Inertial Navigation system of In-pipe Inspection Robot

INERTIAL NAVIGATION SYSTEM OF

IN-PIPE INSPECTION ROBOT

by

Wasim Al-Masri

A Thesis Presented to the Faculty of the
American University of Sharjah
College of Engineering
in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science in
Mechatronics Engineering

Sharjah, United Arab Emirates

May 2016

Approval Signatures

We, the undersigned, approve the Master's Thesis of Wasim Al-Masri.

Thesis Title: Inertial Navigation system of In-pipe Inspection Robot.

Signature	**Date of Signature** (dd/mm/yyyy)
___________________________ Dr. Mamoun Abdel-Hafez Associate Professor, Department of Mechanical Engineering Thesis Advisor	_______________
___________________________ Dr. Mohammad A. Jaradat Associate Professor, Department of Mechanical Engineering Thesis Co-Advisor	_______________
___________________________ Dr. Lotfi Romdhane Professor, Department of Mechanical Engineering Thesis Committee Member	_______________
___________________________ Dr. Mohamed Said Abdou Hassan Associate Professor, Department of Electerical Engineering Thesis Committee Member	_______________
___________________________ Dr. Lotfi Romdhane Director, Mechatronics Engineering Graduate Program	_______________
___________________________ Dr. Mohamed El-Tarhuni Associate Dean, College of Engineering	_______________
___________________________ Dr. Leland Blank Dean, College of Engineering	_______________
___________________________ Dr. Khaled Assaleh Interim Vice Provost for Research and Graduate Studies	_______________

Acknowledgements

First of all, I would like to thank God, the most gracious most merciful for His unlimited gifts. I would like to thank my lovely parents and family for their endless help, support, and patience. Also, I would like to sincerely thank my brilliant mentor and advisor Dr. Mamoun Abdel-Hafez who taught, helped, and guided me patiently throughout my thesis. Thanks as well go to my co-advisor Dr. Mohammad Jaradat for his support and help during my thesis works. I would like to extend my gratitude to Dr. Shayok Mukhopadhyay for his technical advice, allowing me to use his international Aramex box and his effort and guidance throughout my master degree study. I also appreciate the tremendous support I got from the Mechatronics Engineering Graduate Program at the American University of Sharjah for sponsoring my M.Sc degree studies. Finally, I must give thanks to my colleagues and best friends in the Mechatronics Center: Ehab AlKhatib, Hassan Umari, Mohamed Elmustafa, Ali Qahtan and Abdel-Rahman Renawi; for their help, all the discussions and the nights that we have spent studying, working and experimenting in the Lab. Also I would like to thank all my friends and colleagues for their support and encouragement.

This thesis is dedicated to my beloved parents . . .

Abstract

The main goal of this study is to design and implement a robust inertial navigation system (INS) for in-pipe inspection robot. To achieve this goal, different mechanization approaches, that are derived in different frames or have different implementation methods, are investigated. These methods include INS derived in e-frame, INS derived in n-frame and 3D reduced inertial sensor system (RISS). The INS uses the full inertial measurement unit (IMU) data to calculate the navigation solution, whereas RISS uses encoder, one single-axis gyroscope, and two accelerometers. Advantages and disadvantages are highlighted for each approach. Due to accumulated error in the INS or RISS solution, a sensor fusion based on extended Kalman filter is proposed. The INS is proposed to be fused with encoder's derived velocity with velocity constraints, and with the detected pipe length as measurements to correct INS solution. RISS is proposed to be fused with the detected pipe length only as measurements to correct its solution. Subsequently, the accuracy of the proposed algorithm is verified experimentally. An experimental setup, with a prototype of the in-pipe robot, is designed and built to test and validate our algorithms in a real pipe. The accuracy of the proposed algorithms was around ± 3 *cm* after sensor fusion.

Search Terms: ***inertial navigation system, reduced inertial sensor system, extended Kalman filter, sensor fusion, GPS, encoder, nonholonomic constraints***

Table of Contents

List of Figures

List of Tables

Nomenclature

i	ECI frame
e	ECEF frame
n	Navigation (NED) frame
b	Body frame
ψ	Yaw angle
θ	Pitch angle
ϕ	Roll angle
C_x^y	Transformation matrix from a frame, x, to a frame, y
Φ	Geodetic latitude
Λ	Geodetic longitude
h	Height above the plane tangent to the reference ellipsoid
R	Range from the Earth's center of gravity to vehicle's center of gravity
ω_e	Earth's angular velocity ($7.292115 \times 10^{-5}\ rad/s$)
μ	The Earth's gravitational constant ($3986004.418 \times 10^{8}\ m^3/s^2$)
J_2	Coefficient of second zonal harmonics of Earth's potential function ($1.082629989051944 \times 10^{-3}$)
R_e	Semi major axis of the Earth (6378137.0 m)
R_N	Normal radius of earth curvature
R_M	Meridian radius of earth curvature
e_m	Major eccentricity (0.0818191908426)
$\hat{x}$	Estimated states vector
$\bar{x}$	Computed 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

Abbreviation

AI	Artificial Intelligence
ANFIS	Adaptive Neuro-Fuzzy Inference System
COTS	Commercial-off-the-Shelf
DCM	Direction Cosine Matrix
DR	Dead-reckoning
ECEF	Earth-Centered Earth-Fixed
ECI	Earth-Centered Inertial
EKF	Extended Kalman Filter
GPS	Global Positioning System
IDNN	Input-Delayed Neural Network
IF	Information Filter
IMU	Inertial Measurement Unit
INS	Inertial Navigation Solution
KF	Kalman Filter
MEMS	Micro-Electro-Mechanical System
NED	North-East-Down
PND	Personal Navigation Device
RISS	Reduced inertial sensor system
SLAM	Simultaneous Localization and Mapping
VSLAM	Visual Simultaneous Localization and Mapping
UKF	Unscented Kalman Filter

Chapter 1: Introduction

The aim of this thesis is to provide a comprehensive solution to the navigation problem of in-pipe inspection robot, by developing a robust navigation algorithm that enables high-accurate localization. This research focuses mainly on analyzing and carrying out navigation systems using commercially available sensors, integrating them using fusion algorithm based on extended Kalman filter.

1.1. Background

Navigation is a technique for the determination of position and velocity of a moving platform with respect to a known reference, but it can also include the attitude of the platform [1]. Position, velocity and attitude are called the navigation states. A navigation system can either be autonomous or be dependent on external sources, or in some cases, a combination of the two. The fusion of the two systems is traditionally based on the technique of Kalman filtering, developed in 1960 mainly for navigating in space. The sensors for a navigation system typically include accelerometers, gyroscopes and radio receivers. There are two fundamental methods for finding a navigation solution:

- position fixing.
- dead reckoning (DR).

Position fixing is based on the information of external sources with known locations (e.g. GPS, MAP and Active beacons). The main advantage of this system is that the solution is independent of previous location estimate, whereas the disadvantage is that the solution is not always available (i.e. the sampling frequency is very low).

On the other hand, dead reckoning is autonomous and relies on knowledge of the initial location, speed and heading information (e.g. INS, Encoder and odometery). By comparing this system to previous one, the advantage of this system is that the solution is always available (i.e. the sampling frequency is very high), but the disadvantage is the accumulative error because this system is based on integration.

Due to advantages and disadvantages of position fixing and dead reckoning systems, in order to have a compact, low-cost and high-accuracy navigation system, we fuse both systems by using one of the following commonly used fusion algorithms:

- Kalman filter [2–6].
- Particle filter [7, 8].
- Artificial intelligence [9, 10].

1.2. Motivation

Pipelines like the one shown in Figure 1 are considered a necessary media in transporting a fluid or gas to various destinations. The fluid, while traveling, will impose pressure on the internal wall of the pipe, and will cause particles formation and precipitation on the internal wall of the pipe. The surrounding environment of the pipe has a great effect on its condition. For instance, it can cause corrosion of the pipe. These points affect its life expectancy, and with time, the pipe becomes more susceptible to leakages and failures. Thus, automated routine inspection of pipes has become widely needed. In-pipe inspection robots are an innovative way for handling pipe inspection and detecting cracks and leakages. To implement such robots, a robust navigation system is needed to navigate the robot through the pipe and to correctly localize the detected defects. Also, this system should be cheap and based on commercially available sensors.

1.3. Thesis Objectives

This thesis is part of a wider project that aims at enhancing risk assessment and condition inspection of an operating pipe in the field, by investigating and developing Navigation system for the in-pipe robot. To achieve this goal, this thesis has the following objectives:

- Different mechanization methods of INS are investigated and they are intended to be used to find out the best suitable method for our application.

Figure 1: Bu Hasa - Habshan gas pipeline, Abu Dhabi, UAE [11].

- A low-cost hardware composed of an IMU, a GPS antenna, encoder with non-holonomic velocity constraints and light sensor are intended to be used, where their measurements are fused using extended Kalman filter.
- An experimental setup consists of a prototype of the in-pipe robot which is designed and built to test our navigation solution in a real pipe.

1.4. Report Structure

This thesis consists of the following six chapters:

- Chapter 1 is an introduction about the theoretical background behind navigation systems in order to give a better understanding of the topic. Also, the objectives of this thesis are explained in this chapter. This chapter also reviews the literature that has been published about topics in the same field.
- Chapter 2 describes the inertial navigation system (INS), its components, its physical implementation methods and the sources of its error.

- Chapter 3 describes the mechanization methodology of INS. The equations of INS are presented for different frames. Finally reduced inertial sensor system (RISS) is presented and its advantages over regular INS is demonstrated.
- Chapter 4 shows the fusion algorithm of INS with other aided systems. The implementation process of fusion algorithm based on extended Kalman filter is demonstrated.
- Chapter 5 describes the designed experimental setup and the experimental procedure. Also, a set of results obtained experimentally is presented.
- Finally, Chapter 6 shows the conclusion of this thesis and the proposed future work that can be performed related to this project.

The next section consists of a brief literature review of the work that was performed and published about the field of navigation.

1.5. Literature Review

Pipeline-based applications have become an integral part of life, so fully autonomous mobile pipeline inspection robot becomes necessary. The problem of designing an in-pipe inspection robot is being continuously addressed, among theses studies [12–14], which propose a robust automated in-pipe inspection robot. In [12], a new in-pipe leak detection system is proposed. Detection is based on the presence of a pressure gradient in the neighborhood of the leak. The authors validate the concepts by building a prototype and evaluating the systems performance under real conditions in an experimental laboratory setup. The design and implementation of a single-modulated fully autonomous mobile pipeline exploration robot, called FAMPER, is presented in [13]. FAMPER's mechanism provides for the excellent mobility in vertical as well as horizontal pipelines, and proposed the system architecture that would enable FAMPER to be fully autonomous. In [14], The authors present a robotic system for inpipe inspection of underground urban gas pipelines,where robot is developed with the purpose of being utilized as a mobile platform for visual and Non-Destructive Testing (NDT) of the pipeline networks. In this work, the robot is configured as an articulated structure-like snake with a tether cable.

Visual odometry and visual simultaneous localization and mapping (VSLAM) are promising techniques for localization of the in-pipe inspection robot. Many studies have been made to localize the robot by using VSLAM and visual odometry [15–21]. In [15], the authors design a VSLAM method that can accurately provide the position of the AUV in the pipe. A group of sensors, consists of INS, digital camera and laser range finder, is utilized in implementing the proposed VSLAM method. The main advantage of this method is the ability of the proposed method in accurate localization of the AUV inside the pipe, while in the same time reducing the processing load, especially for the image processing. A complementary technique for visual/inertial localization of a mobile robot in sewer pipe network is proposed in [16]. The data fusion, in the proposed system, is based on the Graph SLAM framework, where it uses sensor data from an IMU and from a cable encoder. The proposed system is well suited to be considered as a complementary technique to solve situations where visual feature tracking could fail, due to environment situations. An artificial land mark for vision-based SLAM of water pipe rehabilitation robot is proposed in [17]. A monocular visual odometry method for calculating the displacement of the robot inside the culverts is proposed in [18]. Their method starts with finding the image displacement and its rotation using Fourier transform and then relates it to physical motion using total derivative of perspective camera model. In [19], authors investigated two monocular visual odometry algorithms, dense and sparse, that are designed to estimate camera pose in a straight cylindrical pipe in liquefied natural gas pie system. In [20], authors introduced a noise tolerant landmark detection method using line laser beam projected on internal surface of the pipeline. The localization of the robot is performed by generating unique line pattern, once the line laser is projected on the surface of landmarks such as elbow or branch. In [21], authors proposed a reconstruction method of a piping shape by using a rangefinder constructed with an omni-directional camera and an omni-directional laser, to perform self-localization and 3D model construction.

Many research works have been made to obtain a reliable positioning method for different applications of land vehicle by fusing of INS with various aided systems by using different fusion algorithms. In [22], a low-cost navigation system is proposed for enhancing highway traffic safety even in situations such as fog or rain, by using fusion of velocity constraints with IMU and GPS data using Kalman filter and Information filter. In [23], GPS and encoders, fused with a digital map, which is a layer of General Information System (GIS), are used to estimate a vehicles state. The implementation of such a system would be too expensive and not applicable in places where a precise digital map, built up from small 2-D curves, is not available. Measurements of gyroscope, GPS and a speedometer are used in the proposed solution by [24], where a nonlinear observer which consists of three subfilters is used in developing a vehicle navigation solution. One of the subfilters has two gains and estimates yaw angle and gyro bias. Another subfilter has one gain and is responsible for estimating the speedometers scaling. The third one also has one gain and gives estimates of the vehicles velocity and position. In the filtering stage proposed in [24], nonholomonic constraints are taken into consideration to further enhance the estimates. When tested, the algorithm gave reliable results. However, the algorithm does not take into consideration the accelerometer bias and it also does not use the quaternion approach which is known for its stability in finding a vehicles attitude. Constraints are also fused in a navigation solution for land vehicles proposed in [25]. These constraints depend on the orientation of the vehicle relative to the earth and the relation between the attitude of the vehicle with its velocity, and are mainly used to enhance the estimates, especially the estimates of velocity and position. Velocity constraints are also fused with an INS and GPS in [26], for a navigation solution consisting of nine states. The results of this fusion of the velocity constraints along with the vehicles speed showed that the observability and attitude of the vehicle are guaranteed. This leads to decreases in the errors of these states.

Also, several navigation techniques were developed recently for navigation using artificial intelligence (AI) rather than using Kalman filter (KF) to avoid inadequacies present in the approach of KF [10, 27–32]. Some of the main inadequacies in the KF approach for GPS/INS integration are a stochastic error model for every sensor used, which should be predefined, and values like correlation time, variance and covariance should be known accurately. However, these AI-based techniques relate INS errors to INS output at specific time intervals only. They do not take into consideration the error dependency on past INS readings. To overcome this problem, a study proposes the use of Input-Delayed Neural Networks (IDNN) to model the INS errors based on current and some of the past INS data [10], another study proposes an architecture based on an adaptive neuro-fuzzy inference system for fusing the GPS/IMU measurements [32].

To enhance the integrity of the navigation algorithm, fault detection algorithms are proposed in [3,33]. The proposed fault detection algorithm in [3], is for detection and isolation of the possible bias in the encoder by using limited memory noise estimation, to estimate a possible bias in the encoder velocity measurement. In [33], the authors proposed two GPS fault detection and identification (FDI) techniques; the χ^2 gating function and the multiple model adaptive estimation (MMAE). A sequential FDI scheme is proposed to integrate both methods in order to obtain a high-integrity, high-accuracy state estimate from a filter structure with low-computational requirement.

In the next chapters, the inertial navigation system (INS), its components, its physical implementation methods and the source of its error will be explained.

Chapter 2: Inertial Navigation Systems

2.1. Introduction

Inertial navigation system (INS) employs inertial sensors (gyroscopes and accelerometers). The accelerometers and gyroscopes constitute the inertial sensor assembly (ISA) that is housed along with related electronics in a unit called the inertial measurement unit (IMU) [34]. By measuring angular rates, gyroscopes provide attitude information in three dimensions. Accelerometers measure the specific forces (accelerations) that can be converted to velocity and position by a navigation computer using the process of integration. Starting with initial values of position, velocity and attitude, together with a gravity model, the computer operates on the IMUs outputs to furnish the current position, velocity and attitude of the host vehicle, as shown in Figure 2.

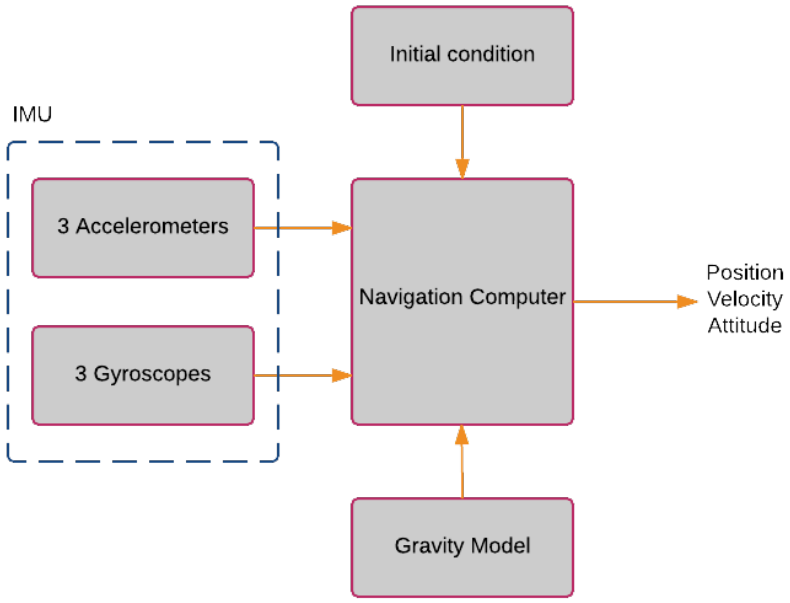


Figure 2: The block diagram of an inertial navigation system (INS).

2.2. Inertial Measurement Unit

The measurements of the acceleration and the rotation of the vehicle are made by a combination of the inertial sensors mounted in a unit called the inertial measurement unit (IMU). As shown in Figure 3, this holds two orthogonal sensor triads; one with three accelerometers and the other with three gyroscopes. Accelerometers measure linear motion in three mutually orthogonal directions, whereas gyroscopes measure angular motion in three mutually orthogonal directions. We assume that, the axes of these two triads are parallel, sharing the origin of the accelerometer triad. The sensor axes are fixed in the body of the IMU, and are therefore called the body axes or body frame. Apart from the inertial sensors, the IMU also contains related electronics to perform self-calibration, and to sample the inertial sensor readings. Subsequentially, these readings are used by the navigation computer.

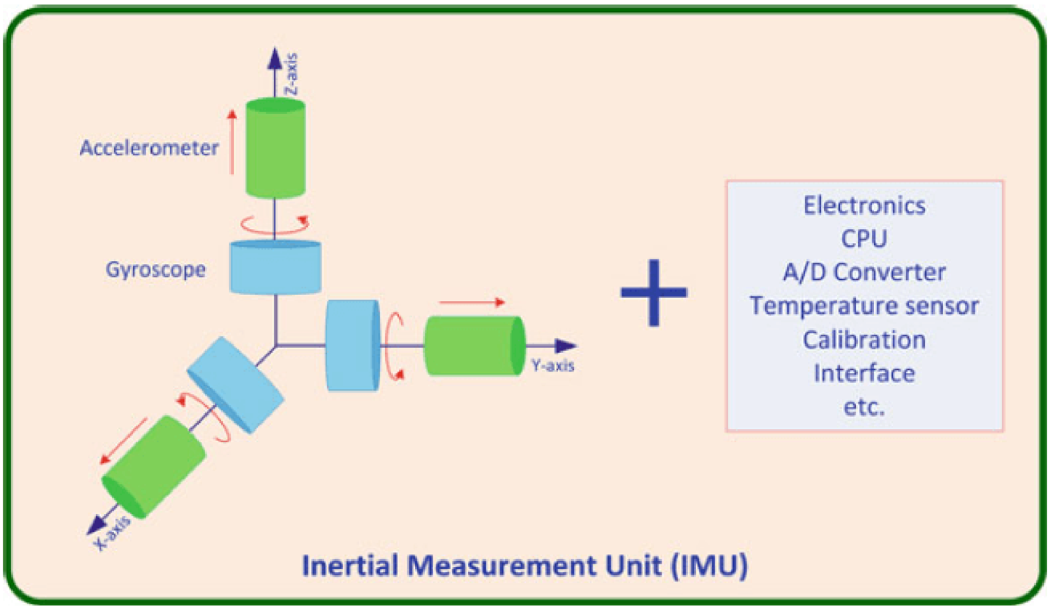


Figure 3: The components of a typical IMU [34].

2.3. Physical Implementation Methods of an INS

Mainly, there are two implementation approaches for an INS:

- Stable platform system (gimbaled system).

- Strapdown system.

As shown in Figure 4(a), in gimbaled systems accelerometers and gyroscopes are fixed on an inner stationary gimbal in a certain direction and pose relative to inertial or earth navigation frame. This state does not change even if the vehicles pose and orientation change. This is because the inner gimbal is isolated from being affected by the motions of the surrounding vehicle. For the inner gimbal to be stable, torque actuators are used along with servo feedback. These systems usually have high accuracy and cost due to the high complexity of their control system and design that can only measure very small changes in deviations. One main drawback of such systems is that they sometimes suffer from gimbal lock. This lock can be fixed by adding a fourth gimbal to the already present three-axis gimbals. However, this solution is usually a very expensive one. On the other hand, strapdown inertial navigation systems are fixed to the body of the vehicle and are not stable in space as they move with the body's motion, as shown in Figure 4(b). The system's angular motion is continuously measured with a set of three perpendicular rate gyros, and the acceleration is measured with a set of three perpendicular accelerometers. Such systems are cheaper, mechanically simpler and stronger than gimbaled ones. However, it is more complex to mathematically analyze, especially if pose in earth and navigation frames are needed.

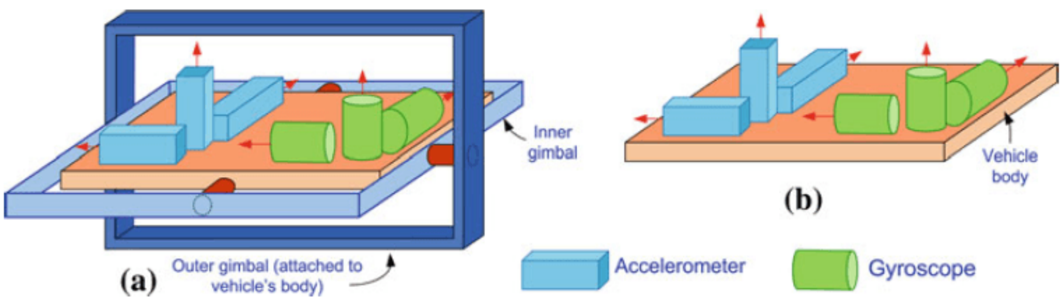


Figure 4: Arrangement of the components of (a) gimbaled system, and (b) strapdown system [34].

2.4. Inertial Sensors Errors

Inertial sensors are subjected to various errors, which get more complex as the price of the sensor decreases. The errors minimize the accuracy of the measured values. We can categorize the errors type as follows.

2.4.1. Systematic errors.

The main systematic errors in accelerometers and gyroscopes come from scale factor, bias and non-linearity in readings. Consequently, any misalignment in the units axes with respect to local navigation axes would cause inaccurate results in the component of acceleration due to gravity, which will affect the position and velocity. These errors are carried on for the next time steps. So, they keep propagating in time. These types of errors can be compensated by laboratory calibration, especially for errors that come from scale factor bias in accelerometers and gyroscopes readings, and among these procedures we have the following:

2.4.1.1. Six-position static test. In this method, the inertial system is mounted on a level table with each sensitive axis pointing alternately up and down (six positions for three axes). Then, estimates of the accelerometer bias and scale factor are obtained by summing and differencing combinations of the inertial system measurements.

2.4.1.2. Angle rate tests. In this type of calibration, the IMU is mounted on a precision rate table, which is rotated through a set of very accurately defined angles. If the table is rotated clockwise and counterclockwise through the same angle then, the biases and scale factors errors of the gyros can be estimated.

2.4.2. Random errors.

Inertial sensors suffer from a variety of random errors, which are usually modeled stochastically in order to mitigate their effects. The most important random error in accelerometers and gyroscopes readings is bias drift and white noise. Mainly, these errors can be minimized by the estimation techniques, which are described in Chapter 4.

2.4.3. Alignment and initialization errors.

One of the most important issues in any navigation solution is the initialization of position, velocity and attitude, and the alignment of the IMU unit. The initialization of velocity can be set to zero thus assuming a stationary vehicle initially. However, the position can be initialized from an external source like a GPS antenna or manually entering initial position values. Also, the initialization of the attitude angles is very important as they are used in the initialization of the quaternions by initializing cosine rotation matrices. Therefore, it should be ensured that the IMU unit is aligned accurately such that the axes of the unit's body frame coincide with the axes of the vehicle's body frame. There are several techniques which were developed for alignment purposes. Some of these are [35]:

- Estimating alignment variables by matching the position with a GPS.
- The usage of a star tracker or an optical line of sight reference to a ground station for estimating alignment.
- Using the gyrocompass alignment approach, where acceleration readings are used to get the rotation directions and local vertical thus estimating north direction.

Chapter 3: Kinematic Modeling and Mechanization of INS

3.1. Introduction

Kinematic modeling is often used where dynamic modeling of a moving object is impractical. It deals with studying the motion of an object without consideration of its mass or the forces acting upon it. In kinematic modeling, the motion of a body is determined only based on position, velocity or acceleration measurements, which are collectively known as observables. In order to measure this acceleration, an accelerometer is needed. Then, this acceleration is integrated once to get the velocity, and twice to get the position with respect to an initial condition. However, the acceleration that we measure with an accelerometer is a combination of acceleration due to gravity and other external forces. The gravity component needs to be removed from the acceleration. In order to do so, the accelerometer's attitude with respect to the computational reference frame needs to be measured, and here comes the role of a gyroscope. The gyroscope measures the angular velocity, which when integrated, gives us the angle change with respect to an initial condition.

Mechanization of INS is the process of converting the output of an IMU into position, velocity and attitude information. The outputs include rotation rates about three body axes ω_{ib}^{b} measured by the gyroscopes triad and three specific forces F^{b} along the body axes measured by the accelerometer triad, all of which are with respect to the computational frame. Mechanization of INS is a recursive process that starts with a specified set of initial values and iterates on the output. A general diagram of INS mechanization is shown in Figure 5.

In order to have a good understanding of how INS and their algorithms work, and before we go into more detail, the coordinate frames used in inertial navigation systems should be understood. In the next section, the main coordinate frames used in inertial navigation systems are demonstrated.

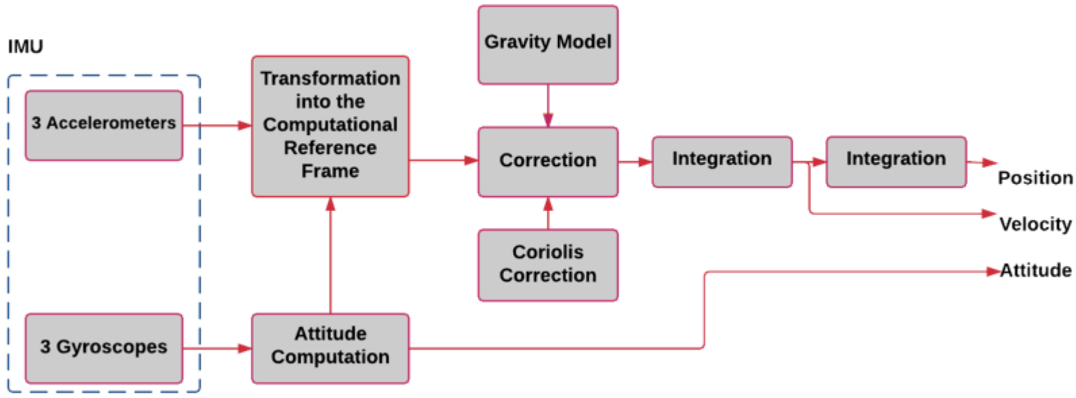


Figure 5: A block diagram of INS mechanization.

3.2. Coordinate Frames

Navigation algorithms involve various coordinate frames and the transformation of coordinates between them. For example, inertial sensors measure motion with respect to an inertial frame which is resolved in the host platforms body frame. This information is further transformed to a navigation frame. A GPS receiver initially estimates the position and velocity of the satellite in an inertial orbital frame. Since the user wants the navigational information with respect to the Earth, the satellites position and velocity are transformed to an appropriate Earth-fixed frame. Since measured quantities are required to be transformed between various reference frames during the solution of navigation equations, it is important to know about the reference frames and the transformation of coordinates between them. So, the listed below are the main coordinate frames used in inertial navigation systems and they are shown in Figure 6.

3.2.1. Earth-centered inertial (ECI) (i-frame).

This frames origin is fixed at the center of the Earth. The axes of this frame do not rotate with the Earth and always point to reference stars. This frame is considered a reference to other coordinate frames.

3.2.2. Earth-centered earth-fixed (ECEF) (e-frame).

In this coordinate frame, we have the origin fixed to the center of the Earth. The z-axis points to the North Pole of the Earth, while the x-axis points to the intersection point

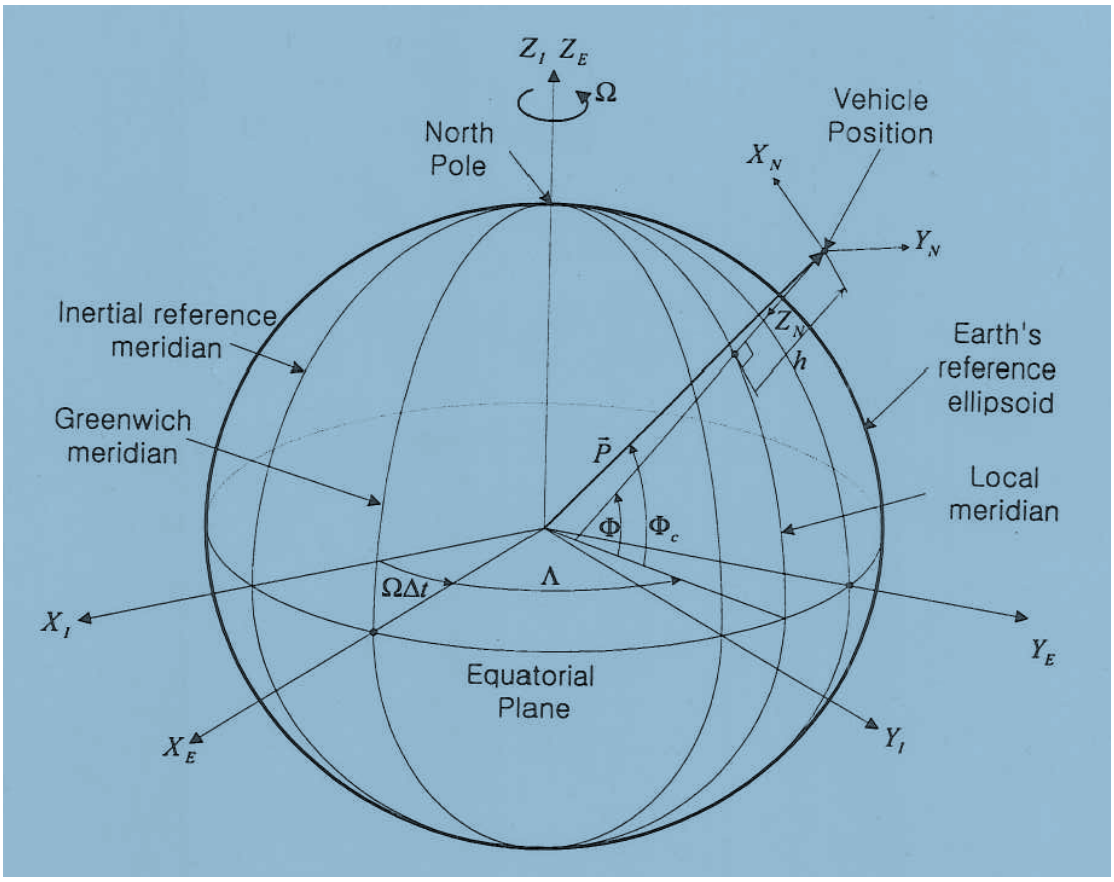


Figure 6: Axes systems

between Greenwich meridian and the equator(0^olatitude, 0^o longitude). The y-axis is orthogonal to both, the x and z axes, and it is found using the right hand rule. Unlike ECI frame, this frame moves with the Earth.

3.2.3. WGS-84 frame.

GPS uses another coordinate frame, called the World Geodetic System (WGS-84). This frame is an ECEF frame. Its origin is at the center of the Earth. The x-axis points to the intersection point between Greenwich meridian and the mean astronomic equator. This equator is the reference meridian which is parallel to the zero-meridian that is defined by the Bureau International de LHeure (BIH), based on adopted longitudes for its stations. The y-axis is measured 90 degrees to the east of the x-axis and completes the right hand rule in the mean astronomic equator plane. The z-axis is parallel to the conventional international origin for a polar motion based on the adopted latitudes for BIH stations.

3.2.4. Navigation frame (n-frame).

In this coordinate frame, the origin is at the location of inertial navigation system. The x and y axes are tangent to the Earth ellipsoid, with the x-axis pointing to the North Pole of the Earth, and the y-axis pointing towards the east. The z-axis points perpendicularly down to the ellipsoid, which means that this axis does not have to pass through the center of the Earth. That is why sometimes this frame is referred to as NED frame, meaning North East Down frame. This frame is also known as the geocentric frame.

The transformation matrix, C_e^n, that transforms a coordinate of the e-frame to the n-frame is given by

$$C_e^n = \begin{bmatrix} -\sin\Phi\cos\Lambda & -\sin\Phi\sin\Lambda & \cos\Phi \\ -\sin\Lambda & \cos\Lambda & 0 \\ -\cos\Phi\cos\Lambda & -\cos\Phi\sin\Lambda & -\sin\Phi \end{bmatrix} \quad (1)$$

where Φ and Λ are geodetic latitude and geodetic longitude of the robot's position.

3.2.5. Body frame (b-frame).

The x-axis of this frame is aligned with the x-axis of the robot's body (the roll axis), pointing forward. The y-axis points perpendicularly to the right (the pitch axis), while the z-axis points perpendicularly down (the yaw axis) as shown Figure 7. This frame is fixed to the body of the robot, so it moves with the robot.

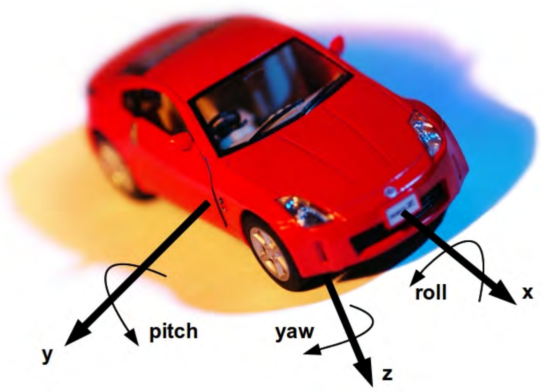


Figure 7: Body frame axes and (roll(ϕ),pitch(θ),yaw(ψ)) angles [22]

The transformation matrix from the b-frame to the n-frame is described by the conventional angles (roll(ϕ),pitch(θ),yaw(ψ)) of the body frame:

$$C_b^n = \begin{bmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\sin\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \tag{2}$$

Then, the transformation matrix from the b-frame to the e-frame is defined as

$$C_b^e = C_e^{n\,T} C_b^n = C_n^e C_b^n \tag{3}$$

3.3. Measurement Model of the Gyroscope and the Accelerometer

The mechanization of an INS depends on the gyroscopes and accelerometers measurements. The error in accelerometer and gyroscope measurements can be eliminated by using the calibration techniques described in section 2.4, but for the white noise and bias uncertainty, which represent the random walk that is assumed to be slowly varying parameters, can be minimized by the estimation techniques described in chapter 4. The measurement equation for the accelerometer and gyroscope is demonstrated in the following subsections.

3.3.1. Gyroscope measurement model.

Measurements of angular rate from the gyroscope can be modeled by the following equation:

$$\bar{\omega}_{ib}^b = \omega_{ib}^b + b_g^b + w_g \tag{4}$$

where

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

ω_{ib}^b is the true gyroscope measurement vector (rad/s)

b_g^b is the gyroscope instrument bias vector (rad/s)

w_g is a vector representing the gyro sensor noise (rad/s).

3.3.2. Accelerometer measurement model.

Measurements of the specific force can be modeled by the following equation:

$$\bar{F}^b = F^b + b_a^b + w_a \quad (5)$$

where

$\bar{F}^b$ is the accelerometer measurement vector (m/s^2)

F^b is the true specific force vector (m/s^2)

b_a^b is the accelerometer instrument bias vector (m/s^2)

w_a is a vector representing the accelerometer sensor noise (m/s^2).

In the next sections, the mechanization of INS in different frames and by using different methods is presented. The experimental results for theses algorithms are presented in Chapter 5.

3.4. Mechanization of INS in e-frame

3.4.1. Position and velocity.

The vehicles kinematics are obtained from the accelerometers and gyroscopes measurements. The time rate of change of the velocity of the vehicle in e-frame can be shown to be given as [34]:

$$\dot{V}^e = C_b^e F^b - 2\Omega_{ie}^e V^e - {\Omega_{ie}^e}^2 P^e + G^e \quad (6)$$

where $2\Omega_{ie}^{e}V^{e}$ represents the Coriolis acceleration and ${\Omega_{ie}^{e}}^{2}P^{e}$ represents the Centrifugal acceleration, where Ω_{ie}^{e} is as follows.

$$\Omega_{ie}^{e} = [\omega_{ie}^{e}]^{\times} \tag{7}$$

$$\omega_{ie}^{e} = \begin{bmatrix} 0 \\ 0 \\ \omega_e \end{bmatrix} \tag{8}$$

where $[\omega_{ie}^{e}]^{\times}$ represents the skew symmetric matrix of ω_{ie}^{e} and ω_e is earth sidereal rotation rate.

G^e in Equation (6), is the gravitational vector in e-frame, which is described by using the Legendre polynomial. If we considered up to second order polynomial, the gravitational force vector components in the ECEF, G^e, are as follows.

$$g_x = -\frac{\mu}{R^2}\left[1 - \frac{3}{2}J_2\left(\frac{R_e}{R}\right)^2\left(5\frac{z_e^2}{R^2} - 1\right)\right]\frac{x_e}{R} \tag{9}$$

$$g_y = -\frac{\mu}{R^2}\left[1 - \frac{3}{2}J_2\left(\frac{R_e}{R}\right)^2\left(5\frac{z_e^2}{R^2} - 1\right)\right]\frac{y_e}{R} \tag{10}$$

$$g_z = -\frac{\mu}{R^2}\left[1 - \frac{3}{2}J_2\left(\frac{R_e}{R}\right)^2\left(5\frac{z_e^2}{R^2} - 3\right)\right]\frac{z_e}{R} \tag{11}$$

where the J_2 is the coefficient of second zonal harmonics of the Earth potential function, the R_e is the semi major axis of the Earth, and the R is the range from the Earth center to the center of the robot.

After obtaining the velocity of the vehicle, we can integrate the velocity to find the vehicle's position, by using the following

$$\dot{P}^e = V^e \tag{12}$$

3.4.2. Attitude.

The rate gyros measure the vehicle angular velocity relative to inertial frame, ω_{ib}^{b}.

The angular velocity of vehicle relative to the ECEF is given by

$$\omega_{eb}^{b} = \omega_{ib}^{b} - \omega_{ie}^{b} \tag{13}$$

where $\omega_{ie}^{b} = C_{e}^{b}\omega_{ie}^{e}$.

For angular motion and attitude calculation, usually, the quaternion approach is used to avoid the singularity problem that is faced with the Euler approach [36]. Let a quaternion $q^e = \begin{bmatrix} q_o & q_1 & q_2 & q_3 \end{bmatrix}^T$ describes the attitude between the b-frame and e-frame. Then the DCM from b-frame to e-frame, C_b^e, is represented by

$$C_b^e = \begin{bmatrix} q_o^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_3q_o) & 2(q_1q_3-q_oq_2) \\ 2(q_1q_2-q_oq_3) & q_o^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3-q_oq_1) \\ 2(q_1q_3-q_oq_2) & 2(q_2q_3-q_oq_1) & q_o^2-q_1^2-q_2^2+q_3^2 \end{bmatrix} \tag{14}$$

The quaternion is governed by following differential equation

$$\dot{q}^e = \frac{1}{2}\bar{\Omega}_{eb}^{b} q^e \tag{15}$$

where

$$\bar{\Omega}_{eb}^{b} = \begin{bmatrix} 0 & -\omega_{eb}^{b\,T} \\ \omega_{eb}^{b} & -\left[\omega_{eb}^{b}\right]^{\times} \end{bmatrix} \tag{16}$$

with a constraint:

$$q_o^2+q_1^2+q_2^2+q_3^2 = 1 \tag{17}$$

Mostly, the initial attitude is represented by (roll(ϕ), pitch(θ), yaw(ψ)) angles. The initial condition of quaternion is necessary to integrate the quaternion differential equation. Let C_{ij} denote the i-th row, j-th column element of initial C_b^e. Initial values of

q^e are obtained from Equation (14):

$$q_o = \frac{1}{2}\sqrt{1+C_{11}+C_{22}+C_{33}} \tag{18}$$

$$q_1 = \frac{C_{32}-C_{23}}{4q_o} \tag{19}$$

$$q_2 = \frac{C_{13}-C_{31}}{4q_o} \tag{20}$$

$$q_3 = \frac{C_{21}-C_{12}}{4q_o} \tag{21}$$

3.4.3. Summary.

The e-frame mechanization equations can be summarized as follows.

$$\begin{bmatrix} \dot{P}^e \\ \dot{V}^e \\ \dot{q}^e \end{bmatrix} = \begin{bmatrix} V^e \\ C_b^e F^b - 2\Omega_{ie}^e V^e - \Omega_{ie}^{e\,2} P^e + G^e \\ \frac{1}{2}\bar{\Omega}_{eb}^b q^e \end{bmatrix} \tag{22}$$

which represents the mechanization equations in the e-frame where the inputs are the sensed accelerations F^b from the accelerometers and rotation rates ω_{ib}^b from the gyroscopes. The outputs are the position vector P^e, the velocity vector V^e, and the quaternions, all expressed in the e-frame.

3.5. Mechanization of INS in n-frame

In many applications the mechanization equations are preferable to be in the n-frame for the following reasons [7]:

- The navigation solution is intuitive to the user on or near the Earth's surface. Also, since the axes of the n-frame are aligned to the local east, north and down directions, the attitude angles (roll(ϕ),pitch(θ),yaw(ψ)) can be obtained directly from the output of the mechanization equations when solved in the n-frame.
- The computational errors in the navigation parameters on the horizontal (E-N) plane are bound by the Schuler effect [37].

3.5.1. Position and velocity.

The time rate of change of the velocity of the vehicle in n-frame can be shown to be given as [34]:

$$\dot{V}^n = C_b^n F^b - (2\Omega_{ie}^n + \Omega_{en}^n)V^n + g^n \tag{23}$$

The g^n mentioned in Equation (23) represents the Earth's gravity field, which equals

$$g^n = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \tag{24}$$

where g is the magnitude of the normal gravity vector over the surface of the ellipsoid, which can be computed as a function of latitude and height by a closed form expression known as the Somigliana formula.

In Equation (23), Ω_{ie}^n is defined as

$$\Omega_{ie}^n = [C_e^n \omega_{ie}^e]^{\times} \tag{25}$$

and Ω_{en}^n is defined as

$$\Omega_{en}^n = [\omega_{en}^n]^{\times} \tag{26}$$

where

$$\omega_{en}^n = \begin{bmatrix} \frac{v_e}{R_N+h} \\ \frac{-v_n}{R_M+h} \\ \frac{-v_e\ tan(\Phi)}{R_N+h} \end{bmatrix} \tag{27}$$

where h is the geodetic height of the robot's position. R_N and R_M represent the normal radius and the meridian radius respectively, which are defined as

$$R_N = \frac{R_e}{\left(1 - e_m^2\, sin^2(\Phi)\right)^{\frac{1}{2}}} \tag{28}$$

$$R_M = \frac{R_e\left(1 - e_m^2\right)}{\left(1 - e_m^2\, sin^2(\Phi)\right)^{\frac{3}{2}}} \tag{29}$$

where e_m represents the major eccentricity of the Earth reference ellipsoid.

After obtaining the velocity of the vehicle, we can integrate the velocity to find the vehicle's position, based on the following differential equation:

$$\dot{P}^e = C_n^e V^n \tag{30}$$

3.5.2. Attitude.

Similar to what is explained in subsection 3.4.2, the angular velocity of vehicle relative to the n-frame is given by

$$\omega_{nb}^b = \omega_{ie}^n - \omega_{en}^n \tag{31}$$

where ω_{in}^b is calculated as follows.

$$\omega_{in}^b = C_n^b \omega_{in}^n \tag{32}$$

where

$$\omega_{in}^n = \omega_{ie}^n + \omega_{in}^n \tag{33}$$

where $\omega_{ie}^n = C_e^n \omega_{ie}^e$, and ω_{en}^n is defined by Equation (27).

The attitude between the b-frame and n-frame is described by quaternion $q^n = \begin{bmatrix} q_o & q_1 & q_2 & q_3 \end{bmatrix}^T$. Then the DCM from b-frame to n-frame, C_b^n is represented by

$$C_b^n = \begin{bmatrix} q_o^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_3q_o) & 2(q_1q_3-q_oq_2) \\ 2(q_1q_2-q_oq_3) & q_o^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3-q_oq_1) \\ 2(q_1q_3-q_oq_2) & 2(q_2q_3-q_oq_1) & q_o^2-q_1^2-q_2^2+q_3^2 \end{bmatrix} \tag{34}$$

The quaternion is governed by following differential equation

$$\dot{q}^n = \frac{1}{2}\bar{\Omega}_{nb}^b q^e \tag{35}$$

where

$$\bar{\Omega}_{nb}^b = \begin{bmatrix} 0 & -\omega_{nb}^{b\,T} \\ \omega_{nb}^b & -\left[\omega_{nb}^b\right]^\times \end{bmatrix} \tag{36}$$

with a constraint:

$$q_o^2+q_1^2+q_2^2+q_3^2 = 1 \tag{37}$$

To convert from quaternion to the (roll(ϕ),pitch(θ),yaw(ψ)) angles, let C_{ij} denote the i-th row, the j-th column element of C_b^n. Then the (roll(ϕ),pitch(θ),yaw(ψ)) angles are given by

$$\phi = tan^{-1}(C_{32}/C_{33}) \tag{38}$$

$$\theta = sin^{-1}(-C_{31}) \tag{39}$$

$$\psi = tan^{-1}(C_{21}/C_{11}) \tag{40}$$

3.5.3. Summary.

The n-frame mechanization equations can be summarized as follows.

$$\begin{bmatrix} \dot{P}^e \\ \dot{V}^n \\ \dot{q}^n \end{bmatrix} = \begin{bmatrix} C_n^e V^n \\ C_b^n F^b - (2\Omega_{ie}^n + \Omega_{en}^n)V^n + g^n \\ \frac{1}{2}\bar{\Omega}_{nb}^b q^n \end{bmatrix} \tag{41}$$

where C_n^e relates V^n with $\dot{P}^e$, P^e is represented in ECEF Cartesian coordinates (x_e, y_e, z_e).

3.6. Mechanization of 3D Reduced Inertial Sensor System

The 3D Reduced Inertial Sensor System (RISS) presented in this section is a 3D navigation solution using encoder, one single-axis gyroscope, and two accelerometers [7]. The gyroscope is aligned with the vertical axis of the vehicle. The pitch and roll angles that would have been provided by the two eliminated gyroscopes are now calculated using the two accelerometers. The 3D RISS integrates the measurements from the vertically aligned gyroscope and the two horizontal accelerometers with speed readings provided by an odometer or wheel encoders.

The advantages of using 3D RISS solution over using full IMU solution are the following:

- By using the accelerometers in full IMU mechanization, the error in velocity is proportional to t, and in position is proportional to t^2. To overcome the error, due to accelerometer biases, the encoder-derived speed will be used in RISS.
- An uncompensated bias in one of the two horizontally aligned gyroscopes, in full IMU mechanization, introduces error in velocity proportional to t^2, and error in position proportional to t^3 [38]. To overcome the error due to horizontal gyroscope biases, the pitch and roll angles will be obtained by combining measurements from the horizontal accelerometers and the encoder.

The mechanization of 3D RISS system is depicted in Figure 8, to obtain a navigation solution of 3D position, 3D velocity and 3D attitude.

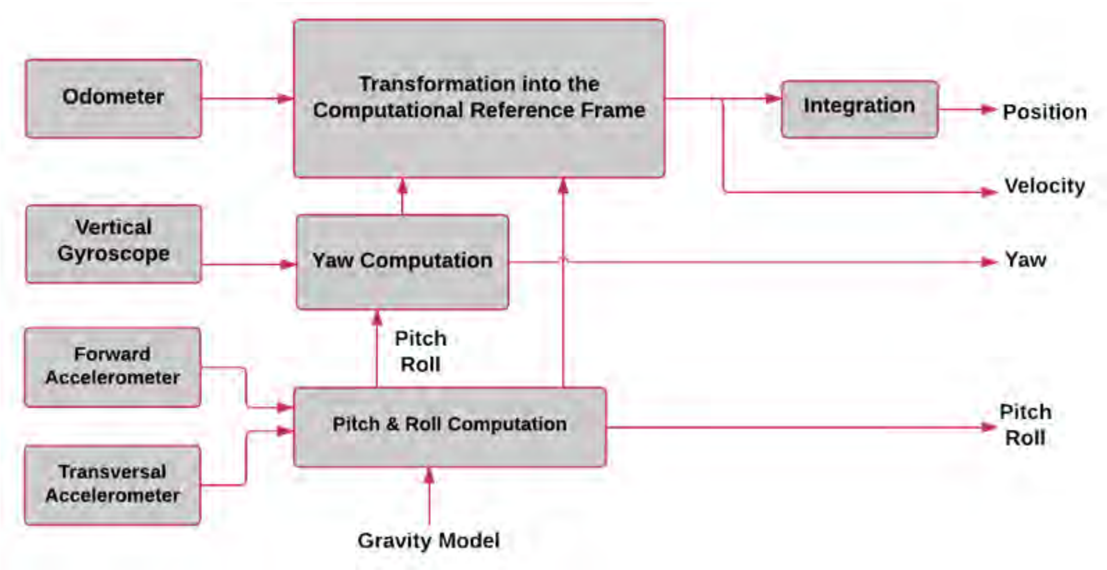


Figure 8: A block diagram of RISS mechanization.

3.6.1. Position and velocity.

As mentioned before, the velocity in 3D RISS is provided directly from the wheel encoder measurements, V^b. Then, the velocity in n-frame can be calculated as

$$V^n = \begin{bmatrix} V^N \\ V^E \\ V^D \end{bmatrix} = C_b^n V^b \tag{42}$$

Then, the position can be calculated by solving the following deferential equation:

$$\dot{P}^e = C_n^e V^n \tag{43}$$

3.6.2. Attitude.

3.6.2.1. Roll and pitch calculation. The forward accelerometer measure the specific force, which includes the forward acceleration of the robot as well as the gravity component in forward direction. Based on that, the pitch angle at time k is calculated by removing gravity effect as follows.

$$\theta_k = sin^{-1}\left[\frac{F^b_{x,k} - a^b_{x,k}}{g_k}\right] \tag{44}$$

where $a^b_{x,k}$ is calculated by differentiating encoder's velocity, $V^b_{x,k}$.

In similar way, to calculate the roll angle, the normal component of robot's acceleration must be subtracted from the lateral accelerometer measurements, then roll angle is given as

$$\phi_k = -sin^{-1}\left[\frac{F^b_{y,k} + V^b_{x,k}\ \omega^b_{z,k}}{g\ cos(\theta_k)}\right] \tag{45}$$

3.6.2.2. Yaw calculation. The first step to calculate the yaw angle, which is the deviation of the robot's forward (x_b) axis from the north, measured clockwise in the E-N plane, is by finding $U^l_{k|k-1}$. Where $U^b_{k|k}$ is the unit vector along the forward direction of the robot at time k observed from the body frame at time k, $U^b_{k|k} = \begin{bmatrix}1 & 0 & 0\end{bmatrix}^T$. $U^b_{k|k-1}$ is the unit vector along the forward direction of the robot at time k, observed from the body frame at time $k-1$. The relation between the two previous mentioned unit vectors is given as

$$U^b_{k|k} = R_{z_b}(\gamma^z_k)\ U^b_{k|k-1} \tag{46}$$

where γ_k represents the angle of clockwise rotation around the vertical downward axis of the body frame of the robot, in time interval Δt between time epochs $k-1$ and k,

$$\gamma^z_k = \omega^z_k\ \Delta t \tag{47}$$

Due to the fact, $R_{z_b}\left(\gamma^z_k\right)$is an orthogonal rotation matrix, $U^b_{k|k-1}$ is given as

$$U^b_{k|k-1} = \left(R_{z_b}\left(\gamma^z_k\right)\right)^T U^b_{k|k} = \begin{bmatrix} cos\left(\gamma^z_k\right) & sin\left(\gamma^z_k\right) & 0 \\ -sin\left(\gamma^z_k\right) & cos\left(\gamma^z_k\right) & 0 \\ 0 & 0 & 1 \end{bmatrix}^T \begin{bmatrix}1\\0\\0\end{bmatrix} = \begin{bmatrix} cos\left(\gamma^z_k\right) \\ sin\left(\gamma^z_k\right) \\ 0 \end{bmatrix} \tag{48}$$

Then, the unit vector along the forward direction of the vehicle at time k seen from the navigation frame at time $k-1$ is given as

$$U^n_{k|k-1} = \begin{bmatrix} U^N \\ U^E \\ U^D \end{bmatrix} = R^n_{b,k-1} U^b_{k|k-1} \tag{49}$$

Finally, by adding the yaw's component due to the rotations performed by the robot due to the angle γ^z_k, which is equal $tan^{-1}\left(\frac{U^N}{U^E}\right)$, to other two components, due to the Earths rotation and the change of orientation of the navigation frame, the resultant of the yaw angle is defined as

$$\psi_k = tan^{-1}\left(\frac{U^E}{U^N}\right) + \omega_e\, sin\left(\Phi_{k-1}\right)\Delta t + \frac{V^E_{k-1}\, tan\left(\Phi_{k-1}\right)}{R_N + h_{k-1}}\Delta t \tag{50}$$

3.6.3. Summary.

The 3D RISS mechanization equations can be summarized in discrete form as follows.

$$\begin{bmatrix} P^e_k \\ V^n_k \\ \phi_k \\ \theta_k \\ \psi_k \end{bmatrix} = \begin{bmatrix} P^e_{k-1} + C^e_{n,k} V^n_k \\ V^n_{k-1} + C^n_{b,k} V^b_k \\ -sin^{-1}\left[\frac{F^b_{y,k} + V^b_{x,k}\omega^b_{z,k}}{g\ cos(\theta_k)}\right] \\ sin^{-1}\left[\frac{F^b_{x,k} - a^b_{x,k}}{g}\right] \\ tan^{-1}\left(\frac{U^E_k}{U^N_k}\right) + \omega_e\, sin\left(\Phi_{k-1}\right)\Delta t + \frac{V^E_{k-1}\ tan(\Phi_{k-1})}{R_N + h_{k-1}}\Delta t \end{bmatrix} \tag{51}$$

where the inputs are encoder velocity, $V^b_{x,k}$, the sensed accelerations in forward direction, F^b_x, and lateral direction, F^b_y, of the robot, and rotation rates, $\omega^b_{ib,z}$, from the vertical gyroscope. The outputs are the position vector, P^e, the velocity vector, V^n, and the attitude represented by (roll(ϕ),pitch(θ),yaw(ψ)) angles.

Chapter 4: Integration of INS with Aided Systems

The positioning solution obtained by INS tends to drift with time due to the integration, which can lead to unbounded accumulation of errors. Inertial navigation alone, especially with low cost sensors, is not suitable for accurate positioning over a long period of time, making it inapplicable to depend on them. This makes it necessary to fuse INS with aided systems such as GPS, Encoder, Camera, LIDAR and Sonars. The aiding systems are needed in order to limit the errors and predict their behavior. The error models are required for the analysis and estimation of the error sources that are associated with any inertial navigation system. The estimator options include Kalman filters (KF) [2–6], particle filters (PF) [4,7,8], and artificial intelligence (AI) techniques [6,9,10]. Traditionally various forms of KF are used for sensor fusion, therefore, in this study, we will restrict ourselves to extended Kalman filter (EKF) technique.

4.1. Error Equation

The INS accuracy is affected by various sources. These include errors during the initial alignment procedure, sensor errors, and the limitations of the processing algorithm. To see the effect of these errors on the navigational output parameters (position, velocity and attitude), it is important to understand their propagation through the navigation equations. Once the nature of the errors is known, one can mitigate them by proper modeling and estimation techniques. Mechanization equations describe physical process by a deterministic dynamic system. Navigation parameters can be determined using these state equations by taking kinematic measurements along a nominal trajectory. Since the solution of these equations incorporate errors (both deterministic and stochastic), sensor error models are required for analysis and estimation purposes. The errors of dynamic systems are variable in time, therefore, they are described by differential equations.

4.1.1. INS error equation in e-frame.

As explained in [39, 40], the computed Direct Cosine Matrix, $C_b^{\bar{e}}$, note that $\bar{e}$

denotes the computed e-frame, is represented as:

$$C_b^{\bar{e}} = C_e^{\bar{e}} C_b^e \tag{52}$$

where $C_e^{\bar{e}}$ is the error in the computation of C_b^e. If the error is assumed to be small,

$$C_e^{\bar{e}} = (I - 2[\delta q^e]^{\times}) \tag{53}$$

where the quaternion δq^e describes the attitude between E and $\bar{E}$ frame:

$$\delta q^e = \begin{bmatrix} \delta q_1 \\ \delta q_2 \\ \delta q_3 \end{bmatrix} \tag{54}$$

Therefore

$$C_b^{\bar{e}} = (I - 2[\delta q^e]^{\times}) C_b^e \tag{55}$$

Let

$$\bar{P}^e = P^e + \delta P^e \tag{56}$$

$$\bar{V}^e = V^e + \delta V^e \tag{57}$$

$$\bar{b}_a^b = b_a^b + \delta b_a^b \tag{58}$$

$$\bar{b}_g^b = b_g^b + \delta b_g^b \tag{59}$$

where '$-$' denotes the computed value of each variable and it is assumed that the quantities $\delta P^e,\ \delta V^e,\ \delta b_a^b\ and\ \delta b_g^b$ are very small. Substituting Equation (56) into Equation (12), Equation (57) into Equation (6), with assuming

$$\dot{\bar{b}}_a^b = 0_{3\times 1} \tag{60}$$

$$\dot{\bar{b}}_g^b = 0_{3\times 1} \tag{61}$$

yield after neglecting higher order terms the following error equations:

$$\delta \dot{P}^e = \delta V^e \tag{62}$$

$$\delta \dot{V}^e = (\bigtriangledown \bar{G}^e - \Omega^2)\delta P^e - 2\Omega \delta V^e - 2\left[C_b^{\bar{e}} \bar{F}^b\right]^\times \delta q^e + C_b^{\bar{e}} \delta b_a^b + C_b^{\bar{e}} w_a \tag{63}$$

$$\delta \dot{q}^e = -\Omega \delta q + \frac{1}{2} C_b^{\bar{e}} \delta b_g^b + \frac{1}{2} C_b^{\bar{e}} w_g \tag{64}$$

$$\delta \dot{b}_a^b = w_{ba} \tag{65}$$

$$\delta \dot{b}_a^b = w_{bg} \tag{66}$$

where gravity tensor is given as

$$\bigtriangledown \bar{G}^e = \begin{bmatrix} \frac{\partial g_x}{\partial x_e} & \frac{\partial g_x}{\partial y_e} & \frac{\partial g_x}{\partial z_e} \\ \frac{\partial g_y}{\partial x_e} & \frac{\partial g_y}{\partial y_e} & \frac{\partial y_x}{\partial z_e} \\ \frac{\partial g_z}{\partial x_e} & \frac{\partial g_z}{\partial y_e} & \frac{\partial g_z}{\partial z_e} \end{bmatrix} \tag{67}$$

If the second zonal harmonic term in the gravitational force is neglected, then

$$\frac{\partial g_x}{\partial x_e} = -\frac{\mu}{\bar{R}^3}[1 - 3(\frac{\bar{x}_e}{\bar{R}})^2] \tag{68}$$

$$\frac{\partial g_y}{\partial y_e} = -\frac{\mu}{\bar{R}^3}[1 - 3(\frac{\bar{y}_e}{\bar{R}})^2] \tag{69}$$

$$\frac{\partial g_z}{\partial z_e} = -\frac{\mu}{\bar{R}^3}[1 - 3(\frac{\bar{z}_e}{\bar{R}})^2] \tag{70}$$

$$\frac{\partial g_x}{\partial y_e} = -3\frac{\mu}{\bar{R}^3}(\frac{\bar{x}_e \bar{y}_e}{\bar{R}}) \tag{71}$$

$$\frac{\partial g_x}{\partial z_e} = -3\frac{\mu}{\bar{R}^3}(\frac{\bar{x}_e \bar{z}_e}{\bar{R}}) \tag{72}$$

$$\frac{\partial g_x}{\partial z_e} = -3\frac{\mu}{\bar{R}^3}(\frac{\bar{y}_e \bar{z}_e}{\bar{R}}) \tag{73}$$

$$\frac{\partial g_y}{\partial x_e} = \frac{\partial g_x}{\partial y_e}, \frac{\partial g_z}{\partial x_e} = \frac{\partial g_x}{\partial z_e}, \frac{\partial g_z}{\partial y_e} = \frac{\partial g_y}{\partial z_e} \tag{74}$$

Representing Equations (62)-(66) in state space representation yields:

$$\dot{x} = Ax + Bw \tag{75}$$

which represents error model equation used by Kalman filter:

$$\begin{bmatrix} \delta\dot{P}^e \\ \delta\dot{V}^e \\ \delta\dot{q}^e \\ \delta\dot{b}_a^b \\ \delta\dot{b}_g^b \end{bmatrix} = \begin{bmatrix} 0_{3\times3} & I_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ \bigtriangledown\bar{G}^e - {\Omega_{ie}^e}^2 & -2\Omega_{ie}^e & -2[C_b^{\bar{e}}\bar{F}^b]^\times & C_b^{\bar{e}} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & -\Omega_{ie}^e & 0_{3\times3} & \frac{1}{2}C_b^{\bar{e}} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{bmatrix} \begin{bmatrix} \delta P^e \\ \delta V^e \\ \delta q^e \\ \delta b_a^b \\ \delta b_g^b \end{bmatrix} + \begin{bmatrix} 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ C_b^{\bar{e}} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & \frac{1}{2}C_b^{\bar{e}} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & I_{3\times3} \end{bmatrix} \begin{bmatrix} w_a \\ w_g \\ w_{ba} \\ w_{bg} \end{bmatrix} \tag{76}$$

4.1.2. INS error equation in n-frame.

In similar way to what is explained in subsection 4.1.1, the error model sate equation for Kalman filter, derived in n-frame [34]:

$$\begin{bmatrix} \delta\dot{P}^e \\ \delta\dot{V}^n \\ \delta\dot{q}^n \\ \delta\dot{b}_a^b \\ \delta\dot{b}_g^b \end{bmatrix} = \begin{bmatrix} 0_{3\times3} & C_{\bar{n}}^{\bar{e}} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & -2[C_b^{\bar{n}}\bar{F}^b]^{\times} & C_b^{\bar{n}} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & -\Omega_{in}^n & 0_{3\times3} & \frac{1}{2}C_b^{\bar{n}} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{bmatrix} \begin{bmatrix} \delta P^e \\ \delta V^n \\ \delta q^n \\ \delta b_a^b \\ \delta b_g^b \end{bmatrix}$$

$$+ \begin{bmatrix} 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ C_b^{\bar{n}} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & \frac{1}{2}C_b^{\bar{n}} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & I_{3\times3} \end{bmatrix} \begin{bmatrix} w_a \\ w_g \\ w_{ba} \\ w_{bg} \end{bmatrix} \tag{77}$$

4.1.3. INS error equation for RISS.

As explained in [34], the error sate equation for Kalman filter for 3D RISS:

$$\begin{bmatrix} \delta\dot{P}^e \\ \delta\dot{V}^n \\ \delta\dot{b}_r \end{bmatrix} = \begin{bmatrix} 0_{3\times3} & C_{\bar{n}}^{\bar{e}} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & A_1 \\ 0_{3\times3} & 0_{3\times3} & A_2 \end{bmatrix} \begin{bmatrix} \delta P^e \\ \delta V^n \\ \delta b_r \end{bmatrix} + \begin{bmatrix} 0_{3\times1} \\ 0_{3\times1} \\ \sigma_r \end{bmatrix} w_r \tag{78}$$

where

$$A_1 = \begin{bmatrix} -V^E & V^N & 0 \\ V^N & V^E & 0 \\ 0 & V^D & 0 \end{bmatrix} \tag{79}$$

$$A_2 = \begin{bmatrix} 0 & 0 & -1 \\ 0 & -\gamma_{od} & 0 \\ 0 & 0 & -\beta_z \end{bmatrix} \tag{80}$$

$$b_r = \begin{bmatrix} b_\psi \\ b_{od} \\ b_{g,z}^b \end{bmatrix} \tag{81}$$

$$\sigma_r = \begin{bmatrix} 0 \\ \sqrt{2\,\gamma_{od}\,\sigma_{od}^2} \\ \sqrt{2\,\beta_z\,\sigma_{g,z}^2} \end{bmatrix} \quad (82)$$

where w_r is the unit-variance white Gaussian noise, b_ψ is the error in the yaw angle, b_{od} is the error of the encoder-derived speed, and $b_{g,z}^b$ is the stochastic drift in the vertical gyroscope. The stochastic errors associated with the gyroscope and the encoder-derived speed are modeled by a Gauss-Markov model, where γ_{od} is the reciprocal of the autocorrelation time for the error in encoder-derived speed, σ_{od}^2 is the variance of the noise associated with γ_{od}, β_z is the reciprocal of the autocorrelation time for the vertical gyroscopes stochastic bias drift, and $\sigma_{g,z}^2$ is the variance of the noise associated with β_z.

4.2. Error Measurement Model

Encoder-derived velocity measurements and pipe length measurements are used to correct the drift in the navigation state of the robot, which are obtained from INS mechanization. In addition to encoder-derived velocity, velocity constraints are taken into consideration to achieve a high-accuracy state estimate. The encoder-derived velocity with zero velocity constraint measurement equation and pipe length measurement equation are derived in the following subsections.

4.2.1. Encoder velocity measurements with nonholonomic constraints.

As explained in [3], in land vehicles applications, the velocity in the x axis of the body frame, V_x^b, is obtained from the wheel encoder and the velocity in the y and z axes of the body frame, V_y^b and V_z^b, are constrained to zero mean with an additive white noise process, where the fusion of these constraints helps in improving the estimated states. The velocity obtained from wheel encoder with zero velocity constraints measurement equation is represented as

$$z_{v_b}^{encoder}(k) = \begin{bmatrix} v_b^x \\ 0 \\ 0 \end{bmatrix} = C_e^b V^e + \begin{bmatrix} \eta_x \\ \eta_y \\ \eta_z \end{bmatrix} \quad (83)$$

where $\begin{bmatrix}\eta_x & \eta_y & \eta_z\end{bmatrix}^T$ is the encoders velocity constraint measurement noise. The INS derived velocity in the body frame is defined as

$$z_{v_b}^{INS}(k) = \bar{V}_k^b = C_{\bar{e}}^b \, \bar{V}^e \tag{84}$$

As explained in subsection 4.1.1, the nominal states are related to the true states as

$$\begin{aligned} C_e^b &= C_{\bar{e}}^b C_e^{\bar{e}} = C_b^{\bar{e}} \left(I - 2\,[\delta q^e]^\times\right) & (85) \\ V^e &= \bar{V}^e - \delta V^e & (86) \end{aligned}$$

by substituting Equation (85) and Equation (86) in Equation (83), we get

$$z_{v_b}^{encoder}(k) = C_{\bar{e}}^b \left(I - 2\,[\delta q^e]^\times\right)(\bar{V}^e - \delta V^e) + \begin{bmatrix}\eta_x \\ \eta_y \\ \eta_z\end{bmatrix} \tag{87}$$

Expanding the terms in the above equation:

$$z_{v_b}^{encoder}(k) = C_{\bar{e}}^b \bar{V}^e - C_{\bar{e}}^b \; \delta V^e - 2C_{\bar{e}}^b \; [\delta q^e]^\times \, \bar{V}^e + 2C_{\bar{e}}^b \; [\delta q^e]^\times \, \delta V^e + \begin{bmatrix}\eta_x \\ \eta_y \\ \eta_z\end{bmatrix} \tag{88}$$

With assuming, the term $2C_{\bar{e}}^b \; [\delta q^e]^\times \, \delta V^e$ is negligible to first order, and defining

$$2C_{\bar{e}}^b \; [\delta q^e]^\times \, \bar{V}^e = -2C_{\bar{e}}^b \; [\bar{V}^e]^\times \, \delta q^e \tag{89}$$

Equation (88) can be given as

$$z_{v_b}^{encoder}(k) = C_{\bar{e}}^b \bar{V}^e - C_{\bar{e}}^b \; \delta V^e + 2C_{\bar{e}}^b \; [\bar{V}^e]^\times \, \delta q^e + \begin{bmatrix}\eta_x \\ \eta_y \\ \eta_z\end{bmatrix} \tag{90}$$

By subtracting Equation (90)from Equation (84), the velocity constraint measurement error is defined as

$$z_{v_b}(k) = z_{v_b}^{INS}(k) - z_{v_b}^{encoder}(k) = C_{\bar{e}}^{b}\ \delta V^{e} - 2C_{\bar{e}}^{b}\ [\bar{V}^{e}]^{\times}\ \delta q^{e} + \begin{bmatrix} \eta_x \\ \eta_y \\ \eta_z \end{bmatrix} \tag{91}$$

The covariance of the velocity constraint measurement error is given as:

$$R_{v_b} = \begin{bmatrix} \sigma_{v_{x_b}} & 0 & 0 \\ 0 & \sigma_{v_{y_b}} & 0 \\ 0 & 0 & \sigma_{v_{z_b}} \end{bmatrix} \tag{92}$$

4.2.2. Pipe length detection measurements.

In the field, the pipeline is composed of standard pieces, which have known length for each and they are usually connected to each other by welding. By detecting the welding line between each two pieces, we can define the beginning and the end for each piece, which has known length. This length measurements can be used to correct the drift in the navigation state obtained by INS mechanization process as explained below.

Let the pipe piece, j, has a known length, l_j, between edge detection measurement, $j-1$, and measurement, j, then the pipe length detection measurement equation is defined as

$$z_{p_b}^{L}(j) = L_j^{b} = \begin{bmatrix} l_j \\ 0 \\ 0 \end{bmatrix} = C_e^{b} L_j^{e} + \begin{bmatrix} \mu_x \\ \mu_y \\ \mu_z \end{bmatrix} \tag{93}$$

where $\begin{bmatrix} \mu_x & \mu_y & \mu_z \end{bmatrix}^T$ is the detected pipe length measurement noise. The INS derived pipe length measurement in the body frame is defined as

$$z_{p_b}^{INS}(j) = \bar{L}_j^b = \begin{bmatrix} \left\| \bar{P}^e(j) - \hat{P}^e(j-1) \right\| \\ 0 \\ 0 \end{bmatrix} = C_{\bar{e}}^b \, \bar{L}_j^e \tag{94}$$

In the same way as explained in subsection 4.2.1, the detected pipe length measurement error is defined as

$$z_{p_b}(j) = z_{p_b}^{INS}(n) - z_{p_b}^{L}(j) = C_{\bar{e},j}^b \, \delta L_j^e - 2C_{\bar{e},j}^b \, \left[\bar{L}_j^e\right]^\times \delta q^e + \begin{bmatrix} \mu_x \\ \mu_y \\ \mu_z \end{bmatrix} \tag{95}$$

It can be shown that, δL_j^e represents δP_j^e and $\bar{L}_j^e$ represents $\bar{P}_j^e - \hat{P}_{j-1}^e$, then Equation (95) becomes

$$z_{p_b}(j) = z_{p_b}^{INS}(j) - z_{p_b}^{L}(j) = C_{\bar{e},j}^b \, \delta P_j^e - 2C_{\bar{e},j}^b \, \left[\bar{P}_j^e - \hat{P}_{j-1}^e\right]^\times \delta q^e + \begin{bmatrix} \mu_x \\ \mu_y \\ \mu_z \end{bmatrix} \tag{96}$$

where $C_{\bar{e},j}^b$ is assumed to be constant for the whole straight pipe piece. This assumption is reasonable because the pipeline is mainly composed from straight pieces. The covariance of the velocity constraint measurement error is given as:

$$R_{p_b} = \begin{bmatrix} \sigma_{p_{x_b}} & 0 & 0 \\ 0 & \sigma_{p_{y_b}} & 0 \\ 0 & 0 & \sigma_{p_{z_b}} \end{bmatrix} \tag{97}$$

4.2.3. Summary.

In subsection 4.2.1 and subsection 4.2.2, based in the deference between the encoders velocity with constraints measurement and the INS derived velocity, and the deference between the detected pipe length measurement and the INS derived pipe length measurement, the error measurement equations are derived for e-frame. The error measurement equations can be represented by matrix form as

$$z(k) = Hx(k) + v(k) \tag{98}$$

where $H(k)$ is observation matrix, and $v(k)$ is the vector of error measurements white noise, its $cov[v(k)]$ is defined as

$$cov[v(k)] = R(k) \tag{99}$$

If only the encoders velocity with constraints measurement is available, then Equation (98) can be written as

$$z(k) = z_{v_b}^{INS}(k) - z_{v_b}^{encoder}(k) = \begin{bmatrix} 0_{3\times3} & C_{\bar{e}}^{b} & 2C_{\bar{e}}^{b}\,[\bar{V}^e]^{\times} & 0_{3\times3} & 0_{3\times3} \end{bmatrix} x(k) + v(k) \tag{100}$$

where $cov[v(k)] = R_{v_b} = R(k)$. In case of both measurements, the encoders velocity with constraints and the detected pipe length, are available, Equation (98) can be written as

$$\begin{aligned} z(k) &= \begin{bmatrix} z_{p_b}^{INS}(j) - z_{p_b}^{L}(j) \\ z_{v_b}^{INS}(k) - z_{v_b}^{encoder}(k) \end{bmatrix} \\ &= \begin{bmatrix} C_{\bar{e},j}^{b} & 0_{3\times3} & 2C_{\bar{e},j}^{b}\left[\bar{P}_j^e - \hat{P}_{j-1}^e\right]^{\times} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & C_{\bar{e},k}^{b} & 2C_{\bar{e},k}^{b}\,[\bar{V}^e]^{\times} & 0_{3\times3} & 0_{3\times3} \end{bmatrix} x(k) + v(k) \end{aligned} \tag{101}$$

where $cov[v(k)]$ is defined as

$$cov[v(k)] = \begin{bmatrix} R_{p_b} & 0_{3\times3} \\ 0_{3\times3} & R_{v_b} \end{bmatrix} = R(k) \tag{102}$$

Similarly, in the case of implementing INS mechanization in n-frame, the error measurement equation, when only the encoders velocity is available, is defined as

$$z(k) = z_{v_b}^{INS}(k) - z_{v_b}^{encoder}(k) = \begin{bmatrix} 0_{3\times3} & C_{\bar{n},k}^{b} & 2C_{\bar{n},k}^{b}[\bar{V}^n]^{\times} & 0_{3\times3} & 0_{3\times3} \end{bmatrix} x(k) + v(k) \tag{103}$$

and the error measurement equation, when both measurements are available, is defined as

$$\begin{aligned} z(k) &= \begin{bmatrix} z_{p_b}^{INS}(j) - z_{p_b}^{L}(j) \\ z_{v_b}^{INS}(k) - z_{v_b}^{encoder}(k) \end{bmatrix} \\ &= \begin{bmatrix} C_{\bar{e},j}^{b} & 0_{3\times3} & 2C_{\bar{e},j}^{b}\left[\bar{P}_j^e - \hat{P}_{j-1}^e\right]^{\times} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & C_{\bar{n},k}^{b} & 2C_{\bar{n},k}^{b}[\bar{V}^n]^{\times} & 0_{3\times3} & 0_{3\times3} \end{bmatrix} x(k) + v(k) \end{aligned} \tag{104}$$

where $cov[v(k)]$ is defined by Equation (102).

In the case of implementing RISS mechanization, encoder-derived velocity measurements are utilized in the mechanization process to get the navigation state. The only available measurement used to correct the drift in the navigation state of the robot, is the detected pipe length measurement. If this measurement is available, then the error measurement equation is defined as

$$z(j) = z_{p_b}^{INS}(j) - z_{p_b}^{L}(j) = \begin{bmatrix} C_{\bar{e},j}^{b} & 0_{3\times3} & 0_{3\times3} \end{bmatrix} x(j) + v(j) \tag{105}$$

where $cov[v(j)] = R_{p_b} = R(j)$.

4.3. INS Kalman Filter based Integration

4.3.1. Discrete-time extended Kalman filter.

The extended Kalman filter (EKF), the nonlinear version of the Kalman filter which linearizes about the propagated state estimate $\hat{x}(k|k-1)$, is used to obtain an updated estimate of the error [41]. EKF corresponds to the closed-loop filter configuration in which the estimated errors are fed back to the inertial system to correct its output as shown in Figure 9, so the INS errors remain small and the linearity assumption required for the KF technique is supported. The discretized dynamic equation of

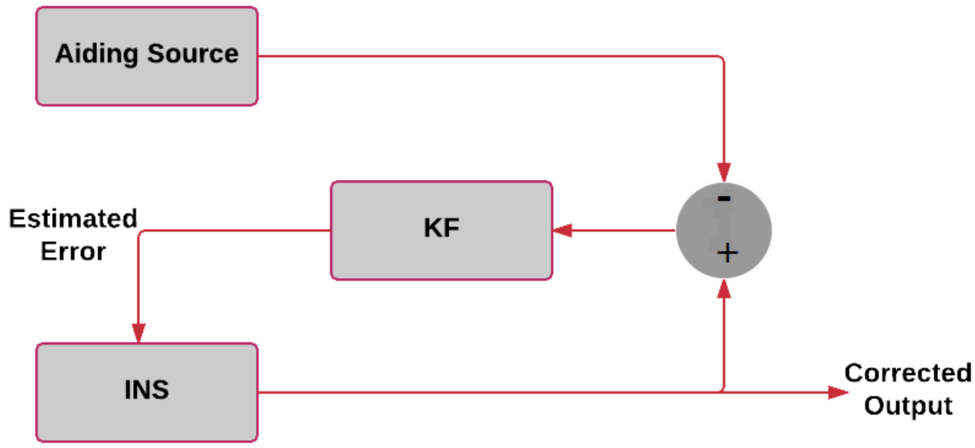


Figure 9: Block diagram of an EKF

the system's error is given as:

$$x(k) = Fx(k-1) + \Gamma w(k-1) \tag{106}$$

where $w(k-1)$ is a dynamics noise, assumed as a Gaussian white noise. In the case of dealing with e-frame and n-frame the covariance of $w(k)$ is defined as

$$cov[w(k-1)] = Q = \begin{bmatrix} I_{3\times3} * {\sigma_a}^2 & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I_{3\times3} * {\sigma_g}^2 & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & I_{3\times3} * \sigma_b^{a2} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & I_{3\times3} * \sigma_b^{g2} \end{bmatrix} \tag{107}$$

where for the case of dealing with RISS the variance of $w_r(k)$ is defined as

$$var[w_r(k-1)] = Q = 1 \tag{108}$$

The F is transition matrix, where $F = I + A\Delta t$. The Γ is the noise distribution matrix, where $\Gamma = B\Delta t$. The discretized error measurement equation is defined by Equation (98).

The implementation of Kalman filter is illustrated in Figure 10, with the initialization assumptions that $x(0 \mid 0)$ and $P(0 \mid 0)$ are known. In this study, $\hat{x}(k \mid k-1)$ will always equal zero because the navigation states after each epoch is updated, as explained in the next section.

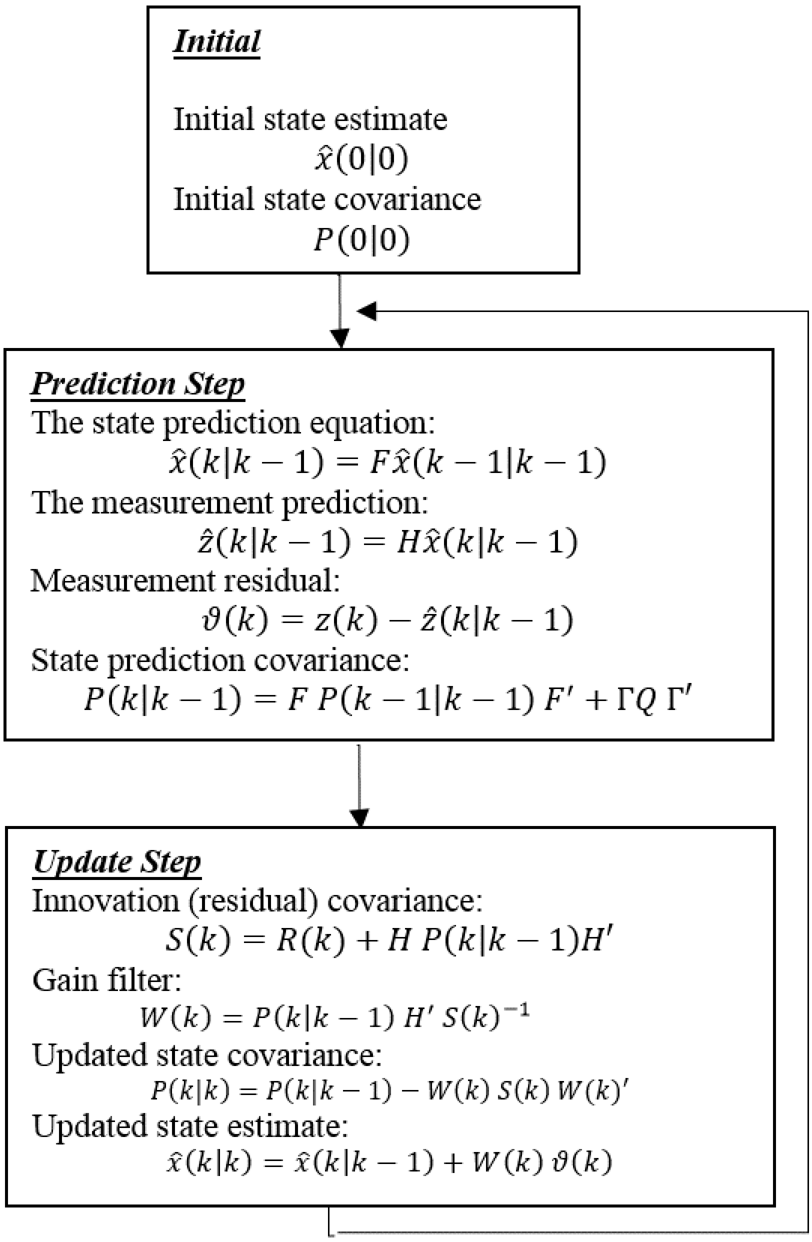


Figure 10: Flowchart of the KF.

4.3.2. State update.

From Kalman filter, we get posteriori estimate of the error states, $\delta\hat{P}^e$, $\delta\hat{V}^e$, $\delta\hat{q}^e$, $\delta\hat{b}_a^b$, and $\delta\hat{b}_g^b$. In the case of doing a full IMU mechanization in e-frame, the navigation state variables are updated with these estimated errors as follows.

$$\begin{aligned}
\hat{P}^e &= \bar{P}^e - \delta\hat{P}^e && (109)\\
\hat{V}^e &= \bar{V}^e - \delta\hat{V}^e && (110)\\
\hat{q}^e &= \delta\hat{q}^e \otimes \bar{q}^e && (111)\\
\hat{b}_a^b &= \bar{b}_a^b - \delta\hat{b}_a^b && (112)\\
\hat{b}_g^b &= \bar{b}_g^b - \delta\hat{b}_g^b && (113)
\end{aligned}$$

where $\otimes$ is the quaternion conjugate and the quaternion multiplication operator

$$\delta\hat{q}^e \otimes \bar{q}^e = \begin{bmatrix} \bar{q}_o & -\bar{q}_1 & -\bar{q}_2 & -\bar{q}_3 \\ \bar{q}_1 & \bar{q}_o & \bar{q}_3 & -\bar{q}_2 \\ \bar{q}_2 & -\bar{q}_3 & \bar{q}_o & \bar{q}_1 \\ \bar{q}_3 & \bar{q}_2 & -\bar{q}_1 & \bar{q}_o \end{bmatrix} \begin{bmatrix} 1 \\ \delta\hat{q}_1 \\ \delta\hat{q}_2 \\ \delta\hat{q}_3 \end{bmatrix} \quad (114)$$

after updating $\hat{q}^e$, it should be normalized. In the case of doing a full IMU mechanization in n-frame, the navigation state variables are updated in similar way to what explained before. Also, for the case of doing RISS, the navigation state variables are updated in similar way to what explained before, but for updating the sensor's bias, $\hat{b}_r$ is updated as

$$\hat{b}_r = \bar{b}_r - \delta\hat{b}_r \quad (115)$$

Chapter 5: Experimental Work

5.1. Pre-Experimental Results: Car Test

Because the building of the experimental setup was still in progress and for validation, the proposed algorithm is implemented and tested using a vehicle driving on a path around the university campus , with an IMU and a GPS antenna. In this test, a loosely coupled INS/GPS integration is used. The position and velocity estimated by GPS receiver are directly used to correct the INS errors. The results of the proposed solution is validated against a commercial off-the shelf (COTS) MIDG solution. Sensors used are described in the following subsection.

5.1.1. Sensors used.

MIDG IIC.

The MIDG IIC is an aided inertial navigation system (INS) for use in applications requiring attitude, position, velocity, acceleration, and angular rates for navigation or control [42]. An internal GPS receiver measures position and velocity and passes this information to the data fusion processor to be combined with the inertial data to generate an optimal solution. An internal three-axis magnetometer provides a magnetic heading reference when needed. The frequency at which readings can be obtained range from 1 to 50 Hz. In this experiment, the sensor is used at 50 Hz. The specifications of the MIDG unit are shown in Figure 11.

GPS Antenna.

A GPS antenna is used to provide us with the position and velocity of the vehicle in ECEF frame at 5 Hz. The GPS works properly when at least four of 24 satellites orbiting the Earth are in the line of sight of the receiver antenna. GPS readings are fused with those obtained from IMU to correct the estimate of our solution as GPS usually gives more accurate readings than IMU but is updated at a slower rate.

Figure 11: MIDG II unit [42].

Power Requirements	
Input Voltage	10 VDC-32 VDC
Power	1.2 W max
Measurements	
Angular Rate	
Range	$\pm 300°/sec$
Non-Linearity	0.1% of FS
Noise Density	$0.05°/sec/\sqrt{Hz}$
3 dB Bandwidth	20 Hz
Acceleration	
Range	$\pm 6\ g$
Non-Linearity	0.3% of FS
Noise Density	$150\ \mu g/\sqrt{Hz}$
3 dB Bandwidth	20 Hz
Attitude Accuracy (Tilt)	$0.4°\ (1\sigma)$
Position Accuracy	2 m CEP, WAAS/EGNOS available
Physical	
Size	$1.50"\ W \times 0.87"H \times 1.67"D$
Weight	55 grams

Figure 12: MIDG specifications [42].

Figure 13: GPS antenna.

5.1.2. Results.

INS mechanization in e-frame without sensor fusion.

As shown in Figure 14, without fusion of INS with any other systems like GPS, our solution will diverge .

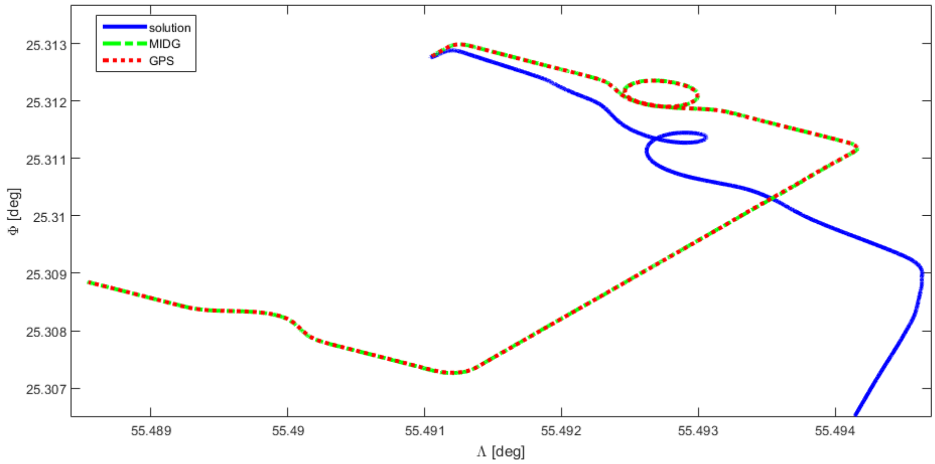


Figure 14: Mechanization of INS in e-frame without sensor fusion.

INS/GPS fusion based on EKF, in e-frame.

The performance of the proposed navigation algorithm is shown in Figures 15-17. The solution of proposed algorithm is achieved by integrating INS with GPS based on EKF in e frame, where the measurements equation is as follows.

$$\begin{aligned} z(k) &= \begin{bmatrix} z_p^{INS}(k) - z_p^{GPS}(k) \\ z_v^{INS}(k) - z_v^{GPS}(k) \end{bmatrix} \\ &= Hx(k) + v(k) \end{aligned} \tag{116}$$

and H is defined as follows.

$$H = \begin{bmatrix} I & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 \end{bmatrix} \tag{117}$$

Figure 16 and Figure 17 show the position and velocity error normal distribution, where the error is defined as the error between the proposed solution and the MIDG solution.

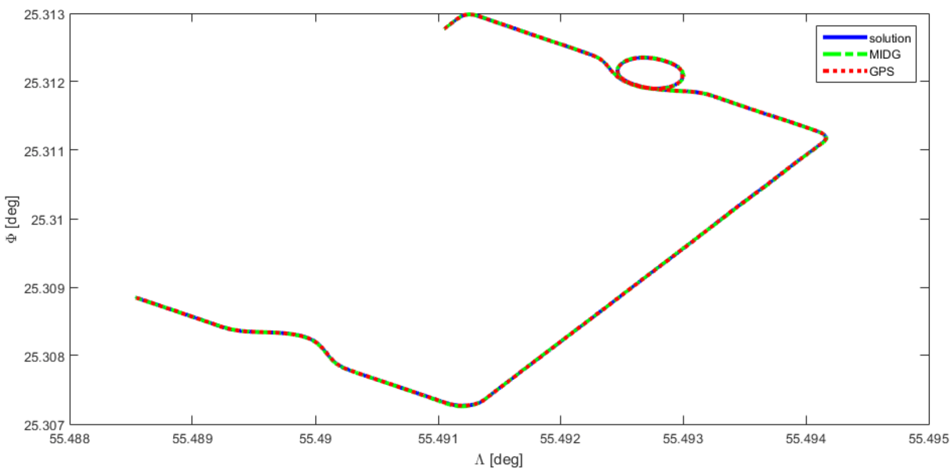


Figure 15: Path estimate of the Vehicle, with using INS/GPS fusion based on EKF in e-frame.

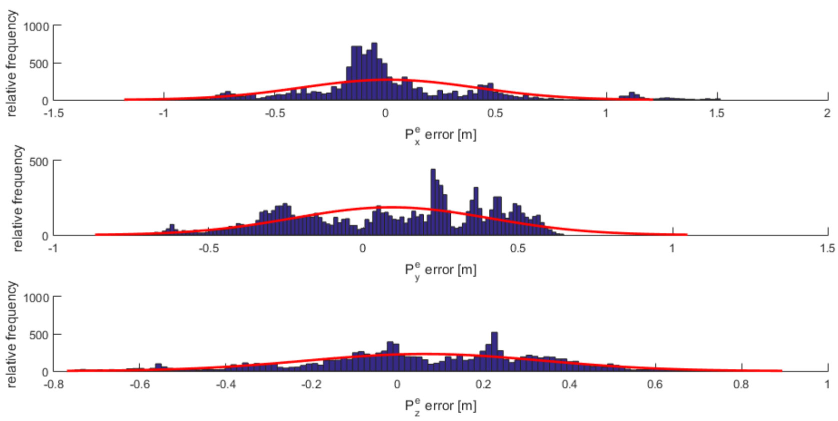


Figure 16: Position error distribution for INS/GPS fusion in e-frame.

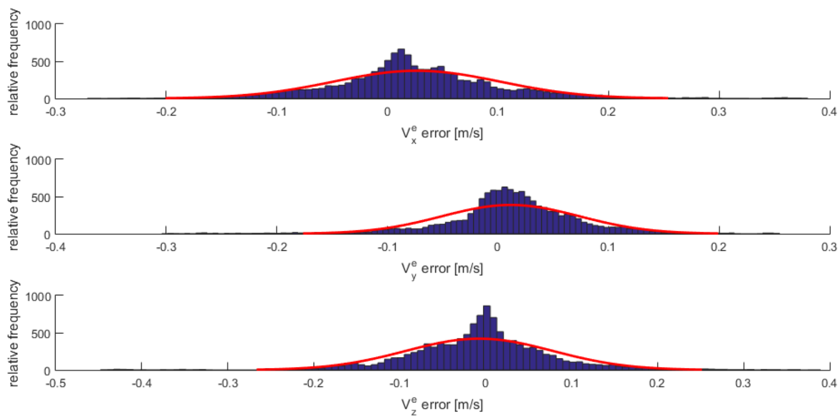


Figure 17: Velocity error distribution for INS/GPS fusion in e-frame.

INS/GPS fusion based on EKF, in n-frame.

The performance of the proposed navigation algorithm, integrating INS with GPS based on EKF in n frame, is shown in Figures 18-20. The results for n-frame is very similar to the previous results.

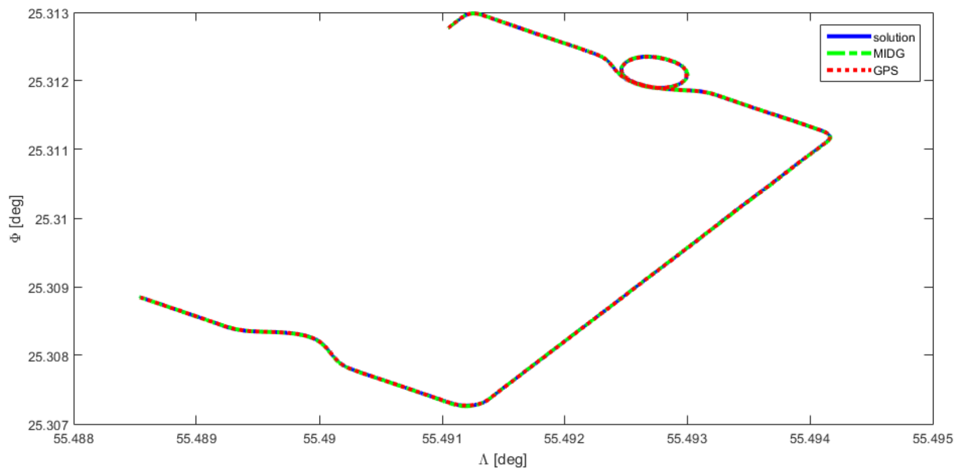


Figure 18: Path estimate of the Vehicle, with using INS/GPS fusion based on EKF in n-frame.

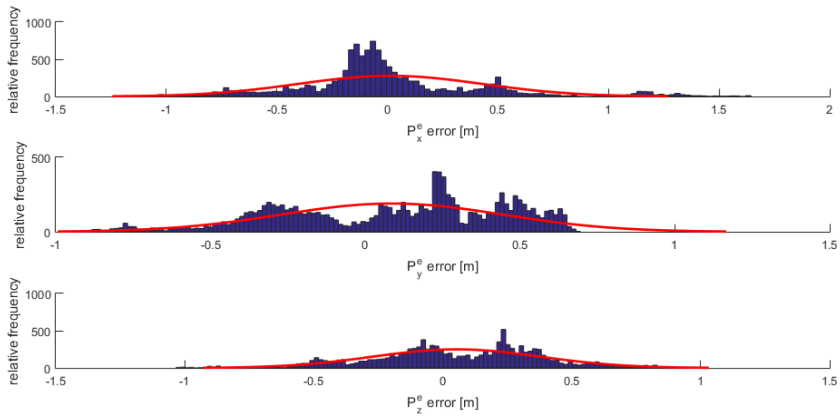


Figure 19: Position error distribution for INS/GPS fusion in n-frame.

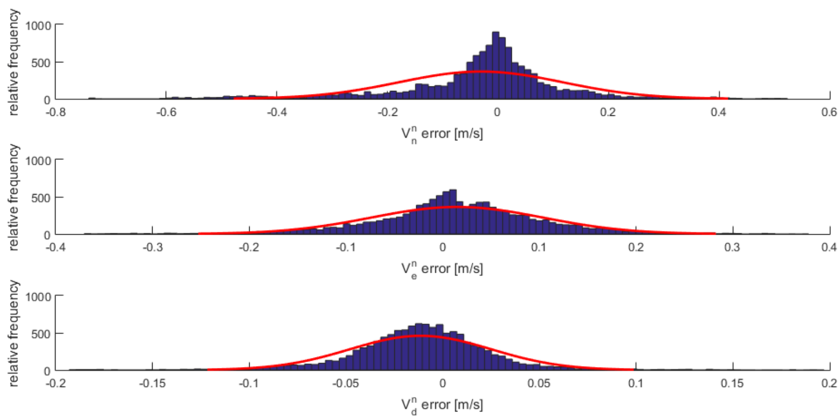


Figure 20: Velocity error distribution for INS/GPS fusion in n-frame.

5.2. Experimental Setup

The designing and building process of the experimental setup is done in collaboration with undergraduate group of students, who are investigating the problem of detecting cracks and leakage in the pipe, as a senior design project. More details about the experimental setup are demonstrated in the following subsections.

5.2.1. Pipe line design.

A transparent acrylic pipe, with 7.75" diameter and 4 *m* length , has been assembled as shown in Figure 21. Different shapes of the the pipe have been used to validate our algorithms.

(a) Straight pipe without elbow. (b) Straight pipe with one elbow.

(c) Straight pipe with two elbows.

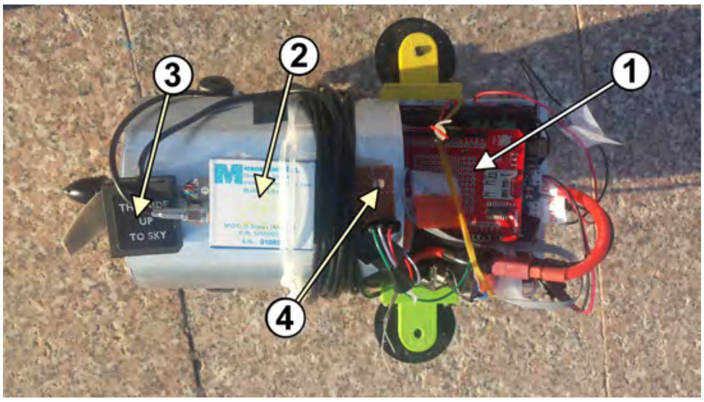
Figure 21: Different used pipe shapes in the experimental work.

5.2.2. Robot model design.

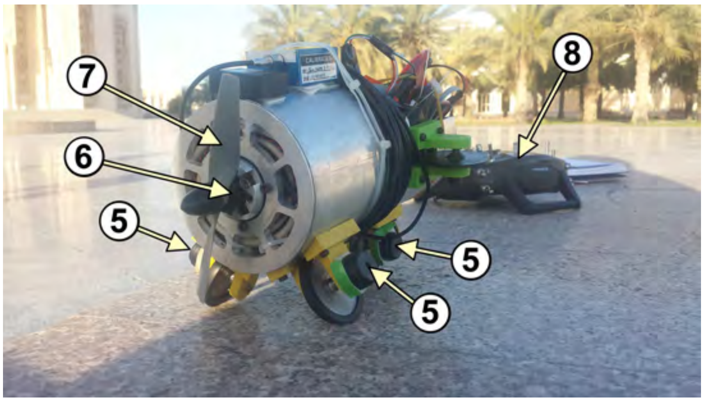
The main components for the in-pipe robot prototype, shown in Figure 22, are:

1. Arduino Due
2. MIDG
3. GPS Antenna
4. Light Sensor
5. Quadrature Encoders
6. Brushless Outrunner Motor
7. Propeller
8. Radio Remote Control (Futaba 8JH 2.4GHZ)

In the experimental test, Arduino due reads the data coming from the sensors and stores it in the SD card. The motor with the propeller drive the robot through the pipe and it is controlled by electronic speed controller which is connected to radio receiver linked with radio remote control.



(a) Top view.



(b) Back view.

Figure 22: In-pipe robot prototype.

5.3. Experimental Results

The first step in the experimental procedure is to bring out the setup outside to an open area because we need the GPS signal to get the initial position, velocity, and attitude. Inside the pipe, we are using the encoder's velocity and the detected pipe length as measurements to correct the INS solution. For detected pipe length measurement, we are using the light sensor to detect the rings shown in Figure 21 and the elbows' outlets and inlets. The distance between each of the all detected features is known and it is used to correct the INS solution. For validation purposes, we are simulating the crack by using a small piece of dark cloth fixed on the top of the pipe to show how our proposed algorithms are accurate in the localization of the detected cracks.

For comparison purposes, the proposed algorithms were applied for two different scenarios. In the first scenario, only the encoder measurement was used to correct the INS solution and there was no measurement used to correct the RISS solution. In the second scenario, the encoder's velocity measurement and the detected pipe length measurement were used to correct the INS solution and only the detected pipe length measurement was used to correct the RISS solution.

5.3.1. First scenario.

5.3.1.1. Experimental results for straight pipe without elbows.

The performance of the proposed navigation algorithms for straight pipe without elbow, for first scenario, is presented in Figures 23-25 and Tables 1-3. The performance of the three algorithms is very similar and the average absolute error is around ± 4 *cm* for all.

INS integration in e-frame.

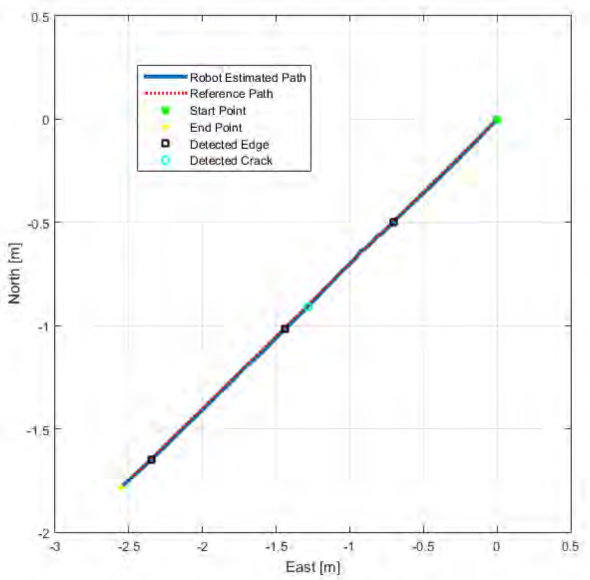


Figure 23: Path estimate of the robot for straight pipe without elbow, derived in e-frame, first scenario.

Table 1: Real length vs. measured length for each part of the pipe without elbow, derived in e-frame, first scenario.

	True Length	Measured Length	Abs. Error
Start edge to edge1	0.82 m	0.86 m	3.98 cm
Edge1 to Edge2	0.86 m	0.901 m	4.1 cm
Edge2 to Edge3	1.09 m	1.116 m	2.6 cm
Edge3 to End Edge	0.24 m	0.242 m	0.19 cm
Start to Crack	1.5 m	1.577 m	7.75 cm
Average Abs. Error			3.73 cm

INS integration in n-frame.

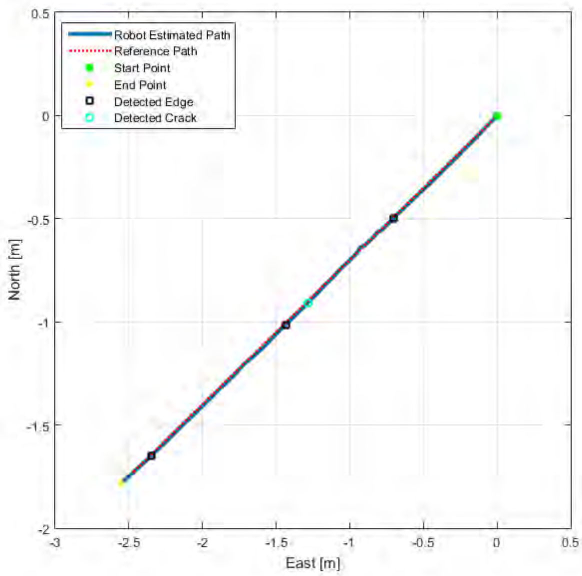


Figure 24: Path estimate of the robot for straight pipe without elbow, derived in n-frame, first scenario.

Table 2: Real length vs. measured length for each part of the pipe without elbow, derived in n-frame, first scenario.

	True Length	Measured Length	Abs. Error
Start Edge to Edge1	0.82 m	0.859 m	3.93 cm
Edge1 to Edge2	0.86 m	0.901 m	4.09 cm
Edge2 to Edge3	1.09 m	1.116 m	2.6 cm
Edge3 to End Edge	0.24 m	0.242 m	0.17 cm
Start to Crack	1.5 m	1.577 m	7.69 cm
Average Abs. Error			3.70 cm

INS integration for RISS.

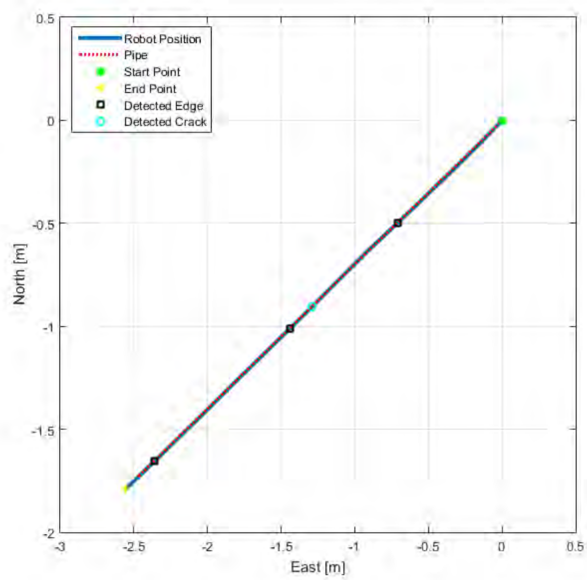


Figure 25: Path estimate of the robot for straight pipe without elbow, given by RISS method, first scenario.

Table 3: Real length vs. measured length for each part of the pipe without elbow, given by RISS method, first scenario.

	True Length	Measured Length	Abs. Error
Start Edge to Edge1	0.82 m	0.864 m	4.35 cm
Edge1 to Edge2	0.86 m	0.878 m	3.78 cm
Edge2 to Edge3	1.09 m	1.119 m	2.95 cm
Edge3 to End Edge	0.24 m	0.243 m	0.342 cm
Start to Crack	1.5 m	1.579 m	7.90 cm
Average Abs. Error			3.87 cm

5.3.1.2. Experimental results for straight pipe with one elbow.

The performance of the proposed navigation algorithms for straight pipe with one elbow, for first scenario, is presented in Figures 26-28 and Tables 4-6. The performance of the three algorithms is very similar and the average absolute error is around ± 7 cm for all.

INS Integration in e-frame.

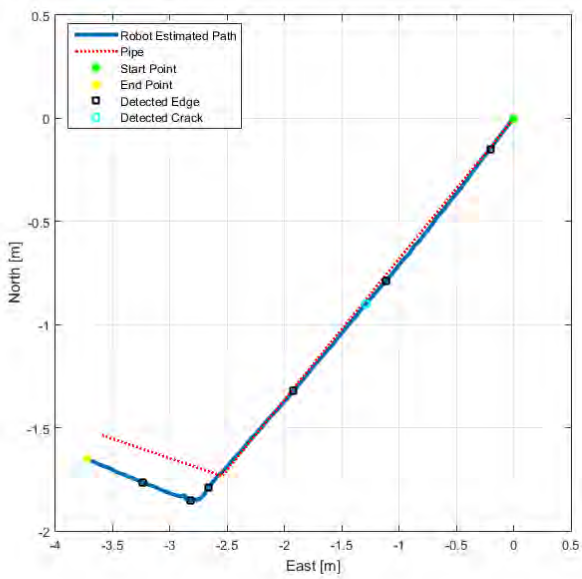


Figure 26: Path estimate of the robot for straight pipe with one elbow, derived in e-frame, first scenario.

Table 4: Real length vs. measured length for each part of the pipe with one elbow, derived in e-frame, first scenario.

	True Length	Measured Length	Abs. Error
Start Edge to Edge1	0.24 m	0.252 m	1.17 cm
Edge1 to Edge2	1.09 m	1.17 m	8.04 cm
Edge2 to Edge3	0.86 m	0.973 m	11.29 cm
Edge3 to Elbow Entrance	0.78 m	0.88 m	9.95 cm
Elbow Exit to Edge4	0.47 m	0.436 m	3.41 cm
Edge4 to End Edge	0.475 m	0.507 m	3.21 cm
Start Edge to Crack	1.5 m	1.630 m	12.98 cm
Average Abs. Error			7.15 cm

INS integration in n-frame.

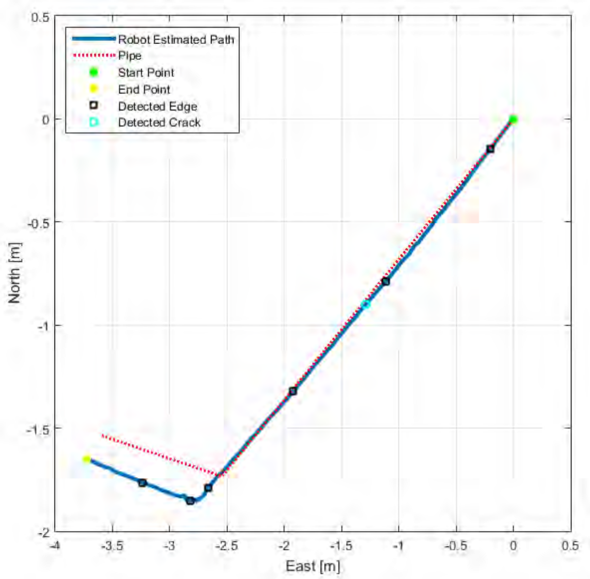


Figure 27: Path estimate of the robot for straight pipe with one elbow, derived in n-frame, first scenario.

Table 5: Real length vs. measured length for each part of the pipe with one elbow, derived in n-frame, first scenario.

	True Length	Measured Length	Abs. Error
Start Edge to Edge1	0.24 m	0.252 m	1.17 cm
Edge1 to Edge2	1.09 m	1.17 m	7.99 cm
Edge2 to Edge3	0.86 m	0.972 m	11.23 cm
Edge3 to Elbow Entrance	0.78 m	0.88 m	9.92 cm
Elbow Exit to Edge4	0.47 m	0.435 m	3.47 cm
Edge4 to End Edge	0.475 m	0.506 m	3.11 cm
Start Edge to Crack	1.5 m	1.63 m	12.92 cm
Average Abs. Error			7.12 cm

INS Integration for RISS.

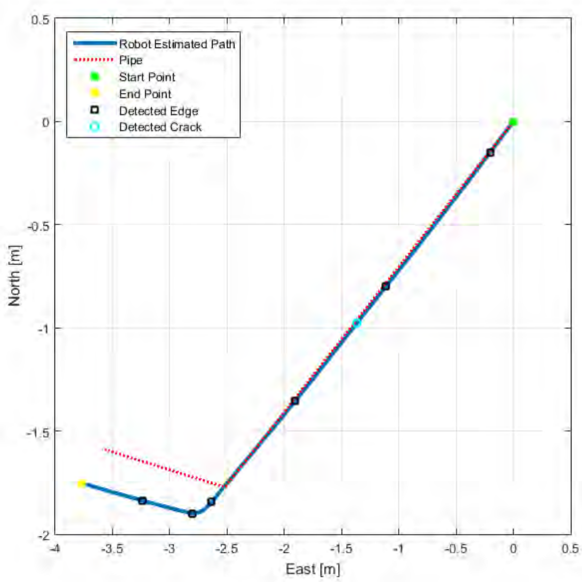


Figure 28: Path estimate of the robot for straight pipe with one elbow, given by RISS method, first scenario.

Table 6: Real length vs. measured length for each part of the pipe with one elbow, given by RISS method, first scenario.

	True Length	Measured Length	Abs. Error
Start Edge to Edge1	0.24 *m*	0.251 *m*	1.11 *cm*
Edge1 to Edge2	1.09 *m*	1.171 *m*	8.12 *cm*
Edge2 to Edge3	0.86 *m*	0.969 *m*	10.86 *cm*
Edge3 to Elbow Entrance	0.78 *m*	0.828 *m*	4.84 *cm*
Elbow Exit to Edge4	0.47 *m*	0.44 *m*	2.96 *cm*
Edge4 to End Edge	0.475 *m*	0.539 *m*	6.42 *cm*
Start Edge to Crack	1.5 *m*	1.681 *m*	18.10 *cm*
Average Abs. Error			7.48 *cm*

5.3.1.3. Experimental results for straight pipe with two elbows.

The performance of the proposed navigation algorithms for straight pipe with two elbows is presented in Figures 29-31 and Tables 7-9. This performance is consistent with previous discussed results and the average absolute error is around ± 9 *cm* for all.

INS integration in e-frame.

Table 7: Real length vs. measured length for each part of the pipe with two elbows, derived in e-frame, first scenario.

	True Length	Measured Length	Abs. Error
Start Edge to Edge1	0.82 *m*	1.031 *m*	21.14 *cm*
Edge1 to Edge2	0.86 *m*	0.828 *m*	3.20 *cm*
Edge2 to Elbow Entrance	0.24 *m*	0.177 *m*	6.33 *cm*
Elbow Exit to Edge3	0.44 *m*	0.523 *m*	8.30 *cm*
Edge3 to Elbow Entrance	0.445 *m*	0.387 *m*	5.77 *cm*
Elbow Exit to Edge4	0.45 *m*	0.436 *m*	1.36 *cm*
Edge4 to End Edge	0.5 *m*	0.551 *m*	5.06 *cm*
Start Edge to Crack	1.5 *m*	1.684 *m*	18.36 *cm*
Average Abs. Error			8.69 *cm*

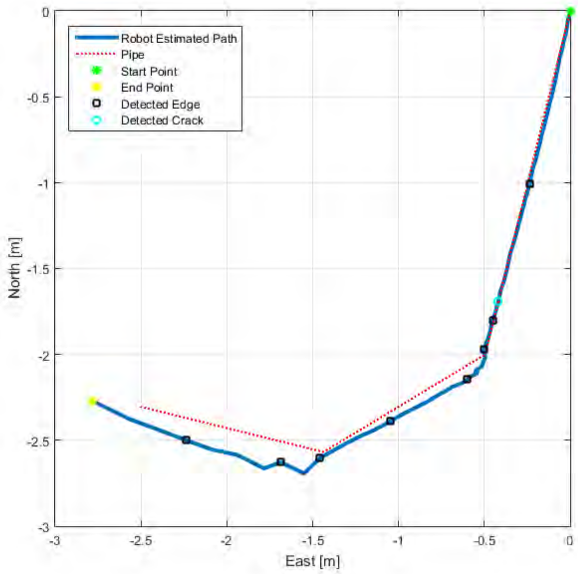


Figure 29: Path estimate of the robot for straight pipe with two elbows, derived in e-frame, first scenario.

INS integration in n-frame.

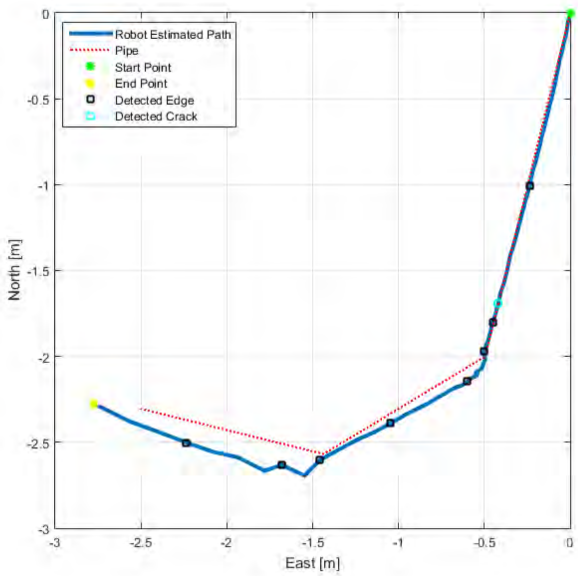


Figure 30: Path estimate of the robot for straight pipe with two elbows, derived in n-frame, first scenario.

Table 8: Real length vs. measured length for each part of the pipe with two elbows, derived in n-frame, first scenario.

	True Length	Measured Length	Abs. Error
Start Edge to Edge1	0.82 m	1.031 m	21.15 cm
Edge1 to Edge2	0.86 m	0.828 m	3.21 cm
Edge2 to Elbow Entrance	0.24 m	0.177 m	6.34 cm
Elbow Exit to Edge3	0.44 m	0.523 m	8.29 cm
Edge3 to Elbow Entrance	0.445 m	0.388 m	5.74 cm
Elbow Exit to Edge4	0.45 m	0.436 m	1.40 cm
Edge4 to End Edge	0.5 m	0.549 m	4.85 cm
Start Edge to Crack	1.5 m	1.684 m	18.36 cm
Average Abs. Error			8.67 cm

INS integration for RISS.

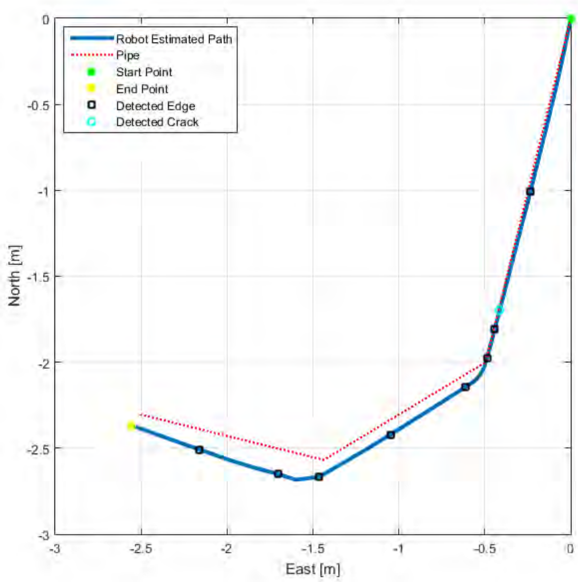


Figure 31: Path estimate of the robot for straight pipe with two elbows, given by RISS method, first scenario.

Table 9: Real length vs. measured length for each part of the pipe with two elbows, given by RISS method, first scenario.

	True Length	Measured Length	Abs. Error
Start Edge to Edge1	0.82 m	1.033 m	21.26 cm
Edge1 to Edge2	0.86 m	0.828 m	3.18 cm
Edge2 to Elbow Entrance	0.24 m	0.176 m	5.43 cm
Elbow Exit to Edge3	0.44 m	0.522 m	8.17 cm
Edge3 to Elbow Entrance	0.445 m	0.414 m	3.06 cm
Elbow Exit to Edge4	0.45 m	0.483 m	3.26 cm
Edge4 to End Edge	0.5 m	0.408 m	9.17 cm
Start Edge to Crack	1.5 m	1.684 m	18.39 cm
Average Abs. Error			8.99 cm

5.3.2. Second scenario.

5.3.2.1. Experimental results for straight pipe without elbows.

The performance of the proposed navigation algorithms for straight pipe without elbow, for second scenario, is presented in Figures 32-34 and Tables 10-12. The performance of the three algorithms is very similar and the average absolute error is around ± 2 cm for all.

INS integration in e-frame.

Table 10: Real length vs. measured length for each part of the pipe without elbows, derived in e-frame, second scenario.

	True Length	Measured Length	Abs. Error
Start edge to edge1	0.82 m	0.82 m	0.04 cm
Edge1 to Edge2	0.86 m	0.879 m	1.94 cm
Edge2 to Edge3	1.09 m	1.105 m	1.52 cm
Edge3 to End Edge	0.24 m	0.243 m	0.26 cm
Start to Crack	1.5 m	1.541 m	4.06 cm
Average Abs. Error			1.57 cm

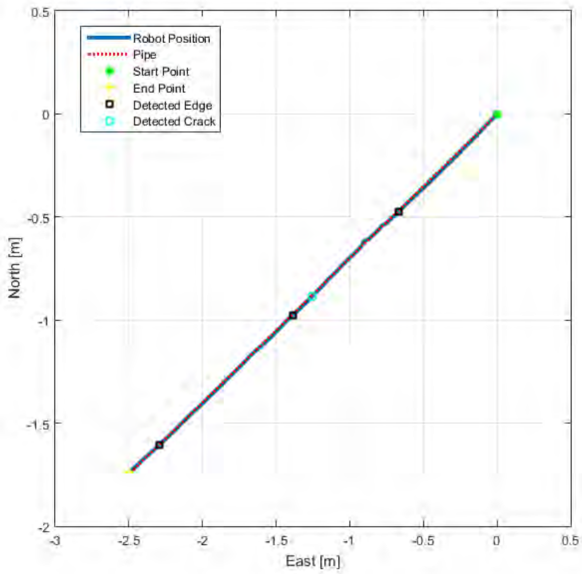


Figure 32: Path estimate of the robot for straight pipe without elbows, derived in e-frame, second scenario.

INS integration in n-frame.

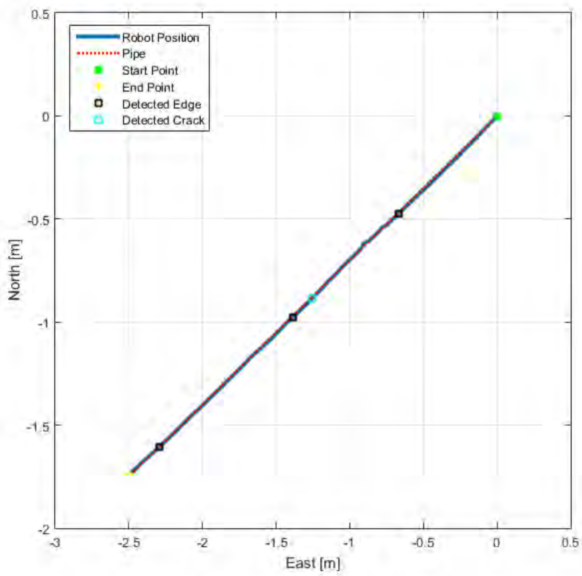


Figure 33: Path estimate of the robot for straight pipe without elbows, derived in n-frame, second scenario.

Table 11: Real length vs. measured length for each part of the pipe without elbows, derived in n-frame, second scenario.

	True Length	Measured Length	Abs. Error
Start Edge to Edge1	0.82 m	0.82 m	0.04 cm
Edge1 to Edge2	0.86 m	0.879 m	1.93 cm
Edge2 to Edge3	1.09 m	1.105 m	1.52 cm
Edge3 to End Edge	0.24 m	0.242 m	0.25 cm
Start to Crack	1.5 m	1.541 m	4.05 cm
Average Abs. Error			1.56 cm

INS integration for RISS.

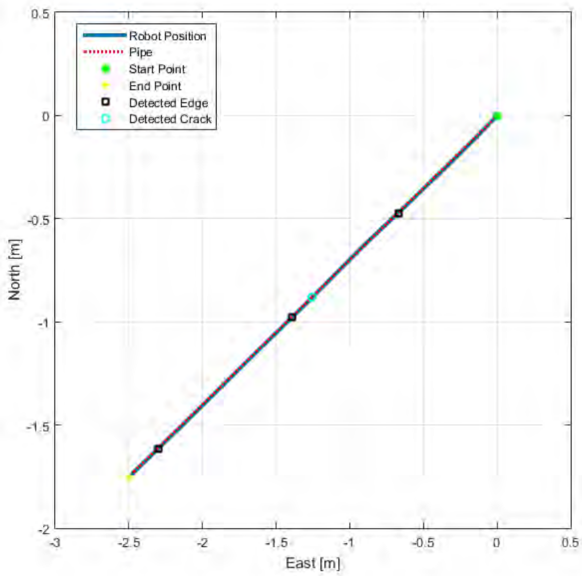


Figure 34: Path estimate of the robot for straight pipe without elbows, given by RISS method, second scenario.

Table 12: Real length vs. measured length for each part of the pipe without elbows, given by RISS method, second scenario.

	True Length	Measured Length	Abs. Error
Start Edge to Edge1	0.82 m	0.82 m	0.04 cm
Edge1 to Edge2	0.86 m	0.879 m	1.89 cm
Edge2 to Edge3	1.09 m	1.109 m	1.93 cm
Edge3 to End Edge	0.24 m	0.242 m	0.23 cm
Start to Crack	1.5 m	1.536 m	3.59 cm
Average Abs. Error			1.54 cm

5.3.2.2. Experimental results for straight pipe with one elbow.

The performance of the proposed navigation algorithms for straight pipe with one elbow, for second scenario, is presented in Figures 35-37 and Tables 13-15. This performance is consistent with previous discussed results.

INS integration in e-frame.

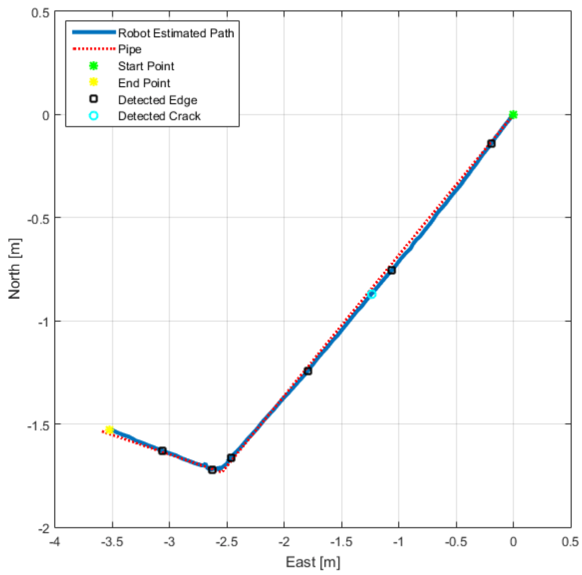


Figure 35: Path estimate of the robot for straight pipe with one elbow, derived in e-frame, second scenario.

Table 13: Real length vs. measured length for each part of the pipe with one elbow, derived in e-frame, second scenario.

	True Length	Measured Length	Abs. Error
Start Edge to Edge1	0.24 m	0.24 m	0.001 cm
Edge1 to Edge2	1.09 m	1.119 m	2.87 cm
Edge2 to Edge3	0.86 m	0.884 m	2.42 cm
Edge3 to Elbow Entrance	0.78 m	0.796 m	1.64 cm
Elbow Exit to Edge4	0.47 m	0.451 m	1.86 cm
Edge4 to End Edge	0.475 m	0.479 m	0.434 cm
Start Edge to Crack	1.5 m	1.569 m	6.89 cm
Average Abs. Error			2.30 cm

INS integration in n-frame.

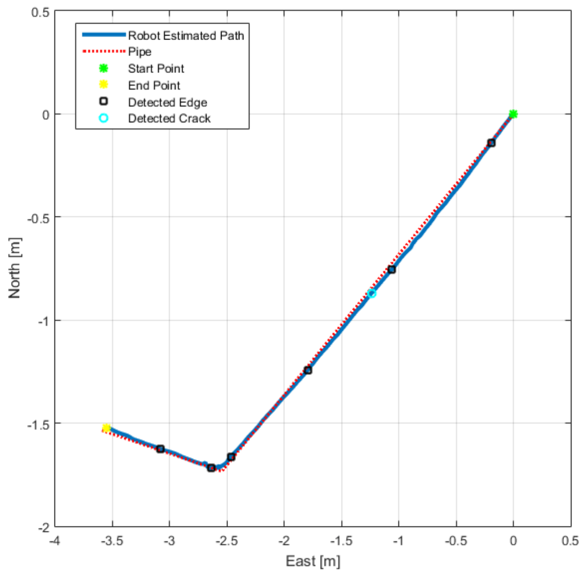


Figure 36: Path estimate of the robot for straight pipe with one elbow, derived in n-frame, second scenario.

Table 14: Real length vs. measured length for each part of the pipe with one elbow, derived in n-frame, second scenario.

	True Length	Measured Length	Abs. Error
Start Edge to Edge1	0.24 m	0.24 m	0.001 cm
Edge1 to Edge2	1.09 m	1.119 m	2.87 cm
Edge2 to Edge3	0.86 m	0.884 m	2.43 cm
Edge3 to Elbow Entrance	0.78 m	0.796 m	1.63 cm
Elbow Exit to Edge4	0.47 m	0.465 m	0.45 cm
Edge4 to End Edge	0.475 m	0.486 m	1.08 cm
Start Edge to Crack	1.5 m	1.569 m	6.91 cm
Average Abs. Error			2.20 cm

INS integration for RISS.

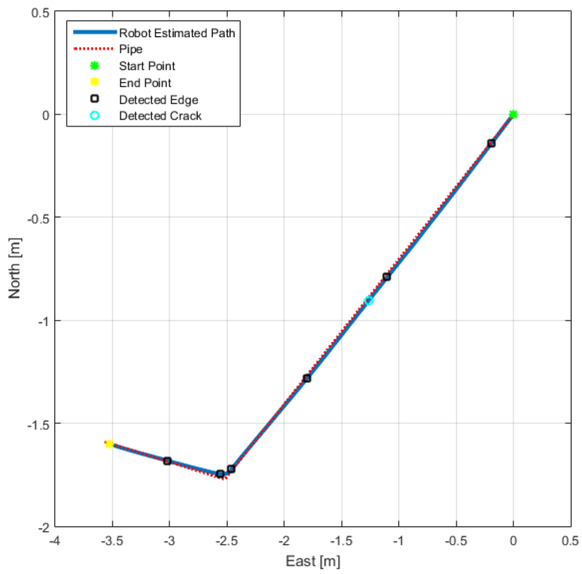


Figure 37: Path estimate of the robot for straight pipe with one elbow, given by RISS method, second scenario.

Table 15: Real length vs. measured length for each part of the pipe with one elbow, given by RISS method, second scenario.

	True Length	Measured Length	Abs. Error
Start Edge to Edge1	0.24 *m*	0.24 *m*	0.00 *cm*
Edge1 to Edge2	1.09 *m*	1.092 *m*	0.22 *cm*
Edge2 to Edge3	0.86 *m*	0.876 *m*	1.61 *cm*
Edge3 to Elbow Entrance	0.78 *m*	0.799 *m*	1.87 *cm*
Elbow Exit to Edge4	0.47 *m*	0.463 *m*	0.70 *cm*
Edge4 to End Edge	0.475 *m*	0.519 *m*	4.4 *cm*
Start Edge to Crack	1.5 *m*	1.554 *m*	5.4 *cm*
Average Abs. Error			2.03 *cm*

5.3.2.3. Experimental results for straight pipe with two elbows.

The performance of the proposed navigation algorithms for straight pipe with two elbows, for second scenario, is presented in Figures 38-40 and Tables 16-18. This performance is consistent with previous discussed results.

INS Integration in e-frame.

Table 16: Real length vs. measured length for each part of the pipe with two elbows, derived in e-frame, second scenario.

	True Length	Measured Length	Abs. Error
Start Edge to Edge1	0.82 *m*	0.82 *m*	0.026 *cm*
Edge1 to Edge2	0.86 *m*	0.889 *m*	2.94 *cm*
Edge2 to Elbow Entrance	0.24 *m*	0.214 *m*	2.62 *cm*
Elbow Exit to Edge3	0.44 *m*	0.448 *m*	0.754 *cm*
Edge3 to Elbow Entrance	0.445 *m*	0.393 *m*	5.20 *cm*
Elbow Exit to Edge4	0.45 *m*	0.454 *m*	0.35 *cm*
Edge4 to End Edge	0.5 *m*	0.536 *m*	3.55 *cm*
Start Edge to Crack	1.5 *m*	1.472 *m*	2.79 *cm*
Average Abs. Error			2.28 *cm*

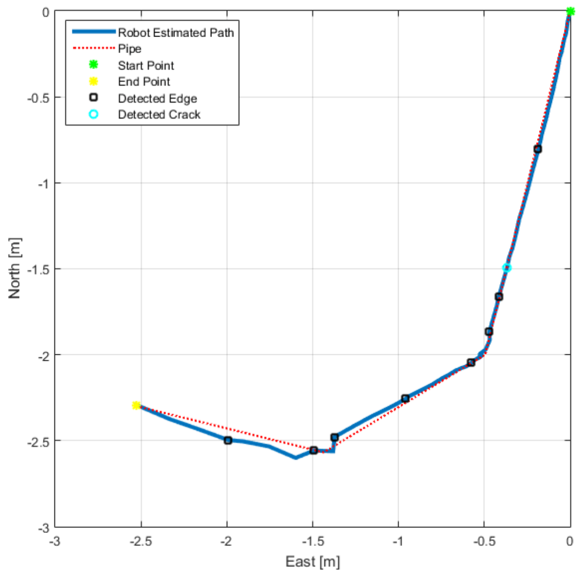


Figure 38: Path estimate of the robot for straight pipe with two elbows, derived in e-frame, second scenario.

INS integration in n-frame.

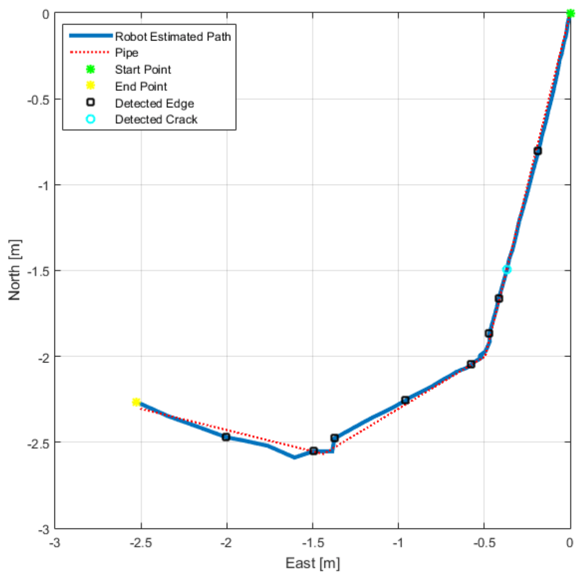


Figure 39: Path estimate of the robot for straight pipe with two elbows, derived in n-frame, second scenario.

Table 17: Real length vs. measured length for each part of the pipe with two elbows, derived in n-frame, second scenario.

	True Length	Measured Length	Abs. Error
Start Edge to Edge1	0.82 *m*	0.82 *m*	0.026 *cm*
Edge1 to Edge2	0.86 *m*	0.889 *m*	2.93 *cm*
Edge2 to Elbow Entrance	0.24 *m*	0.214 *m*	2.63 *cm*
Elbow Exit to Edge3	0.44 *m*	0.448 *m*	0.76 *cm*
Edge3 to Elbow Entrance	0.445 *m*	0.394 *m*	5.11 *cm*
Elbow Exit to Edge4	0.45 *m*	0.456 *m*	0.57 *cm*
Edge4 to End Edge	0.5 *m*	0.531 *m*	3.06 *cm*
Start Edge to Crack	1.5 *m*	1.472 *m*	2.80 *cm*
Average Abs. Error			2.24 *cm*

INS integration for RISS.

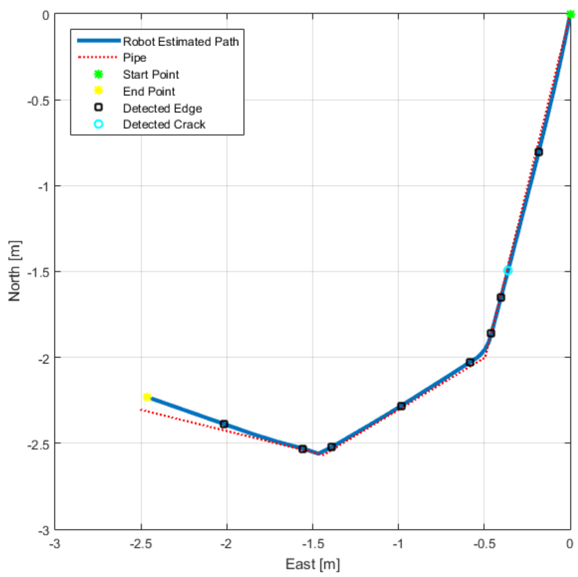


Figure 40: Path estimate of the robot for straight pipe with two elbows, given by RISS method, second scenario.

Table 18: Real length vs. measured length for each part of the pipe with two elbows, given by RISS method, second scenario.

	True Length	Measured Length	Abs. Error
Start Edge to Edge1	0.82 m	0.82 m	0.026 cm
Edge1 to Edge2	0.86 m	0.879 m	2.93 cm
Edge2 to Elbow Entrance	0.24 m	0.215 m	1.47 cm
Elbow Exit to Edge3	0.44 m	0.48 m	3.98 cm
Edge3 to Elbow Entrance	0.445 m	0.414 m	3.06 cm
Elbow Exit to Edge4	0.45 m	0.483 m	3.26 cm
Edge4 to End Edge	0.5 m	0.466 m	3.36 cm
Start Edge to Crack	1.5 m	1.47 m	2.85 cm
Average Abs. Error			2.49 cm

5.3.3. Summary.

To show the effect of using the detected pipe length measurement to correct the INS or RISS solution, the average absolute error of the first scenario vs. the average absolute error of the second scenario, for the three pipe shapes, is shown in Tables 19-21. By using the detected pipe length measurement in addition to encoder's velocity to correct the INS or RISS solution, the absolute error decreased by 50% for straight pipe without elbows as shown in Table 19, the absolute error decreased by 70% for straight pipe with one elbow as shown in Table 20, and the absolute error decreased by 75% for straight pipe with two elbows as shown in Table 21.

Table 19: Average abs. error of the first scenario vs. average abs. error of the second scenario for straight pipe without elbows.

	First Scenario	Second Scenario
INS derived in e-frame	3.73 cm	1.57 cm
INS derived in n-frame	3.70 cm	1.56 cm
RISS	3.87 cm	1.54 cm

Table 20: Average abs. error of the first scenario vs. average abs. error of the second scenario for straight pipe with one elbow.

	First Scenario	Second Scenario
INS derived in e-frame	7.15 *cm*	2.3 *cm*
INS derived in n-frame	7.12 *cm*	2.2 *cm*
RISS	7.48 *cm*	2.03 *cm*

Table 21: Average abs. error of the first scenario vs. average abs. error of the second scenario for straight pipe with two elbows.

	First Scenario	Second Scenario
INS derived in e-frame	8.69 *cm*	2.28 *cm*
INS derived in n-frame	8.67 *cm*	2.24 *cm*
RISS	8.99 *cm*	2.49 *cm*

Chapter 6: Conclusion and Future Work

6.1. Summary and Conclusion

Three navigation algorithms based on extended Kalman filter, to fuse INS with two proposed aided systems, have been investigated and implemented successfully for in-pipe inspection robot and the accuracy of the proposed algorithms was around ± 3 *cm* after sensor fusion. The kinematics equations and the error model state equation have been derived in e-frame in the first algorithm and derived in n-frame in the second algorithm. Instead of using the measurements from the three accelerometers and the three gyroscopes in the IMU in the former two algorithms, 3D RISS uses measurements from the vertically aligned gyroscope and the two horizontal accelerometers with speed readings provided by wheel encoder, to implement the mechanization process.

Because the building and designing of the in-pipe robot prototype was still in progress, an experimental test has been conducted to verify the performance of the proposed algorithms by driving a car around the campus with applying extended Kalman filter to fuse INS with GPS.

Due to the blockage of the GPS signal inside the pipe, so it can not be used in our application. Detection of the edges between each two pieces of the pipe and encoder-derived velocity with zero velocity constraint are used to correct the drift in the navigation state of the robot obtained from the INS mechanization.

The sensor fusion is based EKF, which corresponds to a closed-loop filter configuration. The 15 estimated error states, by regular Kalman filter based on the error between the INS and the aided systems measurements, are fed back to the inertial system to correct its output, so the INS errors remain small, and the linearity assumption required for the KF technique is supported. In the first two algorithms, the 15 estimated error states are the position error in x, y and z-axes, velocity error in x, y and z-axes, quaternions error in x, y and z-axes, accelerometers bias drift in x, y and z-axes, and finally, rate gyro bias drifts in x, y and z-axes, whereas 3D RISS has 9 estimated error states, which are the position error in x, y and z-axes, velocity error in x, y and z-axes, the error in the yaw angle, the error of the encoder-derived speed, and the stochastic bias drift in the vertical gyroscope.

The proposed fusion algorithms of INS with detected pipe lengths and encoder derived speed measurements have been tested and validated in a real pipe by using in-pipe robot designed and made manually for this purpose.

6.2. Future Work

This thesis work represents only the start, so, as a continuation of this project, it is advised that more effort should be put into improving the algorithm performance and implementation. This can be achieved by testing our algorithms in the field. The use of more robust in-pipe robot design equipped with high-performance IMU and sensors to detect the pipe edges and measure the derived encoder velocity should be considered in order to be developed into a commercial product.

Also, we need to apply more effort on the tuning of the process and the measurements covariance. The use of AI methods, like Neural Network (NN) and Adaptive Neuro-Fuzzy Inference Systems (ANFIS), to set these covariance parameters is highly advisable to be considered. Also, AI can be considered instead of mathematical approaches for states estimation, or a hybrid system that combines AI and EKF method.

References

[1] P. D. Groves, *Principles of GNSS, inertial, and multi-sensor integrated navigation systems.* Artech House, 2007.

[2] L. R. Sahawneh *et al.*, "Real-time implementation of GPS aided low-cost strapdown inertial navigation system," *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1, pp. 527–544, 2010. [Online]. Available: http://dx.doi.org/10.1007/s10846-010-9501-0

[3] M. F. Abdel-Hafez, K. Saadeddin, and M. A. Jarrah, "Constrained low-cost GPS/INS filter with encoder bias estimation for ground vehicles applications," *Mechanical Systems and Signal Processing*, vol. 58-59, pp. 285–297, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0888327014005019

[4] J. Georgy *et al.*, "Nonlinear filtering for tightly coupled RISS/GPS integration," in *2010 IEEE/ION Position Location and Navigation Symposium (PLANS)*, May 2010, pp. 1014–1021.

[5] N. El-Sheimy, K. wei Chiang, and A. Noureldin, "The utilization of artificial neural networks for multisensor system integration in navigation and positioning instruments," *IEEE Transactions on Instrumentation and Measurement*, vol. 55, no. 5, pp. 1606–1615, Oct 2006.

[6] A. Noureldin *et al.*, "Performance enhancement of MEMS-based INS/GPS integration for low-cost navigation applications," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 3, pp. 1077–1096, March 2009.

[7] J. Georgy *et al.*, "Enhanced MEMS-IMU/odometer/GPS integration using mixture particle filter," *GPS Solutions*, vol. 15, no. 3, pp. 239–252, 2011. [Online]. Available: http://dx.doi.org/10.1007/s10291-010-0186-4

[8] J. Georgy *et al.*, "Low-cost three-dimensional navigation solution for RISS/GPS integration using mixture particle filter," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 2, pp. 599–615, Feb 2010.

[9] A. Noureldin, A. Osman, and N. El-Sheimy, "A neuro-wavelet method for multi-sensor system integration for vehicular navigation," *Measurement Science and Technology*, vol. 15, no. 2, p. 404, 2004. [Online]. Available: http://stacks.iop.org/0957-0233/15/i=2/a=013

[10] M. B. Aboelmagd Noureldina, Ahmed El-Shafieb, "GPS/INS integration utilizing dynamic neural networks for vehicular navigation," *Information Fusion*, vol. 12, no. 1, pp. 48 – 57, 2011, special Issue on Intelligent Transportation Systems. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1566253510000175

[11] Bu Hasa−Habshan gas pipeline. [Online]. Available: http://www.offshore-technology.com/contractors/design-engineering-construction/consolidated/consolidated4.html [Accessed: Apr 21, 2016].

[12] D. Chatzigeorgiou, K. Youcef-Toumi, and R. Ben-Mansour, "Design of a novel in-pipe reliable leak detector," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 2, pp. 824–833, April 2015.

[13] J.-H. Kim, G. Sharma, and S. Iyengar, "Famper: A fully autonomous mobile robot for pipeline exploration," in *2010 IEEE International Conference on Industrial Technology (ICIT)*, March 2010, pp. 517–523.

[14] H. Choi and S. Ryew, "Robotic system with active steering capability for internal inspection of urban gas pipelines," *Mechatronics*, vol. 12, no. 5, pp. 713 – 736, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957415801000228

[15] D. Krys and H. Najjaran, "Development of visual simultaneous localization and mapping (VSLAM) for a pipe inspection robot," in *2007 International Symposium on Computational Intelligence in Robotics and Automation*, June 2007, pp. 344–349.

[16] A. C. Murtra and J. M. M. Tur, "IMU and cable encoder data fusion for in-pipe mobile robot localization," in *2013 IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*, April 2013, pp. 1–6.

[17] D. Y. Kim *et al.*, "Artificial landmark for vision-based SLAM of water pipe rehabilitation robot," in *2015 12th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, Oct 2015, pp. 444–446.

[18] H. H. Aghdam *et al.*, "Localizing pipe inspection robot using visual odometry," in *2014 IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, Nov 2014, pp. 245–250.

[19] P. Hansen *et al.*, "Monocular visual odometry for robot localization in LNG pipes," in *2011 IEEE International Conference on Robotics and Automation (ICRA)*, May 2011, pp. 3111–3116.

[20] D. H. Lee, H. Moon, and H. R. Choi, "Landmark detection of in-pipe working robot using line-laser beam projection," in *2010 International Conference on Control Automation and Systems (ICCAS)*, Oct 2010, pp. 611–615.

[21] A. Yamashita *et al.*, "Self-localization and 3-D model construction of pipe by earthworm robot equipped with omni-directional rangefinder," in *2011 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dec 2011, pp. 1017–1023.

[22] K. M. Saadeddin, "Estimating vehicle state by GPS/IMU fusion with vehicle dynamics," MSc. thesis, American University of Sharjah, Sharjah, United Arab Emirates, January 2013.

[23] A. Lahrech, C. Boucher, and J. Noyer, “Accurate vehicle positioning in urban areas,” in *31st Annual Conference of IEEE Industrial Electronics Society, 2005. IECON 2005*, Nov 2005, pp. 486–490.

[24] S. Bonnabel and E. Salan, “Design and prototyping of a low-cost vehicle localization system with guaranteed convergence properties,” *Control Engineering Practice*, vol. 19, no. 6, pp. 591 – 601, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0967066111000335

[25] A. Brandt and J. Gardner, “Constrained navigation algorithms for strapdown inertial navigation systems with reduced set of sensors,” in *American Control Conference, 1998. Proceedings of the 1998*, vol. 3, Jun 1998, pp. 1848–1852 vol.3.

[26] G. Dissanayake *et al.*, “The aiding of a low-cost strapdown inertial measurement unit using vehicle model constraints for land vehicle applications,” *IEEE Transactions on Robotics and Automation*, vol. 17, no. 5, pp. 731–747, Oct 2001.

[27] S. Sadhu *et al.*, “Central difference formulation of risk-sensitive filter,” *IEEE Signal Processing Letters*, vol. 14, no. 6, pp. 421–424, June 2007.

[28] S. Peyman, K. Alireza, and F. Ebrahim, “Attitude estimation by divided difference filter-based sensor fusion,” *The Journal of Navigation*, vol. 60, pp. 119–128, 1 2007. [Online]. Available: http://journals.cambridge.org/article_S037346330600405X

[29] J. Rezaie *et al.*, “GPS/INS integration using nonlinear blending filters,” in *2007 Annual Conference SICE*, Sept 2007, pp. 1674–1680.

[30] Y. Zhang, F. Gao, and L. Tian, “INS/GPS integrated navigation for wheeled agricultural robot based on sigma-point Kalman filter,” in *Asia Simulation Conference - 7th International Conference on System Simulation and Scientific Computing, 2008. ICSC 2008.*, Oct 2008, pp. 1425–1431.

[31] K. Saadeddin *et al.*, “Optimization of intelligent-based approach for low-cost INS/GPS navigation system,” in *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, May 2013, pp. 668–677.

[32] M. A. K. Jaradat and M. F. Abdel-Hafez, “Enhanced, delay dependent, intelligent fusion for INS/GPS navigation system,” *IEEE Sensors Journal*, vol. 14, no. 5, pp. 1545–1554, May 2014.

[33] T. S. Abuhashim, M. F. Abdel-Hafez, and M. A. Al-Jarrah, “Building a robust integrity monitoring algorithm for a low cost GPS-aided-INS system,” *International Journal of Control, Automation and Systems*, vol. 8, no. 5, pp. 1108–1122, 2010. [Online]. Available: http://dx.doi.org/10.1007/s12555-010-0520-1

[34] A. Noureldin, T. Karamat, and J. Georgy, *Fundamentals of Inertial Navigation, Satellite-based Positioning and their Integration*. Springer Berlin Heidelberg, 2013. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-30466-8_1

[35] A. P. A. Mohinder S. Grewal, Lawrence R. Weill, *Global Positioning Systems, Inertial Navigation, and Integration, 2nd Edition*. American Institute of Aeronautics and Astronautics, Inc., 2007.

[36] X. Kong, "INS algorithm using quaternion model for low cost IMU," *Robotics and Autonomous Systems*, vol. 46, no. 4, pp. 221 – 246, 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889004000119

[37] A. D. King, "Inertial navigationforty years of evolution," *GEC Review*, vol. 13, no. 3, pp. 140 – 149, 1998.

[38] I. Skog and P. Handel, "In-car positioning and navigation technologies–a survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 1, pp. 4–21, March 2009.

[39] M. F. Abdel-Hafez, "The autocovariance least-squares technique for GPS measurement noise estimation," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 2, pp. 574–588, Feb 2010.

[40] I. Rhee, M. F. Abdel-Hafez, and J. L. Speyer, "Observability of an integrated GPS/INS during maneuvers," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, no. 2, pp. 526–535, April 2004.

[41] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation With Applications to Tracking and Navigation: Theory Algorithms and Software.* New York, NY, USA: Wiley, 2001.

[42] *MIDG II Display and Configuration Program.*, Microbotic, Inc, Winston-Salem, NC, 2007.

Vita

Wasim M. F. Al-Masri was born in Amman, Jordan. He received the B.S. degree in mechanical engineering with a minor in mechatronics from the Jordan University of Science and Technology, Irbid, Jordan, in 2014.

He joined the Mechatronics Program, American University of Sharjah, Sharjah, United Arab Emirates, in 2014, where he is currently a Graduate Research and Teaching Assistant. He got professional training in King Abdullah II Design and Development Bureau, Amman, in different departments, such as Workshops, Test and Evaluation Centre, Design and Development (CAD) Department, in 2013. His current research interests include inertial navigation system, sensor fusion, state estimation, control theory, robotics, unmanned vehicles, and mechatronics design.