

2016

MEMS-based Aided Inertial Navigation System for Small Diameter Pipelines

Sahli, Hussein

Sahli, H. (2016). MEMS-based Aided Inertial Navigation System for Small Diameter Pipelines (Doctoral thesis, University of Calgary, Calgary, Canada). Retrieved from

<https://prism.ucalgary.ca>. doi:10.11575/PRISM/26878

<http://hdl.handle.net/11023/2981>

Downloaded from PRISM Repository, University of Calgary

UNIVERSITY OF CALGARY

MEMS-based Aided Inertial Navigation System for Small Diameter Pipelines

by

Hussein Sahli

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF DOCTOR OF PHILOSOPHY

GRADUATE PROGRAM IN GEOMATICS ENGINEERING

CALGARY, ALBERTA

APRIL, 2016

© Hussein Sahli 2016

Abstract

Pipeline Inspection Gauges (pigs) have been used for many years to perform various maintenance operations in oil and gas pipelines. Different pipeline parameters can be inspected during the pig journey. Although, pigs uses many sensors to detect the required pipeline parameters, matching these data with the corresponding pipeline location is considered a very important parameter that needs to be estimated.

High-end, tactical-grade Inertial Measurement Units (IMUs) are used in pigging applications to locate the detected problems of pipeline using other sensors, and to reconstruct the trajectories of the pig. These IMUs are accurate; however, their high cost and large sizes limit their use in small diameter pipelines.

Calibration would improve the accuracy of the uncertainties that exist in sensor errors behavior. However, intensive calibration would also increase the cost of using IMUs. Therefore, another way to improve the accuracy is used by augmenting IMU with aided sensors (i.e. odometers).

This thesis describes a new methodology for the use of low-cost IMUs using an extended Kalman filter (EKF) and the pipeline junctions to increase the navigation parameters' accuracy and to reduce the total RMS errors even during the unavailability of Above Ground Markers (AGMs).

The results of this new proposed method using micro-electro-mechanical systems (MEMS) based IMU revealed that the position RMS errors were reduced by approximately 85% of the standard EKF solution. Therefore, this approach will enable the mapping of small diameter pipelines, which was not possible before.

Acknowledgements

This work is the result of a Collaborative Research and Development Grant between NSERC (the Natural Sciences and Engineering Research Council of Canada), TECTERRA, and the University of Calgary.

The inertial measurement unit used (SiIMU02) was provided by ROSEN Inc. They are thankfully collected the required data for this thesis.

I would like to express my sincere thanks to my advisor, Dr. Naser El-Sheimy for his generous advice, inspiring guidance and encouragement throughout my graduate studies, research, and dissertation work. His advice on both research as well as on my career have been priceless. This thesis could not have been completed without his generous and professional assistance.

I also appreciate my supervisors committee (Dr. Nouredin and Dr. Yang Gao) and my examiner committee (Dr. Mohammed Tarbouchi and Dr. Mamdouh El-Badry) for carefully reading and providing comments concerning various aspects of this research.

I should say that I am very happy that I have been involved in a project that force me to dig into different fields and gain valuable knowledge.

I would like to thank my colleagues at MMSS research group for their help throughout my thesis.

A special thanks to Dr. Adel Moussa, for his support throughout my study.

Finally, and most importantly, I would like to thank my wife Rawad. Her support, encouragement, quiet patience and unwavering love were undeniably the bedrock upon which the past thirteen years of my life have been built. Her tolerance of my occasional vulgar moods is a testament in itself of her unyielding devotion and love. I appreciate my beloved daughters Sama and Sana for abiding my ignorance and the patience they showed during my thesis writing. Words would never

say how grateful I am to both of you. I consider myself the luckiest in the world to have such a lovely and caring family standing beside me with their love and unconditional support. I thank my parents, Ahmed and Rowaida, for their faith in me and allowing me to be as ambitious as I wanted. It was under their watchful eye that I gained so much drive and an ability to tackle challenges head on.

Dedication

TO

MY LOVING PARENTS, MY SWEET WIFE, MY BELOVED

DAUGHTERS,

AND MY DEAR BROTHERS AND SISTERS

"To All Of You, I Shall Be Indebted Forever"

Table of Contents

Abstract.....	ii
Acknowledgements.....	iii
Dedication.....	v
Table of Contents.....	vi
List of Tables.....	x
List of Figures and Illustrations.....	xi
List of Symbols, Abbreviations and Nomenclature.....	xiv
CHAPTER ONE : INTRODUCTION.....	1
1.1 Background and Problem Statement.....	1
1.2 Current Technology.....	6
1.3 Pipeline Pigging.....	7
1.4 Mapping Smart Pigs.....	8
1.5 Objectives.....	11
1.6 Thesis Structure.....	12
CHAPTER TWO : FUNDAMENTALS OF INERTIAL NAVIGATION.....	14
2.1 Introduction.....	14
2.2 Navigation System Frames.....	17
2.2.1 Earth-Centered Inertial Frame (i-frame).....	17
2.2.2 Earth-Centered Earth-Fixed Frame (e-frame).....	17
2.2.3 Local Level Frame (l-frame).....	21
2.2.4 The Computational Frame (c-frame).....	22
2.2.5 The Platform Frame (p-frame).....	23
2.2.6 The Body Frame (b-frame).....	24
2.3 Attitude Representation.....	24
2.3.1 Direction Cosine Matrix (DCM).....	25
2.3.2 Euler Angles.....	27
2.3.3 Attitude Quaternion.....	30
2.4 INS Mechanization Algorithm.....	33
2.4.1 Measurement Data.....	33
2.4.2 Updating Rotation Matrix and Quaternion.....	38

2.4.3 Attitude Update	40
2.4.4 Velocity Update.....	40
2.4.5 Position Update	41
2.5 Aiding Sensors for INS.....	42
2.5.1 Global Navigation Satellite System (GNSS).....	42
2.5.2 Distance Measurement System.....	43
2.5.3 Other Aiding Sources	44
CHAPTER THREE : CALIBRATION OF INERTIAL MEASUREMENT UNIT	45
3.1 Introduction.....	45
3.2 Inertial Sensor Errors	46
3.2.1 Bias Drift	46
3.2.2 Scale Factor Instability	49
3.2.3 Random Walk.....	50
3.3 Mathematical Models of INS Errors.....	50
3.3.1 Accelerometer Model Errors	51
3.3.2 Gyroscope Model Errors	52
3.4 INS Calibration Methods	53
3.4.1 Six-Position Static Acceleration Test.....	53
3.4.2 Local Level Frame Calibration.....	54
3.4.3 Angle Rate Test	54
3.4.4 Allan Variance Analysis.....	55
3.5 Calibration Results.....	57
3.6 MEMS-INS Alignment.....	59
CHAPTER FOUR : ESTIMATION IN NAVIGATION.....	61
4.1 Introduction.....	61
4.2 Kalman Filtering	63
4.3 INS Error Models Review	66
4.3.1 Psi-Angle Error Model	68
4.3.2 Phi-Angle Error Model.....	72
4.4 Random Processes	73
4.4.1 Sensor Errors	74
4.4.2 Stochastic Process	75

White Noise Process	76
Random Constant Process	77
Random Walk Process	77
Gauss-Markov (GM) Process	79
4.5 Backward Smoothing.....	81
CHAPTER FIVE : PIPELINE NAVIGATION USING CONSTRAINTS	83
5.1 Non-Holonomic Constraints	83
5.2 Pipeline Junctions	84
5.3 Detecting Pipeline Junctions.....	87
5.3.1 Moving Average & Convolution.....	87
5.3.2 Wavelet Transformation.....	88
5.4 Bend Detection Algorithm (BDA).....	91
5.4.1 Angular Velocity Magnitude	92
5.4.2 Mean of Heading & Pitch Angles History	92
5.5 Methodology.....	94
5.5.1 Dynamic Error Model.....	95
5.5.2 Pitch & Heading Measurement Model.....	99
CHAPTER SIX : EXPERIMENTAL RESULTS	104
6.1 Experiment Description (MEMS).....	104
6.1.1 Simulated AGM.....	107
6.1.2 Adding time to original reference trajectory	108
6.2 Forward-Backward Smoothing (MEMS)	109
6.2.1 Scenario #1:	110
6.2.2 Scenario #2:	114
6.3 PLJ/EKF Results (MEMS)	116
6.3.1 Scenario #1	118
6.3.2 Scenario #2	122
6.4 PLJ/EKF Results (LN200).....	126
CHAPTER SEVEN : CONCLUSION AND RECOMMENDATION	131
7.1 Research Contribution	132
7.2 Conclusion	132

7.3 Recommendations for Further Research.....	135
References.....	136
APPENDIX: Pipeline Pigging	141

List of Tables

Table 2-1: Classification of IMU	16
Table 2-2: Quaternion from DCM	32
Table 3-1: IMU Grade Bias Specifications.....	45
Table 3-2: SiIMU02 Calibrated Parameters	57
Table 6-1: North, East & Height Errors - Filtered	110
Table 6-2: North, East & Height Errors – Smoothed.....	110
Table 6-3: MEMS – IMU Specifications (SiIMU02).....	117
Table 6-4: North, East & Height Errors.....	121
Table 6-5: North, East & Height Errors.....	122
Table 6-6: LN200s – IMU Specifications.....	126
Table 6-7: LN200 - Errors	129

List of Figures and Illustrations

Figure 1-1: Smart Pipeline Inspection Gauge (Nord Stream AG).....	3
Figure 1-2: IMUs' Comparison.....	5
Figure 1-3: Simplified Position Algorithm.....	5
Figure 1-4: Smart pig (GEreports.ca).....	10
Figure 2-1: Reference Frames.....	18
Figure 2-2: Geodetic Coordinates.....	20
Figure 2-3: Relationship between true navigation, computer, and platform frames.....	23
Figure 2-4: Rotation Frame.....	25
Figure 2-5: IMU Mechanization Process Block Diagram (Noureldin et al., 2012).....	37
Figure 3-1: Systematic IMU Errors.....	48
Figure 3-2: Bias Drift.....	49
Figure 3-3: Scale Factor.....	50
Figure 3-4: IMU Calibration - Rotation Table.....	55
Figure 3-5: Allan Variance of SiIMU02 Gyroscopes.....	58
Figure 3-6: Allan Variance of SiIMU02 Accelerometers.....	59
Figure 4-1: Block Diagrams (LKF vs EKF).....	63
Figure 4-2: Kalman Filter Algorithm Flow Diagram.....	65
Figure 4-3: Loosely Coupled Architecture using EKF.....	66
Figure 4-4: ACF of GM Process.....	80
Figure 4-5: Block Diagram of First Order GM Process.....	80
Figure 5-1: Pipeline Joints.....	85
Figure 5-2: Accelerometer output - MEMS (SiIMU02).....	86
Figure 5-3: Accelerometer output - LN200.....	86
Figure 5-4: Pipeline Junctions – Sample.....	87

Figure 5-5: Junctions Detection Using Moving Average	88
Figure 5-6: Mexican Hat Mother Wavelet.....	89
Figure 5-7: Pipeline Bends & Junctions	90
Figure 5-8: Detected Pattern Using WL	90
Figure 5-9: Pig Defined Axes	93
Figure 5-10: Selecting Threshold Criteria – Angular Velocity	94
Figure 5-11: Algorithm for corrected state estimation	95
Figure 6-1: Pig Trajectory.....	105
Figure 6-2: Accelerometers Output	106
Figure 6-3: Gyroscopes Output.....	107
Figure 6-4: Detected Junctions	108
Figure 6-5: Add Time to Original Reference Trajectory	109
Figure 6-6: Full Trajectory (Scenario #1).....	111
Figure 6-7: NED Error - Filtering (Scenario #1)	112
Figure 6-8: NED Error - Smoothing (Scenario #1)	112
Figure 6-9: Roll, Pitch and Azimuth - Smoothing (Scenario #1)	113
Figure 6-10: Full Trajectory (Scenario #2).....	114
Figure 6-11: NED Error - Filtering (Scenario #2)	115
Figure 6-12: NED Error - Smoothing (Scenario #2)	115
Figure 6-13: Trajectories - Scenario #1	118
Figure 6-14: Positions RMS Errors - EKF Scenario #1.....	119
Figure 6-15: Positions RMS Errors - EKF/PLJ - Scenario #1	119
Figure 6-16: Maximum Position Errors - Scenario #1.....	120
Figure 6-17: RMS Positions Errors - Scenario #1	120
Figure 6-18: Trajectories - Scenario #2	123

Figure 6-19: Positions RMS Errors - EKF Scenario #2.....	124
Figure 6-20: Positions RMS Errors - EKF/PLJ Scenario #2	124
Figure 6-21: Maximum Position Error - Scenario #2	125
Figure 6-22: RMS Position Errors - Scenario #2.....	125
Figure 6-23: Full Trajectory - LN200.....	127
Figure 6-24: Trajectory - Zoomed area from Figure 6-23	128
Figure 6-25: The effect of EKF/PLJ algorithm.....	130
Figure 7-1: Pipelines in North America. Total length approximation 2,000,000 km.....	143
Figure 7-2: Launcher Receiver Design	145
Figure 7-3: Ball Valves (Wint)	146
Figure 7-4: Gate Valves (Wint)	146

List of Symbols, Abbreviations and Nomenclature

The notations used in the dissertation will be introduced here.

1. Conventions

- a. Matrices are denoted in upper case letters.
- b. Vectors are denoted in lower case letters.
- c. The coordinate frames that are involved in vector transformations are denoted as superscripts and subscripts. As an example, the symbol (R_b^l) is the direction cosine matrix from the body frame (b) to the local-level frame (l).
- d. For angular rate vectors, two subscripts denote the reference and target frames, and superscripts denote the projected frame. As an example, the angular rate vector of the body frame with respect to the inertial frame projected in to the navigation frame is written as (ω_{ib}^l) .

2. Symbols

Symbol	Definition
$(.)^{-1}$: Matrix and quaternion inverse
$(.)^T$: Matrix transpose
$(.\times)$: Cross-product or skew-symmetric form of a vector
$\Omega_{\alpha\beta}^\gamma$: Skew-symmetric matrices corresponding to $(\omega_{\alpha\beta}^\gamma)$
g^l	: Gravity in local level frame

v^n, v^e, v^d : Velocities in north, east and down, respectively.

\times : Cross product

\star : Quaternion product

Ψ : The skew-symmetric matrix of the rotation vector pertaining to the platform tilt error ψ

$E[.]$: Expectation of $[.]$

$diag(.)$: Diagonal matrix form of a vector

θ, ϕ, ψ : Pitch, roll, and heading

φ, λ, h : Latitude, Longitude, and height, respectively.

ψ : The platform tilt error

ω : Angular rate vector

3. Acronyms

AGM: Above Ground Marker

BDA: Bending Detection Algorithm

CUPT: Coordinate Update

DCM: Direct Cosine Matrix

DGPS: Differential GPS

ECEF: Earth Center Earth Fixed Frame

EKF: Extended Kalman Filter

FOG: Fiber Optic Gyro

GCP: Ground control point

GNSS: Global Navigation Satellite System

GPS: Global Positioning System

ILI: Inline Inspection Tool

IMP: Integrity Management Process

IMS: Integrity Management System

IMU: Inertial Measurement Unit

INS: Inertial Navigation System

LLF: Local Level Frame

MEMS: Micro-Electro-Mechanical-Systems

NED: North, East and Down

NEP: National Energy Program

Fig: Pipeline Inspection Gauge

PJC: Pipeline Junction Constraint

PLJ: Pipeline Junctions

PMM: Pipeline maintenance management

RLG: Ring Laser Gyro

RMSE Root Mean Square Error

RTS: Rauch–Tung–Striebel

STD: Standard Deviation

Chapter One : **INTRODUCTION**

Advances in Micro-Electro-Mechanical-Systems (MEMS) technology combined with the miniaturization of electronics, have made it possible to produce low cost and lightweight chip-based inertial sensors. These chips are small, lightweight, consumes very little power, and reliable. It has therefore found a wide spectrum of applications in the automotive and other industrial applications. MEMS technology, therefore, can be used to develop navigation systems that are inexpensive, small, and consume low power (microwatt). The attractive advantages of this MEMS technology have led to remarkable research progress in the field of MEMS inertial sensors. However, on the negative side, the current achieved performance by these low cost sensors is relatively poor due to their sensor errors.

The goal of this chapter is henceforth to present background information on the evolution of MEMS technology and its increased adoption in the inertial navigation field. The main advantages of MEMS inertial sensors as well as the basic limitations and shortcomings, that significantly degrade their performance, are presented. The problem of optimizing the estimation procedure of the MEMS-INS in pipeline navigation is stated. New approach to enhance MEMS sensors performance in pipeline navigation applications is proposed. Finally, research objectives, research contribution, and the dissertation outline are given.

1.1 Background and Problem Statement

Pipelines are the lifelines of a dynamic country's infrastructure; it provides fuel, water, and all other needs that touch millions of lives. In addition, it is one of the safest transportation for cured oil, natural gas, and chemical fluid; it provides a transporting with speed, efficiency and reliability.

Furthermore, it is considered as an eco-friendly option. Canada has a large network of pipelines – over 830,000 km – but they are not the same. Different types of pipelines are used for oil and gas transportation, and each varies in its size and function. Gathering pipelines move oil and gas from the source to the processing facilities. More than 250,000 kilometers of these lines are concentrated in the producing provinces of Western Canada, primarily in Alberta. The size of gathering pipelines varies from 4 to 12 inch. Feeder pipelines move the product to transmission pipelines. There are more than 25,000 kilometers of feeder pipelines in the producing areas of Western Canada. Transmission pipelines carry oil and gas across Canada. There are approximately 117,000 kilometers of transmission lines in Canada. The size of transmission pipelines are varies from 4 to 48 inch. About half are 18 inch or larger, and about one third are 10 inch or smaller. Finally, the distribution pipelines are used to get natural gas to the customer. The size of these pipelines are varies from half to 6 inch. There are about 450,000 kilometers of distribution pipelines lines in Canada.

Pipeline inspection gauge (pig) is a cylindrical device that is inserted into a pipeline to clean the pipeline wall or monitor the internal condition of the pipeline (also called a smart pig). Smart pigs commonly carries out inspection and fault identification of these pipelines Figure 1-1.

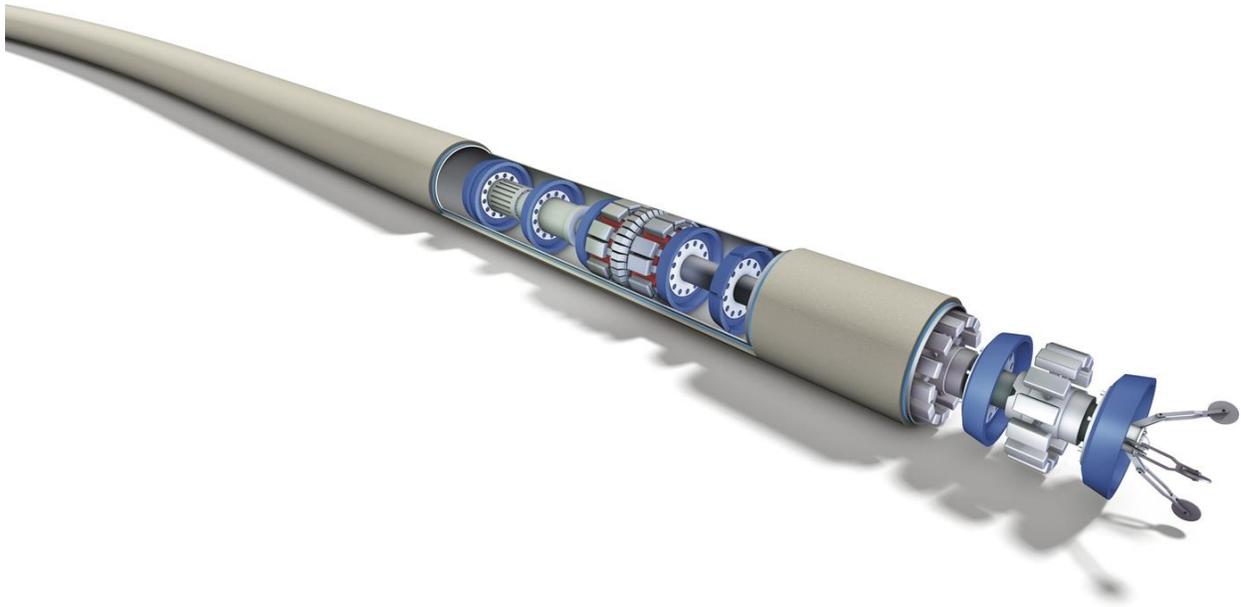


Figure 1-1: Smart Pipeline Inspection Gauge (Nord Stream AG)

Variety of methods are used for pipeline inspection such as ultrasonic techniques, echo sounding, radiography, and cameras. In the past, the position determination of the pig used to be obtained by a set of velocity wheels (odometers). These wheels provides the longitudinal speed of the pig that can be integrated to provide the distance traveled along the pipeline. Fiber Optic Gyro (FOG) and/or Ring Laser Gyro (RLG) based high-end inertial navigation systems has been proposed to be included in the positioning solution (Hanna, 1990), which is currently being used for this purpose in the large diameter inspection tools.

This thesis addresses the issue of providing a low cost, MEMS, aided inertial navigation system for an industrial tool called smart pipeline inspection gauge (pig). The definition of the terms “low cost”, “MEMS” , and “aided inertial navigation” are as follows:

1. Low cost system symbolizes the ability to provide a solution that can be implemented cost effectively by the civilian sector. In our case, we are focusing on low cost navigation systems.
2. Micro-Electro-Mechanical Systems, or MEMS, is a technology that in its most general form can be defined as reduced mechanical and electro-mechanical elements. These elements are built using microfabrication techniques. Perhaps the most interesting elements these days are the micro-sensors and micro-actuators.
3. Inertial Navigation is the implementation of inertial sensors to determine the position and orientation of a motion body. Since the current evaluation of the states of the body motion is formed by the relative increment from the previous known states, hence, inertial sensors are assorted as dead reckoning sensors.

Low cost inertial sensors are characterized by large uncertainties and high noise (i.e. bias, scale factor and non-orthogonality). Therefore, the errors in position, velocity and attitude of the motion body grow rapidly in standalone mode. Therefore, aiding navigation systems become essential for low cost INS to bed the unpredictable problems of sensor errors and noises [1]. In this thesis, the aided information will be derived mainly from odometer sensor, pipeline modeling and Global Navigation Satellite System (GNSS).



Figure 1-2: IMUs' Comparison

From basic of physics, change in position can be calculated by integrating the acceleration twice as shown in Figure 1-3. Similarly, acceleration errors propagate through this double integration.

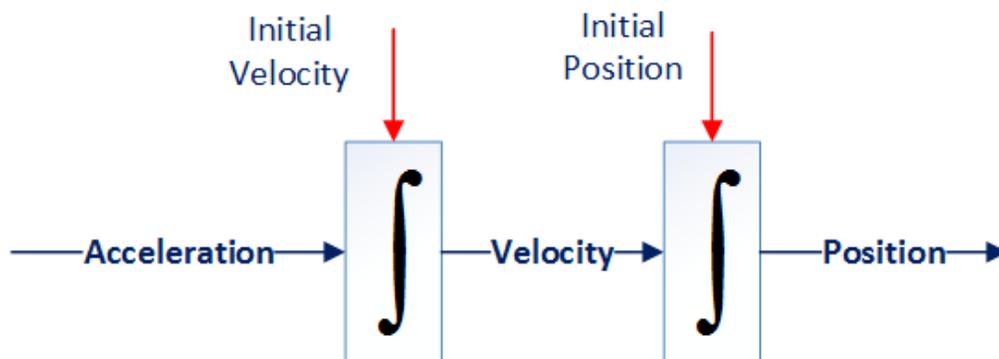


Figure 1-3: Simplified Position Algorithm

An incorrect projection of the acceleration signals onto any defined axes can be caused by an error in the orientation. This causes several problems; firstly, the acceleration of the motion body is integrated in wrong direction. Secondly, gravity cannot be removed correctly in the calculation. Therefore, errors in the angular velocity signals also cause drift in the calculated position. As an example, for gyro drift $6^\circ/hr$, the positional error accumulation after $10\ min$ will be around $10500\ m$.

1.2 Current Technology

The current small diameter pigs use odometers to record the distance travelled during their operations. These distance measurements are then converted into terrestrial coordinates using pipeline maps and external coordinate points.

Despite its simplicity, this standard method has serious limitations. Some small diameter pipelines do not have any maps. For such pipelines, frequent control point surveys are needed which could be very expensive and/or difficult. Furthermore, even if a map exists, these maps may not be sufficiently accurate due to the movement of the pipelines (caused by, for example, underground subsidence or sea bed currents in marine environments) (Hanna, 1990). The errors on the existing maps cause much larger areas to be excavated for maintenance that significantly increases the overall maintenance costs.

In addition to periodical pipeline inspection and maintenance, there are situations for which inline tools are needed to generate new maps or validate existing maps for the pipelines. However, the small diameter smart pig cannot be used for this purpose at all. Currently, the only mapping

solution for small diameter pipelines involves external pipeline survey, which can be extremely costly.

1.3 Pipeline Pigging

An old concept has been used in pipeline sector called "*pipeline pigging*". What do we mean by pipeline pigging? pigs are devices / tools that can be inserted into the pipeline and travel throughout the length of the pipeline, driven forward by the differential pressure across the tool. Mainly, they are divided to two categories, 'Utility pig' that is used for separating and cleaning, and the 'Smart pig' (sometimes called as 'ILI tool', or 'intelligent pig') that provide detailed information of the pipeline (ex. corrosion, thickness ...etc.) and the location of all detected problems (Products & Association, 1995).

Pigging is an operation that needs to be taken seriously. There are often problems that need to be solved with careful control and coordination. There is always a risk that an outlandish body introduced in the pipeline, which will block the flow. Such an incident requires the pipeline to be cut out with all operational expenses. Therefore, the pipeline operator should consider the pipeline need to be pigged, whether it is suitable for pigging, and whether it is economic to do so.

The name pig was originally released to Go-Devil scrapers. These scrapers used to wax of the internal wall of the pipeline by inserting them in the pipeline and drive them by the flowing fluid trailing spring-loaded rakes. The rakes made a characteristic loud squealing noise, hence the name "pig". Nowadays, the term pig is used to describe any device made to pass through a pipeline driven by the pipeline fluid.

Pipelines were mainly pigged to remove paraffin to increase the efficiency in crude oil pipelines that will maximize flow conditions for the war effort in 1940s in the United States. The pigging equipment utilized at that time was limited to a few applications while being very crude in nature. In today's world, pipelines are pigged for a variety of reasons and the pigging equipment used is designed by engineers to perform particular functions, such as (Tiratsoo, 1992):

1. Separation of products.
2. Gauging the internal bore.
3. Locating the obstructions.
4. Measure pipeline geometry.
5. Internal inspection.
6. Internal coating.
7. Corrosion inhibition.

1.4 Mapping Smart Pigs

Smart pig tool (shown in Figure 1-4); sometimes call Inline Inspection (ILI) tool; falls in four basic types (Kuprewicz, 2005):

1. General metal loss.
2. Dimensional.
3. Mapping.
4. Specialty.

Increasingly, regulations demand that pipeline operators document the precise location of pipeline assets. In some cases, however, records are old and of unknown accuracy, or may not include

details of centerline location. Therefore, in this thesis, we are interested in the third type (mapping). The mapping smart pig tool is a geo-positioning or pipeline position tool. This tool uses inertial navigation system (INS) along with distance measurement wheels to position or locate the pipeline. Moreover, above ground markers are used to correct or update the navigation estimation algorithm due to the errors in the inertial navigation system. AGM is a portable or permanently installed device placed on the surface above a pipeline that both detects and records the passage of an in-line inspection tool or transmits a signal that is detected and recorded by the tool.

Overview of Current Smart Pigs

Due to unavailability of GNSS signals inside the pipelines, IMUs with fiber optic gyroscopes (FOG) are used to conduct the overall pig navigation journey. Aiding sensors are used to compensate the growing errors in the stand-alone INS. Ground control point (GCP) are used as coordinate update (CUPT). These control points are available along the pipeline several kilometers apart. Usually, these GCP located at pipeline features (ex. valves) or at AGMs of which the geodetic positions are precisely surveyed by DGPS (Shin & El-Sheimy, 2005). The accuracy of CUPTs is affected by the uncertain lever-arm effect due to the time synchronization issue between the IMU and AGMs, therefore, pipeline trajectories are forced to fit to these points (Liu, 2009). The control point's tracking module detects the magnetic signal of the pig and store the time when the tool passes underneath.

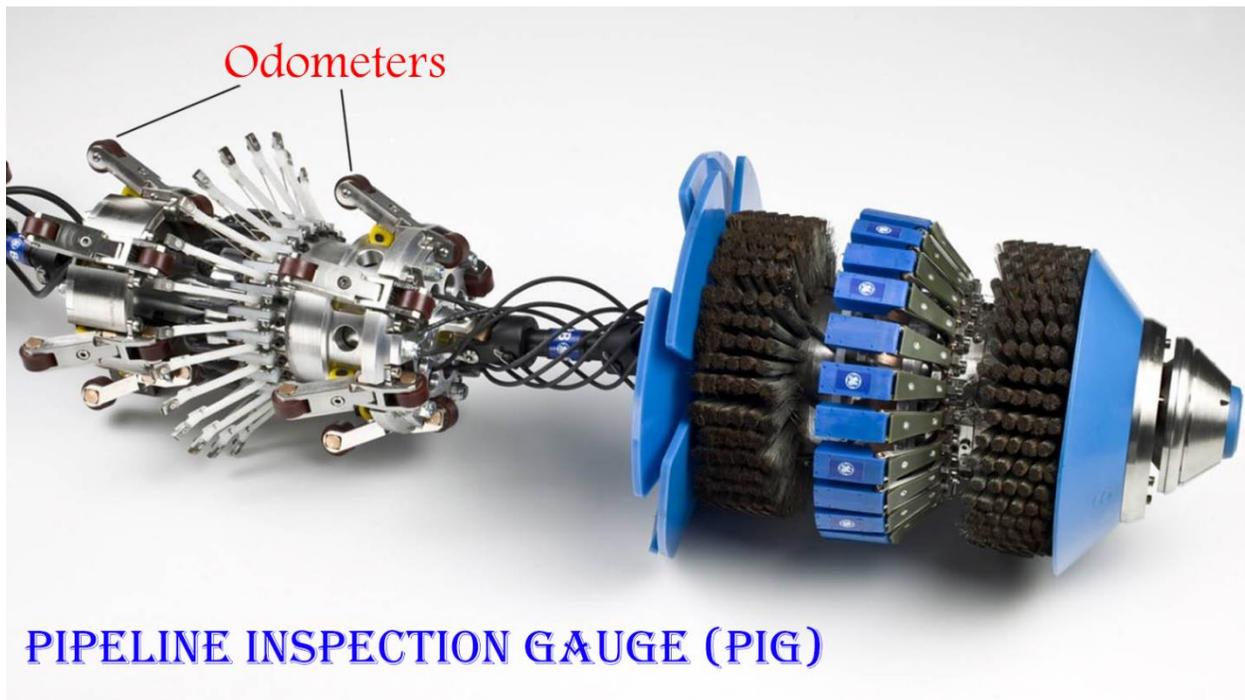


Figure 1-4: Smart pig (GEreports.ca)

Forward distance can be measured using the distance measurement wheels (Odometer). It can provide the forward velocity information by differentiating the distance travelled by the time.

It is worth to highlight that the pig is pushed in and driven through the pipeline using the differential pressure of the medium flow. This may cause some unexpected speed deviation due to mechanical failure or residue on the pipe wall. Moreover, the uncertainty in the wheel measurement can be increased due to the vibration and the varying contacts between the odometer wheels and the pipeline wall.

At the beginning of the pig's surveying journey, it sits in the launcher for sometimes that will be required to calculate the initial alignment of the tool during the post-process calculation. Similar

scenario can be repeated at the end of the journey (receiver). More details about pipeline pigging process are expanded in the appendix.

1.5 Objectives

In summary, the main goal of this thesis is to provide an aided inertial navigation system for small diameter pipelines (less than 8”) which can be used effectively by the pipeline inspection sector and provide an adequate level of integrity not to expose the safety of the overall system. Every pig requires a navigation system to associate its inspection data to a known terrestrial coordinate frame. Due to IMU size (as shown in Figure 1-2) for small diameter pipelines, FOG/RLG IMU systems cannot be employed in the small diameter pigs.

The overall objective is to improve the positioning accuracy for small diameter pigging applications. Therefore, the objective of this thesis can be summarized as follows:

1. Due to the large size of tactical grade IMUs, it is impossible to include them in the current small size diameter’s pig. Current range of small pigs depends on odometers to navigate the positions (Sahli, 2014). Therefore, the main objective of this thesis is to improve the small diameter pig’s positioning capability by including MEMS-based IMU
2. Developing new methodology for aiding the IMU navigation results using pipeline physical characteristics.
3. Developing the methodology for detecting pipeline junctions and bends based on using inertial measurements.
4. Integrating the newly developed pipeline physical characteristics in the navigation filter

As a results of these developments, the overall cost of establishing AGMs will be reduced (by reducing the total number of AGM).

After each operation of the pig, the 3D solution of the INS can be used to map the existing pipeline. As every run of the pig provides an independent mapping solution for the entire pipeline, it will also be possible to improve/verify the existing maps with each pig inspection mission without incurring any additional cost at all.

1.6 Thesis Structure

The thesis is structured in seven chapters and includes a general conclusion and a list of references.

The status of present smart pipeline inspection gauge (pig) technology is documented in Chapter One. This chapter reviews the pipeline integrity system, pipeline pigging technology, history of pipeline pigs, and mapping using smart pigs. More details about pipeline industries are included in the Appendix.

Chapter Two gives the essential building blocks for an aided inertial navigation system. Various reference frames used in navigation field are introduced. Attitude representations used in an INS will be introduced. INS mechanization algorithm will be presented. Finally, mathematical models for aiding sensors will be discussed.

Chapter Three discusses the inertial sensor errors with their mathematical models, required calibration methods, and alignment process for MEMS-based IMUs.

In Chapter Four, the extended Kalman filter (EKF) and the RTS smoother are reviewed. INS error models and the random process modeling.

New methodology for pipeline navigation using pipeline junctions has been introduced in Chapter Five

In Chapter Six, the performance of the PLJ/EKF and EKF will be compared using MEMS IMUs.

Finally, Chapter Seven concludes the research work and provides the contributions and recommendations for future development of the suggested techniques.

Chapter Two : FUNDAMENTALS OF INERTIAL NAVIGATION

Navigation comprises the methods and technologies to determine the time varying position and attitude of a moving object. The three main navigation states presented as time variable functions are position, velocity and attitude. These states contain all necessary navigation information needed to geo-reference a moving object at any moment of time. Positioning is a term that used when the position state of the moving object is only required. In general, positioning is used in place of navigation since most modern navigators actually position a platform in a discrete time. Navigation information can be generated by joining multiple discrete periods. Inertial measurement unit (IMU) are also provide discrete updates. These updates are relative and mathematical integration are required in order to provide positioning in reference frame to a known starting point.

Background information related to the fundamentals of inertial navigation including inertial navigation mechanization models will be provided in this chapter. This information will serve the basis for the development and analysis in later chapters.

2.1 Introduction

Inertia can be defined as a property of matter that causes it to resist changes in velocity. Since an object with a given velocity maintains that velocity unless acted on by an external force; Newton's first law¹ of motion; therefore, inertia is the property of matter that makes this law hold true.

¹ Newton's 1st law states: "an object at rest tends to stay at rest and an object in motion tends to stay in motion with the same speed and in the same direction unless acted upon by an unbalanced force" Lawrence, A. (1993). *Modern inertial technology : navigation, guidance, and control*. New York: Springer.

Inertial navigation is the determination of the motion body position using inertial sensors. The external forces generate accelerations on the body. By measuring these accelerations then integrating them mathematically with respect to the time, the changes in the body's velocities (x, y and z directions) and positions can be determined with respect to the initial conditions.

Inertial sensors can be divided into two categories, accelerometers and gyroscopes. Accelerometers are the sensors that measure the acceleration of the body due to the gravity and all other accelerations due to the external forces. Accelerometers attempt to measure the unbalanced force through the application of Newton's second law². In order to remove the gravity component of the acceleration, the attitude (or tilt) of the accelerometers with respect to the local vertical need to be known. Gyroscopes are the sensors that measure the angular rotation of the moving body. The mathematical integration of the angular rotation with respect to time provides the angular change with respect to initial known angle.

Inertial Measurement Unit (IMU) is a device that contain both inertial sensors (accelerometers & gyroscopes). Using IMU, the determination of the position of the vehicle will be possible. IMUs have been divided into different categories based on their characterization as shown in Table 2-1. IMU that contains three orthogonal accelerometers and three orthogonal gyroscopes called tri-axial IMU.

² Newton's 2nd law states: "the acceleration of an object as produced by a net force is directly proportional to the magnitude of the net force, in the same direction as the net force, and inversely proportional to the mass of the object" *ibid.*

Table 2-1: Classification of IMU

Performance	Unit	Strategic Grade	Navigation Grade	Tactical Grade	Commercial Grade
Gyroscope Drift	\circ/h	$10^{-4} - 10^{-3}$	< 0.01	$1 - 10$	> 10
Gyroscope Random Walk	\circ/\sqrt{h}	$(0.1 - 5)10^{-4}$	< 0.002	$0.05 - 0.2$	Large Variation
Accelerometer Bias	μg	$0.1 - 1$	< 100	$(1 - 4)10^3$	> 5000
Approximate Cost	\$	$> 10^5$	$(0.5 - 1)10^5$	$(5 - 20)10^3$	< 2000
Applications		Submarines Intercontinental ballistic missile	General navigation High precision, Geo-referencing Mapping	Integrated with GPS for mapping, Weapons (short time))	Low cost navigation, Active suspension, Airbags

2.2 Navigation System Frames

Different reference frames are involved in INS algorithms development. This is because a representation of one vector in one frame must frequently be transformed into another. As an example, both angular rates and specific forces (as outputs of the IMU measurements) are measured in the body frame, in order to calculate the navigation states, these values should be transformed to navigation frame (l-frame). In this section, the properties of each reference frame will be described along with their relationship to each other.

2.2.1 Earth-Centered Inertial Frame (*i-frame*)

Inertial frame is defined to be stationary in the space or moving at a constant speed (Noureldin, Karamat, & Georgy, 2012). The i-frame has its origin at the center of the earth as shown in Figure 2-1. It is important to note that all inertial sensors measurements are produced relative to the i-frame and resolved along the instrument's sensitive axes (sensor-frame).

2.2.2 Earth-Centered Earth-Fixed Frame (*e-frame*)

ECEF-frame or simply called as Earth-frame (e-frame). This frame shares the same origin (center of the earth) with i-frame as shown in Figure 2-1. The axes are defined as i-frame. The only difference between e-frame and i-frame that e-frame is rotating with respect to i-frame around z-axis with a rotation speed of ($\omega_e = 7.2921158 \times 10^{-5} rad/s$). In this frame, the acceleration of the moving body with respect to the earth (\dot{v}^e) is equal to the acceleration of the body with respect

to the i-frame minus the Coriolis acceleration³ due to the velocity of the body (v^e) over the angular velocity of the earth rotation (ω_{ie}^e):

$$\dot{v}^e = \dot{v}_i^e - \omega_{ie}^e \times v^e \quad (2-1)$$

where $\omega_{ie}^e = [0 \quad 0 \quad \omega_e]^T$.

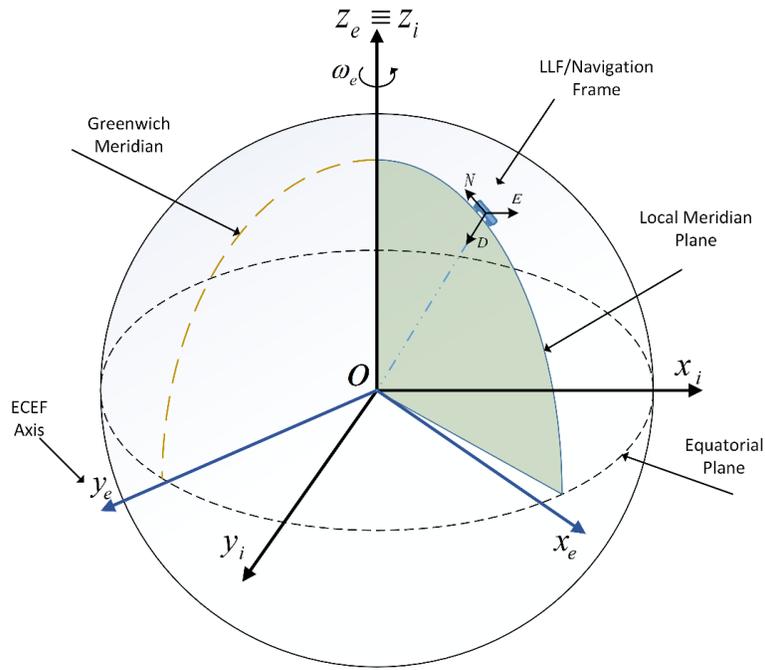


Figure 2-1: Reference Frames

³ The theorem is named after the physicist Gaspard Gustave de Coriolis (1792-1843). “It states that, for a particle moving in a Newtonian frame of reference, which is itself moving with respect to a second frame of reference, the total acceleration for the particle is the vector sum of its acceleration with respect to the first reference, the acceleration of the first to the second reference and a calculated Coriolis acceleration.”

The position vector in e-frame can be expressed in terms of geodetic latitude (φ), longitude (λ), and height (h) as follows (El-Sheimy, 2012; Noureldin et al., 2012):

$$r^e = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} (R_N + h) \cos \varphi \cos \lambda \\ (R_N + h) \cos \varphi \sin \lambda \\ (R_N(1 - e^2) + h) \sin \varphi \end{bmatrix} \quad (2-2)$$

Where e^2 is the first eccentricity ($e^2 = \frac{a^2 - b^2}{a^2}$), and (a) and (b) are the semi-major (Equatorial radius) and semi-minor (Polar radius) axes of the earth, respectively ⁴. (R_N) denotes the radius of curvature in the prime vertical (Figure 2-2).

From Eq.(2-2), the longitude (refer to section 2.2.3) can be computed using the four-quadrant inverse tangent function as follows:

$$\lambda = \tan_2^{-1}(y, x) \quad (2-3)$$

⁴ For WGS84: $e^2 = 0.006694380004260827$, $a = 6378137 \text{ m}$, $b = 6356752.3142 \text{ m}$

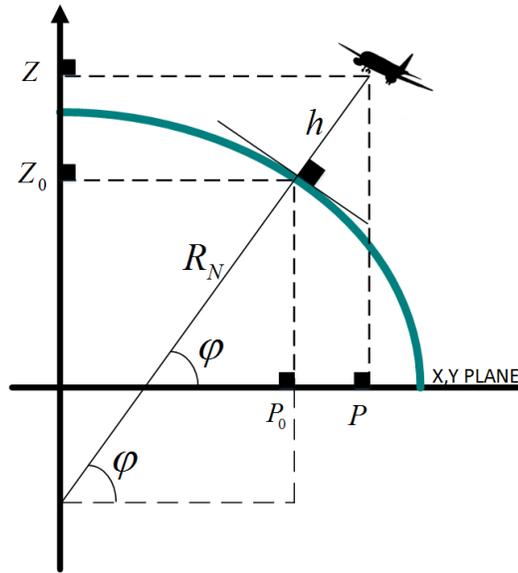


Figure 2-2: Geodetic Coordinates

To calculate the height and latitude, (Shin, 2005) more complicated algorithm will be used:

1. Initialize: $h_0 = 0$, and $\varphi_0 = \tan_2^{-1}(z, (1 - e^2)\sqrt{x^2 + y^2})$.
2. Repeat the following steps to achieve the required convergence:
 - a. Compute the radius of curvature in the prime vertical:

$$R_N = \frac{a}{\sqrt{1 - e^2 \sin^2 \varphi}} \quad (2-4)$$

- b. Update the height as:

$$h = \frac{\sqrt{x^2 + y^2}}{\cos \varphi} - R_N \quad (2-5)$$

- c. Update the latitude as:

$$\varphi = \tan_2^{-1} \left(z, \left(1 - \frac{e^2 R_N}{R_N + h} \right) \right) \quad (2-6)$$

Near the pole, Eq. (2-5), is singular. Therefore, applied non-singular computation procedure to overcome this singularity (Shin, 2005):

$$h = \text{sign}(|z| - |z_0|) \sqrt{(p - p_0)^2 + (z - z_0)^2} \quad (2-7)$$

where $\text{sign}(\cdot)$ Denotes the sign of the argument. And as shown in Figure 2-2,

1. $p_0 = R_N \cos \varphi$
2. $z_0 = R_N(1 - e^2) \sin \varphi$.

2.2.3 Local Level Frame (l-frame)

This is a local geographic frame or sometimes called navigation frame (n-frame). The origin of this frame is defined at the location of the navigation system. The axes of this frame are aligned with the directions of north, east and the local vertical (down) as shown in Figure 2-1.

The converting between the navigation frame (n-frame) and the Earth-Centered Earth-Fixed frame (e-frame) can be expressed using the direct cosine matrix (DCM)⁵ and quaternion, respectively, as follows:

⁵ DCM will be explained in section 2.3.1

$$R_l^e = \begin{bmatrix} -\sin \varphi \cos \lambda & -\sin \lambda & -\cos \varphi \cos \lambda \\ -\sin \varphi \sin \lambda & \cos \lambda & -\cos \varphi \sin \lambda \\ \cos \varphi & 0 & -\sin \varphi \end{bmatrix} \quad (2-8)$$

$$q_l^e = \begin{bmatrix} \cos\left(-\frac{\pi}{4} - \frac{\varphi}{2}\right) \cos \frac{\lambda}{2} \\ -\sin\left(-\frac{\pi}{4} - \frac{\varphi}{2}\right) \sin \frac{\lambda}{2} \\ \sin\left(-\frac{\pi}{4} - \frac{\varphi}{2}\right) \cos \frac{\lambda}{2} \\ \cos\left(-\frac{\pi}{4} - \frac{\varphi}{2}\right) \sin \frac{\lambda}{2} \end{bmatrix} \quad (2-9)$$

The earth rotation rate can be expressed in l-frame using:

$$\omega_{ie}^l = R_e^l \omega_{ie}^e = (R_l^e)^T [0 \quad 0 \quad \omega_e]^T = [\omega_e \cos \varphi \quad 0 \quad -\omega_e \sin \varphi]^T \quad (2-10)$$

2.2.4 The Computational Frame (c-frame)

The computational frame is the frame that the INS computer assumes to be the true navigation frame (Scherzinger, 2004) and it is locally leveled at computed position.

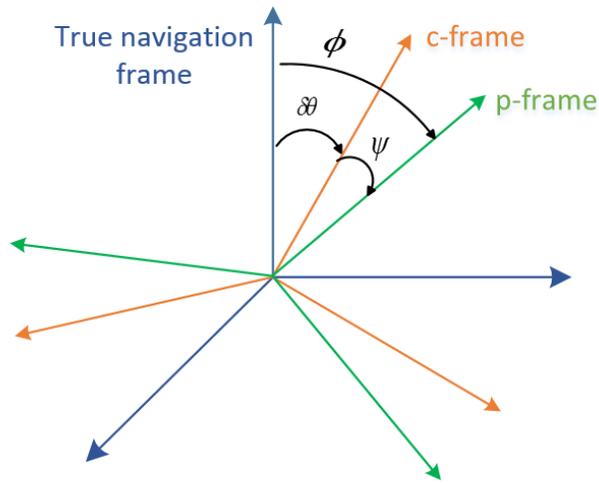


Figure 2-3: Relationship between true navigation, computer, and platform frames (Scherzinger, 1996)

2.2.5 The Platform Frame (p-frame)

This is where the transformed accelerations from the accelerometers and angular rates from the gyros are resolved (Scherzinger, 2004). Therefore, the computed rotation matrix from body to local level frame (\hat{R}_b^l) is the same as R_b^p :

$$\hat{R}_b^l = R_b^p = R_l^p R_b^l \quad (2-11)$$

From Figure 2-3, the ψ -angle is the orientation difference between the p-frame and c-frame. Therefore, for small attitude errors:

$$R_c^p = I - \Psi \quad (2-12)$$

where Ψ is the skew-symmetric matrix of the rotation vector pertaining to the platform tilt error ψ . The angle ϕ is the difference in orientation of the p-frame from the local level frame:

$$R_1^p = I - \Phi \quad (2-13)$$

where Φ is the skew-symmetric matrix of the rotation vector pertaining to the error of the attitude DCM (ϕ).

Therefore,

$$\phi = \psi + \delta\theta \quad (2-14)$$

2.2.6 The Body Frame (b-frame)

The body frame is the frame which the angular rates and accelerations are generated by INS sensor (Scherzinger, 2004). The axes of the b-frame are same as the axes of the IMU sensor. Its orientation is usually in line with that of the platform, but small misalignments are inevitable. The axes of the b-frame are aligned with the pitch (θ), roll (ϕ), and heading (ψ) of the assembly. More details about attitude representation will be discussed in section 2.3.

2.3 Attitude Representation

In (Shuster, 1993), numerous attitude representation techniques have been surveyed. Quaternion, rotation vector, Euler angles, and direction cosine matrix (DCM) are widely used, (Savage, 2000). This section discusses the relationships between all of these techniques. For more information, readers can refer to (Altmann, 2005; Kuipers, 2002).

In this section, special symbols abbreviations will be used. Letters such as α , β , and γ will represent general frames. The transformation from frame (β) to frame (α) will be represented as R_β^α , q_β^α , and ϕ for DCM, quaternion, and rotation vector, respectively. As per (Savage, 2000), the rotation

vector defines an axis of rotation and magnitude for a rotation about the rotation vector (Savage, 2000). Figure 2-4 illustrates two frames (α) & (β). Initially the (β) frame is aligned with frame (α) then rotated about r to obtain the final attitude of (β) frame.

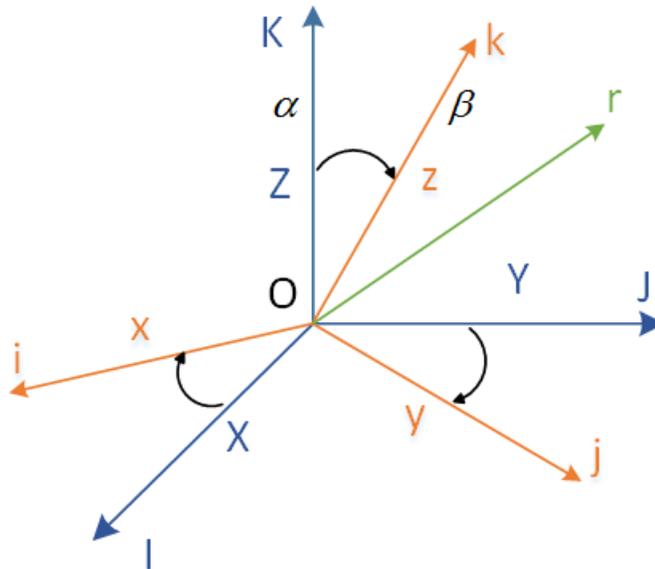


Figure 2-4: Rotation Frame

2.3.1 Direction Cosine Matrix (DCM)

Orientation kinematics deals with calculating the relative orientation of a body relative to a global coordinate system. It is useful to attach a coordinate system to our body frame. Both the global and the body frames have the same fixed origin O (see Figure 2-4). Let us also define i, j, k to be unity vectors co-directional with the body frame's $x, y,$ and z axes – in other words they are versors

of O_{ijk} and let I, J, K be the versors⁶ of global frame O_{IJK} . Thus, by definition, I, J, K can be expressed in terms of global coordinate as:

$$I^G = [1 \ 0 \ 0]^T, \quad J^G = [0 \ 1 \ 0]^T, \quad K^G = [0 \ 0 \ 1]^T \quad (2-15)$$

Similarly, i, j, k can be expressed as follows:

$$i^B = [1 \ 0 \ 0]^T, \quad j^B = [0 \ 1 \ 0]^T, \quad k^B = [0 \ 0 \ 1]^T \quad (2-16)$$

The vectors i, j, k can be expressed in terms of global coordinates as follows:

$$i^G = [i_x^G \ i_y^G \ i_z^G]^T, \quad j^G = [j_x^G \ j_y^G \ j_z^G]^T, \quad k^G = [k_x^G \ k_y^G \ k_z^G]^T \quad (2-17)$$

By analyzing the X coordinate i_x^G , it is calculated as the length of projection of i vector onto the global X axis. Similarly of Y and Z coordinates. Therefore:

$$i_x^G = |i| \cos(X, i) = \cos(I, i) \quad (2-18)$$

where $|i|$ is the norm of the i unity vector and $\cos(I, i)$ is the cosine of the angle formed by the vectors I and i . As $|I| = |i| = 1$,

$$i_x^G = \cos(I, i) = |I||i| \cos(I, i) = I \cdot i \quad (2-19)$$

where $(I \cdot i)$ is the scalar dot product of vectors I and i . Similarly $i_y^G = J \cdot i$ and $i_z^G = K \cdot i$

⁶ Versor of a vector is also known as a normalized vector or in other ward a unit vector.

now a convenient matrix form can be written as:

$$[i^G \quad j^G \quad k^G] = \begin{bmatrix} I.i & I.j & I.k \\ J.i & J.j & J.k \\ K.i & J.k & K.k \end{bmatrix} = \begin{bmatrix} \cos(I,i) & \cos(I,j) & \cos(I,k) \\ \cos(J,i) & \cos(J,j) & \cos(J,k) \\ \cos(K,i) & \cos(K,j) & \cos(K,k) \end{bmatrix} \quad (2-20)$$

This matrix is called Direction Cosine Matrix (DCM). It consists of all possible combinations of body and global versors.

Now, let us assume we have two frames (α) and (β). The DCM from frame (β) to frame (α), denoted as R_β^α , is 3×3 matrix and can be expressed as:

$$R_\beta^\alpha = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \quad (2-21)$$

Where the element (R_{ij}) represents the cosine of the angle between the i^{th} axis of the α -frame and j^{th} axis of the β -frame.

Multiplying the vector quantity (r^β) defined in β -frame with the R_β^α will result the same vector (r^α) expressed in α -frame as follows:

$$r^\alpha = R_\beta^\alpha r^\beta \quad (2-22)$$

2.3.2 Euler Angles

Euler angles is one of the popular transformation method. To transform one coordinate frame (β) to another (α), three successive rotations can be carried out about three different axes as follows:

1. Rotate through angle ψ about z-axis.
2. Rotate through angle θ about y-axis.
3. Rotate through angle ϕ about x-axis.

where ($\psi, \theta, and \phi$) are the Euler rotation angles. The mathematical expression of Euler angles in terms of DCM can be expressed in three different matrices as follows:

$$R_1 = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2-23)$$

$$R_2 = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \quad (2-24)$$

$$R_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \quad (2-25)$$

Then the transformation from α -frame to β -frame can be calculated as:

$$R_\alpha^\beta = (R_\beta^\alpha)^T = R_3 R_2 R_1 \quad (2-26)$$

Therefore, the DCM can be expressed as:

$$\begin{aligned} R_\alpha^\beta &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos \psi \cos \theta & \cos \theta \sin \psi & -\sin \theta \\ \cos \psi \sin \phi \sin \theta - \cos \phi \sin \psi & \cos \phi \cos \psi + \sin \phi \sin \psi \sin \theta & \cos \theta \sin \phi \\ \sin \phi \sin \psi + \cos \phi \cos \psi \sin \theta & \cos \phi \sin \psi \sin \theta - \cos \psi \sin \phi & \cos \phi \cos \theta \end{bmatrix} \end{aligned} \quad (2-27)$$

This is the direction cosine matrix that describes the transformation from α -frame to β -frame. Since all computational frames are orthogonal frames of references, the inverse of a transformation matrix becomes equivalent to its transpose:

$$R_{\beta}^{\alpha} = (R_{\alpha}^{\beta})^{-1} \rightarrow R_{\beta}^{\alpha} = (R_{\alpha}^{\beta})^T \quad (2-28)$$

Therefore, R_{β}^{α} can be written as follows:

$$R_{\beta}^{\alpha} = \begin{bmatrix} \cos \theta \cos \psi & \cos \psi \sin \phi \sin \theta - \cos \phi \sin \psi & \sin \phi \sin \psi + \cos \phi \cos \psi \sin \theta \\ \cos \theta \sin \psi & \cos \phi \cos \psi + \sin \phi \sin \psi \sin \theta & \cos \phi \sin \psi \sin \theta - \cos \psi \sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \quad (2-29)$$

Eq. (2-29) describes the transformation matrix from β -frame to α -frame which can be calculated easily when Euler angles are available.

On the other hand, Euler angles can be obtained when R_{β}^{α} is available from Eq. (2-29):

$$\theta = \tan^{-1} \frac{\sin \theta}{\cos \theta} = \tan^{-1} \frac{-R_{31}}{\sqrt{R_{32}^2 + R_{33}^2}} \quad (2-30)$$

$$\phi = \tan^{-1} \frac{\sin \phi}{\cos \phi} = \tan^{-1} \frac{R_{32}}{R_{33}} \quad (2-31)$$

$$\psi = \tan^{-1} \frac{\sin \psi}{\cos \psi} = \tan^{-1} \frac{R_{21}}{R_{11}} \quad (2-32)$$

From Eq. (2-30), the positive square root solution has been selected to bind the pitch angle (θ) to never be greater than $\left(\frac{\pi}{2}\right)$ (Savage, 2000, pp. 3-34).

Using small-angle approximation, the sine of small angles will approximately equal to the angle itself ($\sin \psi \approx \psi, \sin \theta \approx \theta, \sin \phi \approx \phi$). Moreover, the cosine of small angles approximately equal to unity ($\cos \psi \approx \cos \theta \approx \cos \phi \approx 1$). Therefore, Eq.(2-29), can be reduced approximately to skew symmetric form as follow:

$$R_{\beta}^{\alpha} \approx \begin{bmatrix} 1 & -\psi & \theta \\ \psi & 1 & -\phi \\ -\theta & \phi & 1 \end{bmatrix} \quad (2-33)$$

2.3.3 Attitude Quaternion

Quaternion is a mathematical technique to update the attitude in INS. The efficiency of this method put it at the top of other methods because of its linearity of quaternion differential equations, and because of small number of parameters contained (Farrell & Barth, 1998). This method is based on an idea of transforming one coordinate from a frame (α) to frame (β) by a single rotation about a vector q_{β}^{α} . This vector composed of two parts, scalar (q_s) and three dimensional vector parts (q_v) as shown in below equation:

$$q_{\beta}^{\alpha} \stackrel{\text{def}}{=} \begin{bmatrix} q_s \\ q_v \end{bmatrix} \quad (2-34)$$

The conjugate term of this vector $(q_{\beta}^{\alpha})^{-1}$ can be written as:

$$(q_\beta^\alpha)^{-1} = \begin{bmatrix} q_s \\ -q_v \end{bmatrix} \quad (2-35)$$

Moreover, the product of two quaternions can be defined as:

$$q_\beta^\alpha = q_\gamma^\alpha \star q_\beta^\gamma = \begin{bmatrix} q_{s_1} q_{s_2} - q_{v_1}^T q_{v_2} \\ q_{s_1} q_{v_2} + q_{s_2} q_{v_1} + q_{v_1} \times q_{v_2} \end{bmatrix} \quad (2-36)$$

where \star is the quaternion product (Miller, 1983).

To obtain the DCM from the corresponding quaternion vector as (Savage, 2000, pp. 3-46):

$$R_\beta^\alpha = \begin{bmatrix} q_1^2 + q_2^2 - q_3^2 - q_4^2 & 2(q_2 q_3 - q_1 q_4) & 2(q_2 q_4 + q_1 q_3) \\ 2(q_2 q_3 + q_1 q_4) & q_1^2 - q_2^2 + q_3^2 - q_4^2 & 2(q_3 q_4 - q_1 q_2) \\ 2(q_2 q_4 - q_1 q_3) & 2(q_3 q_4 + q_1 q_2) & q_1^2 - q_2^2 - q_3^2 + q_4^2 \end{bmatrix} \quad (2-37)$$

Therefore, the quaternion elements can be extracted from the DCM using the most robust technique (Savage, 2000; Shuster, 1993) as follows:

$$\begin{aligned} P_1 &= 1 + \text{tr}(R_\beta^\alpha) & P_2 &= 1 + 2R_{11} - \text{tr}(R_\beta^\alpha) \\ P_3 &= 1 + 2R_{22} - \text{tr}(R_\beta^\alpha) & P_4 &= 1 + 2R_{33} - \text{tr}(R_\beta^\alpha) \end{aligned} \quad (2-38)$$

Where $\text{tr}(\cdot)$ denotes the trace of a matrix. Four different scenarios are included based on the maximum value of (P_1, P_2, P_3, P_4) as follows (Savage, 2000, pp. 3-47):

Table 2-2: Quaternion from DCM

<i>max</i>	P_1	P_2	P_3	P_4
Step 1	$q_1 = \frac{1}{2}\sqrt{P_1}$	$q_2 = \frac{1}{2}\sqrt{P_2}$	$q_3 = \frac{1}{2}\sqrt{P_3}$	$q_4 = \frac{1}{2}\sqrt{P_4}$
Step 2	$q_2 = \frac{R_{32} - R_{23}}{4q_1}$	$q_3 = \frac{R_{21} + R_{12}}{4q_2}$	$q_4 = \frac{R_{32} + R_{23}}{4q_3}$	$q_1 = \frac{R_{21} - R_{12}}{4q_4}$
Step 3	$q_3 = \frac{R_{31} - R_{13}}{4q_1}$	$q_4 = \frac{R_{31} + R_{13}}{4q_2}$	$q_1 = \frac{R_{13} - R_{31}}{4q_3}$	$q_2 = \frac{R_{13} + R_{31}}{4q_4}$
Step 4	$q_4 = \frac{R_{21} - R_{12}}{4q_1}$	$q_1 = \frac{R_{32} - R_{23}}{4q_2}$	$q_2 = \frac{R_{21} + R_{12}}{4q_3}$	$q_3 = \frac{R_{32} + R_{23}}{4q_4}$

Note: If $q_1 < 0 \rightarrow q := -q$

Quaternion can also be computed from Euler angles as follows (McGreevy & Litton Systems, 1986) :

$$q_{\beta}^{\alpha} = \begin{bmatrix} \cos \frac{\phi}{2} \cos \frac{\theta}{2} \cos \frac{\psi}{2} + \sin \frac{\phi}{2} \sin \frac{\theta}{2} \sin \frac{\psi}{2} \\ \sin \frac{\phi}{2} \cos \frac{\theta}{2} \cos \frac{\psi}{2} - \cos \frac{\phi}{2} \sin \frac{\theta}{2} \sin \frac{\psi}{2} \\ \cos \frac{\phi}{2} \sin \frac{\theta}{2} \cos \frac{\psi}{2} + \sin \frac{\phi}{2} \cos \frac{\theta}{2} \sin \frac{\psi}{2} \\ \cos \frac{\phi}{2} \cos \frac{\theta}{2} \sin \frac{\psi}{2} - \sin \frac{\phi}{2} \sin \frac{\theta}{2} \cos \frac{\psi}{2} \end{bmatrix} \quad (2-39)$$

2.4 INS Mechanization Algorithm

Mechanization is the process of converting the output of the IMU into position, velocity and attitude information as shown in Figure 2-5. With reference to Newton's second law, and as per (El-Sheimy, 2012; Nouredin et al., 2012; Sahli, Al-Hamad, Ali, & El-Sheimy, 2013; Shin, 2005), the mechanization equations can be expressed as the following set of first order differential equations:

$$\begin{bmatrix} \dot{r}^l \\ \dot{v}^l \\ \dot{R}_b^l \end{bmatrix} = \begin{bmatrix} D^{-1}v^l \\ R_b^l f^b - (2\Omega_{ie}^l + \Omega_{el}^l)v^l + g^l \\ R_b^l(\Omega_{ib}^b - \Omega_{il}^b) \end{bmatrix} \quad (2-40)$$

where (Ω_{ie}^l) and (Ω_{el}^l) are the skew-symmetric matrices corresponding to (ω_{ie}^l) and (ω_{el}^l) , respectively. (Ω_{ib}^b) is the skew-symmetric matrix of the angular velocity measurements provided by the gyroscopes and corresponds to the angular velocity vector (ω_{ib}^b) . More detail information about INS mechanization can be found in (Council on Environmental Quality (U.S.) & Saflianis; El-Sheimy, 2012; KING, 1998; Nassar, 2003; Sahli et al., 2013; Shin, 2001)

2.4.1 Measurement Data

IMU outputs includes accelerations (f^b) and rotation rates (ω_{ib}^b). Where the basic principle of an INS is based on the integration of these accelerations and rotation rates as shown in Eq. (2-41), respectively.

$$p = p_0 + v_0 t + \iint a(t) dt dt \quad (2-41)$$

$$A = A_0 + \int \omega_{gyro} dt$$

where p, v, a, A represent the position, velocity, acceleration and the angle, respectively. Thus, an appropriately initialized inertial navigation system is capable of continuous determination of vehicle position, velocity and attitude without the use of the external information (Britting & R., 1971). It is good to notice that this process is recursive process as shown in Figure 2-5. This means that this process starts with a specified set of initial values and iterates on the output (Noureldin et al., 2012).

Both sensed acceleration and angular rates are digitized with predefined frequency. The output from the INS sensors can be divided into two different types:

1. Incremental angles due to angular motion ($\Delta\tilde{\theta}_{ib}^b$), and incremental velocities due to specific forces ($\Delta\tilde{v}^b$).
2. Angular rate due to angular motion ($\tilde{\omega}_{ib}^b$), and specific forces (\tilde{f}^b).

where $(\tilde{\cdot})$ denotes values corrupted by sensor errors. The following equations describe the relationships between these two types:

$$\Delta\tilde{v}^b = \tilde{f}^b \Delta t \quad (2-42)$$

$$\Delta\tilde{\theta}_{ib}^b = \tilde{\omega}_{ib}^b \Delta t \quad (2-43)$$

where:

$\Delta\tilde{\theta}_{ib}^b$: The angle that describe the change in angular rate during the time interval t

$\Delta\tilde{v}^b$: The velocity that describe the change in specific force during the time interval t

\tilde{f}^b : Specific force (m/s^2)

$\tilde{\omega}_{ib}^b$: Rotation rate of the body frame with respect to the inertial frame and resolved in the body frame (rad/s). This is simply is the output of the gyroscope.

Δt : Sampling interval (sec)

To compensate for the known errors, the following equations can be applied (Shin, 2005):

$$\Delta\theta_{ib}^b = \frac{\Delta\tilde{\theta}_{ib}^b - b_{gyro}\Delta t}{1 + S_{gyro}} \quad (2-44)$$

$$\Delta v^b = \frac{\Delta\tilde{v}^b - b_{acc}\Delta t}{1 + S_{acc}} \quad (2-45)$$

where:

$\Delta\theta_{ib}^b$: Corrected incremental gyroscope output (rad/s)

Δv^b : Corrected incremental accelerometer output (m/s)

b_{gyro} : Gyroscope bias (rad/s)

b_{accel} : Accelerometer bias (m/s^2)

S_{gyro} : Gyroscope scale factor (*ppm*)

S_{accel} : Accelerometer scale factor (*ppm*)

2.4.2 Updating Rotation Matrix and Quaternion

As the mechanization process is recursive, the rotation matrix should be updated every iteration.

The angular rate of the body with respect to the local level frame can be expressed as follows:

$$\omega_{ib}^b = \omega_{ib}^b - \omega_{il}^b \quad (2-46)$$

where (ω_{il}^b) is the rotation rate if the local level frame with respect to the inertial frame and resolved in the body frame. (ω_{il}^b) can be computed as:

$$\begin{aligned} \omega_{il}^b &= R_l^b \omega_{il}^l = R_l^b (\omega_{ie}^l + \omega_{el}^l) \\ &= R_l^b (R_e^l \omega_{ie}^e + \omega_{el}^l) \end{aligned} \quad (2-47)$$

where (ω_{ie}^l) and (ω_{el}^l) can be expressed in local level frame (NED) as follows:

$$\omega_{ie}^l = \begin{bmatrix} -\sin \varphi \cos \lambda & -\sin \varphi \sin \lambda & \cos \varphi \\ -\sin \lambda & \cos \lambda & 0 \\ -\cos \varphi \cos \lambda & -\cos \varphi \sin \lambda & -\sin \varphi \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \omega_e \end{bmatrix} = \begin{bmatrix} \omega_e \cos \varphi \\ 0 \\ -\omega_e \sin \varphi \end{bmatrix} \quad (2-48)$$

$$\omega_{el}^l = \begin{bmatrix} \frac{v_e}{R_N + h} \\ -\frac{v_n}{R_M + h} \\ \frac{v_e \tan \varphi}{R_N + h} \end{bmatrix} \quad (2-49)$$

where (v_e, v_n) are the velocities in east and north, respectively. The height is represented by (h) .

The angular increment of the body rotation with respect to the local level frame can be obtained by integrating Eq. (2-46) for the interval Δt :

$$\theta_{lb}^b = \theta_{ib}^b - \theta_{il}^b \quad (2-50)$$

Now it is the time for quaternion's update. The closed form of quaternion equation is:

$$q_{k+1} = q_k + \frac{1}{2} \left[2 \left(\cos \frac{\theta}{2} - 1 \right) I + \frac{2}{\theta} \sin \frac{\theta}{2} \bar{S}(\omega) \right] q_k \quad (2-51)$$

Witch can be written in expanded form as:

$$\begin{aligned} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}_{k+1} &= \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}_k + \frac{1}{2} \begin{bmatrix} c & s\omega_{lb,z}^b \Delta t & -s\omega_{lb,y}^b \Delta t & s\omega_{lb,z}^b \Delta t \\ -s\omega_{lb,z}^b \Delta t & c & s\omega_{lb,x}^b \Delta t & s\omega_{lb,y}^b \Delta t \\ s\omega_{lb,y}^b \Delta t & -s\omega_{lb,x}^b \Delta t & c & s\omega_{lb,z}^b \Delta t \\ -s\omega_{lb,x}^b \Delta t & -s\omega_{lb,y}^b \Delta t & -s\omega_{lb,z}^b \Delta t & c \end{bmatrix}_k \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}_k \\ &= \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}_k + \frac{1}{2} \begin{bmatrix} c & s\theta_{lb,z}^b & -s\theta_{lb,y}^b & s\theta_{lb,z}^b \\ -s\theta_{lb,z}^b & c & s\theta_{lb,x}^b & s\theta_{lb,y}^b \\ s\theta_{lb,y}^b & -s\theta_{lb,x}^b & c & s\theta_{lb,z}^b \\ -s\theta_{lb,x}^b & -s\theta_{lb,y}^b & -s\theta_{lb,z}^b & c \end{bmatrix}_k \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}_k \end{aligned} \quad (2-52)$$

Where c and s are defined as follows:

$$\begin{aligned} c &= 2 \left(\cos \frac{\theta}{2} - 1 \right) \\ s &= \frac{2}{\theta} \sin \frac{\theta}{2} \end{aligned} \quad (2-53)$$

The magnitude of the incremental angle of the body rotation about a fixed axis is:

$$\theta^b = \sqrt{(\theta_{ib,x}^b)^2 + (\theta_{ib,y}^b)^2 + (\theta_{ib,z}^b)^2} \quad (2-54)$$

2.4.3 Attitude Update

After computing the quaternion, using Eq. (2-37), direct cosine matrix can be computed. All attitude angles can be extracted from the DCM as shown in Eq. (2-29) as express in Eq. (2-30), (2-31), and (2-32).

2.4.4 Velocity Update

From Eq. (2-40), the digital algorithm for velocity update can be written as follows (Noureldin et al., 2012; Savage, 2000):

$$\begin{aligned} \frac{\Delta v^l}{\Delta t} &= R_b^l f^b - (2\Omega_{ie}^l + \Omega_{el}^l)v^l + g^l \\ \Delta v^l &= \Delta \tilde{v}^l - (2\Omega_{ie}^l + \Omega_{el}^l)v^l \Delta t + g^l \Delta t \end{aligned} \quad (2-55)$$

where:

$\Delta \tilde{v}^l$: Measurement velocity increment in local level frame, refer to (2-42).

$(2\Omega_{ie}^l + \Omega_{el}^l)v^l \Delta t$: Coriolis correction that compensates for the earth's rotation and the resulting change of orientation of the local level frame.

$g^l \Delta t$: Gravity correction. Where $g^l = [0 \ 0 \ -g]^T$ and g can be calculated as shown in Eq. (2-56).

$$g = a_1(a + a_2 \sin^2 \varphi + a_3 \sin^4 \varphi) + (a_4 + a_5 \sin^2 \varphi)h + a_6 h^2 \quad (2-56)$$

Finally, the current velocity epoch can be calculated by:

$$v_{k+1}^l = v_k^l + \frac{1}{2}(\Delta v_k^l + \Delta v_{k+1}^l) \quad (2-57)$$

2.4.5 Position Update

After calculating the velocities, the position coordinates (latitude, longitude, and height) is ready to be updated. The change rate of the position vector is expressed as follows (El-Sheimy, 2012):

$$\dot{r}^l = \begin{bmatrix} \dot{\varphi} \\ \dot{\lambda} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} \frac{1}{R_M + h} & 0 & 0 \\ 0 & \frac{1}{(R_N + h) \cos \varphi} & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} v^{n,l} \\ v^e \\ v^d \end{bmatrix} = D^{-1} v^l \quad (2-58)$$

Therefore,

$$\varphi_{k+1} = \varphi_k + \frac{1}{2} \frac{(v_{n,k} + v_{n,k+1})}{R_M + h} \Delta t \quad (2-59)$$

$$\lambda_{k+1} = \lambda_k + \frac{1}{2} \frac{(v_{e,k} + v_{e,k+1})}{(R_N + h) \cos \varphi} \Delta t \quad (2-60)$$

$$h_{k+1} = h_k + \frac{1}{2} (v_{u,k} + v_{u,k+1}) \Delta t \quad (2-61)$$

More detail information about INS mechanization can be found in (Council on Environmental Quality (U.S.) & Saflianis; El-Sheimy, 2012; KING, 1998; Nassar, 2003; Sahli et al., 2013; Shin, 2001)

2.5 Aiding Sensors for INS

High tactical grade INS's errors behavior is well described by Schuler dynamics (with its period about 84.4 minutes). However, a low-cost INS cannot run in stand-alone mode for long periods and, in extreme cases, can even experience computational failures before some part of the Schuler period can be seen. Therefore, an external aided sensors with external navigation related information should be included. This section will investigate two main aiding options that are useful and used in pig positioning.

2.5.1 Global Navigation Satellite System (GNSS)

GNSS signals is not available inside the pipelines. However, AGMs are available sometime as highlighted in section 1.4. The GNSS antenna is located above the ground. Therefore, the position of the IMU (inside the pig) is different from the GNSS antenna. This difference is called lever-arm effect and can be described as follows:

$$r_{GNSS,i}^l = r_{IMU}^l + D^{-1}R_b^l l_{GNSS,i}^b \quad (2-62)$$

Where D^{-1} is shown in Eq. (2-58). $r_{GNSS,i}^l$ and r_{IMU}^l are the positions of the i th GNSS antenna center and the center of the IMU in the local level frame, respectively. The lever-arm vector is represented by $l_{GNSS,i}^b$.

Similarly, in velocity measurement, the lever-arm effect can be considered as follows:

$$v_{GNSS,i}^l = v_{IMU}^l - (\Omega_{ie}^l + \Omega_{el}^l)R_b^l l_{GNSS,i}^b - R_b^l L_{GNSS,i}^b \omega_{ib}^b \quad (2-63)$$

Where $L_{GNSS,i}^b$ is the skew-symmetric matrix of the lever-arm vector.

Since AGM provides only static position (t, φ, λ, h) , loosely coupled integration can be used to aid the INS. Therefore, only Eq. (2-62) can be used directly as a measurement model.

2.5.2 Distance Measurement System

The vehicle frame velocity can be obtained for the odometers and/or speedometers. In general, development of a rigorous model would require information about friction between wheels and the inner side of the pipeline, angular speed of the wheels, and the suspension system. Since it is very hard to acquire information about all of these parameters, a simplified model will be used.

Assume the pig has along-track speed, v , and cross-track velocities equal to zero. Therefore, the velocity in the vehicle frame is $v^v = [v \ 0 \ 0]^T$. The wheels are installed as shown in Figure 1-4. Therefore, the relationship between the velocity of the vehicle (pig) at the center of the IMU,

v_{IMU}^l , which is installed at the center of the pig, and the wheels, v_{wheel}^v , can be expressed as (Shin, 2005):

$$v_{wheel}^v = R_b^v R_l^b v_{IMU}^l + R_b^v \Omega_{lb}^b l_{wheel}^b \quad (2-64)$$

where l_{wheel}^b is the lever-arm vector of the wheel sensor in the body frame. It is good to notice from Figure 1-4 that pig has multiple odometers/ speedometers that work in redundancy. ZUPT will be applied in case of the along-track speed drops below a certain predefined threshold and Eq. (2-64) will not be active for the same epoch.

2.5.3 Other Aiding Sources

Different aiding sources can be added to improve the navigation solution, such as:

1. Visual aiding using laser scanner and/or camera.
2. Magnetic heading sensors.
3. Map matching (aiding by database).

In this thesis, these additional aiding sensors have not been used due to the difficulty of collecting pipeline data with additional sensors. Therefore, measurement models for these sensors will not be considered here.

Chapter Three : CALIBRATION OF INERTIAL MEASUREMENT UNIT

3.1 Introduction

Accelerometers and gyroscopes errors typically include biases, scale factors, triad non-orthogonality and noise. In general, both the stability of the biases and the magnitude of the noise are the two common factors to determine the quality of the sensors. Therefore, the cost of the sensors varies due to these factors. Considering that this research aims to reduce the pipeline navigation cost by replacing the high-tactical grade IMUs with MEMS-based IMUs, which means reducing the system cost and exploiting the size of MEMS-IMU devices for small diameter pipelines. Moreover, the power consumption is much less in MEMS-IMU.

Biases are errors independent and uncorrelated of the angular velocity and specific force ("IEEE Standard for Inertial Sensor Terminology," 2001). The turn-on and the time variant biases are the main MEMS grade IMU bias parts. As shown in Chapter Four, the biases estimated in the filter.

Table 3-1: IMU Grade Bias Specifications

IMU Grade	Gyro Bias	Accelerometer Bias
Strategic	0.0001 ($^{\circ}/h$)	1 μg
Navigation	Around $\frac{1}{1000}$ of earth rotation rate	50 – 100 μg
Tactical	1 – 5 ($^{\circ}/h$)	100 – 1000 μg
MEMS	> 5 ($^{\circ}/h$)	> 1000 μg

The alignment of the IMU is the determination of the DCM (R_b^l). As the navigation equations (2-40), require starting values for position, velocity and attitude. Usually, these values are available from the last epoch of an ongoing iteration. However, for the first epoch, attitude values should be determined.

In this chapter, both calibration and alignment procedures with their effect to our navigation solution will be illustrated.

3.2 Inertial Sensor Errors

In order to assess the inertial sensors, many different characteristics should be considered. Repeatability, Stability, and drift are some general terms used to assess the IMU performance. The errors in the inertial sensors limit the accuracy to which the observables can be measured (Noureldin et al., 2012). These errors can be classified to two broad categories (i.e. Systematic & Random errors).

Systematic errors can be compensated by laboratory calibration (i.e. scale factor, bias offset, non-linearity, non-orthogonality, misalignment, and quantization errors) as shown in Figure 3-1.

The more important errors are the random errors, which are usually, change in time. In general, these errors are modeled stochastically to mitigate their effects.

3.2.1 Bias Drift

A random change in bias over time is called bias drift. It is different from the systematic bias offset. While the IMU is powered on, the initial bias changes over time. The temperature, mechanical

stress, and time are the main reasons of this change. To increase the stability of the measurements, IMUs are often manufactured with temperature compensation.

INS filter continuously estimates the bias drift by using external sources of information (Ex. GNSS, odometer, barometer ...etc.). Then the estimated bias drift value is compensated in the IMU measurements before using them in the mechanization model.

In IMU sensors datasheets, bias drift sometimes called Bias-Stability or In-Run bias. The measurement unit of this bias is ($deg/hr/hr$) or ($m/s^2/hr$). Figure 3-2 illustrates the bias drift concept.

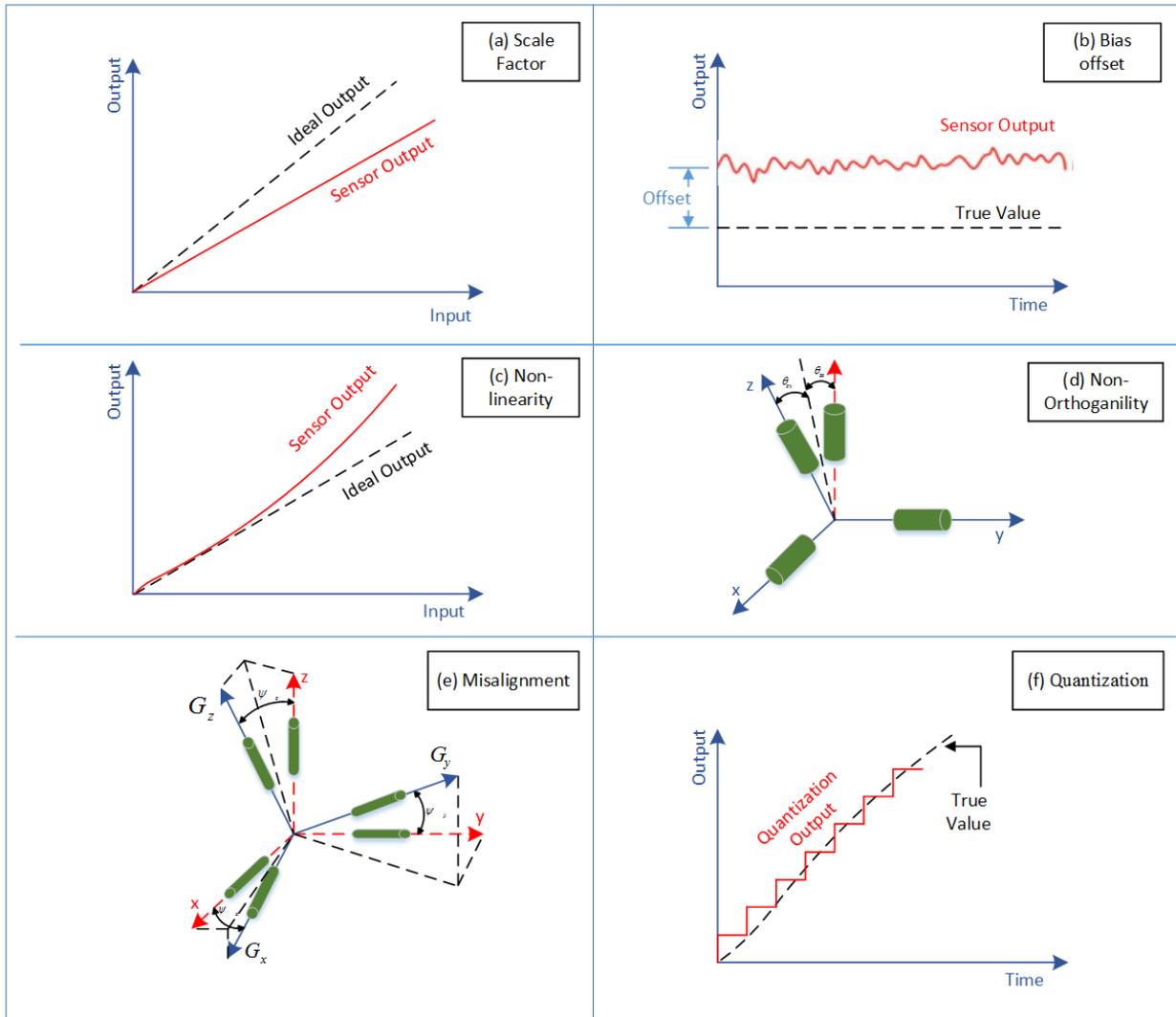


Figure 3-1: Systematic IMU Errors

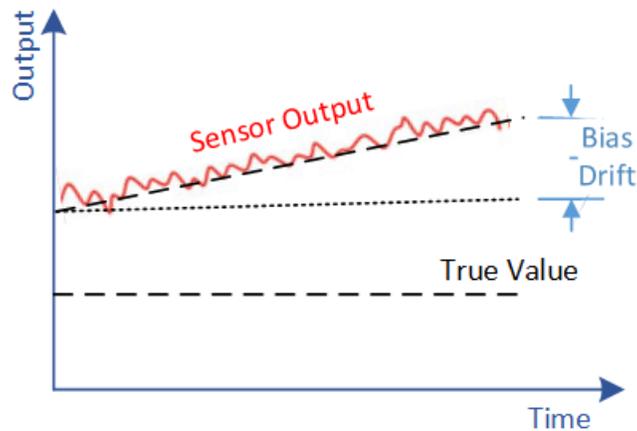


Figure 3-2: Bias Drift

3.2.2 Scale Factor Instability

The error relation between sensor input and output called scale factor (SF). Theoretically, when the input is 100%, the expected output is 100%. However, in reality, the actual output is proportional to the input and scaled. As an example, if the input is $5(m/s^2)$, and assume there is a 3% scale factor error, the output measurement is $5.15 (m/s^2)$. This error can be described or illustrated as a slope of the sensor signal, see Figure 3-3.

Scale factor combine both linear and non-linear parts. The linear part can be measured during the calibration phase in the laboratory. This part is considered as a part of systematic error. However, scale factor error has a nonlinear part that is changed during the run. The change in the scale factor is estimated in the INS filter every iteration. Manufactured combine both linear and nonlinear parts in one value. The measurement unit for this error is (parts per million – *PPM*). For the previous example, the scale factor is $30,000 ppm$. In general, Scale factor effects appear mostly in high acceleration and rotation applications.

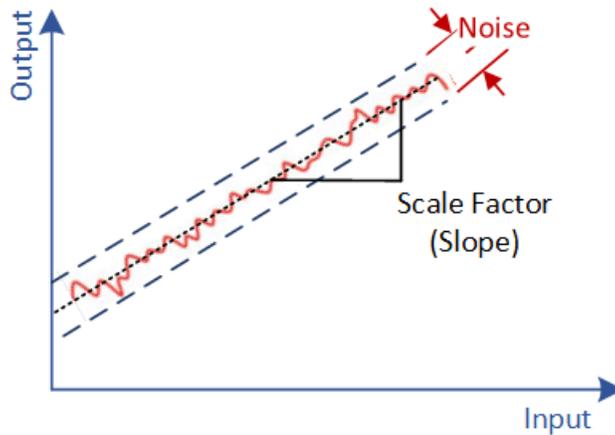


Figure 3-3: Scale Factor

3.2.3 Random Walk

White noise is an uncorrelated noise that is distributed in all frequencies. For a constant measurement signal, a random noise (error) in the measurement is always present. This type of error is described as a stochastic process and is minimized using statistical techniques. In the final solution, a random walk error is appear due to the integration of the random noises in the measurements. Random walk error plays an important role is in static alignment. Therefore, static alignment quality is affected directly by the random walk error of the sensor. Furthermore, the performance of GNSS/INS integration and during the GNSS outage, the sensor noise error causes the solution error to grow unbounded (Navatel, 2014).

3.3 Mathematical Models of INS Errors

INS consists of accelerometers and gyroscopes. Accelerometers and gyroscopes error models are described as follows:

3.3.1 Accelerometer Model Errors

The accelerometer measures specific force and is used to derive linear motion in three mutually orthogonal directions. Measurements of the specific force can be modeled as the following observation equation (El-Sheimy, 2012):

$$\tilde{f}^b = f^b + b_a + S_a f + N_a f + \delta g + \varepsilon_a \quad (3-1)$$

where

\tilde{f}^b : Accelerometer measurement vector (m/s^2)

f^b : Specific force vector (m/s^2)

b_a : Accelerometer bias vector (m/s^2)

S_a : Scale factor error matrix (3×3)

N_a : Non-orthogonality of accelerometer triad matrix (3×3)

δg : Anomalous gravity vector (m/s^2)

ε_a : Sensor noise (m/s^2)

3.3.2 Gyroscope Model Errors

Gyroscope sensor can measure either angular rate or attitude depending on the sensor type whether it is rate sensing type or rate integrating type. Measurements of the angular rate can be modeled as the following observation equation (El-Sheimy, 2012):

$$\tilde{\omega}_{ib}^b = \omega_{ib}^b + b_g + S\omega_{ib}^b + N_g\omega_{ib}^b + \varepsilon_g \quad (3-2)$$

where

$\tilde{\omega}_{ib}^b$: Gyroscope measurement vector (*deg/hr*)

ω_{ib}^b : Angular rate vector (*deg/hr*)

b_g : Gyroscope bias vector (*deg/hr*)

S : Scale factor error matrix (3×3)

N_g : Non-orthogonality of gyroscope triad matrix (3×3)

ε_g : Sensor noise (*deg/hr*)

The scale factor and non-orthogonality matrices can be written as follows:

$$S_i = \begin{bmatrix} S_{ix} & 0 & 0 \\ 0 & S_{iy} & 0 \\ 0 & 0 & S_{iz} \end{bmatrix} \quad N_i = \begin{bmatrix} 0 & \theta_{i,xy} & \theta_{i,xz} \\ \theta_{i,yx} & 0 & \theta_{i,yz} \\ \theta_{i,zx} & \theta_{i,zy} & 0 \end{bmatrix} \quad (3-3)$$

Where the subscript (i) denotes whether the matrix is for accelerometer (a) or gyroscope (g).

3.4 INS Calibration Methods

Calibration is the process of comparing sensor outputs with known reference of information to determine the coefficients that make the output equal to the reference information (Chatfield, 1997). Calibration of INS is needed because the outputs of sensors contain errors. Hence, the required parameters to be determined can be changed based on the IMU manufacture technology. To determine all parameters precisely, some special calibration instruments (i.e. three axial turntable) or special calibration techniques are required.

3.4.1 Six-Position Static Acceleration Test

In this test, the INS is mounted on a level table with each sensitive axis pointing alternately up and down and get the average of (10-15) minutes. The total number of positions for 3 axes is 6. The bias and scale factor for each axis can be computed using the following formula:

$$b_{i,a} = \frac{f_{i,up} + f_{i,down}}{2} \quad , \quad S_{i,a} = \frac{f_{i,down} - f_{i,up} - 2g}{2g} \quad (3-4)$$

where the subscript (*i*) represents the accelerometer axis (*x, y, or z*).

Similarly for gyroscopes, placing the sensor in static mode with the axis being calibrated pointing vertically upward and downward for all three gyros and get the average of (10-15) minutes measurements (ω_{up}):

$$b_{i,g} = \frac{\omega_{i,up} + \omega_{i,down}}{2} \quad , \quad S_{i,g} = \frac{\omega_{i,down} - \omega_{i,up} - 2\omega_e \sin \varphi}{2\omega_e \sin \varphi} \quad (3-5)$$

where ω_e is the earth rotation rate and φ is the latitude of the gyro location. The drawback of Eq. (3-5) is that it will not work with MEMS-based gyroscopes because the output random noise of the MEMS gyroscopes is greater than the magnitude of the Earth's rotation rate (ω_e).

Using this method, the non-orthogonality cannot be determined. Furthermore, the misalignment of the IMU sensitive axes with respect to vertical axis has an effect even if it is very small (Rogers, 2007).

3.4.2 Local Level Frame Calibration

This calibration method is similar to the previous method (Six-Position Static Acceleration Test). The difference here is that the INS sensor should be oriented precisely with respect to the local level frame. Then the IMU is rotated through a series of accurately known angles and positioned in different orientations with respect to the local level frame.

Deterministic sensor errors can be then determined from static measurements of acceleration and angular rate taken in each orientation of the unit. For more details, see (Salychev, 1998).

3.4.3 Angle Rate Test

Using a precision rate table, the IMU rotates through very accurately known angles. Estimates of the gyro biases and scale factor errors can be calculated by comparing these known rotations with estimates of the angles turned through by the systems, derived by integrating the rate outputs provided by the gyros. A leveled rotation table for IMU calibration is shown in Figure 3-4. For more details, see (Titterton & Weston, 2004).

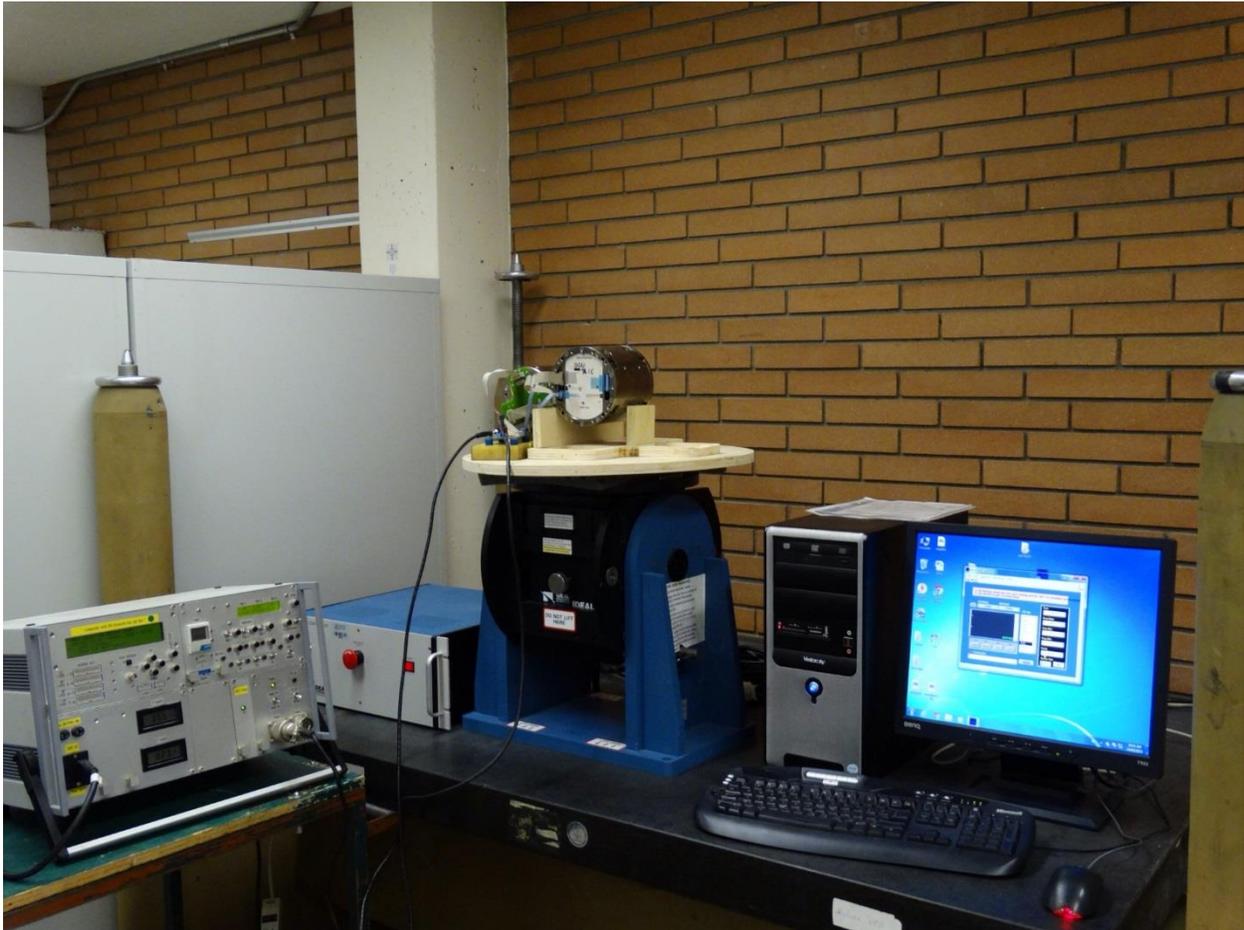


Figure 3-4: IMU Calibration - Rotation Table

3.4.4 Allan Variance Analysis

The Allan variance analysis method is used to identify and quantify random noise modes with different autocorrelation properties and expose their effect when the output signal is integrated over time. In other words, it can be defined as a method of analyzing a time sequence to pull out the intrinsic noise in the system as a function of the averaging time. In the time domain signal, Allan variance analysis consists of computing its root Allan variance $\sigma(\tau)$ for different integration time constants τ and then analyzing the characteristic regions. A log-log scale slopes of the $\sigma(\tau)$

curve can be generated to identify different noise modes (i.e., random components of the signal with different autocorrelation power laws) (Trusov, 2011).

At short averaging times, the Allan Variance is controlled by the noise in the gyro sensor. There is a direct correlation between the noise of the output (standard deviation) and the slope of the Allan Variance at small t (Angle Random Walk – ARW). The variance decreases by increasing the average time. However, due to rate random walk (inherent instability in the output of the sensor); the Allan Variance starts to increase again at some point. The minimum point on the log-log curve of the Allan Variance defines the bias instability used by the inertial sensor.

To perform Allan Variance analysis:

1. a long sequence of data is required. This dataset need to be divided into bins based on an averaging time (τ). There must be enough data for at least 9 bins.
2. Average the data in each bin to obtain a list of averages ($y(\tau)_1, y(\tau)_2, \dots, y(\tau)_n$), where n is the number of bins.
3. Allan variance can be computed as follows:

$$AVAR^2(\tau) = \frac{1}{2(n-1)} \sum_i (y(\tau)_{i-1} - y(\tau)_i)^2 \quad (3-6)$$

The characteristics of underlying noise processes can be determined by taking the square root of the Allan Variance. Allan Deviation is plotted as a function of t on a log-log scale. Different types of random process cause slopes with different gradients to appear on the plot, as shown in Figure 3-5 and Figure 3-6. Moreover, different processes usually appear in different regions of τ . Their presence can be easily identified that will allow us to read its numerical parameters directly. The

important processes for MEMS-based IMUs that we want to measure are random walk and bias instability, which can be identified and read as follows:

From the figures, bias instability shown as a flat region around the minimum. The numerical value is the minimum value on the Allan Deviation curve.

A slope with gradient (-0.5) determines the white noise. The angular random walk of the gyro and velocity angular walk of the accelerometer can be obtained by fitting a straight line through the slope and reading its value at ($\tau = 1$).

For more information and full description of the Allan Variance analysis technique, reader can see (El-Sheimy, Haiying, & Xiaoji, 2008; Hou, 2004; IEEE, 2008; Trusov, 2011).

3.5 Calibration Results

For this thesis, SiIMU02 sensor ("MEMS Inertial Measurement Unit - SiIMU02 Datasheet," 2010) is a MEMS-based IMU. It has been selected for this project as explained in section 6.1. Table 3-2 shows the calibrated parameters for this IMU.

Table 3-2: SiIMU02 Calibrated Parameters

Parameter	Datasheet	Calibration		
		x	y	z
<i>ARW</i> ($^{\circ}/hr$)	0.1 – 0.5	0.391		
<i>VRW</i> ($m/s / \sqrt{hr}$)	0.5	0.178		

Parameter		Datasheet	Calibration	Parameter	Datasheet
			x	y	z
Bias Instability	Gyro ($^{\circ}/hr$)	1.5 - 6.5	0.063	0.131	0.024
	Accel (mg)	< 0.5	0.118	0.097	0.115
Bias Repeatability	Gyro ($^{\circ}/hr$)	50 - 100	-37.544	-65.893	42.499
	Accel (mg)	< 10	-5.408	3.610	-6.801

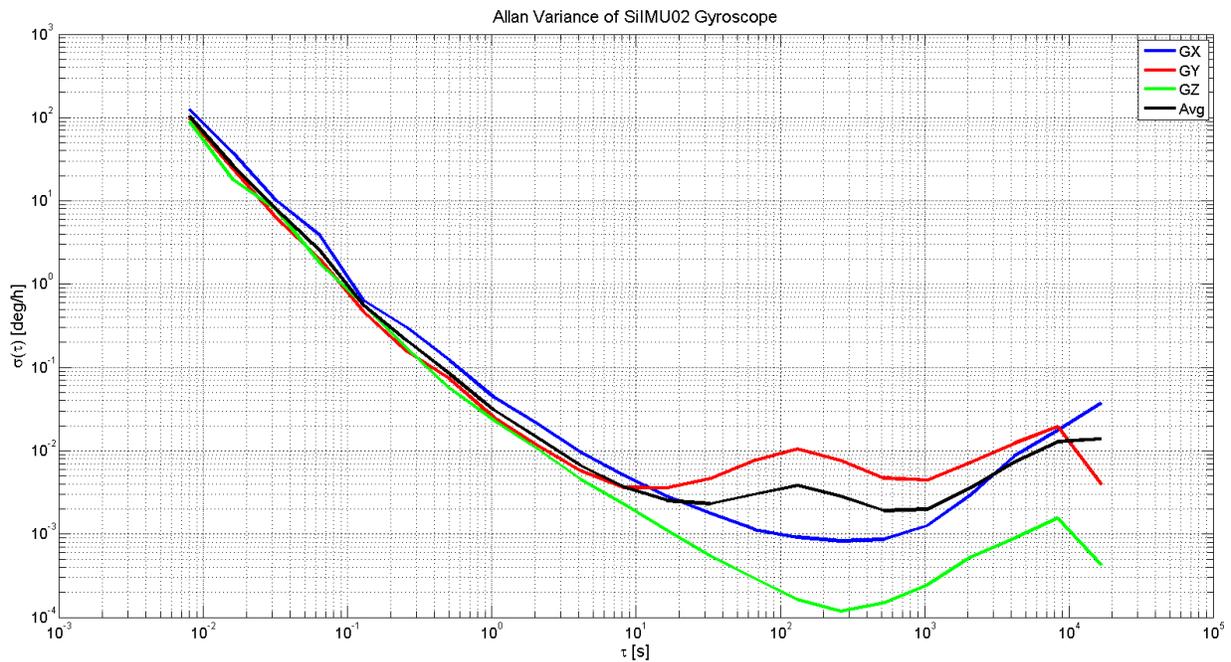


Figure 3-5: Allan Variance of SiIMU02 Gyroscopes

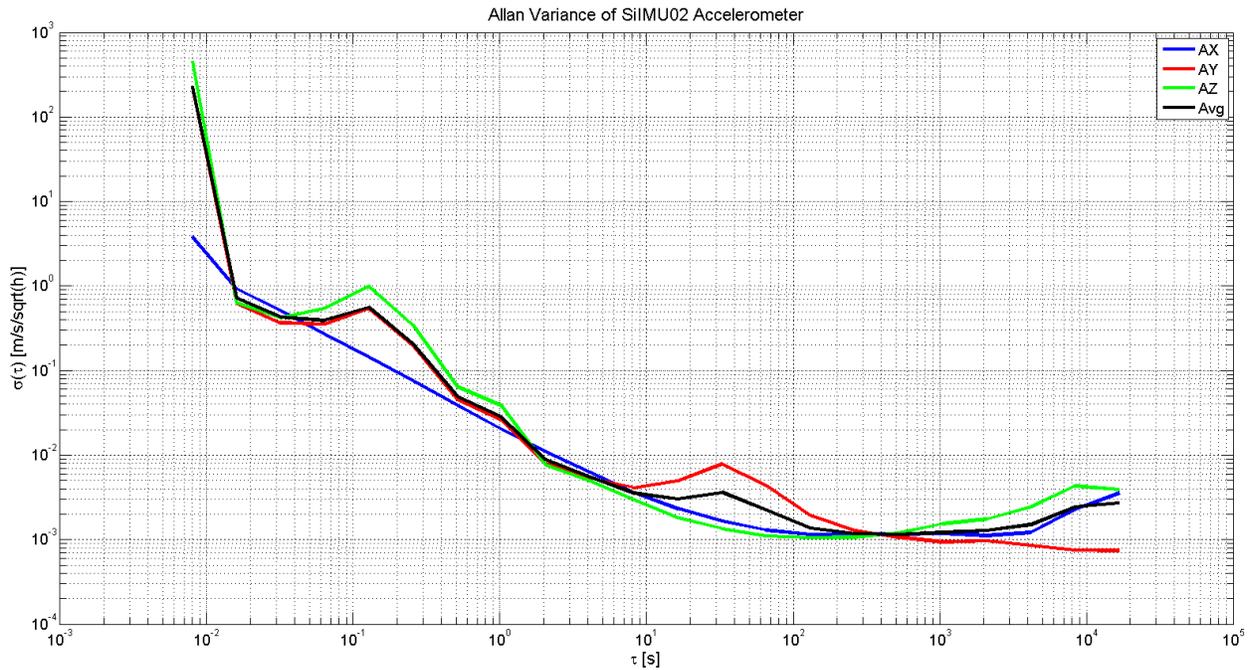


Figure 3-6: Allan Variance of SiIMU02 Accelerometers

3.6 MEMS-INS Alignment

Alignment is referred to the procedure of initializing the INS. Particularly, the attitude information between the body frame and the local level frame. Alignment algorithm can be classified in different ways. Based on the attitude errors, they can be divided to two categories, coarse alignment and fine alignment methods. Typically, the threshold of the attitude errors can reach few degrees between both methods. Based on the dynamics of the vehicle upon initialization, then alignment algorithms can be classified as stationary alignment or in-motion alignment methods.

INS coarse alignment is done in stationary mode using either leveling (by accelerometers) followed by gyro-compassing, or using an analytical method (Britting & R., 1971). The analytical method can be applied in one-step as follows:

$$R_b^l = \begin{bmatrix} -\frac{\tan \varphi}{g} & \frac{1}{\omega_e \cos \varphi} & 0 \\ 0 & 0 & -\frac{1}{g \omega_e \cos \varphi} \\ -\frac{1}{g} & 0 & 0 \end{bmatrix} \begin{bmatrix} (f^b)^T \\ (\omega_{ib}^b)^T \\ (f^b \times \omega_{ib}^b)^T \end{bmatrix} \quad (3-7)$$

Due to the large biases and low signal-to-noise ratio of MEMS-based gyroscopes, analytical coarse alignment and gyro-compassing cannot be applied. However, roll (ϕ) and pitch (θ) can be calculated as follows:

$$\phi = \text{sign}(f_z) \sin^{-1} \frac{f_y}{g} \quad (3-8)$$

$$\theta = -\text{sign}(f_z) \sin^{-1} \frac{f_x}{g} \quad (3-9)$$

where $\text{sign}(\cdot)$ denotes the sign of the value. The sign is used in z-channel because the gravity errors is smaller than the biases in MEMS-based IMUs. For pig applications, the heading should be determined manually or by using top-grade MEMS-based IMU.

The fine alignment can be determined once the coarse alignment is achieved using extended Kalman filter (EKF) with small heading uncertainty (SHU) model.

Chapter Four : ESTIMATION IN NAVIGATION

Estimation is the process of finding an approximation for a usable application. Various types of estimation applications have been studied in previous decades. In navigation field, different estimation techniques have been used and experimented. Kalman Filter (KF) (Gelb, Kasper, Nash, Price, & Sutherland, 1974; Grewal & Andrews, 2001; Grewal, Weill, & Andrews, 2007; Kalman, 1960; Rogers, 2007) with all other different variations were the most successful estimation techniques used. In this chapter, Extended Kalman filter (EKF) will be briefly introduced. For more details about KF with its variations, the reader can refer to (Brown & Hwang, 1997; Chris Goodall & El-Sheimy, 2007; Cossaboom, Georgy, Karamat, & Noureldin, 2012; Gelb et al., 1974; Grewal & Andrews, 2001; Jazwinski, 1970; Petovello, Cannon, & Lachapelle, 2003; Wu & Chen, 1999).

4.1 Introduction

The EKF simply applies the Taylor series expansion for the nonlinear system along with observation equations, and takes terms to the first order, where probability density function (PDF) is approximated by a Gaussian distribution (Gelb et al., 1974; Gordon, Salmond, & Smith, 1993; Grewal & Andrews, 2001; Peter, 1979). However, in practice, EKF has shown several limitations. As per (Julier, Uhlmann, & Durrant-Whyte, 2000), the derivation of the Jacobian matrix is nontrivial for both the system and the observation equations in most applications which leads to significant implementation difficulties. Different styles of filters can be modeled based on using distinct error models such as (large heading uncertainties – LHU, small heading uncertainties – SHU, phi-angle error model, and psi-angle error model) (Benson & O., 1975; Scherzinger, 1996). Therefore, filter designers should be aware of critical errors could occur in case of selecting an

inappropriate error model. Furthermore, in EKF, only small errors are allowed to be delivered to the filter (Sukkarieh, 2000); otherwise, the first order approximation can cause biased solution which will lead to an inconsistency of the covariance update and filter instability in the presence of nonlinear error behavior (Lerro & Bar-Shalom, 1993).

The minimum variance estimator is used in Kalman filter in the navigation field. It can be simply defined through the use of conditional expectation (Meditch, 1969):

$$\hat{x}_{k|i} = E[x_k | z_1, z_2, \dots, z_i] \quad (4-1)$$

where $E[.]$ is the expectation operator; x is a state vector; and z is the measurements vector. Based on the value of k and i , different scenarios can be defined:

1. $k > i \rightarrow$ Prediction.
2. $k = i \rightarrow$ Filtering.
3. $k < i \rightarrow$ Smoothing.

The state vector can be designed as a full state or error states. The error states vector can be considered as either the linearized Kalman filter (LKF), or the extended Kalman filter (EKF). The EKF considers an INS error control loop (feedback). Therefore, EKF corresponds to the closed-loop filter configuration where the fed back errors are used to correct the system output as shown in Figure 4-1.

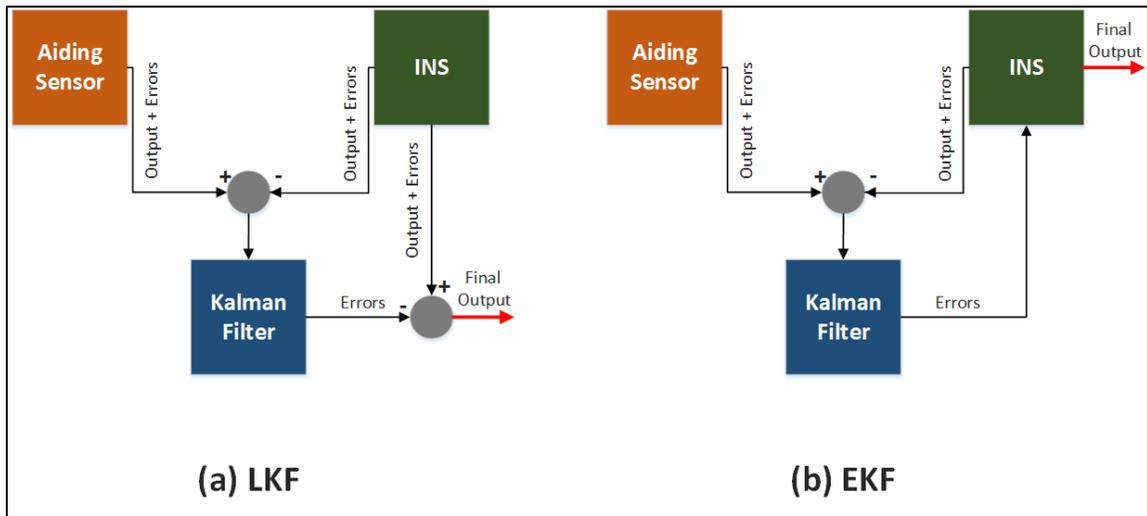


Figure 4-1: Block Diagrams (LKF vs EKF)

Due to the uncertainties in the INS sensors, errors in the navigation parameters will be generated using the INS mechanization equations, Eq. (2-40). Different models have been developed in the literature to describe the time-dependent behavior of these errors. There are two main approaches have been derived of INS error models (Bar-Itzhack & Berman, 1988). One is the true frame approach and called as the phi-angle approach. The second one is the computer frame approach and called as psi-angle approach. Both approaches are reviewed briefly in section 4.3. However, both approach models use the small angle assumption.

4.2 Kalman Filtering

GNSS, distance measurement sensors (odometers and/or speedometers), and inertial navigators are complementary to one another; thus it makes logical sense to combine their measurements. One of the pros of INS that it can provide a continuous solution at a high data rate. However, its errors are time dependent. On the other hand, GNSS provides time-dependent measurements with

accurate time standard, but GNSS is sensitive to signal blockage and radio frequencies (RF) interference (Groves, 2008). Moreover, GNSS provides lower data rate. Distance measurement sensors can provide higher data rate than GNSS. However, they provide only along-track speed.

To minimize inertial error accumulation with time, distance and/or GNSS measurements can be applied. A filtering technique needs to be employed to optimally combine all of the different measurements.

Dynamic inertial error state models can be derived for various error states. The general common states for EKF include position errors, velocity errors, attitude errors, gyroscopes biases errors, accelerometers biases errors, gyroscope scale factors errors, and accelerometer scale factors errors. The dynamic model can be defined within the Kalman filter to predict the next step. The update from either the GNSS or distance measurement sensor is used to correct the prediction and constrain the growth of the initial errors. Of course, a weighting scheme will be applied in the filter based on the errors from both combined measurements.

Since Kalman filter equations are well documented in the past few decades (Brown & Hwang, 1997; Gelb et al., 1974; Grewal & Andrews, 2001; Kalman, 1960), detailed derivation will not be reviewed here. A short review will prove useful of this filter in next chapters. A simple recursive block diagram with EKF equations is provided in Figure 4-2. The noises terms in Kalman filter are considered to be white sequences with known covariance. The state vector is dependent on the architecture that is used to combine GNSS and/or distance measurement with INS measurements. This is normally referred as coupling. In theory, we there are three methods of coupling: loose, tight, and deep (Goodall, 2009). In this thesis, we are interested in loosely couple method only. For more details about other types of coupling, please refer to (Goodall, 2009).

\mathbf{x} :	State Vector
P :	Covariance Matrix
Φ :	Transition Matrix that uses the dynamic model to predict the next state
Q :	Spectral density matrix that defines the noise of the dynamic model
K :	Kalman gain matrix that weights the predictions with the measurement noise
R :	Noise of the measurements matrix that defines the noise of the measurements
H :	Design matrix that defines the noiseless connection between the measurement and state vector
$\hat{\mathbf{x}}_k^-$:	Subscript denotes time epoch
$\hat{\mathbf{x}}_k^+$:	Superscript denotes prediction
$\hat{\mathbf{x}}_k^+$:	Superscript denotes update
$\hat{\mathbf{x}}_k$:	Denotes estimation

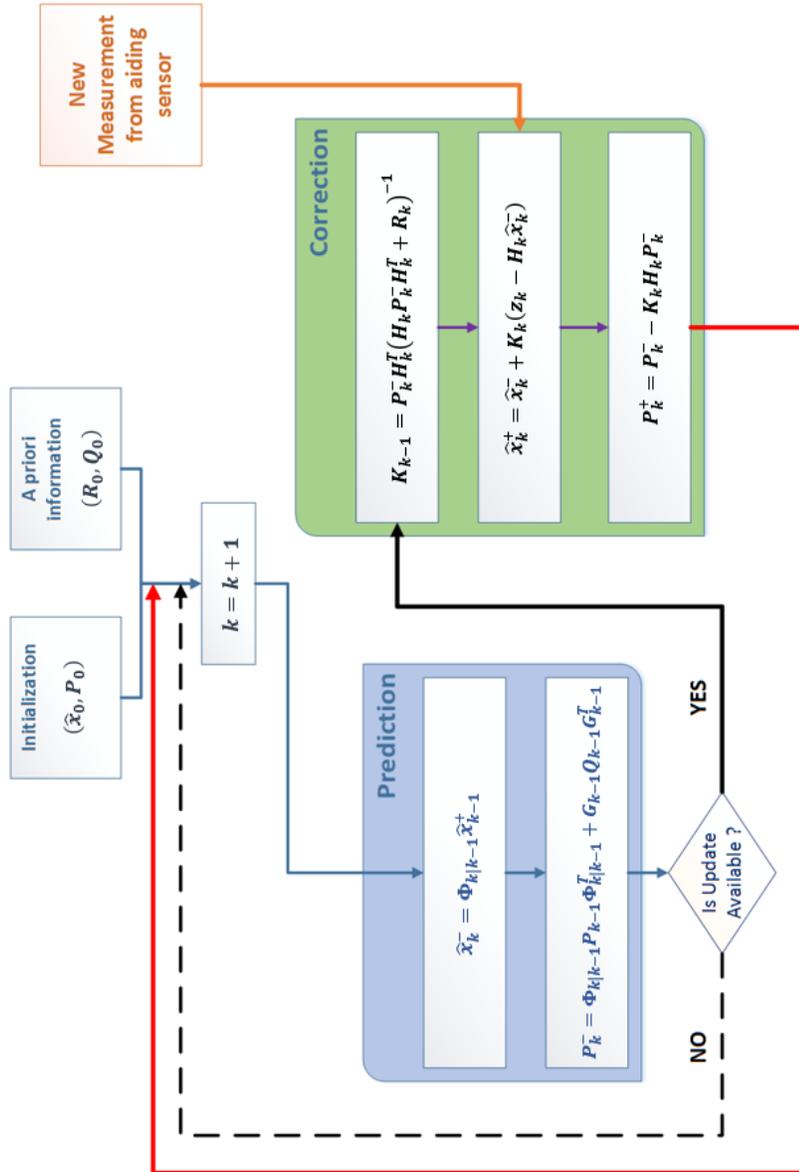


Figure 4-2: Kalman Filter Algorithm Flow Diagram

There are many advantages for loosely coupling method: INS errors are bounded by the updates, INS can be used to bridge the updates, and the update can be useful in inertial errors (deterministic parts) online calibration (Goodall, 2009). The loosely coupled integration strategy, using position update and along-track velocity update using GNSS and odometer, respectively, is shown in Figure 4-3.

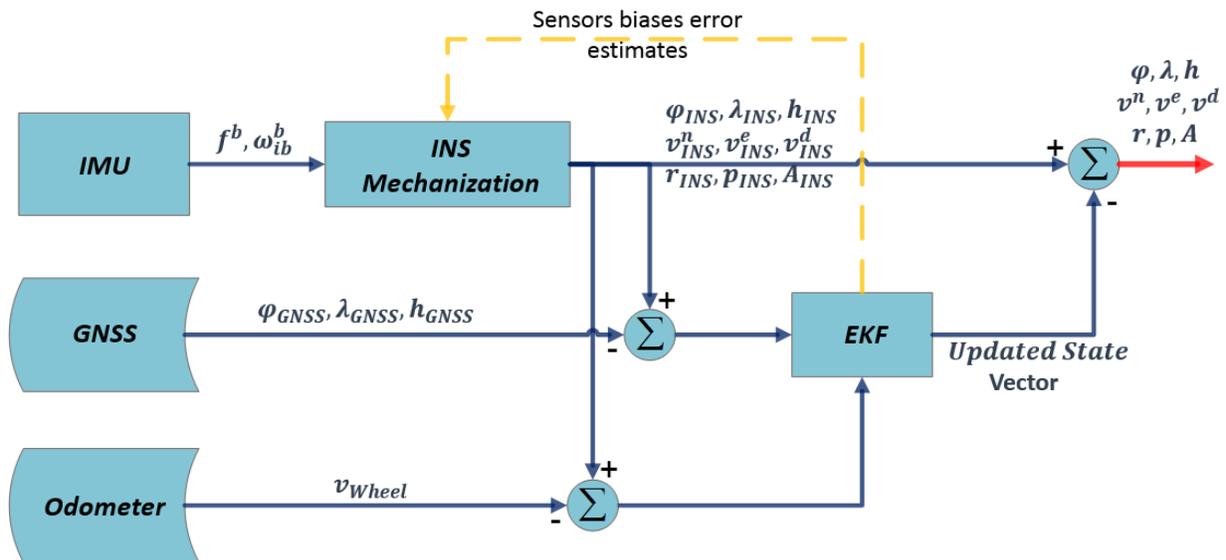


Figure 4-3: Loosely Coupled Architecture using EKF

4.3 INS Error Models Review

Error models are developed by perturbing the nominal differential equations whose solution yields the INS output of position, velocity, and attitude. These equations are based in Newton's law. In 1992, Bar-Itzhack and Goshen-Meskin published a systematic methodology and a unified

approach for INS error models development (Goshen-Meskin & Bar-Itzhack, 1992). Bar-Itzhack and Berman approached the analysis of INS from a control theory point of view (Bar-Itzhack & Berman, 1988) in 1988. Two types of differential equations that describe the behavior of INS error have been described in their paper; i.e. the propagation of the attitude errors and the propagation of translatory errors. Both types can be summarized in two different ways as follows:

1. Attitude errors differential equations depend on whether the equation variables are components of:
 - a. Body to computer frame (Psi-angle approach).
 - b. Body to true-frame (Phi-angle approach).
2. Translatory equations depend on whether the equations are:
 - a. Position error components.
 - b. Velocity error components.

Phi-angle and psi-angle approaches assume small INS attitude errors. Both gyrocompassing and analytic-coarse are the main principles of INS alignment which are based on the earth rate and gravity vectors. Therefore, because of the ability of high-end tactical grade IMUs to measure the earth rate, they are suitable to be used in these types of alignments. On the other hand, low cost IMUs are not sensitive enough to measure the earth rate. Subsequently, navigation-aiding sensors are required to be employed to obtain the INS initial attitude.

Usually, the initial heading may have a large or completely unknown uncertainty. Therefore, large heading uncertainty (LHU) models have been developed (Scherzinger, 1996). Initially, LHU has been developed for in-motion alignment or for in-air alignment until the heading-error is

minimized to few degrees. Then normal fine alignment method for small heading uncertainty (SHU) can be applied. Similarly, for MEMS-based IMUs, large heading uncertainty can be used due to completely unknown heading even in stationary mode because of large gyro-compassing errors.

Two different approaches have been developed using LHU. First, Rogers (Rogers, 2007) used errors of trigonometric functions of the wander azimuth angle as a part of the state vector. In this approach, a very different error model is required in case of heading uncertainty falls below certain threshold. Second, (Scherzinger, 1996) used trigonometric functions of the heading error. Even though, this approach still requiring a switch in the attitude error's dynamics model, it provides a continuous transition. For MEMS-based IMU sensors, and with an absence of aiding navigation sensors, the heading error grows very fast in short time. Similarly, due to the poor observability of the heading in the vehicle (pig) constant speed, the heading error grow fast as well. So far the error model switch can be done in both directions, therefore, Scherzinger approach (Scherzinger, 1996) will be more appropriate to be reviewed.

4.3.1 Psi-Angle Error Model

Two different models have been developed in Scherzinger approach, modified geographic consistent (GC) model and modified platform consistent (PC) model. The different between both models depends upon the type of the aiding sensors. GC model is used in case of using GNSS as an aiding navigation sensor where the position is the main aiding update. However, PC model is used in case of using a body-referenced velocity sensor as an aiding update.

GC model has been implemented and tested in (Shin, 2005), and because our main aiding sensor is the odometer and/or speedometer, the modified PC model will be introduced here.

The navigation parameters in ψ -angle approach between two representations can be represented as follows:

$$\hat{v}^l = v^l + \delta v_1^l = v^c + \delta v_2^c \quad (4-2)$$

$$\hat{g}^l = g^l + \delta g_1^l = g^c + \delta g_2^c \quad (4-3)$$

$$\hat{\omega}_{ie}^l = \omega_{ie}^l + \delta \omega_{ie}^l = \omega_{ie}^c \quad (4-4)$$

$$\hat{\omega}_{il}^l = \omega_{il}^l + \delta \omega_{il}^l = \omega_{ic}^c \quad (4-5)$$

Because we know the position and the transport rate of the c-frame from the navigation computer (Scherzinger, 1996), R_c^l , ω_{ie}^c , and ω_{ic}^c are known without errors. The navigation parameters errors between two representations can be expressed as:

$$\delta v_1^l = \delta v_2^c - \delta \theta \times v^c \quad (4-6)$$

$$\delta g_1^l = \delta g_2^c - \delta \theta \times g^c \quad (4-7)$$

$$\delta \omega_{ie}^l = -\delta \theta \times \omega_{ie}^c \quad (4-8)$$

Since the gravity computation error can be expressed as (Rogers, 2007):

$$\delta g^l = \begin{bmatrix} 0 \\ 0 \\ \frac{2g\delta r_D}{R+h} \end{bmatrix} \quad (4-9)$$

And the misalignment of the c-frame with respect to the true local level frame is (Benson & O., 1975):

$$\delta\theta = \begin{bmatrix} \delta\lambda \cos \varphi \\ -\delta\varphi \\ -\delta\lambda \sin \varphi \end{bmatrix} \quad (4-10)$$

The gravity error in c-frame can be written using Eq. (4-7) as per (Scherzinger, 1996):

$$\delta g_2^c = \begin{bmatrix} \frac{-g\delta r_N}{R_M+h} \\ \frac{-g\delta r_E}{R_N+h} \\ \frac{2g\delta r_D}{R+h} \end{bmatrix} \approx \begin{bmatrix} -\omega_s^2 \delta r_N \\ -\omega_s^2 \delta r_E \\ 2\omega_s^2 \delta r_D \end{bmatrix} \quad (4-11)$$

where ω_s is the Schuler frequency and $R = \sqrt{R_M R_N}$ is the Gaussian mean earth radius of curvature.

The c-frame analysis resulted in the following error model (Scherzinger, 1996):

$$\delta \dot{r}^c = -\omega_{ec}^c \times \delta r^c + \delta v^c \quad (4-12)$$

$$\delta \dot{v}^c = f^c \times \psi - (\omega_{ie}^c + \omega_{ec}^c) \times \delta v^c + \delta g^c + R_b^p \delta f^b \quad (4-13)$$

$$\dot{\psi} = -(\omega_{ie}^c + \omega_{ec}^c) \times \psi - R_b^l \delta \omega_{ib}^b \quad (4-14)$$

The extended misalignment vector ψ_e is defined as follows:

$$\psi_e = \begin{bmatrix} \psi_x \\ \psi_y \\ \sin \psi_z \\ \cos \psi_z - 1 \end{bmatrix} \quad (4-15)$$

where $(\cos \psi_z - 1)$ and $(\sin \psi_z)$ are considered as a random constants when heading error is large (Scherzinger, 2004). The first three elements of Eq. (4-15) define the misalignment vector $\psi' = [\psi_x \ \psi_y \ \sin \psi_z]^T$. Using a simple algebraic expression, the matrix representation of the “extended cross-product type (-)” operator can be defined as follows:

$$A_{e-} \equiv \left[A \left| \begin{array}{c} -a_x \\ -a_y \\ 0 \end{array} \right. \right] \quad (4-16)$$

$$A_{e+} \equiv \left[A \left| \begin{array}{c} a_x \\ a_y \\ 0 \end{array} \right. \right]$$

where A is a skew-symmetric matrix of a vector $a = [a_x \ a_y \ a_z]^T$, then (Scherzinger, 1996) derived the modified PC error equations as follows:

$$\delta \dot{r}^c = -\omega_{ec}^c \times \delta r^c + \Delta v^c + V_{e-}^c \psi_e \quad (4-17)$$

$$\begin{aligned} \delta \dot{v} = & \delta f^p + \Delta g^c - \hat{g} \times \psi - (2\Omega^c + \rho_c^c) \times \delta v - \hat{v} \times (\Omega^c \times \psi) + \hat{v} \\ & \times \epsilon^p - f_{xy}(\hat{f}^p, \psi) + f_c(\psi, \hat{v}) + B \end{aligned} \quad (4-18)$$

$$\dot{\psi}_e = \begin{bmatrix} -(\Omega_{ic}^c)_{e+} \\ 0_{1 \times 4} \end{bmatrix} \psi_e + \begin{bmatrix} 0_{1 \times 2} \\ \omega_x \psi_y - \omega_y \psi_x \\ 0 \end{bmatrix} - \begin{bmatrix} R_b^p \\ 0_{1 \times 3} \end{bmatrix} \delta \omega_{ib}^b \quad (4-19)$$

where $\hat{v} = [\hat{v}_x \quad \hat{v}_y \quad \hat{v}_z]^T$. The nonlinear term $f_{xy}(\hat{f}^p, \psi)$ is not negligible and appears in the x and y components of the velocity error dynamics.

$$f_{xy}(\hat{f}^p, \psi) = \begin{bmatrix} (\Omega_y^c + \rho_y^c)v_y & -(\Omega_x^c + \rho_x^c)v_y & 0 \\ -(\Omega_y^c + \rho_y^c)v_x & (\Omega_x^c + \rho_x^c)v_x & 0 \\ 0 & 0 & 0 \end{bmatrix} \psi \quad (4-20)$$

$$B = \begin{bmatrix} -(3\Omega_y^c + 2\rho_y^c)\hat{v}_z \\ (3\Omega_x^c + 2\rho_x^c)\hat{v}_z \\ \Omega_x^c\hat{v}_y - \Omega_y^c\hat{v}_x \end{bmatrix} (\cos \psi_z - 1) \quad (4-21)$$

4.3.2 Phi-Angle Error Model

The phi-angle error model is the classical INS error analysis approach. The navigation parameters are perturbed with respect to the true local level frame as shown in Eq. (2-40). The basic assumption that all the errors are small enough. The derivation of psi-angle model has been described in many studies in the literatures; for instance, see (Britting & R., 1971; Farrell & Barth, 1998).

The position error vector in local level frame (NED) can be expressed as:

$$\delta r^l = \begin{bmatrix} \delta r_N \\ \delta r_E \\ \delta r_D \end{bmatrix} \quad (4-22)$$

Perturbations on other navigations parameters can be expressed as:

$$\hat{\omega}_{ib}^b = \omega_{ib}^b + \delta\omega_{ib}^b \quad (4-23)$$

$$\hat{f}^b = f^b + \delta f^b \quad (4-24)$$

$$\hat{R}_b^l = [I - \Psi]R_b^l \quad (4-25)$$

$$\hat{\omega}_{ie}^l = \omega_{ie}^l + \delta\omega_{ie}^l \quad (4-26)$$

$$\hat{\omega}_{il}^l = \omega_{il}^l + \delta\omega_{il}^l \quad (4-27)$$

$$\hat{v}^l = v^l + \delta v^l \quad (4-28)$$

$$\hat{g}^l = g^l + \delta g^l \quad (4-29)$$

Then the model perturbation model can be written as follows (Scherzinger, 2004):

$$\delta\dot{r}^l = -\omega_{el}^l \times \delta r^l + \delta\theta \times v^l + \delta v^l \quad (4-30)$$

$$\delta\dot{v}^l = R_b^l \delta f^l + R_b^l f^l \times \phi + \delta g^l - (\omega_{ie}^l + \omega_{il}^l) \times \delta v^l - (\delta\omega_{ie}^l + \delta\omega_{il}^l) \times v^l \quad (4-31)$$

$$\dot{\phi} = -\omega_{il}^l \times \phi + \delta\omega_{il}^l - R_b^l \delta\omega_{ib}^b \quad (4-32)$$

4.4 Random Processes

The gyroscope and accelerometers sensors errors of INS can be categorized to two different parts; the deterministic part that can be determined by calibration process and removed from the raw measurements data. Biases and scale factors are included in this category. Second the stochastic

part that is due the random variations of the sensor errors over time. Stochastic error can be modeled stochastically and then included in the INS error model to be estimated using the filters. The output of the MEMS-based IMUs contains large uncertainties. Hence, it is required to determine the sensor errors as much as possible through the calibration. On the other hand, calibration process increase manufactures cost significantly. Therefore, including sensor errors models in the state vector of the navigation filter may solve huge save many problems. Never the less, filter actual behavior depends on the dynamic of the vehicle.

4.4.1 Sensor Errors

Biases (b), scale factors (S), and non-orthogonality (γ) are the main sensor error known terms. Based on the application and the design engineer, either all of these terms should be included in the sensor error models, or parts of them. The most important part is the bias. Recalling Eq. (3-1) and Eq. (3-2), if it is only considered, then:

$$\begin{aligned}\delta\omega_{ib}^b &= b_g = [b_{gx} \quad b_{gy} \quad b_{gz}]^T \\ \delta f^b &= b_a = [b_{ax} \quad b_{ay} \quad b_{az}]^T\end{aligned}\tag{4-33}$$

where the subscripts (g, a) represent gyroscope and accelerometer, respectively.

For biases and scale factor error model:

$$\begin{aligned}\delta\omega_{ib}^b &= b_g + \text{diag}(\delta\omega_{ib}^b)S_g \\ \delta f^b &= b_a + \text{diag}(\delta f^b)S_a\end{aligned}\tag{4-34}$$

where $\text{diag}(\alpha)$ denotes:

$$diag(\alpha) = \begin{bmatrix} \alpha_x & 0 & 0 \\ 0 & \alpha_y & 0 \\ 0 & 0 & \alpha_z \end{bmatrix} \quad (4-35)$$

Including the non-orthogonality, Eq. (4-34) can be written as:

$$\begin{aligned} \delta\omega_{ib}^b &= b_g + diag(\delta\omega_{ib}^b)S_g + \Gamma_g\gamma_g \\ \delta f^b &= b_a + diag(\delta f^b)S_a + \Gamma_a\gamma_a \end{aligned} \quad (4-36)$$

where:

$$\Gamma_g = \begin{bmatrix} \omega_y & \omega_z & 0 & 0 & 0 & 0 \\ 0 & 0 & \omega_x & \omega_z & 0 & 0 \\ 0 & 0 & 0 & 0 & \omega_x & \omega_y \end{bmatrix} \quad (4-37)$$

$$\Gamma_a = \begin{bmatrix} f_y & f_z & 0 & 0 & 0 & 0 \\ 0 & 0 & f_x & f_z & 0 & 0 \\ 0 & 0 & 0 & 0 & f_x & f_y \end{bmatrix} \quad (4-38)$$

$$\begin{aligned} \delta\gamma_g &= [\gamma_{g,xy} \quad \gamma_{g,xz} \quad \gamma_{g,yx} \quad \gamma_{g,yz} \quad \gamma_{g,zx} \quad \gamma_{g,zy}] \\ \delta\gamma_a &= [\gamma_{a,xy} \quad \gamma_{a,xz} \quad \gamma_{a,yx} \quad \gamma_{a,yz} \quad \gamma_{a,zx} \quad \gamma_{a,zy}] \end{aligned} \quad (4-39)$$

4.4.2 Stochastic Process

The sensor errors must be modeled in order to include it in the state vector. Different general stochastic processes have been studied in literature; ex. White noise, random walk, random

constant, Gauss-Markov, ...etc. Selecting the best model requires an investigation of the behavior of the errors under the operational scenario of the given application. Sensor performance, working environment, and operation time are the main factors that affects model selection. As an example, for very short operation time, the errors can be modeled as random constant.

In this section, white noise, random walk, random constant, and Gauss-Markov stochastic models will be introduced briefly.

White Noise Process

By definition, a white noise process is one whose power spectral density is constant at all frequencies (Petovello, 2012). This can be written mathematically as follows:

$$S_{\omega n}(j\omega) = N \quad (4-40)$$

$$R_{\omega n}(\tau) = N\delta(\tau) \quad (4-41)$$

where the subscript ωn represent the white noise and $\delta(\cdot)$ represents the Dirac delta function. From Eq. (4-41), white noise is uncorrelated with itself except for at zero offset.

The white term has been derived from the concept of white light where all frequencies have equal power. This means such a process would have infinite power. Therefore, white noise is not physically realizable, but it is primarily as a useful and convenient approximation to a situation in which a distributing noise in a wideband compared with the bandwidth of the system (Gelb et al., 1974) and it can be useful to describe the noise in inertial sensors.

Random Constant Process

Random constant process can be thought as an output of integral process without any input, but with unknown initial condition. Therefore, it does not change with time but has an unknown value.

The continuous time system model can be expressed as follows:

$$\dot{x}(t) = 0 \quad (4-42)$$

The discrete system model can be expressed as follows:

$$x_{k+1} = x_k \quad (4-43)$$

In inertial navigation, random constant can be used as initial states an extended heading error states $(\sin \psi_z, \cos \psi_z - 1)$ of the large heading uncertainty models. Moreover, during calibration process, non-orthogonality of sensor triads can be dealt as random constant.

Random Walk Process

Random walk term has been driven from the concept of a person that randomly takes steps forward or backward with equal probability (Petovello, 2012). Therefore, it is by definition the integration of the white noise. Wiener or Brownian motion process (Brown & Hwang, 1997) assumes that the step length are normally distributed. The continuous time system can be expressed as follows:

$$\dot{x}(t) = w(t) \quad (4-44)$$

where w is the white Gaussian random noise process. The corresponding discrete time process is:

$$x_{k+1} = x_k + w_k \quad (4-45)$$

The mean square value is expressed as follow:

$$\begin{aligned}
 E\{x^2(t)\} &= E\left\{\int_0^t w(u)du \times \int_0^t w(v)dv\right\} \\
 &= E\left\{\int_0^t \int_0^t w(u)w(v)dudv\right\} \\
 &= \int_0^t \int_0^t E\{w(u)w(v)\}dudv
 \end{aligned} \tag{4-46}$$

Since $E\{w(u)w(v)\}$ is an auto-correlation function⁷, therefore:

$$\begin{aligned}
 E\{x^2(t)\} &= \int_0^t \int_0^t q\delta(u-v)dudv \\
 &= \int_0^t qdv = qt
 \end{aligned} \tag{4-47}$$

Eq. (4-47) shows that the mean square values grows linearly with time and the random walk processes are not stationary.

In inertial navigation sensors, the angular random walk (ARW) and velocity random walk (VRW) are the terms used to describe these effects.

⁷ Auto-correlation process for two white process is given by the Dirac delta function scaled by the noise spectral density, q .

Gauss-Markov (GM) Process

Gauss-Markov random process model is a stationary process and its stochastic auto-correlation function decays exponentially with time (Figure 4-4). The first ordered Gauss-Markov process can be expressed as:

$$R(\tau) = \sigma^2 e^{-\beta|\tau|} \quad (4-48)$$

where β is the reciprocal of the time constant τ , and σ is the temporal standard deviation of the process.

GM is useful in many engineering applications. It can describe many physical random processes with good approximation (Brown & Hwang, 1997). Most of the present inertial systems use first order Gauss-Markov process to model their residual sensor errors. GM process is used in INS filters to model slowly biases and scale factors because its capability to represent bounded uncertainty.

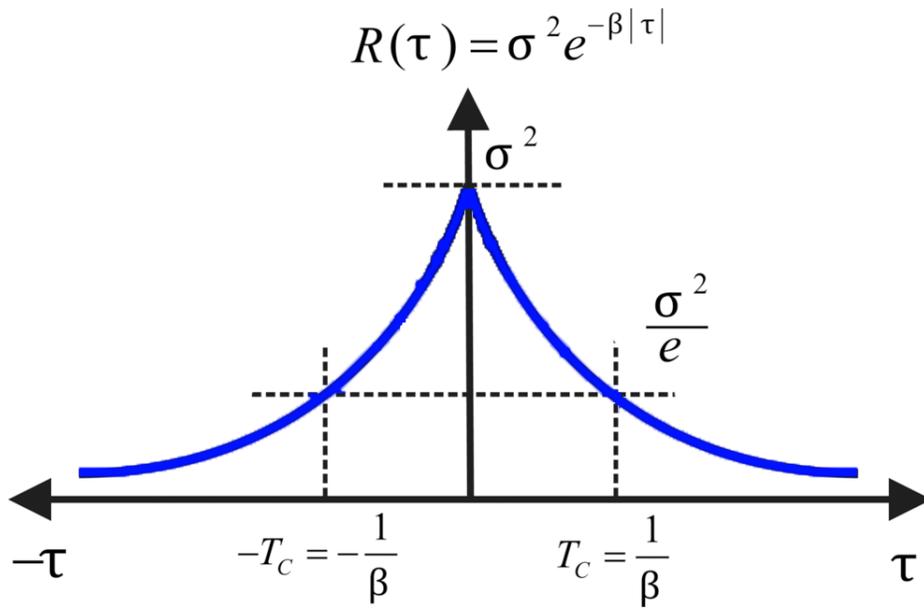


Figure 4-4: ACF of GM Process

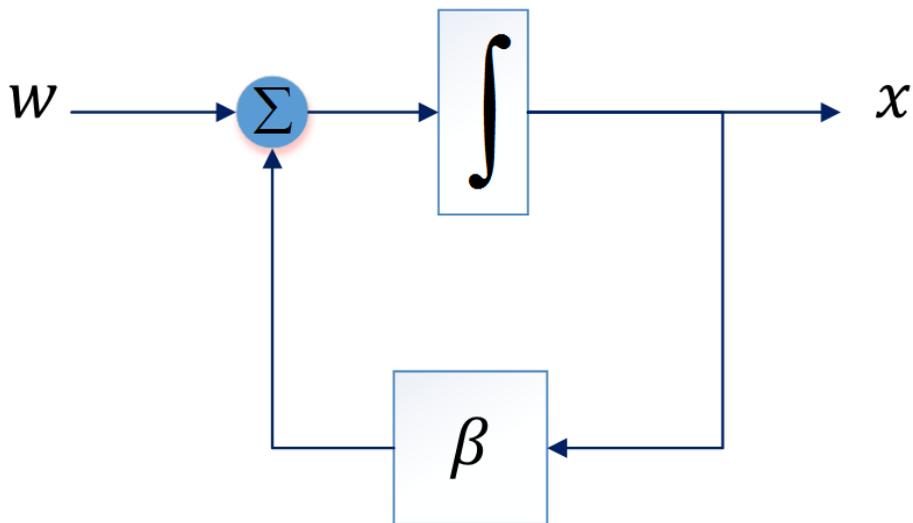


Figure 4-5: Block Diagram of First Order GM Process

From Figure 4-4, a random constant model will be formulated if $T_c \rightarrow \infty$, on the other hand, if $T_c \rightarrow 0$, then a white noise model will be formulated.

Finally, first-order Gauss-Markov processes are commonly used because they tend to adequately approximate a wide range of physical processes. Moreover, they are mathematically simple to work with. A block diagram of a first-order Gauss-Markov process is shown in Figure 4-5.

4.5 Backward Smoothing

The purpose of smoothing is to find an optimal estimate utilizing all past, current and future measurements (Shin, 2005). In other word, smoothing is estimating the states at some prior time based on all measurements taken up to the current time. Smoothing problems have been studied extensively in the literature and classified into three different types (e.g. Fixed-point, Fixed-lag, and Fixed-interval smoothing) (Meditch, 1969).

Combining both forward and backward filter solution can perform smoothing:

$$\mathbf{P}_{sm} = (\mathbf{P}_f^{-1} + \mathbf{P}_b^{-1})^{-1} \quad (4-49)$$

$$\hat{\mathbf{x}} = \hat{\mathbf{x}}_f + \mathbf{P}_{sm} \mathbf{P}_b^{-1} (\hat{\mathbf{x}}_b - \hat{\mathbf{x}}_f) \quad (4-50)$$

Where superscripts f, b and sm are the forward, backward and smoothing solution, respectively. Both equations (4-49) and (4-50) can be used not only for smoothing, but also for combining information coming from a multi-sensor network.

Rauch-Tung-Striebel (RTS) smoother is well known fixed-interval smoother. The algorithm of this smoother was written in (Brown & Hwang, 1997):

$$\delta\hat{\mathbf{x}}_{k|N} = \delta\hat{\mathbf{x}}_{k|k} + \mathbf{A}_k(\delta\hat{\mathbf{x}}_{k+1|N} - \delta\hat{\mathbf{x}}_{k+1|k}) \quad (4-51)$$

$$\mathbf{P}_{k|N} = \mathbf{P}_{k|k} + \mathbf{A}_k(\mathbf{P}_{k+1|N} - \mathbf{P}_{k+1|k})\mathbf{A}_k^T \quad (4-52)$$

$$\mathbf{A}_k = \mathbf{P}_{k|k}\boldsymbol{\Phi}_k^T\mathbf{P}_{k+1|k}^{-1} \quad (4-53)$$

where $k = N - 1, N - 2 \dots 0$, and N is the total number of measurements. The RTS smoother does not require full scale backward filter and the solution is equivalent of combining forward and backward solutions.

Chapter Five : PIPELINE NAVIGATION USING CONSTRAINTS

From previous chapters we noticed that inertial navigation errors grow unboundedly without any assumptions of the motion body and without any aiding sensors. This is the reason that INS cannot provide proper navigation solution as a standalone system for long period. However, the error growth can be limited if the navigation solution can be constrained. Constrain inertial navigation measurements could add some useful values to the system. As an example, forcing the velocity to be zero when the vehicle is not moving (static) will allow to reset the biases. Similarly, if we have a land vehicle (ex. Car, bus, truck ...etc.) that does not have a velocity in body frame with directions perpendicular to the forward motion, all perpendicular velocities can be reset to zero. This is called non-holonomic constraint in navigation society.

When designing a navigation system, it is often useful to consider the environment conditions that affect the motion body. In this chapter, new constraints based on pipeline structure for pig navigation will be reviewed. The effect of these constraints will be analyzed and compared with standard INS navigation using EKF.

Section 5.1 will introduce non-holonomic constraints. Pipeline junctions will be defined in section 5.2. Section 5.4 will discuss pipeline bend's detection technique. Pipeline junction algorithm will be introduced in section 5.5.

5.1 Non-Holonomic Constraints

For land vehicles, the motion can be governed by two non-holonomic constraints. Ideally, when the vehicle does not slide on or jump off the ground, the perpendicular velocities to the forward direction of the vehicle assumed to be zero. However, these assumptions are valid if engine

vibrations, suspension dynamics, and sideslip during cornering are ignored. Therefore, the approximated vertical and horizontal velocities model can be expressed as follows:

$$\begin{aligned} v_y^b &\approx 0 \\ v_z^b &\approx 0 \end{aligned} \tag{5-1}$$

where v_y^b and v_z^b are the horizontal and vertical velocities perpendicular to the forward direction, respectively.

The errors in these velocities can be expressed as:

$$\delta Z_k = \begin{bmatrix} \delta v_y^b \\ \delta v_z^b \end{bmatrix} = \begin{bmatrix} v_{y,INS}^b - v_{y,True}^b \\ v_{z,INS}^b - v_{z,True}^b \end{bmatrix} = \begin{bmatrix} v_{y,INS}^b \\ v_{z,INS}^b \end{bmatrix} \tag{5-2}$$

Since pig tool is tightly moves inside the pipeline, similar approach is applied, only forward motion is allowed. Therefore, non-holonomic constraints approach is applied for pig navigation.

5.2 Pipeline Junctions

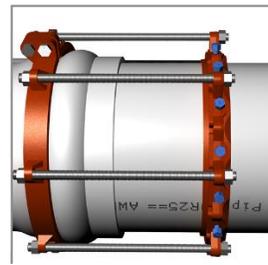
Pipeline is consisting of multiple pieces of pipelines and fittings (bends, T-connections, valves ...etc.). The pipeline pieces are fabricated in straight-line shapes. Different methods can be used to connect these pipeline pieces with each other, such as, push on, flange, and welding techniques as shown in Figure 5-1. In all cases, small gap tolerance will appear between two pieces.

The connection point between two pipelines is called pipeline junction / joint. These junctions can be detected using magnetic flux leakage (MFL) and electromagnetic acoustic transducers for pipeline analysis purposes. However, due to sudden vibration of the pig during the period of passing pipeline junctions, INS sensors are sensitive enough to capture these junctions. Sample

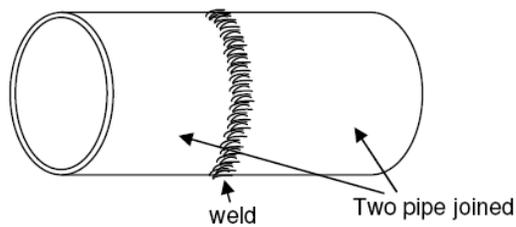
accelerometers pipeline data outputs for MEMS based IMU (SiIMU02) and high-end tactical grade IMU (LN200) are illustrated in Figure 5-2 and Figure 5-3. As shown, repetitive pattern spikes are shown. Taking into consideration the speed of the pig, the distance between two spikes is equal to the length of the fitting or the pipeline pieces. Therefore, the pattern represents the pipeline junction (PLJ). Such information is used as a new constraint to the estimation techniques (will be explained in Section 5.5.2). Figure 5-7 shows an illustration for pipeline junctions with bend fittings.



(a) Flange Type



(b) Push-On Type



(c) Welding Type

Figure 5-1: Pipeline Joints

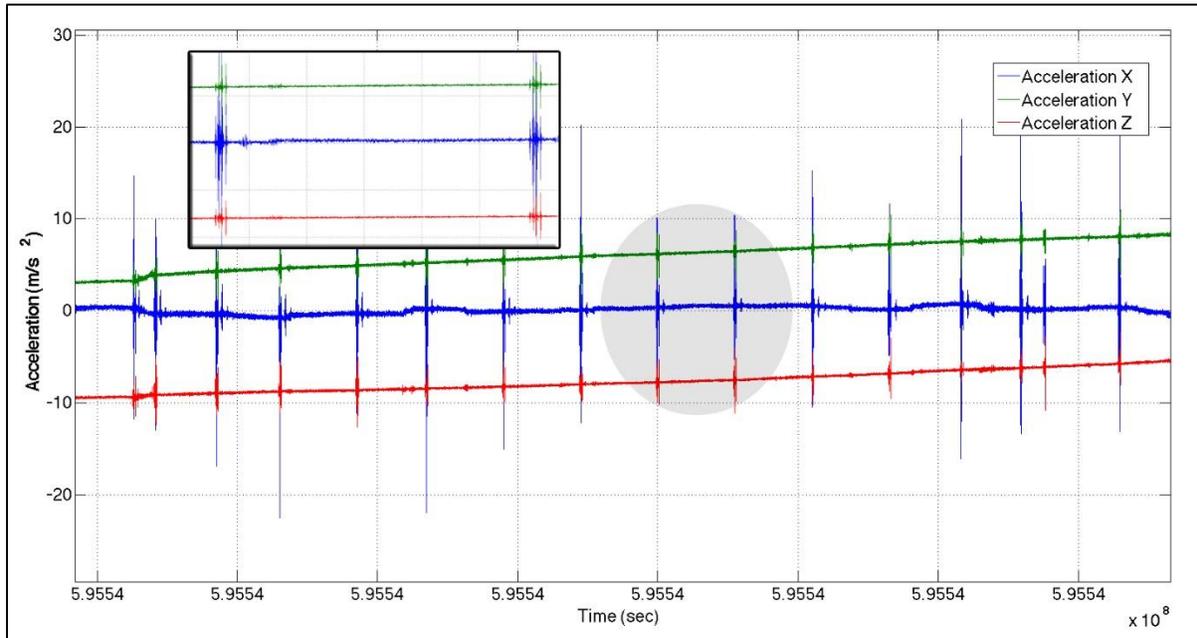


Figure 5-2: Accelerometer output - MEMS (SiIMU02)

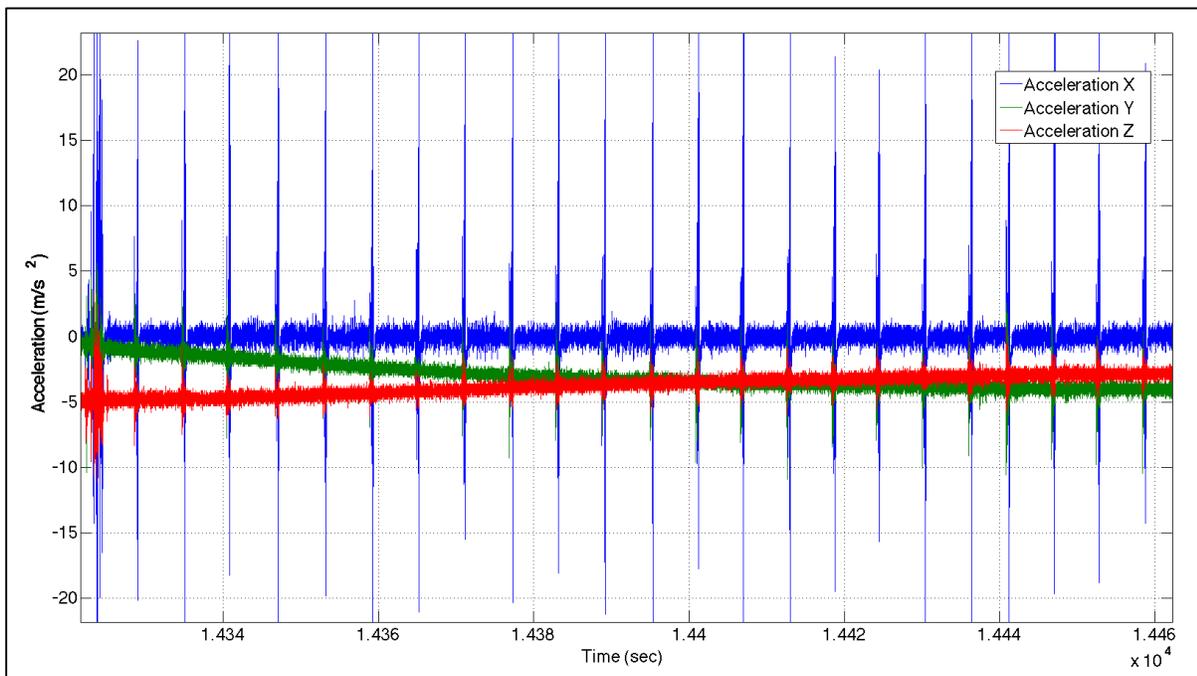


Figure 5-3: Accelerometer output - LN200

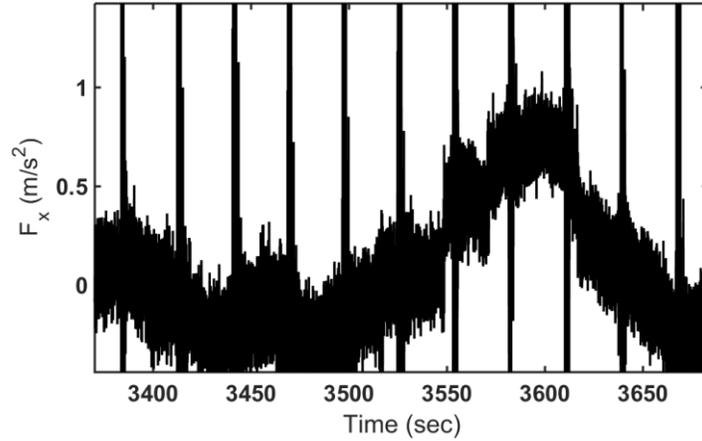


Figure 5-4: Pipeline Junctions – Sample

5.3 Detecting Pipeline Junctions

To detect the INS pipeline junctions signal spikes, different methods can be used (e.g. moving average, neural network (NN), wavelet transformation ...etc.). Two different methods have been applied in this section to detect pipeline junctions signal spikes.

5.3.1 Moving Average & Convolution

Moving average is a simple operation that can be used to suppress noise of a signal. This can be done by setting the value of value of each point to the average of the values in its neighborhood.

Mathematically:

$$y_k = \frac{x_{k-n} + x_{k-n+1} + \dots + x_k + \dots + x_{k+n-1} + x_{k+n}}{w} \quad (5-3)$$

$$n = \frac{w - 1}{2}$$

Where w is the size of the window (supposed to be odd number).

To detect the spikes, convolution method is applied on the accelerometer signal. The result of this process is subtracted from the original signal. The results of this subtraction should be compared with a threshold to define the spikes. Each spike is set to be the center of junction window with a width equal to the frequency of the data. Figure 5-5 shows a sample result of this algorithm.

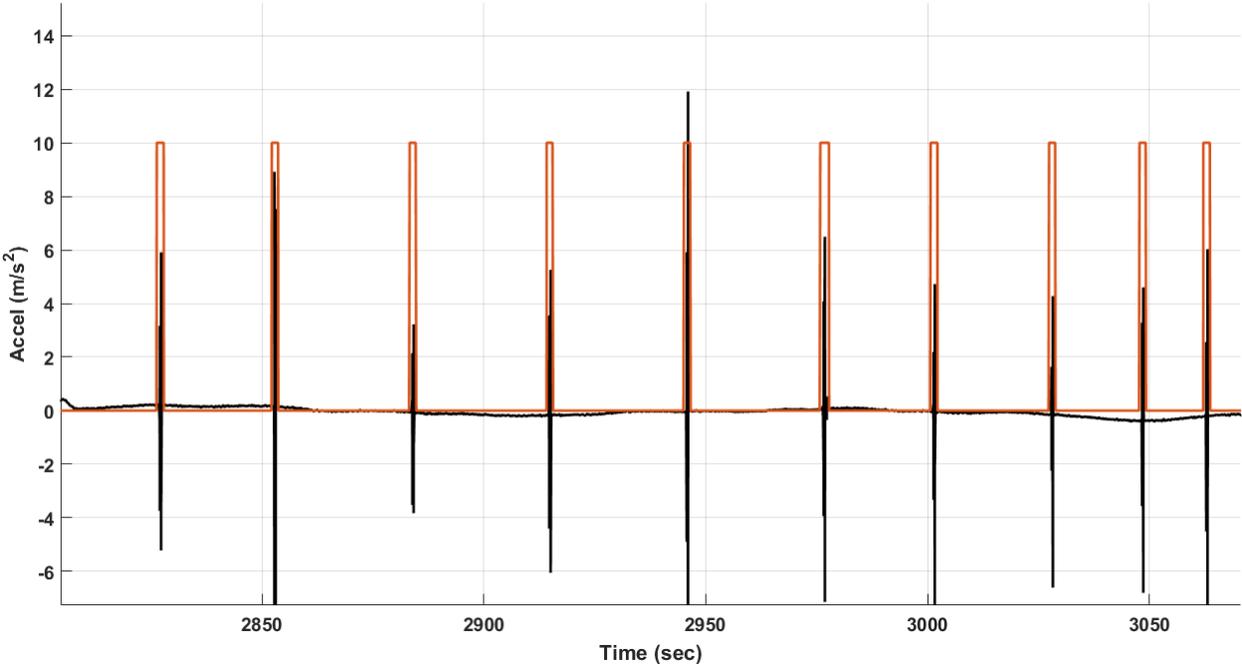


Figure 5-5: Junctions Detection Using Moving Average

5.3.2 Wavelet Transformation

For the simplicity of applying discrete wavelet transformation (DWT), this method is used in our application as well.

The definition of wavelet transform can be written as:

$$X_w(a, b) = \frac{1}{\sqrt{b}} \int x(t) M\left(\frac{t-a}{b}\right) dt \quad (5-4)$$

Where $x(\cdot)$ is the input, and $M(\cdot)$ is the mother wavelet. The parameters (a) is real number that represents a time location. (b) is a positive real number that represents the scaling.

Many types of mother wavelets can be used in signal pattern detection applications, such as Haar basis and Mexican Hat. Figure 5-6 illustrates the Mexican Hat mother wavelet that has the following model:

$$M(t) = \frac{2^{\frac{5}{4}}}{\sqrt{3}} (1 - 2\pi t^2) e^{-\pi t^2} \quad (5-5)$$

Mexican Hat mother wavelet has been applied to detect the signal pattern (pipeline junctions) as shown in Figure 5-8. For more details about the wavelet detection technique, please refer to (Graps, 1995; Mesa, 2005; Meyer, 1993; Vetterli & Herley; Wickerhauser, 1994).

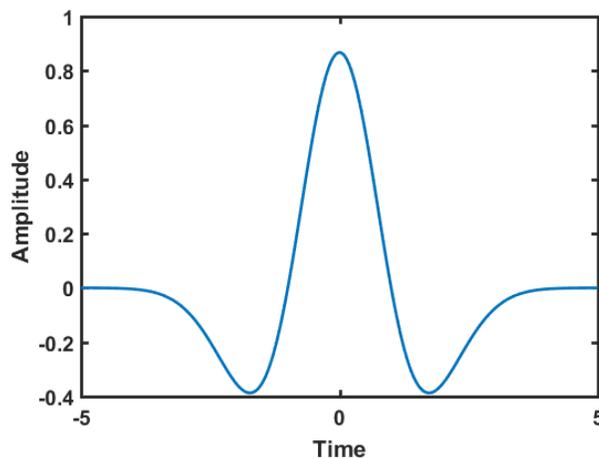


Figure 5-6: Mexican Hat Mother Wavelet



Figure 5-7: Pipeline Bends & Junctions

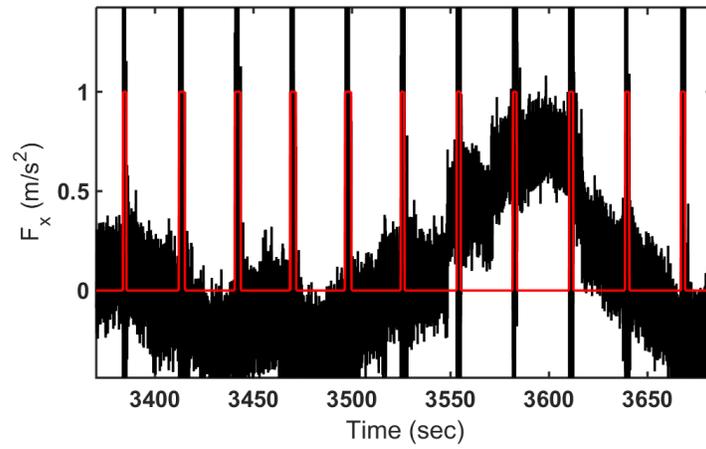


Figure 5-8: Detected Pattern Using WL

Since these patterns represent the pipeline junctions, a constraint is introduced here and called Pipeline Junction Constraint (PJC). This constraint fixes both the heading and pitch angles during the movement inside the pipeline piece (not during the junctions), (El-Sheimy, Sahli, & Moussa, 2015). New measurement model for PJC will be introduced in section 5.5.2. It is noticeable that roll angle is free to rotate along pipeline axis. Therefore, roll angle will not be included in PJC measurement model.

Until now, we assumed that the period between two junctions represents a straight pipe. However, due to the fittings (i.e. bends), this assumption cannot be considered true all the time. Therefore, the PJC model will be supported by bend detection algorithm (BDA) to detect the bends and disable PJC model during the bends periods. BDA is introduced in the next section.

Finally, it is good to mention that junction detection function should be executed prior to navigation offline process to save the pipeline junctions file. This file will be used as an input to the navigation offline process.

5.4 Bend Detection Algorithm (BDA)

Pig has three gyroscopes that measure and record the angular rates $(\omega_x, \omega_y, \omega_z)$ of its motion around three axes x, y, and z, respectively. In this thesis, the pig axes are defined as shown in Figure 5-9. The x-axis angular rate (ω_x) measures the rate change of the roll angle, while ω_y and ω_z measure the rate change of the pitch and heading angles, respectively.

To detect whether the pig is located inside a fitting (bend), the change in heading and/or pitch angles of the pig in motion should not exceed certain threshold (C_{th}) . The selection of this

threshold can be selected by sensor calibration. Different methods can be applied to measure the change in the heading and pitch angles.

5.4.1 Angular Velocity Magnitude

In this method, the change rate for both y and z axes are monitored. The magnitude of the change in these values can be compared to the selected threshold (C_{th}). For easier practice, both values are merged as follows:

$$\omega_R = \sqrt{\omega_y^2 + \omega_z^2} \quad (5-6)$$

Where ω_R is called a resultant angular rate which is equal to the norm of cross product of ω_y and ω_z .

Selecting the threshold is the most important part in this method. By plotting different resultant angular rates for different IMUs, it has been noticed that the best value to select (C_{th}) is the mean value of the static period. Checking the condition of this threshold should be done at every iteration and before applying the PJC constraints as shown in Figure 5-11.

Figure 5-10 illustrates the ω_R values that have been calculated before compensating for gyro biases and scale factors. Threshold (C_{th}) can be selected as the mean value of all resultant angular rates of ω_y and ω_z during the static period of the pig before it begins moving.

5.4.2 Mean of Heading & Pitch Angles History

Since we are dealing with offline system, data history can be more useful for bend detection. To detect the changes in the heading and pitch angles, a certain amount of heading and pitch angles

history can be saved regularly. Each set can be averaged and compared with the previous set. The differences in both average values (heading to heading and pitch to pitch) will help to classify the current location as a bend or not. In this method, the thresholds can be set as the integral value of the drift in the static mode as follows:

$$C_{\delta\theta} = \int b_w dt \quad (5-7)$$

The optimum solution can be obtained by combining both methods in detecting the pipeline bends.

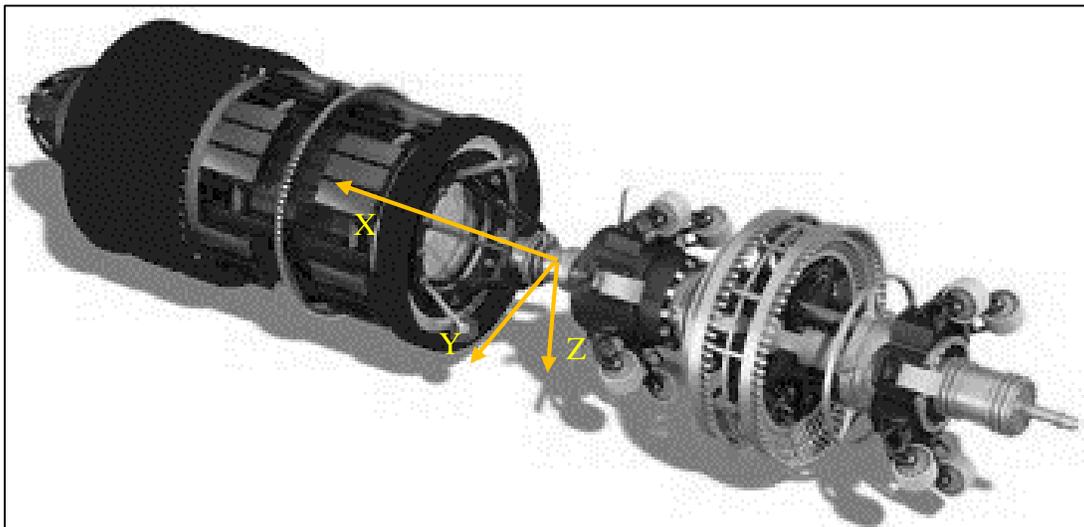


Figure 5-9: Pig Defined Axes

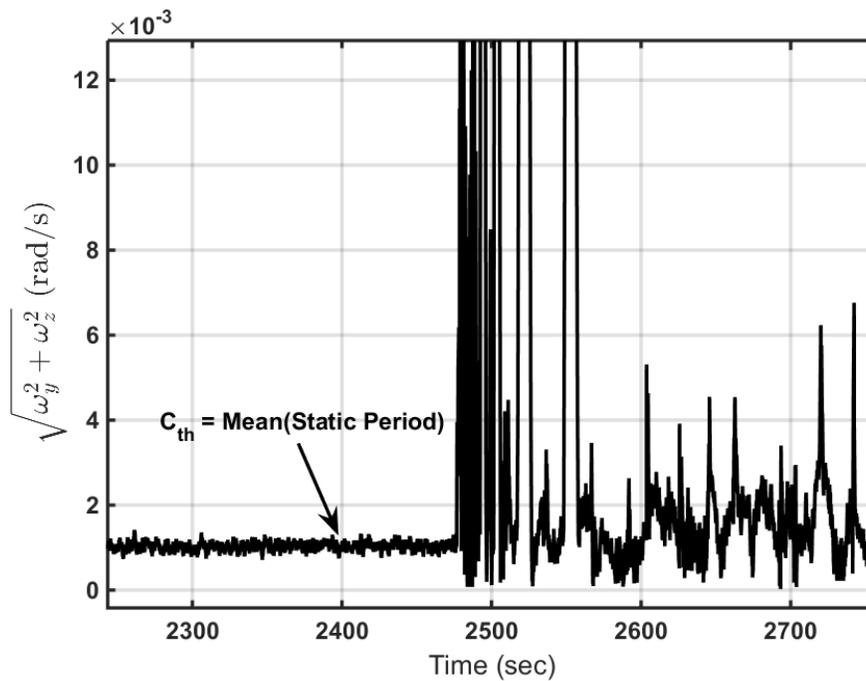


Figure 5-10: Selecting Threshold Criteria – Angular Velocity

5.5 Methodology

To cater for all three-dimensional dynamics of the pig motion, a total of six sensors are used in a full IMU, which comprises three gyroscopes and three accelerometers. In addition to the IMU, the odometer is used to measure the displaced travel distance of the pig. AGMs and their measurement model will be shown in this section, although the target is to use the fewest number of AGMs as possible.

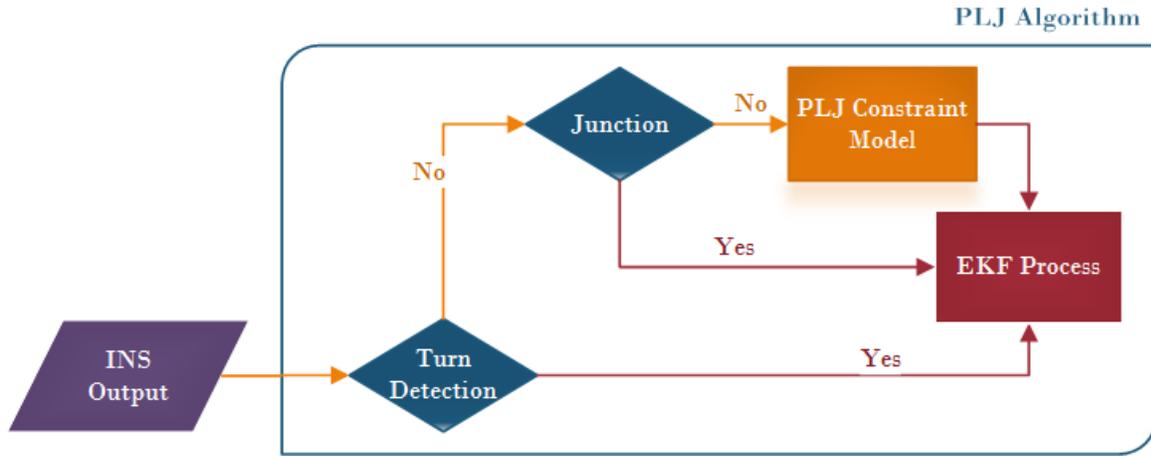


Figure 5-11: Algorithm for corrected state estimation

5.5.1 Dynamic Error Model

MEMS-based IMU was the main sensor used to collect the pig's motion data. EKF was used as an estimation technique to overcome the poor performance and non-linearity of the dynamic system in this work. Both dynamic and measurement models developed in this section to estimate, as accurately as possible, the states of the system. The state vector to be estimated was designed to include the errors associated with the position, velocity, and attitude. The stochastic bias errors associated with the gyroscopes and accelerometers are included as well. The state vector is defined as follows (Noureldin, Karamat, Eberts, & El-Shafie, 2009):

$$\delta x = [\delta r \ \delta v \ \delta \varepsilon \ \delta b_g \ \delta b_a]^T \quad (5-8)$$

where

δr : Position error vector (3×1)

δv : Velocity error vector (3×1)

$\delta \varepsilon$: Attitude error vector (3×1)

δb_g : Gyroscope bias error vector (3×1)

δb_a : Accelerometer bias error vector (3×1)

The dynamic model is non-linear and can be represented in discrete form as follows (El-Sheimy, 2012; Noureldin et al., 2012):

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{k}) + \mathbf{g}(\mathbf{x}_k, \mathbf{k})\mathbf{w}_k \quad (5-9)$$

where f is the dynamic model, g is the stochastic model, and w is the process noise.

The linearized error state system's model can be expressed as:

$$\delta \mathbf{x}_{k+1} = \Phi_k \delta \mathbf{x}_k + \mathbf{G}_k \mathbf{w}_k \quad (5-10)$$

where

$\delta \mathbf{x}_{k+1}$: is the (15×1) state vector

Φ_k : is the (15×15) transition matrix

\mathbf{G}_k : is the (15×1) noise distribution matrix

\mathbf{w}_k : is the unit variance white Gaussian noise

By applying Taylor series expansion and ignoring the higher order terms, the linearized system model in the local level frame (LLF), represented as North, East, and Down (NED), can be expressed as follows:

$$\delta x_{k+1} = \begin{bmatrix} I & F_1 & 0 & 0 & 0 \\ 0 & I & F_2 & 0 & F_3 \\ 0 & F_4 & I & F_3 & 0 \\ 0 & 0 & 0 & F_5 & 0 \\ 0 & 0 & 0 & 0 & F_6 \end{bmatrix} \begin{bmatrix} \delta r_k \\ \delta v_k \\ \delta \varepsilon \\ \delta b_g \\ \delta b_a \end{bmatrix} + \begin{bmatrix} \sigma_r \\ \sigma_v \\ \sigma_\varepsilon \\ F_7 \\ F_8 \end{bmatrix} w_k \quad (5-11)$$

where

$$\delta r_k = [\delta \phi_k, \delta \lambda_k, \delta h_k]^T, \quad \delta v_k = [\delta v_k^N, \delta v_k^E, \delta v_k^D]^T$$

$$\delta \varepsilon_k = [\delta \phi_k, \delta \theta_k, \delta \psi_k]^T, \quad \delta b_g = [\delta b_{gx}, \delta b_{gy}, \delta b_{gz}]^T$$

$$\delta b_a = [\delta b_{ax}, \delta b_{ay}, \delta b_{az}]^T$$

$$F_1 = \begin{bmatrix} \frac{1}{R_M + h} & 0 & 0 \\ 0 & \frac{1}{(R_N + h) \cos \varphi} & 0 \\ 0 & 0 & -1 \end{bmatrix} \cdot \Delta t$$

$$F_2 = \begin{bmatrix} 0 & f_u & -f_n \\ -f_u & 0 & f_e \\ f_n & -f_e & 0 \end{bmatrix} \cdot \Delta t$$

$$F_3 = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \cdot \Delta t$$

$$F_4 = \begin{bmatrix} \frac{1}{R_M + h} & 0 & 0 \\ 0 & \frac{-1}{R_N + h} & 0 \\ 0 & \frac{-\tan \varphi}{R_N + h} & 0 \end{bmatrix} \cdot \Delta t$$

$$F_5 = \begin{bmatrix} -\beta_{\omega x} & 0 & 0 \\ 0 & -\beta_{\omega y} & 0 \\ 0 & 0 & -\beta_{\omega z} \end{bmatrix} \cdot \Delta t$$

$$F_6 = \begin{bmatrix} -\beta_{fx} & 0 & 0 \\ 0 & -\beta_{fy} & 0 \\ 0 & 0 & -\beta_{fz} \end{bmatrix} \cdot \Delta t$$

$$\sigma_r = \begin{bmatrix} \sigma_\phi \\ \sigma_\lambda \\ \sigma_h \end{bmatrix}, \sigma_v = \begin{bmatrix} \sigma_{v^n} \\ \sigma_{v^e} \\ \sigma_{v^d} \end{bmatrix}, \sigma_\varepsilon = \begin{bmatrix} \sigma_\phi \\ \sigma_\theta \\ \sigma_\psi \end{bmatrix}$$

$$F_7 = \begin{bmatrix} \sqrt{2\beta_{\omega x}\sigma_{\omega x}^2} \\ \sqrt{2\beta_{\omega y}\sigma_{\omega y}^2} \\ \sqrt{2\beta_{\omega z}\sigma_{\omega z}^2} \end{bmatrix}, F_8 = \begin{bmatrix} \sqrt{2\beta_{fx}\sigma_{fx}^2} \\ \sqrt{2\beta_{fy}\sigma_{fy}^2} \\ \sqrt{2\beta_{fz}\sigma_{fz}^2} \end{bmatrix}$$

where

β : Reciprocal of the correlation time of the bias instability random process

σ^2 : Variance of the white noise associated with the random process

R_M : Meridian radius of curvature (North-South)

R_N : Prime vertical radius of curvature of the Earth's surface (East-West)

φ, λ, h : Latitude, longitude and height, respectively

f^n, f^e, f^d : Specific forces in east, north and up directions, respectively.

R_{ij} : Rotation matrix (R_b^l) elements from body to local level frame.

Δt : Rate change of time

In this thesis, a new measurement model has been developed specifically for pipeline navigation.

The mathematical equations will be demonstrated in the next section.

The linearized measurement error model can be expressed as:

$$\delta \mathbf{z}_k = \mathbf{H} \delta \mathbf{x}_k + \delta \mathbf{v}_k \quad (5-12)$$

where H is the design matrix and v is the measurement noise. Both process and measurement noises are assumed to be white and uncorrelated to each other. Readers can refer to (Shin, 2005) for more details about measurement models

5.5.2 Pitch & Heading Measurement Model

The attitude of the pig in the pipeline is computed from the elements of the following DCM (Sahli & El-Sheimy, 2016):

$$\hat{R}_v^l = \hat{R}_b^l (R_b^v)^T = [I - \Psi] R_b^l (R_b^v)^T = \begin{bmatrix} \hat{a}_{11} & \hat{a}_{12} & \hat{a}_{13} \\ \hat{a}_{21} & \hat{a}_{22} & \hat{a}_{23} \\ \hat{a}_{31} & \hat{a}_{32} & \hat{a}_{33} \end{bmatrix} \quad (5-13)$$

where (R_j^k) represents the rotation matrix or direct cosine matrix (DCM) from (j) to (k) frames. (Ψ) represents the skew-symmetric matrix of the rotation vector pertaining to the error of the attitude DCM and (b, l, v) represent body, local level (navigation), and vehicle (pig) frames, respectively.

$$\Psi = \begin{bmatrix} 0 & -\delta\psi & \delta\theta \\ \delta\psi & 0 & -\delta\phi \\ -\delta\theta & \delta\phi & 0 \end{bmatrix} \quad (5-14)$$

Let \hat{a}_{ij} , b_{ij} and c_{ij} represent that ij^{th} elements of \hat{R}_v^l , R_b^v , and R_b^l , respectively,

Where the elements of R_b^l can be expressed as follows:

$$c_{11} = \cos \theta \cos \psi$$

$$c_{12} = -\cos \phi \sin \psi + \sin \phi \sin \theta \cos \psi$$

$$c_{13} = \sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi$$

$$c_{21} = \cos \theta \sin \psi$$

$$c_{22} = \cos \phi \cos \psi + \sin \phi \sin \theta \sin \psi$$

$$c_{23} = -\sin \phi \cos \psi + \cos \phi \sin \theta \sin \psi$$

$$c_{31} = -\sin \theta$$

$$c_{32} = \sin \phi \cos \theta$$

$$c_{33} = \cos \phi \cos \theta$$

From (R_b^l) , the computed heading and pitch angles can be written as follows:

$$\hat{\psi} = \tan^{-1} \frac{\sin \hat{\psi}}{\cos \hat{\psi}} = \tan^{-1} \frac{\hat{a}_{21}}{\hat{a}_{11}} \quad (5-15)$$

$$\hat{\theta} = \tan^{-1} \frac{\sin \hat{\theta}}{\cos \hat{\theta}} = \tan^{-1} \frac{-\hat{a}_{31}}{\sqrt{\hat{a}_{32}^2 + \hat{a}_{33}^2}} \quad (5-16)$$

where

$$\hat{a}_{11} = b_{11}(c_{11} + c_{21}\delta\psi - c_{31}\delta\theta) + b_{12}(c_{12} + c_{22}\delta\psi - c_{32}\delta\theta) + b_{13}(c_{13} + c_{23}\delta\psi - c_{33}\delta\theta)$$

$$\hat{a}_{21} = b_{11}(c_{21} + c_{31}\delta\phi - c_{11}\delta\psi) + b_{12}(c_{22} + c_{32}\delta\phi - c_{12}\delta\psi) + b_{13}(c_{23} + c_{33}\delta\phi - c_{13}\delta\psi)$$

$$\hat{a}_{31} = b_{11}(c_{31} + c_{11}\delta\theta - c_{21}\delta\phi) + b_{12}(c_{32} + c_{12}\delta\theta - c_{22}\delta\phi) + b_{13}(c_{33} + c_{13}\delta\theta - c_{23}\delta\phi)$$

$$\hat{a}_{32} = b_{21}(c_{31} + c_{11}\delta\theta - c_{21}\delta\phi) + b_{22}(c_{32} + c_{12}\delta\theta - c_{22}\delta\phi) + b_{23}(c_{33} + c_{13}\delta\theta - c_{23}\delta\phi)$$

$$\hat{a}_{33} = b_{31}(c_{31} + c_{11}\delta\theta - c_{21}\delta\phi) + b_{32}(c_{32} + c_{12}\delta\theta - c_{22}\delta\phi) + b_{33}(c_{33} + c_{13}\delta\theta - c_{23}\delta\phi)$$

Ideally as per the algorithm assumption, the heading and pitch angles do not change in the pipeline piece; the change in these angles should be zero. Therefore, the approximated changes of heading and pitch angles model can be expressed as follows:

$$\delta z_{\theta,\psi}^v = \hat{\epsilon} - \tilde{\epsilon} \quad (5-17)$$

where $\hat{\epsilon}$ is the computed heading and pitch vector, and $\tilde{\epsilon}$ is the measured heading and pitch vector.

The measurement design matrix can be expressed as follows:

$$H_{p,A} = \begin{bmatrix} \frac{\partial \hat{\theta}}{\partial \delta\phi} & \frac{\partial \hat{\theta}}{\partial \delta\theta} & \frac{\partial \hat{\theta}}{\partial \delta\psi} \\ \frac{\partial \hat{\psi}}{\partial \delta\phi} & \frac{\partial \hat{\psi}}{\partial \delta\theta} & \frac{\partial \hat{\psi}}{\partial \delta\psi} \end{bmatrix} \quad (5-18)$$

Finally, the innovation sequence of EKF at each epoch can be calculated as follows:

$$e_k = \delta z_{\theta,\psi}^v - H_{\theta,\psi} \delta x \quad (5-19)$$

where δx_k represents the error state vector $[\delta\phi \quad \delta\theta \quad \delta\psi]^T$.

The elements of the first matrix can be calculated as follows:

$$\begin{aligned} & \frac{\partial \hat{\theta}}{\partial \delta\phi} \\ &= \frac{\left(\frac{EQ_5}{\sqrt{EQ_7}} - \frac{[2(b_{21}c_{21} + b_{22}c_{22} + b_{23}c_{23})EQ_9 + 2(b_{31}c_{21} + b_{32}c_{22} + b_{33}c_{23})EQ_8]EQ_4}{EQ_3} \right) EQ_7}{EQ_1} \end{aligned}$$

$$\begin{aligned} & \frac{\partial \hat{\theta}}{\partial \delta\theta} \\ &= - \frac{\left(\frac{EQ_6}{\sqrt{EQ_7}} - \frac{[2(b_{21}c_{11} + b_{22}c_{12} + b_{23}c_{13})EQ_9 + 2(b_{31}c_{11} + b_{32}c_{12} + b_{33}c_{13})EQ_8]EQ_4}{EQ_3} \right) EQ_7}{EQ_1} \end{aligned}$$

$$\frac{\partial \hat{\theta}}{\partial \delta\psi} = 0$$

$$\frac{\partial \hat{\psi}}{\partial \delta r} = \frac{(b_{11}c_{11} + b_{12}c_{12} + b_{13}c_{13})(b_{11}c_{31} + b_{12}c_{32} + b_{13}c_{33})}{(b_{11}c_{11} + b_{12}c_{12} + b_{13}c_{13})^2 + (b_{11}c_{21} + b_{12}c_{22} + b_{13}c_{23})^2}$$

$$\frac{\partial \hat{\psi}}{\partial \delta\theta} = \frac{(b_{11}c_{21} + b_{12}c_{22} + b_{13}c_{23})(b_{11}c_{31} + b_{12}c_{32} + b_{13}c_{33})}{(b_{11}c_{11} + b_{12}c_{12} + b_{13}c_{13})^2 + (b_{11}c_{21} + b_{12}c_{22} + b_{13}c_{23})^2}$$

$$\frac{\partial \hat{\psi}}{\partial \delta\psi} = - \frac{\left(\frac{(b_{11}c_{21} + b_{12}c_{22} + b_{13}c_{23})^2}{(b_{11}c_{11} + b_{12}c_{12} + b_{13}c_{13})^2} + 1 \right) (b_{11}c_{11} + b_{12}c_{12} + b_{13}c_{13})^2}{(b_{11}c_{11} + b_{12}c_{12} + b_{13}c_{13})^2 + (b_{11}c_{21} + b_{12}c_{22} + b_{13}c_{23})^2}$$

where:

$$EQ_1 = (b_{11}c_{31} + b_{12}c_{32} + b_{13}c_{33})^2 + (b_{21}c_{31} + b_{22}c_{32} + b_{23}c_{33})^2 \\ + (b_{31}c_{31} + b_{32}c_{32} + b_{33}c_{33})^2$$

$$EQ_2 = (b_{11}c_{11} + b_{12}c_{12} + b_{13}c_{13})^2 + (b_{11}c_{21} + b_{12}c_{22} + b_{13}c_{23})^2$$

$$EQ_3 = 2\sqrt{((b_{21}c_{31} + b_{22}c_{32} + b_{23}c_{33})^2 + (b_{31}c_{31} + b_{32}c_{32} + b_{33}c_{33})^2)^3}$$

$$EQ_4 = b_{11}c_{31} + b_{12}c_{32} + b_{13}c_{33}$$

$$EQ_5 = b_{11}c_{21} + b_{12}c_{22} + b_{13}c_{23}$$

$$EQ_6 = b_{11}c_{11} + b_{12}c_{12} + b_{13}c_{13}$$

$$EQ_7 = EQ_8^2 + EQ_9^2$$

$$EQ_8 = b_{31}c_{31} + b_{32}c_{32} + b_{33}c_{33}$$

$$EQ_9 = b_{21}c_{31} + b_{22}c_{32} + b_{23}c_{33}$$

Chapter Six : **EXPERIMENTAL RESULTS**

In this chapter, the performance of the PLJ/EKF approach and the EKF approach will be compared using two datasets collected from MEMS based IMU and FOG based IMU mounted in pipeline inspection gauge (pig). The results will be compared with true reference trajectory that has been made specifically for this research.

This chapter is divided into different sections. Section 6.1 describes the experiment environment. Forward-Backward smoothing results for MEMS based IMU is shown in section 6.2. Section 6.3, shows the experimental results to verify PLJ/EKF algorithm for the MEMS dataset with three different scenarios. In the first two scenarios, AGM point has been introduced in two different locations. In the last scenario, no AGM is used. In all scenarios, comparisons between both algorithms are shown (PLJ/EKF & EKF). Section 6.4 shows the experimental results for FOG based IMU (LN200).

6.1 Experiment Description (MEMS)

This section presents the experiment description to be analyzed in the next sections using different navigation algorithms developed in this thesis. The experimental platform consists of a low cost IMU (SiIMU02) aided by odometer. The dataset was collect by M/S ROSEN using a pipeline inspection gauge in pipeline environment. The standard pipeline piece length used in this project is 24 *m*.

The IMU used in this experiment is a strapdown inertial measurement unit ("MEMS Inertial Measurement Unit - SiIMU02 Datasheet," 2010) which contains three gyroscopes and three accelerometers. The IMU is installed directly in the pig. The outputs of the IMU are in the body

frame of the pig whose origin is defined at the IMU mass center. The pig body frame is defined as IMU body frame. The three accelerometers (x, y, and z) are installed in the (x, y, and z) axes of the body frame to measure the specific force of the body frame. The three gyroscopes (x, y, and z) are installed in the (x, y, and z) axes of the body frame to provide the angular rate of the pig with respect to body frame. The total trajectory length is around (3 km). The pig traveled the total distance in (1 hr) with average speed (0.8 m/s). The data sampling frequency is 125 Hz. The measurement range of the gyroscope (G_x) is ($\pm 9,000$ °/s), and for both (G_y, G_z) is (± 500 °/s). The pig trajectory in the experiment is shown in Figure 6-1. Pig starts the motion from the origin of the local level frame (0,0) after stationary period of (5 min). The starting point is marked in the figure as a red star.

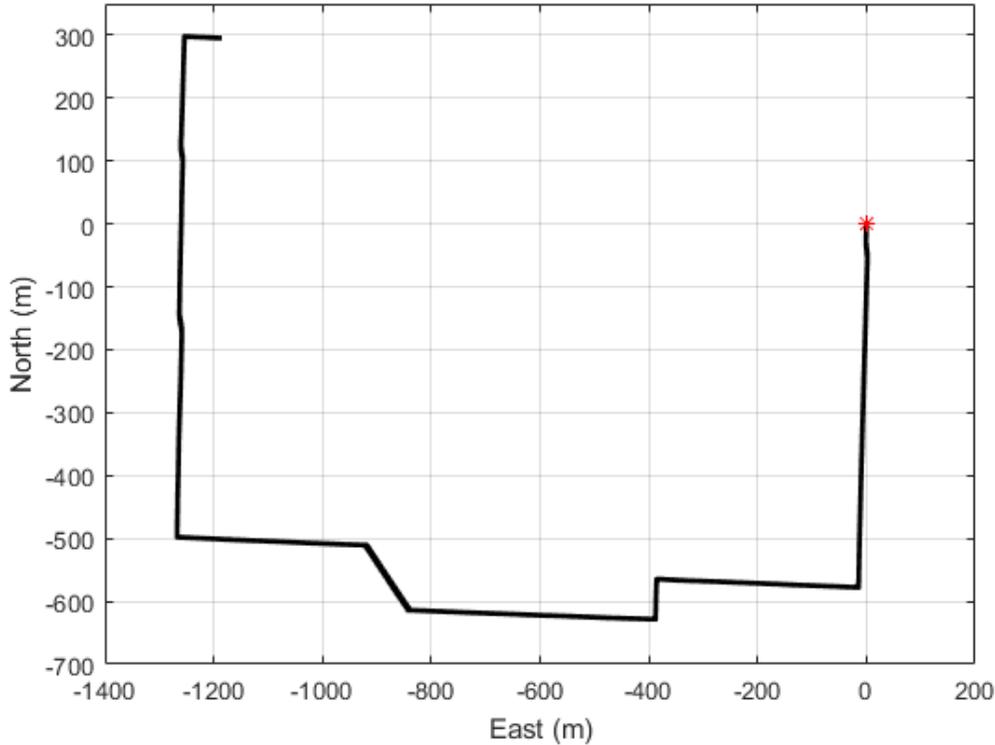


Figure 6-1: Pig Trajectory

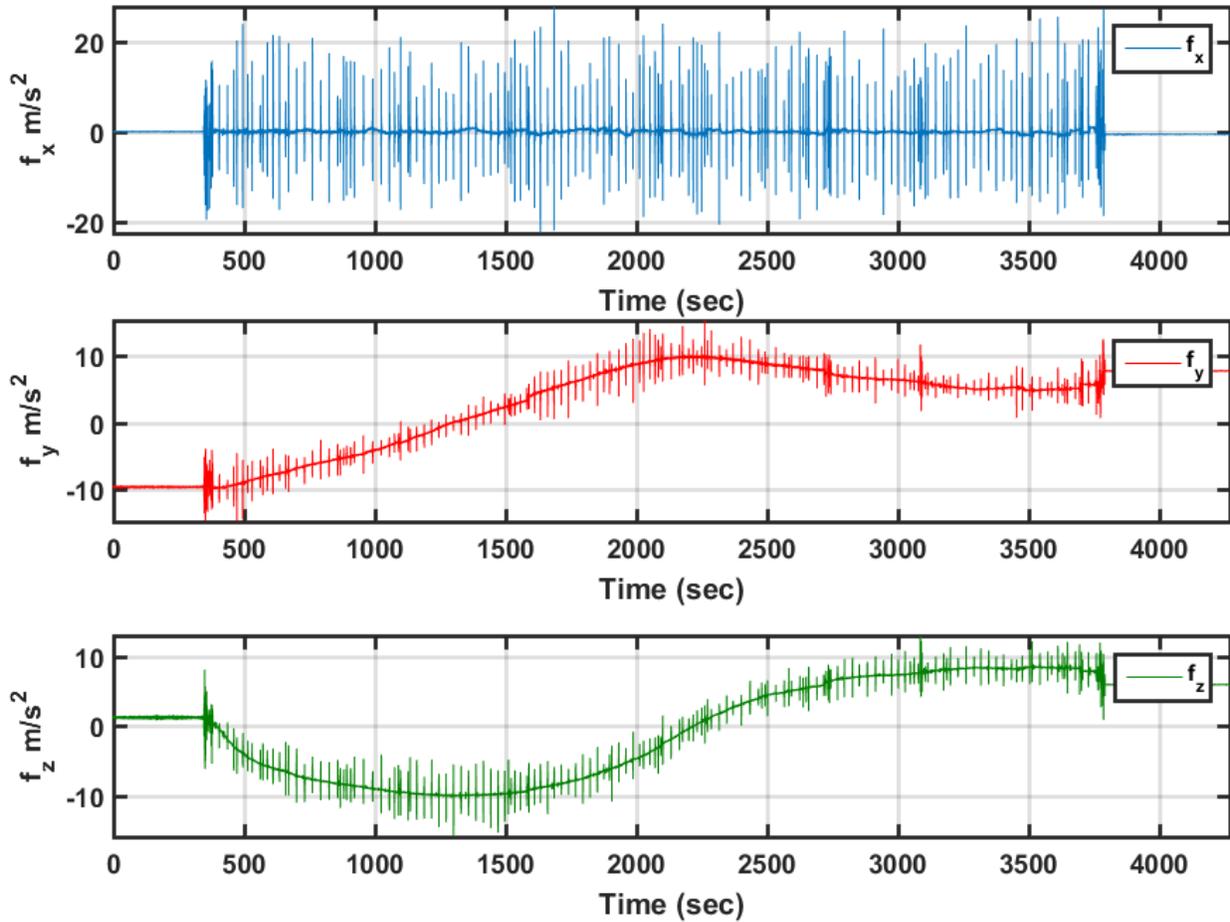


Figure 6-2: Accelerometers Output

The output of the three accelerometers and three gyroscopes after smoothing are shown in Figure 6-2 and Figure 6-3, respectively. The spikes in all figures represents the junctions of the pipeline. These spikes can be detected as shown in section 5.3. A sample of detected junctions are shown in Figure 6-4. From this figure, the time difference between two spikes multiplied by the average speed of the pig ($\approx 0.8 \text{ m/s}$) will be equal to pipe length $(2914 - 2883) \times 0.8 = 24 \text{ m}$.

6.1.1 Simulated AGM

The true reference trajectory has been provided for this pig experiment by M/s ROSEN. No Above Ground Markers (AGMs) have been used in this experiment. Therefore, a simulated AGM has been calculated to simulate the effect of this update coordinate on the system solution. Since the original solution is based on trajectory (latitude, longitude, and height) without any travelling time, the reference solution was calculated based on the distance traveled based on the odometer time and the distance traveled.

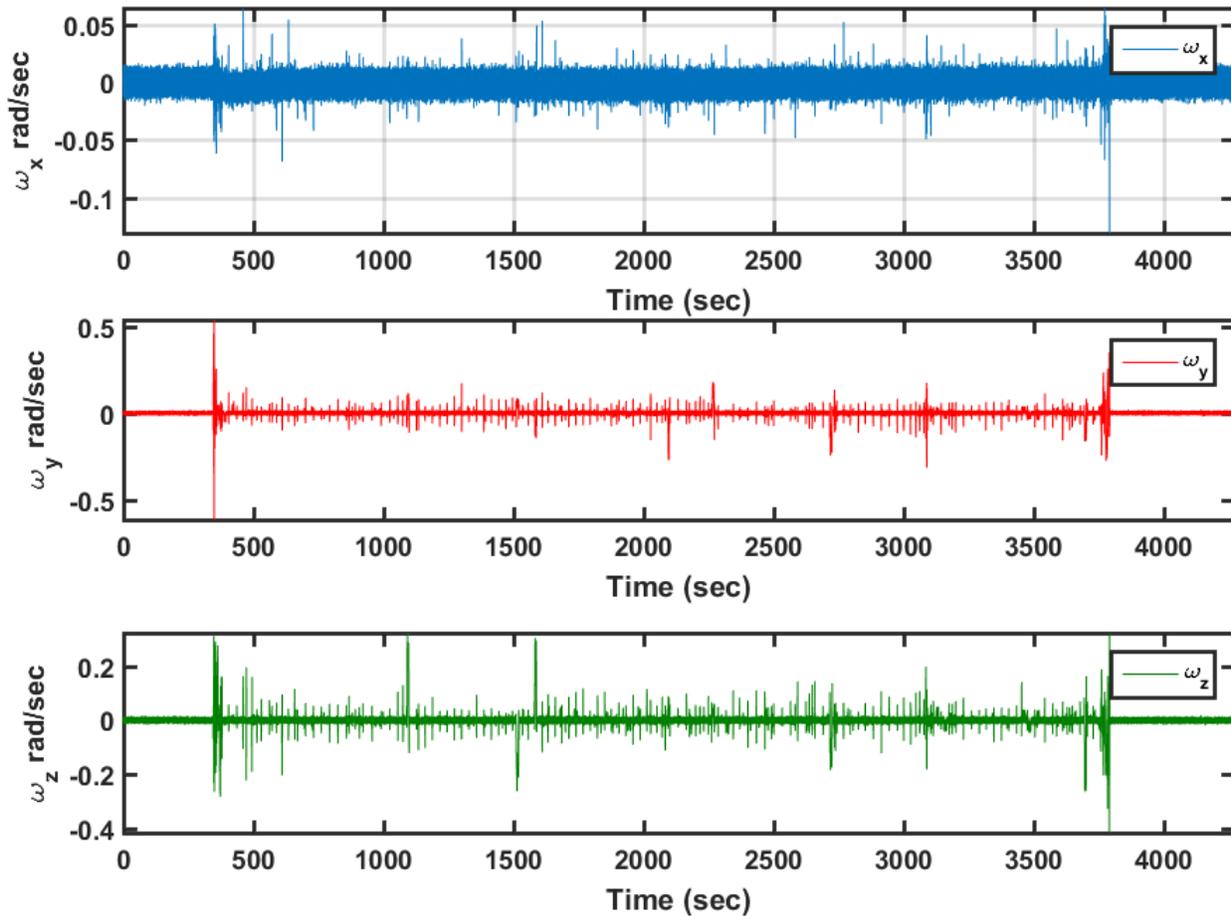


Figure 6-3: Gyroscopes Output

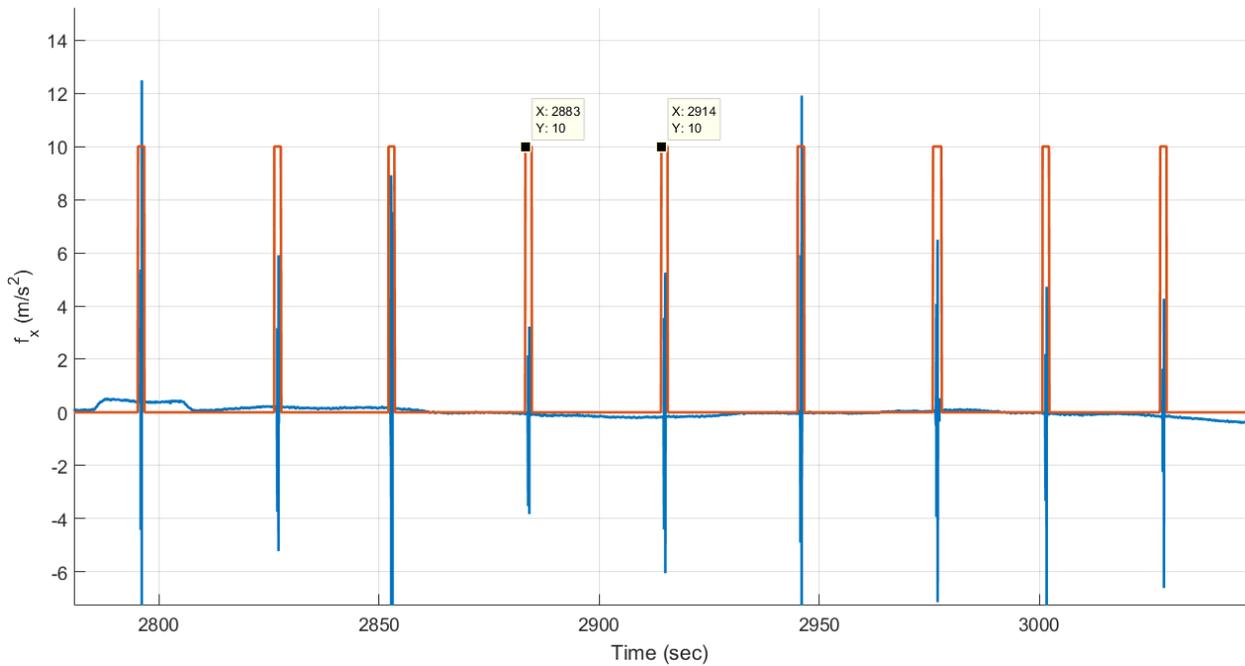


Figure 6-4: Detected Junctions

6.1.2 Adding time to original reference trajectory

Using the original reference (latitude, longitude, and height) along with the odometer output including its time stamp, a new reference trajectory has been generated that includes time stamp as shown in Figure 6-5.

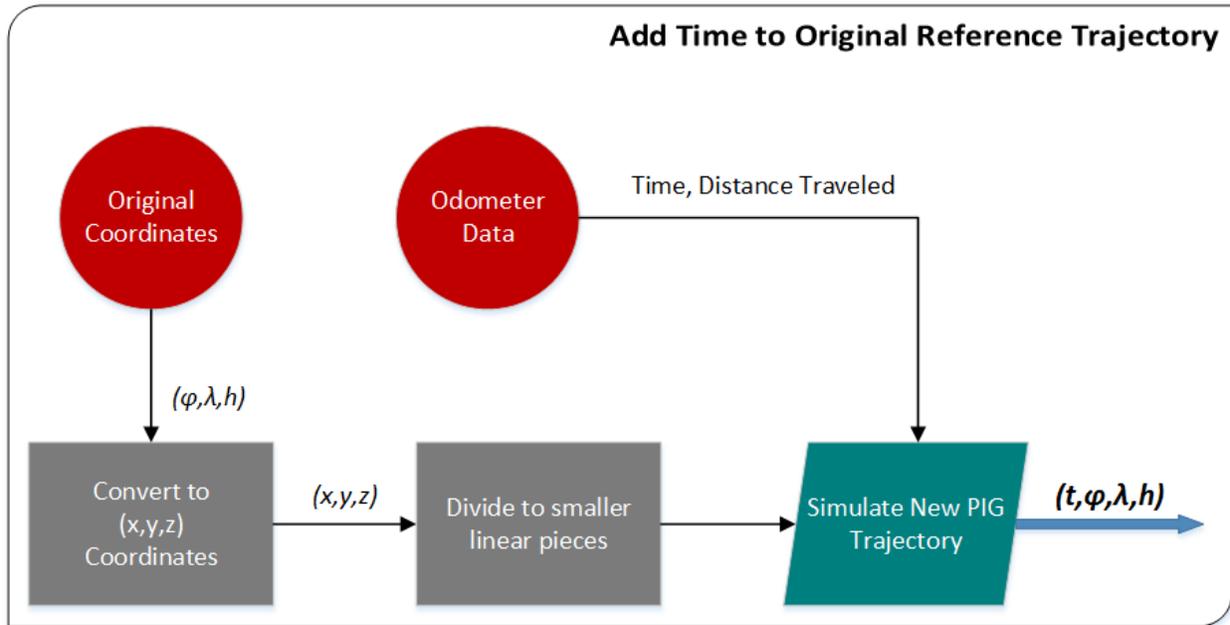


Figure 6-5: Add Time to Original Reference Trajectory

6.2 Forward-Backward Smoothing (MEMS)

In this section, results from Extended Kalman filter with Rauch-Tung-Striebel (RTS) smoother is applied to the dataset. Two different scenarios have been applied as shown below. These results have been discussed in (Sahli, 2014).

1. Scenario #1: Processed IMU & Odometer data using no AGM.
2. Scenario #2: Processed IMU & Odometer data using one AGM point.

The results can be summarized as shown in Table 6-1 and Table 6-2 , where North, East & Height RMS errors for both cases are presented.

Table 6-1: North, East & Height Errors - Filtered

RMS Error			
	North (m)	East (m)	Height (m)
Scenario #1	12.53	26.56	3.74
Scenario #2	10.70	27.50	2.20

Table 6-2: North, East & Height Errors – Smoothed

RMS Error			
	North (m)	East (m)	Height (m)
Scenario #1	1.76	6.77	1.65
Scenario #2	1.16	0.93	0.88

6.2.1 Scenario #1:

Figure 6-6 to Figure 6-9 represent the output results of data processed for scenario #1. In this scenario, no AGM points have been included. However, the starting and ending position are known. In Figure 6-7, we can notice the increments of the estimated errors in all different directions (NED) due to lack of position update. The maximum error in east direction increased to 80 m.

Figure 6-8 shows the errors in North, East, and Height directions, respectively.

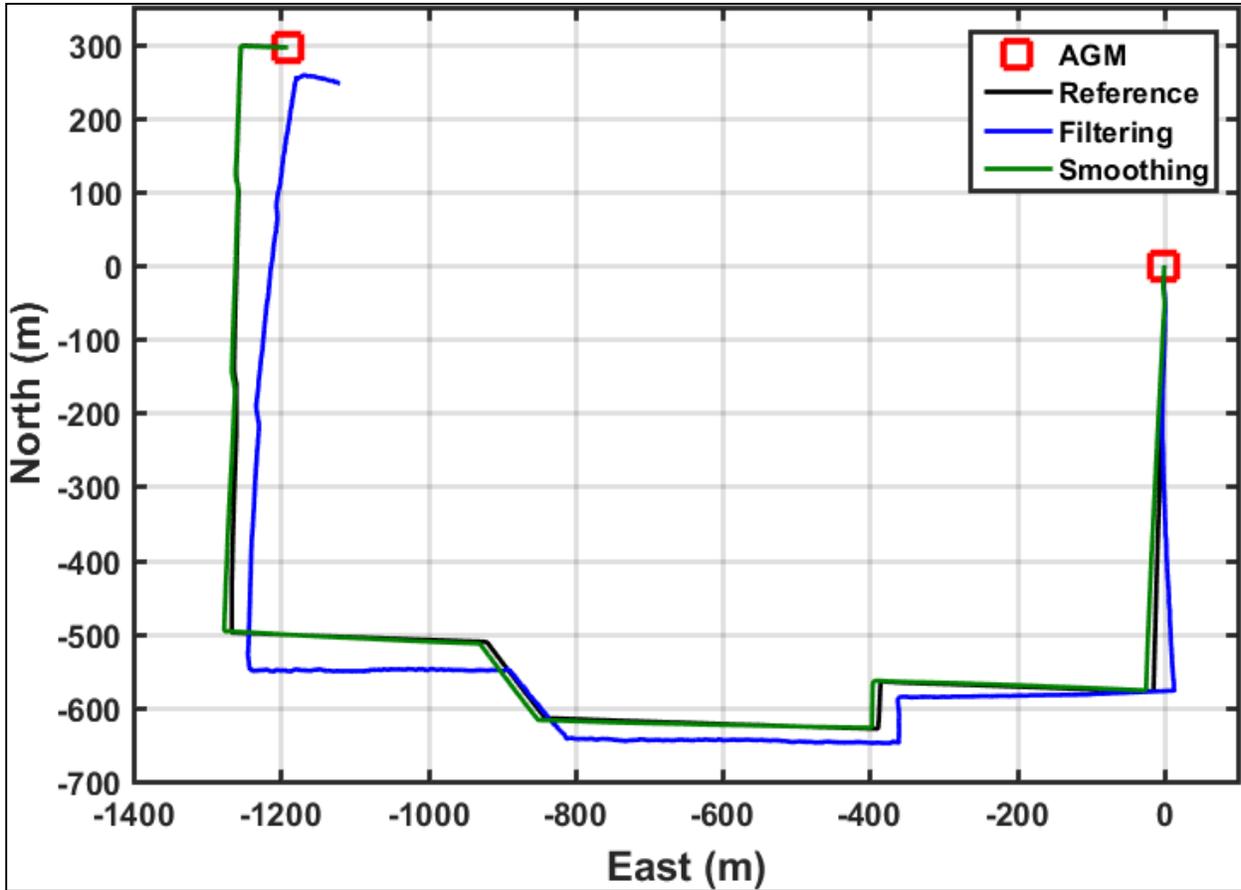


Figure 6-6: Full Trajectory (Scenario #1)

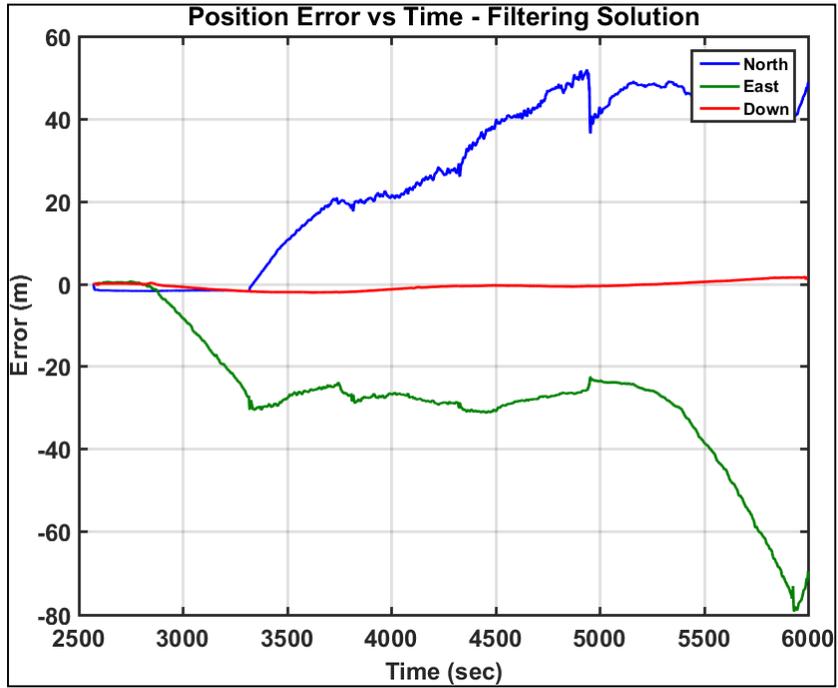


Figure 6-7: NED Error - Filtering (Scenario #1)

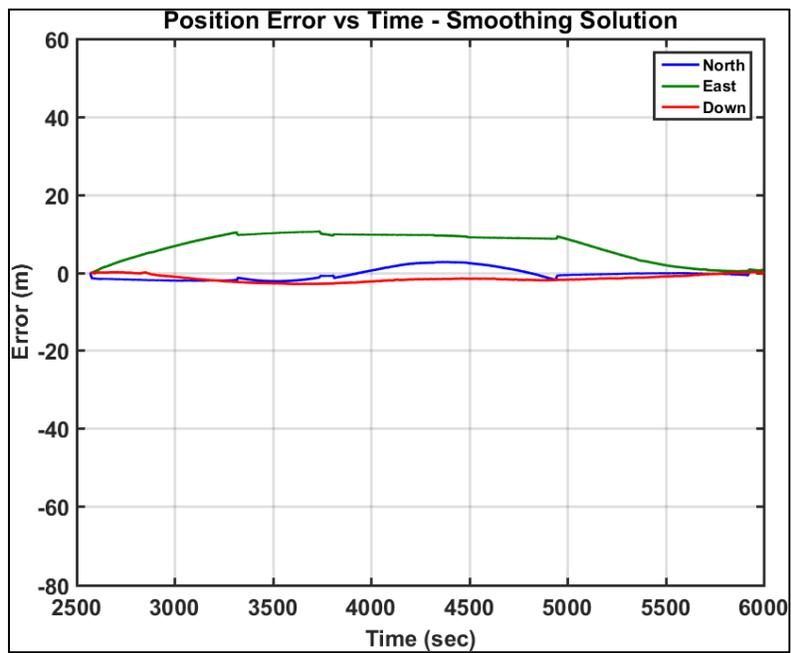


Figure 6-8: NED Error - Smoothing (Scenario #1)

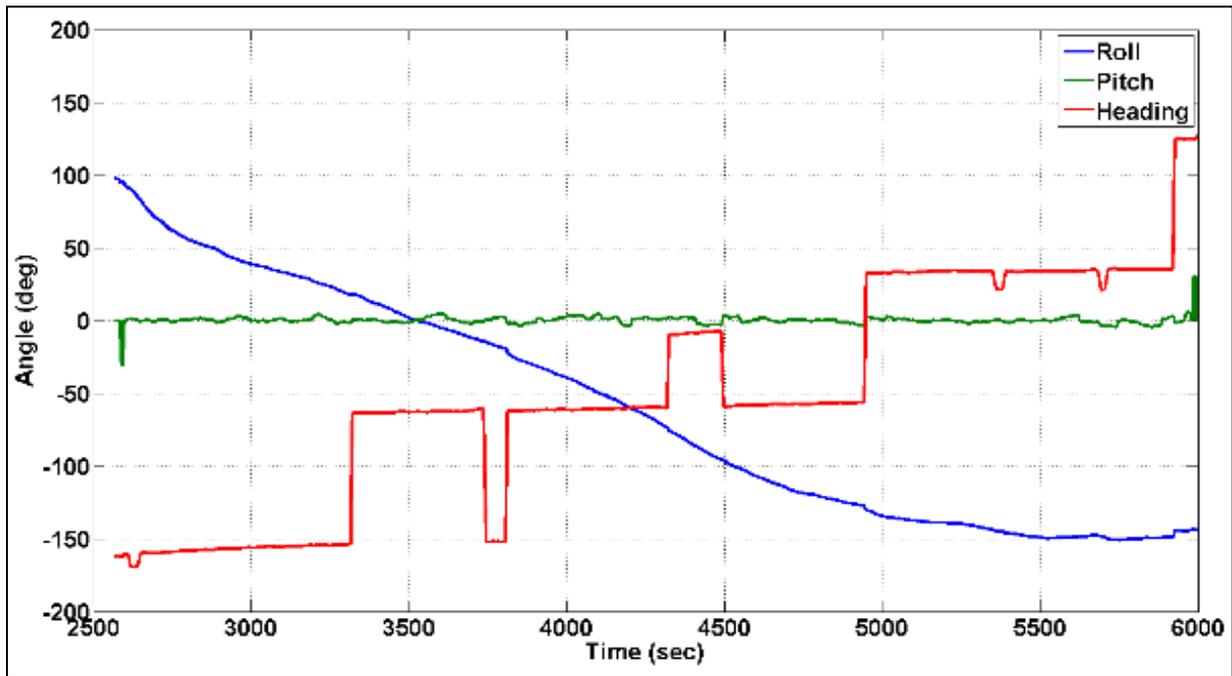


Figure 6-9: Roll, Pitch and Azimuth - Smoothing (Scenario #1)

The three main orientation angles of the pig can be defined as:

- Azimuth: is the Heading of the IMU, which is the angle between the IMU y-axis and magnetic north.
- Pitch: the rotation around the IMU's x-axis.
- Roll: the rotation around the IMU's y-axis.

Since there is no restriction about the rolling motion of the pig (pig is free to roll about its axis), the drop in roll angle's value represents the rolling motion of the pig during its journey through the pipeline as shown in Figure 6-9 (blue curve). The green curve in shows the changes of the pitch angle of the pig. Only a tiny change of this angle is noticeable except at the pig launches at the beginning and the end of the pipeline where the lunches are located above the ground surface.

6.2.2 Scenario #2:

In this scenario, one simulated AGM (as shown in Figure 6-10) has been added to the data as a coordinate update (CUPT). The position error at (4324 sec) has been reset due to AGM update (AGM STD 0.1 m).

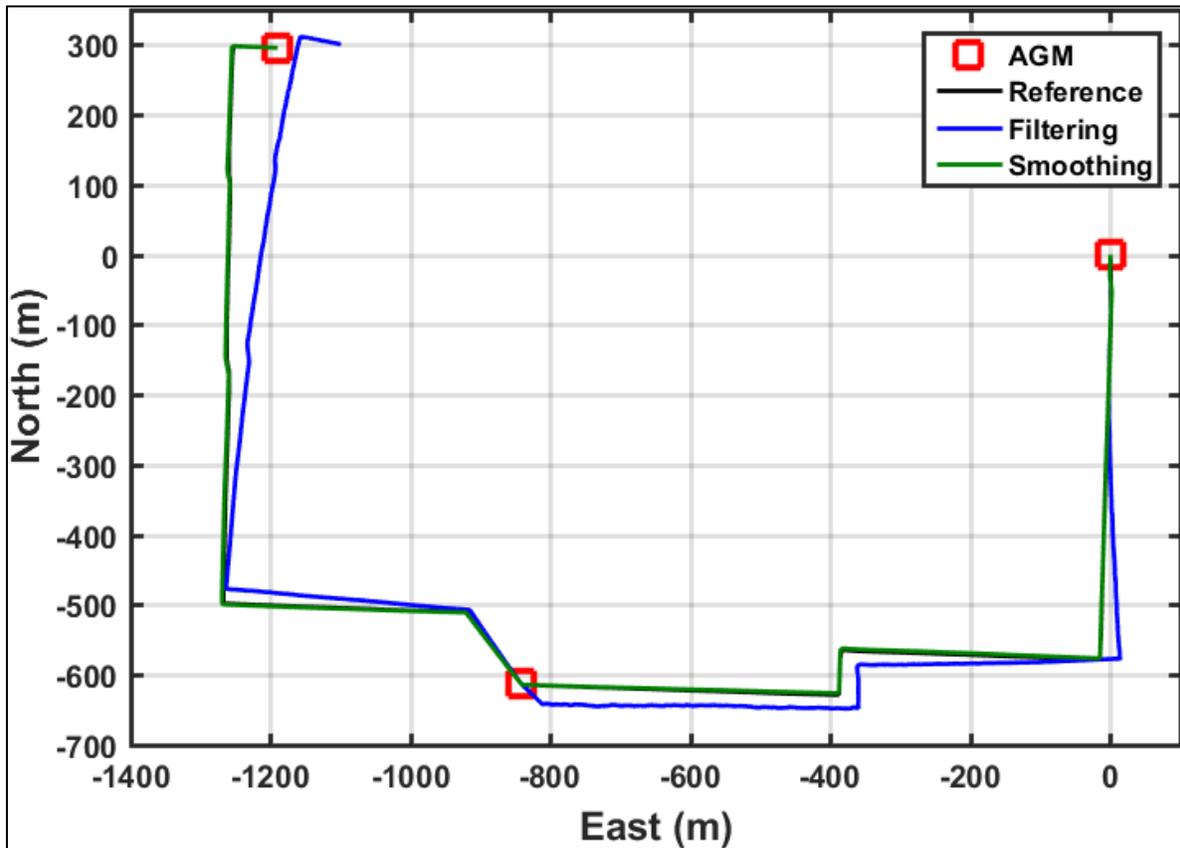


Figure 6-10: Full Trajectory (Scenario #2)

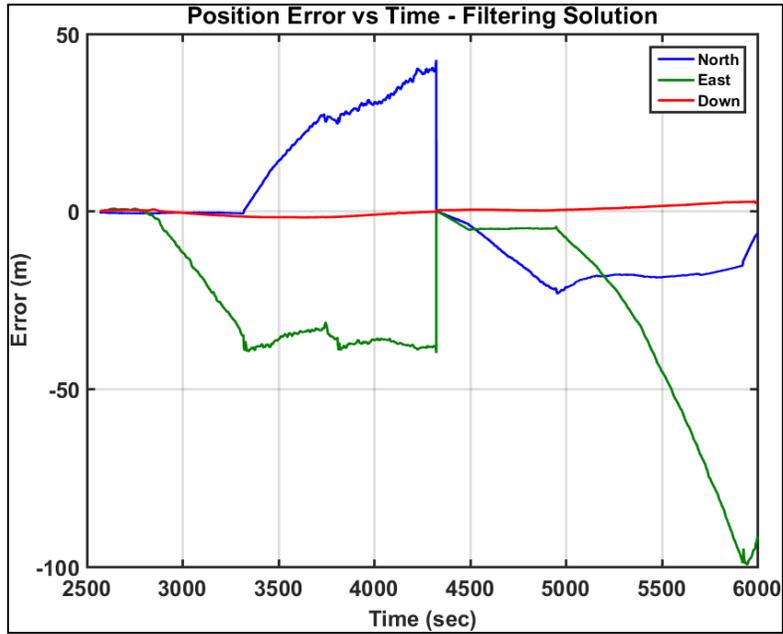


Figure 6-11: NED Error - Filtering (Scenario #2)

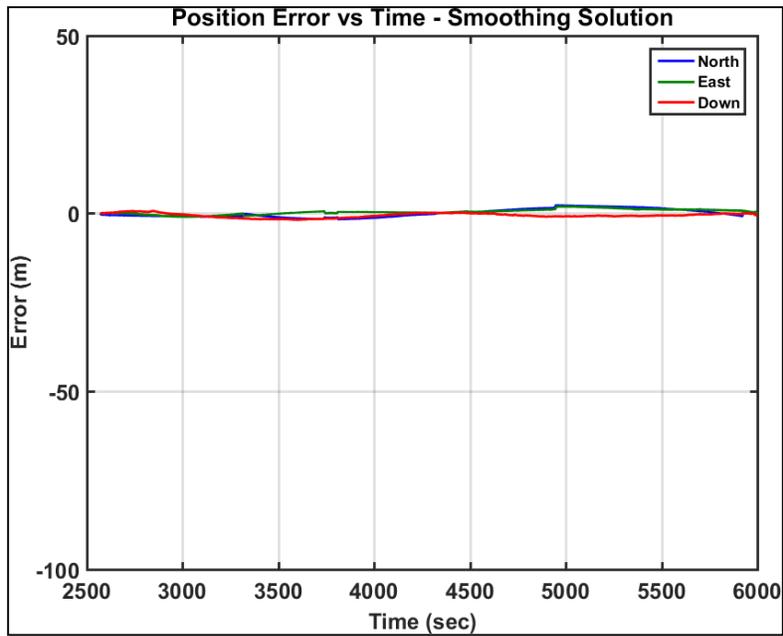


Figure 6-12: NED Error - Smoothing (Scenario #2)

Figure 6-12 shows how the errors have been reduced using the backward smoothing. Huge improvement can be noticed especially in north and east direction.

Adding one position update, improve the backward solution dramatically. However, for long distance pipeline, multiple AGMs need to be added to get the same results. Which will increase the cost of the inspection. Therefore, in the next section, the new proposed algorithm will shows the improvement that can be achieved without adding new position update (AGM) for forward solution.

6.3 PLJ/EKF Results (MEMS)

The new pipeline navigation approach's results using PLJ/EKF algorithm developed in this thesis are shown in this section. The same two scenarios (as shown in section 6.2) for MEMS based IMU are applied here. This section introduces the equipment used and describes the pipeline inspection test performed to assess the efficacy of the PLJ algorithm. The results of the proposed method - pipeline junction's integration - will be discussed in detail and compared to the results of the traditional method of the EKF-based Odometer/AGM integration for pipeline navigation. In figures and tables, EKF/PLJ will be referred to as the new developed method, while EKF will be referred to as the normal EKF method. Please note that Odometer and AGM (if available) are used in both methods. Moreover, in all cases, starting and ending position of the pipeline are known. The developed method was examined through real pipeline inspection trajectories, using as few AGMs as possible. SiIMU02 (by UTC Aerospace Systems) MEMS-based inertial sensors were used for the experiment in this section. SiIMU02 is a six degree of freedom inertial system that uses solid-state devices to measure the angular rate and linear acceleration. Table 6-3 shows the IMU specifications.

The reference solution used to evaluate the proposed method is based on the LN200 high-end tactile grade IMU has been provided by the pipeline contractor. The forward speed (odometer data) was provided by the pig's manufacturer with an odometer STD 0.15 m/s.

One point worth noting is that the pig operator did not provide the AGMs' positions; as a result, artificial AGMs were extracted from the true reference trajectory (Section 6.1.2). The total pig journey distance in this experiment is almost 3km over a total travel time of 1 hour.

Table 6-3: MEMS – IMU Specifications (SiIMU02)

	Gyroscope	Accelerometer
Bias Repeatability (1σ)	$\leq 100^\circ/hr$	$\leq 10\ mg$
Random Walk	$\leq 0.5^\circ/\sqrt{hr}$	$\leq 0.5m/s/\sqrt{hr}$
Size (mm)	Diameter (65.5) Depth (35.5)	

For comparison purposes, the proposed algorithm was applied for two different scenarios. In both scenarios, the position of the first and last point of the trajectory is known. The first point represents the pig pipeline inlet and the last point is the pig pipeline outlet.

- Scenario #1: Processed IMU & Odometer data using no AGM.
- Scenario #2: Processed IMU & Odometer data using one AGM (after 30 min)

6.3.1 Scenario #1

This scenario is similar to scenario #1 earlier without adding any AGM, the difference between both solutions is clear, as shown in Figure 6-13. The position errors for each method are shown in Figure 6-14 and Figure 6-15, respectively. The bar graphs in Figure 6-16 and Figure 6-17 show that the proposed EKF/PLJ method greatly improved the accuracy of the results. EKF/Odometer integration for pipeline navigation, using the developed method of EKF/PLJ, had a maximum position error of 11.59 m in the north direction, and EKF/Odometer solution only had a maximum position error of 76.41 m in the same direction. This is an overall average improvement of 85%.

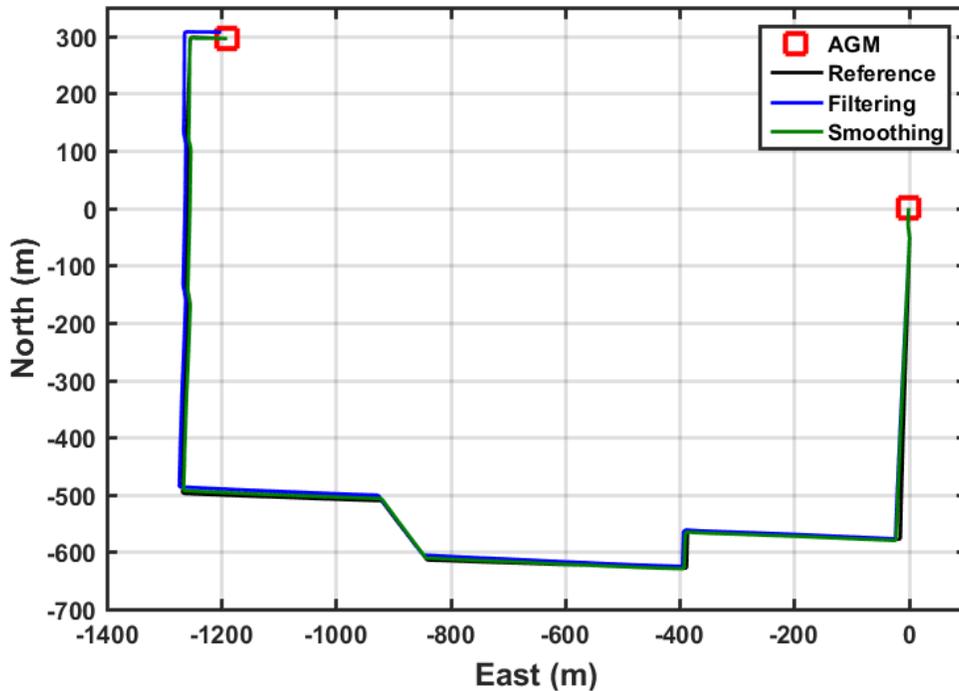


Figure 6-13: Trajectories - Scenario #1

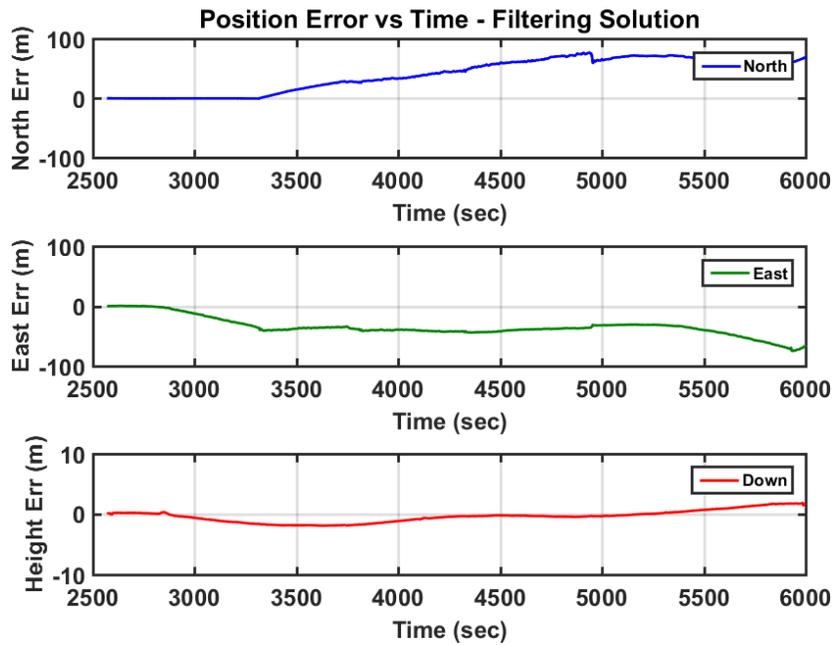


Figure 6-14: Positions RMS Errors - EKF Scenario #1

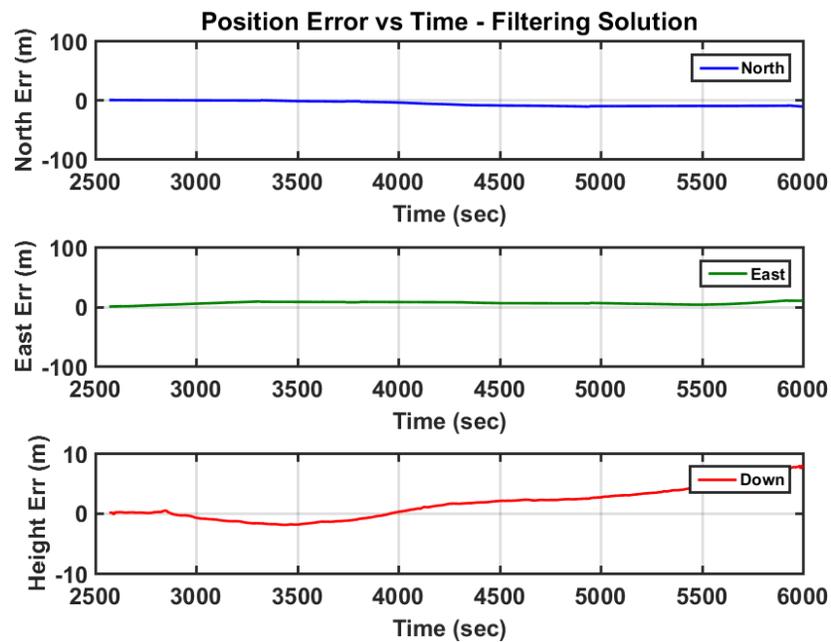


Figure 6-15: Positions RMS Errors - EKF/PLJ - Scenario #1

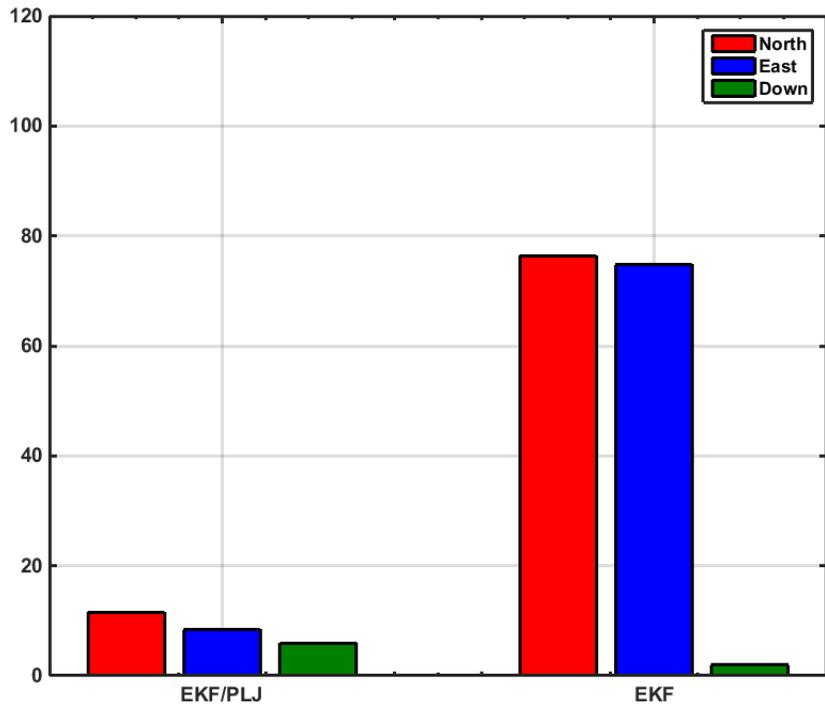


Figure 6-16: Maximum Position Errors - Scenario #1

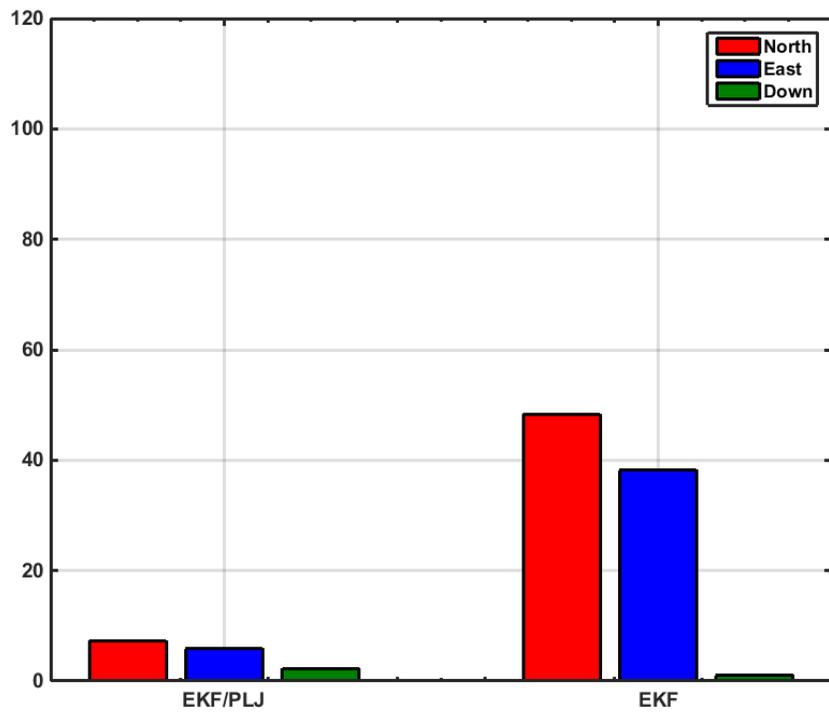


Figure 6-17: RMS Positions Errors - Scenario #1

Despite all the improvements in the horizontal plane (north and east directions), the above bar graphs reveal that height errors have slight increments using the new EKF/PLJ method. This increment does not affect the total solution for the pipeline trajectory, especially, if we know that the pipeline is usually located 1-5 m below the earth surface. Finally, the results can be summarized as shown in Table 6-4 and Table 6-5 where north, east, height maximum and RMS errors for both scenarios are shown.

Table 6-4: North, East & Height Errors

Maximum Error					
Method		North (m)	East (m)	Height (m)	
Scenario #1	EKF	31.99	72.89	3.28	
	EKF/PLJ	11.83	10.77	7.03	
Scenario #2	EKF	76.41	51.94	1.99	
	EKF/PLJ	11.59	8.48	5.99	

Table 6-5: North, East & Height Errors

RMS Error				
Method		North (m)	East (m)	Height (m)
Scenario #1	EKF	17.7	25.05	1.61
	EKF/PLJ	7.11	5.80	2.91
Scenario #2	EKF	46.72	34.58	1.03
	EKF/PLJ	7.35	5.91	2.31

6.3.2 Scenario #2

In this scenario, one AGM was added after 20 minutes of motion to provide position update (CUPT) to the navigation algorithm. Figure 6-18 shows the solutions of the EKF-based and EKF/PLJ-based, where both solutions are compared against the reference trajectory. The EKF/PLJ proposed solution showed an improvement over the standalone EKF solution, which is clearly noticed in Figure 6-19 and Figure 6-20. The figures illustrate the north, east, and height positions RMS errors for both methods. Maximum and RMS position errors (in meters) for each solution are shown by bar graphs in Figure 6-21 and **Figure 6-22**, respectively.

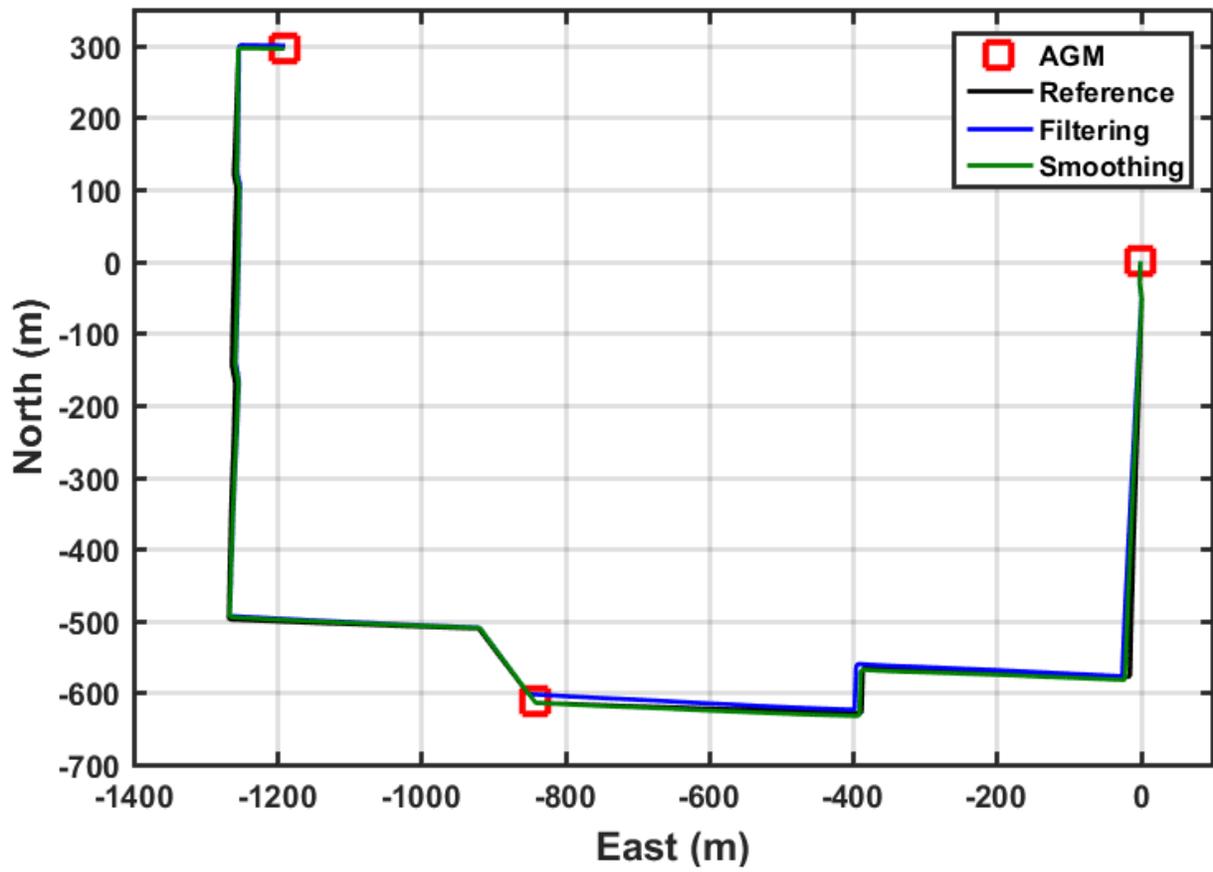


Figure 6-18: Trajectories - Scenario #2

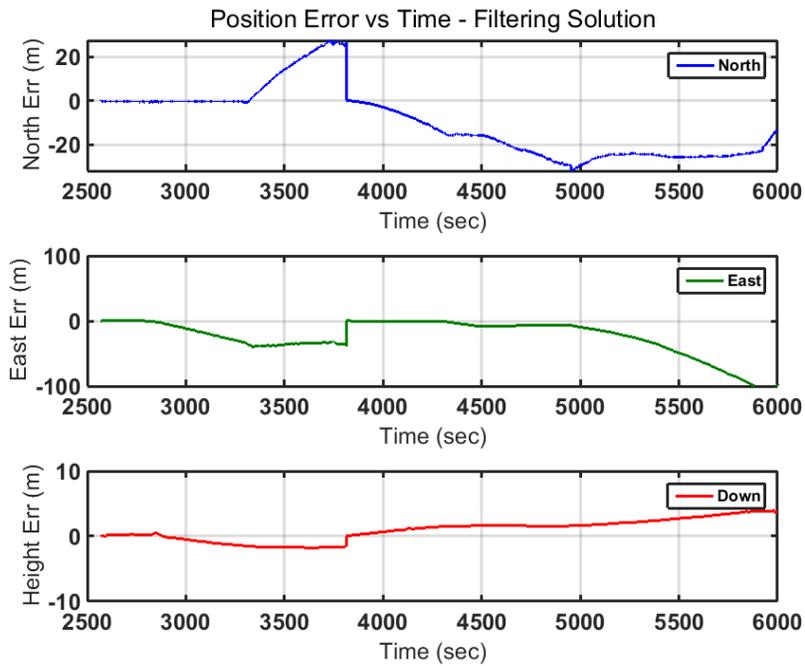


Figure 6-19: Positions RMS Errors - EKF Scenario #2

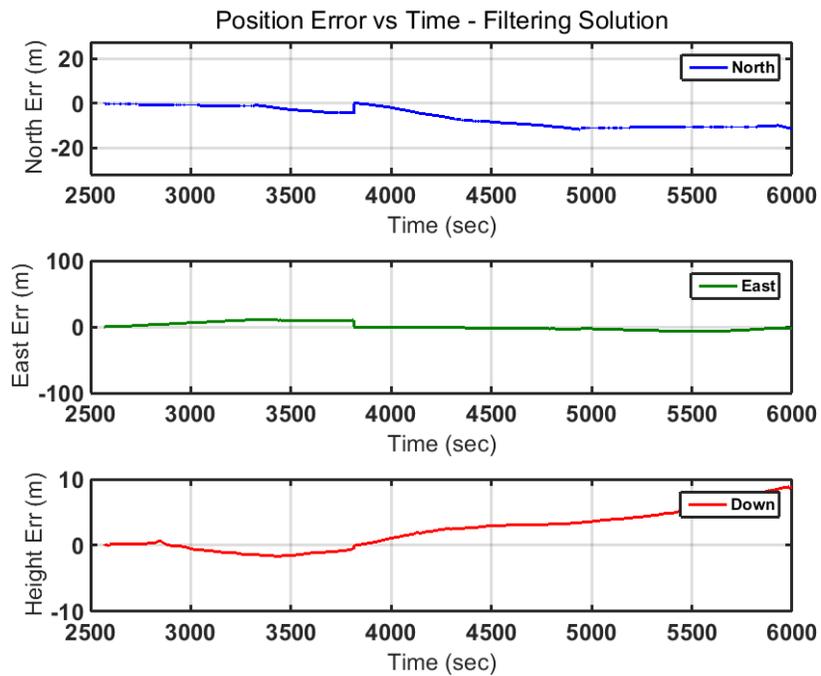


Figure 6-20: Positions RMS Errors - EKF/PLJ Scenario #2

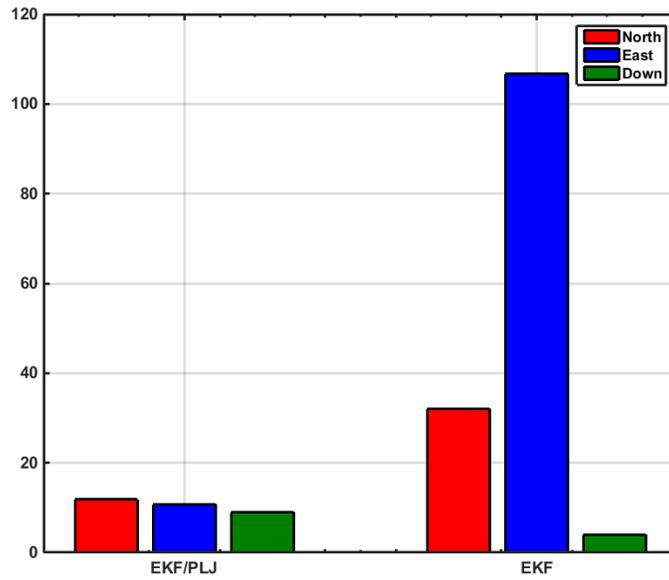


Figure 6-21: Maximum Position Error - Scenario #2

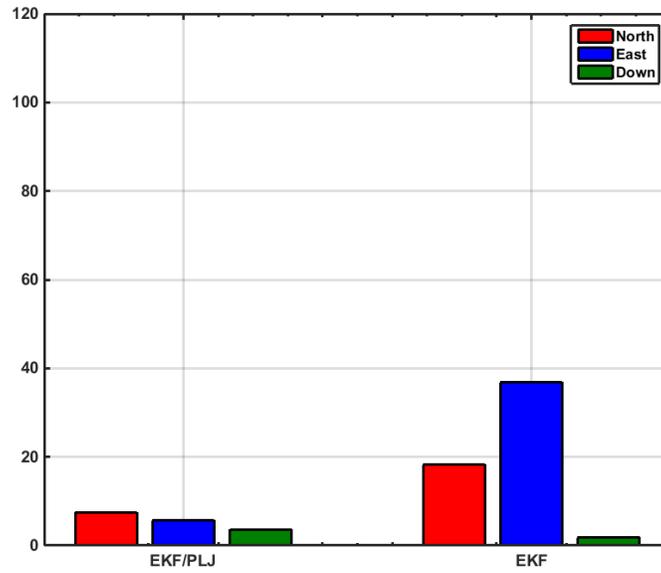


Figure 6-22: RMS Position Errors - Scenario #2

6.4 PLJ/EKF Results (LN200)

In this section results of FOG based IMU (LN200) are demonstrated. The IMU specifications are shown in Table 6-6.

Table 6-6: LN200s – IMU Specifications

	Gyroscope	Accelerometer
Bias Repeatability (1σ)	0.5 °/hr	300 μ g
Random Walk	0.05°/ \sqrt{hr}	0.34 m/s/ \sqrt{hr}
Size (mm)		Diameter (88.9) Height (85.1)

The data of this experiment has been collected for 36 *inch* diameter pig with a velocity average around 0.4 m/s. No reference trajectory has been provided for this test, however, 10 AGMs coordinates have been provided. The pig ran in the pipeline for 21.42 hours and traveled about 31.24 km.

Figure 6-23 illustrates the full trajectory results for LN200 IMU. The “+” markers are the above ground markers. The area inside the rectangle is zoomed and illustrated in Figure 6-24.

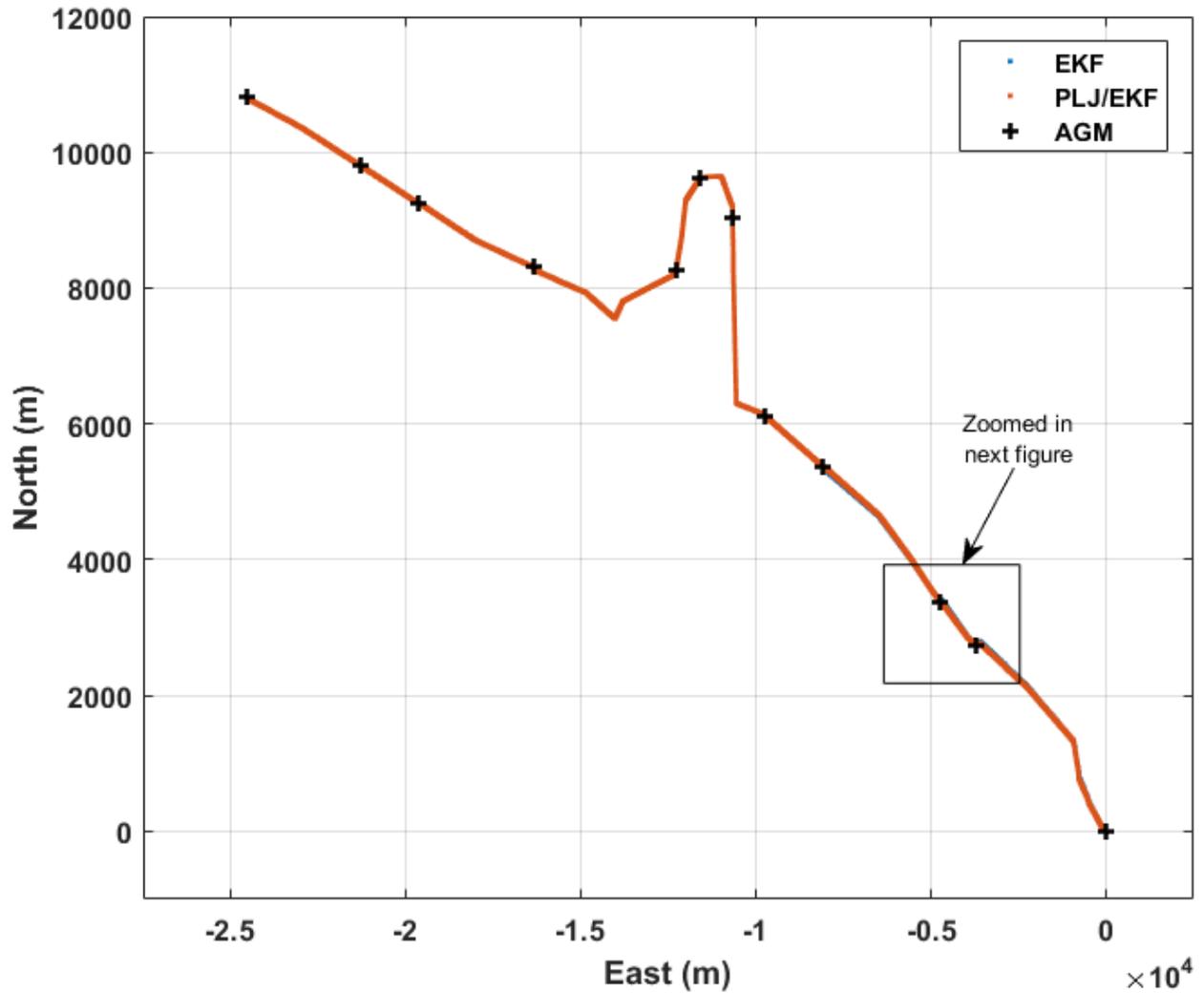


Figure 6-23: Full Trajectory - LN200

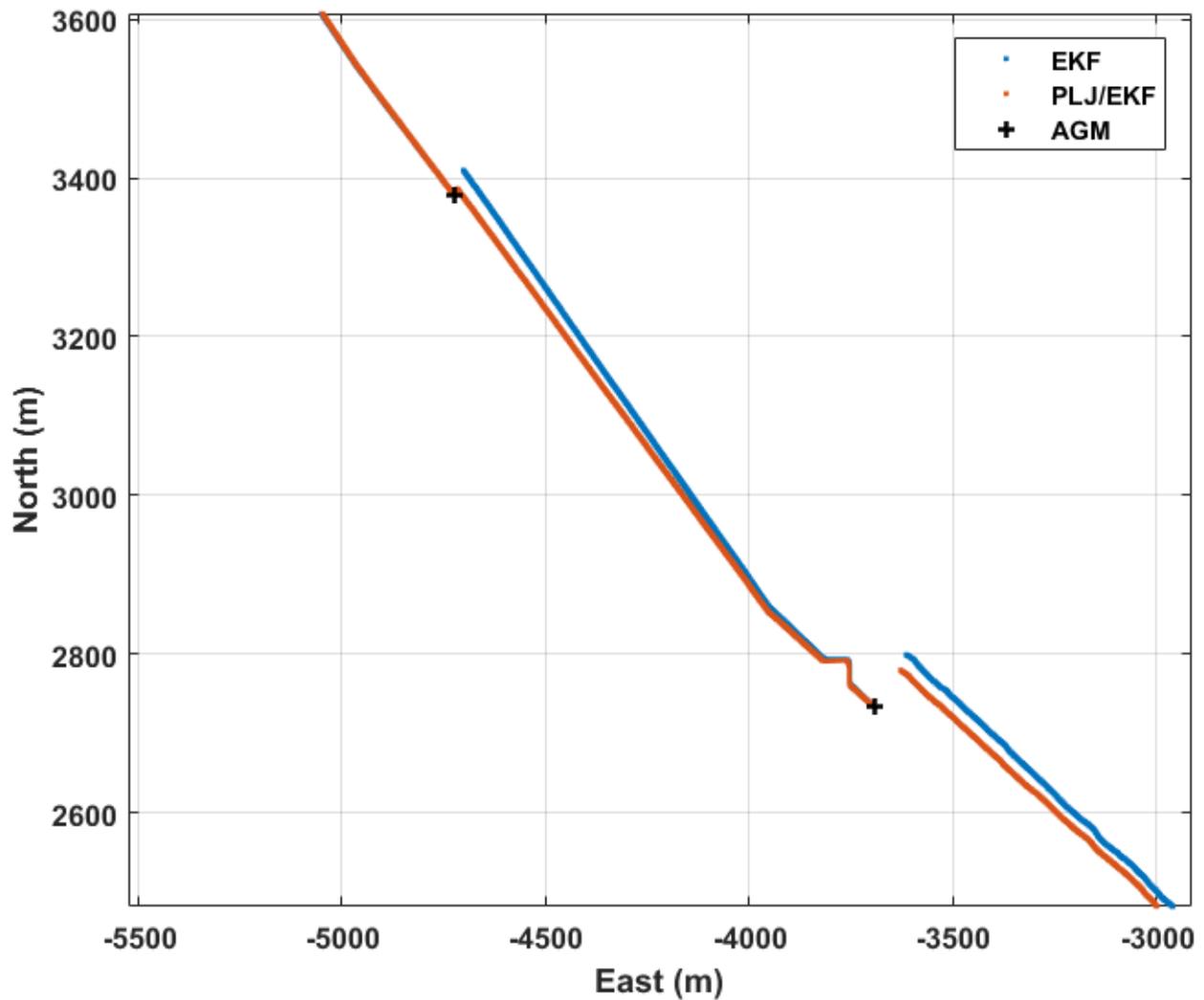


Figure 6-24: Trajectory - Zoomed area from Figure 6-23

The 3D errors has been calculated at the AGM markers by calculating the differences between the position of the AGM and the position of the filtering solution at the same time. Table 6-7 shows all combined errors that includes latitude errors, longitude errors, height errors, 2D errors, and 3D errors.

Figure 6-23 illustrates both AGM #2 and AGM #3. As shown in Table 6-7, the 3D error at AGM #2 has been reduced from 40.215(m) to 14.88(m) using EKF/PLJ algorithm.

Table 6-7: LN200 - Errors

AGM #	Algorithm	Latitude Error (m)	Longitude Error (m)	Height Error (m)	2D Error (m)	3D Error (m)
1	EKF	0.039	0.087	0.203	0.096	0.224
	EKF/PLJ	0.038	0.089	0.141	0.097	0.171
2	EKF	31.36	23.22	9.728	39.021	40.215
	EKF/PLJ	8.039	7.863	9.746	11.245	14.88
3	EKF	18.913	0.246	31.762	18.915	36.967
	EKF/PLJ	7.092	13.9	31.793	15.605	35.416
4	EKF	4.591	6.563	15.465	8.009	17.416
	EKF/PLJ	4.564	6.549	15.371	7.983	17.32
5	EKF	1.259	10.356	32.224	10.432	33.87
	EKF/PLJ	1.622	8.212	32.216	8.371	33.286
6	EKF	5.251	1.927	10.911	5.593	12.261
	EKF/PLJ	5.255	1.918	10.911	5.594	12.261
7	EKF	0.186	0.062	0.015	0.196	0.197
	EKF/PLJ	0.186	0.062	0.015	0.196	0.197
8	EKF	33.823	6.237	39.881	34.393	52.663
	EKF/PLJ	33.788	6.287	39.881	34.368	52.646
9	EKF	0.022	0.064	0.045	0.068	0.081
	EKF/PLJ	0.022	0.064	0.045	0.068	0.081
10	EKF	1.968	1.985	16.157	2.795	16.397
	EKF/PLJ	2.12	2.105	16.156	2.988	16.430

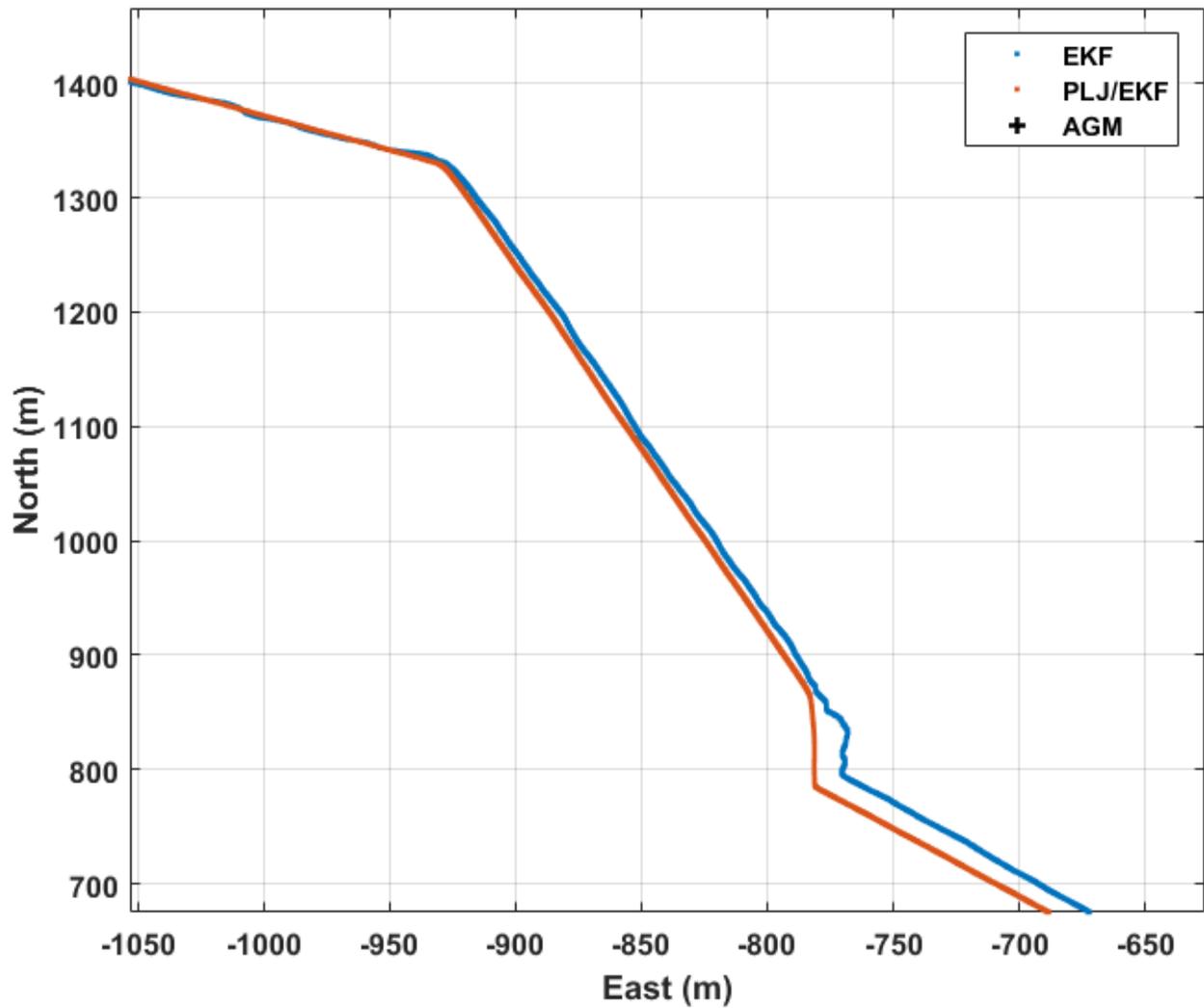


Figure 6-25: The effect of EKF/PLJ algorithm

Figure 6-25 shows the effect of EKF/PLJ algorithm for a short period (red line). It is noticeable that the aliasing appeared using EKF algorithm have been removed (blue line).

Chapter Seven : CONCLUSION AND RECOMMENDATION

This chapter brings together the theoretical and practical results to create a unified view of the integrating pipeline junction algorithm with the EKF. All conclusions and recommendations are made in an attempt to improve usability and adoption of MEMS-based IMU systems for pipeline inspectors.

Although there are many avenues that can be progressed from this thesis, those are the most concern relate to improve the accuracy when using MEMS-based IMUs. Accuracy improvement can result from increasing the accuracy of the inertial sensors, increasing the accuracy of the aiding navigation sensors, through the understanding of the physical source of the initial errors.

In Chapter One an introduction to pipeline navigation has been introduced. The problem statement has been highlighted with the current pigs capabilities. Various types of aiding sensors that can improve navigation algorithms have been investigated in Chapter Two. In section 2.2, the required navigation frames have been introduced. Section 2.3 introduced the different attitude representations used in mechanization algorithm. Gradually mechanization algorithm have been shown in section 2.4. The required aiding sensors for pipeline navigation have been modeled in section 2.5. Inertial sensor errors with their mathematical models have been introduced in Chapter Three. INS calibration methods for MEMS-based IMUs are explained in section 3.4. MEMS-based IMU alignment procedures have been shown in section 3.6. Kalman filter with INS error models (ψ and ϕ) have been reviewed in Chapter Four. Stochastic random processes including Gauss-Markov model have been explained in section 4.4. Pipeline junction algorithm has been introduced in Chapter Five. Pipeline bend detection method are explained in section 5.4. Required constraints

for PLJ has been introduced in section 5.5 . A comparison between EKF and PLJ/EKF algorithms has been illustrated in Chapter Six.

7.1 Research Contribution

There are two main contributions developed in this research work:

1. Enhancing the navigation solution for smart pipeline inspection gauge's (pig) tool using new defined pipeline constraints. Without these constraints, as shown in **Figure 6-6**, the IMU errors cannot be bounded and the solution drifts extremely. In Chapter Five, these constraints have been developed and used to bound the IMU error and reduce the position errors dramatically.
2. In pipeline industry, magnetic flux leakage (MFL) and electromagnetic acoustic transducers are used to detect pipeline junctions. However, in section 5.2, new method for pipeline junctions detection has been introduced using gyroscopes and accelerometers readings.
3. IMU turn detection (pipeline bend detection) has been introduced by combining two methods as follows:
 - a. Using angular velocity magnitude.
 - b. Calculating the average value of heading and pitch angles.

7.2 Conclusion

From the work performed in this thesis, the following conclusions can be drawn:

1. This thesis's objective was to investigate means of improving the small diameter pig's positioning capability by including MEMS-based IMU. Due to the size of tactical grade

IMUs, it is not possible to include them in the current small size diameter pigs (8" and below). Therefore, the need of using MEMS-based IMUs appeared. However, the performance of these MEMS-based IMUs are low compared to the required accuracy by industry. Ideally, the pig should be positioned to within one pipe length so that the correct section may be determined if pipe replacement is required. However, two to three meters should be accepted for in case of excavation for maintenance.

2. Inertial navigation errors grow unboundedly without any assumptions of the motion body and without any aiding sensors. This is the reason that INS cannot provide proper navigation solution as a standalone system for long period. Constraints can limit the growth of these errors. Therefore, understanding the behavior and system environment added useful values to the designed system. Based on the reality of constructing the pipelines (i.e. series of straight pipes), and how they have been connected to each other, new constrains have been added to the measurement model to bounds the errors in IMU drift. Pitch and heading angles have been constrained between two pipeline junctions (excluding the pipes' fittings). This constrain method is called pipeline junction constraints (PJC).
3. In order to implement the PJC, the current pig position need to be marked as inside or outside the straight pipe. Two different methods have been integrated for this purpose:
 - a. Pipeline junctions' detection technique has been developed in section 5.3 using two different methods (moving average technique, and wavelet transformation).
 - b. Bend detection algorithm (BDA) has been developed in section 5.4 using two different methods (angular velocity magnitude, and mean of heading & pitch angles

history). BDA allows the system to decide whether the current position is inside the fitting (bend) or not.

4. By applying PJC, the IMU drift has been reduced. Therefore, the system was capable of reducing the trajectory navigation root mean square errors (RMSE) by around 80% over one hour of operation and without using any AGM along the pig journey using MEMS-based IMU. Hence, the system is capable to reduce the used number of AGMs as shown in both scenarios of section 6.3. However, some external AGMs are still required in order to limit the absolute positional error growth of the INS.
5. Finally, the output solution can be used to generate the new maps, or validate the existing maps, and mark the detected parts of pipeline for maintenance.

In Chapter Six, results from MEMS-based IMU and high-end tactical grade IMU have been provided. Two different scenarios have been discussed for MEMS-based IMU as results of real pipeline data that was collected using a MEMS-IMU-based system. These results show that the newly proposed method is capable of reducing the trajectory navigation RMS error by around 80% over one hour of operation and without using any AGM along the pig journey. Similarly, the proposed algorithm has been tested on pipeline data that has been collected using high-end tactical grade IMU. The algorithm has slight improvement over the high-end tactical grade IMU due to its accuracy. Table 6-7 shows the improvement of 9% in the 3D position errors.

7.3 Recommendations for Further Research

Here are some recommendations for further research to be investigated to improve the current pipeline navigation using MEMS-based IMU:

1. Making a system of cheaper and more numerous inertial sensors can match or exceed the performance of high-end tactical grade IMU using multiple MEMS-based IMUs.
2. Smart pigs contain multiple sensors for different types of measurements. Some of them are able to detect cracks and pipe wall thickness. It is advisable to use their data to detect the pipeline junctions and update PLJ/EKF algorithm.
3. Including different types of aiding sensors can improve the navigation results. Hence, vision sensors (i.e. camera or LIDAR) and magnetometer can be included in future research.
4. In this research, map matching has not been used based on the industrial required. However, if initial maps have been including, using initial engineering maps to integrate them with PLJ/EKF algorithm can improve the results dramatically.
5. Since this is an offline system, other non-linear filtering algorithms can be tested; such as unscented Kalman filter (UKF) or particle filter (PF).

References

- Altmann, S. L. (2005). *Rotations, quaternions, and double groups*. Mineola, N.Y.: Dover Publications.
- Bar-Itzhack, I. Y., & Berman, N. (1988). Control theoretic approach to inertial navigation systems. *Journal of Guidance, Control, and Dynamics*, 11(3), 237-245.
- Benson, & O., D. (1975). A Comparison of Two Approaches to Pure-Inertial and Doppler-Inertial Error Analysis. *Aerospace and Electronic Systems, IEEE Transactions on, AES-11*(4), 447-455.
- Britting, & R., K. (1971). *Inertial navigation systems analysis*. New York: Wiley-Interscience.
- Brown, R. G., & Hwang, P. Y. C. (1997). *Introduction to random signals and applied Kalman filtering : with MATLAB exercises and solutions* (3rd ed.). New York: Wiley.
- Chatfield, A. B. (1997). *Fundamentals Of High Accuracy Inertial Navigation*. Reston, VA, USA: American Institute of Aeronautics and Astronautics.
- Chris Goodall, & El-Sheimy, N. (2007). *Intelligent Tuning of a Kalman Filter Using Low-cost Mems Inertial Sensors*. Paper presented at the 5th International Symposium on Mobile Mapping Technology MMT '07.
- Cossaboom, M., Georgy, J., Karamat, T., & Noureldin, A. (2012). Augmented Kalman Filter and Map Matching for 3D RISS/GPS Integration for Land Vehicles. *International Journal of Navigation and Observation*, 2012, 16.
- Council on Environmental Quality (U.S.), & Saflianis, C. C. *Inertial Navigation Systems aided by G.P.S. (Global Positioning System (I.N.S.))*.
- El-Sheimy, N. (2012). *Introduction to INS - With Applications to Positioning and Mapping* (Graduate Course Lectures): University of Calgary.
- El-Sheimy, N., Haiying, H., & Xiaoji, N. (2008). Analysis and Modeling of Inertial Sensors Using Allan Variance. *Instrumentation and Measurement, IEEE Transactions on*, 57(1), 140-149.
- El-Sheimy, N., Sahli, H., & Moussa, A. (2015). Canada Patent application No. 524.28. Innovate Calgary: Provisional.
- Farrell, J., & Barth, M. (Eds.). (1998). *The global positioning system and inertial navigation*. New York: McGraw-Hill.
- Gelb, A., Kasper, J. F., Nash, R. A., Price, C. F., & Sutherland, A. A. (1974). *Applied Optimal Estimation*. US: THE M.I.T. PRESS.

- Goodall, C. L. (2009). *Improving Usability of Low-Cost INS/GPS Navigation Systems using Intelligent Techniques*. University of Calgary.
- Gordon, N. J., Salmond, D. J., & Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140(2), 107-113.
- Goshen-Meskin, D., & Bar-Itzhack, I. Y. (1992). Unified approach to inertial navigation system error modeling. *Journal of Guidance, Control, and Dynamics*, 15(3), 648-653.
- Graps, A. (1995). An introduction to wavelets. *IEEE Computational Science and Engineering*(2), 50--61.
- Grewal, M. S., & Andrews, A. P. (2001). *Kalman Filtering: Theory and Practice Using MATLAB*: A Wiley-Interscience Publications.
- Grewal, M. S., Weill, L. R., & Andrews, A. P. (2007). *Global positioning systems, inertial navigation, and integration* (2nd ed.). Hoboken, N.J.: Wiley-Interscience.
- Groves, P. D. (2008). Principles of GNSS, inertial, and multisensor integrated navigation systems, GNSS technology and applications seriespp. 518).
- Hanna, P. L. (1990, 9 Jan 1990). *Strapdown inertial systems for pipeline navigation*. Paper presented at the Inertial Navigation Sensor Development, IEE Colloquium on.
- Hou, H. (2004). *Model Inertial Sensors Error Using Allan Variance*. University of Calgary.
- IEEE, S. (2008). IEEE Standard Specification Format Guide and Test Procedure for Single-Axis Interferometric Fiber Optic Gyros, *IEEE Std 952-1997* (pp. i).
- IEEE Standard for Inertial Sensor Terminology. (2001). *IEEE Std 528-2001*, 0_1.
- Jazwinski, A. H. (1970). *Stochastic processes and filtering theory*. sl: sn.
- Julier, S., Uhlmann, J., & Durrant-Whyte, H. F. (2000). A new method for the nonlinear transformation of means and covariances in filters and estimators. *Automatic Control, IEEE Transactions on*, 45(3), 477-482.
- Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. 35-45-.
- KING, A. D. (1998). Inertial Navigation - Forty Years of Evolution. *GEC REVIEW*, 13(3), 140-149.
- Kuipers, J. B. (2002). *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace and Virtual Reality*: Princeton University Press.

- Kuprewicz, R. B. (2005). *Observations on the Application of Smart Pigging on Transmission Pipelines*: Accufacts Inc.
- Lawrence, A. (1993). *Modern inertial technology : navigation, guidance, and control*. New York: Springer.
- Lerro, D., & Bar-Shalom, Y. (1993). Tracking with unbiased consistent converted measurements versus EKF. *Aerospace and Electronic Systems, IEEE Transactions on*, 29(3), 1015-1022.
- Liu, H. (2009). *Optimal smoothing techniques in aided inertial navigation and surveying systems*. University of Calgary.
- McGreevy, J., & Litton Systems, I. (1986). *Fundamentals of Strapdown Inertial Navigation*: Litton Systems, Incorporated.
- Meditch, J. S. (1969). *Stochastic optimal linear estimation and control*. New York: McGraw-Hill.
- MEMS Inertial Measurement Unit - SiIMU02 Datasheet. (2010). Atlantic Inertial Systems is part of Goodrich Corporation.
- Mesa, H. (2005). Adapted Wavelets for Pattern Detection. In A. Sanfeliu & M. Cortés (Eds.), *Progress in Pattern Recognition, Image Analysis and Applications* (Vol. 3773, pp. 933-944): Springer Berlin Heidelberg.
- Meyer, Y. (1993). *Wavelets: Algorithms and Applications*.
- Miller, R. B. (1983). A new strapdown attitude algorithm. *Journal of Guidance, Control, and Dynamics*, 6(4), 287-291.
- Nassar, S. (2003). *Improving the inertial navigation system (INS) error model for INS and INS/DGPS applications*.
- Navatel. (2014). IMU Errors and Their Effects. (APN-064),
- Noureldin, A., Karamat, T. B., Eberts, M. D., & El-Shafie, A. (2009). Performance Enhancement of MEMS-Based INS/GPS Integration for Low-Cost Navigation Applications. *Vehicular Technology, IEEE Transactions on*, 58(3), 1077-1096.
- Noureldin, A., Karamat, T. B., & Georgy, J. (2012). *Fundamentals of Inertial Navigation, Satellite Positioning and Their Integration*. New York: Springer.
- Peter, S. M. (1979). *Stochastic models, estimation, and control* (Vol. Volume 141, Part 1): Elsevier.
- Petovello, M. G. (2012). *Lecture Notes for ENGO 620 - Estimation for Navigation*.

- Petovello, M. G., Cannon, M. E., & Lachapelle, G. (2003). *Kalman Filter Reliability Analysis Using Different Update Strategies*. Paper presented at the Proceedings of the CASI Annual General Meeting, Montreal, QC.
- Products, P., & Association, S. (1995). *An Introduction to Pipeline Pigging*: Gulf Pub.
- Rogers, R. M. (2007). *Applied mathematics in integrated navigation systems* (3rd ed.). Reston, VA: American Institute of Aeronautics and Astronautics.
- Sahli, H., Al-Hamad, A. M., Ali, A. S., & El-Sheimy, N. (2013). *Sensor Fusion for Trajectory Estimation Using Low Cost INS/Magnetometer/Odometer*.
- Sahli, H., & El-Sheimy, N. (2016). A Novel Method to Enhance Pipeline Trajectory Determination Using Pipeline Junctions. *Sensors*, 16(4), 567.
- Sahli, H., Moussa, A. , Nouredin, A. , El-Sheimy, N. (2014, September 8 - 12, 2014). *Small Pipeline Trajectory Estimation Using MEMS based IMU*. Paper presented at the Proceedings of the 27th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2014), Tampa, FL.
- Salychev, O. S. (1998). *Inertial systems in navigation and geophysics*. Moscow: Bauman MSTU Press.
- Savage, P. G. (2000). *Strapdown analytics*. Maple Plain, Minn.: Strapdown Associates.
- Scherzinger, B. (1996, 22-26 Apr 1996). *Inertial navigator error models for large heading uncertainty*. Paper presented at the Position Location and Navigation Symposium, 1996., IEEE 1996.
- Scherzinger, B. M. (1996). Inertial navigator error models for large heading uncertainty. *Position Location and Navigation Symposium, 1996., IEEE 1996*, 477-484.
- Scherzinger, B. M. (2004). *Estimation with Application to Navigation* (Lecture Notes). University of Calgary: University of Calgary.
- Shin, E.-H. (2001). *Accuracy Improvement of Low Cost INS/GPS for Land Applications*. University of Calgary, Calgary, Alberta, Canada.
- Shin, E.-H. (2005). *Estimation Techniques for Low-Cost Inertial Navigation*. University of Calgary, Calgary, Alberta, Canada.
- Shin, E.-H., & El-Sheimy, N. (2005). *Backward Smoothing for Pipeline Surveying Applications*. Paper presented at the Proceedings of the 2005 National Technical Meeting of The Institute of Navigation, San Diego, CA.

- Shuster, M. D. (1993). Survey of attitude representations. *Journal of the Astronautical Sciences*, 41, 439-517.
- Sukkariéh, S. (2000). *Low Cost, High Integrity, Aided Inertial Navigation Systems for Autonomous Land Vehicles*. The University of Sydney.
- Tiratsoo, J. (1992). *Pipeline Pigging Technology* (Second ed.): Butterworth-Heinemann College.
- Tiratsoo, J. (1999). *Pipeline Pigging and Inspection Technology* (2nd ed.). Boston: Gulf Professional Publishing.
- Titterton, D. H., & Weston, J. L. (2004). Strapdown inertial navigation technology, IEE radar, sonar, navigation, and avionics series 17
- Trusov, A. A. (2011). Allan Variance Analysis of Noise Modes in Gyroscopes.
- Veritas, D. N. (2009). Integrity Management of Submarine Pipeline Systems. Article No. DNV-RP-F116, DET NORSKE VERITAS
- Vetterli, M., & Herley, C. Wavelet and filter banks: theory and design, *IEEE transactions on Acoustic Speech Signal Processing* 1992;40(9).
- Wickerhauser, M. V. (1994). *Adapted Wavelet Analysis from Theory to Software* (First Edition ed.): A K Peters/CRC Press.
- Wint, D. Difficult to Pig Pipelines. TDW Services, Inc.
- Woodley, D. (2011). The origin of intelligent pigs. Retrieved Dec. 2015, 2015, from http://pipelinesinternational.com/news/the_origin_of_intelligent_pigs/65044
- Wu, H., & Chen, G. (1999). Suboptimal Kalman filtering for linear systems with Gaussian-sum type of noise. *Mathematical and Computer Modelling*, 29(3), 101-125.

APPENDIX: Pipeline Pigging

Pipeline Integrity

Pipeline system integrity is a key operational issue in oil and gas industry. Pipeline systems starts from the production fields to refineries and finally to the end users. Significant financial losses can be caused by interruption in oil flow due to pipeline system components failure (ex. pipes, valves, flanges ...etc.). A recent example is the Long Lake oilsands incident at facility south of Fort McMurray in northern Alberta. Where M/s Nexen pipeline spills five million liters of bitumen, sand and oil to cover about 16,000 square meters in July 2015. Similarly, in April 2011, a Plains Midstream Canada ULC pipeline leaked 4.5 million liters of crude oil near a First Nations community in northwest Alberta. In 2013, Macondo incident of BP Deepwater Horizon accident in the Gulf of Mexico. A gas leak and subsequent explosion in combination with component failure, ultimately rendering the emergency blow out preventer to seal off the well. The following fire burned for 36 hours before the drilling rig sank. Eleven operators died and an estimate of 3.26 million barrels of oil were released. The operation cost exceeded \$14 billion (bp.com, 2013).

Pipeline maintenance management (PMM) is essential in preserving pipeline integrity. Maintenance objective, strategies, and the responsibilities must be determined in order to achieve effective pipeline maintenance management. The implementation of these through an organized work process is a crucial factor in order to anticipate and prevent pipeline system failure. Any failure affects both the environment and company assets. Pipeline operators' maintenance management is normally based on regulations and industry standards established by national and international regulators. Alberta Energy Regulator (AER) is the applied Albertan industry standard developed, updated, and regulated in cooperation by and for the oil and gas industry. They are

authorized to make decisions on applications for energy development, monitoring for compliance assurance, decommissioning of developments, and all other aspects of energy resource activities. This authority extends to approvals under the public lands and environment statutes that relate to energy resource activities. Standards have been updated constantly and they influence all key aspects of pipeline operations. This includes the origin of the pipelines, the building process, and its operation and maintenance.

The basic design of the oil and gas pipelines has been conceived to ensure a harmonious blend of safety and efficiency. Safety is considered as a top priority, especially for oil and gas transportation.

- **Safe**

Pipelines are considered safe for transporting considerable quantities of oil even over long distances. Since they are not affected by climatic conditions, pipelines run equally efficiently throughout the year.

- **Reliable**

Pipelines are a convenient, safe and reliable means of transportation especially when you consider the high risks of rail and road transportation. There is also the added advantage of least wastage and loss during transit even though large volumes of oil are continuously transported. Generally, pipelines are accepted as the safest natural gas transportation technique.

- **Economical**

Suitable for large volumes of transportation, oil transported by pipelines does not require any handling operations thereby bringing down total costs incurred for medium and even long haul transportation to an affordable figure.

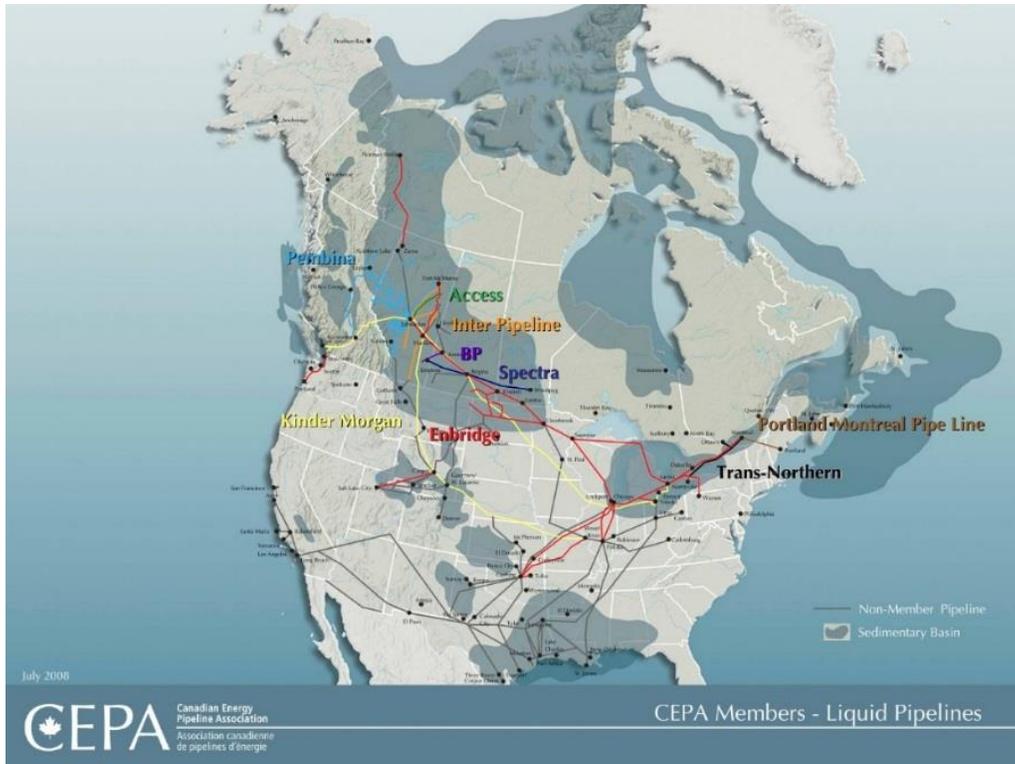


Figure 7-1: Pipelines in North America. Total length approximation 2,000,000 km

Therefore, oil and gas transported by pipelines seems to be the preferred option. However, maintaining the quality of these pipelines along the years is considered as the most important factor in pipeline industry.

Figure 7-1 shows the curdle oil & natural gas pipelines. Several Federal and state standards and regulations regulate oil & gas transported by pipelines (i.e. NEP). They enforce the pipelines clients to ensure both human and environmental safety along with the pipelines safety.

The status of pipelines' lifetime provides true challenge to the pipeline integrity. Therefore, it is important for the pipelines' operators to assess the pipeline conditions regularly. Both internal and external assessments are required to be implemented in order to establish full pipeline condition.

As the vast majority of the pipelines are buried or located at subsea level, external assessment methods are inconvenient (Tiratsoo, 1992). On the other hand, internal and external pipeline assessments are obtainable by accessing the bore of the pipeline. Early detection and identification of developing pipeline threats can be achieved by a feedback of conditional assessment data into the pipeline integrity strategy.

The main responsibility of ensuring pipelines safe operations falls on operators. Hence, the pipelines are inspected by specialists (safety inspectors) to make sure they meet all regulatory requirements.

As more and more emphasis is being placed on the safety of existing pipelines, many of oil and gas pipeline operators' have made rehabilitation of overall pipeline system their top priority (Tiratsoo, 1999).

Launcher and Receiver

In practice, pigging a pipeline starts with a pig trap section in the pipeline that includes a launcher and/ or a receiver. This will allow the system to launch and/or receive a pipeline pig without interruption of the flow. The launcher inserts the pig into the pipeline, where it is either pushed along by line pressure or pulled through the pipe by a cable. The receiver acts as a point to remove the pig from the line, as well as any debris the pig has knocked loose.

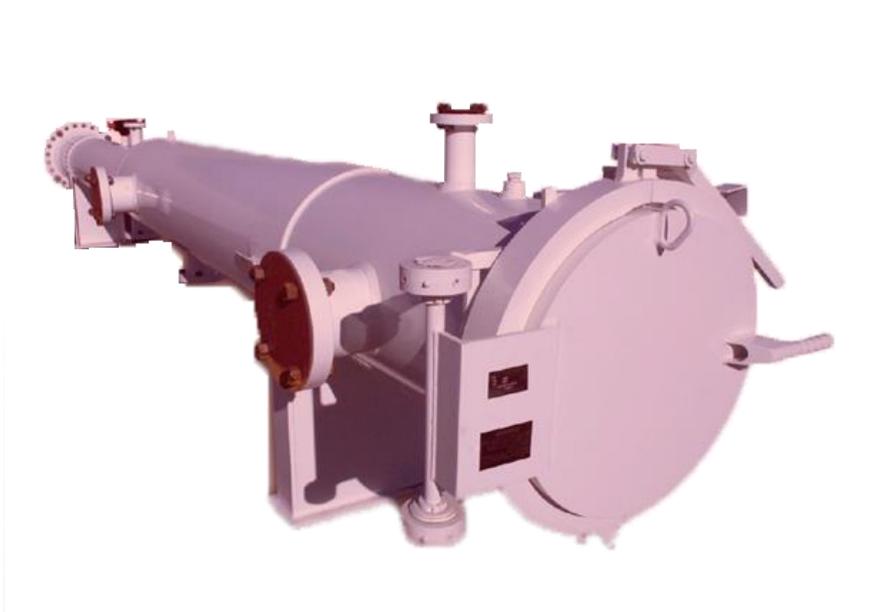


Figure 7-2: Launcher Receiver Design

Piggable Valves

For pipe to be piggable, the in-line valves need to provide a suitable bore. Pipeline can be unpiggable if the in-line valve does not offer a suitable conduit bore. Similarly, all other pipeline fittings and check valves will also need to be assessed. Figure 7-3 illustrates the difference between two types of ball valves. The hollow ball valve cannot be used in piggable pipeline. Similarly, Figure 7-4 shows a difference between two gate valves. Through conduit gate valve is suitable for pipeline pigging.

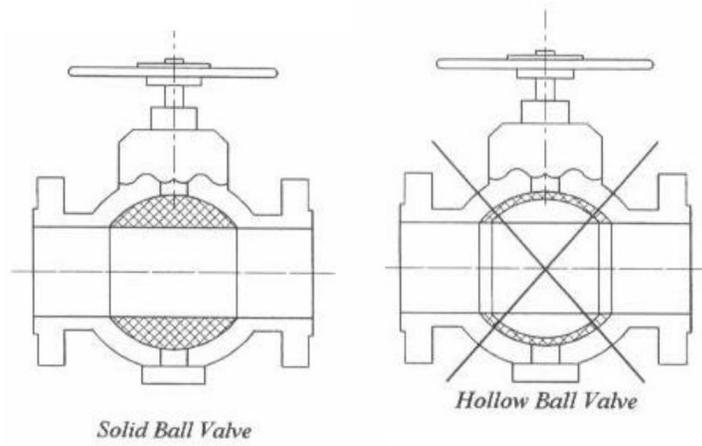


Figure 7-3: Ball Valves (Wint)

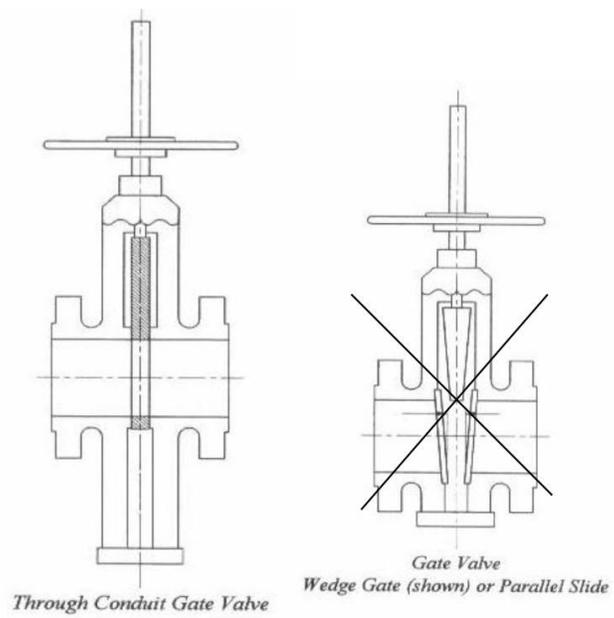


Figure 7-4: Gate Valves (Wint)

Bends

For pipeline to be piggable, pipeline engineers make sure to design pipeline' bends with a bend radius of a minimum three times pipe diameter. Where a pipeline is not designed for pigging, bend radii of 1 or 1.5 D are most common.

History of Pigs Technology

Different reasons to pig pipelines; ex. cleaning, inspection, batching or displacement. Pigging process has been around for a long time. However, the earliest project to retrieve pig information originated in 1959. T.D Williamsons introduced a caliper tool for detecting dents in pipeline in the same year and a "Cooley Tool" was under development by American Petroleum. Cooley tool used the magnetic flux leakage (MFL) technique (Woodley, 2011).

Companies were racing in pipeline tools development. M/s Tuboscope had a tool that could detect defects in down-hole casings where M/s Shell Oil Research was developing a "MacLean tool" that can detect pitting in down-hole casings. In 1962, Tuboscope developed a pig that is able to carry an array of remote field eddy current sensors through a pipeline after they obtained a license from Shell Oil Research for the MacLean tool. Unfortunately, the early test was unsuccessful as the MacLean tool was not able to detect known pits in test spools.

M/s Tuboscope has purchased the Cooley tool patent by Pan-American Petroleum. Therefore, the MacLean tool was abandoned and development switched to Cooley or MFL, which was branded LINALOG. This tool had only four sensors with no distance measurement wheels. It completed its first job in 1965. The pig positioning was identified using a coil on the pig that can detect permanent magnets where they have been placed in the pipe before the pig passed.

Until these days, pigs' development is client driven. In 1966, M/s Tuboscope delivered a prototype of 360° tool.

During these early days, pigs were used to target pipeline operators to make their maintenance more effectively. The transistor was still in its infancy, printed circuit boards were relatively unsophisticated with the circuits often hand-wired, and rare earth magnets were not available. The information recorded by the pigs were recorder using a roll of ultraviolet-sensitive paper via a “visicorder”. The recorded data did not give a suitable information of the status of the pipe unless it has a major metal loss, which will alert the operator to the site where they should dig.

In 1974, British Gas started to build its own tool on a 24-inch diameter, 12-channel analogue tool. To achieve the required accuracy, a high sensor density is required. This means typically ten sensors per pipe inch diameter with a high sampling frequency of 3.3 mm of pig travel. This amount of data could not be recorded onto analogue tapes. Therefore, a completely new approach was needed.

In 1977-1978, digital processing pack and 60-channel digital recorders began. As technology has advanced, data storage and computing power become cheaper, more compact and faster. These days, the digital recorders can store vast amount of pigging data in hundreds of dollars. Similarly, the cost of processing data dropped from around \$US 1 million in 1980s to hundreds or thousands of dollars nowadays. This has enabled companies with low financial support to enter the market.