TERRAIN AIDED NAVIGATION

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

BY

ALPER KARABÖRK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2010

Approval of the thesis:

TERRAIN AIDED NAVIGATION

submitted by **ALPER KARABÖRK** in partial fulfillment of the requirements for the degree of
Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University by,

Prof. Dr. Canan Özgen ______________
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. İsmet Erkmen ______________
Head of Department, **Electrical and Electronics Engineering**

Prof. Dr. Mübeccel Demirekler
Supervisor, **Electrical and Electronics Engineering Dept., METU** ______________

Examining Committee Members:

Prof. Dr. Yalçın Tanık
Electrical and Electronics Engineering Dept., METU ______________

Prof. Dr. Mübeccel Demirekler
Electrical and Electronics Engineering Dept., METU ______________

Prof. Dr. Mete Severcan
Electrical and Electronics Engineering Dept., METU ______________

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

Aykut Arıkan
Senior Design Leader, ASELSAN Inc. ______________

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: ALPER KARABÖRK

Signature :

ABSTRACT

TERRAIN AIDED NAVIGATION

Karabörk, Alper
M.S., Department of Electrical and Electronics Engineering
Supervisor : Prof. Dr. Mübeccel Demirekler

September 2010, 96 pages

An Inertial Navigation System (INS) can individually produce the navigation data, i.e. position, velocity, of the aircraft without any help or aid. However, a large number of errors are introduced by sensors causing to an unacceptable drift in the output. Because of this reason, external aids are used to correct INS. Using these aids an integrated navigation structure is developed.

In an integrated navigation system, INS output is used to calculate current navigation states; aid is used to supply external measurements and different algorithms are used to provide the most probable corrections to the state estimate using all data. One of the integrated navigation approaches is Terrain Aided Navigation (TAN).

Terrain Aided Navigation is a technique to estimate the position of a moving vehicle by comparing the measured terrain profile under the vehicle to a stored map, DTED. This thesis describes the theoretical aspects implementation of a simulation environment, simulations of the implemented Kalman Filtering TAN algorithms with developed INS model.

In order to perform the study, first a thorough survey of the literature on TAN navigation algorithms is performed. Then, we have developed a dynamics simulation environment. A flight profile generator is designed. Since, the main issue of this work is to correct INS, an Strapdown INS model developed using Matlab INS Toolbox. Therefore, to model a Strapdown INS, mathematical equations of INS system are derived and they are linearized to form linear error model. In addition, a radar altimeter simulator is also developed that provides measurement to the error dynamics.

Then, a Kalman filter structure is designed and implemented using Matlab. The simulations are done with different linearization approaches using Kalman filter structure. Finally, the performance of the implemented algorithms are evaluated.

Keywords: terrain, TAN, INS, navigation, Kalman filter

ÖZ

ARAZİ DESTEKLİ SEYRÜSEFER

Karabörk, Alper
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Mübeccel Demirekler

Eylül 2010, 96 sayfa

Bir Ataletsel Seyrüsefer Sistemi (ASS) bir yardım veya destek olmadan uçağın navigasyon verisini, yani konumunu, hızını tek başına üretebilir. Ancak, sensorlerden kaynaklanan birçok hata navigasyon verisinde kabul edilemez sürüklenmelere yol açar. Bu nedenle Ataletsel Seyrüsefer Sistemini (ASS) düzeltmek için harici destekler kullanılır. Bu destekler kullanılarak, bütünleştirilmiş bir navigasyon yapısı kurulur.

Bütünleştirilmiş bir navigasyon sisteminde, Ataletsel Seyrüsefer Sistemi (ASS) çıktısı sistemin navigasyon durumunu hesaplamak için kullanılır; destek ise harici ölçüm vererek farklı algoritmalar navigasyon durumuna en uygun düzeltmeleri sağlamak için kullanılır. Bütünleştirilmiş navigasyon sistemlerine yaklaşımlardan biri de Arazi Destekli Seyrüseferdir (ADS).

Arazi Destekli Seyrüsefer (ADS) hareket eden bir aracın pozisyonunu aracın altındaki ölçülen arazi profili ile yüklü harita verisini karşılaştırarak tahmin etmeye çalışan bir sistemdir. Bu tez bir simulasyon ortamının geliştirilmesini ve gerçeklenen Kalman filtre ADS algortimalarının geliştirilen ASS ile simulasyonlarını ele almaktadır.

Bu çalışmayı yapmak için, öncelikle ADS navigasyon algoritmaları ve uygulamaları

üzerine kapsamlı bir literatür çalışması yapılmıştır. Sonrasında ilk olarak dinamik bir simülasyon ortamı oluşturulmuştur. Bir uçuş profili üretici geliştirilmiştir. Bu çalışmanın ana konusu INS çıktısını düzeltmek olduğundan, bir Strapdown ATS modeli Matlab INS Toolbox kullanılarak geliştirilmiştir. Bunun için ATS'nin matematiksel denklemleri türetilmiş ve bu denklemler sistemin doğrusal modelini oluşturmak için doğrusallaştırıldı. Ayrıca, sistem ölçüm modeli yani radar altimetre simülatörü de geliştirilmiştir.

Sonrasında, Kalman filtresi yapısı tasarlanmış ve Matlab kullanılarak gerçeklenmiştir. Simülasyonlar farklı doğrusallaştırma yaklaşımları ile Kalman filtresi kullanılarak yapılmıştır. Son olarak, uygulanan algoritmalarının performansı değerlendirilmiştir.

Anahtar Kelimeler: arazi, ADS, ASS, navigasyon, Kalman filtre

To My Family and My Endless Love Nage

ACKNOWLEDGMENTS

I would like to express my sincere appreciation to my supervisor Prof. Dr. Mübeccel Demirekler for her continued guidance, also patience and support throughout my research. She provided advice, and assistance that greatly enhanced my studies and experience. And I offer sincere thanks for my wife and my family. My wife, Nagehan, always supported and believed in me. She did not give up her continuous love and understanding.

TABLE OF CONTENTS

ABSTRACT iv

ÖZ vi

DEDICATON viii

ACKNOWLEDGMENTS ix

TABLE OF CONTENTS x

LIST OF TABLES xiv

LIST OF FIGURES xv

CHAPTERS

1 INTRODUCTION 1

1.1 Scope of the Study 1

1.2 TAN 2

1.2.1 TAN Techniques 3

1.3 Radar Altimeter 4

1.4 Barometric Altimeter 4

1.5 DTED 4

SRTM 5

ASTER 6

1.6 Scope of the Thesis 7

1.7 Outline of the Thesis 7

2 INERTIAL NAVIGATION SYSTEMS AND KALMAN FILTER . . 9

2.1 Introduction to INS 9

2.2 Reference Coordinate Frames 10

2.2.1 Inertial Frame 10

2.2.2 Earth Frame 10
2.2.3 Body Frame 11
2.2.4 Inertial Sensor Frame 12
2.3 Inertial Navigation System Configurations 12
2.3.1 Stable Platform Systems 13
2.3.2 Strapdown Inertial Navigation Systems 13
2.4 INS Equations 14
2.4.1 Basic Definitions and Notations 14
2.4.2 The Velocity Equation 17
2.4.3 The Attitude Equation 19
2.4.4 The Position Equation 20
2.4.5 The Gravity Model 20
2.4.6 Summary of the Non-linear Equations of INS . . . 21
2.5 Linear Error Model of INS 21
2.5.1 Error Propagation of Attitude 22
2.5.2 Error Propagation of Velocity 24
2.5.3 Error Propagation of Position 25
2.5.4 Error Propagation of Gravity 25
2.5.5 INS Sensor Error Model 26
2.5.6 Summary of Linear INS Equations 28
2.6 Kalman Filter 30
2.6.1 Introduction to Kalman Filter 30
2.6.1.1 State Model 31
2.6.1.2 Observation Model 31
2.6.2 Kalman Filter Algorithm 32
2.6.2.1 Prediction 32
2.6.2.2 Correction 33
2.6.3 Discrete Time Kalman Filter 34
2.6.3.1 Discretization of Linear State Space Model 34

Approximations 35
2.6.3.2 Discrete Time KF Equations 35
2.6.3.3 Extended Kalman Filter 37
2.6.4 TAN Kalman Filter 38
Measurement model 38
2.6.5 Measurement Model Related Errors 42
3 MAJOR TERRAIN AIDED NAVIGATION METHODS 43
3.1 Introduction 43
3.2 TERCOM 43
3.2.1 Definition of TERCOM 45
3.2.2 Fundamental Phases of the TERCOM 45
Data preparation: 45
Data acquisition: 46
Data correlation: 46
3.2.3 The Measurement process 47
3.2.4 Data Correlation Techniques 48
3.2.5 TERCOM Performance 49
3.3 SITAN 50
3.4 HELI/SITAN 53
3.5 Other TAN Algorithms 55
VATAN 55
Bayesian Approach 55
3.6 The TAN approach used for Thesis 56
4 SIMULATION RESULTS AND SIMULATION TOOLS DEVELOPED 57
4.1 Introduction 57
4.2 INS Toolbox 58
4.3 INS Data Simulator 59
4.4 Flight Profile Generator 63
4.5 Terrain Module 64

Barometric Altimeter 65

Radar Altimeter 65

Terrain Height Generator 67

Measurement Generator 68

4.6 Kalman Filter Module 68

4.7 Simulation specifications 70

4.7.1 Simulations for Terrain Aided Navigation 72

4.7.1.1 Linearized Kalman Filter 73

4.8 Extended Kalman Filter 75

4.8.1 Adaptive **R** and **Q** Approach 81

4.8.2 Different Terrain Stochastic Linearization Approaches 82

Mean Tangent Line (MTL) 83

Nine Point Planar Fit (NPPF) 85

4.8.3 Simulations with Different Flight Path 88

5 DISCUSSIONS CONCLUSIONS AND FUTURE WORK 93

REFERENCES 95

LIST OF TABLES

TABLES

Table 2.1 Discrete Time Kalman Filter Equations 37

Table 3.1 Chronological Overview of TERCOM (taken from [4]) 44

Table 3.2 Extented Kalman Filter Equations for SITAN 54

Table 4.1 LKF (with measurement update interval) Simulation Settings 73

Table 4.2 EKF Nonlinear Solution Simulation Settings 75

Table 4.3 Horizontal Errors 77

Table 4.4 Process Noise 77

Table 4.5 Horizontal Errors 78

Table 4.6 EKF Nonlinear Solution Simulation Settings-2 79

Table 4.7 Horizontal Position Errors 80

Table 4.8 Adaptive EKF TAN Filter Settings 81

Table 4.9 Horizontal Position Errors for Adaptive EKF 82

Table 4.10 Horizontal Position Errors for MTL Technique 85

Table 4.11 Horizontal Position Errors for NPPF Technique 86

Table 4.12 Horizontal Position Errors 88

LIST OF FIGURES

FIGURES

Figure 1.1 Terrain Aided Navigation Principle 2

Figure 1.2 Terrain Aided Navigation Concepts (taken from [17]) 3

Figure 1.3 Radar Altimeter Block Diagram 5

Figure 2.1 Earth Centered Earth Fixed Reference Frame (taken from [26]) . . 11

Figure 2.2 Body Frame of Reference 12

Figure 2.3 A Stable Platform IMU 14

Figure 2.4 A Stable Platform Inertial Navigation Algorithm (taken from [30]) . 15

Figure 2.5 Strapdown Inertial Navigation Algorithm (taken from [30]) 15

Figure 2.6 A Strapdown Navigation System 22

Figure 2.7 Terrain Stochastic Linearization (TSL) (taken from [27]) 41

Figure 3.1 TERCOM Algorithm-1 47

Figure 3.2 TERCOM Algorithm-2 (taken from [25]) 49

Figure 3.3 Terrain Aided Navigation (SITAN) (taken from [12]) 50

Figure 4.1 DCM Generator 60

Figure 4.2 Reference Frame Transformation of the Accelerometer Outputs . . 61

Figure 4.3 Horizontal Position Error 63

Figure 4.4 Ankara to İstanbul Errors 64

Figure 4.5 2-D View of the map from Ankara to İstanbul 65

Figure 4.6 3-D View of the map that we plan to fly over 66

Figure 4.7 Barometric Altimeter Simulator 66

Figure 4.8 Radar Altimeter Simulator 67

Figure 4.9 Radar Altimeter Simulator Output from Ankara to İstanbul 67
Figure 4.10 Terrain Height Generator 68
Figure 4.11 Measurement Generator 68
Figure 4.12 Defining Constant and Initializing the System 70
Figure 4.13 Kalman Filter 71
Figure 4.14 Horizontal Position Errors of INS and LKF Solution (updated with time update intervals) 74
Figure 4.15 Horizontal Position Errors of INS and Filtered solution-1 76
Figure 4.16 Horizontal Position Errors of INS and Filtered solution-3 78
Figure 4.17 Horizontal Position Errors for **R**=100 79
Figure 4.18 Horizontal Position Errors for **R**=2 80
Figure 4.19 Horizontal Position Errors for Adaptive EKF 81
Figure 4.20 Ground Patch Geometry for FTS Technique (taken from [20]) . . . 83
Figure 4.21 Ground Patch Geometry for MTL Technique (taken from [20]) . . . 83
Figure 4.22 Horizontal Position Errors for MTL Technique 84
Figure 4.23 Ground Patch Geometry for NPPF Technique (taken from [20]) . . 85
Figure 4.24 Horizontal Position Errors for NPPF Technique 86
Figure 4.25 Horizontal Position Errors for High Resolution Map 87
Figure 4.26 Terrain profile 2-D 89
Figure 4.27 Terrain profile 3-D 90
Figure 4.28 Horizontal Position Errors 91
Figure 4.29 Zoom Area-1 91
Figure 4.30 Zoom Area-2 92
Figure 4.31 Zoom Area-3 92

LIST OF ABBREVIATIONS

ADS	Arazi Destekli Seyrüsefer
ASS	Ataletsel Seyrüsefer Sistemi
ASTER	Advanced Spaceborne Thermal Emission and Reflection Radiometer
AVRADA	AVionics Reseach And Development Activity
DCM	Direction Cosine Matrix
DMAAC	Defense Mapping Agency Aerospace Center
DTED	Digital Terrain Elevation Data
ECEF	Earth Centered Earth Fixed Frame
ECIF	Earth Centered Inertial Frame
***e*-frame**	Earth frame
EKF	Extended Kalman Filter
FTS	First Order Taylor Series
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
***i*-frame**	Inertial frame

LKF Linearized Kalman Filter

LSSM Long Sample Short Matrix

MAD Mean Absolute Distance

MAP Maximum A Posteriori

MMAE Multiple Model Adaptive Estimation

MSD Mean Squared Distance

MSL Mean Sea level

MTL Mean Tangent Line

NPPF Nine Point Planar Fit

RA Radar Altimeter

RAS Reference Altitude Sensor

SatNav Satellite Navigation

SITAN Sandia Inertial Terrain Aided Navigation

SSLM Short Sample Long Matrix

SRTM Shuttle Radar Topography Mission

TAN Terrain Aided Navigation

TERCOM Terrain Counter Matching

TRF Terrain Referenced Navigation

TSL Terrain Stochastic Linearization

CHAPTER 1

INTRODUCTION

1.1 Scope of the Study

For Navigation systems achieving high accuracy is a very important property to be considered. But for inertial navigation systems navigation errors grow over the time because of the initial alignment accuracy and the imperfections caused by the inertial sensors. Although sensor accuracy can be improved by the use of more accurate sensors, there are limits to the performance before the cost of the system becomes prohibitively large [26].

Aided Navigation approach is an alternative to improve the accuracy of the inertial navigation systems. It can be defined as to employ some external aid to the navigation system which gives an additional navigation information to the inertial navigation system. Aided navigation system can be classified according to the type of the aid employed as external aids or the onboard aids. An external aid means that the measurement used to correct the navigation system error is obtained by receiving signals or by viewing objects outside the platform. The on board aid means that measurements are obtained by using additional sensors carried onboard . An external aids may be radio navigation aids, satellites, star trackers or a ground based radar tracker. The onboard navigation aids may be radar altimeter, barometric altimeter, Doppler radar, airspeed indicator, magnetic sensors, electro-optical imaging systems.

The structure of an aided navigation system is given in Figure 1.1.

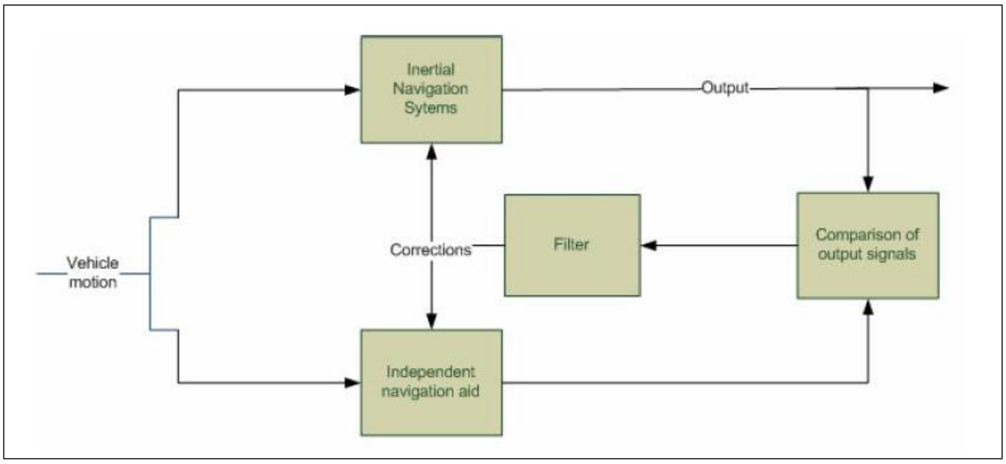

Figure 1.1: Terrain Aided Navigation Principle

1.2 TAN

Terrain Aided Navigation (TAN) or Terrain Referenced Navigation (TRF) is an external measurement type of aided navigation systems. The external measurement sensors are barometric altimeter and radar altimeter. TAN systems uses these measurements in combination with a map elevation database, such as digital terrain elevation data (DTED).

The radar altimeter measurements of heights above ground and the barometric altimeter measurement above mean sea level, provided by INS, are used to construct a ground profile of the platform throughout the flight. Then, the constructed terrain profile is compared with a stored map data to achieve a profile fit, thus the aim is to decide on the position of the vehicle and to correct the INS [26].

In other words TAN is a technique to estimate the position of a moving platform by utilizing the terrain height profile under the platform and a stored terrain elevation map. The performances of the TAN algorithms are generally functions of the terrain roughness, greater the terrain variations better the performance is obtained. The properties of different methods and performances of them are explained in Chapter 3 in detail.

1.2.1 TAN Techniques

There are several Terrain Aided Navigation techniques in the literature. These techniques can be classified into two main group as batch and recursive that are shown in Figure 1.2 [17].

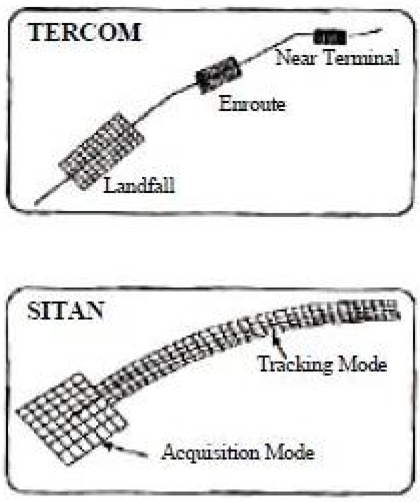

Figure 1.2: Terrain Aided Navigation Concepts (taken from [17])

The most widely known TAN algorithm is TERCOM which is a batch oriented TAN method. The principle of TERCOM is collecting terrain measurement information during the flight path and use these measurements in post processing in a batch fashion to provide a correlation with the terrain elevation map [17]. TERCOM is utilized for missile application, Tomahawk. The map preparation and validation is very crucial point of TERCOM algorithm. The details of TERCOM algorithm is given in Chapter 3.

The most well known recursive TAN algorithm is SITAN which is proposed by Sandia National Laboratories in the late 1970's. SITAN uses an Extended Kalman Filter (EKF) and some linearization techniques. In contrast to batch algortihms recursive algorithms like SITAN operate on the individual measurements as they are obtained.

"SITAN has been developed for the manned aircraft application using the DTED. Data storage and memory access capabilities of embedded vehicle computers have improved dramatically when compared with the 1960' when TAN concepts originally developed. Thus, the computational constraints do not apply as techniques are developed to enhance the performance and to expand the operational envelope of TAN techniques." [17].

1.3 Radar Altimeter

A radar altimeter measurement instrument supplies the height of the platform from the terrain. The radar altimeter working principle is based on the measurement of the time required for an electromagnetic pulse to travel from the platform to the terrain surface below and to return to the platform. The receiver and the transmitter antennas of a radar altimeter are mounted under the platform, i.e. aircraft , helicopter. A radar altimeter block diagram that summarizes the basic components and the working principle of the radar altimeter is given in Figure 1.3.

1.4 Barometric Altimeter

Barometric altimeters are designed to output height above the mean sea level (MSL) relative to the pressure difference [19]. It is also known as pressure altimeter. A barometric altimeter is the altimeter found in most aircrafts. The performance of the barometric altimeters are not good when the high maneuverable scenario is the case. However the barometric altimeters are low cost devices.

1.5 DTED

One of the most important part of TAN algorithm is the stored terrain elevation data. Digital Terrain Elevation Data (DTED) is generally used for military purposes. DTED is a standard of digital datasets which consists of a matrix of terrain elevation values. This standard was originally developed in the 1970s to support aircraft radar simulation and prediction. DTED is one kind of the elevation produced by the U. S.

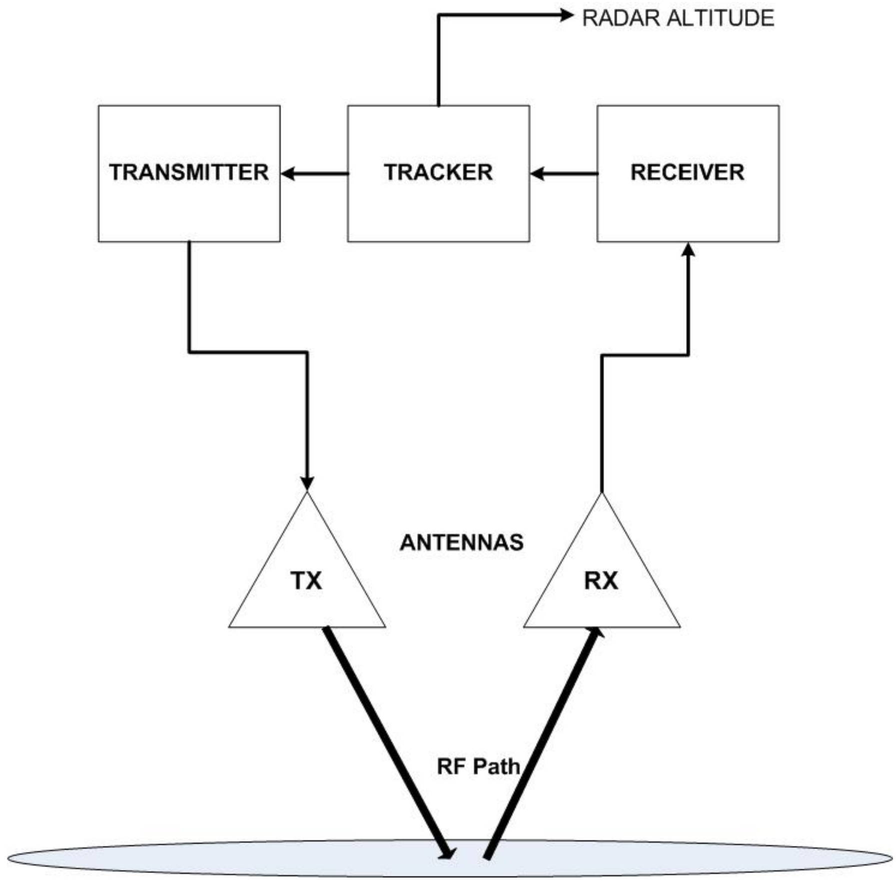

Figure 1.3: Radar Altimeter Block Diagram

Department of Defense.

“DTED Level 0 files have 121-by-121 points for an one terrain section (size of an one terrain section is 1 degree longitude, 1 degree latitude). DTED Level 1 files have 1201-by-1201 points for the same region. The edges of adjacent tiles have redundant records. DTED files are binary. No line ending conversion or byte-swapping is required when downloading a DTED file. The data available to the public is called Level 0 and has a 30 arc second spacing. Other higher resolution data called Level 1 and Level 2 are not available to public.” [10].

Some other elevation data types found in the literature are mentioned below.

SRTM “Shuttle Radar Topography Mission (SRTM) obtained elevation data on a near-global scale to generate the most complete high-resolution digital topographic database of Earth. SRTM consisted of a specially modified radar system that flew onboard the Space Shuttle Endeavour during an 11-day mission in February of 2000. SRTM is an international project spearheaded by the National Geospatial-Intelligence Agency (NGA), NASA, the Italian Space Agency (ASI) and the German Aerospace Center (DLR). There are three resolution outputs available, including 1 kilometer and 90 meter resolutions for the world and a 30 meter resolution for the US. The elevation datasets are affected by mountain and desert no-data areas. These no-data areas can be a problem in areas of very high relief. They affect all summits over 8000 meters, most summits over 7000 meters, many Alpine and similar summits and ridges, and many gorges and canyons. There are some SRTM data sources which have filled these data voids, but some of these have used only interpolation from surrounding data, and may therefore be very inaccurate. If the voids are large, or completely cover summit or ridge areas, no interpolation algorithms will give satisfactory results. Groups of scientists have worked on algorithms to fill the voids of the original SRTM data and similar works are carrying on." [29].

ASTER "Advanced Spaceborne Thermal Emission and Reflection Radiometer is a Japanese sensor which is one of five remote sensory devices on board the Terra satellite launched into Earth orbit by NASA in 1999. The instrument has been collecting surficial data since February 2000. ASTER provides high-resolution images of the Earth in 15 different bands of the electromagnetic spectrum, ranging from visible to thermal infrared light. The resolution of images ranges between 15 to 90 meters. ASTER data are used to create detailed maps of surface temperature of land, emissivity, reflectance, and elevation. Despite the high nominal resolution, however, some reviewers have commented that the true resolution is considerably lower, and not as good as that of SRTM data, and serious artifacts are present. Some of these limitations have been confirmed by METI and NASA, who point out that the current version of the GDEM product is research grade." [29].

1.6 Scope of the Thesis

In this thesis the main aim is to investigate inertial navigation system principles and TAN algorithms. In addition, a simulation environment comprising from Strapdown INS, Flight Profile Generator, Terrain Module etc. to simulate TAN solutions is developed. Implementation of Kalman filter TAN solution is also another crucial part of this thesis.

1.7 Outline of the Thesis

The thesis is composed of five chapters:

1. Introduction
2. Inertial Navigation Systems and Kalman Filter
3. Major Terrain Aided Navigation Methods
4. Simulations, Results and Simulation Tools Developed
5. Discussion, Conclusion and Future Work

In Chapter 1, an introduction to the Terrain Aided Navigation (TAN) is given. The main techniques of TAN are discussed. Inertial navigation integration principle of TAN is another main issue of this chapter.

In Chapter 2, inertial navigation principles are explained starting from the coordinate frames and transformations. Basic notations and concepts needed for navigation applications are investigated and presented. Inertial navigation equations are derived and system error model is developed. The nonlinear equations of a Strapdown INS are derived, then a linear error model is constructed. The Kalman filter algorithm is also explained in the second chapter. The measurement model used throughout thesis is discussed.

In Chapter 3, the major TAN methods found in the literature are investigated. TERCOM and SITAN methods are discussed in detail. VATAN and Bayesian Approach are also explained in third chapter fundamental properties of these major methods are discussed in this chapter.

In Chapter 4, the simulation tools developed throughout the thesis are explained in detail. Then, the Kalman filter solutions implemented are discussed and the simulation results of these algorithms are presented.

Chapter 5 is devoted to the conclusions and the discussions with possible future work suggestions.

CHAPTER 2

INERTIAL NAVIGATION SYSTEMS AND KALMAN FILTER

2.1 Introduction to INS

Navigation is a very ancient skill or art which has become a complex science today. Navigation in basic words is traveling and finding the way from one point to another. The main data produced by navigation systems is the position, velocity, attitude and time information [26].

There are many different kinds of navigation techniques used throughout the world as dead reckoning, piloting, celestial navigation, electronic navigation (radio, radar, satellite) etc. the common point of all is sensors to gather navigational information. Inertial navigation systems are the ones using inertial sensors for navigational purposes.

The operation principle of the inertial navigation system depends on the law of classical mechanics that is formulated by Newton. Newtons law tells us that the motion of a body will continue uniformly unless an external force is acted on the body. These laws also means that the applied force will reason an acceleration in the body motion. When this acceleration is measured, by utilizing this measurement the velocity and the position change can be calculated. Acceleration can be measured by utilizing the accelerometer instrument. An inertial navigation system usually contains three such devices, each of which is capable of detecting acceleration in a single direction. The accelerometers are commonly mounted with their sensitive axes perpendicular to one another.

"Rotational motion of the body with respect to the inertial reference frame may be sensed using gyroscopic sensors and used to determine the orientation of the accelerometers at all times. Given this information, it is possible to resolve the accelerations into the reference frame before the integration process takes place. After gathering two sets of information throughout the accelerometers and gyroscopes, it is possible to define translational motion of the vehicle within the inertial frame and so to calculate its position within that frame." [26].

2.2 Reference Coordinate Frames

Before going any further, it is important to point out that navigation systems work with respect to a reference system. An inertial navigation system uses its sensor readings in a system referenced to inertial frame, but navigation outputs are needed in a system referenced to Earth. So, there is a need to convert the sensor readings to the chosen reference system. Each frame has an orthogonal, right handed axis set and defined by specifying the location of the origin and the direction of the three axes [31]. The fundamental reference frames used throughout the study are defined in the subsequent parts of this section.

2.2.1 Inertial Frame

The origin of the inertial frame (*i*-frame) is at the center of the Earth. The inertial frame axes are fixed with respect to the fixed stars. The axes x_i, y_i, z_i are defined such that z_i is coincident with the Earths polar axis (which is assumed to be invariant in direction). Because of the rotation of the Earth x_i and y_i does not remain constant with respect to the Earth. The inertial frame is also known as Earth Centered Inertial Frame(ECIF). In the equations inertial frame is designated by the subscript $\boldsymbol{i}$.

2.2.2 Earth Frame

The Earth frame (*e*-frame) has its origin at the center of the Earth but different from the *i*-frame the axes are non-rotating with respect to the Earth. The axes are defined as

x_e, y_e, z_e with z_e coincident with the Earth's polar axis and x_e lies along the intersection of the plane of the Greenwich meridian with the Earth's equatorial plane. The e-frame rotates with respect to the i-frame at a rate of w_{ie} around the axes z_e (in other words w_{ie} is the turn rate of earth).

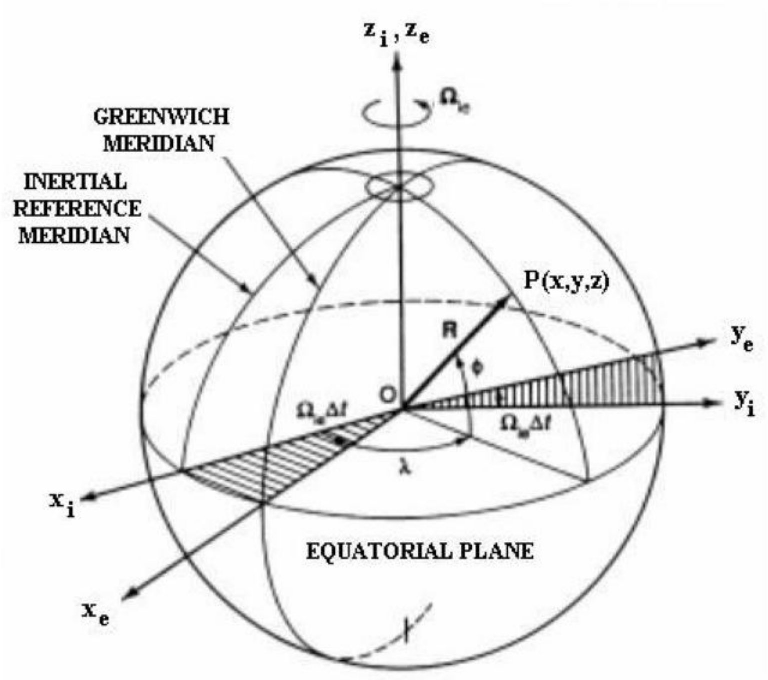

Figure 2.1: Earth Centered Earth Fixed Reference Frame (taken from [26])

The Earth Frame is also known as Earth Centered Earth Fixed Frame((ECEF) and is shown in Figure 2.1. In the equations earth frame is designated by the subscript **e**.

2.2.3 Body Frame

The body reference frame has an orthogonal axis set. The axes of the body frame are the roll, pitch and yaw axes of the vehicle. The origin of the body frame is at the mass center of the platform that the navigation system is mounted on. The axes directions are defined as follows. The roll axis is forward along the longitudinal axis of the platform; the yaw axis is directed downward and the pitch axis is directed 90° to the right which is normal to the roll axis and yaw axis plane. The axes are defined as x_b, y_b, z_b that are coincident with the mass center of the platform. In the equations body frame is designated by the subscript **b**. The orientation of the axes are shown in Figure 2.2.

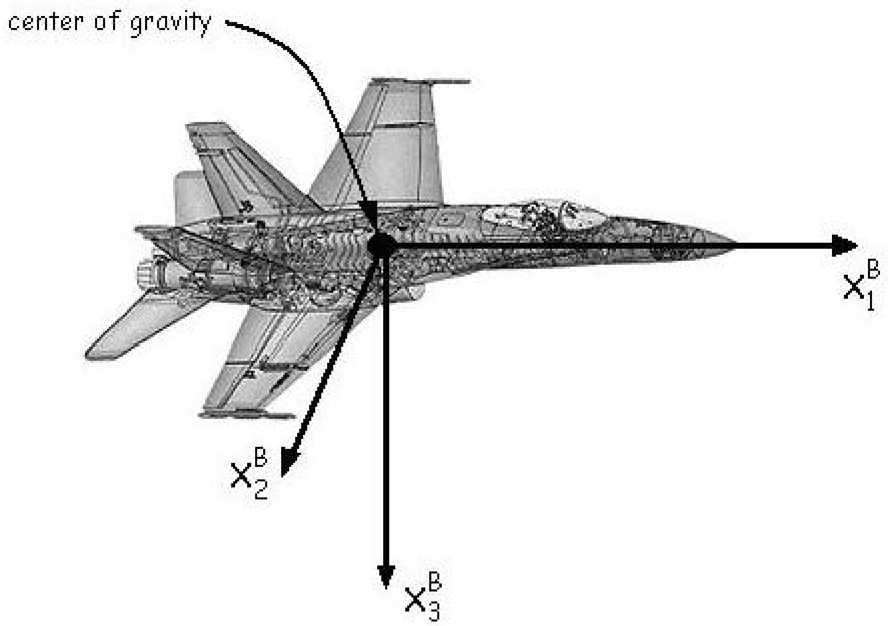

Figure 2.2: Body Frame of Reference

2.2.4 Inertial Sensor Frame

Inertial sensor frame axes are aligned with the axis of each sensor and are placed perpendicular to each other. The three sensitive axes of the sensors describe the Inertial Sensor Frame. The axis of the navigation system made up of this inertial measurement unit (IMU) is described with this frame.

2.3 Inertial Navigation System Configurations

Inertial navigation systems generally can be classified into the two main categories. The fundamental difference between these two main categories is the reference frame in which the gyroscopes and accelerometers operate. In this thesis the navigation systems frame will be referred as body frame and to the reference frame that we are navigating in as the inertial frame.

2.3.1 Stable Platform Systems

“In stable platform type systems the inertial sensors are mounted on a platform which is isolated from any external rotational motion. In other words the platform is held in alignment with the global frame. This is achieved by mounting the platform using gimbals (frames) which allow the platform freedom in all three axes, as shown in Figure 2.3. The platform mounted gyroscopes detect any platform rotations. These signals are fed back to torque motors which rotate the gimbals in order to cancel out such rotations, hence keeping the platform aligned with the global frame. To track the orientation of the device the angles between adjacent gimbals can be read using angle pick-offs. To calculate the position of the platform the signals from the platform mounted accelerometers are double integrated. Note that it is necessary to subtract acceleration due to gravity from the vertical channel before performing the integration. The stable platform inertial navigation algorithm is shown in Figure 2.4.” [26].

2.3.2 Strapdown Inertial Navigation Systems

“In strapdown systems the inertial sensors are mounted rigidly onto the platform and therefore output quantities are measured in the body frame rather than the global frame. To keep track of orientation the signals from the rate gyroscopes are integrated. To track position, the three accelerometer signals are resolved into global coordinates using the known orientation, as determined by the integration of the gyro signals. The global acceleration signals are then integrated as in the stable platform algorithm. This procedure is shown in Figure 2.5. Stable platform and strapdown systems are both based on the same underlying principles. Strapdown systems have reduced mechanical complexity and tend to be physically smaller than stable platform systems. These benefits are achieved at the cost of increased computational complex-

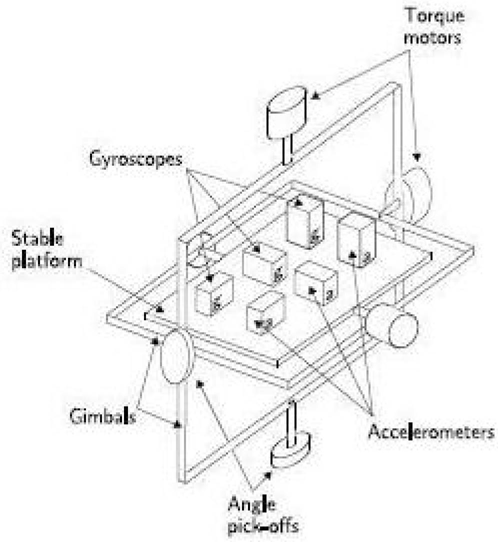

Figure 2.3: A Stable Platform IMU

ity. As the cost of computation has decreased strapdown systems have become the dominant type of INS." [26].

2.4 INS Equations

In this section the fundamental equations of Strapdown Navigation Systems are derived starting from the basic navigation equation. These equations describe the algorithm of an INS. The main aim of this section is to get and show the INS outputs in terms of sensors', accelerometers and gyroscopes, outputs. The derivations are developed referenced to Earth Centered Earth Fixed Frame (ECEF).

2.4.1 Basic Definitions and Notations

In inertial navigation one of the most important concepts is the transformation between the reference frames. Transformation of a vector defined in any reference frame to another frame can be done by using a transformation matrix. This transformation matrix is called as 'Direction Cosine Matrix (DCM)'.

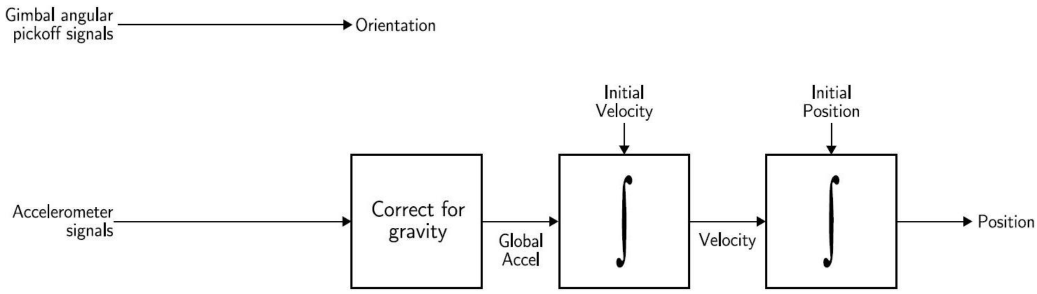

Figure 2.4: A Stable Platform Inertial Navigation Algorithm (taken from [30])

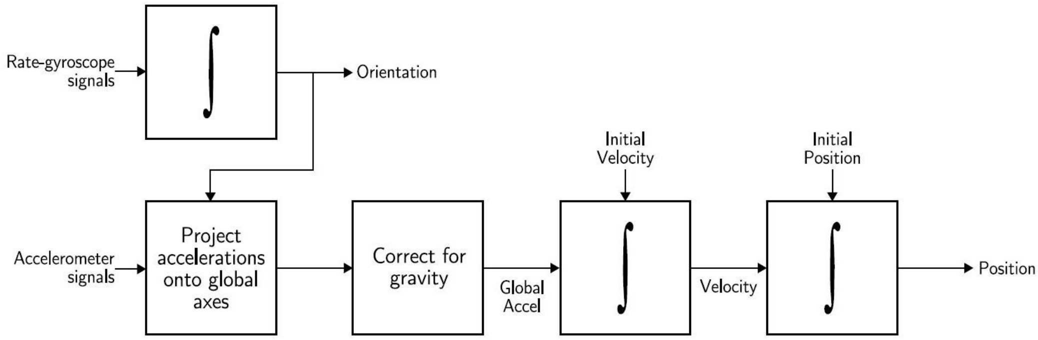

Figure 2.5: Strapdown Inertial Navigation Algorithm (taken from [30])

If $\mathbf{v}_a$ is a vector in a-frame the transformed form of this vector in b-frame is designated by a subscript b as $\mathbf{v}_b$ so the transformation matrix $\mathbf{C}_a^b$ means transformation matrix from a-frame to b-frame.

$$\mathbf{v}_b = C_a^b \mathbf{v}_a$$

A DCM is an orthonormal matrix so it has some properties such as:

$$|\mathbf{C}_a^b| = 1$$

$$[\mathbf{C}_b^a]^T = [\mathbf{C}_b^a]^{-1}$$

Furthermore it ensures the following properties

$$\mathbf{C}_a^b = [\mathbf{C}_b^a]^{-1}$$

$$\dot{\mathbf{C}}_a^b = \mathbf{C}_a^b \mathbf{W}_{ba}^a$$

$$\dot{\mathbf{C}}_b^a = -\mathbf{W}_{ba}^a \mathbf{C}_b^a$$

$\mathbf{W}_{ba}$ in the last two expressions is the skew-symmetric matrix form of the vector $\mathbf{w}_{ba}^a = [w_x w_y w_z]^T$.

$$\mathbf{W}_{ba} = \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix}$$

where $\mathbf{w}_{ba}^a$ is the angular velocity of the frame a with respect to the frame b.

In this work the derivative of a vector with respect to any reference frame is represented as

$$\left.\frac{d\mathbf{R}}{dt}\right|_a$$

In the equation above the subscript a means that this derivation is done with respect to a-frame.

Another very important concept related to the transformation between reference frames is the Coriolis effect.

“The Coriolis effect is an apparent deflection of moving objects when they are viewed from a rotating reference frame. Newton’s laws of motion govern the motion of an object in the inertial frame of reference. When transforming Newton’s laws to a rotating frame of reference, the Coriolis force appears, along with the centrifugal force. If the rotation speed of the frame is not constant, the Euler force will also appear. All three forces are proportional to the mass of the object. The Coriolis force is proportional to the speed of rotation and the centrifugal force is proportional to its square. The Coriolis force acts in a direction perpendicular to the rotation axis and to the velocity of the body in the rotating frame and is proportional to the object’s speed in the rotating frame. The centrifugal force acts outwards in the radial direction and is proportional to the distance of the body from the axis of the rotating frame.” [29].

This effect is formulated as Coriolis equation. If i-frame is rotating with respect to the e-frame with angular velocity $\mathbf{w}_{ie}$, from the Coriolis equation, the time derivative of a vector $\mathbf{R}$ in i-frame is formulated in terms of time derivative of that vector in e-frame as:

$$\left.\frac{d\mathbf{R}}{dt}\right|_i = \left.\frac{d\mathbf{R}}{dt}\right|_e + \mathbf{w}_{ie} \times \mathbf{R} \tag{2.1}$$

where

$$\left.\frac{d\mathbf{R}}{dt}\right|_i : \text{time derivative of } \mathbf{R} \text{ in the } i\text{-frame}$$

$$\left.\frac{d\mathbf{R}}{dt}\right|_e : \text{time derivative of } \mathbf{R} \text{ as seen by an observer in } e\text{-frame}$$

$$\mathbf{w}_{ie} \times \mathbf{R} : \text{change in } \mathbf{R} \text{ caused by rotation of the } e\text{-frame}$$

$$\mathbf{w}_{ie} : \text{angular turn rate of the } e\text{-frame with respect to the } i\text{-frame}$$

The main principle of physics says that the total acceleration of a point in space is the sum of the total force acting on that point and the gravitational acceleration. This is formulated below and is called the "Navigation Equation".

$$\left.\frac{d^2\mathbf{R}}{dt^2}\right|_i = \mathbf{f}_e + \mathbf{g}_e \tag{2.2}$$

The choice of the reference frame is a crucial point for this equation. All vectors should be represented in same frame. If vectors are represented in same frame, navigation equation can be solved in any frame. Throughout this work all derivation is developed for Earth Centered Earth Fixed Frame (ECEF).

2.4.2 The Velocity Equation

The objective is to get the velocity vector $\mathbf{V}_e$ in terms of the sensor outputs and some other know parameters. Starting from the well know equation; the derivative of the position gives the velocity as:

$$\mathbf{V}_e = \left.\frac{d\mathbf{R}_e}{d\mathbf{t}}\right|_e \tag{2.3}$$

where

$\mathbf{V}_e$: Earth relative velocity
$\mathbf{R}_e$: Position vector of the vehicle with respect to the Earth's center

To write the rate of change of $\mathbf{V}_e$ in the inertial frame we need to use the Coriolis equation as:

$$\left.\frac{d\mathbf{R}_e}{dt}\right|_i = \left.\frac{d\mathbf{R}_e}{dt}\right|_e + \mathbf{w}_{ie}^e \times \mathbf{R}_e \tag{2.4}$$

where

$\mathbf{w}_{ie}^{e}$=[0 0 w_{ie}^{e}]^T : turn rate of the Earth with respect to the inertial frame

To get the derivative of $\mathbf{V}_e$ we start from Equation 2.1, go on by taking the derivative of the velocity to develop its final form.

$$\left.\frac{d^2\mathbf{R}_e}{dt^2}\right|_i = \left.\frac{d\mathbf{V}_e}{dt}\right|_i + \dot{\mathbf{w}}_{ie}^{e} \times \mathbf{R}_e + \mathbf{w}_{ie}^{e} \times \left.\frac{d\mathbf{R}_e}{dt}\right|_i \tag{2.5}$$

Substituting the following term from the Coriolis as:

$$\left.\frac{d\mathbf{R}_e}{dt}\right|_i = \left.\frac{d\mathbf{R}_e}{dt}\right|_e + \mathbf{w}_{ie}^{e} \times \mathbf{R}_e$$

Equation 2.5 becomes as:

$$\left.\frac{d^2\mathbf{R}_e}{dt^2}\right|_i = \left.\frac{d\mathbf{V}_e}{dt}\right|_i + \dot{\mathbf{w}}_{ie}^{e} \times \mathbf{R}_e + \mathbf{w}_{ie}^{e} \times \left[\left.\frac{d\mathbf{R}_e}{dt}\right|_e + \mathbf{w}_{ie}^{e} \times \mathbf{R}_e\right] \tag{2.6}$$

Assuming that the Earth is rotating at a constant speed and substituting

$$\mathbf{V}_e = \frac{d\mathbf{R}_e}{dt}$$

We get the equation:

$$\left.\frac{d^2\mathbf{R}_e}{dt^2}\right|_i = \left.\frac{d\mathbf{V}_e}{dt}\right|_i + \mathbf{w}_{ie}^{e} \times \mathbf{V}_e + \mathbf{w}_{ie}^{e} \times [\mathbf{w}_{ie}^{e} \times \mathbf{R}_e] \tag{2.7}$$

Using the basic navigation equation in earth frame form:

$$\left.\frac{d^2\mathbf{R}_e}{dt^2}\right|_i = \mathbf{f}_e + \mathbf{g}_e \tag{2.8}$$

And Equation 2.7 becomes as:

$$\left.\frac{d\mathbf{V}_e}{dt}\right|_i = \mathbf{f}_e - \mathbf{w}_{ie}^{e} \times \mathbf{V}_e + \mathbf{g}_e - \mathbf{w}_{ie}^{e} \times [\mathbf{w}_{ie}^{e} \times \mathbf{R}_e] \tag{2.9}$$

Let's define $\mathbf{g}_l = \mathbf{g}_e - \mathbf{w}_{ie}^{e} \times [\mathbf{w}_{ie}^{e} \times \mathbf{R}_e]$ and use the Coriolis equation one more time, the velocity equation becomes:

$$\left.\frac{d\mathbf{V}_e}{dt}\right|_e = \mathbf{f}_e - 2\mathbf{w}_{ie}^{e} \times \mathbf{V}_e + \mathbf{g}_l \tag{2.10}$$

Finally writing the accelerometer measurements in the body frame by using $\mathbf{C}_b^e$ the velocity equation becomes:

$$\left.\frac{d\mathbf{V}_e}{dt}\right|_e = \mathbf{C}_b^e\mathbf{f}_b - 2\mathbf{w}_{ie}^{e} \times \mathbf{V}_e + \mathbf{g}_l \tag{2.11}$$

2.4.3 The Attitude Equation

Before starting to develop the attitude equation of the strapdown navigation system a brief conceptual introduction will be given for DCM. For this study, $\mathbf{C}_b^e$ is a time dependent matrix that converts the measured specific force vector from the body frame to the earth frame. There is a need to update the DCM because of the movement of the body frame. $\mathbf{C}_b^e$ can be expressed as the multiplication of two DCM's to simplify the representation of the rotations and transformation:

$$\mathbf{C}_b^e = \mathbf{C}_n^e \mathbf{C}_b^n \tag{2.12}$$

$\mathbf{C}_b^n$ includes the roll, pitch and yaw angles and $\mathbf{C}_n^e$ includes terms of geodetic latitude Φ and longitude Λ.

To derive the attitude equation we will start by writing the properties of DCM. The derivative of the $\mathbf{C}_b^e$ satisfies the equation.

$$\dot{\mathbf{C}}_b^e = \mathbf{C}_b^e \Omega_{eb}^b \tag{2.13}$$

Ω_{eb}^b is the skew-symmetric matrix form of the vector $\mathbf{w}_{eb}^b$, body turn rate with respect to the earth frame (expressed at the body frame). This vector also includes the Earth rate. In other words this vector represents the outputs of the gyroscopes and rate of Earth. So $\mathbf{w}_{eb}^b$ can be written in terms of $\mathbf{w}_{ib}^b$=$[p_{roll}\ q_{pitch}\ r_{yaw}]^T$ and Earth angular velocity $\mathbf{w}_{ie}$ as:

$$\mathbf{w}_{eb}^b = \mathbf{w}_{ib}^b - \mathbf{w}_{ie}^b \tag{2.14}$$

Substituting Equation 2.14 into Equation 2.13:

$$\dot{\mathbf{C}}_b^e = \mathbf{C}_b^e[\Omega_{ib}^b - \Omega_{ie}^b] \tag{2.15}$$

Using similarity transformation property of the skew-symmetric matrix we can convert Ω_{ie}^b to Ω_{ie}^e as:

$$\Omega_{ie}^b = \mathbf{C}_e^b \Omega_{ie}^e \mathbf{C}_b^e \tag{2.16}$$

Combining Equations 2.16 and 2.14 we get the final attitude equation as:

$$\dot{\mathbf{C}}_b^e = \mathbf{C}_b^e \Omega_{ib}^b - \Omega_{ie}^e \mathbf{C}_b^e \tag{2.17}$$

2.4.4 The Position Equation

For completeness of the differential equations related to the inertial navigation system variables we write the position equation as the integral of velocity such as:

$$\left.\frac{d\mathbf{R}_e}{dt}\right|_e = \mathbf{V}_e \tag{2.18}$$

Our objective is to relate the inertial navigation system variables position, velocity and attitude to the sensor outputs $\mathbf{f}_b$ and $\mathbf{w}_{ib}^b$. As a result we derived the non-linear differential equations of the position, velocity and attitude.

2.4.5 The Gravity Model

In real world the magnitude and also the direction of the gravity vector change with the position on the Earth and the height above the Earth. This variation stems from the difference between the mass attraction of the Earth and the centrifugal acceleration being a function of latitude. The gravity model used in this thesis is taken from Britting's work which discusses the mathematical representations of the Earth's gravitational field in depth [11]. The Gravity mass attraction vector $\mathbf{g}_e$ in Equation 2.2 is expressed in Earth reference frame. The Gravity model is given in the following equations.

$$R_e = \left(R_x^2 + R_y^2 + R_z^2\right)^2$$

$$\mathbf{g}_l = \mathbf{g}_e - \Omega_{ie}^e \Omega_{ie}^e \mathbf{R}_e \tag{2.19}$$

$$\mathbf{g}_e = \begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix}$$

$$g_x = -\frac{GM}{R_e^3}\left(1 - \frac{3}{2}J_2\left(\frac{a}{R_e}\right)^2\left(1 - 5\left(\frac{R_z}{R_e}\right)^2\right)\right) \cdot R_x \tag{2.20}$$

$$g_y = -\frac{GM}{R_e^3}\left(1 - \frac{3}{2}J_2\left(\frac{a}{R_e}\right)^2\left(1 - 5\left(\frac{R_z}{R_e}\right)^2\right)\right) \cdot R_y \tag{2.21}$$

$$g_z = -\frac{GM}{R_e^3}\left(1 - \frac{3}{2}J_2\left(\frac{a}{R_e}\right)^2\left(3 - 5\left(\frac{R_z}{R_e}\right)^2\right)\right) \cdot R_z \tag{2.22}$$

GM : Earth's Gravitational Constant ($3986004.418 \times 10^8 m^3/s^2$)

a : Earth equatorial radius (6378137 meters)

J_2 : Second degree harmonic (-1.082629821313)

To sum up the equations above, the gravitational acceleration vector g_l can be written as:

$$\mathbf{g}_l = \begin{bmatrix} g_x + \mathbf{w}_{ie}^2 R_x \\ g_y + \mathbf{w}_{ie}^2 R_y \\ g_z \end{bmatrix}$$

Since Earth's rotation is around the z axis, there is no contribution of it to the z component of the gravitational acceleration.

2.4.6 Summary of the Non-linear Equations of INS

$$\dot{\mathbf{C}}_b^e = \mathbf{C}_b^e \Omega_{ib}^b - \Omega_{ie}^e \mathbf{C}_b^e \tag{2.23}$$

$$\dot{\mathbf{V}}_e = \mathbf{f}_e - 2\mathbf{w}_{ie}^e \times \mathbf{V}_e + \mathbf{g}_l \tag{2.24}$$

$$\dot{\mathbf{R}}_e = \mathbf{V}_e \tag{2.25}$$

$$\mathbf{g}_l = \mathbf{g}_e + \mathbf{w}_{ie}^e \times [\mathbf{w}_{ie}^e \times \mathbf{R}_e] \tag{2.26}$$

$$\mathbf{f}_e = \mathbf{C}_b^e \mathbf{f}_b \tag{2.27}$$

2.5 Linear Error Model of INS

The equations analysized in the previous sections are the non-linear equations of INS needed to calculate approximate INS solutions of a vehicle. In this section, the error sources of the INS are summarized and these non-linear equations are linearized to get a linear error model.

The error dynamic equations are obtained when the nonlinear equations are perturbed.

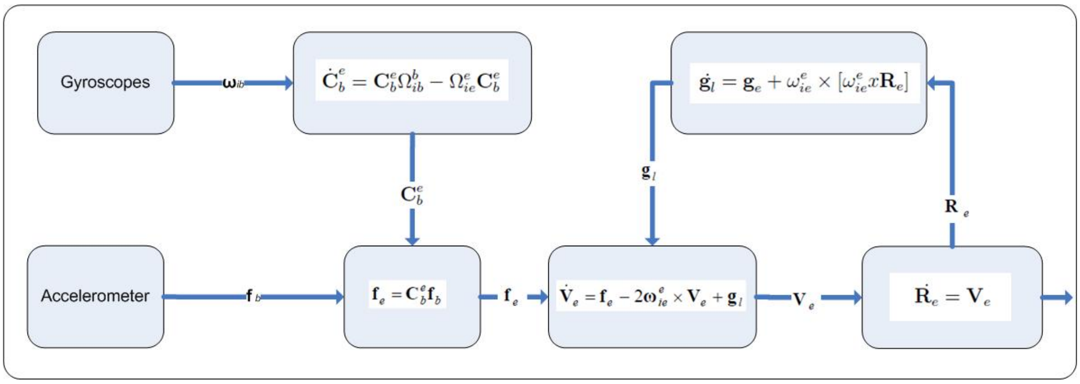

Figure 2.6: A Strapdown Navigation System

$$\hat{\mathbf{R}}_e = \mathbf{R}_e + \delta\mathbf{R}_e \tag{2.28}$$

$$\hat{\mathbf{V}}_e = \mathbf{V}_e + \delta\mathbf{V}_e \tag{2.29}$$

$$\hat{\mathbf{C}}_b^e = \mathbf{C}_b^e + \delta\mathbf{C}_b^e \tag{2.30}$$

$$\hat{\mathbf{g}}_l = \mathbf{g}_l + \delta\mathbf{g}_l \tag{2.31}$$

Error equation, by definition, is the difference between the estimated and the true value of a variable. For any variable x error is formulated as:

$$\delta\mathbf{x} = \hat{\mathbf{x}} - \mathbf{x}$$

When substitutions of the above type are made for dependent variables in the non-linear differential equations and product terms of the errors are ignored, the linear equations involving only the error quantities emerge [11].

2.5.1 Error Propagation of Attitude

Define the attitude errors about the three axes of rotation as the vector $\psi = \begin{bmatrix} \delta\alpha \\ \delta\beta \\ \delta\gamma \end{bmatrix}$.

Then from Equation 2.30 the relation between the the estimated and the true DCM is as:

$$\delta\mathbf{C}_b^e = \hat{\mathbf{C}}_b^e - \mathbf{C}_b^e \tag{2.32}$$

In the computation of the $\delta\mathbf{C}_b^e$ there are only three independent elements, the elements of the ψ. So we will first model the $\delta\mathbf{C}_b^e$ as:

$$\delta\mathbf{C}_b^e = -\Psi\mathbf{C}_b^e$$

or

$$\hat{\mathbf{C}}_b^e = [\mathbf{I} - \Psi]\mathbf{C}_b^e$$

Ψ is the skew-symmetric matrix form of the ψ. Then obtain a differential equation for it.

To derive the linear attitude error differential equation we need to start from Equation 2.32.
Rewriting the equation in a different form

$$\Psi = \mathbf{I} - \hat{\mathbf{C}}_b^e\mathbf{C}_b^{eT} \tag{2.33}$$

When we differentiate above equation with respect to time:

$$\dot{\Psi} = -\dot{\hat{\mathbf{C}}}_b^e\mathbf{C}_b^{eT} - \hat{\mathbf{C}}_b^e\dot{\mathbf{C}}_b^{eT} \tag{2.34}$$

Using the differential Equation 2.17 for the DCM:

$$\dot{\mathbf{C}}_b^e = \mathbf{C}_b^e\Omega_{ib}^b - \Omega_{ie}^e\mathbf{C}_b^e \tag{2.35}$$

$$\dot{\hat{\mathbf{C}}}_b^e = \hat{\mathbf{C}}_b^e\Omega_{ib}^b - \Omega_{ie}^e\hat{\mathbf{C}}_b^e \tag{2.36}$$

The $\hat{\Omega}_{ie}^e = 0$ assumption is used here. Combining Equations 2.34, 2.35, 2.36 and using the property that the transpose of a skew-symmetric is equal to its negative, we get the following equation:

$$\begin{aligned}
\dot{\Psi} &= -[\hat{\mathbf{C}}_b^e\hat{\Omega}_{ib}^b - \Omega_{ie}^e\hat{\mathbf{C}}_b^e]C_b^{eT} - \hat{\mathbf{C}}_b^e[C_b^e\Omega_{ib}^b - \Omega_{ie}^e\mathbf{C}_b^e]^T && (2.37)\\
\dot{\Psi} &= -\hat{\mathbf{C}}_b^e\hat{\Omega}_{ib}^b\mathbf{C}_b^{eT} + \Omega_{ie}^e\hat{\mathbf{C}}_b^e\mathbf{C}_b^{eT} + \hat{\mathbf{C}}_b^e\Omega_{ib}^bC_b^{eT} - \hat{\mathbf{C}}_b^e\mathbf{C}_b^{eT}\Omega_{ie}^e && (2.38)\\
\dot{\Psi} &= -\hat{\mathbf{C}}_b^e(\hat{\Omega}_{ib}^b - \Omega_{ib}^b)\mathbf{C}_b^{eT} + \Omega_{ie}^e\hat{\mathbf{C}}_b^e\mathbf{C}_b^{eT} - \hat{\mathbf{C}}_b^e\mathbf{C}_b^{eT}\Omega_{ie}^e && (2.39)
\end{aligned}$$

Let $\delta\Omega_{ib}^b = \triangleq \hat{\Omega}_{ib}^b - \Omega_{ib}^b$. By using this definition and Equation 2.32

$$\dot{\Psi} = -[\mathbf{I} - \Psi]\mathbf{C}_b^e\delta\Omega_{ib}^b\mathbf{C}_b^{eT} + \Psi\Omega_{ie}^e - \Omega_{ie}^e\Psi \tag{2.40}$$

Then ignoring the products of the error terms and using the properties of the skew-symmetric matrices we get the final equation as:

$$\dot{\Psi} = -\mathbf{C}_b^e \delta \mathbf{w}_{ib}^b - \Omega_{ie}^e \Psi \tag{2.41}$$

2.5.2 Error Propagation of Velocity

From the perturbation model we know that:

$$\delta \mathbf{V}_e = \hat{\mathbf{V}}_e - \mathbf{V}_e \tag{2.42}$$

Using Equation 2.11 and rewriting $\dot{\mathbf{V}}_e$ and $\dot{\hat{\mathbf{V}}}_e$ for completeness

$$\begin{aligned}
\dot{\mathbf{V}}_e &= \mathbf{C}_b^e f_b - 2\mathbf{w}_{ie}^e \times \mathbf{V}_e + \mathbf{g}_l \\
\dot{\hat{\mathbf{V}}}_e &= \hat{\mathbf{C}}_b^e \hat{\mathbf{f}}_b - 2\mathbf{w}_{ie}^e \times \hat{\mathbf{V}}_e + \hat{\mathbf{g}}_l \\
\delta \dot{\mathbf{V}}_e &= \hat{\mathbf{C}}_b^e \hat{\mathbf{f}}_b - 2\mathbf{w}_{ie}^e \times \hat{\mathbf{V}}_e + \hat{\mathbf{g}}_l - \mathbf{C}_b^e \mathbf{f}_b + 2\mathbf{w}_{ie}^e \times \mathbf{V}_e - \mathbf{g}_l
\end{aligned} \tag{2.43}$$

Substituting Equation 2.32 and the definitions below:

$$\begin{aligned}
\delta \mathbf{f}_b &= \hat{\mathbf{f}}_b - \mathbf{f}_b \\
\delta \mathbf{g}_l &= \hat{\mathbf{g}}_l - \mathbf{g}_l \\
\mathbf{f}_e &= \mathbf{C}_b^e \mathbf{f}_b^e
\end{aligned}$$

Rewriting Equation 2.43 and ignoring the errors product terms

$$\delta \dot{\mathbf{V}}_e = \mathbf{C}_b^e \delta \mathbf{f}_b - \Psi \mathbf{f}_e - 2\Omega_{ie}^e \delta \mathbf{V}_e + \delta \mathbf{g}_l$$

Using the property of skew-symmetric matrix which is $skew[\mathbf{v}] \cdot \mathbf{w} = -skew[\mathbf{w}] \cdot \mathbf{v}$ the equation becomes:

$$\delta \dot{\mathbf{V}}_e = \mathbf{C}_b^e \delta \mathbf{f}_b + skew[\mathbf{f}_e]\psi - 2\Omega_{ie}^e \delta \mathbf{V}_e + \delta \mathbf{g}_l \tag{2.44}$$

In this equation $skew[\mathbf{f}_e]$ is the matrix form of the vector $\mathbf{f}_e$.

2.5.3 Error Propagation of Position

The error equation of the position is the well-known physics formula.

$$\delta\dot{\mathbf{R}}_e = \delta\mathbf{V}_e \tag{2.45}$$

2.5.4 Error Propagation of Gravity

Using the equation of the gravity, Equation 2.19, and the perturbation model $\delta\mathbf{g}_l$ can be written as:

$$\delta\mathbf{g}_l = \delta\mathbf{g}_e - \mathbf{w}_{ie}^e \times [\mathbf{w}_{ie}^e \times \delta\mathbf{R}_e]$$

From the definition of the gravity mass attraction vector $\mathbf{g}_e$ becomes as:

$$\mathbf{g}_e = \mathbf{g}_0 + \Delta\mathbf{g} = -GM \cdot \mathbf{R}/R + \delta\mathbf{g}$$

$\mathbf{g}_0$:Spectral central force acceleration
$\Delta\mathbf{g}$:Earth oblateness effects
GM :Earth's gravitational constant
R : $\|\mathbf{R}\|_2$

$\mathbf{g}_e$ can be expressed as:

$$\delta\mathbf{g}_e = (\mathbf{G} + \Delta\mathbf{G}) \cdot \mathbf{R}_e$$

The gravity gradient matrices used above are defined as [22]:

$$\mathbf{G} = -\frac{GM}{R_3}\left[\mathbf{I} - 3\mathbf{u}_R\mathbf{u}_R^T\right]$$

$$\Delta\mathbf{G} = -\frac{3}{2}\frac{GM}{R_3}J_2\left[\frac{R_e}{R}\right]^2\left[\mathbf{I} - 5\mathbf{u}_R\mathbf{u}_R^T\right]$$

In the above equations $\mathbf{u}_R$ is the unit vector in $\mathbf{R}_e$ direction and $\mathbf{J}_2$ is the oblate gravity potential second harmonic coefficient.
Finally we can write the differential equation of the gravity error as:

$$\delta\mathbf{g}_l = (\mathbf{G} + \Delta\mathbf{G} - \Omega_{ie}^e\Omega_{ie}^e) \cdot \delta\mathbf{R}_e \tag{2.46}$$

2.5.5 INS Sensor Error Model

As discussed before the sensors are subject to errors which limit the accuracy of an inertial navigation system. Errors of the sensors propagate through the other states of the system and create errors in the computed outputs of the navigation system (position, velocity, attitude). So, an important part of the design and modelling of a navigation system is to model and simulate errors associated with gyros and accelerometers [7].

The measurement error associated with an inertial sensor is dependent on the physical operational principle of the sensor itself. Sensors are often compared on the basis of some errors like bias, scale factor, random noise etc.

The major sources of errors for accelerometers are:

- fixed bias
- random bias
- scale factor errors
- cross-coupling errors
- misalignment errors

The major sources of errors for gyroscopes are:

- fixed bias
- acceleration dependent bias
- scale factor errors
- cross-coupling errors
- misalignment errors

From the work of Titterton and Weston, Strapdown Navigation Technology [26], we can write the error equation of the sensors as follows:

For gyroscopes:

$$\begin{bmatrix} \delta w_x \\ \delta w_y \\ \delta w_z \end{bmatrix} = \mathbf{b} + \mathbf{b}_g \begin{bmatrix} \delta f_x \\ \delta f_y \\ \delta f_z \end{bmatrix} + \boldsymbol{S}_f \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} + \mathbf{M} \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} + \mathbf{w}$$

For accelerometer:

$$\begin{bmatrix} \delta f_x \\ \delta f_y \\ \delta f_z \end{bmatrix} = \mathbf{b} + \boldsymbol{S}_f \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} + \mathbf{M} \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} + \mathbf{w}$$

where

f_x, f_y, f_z : actual accelerations

w_x, w_y, w_z : actual angular rates

$\mathbf{b}$: fixed biases(a [3x1] matrix)

$\mathbf{b}_g$: g dependent bias coefficient(a [3x3] matrix)

$\boldsymbol{S}_f$: scale factor errors (a diagonal matrix)

$\mathbf{M}$: mounting misalignment matrix and cross-coupling terms

$\mathbf{w}$: the random bias

To handle the complexity reduced forms of these error models i.e the random bias w is used for this study. Inertial instrument errors can be modeled by random constants, random walk, random ramp or exponentially correlated random variables [31].

Then state equations for the bias errors which we modeled as the random walk for sensors becomes as:

$$\delta \dot{\mathbf{w}}_b = 0 + noise \tag{2.47}$$

$$\delta \dot{\mathbf{f}}_b = 0 + noise \tag{2.48}$$

where noise is assumed to be white Gaussian.

2.5.6 Summary of Linear INS Equations

Perturbation of inertial navigation equations gives linear differential equations of the basic inertial error states. Below are the linear error differential equations for inertial navigation system as:

$$\delta\dot{\mathbf{R}}_e = \delta\mathbf{V}_e \tag{2.49}$$

$$\delta\dot{\mathbf{V}}_e = \mathbf{C}_b^e\delta\mathbf{f}_b + \tilde{\mathbf{f}}_e\Psi - 2\Omega_{ie}^e\delta\mathbf{V}_e + \delta\mathbf{g}_l \tag{2.50}$$

$$\dot{\Psi} = -\mathbf{C}_b^e\delta\mathbf{w}_{ib}^b - \Omega_{ie}^e\Psi \tag{2.51}$$

Their representation in the matrix form is given in Equation 2.52

$$
\frac{d}{dt}\begin{bmatrix} \delta R_x \\ \delta R_y \\ \delta R_z \\ \delta V_x \\ \delta V_y \\ \delta V_z \\ \delta\alpha \\ \delta\beta \\ \delta\gamma \\ \delta w^b_{ib,x} \\ \delta w^b_{ib,y} \\ \delta w^b_{ib,z} \\ \delta f_{b,x} \\ \delta f_{b,y} \\ \delta f_{b,z} \end{bmatrix}
=
\begin{bmatrix}
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\delta g_{11} + w_{ie}^2 & 0 & 0 & 0 & 2w_{ie} & 0 & 0 & -f_{ez} & f_{ey} & 0 & 0 & 0 & & & \\
0 & \delta g_{22} + w_{ie}^2 & 0 & -2w_{ie} & 0 & 0 & f_{ez} & 0 & -f_{ex} & 0 & 0 & 0 & & \mathbf{C}_b^e & \\
0 & 0 & 0 & 0 & 0 & 0 & -f_{ey} & -f_{ex} & 0 & 0 & 0 & 0 & & & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & w_{ie} & 0 & & & & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -w_{ie} & 0 & 0 & & \mathbf{C}_b^e & & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & & & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix} \delta R_x \\ \delta R_y \\ \delta R_z \\ \delta V_x \\ \delta V_y \\ \delta V_z \\ \delta\alpha \\ \delta\beta \\ \delta\gamma \\ \delta w^b_{ib,x} \\ \delta w^b_{ib,y} \\ \delta w^b_{ib,z} \\ \delta f_{b,x} \\ \delta f_{b,y} \\ \delta f_{b,z} \end{bmatrix}
+ \mathbf{w}_k \qquad (2.52)
$$

2.6 Kalman Filter

In 1960, R.E. Kalman published his famous paper describing a recursive solution to the discrete-data linear filtering problem [3]. Since that time, due in large part to advances in digital computing, the Kalman filter has been the subject of extensive research and application, particularly in the area of autonomous or assisted navigation [28].

The Kalman filter is a set of mathematical equations that provide an efficient computational (recursive) means to estimate the state of a process in a way that minimizes the mean of the squared error. The filter is very powerful in several aspects as it supports estimations of past, present, and even future states and it can do so even when the precise nature of the modeled system is unknown [28].

The derivation of the detailed Kalman filter equations and the probabilistic origin of the filter are out of the scope of this study but a general information and the algorithm of this recursive process is given in the following sections.

2.6.1 Introduction to Kalman Filter

The Kalman Filter is a recursive predictive filter that is based on the use of the state space techniques and recursive algorithms. Kalman Filter estimates the state of a dynamic system. Generally this dynamic system is exposed to noise which is assumed to be white. Kalman Filter uses the measurements which are related to the state of the dynamic system to get a better estimate.

Kalman Filter has two main phases as prediction (or time update) and correction (or measurement update). In the first phase the state is predicted using the state model. In the second phase this predicted state is corrected with the measurement (or observation). The main aim of the filter is to minimize the error covariance of the estimator. The fundamental components of the Kalman Filter are the *state vector*, the *state model* and the *observation model*. The state vector consist of the state variables of interest. It describes the state of the dynamic system and represents the degree of freedom of the dynamic model. The state vector cannot be measured directly but it can be approximated from the values which are measurable [21]. The vector has two values at a point of time, these are the a priori and the a posteriori values. The a priori value is the predicted value before the measurement update is done. The a posteriori value is the corrected value after the measurement update. Throughout this study $\hat{\mathbf{x}}^{-}$ is used for a priori estimate and $\hat{\mathbf{x}}^{+}$ is used for a posteriori estimate.

2.6.1.1 State Model

The fundamental concept of the Kalman filter is the system state model. The state model describes the transformation of the state vector in time. The general formulation of a continuous time system state model is as:

$$\dot{\mathbf{x}}(t) = f(t, \mathbf{x}(t), \mathbf{u}(t)) \tag{2.53}$$

For the linear case, with no control input, the model can be represented as:

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{w}(t) \tag{2.54}$$

$\mathbf{F}$(t) : the dynamic matrix
$\mathbf{x}$(t) : the state vector of the system
$\mathbf{w}$(t) : process noise

The process noise is assumed to be white with covariance matrix $\mathbf{Q}$(t).

2.6.1.2 Observation Model

The observation model describes the relationship between the state vector and the measurements. For linear case observation model is a set of linear equations that depend on the state vector. Generally the measurements are taken at discrete times, so writing the observation model for discrete time steps, t_k:

$$\mathbf{z}(t_k) = h(t_k, \mathbf{x}(t_k), \mathbf{v}(t_k)) \tag{2.55}$$

The linear form of the measurement representation is as:

$$\mathbf{z}(t_k) = H(t_k) \cdot \mathbf{x}(t_k) + \mathbf{v}(t_k) \tag{2.56}$$

where

$\mathbf{H}(t_k)$: the observation matrix

$\mathbf{z}(t_k)$: the observation vector at time t_k

$\mathbf{v}$(t) : the noise of the measurement (or observation noise)

2.6.2 Kalman Filter Algorithm

2.6.2.1 Prediction

In this section we will give the formulas for a time invariant system although they can be extended to a time varying one.

For a dynamic model defined by the differential equation given below:

$$\dot{\mathbf{x}}(t) = \mathbf{F} \cdot \mathbf{x}(t) + \mathbf{w}(t)$$

The prior state is calculated by ignoring the contribution of the noise component because it is a zero mean function and not correlated with the earlier noise components [15].

$$\dot{\mathbf{x}}^{-}(t) = \mathbf{F} \cdot \mathbf{x}^{-}(t) \tag{2.57}$$

The solution is:

$$\mathbf{x}^{-}(t) = \Phi_0^t \cdot \mathbf{x}^{-}(t_0) \tag{2.58}$$

Φ_0^t is the *state transition matrix* which transforms the state at time t_0 to state at time t.

For the time invariant case:

$$\Phi_0^t = e^{\mathbf{F}(t-t_0)} = \sum_{k=0}^{\infty} \frac{\mathbf{F}^k (t-t_0)^k}{k!}$$

Defining

$$\Phi_{k-1}^k = e^{\mathbf{F}(t_k - t_{k-1})}$$

From the law of error propagation the a priori covariance matrix can be obtained

$$\mathbf{P}^{-}(t_k) = \Phi_{k-1}^k \cdot \mathbf{P}(t_{k-1}) \cdot (\Phi_{k-1}^k)^T + \mathbf{Q} \tag{2.59}$$

Q is assumed to be constant over time.

2.6.2.2 Correction

In the correction (or measurement update) phase the predicted state, a priori estimate $\mathbf{x}^{-}(t_k)$ is corrected with the measurement taken at time t_k. The a posteriori estimate is corrected as:

$$\mathbf{x}(t_k) = \mathbf{x}^{-}(t_k) + \Delta\mathbf{x}(t_k)$$

The covariance matrix is updated as:

$$\mathbf{P}^{+}(t_k) = \mathbf{P}^{-}(t_k) + \Delta\mathbf{P}(t_k)$$

where $\Delta P(t_k)$is as:

$$\Delta\mathbf{P}(t_k) = E\left[\Delta\mathbf{x}(t_k)\Delta\mathbf{x}(t_k)^T|\mathbf{z}(t_1), \mathbf{z}(t_2), \cdots \mathbf{z}(t_k)\right]$$

$\Delta x(t_k)$is as:

$$\Delta\mathbf{x}(t_k) = \mathbf{P}^{-}\mathbf{H}^{t}\left(\mathbf{H}\mathbf{P}^{-}\mathbf{H}^{T} + \mathbf{R}(t_k)\right)^{-1} \cdot (\mathbf{z}(t_k) - \mathbf{H}\mathbf{x}^{-}(t_k))$$

Defining

$$\mathbf{K}(t_k) = \mathbf{P}^{-}\mathbf{H}^{t}\left(\mathbf{H}\mathbf{P}^{-}\mathbf{H}^{T} + \mathbf{R}(t_k)\right)^{-1}$$

The matrix **K** is called the Kalman gain matrix.Equation 2.6.2.2 becomes as:

$$\Delta\mathbf{x}(t_k) = \mathbf{K}(t_k) \cdot (\mathbf{z}(t_k) - \mathbf{H}\mathbf{x}^{-}(t_k))$$

The difference $(\mathbf{z}(t_k) - \mathbf{H}\mathbf{x}^{-}(t_k))$ is called the innovation. Defining $\mathbf{z}^{-}(t_k) = \mathbf{H}\mathbf{x}^{-}(t_k)$. When we rewrite the updated state formula as:

$$\mathbf{x}_{+}(t_k) = \mathbf{x}^{-}(t_k) + \mathbf{K}(t_k) \cdot (\mathbf{z}(t_k) - \mathbf{z}^{-}(t_k))$$

The covariance matrix of the posterior state becomes:

$$\mathbf{P}^{+}(t_k) = \mathbf{P}^{-}(t_k) - \mathbf{K}(t_k)\mathbf{H}\mathbf{P}^{-}(t_k)$$

$$\mathbf{P}^{+}(t_k) = (\mathbf{I} - \mathbf{K}(t_k)\mathbf{H})\,\mathbf{P}^{-}(t_k)$$

2.6.3 Discrete Time Kalman Filter

In this section the fundamental equations of the discrete time Kalman filter are given. Before representing these equations, the discretization of a linear system given in the state space form will be shortly explained.

2.6.3.1 Discretization of Linear State Space Model

"Discretization is concerned with the transformation of continuous differential equations into discrete difference equations, suitable for numerical computing." [29].

Consider the following continuous-time state space model.

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{w}(t)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{v}(t)$$

where v and w are continuous zero-mean white noise sources with covariances:

$$\mathbf{w}(t) \sim N(0, \mathbf{Q})$$

$$\mathbf{v}(t) \sim N(0, \mathbf{R})$$

The discrete time representation, assuming zero-order hold approximation is given as follows:

$$\mathbf{x}[k+1] = \mathbf{A}_d\mathbf{x}[k] + \mathbf{B}_d\mathbf{u}[k] + \mathbf{w}[k]$$

$$\mathbf{y}[k] = \mathbf{C}_d\mathbf{x}[k] + \mathbf{v}[k]$$

with covariances

$$\mathbf{w}[k] \sim N(0, \mathbf{Q}_d)$$

$$\mathbf{v}[k] \sim N(0, \mathbf{R}_d)$$

where

$$\mathbf{A}_d = e^{\mathbf{AT}}$$

$$\mathbf{B}_d = \left(\int_{\tau=0}^{T} e^{\mathbf{A}\tau} d\tau\right)\mathbf{B} = \mathbf{A}^{-1}(\mathbf{A}_d - I)\mathbf{B}$$

if $\mathbf{A}$ is nonsingular.

$$\mathbf{C}_d = \mathbf{C}$$
$$\mathbf{Q}_d = \int_{\tau=0}^{T} e^{\mathbf{A}\tau}\mathbf{Q}e^{\mathbf{A}^T\tau}d\tau$$
$$\mathbf{R}_d = \mathbf{R}$$

and T is the sampling period.

Approximations Zero-order hold discretization may sometimes be intractable due to the heavy matrix exponential and integral operations involved. It is much easier to calculate an approximate discrete model, based on small time step assumption:

$$e^{\mathbf{A}T} \approx \mathbf{I} + \mathbf{A}T$$

The approximate solution then becomes

$$\mathbf{x}[k+1] \approx (\mathbf{I} + \mathbf{A}T)\mathbf{x}[k] + (\mathbf{I}T + \frac{1}{2}\mathbf{A}T^2)\mathbf{B}\mathbf{u}[k] + \mathbf{w}[k]$$

which can further be approximated if $\frac{1}{2}\mathbf{A}T^2$ is small; yielding

$$\mathbf{x}[k+1] \approx (\mathbf{I} + \mathbf{A}T)\mathbf{x}[k] + T\mathbf{B}\mathbf{u}[k] + \mathbf{w}[k]$$

2.6.3.2 Discrete Time KF Equations

The KF addresses the general problem of estimating the state $\mathbf{x}$ of a discrete-time process that is governed by the linear stochastic difference equation:

$$\mathbf{x}_k = \mathbf{A}_k\mathbf{x}_{k-1} + \mathbf{w}_{k-1} \tag{2.60}$$

with a measurement $\mathbf{z}_k$

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \tag{2.61}$$

The variables $\mathbf{w}_k$ and $\mathbf{v}_k$ represent the process and measurement noise respectively. They are assumed to be independent of each other, white, and with normal probability distributions $\mathbf{w}_k \sim \mathrm{N}(0,\mathbf{Q})$ and $\mathbf{v}_k \sim \mathrm{N}(0,\mathbf{R})$.

An initial estimate of the process at some discrete time t_k is assumed, and this estimate is based on our knowledge of the process prior to k. Let this a priori estimate be denoted by $\hat{x}_k^-$, where the hat denotes estimate, and the super minus reminds us that this is the best estimate we have prior to assimilating the measurement at k. Assuming that the error covariance matrix associated with $\hat{\mathbf{x}}_k^-$ is also known, and is denoted by $\mathbf{P}_k^-$, define the estimation error as:

$$\hat{\mathbf{e}}_k^- = \mathbf{x}_k^- \hat{\mathbf{x}}_k^- \tag{2.62}$$

The associated covariance matrix is as:

$$\mathbf{P}_k^- = E[\hat{\mathbf{e}}_k - \hat{\mathbf{e}}_k^{-T}] \tag{2.63}$$

Since we have assumed a prior estimate $\hat{\mathbf{x}}_k^-$, we use $\mathbf{z}_k$ to improve the prior estimate, using the following equation.

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-) \tag{2.64}$$

where $\hat{\mathbf{x}}_k^+$ is the updated estimate and $\mathbf{K}_k$ is the blending factor or Kalman gain that minimizes the a posteriori error covariance equation.

$$\mathbf{P}_k^+ = E[\hat{\mathbf{e}}_k^+ \hat{\mathbf{e}}_k^{+T}] \tag{2.65}$$

where

$$\hat{\mathbf{e}}_k^+ = \mathbf{x}_k - \hat{\mathbf{x}}_k^+$$

The error covariance matrix is given as:

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\,\mathbf{P}_k^- \tag{2.66}$$

With the Kalman gain

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_k^T\left(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k\right)^{-1} \tag{2.67}$$

For the next time step the next time prediction is done with the equations

$$\hat{\mathbf{x}}_{k+1}^- = \mathbf{A}_k\hat{\mathbf{x}}_k^+ \tag{2.68}$$

$$\mathbf{P}_{k+1}^- = \mathbf{A}_k\mathbf{P}_k^+\mathbf{A}_k^T + \mathbf{Q}_k \tag{2.69}$$

The summary of the discrete time Kalman filter equations are given in Table 2.1 [5].

Table 2.1: Discrete Time Kalman Filter Equations

State Model	$\mathbf{x}_k = \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{w}_k$, $\mathbf{w}_k \sim$ N(0,$\mathbf{Q}$)
Measurement Model	$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k$, $\mathbf{v}_k \sim$ N(0,$\mathbf{R}$)
Initial Conditions	$E[\mathbf{x}(0)] = \hat{\mathbf{x}}_0 \; E\left[(\mathbf{x}(0) - \hat{\mathbf{x}}(0))(\mathbf{x}(0) - \hat{\mathbf{x}}(0))^T\right] = \mathbf{P}_0$
Assumptions hline State Estimate Extrapolation	E[$w_i v_j$]=0 for all i and j w_i nad v_j are white $\hat{\mathbf{x}}_k^- = \mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1}^+$
Error Covariance Extrapolation	$\mathbf{P}_k^- = \mathbf{F}_{k-1}\mathbf{P}_{k-1}^+\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1}$
State Estimate Update	$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-)$
Error Covariance Update	$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\,\mathbf{P}_k^-$
Kalman Gain Matrix	$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_k^T\left(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k\right)^{-1}$

2.6.3.3 Extended Kalman Filter

Kalman filter estimates the states of a linear system in an optimal way. However most of the systems are not linear. To apply Kalman filtering to nonlinear systems usually a linearization is applied. Assume that the state space representation of a nonlinear system is given as:

$$x_{k+1} = f(k, x_k) + w_k$$

$$y_k = h(k, x_k) + v_k$$

The time update of the state is done in the most natural way as follows:

$$x_{k+1}^- = f(k, x_k^+)$$

To update the covariance matrix, the state and the measurement equations are linearized around the $\hat{x}_k^+$ to obtain

$$\mathbf{F}_k = \left.\frac{\partial f}{\partial x}\right|_{\hat{x}_k^+}$$

$$\mathbf{H}_k = \left.\frac{\partial h}{\partial x}\right|_{\hat{x}_k^+}$$

These matrices are directly used in the update equations as if the system is linear

$$\mathbf{P}_{k+1}^{-} = \mathbf{F}_k \mathbf{P}_k^{+} \mathbf{F}_k^{T} + \mathbf{Q}_k \tag{2.70}$$

For the measurement update the innovation is computed using the nonlinear measurement equation.

$$z_{k+1} - h(k+1, \hat{x}_{k+1}^{-}) \tag{2.71}$$

Kalman gain and the state covariance are calculated as:

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1}^{-} \mathbf{H}_k^{T} \left(\mathbf{H}_k \mathbf{P}_{k+1}^{-} \mathbf{H}_k^{T} + \mathbf{R}_{k+1} \right)^{-1} \tag{2.72}$$

$$\mathbf{P}_{k+1}^{+} = \left(\mathbf{I} - \mathbf{K}_{k+1} \mathbf{H}_{k+1} \right) \mathbf{P}_{k+1}^{-} \tag{2.73}$$

The measurement update of the state is as follows:

$$\hat{\mathbf{x}}_{k+1}^{+} = \hat{\mathbf{x}}_{k+1}^{-} + \mathbf{K}_{k+1} [\mathbf{z}_{k+1} - h(k+1, \hat{\mathbf{x}}_{k+1}^{-})] \tag{2.74}$$

2.6.4 TAN Kalman Filter

For Terrain Aided Navigation Kalman Filter we use the linear INS error model equations to construct the dynamic model, so that our aim is to estimate the error for each time to correct the solution. The estimation is done by Kalman filtering. The Kalman filter structure used for this study is expressed as TAN Kalman in this thesis because this structure is not fully same with any other algorithm found in literature. To construct the TAN Kalman filter, we have to develop our measurement model for aided navigation. As we have mentioned before the measurements are the terrain heights from DTED and the radar altimeter.

Measurement model We know that the measurement model equation is

$$\mathbf{z}(k) = \mathbf{H}(k) \delta \mathbf{x}(k) + \mathbf{v}(k) \tag{2.75}$$

In general, the linearized form of the measurement equation is written as:

$$\mathbf{z}(k) = \hat{\mathbf{z}}_k - \mathbf{z}_k = \mathbf{H}_k \delta \mathbf{x} + \mathbf{v}_k \tag{2.76}$$

where $\hat{\mathbf{z}}_k$ is the predicted value of the height and the $\mathbf{z}_k$ is the measured height above the terrain, $\mathbf{v}_k$ is the measurement noise which is white Gaussian and $\mathbf{H}$ is the observation matrix.

To estimate the errors of the INS system the following measurements are used as aid.

- Radar altimeter measurement: measures the height difference between the platform and the terrain
- DTED measurement gives the height of any point on the Earth as a function of x (latitude) and y (longitude)
- Barometric altimeter: We actually do not have a barometric altimeter, instead the $\hat{R}_z$ output of the INS system gives this value

We must say that all these measurements are noisy. Furthermore, the measurement of the barometric altimeter depends on R_z, the radar altimeter gives the measurements according to R_x and R_y positions of the platform and DTED gives its output according to the estimated R_x and R_y. If all the measurements were error free and estimated R_x and R_y were correct we would have:

$$h_{baro}(\hat{R}_z) - h_{ter}(\hat{R}_x, \hat{R}_y) - h_{RA}(R_x, R_y) = 0 \tag{2.77}$$

where

$$h_{baro}(R_z) = \hat{R}_z$$

The deviations of the right hand side of this equation form the ideal value which is zero contains the information about the errors of the estimation. To clarify this point assume measurements are error free and the only error is due to the INS system.

$$z = \hat{R}_z - h_{ter}(\hat{R}_x, \hat{R}_y) - h_{RA}(R_x, R_y) \tag{2.78}$$

Define

$$\delta R_x = R_x - \hat{R}_x$$

$$\delta R_y = R_y - \hat{R}_y$$

$$\delta R_z = R_z - \hat{R}_z$$

Then the error measurement z can be written as follows:

$$z = R_z - \delta R_z - h_{ter}(\hat{R}_x, \hat{R}_y) - h_{RA}(R_x, R_y) \tag{2.79}$$

Note that

$$h_{ter}(R_x, R_y) = h_{ter}(\hat{R}_x + \delta R_x, \hat{R}_y + \delta R_y)$$

Writing the Taylor series expression of the $h_{ter}(R_x, R_y)$ around R_x and R_y and omitting the second order terms we obtain

$$h_{ter}(R_x, R_y) = h_{ter}(\hat{R}_x, \hat{R}_y) + \left.\frac{\partial h_{ter}}{\partial R_x}\right|_{\hat{R}_x,\hat{R}_y} \delta R_x + \left.\frac{\partial h_{ter}}{\partial R_y}\right|_{\hat{R}_x,\hat{R}_y} \delta R_y \tag{2.80}$$

or using this expression we can write that

$$h_{ter}(\hat{R}_x, \hat{R}_y) = h_{ter}(R_x, R_y) - \left.\frac{\partial h_{ter}}{\partial R_x}\right|_{\hat{R}_x,\hat{R}_y} \delta R_x - \left.\frac{\partial h_{ter}}{\partial R_y}\right|_{\hat{R}_x,\hat{R}_y} \delta R_y \tag{2.81}$$

Substituting the above expressions into Equation 2.79 gives

$$z = R_z - \delta R_z - h_{ter}(R_x, R_y) + \left.\frac{\partial h_{ter}}{\partial R_x}\right|_{\hat{R}_x,\hat{R}_y} \delta R_x + \left.\frac{\partial h_{ter}}{\partial R_y}\right|_{\hat{R}_x,\hat{R}_y} \delta R_y - h_{RA}(R_x, R_y) \tag{2.82}$$

Since

$$R_z = h_{ter}(R_x, R_y) - h_{RA}(R_x, R_y)$$

We obtain

$$z = -\delta R_z + \left.\frac{\partial h_{ter}}{\partial R_x}\right|_{\hat{R}_x,\hat{R}_y} \delta R_x + \left.\frac{\partial h_{ter}}{\partial R_y}\right|_{\hat{R}_x,\hat{R}_y} \delta R_y \tag{2.83}$$

Then the measurement equation is given by the following relation

$$z(k) = \mathbf{H}(k)\delta\mathbf{x}(k) + v(k) \tag{2.84}$$

where

H(k): Observation matrix

$$H(k) = \begin{bmatrix} -h_x(k) & -h_y(k) & 1 \end{bmatrix} \tag{2.85}$$

$$x(k) = \begin{bmatrix} \delta R_x & \delta R_y & \delta R_z \end{bmatrix}^T \tag{2.86}$$

where

$$h_x(k) = \left.\frac{\partial h_{ter}}{\partial R_x}\right|_{\hat{R}_x,\hat{R}_y}$$

$$h_y(k) = \left.\frac{\partial h_{ter}}{\partial R_y}\right|_{\hat{R}_x,\hat{R}_y}$$

The correlation between the estimated and the actual positions is the most important key feature of the TAN algorithms.

At the original paper of the SITAN the terrain fitting curve and linearization is discussed in some detail [12]. Consider a fitted function f(x, y) to the terrain profile being expanded near INS position $(\hat{x}, \hat{y})$.

$$f(x,y) = f(\hat{x},\hat{y}) + \frac{\partial f}{\partial x}(\hat{x} - x) + \frac{\partial f}{\partial y}(\hat{y} - y) \tag{2.87}$$

where

$\partial f / \partial x$: Terrain slope along eastward direction

$\partial f / \partial y$: Terrain slope along northward direction

Define terrain slope along eastward direction as h_x and terrain slope along northward direction as h_y.

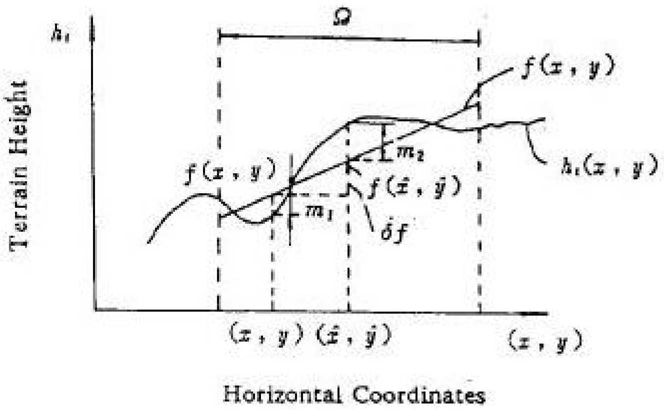

Figure 2.7: Terrain Stochastic Linearization (TSL) (taken from [27])

Figure 2.7 shows the mechanism of the Terrain Stochastic Linearization(TSL)[27].

$$h(x,y) = f(x,y) + m_1 \tag{2.88}$$

$$h(\hat{x},\hat{y}) = f(\hat{x},\hat{y}) + m_2 \tag{2.89}$$

where

m_1 : TSL error at real point (x,y)

m_2 : TSL error at the estimate point $(\hat{x}, \hat{y})$

2.6.5 Measurement Model Related Errors

The errors of the measurement model can be classified as three parts [27]

- DTED error which is introduced during DTED generation, ϵ_{dted}
- The theoretical error of TSL, from TSL at the actual position

$$m_1 = h(x, y) - f(x, y)$$

- The methodological error of TSL, from TSL at the estimated position

$$m_2 = h(\hat{x}, \hat{y}) - f(\hat{x}, \hat{y})$$

In addition, m_2 is dependent to the sampling rate of the system. Modelling of the DTED related errors is an important part of the TAN algorithms and performance. It is a different and huge issue that is studied by many researchers.

Improvement of the estimates by obtaining better TSL is beyond the subject of this thesis. A detailed study can be found at the paper of Wei Wang and Zhe Chen [27].

The measurement model described here is similar to the measurement model of one of the terrain aided navigation systems called SITAN. In the following chapter we will give basic information about the TAN algorithms.

CHAPTER 3

MAJOR TERRAIN AIDED NAVIGATION METHODS

3.1 Introduction

There are several algorithms that solve the terrain aided navigation problem. In this chapter, two major terrain aided navigation algorithms, TERCOM and SITAN, will be explained in detail. In addition, other major terrain aided navigation approaches will be mentioned to give a general information about the terrain aided navigation methods. Since the development of Terrain Aided Navigation has been driven by military interests, most of the algorithms are not very well documented in the literature. Because of this reason research in this topic becomes more challenging.

3.2 TERCOM

"TERCOM algorithm was first patented in 1958, relies for its operating principle on the simple fact that the altitude of the ground above sea level varies as a function of location." [4].

The chronological overview of TERCOM process is given in Table 3.1.

Table 3.1: Chronological Overview of TERCOM (taken from [4])

PROGRAM	YEAR	OBJECTIVES
Fingerprint	1958	Guidance package for SLAM missile TERCOM concept first proposed.
TERCOM	1960-1961	Feasibility study of terrain contour matching.
LACOM (Low Altitude Contour Matching)	1963-1965	Design and development of a complete fix-taking subsystem.
RACOM (Rapid Contour Matching)	1963-1966	Improve TERCOM computation procedures and increase accuracy.
SAMSO (USAF's Space and Missiles Systems Organization) Programs (a) TPLS (Terminal Position Location System (b) TERSE (Terminal Sensing Experiment) (c) TERF (Terminal Fix). (d) TSOFT (Terminal Sensor Overland Flight Test).	1963-1971	Application of terrain correlation techniques for ballistic missiles.
Avionics Update	1972-1975	Study and define a TERCOMI drone system capable of operational deployment.
TAINS (Terrain Aided INS) TERCOM	1972-1974	Feasibility study for incorporation in cruise missile and evaluation of snow coverage effects on terrain profile acquisition.
Competitive Flyoff	1975	McDonnell – Douglas Astrodynamics awarded a contract for TERCOM system.
RACOM (Recursive All Weather Contour Matching)	1975	Improve terrain correlation update accuracy.

3.2.1 Definition of TERCOM

Terrain counter Matching (TERCOM) is in basic words is an algorithm to decide the position fix of any moving platform utilizing the information of terrain height profile and its correlation with the stored digital terrain elevation map.

"TERCOM is a form of correlation guidance based on a comparison between the measured and prestored features of the profile of the terrain (ground) over which a missile or aircraft is flying. The TERCOM technique relies for its operating principle on the simple fact that the altitude of the ground above sea level as a function of location." [25].

3.2.2 Fundamental Phases of the TERCOM

TERCOM algorithm consists of a radar altimeter, a computer to calculate position fix and a digital elevation database that is stored on the computer of the system that TERCOM operate on.

The process of determining air vehicle position by the use of terrain counter matching can generally be described as consisting of three basic steps:

- Data preparation
- Data acquisition
- Data correlation

Data preparation: Data preparation consists of selecting a fix area large enough to accommodate the crosstrack and downtrack navigation arrival uncertainties and digitizing the terrain elevation data into a matrix of cells oriented along the intended flight path. The resulting reference matrix consists of an array of numbers that represent discrete terrain heights corresponding to a sampling interval (resolution) equal to the desired cell size.

This reference matrix or map is then stored in the vehicle's onboard digital computer memory prior the flight. One of the most important aspects of TERCOM data preparation is selecting the area that is to be digitized and used as a reference matrix.

Data acquisition: As the vehicle flies over the fix point area, data are acquired by sampling the altitude of the vehicle above the terrain directly below it at an interval equal to the reference map cell size.

This sensed altitude is measured with a radar altimeter. At the same time, the vehicle's altitude above mean sea level is measured using a combination of a barometric altimeter and vertical accelerometers to provide the system reference altitude. The acquired terrain elevation samples are then stored in a file in the vehicle's onboard digital computer. This terrain elevation file represents a discrete elevation profile of the terrain along a line coincident with the vehicle's ground track.

The collected data are in the form of a horizontally arranged matrix of digital elevation numbers that describe a terrain profile. The length of the counter profile necessary for a unique fit is a function of terrain roughness. TERCOM profile acquisition system consists of a radar terrain sensor (RTS) or a radar altimeter (RA) and a reference altitude sensor (RAS) or a barometric altimeter.

Data correlation: The last step in the TERCOM process is the correlation of the data in the file with each column of the reference matrix. Basically the reference column that has the greatest correlation with the terrain elevation file is the column down which the vehicle has flown. With no navigation error, the matching column would be the center column of the terrain elevation file, since that is the ground track the navigation system is steering along. However, with downtrack and crosstrack errors, it is probable that some column other than the center one will be flown down.

In this case, the system computes the downtrack and crosstrack distance from the center of the map and uses these errors to correct the vehicle's navigation error. Generalized block diagram of the TERCOM system is shown in Figure 3.1 [25].

The left side of the diagram describes the reference data loop. Source materials in the form of survey maps or stereo-photographs of the terrain are used to collect the set of

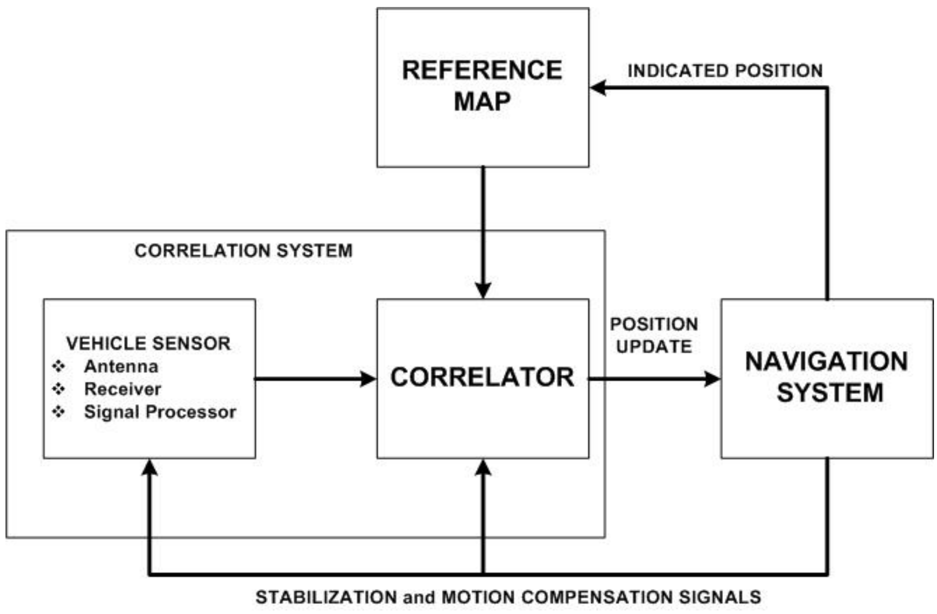

Figure 3.1: TERCOM Algorithm-1

altitudes that constitute the reference matrix. The right side of the diagram describes the data acquisition loop.

The radar altimeter acquires altitude estimates above terrain. The radar altimeter output is differenced with the system's reference altitude. Then arithmetic operations (ex: mean removal, quantization) are performed on the differenced data. Finally the correlation between the stored and acquired data is performed, and a position fix is determined [25].

3.2.3 The Measurement process

When the platform flies over the terrain, the radar altimeter measurements are gathered. These measurements are taken with a sampling rate of at least passing from each grid area to another. When more than one measurement is taken for that grid area, the average measurement is gathered as radar altimeter measurement for that grid. As stated before the radar altimeter measures the height of the platform from the terrain on which it flies. In addition INS also gives the barometric altimeter supported altitude for the platform. The terrain height profile is generated by using these two measurement. The resulting terrain height profile is used in combination with the digital terrain elevation map to correlate with any terrain profile on the map [25].

3.2.4 Data Correlation Techniques

"Since the TERCOM system is not noiseless, the terrain profile measured during flight will probably never exactly match one of the reference matrix profiles. A fundamental assumption of the terrain correlation process is that the geographic distance between the measured terrain elevation profile and the best-matching reference matrix column provide an excellent measure of the downtrack and crosstrack position errors of the vehicle as it flies over the reference matrix area. There are a number of correlation algorithms as Mean Squared Distance (MSD), Mean Absolute Distance (MAD), the normalized MAD, the normalized MSD, the product method which have varying complexity and accuracy that can be used to correlate the measured data with the reference data. The MAD algorithm provides the best combination of accuracy and computational efficiency for performing real time terrain counter matching in an onboard computer environment." [25].

There are two main approaches for the computation of correlation. First is called Short Sample Long Matrix (SSLM) and the other is Long Sample Short Matrix (LSSM). We will briefly explain SSLM methods. To apply the method we get a batch of terrain heights from the measuring instrument like barometric altimeter or radar altimeter during the flight. These measurements form the short sample sequence and denoted by $\mathbf{h}_k, = (h_1, h_2, \cdots, h_N)$, i.e., $\mathbf{h}_k$ is a vector of size N which is the order of 10's. Note that during the complete flight the overall measurement vector will be in the form $[\mathbf{h}_1^T \mathbf{h}_2^T \cdots \mathbf{h}_{total}^T]^T$ when N*T is the number of total measurement. We will partition the DTED matrix $\mathbf{H}$ similarly and write it as $\mathbf{H}$=$[\mathbf{H}_1^T \mathbf{H}_2^T \cdots \mathbf{H}_{total}^T]^T$ when $\mathbf{H}_i$ is a NxM matrix of the elevation map. The idea of TERCOM is to compare $\mathbf{h}_i$ with all columns of $\mathbf{H}_i$ and select the one which minimizes the l_1 norm of $||\mathbf{h}_i - \mathbf{H}_i^m||$ where $\mathbf{H}_i^m$ is the m^{th} column of $\mathbf{H}_i$.

The Terrain correlation processing is showed in Figure 3.2 [25].

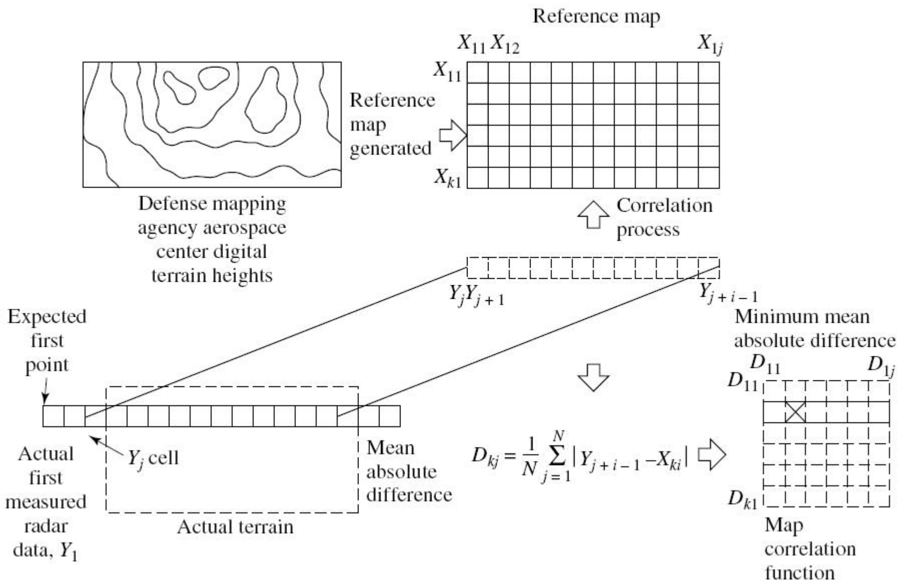

Figure 3.2: TERCOM Algorithm-2 (taken from [25])

3.2.5 TERCOM Performance

The performance of the terrain correlation system is achieved by taking into account the uncontrollable errors in the system. These error parameters can be summarized as:

- Map
  - Terrain mapping errors that are a function of DMAAC (Defense Mapping Agency Aerospace Center) procedures and equipment
  - Map quantization
  - Cell-size sampling errors
- Inertial platform errors allowed by its specification
- Radar altimeter
  - RA noise errors including beamwidth blurring as a function of altitude allowed by its specification
  - Natural errors in the elevation such as snow, tree leaves and buildings.

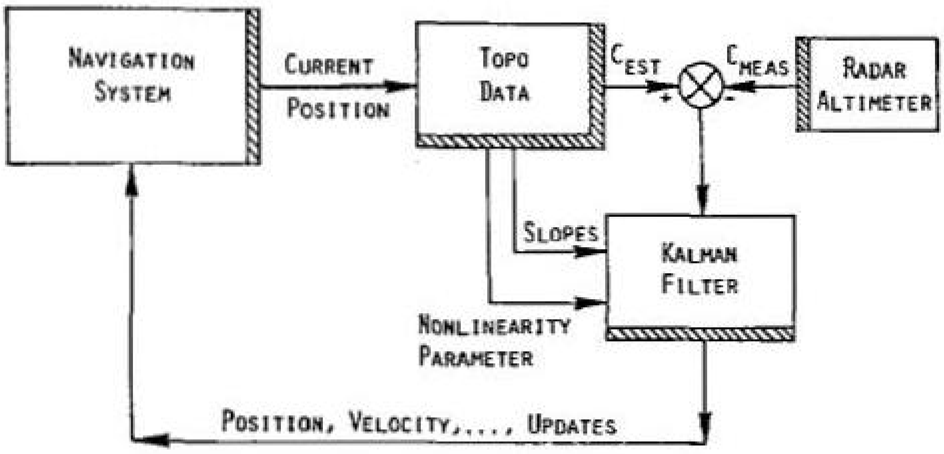

Figure 3.3: Terrain Aided Navigation (SITAN) (taken from [12])

3.3 SITAN

Major and most famous recursive terrain aided navigation algorithm found in literature is SITAN. In order to investigate SITAN, original work of Hostetler and Andreas[12] about SITAN will be investigated in detail.

The basic configuration for SITAN is shown in Figure 3.3. This structure is typical for Kalman filtering in which nonlinear auxiliary measurements are iteratively processed to estimate and compensate the errors of the navigation system. At each measurement update time the current state estimate in conjunction with stored topographical data (i.e. terrain elevation data) is used to obtain a prediction of what the radar ground clearance measurement should be. The actual radar measurement is then compared with this predicted measurement, and their difference is processed by the Kalman filter to generate estimates of the navigation system's error states. The measurement matrix in this case is related to the downrange and cross range terrain slopes calculated from the stored data. The error estimates are then fed back to compensate the navigation system and thus provide an improved estimate of the actual state (position, velocity. etc.) of the system. This process is iterated approximately for every 30-50 m of distance traveled, as the system maneuvers along its trajectory, thus providing essentially continuous updating to the navigation system [12].

To apply Kalman filter propagation of the navigation error states are needed. For SITAN algorithm the state error dynamics are assumed to be good linear approximations. The bias errors are included in the state vector and constantly updated to maintain the adequacy of such a representation [12].

For SITAN algorithm the navigation state vector is given as:

$$\bar{\mathbf{x}} = [x\ y\ h\ v_x\ v_y]^T$$

where

x : horizontal coordinates along eastward direction

y : Horizontal coordinates along northward direction

v_x : velocity along eastward direction

v_y: velocity along northward direction

h : height above sea level

The error state $\delta\bar{\mathbf{x}}$ for the above navigation state vector $\bar{x}$ becomes as:

$$\delta\bar{x} = \begin{bmatrix} \delta x \\ \delta y \\ \delta h \\ \delta v_x \\ \delta v_y \end{bmatrix}$$

Then for a sampling rate of T, the error state equation is written as:

$$\delta\bar{\mathbf{x}}(k+1) = \mathbf{F}(k) \cdot \delta\bar{\mathbf{x}}(k) + \mathbf{w}(k)$$

where

$\delta\bar{\mathbf{x}}$: error state vector

$\mathbf{F}$(k) : state transition matrix

$\mathbf{w}$(k) : process noise

$$\mathbf{F}(k) = \begin{bmatrix} 1 & 0 & 0 & T & 0 \\ 0 & 1 & 0 & 0 & T \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{w}(k) = \begin{bmatrix} w_x(k) \\ w_y(k) \\ w_h(k) \\ w_{v_x}(k) \\ w_{v_y}(k) \end{bmatrix}$$

In order to implement Kalman filter we also need the observation model. The only measurement is the terrain-clearance measurement. The following approximation approach is used for SITAN algorithm:

$$c_k = c_k(x_k) + v_k$$

$$c_k = z_k - h(CR_k, DR_k) + v_k$$

where

$h(CR_k, DR_k)$: height of the terrain at position (CR_k, DR_k)
CR_k : Cross Range position
DR_k : Down Range position
v_k : the measurement error

The measurement error consists of both radar altimeter error and reference map errors. Then δh becomes as:

$$\delta h = C_{est} - C_{meas}$$

where

C_{est} : the difference between estimated height above sea level from barometric altimeter and the terrain height from DTED based on the INS data

C_{meas} : the measured relative height

The system measurement equation of SITAN in discrete time form can be formulated as:

$$\mathbf{z}(k) = \mathbf{H}(k) \cdot \delta\bar{x}(k) + \mathbf{w}_{meas}(k)$$

$$\mathbf{H}(\mathrm{k}) = [h_x \ h_y \ 1 \ 0 \ 0]$$

where h_x and h_y are slopes in the x and y directions, respectively.

The Extended Kalman filter equations are presented in Table 3.2.

If the position errors are sufficiently small relative to the terrain correlation length, it can be assumed that the terrain is linear and simple first-order derivatives yields satisfactory performance. When the position errors are comparable to the terrain correlation length, more sophisticated techniques are required [12].

At the original work of the SITAN, three different linearization approaches for treating nonlinearities are discussed. First two of them are local slope linearization and stochastic linearization which attempt to maintain satisfactory performance using a single Kalman filter. The third approach utilizes parallel Kalman filters, each linearized over a different region [12]. The original work of the SITAN is not a sufficient material to fully understood and implement the SITAN algorithm, one need to check the details of the algorithm from different references since only the general philosophy of the algorithm is expressed in this work.

3.4 HELI/SITAN

“Heli/SITAN is a terrain referenced navigation algorithm that uses the radar altimeter measurement in combination with a conventional navigation system and a stored dig-

Table 3.2: Extented Kalman Filter Equations for SITAN

SITAN System Error Model	$\delta\bar{\mathbf{x}}(k+1) = \mathbf{F}(k) \cdot \delta\bar{\mathbf{x}}(k) + \mathbf{w}(k)$
SITAN Measurement Model	$\mathbf{z}(k) = \mathbf{H}(k) \cdot \delta\bar{\mathbf{x}}(k) + \mathbf{w}_{meas}(k)$
Initial Conditions	$\delta\bar{\mathbf{x}}_0 \sim N(\delta\mathbf{x}_0, \mathbf{P}_0)$
Assumptions	$E[w(k) \cdot w_{meas}(k)^T] = 0$ for all i and j
State Estimate Propagation	$\delta\hat{\bar{\mathbf{x}}}(k\|k-1) = \mathbf{F}(k-1)\delta\hat{\bar{\mathbf{x}}}(k-1\|k-1)$
Error Covariance Propagation	$\mathbf{P}(k\|k-1) = \mathbf{F}(k-1)\mathbf{P}(k-1\|k-1)\mathbf{F}(k-1)^T + \mathbf{Q}(k-1)$
State Estimate Update	$\delta\hat{\bar{\mathbf{x}}}(k\|k) = \delta\hat{\bar{\mathbf{x}}}(k\|k-1) + \mathbf{K}(k)[\mathbf{z}(k) - \mathbf{H}(k)\delta\hat{\bar{\mathbf{x}}}(k\|k-1)]$
Error Covariance Update	$\mathbf{P}(k\|k) = (\mathbf{I} - \mathbf{K}(k)\mathbf{H}(k))\,\mathbf{P}(k\|k-1)$
Kalman Gain Matrix	$\mathbf{K}(k) = \mathbf{P}(k\|k-1)\mathbf{H}(k)^T \left(\mathbf{H}(k)\mathbf{P}(k\|k-1)\mathbf{H}(k)^T + \mathbf{R}(k)\right)^{-1}$

ital elevation map to accurately estimate a helicopter's position. Heli/SITAN utilizes Multiple Model Adaptive Estimation (MMAE) techniques. It is developed by Sandia National Laboratories for a work which is founded by the U.S. Army AVionics Reseach And Development Activity (AVRADA). The Heli/SITAN algorithm estimates the errors in a reference navigation system by processing the radar altimeter measurements, the barometric altimeter measurements and the stored digital terrain elevation values with a bank of one state Kalman filters. Since Kalman filters are recursive, the altimeter measurements are incorporated into the current position estimate as they become available. This is the main difference of Heli/SITAN from the batch oriented processing techniques. As stated before Heli/SITAN algorithm is designed for helicopters which tend to fly short distances at slow speeds. Thus, for helicopters it is important to provide position fixes as often as possible. Besides, high immunity to false fixes is another crucial requirement for helicopters. The MMAE technique utilized by Heli/SITAN consists of a bank of Kalman filters and some position fixing algorithms for producing a position estimate based on information from the filters. The single state of Heli/SITAN is a slowly varying vertical channel bias. The sources of this bias are barometric altimeter bias, radar altimeter bias, and DTED bias. However none of these biases are individually observable, the collective bias that results from all biases is observable." [9]. A real-time implementation test results and the details of the Heli/SITAN algorithm are explained at the original work of Hollowell [9].

3.5 Other TAN Algorithms

VATAN is another Terrain Aided Navigation technique which uses the Viterbi Algorithm. The VA is a maximum a posteriori (MAP) estimator that estimates a sequence of system states from a sequence of observation values. One of the main advantages of VATAN is that the VA does not require linearization, and it is robust with respect to partially observable system models; thus, VATAN promises good performance for all aircraft over a wide range of terrain without being susceptible to the divergence problems encountered using EKF. At the original work of VATAN it is claimed that VATAN algorithm provides more accurate position estimates for both relatively flat and extremely rough terrain [24].

The simulation results and the details of the algorithm are explained at the original work of Enss and Morrell in detail [24]. The simulation results given shows that VATAN provides good performance for low-velocity aircraft flying over all types of terrain.

Bayesian Approach Terrain navigation using Bayesian statistics is a different approach of terrain referenced navigation. As other terrain navigation methods Bayesian Approach also utilizes variations in the terrain height along the aircraft flight path.

The problem of combining navigation information and height profile information has been solved by local linearization of the terrain at one or several assumed aircraft positions for most methods found in the literature, like TERCOM and SITAN. Due to changing terrain characteristics, these linearizations will in some cases result in diverging position estimates. Instead of approximating the model of the estimation problem, at the Bayesian approach the analytical solution is approximately implemented. The navigation filter computes a probability mass distribution of the aircraft position and updates this description recursively with each new measurement [18].

In contrast to the other sources of terrain navigation methods, the implementation of the algorithm and the results of the simulations are very well documented at the original work of Bayesian approach [18].

3.6 The TAN approach used for Thesis

The algorithm presented and implemented in this thesis uses the measurement model of the SITAN algorithm. However we have used INS equations directly in the state model. Such an approach is similar to the approach of tightly coupled GPS/INS integration.

The LKF and EKF TAN algorithms implemented using MATLAB. The performances of these algorithms are observed by applying them to the data generated by the simulation environment implemented. The details of the EKF structure and the the results of the simulations are given in Chapter 2 and Chapter 4 respectively. Like SITAN this is a recursive approach. Throughout the thesis the implemented algorithms are expressed as LKF TAN and EKF TAN algorithms.

CHAPTER 4

SIMULATION RESULTS AND SIMULATION TOOLS DEVELOPED

4.1 Introduction

In this chapter, simulation results of the implemented TAN algorithms are presented for a Strapdown INS model. First, a flight profile generator tool is developed in order to get a dynamic simulation environment. Next an INS simulation tool is developed that includes the navigation system errors.

Other simulation tools such as Terrain Module, Radar Altimeter Simulator are also implemented for a complete system simulation. Strapdown INS simulations are performed with different flight profiles and different sampling rates.

The main aim of the thesis is to use the terrain information to correct the errors of the INS system. The correction is done by integrating the terrain information to INS by using Kalman filtering. In the simulations a TAN algorithm, which is basically an EKF, is implemented and its performance is discussed for different terrain characteristics and for different terrain linearization methods.

Before explaining the simulation tools, a brief information about the MATLAB INS toolbox is given in this chapter. The Strapdown INS Module used in this study is developed by MATLAB INS toolbox which is becoming popular for navigation studies and simulations.

4.2 INS Toolbox

Inertial Navigation Toolbox is a third-party toolbox of the MATLAB which is developed by GPSoft [13]. Inertial Navigation System Toolbox allows one to simulate the vehicle dynamics, raw measurements (from the accelerometers and gyros) and also the data processing.

For this toolbox all functions and calculations are made with the assumption of Strapdown INS, which is appropriate for this study. The vehicle dynamics mentioned above are generated for a trajectory that can be planned using the INS toolbox. There are a few different methods that one can plan a flight trajectory using the properties of the INS toolbox from a start point to a stop point.

Using INS toolbox one can also model the error source emulator as gyro and accelerometer biases, scale factors and noise, initial position, velocity and tilt errors. The raw measurements of the accelerometers and gyros are generated by using the sensor output generator functions and noise models that are determined by the user. Also Direction Cosine Matrix or quaternion updating can be done using INS toolbox. The flight trajectory may be planned by using either the great circle path generator or profile/trajectory generator which are two different methods of planning a flight path.

In addition, Coordinate frame conversions can be done by using appropriate functions. The coordinate frames that can be worked with the INS toolbox are ECEF, ENU, earth-frame, body-frame, navigation-frame and local level-frame. Another important property of this toolbox is that the ellipsoidal and rotating Earth effects are included. So, one can model the error source emulators (sensor models), include the Earth ellipsoidal and rotating effects to model an INS that is operating as a realistic Strapdown INS. The other toolboxes of MATLAB like Kalman Filtering Toolbox or Mapping Toolbox (the interests of this study) are also compatible with INS toolbox.

Other navigation related toolboxes of the GPSoft are Satellite Navigation (SatNav) Toolbox and Navigation System Integration-Kalman Filter Toolbox which are widely used for navigation studies and simulations. These toolboxes are not used during this study. The details of INS toolbox will be more precise in the following section when we describe how we model a Strapdown INS using it.

4.3 INS Data Simulator

The INS Module is developed by using MATLAB INS Toolbox that we have mentioned in the previous section. To apply TAN algorithm and make simulations we need to develop an INS module with some specified errors rates. The properties of INS, error sources, and navigation equations are given in Chapter 2 in detail.

To design an INS by modeling all error sources is a challenging and time consuming work but is very informative to prevail all the details of an INS. Another advantage of designing an INS is that during simulations there is a chance to work with different qualities of INSs that we can model by changing error characteristics. In addition, by changing sampling rate, velocity and height we can see the performance variations of the system. Designing own INS simulator brings us the simulation flexibility with different conditions. Thus, for this study a Strapdown INS is designed from the beginning by using INS toolbox.

Error modelling is an important element of INS modelling. First we observed the solution of an errorless INS to have some idea about the total horizontal errors made by the system. Actual solution is obtained by modeling the error sources and adding these effects into errorless INS model. For our strapdown INS we need a flight path that we will generate INS data. The flight path is generated by flight path generator module that will be explained in the following section. We generate our INS data using the latitude and longitude profile, total distance and the vector of true course from each waypoint to the next (these are generated by flight path generator module).

The Next step is to create a series of accelerometer measurements (delta-V's) and a series of gyro measurements (delta-theta's). The functions *gendv* and *gendthet* are utilized to generate the accelerometer and gyroscope measurements, respectively. Once the measurements have been formed, the inertial navigation processing algorithms can be applied to the data created. When creating delta-V and delta-theta measurements we need to consider the Earth shape. INS toolbox has the flexibility to model Earth as flat, spherical or ellipsoidal. For this study the final form of INS module is designed by the assumption of ellipsoidal Earth.

To generate delta-theta measurements we first form the body frame to navigation

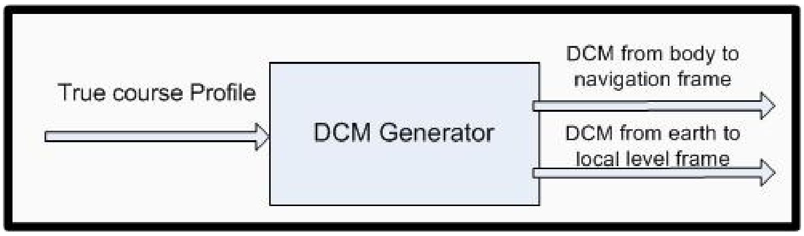

Figure 4.1: DCM Generator

frame DCM series using the vector of true course from each waypoint to the next. So body motion component of delta-theta is created by using the body frame to navigation frame DCM for each waypoint and delta-theta measurement generator function. Since we know the true course profile we can generate the local level to Earth frame DCM for each waypoint. Using local level to Earth frame DCM's we can generate the additive component of the delta-theta that is associated with the rotation of the Earth. Another component of the delta-theta associated with the craft rate is generated using true course profile, true height profile, appropriate DCM's and the velocity profile that is generated by flight path generator. For error-free INS we model sensor errors as zero, so Earth-rate and craft rate included delta-theta outputs are the ideal outputs. The block diagram given in Figure 4.1 is the DCM generator that is explained above.

For our INS model, platform is nominally level for the entire flight path that means that the initial roll and the initial pitch are "0". Wind is not modeled so true yaw becomes true course. It is always required that the desired parameters (attitude, velocity and position) must be initialized for inertial navigation. Although in the real world these parameters must be estimated, in this study we initialize them manually with the true data.

The accelerometer measurements (delta-V) are generated using the velocity profile associated with the flight path. The velocity profile is generated by using the true course vector and the total velocity profile that are generated by the flight path generator module. The component of the delta-V associated with the body motion relative to the Earth is generated by using *gendv* function, velocity profile and the body frame to navigation frame DCM's for each waypoint. The Coriolis effect is also considered

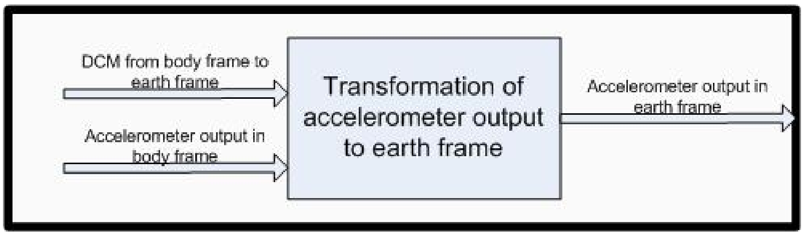

Figure 4.2: Reference Frame Transformation of the Accelerometer Outputs

when generating the delta-V measurements. By using *gendvcor* and appropriate inputs to this function the Coriolis components of the delta-V measurements are generated. Up to this point true course profile for desired inertial navigation system outputs are created, inertial navigation system is initialized and the delta-theta and delta-V measurements are generated. Now navigation computations can be done with these inputs to create navigation system outputs. The block diagram of the reference frame transformation is given in Figure 4.2.

The Gyroscope measurements allow the determination of the platform attitude and the orientation of the body frame relative to the navigation frame. The updating of the body frame to navigation frame DCM is performed by *bodupdat* function using DCM of the previous waypoint and the current delta-theta measurements. After updating the attitude of the platform, one can transform the accelerometer measurements to the navigation frame with the help of the body frame to navigation frame DCM's. The conversion from the navigation frame to the local level frame can be simply done by using the DCM from the navigation frame to the local level frame. So, the accelerometer measurements are transformed into the local level frame. By accumulating the accelerometer outputs the current velocity is calculated and also by integrating the velocity the position is updated.

The next step is to design a noisy (or realistic) INS by modelling the sensor errors. The noisy INS model includes the Coriolis components, Earth rate components and sensor error components. The effect of these components to the gyroscope and accelerometer measurements are also taken into consideration. To create accelerometer and gyroscope errors the functions *gendverr* and *gendtherr* are used respectively.

The following sensor errors are simulated:

- 40 micro-g x-accel bias
- 50 micro-g y-accel bias
- 0.01 deg/hr x-gyro bias
- 0.015 deg/hr y-gyro bias
- 0.009 deg/root-hour x-gyro white noise standard deviation
- 0.005 deg/root-hour y-gyro white noise standard deviation

By using these sensor errors, the final INS horizontal position error becomes 1 nautical-mile per hour which is a typical error for military aircraft and commercial airliners[8]. The horizontal position error is the displacement of INS output from the actual coordinates in x and y directions.

For a flight from Ankara to Istanbul with a constant velocity of 100 knots and constant height of 10000 meters the INS is simulated by using the sensor errors defined above. The horizontal position error for this simulation is given in Figure 4.3. Note that we have defined the horizontal error as the total error in the position excluding the vertical error position.

The latitude-longitude and the velocity errors are shown in Figure 4.4.

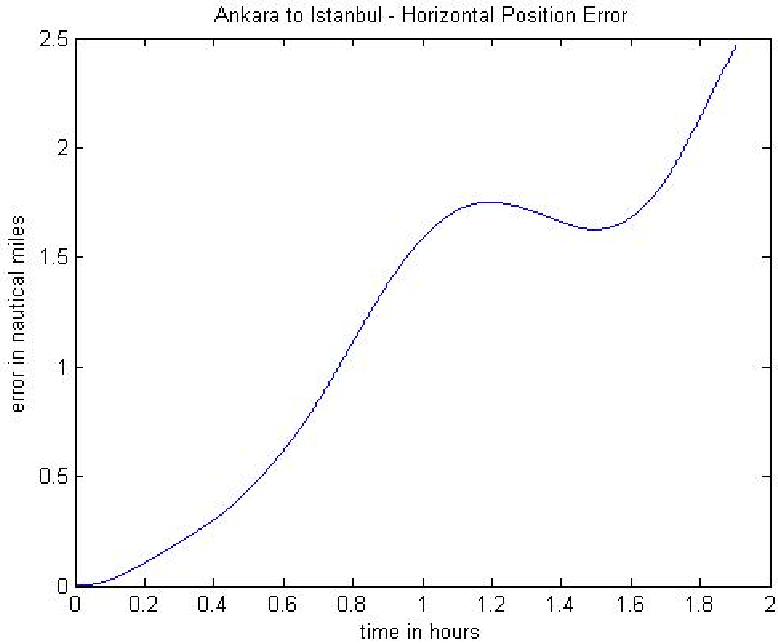

Figure 4.3: Horizontal Position Error

4.4 Flight Profile Generator

Flight Profile Generator is developed to create a flight path by using *greatcir* function which generates the great circle path between any two points on the Earth that are determined by the user. The great-circle distance or orthodromic distance is the shortest distance between any two points on the surface of a sphere [29].

To create a flight path one have to first define the latitude and the longitude of the start and the stop points. The velocity of the flight is constant but user can change the velocity for each separate simulation. Also the height is constant for each flight. The sampling rate is determined in terms of the distance between waypoints (sampling points). The flight profile sampling rate determines the sampling rate of the INS.

To sum up, Flight Profile Generator module generates true latitude-longitude profiles, vector of true course from each waypoint to next, total distance of flight, total time of flight, number of waypoints and the time profile tagged by each waypoint. Also one can plan any flight path with either the perfect sphere Earth or ellipsoidal Earth assumption. For this study ellipsoidal Earth assumption is used.

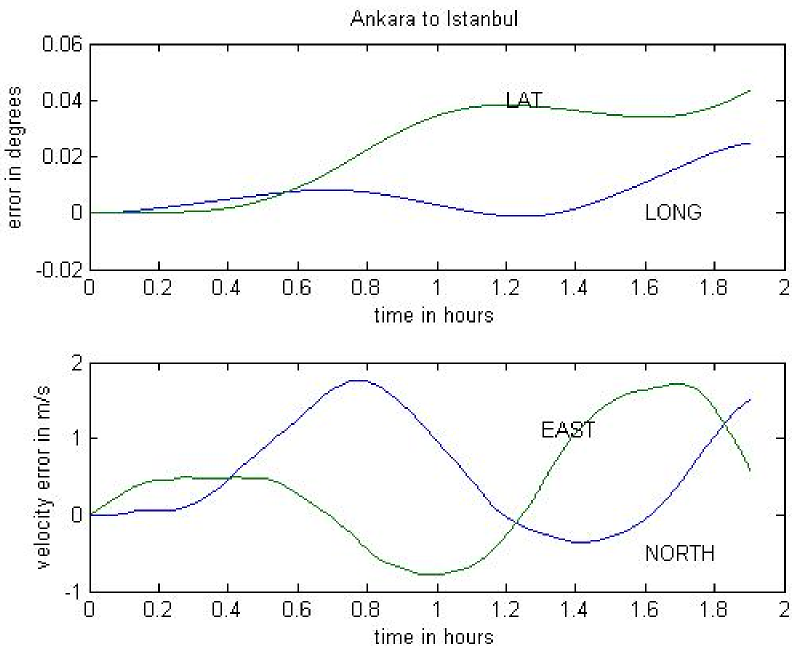

Figure 4.4: Ankara to İstanbul Errors

4.5 Terrain Module

In order to read DTED files for simulations, MATLAB Mapping Toolbox is used [10]. Binary DTED files are directly used as text files in the simulations with the help of the Mapping Toolbox. Using this module we can read the DTED files separately or we can read the directory file and load all the DTED to the workspace in a matrix form that contains the terrain height values in grid form.

The level of the DTED file is not an issue for this module but, in this study we use DTED Level 0 files which are free to public. For our purpose it is sufficient to load only the terrain that we plan to fly over, by using the start and stop latitude longitude form of DTED loading functional module. To check the terrain roughness and decide or predict the performance of the TAN algorithm for the terrain selected, a 2-D and 3-D map plots are also generated. These are given in Figure 4.5 and Figure 4.6 respectively.

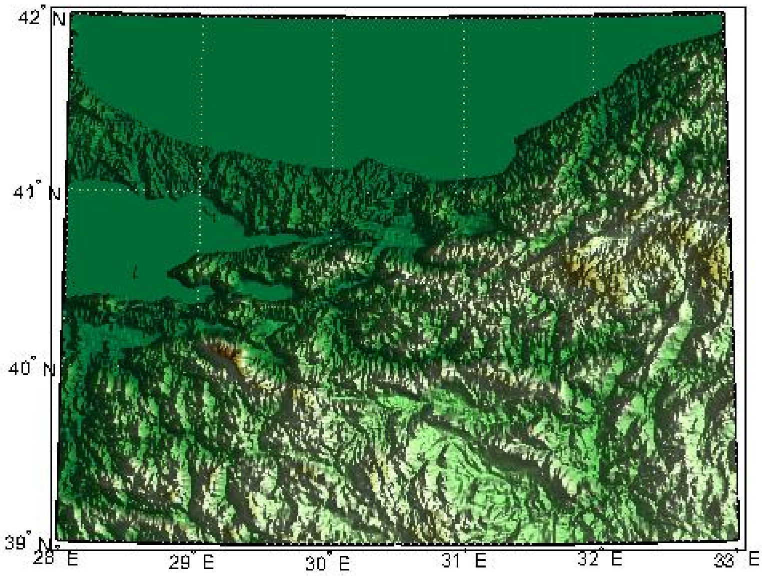

Figure 4.5: 2-D View of the map from Ankara to İstanbul

Barometric Altimeter To generate barometric altimeter outputs we use the constant height profile that is generated by the Flight Profile Generator. In other words, we assume that we know the exact mean sea level. So, this block of the simulation environment is a dummy block. It only takes the constant height that we planned to fly and produces the time tagged heights. The block diagram of the barometric altimeter measurement generator is shown in Figure 4.7.

Radar Altimeter To simulate the radar altimeter we need a radar altimeter generator. Radar altimeter model given in the work of Gray R. A. is used in this study [6]. This model is given below:

$$h_{RA}(generated) = h_{RA}(true) + sqrt(0.01^2) \times h_{RA}(true) + v_{RA}$$

where

$v_{RA} \sim$ N(0,50)

4.6.

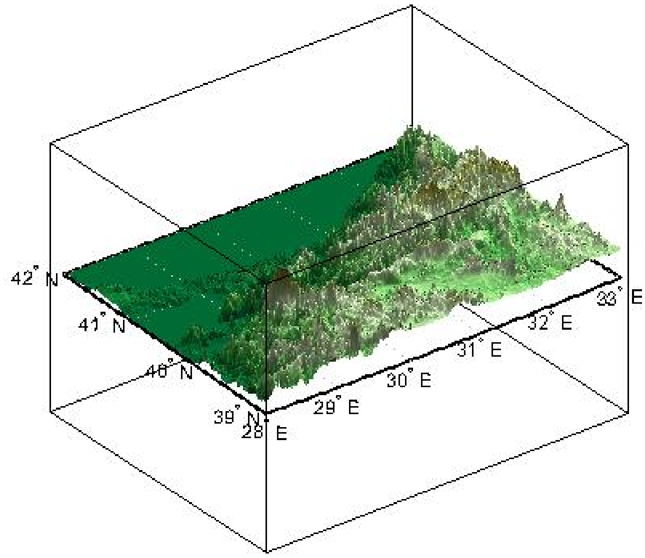

Figure 4.6: 3-D View of the map that we plan to fly over

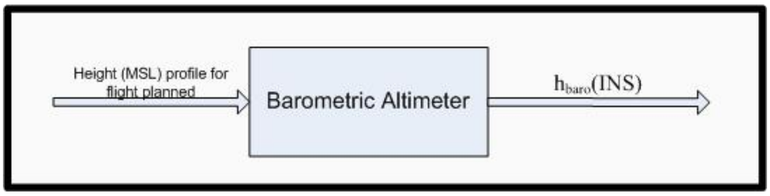

Figure 4.7: Barometric Altimeter Simulator

To obtain the radar altimeter outputs we use the planned latitude and longitude profiles that is generated by Flight Profile Generator. This means that the terrain height is obtained using the true latitude-longitude values and the DTED. In other words when generating the radar altimeter outputs the true latitude longitude profile is obtained from Flight Path Generator, then using the true terrain height from DTED the radar altimeter measurements are generated by using the modeled noise characteristics.

The block diagram of the radar altimeter measurement generator is shown in Figure 4.8.

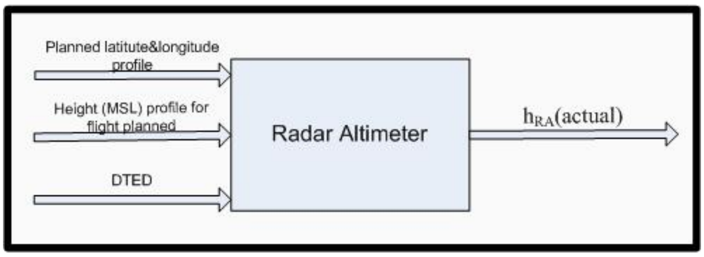

Figure 4.8: Radar Altimeter Simulator

Figure 4.9 shows the output of the radar altimeter simulator for flight from Ankara to İstanbul.

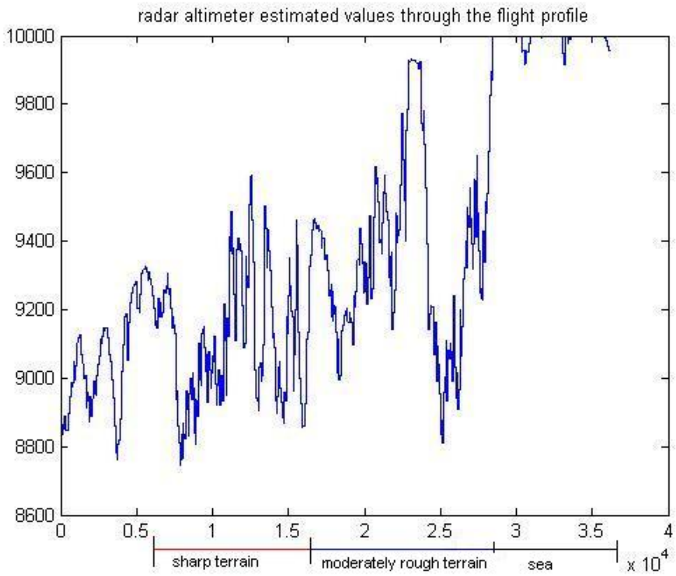

Figure 4.9: Radar Altimeter Simulator Output from Ankara to İstanbul

Terrain Height Generator We need to generate the terrain profile (or map profile) to apply and simulate the TAN algorithms. The INS data generated by INS module, planned true profile and the DTED is used to generate actual map profile (heights of the waypoints we plan to fly over) and the INS estimated map profile (heights of the waypoints that are generated by INS module). The block diagram of the terrain height generator is shown in Figure 4.10

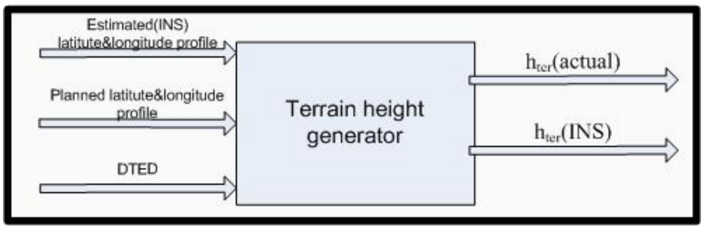

Figure 4.10: Terrain Height Generator

Measurement Generator By combining the radar altimeter, barometric altimeter and terrain height generator outputs we generate our measurement δh. The block diagram of the measurement generator is shown in Figure 4.11

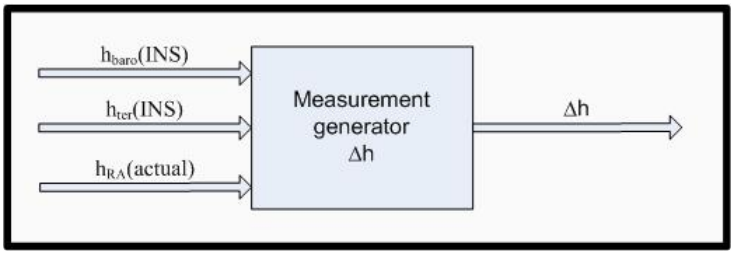

Figure 4.11: Measurement Generator

4.6 Kalman Filter Module

In this chapter the implementation of the Kalman Filter Module for TAN solution is explained in detail. The Kalman Filter and the TAN algorithms are studied in the previous chapters, so implementation is the main issue of this section.

Terrain Aided Navigation (TAN) algorithms estimate the position of a platform by comparing the measured terrain elevation profile under the platform to a stored elevation map. Therefore, the critical part of the TAN is the DTED. Batch and recursive algorithms are used for TAN as explained before. For batch algorithm, i.e. TERCOM, only DTED is used in order to estimate the position of the vehicle. On the other hand, for recursive algorithms, like SITAN, besides DTED, the system dynamics should also be modeled. For TAN Kalman Filter solutions we also need to model the system dynamics. The details of the Kalman Filter state equations and the model of system dynamics are explained in Chapter 2.

The Kalman Filter Module used for this thesis is developed by using MATLAB. Functional outline of the implementation of the Kalman Filter Module is given below.

1. Define constants and variables
    - Earth angular velocity
    - local level to navigation frame DCM
    - sampling rate
    - iteration number
2. Set initial values of the state variables
3. Generate body to Earth frame DCM using the estimated body to navigation and the estimated Earth to local level frame DCM's that are obtained from the INS Module
4. Generate accelerometer outputs using the DCM from body to Earth and the gyro outputs that are given by the INS Module (C_b^n)
5. Generate radar altimeter output using the true latitude-longitude profiles and the true terrain height values for that flight profile
6. Generate measurement vector δh using the radar altimeter, barometric altimeter and terrain height profiles
7. Set process and measurement noise covariance matrices
8. At iteration index T=0 initialize the mean and the covariance matrix of the state
9. At iteration index T=1 compute the state transition matrix **F** using the generated DCM profiles and accelerometer outputs
10. Generate the terrain slopes to develop the measurement matrix **H**
11. Calculate Kalman Gain matrix
12. Update the estimated state variables using the innovation

13. Update the estimated state covariance matrix
14. Calculate the predicted state variables and state covariance matrix
15. Go to the step 9 with iteration index T=T+1
16. Run the above cycle for all profile

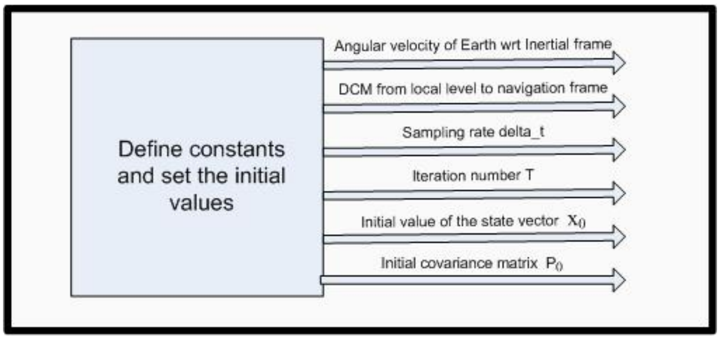

Figure 4.12: Defining Constant and Initializing the System

The Figure 4.12 shows the constants and initial values that one need to determine before any simulation.

The Kalman Filter structure is seen in Figure 4.13 as a block diagram.

The aim of these algorithms is to correct the path that is generated by the INS by using the terrain elevation data as measurement. For TAN algorithms, the only measurement is the height difference δh defined in Equation 2.84 of the previous chapter. When generating the measurement δh we combine the information given by the inertial navigation system, radar altimeter and the map. The details of generating δh is described in Chapter 2, in measurement model of the Kalman Filter section.

4.7 Simulation specifications

In this section we give the two examples of terrain aided navigation corresponding to the two different flight paths. The INS Model used in the simulations has a typical 1nm/hour horizontal position error that is explained in INS Module section in detail.

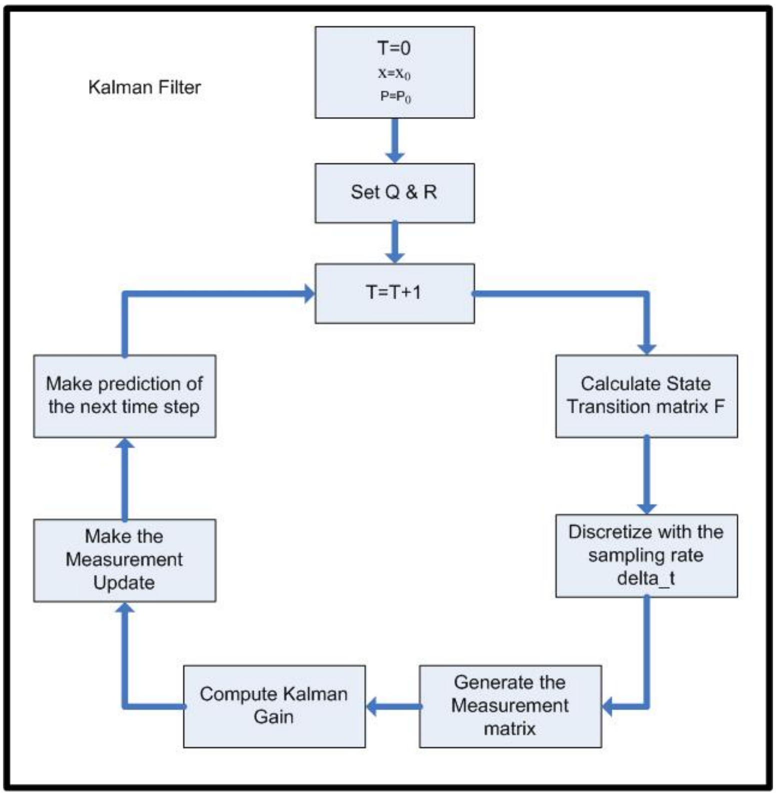

Figure 4.13: Kalman Filter

The flight profile planned for these simulations are from Ankara to İstanbul and from Eskişehir to Karaman. These paths are selected to observe the performances of the algorithms for different terrain characteristics. These flight paths include flat terrain, rough terrain and some region on the sea. Besides, these flight scenarios are realistic scenarios. The flight scenarios have constant velocity of 100 knots (51.4 m/sec) and a constant height of 10000 meters (MSL) flight. The sampling rate of the INS data is also a simulation parameter that we change.

All simulations are done with DTED Level 0. Since the distance between each grid of the DTED Level 0 is approximately 1 km, we need to define a meaningful measurement update interval considering the DTED resolution. In other words, independent of the measurement and process sampling rates a measurement update interval should be decided. For this study, the measurement update interval is optimized as 1 nautical mile (this is given in terms of distance because the sampling rate of the INS model is also implemented in terms of distance). The simulations are done with a measurement update interval of 1 nautical mile. This means when we approximately pass through a DTED grid, we use our measurement and make an update for our state variables.

4.7.1 Simulations for Terrain Aided Navigation

In the simulations we have used two versions of Linearized Kalman Filter (LKF) and two versions of Extended Kalman Filter (EKF). In the following sections we we will give the results of these filters.

In the simulations distances are in meters and the angles are measured in radians. And the state variables, measurement and process noises are:

$$\mathbf{x}_0 \sim N(\mathbf{x}_0, \mathbf{P}_0)$$

$$\mathbf{w}_k \sim N(0, \mathbf{Q})$$

$$\mathbf{v}_k \sim N(0, \mathbf{R})$$

where

$\mathbf{x}_0$: initial state estimate

$\mathbf{P}_0$: initial state covariance matrix

$\mathbf{Q}$: process noise covariance matrix

$\mathbf{R}$: observation noise covariance matrix

$\mathbf{Q}$ matrix shows the uncertainty in the error state. In this study it is selected as a diagonal matrix assuming that the components of the error state are independent. The diagonal elements are functions of sampling time, i.e., larger the time difference between two updates the larger variance. They are selected in a rather intuitive form by considering the magnitudes of the related quantities. As an example the variance of δR_x and δR_y are selected equal to each other while the variance of attitudes are much smaller than the variances of δR_x and δR_y.

4.7.1.1 Linearized Kalman Filter

LKF is another way of using KF for nonlinear systems. The structure of LKF will be briefly explained here. Assume that a discrete time nonlinear system represented in the state space form as:

$$x_{k+1} = f(x_k) + w_k$$

$$z_k = h(x_k) + v_k$$

Let x_k^+ be the estimate of the state at time k after the measurement z_k. To obtain the time update the system is linearized around x_k^+ and the standard time update and measurement update procedure is applied. The standard procedure is explained in Chapter 2 in detail.

For this approach both mean and the covariance matrices are updated according to the linearization of the nonlinear system at each measurement time.

The LKF TAN algorithm simulations applying the linearization with measurement update rate is performed with the assumptions given in Table 4.1.

Table 4.1: LKF (with measurement update interval) Simulation Settings

$\mathbf{x}_0$	zeros(15,1)
$\mathbf{P}_0$	diag($\mathbf{p}$)
$\mathbf{Q}$	diag($\mathbf{v}$)
$\mathbf{R}$	$500m^2$

where

$\mathbf{v}$=[0.1 0.1 0.01 10^{-3} 10^{-3} 10^{-5} 10^{-5} 10^{-5} 10^{-5} 10^{-6} 10^{-6} 10^{-6} 10^{-6} 10^{-6} $10^{-6}]^T$

$\mathbf{p}$=$10_{-3}\times$[0.1 0.1 0.01 10^{-3} 10^{-3} 10^{-5} 10^{-5} 10^{-5} 10^{-5} 10^{-6} 10^{-6} 10^{-6} 10^{-6} 10^{-6} $10^{-6}]^T$

The measurement update is done with intervals of 1 nautical mile, that corresponds to 36 seconds. This interval is not small enough to assume that the INS is linear. However for the sake of completeness, to see the divergence of filter the simulations are done. As expected the filter diverges and meaningless results are obtained. This shows that assuming the INS as linear between 36 seconds for a flight of constant velocity of 100 knots is not a reasonable assumption.

LKF TAN algorithm with time update of intervals of 1/190 nautical mile, i.e., 0.19 seconds is the second application. This corresponds to a flight speed of 100 knots. Between measurement updates the time updates are also applied for no measurement update intervals. The simulation result is shown in Figure 4.14.

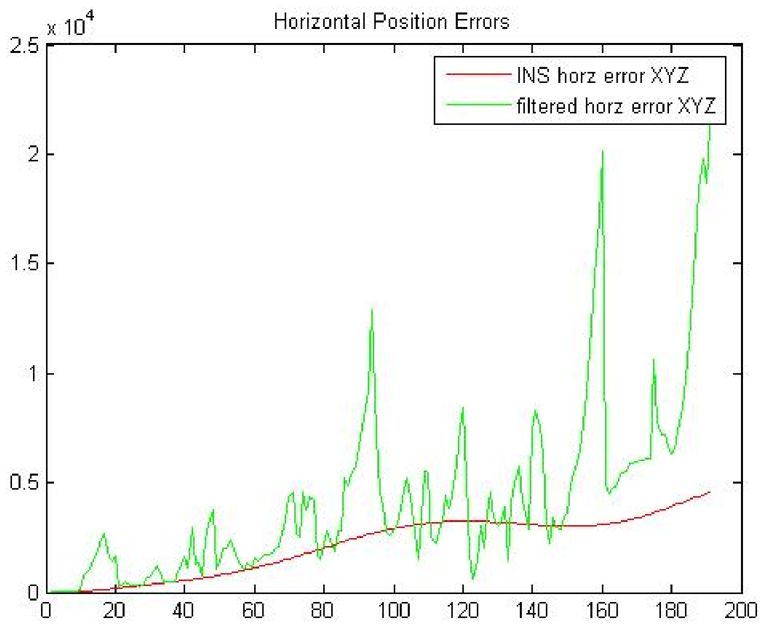

Figure 4.14: Horizontal Position Errors of INS and LKF Solution (updated with time update intervals)

The result shows that the linear assumption of INS for such intervals is reasonable. However because of the high nonlinear characteristics of the INS the filter cannot give the optimal solution. Figure 4.14 clearly indicates an increase in the total error compared to non-aided INS. As stated, this is the expected result. Therefore, LKF approaches does not improve the INS.

4.8 Extended Kalman Filter

As expressed before for most of the practical systems the state and the observation models may be nonlinear. For such cases Kalman filter is adapted to cope with nonlinearity where the resultant filter is called Extended Kalman Filter (EKF). EKF is based on the linearization of the nonlinear equations of the state and the observation models about the current state estimate. The theory of the EKF is explained in Chapter 2 in detail.

In the standard EKF the mean is updated according to the nonlinearity $\mathbf{x}_{k+1} = f_k(\mathbf{x}_k)$, but to update the covariance matrix, a linearization is done at time k. In our application there is no explicit function $f(\cdot)$, but the same effect is obtained by using (unaided) INS solutions on the time interval of two consecutive DTED grids.

The standard procedure for covariance matrix time update is to use the linearization done at the beginning of the interval (t_k, t_{k+1}), i.e., $F_k = \left.\frac{\partial f}{\partial x}\right|_{x_k}$. In this thesis, when implementing the EKF TAN algorithm different linearization approaches are used. Since our observation model is linear, linearization is only applied to the state model.

The EKF TAN algorithm simulations uses the parameters given in Table 4.2.

Table 4.2: EKF Nonlinear Solution Simulation Settings

$\mathbf{x}_0$	zeros(15,1)
$\mathbf{P}_0$	diag($\mathbf{p}$)
$\mathbf{Q}$	diag($\mathbf{v}$)
T_{INS}	0.189 seconds
T_{meas}	36 seconds

where
T_{INS} : INS sampling rate
T_{meas} : Measurement sampling rate

Figure 4.15 shows the horizontal position errors at each iteration instant. We have defined the horizontal error as the total error made in the horizontal plane. The red line shows the horizontal error of the INS in XYZ coordinate frame (the coordinate frame is not an issue because the horizontal error does not change with coordinate frames). The green line shows the filtered (or corrected) INS error.

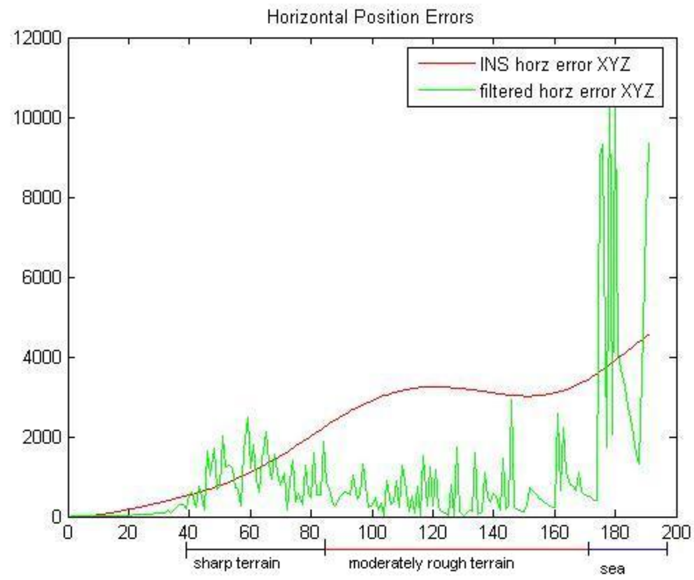

Figure 4.15: Horizontal Position Errors of INS and Filtered solution-1

It is seen from the figure that the behavior of the filter changes along the flight path. After time index 160 (approximately) there is a drastic change in the performance of the filter. This is the result that we expected because when we look at the flight profile and the terrain characteristics, after nearly 160^{th} waypoint the flight is carried out over the sea. Thus, external navigation aid becomes a bad sensor over flat terrain (sea), as a result of the bad external aid the filter can not correct the INS.

To observe the performance of the filter for different terrain characteristics, the total INS error and the total error using terrain aid are compared for different intervals.

Table 4.3 shows that the total errors made by INS and aided INS for the total flight path from Ankara to İstanbul and also for the flight path excluded the over sea flight region, i.e., after the 160^{th} point. As can be seen the total error is reduced to almost 1/3 of the INS error when flying over a rough terrain region.

In the EKF TAN structure the covariance matrix update is done with the measurement update frequency. Note that the selection of the process noise covariance matrix, i.e.,

Table 4.3: Horizontal Errors

	Total Error	Up to the 160 th point
INS horizontal total error	4.090331e+5 meters	2.415239e+5 meters
Filtered horizontal total error	1.866052e+5 meters	7.773223e+4 meters

Q should be done according to the length of the time update interval. This is because of the fact that impact of the unmodelled errors are small for small time intervals. So the **Q** matrix is selected after some tunning process.

The second version of the EKF is based on the idea of using nonlinear nature of INS also for update covariance, i.e., the time update. Note that time update formula of covariance matrix is

$$\mathbf{P}_{k+1} = \mathbf{F}_k \mathbf{P}_k \mathbf{F}_k^T + \mathbf{Q}_k$$

where $\mathbf{F}_k$ is obtained as a linearization of nonlinear function $f(\cdot)$ around the estimate. It may be reasonable to change F along the path from one measurement update to the second one by using INS outputs.

To give an idea about the effect of process covariance matrix, some simulations are done for different **Q** matrices. To interpret the effect of this issue an example is given below. When the simulation is done with the process noise assumption given in Table 4.4 the result given in Figure 4.16 is observed. For this simulation, the total error that is corrected by this filter and the error over moderate terrain (up to the sea) are given in Table 4.5. Results show that there is no improvement for error characteristics of INS, moreover the error is increased when the INS data is filtered.

Table 4.4: Process Noise

Q	$\mathrm{diag}(\mathbf{v}) * 10^{-8}$

The observation noise covariance matrix is another crucial parameter of filter. So simulations are done to observe the effect of observation noise covariance matrix for

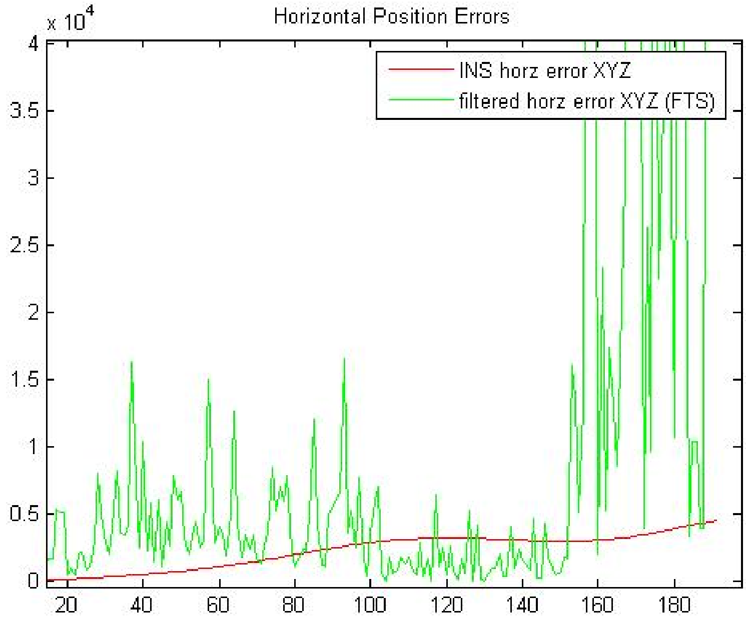

Figure 4.16: Horizontal Position Errors of INS and Filtered solution-3

Table 4.5: Horizontal Errors

	Total Error	Up to the 160 th point
INS horizontal total error	4.090331e+5 meters	2.415239e+5 meters
Filtered horizontal total error	8.346210e+5 meters	2.887589e+5 meters

different terrain types with different simulations including some extreme assumptions. The expectation is to observe an increase in total error because of the flat terrain region, on the sea, when the observation noise is decreased. With small observation noise covariance matrix, over flat region filter relies on the aided data more, thus at the filter output increasing error is expected to observe. The simulations are performed with the assumptions given in Table 4.6.

The result of the simulation is given in Figure 4.17.

To see the effect of variation on **R**, an extreme example is simulated with **R**=2. The result of the simulation is given in Figure 4.18.

Table 4.6: EKF Nonlinear Solution Simulation Settings-2

$\mathbf{x}_0$	zeros(15,1)
$\mathbf{P}_0$	diag($\mathbf{p}$)
$\mathbf{Q}$	diag($\mathbf{v}$)
$\mathbf{R}$	$100m^2$

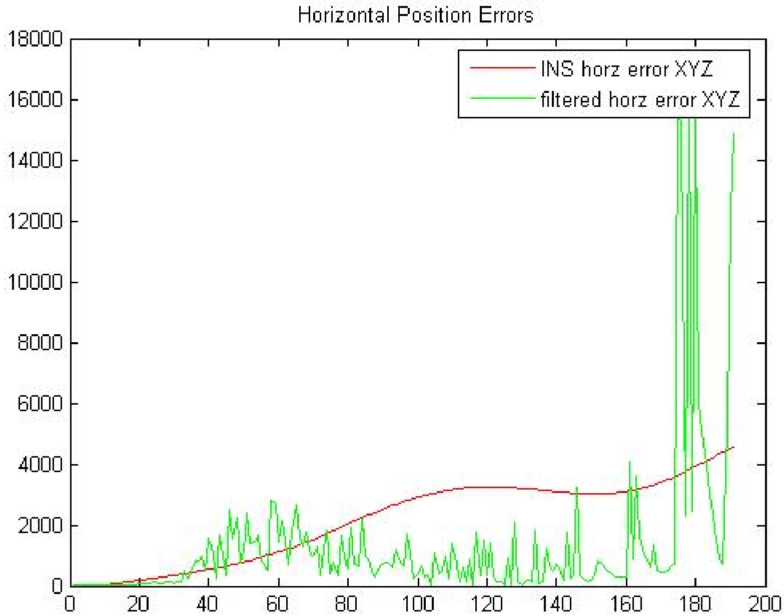

Figure 4.17: Horizontal Position Errors for **R**=100

Total horizontal position errors observed with **R**=100 and **R**=2 are given in Table 4.7. The horizontal position errors over moderate terrain (up to the 160 th measurement point) are also given in this table.

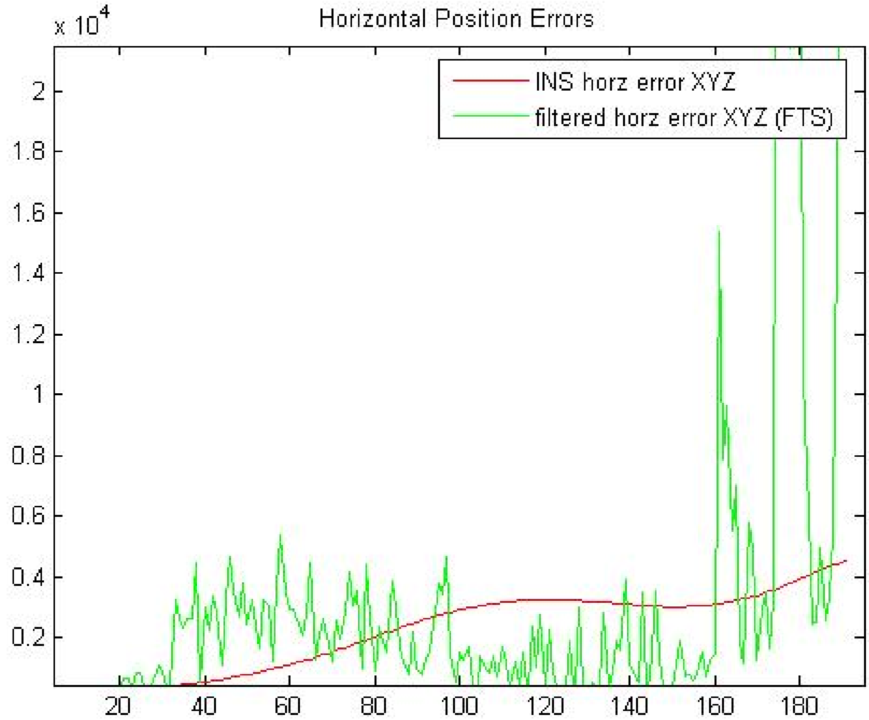

Figure 4.18: Horizontal Position Errors for **R**=2

Table 4.7: Horizontal Position Errors

	R=100		**R**=2	
INS error	4.090331e+5	2.415239e+5	4.090331e+5	2.962729e+5
Filtered error	4.009735e+5	1.511438e+5	5.112878e+5	1.987078e+5

4.8.1 Adaptive R and Q Approach

Since the performance of the filter changes with aid quality and terrain type, an adaptive approach is reasonable. We are applying this algorithm for a planned flight path, hence with the knowledge of terrain characteristics we can set appropriate process and the measurement noise covariance matrices. Using the simulation results of the previous section, the appropriate **R** and **Q** settings are selected as given in Table 4.8.

Table 4.8: Adaptive EKF TAN Filter Settings

Region	0¡index¡41	40¡index¡81	80¡index¡161	160¡index¡191
Q	diag(**v**)	diag(**v**)	diag(**v**)	diag(**v**)
R	500	10000	100*	3000

Figure 4.19 shows the adaptive **R** and **Q** EKF solution results.

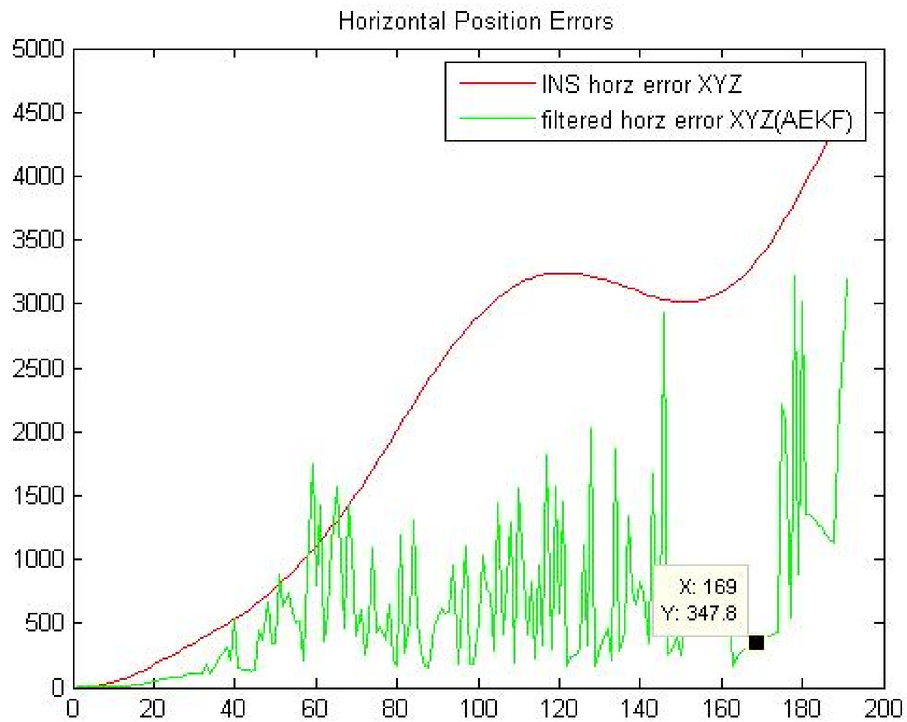

Figure 4.19: Horizontal Position Errors for Adaptive EKF

The total horizontal position error is given in Table 4.9.

Table 4.9: Horizontal Position Errors for Adaptive EKF

INS horizontal total error	4.090331e+005 meters
Filtered horizontal total error	2.348888e+005 meters

Hence for a planned flight profile using adaptive EKF protects error characteristics from bad external aids.

4.8.2 Different Terrain Stochastic Linearization Approaches

For TAN type of algorithms terrain linearization is a crucial issue that is studied by many researchers. The simulation done up to this section use First Order Taylor Series (FTS) terrain stochastic linearization technique. In FTS technique, the values of the elevation data and the slopes are calculated as:

$$a = h(m,n) \tag{4.1}$$

$$h_x = \frac{h(m+1,n) - h(m-1,n)}{2d} \tag{4.2}$$

$$h_y = \frac{h(m,n+1) - h(m,n-1)}{2d} \tag{4.3}$$

where d is the DTED grid spacing or the DTED resolution and (m,n) denotes the position on DTED.

Figure 4.20 shows the ground patch geometry used for this technique [20].

Since we use DTED Level 0 and the resolution of the DTED Level 0 is 1 km, the FTS technique is reasonable when we think about the linear assumed region. To see the performance of the EKF TAN solution with different terrain stochastic linearization methods, Mean Tangent Line (MTL)and the Nine Point Planar Fit (NPPF) methods are implemented. The theory of these methods and simulation results are given in the following sections.

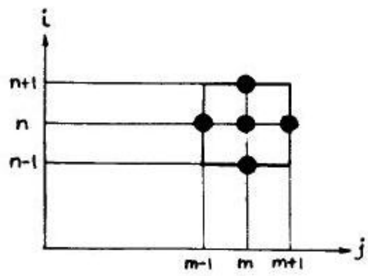

Figure 4.20: Ground Patch Geometry for FTS Technique (taken from [20])

Mean Tangent Line (MTL) The MTL technique is an improvement of the FTS technique. In MTL technique, the patch area is determined by using the standard deviation of position errors, σ_x and σ_y.The grid points that are in the selected area one used in the slope computations [20]. For MTL technique patch area is selected as $5\sigma_x \times 5\sigma_y$ in size and centered at the point (m,n). In our work the instantaneous horizontal errors are small compared with the DTED Level 0 resolution, so the method is not applicable since it give M=N=1. However for the sake of completeness we will explain the method and give a result for our terrain for M=N=2, where M and N are the horizontal and vertical data grid points used for linearizations in the x and y directions respectively.

The geometry of MTL is shown in Figure 4.21

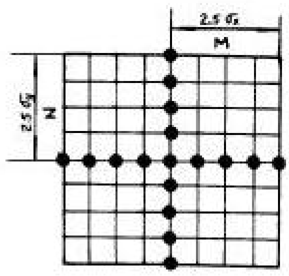

Figure 4.21: Ground Patch Geometry for MTL Technique (taken from [20])

Let (m,n) be a position on the DTED. The values of the elevation data and the slopes are calculated as:

$$a = h(m, n) \tag{4.4}$$

$$h_x = \frac{1}{2Md} \sum_{k=-M}^{M} \frac{h(m+k, n) - h(m, n)}{|k|} \tag{4.5}$$

$$h_y = \frac{1}{2Nd} \sum_{k=-N}^{N} \frac{h(m, n+k) - h(m, n)}{|k|} \tag{4.6}$$

where $k \neq 0$.

The simulation result is given in Figure 4.22.

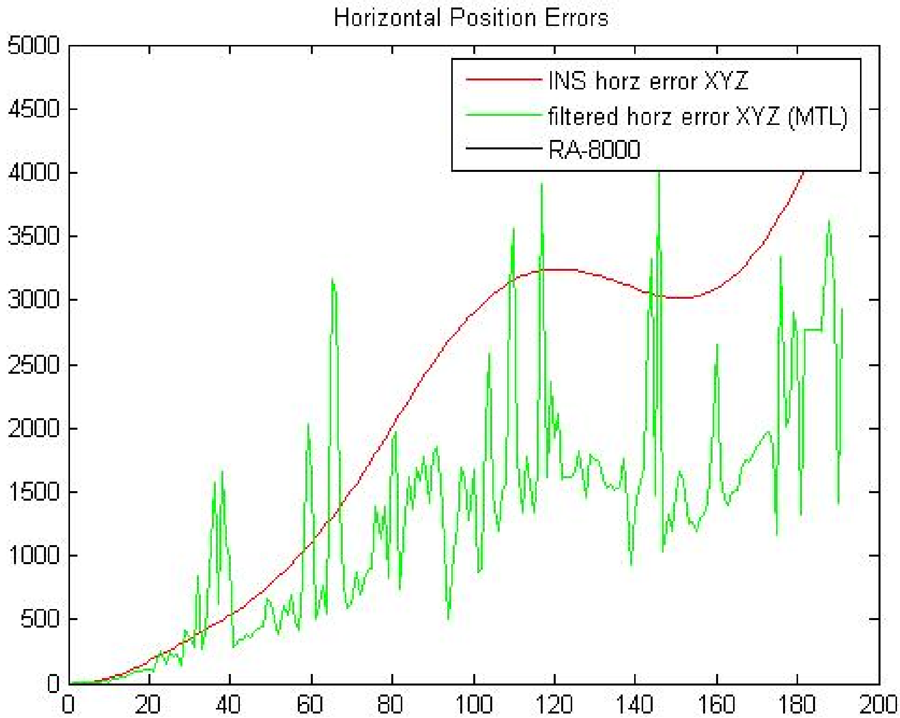

Figure 4.22: Horizontal Position Errors for MTL Technique

The total horizontal position error is given in Table 4.10.

The MTL Technique improve the solution for bad external data regions like sea. However the error increases at sharp regions because of our 4 km linearization assumption. Since we utilize a region (4 km X 4 km in x and y directions) to calculate the terrain

Table 4.10: Horizontal Position Errors for MTL Technique

INS horizontal total error	4.090331e+005 meters
Filtered horizontal total error	2.490790e+005 meters

slopes, the sharp terrain characteristics are linearized means the slope of that point is not accurately represented. As a result of this issue at these points MTL response is worse than the FTS technique which is expected. In addition, the filter total response is also worse than the FTS technique. This claim is also supported by the filter response given in Figure 4.22.

Nine Point Planar Fit (NPPF) The NPPF technique uses a rectangular patch centered at (m,n). The geometry of the method is given in Figure 4.23. Similar to the MTL technique the patch is approximated by using the standard deviations of horizontal position error and decided as $3\sigma_x \times 3\sigma_y$ [20].

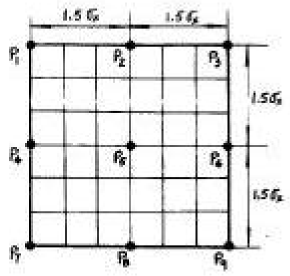

Figure 4.23: Ground Patch Geometry for NPPF Technique (taken from [20])

Let (m,n) be a position on the DTED and the h_l, represents the height value of the point P_l. The values of the elevation data and the slopes are calculated as:

$$a = \frac{1}{9}(h_1 + h_2 + \cdots + h_9) \tag{4.7}$$

$$h_x = \frac{1}{6d}(h_3 + h_6 + h_9 - h_1 - h_4 - h_7) \tag{4.8}$$

$$h_y = \frac{1}{6d}(h_1 + h_2 + h_3 - h_7 - h_8 - h_9) \tag{4.9}$$

The simulation result is given in Figure 4.24.

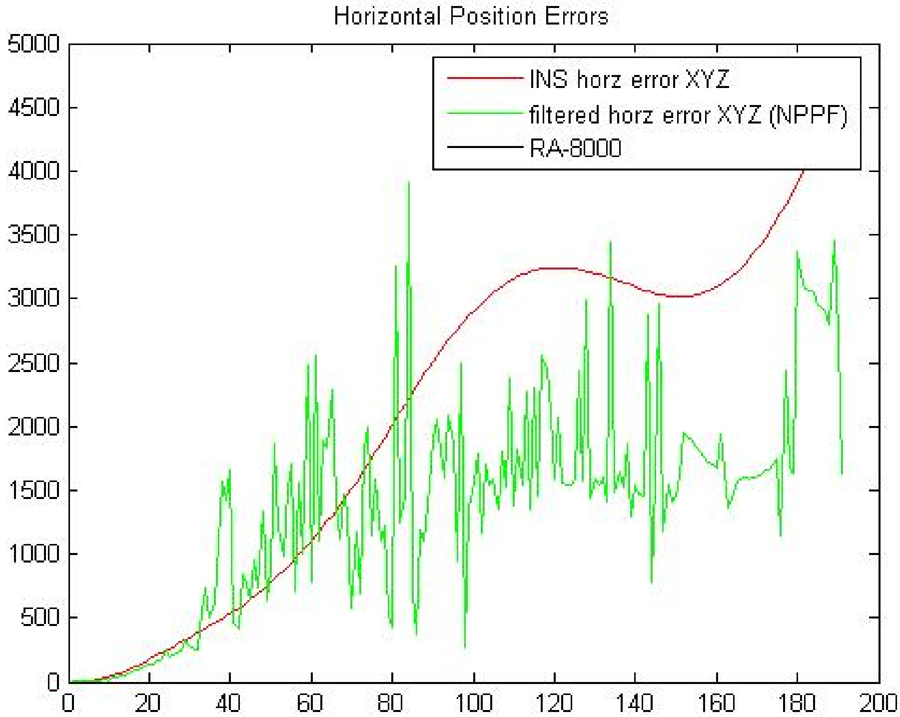

Figure 4.24: Horizontal Position Errors for NPPF Technique

The total horizontal position error is given in Table 4.11.

Table 4.11: Horizontal Position Errors for NPPF Technique

INS horizontal total error	4.090331e+005 meters
Filtered horizontal total error	2.108213e+005 meters

The NPPF Technique is also improved the solution for bad external data region. However, similar to the MTL technique, the error increases at the sharp regions. Since 2 km X 2 km region is used to calculate the terrain slopes, the sharp terrain characteristics are linearized and at these points the calculated terrain slopes are not sensible when compared with the actual slopes. Hence at these points our filter response is worse than the FTS technique. Besides, total response of the filter is also worse than the FTS technique. This claim is also supported by the filter response given in Figure 4.24. But when we compare the MTL and the NPPF techniques for this application, the MTL technique is better for moderate terrain, but it is not good for sharp terrain regions. The NPPF technique is better than MTL technique when compared to the sharp terrain regions, but its performance is similar but worse than MTL for moderate terrain regions. These terrain linearization techniques are expected to improve INS more for an application of higher terrain elevation data resolution but for this application, as stated before, it is not the case.

The main problem that is caused by low resolution data is the poor performance of the filter with TSL techniques such as MTL and NPPF. The problem of instantaneous peak errors is expected to be handled when a high resolution data is used. To observe the filter performance with a high resolution data, some simulations are done using nearly 150 meters resolution elevation data. The result is given in Figure 4.25.

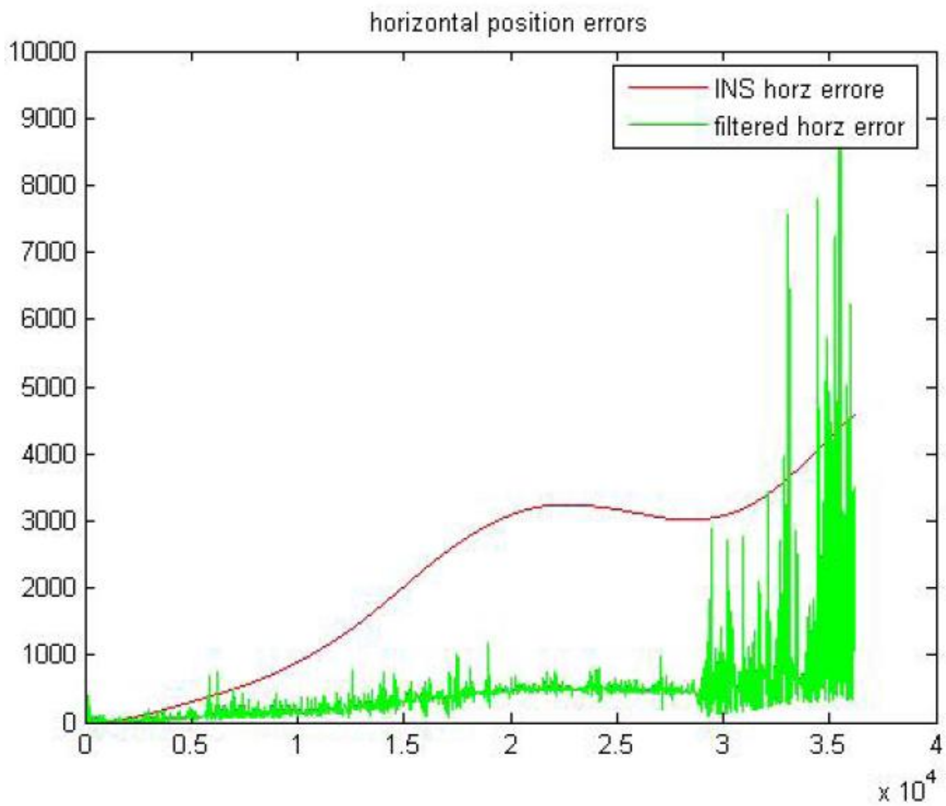

Figure 4.25: Horizontal Position Errors for High Resolution Map

It is clear from the figure that a high resolution data supplies more accurate terrain information. So, with more correct slopes the filter performance is improved as expected. The instantaneous peak errors which are the main problem of our TAN solutions still occur, but it is clear that they are much smaller. The mean error is also decreased with increasing map resolution with NPPF TSL technique. The simulation tools developed for this thesis can be modified and used for simulation with other different elevation data formats such as SRTM. The main problem of SRTM elevation map is no-data areas, because of this reason many researchers have been studying on this issue. To develop the free and void-filled SRTM data for Turkey a Tubitak supported project is carrying on [16]. SRTM and ASTER free data sets will be more reliable and widely used by many researchers in coming years.

4.8.3 Simulations with Different Flight Path

To see the EKF TAN performance for over a different terrain a different flight profile is generated using the simulation tools developed and INS module is simulated with the same error characteristics. The path is selected over a moderate terrain. The start and stop latitude and longitudes are [40 31] and [37 34] respectively, in other words from Eskişehir to Karaman. Figure 4.26 and 4.27 show the 2-D and 3-D terrain characteristics of this flight region.

The simulation results of EKF TAN of this flight path is given in Figure 4.28.

Table 4.12: Horizontal Position Errors

	Horizontal Position Errors
INS error	5.951874e+5 meters
Filtered error	2.530127e+5 meters

Figure 4.28 consists of radar altimeter output as well as INS and aided INS errors. The performance of the EKF TAN algorithm is a function of the terrain characteristics that is also observed in the result of this simulation. In general, for all TAN algorithms it

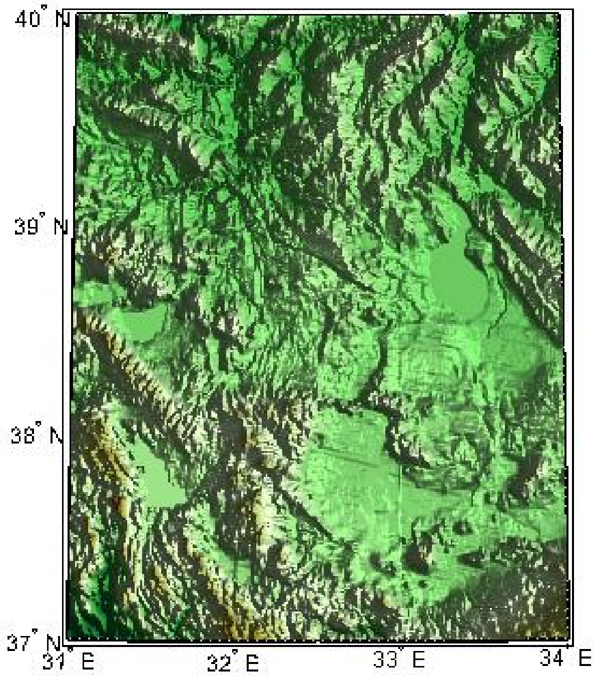

Figure 4.26: Terrain profile 2-D

is a well know fact that a sharp terrain characteristic, as canyons, is a main concern that causes peak errors. To emphasize this relation between terrain and filter output three different sharp regions are zoomed and discussed in detail.

Figure 4.29 shows the zoom area-1. There is a canyon at this region. The slope of the terrain is computed by using the points X=54 and X=56 which is certainly not representing the actual slope. As a result of this highly nonlinear characteristic of the terrain, at the filter output we observed an instantaneous peak error. However, after this region the terrain slopes become informable about the terrain so the output of the filter shows an improvement in the error of aided INS.

Figure 4.30 shows the zoom area-2. At this region the terrain has a hill. Hence, the slope of the terrain is computed by using the points X=115 and X=117 which is certainly not representing the actual slope. At the output of the filter we also observe a peak error.

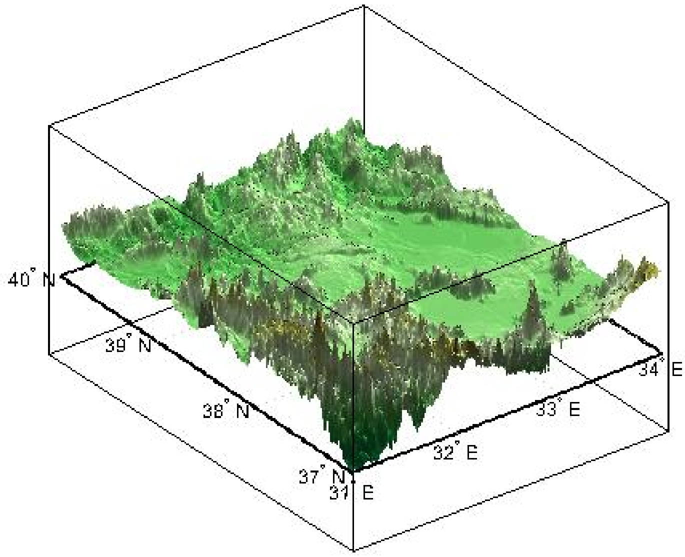

Figure 4.27: Terrain profile 3-D

Figure 4.31 shows the zoom area-3. At this region the terrain has a high hill. As a result, the terrain slopes are very wrong although we use the same uncertainty. As expected, at the output of the filter a huge peak error is observed.

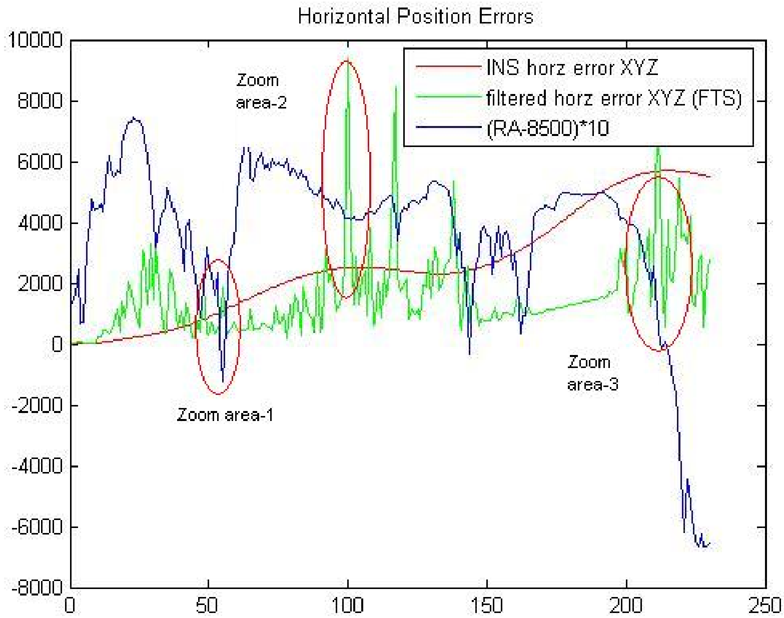

Figure 4.28: Horizontal Position Errors

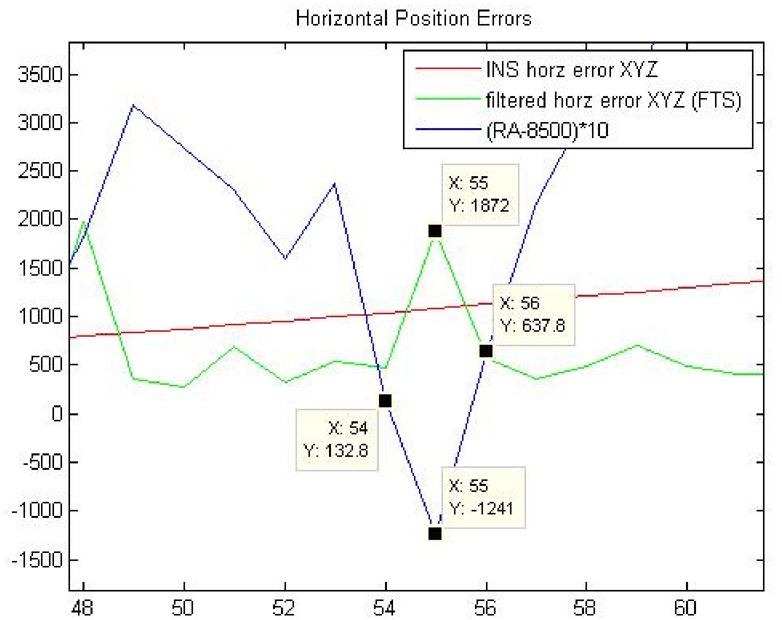

Figure 4.29: Zoom Area-1

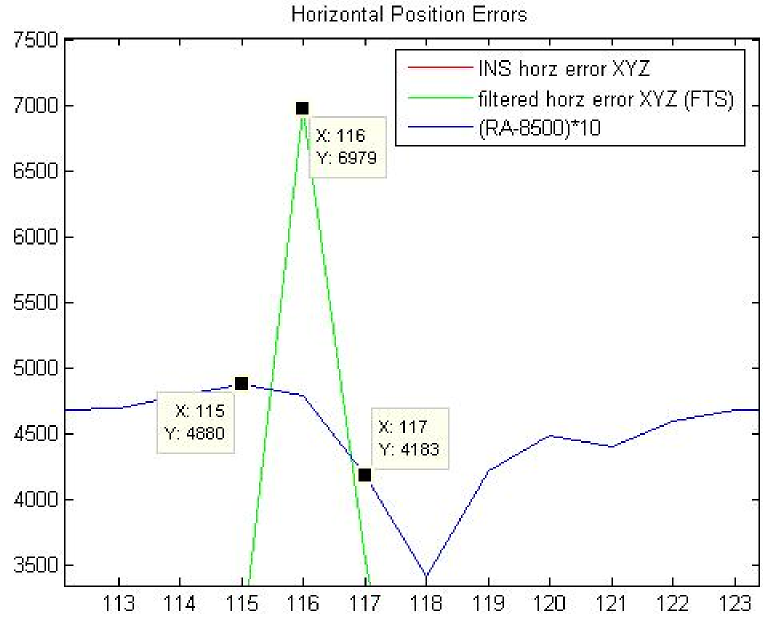

Figure 4.30: Zoom Area-2

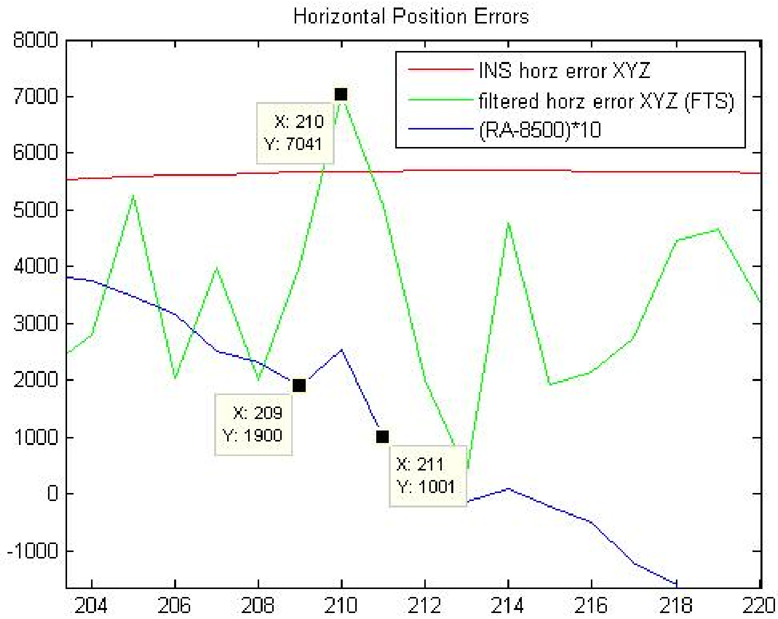

Figure 4.31: Zoom Area-3

CHAPTER 5

DISCUSSIONS CONCLUSIONS AND FUTURE WORK

In this study, first of all a wide and detailed research is done about inertial navigation systems and Terrain Aided Navigation (TAN) algorithms. The equations of the Strapdown INS system are derived. The INS data simulator tool is developed using the model constructed. This INS data simulator tool is implemented and simulated using MATLAB INS Toolbox which is becoming popular in navigation studies. In this regard, a flight path generator is also implemented using INS Toolbox. In addition, some new softwares were also developed as a part of this thesis and were used to simulate INS data. Hence a dynamic simulation environment is developed to apply TAN algorithms. An important advantage of this simulation environment is the ability to make simulations with different sampling rates and with different flight scenarios.

A fifteen state Kalman Filter is designed and implemented using the perturbation theory to model the position, velocity, attitude and the sensor errors. The LKF and the EKF TAN algorithms are developed and implemented. The results of the EKF TAN algorithm that is explained in detail in the previous chapter was better than LKF TAN algorithm. Besides, the performance of the algorithm is obtained for two different flight paths over different terrains and observed that the improvements are similar.

Another main observation is that the performance of terrain navigation depends on the norm of the terrain gradient in the flight region. Best results are obtained for a moderately rough terrain. The high nonlinearities of the terrain characteristic result with peak errors at the filter output.

As a second issue Kalman filter based TAN solution is simulated with different terrain stochastic linearization methods as MTL and NPPF. The MTL and the NPPF linearization methods are expected to increase the performance of any terrain based aided navigation. However, since DTED Level 0 is used throughout this study, these linearization methods becomes inappropriate because of DTED Level 0 resolution of 1 km. For a work of higher resolution these terrain linearizations approaches may result with better performance.

Another main disadvantage of the low resolution of elevation data is that the nonlinearities of terrain effect the performance of the filter by introducing instantaneous peak errors. The observed peaks are because of the terrain elevation data gird resolution.

The contribution of this master thesis is the assessment of the performance of the Kalman Filter based TAN algorithm within a dynamic simulation environment. It could be interesting to look into the problems below in the future:

- How to deal with the problem of false peaks in the matching procedure of the Kalman filter
- Discussion of DTED usage with different extrapolation approaches
- Using different terrain elevation data sets with higher resolution such as SRTM and ASTER
- Applying a highly nonlinear algorithm to this problem as particle filter
- Using two or more height measurement instrument as external aid
- Real-time applications of these algortihms can be applied with a real flight scenario

REFERENCES

[1] Daniel J. Biezad. *Integrated Navigation and Guidance Systems*. American Institude of Aeronautics and Astronatics, 1999.

[2] A.B. Chatfield. *Fundamentals of High Accuracy Inertial Navigation*. AIAA Inc., 1997.

[3] Kalman R. E. *A New Approach to Linear Filtering and Prediction Problems*. *Transaction of the ASME-Journal of Basic Engineering*, pages pp. 35–45, 1960.

[4] Vedat Ekütekin. Navigation and Control Studies on Cruise Missiles. PhD thesis, Middle East Technical University, 2007.

[5] A. Gelb. *Applied Optimal Estimation*. The M.I.T. Press, 1974.

[6] R. A. Gray. *In flight Detection of Errors for Enhanced Aircraft Flight Safety and Vertical Accuracy Improvement Using Digital Terrain Elevation Data With and Inertial Navigation System, Global Positioning System and Radar Altimeter*, 1999.

[7] Weill L.R. Grewal M.S. *Global Positioning Systems, Inertial Navigation and Integration*. John Wiley and Sons, Inc., 2001.

[8] Paul D. Groves. *GNSS, Inertial and Multisensor Integrated Navigation Systems*. Artech House,GNSS Technology and Applications Series, 2001.

[9] Jeff Hollowell. *HELI/SITAN: A Terrain Referenced Navigation Algorithm For Helicopters*. Technical Report CH2811-8/90/0000/0616, Sandia National Laboratories, 1990.

[10] The MathWorks Inc. *MATLAB, Mapping Toolbox, http://www.mathworks.com/products/mapping*.

[11] Britting K. *Inertial Navigation System Analysis*. Wiley Interscience,New York, 1971.

[12] Ronald D. Andreas Larry D. Hostetler. *Nonlinear Kalman Filtering Techniques for Terrain-Aided Navigation*. *IEEE, Catalog No:001 8-9286/83/0300*, pages pp. 315–323, 1983.

[13] GPSoft LLC. *GPSoft, INS Toolbox 2.0, http://www.gpsoftnav.com/inertial.html*.

[14] Angus P. Andrews Mohinder S. Grewal. *Kalman Filtering Theory and Practice Using Matlab*. Wiley Interscience Publication,John Wiley and Sons Inc., 2001.

[15] Vikas Kumar N. *Integration of Inertial Navigation System and Global Positioning System Using Kalman Filtering*, 2004.

[16] Alpay Abbak A. Özgör Doğru İ.Öztuğ Bildirici Aydın Üstün Necla Uluğtekin, H.Zahit Selvi. *SRTM Verilerine Dayalı Ülke Bazında 3x3 Çözünürlüklü Sayısal Yükseklik Modelinin Oluşturulması*. 2007.

[17] Gene Howell Newton Johson, Wang Tang. *Terrain Aided Navigation Using Maximum a Posteriori Estimation*. *IEEE, Position Location and Navigation Symposium*, pages pp. 464–469, 1990.

[18] Lennart Ljung Niclas Bergman and Fredrik Gustafsson. *Terrain Navigation using Bayesian Statistics*. Technical Report LiTH-ISY-R-2139, Department of Electrical Engineering Linköping University, 1999.

[19] Department of the Air Force. A-10 recovery operation, *http://www.spacecom.af.mil/2 1 sw/alO/index.htm*.

[20] Zhe Chen Pei-jun Yu and James C. Hung. *Performance Evalution of Six Terrain Stochastic Linearization Techniques for TAN*. *IEEE, Catalog No:91CH3007-2*, pages pp. 382–389, 1991.

[21] Kleinbauer Racheal. *Kalman Filtering Implementation with Matlab*, 2004.

[22] F.J. Regan. *Strapdown Systems, Part II, Algorithms*. Naval Surface Warfare Center, 1981.

[23] Robert M. Rogers. *Applied Mathematics in Integrated Navigation Systems, Third Edition*. American Institude of Aeronautics and Astronatics, 2007.

[24] Darryl Morrell Russell Enns. *Terrain-Aided Navigation Using the Viterbi Algorithm*. *Journal of Guidance, Control and Dynamics*, pages pp. 1443–1449, 1995.

[25] George M. Siouris. *Missile Guidance and Control Systems*. Springer-Verlag New York, Inc., 2004.

[26] D. H. Titterton and J.L. Weston. *Strapdown Inertial Navigation Technology*. The Lavebham Press Ltd., 1997.

[27] Wei Wang and Zhe Chen. *Error Model Identification on Digital Map of TAN System Based on EKF*. pages pp. 818–822, 1994.

[28] Greg Welch and Gary Bishop. *An Introduction to the Kalman Filter*. Technical Report TR 95-041, University of North Carolina at Chapel Hill, 2006.

[29] Wikipedia. Wikipedia encylopedia, *http://www.wikipedia.com, 30 August 2010*.

[30] Oliver J. Woodman. *An Introduction to Inertial Navigation*. Technical Report CUCAM-CL-TR-696, University of Cambrige, 2007.

[31] Alper Öztürk. *Development, Implementation and Testing of a Thightly Coupled Integrated INS/GPS System*, 2003.