UNIVERSITY OF TURKISH AERODYNAMIC ASSOCIATION

INSTITUE OF SCIENCE AND TECHNOLOGY

# A NEW PIG ROBOT NAVIGATION ALGORITHM FOR SEARCHING THE LEAKAGE AND CRASH POINTS IN THE UNDER SERVICE OIL-PIPELINE

MASTER THESIS

MOHAMMED FALAH HASAN ALKHATABE

INSTITUTE OF SCIENCE AND TECHNOLOGY

MECHANICAL AND AERONAUTICAL ENGINEERING DEPARTMENT

MASTER THESIS PROGRAM

July 2017

**UNIVERSITY OF TURKISH AERODYNAMIC ASSOCIATION**

**INSTITUE OF SCIENCE AND TECHNOLOGY**

# A NEW PIG ROBOT NAVIGATION ALGORITHM FOR SEARCHING THE LEAKAGE AND CRASH POINTS IN THE UNDER SERVICE OIL-PIPELINE

**MASTER THESIS**

**MOHAMMED FALAH HASAN ALKHATABE**

**1403730058**

**IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF SCIENCE IN MECHANICAL AND AERONAUTICAL ENGINEERING**

**Supervisor: Assist. Prof. Dr. Habib Ghanbarpourasl**

**July 2017**

Mohammed Alkhatabe, having student number 1403730058 and enrolled in the Master Program at the Institute of Science and Technology at the University of Turkish Aeronautical Association, after meeting all of the required conditions contained in the related regulations, has successfully accomplished, in front of the jury, the presentation of the thesis prepared with the title of: "A NEW PIG ROBOT NAVIGATION ALGORITHM FOR SEARCHING THE LEAKAGE AND CRASH POINTS IN THE UNDER SERVICE OIL-PIPELINE".

**Supervisor** : **Assist. Prof. Dr. Habib Ghanbarpourasl** ..........................
**The University of Turkish Aeronautical Association**

**Jury Members** : **Assist. Prof. Dr. Habib Ghanbarpourasl** ..........................
**The University of Turkish Aeronautical Association**

: **Assoc. Prof. Dr. Fahd Jarad** ..........................
**Çankaya University**

: **Assist. Prof. Dr. Yuri Alyeksyeyenkov** ..........................
**The University of Turkish Aeronautical Association**

**Thesis Defense Date:** 31.07.2017

## STATEMENT OF NON-PLAGIARISM PAGE

I hereby declare that all information in this document as my Master Thesis, which is titled: “A NEW PIG ROBOT NAVIGATION ALGORITHM FOR SEARCHING THE LEAKAGE AND CRASH POINTS IN THE UNDER SERVICE OIL-PIPELINE” has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

**24.07.2017**

**Mohammed Alkhatabe**

# ACKNOWLEDGEMENTS

First, I say thanks to our God, I would like to thank my supervisor Dr. Habib Ghanbarpourasl., who taught, helped, shared of his expertise and guided me patiently through my research term. I would like to thank my extended family especially my lovely parents and my special family especially my wife for their endless patience, pray and support.

I would like to thank my friends Eng. SALIM ALKTASH, Mr. LAITH TALIB, Dr. HAIDER ALMASOODI and RASOL ALSAIDI with my sincere gratitude. I would like to thank my country Al Iraq: the people and the government for giving me the opportunity to complete my Master's degree. Also, I sincerely thank the immense support that I have got from the Mechanical Engineering Graduate Program at the T.H.K University to achieve my M.Sc. degree study.

Finally, for all who gave or tried to give me encouragement and support without regrets get me from the beginning to the end of my studying period. Just ALLAH knows how much I owe them.

**July, 2017** **Mohammed Alkhatabe**

# CONTENTS

**ACKNOWLEDGEMENTS........................................................................IV**

**CONTENTS.......................................................................................... V**

**LIST OF TABLES ........................................................................... VII**

**LIST OF FIGURES .........................................................................VIII**

**ABSTRACT...........................................................................................XI**

**ÖZET...................................................................................................XIII**

**LIST OF SYMBOLS ......................................................................... XV**

**CHAPTER ONE ..................................................................................... 1**

**INTRODUCTION.................................................................................... 1**

1.1 BACKGROUND ABOUT THE DOMAIN OF PIPELINES AND PIG ROBOTS ................... 1
1.2 THE MAIN FEEDING METHODS OF A PIG ROBOT TO THE PIPELINES ........................ 4
1.3 COMPONENTS OF PIG ROBOT............................................................................ 6
1.4 LITERATURE REVIEW ......................................................................................... 6
    1.4.1 Review of integrated of navigation of PIG system..................................... 6
    1.4.2 Review of Kalman filters.............................................................................. 8
    1.4.3 Review of PIG robot's navigation algorithms based on Kalman filters...... 9
    1.4.4 Costs' review .............................................................................................. 12
1.5 OBJECTIVES........................................................................................................ 13
1.6 ORGANIZATION OF THE THESIS ......................................................................... 13

**CHAPTER TWO .................................................................................. 15**

**INERTIAL NAVIGATION SYSTEMS AND AIDING TECHNIQUES........... 15**

2.1 OVERVIEW OF AIDED INERTIAL NAVIGATION SYSTEMS .................................... 15
2.2 OVERVIEW OF REFERENCE FRAMES AND ATTITUDE PARAMETERIZATION......... 17
    2.2.1 Reference Frames ...................................................................................... 17
    2.2.2 Attitude Representations............................................................................ 19
    2.2.3 Reference Frame Transformations............................................................ 21
2.3 INERTIAL NAVIGATION SYSTEM (INS) FUNDAMENTALS .................................. 24
    2.3.1 INS Navigation Equations ......................................................................... 24
    2.3.2 Inertial Sensor Calibration and Measurement Error Compensation.......... 25
    2.3.3 INS Mechanization .................................................................................... 26
    2.3.4 INS Error Model ........................................................................................ 29
    2.3.5 Initial Alignment........................................................................................ 31
2.4 MEASUREMENT MODELS FOR AIDING SOURCES.............................................. 32
    2.4.1 Global Positioning System (GPS) ............................................................ 32
    2.4.2 Odometer and Non-Holonomic Constraints ............................................. 33

**CHAPTER THREE ............................................................................. 36**

**KALMAN FILTERS ........ 36**

3.1 KALMAN FILTER (KF) ........ 36
3.2 THE EXTENDED KALMAN FILTER (EKF) ........ 39
3.3 THE UNSCENTED KALMAN FILTER (UKF) ........ 43
  3.3.1 The Unscented Kalman Filter (UKF) ........ 44
    3.3.1.1 Unscented Transformations (UT) ........ 45
    3.3.1.2 Unscented Kalman Filter ........ 47
3.4 IMPLEMENTATION VARIATIONS ........ 48

**CHAPTER FOUR ........ 51**

**IMPLEMENTATION OF UKF FOR NAVIGATION OF PIG ROBOT ........ 51**

4.1 INTRODUCTION ........ 51
4.2 DYNAMIC OF PIG ROBOT WITH NON-HOLONOMIC CONSTRAINTS ........ 53
4.3 DEFINETION OF NAVIGATION PARAMETERS ........ 57
4.4 THE ODOMETER MEASUREMENT AND COUNTING SYSTEM ........ 59
4.5 THE COUNTING SYSTEM ........ 60
4.6 IMPLEMENTATION OF UKF FOR PIG ROBOT ........ 61

**CHAPTER FIVE ........ 65**

**SIMULATION STUDY ........ 65**

5.1 INTRODUCTION ........ 65
5.2 SENSORS PARAMETERS ........ 65
5.3 NOMINAL TRAJECTORY ........ 66
5.4 SENSORS OUTPUT ........ 72
5.5 INERTIAL NAVIGATION PARAMETERS ERRORS AND STANDARD DEVIATIONS ........ 75
5.6 NAVIGATED PARAMETERS ERRORS ........ 76
5.7 ERRORS OF INS/ODOMETER ........ 84
5.8 ERRORS OF INERTIAL NAVIGATION SYSTEM ........ 92

**CHAPTER SIX ........ 95**

**CONCLUSION AND RECOMMENDATIONS ........ 95**

6.1 BRIEF REVIEW OF THESIS ........ 95
6.2 THE RESEARCH CONTRIBUTIONS ........ 95
6.3 CONCLUSION ........ 96
6.4 RECOMMENDATIONS ........ 97

**REFERENCES ........ 98**

**CURRICULUM VITAE ........ 105**

# List of Tables

**TABLE 1. 1** COMPARISON OF SOME PIPELINE MONITORING SYSTEMS' COSTS............... 13

**TABLE 3. 1** EQUATIONS OF UNSCENTED KALMAN FILTER [35]................................... 48
**TABLE 3. 2** UNSCENTED KALMAN FILTER–ADDITIVE (ZERO MEAN) NOISE CASE [35]. 49

**TABLE 5. 1** PARAMETERS OF SENSORS. ....................................................................... 66
**TABLE 5. 2** NOMINAL VALUES OF PARAMETERS......................................................... 66
**TABLE 5. 3** INITIAL VALUE (OR INITIAL ERROR) OF NAVIGATION SYSTEM. .................. 75
**TABLE 5. 4** INITIAL STANDARD DEVIATION OF NAVIGATION SYSTEMS STATES. ........... 76
**TABLE 5. 5** SUMMERY OF POSITION ERRORS ............................................................. 94

# List of Figures

**FIGURE 1. 1** GENERAL PATTERNS OF IN-PIPE ROBOTS, [2]. ........................................... 1
**FIGURE 1. 2** THE MAIN KINDS OF PIG ROBOTS, [3]. ....................................................... 1
**FIGURE 1. 3** COMPARING PRICES OF ROBOT WITH COSTS OF HUMAN LABOR IN THE 1990S, [5]. ........................................................................................................ 2
**FIGURE 1. 4** DIFFERENT KINDS OF ROBOTS, [6]. .......................................................... 3
**FIGURE 1. 5** SOME UNWANTED JOBS AND MANY TROUBLES IN PIPELINE FAILURE, [6]. .. 3
**FIGURE 1. 6** PIG'S INSERTING METHODS, [7]. ................................................................ 4
**FIGURE 1. 7** EXAMPLES OF REAL LAUNCHING AND RECEIVING OPERATIONS, [8]. .......... 5
**FIGURE 1. 8** PIG'S LAUNCHER / RECEIVING SYSTEM, [9]. ............................................ 5
**FIGURE 1. 9** THE FLEXIBILITY OF PIG ROBOT THROUGH THE ELBOW POSITIONING, [10]. 5
**FIGURE 1. 10** EXAMPLES OF THE OIL PIPELINE INSPECTION ROBOTS, [11]. .................... 6
**FIGURE 1. 11** CONCEPTS DRAWING FOR THE ARTICULATED PROPOSED PIG ROBOT, [12].6
**FIGURE 2. 1** INVESTIGATION OF GYROSCOPE TECHNOLOGY [25]. ................................ 16
**FIGURE 2. 2** INVESTIGATION OF ACCELEROMETER TECHNOLOGY [25]. ........................ 16
**FIGURE 2. 3** REFERENCE FRAMES [60]. ....................................................................... 17
**FIGURE 2. 4** BODY REFERENCE FRAME [36]. .............................................................. 18
**FIGURE 2. 5** I-FRAME, E-FRAME AND N-FRAME [24]. ................................................... 22
**FIGURE 2. 6** N-FRAME, C-FRAME AND P-FRAME [24]. .................................................. 24
**FIGURE 2. 7** NON-HOLONOMIC CONSTRAINTS [44]. .................................................... 35
**FIGURE 3. 1** KALMAN FILTER ALGORITHM [44]. ........................................................ 39
**FIGURE 3. 2** EXTENDED KALMAN FILTER [77]. ........................................................... 41
**FIGURE 3. 3** RELATIONSHIP OF FILTERING OUTCOMES WITH LKF NOMINAL TRAJECTORY [44]. ............................................................................................................ 42
**FIGURE 3. 4** RELATIONSHIP OF FILTERING OUTCOMES WITH EKF NOMINAL TRAJECTORY [44]. ............................................................................................................ 43
**FIGURE 3. 5** THE CONCEPT OF THE UKF [24]. ............................................................. 44
**FIGURE 3. 6** BLOCK DIAGRAM OF THE UT [35]. ........................................................... 46
**FIGURE 3. 7** EXAMPLE OF COMPARING THE UT AND EKF PERFORMANCE FOR MEAN AND COVARIANCE PROPAGATING WITH ACTUAL SAMPLING [35]. ................................ 47
**FIGURE 3. 8** A HIGH LEVEL OF THE OPERATION OF THE UNSCENTED KALMAN FILTER [86]. ............................................................................................................ 50
**FIGURE 4. 1** PIPELINE INSPECTION GAUGE (PIG) ROBOT [52]. ...................................... 51
**FIGURE 4. 2** SAMPLES OF PIPELINE JOINTS [52]. ......................................................... 52
**FIGURE 4. 3** BLOCK DIAGRAM OF THE IMU MECHANIZATION PROCESS [88]. ................ 54
**FIGURE 4. 4** COMPARISON OF IMUS [52]. ................................................................... 55
**FIGURE 4. 5** DEFINITION OF PIG AXES [52]. ............................................................... 57
**FIGURE 4. 6** ODOMETERS OF SMART PIG ROBOT [52]. ................................................ 60
**FIGURE 4. 7** SCENARIO OF THE ALGORITHM FOR CORRECTED THE ESTIMATION STATE OF INS BY ODOMETER OUTPUT AND COUNTING SYSTEM. .......................................... 60
**FIGURE 4. 8** DIAGRAM OF SYSTEM IMPLEMENTATION. ................................................ 64

**FIGURE 5. 1** COMMAND FOR VELOCITY CONTROL SYSTEM (IN BODY COORDINATE)..... 67
**FIGURE 5. 2** COMMAND FOR ATTITUDE CONTROL SYSTEM. .......................................... 67
**FIGURE 5. 3** SPECIFIC FORCES ALONG X, Y AND Z DIRECTION. ..................................... 68
**FIGURE 5. 4** ANGULAR VELOCITY OF ROBOT RELATIVE TO INERTIAL COORDINATE SYSTEM. ........................................................................................................ 69
**FIGURE 5. 5** VELOCITY ALONG THE NORTH, EAST AND DOWN DIRECTION. .................. 69
**FIGURE 5. 6** PROJECTION OF VELOCITY IN THE BODY FRAME. ..................................... 70
**FIGURE 5. 7** ATTITUDE OF PIG. ........................................................................ 70
**FIGURE 5. 8** POSITION OF PIG. ........................................................................ 71
**FIGURE 5. 9** TRAJECTORY OF PIG. ..................................................................... 71
**FIGURE 5. 10** ACCELEROMETERS' ERRORS. ........................................................... 72
**FIGURE 5. 11** GYROSCOPES' ERRORS. ................................................................. 73
**FIGURE 5. 12** OUTPUT OF ODOMETERS ................................................................. 73
**FIGURE 5. 13** COUNTING SYSTEM OUTPUT (THE LENGTH OF PIPES). ............................. 74
**FIGURE 5. 14** COUNTING SYSTEM OUTPUT AT THE FIRST 120 SECONDS. ....................... 75
**FIGURE 5. 15** ERROR OF X-ACCELEROMETER BIAS AND ITS THREE-SIGMA BOUNDS. .... 76
**FIGURE 5. 16** ERROR OF Y-ACCELEROMETER BIAS AND ITS THREE-SIGMA BOUNDS. .... 77
**FIGURE 5. 17** ERROR OF Z-ACCELEROMETER BIAS AND ITS THREE-SIGMA BOUNDS...... 77
**FIGURE 5. 18** ERROR OF X-GYROSCOPE BIAS AND ITS THREE-SIGMA BOUNDS ............. 78
**FIGURE 5. 19** ERROR OF Y-GYROSCOPE BIAS AND ITS THREE-SIGMA BOUNDS. ............. 78
**FIGURE 5. 20** ERROR OF Z-GYROSCOPE BIAS AND ITS THREE-SIGMA BOUNDS. ............. 79
**FIGURE 5. 21** ERROR OF ROLL AND ITS THREE-SIGMA BOUNDS. ................................... 79
**FIGURE 5. 22** ERROR OF PITCH AND ITS THREE-SIGMA BOUNDS. .................................. 80
**FIGURE 5. 23** ERROR OF YAW AND ITS THREE-SIGMA BOUNDS. .................................... 80
**FIGURE 5. 24** ERROR OF THE NORTH VELOCITY AND ITS THREE-SIGMA BOUNDS. ......... 81
**FIGURE 5. 25** ERROR OF EAST VELOCITY AND ITS THREE-SIGMA BOUNDS. ................... 81
**FIGURE 5. 26** ERROR OF THE DOWN VELOCITY AND ITS THREE-SIGMA BOUNDS. .......... 82
**FIGURE 5. 27** ERROR OF LATITUDE AND ITS THREE-SIGMA BOUNDS. ............................ 82
**FIGURE 5. 28** ERROR OF THE LONGITUDE AND ITS THREE-SIGMA BOUNDS. .................. 83
**FIGURE 5. 29** ERROR OF THE ALTITUDE AND ITS THREE-SIGMA BOUNDS. ..................... 83
**FIGURE 5. 30** ERROR OF THE X-ACCELEROMETER'S BIAS AND ITS THREE-SIGMA BOUNDS. ........................................................................................................ 84
**FIGURE 5. 31** ERROR OF THE X-ACCELEROMETER'S BIAS AND ITS THREE-SIGMA BOUNDS. ........................................................................................................ 85
**FIGURE 5. 32** ERROR OF THE X-ACCELEROMETER'S BIAS AND ITS THREE-SIGMA BOUNDS. ........................................................................................................ 85
**FIGURE 5. 33** ERROR OF THE X-GYROSCOPE'S BIAS AND ITS THREE-SIGMA BOUNDS.... 86
**FIGURE 5. 34** ERROR OF THE Y-GYROSCOPE'S BIAS AND ITS THREE-SIGMA BOUNDS.... 86
**FIGURE 5. 35** ERROR OF THE Z-GYROSCOPE'S BIAS AND ITS THREE-SIGMA BOUNDS. ... 87
**FIGURE 5. 36** ERROR OF THE ROLL AND ITS THREE-SIGMA BOUNDS. ............................ 87
**FIGURE 5. 37** ERROR OF THE PITCH AND ITS THREE-SIGMA BOUNDS. ........................... 88
**FIGURE 5. 38** ERROR OF THE YAW (HEADING) AND ITS THREE-SIGMA BOUNDS. ........... 88
**FIGURE 5. 39** ERROR OF THE NORTH VELOCITY AND ITS THREE-SIGMA BOUNDS. ......... 89
**FIGURE 5. 40** ERROR OF THE EAST VELOCITY AND ITS THREE-SIGMA BOUNDS. ............ 89

**FIGURE 5. 41** ERROR OF THE DOWN VELOCITY AND ITS THREE-SIGMA BOUNDS. .......... 90
**FIGURE 5. 42** ERROR OF THE LATITUDE AND ITS THREE-SIGMA BOUNDS. ..................... 90
**FIGURE 5. 43** ERROR OF THE LONGITUDE AND ITS THREE-SIGMA BOUNDS. .................. 91
**FIGURE 5. 44** ERROR OF THE ALTITUDE AND ITS THREE-SIGMA BOUNDS. ..................... 91
**FIGURE 5. 45** ERROR OF THE ATTITUDE (ROLL, PITCH, AND YAW)............................... 92
**FIGURE 5. 46** ERROR OF THE VELOCITY (NORTH, EAST, AND DOWN VELOCITIES)......... 92
**FIGURE 5. 47** ERROR OF THE LATITUDE, LONGITUDE, AND ALTITUDE. ......................... 93

# Abstract

## A new PIG robot navigation algorithm for searching the leakage and crash points in the under service oil-pipeline

Mohammed Alkhatabe
Master. Department of Aeronautics and Mechanical Engineering
Supervisor: Assist. Prof. Dr. Habib Ghanbarpourasl.
July, 2017, 123 Pages

**Abstract**

The field of robotics applications was rapidly growing in the last decade and continues to develop. The computational challenges of the robotic applications and translations of actions using sensors have helped to reduce the costs and size of any robot. The Pipe Inspection Gauge (PIG) robot is one of the most challenging fields for the robotics applications in pipeline-based applications, which has been used for many years to perform various maintenance operations in oil and gas pipelines. The PIG robot can inspect different parameters of pipeline during its journey. Although PIG robots use many sensors to detect the required pipeline parameters, matching these data with the corresponding pipeline location represents a very important parameter. High-end, tactical-grade inertial measurement units (IMUs) uses in the pigging applications to locate the detected problems of the pipeline by using other sensors, and to reconstruct the trajectories of the PIG robot.

In this project, an integrated navigation system algorithm developes as a navigation system based on Unscented Kalman filter (UKF) for Pipe Inspection Gauge robot. Unfortunately, the errors of the inertial navigation system (INS) increase during the time due to, the uncalibrated and random errors of IMU’s sensors. There are many solutions for reducing the errors of the navigation system. Where, the system can use a very high tactical grid IMU, but in this case, the cost of the system will be high. The GPS in the most famous sensors can stabilize the errors of the INS, but in the isolated environments like inside of the pipeline, there is no GPS signal. Some researchers have used new methods such as adding odometer as an external sensor, the Non-holonomic constraints for the INS, the Zero-velocity Updates and Coordinate Updates methods for stabilizing the errors of the system. Furthermore, there are many smoothing methods such as (KF) and (UKF) for reducing the errors, but always all methods only can reduce the rate of the errors. The solution in this thesis aims to stabilize the errors of the navigation system for a moving (PIG) robot by integrating of the IMU with an odometer and counting the number of the known length pipes, in an assumption oil pipeline, which joints between two locations in Karbala city in Iraq. The geometry of the pipeline contains the latitude, longitude and altitude of the pipeline in any position. The simulation of the PIG robot moves inside the pipeline is with known angular velocity and specific acceleration. The IMU’s data is simulated by adding some biases and noise to the sensors’ output. Furthermore, the output of the odometer is generated by adding noise to the output of the odometer. The algorithm of the navigation system is applied to the output of the IMU. The output of the navigation system is compered only against the INS and INS/odometer integration. So the navigation errors are discussed in the final part of the thesis.

**Key words**: Pipe Inspection Gauge (PIG) robot, Inertial Measurement Units (IMU), Kalman Filter (KF), Unscented Kalman Filter (UKF), Inertial Navigation System (INS), Integrated Navigation System (INS).

# Özet

## Hizmet alti petrol boru hattındaki kaçak ve kaza noktalarını aramak için yeni bir PIG robotunrn navigasyon algoritması

Yüksek Lisans Tezi, Makine Mühendisliği Anabilim Dalı
Tez Danışmanı: Yrd. Doç. Dr. Habib Ghanbarpourasl.
Temmuz, 2017, 123 sayfa

### Özet

Son on yılda robotik uygulama alanları hızla büyümüş ve gelişmeye devam etmektedir. Robotik uygulamalarını işleme dönüştürmek için yapılan hesaplamalardaki zorluklar ve sensörleri kullanarak hareketlerin veriye dönüştürülmesi, herhangi bir robotun maliyetlerini ve boyutunu azaltmaya yardımcı olmuştur. Boru Denetleme Aracı (PIG) robotu, uzun yıllardır petrol ve gaz boru hatlarında çeşitli bakım işlemlerini yapmak için kullanılan boru hattı tabanlı uygulamalarındaki robotik uygulamalar içinde en zor alanlardan biridir.PIG robotu, yolculuğu sırasında boru hattının farklı parametrelerini denetleyebilir. PIG robotları, gerekli boru hattı parametrelerini algılamak için birçok sensör kullanmasına rağmen, bu verileri boru hattındaki ilgili konum ile eşleştirmek çok önemli bir parametreyi ifade etmektedir. En son teknolojiye sahip, taktik sınıf atalet ölçüm birimleri (IMU), diğer sensörleri kullanarak boru hattında algılanan sorunları bulmak ve PIG robotunun yörüngelerini yeniden oluşturmak için tarama uygulamalarında kullanılır.

Bu projede, Entegre edilmiş bir navigasyon sistemi algoritması , Boru Muayene Gauge robotu için Sezgisiz Kalman Süzgecine (UKF) dayanan bir navigasyon sistemi olarak geliştirilmektedir .Ne yazık ki, Inersiyal Navigasyon Sistemleri (INS) hataları, IMU sensörlerinin kalibre edilmemiş ve rastgele hataları nedeniyle kullanım zamanına bağlı olarak artmaktadır .Navigasyon sisteminin hatalarını azaltmak için pek çok çözüm yolu bulunmaktadır. Sistem, çok yüksek bir taktik sınıf atalet ölçüm birimi IMU kullanabilir, ancak bu durumda, sistemin maliyeti yüksek olacaktır. En ünlü sensörlerde yer alan GPS, INS hatalarını dengeleyebilir ancak boru hattının iç kısmı gibi izole edilmiş ortamlarda GPS sinyali yoktur. Bazı araştırmacılar sistemin kararlı hale getirilmesiiçin kilometre sayacını harici bir sensör olarak ekleme , INS için holonomik olmayan sabitleyiciler, sıfır hız güncellemeleri ve koordinat güncellemeleri gibi yeni yöntemler kullanmışlardır.Ayrıca, hataları azaltmak için

(KF) ve (UKF) gibi birçok düzleme yöntemi vardır, ancak her zaman tüm yöntemler yalnızca hataların oranını düşürebilir. Bu tezde ortaya atılan çözüm, IMU'nun bir kilometre sayacıyla birleştirilmesi ve uzunluğu biliinen borularının sayısının sayılması yoluyla hareket eden (PIG) robotu için navigasyon sisteminin, Irak'taki Kerbela şehrindeki iki konum arasındaki bağlantıların varsayımsal bir petrol boru hattında hatalarını stabilize etmeyi amaçlıyor.Boru hattının geometrisi, herhangi bir konumdaki boru hattının enlem, boylam ve yüksekliğini içerir. PIG robotunun simülasyonu boru hattı içerisinde, bilinen açısal hız ve belirli ivmelenme ile hareket eder. IMU'nun verileri, sensörlerin çıktılarına bazı sapma ve parazitler ekleyerek simüle edilir.Ayrıca kilometre sayacının çıktısı, kilometre sayacının çıkışına parazitler ekleyerek oluşturulur.Navigasyon sisteminin algoritması IMU'nun çıktısına uygulanır. Navigasyon sisteminin çıktısı yalnızca INS ve INS / kilometre sayacı entegrasyonu ile karşılaştırılmıştır .Dolayısıyla navigasyon hataları, tezin son bölümünde tartışılmaktadır.

**Anahtar kelimeler**: Boru Denetleme Aracı (PIG) robotu, Atalet Ölçüm Birimleri (IMU) Kalman Süzgeci (KF), Sezgisiz Kalman Süzgeci (UKF), Inersiyal Navigasyon Sistemi (INS), Entegre Navigasyon Sistemi (INS).

## List of Symbols

| Symbol | Definition |
|---|---|
| $(.)-1$ | Matrix and quaternion inverse |
| $(.)^T$ | Matrix transpose |
| $(.\times)$ | Cross-product or skew-symmetric form of a vector |
| $g^l$ | Gravity in local level frame |
| $\mathrm{v}_\mathrm{n}\mathrm{v}_\mathrm{e}\mathrm{v}_\mathrm{d}$ | Velocities in north, east and down, respectively |
| $\times$ | Cross product |
| $\star$ | Quaternion product |
| $E[.]$ | Expectation of [.] |
| $diag(.)$ | Diagonal matrix form of a vector |
| $\theta, \phi, \psi$ | Pitch, roll, and heading angle |
| $\varphi, \lambda, h$ | Latitude, Longitude, and height, respectively |
| $\psi$ | The platform tilt error |
| $\Psi$ | The skew-symmetric matrix of the rotation vector pertaining to the platform tilt error $\psi$ |
| $\omega$ | Angular velocity |
| $\omega_e$ | Earth's angular velocity ($7.2921158 \times 10^{-5}\ rad/s$) |
| $i$ | ECI frame |
| $e$ | ECEF frame |
| $n$ | Navigation (NED) frame |
| $C_x^y$ | Transformation matrix from a frame, $x$, to a frame, $y$ |
| $\hat{x}$ | Estimated states vector |
| $\bar{x}$ | Mean state vector of vector $x$ |
| $(k\|k-1)$ | Denotes a quantity at time $k$ just before the measurement update |
| $(k\|k)$ | Denotes a quantity at time $k$ just after the measurement update |

## Abbreviations

| | |
|---|---|
| BKF | Backward Kalman Filter |
| CUPT | Coordinate Update |
| DCM | Direct Cosine Matrix |
| DGPS | Differential Global Position System |
| DR | Dead-Reckoning |
| ECEF | Earth-Centered Earth-Fixed |
| EKF | Extended Kalman Filter |
| FKF | Forward Kalman Filter |
| FOG | Fiber Optical Gyroscope |
| GINS | Global Navigation Satellite Systems |
| GPS | Global Position System |
| GRV | Gaussian Random Variable |
| IMU | Inertial Measurement Unit |
| INS | Inertial Navigation System |
| KF | Kalman Filter |
| LKF | Linearized Kalman Filter |
| LOS | Line-Of-Sight |
| MEMS | Micro Electro-Mechanical system |
| NDT | Non-Destructive Testing |
| NED | North East Down |
| Odo | Odometer |
| PDF | Probability Density Function |
| PF | Particle Filter |
| PIG | Pipe Inspection Gauge |
| RF | Radio Frequency |
| RLG | Ring Laser Gyroscope |

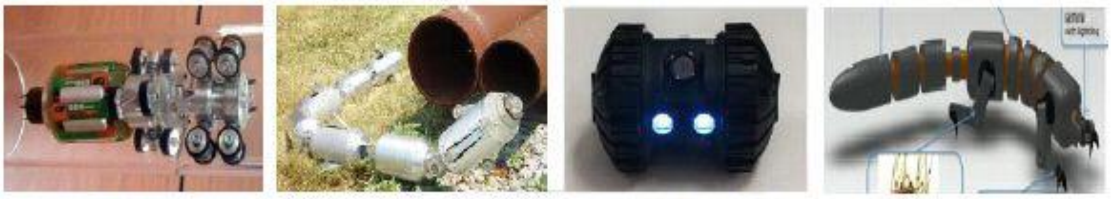
| | |
|---|---|
| RTS | Rauch-Tung-Striebel |
| RTSS | Rauch-Tung-Striebel Smoother |
| RV | Random Variable |
| SICK | 2D or 3D Laser Scan |
| SINS | Strapdown Inertial Navigation Systems |
| SLAM | Simulation Localization And Mapping |
| STD | Standard Deviation |
| TFS | Two-filter Smoother |
| UT | Unscented Transformation |
| UKF | Unscented Kalman Filter |
| ZUPT | Zero-Velocity Update |

# CHAPTER ONE

## INTRODUCTION

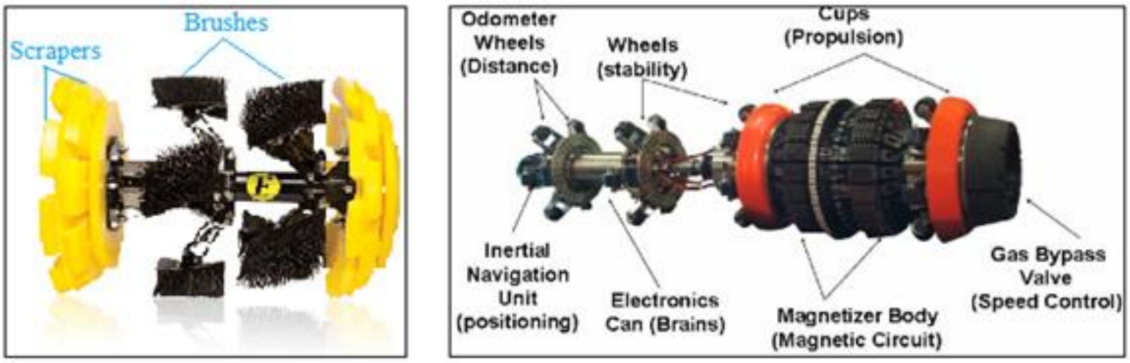
### 1.1 Background about The Domain of Pipelines and PIG Robots

PIG robot is well known as an old term in the pipeline field, which means pipe inspection gauge robot. PIG robots are equipment/devices, which have several patterns as shown in Fig. 1.1. The PIG robots can be fed/inserted through the pipeline and travel along of its length. The PIG robot is driven forward or backward by a differential pressure across the equipment. In now days, the term PIG is being used to define any equipment/device create to pass through a pipeline driven by the fluid of the pipeline. Mostly, they are classified into two categories as shown in Fig. 1.2, the first is named Utility PIG that is employed for cleaning, and the second is named Smart/intelligent PIG that collects and provides detailed information of the pipeline like information of corrosion, thickness, leakage and crash points and the position of each detected problem [1].



Wheels Pattern Snak Pattern Ball Pattern Walking Pattern

**Figure 1.1:** General patterns of in-pipe robots, [2].


Brushes
Scrapers
Odometer Wheels (Distance)
Wheels (stability)
Cups (Propulsion)
Inertial Navigation Unit (positioning)
Electronics Can (Brains)
Magnetizer Body (Magnetic Circuit)
Gas Bypass Valve (Speed Control)

(a) Utility PIG robot (b) Smart/intelligent PIG robot

**Figure 1.2:** The main kinds of PIG robots, [3].

An automatically robot domain operated machine became instead of a human effort. At first, it was difficult to imagine; especially with the view of appearance or perform functions in a humanlike behavior. With time, the robotics' design, construction, and operation have developed. A robot in now days has a brain of its own. The advanced in robot domain mainly depended on micro electro-mechanical system (MEMS) based on some computational smoothing methods, which contributed reducing the cost of the intelligent parts and developing of their abilities, in addition to reduce the robot's size and cost. In all over the world in the first half of 2008 there were more than one million robots in operation, with roughly half in Asia, 32% in Europe, 16% in North America, 1% in Australia and 1% in Africa [4]. In industrial and commercial fields, robots were used widely through the decade of the 1990s, these robots characterized with greater accuracy and more reliable than human with no labor cost as Fig. 1.3 indicates. Based on the type of the job doing by robots can classify robots into roughly two classifications. The first category contains tasks which a robot can do with higher productivity, accuracy, or endurance than the ability of a human, and the second category includes of doing dirty, dangerous or dull jobs which humans find unwanted. Some examples of different kinds of robots, which are currently in service, as shown in Fig. 1.4, and some unwanted jobs shown in Fig. 1.5.

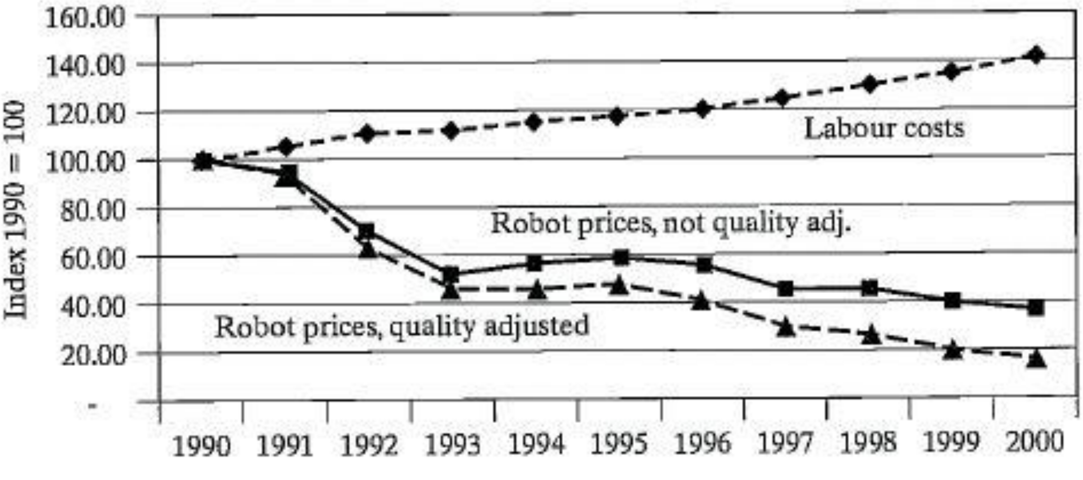


**Figure 1.3:** Comparing prices of robot with costs of human labor in the 1990s, [5].

(a) Manufacturing assembly robots
(b) Pick and putting robot for electronics assemblage
(c) Surgery robot

(d) Domestic robot for cleaning
(e) Robot for military purposes
(f) (Asimo) Humanoid robot

**Figure 1.4**: Different kinds of robots, [6].

(a) an explosive gas pipeline
(b) a clog city sewer
(c)Worker unclogging sewer, India, Chennai

**Figure 1.5:** Some unwanted jobs and many troubles in pipeline failure, [6].

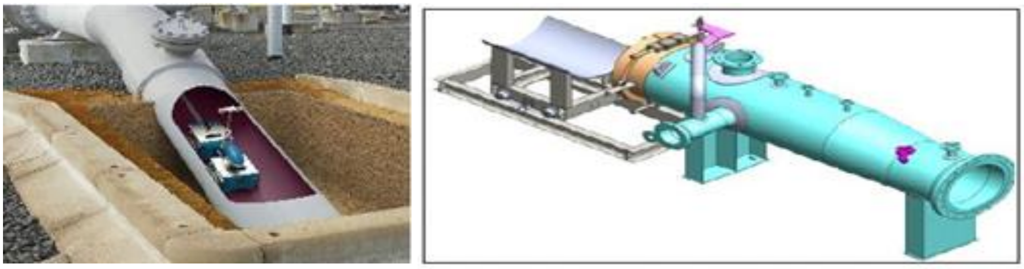
In pipeline inspection robots domain, robots became instead of a human in many areas due to the widespread of using pipelines in major utilities for a long time, which start represent the most challenging areas to maintenance their pipelines. Over billions of places, from the huge plants to an individual house are using pipelines in many fields and for many reasons with the above-mentioned; problems in pipelines and as shown above in Fig. 1.4 and Fig. 1.5, robots became extremely instead of people for pipelines maintenance purposes.

In 1893, Dmitri Mendeleev suggested pipelines to transport Petroleum, and then these pipes started to get widely used in all over the world. In 2007, the total length of the gas and oil pipelines in the world was nearly two millions km, and the length of oil/gas pipelines in the United States reached 793,285 km. Generally, pipelines represent the most economical way to transport large amounts of natural gas or oil over land. Comparison with the railroad, they have lower cost per unit with higher capacity.

The oil pipes are manufactured from steel or plastic tubes with inner diameter typically varying from 4 to 48 inches. Pipelines are mostly buried underground at a typical depth of about 3 to 6 feet. The oil is usually injected and kept in motion by pump stations along the pipeline; with flow rates about 1 to 6 m/s. Multi-product pipelines are used to transport two or more different products alternately on the same pipeline. Usually there is no physical separation between the different products in multi-product pipelines. Therefore some mixing of adjacent products causes, producing interface. Removing this interface from the pipeline is at receiving facilities and segregated to prevent contamination. Oil contains varying amounts of paraffin, or wax. In the colder weathers, wax accumulation may happen within a pipeline. Often these pipelines are inspected and cleaned using pipeline inspection gauges [6].

## 1.2 The Main Feeding Methods of A PIG Robot to The Pipelines

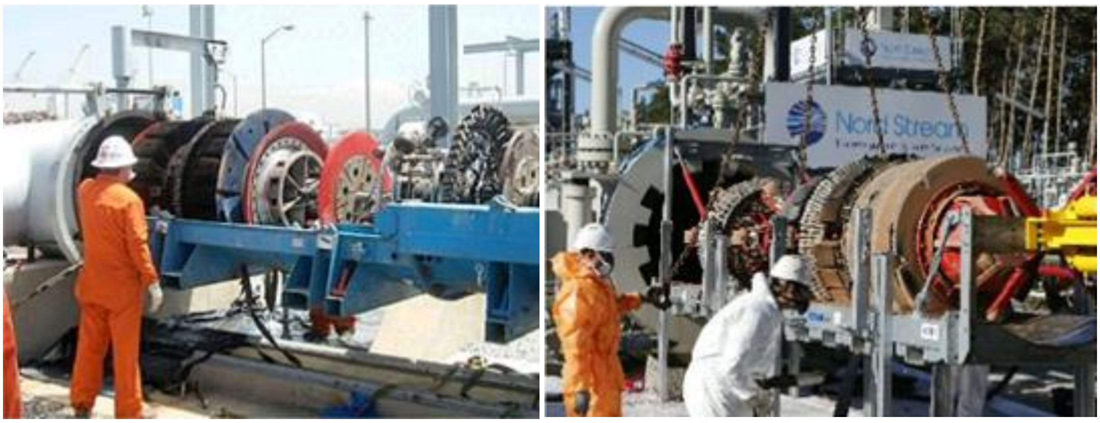
PIG robot is pushed in and driven along the pipeline by using the differential pressure of the fluid flow. At the starting point of the PIG robot’s surveying journey, it is put in the launcher in deferent methods as shown in Fig. 1.6 with the same procedure to launch or to receive it. Similar scenario is done at the beginning and the end of the PIG robot’s surveying journey in a special launcher/receiving system as shown in Fig. 1.7 and Fig. 1.8. In addition, the PIG robot should has some significant mechanical features like, it should be fitted with the inner diameter of the pipeline with enough tolerance to move smoothly, and it should has a good design with a high flexibility to pass seamlessly through the elbow positioning Fig. 1.9.



(a) Lateral feeding method

(b) Injection method

**Figure 1.6:** PIG’s inserting methods, [7].



(a) PIG robot under launching operation (b) PIG robot under receiving operation

**Figure 1.7:** Examples of real launching and receiving operations, [8].

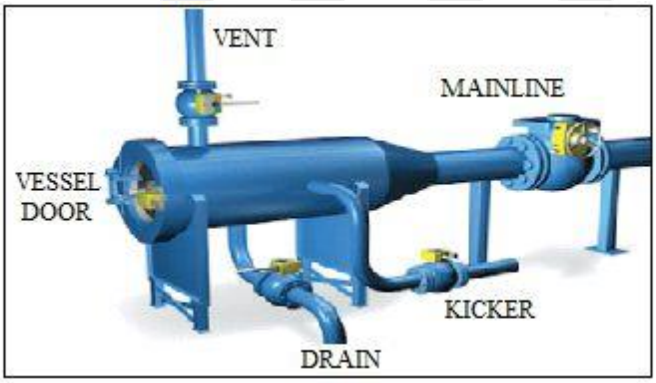


**Figure 1.8:** PIG's launcher / receiving system, [9].

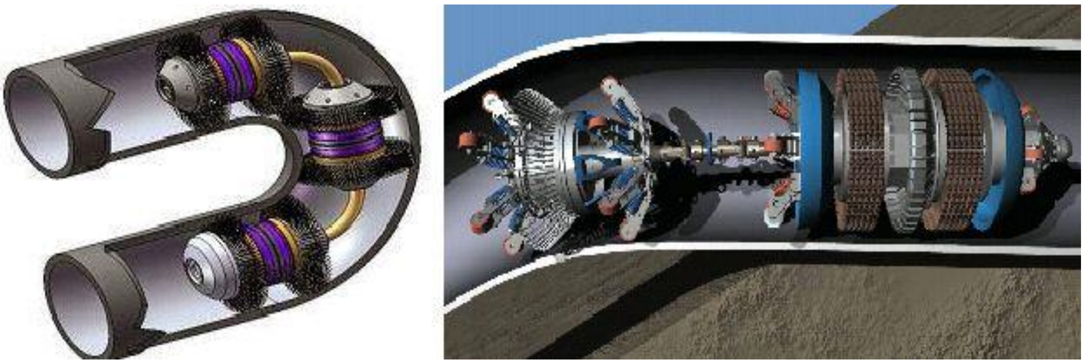
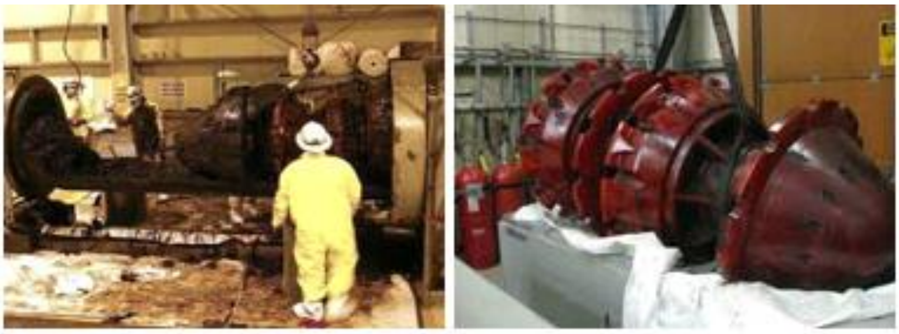


**Figure 1.9:** The flexibility of PIG robot through the elbow positioning, [10].

Varies oil-pipeline inspection robots are available like PIGS also known as SCRAPERS, as shown in Fig. 1.10 [11]. These devices are usually launched from PIG-launcher stations and travel into the pipeline to be received at any other station down-stream; Cleaning wax accumulates and material that may have deposited along the line and to locate any pre or advanced detected or undetected damages in wall of pipeline during the journey of a PIG robot.



(a) PIGS pipeline inspection robot. (b) SCRAPERS pipeline robot.

**Figure 1.10:** Examples of the oil pipeline inspection robots, [11].

### 1.3 Components of PIG Robot

By summarizing the details of Fig. 1.2b in the Fig. 1.11 that shows the smart PIG has three main intelligent units one of these consists the accelerometers and the gyroscopes, which is well known inertial measurement unite (IMU). Other one sensors unit that consists the suitable sensors for the robot's task such as odometer sensor, counting sensor, the Global Positioning Systems (GPS) sensor and other sensors which can be added due to the robot's task. The last unit has termed control or process unit that includes integrating of the sensor measurements data using fusion algorithm based on one of Kalman filters to get the robot fully autonomous.

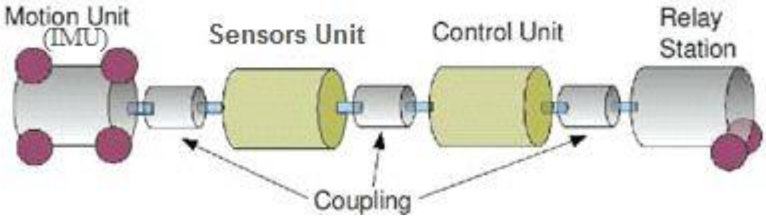


**Figure 1.11:** Concepts drawing for the articulated proposed PIG robot, [12].

### 1.4 Literature Review

#### 1.4.1 Review of integrated of navigation of PIG system

The PIG robots are well-known as fully autonomous robots, which depends mainly on Inertial Navigation Systems, that were widespread to be applied as either the associated or the main controlling instruments for vehicles navigations in long-range travelling, like submarines and airliners. It has now been 47 years since 1970s, when optical gyroscope based INS was used as the well-functioned complement in the integrated systems of radio navigation in flying applications [13]. With the enormous advance of Micro-Electro-Mechanical Systems (MEMS) inertial sensors and GPS, high-end, tactical-grade and decreasing in cost of Inertial Measurement Units (IMUs) have contributed widely interests in industrial, commercial and civilian fields in the last decade. It has been implemented and proved that the INS/GPS integration is the best technique for vehicular navigation [14]. In now days, promising potentials using INS exist in civilian and commercial applications for auto pilot, drone flying, personal navigation, unnamed vehicles, horizontal drilling, etc. [15, 16, 17 and 18].

The Dead-Reckoning (DR) nature of the stand-alone INS causes the error accumulation of navigation parameters. Furthermore, low-cost INS faces the problem of large and unpredictable sensor noises and errors [19]. Therefore, aiding navigation information becomes fundamental to cope these weaknesses. GPS, a Radio-Frequency (RF) signal-based system, is able to provide absolute positioning solutions with long-term accuracy under any weather conditions [20]. But, this rendering is usually discontinued by frequent signal outages. Therefore, it is considered not suitable to be used in isolated environments like mines and channels. The integration of INS and GPS takes advantage of the integral features of both systems and outperforms either single system run alone [21]. Since the last decade, different integration strategies, i.e. loose-coupled, tightly-coupled, and deeply-coupled INS/GPS integrations, have been researched and developed [22].

By depending on Line-Of-Sight (LOS) measurements, the high-accuracy, continuous GPS positioning updates are not available in the isolated environments like tunnels, mines or indoor areas. At these conditions, the information from alternative navigation-related techniques must be integrated with the stand-alone INS to stabilize the navigation error increase. Like them, the aiding performance of odometers, magnetometers, and non-holonomic constraints are most commonly used when GPS signals are discontinuous in land-vehicle navigation [23, 24]. Zero-Velocity Update (ZUPT) is another active method to enhance the navigation accuracy by stabilizing the increasing velocity errors with appropriately selection time intervals [25]. Moreover, Coordinate Update (CUPT), sometimes available at certain predetermined surveying stations (i.e. control points), is capable of aiding to enhance the navigation performance and get high accuracy positioning measurements. The integration strategies with INS and the aiding techniques have been demonstrated to be workable for pipeline surveys, pedestrian navigation, and vertical mine shaft surveys [18, 26 and 27].

### 1.4.2 Review of Kalman filters

Kalman Filter (KF) is recognized as the classic real-time estimation method to integrate multi-sensor information from INS and aiding sources. In the KF context of integration systems, INS provides the predictions as well as the system knowledge, while the aiding sensors provide the measurement updates. Extended Kalman Filter (EKF) is utilized to resolve the nonlinearity problem in the INS navigation equations; it simply applies the Taylor series expansion on the nonlinear system along with observation equations, and takes terms to the first order, where the Probability Density Function (PDF) is approximated by a Gaussian distribution [28]. KF is a recursive algorithm that implements a series of prediction and measurement update steps to obtain the optimal estimates based on minimum variance criterion [29]. It will only work in prediction mode during measurement gaps where the navigation solution accuracy degrades rapidly with time. As a result, this performance cannot meet the accuracy requirements of several navigation and surveying applications. Hence, post-processing methods such as backward smoothing can be employed in this case to yield better navigation solutions.

Furthermore, the EKF simply uses the Taylor series expansion for the nonlinear system along with the observation equations, and it is specialist to handle only to the first order terms while the probability density function PDF is approximated depending on a Gaussian distribution. In practice however, EKF has shown several limitations and easily exhibits divergent characteristics, [30, 31 and 32].

Julier and Uhlmann were pioneered in the development of the unscented transformation (UT) to approximate two of the first statistical moments [33, 34]. The unscented Kalman filter (UKF) was developed based on UT with the underlying supposition that approximating a Gaussian distribution is easier compering with approximating a nonlinear transformation [33, 35]. The UKF depends on using deterministic sampling to approximate the state distribution as a Gaussian Random

Variable (GRV). The sigma points are selected to capture the true mean and covariance of state distribution and are transmitted through the nonlinear system. The following mean and covariance are then calculated from the transferred sigma points. The UKF determines the mean and covariance accurately to the second order [34], while the EKF is only capable to obtain first order accuracy [34]. Thus, the UKF provides superior state estimates for nonlinear systems [35]. However, UKF needs calculation the new set sigma points at each sample time that must involves using a matrix square root for the state covariance matrix. While the square-root of covariance matrix is an integral part of the UKF, it is still the full covariance which is recursively updated. In the Square-root form of UKF SR-UKF implementation, square-root matrix of the state covariance will be transmitted directly, avoiding the need to re-factorize at each time step [36]. SR-UKF uses QR factorization method, Cholesky of the Rank 1 update and efficient least square solution for linear systems [37, 38 and 39].

### 1.4.3 Review of PIG robot's navigation algorithms based on Kalman filters

Many studies have been made to get a reliable positioning method for various applications of PIG robot by fusing of INS with different aided systems by using various fusion algorithms. In some research works, Kalman filters were used as a tool for reducing the size of inertial measurement unit IMU with maintaining on its quality and to create navigation algorithm for processing and reducing the sensors errors to get the navigation performance of a PIG robot with a high accuracy. Most pipeline inspection applications have become an important integral part of life, therefore fully autonomous mobile PIG robot becomes necessary.

**Rauch, *et al.*, (1965), [40].** Have been applied the Rauch-Tung-Striebel Smoother (RTSS) in the navigation due to its robustness and effectiveness. The RTSS does not require the process of the full-scale Backward Kalman Filter (BKF). By utilizing all the information stored in the Forward Kalman Filter (FKF), the RTSS recursively updates the smoothed estimate and its covariance in a backward sweep.

**Fraser and Potter, (1969), [41].** Proposed that the fixed-interval smoother can be accomplished by a combination of two Kalman filters manipulated forward and backward, i.e. FKF and BKF, using a series of convenient discrete-time equations. It has been demonstrated that the aforementioned Two Filter Smoother (TFS) and the

RTS smoother are mathematically equivalent in linear cases [42]. However, the traditional TFS was originally designed for linear systems. Therefore, it was not applicable for INS-based multi-sensor systems because of the high nonlinear characteristics in the INS navigation equations. The further attempt of applying the common EKF both forward and backward failed to accurately estimate the smoothing INS error states. This problem was resolved by a revised smoothing algorithm that was proposed specifically for pipeline surveys using inertial measurements units [43]. The main idea in such modification was that the BKF nominal trajectory is assumed to track both the FKF prediction and update results rather than the predictions only [14].

**Hang Liu, (2009), [44].** Employed Kalman filter as the real-time estimation method to fuse the multi-sensor information, optimal smoothing has been utilized as the post-processing methodology to provide better navigation solutions in stand-alone mode for a PIG robot. The optimal smoothing included utilizing and evaluating two different fixed-interval smoothing algorithms. The first one was the Two Filter Smoother TFS and the other was the Rauch-Tung-Streibel Smoother RTSS. The TFS is performed by combining the results of Forward Kalman Filtering FKF and Backward Kalman Filtering BKF through minimizing the smoother error covariance without depending on the GPS signal. The integration strategies of INS and the aiding techniques; odometers, non-holonomic constraints, Zero-velocity Updates (ZUPTs) and Coordinate Updates (CUPTs) were proved to be applicable and effective to overcome the problem of INS time-dependent error. The achieved results of the used INS-based all the aiding applications mentioned earlier showed that the TFS substantially improve the position estimation accuracy over the corresponding filtered solution. Furthermore, the TFS' estimation efficiency was compared with the commonly used RTSS.

**Wasim Al-Masri, (2016), [45].** According to the accumulated error in the INS and reduced inertial sensor system RISS solution, proposed the INS to be combined with velocity constraints, and with the detected pipe length as measurements to correct the solution of INS. So, to achieve RISS solution corrected, proposed the RISS to be combined only with the detected pipe length as measurements. And setup an experimental, with a prototype of the in-pipe robot, to exam and validate the proposed algorithms based on EKF with a real pipe. The proposed algorithms' accuracy was around  after sensor fusion where a sensor fusion was proposed to be based on EKF.

**D. Chatzigeorgiou, *et al.*, (2015), [46].** Was proposed a new system for in-pipe leak detection. Detection was proposed to be depended on the presence gradient in a pressure in the neighborhood that contains the leak. So, the study based on EKF algorithms to validate the concepts by construction a prototype to evaluate the systems performance under real environment conditions by setting up an experiment in the laboratory.

**J.-H. Kim, *et al.*, (2010), [47].** Was presented the design and application of a single-modulated fully autonomous mobile pipeline exploration robot (FAMPER). The mechanism of FAMPER provides for the first-rate mobility in horizontal as well as vertical pipelines, and to achieve the system fully autonomous, the study was proposed the system architecture enables FAMPER.

**H. Choi and S. Ryew, (2002), [48].**Was presented a PIG robot navigation algorithm for internal inspection of underground urban gas pipelines, where the PIG robot was developed to be employed as a mobile platform for visual and Non-Destructive Testing (NDT) of the pipeline nets. In the work of this study, the robot was constructed as an articulated structure to be similar as a snake and was provided with a tether cable.

**A. C. Murtra and J. M. M. Tur, (2013), [49].** Was proposed for visual/inertial localization of a mobile robot in sewer pipe net a complementary technique. In the proposed system of this study, the fusion of data was based on the graph of simultaneous localization and mapping (SLAM) framework, where it was included usage sensor data from a cable encoder and from an IMU. The proposed system was so suited to be considered as a complementary technique to solve situations comparing with visual feature tracking which could fail, due to the situations of environment.

**S. Bonnabel and E. Salan, (2011), [50].** Was proposed solution that includes using measurements of GPS, gyroscope and an accelerometer to provide a nonlinear observer, which is used to develop a solution of a vehicle navigation. This nonlinear observer has three subfilters; the first one has two gains and estimates yaw angle and gyroscopes bias. The second subfilter has one gain and the third one has one gain. The second subfilter is to estimate the accelerometer’s scaling. The third one is to give estimates of the vehicle’s velocity and position. The filtering stage proposed in this study was included the non-holomonic constraints to get extra enhancement for the estimates. Under testing, the algorithm showed reliable results. However, the accelerometer bias was not taken into the consideration of the algorithm and it also did not use the quaternion approach which has stability in finding vehicle’s attitude.

**A. Brandt and J. Gardner, (1998), [51].** Was taken the non-holomonic constraints into the consideration , where these constraints were fused in a navigation solution for land vehicle and they were depended on the direction of the vehicle relative to the earth and the attitude of the vehicle relative with its velocity. The main cause of using these constraints was especially to enhance the estimates of velocity and position.

**Hussein Sahli, (2016), [52].** Handled with robot navigates in a small diameter pipeline. In this case, the challenge was how to exceed the usage of large sizes IMUs with the same accuracy. So, there was using an augmenting IMU with odometers. In addition , was presented a new methodology to enable using low-cost IMUs with employing the EKF and the pipeline junctions detection system to increase the accuracy of the navigation parameters and to decrease the total root mean square errors even during the unavailability of above ground markers. The results showed that the position of the total root mean square errors were reduced by approximately 85% of the standard EKF solution. Thus, the mapping of small diameter pipelines became possible after that.

### 1.4.4 Costs’ Review

There is a cost comparing of some laterture solutions in Table 1.1. This comparing is according to the five main limitations and characteristics in any literature solution, which consist: activity of the solution in the PIG robot’s applications, sensing mode, method of positioning detection, efficiency of the method, autonomy of the system.

**Table 1.1:** Comparison of some pipeline monitoring systems' costs.

| Project | Active | Sensing | Method | Efficiency | Auto-nomy | Cost |
|---|---|---|---|---|---|---|
| **Hagen Schempf, (2005), [53].** A network system. | Passive | Static | Position sensors | Fair<br>(no free error) | No | High |
| **A.A.F. Nassiraei, *et al.* (2007) [54].** A fully autonomous robot. | Active | Mobile | Robot wheel rotations | Low<br>(many slip errors) | Yes | Not<br>avail-able |
| **Jong-Hoon Kim**, **(2011) [6].** An autonomous monitoring robotic system. | Active | Mobile | Radio-frequency identification systems | Fair<br>(no free-error) | Yes | Low |
| **Hagen Schempf, *et al.* (2003) [55].** Sub autonomous assessment robotic system. | Active | Mobile | Counting locations of pipe joint | Fair<br>(depended on detection) | Yes | Low |
| **Hussein Sahli, (2016), [52].** Sub autonomous robotic system with GPS' aiding. | Active | Mobile | Counting locations of pipe joint | Fair<br>(depended on detection) | Yes | Low |

### 1.5 Objectives

This thesis aims to provide a comprehensive solving to the navigation error problem of a PIG robot, by a new robust navigation algorithm that enables a high accurate localization. In this thesis, the attention will be devoted to discussing and analyzing the UKF to reduce the rate of errors for INS-based PIG robot application in its journey. The Matlab program will be the tool of this work.

### 1.6 Organization of the thesis

The thesis is organized to be contained on six chapters and a references list.

Chapter one includes the historical review and presentation of the pipeline inspection gauge PIG technology. It includes explanation for the main parts of PIG robot with the sending and receiving procedure, it at the beginning and end of the pipeline. In addition, this chapter reviews the literature studies that include the pigging technology integrity systems with their costs and efficiency in a briefly set of examples, and includes an introduction to Kalman filters that represent the tool to create the algorithm of the PIG robot in the posterior chapters in this thesis.

Chapter two consists an inertial navigation system with essential building blocks. It introduces different reference frames that are used in navigation field, and attitude representations that are used in an inertial navigation system. Finally, the INS fundamentals, inertial sensor calibration and compensation of the measurement error, mechanization of INS, mathematical models for aiding sensors and the error of INS model will be presented and discussed.

Chapter three reviews the KF, EKF and the unscented transformation UT with UKF and their algorithms with their mathematical models for filtering task.

Chapter four includes a briefly review for the previous chapters and shows how to combine most parts of previous chapters together in the work of this thesis to create the new PIG robot navigation algorithm. Where it consists definition for the navigation parameters of PIG robot with its trajectory constraints, a brief description for the odometer measurement and the counting system, implementation of UKF for the system of the robot navigation with the non-holonomic constraints and the error models of INS and modeling of the random process.

Chapter five includes the results of the new methodology for in-pipeline robot navigation by aiding the junctions of pipeline based on UKF algorithm that has been introduced in Chapter four. And includes comparing and discussing the navigation errors for the results depending on some assumptions.

Finally, Chapter six concludes the research results and offers the recommendations for developing the future suggested techniques.

# CHAPTER TWO

# INERTIAL NAVIGATION SYSTEMS AND AIDING TECHNIQUES

## 2.1 Overview of Aided Inertial Navigation Systems

The publication of Schuler Pendulum principle issued the theory reference for inertial navigation which was firstly applied by Germany in 1942 [13]. The upcoming Gimbaled Inertial Navigation Systems (GINSs) that were successfully designed for aircrafts and submarines, however, relied on complex, sizable, expensive but precise gimbaled platforms and gyroscopes [56]. The invention of lightweight digital computers permitted to remove the mechanical parts and triggered the appearance of Strapdown INSs (SINSs) [57]. With the development of miniaturized optical and MEMS gyroscopes, SINS gained many advantages including smaller volume, less power requirement, lower cost and faster respond. After the Global Positioning System (GPS) Selective Availability (SA) error removal and the Galileo plan agreement [20], low-cost INS/GPS integration was widely researched and applied in civilian navigation fields during the last decade.

INS is built with inertial sensors: accelerometers sensing linear accelerations and gyroscopes (gyros) sensing angular rotation rates. Orthogonally mounted inertial sensor triads on a rigid body compose the Inertial Measurement Unit IMU, the key component of a SINS. IMU computes navigation solutions by processing the inertial sensor measurements through the mechanization equations with respect to the predefined reference frame. With the IMU rigidly tied on the host, SINS is considered to be a self-contained Dead-Reckoning (DR) system as it is capable of providing the complete 3-D navigation parameters, namely positions, velocities and attitudes, without any external signal receiving or transmission [58]. Generally speaking, the IMU performance is dominated by the gyroscope accuracy [59]. According to the sensor characteristics including biases and scale factors, gyroscopes are usually classified into several categories: strategic-grade, navigation-grade, tactical-grade and customer-grade gyroscopes [25]. Another classification is based on the manufacture principles: mechanical gyros, suspended gyros, Ring Laser Gyros (RLG), Fiber Optical Gyros (FOG) and MEMS gyros. Considering the requirement of low-cost and miniaturization, SINS is based on the tactical-grade and customer-grade gyroscopes [60]. An investigation of gyroscope technology with respect to the sensor bias and scale factor is roughly described in Fig. 2.1.

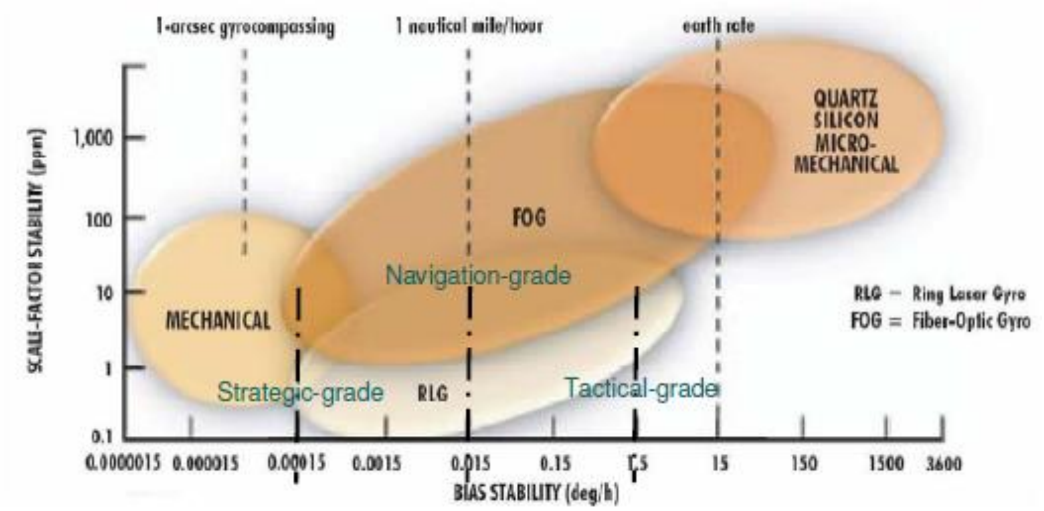


**Figure 2.1:** Investigation of Gyroscope Technology [25].

The development of gyroscopes and accelerometers led for much more successfully design and led to achieve the miniaturization and inexpensiveness simultaneously. Performance of accelerometer technology is described with respect to the sensor bias and scale factor, as shown in Fig. 2.2.

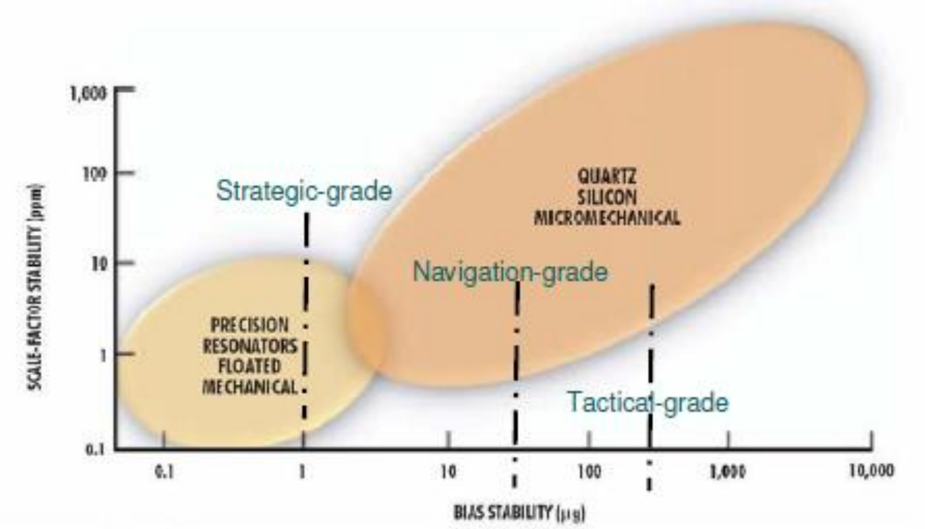


**Figure 2.2:** Investigation of Accelerometer Technology [25].

Compared to the higher-grade systems, low-cost INS confronts the problems of large and unpredictable sensor errors and noises. This inadequacy leads to the fast navigation error accumulation over short time intervals [61]. The inertial sensor calibration techniques are essential to model the determinant errors and uncertainties.

Another practical way to improve the accuracy is aiding the INS with other complementary sensors or navigation-related information [24]. In this thesis, we will use the integration of IMU/Odometer for a main navigation system and will use the pipe counting system for another external measuring system. The data of the counting system will be available when the system is between two pipes in welding points, so the updating by counting sensor have low frequency but the updating with odometer is high frequency updating.

## 2.2 Overview of Reference Frames and Attitude Parameterization

SINS algorithms require frequent transformations between different reference frames, in which the sensor measurements and navigation states are defined. On the other hand, background information about attitude representations and their conversions is a foundation for the reference frame transformations. The details related to the reference frames and attitude parameterization will be discussed in this section.

### 2.2.1 Reference frames

The reference frames and their axes are shown in the Fig. 2.3 and they are frequently used in SINS modeling as they are listed following:

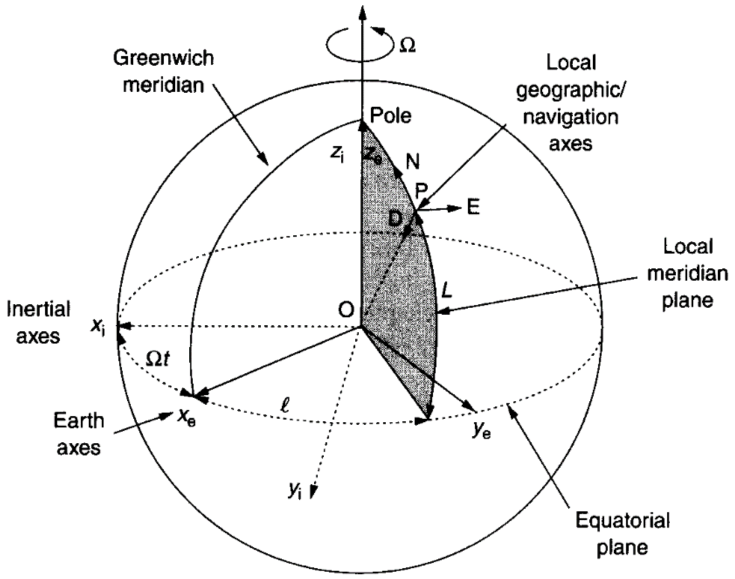


**Figure 2.3:** Reference Frames [60].

### Inertial frame (i-frame)

An inertial frame is idealized as a right-handed orthogonal, non-rotating and nonaccelerating frame with respect to fixed stars. An operational i-frame is realized by defining its origin at the Earth center, its axes are denoted by $x_i, y_i, z_i$, as shown in Fig. 2.3, with $z_i$ coincident with Earth's polar axis (which is assumed to be invariant in direction ) [60]. For its x-axis pointing towards the vernal equinox [22], and x-y directions are in the equatorial plane.

### Earth-Centered Earth-Fixed Frame (ECEF or e-frame):

ECEF is defined as a right-handed orthogonal frame which has its origin at the Earth center, which has its axes are denoted as $x_e, y_e, z_e$ as shown in Fig. 2.3, where its z-axis ($z_e$) parallel to the Earth mean spin axis, and its x-axis ($x_e$) pointing towards the mean meridian of Greenwich.

### Navigation Frame (n-frame)

The navigation frame is a local geographic frame which has its origin point at the location of the navigation system. In this thesis, it is defined as the North-East-Down NED right-handed frame as shown in Fig. 2.3. The turn rate of the navigation frame, with respect to the Earth-fixed frame ( $\omega_{en}$) is governed by the motion of the point **P** with respect to the Earth. This is often referred to as the transport rate [60].

### Body Frame (b-frame)

The body frame is illustrated in Fig. 2.4, is an orthogonal axis set which is aligned with the Roll, Pitch, Yaw axes of the vehicle in which the navigation system is installed [60]. The body frame is the same as the IMU orthogonal body axis in which the accelerations and angular rotation rates by inertial sensors are resolved [62]. In SINS, the b-frame and n-frame are assumed to be overlapped for convenience.

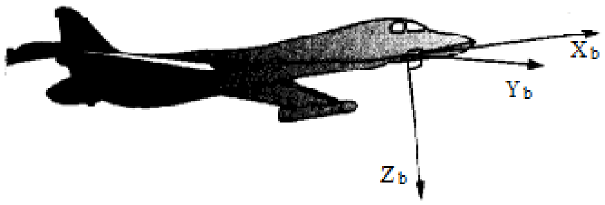


**Figure 2.4:** Body Reference Frame [36].

**Computer Frame (c-frame) and Platform Frame (p-frame)**

The computer frame is near to the navigation frame that assumed by the navigation system computer. The platform frame is the assumed inertial stabilized platform axis set in which the measurements from the hypothesized inertial sensors are resolved [62]. P-frame is actually the b-frame counterpart in gimbaled INSs.

### 2.2.2 Attitude Representations

In SINS, used attitude representation methods are often listed as following:

**Angular Rotation Vector and Angular Velocity**

The transformation of a frame from its initial orientation to its final destination, or the transformation between two different frames is preferred to be represented with a single rotation operation around its rotation axis. Angular rotation vector describes the magnitude and the direction of this rotation in a 3×1 vector $\mu = [\mu_x \quad \mu_y \quad \mu_z]^T$.

Angular velocity describes the rotation speed and its rotation around which axis direction occurs. It is usually given by a vector form from three components as shown below:

$$\omega_{nb}^{\gamma} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{2.1}$$

Where the superscript $\gamma$ refers to the name of the coordinate frame that contains the projection of the angular velocity components (it is normally set as the frame b); the (b, n) subscript means that the coordinate n-frame rotates with respect to b-frame.

The skew-symmetric matrix form is an alternative expression of angular velocity, which is as following [25]:

$$\Omega_{nb}^{b} = \left(\omega_{nb}^{b} \times\right) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{2.2}$$

The angular velocity and the angular rotation vector are given in one relationship which is as following [63]:

$$\dot{\mu} \approx d\mu/d\mathrm{t} \approx \omega_{nb}^{b} + \frac{1}{2}\mu \times \omega_{nb}^{b} + \frac{1}{12}\mu \times (\mu \times \omega_{nb}^{b}) \tag{2.3}$$

### Euler Angles and Direction Cosine Matrix (DCM)

An Euler angle is defined as a rotation angle about one coordinate frame axis. The relative orientation between two frames can be decomposed as a sequence of three rotations expressed by Euler angles. Mathematically, it can be explained as a product of three elementary rotation matrix obtained by Euler angles [57]. This product is defined as Direction Cosine Matrix (DCM) as one of the main methods for attitude parameterization.

The Euler angle elementary matrix and the corresponding DCM are formulated as,

$$C_b^n = R_x(\phi)R_{\mathrm{y}}(\theta)R_{\mathrm{z}}(\psi)$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

Where:

$\phi, \theta$ and $\psi$: are Roll, Pitch, and Heading (Yaw) angles, respectively which denote the three components of Euler angles;

$C_b^n$: denotes the DCM from b-frame to n-frame.

$R$: denotes an elementary rotation matrix, and its subscript denotes the instantaneous axis about which the Euler angle is rotated.

### Attitude Quaternion

Quaternion implementation is preferred in updating the attitude in INS as the linearity of quaternion differential equations, the lack of trigonometric functions, allow efficient algorithm [64]. Similar to the angular rotation vector, quaternion defines the frame transformation using a single rotation about its direction axis. It is represented in a $4 \times 1$ parameter vector by the rotation vector as [57]:

$$q_n^b = \begin{bmatrix} \frac{\sin\|0.5\mu\|}{\|\mu\|}\mu \\ \cos\|0.5\mu\| \end{bmatrix} = \begin{bmatrix} \frac{sin\|0.5\mu\|}{\|\mu\|}\mu_{\mathrm{x}} \\ \frac{sin\|0.5\mu\|}{\|\mu\|}\mu_y \\ \frac{sin\|0.5\mu\|}{\|\mu\|}\mu_{\mathrm{z}} \\ cos\|0.5\mu\| \end{bmatrix} \tag{2.5}$$

Where:

$q_n^b$: denotes the quaternion, which signifies the rotation from n-frame b-frame.

$||\mu||$: denotes the Euclidean norm of the rotation vector, which is the rotation magnitude as,

$$||\mu|| = \sqrt{\mu_x^2 + \mu_y^2 + \mu_z^2} \tag{2.6}$$

The product of quaternion vectors represents a series of continuous rotations as:

$$q_n^b = q_\gamma^b \bullet q_n^\gamma = \begin{bmatrix} V_1 \\ S_1 \end{bmatrix} \bullet \begin{bmatrix} V_2 \\ S_2 \end{bmatrix} = \begin{bmatrix} S_1 V_2 + S_2 V_1 + V_1 \times V_2 \\ S_1 S_2 - V_1^T V_2 \end{bmatrix} \tag{2.7}$$

Where:

● denotes the quaternion product; × denotes the vector cross product;

$V$ denotes the vector part of a quaternion, which is composed of first three components;

$S$ denotes the scalar part of a quaternion, which is the last component.

The conjugate quaternion is described as,

$$(q_n^b)^{-1} = q_n^b = \begin{bmatrix} -\frac{\sin||0.5\mu||}{||\mu||}\mu \\ \cos||0.5\mu|| \end{bmatrix} \tag{2.8}$$

Where $(q_n^b)^{-1}$denotes the conjugate quaternion of $q_n^b$.

### 2.2.3 Reference Frame Transformations

Frequently used reference frame transformations are discussed below.

**Transformations between i-frame, e-frame and n-frame**

The relationship between i-frame, e-frame and n-frame are depicted in Fig. 2.5. The DCM from e-frame to n-frame is expressed in terms of the geodetic latitude $\varphi$ and longitude $\lambda$ as:

$$C_e^n = R_y\left(-\varphi - {}^{\pi}/_{2}\right) R_z(\lambda) = \begin{bmatrix} -\sin\varphi\cos\lambda & -\sin\varphi\sin\lambda & \cos\varphi \\ -\sin\lambda & \cos\lambda & 0 \\ -\cos\varphi\cos\lambda & -\cos\varphi\sin\lambda & -\sin\varphi \end{bmatrix} \tag{2.9}$$

The corresponding quaternion is,

$$q_n^e = \begin{bmatrix} -\sin(-\pi/4 - \varphi/2)\sin(\lambda/2) \\ \sin(-\pi/4 - \varphi/2)\cos(\lambda/2) \\ \cos(-\pi/4 - \varphi/2)\sin(\lambda/2) \\ \cos(-\pi/4 - \varphi/2)\cos(\lambda/2) \end{bmatrix} \tag{2.10}$$

The angular velocities frequently used are listed as [25, 60]:

$$\omega_{ie}^{e} = [0 \quad 0 \quad \omega_e]^T; \omega_e = 7.2921158 \times 10^{-5} rad/s \tag{2.11}$$

$$\omega_{ie}^{n} = C_e^n \omega_{ie}^{e} = [\omega_e \cos\varphi \quad 0 \quad -\omega_e \sin\varphi]^T \tag{2.12}$$

$$\begin{aligned}\omega_{en}^{n} &= [\dot{\lambda}\cos\varphi \quad -\dot{\varphi} \quad -\dot{\lambda}\sin\varphi]^T \\ &= [v_E/(N+h) \quad -v_N/(M+h) \quad -v_E \tan\varphi/(N+h)]^T\end{aligned} \tag{2.13}$$

Where:

$v_E$ , $v_N$: are the east and north velocities;

$N$, $M$: are the meridian and prime vertical radii of curvature.

$h$: is the ellipsoidal height;

$$\dot{\lambda} = v_E/(N+h)/\cos\varphi\,; \dot{\varphi} = v_N/(M+h) \tag{2.14}$$

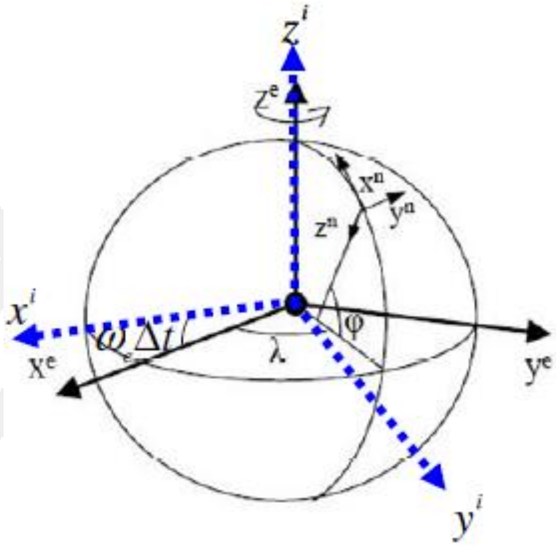


**Figure 2.5:** i-frame, e-frame and n-frame [24].

**Transformations between b-frame and n-frame**

The DCM from b-frame to n-frame is given as [23]:

$$C_b^n = R_z(-\psi)R_y(-\theta)R_x(-\phi)$$

$$= \begin{bmatrix}\cos\psi & -\sin\psi & 0\\ \sin\psi & \cos\psi & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\cos\theta & 0 & \sin\theta\\ 0 & 1 & 0\\ -\sin\theta & 0 & \cos\theta\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\ 0 & cos\,\phi & -sin\,\phi\\ 0 & \sin\phi & cos\,\phi\end{bmatrix} = \tag{2.15}$$

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

Where $\phi, \theta, \psi$, are Euler angles; the Roll, Pitch, and Yaw (Heading) angles.

The conversion from DCM to the Euler angles is shown as [64]:

$$\theta = \tan^{-1}(\frac{C_{31}}{\sqrt{1 + {C_{31}}^2}})$$

$$\phi = \tan^{-1}(C_{32}/C_{33})$$

$$\psi = \tan^{-1}(C_{21}/C_{11}) \tag{2.16}$$

Where $C_{ij}$ is the $(i, j)$ element in DCM $C_b^n$.

The conversions between quaternion and DCM are shown as [23]:

$$C_b^n = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_{2-}q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 + q_3q_4) & q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 + q_1q_4) \\ 2(q_1q_3 - q_2q_4) & 2(q_2q_3 + q_1q_4) & q_1^2 - q_2^2 - q_3^2 + q_4^2 \end{bmatrix} \tag{2.17}$$

and,

$$q_n^b = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} 0.25(C_{32} - C_{23})/0.5/\sqrt{1 + C_{11} + C_{22} + C_{33}} \\ 0.25(C_{13} - C_{31})/0.5/\sqrt{1 + C_{11} + C_{22} + C_{33}} \\ 0.25(C_{21} - C_{12})/0.5/\sqrt{1 + C_{11} + C_{22} + C_{33}} \\ 0.5\sqrt{1 + C_{11} + C_{22} + C_{33}} \end{bmatrix} \tag{2.18}$$

**Transformations between n-frame, c-frame and p-frame**

The relationship between n-frame, c-frame and p-frame is illustrated in Fig. 2.6, where the perturbation angle from n-frame to c-frame is defined as $\delta\theta$, the perturbation angle from n-frame to p-frame is defined as $\phi$, and the perturbation angle from c-frame to p-frame is defined as $\psi$ . Since all these misalignments are small angles, the following equations are yielded as [24, 62],

$$\delta\theta = [\delta\lambda cos\varphi \quad -\delta\varphi \quad -\delta\lambda \sin\varphi]^T = [\frac{\delta v_E}{N+h} - \frac{\delta v_N}{M+h} - \delta v_E \tan\frac{\varphi}{N+h}]^T \tag{2.19}$$

$$C_n^c = I - (\delta\theta \times); \quad C_n^p = I - (\psi \times);\ C_n^p = I - (\phi \times) \tag{2.20}$$

$$\phi = \psi + \delta\theta \tag{2.21}$$

Where,

$\delta\lambda,\ \delta\varphi$: denote the latitude and longitude errors;

$\delta v_N, \delta v_E$: denote the north and east velocity errors;

$(\times)$ denotes the skew symmetric matrix of a three-element vector.

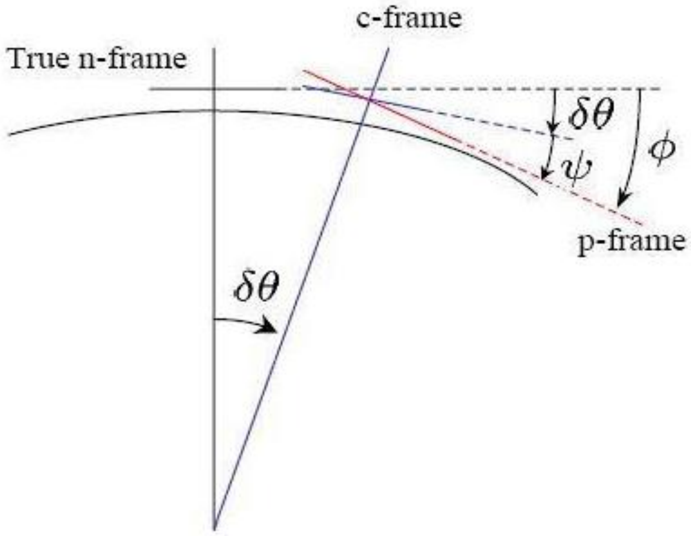


**Figure 2.6:** n-frame, c-frame and p-frame [24].

## 2.3 Inertial Navigation System (INS) Fundamentals

### 2.3.1 INS navigation equations

Without a detailed derivation, the INS navigation equations in the n-frame which define the dynamics model of the navigation states in continuous-time domain can be described as [57, 65]:

$$\begin{bmatrix} \dot{r}^n \\ \dot{v}^n \\ \dot{C}_b^n \end{bmatrix} = \begin{bmatrix} D^{-1}v^n \\ C_b^n f^b - (2\omega_{ie}^n + \omega_{en}^n) \times v^n + g^n \\ C_b^n(\omega_{ib}^b \times) - (\omega_{in}^n \times) C_b^n \end{bmatrix} \tag{2.22}$$

Where,

$r^n = [\varphi \quad \lambda \quad h]^T$: is defined as the position vector, which is essentially the polar coordinate expression in e-frame; its Cartesian coordinate counterpart is:

$$r^e = [r_x \quad r_y \quad r_z]^T =$$
$$[(N+h)\cos\varphi\cos\lambda \quad (N + h\cos\varphi\sin\lambda) \quad (N(1-e^2)+h)\sin\varphi]^T \tag{2.23}$$

Where $e$ is the first eccentricity of reference ellipsoid.

$f^b$ , $\omega_{ib}^b$: are the specific force and angular rate measurements from inertial sensors projected in b-frame, which are the time-varying parameters in navigation equations;

$g^n$: denotes the gravity vector in n-frame,

$$D^{-1} = \begin{bmatrix} 1/(M+h) & 0 & 0 \\ 0 & 1/(M+h)/\cos\varphi & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{2.24}$$

### 2.3.2 Inertial sensor calibration and measurement error compensation

Generally, the raw outputs of inertial sensors are corrupted by biases, scale factors, non-orthogonalities and noises, shown as in Eq. (2.25) and (2.26).

$$\tilde{\omega}_k = \begin{bmatrix} \tilde{\omega}_k^x \\ \tilde{\omega}_k^y \\ \tilde{\omega}_k^z \end{bmatrix} = b_{g0} + b_{gk} + \left(I + L_{g0} + L_{gk}\right)\omega_k + w_{gk} \tag{2.25}$$

$$\tilde{f}_k = \begin{bmatrix} \tilde{f}_k^x \\ \tilde{f}_k^y \\ \tilde{f}_k^z \end{bmatrix} = b_{acc0} + b_{acck} + (I + L_{acc0} + L_{acck})f_k + w_{acck} \tag{2.26}$$

Where,

The superscripts $x$, $y$, $z$ denote the sensor triad axes,

The subscripts 0 denote the determinant sensor error; the subscripts $k$ denote the random sensor error at time epoch $\mathrm{t}_k$,

$g$: denotes to the gyroscope.

$acc$: denotes to the accelerometer.

$\tilde{f}, f$: denote the vectors of the raw accelerometer outputs and the true specific force.

$\tilde{\omega}, \omega$: denote the vectors of the raw gyro outputs and the true angular velocity.

$b$: denotes the bias vector.

$w$: denotes the random noise.

$L$: denotes the linear sensor error matrix with scale factor $SF$ and non-orthogonality $r$ as:

$$L_{g0} = \begin{bmatrix} SF_{g0}^x & r_{g0}^{xy} & r_{g0}^{xz} \\ r_{g0}^{yx} & SF_{g0}^y & r_{g0}^{yz} \\ r_{g0}^{zx} & r_{g0}^{zy} & SF_{g0}^z \end{bmatrix}; L_{g,k} = \begin{bmatrix} SF_{gk}^x & r_{gk}^{xy} & r_{gk}^{xz} \\ r_{gk}^{yx} & SF_{gk}^y & r_{gk}^{yz} \\ r_{gk}^{zx} & r_{gk}^{zy} & SF_{gk}^z \end{bmatrix}$$

$$L_{acc0} = \begin{bmatrix} SF_{acc0}^x & r_{acc0}^{xy} & r_{acc0}^{xz} \\ r_{acc0}^{yx} & SF_{acc0}^y & r_{acc0}^{yz} \\ r_{acc0}^{zx} & r_{acc0}^{zy} & SF_{acc0}^z \end{bmatrix}; L_{g,k} = \begin{bmatrix} SF_{acc0}^x & r_{acc0}^{xy} & r_{acc0}^{xz} \\ r_{acc0}^{yx} & SF_{acc0}^y & r_{acc0}^{yz} \\ r_{acc0}^{zx} & r_{acc0}^{zy} & SF_{acc0}^z \end{bmatrix} \tag{2.27}$$

The process of calculation of sensor error parameters, i.e. biases, scale factors, and non-orthogonalities, is known as sensor calibration.

Determination of sensor errors are preferred to be calibrated beforehand in laboratory. Normal SINS laboratory calibration technologies are following the idea to compare the IMU outputs with the reference information including gravity and earth rotation rate [66]. The random errors are always mathematically modeled as stochastic processes [61]. Allan Variance method is utilized as part of the laboratory work to determine the model types and estimate the model parameters for the random noise [67]. Random sensor error parameters can be calibrated in field tests or be estimated on-line in the integrated navigation systems. The effect of random errors will be suppressed using optimal estimation methods with aiding sources, as discussed in the succeeding chapters.

With calibrated parameters, sensor errors can be compensated from raw outputs as:

$$\omega_k = (I + L_{g0} + L_{gk})^{-1}(\tilde{\omega}_k - b_{g0} - b_{gk}) \tag{2.28}$$

$$f_k = (I + L_{acc0} + L_{acck})^{-1}(\tilde{f}_k - b_{acc0} - b_{acck}) \tag{2.29}$$

Instead of specific forces and angular rates, incremental velocities and angles are the outputs in most of the high-grade IMUs. Integration procedures to relate the two types of IMU outputs are introduced as follows [57]:

$$\Delta\Theta_k = \int_{t_{k-1}}^{t_k} \omega dt \approx \omega_k \Delta t_k \tag{2.30}$$

$$\Delta v_k = \int_{t_{k-1}}^{t_k} f dt \approx f_k \Delta t_k \tag{2.31}$$

Where,

$\Delta\Theta\,, \Delta v$ denote the incremental angles and velocities;

$\Delta t = t_k - t_{k-1}$ is the time increment.

### 2.3.3 INS mechanization

SINS mechanization is defined as the integration process to calculate the navigation states, i.e. positions, velocities and attitudes, with raw inertial sensor measurements. Therefore, the mechanization algorithm can be regarded as the discrete-time form of the INS navigation equations. Several approximation methods were applied to solve the quaternion differential equations in attitude integration. Further, a single-speed mechanization algorithm considering midway navigation states and applying quaternion algebras was developed [57]. SINS mechanization is the integration process to determine the navigation states from the previous time epoch $t_{k-1}$ to the current time epoch $t_k$ using compensated IMU outputs. Its simplification in [24], which will be summarized in this thesis.

**Velocity Integration**

The discrete-time form of the second component in Eq. (2.22) can be written as:

$$v_k^n = v_{k-1}^n + \Delta v_{fk}^n + \Delta v_{ek}^n \tag{2.32}$$

$$\Delta v_{f_k}^n = [I - (0.5\varsigma_k \times)]C_{b_{k-1}}^{n_{k-1}} \Delta v_{f_k}^{b_{k-1}} \tag{2.33}$$

$$\varsigma_k = [\omega_{ie}^n \omega_{en}^n]_{k-1/2} \Delta \mathrm{t}_k \tag{2.34}$$

$$\Delta v_{f_k}^{b_{k-1}} = \Delta v_{f_k}^b + \frac{1}{2}\Delta\theta_k \times \Delta v_{f_k}^b + \frac{1}{12}(\Delta\theta_{k-1} \times \Delta v_{f_k}^b \times \Delta v_{f_{k-1}}^b \times \Delta\theta_k) \tag{2.35}$$

$$\Delta v_{e_k}^n = [g^n - (2\omega_{ie}^n + \omega_{en}^n) \times v^n]_{k-1/2} \Delta \mathrm{t}_k \tag{2.36}$$

Where:

The subscripts *k*-1, *k*-1/2 and *k* denote the previous, midway and current time epochs $\mathrm{t}_{k-1}, \mathrm{t}_{k-1/2}, \mathrm{t}_k$ respectively.

$b_k$: denote the corresponding variable is projected to the b-frame at $\mathrm{t}_k$.

$\Delta v_{e_k}^n$: is the increment induced by gravity and Coriolis force.

$\Delta v_{f_k}^n$: is the increment induced by specific force.

$\varsigma_k$ is the n-frame rotation vector from ($n_{k-1}$-frame) to ($n_k$-frame).

The second and third terms at the right of Eq. (2.35) are the rotational and sculling motion.

Positions at midway are required to be extrapolated from the previous time navigation states as:

$$h_{k-1/2} = h_{k-1} - \frac{v_{Dk-1}\,\Delta t_k}{2} \tag{2.37}$$

$$\delta\varphi_{k-1/2} = v_{Nk-1}/(M + h_{k-1/2})/2 \tag{2.38}$$

$$\delta\lambda_{k-1/2} = v_{Ek-1}/(N + h_{k-1/2})/\cos(\varphi_{k-1})\,/2 \tag{2.39}$$

$$\delta\theta_{k-1/2} = [\delta\lambda_{k-1/2}\cos\varphi_{k-1} \; -\delta\varphi_{k-1/2} \; -\delta\lambda_{k-1/2}\sin\varphi_{k-1}]^T \tag{2.40}$$

$$q_{\delta\theta_{k-1/2}} = \begin{bmatrix} \frac{\sin\|0.5\delta\theta_{k-1/2}\|}{\|\delta\theta_{k-1/2}\|}\delta\theta_{k-1/2} \\ \cos\|0.5\delta\theta_{k-1/2}\| \end{bmatrix} \tag{2.41}$$

$$q_{n_{k-1/2}}^{e_{k-1/2}} = q_{n_{k-1}}^{e_{k-1}} \bullet \; q_{\delta\theta_{k-1/2}} \tag{2.42}$$

Where the midway latitude and longitude can be extracted from quaternion $q_{n_{k-1/2}}^{e_{k-1/2}}$.

Velocity at midway are extrapolated as:

$$v_{k-1/2}^{n} = v_{k-1}^{n} + \frac{\Delta v_{k-1}^{n}}{2} = v_{k-1}^{n} + \frac{\Delta v_{f_{k-1}}^{n} + \Delta v_{e_{k-1}}^{n}}{2} \tag{2.43}$$

Where $\Delta v_{k-1}^{n}$ is the second and third velocity increments at the right of Eq. (2.32) stored in the previous epoch.

**Position Integration**

The midway velocity can be updated by interpolation as:

$$v_{k-1/2}^{n} = \frac{v_{k-1}^{n} + v_{k}^{n}}{2} \tag{2.44}$$

The height can be updated by the midway downside velocity as:

$$h_k = h_{k-1} - v_{Dk-1/2} \Delta t_k \tag{2.45}$$

The current time quaternion $q_{n_k}^{e_k}$ containing position information can be updated by the products of e-frame and n-frame rotations as:

$$q_{n_k}^{e_{k-1}} = q_{n_{k-1}}^{e_{k-1}} \bullet q_{n_k}^{n_{k-1}} \tag{2.46}$$

$$q_{n_k}^{e_k} = q_{e_{k-1}}^{e_k} \bullet q_{n_k}^{e_{k-1}} \tag{2.47}$$

$$q_{n_k}^{n_{k-1}} = \begin{bmatrix} \frac{\sin\|0.5\varsigma_k\|}{\|\varsigma_k\|} \varsigma_k \\ \cos\|0.5\varsigma_k\| \end{bmatrix} \tag{2.48}$$

$$q_{e_{k-1}}^{e_k} = \begin{bmatrix} \frac{\sin\|0.5\xi_k\|}{\|\xi_k\|} \xi_k \\ \cos\|0.5\xi_k\| \end{bmatrix} \tag{2.49}$$

$$\xi_k = \omega_{ie}^{e} \Delta t_k \tag{2.50}$$

Where:

$\varsigma_k$: is recalculated with the renewed midway velocity using Eq. (2.34).

$\xi_k$: denotes the e-frame rotation vector from ($e_{k-1}$-frame) to ($e_k$-frame).

$q_{n_k}^{n_{k-1}}, q_{e_{k-1}}^{e_k}$: denote the quaternion vectors corresponding to the above rotation vectors.

**Attitude Integration**

The midway positions can be renewed by interpolation as:

$$\varphi_{k-1/2} = \frac{\varphi_{k-1}+\varphi_k}{2} \tag{2.51}$$

$$\lambda_{k-1/2} = \frac{\lambda_{k-1}+\lambda_k}{2} \tag{2.52}$$

$$h_{k-1/2} = \frac{h_{k-1}+h_k}{2} \tag{2.53}$$

The quaternion $q_b^n$ containing attitude information can be updated by the products of n-frame and e-frame rotations as:

$$q_{b_k}^{n_{k-1}} = q_{b_{k-1}}^{n_{k-1}} \bullet q_{b_k}^{b_{k-1}} \tag{2.54}$$

$$q_{b_k}^{n_k} = q_{n_{k-1}}^{n_k} \bullet q_{b_k}^{n_{k-1}} \tag{2.55}$$

$$q_{b_k}^{b_{k-1}} = \begin{bmatrix} \frac{\sin\|0.5\phi_k\|}{\|\phi_k\|}\phi_k \\ \cos\|0.5\phi_k\| \end{bmatrix} \tag{2.56}$$

$$q_{n_{k-1}}^{n} = \begin{bmatrix} -\frac{\sin\|0.5\varsigma_k\|}{\|\varsigma_k\|}\varsigma_k \\ \cos\|0.5\varsigma_k\| \end{bmatrix} \tag{2.57}$$

$$\phi_k \approx \Delta\theta_k + \frac{1}{12}\Delta\theta_{k-1} \times \Delta\theta_k \tag{2.58}$$

Where:

$\phi_k$: is the b-frame rotation vector from ($b_{k-1}$-frame) to ($b_k$-frame).

The second term at the right of Eq. (2.58) denotes the second-order coning correction.

### 2.3.4 INS error model

SINS navigation equations are the non-linear models to describe the dynamics of the navigation states. The linearized equations can be derived by perturbation analysis [68], which are transferred as the models of the navigation error states, i.e. position errors, velocity errors, and attitude angle errors. For convenience, the $\psi$-angle error model, which indicates the perturbation is conducted with respect to the computer frame, will be utilized in this thesis. In addition, the random sensor error parameters including residual bias and measurements noise are modeled as constant and white noise. Materials [24, 61, 62 and 69] for detailed deductions of the above algorithm models are recommended to readers with interest.

The continuous-time $\psi$-angle error model for navigation states and sensor error states are shown as:

$$\begin{aligned} &\delta \dot{r}^c = -\omega_{ec}^c \times \delta r^c + \delta v^c \\ &\delta \dot{v}^c = f^c \times \psi - (2\omega_{ie}^c + \omega_{ec}^c) \times \ \delta v^c + \delta g^c + C_b^n \delta f^b \\ &\dot{\psi} = -(\omega_{ie}^c + \omega_{ec}^c) \times \psi - C_b^n \delta \omega_{ib}^b \end{aligned} \tag{2.59}$$

Where,

The position error vector is $\delta r^c = [\delta r_N \quad \delta r_E \quad \delta r_D]^T$.

The superscript *c* denotes the computer frame.

The sensor measurement errors are written as:

$$\delta f^b = b_{acc} + f^b + w_{acc} \tag{2.60}$$

$$\delta \omega_{ib}^b = b_g + \omega_{ib}^b + w_g \tag{2.61}$$

The gravity perturbation is:

$$\delta g^c = D_g \delta r^c = diag([\frac{-g}{M+h} \quad \frac{-g}{N+h} \quad \frac{-g}{\sqrt{MN}+h}])\delta r^c \tag{2.62}$$

Where the *diag*(●) denote the diagonal matrix form of a vector.

The stochastic models for the sensor random biases are given as:

$$\dot{b}_{acc} = 0 \tag{2.63}$$

$$\dot{b}_g = 0 \tag{2.64}$$

Where,

$b_{acc}$ and $b_g$ are biases of accelerometers and gyroscopes, respectively.

The combination of Eq. (2.59) and Eq. (2.63)-(2.64) yields the forward linear dynamics process model with both the navigation error parameters and the sensor error parameters defined as the system states. The discrete-time form of this process model can be given by [70, 71]:

$$\begin{aligned} Sx_k = F_{k-1}\delta x_{k-1} + w_{k-1} &= \left( I + \begin{bmatrix} (F_1)_{9\times9} & (F_2)_{9\times6} \\ 0_{6\times9} & (F_3)_{6\times6} \end{bmatrix}_{k-1} \Delta t_{k-1} \right) \delta x_{k-1} + w_{k-1} \\ &= \begin{bmatrix} (I + F_1 \Delta t_{k-1})_{9\times9} & (F_2 \Delta t_{k-1})_{9\times6} \\ 0_{6\times9} & (I + F_3 \Delta t_{k-1})_{6\times6} \end{bmatrix}_{k-1} \delta x_{k-1} + w_{k-1} \\ &= \begin{bmatrix} (F_1{}')_{9\times9} & (F_2{}')_{9\times6} \\ 0_{6\times9} & (F_3{}')_{6\times6} \end{bmatrix}_{k-1} \delta x_{k-1} + w_{k-1} \end{aligned} \tag{2.65}$$

Where:

The 15 system state vector is,

$$Sx_k = \left[(\delta_{r^c})^T \quad (\delta_{v^c})^T \quad (\psi)^T \quad \left(\delta b_g\right)^T \quad (\delta b_{acc})^T\right]_k^T \tag{2.66}$$

$F_{k-1}$ is the state transition matrix, obtained from the numerical approximation of the continuous-time dynamics matrix $F(t)$ , which is composed of the following matrix,

$$F_1 = \begin{bmatrix} -(\omega_{ec}^c \times)_{3\times3} & I_{3\times3} & 0_{3\times3} \\ (D_g)_{3\times3} & [-(2\omega_{ie}^c + \omega_{ec}^c)\times]_{3\times3} & (f^c \times)_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & [-(\omega_{ie}^c + \omega_{ec}^c)\times]_{3\times3} \end{bmatrix} \tag{2.67}$$

$$F_2 = \begin{bmatrix} 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & C_b^n 0_{3\times3} \\ -C_b^n & 0_{3\times3} \end{bmatrix} \tag{2.68}$$

The covariance matrix for the driving noise *w* in discrete-time domain is:

$$Q = diag\begin{bmatrix} Q_{acc} & 0 \\ 0 & Q_{acc} \end{bmatrix} \tag{2.69}$$

Where:

$Q_{acc}$ and $Q_{acc}$: are covariance of accelerometers and gyroscopes, respectively.

### 2.3.5 Initial alignment

INS initial alignment is defined as the process to determine the initial values of the navigation parameters. Dependable position and velocity information can be provided by high-accuracy GPS solutions. Since the accuracy of the initial attitudes predominantly governs the navigation error accumulation, initial alignment is narrowly considered as the procedure to initialize the attitude information, contained in the DCM, $C_b^n$ [68]. For IMUs whose gyro bias and noise levels are smaller than the values of the Earth rotation rate, such as navigation-grade or high-end tactical-grade IMUs, a coarse alignment followed by a fine alignment can be applied to estimate the initial attitude parameters. The coarse alignment is an analytic method providing the averaged solutions. It can be decomposed as the levelling step, determining the initial roll and pitch, and the Gyro-compassing step, determining the heading angle [60]. In many cases with the established DCM from the b-frame to the n-frame, the fine alignment is an optimal estimation method by an INS-only KF using horizontal specific force and east-channel gyro error measurements. Both of the two alignment methods are processed in stationary mode and implemented on the basis of the reference information including gravity and Earth rotation rate [64].

For low-cost IMUs, the poor gyroscope characteristics result in the failing of initial heading alignment [72]. On the other hand, stationary alignment cannot meet the real-time consideration in civilian and commercial applications, such as vehicle navigation [23]. Aiding sources including magnetometers, GPS multi-antenna systems and/or GPS-derived velocity information are indispensable to the in-motion alignment techniques. Besides, kinematic alignment is researched from the system observability point of view in aircraft applications [73].

### 2.4 Measurement Models for Aiding Sources

Since the EKF is commonly applied to resolve the non-linearity in system model, it is the measurement disclosure, or the difference between the INS mechanization outputs and the observations from aiding sources that is concerned in the INS-based integration systems,

$$\delta\tilde{z}_k = \tilde{z}_{INS} - \tilde{z}_{aiding} \qquad (2.70)$$

Where, $\tilde{z}_{INS}$ is the INS mechanization solution; $\tilde{z}_{aiding}$ is the aiding sensor observation.

#### 2.4.1 Global Positioning System (GPS)

The GPS is a Global Navigation Satellite System (GNSS) developed by the United States Department of Defence (DOD), which provides absolute positioning information and long term accuracy under all weather conditions [20]. Due to its dependency on radio signal transmission and line-of-sight (LOS) measurements, GPS suffers from various error sources and poor satellites geometry. To eliminate or mitigate the common errors between receivers, epochs, satellites or stations, the Differential GPS (DGPS) technique is implemented to improve the positioning accuracy to centimeter level. Several strategies were performed to integrate the GPS and INS data to overcome their individual disadvantages and reach superior performance. In this thesis, loosely-coupled integration is introduced which utilizes position or position/velocity measurements from GPS-only filter to aid INS solutions. The measurement model using GPS position solutions considering the lever arm effect can be written as [24]:

$$\delta z_k = \delta r_k^c + \left(C_b^n l_{GPS}^b \times\right)\psi_k + \nu_k \qquad (2.71)$$

Where,

$l^b_{GPS}$: denotes the lever-arm effect between the GPS antenna and IMU mass center projected in the b-frame;

$\nu_k$: is the GPS position measurement noise, with the spectral density matrix obtained from statistic and/or kinematic GPS data processing as,

$$R_k = diag\left(\left[\sigma_\varphi^2 \quad \sigma_\lambda^2 \quad \sigma_h^2\right]\right) \tag{2.72}$$

The measurement vector is:

$$\delta\tilde{\mathcal{Z}}_k = D(\tilde{r}^n_{INS_k} - \tilde{r}^n_{GPS_k}) + \tilde{C}^n_b l^b_{GPS} \tag{2.73}$$

$$D = \begin{bmatrix} (M+h) & 0 & 0 \\ 0 & (M+h)cos\varphi & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{2.74}$$

Where, $(\tilde{r}^n_{INS_k}, \tilde{r}^n_{GPS_k})$: denote the position vectors achieved by INS and GPS.

The measurement model using GPS velocity solutions considering the lever arm effect can be written as:

$$\delta_{\mathcal{Z}_k} = \delta v_k^c - (\omega^n_{in} \times)C^n_b\left(l^b_{GPS} \times\right)\psi_k - C^n_b\left(l^b_{GPS} \times\right)\delta\omega^b_{ib} + \nu_k \tag{2.75}$$

Where:

$\nu_k$: is the GPS velocity measurement noise, with the covariance matrix obtained from statistic and/or kinematic GPS data processing as:

$$R_k = diag(\left[\sigma^2_{v_N} \quad \sigma^2_{v_E} \quad \sigma^2_{v_D}\right]) \tag{2.76}$$

The measurement vector is:

$$\delta_{\tilde{\mathcal{Z}}_k} = \tilde{v}^n_{INS_k} - (\tilde{\omega}^n_{in} \times)\tilde{C}^n_b l^b_{GPS} - \tilde{C}^n_b\left(l^b_{GPS} \times\right)\tilde{\omega}^b_{ib} - \tilde{v}^n_{GPS_k} \tag{2.77}$$

Where, $(\tilde{v}^n_{INS_k}, \tilde{v}^n_{GPS_k})$ denote the velocity vectors of INS and GPS.

### 2.4.2 Odometer and Non-Holonomic Constraints

Odometers, or milometers, are applied in land-vehicle navigation and pipeline surveys to provide augmented host velocity observations. The measurement model using odometer velocity measurements considering the misalignment between h-frame and b-frame can be written as [74]:

$$\delta_{z_k} = C_b^h C_n^b \delta v_k^c - C_b^h C_n^b (v_{INS}^n \times)\psi_k - C_b^h (l_{odometer}^b \times)\delta\omega_{ib}^b + \nu_k \qquad (2.78)$$

Where,

$C_b^h$: denotes the DCM from b-frame to h-frame;

$l_{odometer}^b$: denotes the lever arm effect between the odometer and IMU mass center projected in b-frame.

$v_{INS}^n$: is the velocity vector of INS mechanization.

$\nu_k$: is the odometer measurement noise, with the covariance matrix evaluated by the priori knowledge on sensor characteristics.

The measurement vector is,

$$\begin{aligned}\delta_{\tilde{z}_k} &= \tilde{C}_b^h \tilde{C}_n^b \tilde{v}_{INS_k}^n + \tilde{C}_b^h \left(\tilde{\omega}_{nb}^b \times\right) l_{odometer}^b - v_{odometer_k}^h \\ &= \tilde{C}_b^h \tilde{C}_n^b \tilde{v}_{INS_k}^n + \tilde{C}_b^h \left(\tilde{\omega}_{nb}^b \times\right) l_{odometer}^b - \left[v_{odometer_k}^x \quad 0 \quad 0\right]^T \qquad (2.79)\end{aligned}$$

Where:

$v_{odometer}^h$: denotes the velocity vector observed by odometer projected to host-frame;

$v_{odometer}^x$: denotes the odometer observation along the forward direction in h-frame.

Non-holonomic constraints is defined as the fact that unless the vehicle jumps off the ground or slides on the ground, the velocity of the vehicle in the plane perpendicular to the forward direction is almost zero, as in Eq. (2.80) [72, 75 and 76]. This is illustrated as in Fig. 2.7.

$$\delta_{\tilde{v}_k^y} \approx 0\,,\ \delta_{\tilde{v}_k^z} \approx 0 \qquad (2.80)$$

Where, the superscripts $y$, $z$ denote the transversal and down directions in h-frame.

Simplified from Eq. (2.78), the measurement model using non-holonomic constraints can be written as:

$$\delta_{z_k} = \begin{bmatrix} \delta_{z_k^y} \\ \delta_{z_k^z} \end{bmatrix} = \left(C_b^h C_n^b\right)_{2:3,3:3} \delta v_k^c - \left[C_b^h C_n^b (v_{INS}^n \times)\right]_{2:3,3:3} \psi_k + \nu_k \qquad (2.81)$$

Where:

The subscript 2: 3, 3: 3 denote the last two rows' elements of a 3 × 3 matrix.

$v_k$: is the assumed non-holonomic constraints noise.

The measurement vector is:

$$\delta_{\tilde{z}_k} = (\tilde{C}_b^h \tilde{C}_n^b)_{2:3,3:3} \begin{bmatrix} v_{INS}^y \\ v_{INS}^z \end{bmatrix} \tag{2.82}$$

Where, $v_{INS}^y$ , $v_{INS}^z$ are the velocities achieved by INS in east and down directions.

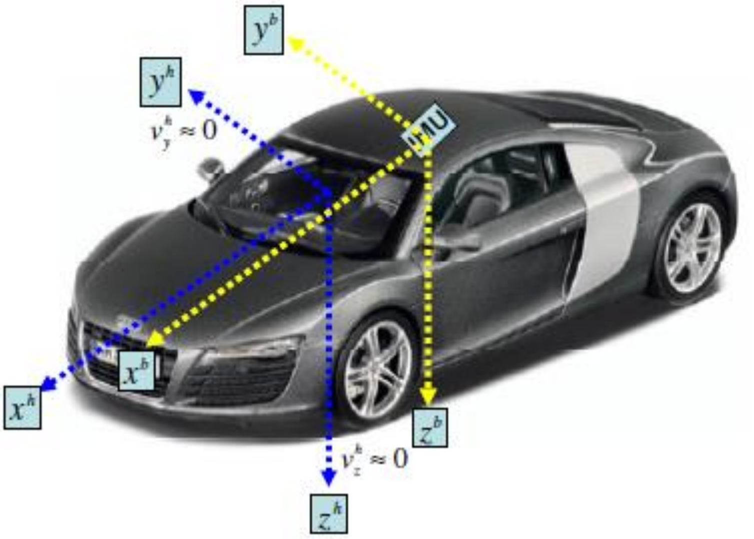


**Figure 2.7:** Non-holonomic Constraints [44].

# CHAPTER THREE

## KALMAN FILTERS

### 3.1 Kalman Filter (KF)

The Kalman filter is a mathematical power apparatus that is playing as an increasingly essential part in fields of mechanical engineering as we incorporate sensing of the actual world in our frameworks. The great news is you do not need to be a mathematical genius to comprehend and successfully utilize Kalman filter.

Kalman filter is known as the one of the best and the most significant mathematical tools, which can be often used for stochastic estimation from noisy sensor measurements, and it has named after Rudolph E. Kalman, who published his well-known paper in 1960, which is telling a recursive solution to the discrete-data linear filtering problem.

While the Kalman filter has been around for more than 45 years, it (and related ideal estimators) have as of late begun flying up in a wide assortment of computer graphics in many applications. These applications span from simulating melodic instruments in Virtual Reality, to head tracking, to removing lip movement from video arrangements of talkers, to fitting spline surfaces over accumulations of focuses [77].

The Kalman filter is basically a group of mathematical equations that applying a predictor-corrector style estimator that is ideal to minimize the estimated error covariance while some assumed conditions are met. Since the Kalman filter has been introduced, it was started to be extensively used in many research and application, especially in the autonomous field or aided navigation. This is likely due in a vast part to get advances in digital computing that made the utilization of the filter workable, in any case, but due to the relative simplicity of the filter and the filter’s sturdy nature. Seldom do the conditions, which are necessary for optimality, really exist, but the filter didn’t show any weakness yet for several applications in spite of this case [77]. Unfortunately, in most of the navigation applications the models are needed to be non-linear. Therefore, the EKF is more utilization to be employed than the KF in some parts of INS.

The Kalman filters are based on the linear dynamics systems discretized in the time domain. Employing KF to deal with any linear state-space system needs building a model to express about the linear state-space dynamically, and using the observation relationship between the system states and measurement quantities. Therefore KF can be described the linear dynamics systems as the continuous-time system Eq. (3.1) and the discrete time measurement Eq. (3.2), as;

$$x(t) = F(t)x(t) + G(t)w(t) \tag{3.1}$$

$$z_k = H_k x_k + v_k \tag{3.2}$$

Where:

$t$: denotes the continuous time.
$k$: refers to epoch of the discrete time $t_k$.
$x$: indicates the system state in vector form.
$z$: indicates the measurement in vector form.
$w$: indicates the process/ system noise in vector form, which is assumed to be a Gaussian white noise with the covariance matrix, $E[w(t)w(\tau)^T] = Q(t)\delta(t-\tau)$, where the operator $\delta(\cdot)$ refers to the Dirac delta function and $Q$ is called the spectral density matrix [29].
$v_k$: refers to the measurement/ sensors' white noise in vector form.
*F*: refers to the system dynamics prediction matrix.
*G*: refers to the system noise control/ shaping matrix;
*H*: refers to the observation design matrix for model the sensors.

For digital implementation, the Eq. (3.1) is preferred to be in the following form:

$$x_k = \Phi_{k,k-1} x_k + w_{k-1} \tag{3.3}$$

Where:

( $w_{k-1} = \int_{t_{k-1}}^{t_k} \Phi_{k,r}\, G(\tau)w(\tau)d\tau$ ): is suitable at $t_k$ which dues to the existence of the input white noise throughout the time interval $(t_{k-1}, t_k)$ [71]:

Where:

$\Phi_{k,k-1}$: refers to the system prediction/ transition matrix during the epoch from $t_{k-1}$ to $t_k$.
$k-1$: refers to the time epoch $t_{k-1}$.

In reality, for most system models, the dynamics prediction matrix $F(t)$ is considered to be time invariant during the short time interval $\Delta t = t_k - t_{k-1}$. So, the transition matrix ($\Phi_{k,k-1}$) can be achieved from the dynamics matrix by the following simple numerical approximation [78];

$$\Phi_{k,k-1} = \mathcal{L}^{-1}[(sI - F)^{-1}] = e^{F\Delta t} \approx I + F\Delta t + \frac{\Delta t^2}{2!}F^2 + \frac{\Delta t^3}{3!}F^3 + \cdots\cdots \tag{3.4}$$

For the reason that a white sequence is a zero-mean random variable sequence, which is uncorrelated time wise, the covariance matrix combines $w_k$ with $\nu_k$ is given in [71] as following:

$$E[w_k w_i^T] = \begin{cases} Q_k \ , i = k \\ \ 0 \ , i \neq k \end{cases} \tag{3.5}$$

$$E[\nu_k \nu_i^T] = \begin{cases} R_k \ , i = k \\ \ 0 \ , i \neq k \end{cases} \tag{3.6}$$

$$E[w_k \nu_i^T] = 0 \ \forall i, k \tag{3.7}$$

Where the process/ system noise matrix $Q_k$ is resulting as [44]:

$$\begin{aligned} Q_k = E[w_k w_i^T] &= E\left\{\left[\int_{t_k}^{t_{k+1}} \Phi_{k+1,\xi} G(\xi) w(\xi) d\xi\right]\left[\int_{t_k}^{t_{k+1}} \Phi_{k+1,\eta} G(\eta) w(\eta) d\eta\right]^T\right\} \\ &= \int_{t_k}^{t_{k+1}}\int_{t_k}^{t_{k+1}} \Phi_{k+1,\xi} G(\xi) E[w(\xi) w^T(\eta)] G^T(\eta) \Phi_{k+1,\eta}^T d\xi d\eta \\ &\approx \frac{1}{2}[\Phi_k G(t_k) Q(t_k) G^T(t_k) \Phi_k^T + G(t_{k+1}) Q(t_{k+1}) G^T(t_{k+1})]\Delta t_{k+1} \end{aligned} \tag{3.8}$$

The placement of the estimation of the initial state $\hat{x}_0$ and its covariance $P_0 = E[\hat{x}_0 \hat{x}_0^T]$ is the first step of the Kalman Filter algorithm. With a priori information of the noise characteristics and a priori information of the initial conditions, the Kalman Filter algorithm can be applied refining the a series of estimation with measurement update steps [29].

The Kalman Filter estimation/ prediction step of the system model, which is given by the prediction state estimate in Eq. (3.9) and the prediction covariance matrix in Eq. (3.10):

$$\hat{x}_k^- = \Phi_{k,k-1} \hat{x}_{k-1}^+ \tag{3.9}$$

$$P_k^- = \Phi_{k,k-1} P_k^+ \Phi_{k,k-1}^T + Q_{k-1} \tag{3.10}$$

Where:

$\hat{x}^-$: indicates the prediction state estimate.

$\hat{x}^+$: indicates the update state estimate.

$P^-$: indicates the prediction covariance matrix.

$P^+$: indicates the update covariance matrix.

In the measurement update step, the best state estimate and its covariance are updated with the predictions and the observations. The following set of equations treats the measurement update step:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \tag{3.11}$$

$$v_k = z_k - H_k \hat{x}_k^- \tag{3.12}$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k v_k \tag{3.13}$$

Where:

$P_k^+ = (I - K_k H_k) P_k^-$;

$v_k$: denotes the innovation sequence, which refers to the variance between the prediction and the observation.

$K_k$: denotes the gain of Kalman Filter, or the weighting matrix, which determines how much of the new information contained in the innovations that should be accepted by the system [22]. The gain matrix is derivate based on the minimum variance norm. In Fig. 3.1, the KF algorithm is outlined:

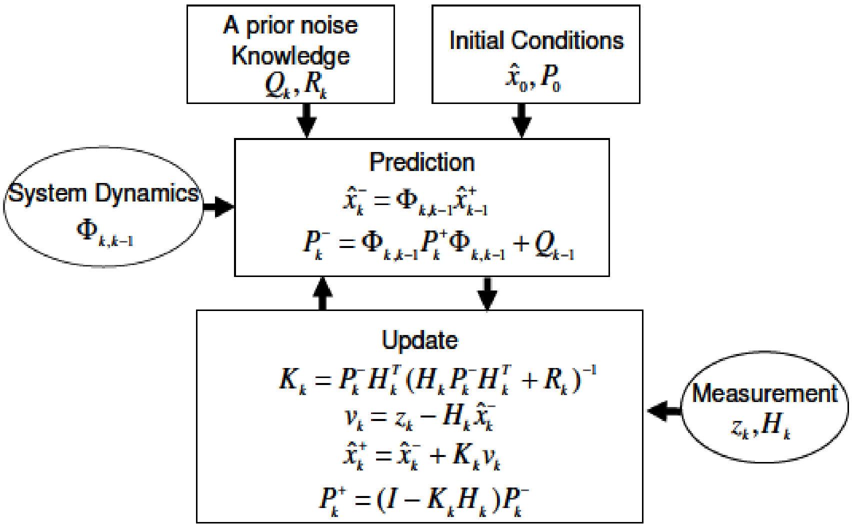

A prior noise Knowledge
$Q_k, R_k$
Initial Conditions
$\hat{x}_0, P_0$
System Dynamics
$\Phi_{k,k-1}$
Prediction
$\hat{x}_k^- = \Phi_{k,k-1}\hat{x}_{k-1}^+$
$P_k^- = \Phi_{k,k-1} P_k^+ \Phi_{k,k-1} + Q_{k-1}$
Update
$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}$
$v_k = z_k - H_k \hat{x}_k^-$
$\hat{x}_k^+ = \hat{x}_k^- + K_k v_k$
$P_k^+ = (I - K_k H_k) P_k^-$
Measurement
$z_k, H_k$

**Figure 3.1:** Kalman Filter Algorithm [44].

### 3.2 The Extended Kalman Filter (EKF)

As described above in Section 3.1, the Kalman filter has mathematical models, which are assumed to be linear. Therefore, KF governs its attempts to estimate the state $x$ of a discrete-time linear by a linear stochastic difference equation. However, for estimated a non-linear process and its measurement relationship the EKF will be more appropriate than the KF because of the EKF modifies some of the concepts that are offered in Section 3.1 to be the typical discrete-time non-linear system model and its measurement relationship as showed in the below equations [70]:

$$x_k = f(x_{k-1}, k-1) + w_{k-1} \tag{3.14}$$

$$z_k = h(x_k, k) + v_k \tag{3.15}$$

Where:

$f\,(\cdot)$: is the non-linear function, which relates the state at the previous time step $k-1$ to the state at the current time step $k$.

$h\,(\cdot)$: is the non-linear function, which relates the state $x_k$ to the measurement $z_k$.

The rule strategy to manage the non-linear estimation issues is to linearize the models around a nominal trajectory, which is predetermined or immediate. The definition of this nominal trajectory is as the trace, which tracks a vector of time-changing parameter that refers usually to the succession of the system state vectors with the predicted or evaluated values [70]. By utilizing Taylor series expansion, the linearization procedure of the non-linear system model is given as following:

$$x_k^{NOM} = f(x_{k-1}^{NOM}, k-1) \tag{3.16}$$

$$x_k = x_k^{NOM} + \delta_{x_k} \tag{3.17}$$

$$\begin{aligned} x_k &= f(x_{k-1}, k-1) + w_{k-1} \\ &\approx f(x_{k-1}^{NOM}, k-1) + \frac{\partial f(x(t), t)}{\partial x} \mid_{x=x_{k-1}^{NOM}, t=k-1} (x_{k-1} - x_{k-1}^{NOM}) \\ &\quad + w_{k-1} \end{aligned}$$

$$= x_k^{NOM} + \frac{\partial f(x(t), t)}{\partial x} \mid_{x=x_{k-1}^{NOM}, t=k-1} (x_{k-1} - x_{k-1}^{NOM}) + w_{k-1} \tag{3.18}$$

Or,

$$\begin{aligned} \delta x_k &= \frac{\partial f(x(t), t)}{\partial x} \mid_{x=x_{k-1}^{NOM}, t=k-1} \delta x_{k-1} + w_{k-1} \\ &= F_{k-1} \delta x_{k-1} + w_{k-1} \end{aligned} \tag{3.19}$$

Where:

$x^{NOM}$: symbolizes to the nominal trajectory.

$\delta x$: denotes what is termed “error state” that represents the system state disturbance from the nominal trajectory.

$f(x(t),t)$: refers to the continuous non-linear function.

$F_{k-1}$: refers to the linearized system dynamics matrix, relating to the system transition matrix $\Phi_{k,k-1}$ in the linear situation.

Similarly, the linearization procedure of the non-linear system m measurement model is given as following;

$$z_k = h(x_k,k) + v_k \approx h(x_k^{NOM},k) + \frac{\partial h(x(t),t)}{\partial x}\Big|_{x=x_k^{NOM},t=k}(x_k - x_k^{NOM}) + v_k$$

$$= z_k^{NOM} + \frac{\partial h(x(t),t)}{\partial x}\Big|_{x=x_k^{NOM},t=k}(x_k - x_k^{NOM}) + v_k \tag{3.20}$$

or,

$$\delta z_k = z_k - h(x_k^{NOM},k)$$

$$= \frac{\partial h(x(t),t)}{\partial x}\Big|_{x=x_k^{NOM},t=k}\delta x_k + v_k$$

$$= H_k \delta x_k + v_k \tag{3.21}$$

Where:

$\delta z_k$: denotes what is termed “measurement closing” that relates the actual to the predicted measurements. Which represents the measurement disturbance from the nominal trajectory.

$H_k$: denotes the linearized observation design matrix.

The Fig. 3.2 illustrates the EKF algorithm briefly as outlined below,

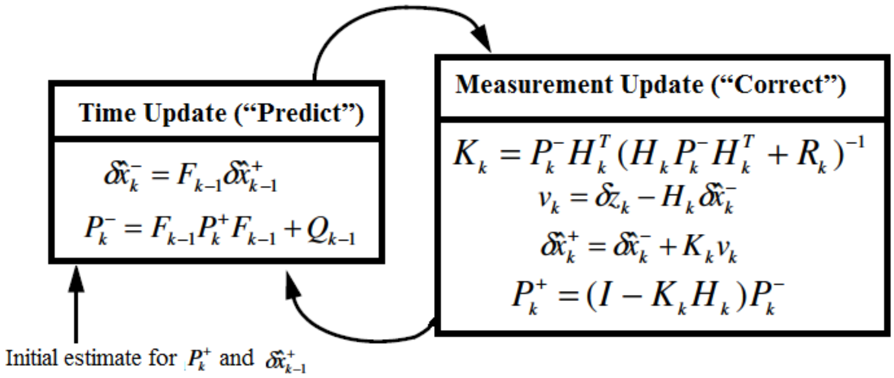

Time Update (“Predict”)
$\delta x_k^- = F_{k-1}\delta x_{k-1}^+$
$P_k^- = F_{k-1}P_k^+F_{k-1} + Q_{k-1}$
Measurement Update (“Correct”)
$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}$
$v_k = \delta z_k - H_k \delta x_k^-$
$\delta x_k^+ = \delta x_k^- + K_k v_k$
$P_k^+ = (I - K_k H_k)P_k^-$
Initial estimate for $P_k^+$ and $\delta x_{k-1}^+$

**Figure 3.2:** Extended Kalman Filter [77].

By combination of Eq. (3.19) and (3.21) the linear system model and measurement model can be reconstructed. Consequently, the optimal estimates of the error states can be achieved by the application of KF on this set of models. Finally, the original system state that is termed state “reset” can be achieved by Eq. (3.17). If this reset step is overcome in closed loop after every Kalman filter measurement update step, or another way to say that, if the feedback on the nominal trajectory happens, the nominal trajectory will belief the estimation outcomes and will diverge consistent with them. Subsequently, this error state will be established as “zero” to show the nominal value as the estimated value. This methodology is termed as Extended Kalman Filter EKF. In contrast, if the reset step is overcome in open loop, or another way to say that, if just the feed forward on the Kalman filter estimates happens, the nominal trajectory will be detected priori and will disregard the estimates of the Kalman filter. This what is termed Linearized Kalman Filter LKF [24, 79]. To describe how the filtering consequences relate to the nominal trajectory for each of the LKF and EKF are illustrated in Fig. 3.3 and Fig. 3.4 respectively. It reveals to us that while the LKF nominal trajectory is independent of the estimation consequences of the filtering, the nominal trajectory of the EKF will track them if the rate of the feedback is similar with the rate of the estimation update.

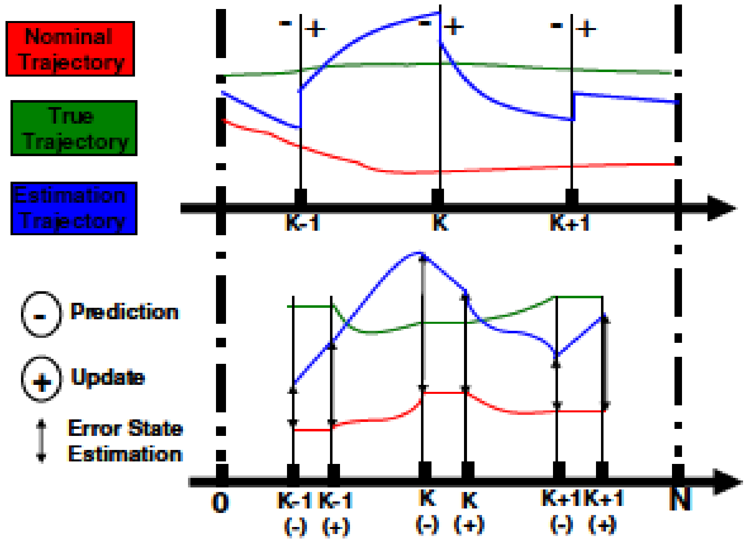


**Figure 3.3:** Relationship of Filtering Outcomes with LKF Nominal Trajectory [44].

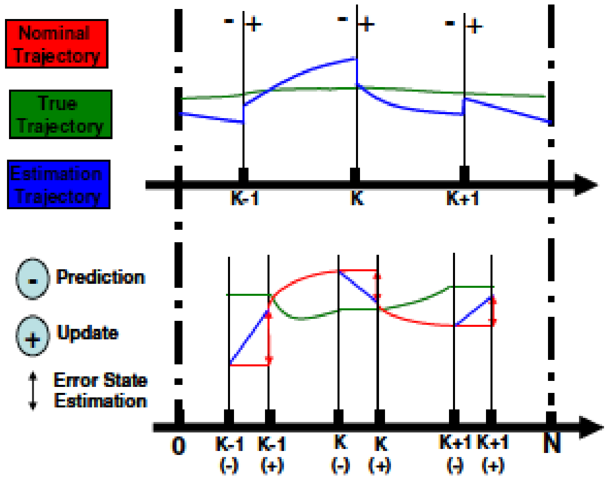


**Figure 3.4:** Relationship of Filtering Outcomes with EKF Nominal Trajectory [44].

### 3.3 The Unscented Kalman Filter (UKF)

One of the most widely used methods for tracking and estimation is the Kalman filter KF, due to its simplicity and optimality. However, the implementation of the KF to nonlinear systems can be difficult. The most prevalently approach is to use the Extended Kalman Filter EKF, which simply transforms all nonlinear models to the linear models for that the traditional linear Kalman filter can be applied. Although the Extended Kalman Filter in many forms is openly used filtering strategy. However more than thirty years of experience with it has led to a general assent within the tracking and control community, which it is difficult to implement, difficult to tune, and only reliable when the systems are almost linear on the time scale of the update intervals. A new linear estimator which is called Unscented Kalman Filter has been developed to support the nonlinear systems. Utilizing the principle that a set of sampled points can be used to parameterize mean and covariance, the estimator gives performance equivalent to the KF for the linear systems, and it applies for the nonlinear systems without the linearization steps required by the EKF. The new approach shows analytically that its performance is superior to that of the EKF in virtually all applications and, indeed, is directly apply to the second order Gauss filter. The method is not restricted to assuming that the distributions of noise sources are Gaussian [80].

The basic concept of the UKF is illustrated in Fig. 3.5. The EKF simply transforms the mean through the given nonlinear function and considers it as the transformed mean, which is valid if the given transformation is linear. The UKF, however, samples several points from the given mean and covariance of the PDF and transforms all of the points through the given nonlinear transformation. The transformed mean and covariance are constructed from the transformed points.

First applied the UKF to INS/GPS integration and demonstrated the UKF's capability of dealing with large and small attitude errors seamlessly [81]. Unfortunately, it was possible for singularities to occur, since attitude was expressed in terms of the Euler angles, and the metric required to measure the attitude difference was incomplete.

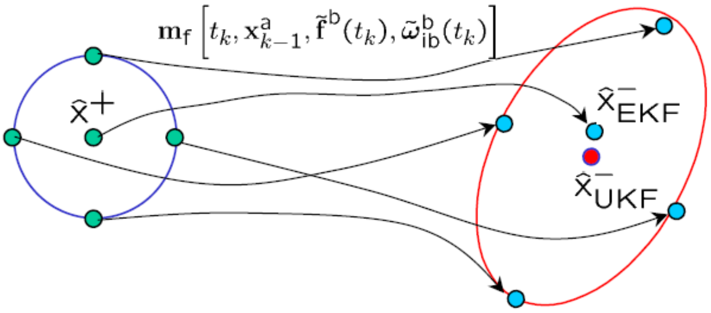

$\mathbf{m}_\mathrm{f}\left[t_k, \mathbf{x}^\mathrm{a}_{k-1}, \tilde{\mathbf{f}}^\mathrm{b}(t_k), \tilde{\omega}^\mathrm{b}_\mathrm{ib}(t_k)\right]$
$\hat{x}^+$
$\hat{x}^-_{\mathrm{EKF}}$
$\hat{x}^-_{\mathrm{UKF}}$

**Figure 3.5:** The concept of the UKF [24].

With the use of quaternion attitude representation, singularities can be resolved. For quaternion-based UKFs for attitude, determination can review [82, 83]. Where according to [82] has been used rotation vectors (singular at ±180°) to express attitude covariance while in [83] has been used a generalized representation, where the singularity can be placed anywhere from 180° to 360°. In section, 3.3.1.1 will review of the unscented transformation (UT). Then, in Section 3.3.1.2, the conception of the UKF will be reviewed. Finally, Section 3.4 Implementation Variations and the mechanism behind UKF to reduce the errors.

### 3.3.1 The Unscented Kalman Filter (UKF)

The UKF handles the approximation issues of the EKF. The state distribution is again represented by a Gaussian Random Variables (GRV). However, it is being specified using a minimal set of carefully selected sample points. These sample points fully capture the true mean and covariance of the GRV, and when propagated through the true nonlinear system, capture the posterior mean and covariance accurately to the second order (Taylor series expansion) for any nonlinearity. To elaborate on this, starting with explaining the unscented transformation [24].

#### 3.3.1.1 Unscented Transformations (UT)

The unscented transformation (UT) refers to the procedure for obtaining a set of weights, $W_i$'s, and sigma points, $x_i$'s, from the given mean, $\bar{x}$, and covariance, $P_x$. The UT method used for calculating the statistics of a random variable, which undergoes a nonlinear transformation [84]. Consider propagating a random variable x (dimension L) through a nonlinear function, y = $f$ (x). Assume x has mean $\bar{x}$ and covariance $P_x$. To calculate the statistics of y, the flowing equations form a matrix $x$ of 2L+1 sigma victors $x_i$:

$$\begin{aligned} &\mathcal{X}_0 = \bar{x} \\ &\mathcal{X}_i = \bar{x} + (\sqrt{(L+\lambda)P_x})_i \qquad i = 1, \dots \dots .., L \\ &\mathcal{X}_i = \bar{x} + (\sqrt{(L+\lambda)P_x})_{i-n} \qquad i = L+1, \dots \dots .., 2L \end{aligned} \tag{3.22}$$

Where $\lambda = \alpha^2(L+k) - L$ is a scaling parameter. The constant $\alpha$ determines the spread of the sigma points around $\bar{x}$ and is usually set to a small positive value (e.g.$10^{-4} \leq \alpha \leq 1$) [35].

The constant $k$ is a secondary scaling parameter which is usually set to 0 or $3 - L$ [85], and $\beta$ is used to incorporate prior knowledge of the distribution of $x$ (for Gaussian distribution, $\beta = 2$ is the optimal [35]. $(\sqrt{(L+\lambda)P_x})_i$ is the $i$th column of the matrix square root (e.g., lower triangular Cholesky factorization). These sigma vectors are propagated through the nonlinear function:

$$y_i = f(x_i) \qquad i = 0, \dots .., 2L, \tag{3.23}$$

and the mean and the covariance for $\boldsymbol{y}$ are approximated using a weighted sample mean and covariance of the posterior sigma points;

$$\bar{y} \approx \sum_{i=0}^{2L} W_i^{(m)} y_i \tag{3.24}$$

$$P_y \approx \sum_{i=0}^{2L} W_i^{(c)} (y_i - \bar{y})(y_i - \bar{y})^T \tag{3.25}$$

With weights $W_i$ given by:

$$W_0^{(m)} = \lambda/(L+\lambda) \tag{3.26}$$

$$W_0^{(c)} = \frac{\lambda}{L+\lambda} + (1 - \alpha^2 + \beta) \qquad (3.27)$$

$$W_i^{(m)} = W_i^{(c)} = \frac{1}{2(L+\lambda)} \qquad i = 1, \ldots.,2L. \qquad (3.28)$$

A block diagram is shown in Fig. 3.6 illustrates the steps in performing the UT. The UT method varies basically from general Monte-Carlo sampling methods which orders of magnitude more sample points in a try to propagate an accurate (possibly non-Gaussian) distribution of the state. The UT results in approximations are precise to the 3rd order for Gaussian inputs for all nonlinearities. For non-Gaussian inputs, its approximations are precise to at least the 2nd order, the accuracy of the third and higher order moments is determined by the choice of $\alpha$ and $\beta$.

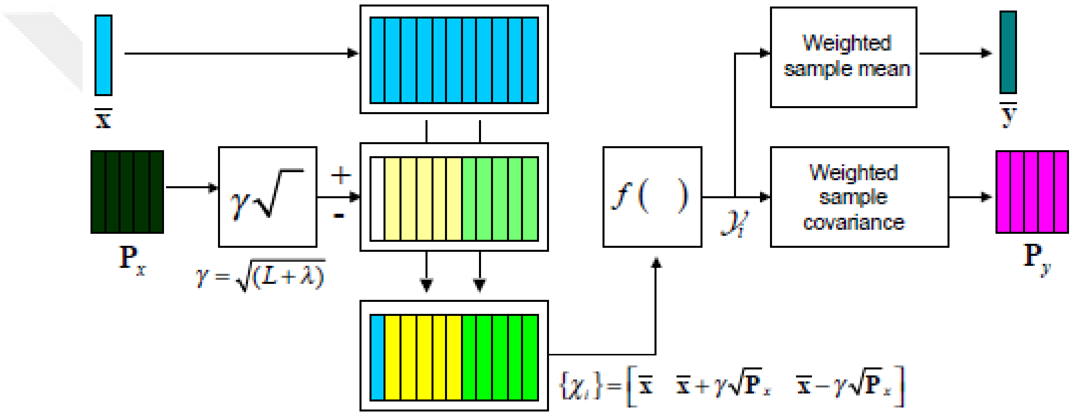

$\bar{\mathbf{x}}$
$\mathbf{P}_x$
$\gamma\sqrt{\ }$
+
-
$\gamma = \sqrt{(L+\lambda)}$
$f(\ )$
$\mathcal{Y}_i$
Weighted sample mean
Weighted sample covariance
$\bar{\mathbf{y}}$
$\mathbf{P}_y$
$\{\chi_i\} = [\bar{\mathbf{x}} \quad \bar{\mathbf{x}} + \gamma\sqrt{\mathbf{P}_x} \quad \bar{\mathbf{x}} - \gamma\sqrt{\mathbf{P}_x}]$

**Figure 3.6:** Block diagram of the UT [35].

In Fig. 3.7 a simple example for a 2-dimentional system, which includes three schemes. The left scheme shows the true mean and covariance propagation of using Monte-Carlo sampling, the schemes in middle show the results of using a linearization approach as would be done in the EKF; the schemes on the right side show the performance of the UT (note just 5 sigma points are required). The superior performance of the UT is clear.

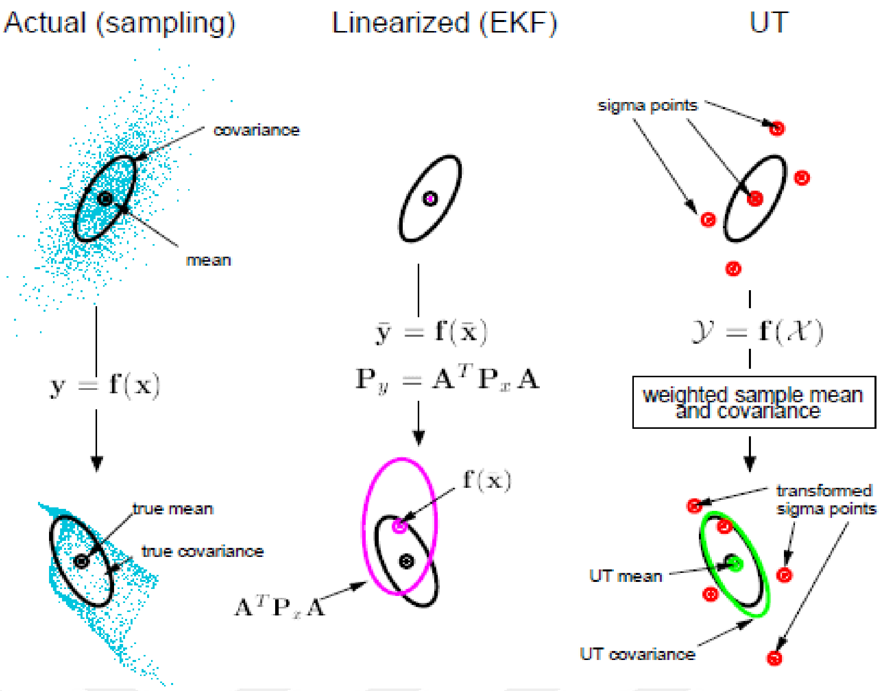

Actual (sampling)
Linearized (EKF)
UT
covariance
mean
$\mathbf{y} = \mathbf{f}(\mathbf{x})$
true mean
true covariance
$\bar{\mathbf{y}} = \mathbf{f}(\bar{\mathbf{x}})$
$\mathbf{P}_y = \mathbf{A}^T \mathbf{P}_x \mathbf{A}$
$\mathbf{f}(\bar{\mathbf{x}})$
$\mathbf{A}^T \mathbf{P}_x \mathbf{A}$
sigma points
$\mathcal{Y} = \mathbf{f}(\mathcal{X})$
weighted sample mean and covariance
transformed sigma points
UT mean
UT covariance

**Figure 3.7:** Example of comparing the UT and EKF performance for mean and covariance propagating with actual sampling [35].

### 3.3.1.2 Unscented Kalman Filter

The unscented Kalman filter UKF is an explicit extension of the UT to the recursive estimation in the following equation:

$$\hat{x}_k = (\text{prediction of } x_k) + K_k[y_k - (\text{prediction of } y_k)] \tag{3.29}$$

Observations $y_k$, estimate state $x_k$, $K_k$ is the gain term and $\hat{x}_k$ is the conditional mean.

Where the state Random Variables RV is redefined as the sequence of the original state $\mathrm{x}_k$, the process noise $\mathrm{v}_k$ and the observation noise $\mathrm{n}_k$ variables: $\mathrm{x}_k^a = [\mathrm{x}_k^T \ \mathrm{v}_k^T \ \mathrm{n}_k^T]^T$. The UT sigma points selection scheme, Eq. (3.22), is applied to this augmented state RV to compute the corresponding sigma matrix, $\mathrm{x}_k^a$. The UKF equations are given in Table 3.1 for the following system:

$$x_k = F(x_{k-1}, u_{k-1}, w_{k-1}) \tag{3.30}$$

$$y_k = H(x_k, v_k) \tag{3.31}$$

Note that no explicit calculations of Jacobians are necessary to implement this algorithm. Furthermore, the overall number of calculations is of the same order as the EKF [35].

**Table 3.1:** Equations of Unscented Kalman Filter [35].

Initialize with

$$\hat{\mathbf{x}}_{\mathbf{0}} = \mathbb{E}[\mathbf{x}_{\mathbf{0}}], \tag{3.32}$$

$$\mathbf{P}_{\mathbf{0}} = \mathbb{E}[(\mathbf{x}_{\mathbf{0}} - \hat{\mathbf{x}}_{\mathbf{0}})(\mathbf{x}_{\mathbf{0}} - \hat{\mathbf{x}}_{\mathbf{0}})^{\boldsymbol{T}}], \tag{3.33}$$

$$\hat{\mathbf{x}}_{\mathbf{0}}^{\boldsymbol{a}} = \mathbb{E}[\mathbf{x}^{\boldsymbol{a}}] = [\hat{\mathbf{x}}_{\mathbf{0}}^{\boldsymbol{T}} \quad \mathbf{0} \quad \mathbf{0}]^{\boldsymbol{T}}, \tag{3.34}$$

$$\mathbf{P}_{\mathbf{0}}^{\mathrm{a}} = \mathbb{E}[(\mathbf{x}_{\mathbf{0}}^{\mathrm{a}} - \hat{\mathbf{x}}_{\mathbf{0}}^{\mathrm{a}})(\mathbf{x}_{\mathbf{0}}^{\mathrm{a}} - \hat{\mathbf{x}}_{\mathbf{0}}^{\mathrm{a}})^{\mathrm{T}}] = \begin{bmatrix} \mathbf{P}_{\mathbf{0}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}^{\mathrm{v}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R}^{\mathrm{n}} \end{bmatrix}. \tag{3.35}$$

For $k \in \{\mathbf{1}, \dots, \infty\}$,

calculate the sigma points:

$$\boldsymbol{\mathcal{X}}_{\mathbf{k-1}}^{\mathbf{a}} = [\hat{\mathbf{x}}_{\mathbf{k-1}}^{\mathbf{a}} \quad \hat{\mathbf{x}}_{\mathbf{k-1}}^{\mathbf{a}} + \boldsymbol{\gamma}\sqrt{\mathbf{P}_{\mathbf{k-1}}^{\mathbf{a}}} \quad \hat{\mathbf{x}}_{\mathbf{k-1}}^{\mathbf{a}} - \boldsymbol{\gamma}\sqrt{\mathbf{P}_{\mathbf{k-1}}^{\mathbf{a}}}] \tag{3.36}$$

The time-update equations are:

$$\boldsymbol{\mathcal{X}}_{\boldsymbol{k|k-1}}^{\mathbf{x}} = \mathbf{F}(\boldsymbol{\mathcal{X}}_{\boldsymbol{k-1}}^{\boldsymbol{x}}, \mathbf{u}_{\boldsymbol{k-1}}, \boldsymbol{\mathcal{X}}_{\boldsymbol{k-1}}^{\boldsymbol{v}}) \tag{3.37}$$

$$\hat{\mathbf{x}}_{\boldsymbol{k}}^{-} = \sum_{\boldsymbol{i=0}}^{\boldsymbol{2L}} \boldsymbol{W}_{\boldsymbol{i}}^{(\boldsymbol{m})} \boldsymbol{\mathcal{X}}_{\boldsymbol{i,k|k-1}}^{\boldsymbol{x}} \tag{3.38}$$

$$\boldsymbol{P}_{\boldsymbol{k}}^{-} = \sum_{\boldsymbol{i=0}}^{\boldsymbol{2L}} \boldsymbol{W}_{\boldsymbol{i}}^{(\boldsymbol{c})} \left(\boldsymbol{\mathcal{X}}_{\boldsymbol{i,k|k-1}}^{\boldsymbol{x}} - \hat{\boldsymbol{x}}_{\boldsymbol{k}}^{-}\right)\left(\boldsymbol{\mathcal{X}}_{\boldsymbol{i,k|k-1}}^{\boldsymbol{x}} - \hat{\boldsymbol{x}}_{\boldsymbol{k}}^{-}\right)^{\boldsymbol{T}}, \tag{3.39}$$

$$\boldsymbol{Y}_{\boldsymbol{i,k|k-1}} = \mathbf{H}\left(\boldsymbol{\mathcal{X}}_{\boldsymbol{i,k|k-1}}^{\boldsymbol{x}}, \boldsymbol{\mathcal{X}}_{\boldsymbol{i,k-1}}^{\boldsymbol{n}}\right), \tag{3.40}$$

$$\hat{\mathbf{y}}_{\boldsymbol{k}}^{-} = \sum_{\boldsymbol{i=0}}^{\boldsymbol{2L}} \boldsymbol{W}_{\boldsymbol{i}}^{(\boldsymbol{m})} \boldsymbol{Y}_{\boldsymbol{i,k|k-1}}, \tag{3.41}$$

and the measurement-update equations are:

$$\mathbf{P}_{\bar{\boldsymbol{y}}_{\boldsymbol{k}}\bar{\boldsymbol{y}}_{\boldsymbol{k}}} = \sum_{\boldsymbol{i=0}}^{\boldsymbol{2L}} \boldsymbol{W}_{\boldsymbol{i}}^{(\boldsymbol{c})} (\boldsymbol{Y}_{\boldsymbol{i,k|k-1}} - \hat{\mathbf{y}}_{\boldsymbol{k}}^{-})(\boldsymbol{Y}_{\boldsymbol{i,k|k-1}} - \hat{\boldsymbol{y}}_{\boldsymbol{k}}^{-})^{\boldsymbol{T}}, \tag{3.42}$$

$$\mathbf{P}_{\mathbf{x}_{\boldsymbol{k}}\boldsymbol{y}_{\boldsymbol{k}}} = \sum_{\boldsymbol{i=0}}^{\boldsymbol{2L}} \boldsymbol{W}_{\boldsymbol{i}}^{(\boldsymbol{c})} (\boldsymbol{\mathcal{X}}_{\boldsymbol{i,k|k-1}} - \hat{\mathbf{x}}_{\boldsymbol{k}}^{-})(\boldsymbol{Y}_{\boldsymbol{i,k|k-1}} - \hat{\boldsymbol{y}}_{\boldsymbol{k}}^{-})^{\boldsymbol{T}}, \tag{3.43}$$

$$\boldsymbol{K}_{\boldsymbol{k}} = \mathbf{P}_{\boldsymbol{x}_{\boldsymbol{k}}\boldsymbol{y}_{\boldsymbol{k}}} \mathbf{P}_{\bar{\boldsymbol{y}}_{\boldsymbol{k}}\bar{\boldsymbol{y}}_{\boldsymbol{k}}}^{\mathbf{-1}} \tag{3.44}$$

$$\hat{\mathbf{x}}_{\mathbf{k}} = \hat{\mathbf{x}}_{\boldsymbol{k}}^{-} + \boldsymbol{K}_{\boldsymbol{k}}(\boldsymbol{y}_{\boldsymbol{k}} - \hat{\boldsymbol{y}}_{\boldsymbol{k}}^{-}) \tag{3.45}$$

$$\mathbf{P}_{\boldsymbol{k}} = \mathbf{P}_{\boldsymbol{k}}^{-} - \boldsymbol{K}_{\boldsymbol{k}} \mathbf{P}_{\bar{\boldsymbol{y}}_{\boldsymbol{k}}\bar{\boldsymbol{y}}_{\boldsymbol{k}}} \boldsymbol{K}_{\boldsymbol{k}}^{\boldsymbol{T}}, \tag{3.46}$$

Where,

$\mathbf{x}^{\boldsymbol{a}} = [\mathbf{x}^{\boldsymbol{T}} \quad \mathbf{v}^{\boldsymbol{T}} \quad \mathbf{n}^{\boldsymbol{T}}]^{\boldsymbol{T}}$, $\quad \boldsymbol{\mathcal{X}}^{\boldsymbol{a}} = [(\boldsymbol{\mathcal{X}}^{\boldsymbol{x}})^{\boldsymbol{T}} \quad (\boldsymbol{\mathcal{X}}^{\boldsymbol{v}})^{\boldsymbol{T}} \quad (\boldsymbol{\mathcal{X}}^{\boldsymbol{n}})^{\boldsymbol{T}}]^{\boldsymbol{T}}$, $\quad \boldsymbol{\gamma} = \sqrt{\boldsymbol{L} + \boldsymbol{\lambda}}$ , the composite scaling parameter; $\boldsymbol{\lambda}$, the dimension of the augmented state; $L$, the process noise covariance; $\mathbf{R}^{\mathbf{v}}$, the measurement-noise covariance; $\mathbf{R}^{\mathbf{n}}$, and as in Eq. (3.28) the weights ($\boldsymbol{W}_{\boldsymbol{i}}$) are calculated.

### 3.4 Implementation Variations

For the special case (often encountered) where the process and measurement noise are purely additive, the computational complexity of the UKF can be reduced. In such a case, the system state require not to be increased with the noise RVs. This reduces the dimension of the sigma points as well as the whole number of sigma points used. The all covariance of the noise source are then joined into the state covariance using a simple additive procedure. In Table 3.2 this implementation was given. The complexity of the algorithm is represented in order $L^3$, where $L$ is the dimension of the state. This is the similar complexity as the EKF. The forming of the sample previous covariance matrix $\mathrm{P}_k^-$ represents the most costly operation. Depending on the form of $F$, this may be simplified; for example, with parameter estimation, the complexity reduces to order $L^2$ [35].

When the noise of the system are in additive form, we can simplify the UKF algorithm. Consider the system dynamic equation and measurement equation to be in the forms:

$$x_k = F(x_{k-1}, u_{k-1}) + w_k \tag{3.47}$$

$$y_k = H(x_{k-1}) + v_k \tag{3.48}$$

The Table 3.2 includes the implementation of the additive version of UKF for the above system.

**Table 3.2:** Unscented Kalman Filter–additive (zero mean) noise case [35].

Initialize with

$$\mathbf{\hat{x}_0} = \mathbb{E}[\mathbf{x_0}], \tag{3.49}$$

$$\mathbf{P_0} = \mathbb{E}[(\mathbf{x_0} - \mathbf{\hat{x}_0})(\mathbf{x_0} - \mathbf{\hat{x}_0})^{\boldsymbol{T}}]. \tag{3.50}$$

For $k \in \{\mathbf{1}, \dots, \infty\}$,

calculate the sigma points:

$$\boldsymbol{\mathcal{X}}_{\mathbf{k-1}} = [\mathbf{\hat{x}_{k-1}} \quad \mathbf{\hat{x}_{k-1}} + \boldsymbol{\gamma}\sqrt{\mathbf{P_{k-1}}} \quad \mathbf{\hat{x}_{k-1}} - \boldsymbol{\gamma}\sqrt{\mathbf{P_{k-1}}}]. \tag{3.51}$$

The time-update equations are:

$$\boldsymbol{\mathcal{X}}^*_{\boldsymbol{k|k-1}} = \mathbf{F}(\boldsymbol{\mathcal{X}}_{\boldsymbol{k-1}}, \mathbf{u}_{\boldsymbol{k-1}}), \tag{3.52}$$

$$\mathbf{\hat{x}}_{\boldsymbol{k}}^- = \sum_{\boldsymbol{i=0}}^{\boldsymbol{2L}} \boldsymbol{W_i^{(m)}}\, \boldsymbol{\mathcal{X}}^*_{\boldsymbol{i,k|k-1}} \tag{3.53}$$

$$\mathbf{P}_{\boldsymbol{k}}^- = \sum_{\boldsymbol{i=0}}^{\boldsymbol{2L}} \boldsymbol{W_i^{(c)}} \left(\boldsymbol{\mathcal{X}}^*_{\boldsymbol{i,k|k-1}} - \mathbf{\hat{x}}_{\boldsymbol{k}}^-\right)\left(\boldsymbol{\mathcal{X}}^*_{\boldsymbol{i,k|k-1}} - \mathbf{\hat{x}}_{\boldsymbol{k}}^-\right)^{\boldsymbol{T}} + \mathbf{R}^{\mathrm{v}}, \tag{3.54}$$

(augment sigma points)

$$\boldsymbol{\mathcal{X}}_{\boldsymbol{k|k-1}} = [\boldsymbol{\mathcal{X}}^*_{\boldsymbol{k|k-1}} \quad \boldsymbol{\mathcal{X}}^*_{\boldsymbol{0,k|k-1}} + \boldsymbol{\gamma}\sqrt{\mathbf{R^v}} \quad \boldsymbol{\mathcal{X}}^*_{\boldsymbol{0,k|k-1}} - \boldsymbol{\gamma}\sqrt{\mathbf{R^v}}], \tag{3.55}$$

$$\boldsymbol{Y}_{\boldsymbol{k|k-1}} = \mathbf{H}\left(\boldsymbol{\mathcal{X}}_{\boldsymbol{k|k-1}}\right), \tag{3.56}$$

$$\boldsymbol{\hat{y}}_{\boldsymbol{k}}^- = \sum_{\boldsymbol{i=0}}^{\boldsymbol{2L}} \boldsymbol{W_i^{(m)}}\, \boldsymbol{Y}_{\boldsymbol{i,k|k-1}}, \tag{3.57}$$

and the measurement-update equations are:

$$\mathbf{P}_{\boldsymbol{\bar{y}_k\bar{y}_k}} = \sum_{\boldsymbol{i=0}}^{\boldsymbol{2L}} \boldsymbol{W_i^{(c)}} \left(\boldsymbol{Y}_{\boldsymbol{i,k|k-1}} - \boldsymbol{\hat{y}}_{\boldsymbol{k}}^-\right)\left(\boldsymbol{Y}_{\boldsymbol{i,k|k-1}} - \boldsymbol{\hat{y}}_{\boldsymbol{k}}^-\right)^{\boldsymbol{T}} + \mathbf{R^n}, \tag{3.58}$$

$$\mathbf{P}_{\mathbf{x}_{\boldsymbol{k}}\boldsymbol{y_k}} = \sum_{\boldsymbol{i=0}}^{\boldsymbol{2L}} \boldsymbol{W_i^{(c)}} (\boldsymbol{\mathcal{X}}_{\boldsymbol{i,k|k-1}} - \mathbf{\hat{x}}_{\boldsymbol{k}}^-)(\boldsymbol{Y}_{\boldsymbol{i,k|k-1}} - \boldsymbol{\hat{y}}_{\boldsymbol{k}}^-)^{\boldsymbol{T}}, \tag{3.59}$$

$$\boldsymbol{K_k} = \mathbf{P}_{\boldsymbol{x_k y_k}} \mathbf{P}^{\boldsymbol{-1}}_{\boldsymbol{\bar{y}_k\bar{y}_k}} \tag{3.60}$$

$$\mathbf{\hat{x}_k} = \mathbf{\hat{x}}_{\boldsymbol{k}}^- + \boldsymbol{K_k}(\boldsymbol{y_k} - \boldsymbol{\hat{y}}_{\boldsymbol{k}}^-) \tag{3.61}$$

$$\mathbf{P}_{\boldsymbol{k}} = \mathbf{P}_{\boldsymbol{k}}^- - \boldsymbol{K_k}\mathbf{P}_{\boldsymbol{\bar{y}_k\bar{y}_k}}\boldsymbol{K}_{\boldsymbol{k}}^{\boldsymbol{T}}, \tag{3.62}$$

where $\boldsymbol{\gamma} = \sqrt{\boldsymbol{L} + \boldsymbol{\lambda}}$ , $\boldsymbol{\lambda}$ is the composite scaling parameter, the dimension of the state is $L$, the process-noise covariance is $\mathbf{R}^{\mathbf{v}}$, the measurement-noise covariance is $\mathbf{R}^{\mathbf{n}}$ and as in Eq. (3.28) the weights ($\boldsymbol{W}_{\boldsymbol{i}}$) are calculated.

---

Here the sigma points are augmented with additional points derived from the matrix square root of the process noise covariance. This needs setting $L \rightarrow 2L$ and recalculating the various weights $\boldsymbol{W}_{\boldsymbol{i}}$ accordingly. Alternatively, may redraw a whole new set of sigma points, i.e. $\boldsymbol{\mathcal{X}}_{\boldsymbol{k}|\boldsymbol{k-1}} = [\hat{\mathbf{x}}_{\mathbf{k}}^{-} \quad \hat{\mathbf{x}}_{\mathbf{k}}^{-} + \boldsymbol{\gamma}\sqrt{\mathbf{P}_{\mathbf{k}}^{-}} \quad \hat{\mathbf{x}}_{\mathbf{k}}^{-} - \boldsymbol{\gamma}\sqrt{\mathbf{P}_{\mathbf{k}}^{-}}]$. This alternative approach results in fewer sigma points being used, and also rejects any odd-moments information captured by the original propagated sigma points.

The Fig. 3.8 illustrates the UKF algorithm briefly as outlined below:

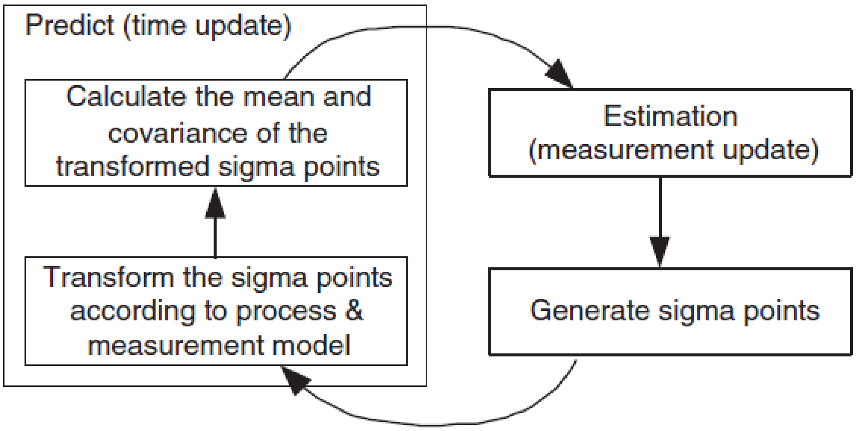

Predict (time update)
Calculate the mean and covariance of the transformed sigma points
Transform the sigma points according to process & measurement model
Estimation (measurement update)
Generate sigma points

**Figure 3.8:** A high level of the operation of the Unscented Kalman filter [86].

# CHAPTER FOUR

## IMPLEMENTATION OF UKF FOR NAVIGATION OF PIG ROBOT

### 4.1 Introduction

In this chapter there will be a briefly review for each previous chapters and how to combine their most parts together in the work of this thesis.

Thus, the Pipe Inspection Gage PIG robots as shown in Fig. 4.1 below where they have been employed for many years to achieve an important benefit in field of non-destructive tests for detection the several of operational defects in gas, oil pipelines and other pipelines that usually happen during their actual service life.

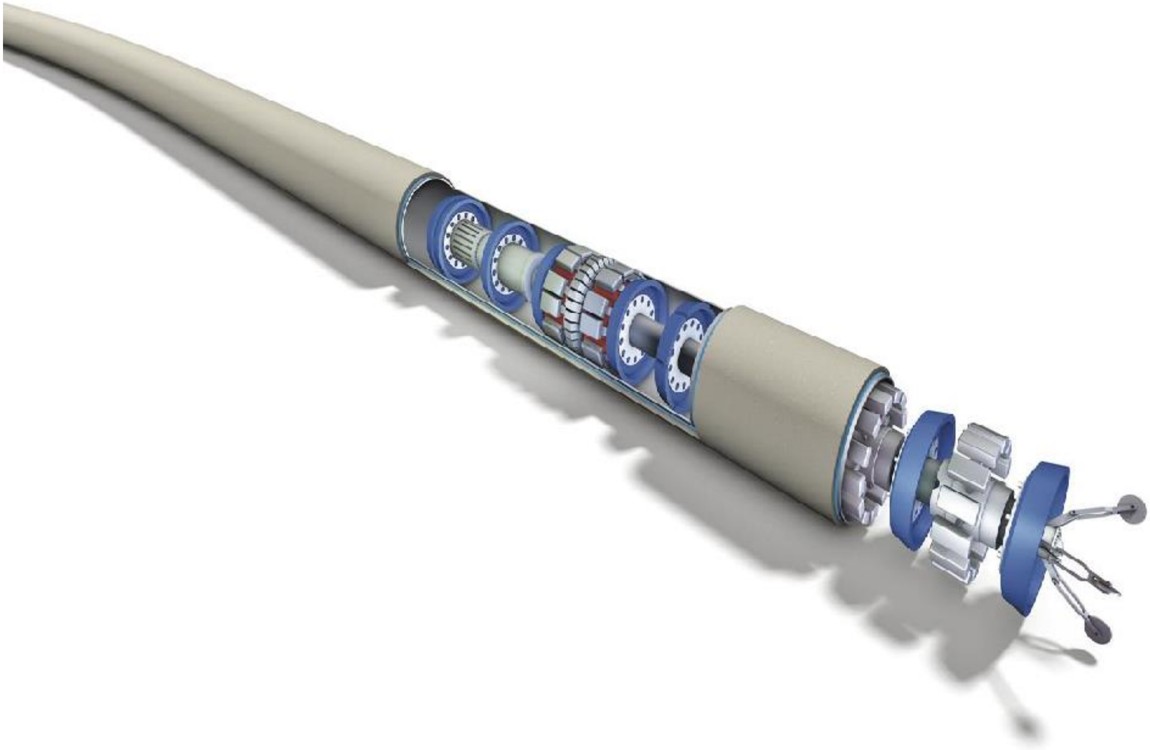
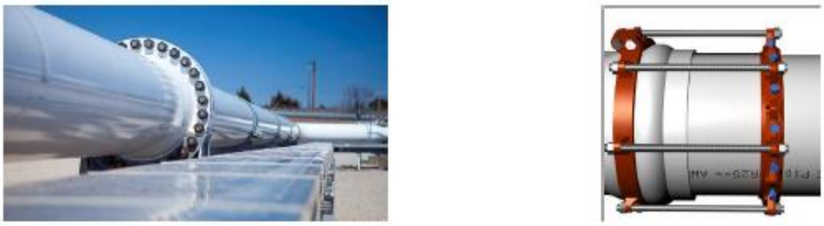


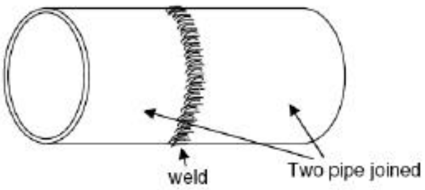
**Figure 4.1:** Pipeline Inspection Gauge (PIG) Robot [52].

In this project, we developed an integrated navigation system algorithm for a PIG robot. In which is usually used many sensors for detecting the desired pipeline's parameters, but the important task is to combine the desired pipeline parameters that are obtained from the sensors' data with corresponding to the pipeline location.

As we deal with INS in this project, it is usually has errors and for stabilizing these errors there are many methods that are represented by adding extra sensors. One of the most famous and practical sensor for this purpose is the GPS sensor. However, unfortunately, in our project the errors of INS cannot be stabilized depending on GPS sensor, because the journey of our PIG robot will be through an assumed oil pipeline that is well-known for most oil pipelines to be buried under ground far away from the GPS signal. Therefore, there are many new methods that are only used for reducing the errors rate of the INS, such as adding odometer as an external sensor, zero velocity updates and coordinate updates or by using some smoothing methods like KF and UKF. However, in our navigation algorithm, we will try to stabilize the errors of INS by using counting for the number of pipes in an assumed oil pipeline that has known length. This assumed oil pipeline represents the journey of our PIG robot inside of Karbala city in Iraq. Karbala city is situated at 32.62 latitude, 44.02 longitude and at height 32 meters above the sea level where our PIG robot starts its journey. The idea behind of counting the pipes' number in the oil pipeline is depending on simulation of a counting sensor for the junctions between the pipes, which might be jointed with each other by welding or by other methods as shown in Fig. 4.2 below:



(a) Flange Type (b) Push-On Type



(c) Welding Type

**Figure 4.2:** Samples of Pipeline Joints [52].

According to the previous studies, become well known that the errors of inertial navigation grow unlimitedly, but if the navigation solutions were constrained then the growth of errors would be limited. So our PIG robot does not have a velocity meter in the directions perpendicular to the forward movement, because all of the perpendicular velocities will be reset to zero as a concept of the constrain inertial navigation that allow to reset the biases. This is named non-holonomic constraints in navigation terms.

To design a navigation system, it is necessary to take in account the consideration of the environment conditions that influence the body motion. In this chapter, the constraints depended on pipeline measurement structure for pig navigation will be reviewed. The effect of these constraints will be analyzed and compared with measurement data of INS navigation using UKF throgh the next chapter.

In the previous studies, has been developed a quaternion-based UKF for the integration of GPS and INS following the former approach [87]. However in this study, because it is assumed the oil pipeline to be buried under the ground so the interrupted GPS signal there may cause increasing in the errors of the positions. For that reason, there will be instead of GPS, a counter for counting pipes, which are found in the pipeline that represents the trajectory of the PIG robot to support this approach.

## 4.2 Dynamic of PIG robot with Non-Holonomic Constraints

Inertial Measurement Unit IMU is a device that includes the two significant inertial sensors set (accelerometers and gyroscopes). By utilizing IMU and INS algorithm, the determination of the position, velocity and attitude of the PIG robot will be possible as shown in Fig. 4.3 and the noise of IMU is inversely propotional with its size as shown in Fig. 4.4.

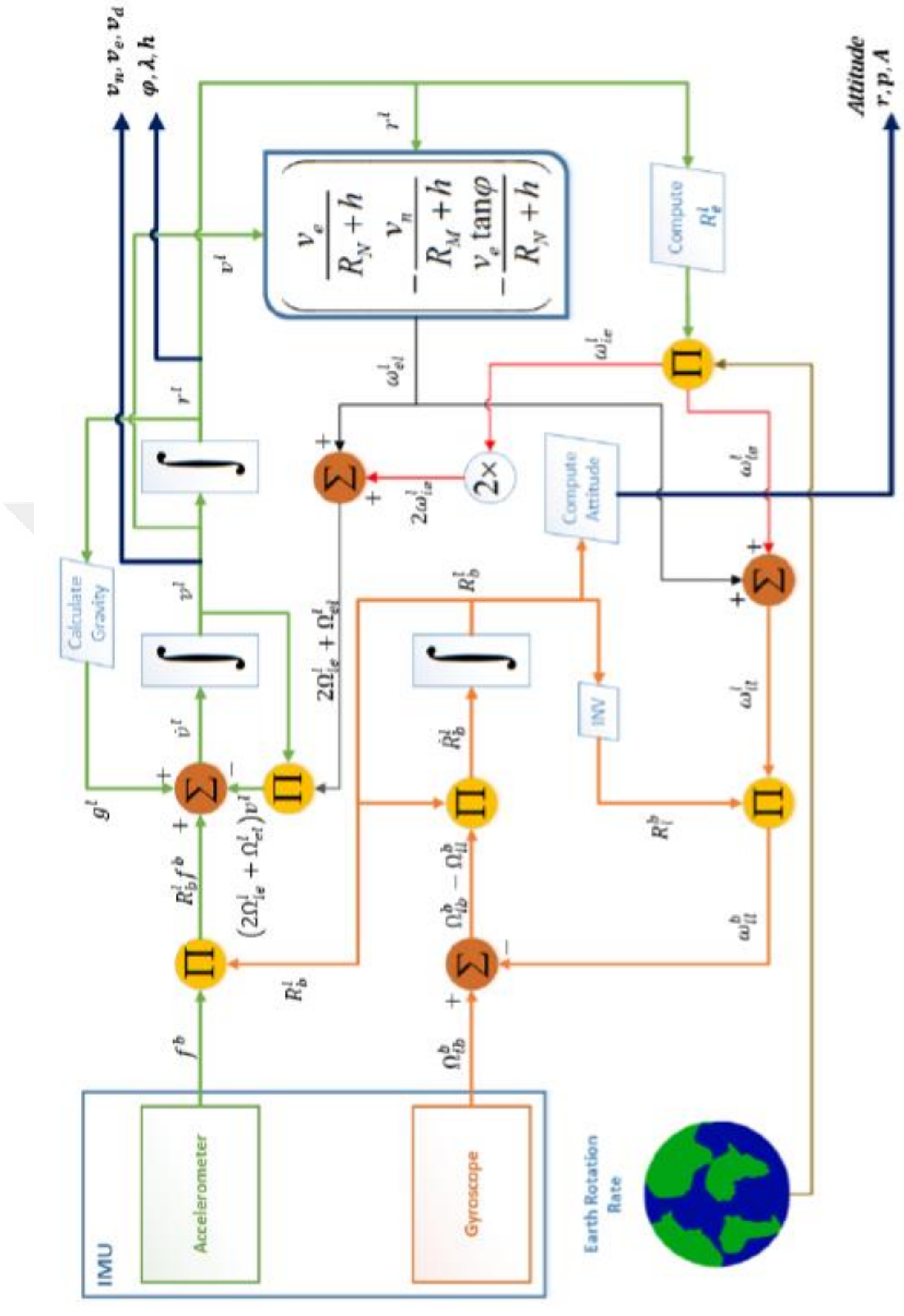


**Figure 4.3:** Block Diagram of The IMU Mechanization Process [88].

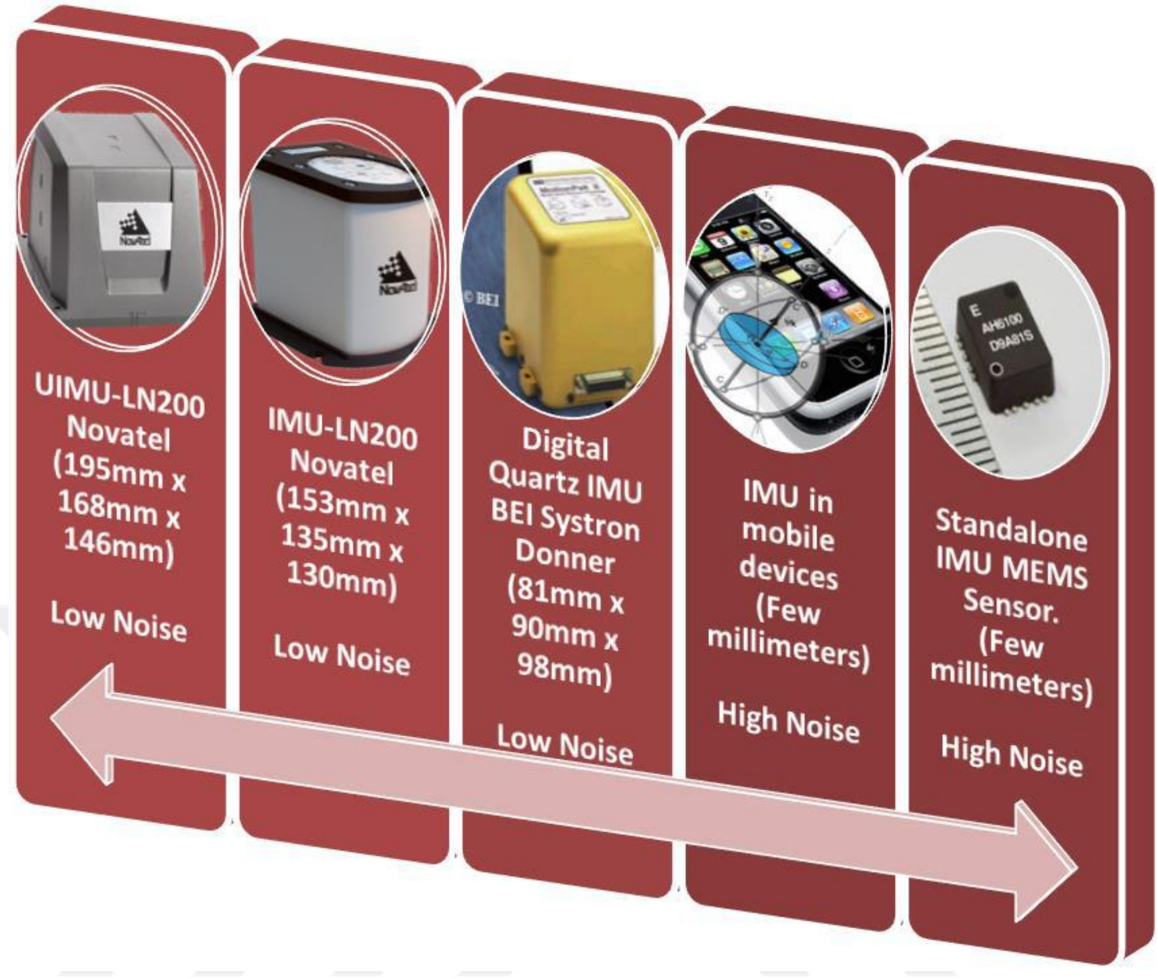

UIMU-LN200 Novatel (195mm x 168mm x 146mm)
Low Noise
IMU-LN200 Novatel (153mm x 135mm x 130mm)
Low Noise
Digital Quartz IMU BEI Systron Donner (81mm x 90mm x 98mm)
Low Noise
IMU in mobile devices (Few millimeters)
High Noise
Standalone IMU MEMS Sensor. (Few millimeters)
High Noise

**Figure 4.4:** Comparison of IMUs [52].

In the body frame of our PIG robot, the output of its accelerometers are given the specific body force that is denoted by $f^b$, which provides its linear body velocity that is denoted by $v^b$, which is expressed in vector form in three directions by the following form:

$$v^b = \begin{bmatrix} v_x^b \\ v_y^b \\ v_z^b \end{bmatrix} \tag{4.1}$$

Since our PIG robot is considered as a land vehicle, so its motion can be run by two non-holonomic constraints. Firstly, according to the dynamic of its movement inside of the pipeline, the robot does not slide on or jump off the ground, therefore the perpendicular velocities to the forward direction are assumed to be zero. Secondly, because of these assumptions it is necessary to ignore accelerometer vibrations, suspension dynamics, and assume the trajectory of our PIG robot inside the oil pipeline to be smoothly or in other words ignoring sideslip of the PIG robot during the turns in its trajectory. So, depending on the PIG axes defined in Fig. 4.5 the approximated model of the horizontal and vertical velocities perpendicular to the forward direction can be stated respectively as following:

$$v_y^b \approx 0$$

$$v_z^b \approx 0 \qquad (4.2)$$

The expression of the errors in the horizontal and vertical velocities can be as:

$$\delta z_k = \begin{bmatrix} \delta v_x^b \\ \delta v_y^b \\ \delta v_z^b \end{bmatrix} = \begin{bmatrix} v_{x,INS}^b - v_{x,True}^b \\ v_{y,INS}^b - v_{y,True}^b \\ v_{z,INS}^b - v_{z,True}^b \end{bmatrix} = \begin{bmatrix} v_{x,INS}^b - v_{x,True}^b \\ v_{y,INS}^b \\ v_{z,INS}^b \end{bmatrix} \qquad (4.3)$$

Since our robot is tightly moves inside the assumption oil pipeline, for this reason our PIG robot will has only forward motion. So, non-holonomic constraints approach is convenient to be applied for the navigation of the robot.

The gravity and the pressure inside the pipeline generate the specific force. The following model gives the expression of the specific force:

$$f^b = C_n^b(\dot{v}^n + g^n + \Delta f_{cent.}^n) \approx g^b \qquad (4.4)$$

Where;

$f^b$; is the total specific force that represents the output of accelerometer.

$g^n$ ; is the projection of gravity acceleration along the navigation frame.

$\Delta f_{cent.}^n$ ; are related to the centrifugal terms.

When the velocity of the robot is a constant velocity, the spesific force will be near the gravity acceleration at body frame.

The output of gyroscope is angular velocity, which is denoted by $\omega_{ib}^b$. Normally, there are three gyroscopes in the PIG robot and these are responsible about the measureing and recording of the angular velocity along the three axes x, y and z as shown in Fig. 4.3, and these rates according to the three axes x, y and z are denoted by ($\omega_{xb}^b$, $\omega_{yb}^b$, $\omega_{zb}^b$), which are given in the following vector form:

$$\omega_{ib}^b = \begin{bmatrix} \omega_x^b \\ \omega_y^b \\ \omega_z^b \end{bmatrix} \qquad (4.5)$$

Where, the symbols $\omega_x^b, \omega_y^b$ and $\omega_z^b$ represent the angular velocity vectors along the $x^b, y^b$ and $z^b$ axis respectively.

The motion of the robot is assumed to be in a smooth trajectory and the PIG robot is assumed to be fitted to the inner diameter of pipeline. This fitting should include a permissible tolerance that is necessary to make the PIG robot pass smoothly through the junctions' rings inside the pipeline. Thus, the angular velocity of the system will be near the zero value:

$$\omega_{ib}^b \approx \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.6}$$

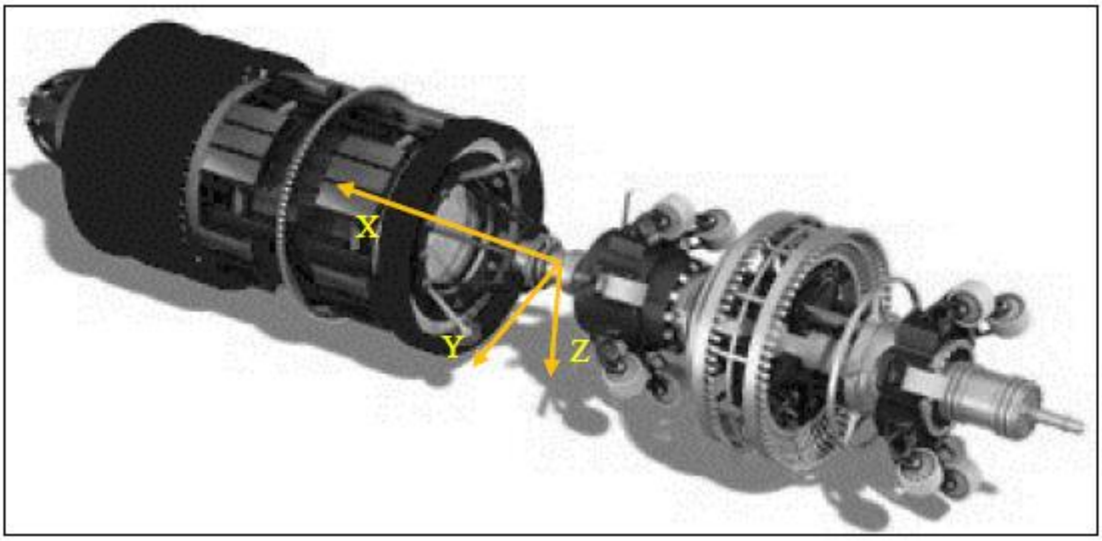


**Figure 4.5:** Definition of PIG Axes [52].

### 4.3 Defination of Navigation Parameters

Always the outputs of IMU have some errors. In this study, we will consider the constant bias and white noise for sensors of our robot. Where, the symbols $b_a$, $b_g$, $w_a$ and $w_g$ will represent the accelerometers' bias, gyroscopes' fixed bias, white noise of accelerometers and white noise of gyroscopes respectively. The biases are given in following forms:

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

$$b_g = [b_{gx} \quad b_{gy} \quad b_{gz}]^T \tag{4.7}$$

Where, the index $a$ and $g$ refer to the accelerometer and gyroscope, respectively.

So, the noise vector of IMU can be modelized as:

$$w_k = \begin{bmatrix} w_a \\ w_g \end{bmatrix} \tag{4.8}$$

Where:

$w_k$; is the sensors noise vector of the accelerometer and gyroscope.

Thus, the observation model for the accelerometer measurements is [89]:

$$\tilde{f}_b = z_{f_b} = f^b + b_a + w_a \tag{4.9}$$

And the observation model for the gyroscope measurements is [89]:

$$\widetilde{\omega}_{ib}^b = z_{g_b} = \omega_{ib}^b + b_g + w_g \tag{4.10}$$

The expression of the accelerometer and gyroscope measurements in vector model form will be as following;

$$\tilde{u} = \begin{bmatrix} \tilde{f}_b \\ \widetilde{\omega}_{ib}^b \end{bmatrix} \tag{4.11}$$

Where;

$\tilde{u}$ ; is the accelerometer and gyroscope measurements vector (IMU outputs).

$\tilde{f}_b$ ; is the accelerometer measurement vector$(m/s^2)$.

$\widetilde{\omega}_{ib}^b$ ; is the gyroscope measurement vector$(rad/s)$.

Since the navigation states vector $(x)$ of the robot is given in the following form:

$$x = [\, p^T \;\; v^T \;\; \Theta^T \;\; b_g^T \;\; b_a^T \,]_{15\times 1}^T \tag{4.12}$$

Where:

$p$; refers to the robot position vector $(3 \times 1)$ in a coordinate form (latitude$(\varphi)$, longitude $(\lambda)$, altitude $(h)$) in a location of its trajectory.

$v$; refers to the velocity vector $(3 \times 1)$ of the robot along the North, East and Down in a location of the robots trajectory.

$\Theta$ ; is the attitude vector $(3 \times 1)$, which represents the angles (Roll, Pitch and Yaw) of the robot in a location of its trajectory.

So, both of $b_a$and $b_g$ should be in vector of $(3 \times 1)$.

By using of the formulation in chapter two, the state space form of navigation algorithem will be modelized as following:

$$x_k = f(x_{k-1}, \tilde{u}_{k-1}, w_{k-1}) \tag{4.13}$$

### 4.4 The Odometer Measurement and Counting System

Generally, odometers are used for measuring the forward distance by utilizing the distance measurement wheels technique as shown in Fig. 4.6. By differentiating the travelled distance by the wheels over the time, the odometer can give the information of the forward velocity. Thus, it is used in our PIG robot for the same purpose. However, when the PIG robot is driven through the pipeline that might has medium flow with differential pressure. The differential pressure may led for some sudden variation in the standard velocity magnitude of the PIG robot during its journey, which is either because of the losing in its mechanical performance or according to the sliding of its wheels on the pipe wall. Moreover, because of some lubrication that might be found on the pipe wall the sliding of wheels will occur with or without existence the differential pressure in the medium flow.

Therefore, depending on odometers to navigate the positions needs to be supported with an extra technique like GPS. However, in this study, GPS signal assumed to be not available, so instead of GPS we will use a counting system to count the number of pipes in the pipeline to update navigation errors. This is considered the main objective of this thesis, which is depending in its work on a sensor to sense the junctions between the pipes in the pipeline as shown in Fig. 4.2. In many cases, the pipelines are constructed by using pipes have the same length with small standard deviation (STD). So, in such cases using the counting system to reducing error of the position will be more beneficial than GPS.

So, the vector form of odometer output is:

$$z_{odo} = \begin{bmatrix} z_{odo_x} \\ z_{odo_y} \\ z_{odo_z} \end{bmatrix} \tag{4.14}$$

Scince the robot motion is under non-holonomic constraints, so the output of the odometer will be just along x-direction so:

$$z_{odo_y} \approx z_{odo_z} \approx 0$$

Thus, the odometer will correct the navigation output according to the scenario in Fig. 4.7 sharing with the counting system over a short period ( pipeline length).

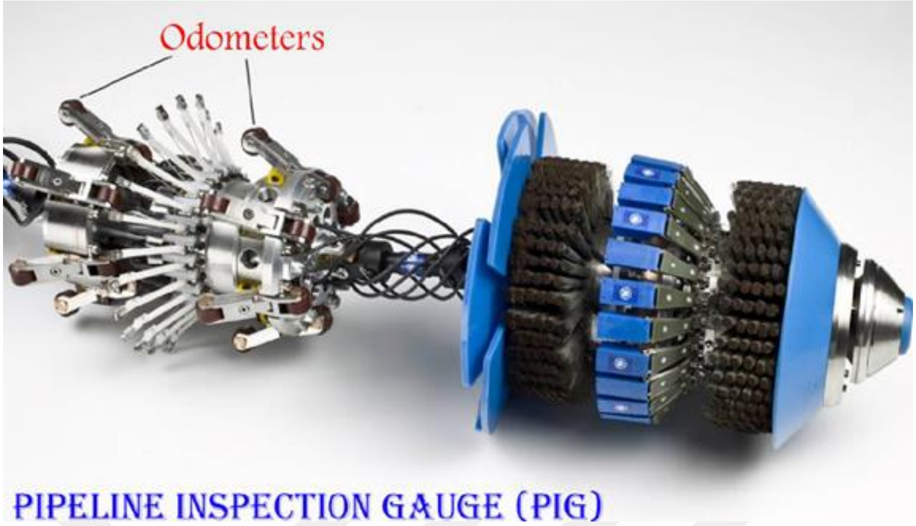


**Figure 4.6:** Odometers of Smart PIG Robot [52].

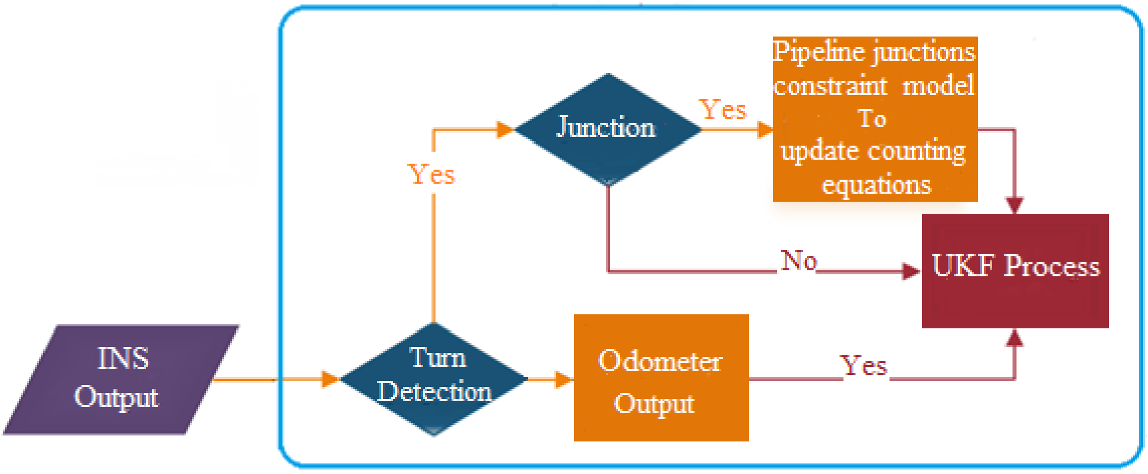


**Figure 4.7:** Scenario of the algorithm for corrected the estimation state of INS by odometer output and counting system.

### 4.5 The Counting System

The spikes of counting sensor can detect junctions between every two pieces. Every piece in pipeline has beginning and end detected junction lines with its previous and posterior pieces respectively respect to the direction of the motion. The distance between beginning and end detected for each piece represents its standard length, in this work it is ($20\ m$). Totally, these length measurements can be used to update the navigation states to reduce the errors of INS. Where, we can achieve the length of any piece in the sequence of the pipes in the pipeline: By integration, the multiplication of the velocity along the x-direction with the typical output of the counting system between two times relate to the start and end junction lines of one piece.

### 4.6 Implementation of UKF for PIG Robot

Let the process equation of the state vector with process noise is given as following:

$$x_k = f(x_{k-1}, \tilde{u}_{k-1}, w_{k-1}) \tag{4.15}$$

Where:

$x_k$ : is the navigation state vector at time $k$.

$\tilde{u}_{k-1}$ : is the accelerometer and gyroscope output vector.

$w_{k-1}$ : is the process noise vector with zero mean and covariance $R^w$ for IMU output.

Let suppose the output odometer to be given as:

$$y_k = h(x_k, \vartheta_k) = C_n^b v^n + \vartheta_k \tag{4.16}$$

Where:

$y_k$ : is the output vector of odometer output at time $k$.

$\vartheta_k$: is the measurement noise vector with zero mean and covariance $R^\vartheta$ for odometer output.

If there was counting system, there would be the following form:

$$z_k = g(y_k, \gamma_k) = \sqrt{((\varphi_0 - \varphi_f)R_e)^2 + ((\lambda_0 - \lambda_f)R_e)^2 + ((h_0 - h_f)R_e)^2} + \gamma_k \tag{4.17}$$

Where:

$z_k$ : is the output vector of counting system and always it will be $20m$.

$\varphi_0$, $\lambda_0$ and $h_0$: are latitude, longtitude and altitude coordenaits of the robot at the starting point of the pipeline.

$\varphi_f$, $\lambda_f$ and $h_f$: are latitude, longtitude and altitude coordenaits of the robot at the end point of the pipeline.

$R_e$: is the Earth’s radius at the starting point of the pipeline.

$\gamma_k$: is the measurement noise vector with zero mean and covariance $R^{\gamma}$for counting system.

Hence, ( $x_k \in R^n$), ($y_k \in R^m$) and ($z_k \in R^{\gamma}$). ($w_k$ ,$\vartheta_k$,$\gamma_k$) assumed to be Gausian weight noise.

The UKF can be implemented as following:

### I. Initialization of UKF

$$x_0 = [p_0^T \quad v_0^T \quad \Theta_0^T \quad 0 \quad 0]^T = E[x_0] = \bar{x}_0$$

$$P_0 = \begin{bmatrix} cov(p) & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & cov(v) & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & cov(\Theta) & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & cov(b_a) & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & cov(b_g) \end{bmatrix} = E[(x_0 - \bar{x}_0)(x_0 - \bar{x}_0)^T] \quad (4.18)$$

Hence, $E[\cdot]$ refers to expected value of $[\cdot]$.

### II. Generation the sigma points for navigation states

$$\mathcal{X}_{k-1} = [\bar{x}_{k-1} \quad \bar{x}_{k-1} + \gamma\sqrt{P_{k-1}} \quad \bar{x}_{k-1} - \gamma\sqrt{P_{k-1}}] \quad (4.19)$$

### III. Prediction phase

For prediction the sigma points, will be by using the process equation as given below:

$$x_{i,k} = f(\mathcal{X}_{i,k-1}^{x_{k-1}},\ \mathcal{X}_{i,k-1}^{\tilde{u}_{k-1}},\ \mathcal{X}_{i,k-1}^{w_{k-1}}) \quad (4.20)$$

Calculation the mean and covariance of INS predicted parameters:

$$\bar{x}_k = \sum_{i=0}^{2L} W_i^{(m)}\, x_{i,k} \quad (4.21)$$

$$\bar{P}_{x_k x_k} = \sum_{i=0}^{2L} W_i^{(c)} \left(x_{i,k} - \bar{x}_{i,k}\right)\left(x_{i,k} - \bar{x}_{i,k}\right)^T + R^w \quad (4.22)$$

Hence, the $\boldsymbol{W_i^{(m)}}$and $W_i^{(c)}$are mean and covariance weights that use to caculate the mean and covariance.

The $L$ in the above equations refers to the dimension of the expended state space, where it is equivalent to $L_x + L_w + L_\vartheta + L_\gamma$.

$L_x$: refers to the dimension of the original state $x_k$.

$L_w$, $L_\vartheta$ and $L_\gamma$ are the dimension of the IMU measurements noise $w_k$ ,$\vartheta_k$,$\gamma_k$ respectively.

$W_i$ is the weight calculated by Eq. (3.28).

Prediction of the sigma points of the odometer output is by applaying the obeservation equation:

$$y_{i,k} = h\left(\mathcal{X}_{i,k}^{x_k},\ \mathcal{X}_{i,k}^{\vartheta_k}\ \right) = h(\mathcal{X}_{i,k}^{x_k}, 0) \tag{4.23}$$

The mean of the predicted odometer output:

$$\bar{y}_k = \sum_{i=0}^{2L} W_i^{(m)}\, y_{i,k} \tag{4.24}$$

If a counting system output is available, the sigma points of predicted output will be:

$$z_{i,k} = g\left(\mathcal{X}_{i,k}^{y_k},\ \mathcal{X}_{i,k}^{\gamma_k}\right) \tag{4.25}$$

So, the predicted mean of the counting system output will be:

$$\bar{z}_k = \sum_{i=0}^{2L} W_i^{(m)}\, z_{i,k} \tag{4.26}$$

**IV. Measurement Update**:

The odometer residual covariance is:

$$P_{y_k y_k} = \sum_{i=0}^{2L} W_i^{(c)}\left(y_{i,k} - \bar{y}_{i,k}\right)\left(y_{i,k} - \bar{y}_{i,k}\right)^T + R^w \tag{4.27}$$

Prediction of the covariance of the correction between the residual odometer and states will update as:

$$P_{x_k y_k} = \sum_{i=0}^{2L} W_i^{(c)}\left(x_{i,k} - \bar{x}_{i,k}\right)\left(y_{i,k} - \bar{y}_{i,k}\right)^T \tag{4.28}$$

Prediction of the covariance of the counting system residual:

$$P_{z_k z_k} = \sum_{i=0}^{2L} W_i^{(c)}\left(z_{i,k} - \bar{z}_{i,k}\right)\left(z_{i,k} - \bar{z}_{i,k}\right)^T + R^\gamma \tag{4.29}$$

Prediction of the covariance of the correction between the residual counting system and states will update as:

$$P_{y_k z_k} = \sum_{i=0}^{2L} W_i^{(c)}\left(x_{i,k} - \bar{x}_{i,k}\right)\left(z_{i,k} - \bar{z}_{i,k}\right)^T \tag{4.30}$$

When the data of odometer is available;

Define the Kalman gain for odometer as:

$$K_k^{odo} = P_{x_k y_k} P_{y_k y_k}^{-1} \tag{4.31}$$

Update the states as:

$$\hat{x}_k = \bar{x}_{k-1} + K_k^{odo}(y_k - \bar{y}_k) \tag{4.32}$$

Update the covariance by measurment as:

$$P_k = \bar{P}_k - K_k^{odo} P_{x_k y_k}^T \tag{4.33}$$

When the data of counting system is availabile:

$$K_k^{count} = P_{x_k z_k} P_{z_k z_k}^{-1} \tag{4.34}$$

Update the states of navigation system by counting measurment as:

$$\hat{x}_k = \bar{x}_k + K_k^{count}(z_k - \bar{z}_k) \tag{4.35}$$

Update the covariance of states:

$$P_k = \bar{P}_k - K_k^{count} P_{x_k z_k}^T \tag{4.36}$$

$\bar{P}_k$: the update of $\bar{P}_{x_k x_k}$. The figure below shows briefly the steps of above system.

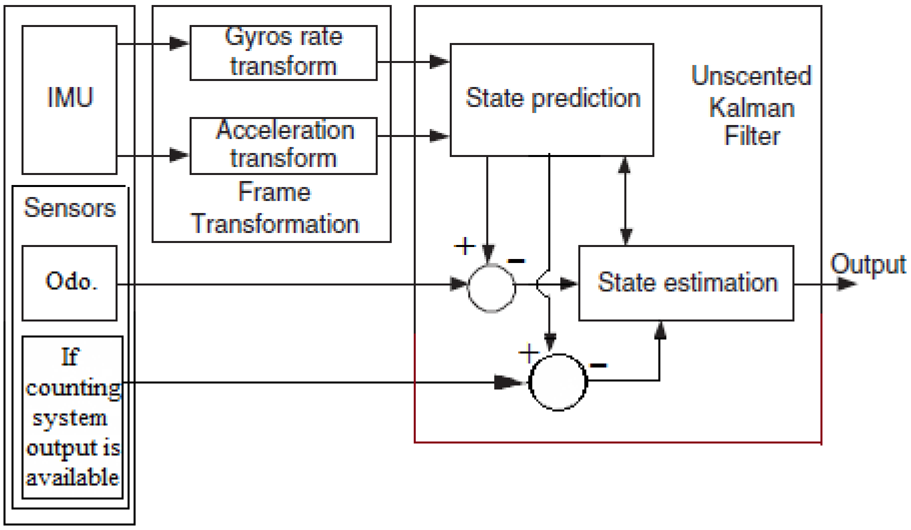


**Figure 4.8:** Diagram of system implementation.

# CHAPTER FIVE

## SIMULATION STUDY

### 5.1 Introduction

In this chapter, the simulation study of the integrated navigation system for PIG application will be introduced. At the first step, the simulations of robot and sensors' outputs are generated, and in the second step, the output of sensors is processed by UKF algorithm that introduced in the previous chapter. It should be mention that the inertial sensors that used for navigation are in the navigation grads, because the navigation of PIG with low cost sensors has large errors and the external sensors like odometers and counting sensor isn't able to stabilize the errors within the navigation system. We know the INS is a stand-alone system, and can calculate the navigation parameters which are position, velocity, and attitude, but the errors of such a system are unstable.

Odometers are able to measure the velocity of system in body frame and counting system is able to count the number of pipes, so the error of position will be unstable. The integrated system just can reduce the rate of errors and in long time, the error will be unstable.

### 5.2 Sensors parameters

Today we know the inertial sensors (gyroscopes and accelerometers) have many type of errors like fixed bias, nonlinearity, misalignments, gain errors and noise errors. In this study, we will suppose that IMU is calibrated system, and has only biases (due to drift from calibration condition or random bias) and white noises. The aiding sensors have only measurement white noise. In addition, the out of counting sensor is a pulse and do not has any data, actually here the time of passing from the welding points is important. The parameters of sensors are represented in Table 5.1.

**Table 5.1:** Parameters of sensors.

| raw | name | value | unit |
|---|---|---|---|
| 1 | Gyroscopes unknown bias | 0.0083, -0.0062, -0.0041 | Deg/hour |
| 2 | Std of gyroscopes' noise | 0.0103, 0.0103, 0.0103 | Deg/hour |
| 3 | accelerometers unknown bias | -0.0003, 0.0002,0.0004 | m/s^2 |
| 4 | Std of accelerometers' noise | 0.0002, 0.0002, 0.0002 | m/s^2 |
| 5 | Std of odometer's noise | 0.001 | m/sec |
| 6 | Std of pip's length | 0.3 | m |

In Table 5.1 vector parameters' data are represented in a line are for x, y and z directions respectively.

### 5.3 Nominal Trajectory

For generating of nominal states and generating of nominal values of sensors, we used an attitude control system and a velocity control system. The starting point is the coordinate of the Karbala city. By sending, some command to the control system simulated path is generated. In Table 5.2, the initial condition of simulation sates (nominal values) are shown.

**Table 5.2:** Nominal values of parameters.

| raw | name | value | unit |
|---|---|---|---|
| 1 | Initial latitude (latitude of Karbala) | 32.3233 | Deg |
| 2 | Initial longitude (longitude of Karbala) | 44.0594 | Deg/hour |
| 3 | Initial altitude (altitude of Karbala) | 32 | m |
| 4 | Angular velocity of PIG | 6.8697, -10.6989, -.0378 | deg/hour |
| 5 | Attitude (Roll, Pitch, Yaw) | 0.0, 0.0, 57.2958 | deg |
| 6 | Specific force | 0.0, 0.0, 9.8004 | m/sec^2 |

The command to the velocity and attitude control system are:

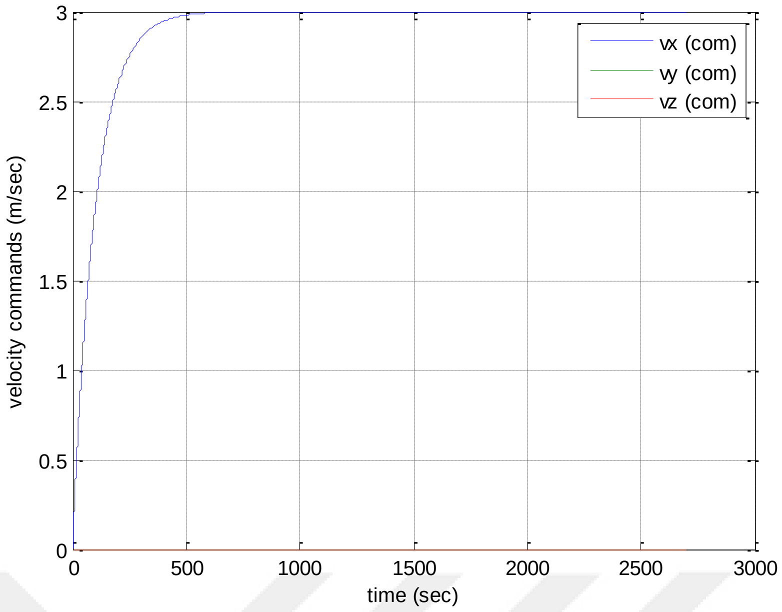


**Figure 5.1:** Command for velocity control system (in body coordinate).

As it shown in Fig. 5.1, command velocity along the x direction is 3 m/sec, and along the y and z direction is zero, so the velocity of robot will increase and finally will receive to the 3 m/sec.

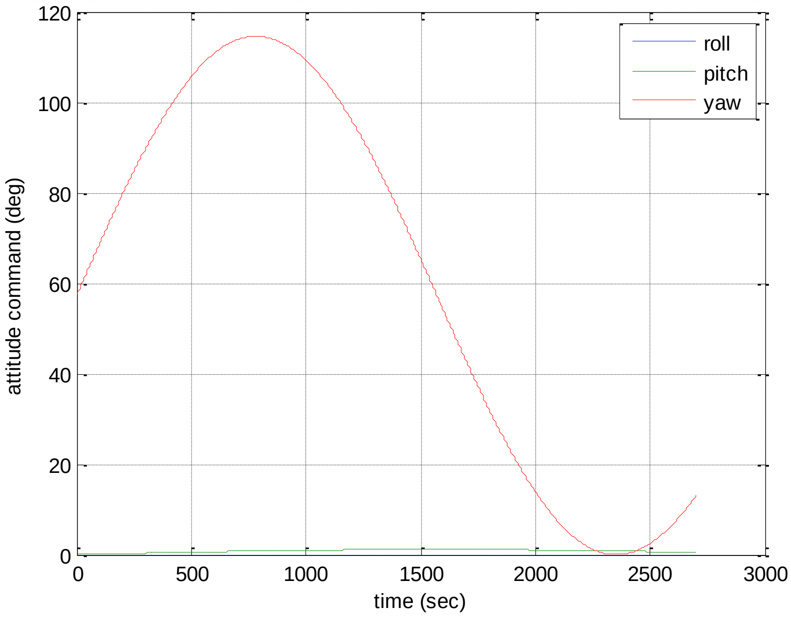


**Figure 5.2:** Command for attitude control system.

The commands to the attitude control system are shown in Fig. 5.2. As we can see, the heading and pitch angles are a sine signal, but the roll command is zero signal. After simulation of dynamic equations for PIG robot, the nominal parameters are generated.

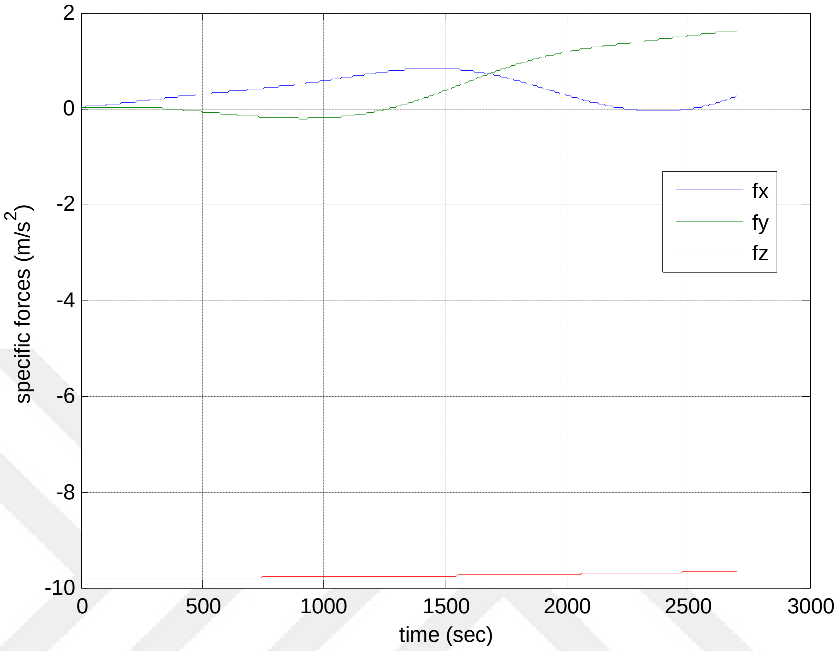


**Figure 5.3:** Specific forces along x, y and z direction.

In Fig. 5.3, ideal outputs of accelerometers are shown. After some time, the velocity of system will be constant, so the acceleration of system will be zero, and output of accelerometers will be projection of $-g$ along the body frame. Consequently, the specific forces along the x and y directions are due to orientation of PIG (roll and pitch angles), and along the z direction is near the $-|g|$.

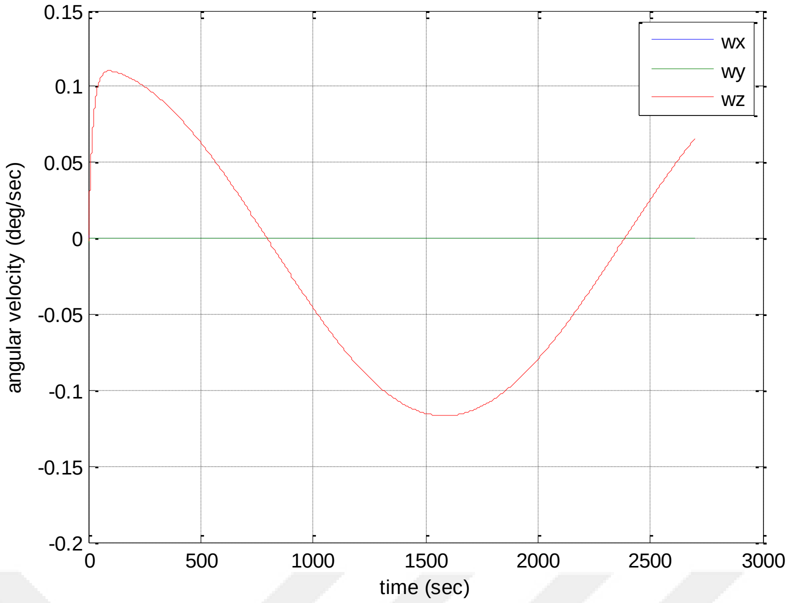


**Figure 5.4:** Angular velocity of robot relative to inertial coordinate system.

The ideal output of rate gyros are shown in Fig. 5.4. The $\omega_z$ is like Sine signal and angular velocity along the x and y directions are zero.

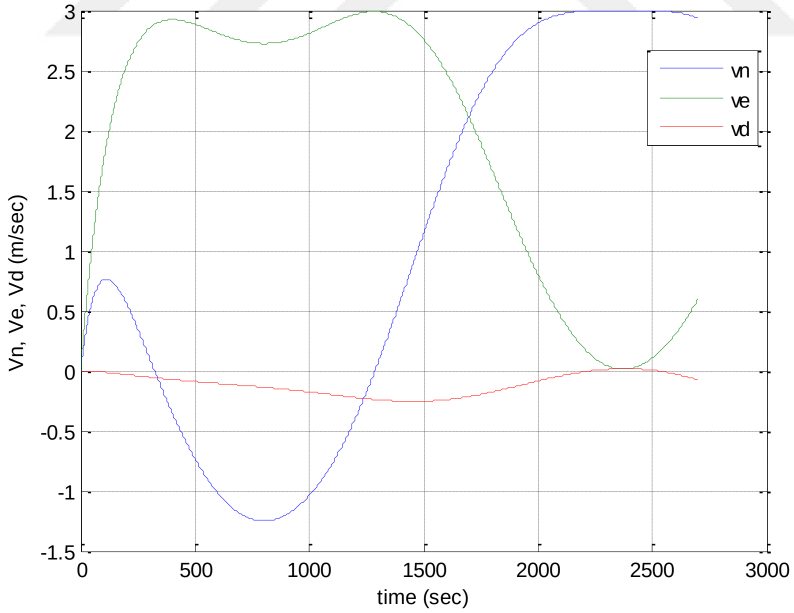


**Figure 5.5:** Velocity along the north, east and down direction.

In Fig. 5.5, velocity of robot relative to Earth that projected in navigation frame are shown. The projection of velocity of robot relative to Earth the projected in the body frame are shown in the Fig. 5.6.

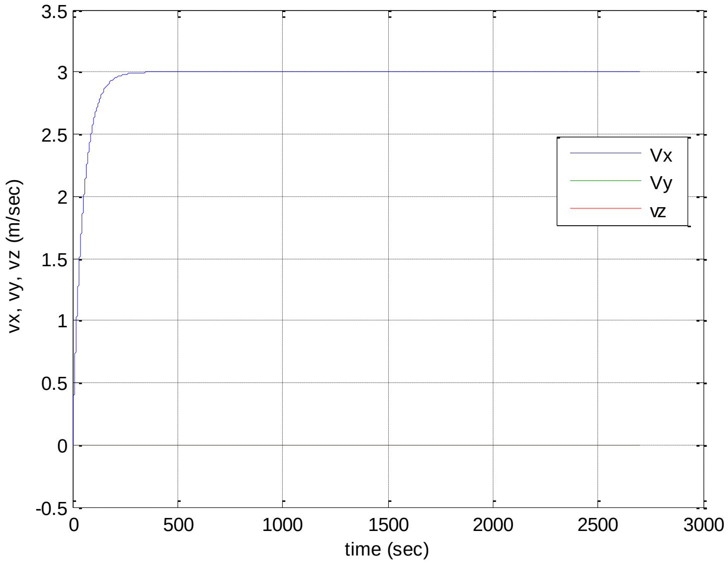


**Figure 5.6:** Projection of velocity in the body frame.

These velocities are near the command velocities, because the control system tries to keep the system's velocity near the command signals that represented in Fig. 5.1.

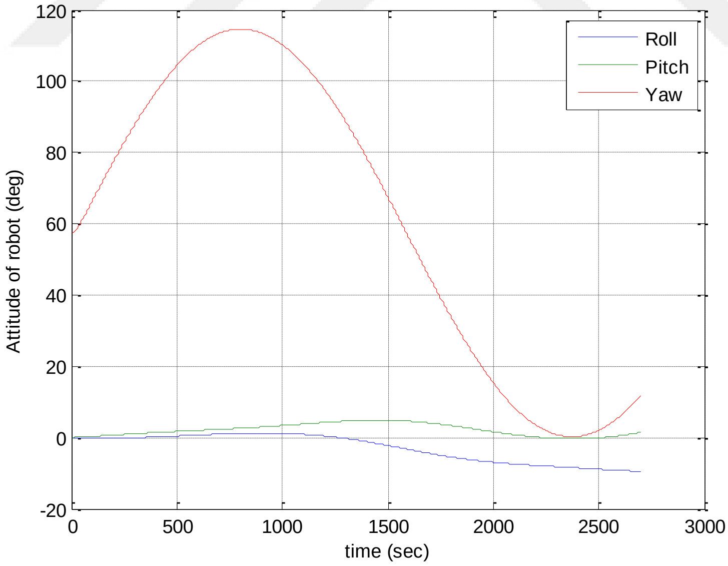


**Figure 5.7:** Attitude of PIG.

The attitude of PIG is represented in Fig. 5.7, these attitude are very close to command attitude that represented in Fig. 5.2, because the control system tries to keep the attitude near the desired attitude.

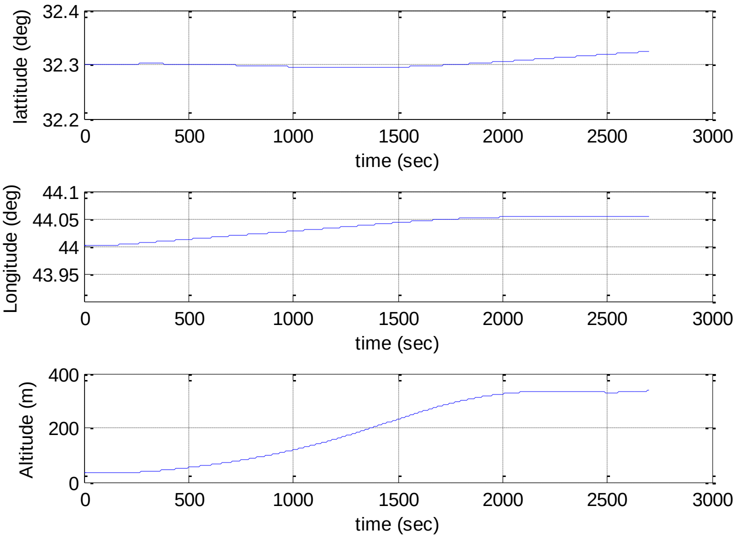


**Figure 5.8:** Position of PIG.

The latitude, longitude, and altitude of robot are shown in the Fig. 5.8. These coordinates are for the Karbala city.

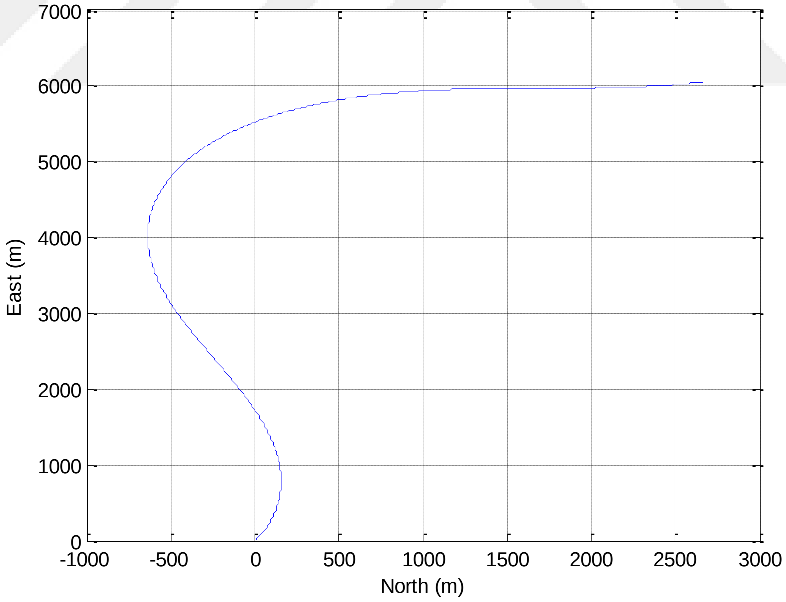


**Figure 5.9:** Trajectory of PIG.

Fig. 5.9 shows the trajectory of PIG. As it can extracted from the figure, the robot has a journey along the north and east less than 3 km and 7 km inside of 45 min.

## 5.4 Sensors Output

The output of ideal accelerometers are shown in the Fig. 5.3, by adding the bias and white noise to that signals, out of accelerometers will be simulated. The errors of accelerometers are shown in the Fig. 5.10.

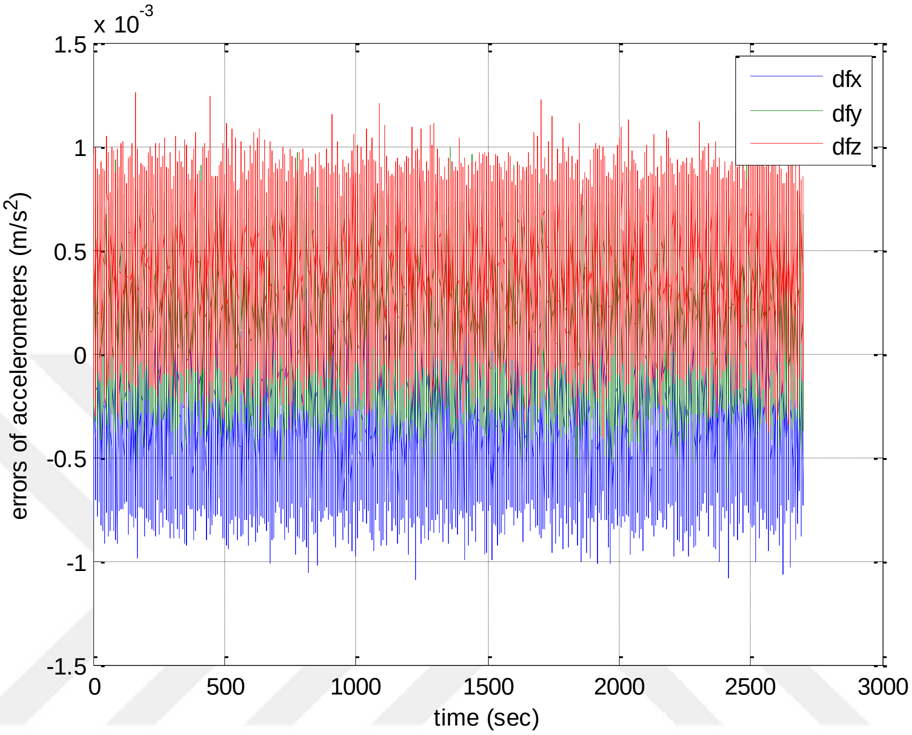


**Figure 5.10:** Accelerometers’ errors.

The ideal output of gyroscopes are shown in the Fig. 5.10, by adding bias and white noise error to these signals the output of gyros are simulated. The errors of gyroscopes are shown in the Fig. 5.11.

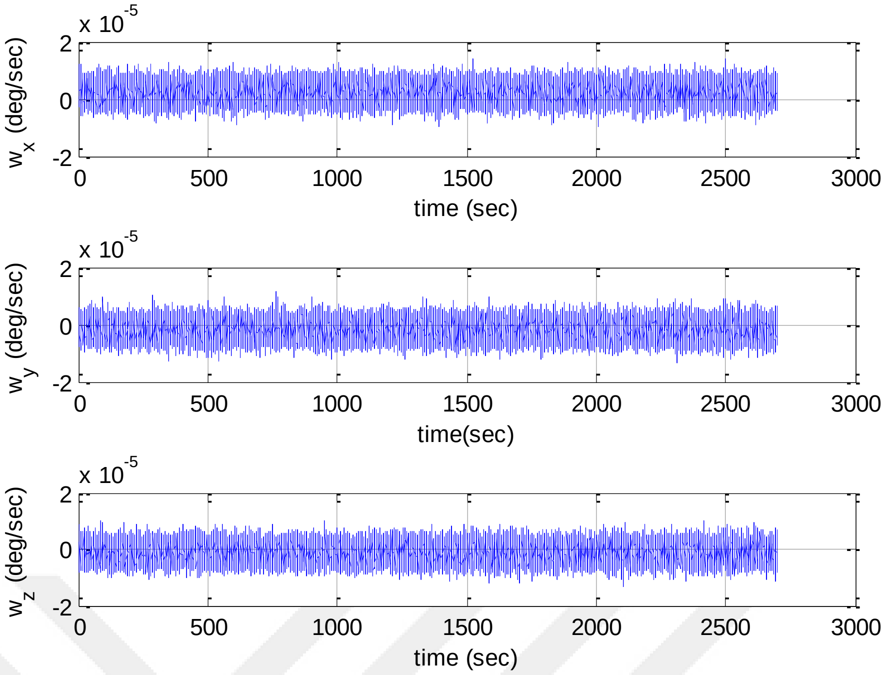


**Figure 5.11:** Gyroscopes' errors.

The integrated navigation system has two aided sensors. One of them is odometer that output of odometers are illustrated in the Fig. 5.12.

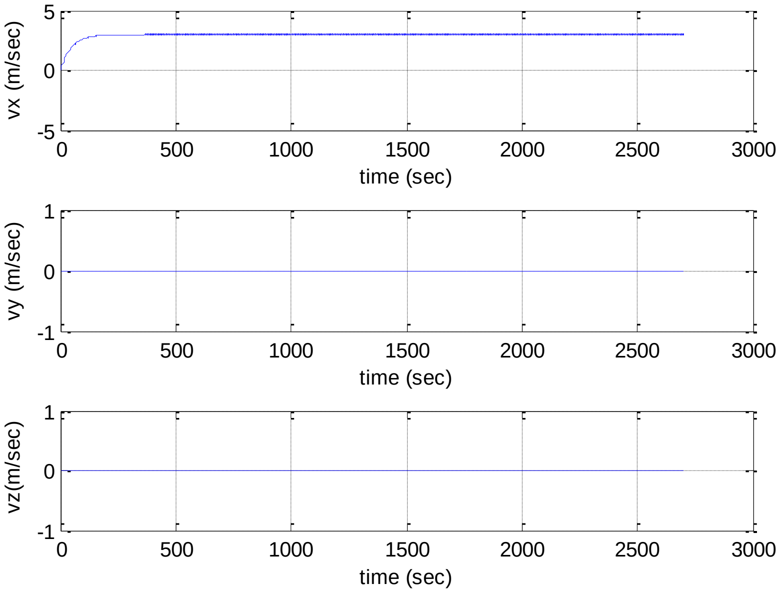


**Figure 5.12:** Output of odometers.

As we can see, the velocity along the y and z directions are just noise of measurement, and along the x direction after 3 minutes are constant.

Simulating of the counting system is more complex than the other sensors. In this simulation when the time is going ahead the program calculated the length of pipe from starting point when the length is greater than the 20 m, a noise with zero mean and 0.3 m standard deviation is added to output of sensor and will used in the integrated navigation algorithm.

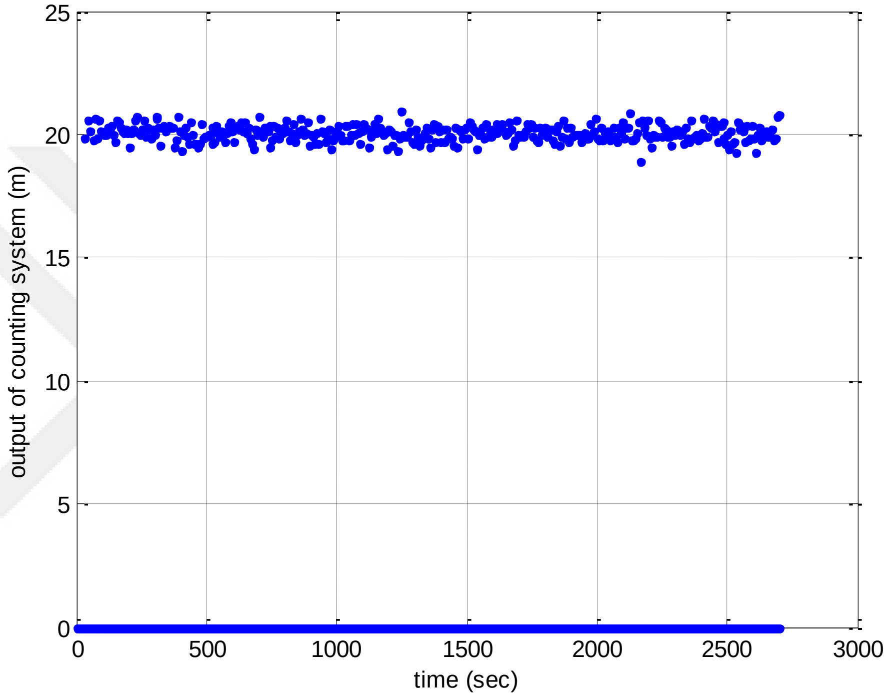


**Figure 5.13:** Counting system output (the length of pipes).

As seen in Fig 5.13, the nominal value of pipes are 20 meters, but some time that is lower and some time that is greater than the nominal value. In addition, the frequency of available data from counting system is different. When the velocity is low, the frequency of data is low and when the velocity is high, the frequency will be high.

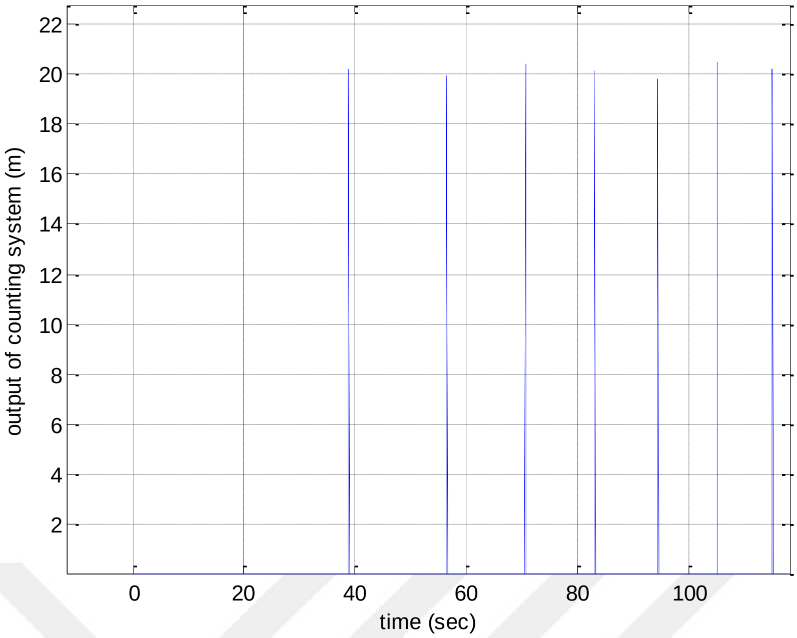


**Figure 5.14** Counting system output at the first 120 seconds.

### 5.5 Inertial navigation parameters errors and standard deviations

The initial condition of navigation system has some errors; these errors are classified in Table 5.3.

**Table 5.3:** Initial value (or initial error) of navigation system.

| raw | name | Value | unit |
|---|---|---|---|
| 1 | Initial latitude error | 0.5 | m |
| 2 | Initial longitude error | -0.5 | m |
| 3 | Initial altitude error | 0.3 | m |
| 4 | Errors of initial velocity | 0.001, -0.002, -0.003 | m/sec |
| 5 | Initial attitude (Roll, Pitch, Yaw) | 0.01, -0.02., -0.03 | deg |
| 6 | Accelerometer estimated bias | 0.0, 0.0, 0.0 | m/sec^2 |
| 7 | Gyroscope estimated bias | 0.0, 0.0, 0.0 | deg/sec |

The standard deviation of navigation system states at initial time are illustrated in table 5.4.

**Table 5.4:** Initial standard deviation of navigation systems states.

| raw | name | value | unit |
|---|---|---|---|
| 1 | Std of initial latitude | 1 | m |
| 2 | Std of initial longitude | 1 | m |
| 3 | Std of initial altitude | 1 | m |
| 4 | Std of initial velocity | 0.003, 0.003, 0.003 | m/sec |
| 5 | Std of initial attitude (Roll, Pitch, Yaw) | 0.05, 0.05, 0.05 | deg |
| 6 | Std of accelerometer estimated bias | 0.5e-3, 0.5e-3, 0.5e-3 | m/sec^2 |
| 7 | Std of Gyroscope estimated bias | 0.01, 0.01, 0.01 | deg/hour |

### 5.6 Navigated parameters errors

In this section, the navigation parameters' errors and its 3-sigma will represent. The error of accelerometers' bias for x, y and z directions are represented in figures from (5.15) up to (5.17) respectively.

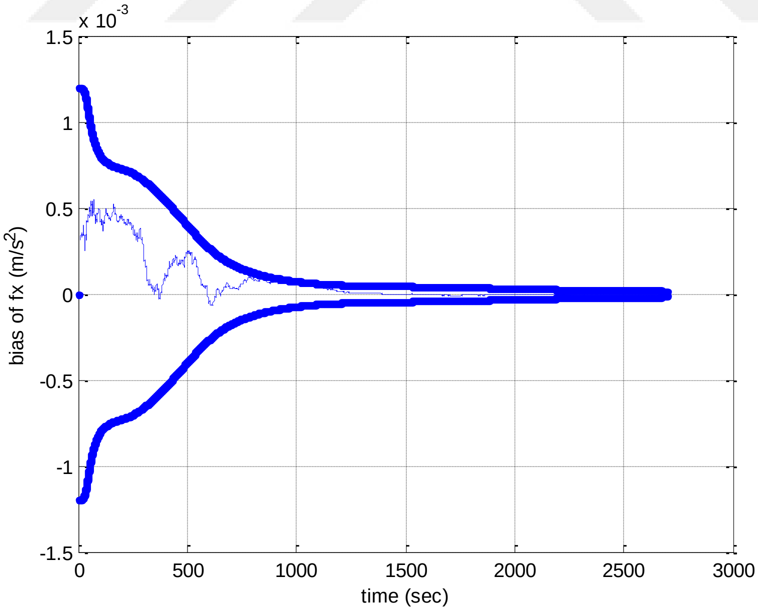


**Figure 5.15:** Error of x-accelerometer bias and its three-sigma bounds.

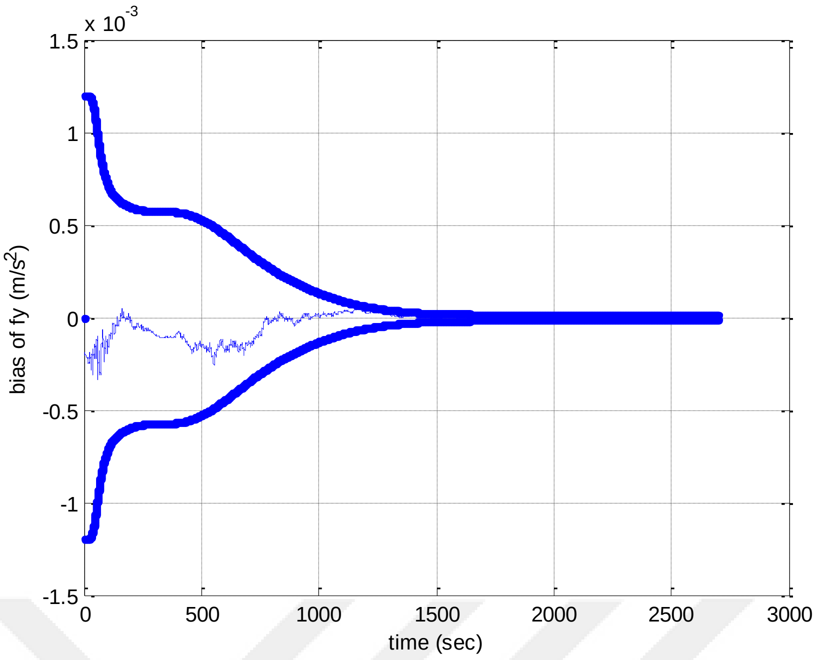


**Figure 5.16:** Error of y-accelerometer bias and its three-sigma bounds.

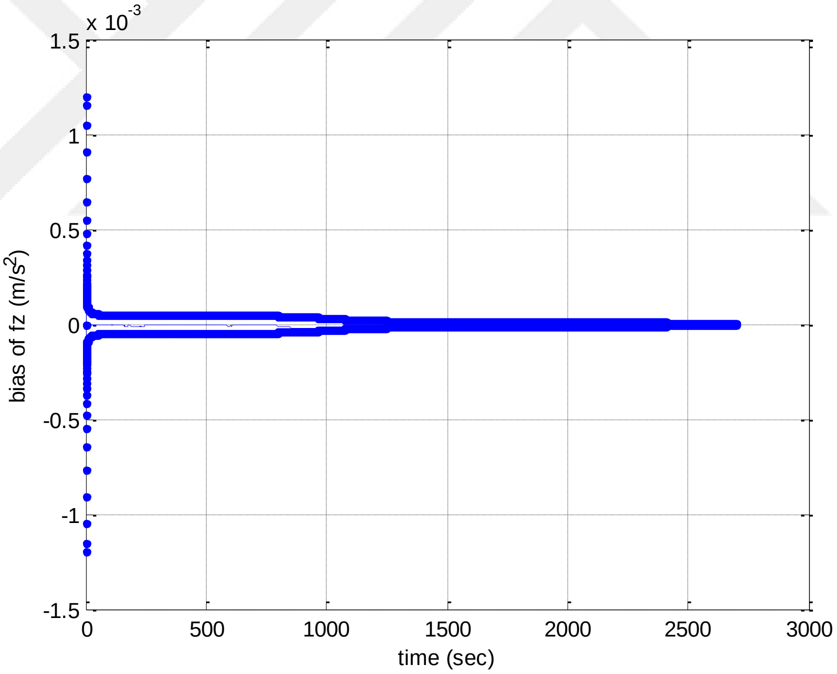


**Figure 5.17:** Error of z-accelerometer bias and its three-sigma bounds.

As we can see in the above figures, all of estimated biases are in three-sigma bound and have small errors. We can see after around 1000 seconds all of the standard deviations will rich to steady state condition. The error of estimated gyroscopes’ biases are shown in the figures from (5.18) to (5.20).

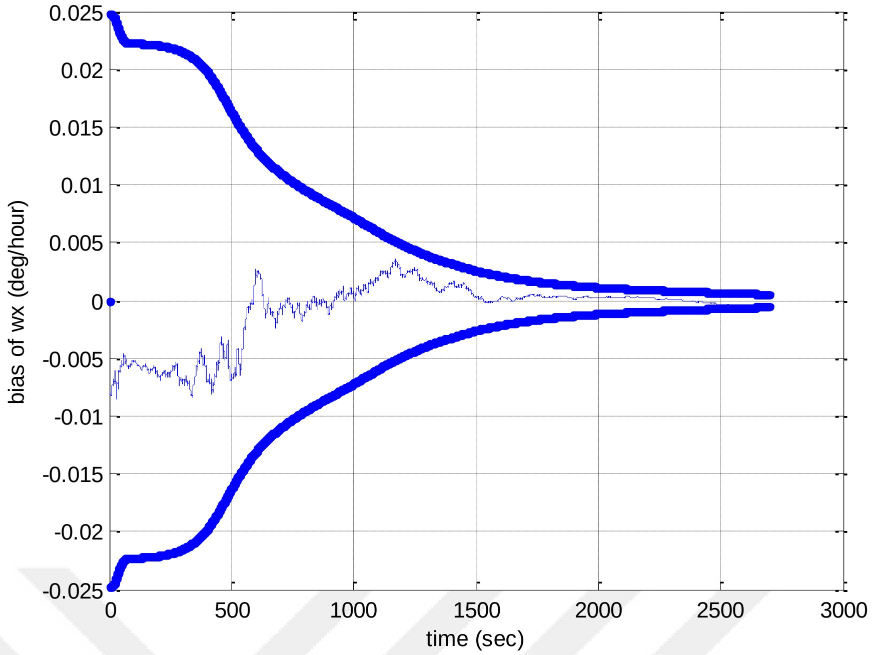


**Figure 5.18:** Error of x-gyroscope bias and its three-sigma bounds

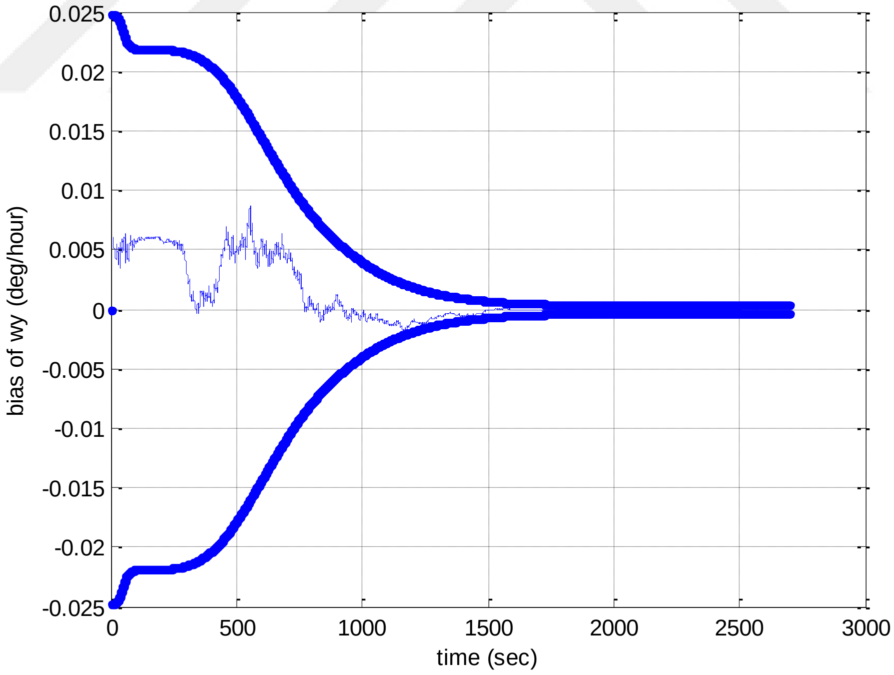


**Figure 5.19:** Error of y-gyroscope bias and its three-sigma bounds.

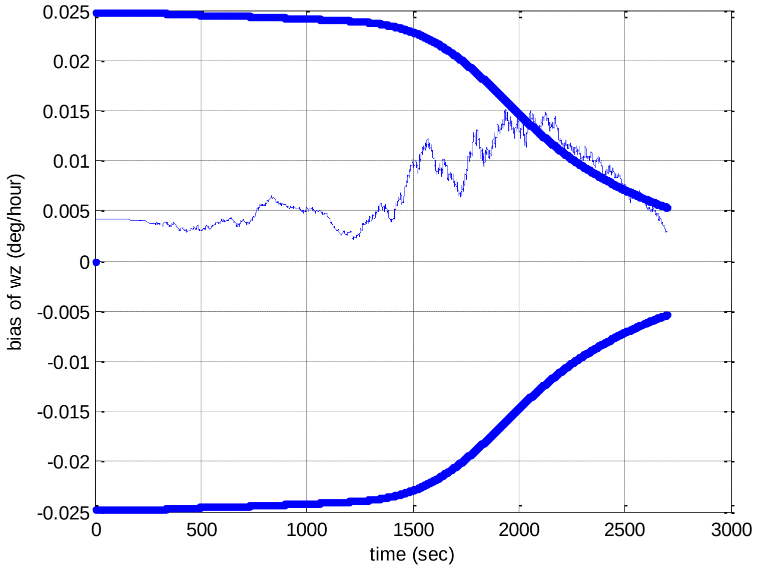


**Figure 5.20:** Error of z-gyroscope bias and its three-sigma bounds.

As we can see from the above figures, the bias of gyroscopes are remained in their bound, but the bias of z-gyroscope has large errors. Because the observability of this parameter come from the observability of heading, again the lack of good external senor is fleeing the errors of yaw angle. The attitude errors are shown in figures from (5.21) up to (5.23).

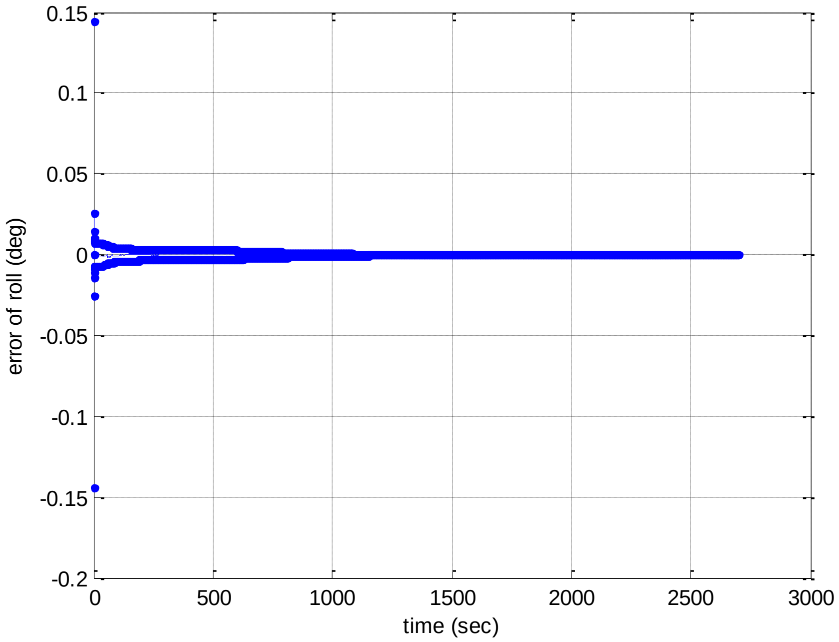


**Figure 5.21:** Error of roll and its three-sigma bounds.

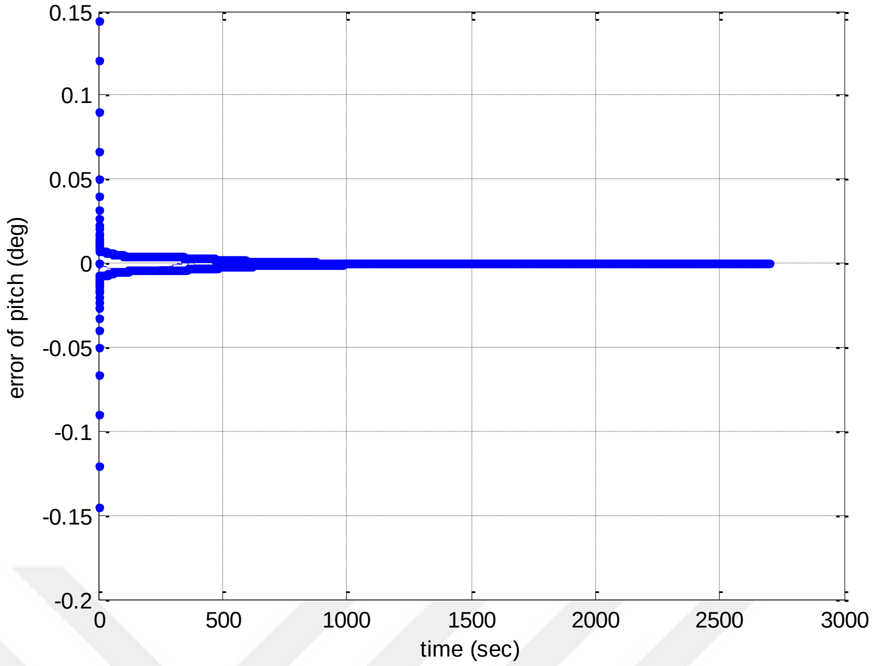


**Figure 5.22:** Error of pitch and its three-sigma bounds.

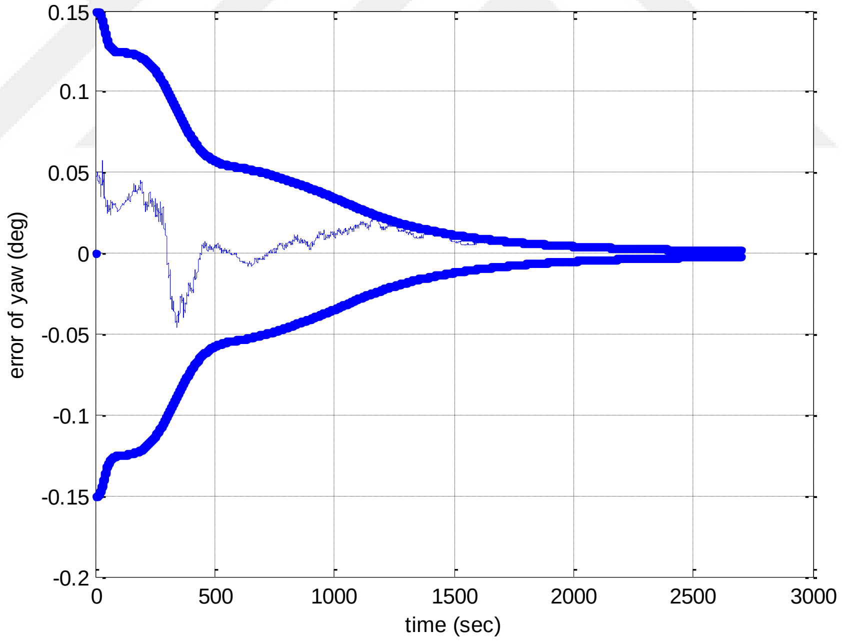


**Figure 5.23:** Error of yaw and its three-sigma bounds.

As we can see the errors of roll and pith are near the zero and its bounds, but the heading's error is more than roll and pitch angles' errors, because the heading is not affected from the counting system. The errors of velocity in navigation frame are represented in figures from (5.24) to (5.26).

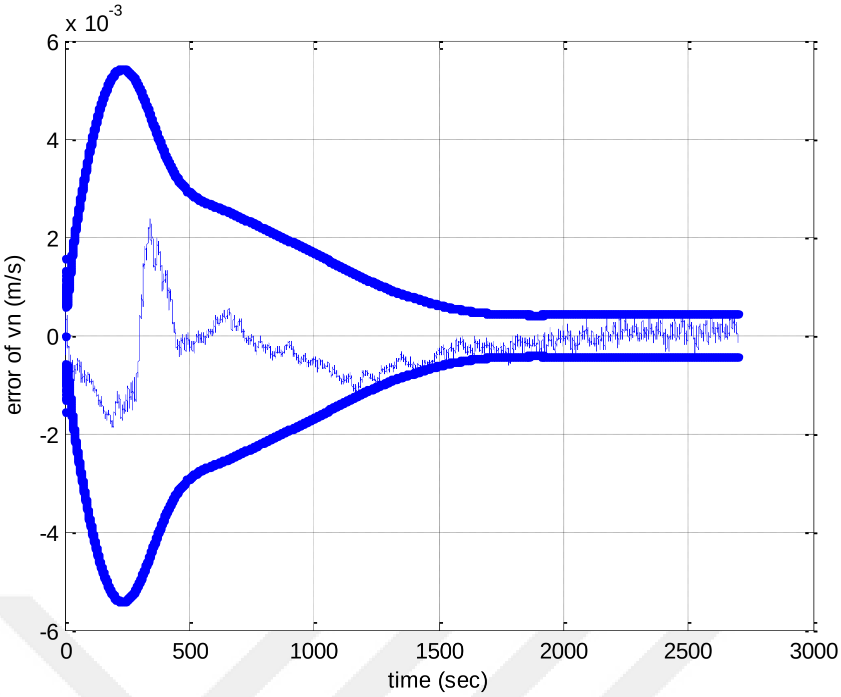


**Figure 5.24:** Error of the north velocity and its three-sigma bounds.

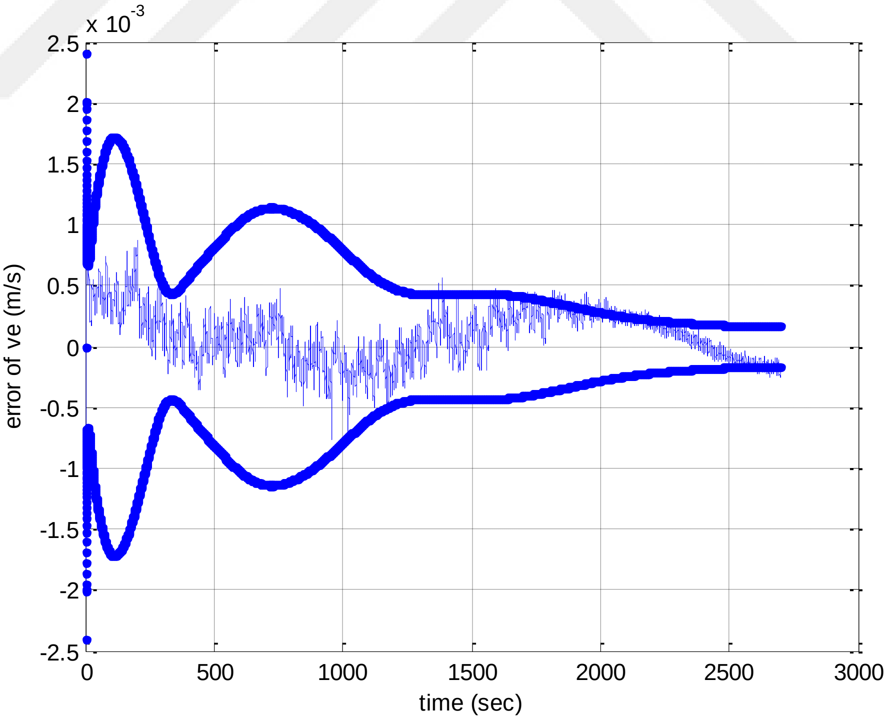


**Figure 5.25:** Error of east velocity and its three-sigma bounds.

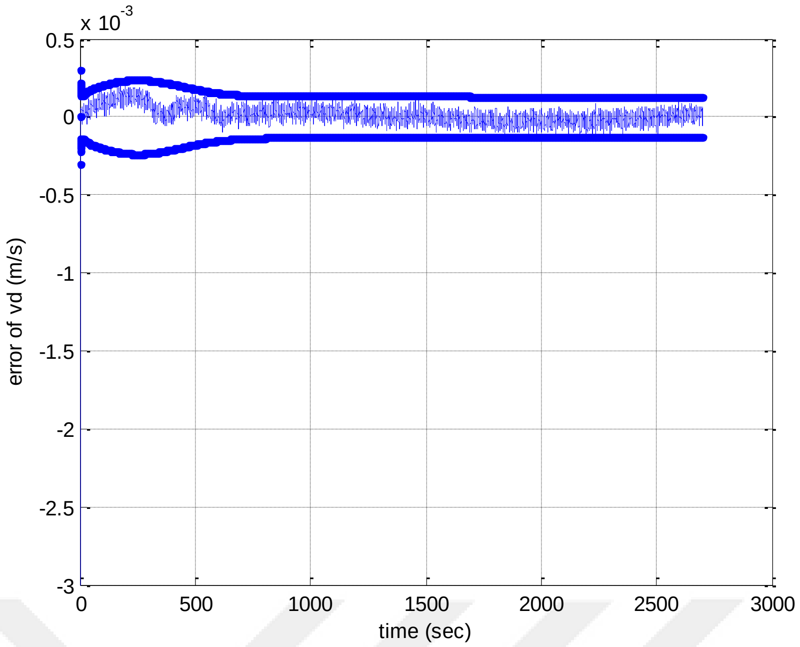


**Figure 5.26:** Error of the down velocity and its three-sigma bounds.

As we can see from the above three figures, the velocity error along the down is lower than velocities' errors along the north and east directions.

The position error of the navigation system are represented in figures from (5.27) to (5.29).

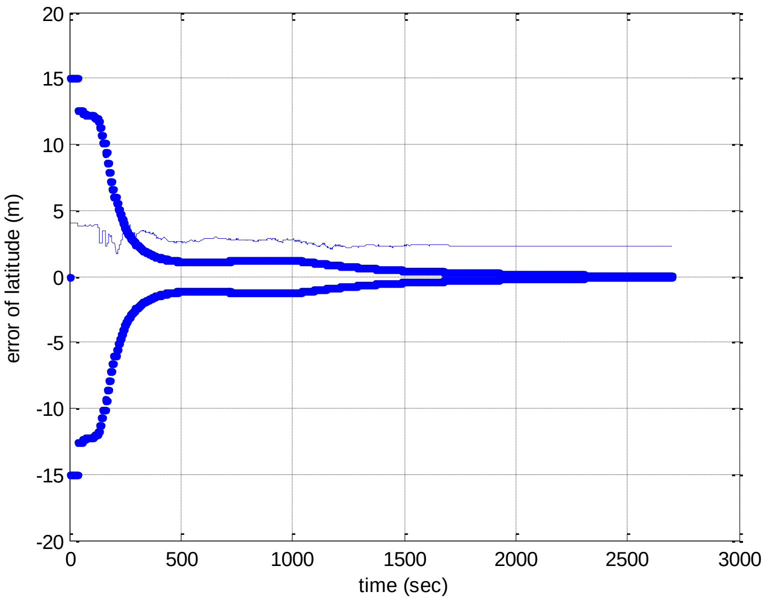


**Figure 5.27:** Error of latitude and its three-sigma bounds.

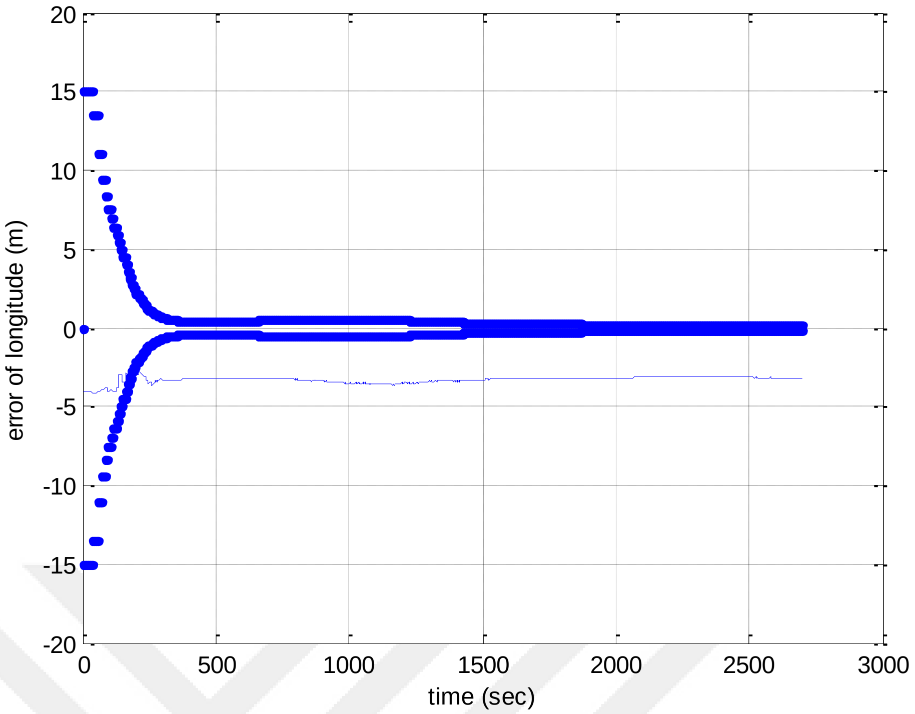


Figure 5.28: Error of the longitude and its three-sigma bounds.

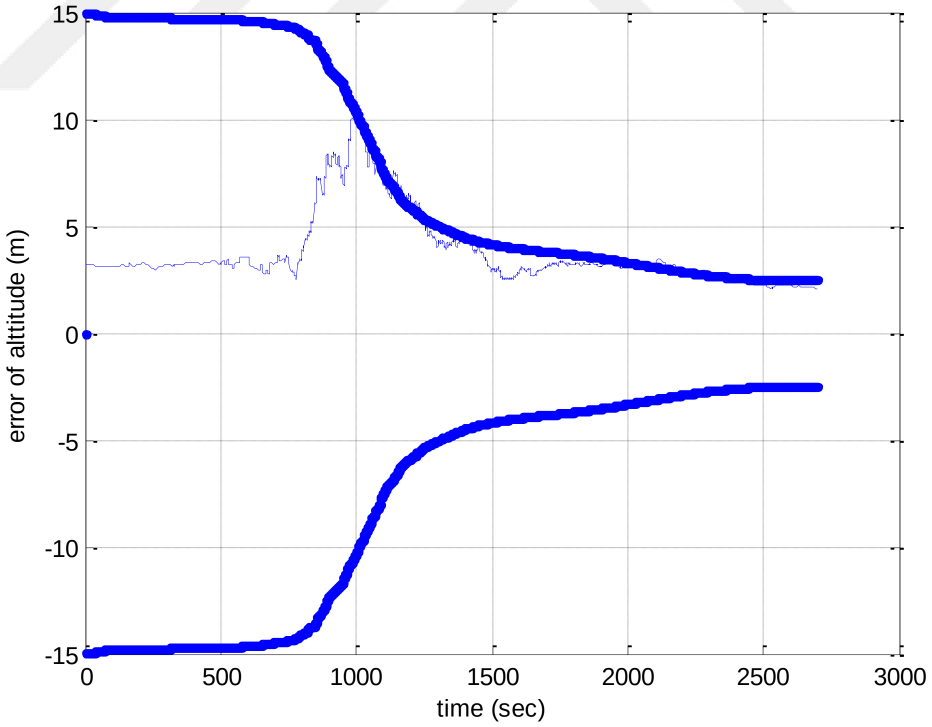


Figure 5.29: Error of the altitude and its three-sigma bounds.

As we can see the error along the north and east directions error are lower than the 2.5 m and 3.5 m respectively, but altitude error is very small, less than the 2.5 m. we can say the total error of navigation system in 45 minutes is close to length of one pipe and it is god result.

### 5.7 Errors of INS/Odometer

Here, there is a simulation in the absence of counting system. When the integrated navigation system does not uses the counting system, the system does not has any sense of length or position so the position will be completely unobservable. The figures (5.30) to (5.31) are represents the accelerometers' biases and its three sigma bounds.

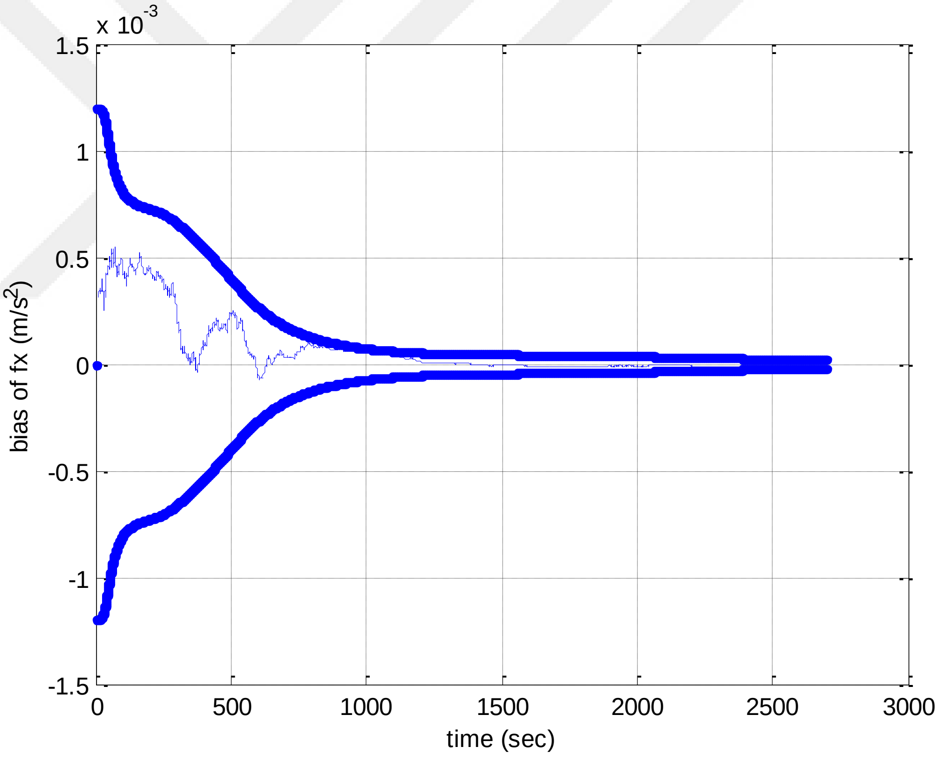


**Figure 5.30:** Error of the x-accelerometer's bias and its three-sigma bounds.

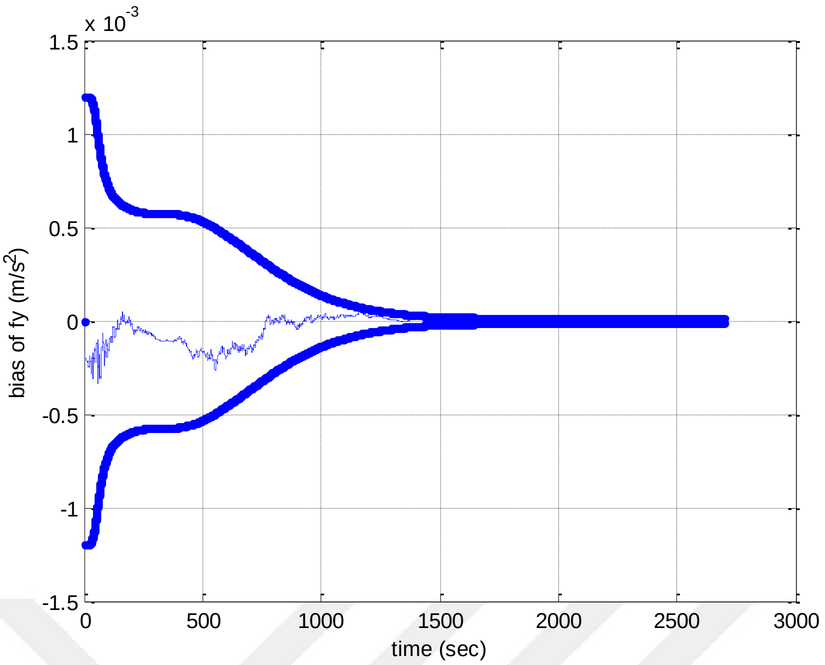


**Figure 5.31:** Error of the y-accelerometer's bias and its three-sigma bounds.

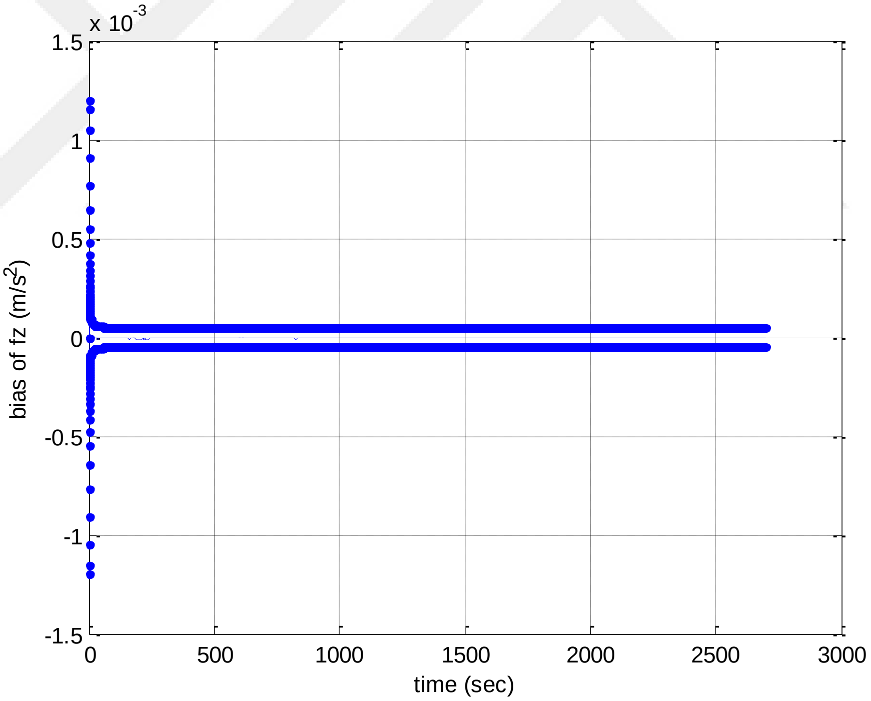


**Figure 5.32:** Error of the z-accelerometer's bias and its three-sigma bounds.

We can see the all accelerometers' biases are observable and its three sigma bound. Actually, the observability of accelerometers' biases come from the velocity measurements. Therefore, when the system has odometers these parameters will be observable.

The figures (5.33) to (5.35) shows the performance of filter for estimation of gyroscope's biases.

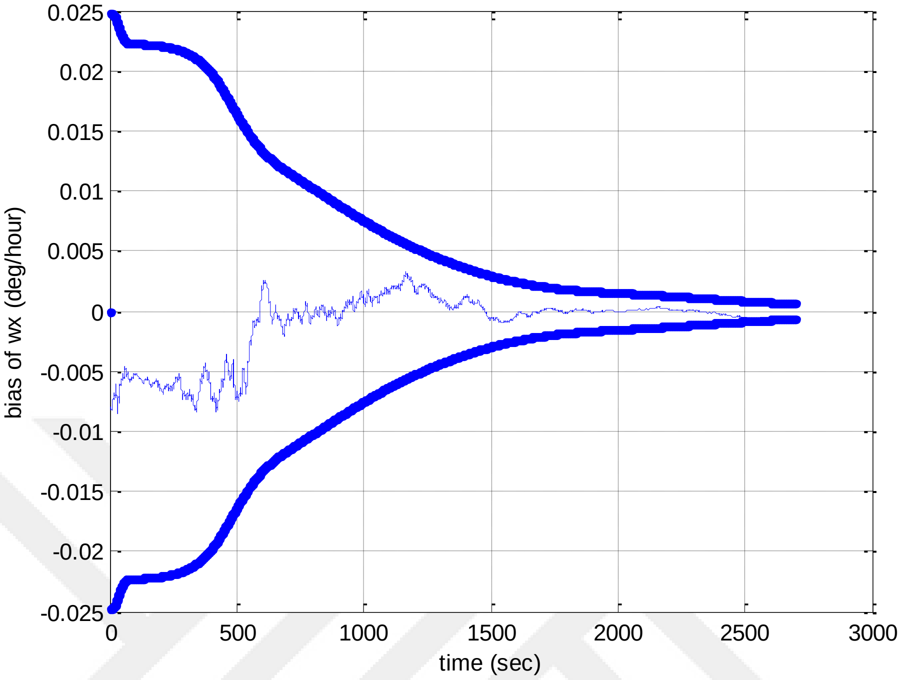


**Figure 5.33:** Error of the x-gyroscope's bias and its three-sigma bounds.

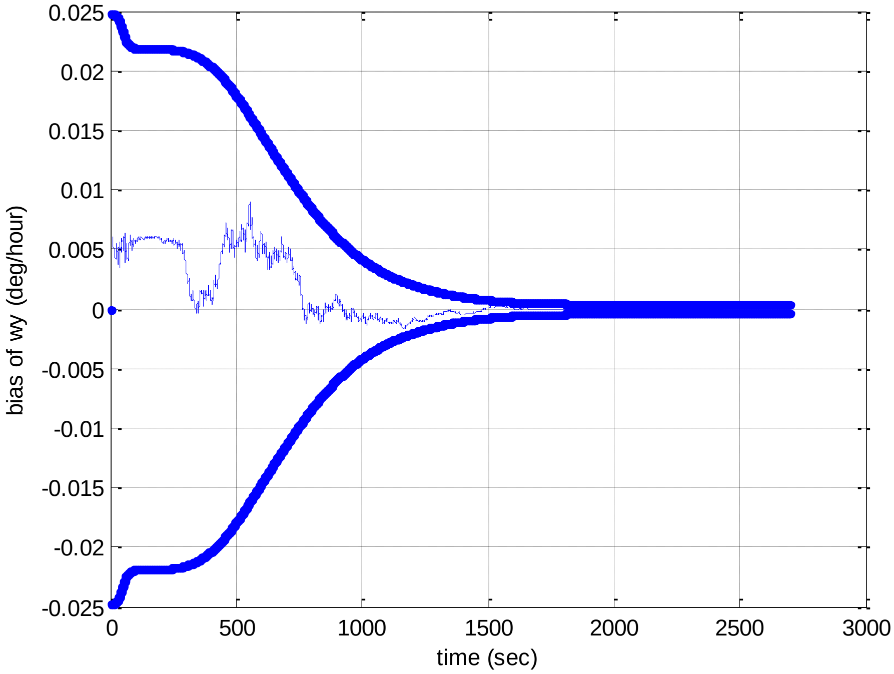


**Figure 5.34:** Error of the y-gyroscope's bias and its three-sigma bounds.

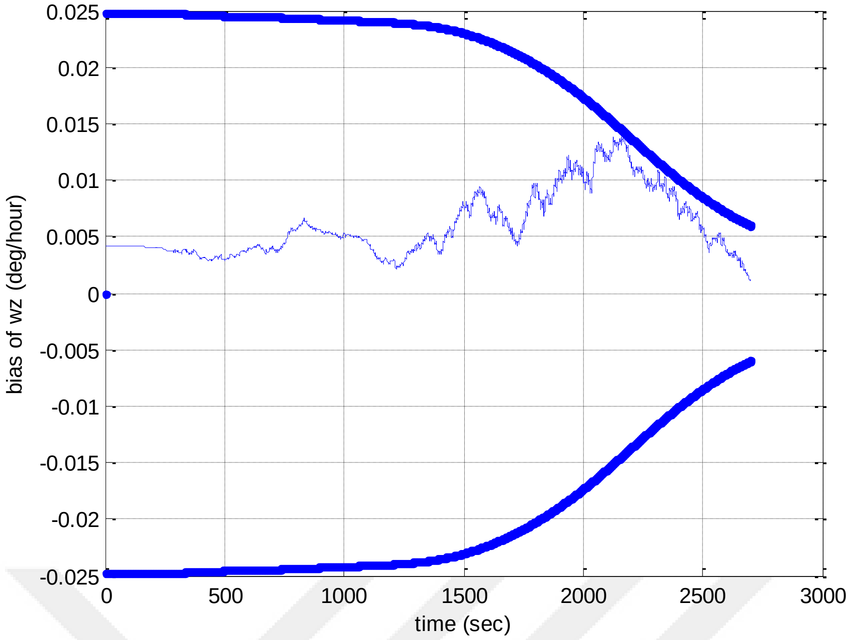


**Figure 5.35:** Error of the z-gyroscope's bias and its three-sigma bounds.

It is shown that the bias of x-axis and y-axis have low errors and z gyro has large error.

Figures (5.36) to (5.38) are show the attitude errors of integrated system.

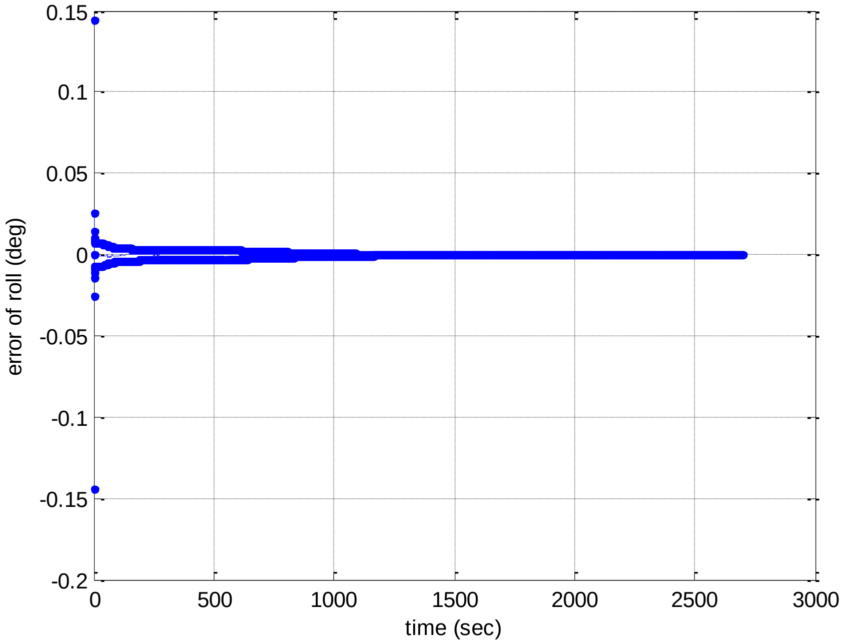


**Figure 5.36:** Error of the roll and its three-sigma bounds.

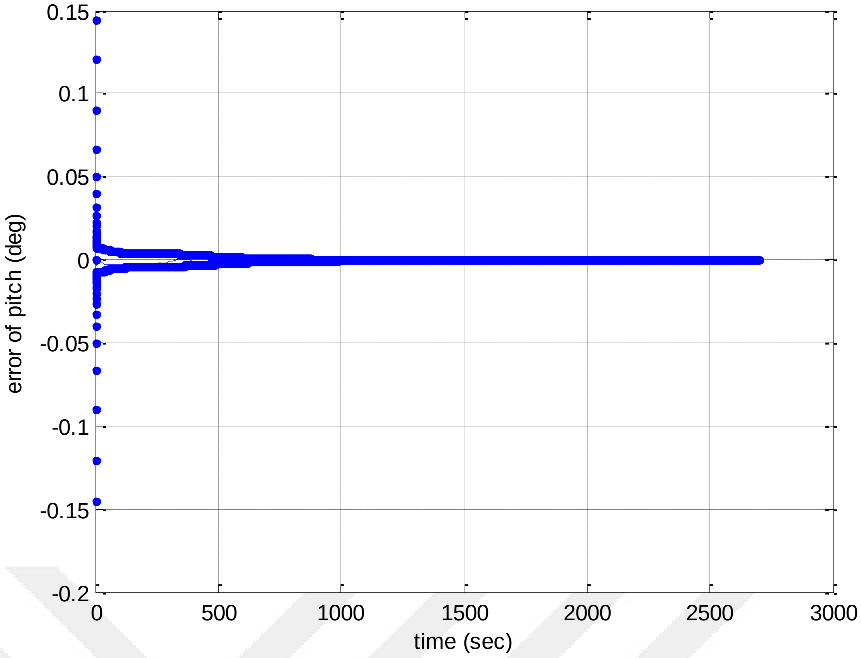


**Figure 5.37:** Error of the pitch and its three-sigma bounds.

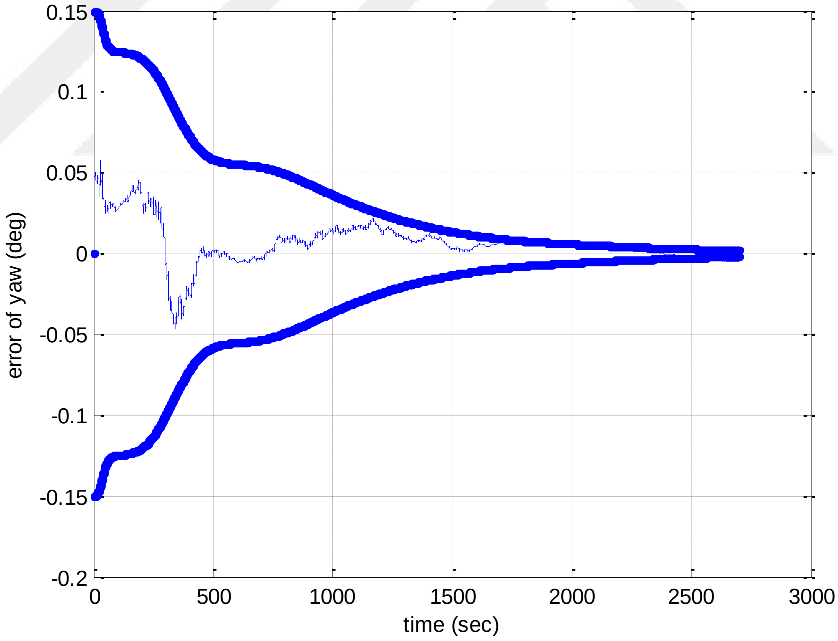


**Figure 5.38:** Error of the yaw (heading) and its three-sigma bounds.

It is shown that error of roll and pitch immediately reduce, but the error of heading is reduced very slowly.

The figures (5.39) to (5.41) represent the errors of velocity in navigation frame.

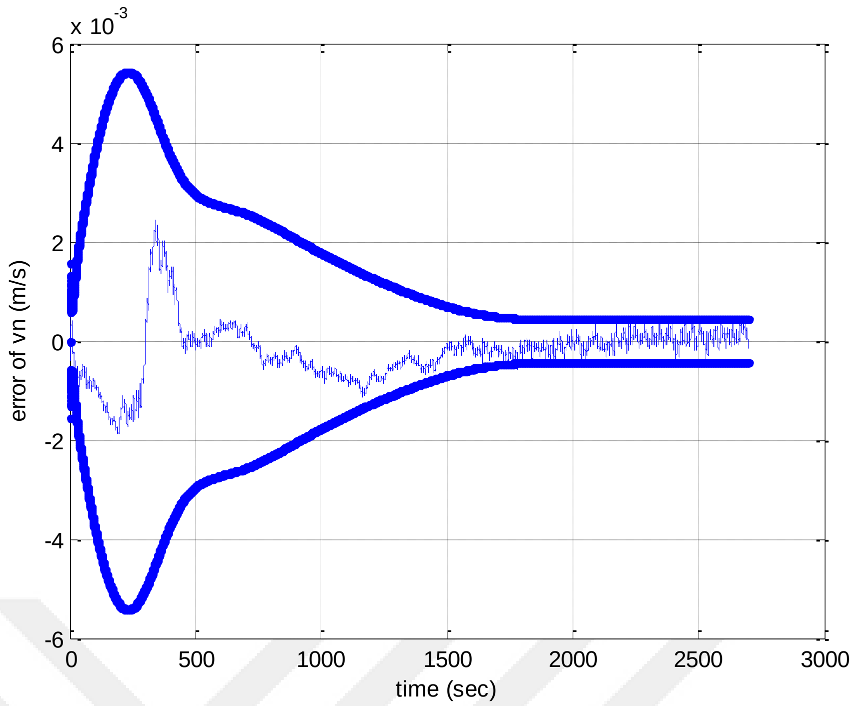


**Figure 5.39:** Error of the north velocity and its three-sigma bounds.

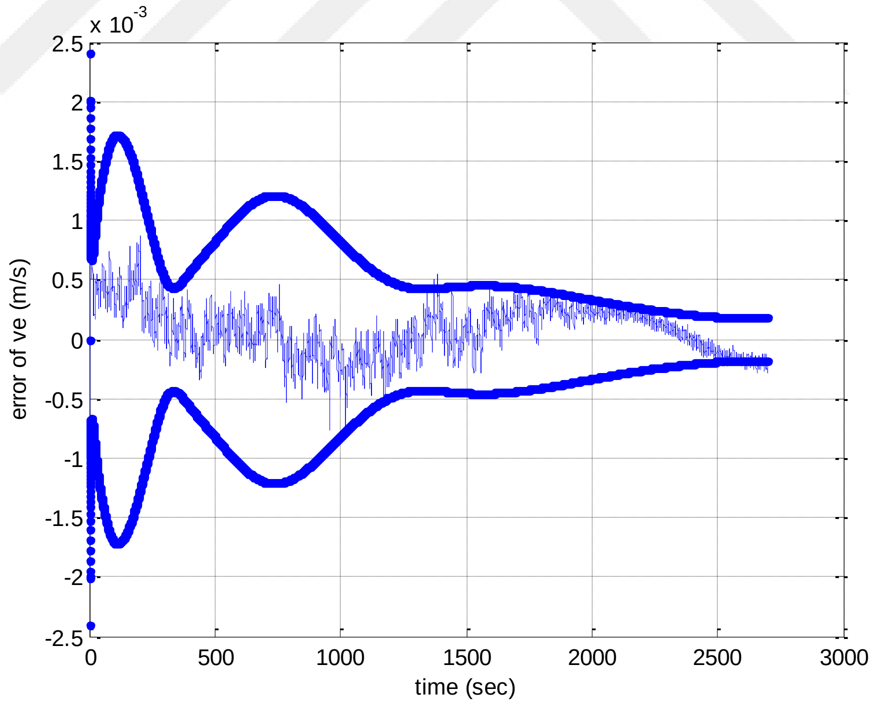


**Figure 5.40:** Error of the east velocity and its three-sigma bounds.

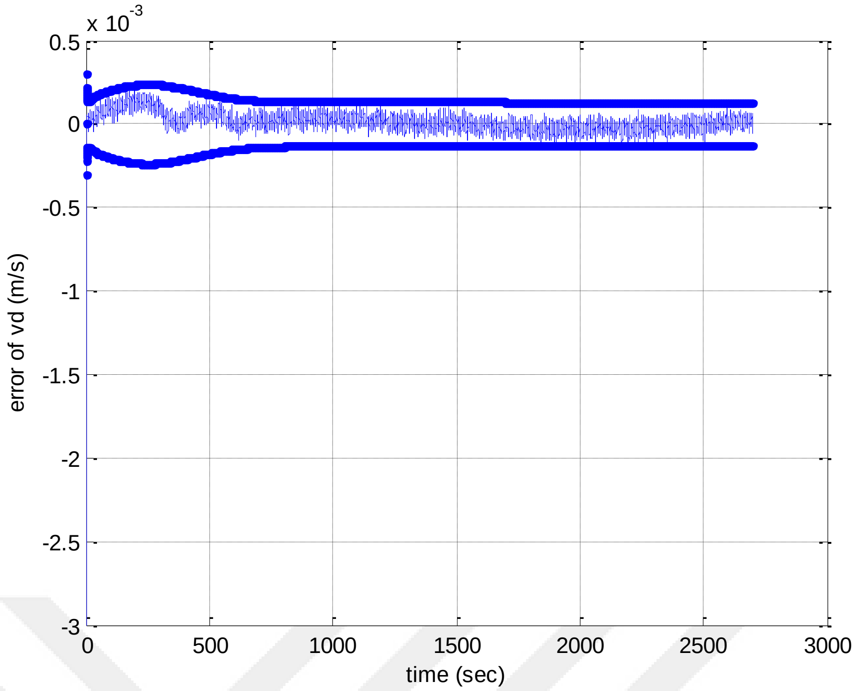


**Figure 5.41:** Error of the down velocity and its three-sigma bounds.

It is shown that the errors of velocity in navigation has bounded error and these errors are in 3-sigma bounds. Figures (5.42) to (5.44) represents the position errors.

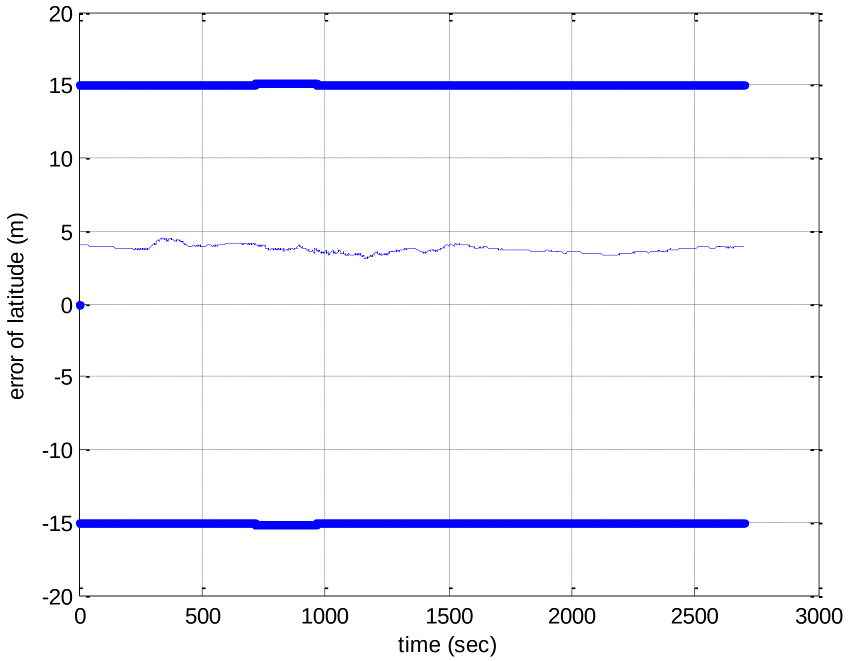


**Figure 5.42:** Error of the latitude and its three-sigma bounds.

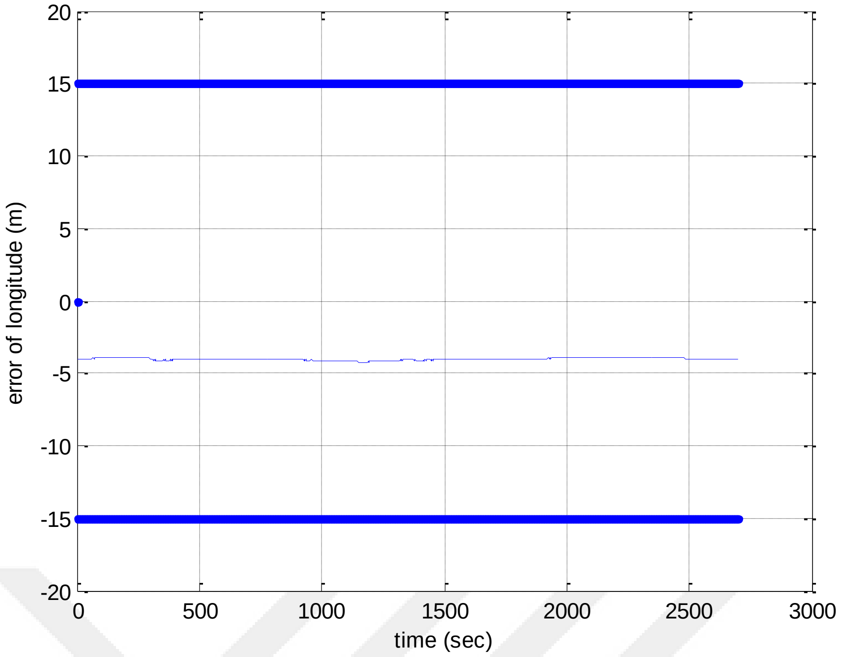


**Figure 5.43:** Error of the longitude and its three-sigma bounds.

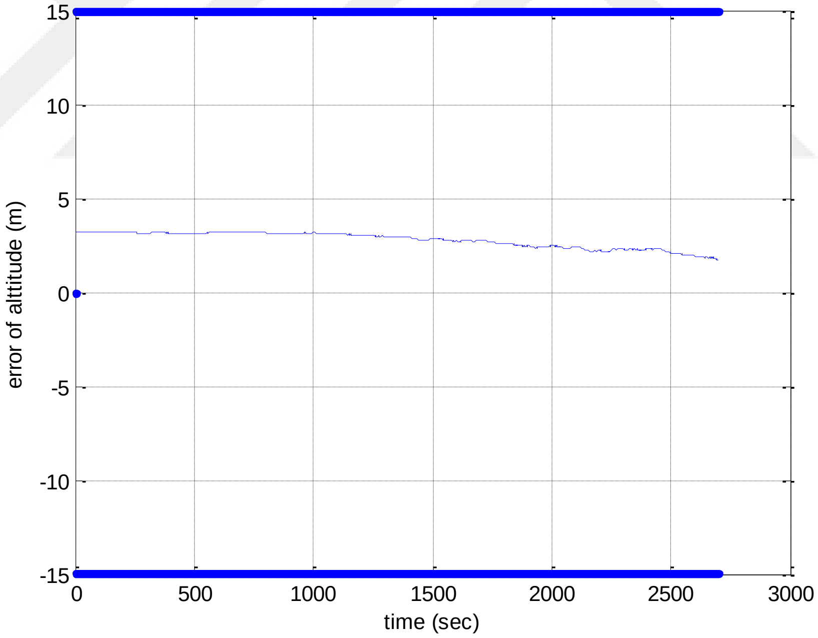


**Figure 5.44:** Error of the altitude and its three-sigma bounds.

It is clear that the lack of position sensor is the first reason of position errors. It is shown that the system cannot compensate the errors of position. Therefore, the error of initial position will be increase.

## 5.8 Errors of inertial navigation system

When the navigation system uses only the data of IMU, errors of inertial navigation system will be very large, because the system don't has any aiding system, so the all of errors will be increase. In the below errors of attitude are shown in the Fig. 5.45.

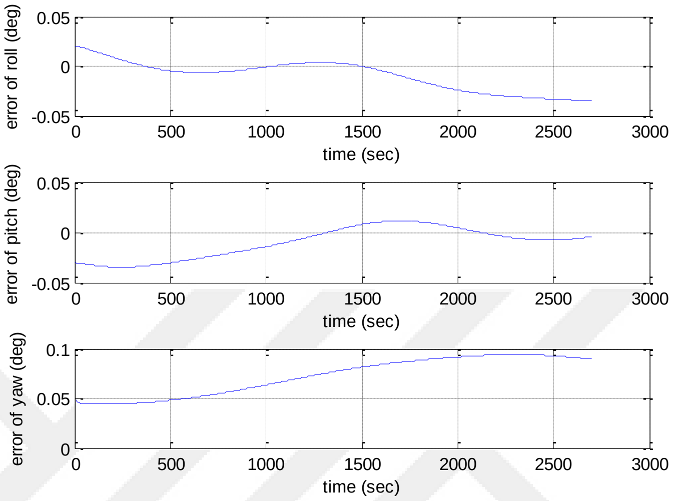


**Figure 5.45:** Error of the attitude (roll, pitch, and yaw).

We can see that the errors of roll, pitch, and yaw are completely oscillating unstable. If the time approach to infinitive, the errors of attitude will approach to infinitive. The velocity error is shown in the below figure.

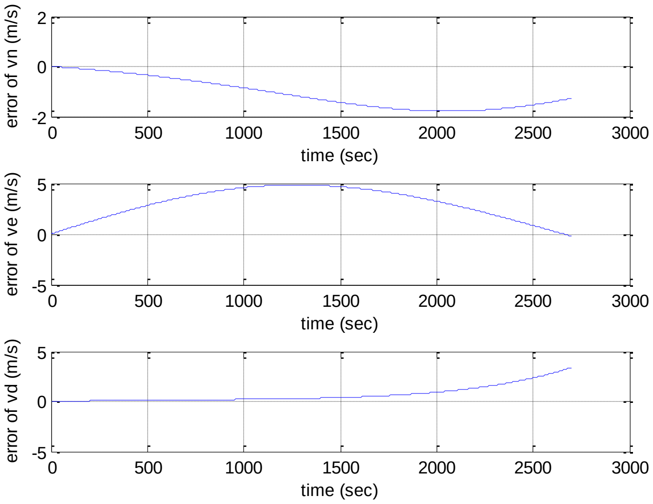


**Figure 5.46:** Error of the velocity (north, east, and down velocities).

Again, the errors of velocity is oscillating unstable. Fig. 5.47 represents the error of position.

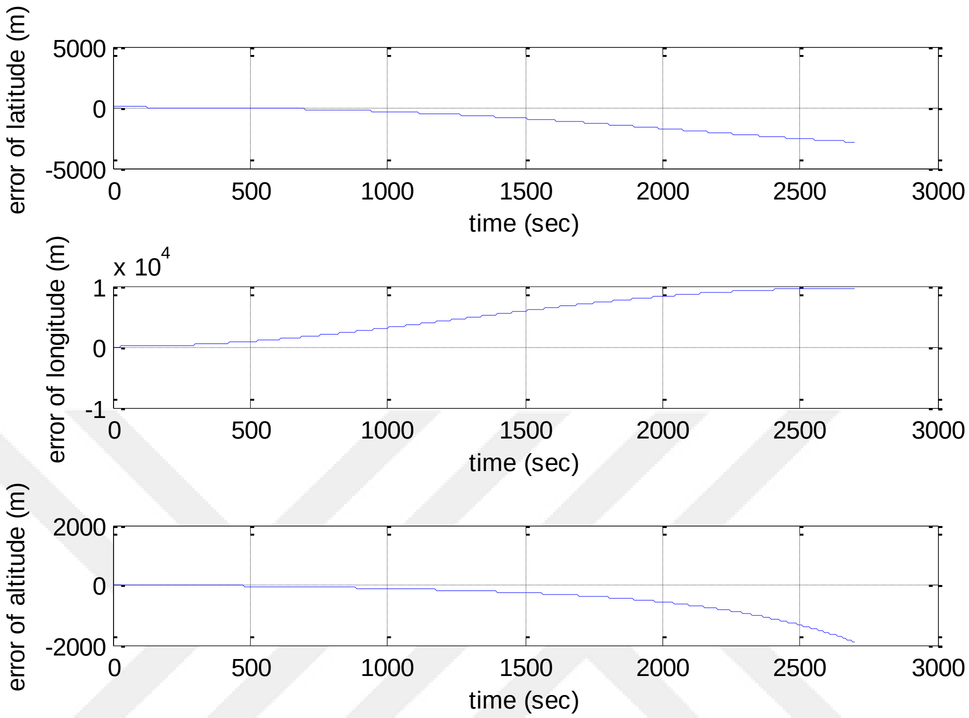


**Figure 5.47:** Error of the latitude, longitude, and altitude.

The position error plot shows the INS has near the 3, 10, and 2 km errors along the north, east and down directions respectively. These errors are large errors and for this application actually it in not acceptable.

Table 5.5 represents the comparison of position error for three type of systems. The first system consider the integration of INS/Odometer and counting system, the second system consider the integration of INS and odometer and third system consider only the INS system. The errors of these system are in the below table.

**Table 5.5:** Summery of position errors.

| raw | Name (maximum error) | INS/Odometer/Counting System (m) | INS/Odometer (m) | INS (km) |
|---|---|---|---|---|
| 1 | Latitude | 2.5 | 4 | 3 |
| 2 | Longitude | 3.5 | 4 | 10 |
| 3 | Altitude | 2 | 1.8 | 1.8 |
| 4 | Mean square error | 4.7 | 5.9 | 10.59 |

We can see the role of the counting system in the reduction of position errors.

# CHAPTER SIX

## CONCLUSION AND RECOMMENDATIONS

In this chapter, a brief review of the thesis and contributions and some recommendation will discuss.

### 6.1 Brief review of thesis

In chapter one, there was an introduction to in-pipeline robot navigation. The current problem with smart PIG robot is the navigation errors that can be stabilized by several types of aiding sensors to improve the errors of navigation system, which have been discussed in the next chapters. In addition, chapter two included introduction for the navigation frames, the errors of inertial sensor with their mechanization and with their mathematical models and the explanation for the calibration of inertial sensor and measurement error compensation. The filtering explanation was included in chapter three. Error models of INS with UKF have been offered in chapter four and the stochastic processes included UT model. Thus, the entire implementation algorithm of UKF for pipeline navigation has been shown in chapter four. Controlling/stabilizing the deviation in the pipeline navigation has been illustrated in a comparing between INS/odometer and INS/odometer/counting system in chapter five.

### 6.2 The research contributions

This research includes two contributions:

- Improving the navigation errors for in-pipeline smart inspection robot by defining a new algorithm based on integration of pipeline constraints as odometer measurements along y and z direction, odometer, and counting system. With absence of these constraints and sensors as shown in Table 5.5, the errors of INS are unstable and, so the navigation parameters have large errors. In chapter four, proposed algorithm is presented and discussed. The efficiency of algorithm are shown in chapter five.
- In the industry of pipeline, there are several methods for detection the pipeline junctions such as, electromagnetic acoustic transducers, magnetic

flux leakage and by utilizing accelerometers and gyroscopes (IMUs' sensors) readings that capture junctions depending on their sensitivity for the sudden vibration of a PIG robot during the period of crossing these junctions. However, a new method in section 4.5 has been introduced. This method is based on sensor with spikes that may possible to be connected to the odometers' arms by moving joints to sense the changing in the inner diameter of pipeline at its junctions during the PIG moving along in it. These sensing readings of the sensor is provided to the counting system to update the measurement model for correction the positioning errors.

## 6.3 Conclusion

From the research in this thesis can be concluded the following:

- This thesis's objective was to provide a comprehensive solving to the navigation error problem of a PIG robot, to enable a high accurate localization. A new navigation algorithm based on INS/odometer/counting system achieves this task to overcome from the increasing in navigation errors when depending on a tactical grade IMUs. The correction task in this research aimed to get the robot to be localized to within length of one pipe that may be needed to replace with a tolerance that might reach to less than three meters to be accepted in the case of burrowing for maintenance.
- With absence together the aiding sensors (GPS, land marks, etc.) and the constraints of the motion body the needing for a standalone system becomes necessary to enable INS provides appropriate navigation solving for along traveling period. However, by comprehensive the system manner and by aiding useful system environment parameters will be possible to design system with appropriate constraints to limit/stabilize the growth of navigation errors along the whole traveling. So, new constraints (pipeline length) have been taken in measurement model to limit the drift of the INS errors over short period. The fitting PIG nearly to the pipe inner diameter excludes the consideration of roll angle with staying on a small value.
- In order to implement to get accurate localization for the required parameters under PIG searching travel, the robot positioning needs to be marked by either inside or outside of the straight pipe. Thus, the detection operation for

pipeline junctions that provides junctions data to the counting system to support the measurement model as it has been explained in section 4.5 and shown in results in section 5.4 Fig. 5.14.

The results show that the newly suggested algorithm can improve the errors of INS/odometer from 5.9m to 4.7m that means the new method can reduce the position errors of system nearby 20%. In addition, the INS errors are very large nearby 10.5km. So, the integrated system is very efficient method to reduce the errors of INS.

## 6.4 Recommendations

Here are some recommendations for future researches that may be investigated to enhance the accuracy of the current in-pipeline navigation utilizing navigation grade based on IMU:

- Creation a system of low cost with few inertial sensors can match or exceed the performance of high-end tactical grade IMU using multiple navigation grade based IMUs.
- Including different aiding sensors can improve the positioning of the navigation results. Hence, camera or SICK as vision sensors and magnetometers can be involved in the future research.
- In the application field, the output solving can be validated with an existing map, to detect with optimal efficiency the pipeline parts that require to be maintenance.
- Using the special communication system for length measurement.
- Using the map of pipeline for positioning of Robot.
- Since this is an offline system, still there is other algorithm for non-linear filtering can be examined; such as particle filter (PF).

# REFERENCES

[1] Products, P., and Association, “An Introduction to Pipeline Pigging”, Gulf Pub”, S. 1995.

[2] John Faber and Marcelo Becker, “Study of Robots to Pipelines, Mathematical Models and Simulation”, Mechanical Engineering Department, 2013, Universidade de São Paulo, Brazil.

[3] https://www.linkedin.com/pulse/intelligent-pigging-vs-rov-offshore-pipeline-ahmed-raafat-orabi

[4] World Robotics, “ifr Presents the 2007 World Robotics Statistics Survey”, October 2007, Retrieved from: http://www.worldrobotics.org/index.php.

[5] The International Federation of Robotics, and the United Nations, “World Robotics 2001”, Statistics, Market Analysis, Forecasts, Case Studies and Profitability of Robot Investment, United Nations Publication, 2001, New York and Geneva.

[6] Jong-Hoon Kim., “Sensor-Based Autonomous Pipeline Monitoring Robotic System”, the Department of Computer Science, December 2011, the University of Louisiana State, Ph.D. thesis, USA.

[7] https://www.pipelinesinternational.com/2017/01/10/making-the-graid/

[8] http://www.nord-stream.com/press-info/images/arrival-of-the-inline-inspection-tool-in-lubmin-3503/

[9] http://www.smithflowcontrol.com/news/mechanical-safety-interlocking-simple-complex-pig-trap-procedures/

[10] http://mechanicalgalaxy.blogspot.com.tr/2014/07/

[11] Pipeline and Gas journal. Pigging: “Increasing Appreciation of an Industry Truly Come of Age”. Staff report in pipeline and gas journal, p.48 and 51. Reviewed: http://www.oildompublishing.com/pgj/pgjarchive/aug02articles/piggingreport.pdf.

[12] Fujiwara, S., Kanehara, R., Okada, T., & Sanemori, T., “An Articulated Multi-Vehicle Robot for Inspection and Testing of Pipeline Interiors”. In Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems’ 93, IROS’93, Tokyo, Japan (vol. 1).

[13] King, A.D., “Inertial Navigation-Forty Years of Evolution”, 1998, V.13, N.3, pp. 140-149, GEC Review.

[14] Liu, H., Nassar, S., and El-Sheimy, "Performance Evaluation of Different Optimal Smoothers for Land-Vehicle Navigation Using Integrated GPS/INS Systems", Proceeding of ENC-GNSS 2009, Naples, Italy.

[15] Noureldin, Aboelmagd, "New Measurement While Drilling Surveying Technique Utilizing Sets of Fiber Optic Rotation Sensors", the University of Calgary, 2002, PhD thesis, UCGE Reports, Calgary, Alberta, Canada.

[16] Kim, J. and Sukkarieh, S., "Flight Test Results of GPS/INS Navigation Loop for an Autonomous Unmanned Aerial Vehicle (UAV)", ION GPS 2002, Portland, OR, USA.

[17] El-Gizawy, M. L., "Continuous Measurement-While-Drilling Surveying System Utilizing MEMS Inertial Sensors", the University of Calgary, 2009, PhD thesis, UCGE Report 20284, , Calgary, Alberta, Canada.

[18] Syed, Z.F., "Design and Implementation Issues of a Portable Navigation System", the University of Calgary, 2009, PhD thesis, UCGE Reports Number 20288, Calgary, Alberta, Canada.

[19] Niu, X. and El-Sheimy, "The Development of Low-Cost MEMS-Based IMU for Land Vehicle Navigation Applications Using Auxiliary Velocity Updates in the Body Frame", ION GNSS 2005, Long Beach, CA, USA.

[20] Kaplan, E.D. and Hegarty, C.J., "Understanding GPS Principles and Applications", 2006, Second Edition, Artech house, Inc.

[21] Yang, Y., "Tightly Coupled MEMS INS/GPS Integration with INS Aided Receiver Tracking Loops", the University of Calgary, 2008, PhD thesis, UCGE Reports Number 20270, Calgary, Alberta, Canada.

[22] Petovello, M.G., "Real-Time Integration of a Tactical-Grade IMU and GPS for High-Accuracy Positioning and Navigation", the University of Calgary, 2003, PhD thesis, UCGE Reports Number 20173, Calgary, Alberta, Canada.

[23] Shin, E., "Accuracy Improvement of Low Cost INS/GPS for Land Applications", the University of Calgary, 2001, MSc thesis, UCGE Reports Number 20156, Calgary, Alberta, Canada.

[24] Shin, E., "Estimation Techniques for Low-Cost Inertial Navigation", the University of Calgary, 2005, PhD thesis, UCGE Reports Number 20219, Calgary, Alberta, Canada. Reviewed from: http://www.geomatics.ucalgary.ca/links/GradTheses.html

[25] El-Sheimy, "Inertial Techniques and INS/GPS Integration", Department of Geomatics Engineering, the University of Calgary, November 2007, ENGO 623 Lecture Notes, Canada.

[26] Shin, E. and El-Sheimy, "Navigation Kalman Filter Design for Pipeline Pigging", Journal of Navigation, November 2005, Vol.58, pp.283-295.

[27] Skaloud, J. and Schwarz, K.P., "Application of Inertial Technology to Underground Positioning: The Soudan Mine Shaft Survey", Zeitschrift für Vermessungswesen (ZfV), 2000, Vol 8: pp. 292-299.

[28] Gordon, N. J., Salmond, D. J., and Smith, "Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation", IEE Proceedings Part F: Communications, Radar, and Signal Processing, A. F. M. 1993, 140(2), 107–113.

[29] Gelb, A., "Applied Optimal Estimation", The M.I.T. Press, Massachusetts Institute of Technology, Cambridge, 1974, Massachusetts, USA.

[30] A. H. Jazwinski., "Stochastic Processes and Filtering Theory", Academic Press, 1970, New York.

[31] S. J. Julier and J. K. Uhlmann., "A General Method for Approximating Nonlinear Transformations of Probability Distributions", Technical report, Robotics Research Group, Department of Engineering Science, University of Oxford, 1994, Reviewed from: http://www.robots.ox.ac.uk/ fi siju/index.html.

[32] S. Haykin, "Adaptive Filter Theory", Fourth Edition, 2002, Prentice Hall, Inc.

[33] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte, "A New Approach for the Nonlinear Transformation of Means and Covariance in Linear Filters", IEEE Transactions on Automatic Control, March 2000, Vol. 5, No. 3, pp. 477-482.

[34] S. J. Julier and J. K. Uhlmann, "Unscented Filtering and Nonlinear Estimation" Proc. IEEE, March 2004, vol. 92, no. 3, pp. 401–422.

[35] Wan, E. A. and van der Merwe, R., "Kalman Filtering and Neural Networks", Haykin, S. (Ed.), 2001, Ch. 7, pp.229. John Wiley and Sons, New York.

[36] R. v. d. Merwe, "Sigma-point Kalman Filters for Probabilistic Inference in Dynamic State-space Models", Electrical and Computer Engineering, the University of Portland, 2004, PhD thesis, Oregon Health Sciences.

[37] R. van der Merwe and E. A. Wan, "The Square-Root Unscented Kalman Filter for State and Parameter- Estimation", In International Conference on Acoustics, Speech, and Signal Processing, May 2001, Salt Lake City, Utah, Vol. 6, , pp. 3461-3464.

[38] R. van der Merwe and E. A. Wan, "Efficient Derivative-Free Kalman Filters for Online Learning", In European Symposium on Artificial Neural Networks (ESANN), April 2001, Bruges, Belgium.

[39] Eric A. Wan and Rudolph van der Merwe and Alex T. Nelson, "Dual Estimation and the Unscented Transformation", In Advances in Neural Information Processing Systems 12, November 2000, pp. 666-672, MIT Press, Eds. S.A. Solla and T. K. Leen and K.-R. Muller.

[40] Rauch, H.E., Tung, F. and Striebel, C.T., “Maximum Likelihood Estimates of Linear Dynamic Systems”, AIAA Journal, 1965, V.3, N.8.

[41] Fraser, D.C. and Potter, J.E., “The Optimum Linear Smoother as a Combination of Two Optimum Linear Filters”, IEEE Transactions on Automatic Control, 1969, pp. 387-390.

[42] Crassidis, J.L. and Junkins, J.L., “Optimal Estimation of Dynamic Systems”, Chapman and Hall/CRC, 2004.

[43] Yu, J., Lee, J. G., Park, C. G and Han, H. S., “An Off-Line Navigation of A Geometry PIG Using A Modified Nonlinear Fixed-Interval Smoothing Filter”, Control Engineering Practice, 2005, pp. 1403-1411.

[44] Hang Liu, “Optimal Smoothing Techniques in Aided Inertial Navigation and Surveying Systems”, University of Calgary, November 2009, PhD Thesis.

[45] Wasim Al-Masri, “Inertial Navigation System of In-Pipe Inspection Robot”, Mechatronics Engineering Department, American University of Sharjah College of Engineering, May 2016, MSc thesis, United Arab Emirates.

[46] D. Chatzigeorgiou, K. Youcef-Toumi, and R. Ben-Mansour, “Design of A Novel In-Pipe Reliable Leak Detector”, IEEE/ASME Transactions on Mechatronics, April 2015, vol. 20,no. 2, pp. 824–833.

[47] J.-H. Kim, G. Sharma, and S. Iyengar, “A Fully Autonomous Mobile Robot for Pipeline Exploration”, In 2010 IEEE International Conference on Industrial Technology (ICIT), March 2010, pp. 517–523.

[48] H. Choi and S. Ryew, “Robotic System with Active Steering Capability for Internal Inspection of Urban Gas Pipelines”, Mechatronics, 2002, vol. 12, no.5,pp.713–736,[Online].Available: http://www.sciencedirect.com/science/article/pii/ S0957415801000228.

[49] A. C. Murtra and J. M. M. Tur, “IMU and Cable Encoder Data Fusion for In-Pipe Mobile Robot Localization”, In IEEE International Conference on Technologies for Practical Robot Applications (TePRA), April 2013, pp. 1–6.

[50] S. Bonnabel and E. Salan, “Design and Prototyping of A Low-Cost Vehicle Localization System with Guaranteed Convergence Properties” Control Engineering Practice, 2011, vol. 19, no. 6, pp.591–601. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0967066111000335.

[51] A. Brandt and J. Gardner, “Constrained Navigation Algorithms for Strapdown Inertial Navigation Systems with Reduced Set of Sensors” In American Control Conference, Jun 1998, Proceedings of the 1998, vol. 3, pp. 1848–1852 vol.3.

[52] Hussein Sahli, “MEMS-based Aided Inertial Navigation System for Small Diameter Pipelines”, University of Calgary, April 2016, PhD Thesis, Alberta.

[53] Hagen Schempf, Gasnettm: "Sensing and Communications Network System for Low- and High Pressure Distribution Gas Mains". In Natural Gas Technologies III Conference, January 2005.

[54] A.A.F. Nassiraei, Y. Kawamura, A. Ahrary, Y. Mikuriya, and K. Ishii., "Concept and Design of A Fully Autonomous Sewer Pipe Inspection Mobile Robot KANTARO". In Robotics and Automation, IEEE International Conference on April 2007, pages 136–143.

[55] Hagen Schempf, Edward Mutschler, Vitaly Goltsberg, and William Crowley., Grislee: "Gas-Main Repair and Inspection System for Live Entry Environments". The International Journal of Robotics Research, July 2003, 22(1):603-616.

[56] Stevenson, K.M., Estates, P.V., and Berg, R.L., "Inertial Navigation System", U.S. Patent, 1970, N.O. 3509765.

[57] Savage, P.G., "Strapdown Analytics, Part 1", Strapdown Associates, 2004, Inc., Maple Plain, Minnesota.

[58] Jekeli, C., "Inertial Navigation Systems with Geodetic Applications", Walter de Gryter GmBH and Co, 2001.

[59] Abdel-Hamid, W., "Accuracy Enhancement of Integrated MEMS-IMU/GPS Systems for Land Vehicular Navigation Applications", the University of Calgary, 2005, PhD thesis, UCGE Reports Number 20207, Calgary, Alberta, Canada.

[60] D.H. Titterton and J.L. Weston, "Strapdown Inertial Navigation Technology", 2004, Volume 207, Second Edition, UK.

[61] Nassar, S., "Improving the Inertial Navigation System (INS) Error Model for INS and INS/DGPS Applications", the University of Calgary, 2003, PhD thesis, UCGE Reports Number 20183, Calgary, Alberta, Canada.

[62] Scherzinger, B.M., "Inertial Navigator Error Models for Large Heading Uncertainty", IEEE Position Location and Navigation Symposium, 1996, pp. 477-484.

[63] Bortz, J.E., "A New Mathematical Formulation for Strapdown Inertial Navigation", IEEE Transactions on Aerospace and Electronic Systems, 1971, AES-7 (1):61-66.

[64] Farrell, J.A. and Barth, M., "The Global Positioning System and Inertial Navigation", McGraw-Hill, 1998.

[65] Schwarz, K.P. and Wei, M., "INS/GPS Integration for Geodetic Applications", ENGO 623 Lecture Notes, Department of Geomatics Engineering, the University of Calgary, 1999, Canada.

[66] Niu, X., Yang, Y. and Hassan, T., “Lab Testing and Calibration of MEMS IMUs”, Moblie Multi-Sensor Systems (MMSS) Research Group, Department of Geomatics Engineering, the University of Calgary, 2006, Canada.

[67] Hou, H., “Modeling Inertial Sensor Errors Using Allan Variance”, the University of Calgary, 2004, MSc thesis, UCGE Reports Number 20201, Calgary, Alberta, Canada.

[68] Britting, K.R., “Inertial Navigation Systems Analysis”, John Wiley & Sons, 1971, Inc.

[69] Weinred, A. and Bar-Itzhack, I.Y., “The Psi-Angle Error Equation in Strapdown Inertial Navigation Systems”, IEEE Aerospace and Electronic Systems, 1978, AES-14, NO.3, pp.539-542.

[70] Grewal, M.S. and Andrews, A.P., “Kalman Filtering: Theory and Practice”, Prentice-Hall, 2001, Inc.

[71] Brown, R.G. and Hwang, P.Y.C., “Introduction to Random Signals and Applied Kalman Filtering”, John Wiley & Sons, 1997, Inc.

[72] Godha, S., “Performance Evaluation of Low Cost MEMS-Based IMU Integrated with GPS for Land Vehicle Navigation Application”, the University of Calgary, 2006, MSc thesis, UCGE Reports Number 20239, Calgary, Alberta, Canada.

[73] Bar-Itzhack, I.Y. and Porat, B., “Azimuth Observability Enhancement during Inertial Navigation System In-Flight Alignment”, Journal of Guidance, Control and Dynamics, 1980, Vol.3, NO.4, pp.337-343.

[74] Shin, E. and El-Sheimy, “Backward Smoothing for Pipeline Surveying Applications”, Proceeding of ION NTM, November 2005, San Diego, CA, U.S.

[75] Sukkarieh, S., “Low Cost, High Integrity, Aided Inertial Navigation Systems for Autonomous Land Vehicles”, Department of Mechanical and Mechatronic Engineering, the University of Sydney, 2000, PhD Thesis, Australian Centre for Field Robotics, Sydney, Australia.

[76] Nassar, S., Syed, Z., Niu, X. and El-Sheimy, N., “Improving MEMS IMU/GPS Systems for Accurate Land-Based Navigation Applications”, The Institute of Navigation National Technical Meeting (ION NTM 2006), Monterey, California, USA, pp. 523-529, January 18-20.

[77] http://www.cs.unc.edu/~{welch, gb}

[78] Gao, Y., “Advanced Estimation Methods and Analysis”, ENGO 629 Lecture Notes, Department of Geomatics Engineering, the University of Calgary, 2007, Canada.

[79] Nassar, S., Shin, E., Niu, X. and El-Sheimy, N., “Accurate INS/GPS Positioning with Different Inertial Systems Using Various Algorithms for Bridging GPS Outages”, Proceeding of International Technical Meeting of ION-GNSS, 2005, Long Beach, CA, U.S.

[80] Simon J. Julier and Jerey K. Uhlmann, “A New Extension of the Kalman Filter to Nonlinear Systems”, The Robotics Research Group, Department of Engineering Science, the University of Oxford.

[81] Shin, E.-H. and El-Sheimy, November, “An Unscented Kalman Filter for In-Motion Alignment of Low-Cost IMUs”, In Proceedings of IEEE Position, Location, and Navigation Symposium, 2004, pages 273–279, Monterey, CA.

[82] Kraft, E., “A Quaternion-Based Unscented Kalman Filter for Orientation Tracking”, In Proceedings of Fusion, 2003, Cairns, Australia.

[83] Crassidis, J. L. and Markley, “Unscented Filtering for Spacecraft Attitude Estimation. Journal of Guidance”, Control, and Dynamics, F. L. 2003, 26(4):536–542.

[84] S.J. Julier and J.K. Uhlmann, ‘‘A New Extension of the Kalman Filter to Nonlinear Systems’’, In Proceedings of Aero Sense: The 11th International Symposium on Aerospace/Defence Sensing, 1997, Simulation and Controls.

[85] S.J. Julier, J.K. Uhlmann, and H. Durrant-Whyte, ‘‘A New Approach for Filtering Nonlinear Systems,’’ In Proceedings of the American Control Conference, 1995, pp. 1628–1632.

[86] Pifu Zhang, Jason Gu, Evangelos E. Milios, and Peter Huynh, “Navigation with IMU/GPS/Digital Compass with Unscented Kalman Filter”, International Conference on Mechatronics and Automation, July 2005, Dalhousie University, Niagara Falls, Canada.

[87] Shin, E.-H. , “A Quaternion-Based Unscented Kalman Filter for the Integration of GPS and MEMS INS”. In Proceedings of ION GNSS, 2004, pages 1060-1068, Long Beach, CA.

[88] Noureldin, A., Karamat, T. B., and Georgy, “Fundamentals of Inertial Navigation, Satellite Positioning and Their Integration”, July 2012, New York: Springer.

[89] El-Sheimy, “Introduction to INS - With Applications to Positioning and Mapping”, University of Calgary, November 2012, Graduate Course Lectures.

# CURRICULUM VITAE

## PERSONAL INFORMATION

| | |
|---|---|
| Name, Surname | : Mohammed Alkhatabe |
| Date and birth place | : 03.06.1984 / Iraq |
| Email | : mohammad84.mf@gmail.com |