TERRAIN REFERENCED NAVIGATION OF AN AIRCRAFT USING PARTICLE FILTER

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

BURAK TURAN

IN PARTIAL FULLFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
AEROSPACE ENGINEERING

AUGUST 2017

Approval of the thesis:

TERRAIN REFERENCED NAVIGATION OF AN AIRCRAFT USING PARTICLE FILTER

submitted by **BURAK TURAN** in partial fulfillment of the requirements for the degree of **Master of Science in Aerospace Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver _____
Dean**,** Graduate School of **Natural and Applied Sciences**

Prof. Dr. Ozan Tekinalp _____
Head of Department, **Aerospace Engineering**

Asst. Prof. Dr. Ali Türker Kutay _____
Supervisor, **Aerospace Engineering Dept., METU**

Examining Committee Members:

Assoc. Prof. Dr. İlkay Yavrucuk _____
Aerospace Engineering Dept., METU

Asst. Prof. Dr. Ali Türker Kutay _____
Aerospace Engineering Dept., METU

Prof. Dr. Kemal Leblebicioğlu _____
Electrical and Electronics Engineering Dept., METU

Assoc. Prof. Dr. Umut Orguner _____
Electrical and Electronics Engineering Dept., METU

Asst. Prof. Dr. Gökhan Soysal _____
Electrical and Electronics Engineering Dept., Ankara University

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Burak TURAN

Signature:

ABSTRACT

TERRAIN REFERENCED NAVIGATION OF AN AIRCRAFT USING PARTICLE FILTER

Turan, Burak

M.S., Department of Aerospace Engineering

Supervisor: Asst. Prof. Dr. Ali Türker Kutay

August 2017, 81 Pages

The need for Terrain Referenced Navigation (TRN) arises when Global Navigation Satellite System (GNSS) signals are unavailable. In recent years, research on the application of TRN to aerial and underwater vehicles has been increased rapidly with the developments in the accuracy of digital terrain elevation database (DTED). Since the land and sea floor profiles are inherently nonlinear, TRN becomes a nonlinear estimation problem. Because of the highly nonlinear and non-Gaussian problem, linear or linearized estimation techniques such as Kalman or Extended Kalman Filter (EKF) do not work properly for many terrain profiles. Hence, this thesis focuses on the particle filter (PF) for dealing with nonlinearities and different types of probability distributions even multi modal. Two different particle filter (PF) implementations are studied, Sequential Importance Sampling with effective resampling (SIS-R) and Sampling Importance Resampling (SIR). Both algorithms are tested for an aircraft sample scenario over a DTED map. Simulations with different number of particles, inertial measurement units (IMUs) having various error specifications are performed and investigated.

Keywords: Terrain referenced navigation, particle filter, nonlinear estimation, SIR, SIS, DTED

ÖZ

PARÇACIK FİLTRE İLE BİR HAVA ARACININ YERYÜZÜ REFERANSLI NAVİGASYONU

Turan, Burak

Yüksek Lisans, Havacılık ve Uzay Mühendisliği Bölümü

Tez Yöneticisi: Asst. Prof. Dr. Ali Türker Kutay

Ağustos 2017, 81 Sayfa

Yeryüzü referanslı navigasyona küresel navigasyon uydu sistemleri sinyallerinin bulunmadığı durumlarda ihtiyaç duyulmaktadır. Son yıllarda, hava ve sualtı araçlarına yeryüzü referanslı navigasyon uygulaması üzerine araştırmalar sayısal arazi yükseklik veritabanlarının daha doğru üretilmesiyle hızla artmaktadır. Kara yüzeyi ve deniz tabanının doğrusal olmayan yapısı yeryüzü referanslı navigasyonu doğrusal olmayan bir kestirim problemine dönüştürmektedir. Problemin doğrusal olmaması ve Gauss dağılım göstermemesi sebebiyle, Kalman ve genişletilmiş Kalman filtresi gibi doğrusal veya doğrusallaştırılmış kestirim yöntemleri çoğu arazi tipleri üzerinde düzgün çalışmamaktadır. Bu nedenle bu tez, doğrusal olmamayla ve farklı hatta çoklu türde olasılık dağılımlarıyla başaçıkabilmesi nedeniyle parçacık filtre üzerine odaklanmaktadır. Etkin Yeniörneklemeli Ardışık Öncelik Örnekleme ve Öncelik Yeniörneklemeli Örnekleme olmak üzere iki farklı parçacık filtre uygulaması yapılmıştır. Sayısal arazi verisi kullanılarak örnek bir uçak senaryosunda her iki algoritma da test edilmiştir. Farklı parçacık sayıları ve farklı hata değerlerine sahip ataletsel ölçüm birimleriyle benzetimler gerçekleştirilip incelenmiştir.

Anahtar Kelimeler: Yeryüzü referanslı konumlama, parçacık filtre, doğrusal olmayan kestirim, EYAÖÖ, ÖYÖ, SAYV

To my family and the Peaks…

ACKNOWLEDGMENTS

First of all, I would like to express my gratitude and thanks to my supervisor Asst. Prof. Dr. Ali Türker KUTAY who guided me throughout this work with his valuable comments, support, understanding and patience. I would also like to thank Asst. Prof. Dr. Gökhan SOYSAL for his help to grasp the estimation theory.

I would like to express my appreciation to my colleagues especially Lisan Ozan YAMAN, Bağış ALTINÖZ, Seyit TUNÇ and Hüseyin DUMAN for providing me a peaceful working environment and your helpful discussions, you made me realize the importance of breathful coffee breaks.

I am grateful to my teacher Suay ŞEKKELİ and my dear friend Çağrı GEZGÜÇ for believing in me and for their never ending supports throughout my life.

Finally yet importantly, I would like to express my special thanks to my parents Yüksel - Veyis TURAN, my wife Meltem DİRİM TURAN and my other family members for their sincere love and understanding.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS/SYMBOLS

ABBREVIATIONS

TRN	:	Terrain Referenced Navigation
GNSS	:	Global Navigation Satellite Systems
INS	:	Inertial Navigation System
IMU	:	Inertial Measurement Unit
DTED	:	Digital Terrain Elevation Data
EKF	:	Extended Kalman Filter
UKF	:	Unscented Kalman Filter
PMF	:	Point Mass Filter
PF	:	Particle Filter
SIR	:	Sampling Importance Resampling
SIS-R	:	Sequential Importance Sampling with effective Resampling
NED	:	North-East-Down
MEMS	:	Microelectromechanical system
FOG	:	Fiber Optic Gyroscope
IFOG	:	Interferromagnetic Fiber Optic Gyroscope
RLG	:	Ring Laser Gyroscope
WGS	:	World Geodetic System
DCM	:	Direction cosine matrix
GLONASS	:	Globalnaya Navigatsionnaya Sputnikovaya Sistema
GPS	:	Global Positioning System
MSL	:	Mean Sea Level
UAV	:	Unmanned Air Vehicles
RMS	:	Root Mean Square
SITAN	:	Sandia Inertial Terrain Aided Navigation
pdf	:	probability density function
mg	:	milli-g (gravity)

mrad	:	milli-radian
deg:	:	degree
s	:	second
baro	:	barometer
hr	:	hour
m	:	meter
rad	:	radian
ppm	:	parts per million

SYMBOLS

b^a	:	Accelerometer bias error
b^g	:	Gyroscope bias error
C_b^n	:	Transformation matrix from b-frame to n-frame
C_e^n	:	Transformation matrix from e-frame to n-frame
$d\,°, m\,', s\,''$	:	*d* degrees, *m* arcminutes and *s* arcseconds
$\delta(.)$	:	Error state or Dirac delta function
δz	:	Measurement innovation
$E\{.\}$	:	Expectation operator
e_t	:	Measurement noise at time *t*
e	:	Eccentricity of the ellipsoid
f	:	Flattening of the ellipsoid
f^b	:	Reference specific force in b-frame
$\tilde{f}^b$	:	Accelerometer measurements in b-frame
$F(t), F_k$	:	Continuous-time state transition matrix
$F_{(..)}$	:	Component of state transition matrix
$G(t), G_k$	:	Continuous-time noise to state matrix
$G(k+1,k)$	:	Discrete-time noise to state matrix from step k to $k+1$
$g_0(L)$	:	Gravity at the ellipsoid surface
$g(L,h)$	:	Gravitation scaled wrt. altitude
g_l^n	:	Local gravity vector

$h(.,.)$	:	Terrain elevation database
Hz	:	Hertz
L, λ, h	:	Latitude, longitude and altitude
$M_{..}^{a}$	:	Accelerometer misalignment error
$M_{..}^{g}$	:	Gyroscope misalignment error
N	:	Number of particles
N_{eff}, N_{th}	:	Effective and threshold number of particles
$N(m, \sigma^2)$	:	Normal probability density with mean m and variance σ^2
ω_{ie}	:	Earth rotation rate
ω_{en}^{n}	:	Rotation vector of n-frame relative to i-frame expressed in n-frame
$\omega_{ib}^{b}, \widetilde{\omega}_{ib}^{b}$	:	Rotation of b-frame relative to i-frame expressed in b-frame as a vector and a matrix respectively
$\omega_{in}^{n}, \widetilde{\omega}_{in}^{n}$	:	Rotation of n-frame relative to i-frame expressed in n-frame as a vector and a matrix respectively
$p(a)$	:	Probability density function for the random variable a
$p(a\|b)$	:	Conditional probability density function for a given b
$\Pr\{.\}$	:	Discrete probability function
$P_{t\|t}$	:	Covariance of estimate $\hat{x}_{t\|t}$ at time t
ϕ, θ, ψ	:	Roll, pitch and yaw angles
$\Phi(k+1, k)$	:	Discrete-time state transition matrix from step k to $k+1$
Q_k	:	Process noise vector
r_{eS}^{e}	:	Geocentric radius
R_0, R_p	:	Equatorial radius, polar radius
R_N, R_E	:	Meridian and transverse radius of curvatures
S^a	:	Accelerometer scale factor error
S^g	:	Gyroscopes scale factor error
T_s	:	Sampling time
$U(a, b)$	:	Uniform probability density with mean $(a+b)/2$ and variance $(b-a)^2/12$

$v(t), v_t$	:	Process noise at time *t*
v^a	:	Accelerometer noise
v^g	:	Gyroscope noise
v^{ab}	:	Accelerometer bias noise
v^{gb}	:	Gyroscope bias noise
$w_t^{(i)}, \overline{w}_t^{(j)}$	:	Importance weights and normalized importance weights
x^e, y^e, z^e	:	Axes of the e-frame
x^i, y^i, z^i	:	Axes of the i-frame
x^n, y^n, z^n	:	Axes of the n-frame
x_t	:	State vector at time *t*
$\hat{x}_{t\,\vert\,t}$	:	Estimated state at time *t*
y_t	:	Measurement at time *t*
Y_t	:	Measurement vector at time *t*

CHAPTER 1

INTRODUCTION

Navigation is to determine the position, velocity or orientation of a vehicle relative to some reference frame [1]. In order to navigate, some means or methods (e.g. astronomy, geometry, radio signals, compass) have been developed and used by human race since the known ages. In early periods of history even now, people observed geographic features, stars and moon in order to find their ways. In the last century, fundamentals of modern navigation methods are established. By means of the recent developments in navigation, electronic and manufacturing technology, Global Navigation Satellite Systems (GNSS), inertial measurement units (IMUs), inertial navigation systems (INS), magnetometers etc. are commonly used as a necessity in both civil and military platforms [2].

INS are the main part of the integrated navigation for most of the aerial and underwater vehicles. However, the accuracy of an inertial navigation solution decreases with time as the inertial instrument (e.g., gyroscope and accelerometer) errors are integrated through the navigation equations. Therefore, different aiding techniques are used to bound the drift in these systems. One of the commonly used techniques is the integration of INS with GNSS signals. By means of this integration, the advantages of both technologies are combined to give a complete navigation solution [2, 3].

Terrain Referenced Navigation (TRN) has been in use since the beginning of the flight when aircraft position was estimated by the pilot recognition of landmarks. This technique, still in use today, is known as pilotage [3]. In modern world, the need for TRN appears when GNSS signals are unavailable, blocked or jammed. By means of barometer/depth sensor, radar altimeter/sonar, inertial measurement unit and digital terrain elevation database (DTED), TRN can be applied to both aerial or underwater vehicles.

1.1 Literature Review

In literature, there are batch and recursive types of algorithms that try to solve TRN problem. Batch-oriented algorithms are based on the correlation of collected terrain profile measurements without any system model. Therefore, during data collection a straight trajectory has to be followed without any maneuver. Major drawback of the batch-based algorithms is additional effort requirements for mission planning [4]. Most of the recursive algorithms based on EKF uses the approximation of the estimation model. Inherent highly nonlinear structure of terrain profile leads the TRN problem to become nonlinear. Therefore, traditional linear or linearized estimation techniques such as Kalman or Extended Kalman Filter (EKF) are not suitable for many terrain types [4].

Thanks to the recent developments in the processor technology, nonlinear techniques become a popular research area on the engineering problems to overcome the disadvantages of classical methods. Researchers all around the world spend more and more time on the nonlinear methods such as Unscented Kalman Filter (UKF), Point-Mass Filter (PMF) and Particle Filter (PF) to solve the problems that includes severe nonlinearities and various type of probability distributions [4, 5, 6].

SITAN(Sandia Inertial Terrain Aided Navigation) developed by Hostetler and Andreas [7] is the one of the famous recursive algorithms found in the terrain navigation literature. In this work, continuous updating of INS using nonlinear Kalman filtering techniques is achieved. They use the EKF with different approaches including the simple first order local derivatives, a modified stochastic linearization technique and parallel filters utilizing bank of reduced order Kalman filters, which are needed when the initial position uncertainties are large. Yet, filter does not work properly in large uncertainties because of the repetitive terrain profiles.

Anonsen [4] investigates the position estimation performance of the nonlinear techniques in an autonomous underwater vehicle application with a low-dimensional state-space model leading to discrepancy between the true system and the filter model. This doctoral work provides an overview of the nonlinear UKF, PMF, PF

techniques in terrain navigation. In this work, drawbacks and benefits of each technique are discussed.

Bergman [5] studied the nonlinear estimation problem in terrain navigation and tracking applications. Instead of simplifying the model of the problem by means of local linearization, this doctoral work focuses on solving the problem analytically and approximates the implementation of the solution. In order to solve the problem, the Bayesian solution is derived using PMF and PF. PMF is based on the adaptive grids calculating probability of a set of points on a dense mesh deterministically. The Cramer-Rao bound is also utilized to evaluate the algorithm. The drawback of the PMF algorithm is caused by the computational aspects. The complexity increases exponentially with the dimension. In this work, different particle filters are applied to terrain navigation and evaluated against the PMF. Particle filters offer an alternative approach to recursive estimation and gives tractable solutions to general high dimensional problems. Results are shown to have equal performance to the PMF.

One of the most interesting algorithms for TRN is proposed by Nordlund [8]. He proposes a combination of algorithm composed of a particle filter and Kalman filter using a low-dimensional state-space model. This technique is known as Rao-Blackwellization method that requires the marginalization of the full conditional posterior density with respect to the linear part. In this method to reduce the computational load, nonlinear states, i.e. horizontal positions, are estimated by means of particle filter and linear altitude state is estimated by multiple Kalman filters.

In literature, as mentioned above there are very useful papers investigating the effect of different types of algorithms on the solution of the TRN problem. All of these studies were done with a prefixed IMU, which is the most critical element of navigation system. However, it is not found any work that investigates the effect of IMU selection on TRN performance together with number of particle size. This gives the main trigger of this study.

1.2 Scope and Contributions

The main objective of this thesis is to apply one of the nonlinear estimation methods, which is the PF on TRN problem. Two different PF implementations are studied, Sampling Importance Resampling (SIR) and Sequential Importance Sampling with effective Resampling (SIS-R) [5, 9]. In order to compare the designed filters, convergence performance and horizontal position accuracy are investigated regarding the IMU quality and the number of particles. The importance of IMU selection for TRN application is especially emphasized. These implementations do not require the linearization of system and measurement models. However, linearized INS system model is used due to the slower dynamics than the nonlinear INS mechanization equations [10]. In addition, an IMU model is developed including some stochastic and deterministic errors such as bias, misalignment, scale factor and noise [2, 10]. Moreover, it is assumed that the vertical channel of INS is compensated by barometer. By means of this assumption, the altitude and the down velocity component of the vehicle are not to be estimated instead supplied to PF directly. INS/barometer integration is not in the scope of this thesis, for more information about this subject one may look at the references [11, 12].

The contributions of this study to the research area of TRN are summarized as follows:

- The comparison of SIR and SIS-R algorithms for the TRN of an aircraft using different IMUs and number of particles. The effect of IMU selection on filter convergence and horizontal position estimation accuracy is presented. The first results of this work were published in [13].
- The development of a terrain referenced navigation algorithm with embeddable navigation equations.
- The development of a terrain referenced navigation algorithm in an open-loop architecture that is not dependent on a specific navigation system. Hence, applied algorithms can easily be ported to utilize on other platforms operating with different navigation systems. Available navigation systems on the platform can be used without any intervention.

1.3 Organization of the Thesis

The basics of the inertial navigation systems are given in CHAPTER 2. Some fundamental information is presented about gyroscope and accelerometer technology. The model and error types of the IMU, the mechanization equations of the INS in the North-East-Down (NED) frame and the linear error model of the INS are all mentioned in this chapter.

CHAPTER 3 presents the brief background theory about the recursive Bayesian estimation techniques and introduces the main steps of the particle filtering. The implemented algorithms SIR and SIS-R are also discussed. Pseudo codes for each technique are given in related sections.

CHAPTER 4 covers how the sensor fusion is handled. It starts by explaining the operation principle of TRN and the correction configuration of the INS. The system model and the measurement model for the sensor fusion are given in this chapter.

CHAPTER 5 is dedicated to present the conducted simulations and the obtained results. The performance evaluation of the utilized sensor fusion algorithms is given for a reference scenario.

CHAPTER 6 discusses some conclusions of the thesis study and makes suggestions for the future work.

CHAPTER 2

INERTIAL NAVIGATION SYSTEMS

This chapter presents the fundamentals about the inertial navigation systems, which is required for understanding the basic principles of the inertial navigation. First of all, it starts with an introduction to IMU sensor technology. Some beneficial information about gyroscopes and accelerometers is mentioned and different error parameters are explained for modeling these sensors. Thanks to this part, an IMU model is developed for the simulations. Remaining part of the chapter is dedicated to the modeling of the inertial navigation, which combines the frames of reference, the Earth surface, and gravity models, INS mechanization equations, INS error model and state space system model for the sensor fusion.

2.1 IMU Technology

An IMU combines two types of sensors which are gyroscopes and accelerometers, and consists of other electronic parts a processor and an integrated power supply [2, 10]. The sensors are fixed to the three orthogonal axes in such a way that each axis has one gyroscope and one accelerometer. In this section, operational and technological basics of several sensor types are discussed.

2.1.1 Accelerometers

Accelerometers measure specific force along a single sensitive axis with respect to inertial space. There are many types of accelerometers. Most of them use vibrating beams or are pendulous. Most of the pendulous type of accelerometers is based on closed loop configuration. A proof mass is connected to the accelerometer case by means of a hinge and pendulous arm. The proof mass can move only in the input axis direction by damping it with a hinge in other axes. When the mounted platform starts to move, a force is exerted on the proof mass resulting in a position change of the

proof mass. To stabilize the proof mass an external force is needed. A pick-off system is used to measure the displacement of the proof mass. This position change is compensated using a torquer so that the proof mass is fixed to its initial condition. Figure 1 presents the working principle of an accelerometer [2].

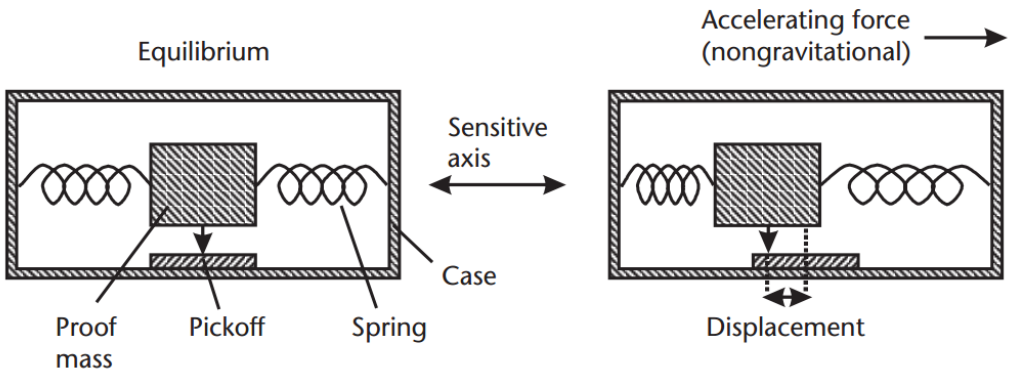

Figure 1 Basic working principle of an accelerometer [2]

The performance of the accelerometer is affected by the total integration quality of the components and control electronics. The performance grades for various prices can be attainable by changing the component quality [2].

Another type of accelerometers is vibrating beam accelerometer also known as resonant accelerometer. Similar components are used in the design of this accelerometer as in closed loop pendulous accelerometer. The main difference between these accelerometers is that in the vibrating beam accelerometer the proof mass is supported along the sensitive axis by a vibrating beam at its resonant frequency to remain fixed with respect to the case. This can be done such a way that the proof mass is pulled and pushed by the beam resulting in tension and compression in the vibrating beam that limits its motion. As a result, specific force on the system can be measured by determining the resonant frequency of the vibrating beam. In addition, there is no need to feedback the displacement change, therefore, resonant accelerometer is naturally in an open loop structure. Figure 2 shows the simple structure of an vibrating beam accelerometer [2].

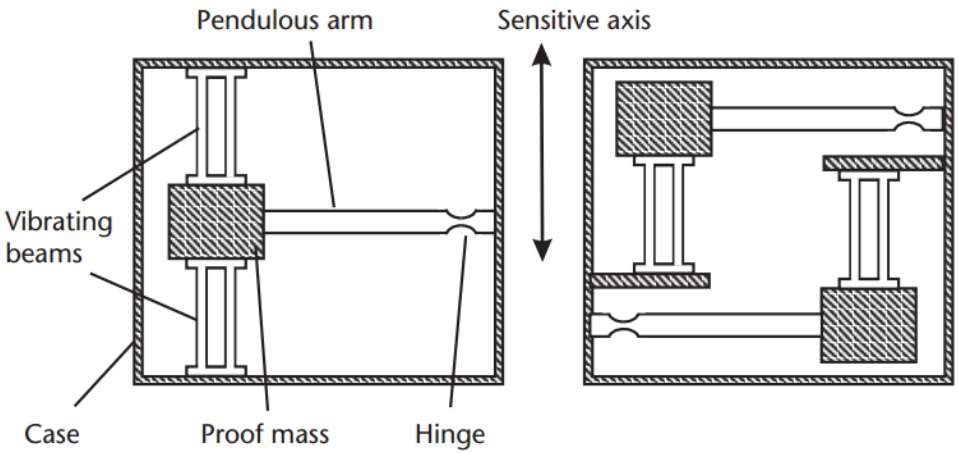

Figure 2 Simple structure of a vibrating beam accelerometer [2]

Still today, highest-grade accelerometers have mechanical type of technology. In addition to mechanical accelerometers, there are accelerometers based on microelectromechanical system (MEMS) technology. In recent years, with the help of achievements in the micromachining process and microelectronic industry, MEMS sensors have been utilized increasingly by the aerospace, automotive and mobile device industries. In contrast to mechanical systems, the large amount of production capability leads these systems to be low cost besides being small sized and having low power consumption. By means of this technology overall system can be manufactured in a single microchip. Hence, all of mentioned reasons above, MEMS accelerometers overcome the mechanical ones. The working principle of MEMS accelerometers is the same as the mechanical types. Only difference comes from the manufacturing method. In fact, both pendulous and resonant types of accelerometers can be produced by MEMS technology [2].

There are various types of MEMS accelerometers such as piezo-resistive, capacitive, piezo-electric, tunneling. The classification of these accelerometers is based on the transduction type, which is utilized for converting the position change of the proof mass to electrical signal. On a common microchip, quartz or silicon accelerometers and electronic parts can be integrated in three orthogonal axes. As the dimensions of the sensing elements decrease, a reduction in performance and sensitivity occurs inevitably meaning a higher noise level and sensor error in measurements. MEMS inertial sensors are not suitable to be utilized as a standalone navigation system if the

accuracy of measurement is poor or not enough to meet the navigation system requirements. Therefore, deterministic and stochastic errors have to be accurately calibrated before and during the operation in order to obtain desired performance level [14, 2].

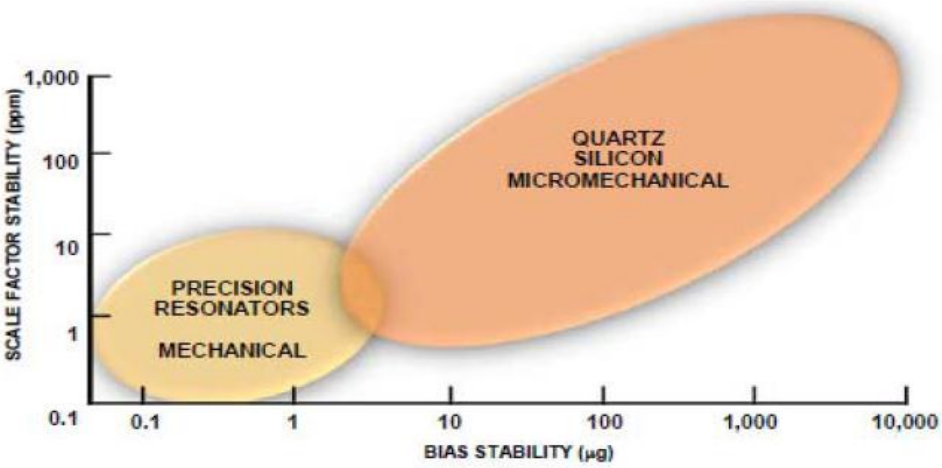

Figure 3 Different accelerometer error budgets [15]

In Figure 3, the error budgets of MEMS based and mechanical accelerometers are compared. In the near future, it is expected that mechanical sensors will be replaced by very accurate MEMS based quartz or silicon sensors [15].

2.1.2 Gyroscopes

Gyroscopes measure the angular velocity of the host vehicle around a single sensitive axis with respect to inertial space. By means of gyroscopic measurements, rotation matrices, which are required to transform one coordinate system to another, can be formed. Relative orientation angles between coordinate systems are calculated to obtain velocity and position information of the platform using these matrices.

Gyroscope technology is separated into more categories than accelerometer technology. There are many complex and different types of gyroscopes. Gyroscopes can be divided into three main groups namely spinning mass, optical and vibratory gyroscopes [16].

The operation principle of the spinning mass gyros is based on the conservation of angular momentum, which states that the angular momentum of a rotating body will remain constant with respect to inertial frame if there is no external torque applied on it. The basic way to measure angular velocity of a rotating body is based on gyroscopic theory. When a spinning mass is rotated around its input axis perpendicular to its spin axis, a reaction will take place in the third axis, which is perpendicular to both. By measuring this reaction, the rate of input rotation can be determined. This principle works for all mechanical gyroscopes. Figure 4 depicts the basic schematic of single-degree-of-freedom spinning mass gyro [2].

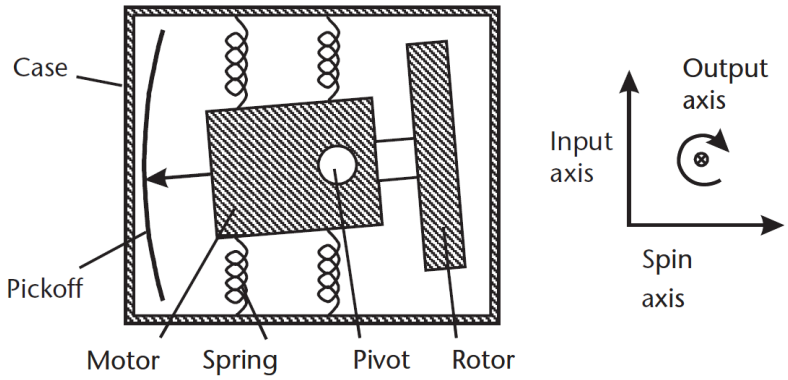

Figure 4 Schematic of simple spinning mass gyro [2]

There are various types of spinning mass gyroscopes. These devices are categorized with respect to their techniques to minimize the unwanted effects such as nonlinearity and friction. Dynamically tuned, electrostatically suspended, magnetically suspended are the examples of spinning mass gyroscope. A dynamically tuned gyroscope eliminates the movement in the reaction axis by using an electrical rebalance loop. The input rotation rate is proportional to the amount of required current for balancing the spinning mass. The different graded gyros for various prices can be attainable by changing the component quality and electronics [2, 16].

Another type of gyroscopes is the optical gyroscopes. Optical gyroscopes can be grouped as ring laser and fiber optic gyroscopes. Both of them are based on the same working principle using the Sagnac effect [16].

Ring laser gyroscope (RLG) combines a triangular laser hollow with mirrors located at the corners of closed loop with three or more arms as shown in Figure 5 [2].

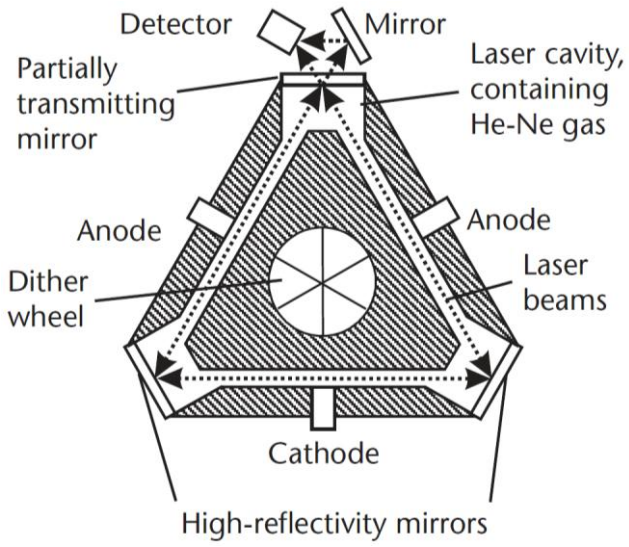

Figure 5 Ring Laser Gyro [2]

A ring laser has two opposite lasing beams, one of which travels in the clockwise direction and the other in the counter-clockwise direction. Mirrors at the corners are utilized to generate a continuous light path. Laser beams moves around this path being reflected from each mirror, and return to its first point. If the sensor is not rotated, both beams have the same wavelength and travels at the same frequency. However, if the laser hollow is rotated about an axis perpendicular to its plane, the hollow length is increased for the beam in the direction of rotation and decreased for the beam in the opposite direction. As a result, the beam in the opposite direction of rotation experiences a increase in frequency and decrease in wavelength, while the converse is valid for the other beam. This results in a phase difference between two beams. By means of an interferometer the phase shift can be determined. The Sagnac effect is the name of this phenomenon and illustrated below in Figure 6 [2]. Because of the laser scattering in the hollow, there is a problem called as laser lock-in occurring at low angular rates. When this phenomenon occurs, laser beams travelling in opposite directions cannot be distinguished from each other. In this condition, the frequencies and wavelengths of the beams become nearly equal. Therefore, no phase difference between the beams can be observed resulting that the RLG is unable to

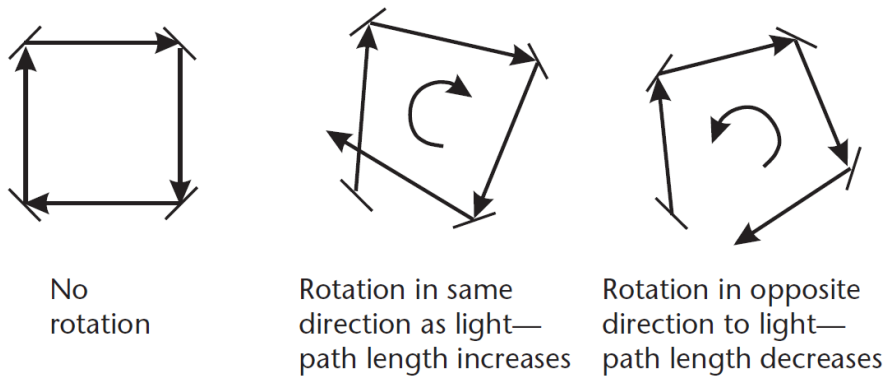

Figure 6 The Sagnac phenomenon [2]

generate accurate measurements. To overcome this problem, an interesting solution method known as dithering is used. In dithering application, a low amplitude, high frequency vibration is exerted onto the hollow about the input axis with respect to the laser case. The use of dithering method results an increase in complexity, cost and size. Moreover, the real angular measurements have to be distinguished from the vibratory dithering motion and this can be realized electronically or optically. The disadvantage of RLG technology is the requirement for producing the faces of the laser blocks and difficulties in polishing the mirrors. High technology methods, which increase the cost of the device, must be used to meet the requirements by controlling scatter, surface quality and loss [2, 16].

Another type of optical gyroscopes is the fiber optic gyroscope (FOG). As in the RLG, the working principle of the FOG is based on the Sagnac effect. Instead of travelling the loop once in opposite directions, the laser beams travel the loops many times in the fiber optic cable. An external laser source exists in the FOG, which generates travelling beams both in the clockwise and in the counter-clockwise directions. When the device experiences inertial rotation, the distances travelled by two counter waves become different. This operation results in two different frequencies for two beams that mean a phase difference between the laser beams. The output phase difference measured is proportional to the rotation rate of the device. The change in temperature and accelerations may ruin the structure of the optical fiber. This is one of the sources of error that should be minimized by using design techniques. On the other hand, fiber optic gyroscopes do not include mirrors

in their internal design. Furthermore, lock-in phenomenon is not inherent in FOG technology unlike the RLG. These advantages make the FOG technology cheaper and simpler compared to RLG. In these days, FOGs are replacing RLGs in the lower performance tactical and commercial applications. Developments in the FOG performance will make it possible to use these devices in strategic applications [2].A typical FOG is comprised of a fiber optic coil, a light source, a photo detector, a beam splitter, and a couple of lenses. In the Figure 7 below, internal structure of a typical FOG is illustrated [16].

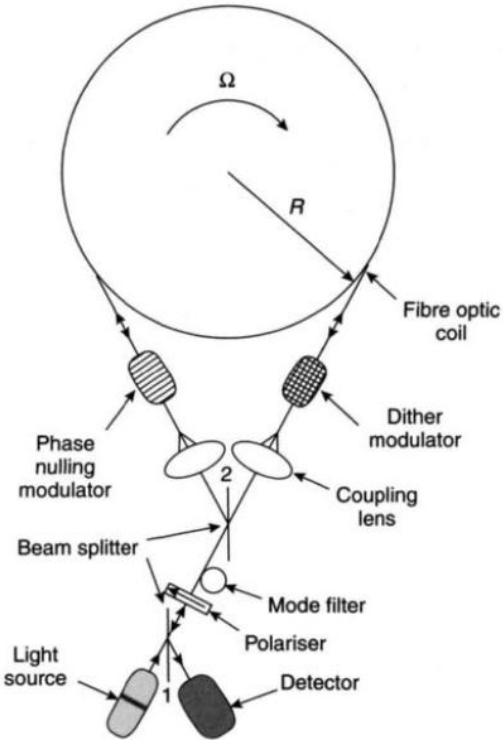

Figure 7 Internal structure of a FOG [16]

By the help of the developments in the FOG technology, Interferromagnetic Fiber Optical gyroscope (IFOG) is invented. This device has brought substantial increases in the performance of gyroscopes. This technology has been used in many lower grade applications as unmanned vehicles and stabilization systems. The accuracy of an IFOG is nearly as the RLG. However, due to high sale prices, it is not widely used by the industry for now. Actually, in the near future, with increasing industrial investments, the costs will decrease and IFOG is expected to replace RLG in high performance applications. By means of IFOG technology, integrated navigation

systems have the chance of having lower costs with high reliability and suitability for a wide range of military and commercial applications [17, 2, 16].

Finally, the last gyroscope type mentioned in this thesis is the vibratory gyroscopes. A vibratory gyroscope has a component that oscillates in a harmonic motion. This vibrating component can be a tuning fork, hemisphere, ring, spring or a pair of beams. The working principle, which is same for all, is to obtain the Coriolis acceleration of the vibrating element when it experiences a rotational motion. A simple harmonic motion is triggered by the Coriolis acceleration in the plane perpendicular to vibration. The amplitude of the triggered vibration is proportional to the applied rotational rate. Most of the vibratory gyroscopes are based on MEMS technology whereby these sensors are low performance and low cost. In addition, quartz MEMS have better performance than silicon ones. As an exception, hemispherical resonator gyroscope offers much higher performance than other vibratory gyroscopes. Actually, it is suitable for space applications [2, 16, 17].

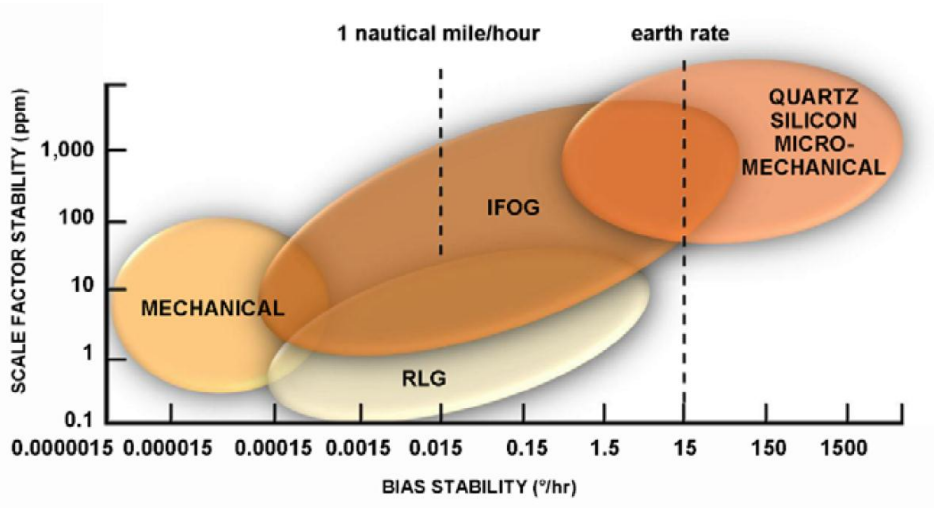

Figure 8 Gyroscopes technology applications [15]

The evaluation of gyroscope technologies is illustrated in Figure 8 [15]. It is expected that mechanical sensors will be replaced by accurate other sensors. This figure indicates that FOG, RLG sensors have better performance characteristics than the quartz and MEMS gyros [15].

2.2 IMU Errors

The measurement accuracy of the inertial sensors is dependent on different types of errors, which decrease the performance of the navigation system. The output of the accelerometers and gyroscopes includes the error types of bias, misalignment, scale factor and noise. Each of these error types shows different characteristics namely, a fixed deterministic part, a temperature dependent part, a turn-on to turn-on part and an in-run variation part. By means of the IMU calibration process, first two deterministic components of sensor errors can be removed from the IMU output.

The remaining two error parts change in a stochastic manner, which mostly determines the accuracy of the IMU [18]. In this study, only stochastic errors are modeled to form the inertial data outputs and it is assumed that first two deterministic components of sensor errors are completely removed by means of the pre-calibration. Magnitudes of sensor errors also determine the quality of an IMU such as navigation grade, tactical grade, and automotive grade. These different categories are given in Table 1 mostly accepted by the industry whereas performance boundaries are not strictly defined [19].

Table 1 IMU classification [19]

Performance	**Navigation Grade**	**Tactical Grade**	**Automotive Grade**
Accelerometer Bias (mg)	< 0.05	0.2 - 10	> 10
Gyroscope Bias (deg/h)	< 0.01	0.1 - 10	> 100

Stochastic error types of accelerometers and gyroscopes are explained briefly in the following subsections.

2.2.1 Bias Error

In an ideal case, when no input is applied to an inertial sensor, the measurements obtained from the sensor should be zero [18]. While operating the inertial sensors, a fixed error remains on the output of the sensor. This fixed error is known as the sensor bias observed at any time in the measurements [16]. Unit of the bias is mg or m/s^2 for accelerometer and deg/h for gyroscope. Bias error is not dependent on the applied acceleration or angular rate onto the sensors. It can be divided in two parts as static and dynamic biases. In the literature, static bias is also considered as bias repeatability or turn-on to turn-on bias. This part of the bias is formed at the startup due to the internal electronics and it is fixed during IMU operation time. However, it changes from startup to startup. Even after calibrating the sensor, the residual bias is also included in bias repeatability [2].

Bias repeatability can be modeled as a random constant as follows,

$$\begin{aligned} \dot{b}^a &= v^{ab} \\ \dot{b}^g &= v^{gb} \end{aligned} \tag{2.1}$$

where $v^{ab} \sim N(0, \sigma^2_{v^{ab}})$ and $v^{gb} \sim N(0, \sigma^2_{v^{gb}})$ are zero-mean Gaussian white noises for accelerometer and gyroscope bias, respectively.

As for the dynamic bias, it is considered as bias stability or in-run bias change. It changes over periods of time depending on the type of sensors. Moreover, it also contains temperature dependent bias that remains after the calibration of sensors. Bias stability component is generally much less than the bias repeatability component. First order Gauss-Markov process can be used to model the bias stability [20, 2].

2.2.2 Misalignment Error

The misalignment error arises in two different geometrical ways. First one the nonorthogonality between the IMU axis and the axis of the carrier platform. This is shown in Figure 9 below [21].

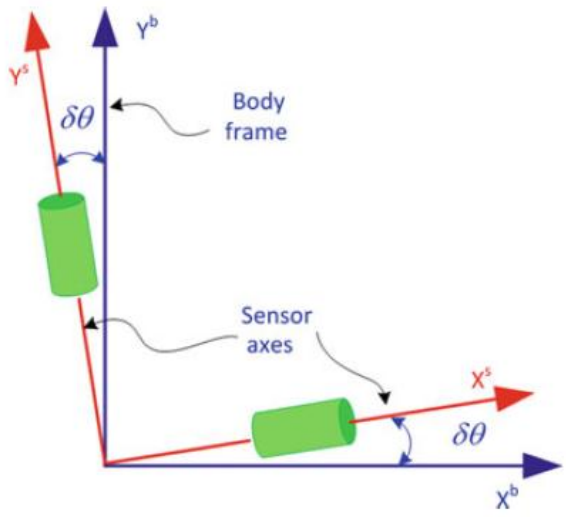

Figure 9 Angle between the body frame and the sensor frame determines the misalignment error [21]

The nonorthogonal mounting of sensors in IMU itself due to the manufacturing technology capabilities of producers is the second reason for misalignment error. Figure 10 illustrates this case [21].

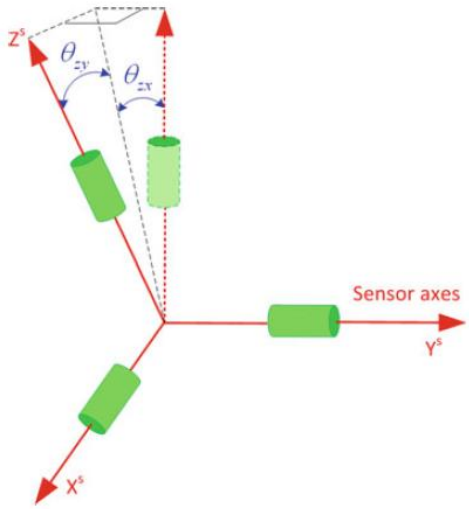

Figure 10 Non-orthogonality error between sensor axes [21]

The output of the sensors is continuously cross-coupled because of the misalignment angle errors. By means of today's manufacturing technology, the misalignment error is in the order of a few milliradians [18, 21].

2.2.3 Scale Factor Error

Scale factor is defined as the deviation of slope of the input to output ratio curve of an inertial sensor from unity. Scale factor is a parameter that is used for calculating acceleration or angular rates from the voltage outputs of accelerometers and gyroscopes. The expected value of the scale factor is 1. However, this value is generally not observed in most cases. The unit of the scale factor error is ppm (parts per million). The scale factor errors are proportional to outputs of the sensors along the sensitive axis. As the bias error, scale factor is formed at the startup of the instrument and is fixed during IMU operation time. It also includes the residual error remaining after the calibration of sensor [18, 2]. Scale factor can be modeled as a random constant and depicted in Figure 11 [21].

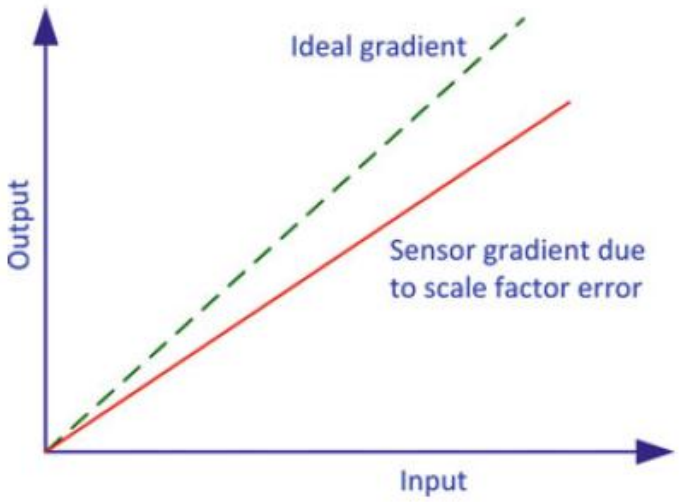

Figure 11 Scale factor error for an inertial sensor [21]

2.2.4 Random Noise Error

A random noise error is observed on the IMU measurements because of the internal electronics or electromechanical manufacturing deficiencies of the sensors. It is a an output of stochastic process which cannot be calculated analytically. A zero-mean white noise model can be used to simulate the random noise error of the sensors. When this error is integrated over time for the accelerometers and gyroscopes, a

random walk behavior is seen in the velocity and angle calculations, respectively. The square root of the integration time affects the magnitudes of velocity and angle random walks proportionally [2, 20].

There are additional errors types that inertial sensors have such as scale factor nonlinearity, g-dependent bias, quantization noise. However, these errors have negligible effects on the system performance in comparison with the error types explained before. Therefore, they are out of scope of this thesis and for more information on the error characterization and compensation of IMU errors refer to [2, 18]. The mathematical model developed to simulate the IMU by adding sensor errors to the reference acceleration and angular velocity in vehicle body frame are provided in equations (2.2) and (2.3) respectively [2].

$$\begin{bmatrix} \tilde{f}_x^b \\ \tilde{f}_y^b \\ \tilde{f}_z^b \end{bmatrix} = \left\{ \begin{bmatrix} b_x^a \\ b_y^a \\ b_z^a \end{bmatrix} + \begin{bmatrix} 1 & M_{xy}^a & M_{xz}^a \\ M_{yx}^a & 1 & M_{yz}^a \\ M_{zx}^a & M_{zx}^a & 1 \end{bmatrix} \begin{bmatrix} f_x^b \\ f_y^b \\ f_z^b \end{bmatrix} \right\} \begin{bmatrix} 1+S_x^a & 0 & 0 \\ 0 & 1+S_y^a & 0 \\ 0 & 0 & 1+S_z^a \end{bmatrix} + \begin{bmatrix} v_x^a \\ v_y^a \\ v_z^a \end{bmatrix} \tag{2.2}$$

where $\tilde{f}_x^b$, f_x^b, b_x^a, M_x^a, S_x^a, v_x^a represent erroneous accelerometer measurement, reference acceleration, bias, misalignment, scale factor and noise parameters only for the accelerometer on the x-axis of the body frame respectively. The notation for the y and z axes of the body frame is adopted similarly.

$$\begin{bmatrix} \tilde{\omega}_x^b \\ \tilde{\omega}_y^b \\ \tilde{\omega}_z^b \end{bmatrix} = \left\{ \begin{bmatrix} b_x^g \\ b_y^g \\ b_z^g \end{bmatrix} + \begin{bmatrix} 1 & M_{xy}^g & M_{xz}^g \\ M_{yx}^g & 1 & M_{yz}^g \\ M_{zx}^g & M_{zx}^g & 1 \end{bmatrix} \begin{bmatrix} \omega_x^b \\ \omega_y^b \\ \omega_z^b \end{bmatrix} \right\} \begin{bmatrix} 1+S_x^g & 0 & 0 \\ 0 & 1+S_y^g & 0 \\ 0 & 0 & 1+S_z^g \end{bmatrix} + \begin{bmatrix} v_x^g \\ v_y^g \\ v_z^g \end{bmatrix} \tag{2.3}$$

where $\tilde{\omega}_x^b$, ω_x^b, b_x^g, M_x^g, S_x^g, v_x^g represent erroneous gyroscope measurement, reference angular velocity, bias, misalignment, scale factor and noise parameters only for the gyroscope on the x-axis of the body frame respectively. The notation for the y and z axes of the body frame is adopted similarly.

2.3 Navigation Mathematics

Inertial navigation systems are used to provide a reliable and self-contained navigation data such as position, velocity and attitude information to the vehicles. In order to calculate the navigation states, the INS processor utilizes acceleration and angular velocity measurements provided by IMU and gravity exerted on the platform using the gravity model. After feeding these data into INS mechanization equations, inertial solution can be found. Throughout this part of the study, mathematical foundations for the principles of navigation systems are described in detail.

2.3.1 Coordinate Frames

A coordinate frame can be defined as a reference that comprises of three orthogonal axes and an origin in space for a moving platforms. It has to be defined according to the right-hand rule. In simple problems, motion is modeled with respect to the Earth assuming that the Earth is flat and non-rotating. However, in the real navigation problems these assumptions cannot be applicable because the shape and the rotational rate of the earth have a significant effect on the accuracy of the navigation solution. Navigation studies require different coordinate frames because of the relative motion of the objects. The outputs of IMU are expressed in body frame of the vehicle but referenced to a non-rotating inertial frame. Therefore, for accurate navigation, the relationship between the different coordinate frames has to be properly modeled [2, 10, 21].

2.3.1.1 Earth-Centered Inertial Frame (i)

Earth-centered inertial frame is usually called the inertial reference frame. This frame has to be placed at some origin and neither accelerate nor rotate with respect to the distant stars. It turns out that by placing this frame at the center of the earth, but not rotating with the earth, is suitable for near earth type of applications. The z-axis of

the inertial frame points through the north pole. Figure 12 shows the axes of inertial frame [2].

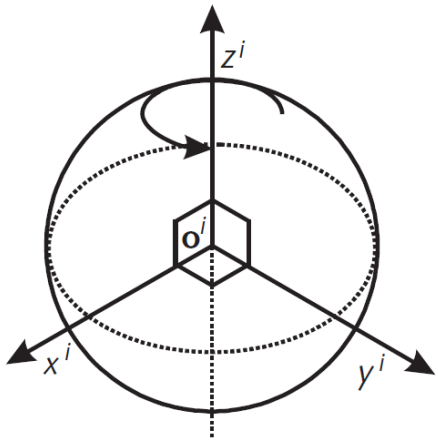

Figure 12 Axes of the inertial frame [2]

2.3.1.2 Earth-Centered Earth-Fixed Frame (e)

Earth-centered earth-fixed frame is usually known as the earth frame. This frame also has its origin at the center of the earth, but it also rotates with the earth rate. The position is often given relative to this frame as latitude, longitude and altitude. The z-axis points through the north pole of the earth, the x-axis through the zero-meridian and the y-axis completes the right hand rule. The axes of the earth frame are depicted in Figure 13 [2].

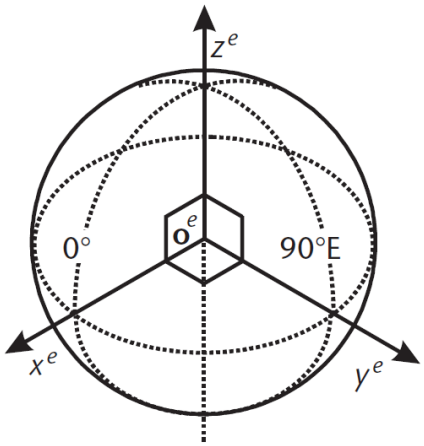

Figure 13 Axes of the earth frame [2]

2.3.1.3 Local Navigation Frame (n)

The local navigation frame also known as the north-east-down frame is illustrated in Figure 14 [10]. The frame is placed at the center of the vehicle and moves with the vehicle. The z-axis points downwards along the normal of the surface of the earth, the x-axis points to the north and the y-axis completes the right hand system, i.e., points to the east. The local navigation frame is important in navigation because the attitude of the vehicle relative to the north, east, and down directions can be calculated. In addition, it provides a convenient set of resolving axes for position and velocity [2].

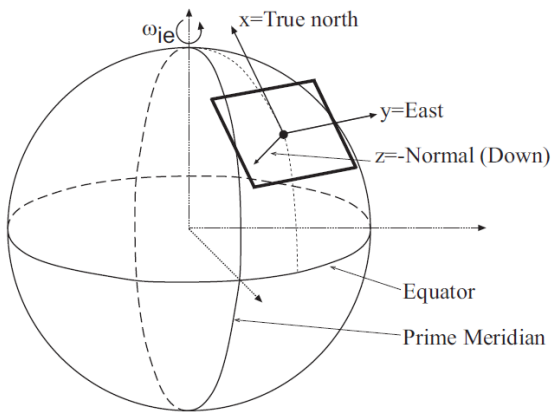

Figure 14 Axes of the local navigation frame [10]

2.3.1.4 Body Frame (b)

Navigation solution is generally sought for the body frame. The body frame is also placed at the center of the vehicle, but it also rotates with the vehicle. The x-axis points through the nose of the vehicle, the z-axis down through the belly and the y-axis completes the right hand system, i.e., points through the right. For angular motion, the x, y, and z-axes are called as the roll, pitch, yaw axes respectively. Figure 15 illustrates the body frame [2].The body frame is important for navigation because it describes the object that is moving. All strapdown inertial sensors measure the motion of the body frame with respect to the inertial frame [2, 10].

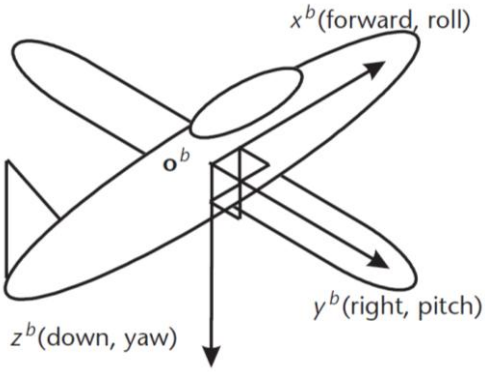

Figure 15 Axes of the body frame [2]

2.3.2 Earth Surface Model

In order to navigate near the earth, i.e. calculate the position and the velocity of the vehicle with respect to the earth, it is necessary to define the surface of the earth. Therefore, a set of coordinates for relating position with respect to that surface, the latitude, longitude and height should be defined. Unfortunately, the surface of the earth is not uniform. Hence, it is approximated to a known shape, which is then fitted to the true surface of the Earth at mean sea level. The Earth's surface is modeled as an ellipsoid in most navigation applications. The ellipsoid shape defined by two radii is symmetric about the north-south axis and the equatorial plane. Figure 16 illustrates the surface model of the earth [2].

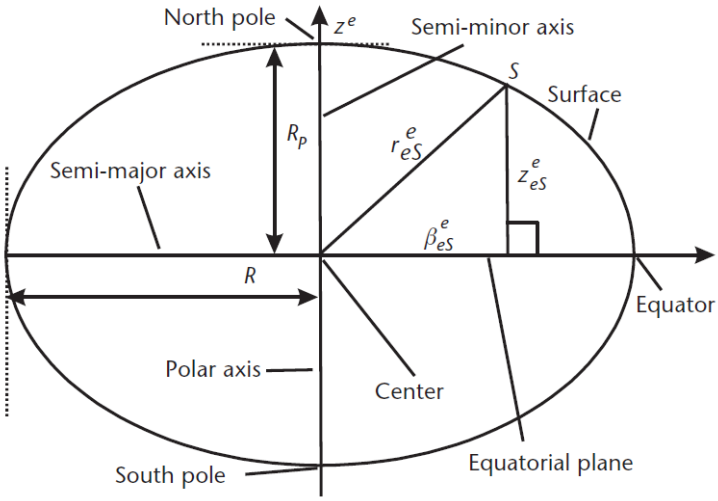

Figure 16 Surface model of the earth [2]

The equatorial radius, R_0, is the distance from the center to the equator and the polar radius, R_p, is the distance from the center to each pole. The equatorial radius and the polar radius are also given as the lengths of the semi-major axis and the semi-minor axis, respectively. The ellipsoid is commonly defined in terms of the equatorial radius and either the eccentricity of the ellipsoid, e, or the flattening of the ellipsoid, f. WGS84 model defines the reference ellipsoid in terms of the equatorial radius and the flattening. The equatorial radius and the polar radius values are

$$R_0 = 6378137\ m$$
$$R_P = 6356752.3142\ m$$

The flattening and eccentricity may be derived accordingly.

$$f = \frac{R_0 - R_P}{R_0}, \qquad e = \sqrt{1 - \frac{{R_P}^2}{{R_0}^2}} \tag{2.4}$$

The transverse radius of curvature, R_E, and the meridian radius of curvature, R_N, which will be used to calculate the rates of change of latitude and longitude, can also be derived in accordance with the ellipsoidal definition of the Earth.

$$R_N = \frac{R_0(1 - e^2)}{(1 - e^2 \sin^2 L)^{\frac{3}{2}}}$$
$$R_E = \frac{R_0}{(1 - e^2 \sin^2 L)^{1/2}} \tag{2.5}$$

For navigation applications, rotation rate of the Earth is assumed as a constant based on the WGS84 datum. The value of the Earth angular rate is given as $\omega_{ie} = 7.292115 \times 10^{-5}\ rad\ s^{-1}$ [2]. Rotation of the Earth is in counter-clockwise direction around z-axis of i-frame and e-frame. Earth rotation vector can be expressed in these frames as follows.

$$\omega_{ie}^{i} = \omega_{ie}^{e} = (0\ 0\ \omega_{ie})^T \tag{2.6}$$

Earth rotation vector can be expressed in n-frame as a function of geodetic latitude.

$$\omega_{ie}^{n} = (cosL\ \ 0 - sinL)^T \omega_{ie} \tag{2.7}$$

2.3.3 Earth Gravity Model

The gravity vector is the vector sum of the gravitational force of the Earth mass and the centrifugal force due to Earth rotation. The relative size and orientation of these two forces is dependent on the point of evaluation. For example, the centrifugal force is maximum at the equator and zero along the Earth axis of rotation. The geodetic surface of the Earth is defined to be everywhere normal to the gravity vector (i.e., an equipotential surface relative to the force of gravity). The geodetic surface is distinct from the actual irregular shape of the surface, but may be imagined as the mean shape that the Earth would take if the solid surface of the Earth were completely covered with sea water. WGS84 model of gravity at the ellipsoid surface as a function of latitude is given by Eqn.(2.8).

$$g_0(L) = 9.7803255359 \times \frac{1 + 0.001931853 \sin^2 L}{(1 - e^2 \sin^2 L)^{1/2}} \quad m/s^2 \tag{2.8}$$

This gravity field model is known as Somigliana model [2]. The variation of the gravitational field with altitude can be modeled using a scaling parameter, which is a function of latitude and height above the ellipsoid as follows.

$$g(L, h) = \frac{r_{eS}^{e}(L)^2}{(r_{eS}^{e}(L) + h)^2} g_0(L) \tag{2.9}$$

where $r_{eS}^{e}(L)$ is the geocentric radius given by

$$r_{eS}^{e}(L) = R_E \, (\cos^2 L + (1 - e^2)^2 \sin^2 L)^{1/2} \tag{2.10}$$

The sum of the accelerations caused by mass attraction and centripetal forces forms the local gravity vector, g_l^n, which is perpendicular to the reference ellipsoid and in the direction of the third axis of the n-frame. It can be represented as given in Eqn. (2.11).

$$g_l = g(L, h) - \omega_{ie} \times \omega_{ie} \times (R_E + h) \tag{2.11}$$

2.3.4 Mathematical Mechanization of INS

This section of the thesis presents the equations derived to calculate navigation states, namely position, velocity and attitude using acceleration and angular rate measurements of IMU. Figure 17 shows a block diagram of steps utilized in inertial navigation processing [16]. Once the initial navigation states are set, attitude update step is done in order to transform the specific force from b-frame to n-frame. Then, velocity and position are computed using gravity model and Coriolis correction through the navigation mechanization equations.

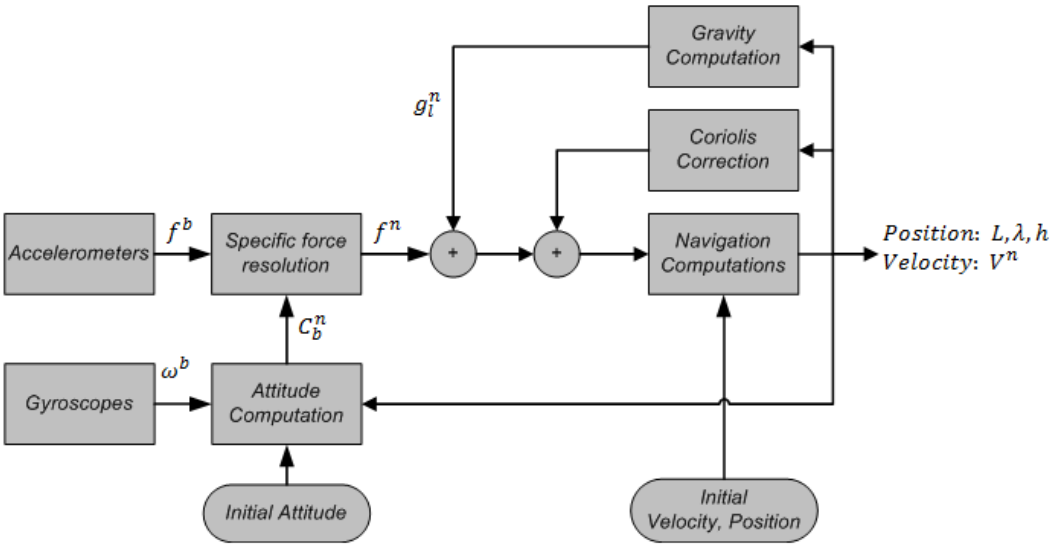

Figure 17 Block diagram of INS mechanization [16]

INS mechanization model is derived in the form of continuous time differential equations [20]. Mechanization equations for attitude, velocity and position are presented in the following sections.

2.3.4.1 Mechanization of Attitude

Attitude is used for the transformation of vectors between different reference frames. Direction cosine matrix (DCM) is utilized for updating attitude solution. Then, Euler angles can be computed from the DCM. Differential equation describing the rate of change of DCM that is used to transform vectors represented in b-frame to vectors in

n-frame is given below [10]. The angular rates $\widetilde{\omega}_{ib}^{b}$, $\widetilde{\omega}_{in}^{n}$ are skew-symmetric forms of the corresponding vectors.

$$\dot{C}_b^n = C_b^n \widetilde{\omega}_{ib}^b - \widetilde{\omega}_{in}^n C_b^n \tag{2.12}$$

Angular rates existing in the equation above are defined as follows.

$$\omega_{in}^n = \omega_{ie}^n + \omega_{en}^n \tag{2.13}$$

where Earth rate in n-frame is given before in Eqn.(2.7). Transport rate ω_{en}^n is defined as the angular rate of the n-frame with respect to the e-frame represented in the n-frame [16],

$$\omega_{en}^n = \begin{pmatrix} \frac{V_E}{R_E + h} & \frac{-V_N}{R_N + h} & \frac{-V_E tanL}{R_E + h} \end{pmatrix} \tag{2.14}$$

where L, λ stand for latitude, longitude and V_E, V_N stand for east and north components of V^n. DCM matrix, C_b^n, which relates b-frame to n-frame is obtained by using Euler angles as follows [10],

$$C_b^n = \begin{pmatrix} c\theta s\psi & -c\phi s\psi + s\phi s\theta c\psi & s\phi s\psi + c\phi s\theta c\psi \\ c\theta s\psi & c\phi c\psi + s\phi s\theta s\psi & -s\phi c\psi + c\phi s\theta s\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{pmatrix}, \qquad \begin{matrix} c\alpha: cos\alpha \\ s\alpha: sin\alpha \end{matrix} \tag{2.15}$$

where $\phi,\ \theta,\ \psi$ are roll, pitch and yaw angles respectively.

2.3.4.2 Mechanization of Velocity

Navigation is to know the position and the velocity of an object with respect to a known reference point. The position with respect to the Earth is commonly expressed in curvilinear coordinates by latitude, longitude and altitude. Therefore, the velocity of the object is calculated in the n-frame. Differential mechanizations of velocity are given in the i-frame, e-frame and n-frame respectively [16, 20].

Velocity in i-frame:

$$\dot{V}^i = f^i + g^i - \omega_{ie}^i \times \omega_{ie}^i \times r - \omega_{ie}^i \times V^i \tag{2.16}$$

Velocity in e-frame:

$$\dot{V}^e = f^e + g^e - \omega_{ie}^e \times \omega_{ie}^e \times r - 2\omega_{ie}^e \times V^e \tag{2.17}$$

Velocity in n-frame:

$$\dot{V}^n = f^n + g^n - \omega_{ie}^n \times \omega_{ie}^n \times r - (2\omega_{ie}^n + \omega_{en}^n) \times V^n \tag{2.18}$$

2.3.4.3 Mechanization of Position

In order to visualize, the position is generally given in curvilinear coordinates. The rate of change of the curvilinear position can be expressed as follows [10];

$$\dot{L} = \frac{V_N}{R_N + h}, \qquad \dot{\lambda} = \frac{-V_E}{R_E + h}, \qquad \dot{h} = -V_D \tag{2.19}$$

where V_N, V_E, V_D represent the north, east and down components of the velocity vector resolved in the n-frame respectively. The vectors expressed in the e-frame can be transformed into the n-frame by means of a transformation matrix. This matrix is function of latitude and longitude and given below [2].

$$C_e^n = \begin{pmatrix} -\sin L \cos\lambda & -\sin L \sin\lambda & \cos L \\ -\sin L & \cos\lambda & 0 \\ -\cos L \cos\lambda & -\cos L \sin\lambda & -\sin L \end{pmatrix} \tag{2.20}$$

A summary of the differential mechanization equations of INS is given in the table below.

Table 2 Summary of INS mechanization [2]

Attitude	$\dot{C}_b^n = C_b^n \tilde{\omega}_{ib}^b - \tilde{\omega}_{in}^n C_b^n$
Velocity	$\dot{V}^n = f^n + g_l^n - (2\omega_{ie}^n + \omega_{en}^n) \times V^n$
Position	$\dot{L} = \frac{V_N}{R_N + h}; \ \dot{\lambda} = \frac{V_E}{R_E + h}; \ \dot{h} = -V_D$

2.3.5 INS Error Model

An error state system model will be developed for the INS and TRN integration because of the fact that the INS dynamics is fast; it is more convenient to estimate the error in the INS. Therefore, nonlinear equations for position, velocity and attitude given in the previous section need to be linearized about the most recent navigation solution. For this linearization operation, nonlinear equations are perturbed for the state space variables. In the derivation step of this model, second order terms in the error quantities are ignored [20].

2.3.5.1 Attitude Error Model

For the attitude errors, small angle attitude error model is utilized. The attitude mechanization equation is linearized with respect to the state space variables and the following results are obtained [20],

$$\delta\dot{\psi} = -\widetilde{\omega}_{in}^{n}\delta\psi + \delta\omega_{in}^{b} + C_{b}^{n}\delta\omega_{ib}^{b} \tag{2.21}$$

where $\delta\psi$ is the attitude error present in the state estimate, $\delta\omega_{ib}^{b} = b^{g} + v^{g}$ is the gyroscope error vector resolved in the b-frame and $\delta\omega_{in}^{n}$ is the error in the angular rate of the n-frame with respect to the i-frame which is composed of earth rate error and transport rate error as given below [2].

$$\delta\omega_{in}^{n} = \delta\omega_{ie}^{n} + \delta\omega_{en}^{n} \tag{2.22}$$

Earth rate error $\delta\omega_{ie}^{n}$ is obtained by perturbing the Earth rate ω_{ie}^{n} by latitude error δL. Assuming that perturbation of the Earth rate is equal to zero [2].

$$\delta\omega_{ie}^{n} = (-\omega_{ie}\sin L\,\delta L \quad 0 \quad -\omega_{ie}\cos L\,\delta L)^{T} \tag{2.23}$$

Transport rate error $\delta\omega_{en}^{n}$ is similarly obtained by perturbing ω_{en}^{n} by four of the states δV_{E}, δV_{N}, δh, and δL [2].

$$\delta\omega_{en}^{n} = \begin{pmatrix} \dfrac{\delta V_E}{R_E + h} - \dfrac{V_E}{(R_E + h)^2}\delta h \\ -\dfrac{\delta V_N}{R_N + h} + \dfrac{V_N}{(R_N + h)^2}\delta h \\ \dfrac{-\tan L\, \delta V_E}{R_E + h} + \dfrac{V_E \tan L\, \delta h}{(R_E + h)^2} - \dfrac{V_E \delta L}{(R_E + h)\cos^2 L} \end{pmatrix} \tag{2.24}$$

Attitude error model can be formed in state space form as follows [20]. Definitions for the elements of the state space are given in Section 2.3.5.4.

$$\delta\dot{\psi} = \begin{pmatrix} F_{\psi P} & F_{\psi V} & F_{\psi\psi} \end{pmatrix} \begin{pmatrix} \delta P \\ \delta V \\ \delta\psi \end{pmatrix} + C_b^n \delta\omega_{ib}^b \tag{2.25}$$

2.3.5.2 Position Error Model

Position error model is formed by perturbing the position mechanization equation given in the Table 2 [2].

$$\begin{gathered} \delta\dot{L} = \frac{\delta V_N}{R_N + h} - \frac{V_N}{(R_N + h)^2}\delta h \\ \delta\dot{\lambda} = \frac{\delta V_E}{R_E + h} - \frac{V_E \delta h}{(R_E + h)^2 \cos L} - \frac{V_E \tan L\, \delta L}{(R_E + h)\cos L} \\ \delta\dot{h} = -\delta V_D \end{gathered} \tag{2.26}$$

Position error model can be represented in state space form as follows [20]. Definitions for the elements of the state space are given in Section 2.3.5.4.

$$\delta\dot{P} = \begin{pmatrix} F_{PP} & F_{PV} & F_{P\psi} \end{pmatrix} \begin{pmatrix} \delta P \\ \delta V \\ \delta\psi \end{pmatrix} \tag{2.27}$$

2.3.5.3 Velocity Error Model

Similarly, velocity error model is obtained by perturbing the velocity mechanization equation given in Table 2 by state vectors, Earth rate and transport rate errors. The

rate of change of velocity error is given below as a function of state variables and gravity error [2],

$$\delta\dot{V}^n = -\psi C_b^n f^b + C_b^n \delta f^b - (2\omega_{ie}^n + \omega_{en}^n) \times \delta V^n - (2\omega_{ie}^n + \omega_{en}^n) \times V^n - \frac{2g_0(L)}{r_{es}^e(L)}\delta h \tag{2.28}$$

where δV^n is the velocity error present in the state estimate and $\delta f^b = b^a + v^a$ is the accelerometer error vector resolved in the b-frame.

Velocity error model can be represented in state space form as follows [20]. Definitions for the elements of the state space are given in Section 2.3.5.4.

$$\delta\dot{V}^n = \begin{pmatrix} F_{VP} & F_{VV} & F_{V\psi} \end{pmatrix} \begin{pmatrix} \delta P \\ \delta V \\ \delta\psi \end{pmatrix} + C_b^n \delta f^b \tag{2.29}$$

2.3.5.4 State Space Error Model

State space for the general integration process is composed of fifteen elements namely position, velocity and attitude errors, accelerometer biases and gyroscope biases. Each of these states is a three dimensional vector. This state space model will be utilized as the basis for linearized system model of the estimators of TRN integration process. State transition matrix without process noises is obtained by using the state space forms of individual error models for each state space variable as follows [2, 10, 20].

$$\begin{pmatrix} \delta\dot{P} \\ \delta\dot{V} \\ \delta\dot{\psi} \end{pmatrix} = \begin{pmatrix} F_{PP} & F_{PV} & F_{P\psi} & 0 & 0 \\ F_{VP} & F_{VV} & F_{V\psi} & C_b^n & 0 \\ F_{\psi P} & F_{\psi V} & F_{\psi\psi} & 0 & C_b^n \end{pmatrix} \begin{pmatrix} \delta P \\ \delta V \\ \delta\psi \\ b^a \\ b^g \end{pmatrix} \tag{2.30}$$

Components of this state transition matrix are given in the equations (2.31) to (2.44) [2, 10, 20, 21].

$$F_{PP} = \begin{pmatrix} 0 & 0 & -\dfrac{V_N}{(R_N+h)^2} \\ \dfrac{V_E \tan L}{(R_E+h)\cos L} & 0 & \dfrac{-V_E}{(R_E+h)^2 \cos L} \\ 0 & 0 & 0 \end{pmatrix} \tag{2.31}$$

$$F_{PV} = \begin{pmatrix} \dfrac{1}{R_N+h} & 0 & 0 \\ 0 & \dfrac{1}{(R_E+h)\cos L} & 0 \\ 0 & 0 & -1 \end{pmatrix} \tag{2.32}$$

$$F_{P\psi} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{2.33}$$

$$F_{VP} = [F_{VP}(:,1) \;\; F_{VP}(:,2:3)] \tag{2.34}$$

where the columns of the F_{VP} 3x3 matrix are as follows.

$$F_{VP}(:,1) = \begin{pmatrix} \dfrac{-{V_E}^2}{(R_E+h)\cos^2 L} - 2V_E\omega_{ie}\cos L \\ \dfrac{{V_E}^2}{(R_E+h)\cos^2 L} + 2V_N\omega_{ie}\cos L - 2V_D\omega_{ie}\sin L \\ 2V_E\omega_{ie}\sin L \end{pmatrix} \tag{2.35}$$

$$F_{VP}(:,2:3) = \begin{pmatrix} 0 & \dfrac{{V_E}^2 \tan L}{(R_E+h)^2} - \dfrac{V_E V_D}{(R_N+h)^2} \\ 0 & \dfrac{-V_E V_N \tan L - V_E V_D}{(R_E+h)^2} \\ 0 & \dfrac{{V_E}^2}{(R_E+h)^2} + \dfrac{{V_N}^2}{(R_N+h)^2} \end{pmatrix} \tag{2.36}$$

$$F_{VV} = \begin{pmatrix} \dfrac{V_D}{R_N+h} & \dfrac{-2V_E \tan L}{R_E+h} - 2\omega_{ie}\sin L & \dfrac{V_N}{R_N+h} \\ \dfrac{V_E}{R_E+h} + 2\omega_{ie}\sin L & \dfrac{V_N \tan L + V_D}{R_E+h} & \dfrac{V_E}{R_E+h} + 2\omega_{ie}\cos L \\ \dfrac{-2V_N}{R_N+h} & \dfrac{-2V_E}{R_E+h} - 2\omega_{ie}\cos L & 0 \end{pmatrix} \tag{2.37}$$

$$F_{V\psi} = \begin{pmatrix} 0 & -f_D & f_E \\ f_D & 0 & -f_N \\ -f_E & f_N & 0 \end{pmatrix} \tag{2.38}$$

where the f_N, f_E, f_D elements of the $F_{V\psi}$ matrix are can be computed as below.

$$\begin{bmatrix} f_N \\ f_E \\ f_D \end{bmatrix} = C_b^n \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} \tag{2.39}$$

$$F_{\psi P} = \begin{pmatrix} \omega_{ie} \sin L & 0 & \dfrac{V_E}{(R_E + h)^2} \\ 0 & 0 & \dfrac{-V_N}{(R_N + h)^2} \\ \omega_{ie} \cos L + \dfrac{V_E}{(R_E + h) \cos^2 L} & 0 & \dfrac{-V_E \tan L}{(R_E + h)^2} \end{pmatrix} \tag{2.40}$$

$$F_{\psi V} = \begin{pmatrix} 0 & \dfrac{-1}{R_E + h} & 0 \\ \dfrac{1}{R_N + h} & 0 & 0 \\ 0 & \dfrac{\tan L}{R_E + h} & 0 \end{pmatrix} \tag{2.41}$$

$$F_{\psi\psi} = \widetilde{\omega}_{in}^n \tag{2.42}$$

where the columns of the $F_{\psi\psi}$ 3x3 matrix are given below respectively.

$$F_{\psi\psi}(:,1) = \begin{pmatrix} 0 \\ \omega_{ie} \sin L + \dfrac{V_E \sin L}{(R_E + h) \cos L} \\ \dfrac{-V_N}{R_N + h} \end{pmatrix} \tag{2.43}$$

$$F_{\psi\psi}(:,2) = \begin{pmatrix} -\omega_{ie}\sin L - \dfrac{V_E \sin L}{(R_E + h)\cos L} \\ 0 \\ -\omega_{ie}\cos L - \dfrac{V_E}{(R_E + h)} \end{pmatrix} \tag{2.44}$$

$$F_{\psi\psi}(:,3) = \begin{pmatrix} \dfrac{V_N}{R_N + h} \\ \omega_{ie}\cos L + \dfrac{V_E}{(R_E + h)} \\ 0 \end{pmatrix} \tag{2.45}$$

CHAPTER 3

ESTIMATION TECHNIQUE

This chapter of the thesis provides background information about the nonlinear estimation algorithms used in the TRN. In the first section, a short review of the recursive Bayesian estimation process is given with the mathematical models involved. In Sections 3.2 and 3.3, the general structure of the PF and particle selection algorithm, i.e. resampling, are discussed, respectively. The SIR and SIS-R schemes utilized for this thesis are described in the last section.

3.1 Recursive Bayesian Estimation

As generally the case, the aim of the estimation is to determine the posterior probability density of the states by means of some observations. In other words, the estimation process transforms the information in the measurements Y_t to the state x_t. Consider the discrete nonlinear system and measurement model respectively, where x_t is the state vector to be estimated, y_t is the measurement, v_t is the process noise and e_t is the measurement noise [13, 5, 9].

$$\begin{aligned} x_{t+1} &= f_t(x_t) + v_t \\ y_t &= h(x_t) + e_t \end{aligned} \tag{3.1}$$

Alternatively, the state x_t and the measurement y_t can be linked with probability density functions (pdf) as follows,

$$\begin{aligned} x_{t+1} &\sim p(x_{t+1}|x_t) \\ y_t &\sim p(y_t|x_t) \end{aligned} \tag{3.2}$$

where $p(x_{t+1}|x_t)$ is the probability density of the dynamics and $p(y_t|x_t)$ is the measurement likelihood, respectively.

Let $Y_t = \{y_0, y_1, y_2, \dots, y_t\}$ be the set of measurements up to present time t and let $p(x_t|Y_t)$ be the posterior distribution for state x_t conditioned on all the measurements Y_t The probabilistic Bayesian approach gives the solution to calculate the pdf $p(x_t|Y_t)$ that includes all the information about the state. According to the Bayes' rule, the conditional density can be expressed as below [5, 9, 13].

$$p(a|b) = \frac{p(b|a)\, p(a)}{p(b)} \tag{3.3}$$

The functions $p(b|a)$, $p(a)$, and $p(b)$ are called as the likelihood, the prior and the normalizing constant, respectively. Applying this rule to the last measurement of Y_t and assuming that the observation y_t is conditionally independent of old measurements given the state x_t yields the following equation [5, 9, 13].

$$p(x_t|Y_t) = \frac{p(y_t|x_t, Y_{t-1})\, p(x_t|Y_{t-1})}{p(y_t|Y_{t-1})} = \frac{p(y_t|x_t)\, p(x_t|Y_{t-1})}{p(y_t|Y_{t-1})} \tag{3.4}$$

Equation (3.4) is known as the measurement update in the Bayesian estimation. As for the time update step, assuming that x_{t+1} is independent of Y_t when x_t is given, the propagating pdf $p(x_{t+1}|Y_t)$ can be found in a way that in [13, 5, 9].

$$\begin{aligned} p(x_{t+1}|Y_t) &= \frac{p(x_{t+1}, Y_t)}{p(Y_t)} = \int_{\mathbb{R}^n} \frac{p(x_{t+1}, x_t, Y_t)}{p(Y_t)} dx_t \\ &= \int_{\mathbb{R}^n} \frac{p(x_{t+1}, x_t, Y_t)}{p(x_t, Y_t)} \frac{p(x_t, Y_t)}{p(Y_t)} dx_t \\ &= \int_{\mathbb{R}^n} p(x_{t+1}|x_t, Y_t)\, p(x_t|Y_t) dx_t \\ &= \int_{\mathbb{R}^n} p(x_{t+1}|x_t)\, p(x_t|Y_t) dx_t \end{aligned} \tag{3.5}$$

With the equations (3.4) and (3.5), the Bayesian update solution of pdf is described recursively. It can be seen easily from the equations above how the measurement y_t and the state dynamics $p(x_{t+1}|x_t)$ affect the estimation process. In Eqn.(3.4) as the new measurement comes, the previous prediction density $p(x_t|Y_{t-1})$ is multiplied by the likelihood of the measurement. Hence, this results in that the filtering probability

will increase in likely regions and decrease in unlikely regions. In Eqn.(3.5), the filtering density $p(x_t|Y_t)$ is propagated according to the state dynamics and convolved with the probability of the new state with regard to the previous state. Therefore, the filtering density will be shaped with respect to the measurements and the state dynamics [13]. There are different Bayesian techniques given in Figure 18 to solve the measurement and time update steps recursively [22].

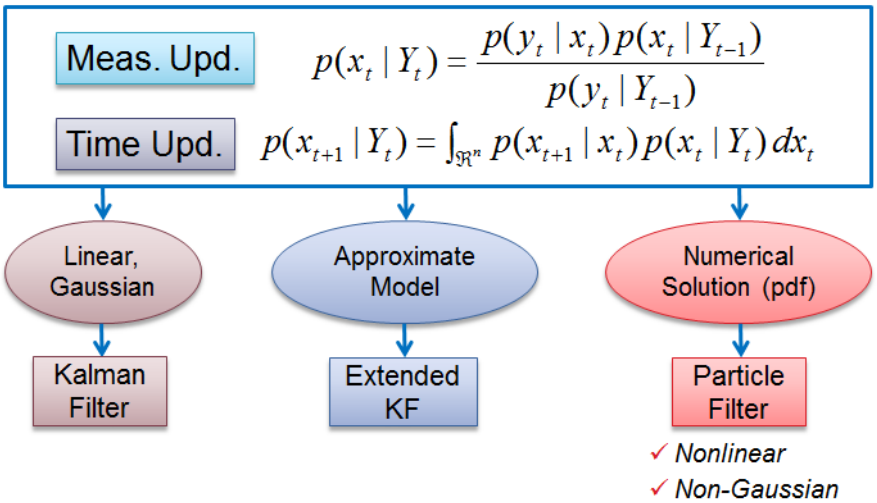

Figure 18 Bayesian estimation techniques [22]

For linear and Gaussian problems, the solution to these steps is the Kalman filter. For nonlinear and Gaussian cases then the problem can be solved by extended Kalman filter, which needs linearization. However, if the problem is nonlinear and non-Gaussian, the problem cannot be solved analytically. Hence, the solution has to be calculated numerically and the solution will be approximate [5]. Therefore, particle filter is selected to solve nonlinear and non-Gaussian TRN problem.

3.2 Particle Filter

Particle filter is an approximate optimal filter on correct model and based on particle representation of probability density [23]. A set of state candidates are drawn statistically considering the related stochastic process. This candidate cloud adapts itself with measurements and evolves in time with system dynamics recursively. A set

of N particles $\left\{x_t^{(i)}\right\}_{i=1}^{N}$ is utilized to approximate the filtering density $p(x_t|Y_t)$. By means of the relative weight $w_t^{(i)}$ associated to each particle such that the sum of all weights equal to unity, every particle carries information about how well it represents the true state. A weighted sample representation of an arbitrary pdf is shown in Figure 19 below [22, 24].

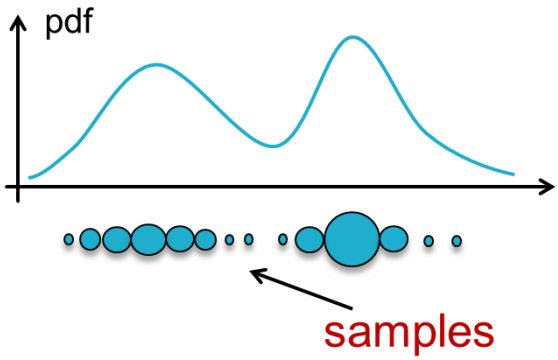

Figure 19 Representation of pdf with samples [22]

As the number of particles becomes very large, the sampled pdf approaches the functional equivalent of the posterior pdf. The idea behind the particle filter is to calculate the posterior density $p(x_t|Y_t)$ numerically with a sum of the Dirac delta functions located in the samples $x_t^{(i)}$ and with importance weights $w_t^{(i)}$. The importance weights can be sequentially updated by Eqn.(3.6) given below, refer to [9, 24] for details,

$$w_t^{(i)} = w_{t-1}^{(i)} \frac{p\left(y_t \middle| x_t^{(i)}\right) p\left(x_t^{(i)} \middle| x_{t-1}^{(i)}\right)}{q\left(x_t^{(i)} \middle| x_{t-1}^{(i)}, Y_t\right)} \tag{3.6}$$

where $q(x)$ is the proposal density from which samples has to be easily drawn [24, 25]. The posterior density can be then calculated as follows

$$p(x_t|Y_t) \approx \sum_{i=1}^{N} \overline{w}_t^{(i)} \delta(x_t - x_t^{(i)}) \tag{3.7}$$

where the normalized importance weights are defined as

$$\bar{w}_t^{(i)} = \frac{w_t^{(i)}}{\sum_{j=1}^{N} w_t^{(j)}}, \quad i = 1,2,\dots,N. \tag{3.8}$$

Finally, the estimated state $\hat{x}_{t|t}$ and its covariance $P_{t|t}$ can be computed as in equations (3.9), (3.10) and (3.11), respectively [25].

$$\hat{x}_{t|t} = E\{x_t\} \approx \sum_{i=1}^{N} \bar{w}_t^{(i)} x_t^{(i)} \tag{3.9}$$

$$P_{t|t} = E\{(x_t - E\{x_t\})(x_t - E\{x_t\})^T\} \tag{3.10}$$

$$\approx \sum_{i=1}^{N} \bar{w}_t^{(i)} \left(x_t^{(i)} - \hat{x}_{t|t}\right)\left(x_t^{(i)} - \hat{x}_{t|t}\right)^T \tag{3.11}$$

This is the basic filtering structure known as the Sequential Importance Sampling (SIS) algorithm.

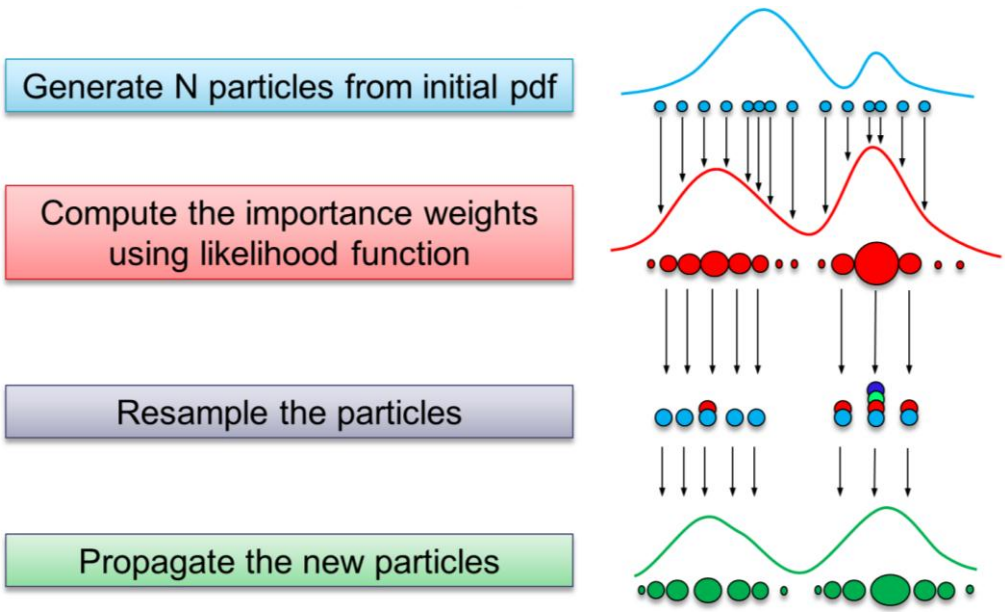

Figure 20 Main steps in particle filtering [22]

Figure 20 illustrates the main steps involved in particle filtering [22, 25]. First step of filtering is the generation of weighted particles from the initial pdf of state. Second step is the computation of the importance weights using the measurement likelihood function. In this representation, the size of the circle is a measure of its weight. As can

be seen from the Figure 20, weights are changing with respect to the region. In high likely region, weights get higher. Next step is resampling or selecting the particles with high importance. Particles with low importance weights die out and particles with high importance weights are multiplied and again weights are reset to 1/ N. Final step is the propagation of the selected particles to the next time step.

3.3 Resampling

After some number of recursive steps, particles will have relatively small weights because of the increasing variance over time. Hence, the accuracy of the filtering is degenerated step by step and excessive computations are dedicated to propagate and update particles whose contribution for estimating posterior density is almost zero. In order to overcome the degeneracy problem, a solution strategy, e.g. resampling, is necessarily implemented [9].

Resampling is a selection algorithm that discards particles with low importance weights and duplicates particles with high importance weights. By means of this selection, only the fittest particles survive and a new set of uniformly weighted most likely candidates is generated. In this work, as a measure of the degeneracy, effective sample size, N_{eff}, is utilized. It is shown in [24] that the effective sample size is estimated by

$$N_{eff} = \frac{1}{\sum_{i=1}^{N}\left(\overline{w}_t^{(i)}\right)^2} \tag{3.12}$$

Resampling stage is done when the effective sample size becomes less than a threshold sample number, N_{th}. In spite of decreasing the influence of degeneracy on the estimation of posterior density, resampling causes lower variance among the particles in the case of low process noise in state dynamics, as the resampled new set may include many repeated particles. In this condition, all candidates may converge to a single particle after a few steps. Some of the issues and detailed explanations of the particle filtering are addressed in [9, 22, 24].

In the literature, different selection algorithms exist for resampling step, i.e., random resampling, residual resampling, systematic resampling [26, 22]. These algorithms are unbiased, but they are different in terms of variance and computational complexity. Among all, systematic method has the lowest resampling variance and needs less computational power [26]. Therefore, systematic algorithm is selected as the resampling method in this work.

In systematic resampling algorithm, each particle takes a unique normalized importance weight, $\overline{w}_t^{(i)}$, between 0 and 1. Then, $(u_1, u_2, \dots, u_N)$, an ordered set of independent and identically distributed N random numbers from the uniform distribution $U[0,1]$ is generated and compared with the cumulative sum of normalized weights. Particles, which exceed the range of random number, will be selected and multiplied with respect to its weight. The systematic resampling algorithm [26, 22] is summarized in Table 3.

Table 3 Systematic resampling algorithm [22]

1. Set i = 1, c_1 = 0 and select a starting point,

$$u_1 \sim U[0, 1/N] \,.$$

2. Compute the cumulative sum of the importance weights

$$c_i = c_{i-1} + w_t^{(i)} \,, \qquad i = 2,3, \dots, N.$$

3. For $j = 1,2, \dots, N$, generate

$$u_j = u_1 + \frac{j-1}{N}$$

- while $u_j > c_i$

  $* \; i = i + 1$

- otherwise

  $* \; set\ sample\ x_t^{(j)} = x_t^{(i)}$

4. Increase j and continue from step 3.

3.4 Applied Filtering Algorithms

Two different particle filter algorithms based on the equations and descriptions above are utilized in this study. The particle filter algorithm that uses the effective sample size in resampling forms the Sequential Importance Sampling with effective resampling (SIS-R) algorithm [5]. In this algorithm, resampling is only performed when actually required. As for the other particle filter implementation, the Sampling Importance Resampling (SIR) algorithm [5], SIR filter is also a derivative form of the SIS algorithm by executing the resampling stage at every time step. The major differences between these implementations are that in SIS-R algorithm, the importance weights are calculated sequentially and the execution frequency of resampling is less than in SIR algorithm. Hence, the SIS-R algorithm has less computational load on the processor [25]. Pseudo codes for both SIS-R and SIR algorithms [5, 25] are given below in Table 4 and Table 5, respectively.

Table 4 Pseudo code for SIS-R algorithm [5]

1. Set t = 0 and generate N samples $\left\{x_0^{(i)}\right\}_{i=1}^{N}$ from the initial distribution $q(x_0)$.
2. Compute the importance weights
$$w_t^{(i)} = w_{t-1}^{(i)} \frac{p\left(y_t \middle| x_t^{(i)}\right) p\left(x_t^{(i)} \middle| x_{t-1}^{(i)}\right)}{q\left(x_t^{(i)} \middle| x_{t-1}^{(i)}, Y_t\right)}, \quad i = 1,2,\dots,N.$$
3. Normalize the weights, i.e., $\overline{w}_t^{(i)} = w_t^{(i)} / \sum_{j=1}^{N} w_t^{(j)}$.
4. If $N_{eff} \geq N_{th}$ move to step 6.
5. Generate a new set $\left\{x_t^{(i^*)}\right\}_{i=1}^{N}$ by resampling with replacement N times from the set $\left\{x_t^{(j)}\right\}_{j=1}^{N}$, with probability $\overline{w}_t^{(j)} = \Pr\{x_t^{(i^*)} = x_t^{(j)}\}$
6. Propagate new particles, i.e., $x_{t+1}^{(i)} \sim q\left(x_{t+1} \middle| x_t^{(i^*)}, Y_{t+1}\right)$, using different noise realizations for the particles.
7. Increase t and continue from step 2.

Table 5 Pseudo code for SIR algorithm [5]

1. Set t = 0 and generate N samples $\left\{x_0^{(i)}\right\}_{i=1}^{N}$ from the initial distribution $p(x_0)$.
2. Compute the importance weights
$$w_t^{(i)} = p\left(y_t \middle| x_t^{(i)}\right) w_{t-1}^{(i)}, i = 1,2,\dots,N.$$
3. Normalize the weights, i.e., $\overline{w}_t^{(i)} = w_t^{(i)} / \sum_{j=1}^{N} w_t^{(j)}$.
4. Generate a new set $\left\{x_t^{(i^*)}\right\}_{i=1}^{N}$ by resampling with replacement N times from the set $\left\{x_t^{(j)}\right\}_{j=1}^{N}$, with probability $\overline{w}_t^{(j)} = \Pr\{x_t^{(i^*)} = x_t^{(j)}\}$.
5. Propagate new particles, i.e., $x_{t+1}^{(i)} \sim p\left(x_{t+1} \middle| x_t^{(i^*)}\right)$, using different noise realizations for the particles.
6. Increase t and continue from step 2.

CHAPTER 4

SENSOR FUSION

In the previous chapter, the recursive Bayesian estimation techniques and the idea behind the particle filtering were explained. In this chapter, the sensor fusion scheme used to solve the nonlinear TRN problem is provided. In Section 4.1, the working principle of TRN is given in order to understand the operation of terrain aided positioning. Once getting the idea, INS correction architecture that integrates the utilized sensors, particle filter and terrain database will be discussed in Section 4.2. Then, the system and measurement models utilized for the PF are described in the preceding sections.

4.1 Operation Principle of TRN

The working principle of TRN illustrated in Figure 21 is to measure the terrain elevations under the path of the vehicle and compare these measurements with an onboard database [13].

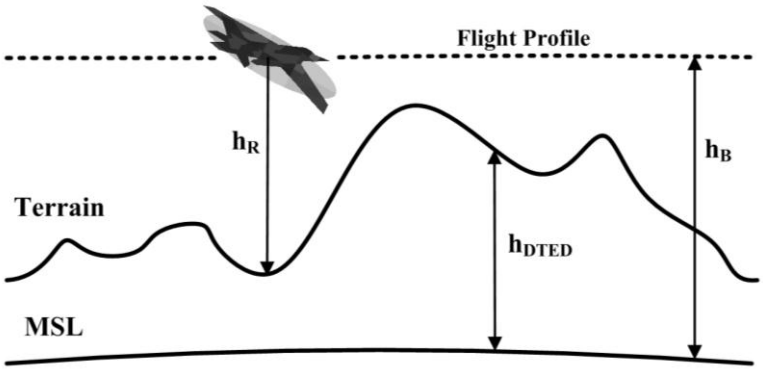

Figure 21 TRN using DTED, barometric and radar altimeter [13]

By means of a barometric altimeter, the altitude of the vehicle over the mean sea level h_B is measured. The ground clearance h_R , i.e. the distance between the vehicle and the ground, is determined by using a radar altimeter. The difference between these measurements gives the terrain elevation measurements. The terrain elevation

value read from the onboard database as a function of latitude and longitude is compared with the measured terrain elevation, thereby determining the vehicle position [5, 13, 25].

4.2 INS Correction Configuration

An error state particle filter structure is developed assuming that the vertical channel of INS is stabilized by another filter, which integrates the INS with a barometric altimeter [11, 12]. Different methods can be utilized to correct INS, e.g. open-loop or closed-loop. In the open-loop method, estimated error states are used in the filter algorithm and not fed back to the INS. INS only solution is also available for integrity monitoring [2]. Open-loop architecture is selected due to its simplicity and shown in Figure 22 [13].

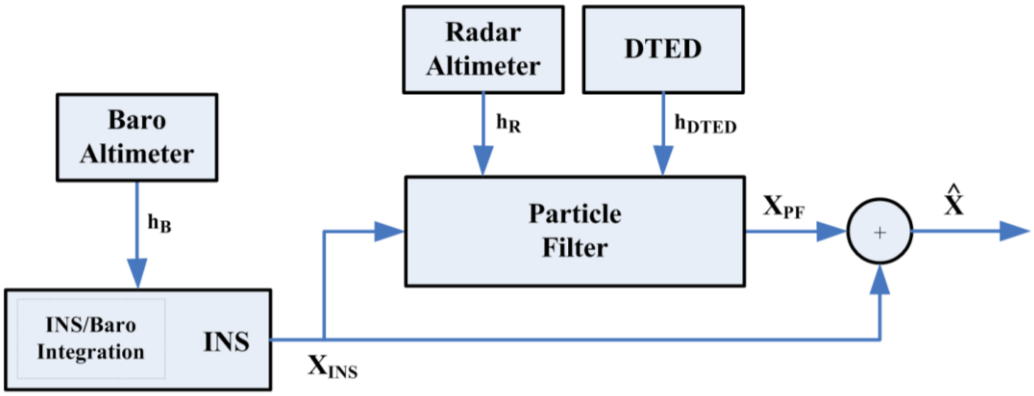

Figure 22 Open-loop correction of INS [13]

The altitude channel of an aircraft is often stabilized with a barometric sensor for a long time even before the global navigation satellite systems, e.g. GPS, GLONASS, are in service. By measuring the static air pressure using a pressure sensor, e.g. transducer, the altitude over mean sea level can be computed by means of standard atmosphere model [2]. A barometric sensor alone is not accurate enough but combined with the data from the INS it provides accurate altitude measurements. This combination is known as INS/Baro integration, which infers the altitude and the vertical velocity component of vehicle [27, 12, 11]. In the scope of this study, it is

reasonable to assume that the INS/Baro integration was already done and the output of this integration only includes the error, which is in the form of a zero-mean white noise. Therefore, the INS altitude error follows the INS/Baro integration error resulting in the equations below,

$$\delta h_B = e_B \tag{4.1}$$

$$\delta V_{D,B} = e_{V_{D,B}} \tag{4.2}$$

where δh_B and $\delta V_{D,B}$ are the INS/Baro integration altitude and vertical velocity errors, e_B and $e_{V_{D,B}}$ stand for the zero-mean normally distributed white noises, which are assumed to have probability densities $N(0, {\sigma_B}^2)$ and $N(0, {\sigma_{V_{D,B}}}^2)$, respectively.

In the operation of terrain aided navigation, the ground clearance need to be measured by utilizing a distance measurement device, e.g. the radar altimeter, as shown Figure 21. The radar altimeter emits the radio signals directly straight down with respect to the vehicle body. The distance to the ground is obtained from the returning signal by using RF signal processing algorithms. Sometimes the beam is reflected anomalously due to the multipath resulting a false measurement [2, 5, 25]. In the scope of this study, it is assumed that there are no false alarms in measurements and the radar altimeter has a measurement error given as follows,

$$\delta h_R = e_R \tag{4.3}$$

where e_R is a zero-mean normally distributed white noise with a probability density $N(0, {\sigma_R}^2)$.

4.3 System Model of the Particle Filter

The error-state system model is obtained by linearizing the nonlinear navigation mechanization equations about the most recent navigation solution as given in Section 2.3.5. The continuous-time expression in Eqn.(4.4) defines the system model for the designed filter,

$$\dot{x}(t) = F(t)x(t) + G(t)v(t) \tag{4.4}$$

where x is the state vector given in Eqn.(4.5), F is the system to state matrix, G is the noise to state matrix, v is the process noise vector.

In the context of baro-aided INS correction configuration, a state space, which is composed of twelve error states namely latitude, longitude, north and east components of velocity vector, roll, pitch, yaw angle errors, accelerometer biases in x and y-axis and gyroscope biases in x, y, z-axis, is constructed as in Eqn.(4.5).

$$x = \left(\delta L, \delta\lambda, \delta V_N, \delta V_E, \delta\psi_R, \delta\psi_P, \delta\psi_Y, b_x^a, b_y^a, b_x^g, b_y^g, b_z^g\right)^T \tag{4.5}$$

Eqn.(4.4) can be written in the state space form as follows [2, 10, 20],

$$\begin{pmatrix} \delta\dot{P} \\ \delta\dot{V} \\ \delta\dot{\psi} \\ \dot{b}^a \\ \dot{b}^g \end{pmatrix} = \begin{pmatrix} F_{PP} & F_{PV} & F_{P\psi} & 0 & 0 \\ F_{VP} & F_{VV} & F_{V\psi} & C_b^n & 0 \\ F_{\psi P} & F_{\psi V} & F_{\psi\psi} & 0 & C_b^n \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \delta P \\ \delta V \\ \delta\psi \\ b^a \\ b^g \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ C_b^n & 0 & 0 & 0 \\ 0 & C_b^n & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{pmatrix} \begin{pmatrix} v^a \\ v^g \\ v^{ab} \\ v^{gb} \end{pmatrix} \tag{4.6}$$

Eqn.(4.4) is in the form of a continuous-time differential equation. Since the filter implementation is a discrete-time process, this equation must be digitized to sample at each time step. The discrete-time form of the equation for the system can be obtained as follows [1],

$$x_{k+1} = \underbrace{\Phi(k+1,k)x_k}_{initial\ condition\ part} + \underbrace{\int_{kT_s}^{(k+1)T_s} \Phi(k+1,\tau)x_k G(\tau)\, v(\tau) d\tau}_{driven\ part\ by\ v(t)} \tag{4.7}$$

where x_k is the state vector, T_s is the sampling time, Φ is the state transition matrix from time step k to $k+1$. For a small enough sampling time, $F(t)$ is assumed to be constant as F_k and the state transition matrix can be discretized with matrix exponential using Taylor series approximation.

$$\Phi(k+1,k) \approx e^{F_k T_s} \approx I + F_k T_s + \frac{1}{2}\,(F_k T_s)^2 + \frac{1}{3!}\,(F_k T_s)^3 + \cdots \tag{4.8}$$

As for the discrete-time equivalent of the noise to state matrix, again for a small sampling time, $G(t)$ is assumed to be constant as G_k and the noise to state transition matrix can be approximated by the expression given below [28].

$$G(k+1,k) = \int_{kT_s}^{(k+1)T_s} (I + ((k+1)T_s - k\tau)F_k)G_k\, d\tau$$

$$= T_s(I + \frac{T_s}{2}F_k)G_k \tag{4.9}$$

Finally the corresponding process noise vector, v, for the system model of the filter is given by the expression below,

$$v = (v^a \quad v^g \quad v^{ab} \quad v^{gb})^T \tag{4.10}$$

where $v^a \sim N(0, \sigma_{v^a}^2)$ and $v^g \sim N(0, \sigma_{v^g}^2)$ are zero-mean Gaussian white noises for accelerometer and gyroscope outputs, $v^{ab} \sim N(0, \sigma_{v^{ab}}^2)$ and $v^{gb} \sim N(0, \sigma_{v^{gb}}^2)$ are zero-mean Gaussian white noises for accelerometer and gyroscope bias, respectively. Eqn.(4.10) is used to define the process noise matrix $Q = E\{vv^T\}$, where E is the expectation operator. Then using the matrix, Q, and sampling time, T_s , the discrete process noise matrix can be approximated as follows [3],

$$Q_k = T_s Q \tag{4.11}$$

4.4 Measurement Model of the Particle Filter

By measuring the pressure using a barometer the altitude over mean-sea level, h_B, can be obtained using the standard atmosphere model. Meanwhile the ground clearance, h_R, i.e. the distance between the aircraft and the ground, is measured using a radar altimeter. The difference between these two measurements provides information on the location where the measurement was collected by comparing with

database. Therefore, the nonlinear measurement model for the terrain referenced navigation can be formed as given below [29].

$$y_k = h_B - h_R = h(L_k, \lambda_k) + e_k \tag{4.12}$$

In 4.8, y_k is the measured terrain height, $h(L_k, \lambda_k)$ is the terrain height obtained from the onboard database as a function of latitude and longitude of the aircraft. In addition, e_k stands for the summation of the zero-mean normally distributed measurement noises of the baro- and radar altimeters, which are assumed to have probability densities $N(0, {\sigma_B}^2)$ and $N(0, {\sigma_R}^2)$, respectively. Moreover, in order to show the implemented particle filter performance in the availability of accurate GNSS horizontal position measurements, i.e. latitude and longitude, an INS/GNSS integration model is configured. This is simply done by changing the measurement model of the particle filter as given the equation below.

$$\begin{gathered} \delta z_k = \begin{pmatrix} L_{GNSS} - \hat{L}_{PF} \\ \lambda_{GNSS} - \hat{\lambda}_{PF} \end{pmatrix}_k \\ L_{GNSS} = L_{REF} + \sigma_{lat} \\ \lambda_{GNSS} = \lambda_{REF} + \sigma_{lon} \end{gathered} \tag{4.13}$$

In 4.13, δz_k is the measurement innovation vector between the GNSS measurements and the estimates of the filter. In addition, σ_{lat} and σ_{lon} are the zero-mean Gaussian distributed latitude and longitude measurement errors of GNSS receiver, which are assumed to be 5 meters [1].

CHAPTER 5

SIMULATIONS AND RESULTS

In this chapter the simulations and the results of the conducted simulations are presented. In the first part, the reference scenario is introduced. Explanations on the used terrain database and IMUs are extended. In addition, a summary of the simulation parameters is given. In the second part, to determine the estimation performance of the particle filter on each state, the results of the single run test case are presented. Moreover, in order to compare the implemented SIR and SIS-R algorithms for TRN, Monte-Carlo simulation method is used because of the stochastic nature of the utilized estimation techniques. Simulations with different number of particles and IMUs having various error measures are performed and investigated. Implemented algorithms are evaluated in terms of horizontal position accuracy and convergence.

5.1 Simulations

In this part, the elements of the simulation are explained. The reference scenario is constructed to compare the implemented filters. Additional information on the terrain map and specifications of the modeled IMUs are presented. Also, design parameters for the filters are given. In order to investigate the estimation performance of the designed nonlinear particle filter on each state, single run is utilized to show whether the PF is suitable or not. Moreover, Monte-Carlo simulations are performed considering the effect of different number of particles and IMUs on the horizontal position estimation performance of the filters. As the number of particles becomes very large, the sampled pdf approaches the functional equivalent of the posterior pdf [13]. Particles of 1000, 2000 and 5000 are utilized to present the influence of the number of particles on TRN. As for showing the effect of IMU on TRN, two different tactical grade IMUs having various error characteristics are used.

5.1.1 Reference Scenario

Before moving to the obtained results, some remarks on the simulated scenario have to be mentioned. In order to setup a reference ground truth solution, scenario flight is simulated without any sensor errors. Hereby, this error-free solution is used to compare with the erroneous INS solution and with the solution of the implemented filters in the following sections. Total scenario takes about 500 seconds. At the beginning, scenario starts with a straight flight to the north direction. Then, the aircraft turns its heading with a rolling maneuver in order to fly to the east direction. Finally, it experiences a roll-to-turn maneuver again to change its heading to the south. The position, velocity and attitude of the generated reference flight over the entire course are represented below as a function of time in Figure 23, Figure 24, Figure 25, respectively.

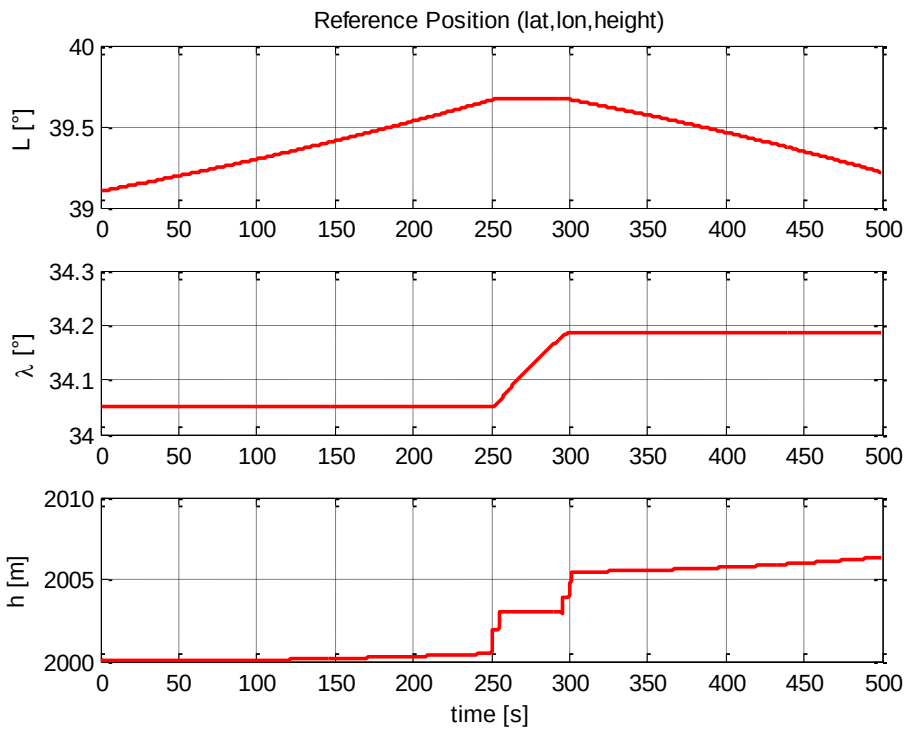

Figure 23 Position vs. time for the reference scenario

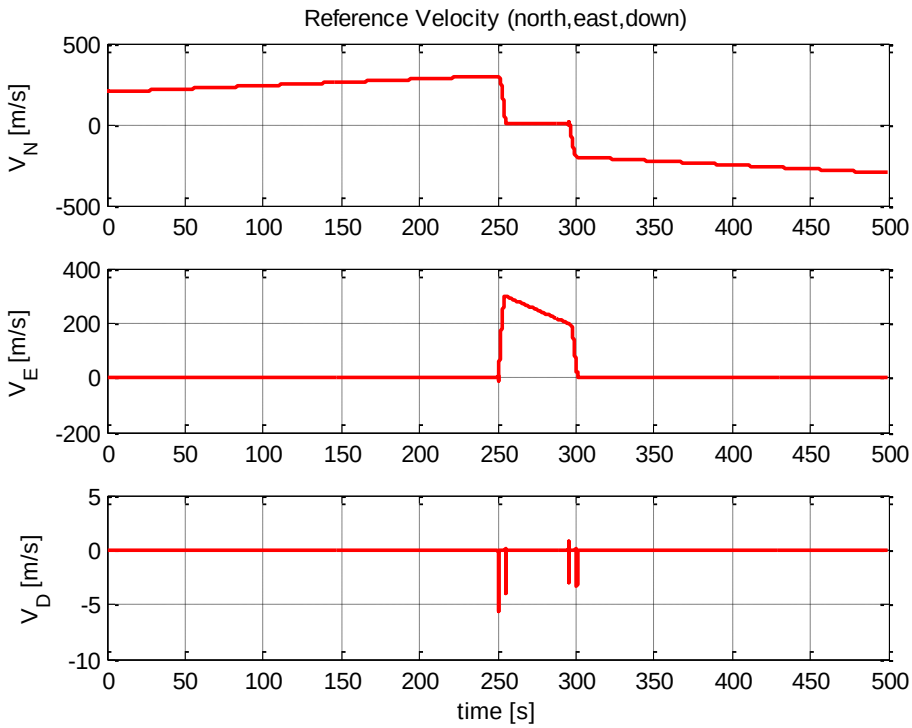

Figure 24 Velocity vs. time for the reference scenario

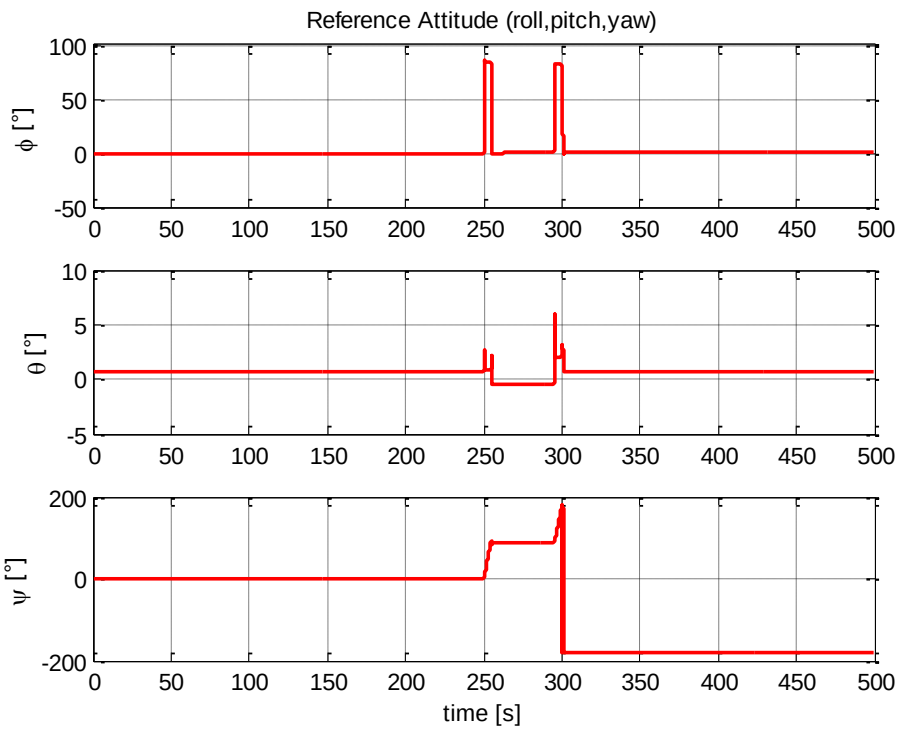

Figure 25 Attitude vs. time for the reference scenario

5.1.2 Terrain Database

The terrain database is critically important for terrain aided positioning applications. Digital terrain elevation databases are used especially in military systems, (e.g., fighter jets, submarines, helicopters), in the case of GNSS failure [29]. Some of the countries around the world produce their different levels of terrain databases. The data is available for horizontal resolutions ranging from about 1 kilometer to 1 meter. The lowest resolution data called Level 0 is available to the public. Generally these databases are generated with respect to mean sea level (MSL), i.e. orthometric height [5]. Most of the military TRN systems use Level 1 Digital Terrain Elevation Data (DTED), which has a grid resolution of about 100m. Almost 2 GB of memory is needed for a whole Earth database [30]. US military standard defines the performance specifications of DTED levels and one can refer to this document for more detailed information [31]. The properties of DTED generated by Harita Genel Komutanlığı (General Commandership for Mapping) are given in Table 6 [32].

Table 6 Specifications of DTED files [32]

Specification	**DTED**	
Level	1	2
Confidential	+	+
Coverage	1°x1°	7.5'x7.5'
Resolution	3"x3"	1"x1"
Datum	MSL	MSL

Filter implementations are tested for an aircraft scenario over a commercial DTED-1 level map. This map belongs to the central part of the country and has a resolution of 3" x 3" in latitude and longitude. In Figure 26 and Figure 27 below, downsampled terrain data and reference aircraft flight scenario used in simulations are illustrated together.

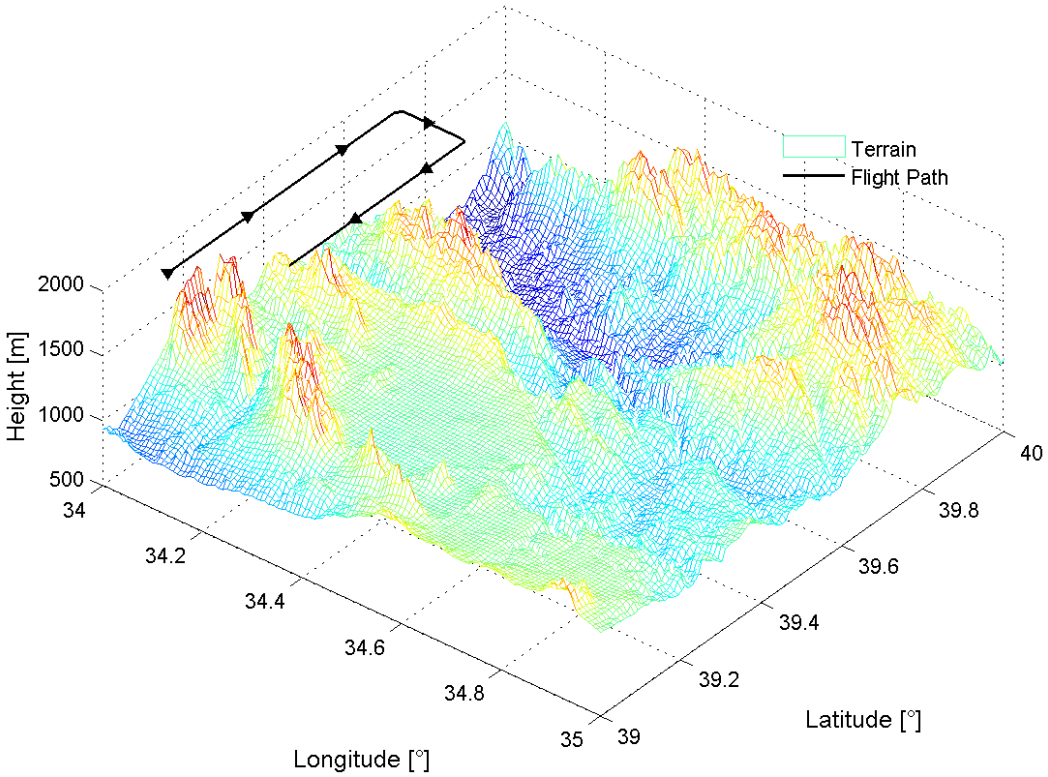

Figure 26 Terrain and reference scenario

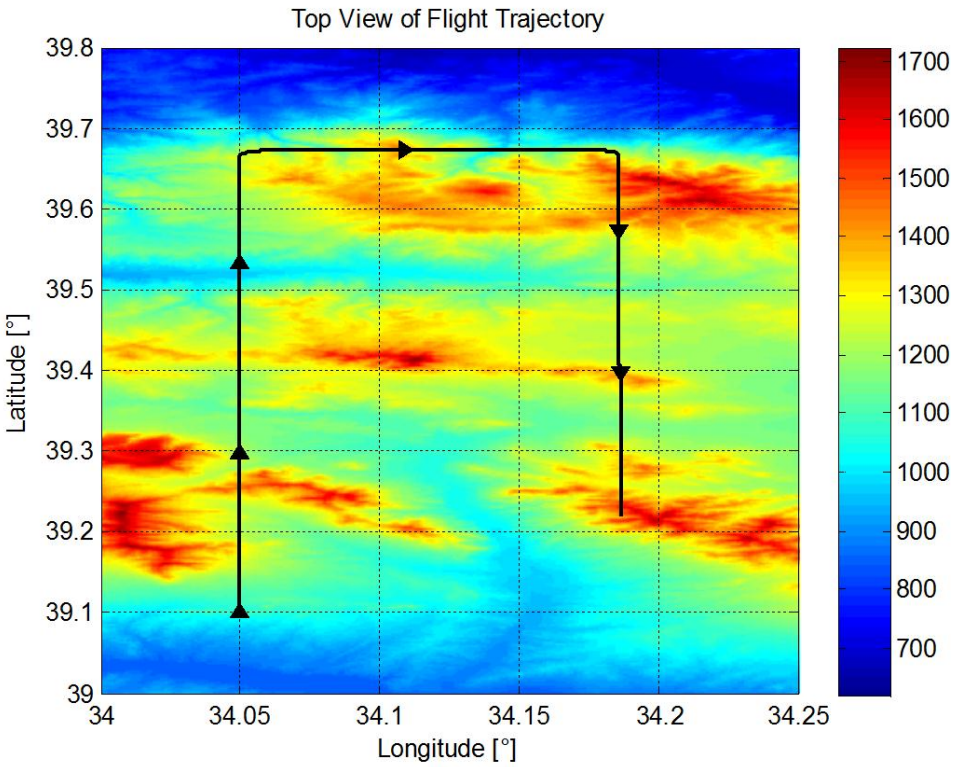

Figure 27 Reference scenario track over the elevation map (top view)

The terrain elevation profile beneath the aircraft is also crucial for the performance of estimation algorithms. This profile needs to contain enough information about where the measurements are taken. The characteristics of the terrain determine the accuracy of the filter, even the convergence. If the region where the aircraft flies over is relatively flat, the filter diverge most probably [13, 29, 5]. This divergence is mainly caused by the lack of required information to distinguish between different positions. Therefore, the terrain profile is taken into account before the simulations are performed.

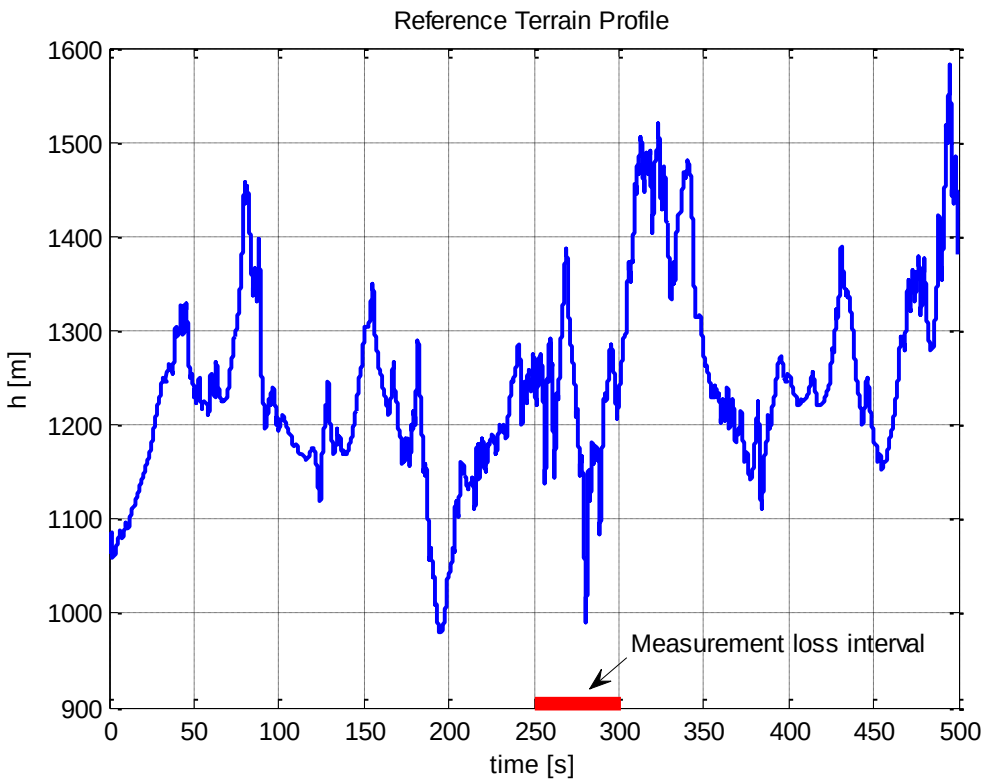

Figure 28 Terrain profile for the reference scenario

Between 250–300 seconds from the start of the scenario, the measurements collected using the radar altimeter are cut synthetically to evaluate the performance of the filters in the loss of data condition. This condition may occur when the aircraft experiences a highly maneuvered flight or there is a malfunction of the radar altimeter [13]. During 50 seconds of this period, no measurement update step is achieved and last valid importance weights are used to resample and to propagate the

particles one step forward. Reference terrain profile for the generated reference scenario and the region of measurement data loss are shown in Figure 28.

5.1.3 Inertial Measurement Units

An IMU model is developed including errors, e.g. bias, misalignment, scale factor and noise parameters as explained in the previous sections. In the content of this work, tactical grade IMUs providing similar performance to HG1930 and HG1700, are selected to be used in the navigation system. Both sensors are shown in Figure 29 below.

Figure 29 HG1930 and HG1700 [33, 34]

The HG1930 is an available commercial off-the-shelf product of Honeywell Inc. and providing industry leading microelectromechanical sensor performance. HG1930 is a MEMS based inertial measurement unit designed to exploit in a range of applications including ground and oceanographic surveying, robotics, control and navigation of unmanned aerial vehicles for both civil and military applications [33]. The HG1700 is in the class of high performance tactical grade IMUs designed to be used in a broad range of guidance and control applications. The HG1700 has three ring laser gyroscopes, three quartz resonating beam accelerometers. The HG1700 also employs an external environmental ring isolator to filter unwanted sensor inputs commonly encountered in real world applications. The HG1700 has been mounted on a wide

range of unmanned air vehicles (UAV), stabilized platforms and commercial applications [34].

In the remaining part of this study, the sensor that shows similar performance to the HG1930 is denoted as IMU1 and the HG1700 is denoted as IMU2. In the sequel, the performance specifications of the selected sensors are given below in Table 7 and Table 8, respectively.

Table 7 Accelerometers error measures

Error	**Unit**	**Accelerometers**	
		IMU1	**IMU2**
Bias $[\boldsymbol{b^a}]$	mg	10	2
Misalignment $[\boldsymbol{M^a}]$	mrad	2	1
Scale Factor $[\boldsymbol{S^a}]$	ppm	300	300
Noise $[\boldsymbol{v^a}]$	$\mathrm{m/s^2/\sqrt{Hz}}$	0.0020	0.0033

Table 8 Gyroscopes error measures

Error	**Unit**	**Gyroscopes**	
		IMU1	**IMU2**
Bias $[\boldsymbol{b^g}]$	°/hr	60	3
Misalignment $[\boldsymbol{M^g}]$	mrad	2	1
Scale Factor $[\boldsymbol{S^g}]$	ppm	300	150
Noise $[\boldsymbol{v^g}]$	$°/\mathrm{s}/\sqrt{\mathrm{Hz}}$	0.0029	0.0050

Before the start of each simulation run, bias, misalignment and scale factor error parameters are selected randomly and used as fixed over the entire simulation. However, as for the random noise error, it is produced for each time step of the run as a white noise sampled from a zero-mean Gaussian distribution and its standard deviation is given by the Table 7 and Table 8 above.

5.1.4 Simulation Parameters Summary

There are different design parameters, which are to be chosen. The threshold value for the effective number of particles (N_{th}), INS/baro integration errors for vertical channel (σ_B and $\sigma_{V_{D,B}}$), radar altimeter measurement noise (σ_R), initial values for the

particle filter estimates (δP_0, δV_0, $\delta\psi_0$, b_0^a and b_0^g), and initial state covariance matrix ($P_{0|0}$) are selected as given in Table 9.

Table 9 Parameters for the particle filter

Parameter	**Value**
IMU type	IMU1, IMU2
$\boldsymbol{N}$	1000, 2000, 5000
$\boldsymbol{N_{th}}$	$2N/3$
$\boldsymbol{T_s}$	0.01s
$\boldsymbol{\sigma_B}$, $\boldsymbol{\sigma_{V_{D,B}}}$, $\boldsymbol{\sigma_R}$	20m, 1m/s, 10m
$\boldsymbol{\delta P_0}$, $\boldsymbol{\delta V_0}$, $\boldsymbol{\delta\psi_0}$, $\boldsymbol{b_0^a}$, $\boldsymbol{b_0^g}$	800m, 0.5m/s, 0.1°/s, 0mg, 0°/hr
$\boldsymbol{P_{0\|0}(\delta P, \delta V, \delta\psi, b^a, b^g)}$	(800m, 1m/s, 0.2°/s, $b^a, b^g)^2$
$\boldsymbol{\sigma_{v^{ab}}}$	$b^a T_s$
$\boldsymbol{\sigma_{v^{gb}}}$	$b^g T_s$

5.2 Results

In this part of the thesis, the results of the conducted simulations are presented. This section is divided into two parts. In the first part, the results obtained from a single run are provided in order to investigate the states estimated by designed filters individually. In the second part, as mentioned in the beginning of this chapter, the results obtained by means of Monte-Carlo simulation technique are presented to illustrate the estimation performance of the filters statistically.

5.2.1 Single Run

The main objective of this part is to investigate the estimation performance of the designed nonlinear particle filter on each state. In other words, the results obtained by utilizing a single run show the estimation performance of particle filter on each state individually.

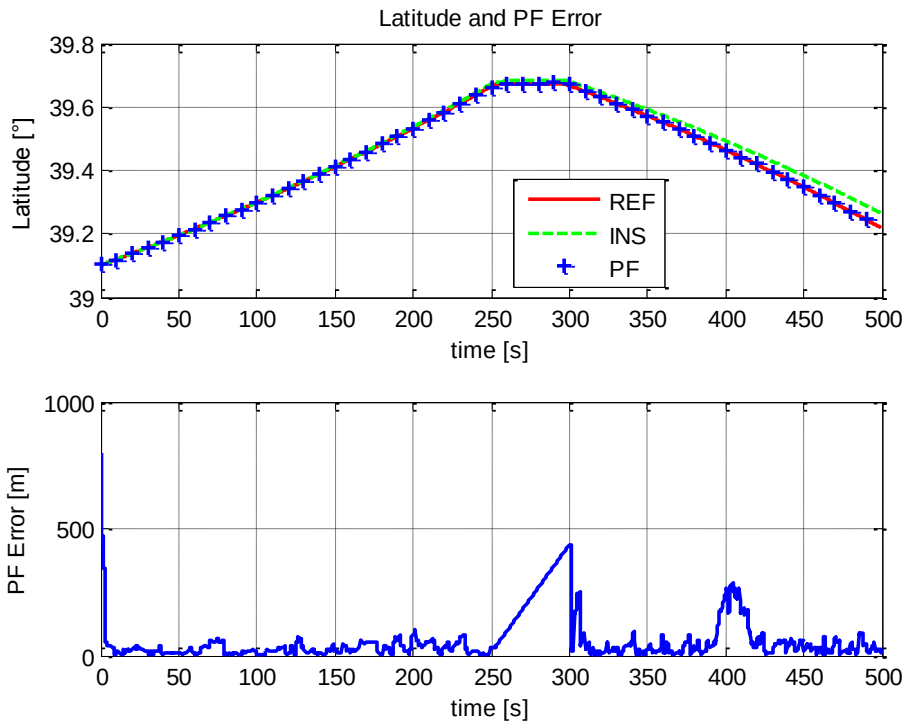

Figure 30 Latitude estimation of filter for a single run

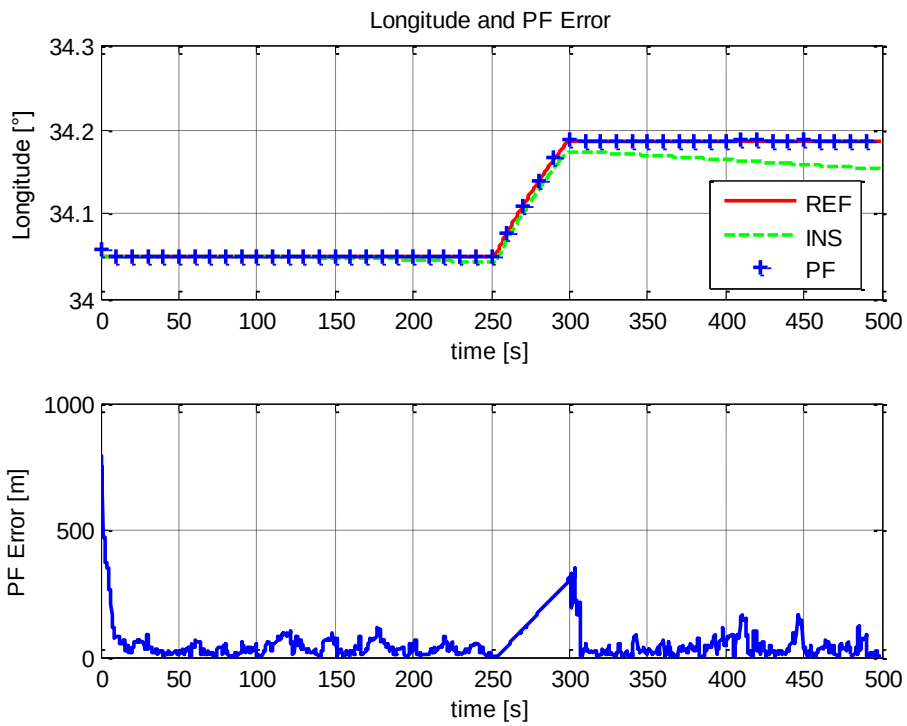

Figure 31 Longitude estimation of filter for a single run

Horizontal position, i.e. latitude L and longitude λ, estimates of the designed filter, reference and erroneous INS position solution obtained from a single run are provided above in Figure 30 and Figure 31, respectively. As can be inferred from these figures, the designed particle filter algorithm tracks the reference path successfully whereas the output of the unaided INS diverges from the true path.

North velocity V_N and east velocity V_E, estimates of the implemented filter, reference and erroneous INS velocity outputs are illustrated below in Figure 32 and Figure 33, respectively. As can be seen from these figures that the filter tracks both reference north and east velocities. However, the velocity estimation of the filter is not very accurate. Velocity estimation error exceeds 10 m/s in some instances although radar altimeter measurements are available. Therefore, the designed particle filter algorithm does not work efficiently in the estimation of velocity states; even the position estimation error of the filter is less than the INS-only case.

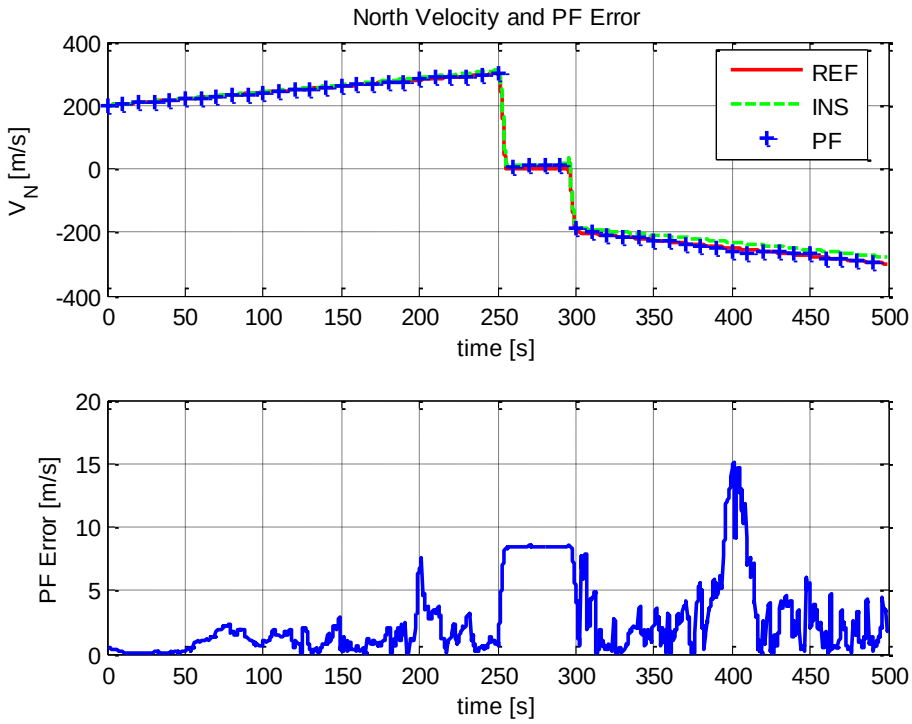

Figure 32 North velocity estimation of filter for a single run

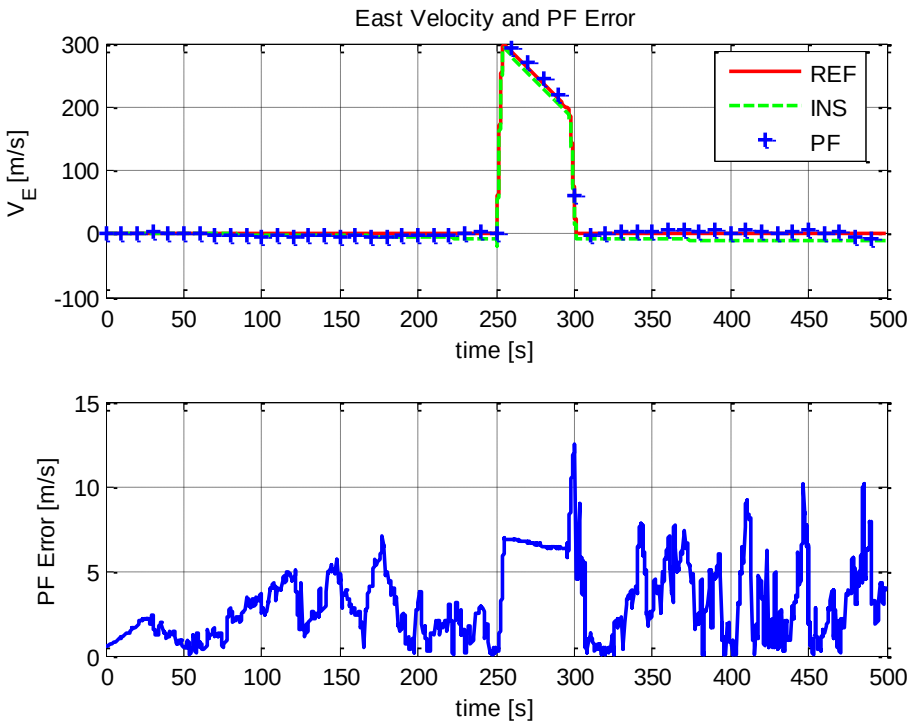

Figure 33 East velocity estimation of filter for a single run

As for the attitude estimates, roll ϕ, pitch θ and yaw ψ, of the utilized filter, reference and unaided INS-only attitude outputs are given below in Figure 34, Figure 35 and Figure 36, respectively. It is clear from these figures that the particle filter algorithm does not accurately estimate the weakly observable attitude errors.

Accelerometer bias a_b and gyroscope bias ω_b estimates of the filter, reference bias values of the utilized sensors are illustrated below in Figure 37 and Figure 38, respectively. As can be seen easily from these figures, the designed particle filter algorithm does not work efficiently in the estimation of weakly observable bias states.

As shown by means of all figures in this part of thesis, implemented particle filter algorithm does not accurately estimate all states except from the observable position states whose information is inherently embedded in the nonlinear DTED database. Therefore, only the horizontal position estimation results are presented in the remaining part of the thesis.

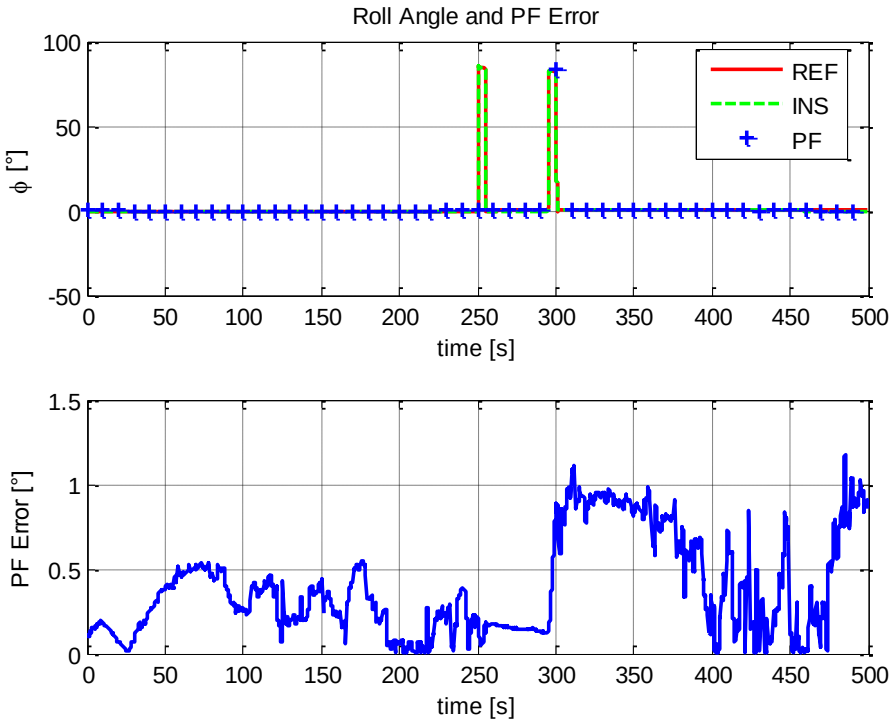

Figure 34 Roll angle estimation of filter for a single run

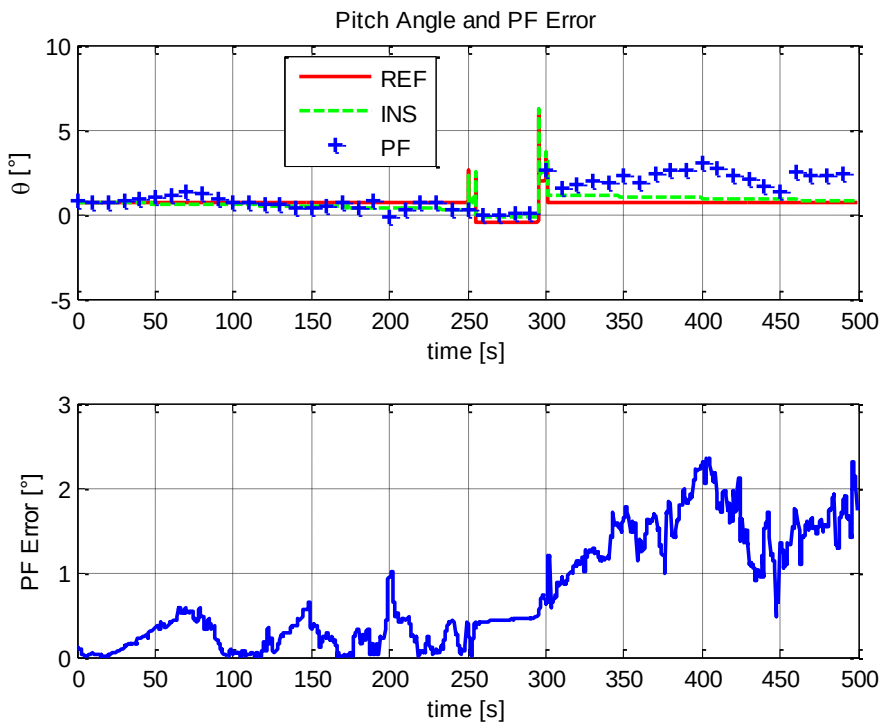

Figure 35 Pitch angle estimation of filter for a single run

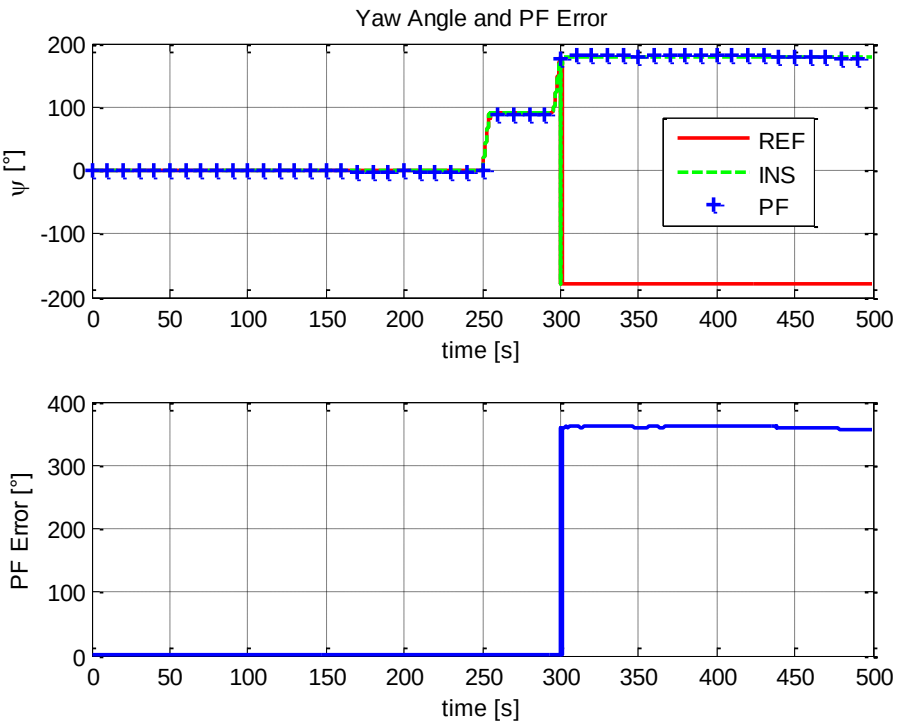

Figure 36 Yaw angle estimation of filter for a single run

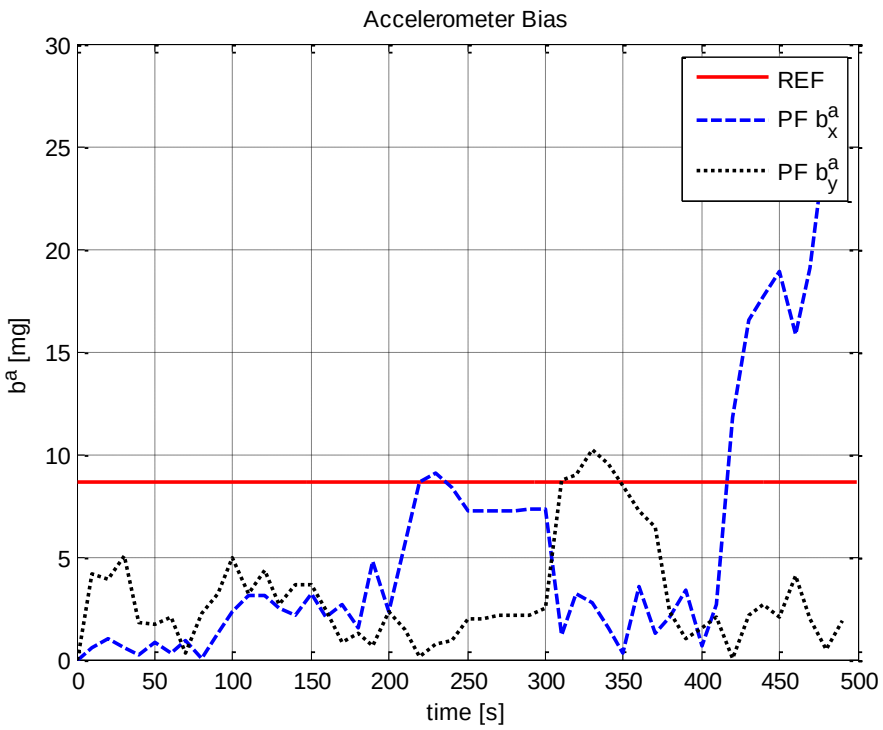

Figure 37 Accelerometer bias estimation of filter for a single run

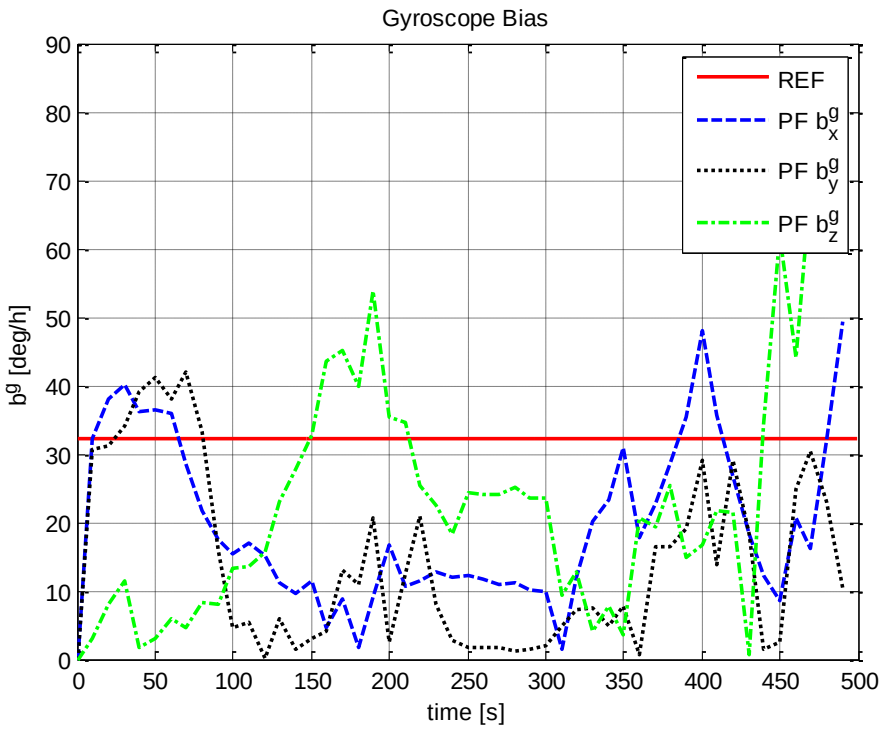

Figure 38 Gyroscope bias estimation of filter for a single run

Before moving to the obtained Monte-Carlo simulation results, it will be quite intuitive to explain how the implemented particle filters work in the simulated scenario over DTED map without equations.

In Figure 39, total simulation is presented from the top view of the map. Colored legend shows the terrain height in meters with respect to the horizontal position. Blue stars represent the position of utilized particles, green and red plus signs represent the filter estimate and true position, respectively. At the initialization step, N number of particles, $\left\{x_0^{(i)}\right\}_{i=1}^{N}$, are generated from the initial distribution $p(x_0)$ and initial errors, δP_0, given in Table 9. After collecting valid measurements from the altimeters, importance weight for each particle is calculated using likelihood function. According to this weight, high likely particles are selected and duplicated, low likely candidates are eliminated by means of resampling algorithm given in Table 3. After resampling selected particles are propagated in time up to the next measurement using the system model of the state dynamics given in Section 4.3.

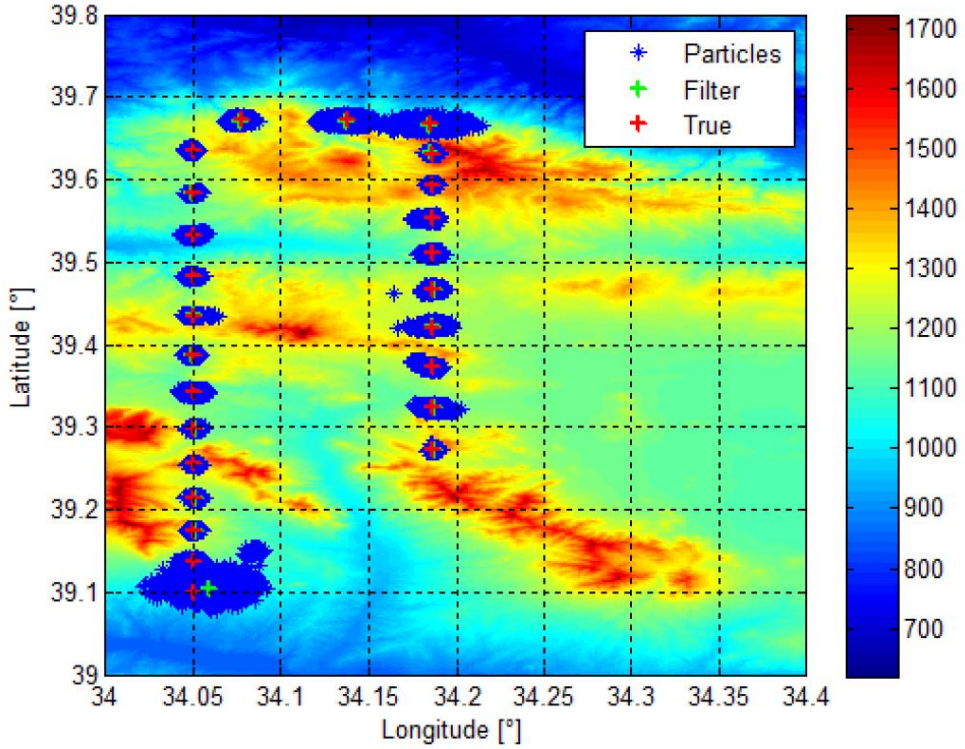

Figure 39 Particle cloud, filter estimate and true position for a single run

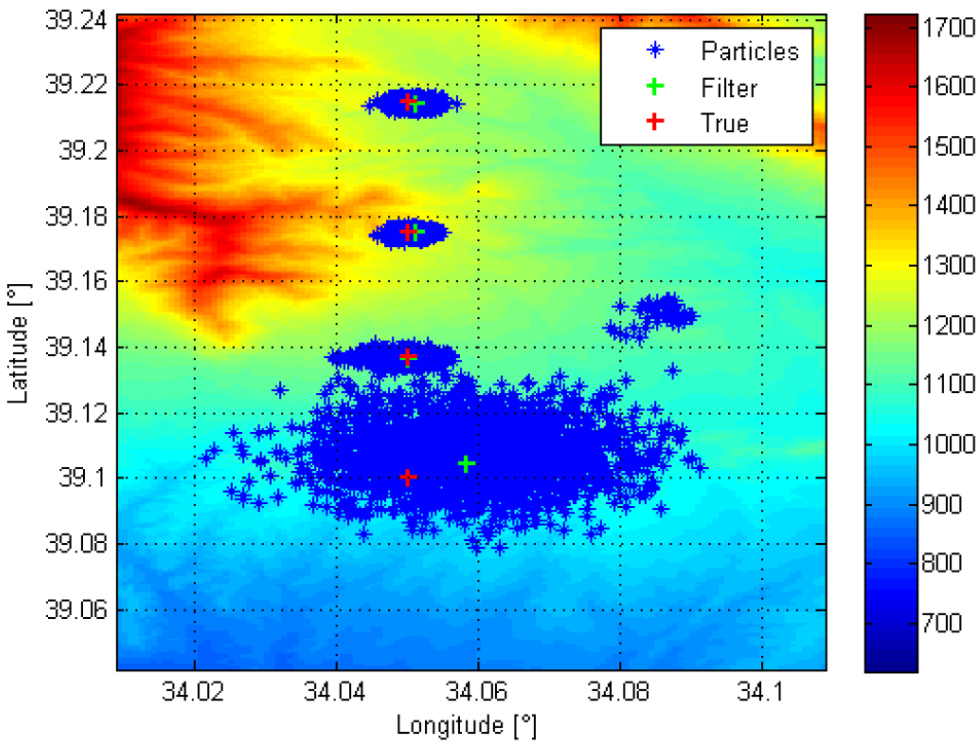

Figure 40 Initialization and early stages of the simulation

Figure 40 shows the initialization and the early stages of the sample simulation run. As seen clearly from this figure, after a few measurement update, the variance, i.e. dispersion, of the particles drops and implemented filter tracks the true position closely.

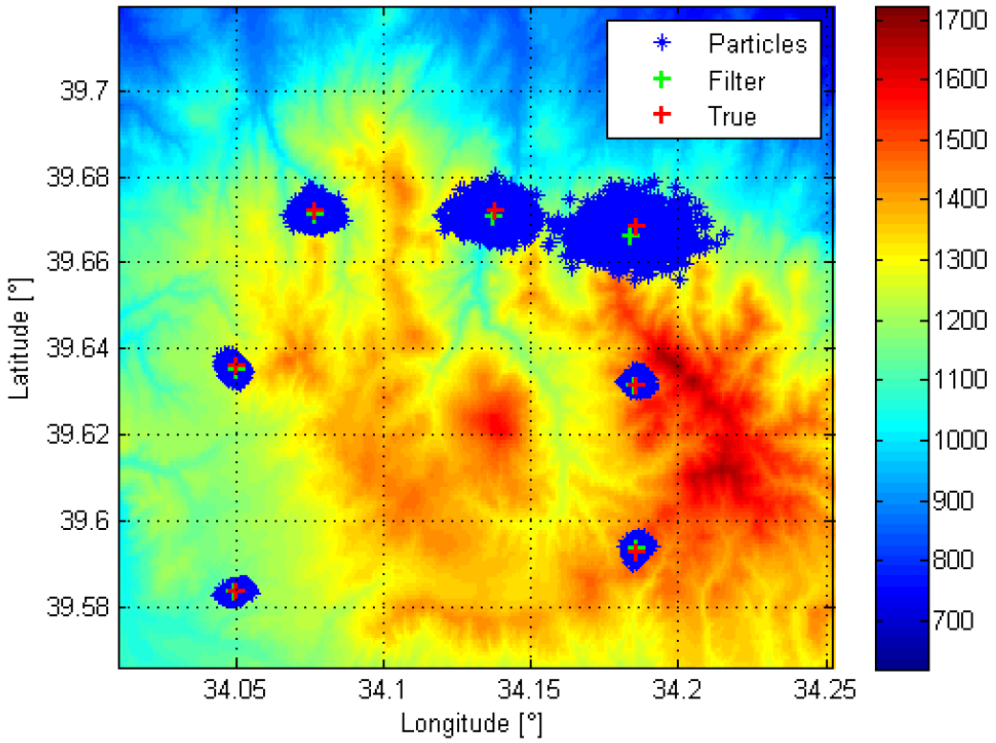

Figure 41 Particles during the measurement loss interval

Figure 41 is taken from the measurement loss interval of the sample simulation run. It can be easily inferred from this figure, once the radar altimeter measurements are lost, the variance of the particles starts to increase due to the free propagation of error with time according to state dynamics. Due to this dispersion, the estimation accuracy of the particle filter is deteriorated as expected. After a few measurement update, the variance, i.e. dispersion, of the particles drops and implemented filter tracks the true position closely. After collecting valid measurements from the radar altimeter, the aircraft flies over the suited rough terrain, the performance of the filter is recovered again.

5.2.2 Monte-Carlo Runs

In this section, the results obtained by means of 100 Monte-Carlo runs are given to show the estimation performance of the filters statistically. Monte-Carlo results of both algorithms are evaluated in terms of convergence and horizontal position accuracy.

5.2.2.1 Convergence Comparison

In this part of the work, the convergence results of both SIR and SIS-R filters are presented and investigated. Filter divergence is determined by comparing the sum of the importance weights with a threshold value, which is very close to zero, e.g. 10^{-8}. If the sum is less than this threshold value then the filter is said to be diverged [13, 35]. In Table 10, after utilizing these filters in MC simulation method, the number of converged runs out of 100 simulations is tabulated.

Table 10 Convergence results of Monte-Carlo runs

Number of Particles	**SIR**		**SIS-R**	
	IMU1	***IMU2***	***IMU1***	***IMU2***
1000	63	75	60	74
2000	86	93	84	92
5000	92	99	91	99

According to Table 10, if the IMU type is fixed, with the increasing number of particles, the convergence performance of the both filters increases for all cases. This result means that the larger the number of particles becomes, the better the pdf is represented and supports the explanations mentioned previous sections. Moreover, by increasing the quality of the inertial sensors from IMU1 to IMU2, the robustness of both algorithms is considerably improved for a given fixed number of particles. This is mainly caused by the lower process noise of the better quality inertial sensors. For the case in which poorer performance sensors are utilized, higher process noise leads the particle cloud to become more diverged from the true path. During all simulations

with smallest sample size of 1000 and lower quality IMU1, tracking of the true state is lost frequently as almost 40%. For the largest sample size, the divergence of the filters occurs rather seldom and only 1% with the higher quality IMU2. SIR and SIS-R algorithms yield similar results for a given number of particles and fixed IMU type. Therefore, it can be deduced from these results that the robustness of both algorithms is almost equal.

5.2.2.2 Horizontal Position Comparison

In this section of the thesis, the horizontal position accuracy comparison of both SIR and SIS-R algorithms is presented. As can be inferred from Table 10 in the previous section, for the same sample size and IMU configuration, both filters perform almost equally in terms of convergence. However, these algorithms do not work efficiently in cases where the smallest sample size of 1000 particles is selected and the lower quality IMU1 is utilized. Hence, the results obtained with the larger sample sizes of 2000, 5000 particles are presented only. After all MC runs finished, in order to calculate the Root Mean Square (RMS) error, the results from the diverged cases are removed from all runs and the remaining converged simulations are used in computing the estimation error of the related filter. Therefore, the simulations succeeding in 80% or higher convergence are illustrated for the confidence in comparison. The horizontal position RMS error can be calculated by Eqn.(5.1), where C_R represents the number of converged runs [5].

$$RMSE_t = \left(\frac{1}{C_R} \sum_{i=1}^{C_R} \left\| \hat{x}_t^{(i)} - x_t^{ref} \right\|^2 \right)^{\frac{1}{2}} \tag{5.1}$$

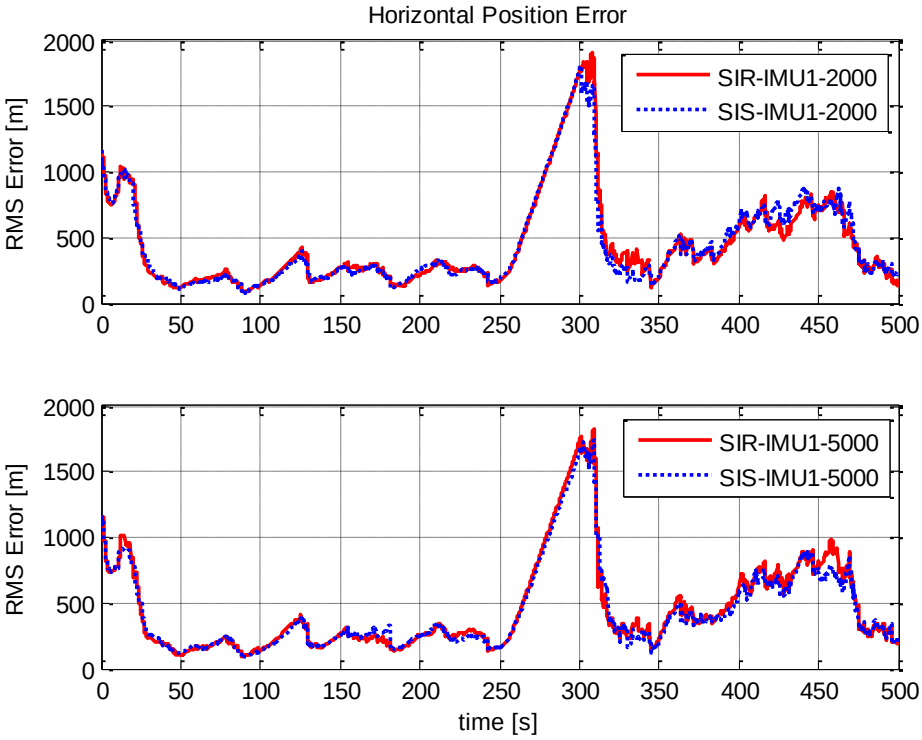

Figure 42 SIR and SIS-R horizontal errors for IMU1 and N = {2000, 5000}

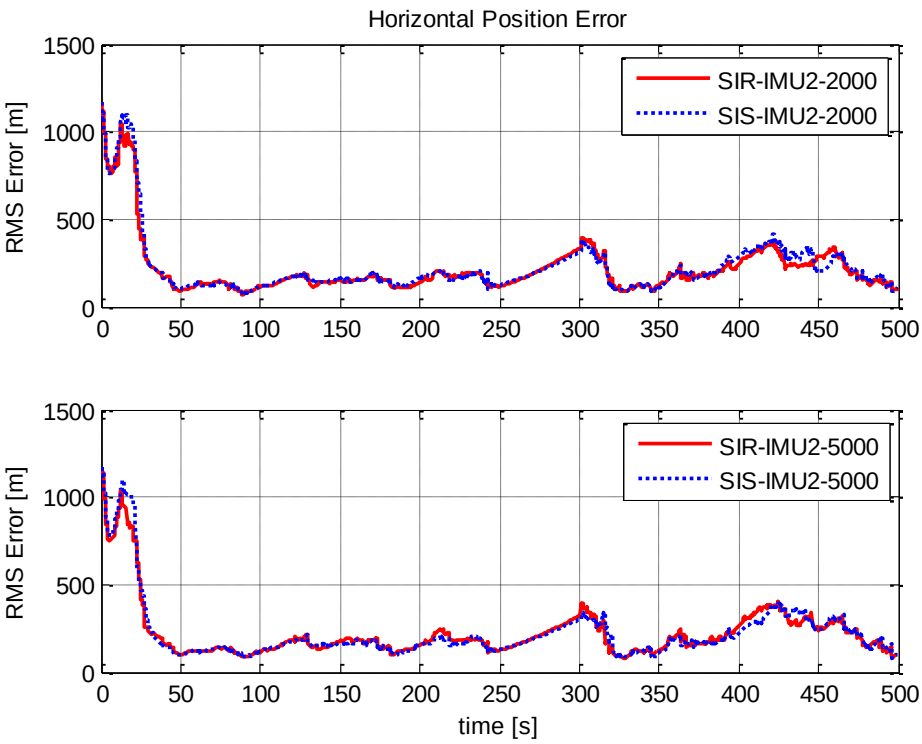

Figure 43 SIR and SIS-R horizontal errors for IMU2 and N = {2000, 5000}

Figure 42 and Figure 43 show the horizontal position errors for the two filters compared to reference flight profile when they are simulated selecting the same IMU type and utilizing the equal number of particles. It can be deduced from these figures that the SIS-R filter produces accurate estimates as the SIR filter does in all cases.

After presenting that the SIS-R filter yields comparable results to those of the SIR filter, we move onto the results gathered by exploiting the SIS-R algorithm only. For all cases in Figure 44 and Figure 45, the SIS-R filter works efficiently over the suitable terrain profile with an estimation accuracy of slightly more than 100 meters, e.g. close to the dimension of the map cell. However, as seen from these figures, there are two remarkable regions where the performance of the filter decreases. The first one is the region where the radar altimeter measurements are unavailable between 250–300 seconds of the flight. The second region, along the duration around 350–420 seconds of the simulation, corresponds to the relatively repetitive terrain in Figure 27 and Figure 28. Actually, this condition can be thought as a similar case when the aircraft flies over a relatively flat region because repetitive terrain surfaces do not include the required information to make the particles distinguish between different positions [13]. Hence, the horizontal position error grows significantly over these regions as expected. After collecting valid measurements from the radar altimeter and passing over the repetitive region, the performance of the filter is recovered again by flying over the suited rough terrain. Furthermore, in these figures, the result of the configured INS/GNSS integration utilizing SIS-R particle filter is presented as a lower bound for the implemented TRN algorithms. Since the GNSS position measurements are quite accurate, the outcome of the INS/GNSS integration with SIS-R filter also more accurate than our algorithm as expected. It is important to note that the performance of the implemented TRN algorithm is closely dependent on the radar and barometric altimeter measurement errors. Therefore, horizontal position estimation error of our TRN algorithm can be reduced by utilizing more accurate altimeter sensors to get closer to the lower bound of INS/GNSS integration.

Figure 44 shows the effect of the IMU type on the estimation accuracy of the SIS-R filter for a fixed sample size. As the quality of the sensors from IMU1 to IMU2

increases, filter accuracy is also improved typically for all number of particles. The lower process noise of the higher quality inertial sensors of IMU2 permits the particle cloud to concentrate around the true path. However, it is important to note that if this process noise is too low with respect to the resolution of the map, the particle cloud shrinks easily into a single map cell. This situation may cause to the lack of diversity among the particles, finally to the filter divergence. Hence, the IMU quality and the dimension of the map cell have to be considered together in the design phase of the filter and if needed artificial noise can be introduced to the system dynamics. In this work, 0.005 percent of the estimated covariance is empirically added to the system.

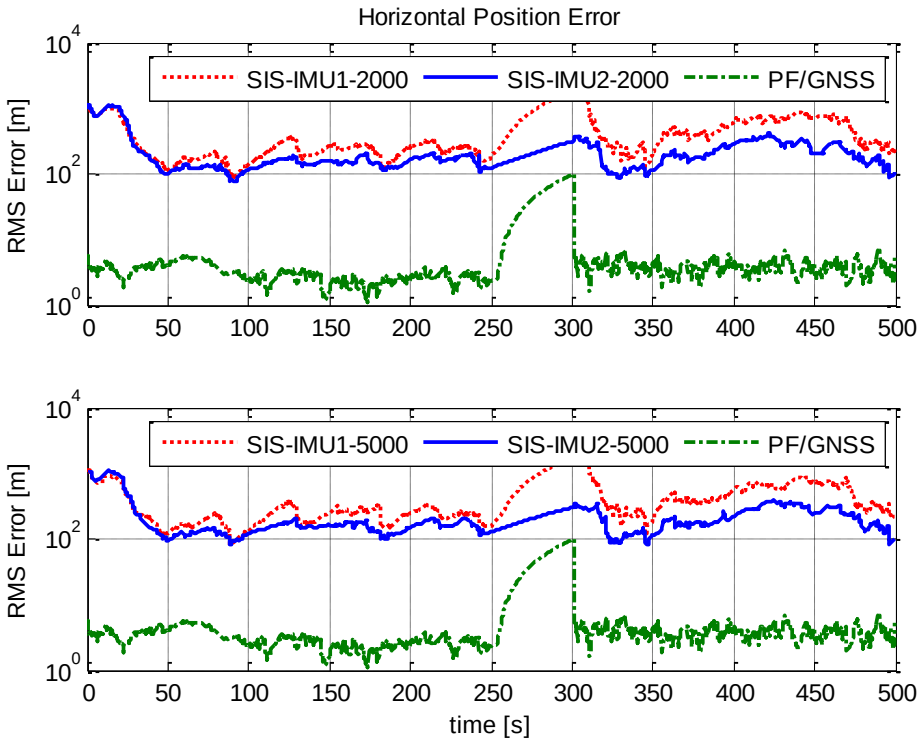

Figure 44 SIS-R horizontal errors for IMU1 and IMU2 (fixed sample size)

Figure 45 presents the obtained results from the SIS-R filter with the different number of particles for a fixed IMU type. As can be inferred from this figure, the larger sample size of 5000 particles does not improve the estimation accuracy significantly in general, whereas it is more robust than the sample size of 2000 as already known from the previous part Table 10.

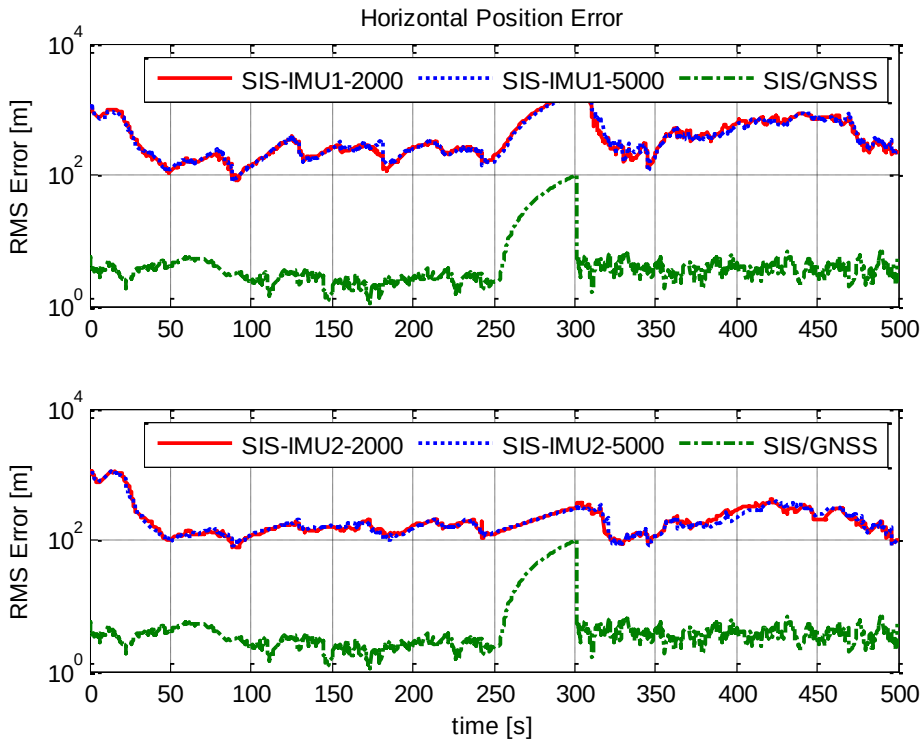

Figure 45 SIS-R horizontal errors for N = {2000, 5000} (fixed IMU type)

CHAPTER 6

DISCUSSIONS AND FUTURE WORK

In this thesis, sensor fusion algorithms based on particle representation of probability density have been utilized. These methods provide an alternative approximate solution to the Bayesian recursive estimation by discretizing the probability density functions involved unlike classical linear methods. In the context of study, two different implementations of particle filtering, i.e. SIR and SIS-R, are utilized to solve the nonlinear and non-Gaussian terrain aided navigation problem. The results presented in Chapter 5 indicate that how well these algorithms perform for the state estimation considering the effect of IMU selection. At the end, this chapter discusses the potential of the particle filter algorithm and gives some suggestions on future work.

6.1 Discussions

The main objective of this thesis has been to show the effect of IMU and number of particles on the performance of the particle filter algorithms in terrain referenced navigation. First of all, by means of this thesis it is shown that the IMU quality and the dimension of the map cell have to be considered together in the design phase of the filter. In addition, the SIS-R routine gives almost equal results as the SIR filter does in terms of the filter convergence and horizontal position estimation accuracy. The comparison of the SIS-R algorithm with the SIR showed very similar performance in cases when there is a measurement loss problem or when the aircraft flies over the repetitive terrain. Therefore, it can be concluded that besides putting less burden on the computational resources the SIS-R particle filter works robustly and accurately in the terrain aided positioning problem.

The results also showed that applied particle filter algorithms for the solution of terrain aided navigation problem allow to estimate the observable nonlinear position states whereas the weakly observable linear states, i.e. velocity, attitude and sensor

biases, were not accurately estimated. It is important to note that filter needs time to converge and work efficiently, then throughout this period suitable terrain profile is inevitably necessary. In addition, by using more accurate altimeter sensors, which enable the filter to distinguish between the different positions, the estimation performance of the implemented TRN algorithm can be closer to the INS/GNSS integration lower bound.

All of the work done to here is based on virtual simulation results. Therefore, fine parameter tunings may be required to get desired performance from the utilized filters on real flight tests.

6.2 Future Work

In aircraft navigation, accurate and reliable navigation solution is crucial. This thesis has rarely mentioned about how the unobservable states can be estimated accurately. The unobservable state estimation accuracy of the implemented algorithm is one issue that needs to be investigated further. While the idea behind the particle filter is quite simple, the main problem is that it puts a burden on mission computer. With the increasing state dimension, computational requirement also increases rapidly. In order to solve this problem, a marginalized filter that combines both particle and Kalman filter may be utilized [8, 36]. By means of this marginalization, particle filter can handle the nonlinear states and the remaining linear states can be inferred with Kalman filter. Moreover, if higher resolution height database, e.g. DTED Level 2, obtained, the effect of database resolution on the estimation accuracy can be investigated in the future studies. Furthermore, if applicable, by means of flight tests assisted with GNSS equipment, implemented algorithms may be validated using real sensors.

REFERENCES

[1] Grewal, Mohinder S.; Andrews, Angus P.; Bar, Chris G., Global Navigation Satellite Systems, Inertial Navigation, and Integration, Wiley, 2013.

[2] P. D. Groves, Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems, Artech House, 2008.

[3] Gleason, S; Gebre-Egziabher, D., GNSS Applications and Methods, Artech House, 2009.

[4] K. B. Anonsen, Advances in Terrain Aided Navigation for Underwater Vehicles, PhD. Thesis, Norwegian University of Science and Technology, 2010.

[5] N. Bergman, Recursive Bayesian Estimation Navigation and Tracking Applications, PhD. Thesis, Linköping University, 1999.

[6] Wan, E.A; van der Merwe, R., "The Unscented Kalman Filter," in *Kalman Filtering and Neural Networks*, New York, Wiley, 2001, pp. Chapter 7, S. Haykin (Ed.).

[7] Hostetler, L.; Andreas,R., "Nonlinear Kalman Filtering Techniques for Terrain-Aided Navigation," *IEEE Transactions on Automatic Control*, vol. 28, issue 3, pp. 315-323, March 1983.

[8] P. J. Nordlund, Recursive Estimation of Three-Dimensional Aircraft Position Using Terrain-Aided Positioning, Technical Report, Department of Electrical Engineering, Linköping University, 2001.

[9] Ristic, B.; Arulampalam, S.; Gordon, N.J. , Beyond the Kalman Filter: Particle Filters for Tracking Applications, Artech House, 2004.

[10] J. Farrell, Aided Navigation: GPS with High Rate Sensors, New York: McGraw-Hill, 2008.

[11] J. S. Ausman, "Baro-Inertial Loop for the USAF Standard RLG INU," *NAVIGATION, Journal of the Institute of Navigation*, vol. 38, no. 2, pp. 205-220, 1991.

[12] J. S. Ausman, "A Kalman Filter Mechanization for the Baro-Inertial Vertical Channel," *Proceedings of the 47th Annual Meeting of the Institute of Navigation*, pp. 153-159, June 1991.

[13] Turan, B.; Kutay, Ali T., "Particle Filter Studies on Terrain Referenced Navigation," in *Proceedings of IEEE/ION PLANS 2016*, Savannah, GA, April 2016.

[14] Aggarwal, P.; Syed, Z.; Noureldin, A.; El-Shimy, N., MEMS-Based Integrated Navigation, Artech House, 2010.

[15] Eren, M.; Davison, S.; Schmidt, G.; et al., "Basic Guide to Advanced Navigation," NATO RTO Publication, 2003.

[16] Titterton, David H.; Weston, John L., Strapdown Inertial Navigation Technology - 2nd Edition, Reston: The American Institute of Aeronautics and Astronautics, 2004.

[17] Barbour, N.; Schmidt, G., "Inertial Sensor Technology Trends," *IEEE Sensors Journal*, vol. 1, no. 4, pp. 332-339, 2001.

[18] B. Altinöz, Identification of Inertial Sensor Error Parameters, M.Sc. Thesis, METU, 2015.

[19] Brown, Robert G.; Hwang, Patrick Y. C., Introduction to Random Signals and Applied Kalman Filtering - 4th Ed., John Wiley & Sons, Inc., 2012.

[20] T. Akça, An Adaptive Unscented Kalman Filter for Tightly Coupled INS/GPS Integration, M. Sc. Thesis, Middle East Technical University, 2012.

[21] Noureldin, A.; Karamat, T. B.; Georgy, J., Fundamentals of Inertial Navigation, Satellite-based Positioning and their Integration, Berlin: Springer-Verlag, 2013.

[22] D. J. Lee, Nonlinear Bayesian Filtering with Applications to Estimation and Navigation, PhD. Thesis, Texas A&M University, May 2005.

[23] Arulampalam, M. S.; Maskell, S.; Gordon, N.; Clapp, T., "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174-188, 2002.

[24] Doucet, A.; De Freitas, N.; Gordon, N., Editors, Sequential Monte Carlo Methods in Practice, Springer, 2001.

[25] R. Karlsson, Particle Filtering for Positioning and Tracking Applications, PhD. Thesis, Linköping University, 2005.

[26] J. D. Hol, Resampling in Particle Filters, MSc. Thesis, Linköping, Sweden: Linköping University, 2004.

[27] Widnall, W. S.; Sinha, P. K., "Optimizing the Gains of the Baro-Inertial Vertical Channel," *Journal of Guidance, Control and Dynamics, AIAA*, vol. 3, no. 2, pp. 172-178, 1980.

[28] P. J. Nordlund, Sequential Monte Carlo Filters and Integrated Navigation, PhD. Thesis, Linköping University, 2002.

[29] Bergman, N.; Ljung, L.; Gustafsson, F., "Terrain Navigation using Bayesian Statistics," *IEEE Control Systems*, vol. 19, no. 3, pp. 33-40, 1999.

[30] Perrett, M.; Krempasky, J., "Terrain Aiding for Precision Navigation in Heavy GPS Jamming," in *Proceedings of ION GPS 2001*, Salt Lake City, UT, September 2001.

[31] MIL-PRF-89020A, "Performance Specification Digital Terrain Elevation Data," US Department of Defense, 1996.

[32] "Sayısal Coğrafi Ürün Örnekleri," Harita Genel Komutanlığı, [Online]. Available: http://www.hgk.msb.gov.tr/u-6-sayisal-cografi-urun-ornekleri.html. [Accessed January 2017].

[33] "HG1930 Inertial Measurement Unit," Honeywell International Inc., [Online]. Available: https://aerospace.honeywell.com/en/products/navigation-and-sensors/hg1930. [Accessed February 2017].

[34] "HG1700 Inertial Measurement Unit," Honeywell International Inc., [Online]. Available: https://aerospace.honeywell.com/en/products/navigation-and-sensors/hg1700. [Accessed February 2017].

[35] T. Karlsson, Terrain Aided Underwater Navigation using Bayesian Statistics, MSc. Thesis, Linköping University, 2002.

[36] Schön, T.; Gustafsson, F.; Nordlund, P., "Marginalized particle filters for mixed linear/nonlinear state-space models," *IEEE Transactions on Signal Processing*, vol. 53, no. 7, pp. 2279-2289, 2005.