 'rigrsure', 's', scal, 5, 'db5');
Yaccel_new = wden(Yaccel, 'rigrsure', 's', scal, 5, 'db5');
```

```
Zaccel_new = wden(Zaccel, 'rigrsure', 's', scal, 5, 'db5');
Alignment = [Xgyro_new Ygyro_new Zgyro_new Xaccel_new Yaccel_new Zaccel_new
Xmagnetometer Ymagnetometer Zmagnetometer];
wx_align = Alignment(1:17800,1);
wy_align = Alignment(1:17800,2);
wz_align = Alignment(1:17800,3);
fx_align = Alignment(1:17800,4);
fy_align = Alignment(1:17800,5);
fz_align = Alignment(1:17800,6);
%Magnetometer measurements along x  and y axis in horizontal plane
mx_align = Alignment(1:17800,7);
my_align = Alignment(1:17800,8);
%Output message rate of MEMSense IMU
fs = 150;
 %reference measurement sampling since we take it as stable velocity
%LSB value has been converted by the IMU in degrees.
%It has to be converted to radians!
%%Magnetometer Calibration%%
%load magnetometer calibration file
load magnetometer_calibration.txt
mx_cal = magnetometer_calibration(:,7);
my_cal = magnetometer_calibration(:,8);
%Scale Factors
```

```
Sx = max(1, (max(my_cal)-min(my_cal))/((max(mx_cal)-min(mx_cal))));

Sy = max(1, (max(mx_cal)-min(mx_cal))/((max(my_cal)-min(my_cal))));

%Biases

Bx = ((max(mx_cal)-min(mx_cal))*1/2 - max(my_cal))*Sx;

By = ((max(my_cal)-min(my_cal))*1/2 - max(mx_cal))*Sy;

Mx_align = Sx*mean(mx_align) + Bx;

My_align = Sy*mean(my_align) + By;

%MEMS accelerometers levelling can only determin initial attitude angles r and p

Roll(1)= asin(mean(fxn2)/NormalGravity); % [radians]

Pitch(1)= asin(mean(fyn2)/NormalGravity); % [radians]

%Azimuth

Azimuth(1) = atan2(Mx_align,My_align)*180/pi - 14.666; %The first part is local heading

%%%%%%%%%%%

% Initialize transformation matrix R_b_l

%Order is R3(A)R2(r)R1(p)

R_b_l(1,1) = cos(Azimuth(1))*cos(Roll(1)) - sin(Azimuth(1))*sin(Pitch(1))*sin(Roll(1));

R_b_l(1,2) = -sin(Azimuth(1))*cos(Pitch(1));

R_b_l(1,3) = cos(Azimuth(1))*sin(Roll(1)) + sin(Azimuth(1))*sin(Pitch(1))*cos(Roll(1));

R_b_l(2,1) = sin(Azimuth(1))*cos(Roll(1)) + cos(Azimuth(1))*sin(Pitch(1))*sin(Roll(1));

R_b_l(2,2) = cos(Azimuth(1))*cos(Pitch(1));

R_b_l(2,3) = sin(Azimuth(1))*sin(Roll(1)) - cos(Azimuth(1))*sin(Pitch(1))*cos(Roll(1));

R_b_l(3,1) = -cos(Pitch(1))*sin(Roll(1));

R_b_l(3,2) = sin(Pitch(1));
```

```
R_b_l(3,3) = cos(Pitch(1))*cos(Roll(1));
% Free memory
clear magnetometer_calibration.txt;
clear zup4min.txt;
clear zdown4min.txt;
clear xup4min.txt;
clear xdown4min.txt;
clear yup4min.txt;
clear ydown4min.txt;
clear xCCW3min.txt;
clear xCW3min.txt;
clear yCCW3min.txt;
clear yCW3min.txt;
clear zCCW3min.txt;
clear zCW3min.txt;
% Declare normal gravity constants
g_l = [0;0;NormalGravity]; % Initial gravity vector  %in the function
%Start the navigation algorithm
% initialize quaternion with alignment data
%is used to get and update rotation matrix
% Reference: Section 2, ENGO623 Course Notes
q4 = (1/2)*sqrt(1+R_b_l(1,1)+R_b_l(2,2)+R_b_l(3,3));
q1 = ((1/4)*(R_b_l(3,2) - R_b_l(2,3)))/q4;
```

```
q2 = ((1/4)*(R_b_l(1,3) - R_b_l(3,1)))/q4;

q3 = ((1/4)*(R_b_l(2,1) - R_b_l(1,2)))/q4;

Q_prev = [q1;q2;q3;q4];

% Declare Navigation Constants - IMU is initially at rest

Vn = 0; % Assume initial North velocity is zero, local level frame

Ve = 0; % Assume initial East velocity is zero, local level frame

Vu = 0; % Assume initial Up velocity is zero, local level frame

V_l(:,1) = [Ve; Vn; Vu]; % *Initial velocity 0 for N E U this is the local level frame velocity

%Reference: Section 2, ENGO623 Course Notes

delta_V_l_prev = [0;0;0];   %Initialize delta_V_l for navigational algorithm

w_il_b(1) = 0; % Initially no rotation between reference frames

% Starting Position

latitude = lab_latitude;

longitude = lab_longitude;

altitude = lab_altitude;

% Load navigation data

load ZUPT_2_2_15sec.txt

%include 2min static,15 seconds rotation at 5rev, and 105 seconds ZUPT

Raw_data = (ZUPT_2_2_15sec);

Xgyro = Raw_data(:,1);

Ygyro = Raw_data(:,2);

Zgyro = Raw_data(:,3);

Xaccel = Raw_data(:,4);
```

```
Yaccel = Raw_data(:,5);

Zaccel = Raw_data(:,6);

Xmagnetometer = Raw_data(:,7);

Ymagnetometer = Raw_data(:,8);

Zmagnetometer = Raw_data(:,9);

changdu = length(Xgyro);

%Wavelet denoising level 5

scal = 'one';

Xgyro_new = wden(Xgyro, 'rigrsure', 's', scal, 5, 'db5');

Ygyro_new = wden(Ygyro, 'rigrsure', 's', scal, 5, 'db5');

Zgyro_new = wden(Zgyro, 'rigrsure', 's', scal, 5, 'db5');

Xaccel_new = wden(Xaccel, 'rigrsure', 's', scal, 5, 'db5');

Yaccel_new = wden(Yaccel, 'rigrsure', 's', scal, 5, 'db5');

Zaccel_new = wden(Zaccel, 'rigrsure', 's', scal, 5, 'db5');

Navigation = [Xgyro_new Ygyro_new Zgyro_new Xaccel_new Yaccel_new Zaccel_new

Xmagnetometer Ymagnetometer Zmagnetometer];

wx_nav = Navigation(:,1);

wy_nav = Navigation(:,2);

wz_nav = Navigation(:,3);

fx_nav = Navigation(:,4);

fy_nav = Navigation(:,5);

fz_nav = Navigation(:,6);

%It requires incremental value for algorithm
```

```
fx_nav = (1/fs).*fx_nav;

fy_nav = (1/fs).*fy_nav;

fz_nav = (1/fs).*fz_nav;

% Convert wb to [r/s]

wx_nav = (1/fs).*wx_nav*pi/180;

wy_nav = (1/fs).*wy_nav*pi/180;

wz_nav = (1/fs).*wz_nav*pi/180;

% Remove deterministic errors from navigation file

fx = (fx_nav + Bias_accl_X)*(1+Sca_fact_accl_x);

fy = (fy_nav + Bias_accl_Y)*(1+Sca_fact_accl_y);

fz = (fz_nav + Bias_accl_Z)*(1+Sca_fact_accl_z);

wx = wx_nav + Gyro_Error_x*pi/180;

wy = wy_nav + Gyro_Error_y*pi/180;

wz = wz_nav + Gyro_Error_z*pi/180;

Navigation_Length = length(fx_nav);

%Start of RIDA Constants

k = 0;

Lat = lab_latitude;   % Lab latitude [radians]

we = EarthRotation;  % Earth rotation [rad/sec]

g = NormalGravity;       % Gravity [m/sec^2]

Lam_dot = 0;    % Rate change in latitude during RIDA=0

Long_dot = 0;   % Rate change in longtitude during RIDA=0

% Construct transition matrix
```

```
% x=Fx+Gw, the dynamics matrix F. For the F matrix, include OM and OM_hat
Om_N_hat = (2*we + Lam_dot)*cos(Lat); %2*earthrotation*cos(latitude)
Om_E_hat = -Long_dot; %0, there is no earthrotation devation in East
Om_D_hat = -(2*we+Lam_dot)*sin(Lat); %He chose NED
OM_hat = [0, -Om_D_hat, Om_E_hat; Om_D_hat, 0, -Om_N_hat; -Om_E_hat, Om_N_hat, 0];
%skew-symmetric matrix
Om_N = (we + Lam_dot)*cos(Lat);
Om_E = -Long_dot;
Om_D = -(we + Lam_dot)*sin(Lat);
OM = [0, -Om_D, Om_E; Om_D, 0, -Om_N;-Om_E, Om_N, 0];
o_hat = zeros(3); %To form the dynamic matrix F
I_hat = eye(3);
oo = zeros(6,12); % for RIDA w0 = 281.25*pi/180;
% and dt = 1/fs; oo(9,9)=-w0*sin(w0*dt); oo(12,12)=Acc_Process_Noise_fz%
%add a gyroscope and accelerometer bias to the dynamic matrix
% Put the beta values in the dynamic matrix
oo(1,7)=-1/30;
oo(2,8)=-1/30;
oo(3,9)=-1/30;
oo(4,10)=-1/15;
oo(5,11)=-1/15;
oo(6,12)=-1/15;
% Measurement matrix Z=HX+V
```

```
H = zeros(3,12);

H(1,1) = 1; % Vn

H(2,2) = 1; % Ve

H(3,3) = 1; % Vu

% Velocity measurement noise

R(1,1) = 0.288; %0.288 for the encoder

R(2,2) = 0.288;

R(3,3) = 0.288;

% Reference: Introduction to Random Signals and Applied Kalman

% Filtering, Brown & Hwang

% R_pred used in the KF gain calculation, These 3 are very big numbers

R_pred(1,1) = 1e20; % It is the format of the variable

R_pred(2,2) = 1e20;

R_pred(3,3) = 1e20;

% Accelerometer process noise

% Data is from Allen analysis(each sensor parameters were determined)

% MEMS sensor has shorter correlation time than FOGs

Alpha_fx = 1/15; % Units [1/s],

Acc_Std_Dev_fx = 0.0020;

Acc_Process_Noise_fx = Acc_Std_Dev_fx*sqrt(2*Alpha_fx);

Alpha_fy = 1/15; % Units [1/s], from LN-200 manual - sheet 41

%Acc_Std_Dev_fy = 0.003; % Wavelet denosing

Acc_Std_Dev_fy = 0.0019;
```

```
Acc_Process_Noise_fy = Acc_Std_Dev_fy*sqrt(2*Alpha_fy);

Alpha_fz = 1/15; % Units [1/s],

%Acc_Std_Dev_fz = 0.0002112; % wavelet analysis

Acc_Std_Dev_fz = 0.0022;

Acc_Process_Noise_fz = Acc_Std_Dev_fz*sqrt(2*Alpha_fz);

%Gyroscope

%Gyro_Std_Dev_wx = 0.0817;% From wavelet analysis

%Actual value for gyro STD is bigger, from the data sheet

Gyro_Std_Dev_wx = 0.65; % Units [deg/s],

%The correlation time for gyros should be very small

Beta_wx = 1/5; % 5 sec correlation time for the gyro, while LN-200 is 100sec

Gyro_Process_Noise_wx = Gyro_Std_Dev_wx*sqrt(2*Beta_wx);

Beta_wy = 1/5; % Units [1/s]

Gyro_Std_Dev_wy = 0.65; % Units [deg/s],

Gyro_Process_Noise_wy = Gyro_Std_Dev_wy*sqrt(2*Beta_wy);

Beta_wz = 1/5; % Units [1/s],

Gyro_Std_Dev_wz = 0.65; % Units [deg/s],

Gyro_Process_Noise_wz = Gyro_Std_Dev_wz*sqrt(2*Beta_wz);

%Construct the process noisy matrix

w = [0;0;0;0;0;0;Gyro_Process_Noise_wx;Gyro_Process_Noise_wy;Gyro_Process_Noise_wz;...

 Acc_Process_Noise_fx;Acc_Process_Noise_fy;Acc_Process_Noise_fz];

Qd = w*w';

% Error Covariance Matrix - Initialize each covariance matrix value
```

```
% E. Pecht's thesis - Pg. 56
P(1,1) = 0.2^2;  % North velocity error [m/s]
P(2,2) = 0.2^2;  % East velocity error [m/s]
P(3,3) = 0.2^2;  % Down velocity error [m/s]
P(4,4) = (1*pi/180)^2;    % Initial roll (North direction) error [rad]
P(5,5) = (1*pi/180)^2;    % Initial pitch (East direction) error [rad]
P(6,6) = (2*pi/180)^2;    % Initial azimuth (Down direction) error [rad]
%The bias is 30mg for accelerometer
P(7,7) = (9.8*30*100^(-3))^2; % Initial north accelerometer drift [m/s^2]^2
P(8,8) = (9.8*30*100^(-3))^2; % Initial east accelerometer drift [m/s^2]^2
P(9,9) = (9.8*30*100^(-3))^2; % Initial down accelerometer drift [m/s^2]^2
%The drift is 20-30deg/h for gyroscope
P(10,10) = (30*(pi/180)*(1/3600))^2; % Initial north (roll) gyro drift [rad/s]^2
P(11,11) = (30*(pi/180)*(1/3600))^2; % Initial east (pitch) gyro drift [rad/s]^2
P(12,12) = (30*(pi/180)*(1/3600))^2; % Initial down (azimuth) gyro drift [rad/s]^2
n = 12; % Number of states does not use
I = eye(12);
PMINUS = P; %P(-) is the initial P
dt = 1/fs;
%dt = T/dt;  number of steps
%Initial error states
% Velocity errors in N, E, and U
Vn = sqrt(P(1,1)); Ve = sqrt(P(2,2)); Vu = sqrt(P(3,3));
```

```
% Altitude angle errors in r, p, and A
Phi_n = sqrt(P(4,4)); Phi_e = sqrt(P(5,5)); Phi_u = sqrt(P(6,6));
% Accelerometer bias N, E, and Up
Bn = sqrt(P(7,7)); Be = sqrt(P(8,8)); Bu = sqrt(P(9,9));
% Gyroscope drift in N, E, Up
Dn = sqrt(P(10,10)); De = sqrt(P(11,11)); Du = sqrt(P(12,12));
% Initial estimate x(-) which is also initial system state vector
x_est_prior_k =[Vn; Ve; Vu; Phi_n; Phi_e; Phi_u; Bn; Be; Bu; Dn; De; Du];
Navigation_Length = length(fx_nav);
Coarse_Align_Start = ((2 * 60) +1) * 150;
for nav_increment = Coarse_Align_Start:Navigation_Length
%     For the measurement outages
    Outage_Start = Coarse_Align_Start;
    outage_length = 15; %This will give us 15 seconds of outage length
% Change the outage_length for however long you want
    Outage_Stop = (outage_length * 150) + Coarse_Align_Start;
  % Estimate the effect of earth rotation in the l-frame
  % Frame change
    Ve_tk = V_l(1); %Here V_1(1)=V_1(2)=V_1(3)=0
    Vn_tk = V_l(2);
    Vu_tk = V_l(3);
    % Reference: G. Siouris "Aerospace Avionics Systems: A Modern
    % Synthesis" Academic Press, Portland
```

```
h = altitude;
% Radius of curvature in the east-west direction
N = SemiMajorAxis/sqrt(1-e2*(sin(latitude)^2));
% Radius of curvature in the north-south direction
M = SemiMajorAxis*(1-e2)/((1-e2*(sin(latitude)^2))^(3/2));
% Attitude computation
% Reference: Section 2 ENGO623 Course Notes
w_ie_l = [0;EarthRotation*cos(latitude);...
      EarthRotation*sin(latitude)];
w_el_l = [-Vn_tk/(M + h);Ve_tk/(N + h);(Ve_tk*tan(latitude))/(N + h)];
w_il_l = w_ie_l + w_el_l;
R_l_b = R_b_l';
w_il_b = R_l_b * w_il_l;
theta_il_b = w_il_b./fs;    % theta_il_b is at tk (R_l_b was calculated last loop)
% Update the quaternion vector
% Calculate angular increments at tk+1
% nav_increment = 2:Navigation_Length
delta_wx = wx(nav_increment)/fs;
delta_wy = wy(nav_increment)/fs;
delta_wz = wz(nav_increment)/fs;
theta_ib_b = [delta_wx;delta_wy;delta_wz];  % theta_ib_b is at tk+1
theta_lb_b = theta_ib_b - theta_il_b;
% Build skew-symmetric matrix
```

```
% Reference: Section(2), ENGO623 Course Notes
tx = theta_lb_b(1,1); % measurement from gyro wx
ty = theta_lb_b(2,1); % measurement from gyro wy
tz = theta_lb_b(3,1); % measurement from gyro wz
omega1 = [0,tz,-ty,tx];
omega2 = [-tz,0,tx,ty];
omega3 = [ty,-tx,0,tz];
omega4 = [-tx,-ty,-tz,0];
omega = [omega1;omega2;omega3;omega4];  % omega is at tk+1;
% Calculate new quaternion
Q = Q_prev + (1/2)*omega*Q_prev;
% Computational errors occur in calculating the new quaterion vectors
% that must be compensated for.
% Reference: Section(2) P41, ENGO623 Winter 2003 Course Notes
delta = 1 - (q1^2 + q2^2 + q3^2 + q4^2);
Q = Q*(1 + delta/2);
q1 = Q(1);
q2 = Q(2);
q3 = Q(3);
q4 = Q(4);
% Calculate R_b_l and update attitude angles
R_b_l(1,1) = q1^2 - q2^2 - q3^2 + q4^2;
R_b_l(1,2) = 2*(q1*q2 - q3*q4);
```

```
  R_b_l(1,3) = 2*(q1*q3 + q2*q4);
  R_b_l(2,1) = 2*(q1*q2 + q3*q4);
  R_b_l(2,2) = -q1^2 + q2^2 - q3^2 + q4^2;
  R_b_l(2,3) = 2*(q2*q3 - q1*q4);
  R_b_l(3,1) = 2*(q1*q3 - q2*q4);
  R_b_l(3,2) = 2*(q2*q3 + q1*q4);
  R_b_l(3,3) = -q1^2 - q2^2 + q3^2 + q4^2;
  Pitch = atan(R_b_l(3,2)/(sqrt(R_b_l(1,2)^2 + R_b_l(2,2)^2)));
  Roll = atan(R_b_l(3,1)/R_b_l(3,3));
  Azimuth(nav_increment) = atan(-R_b_l(1,2)/R_b_l(2,2));
% Determine the body velocity change
% f_b at tk+1
  f_b = [fx(nav_increment);fy(nav_increment);fz(nav_increment)];
  delta_v_b = (1/fs).*f_b; % The same with gyro increment
  Measured_Velocity = R_b_l*delta_v_b; % Measured velocity incremental from body
  %frame to local level frame. at tk+1
  % Build skew-symmetric matricies
  x = w_ie_l(1,1); y = w_ie_l(2,1); z = w_ie_l(3,1); % w_ie_l is at tk
  omega_ie_l = [0,-z,y;z,0,-x;-y,x,0];
  x = w_el_l(1,1); y = w_el_l(2,1); z = w_el_l(3,1); % w_el_l is at tk
  omega_el_l = [0,-z,y;z,0,-x;-y,x,0];
  % Okay to use w_ie_l, w_el_l, g_l at tk because of small difference between samples at fs.
  Coriolis_Correction = (1/fs).*(2.*omega_ie_l + omega_el_l)*V_l;
```

```
Gravity_Correction = (1/fs).*g_l;
 % Subtract gravity correction because g in -ve z direction
delta_V_l = Measured_Velocity - Coriolis_Correction - Gravity_Correction;
% Compute the velocity components
V_l = V_l + (1/2)*(delta_V_l_prev + delta_V_l);
% Compute change is altitude (height)
Vu_tk1 = V_l(3);
altitude = altitude+ (1/2)*(Vu_tk1 + Vu_tk)*(1/fs);
% Update the latitude
Vn_tk1 = V_l(2);
latitude = latitude + (1/2)*((Vn_tk1 + Vn_tk)/(EarthMeanRadius +...
   altitude))*(1/fs);
% Update the longitude
Ve_tk1 = V_l(1);
longitude = longitude + (1/2)*((Ve_tk1+Ve_tk)/...
   ((EarthMeanRadius + altitude)*cos(latitude)))*(1/fs);
% Here begins Kalman Filter - Fine Alignment
% Scenerio 1 - No reference measurements available (outage).  The
%              Kalman Filter acts as an optimal predictor
% Scenerio 2 - ZUPT - Measurements available, no movement
% Scenerio 3 – R-IDA - Measurements available with movement
% Note all 3 scenerios have the same steps, except KF Step 2.
f_b = [fx(nav_increment);fy(nav_increment);fz(nav_increment)];
```

```
f_LLF = R_b_l*f_b;

Fd = -f_LLF(3);%Used the NED coordinate

Fe = f_LLF(1);

Fn = f_LLF(2);

Fo = [0 -Fd Fe; Fd 0 -Fn; -Fe Fn 0];

% Reference: Fundamental of INS/GPS integration

A = [OM_hat Fo R_b_l o_hat; o_hat OM o_hat R_b_l; oo];

% Discrete transition matrix

PHI = expm(A*dt);

% Measurements available - manually set the time outages

% During measurement outages, do not update the error estimate with the

% Kalman Gain (minimizing the covariance), only the transition matrix

if ((nav_increment > Outage_Start) && (nav_increment < Outage_Stop))

        %KF step 1 compute Kalman gain

        K_gain = (PMINUS*H')*inv(H*PMINUS*H' + R_pred);

        x_est = x_est_prior_k + K_gain*(z_k - H*x_est_prior_k);

else

        K_gain = (PMINUS*H')*inv(H*PMINUS*H' + R);

        simulated_ZUPT = rand(3,1)*0.0001;

        z_k = simulated_ZUPT;

        x_est = x_est_prior_k + K_gain*(z_k - H*x_est_prior_k)

end

%KF step 3. Compute error covariance for updated estimate
```

```
    PPLUS = (I - K_gain*H)*PMINUS; %PPLUS is the updated P(+), from P(-)
    PPLUS = 0.5*(PPLUS + PPLUS.'); % Ensure matrix is symmetric
    COVAR(nav_increment) = PPLUS(6,6);
    PMINUS = PHI*PPLUS*PHI' + Qd; %Update the priori covariance matrix
    %Qd = w*w' w is the process noise matrix
    % KF step 4. Project ahead for next scan
    x_est_prior_k = PHI*x_est;
    % Construct error estimates for plotting
    errors(nav_increment) = x_est(6);
    xgyroerrors(nav_increment) = x_est(10);
    ygyroerrors(nav_increment) = x_est(11);
    zgyroerrors(nav_increment) = x_est(12);
    xveloerrors(nav_increment) = x_est(1);
    yveloerrors(nav_increment) = x_est(2);
    zveloerrors(nav_increment) = x_est(3);
    xaccleerrors(nav_increment) = x_est(7);
    yaccleerrors(nav_increment) = x_est(8);
    zaccleerrors(nav_increment) = x_est(9);
    % Updata variables
    Q_prev = Q; %update quaterion matrix
    delta_V_l_prev = delta_V_l; %update velocity error states
end
%For RIDA if ((nav_increment > Outage_Start) && (nav_increment < Outage_Stop))
```

```
%A = [OM_hat Fo R_b_l o_hat; o_hat OM o_hat R_b_l; oo];
%PHI = expm(A*dt); PHI = exp(F*(1/fs))
 %K_gain = (PMINUS*H')*inv(H*PMINUS*H' + R_pred);
%x_est = x_est_prior_k + K_gain*(z_k - H*x_est_prior_k);
%else
%oo=zeros(6,12);
%w0 = 281.25*pi/180;
%dt = 1/fs;
%oo(9,9)=-w0*sin(w0*dt);
%oo(12,12)=Acc_Process_Noise_fz;
%A = [OM_hat Fo R_b_l o_hat; o_hat OM o_hat -R_b_l; oo];
%PHI = expm(A*dt);
%K_gain = (PMINUS*H')*inv(H*PMINUS*H' + R);
%Ref_gyro = [0.0001;0.0001;281.25*pi/180];
%z_k = rand(2,1)*0.0001;
%x_est = x_est_prior_k + K_gain*(z_k - H*x_est_prior_k);
%end
% Results
Azimuth_deg = (180/pi).*Azimuth;
Azimuth_Deg_Err = (180/pi).*errors;
time_axis_err = [0:1/fs:(length(Azimuth_Deg_Err)-1)/fs];
% Determine converged value for statistics
```

```
STD_of_Azimuth_Estimated_Error_Deg = (180/pi).*sqrt(COVAR);

end_of_array = length(Azimuth_Deg_Err);

converged_value = Azimuth_Deg_Err(end_of_array);

plot(time_axis_err,Azimuth_Deg_Err)

title('Estimated Azimuth Error vs. Time')

xlabel('Time [seconds]');

ylabel('Estimated Azimuth Error [degrees]');

grid on
```

Appendix D: MCU LPC11u24 code

```
//Wireless navigation module code
//Utilized API mode
#include "mbed.h"
#include "stdio.h"
#include "xbee.h"
#include "SDFileSystem.h"
//Set up spi interface on pins 5,6,7
SPI imu(p5, p6, p7);
//Use pin 8 as chip select
DigitalOut cs(p8);
// initalise Xbee_lib
xbee XbeeA(p9,p10,p27);
//Set SPI interface between SD card and IMU
//p11 MOSI, p12 MISO, p13 SCLK, p14 CS
SDFileSystem sd(p11,p12,p13,p14, "sd");
//USB interface to host terminal
Serial pc(USBTX, USBRX);
int main(){
//Open an file in SD card
  FILE *File = fopen("/sd/sdfile.txt","w");
  char send_data_Robin[202];     //Xbee buffer size is 202 bytes
  unsigned short buffer[12];
```

```
 int16_t data[6];
 float x_gyro, y_gyro, z_gyro,x_accl, y_accl, z_accl;
 //char read_data_Robin[202];     //Xbee buffer size is 202 bytes
  cs=1;
//SPI model: 16 bit data, mode 3
  imu.format(16,3);
  imu.frequency(1000000);
  wait_ms(500);
  cs=0;
  imu.write(0x8000);     //1MHz clock rate
  cs =1;
  while(1) {
     cs=0;
     buffer[0]=imu.write(0x1000);     // request X_GRYO_LOW, ignore returned result
     cs = 1;
     wait_us(2);    //Stall time is 2 us
     cs = 0;         //pull cs signal high to avoid the noise
     buffer[0]=imu.write(0x1200);     // request X_GRYO_OUT, read in X_GRYO_LOW
     cs = 1;
     wait_us(2);    //stall time is 2us
     cs = 0;
     buffer[1]=imu.write(0x1400);     // request Y_GRYO_LOW, read in X_GRYO_OUT
     cs = 1;
```

```
wait_us(2);    //might not be required, stall time is 2us
cs = 0;
buffer[2]=imu.write(0x1600);     // request Y_GRYO_OUT, read in Y_GRYO_LOW
cs = 1;
wait_us(2);    //might not be required, stall time is 2us
cs = 0;
buffer[3]=imu.write(0x1800);     // request Z_GRYO_LOW, read in Y_GRYO_OUT
cs = 1;
wait_us(2);    //might not be required, stall time is 2us
cs = 0;
buffer[4]=imu.write(0x1A00);     // request Z_GRYO_OUT, read in Z_GRYO_LOW
cs = 1;
wait_us(2);    //might not be required, stall time is 2us
cs = 0;
buffer[5]=imu.write(0x1C00);     // request X_ACCL_LOW, read in Z_GRYO_OUT
cs = 1;
wait_us(2);    //might not be required, stall time is 2us
cs = 0;
buffer[6]=imu.write(0x1E00);     // request X_ACCL_OUT, read in X_ACCL_LOW
cs = 1;
wait_us(2);    //might not be required, stall time is 2us
cs = 0;
buffer[7]=imu.write(0x2000);     // request Y_ACCL_LOW, read in X_ACCL_OUT
```

```
  cs = 1;
  wait_us(2);    //might not be required, stall time is 2us
  cs = 0;
  buffer[8]=imu.write(0x2200);     // request Y_ACCL_OUT, read in Y_ACCL_LOW
  cs = 1;
  wait_us(2);    //Stall time is 2us
  cs = 0;
  buffer[9]=imu.write(0x2400);     // request Z_ACCL_LOW, read in Y_ACCL_OUT
  cs = 1;
  wait_us(2);    //Stall time is 2us
  cs = 0;
  buffer[10]=imu.write(0x2600);     // request Z_ACCL_OUT, read in Z_ACCL_LOW
  cs = 1;
  wait_us(2);    //Stall time is 2us
  cs = 0;
  buffer[11]=imu.write(0x2800);       //request X_MAGN_OUT, read in Z_ACCL_OUT
  cs = 1;
 //Convert gyroscope & accelerometer data to int16_t format
  data[0]=buffer[1];
  data[1]=buffer[3];
  data[2]=buffer[5];
  data[3]=buffer[7];
  data[4]=buffer[9];
```

```
        data[5]=buffer[11];
        wait(0.1);
       //LSB for gyroscope measurement is 0.02
        x_gyro= 0.02*data[0];
        y_gyro= 0.02*data[1];
        z_gyro= 0.02*data[2];
       //LSB for gyroscope measurement is 0.0008
        x_accl= 0.0008*data[3];
        y_accl= 0.0008*data[4];
        z_accl= 0.0008*data[5];
        fprintf(File, "x_gyro= %+1.2fdeg/s\t y_gyro= %+1.2fdeg/s\t z_gyro= %+1.2fdeg/s\t
x_accl= %+1.4fg\t y_accl= %+1.4fg\t z_accl=
%+1.4fg\t\n\r",x_gyro,y_gyro,z_gyro,x_accl,y_accl,z_accl);
        fclose(File);
        send_data_Robin[0]=data[0]<<8;
        send_data_Robin[1]=data[0]>>8;
        send_data_Robin[2]=data[1]<<8;
        send_data_Robin[3]=data[1]>>8;
        send_data_Robin[4]=data[2]<<8;
        send_data_Robin[5]=data[2]>>8;
        send_data_Robin[6]=data[3]<<8;
        send_data_Robin[7]=data[3]>>8;
        send_data_Robin[8]=data[4]<<8;
```

```
        send_data_Robin[9]=data[4]>>8;
        send_data_Robin[10]=data[5]<<8;
        send_data_Robin[11]=data[5]>>8;
        //Send data to receiver XBP
        XbeeA.SendData(send_data_Robin);
        //XbeeA.RecieveData(read_data_Robin,0);  read data from the Xbee not continuously
        cs=1;
        pc.printf("x_gyro= %+1.2fdeg/s\t y_gyro= %+1.2fdeg/s\t z_gyro= %+1.2fdeg/s\t x_accl=
%+1.4fg\t y_accl=%+1.4fg\t z_accl=%+1.4fg\t\n\r", x_gyro, y_gyro, z_gyro, x_accl, y_accl,
z_accl);
        }
    }
```