

SATELLITE ORBIT ESTIMATION USING KALMAN FILTERS

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

BY

MEHMET İPEK

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

FEBRUARY 2017

Approval of the thesis:

SATELLITE ORBIT ESTIMATION USING KALMAN FILTERS

submitted by **MEHMET İPEK** 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. Gülbin Dural Ünver _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Tolga Çiloğlu _____
Head of Department, **Electrical and Electronics Engineering**

Assoc. Prof. Dr. Umut Orguner _____
Supervisor, **Electrical and Electronics Engg. Dept., METU**

Examining Committee Members:

Prof. Dr. Sencer Koç _____
Electrical and Electronics Engineering Department, METU

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

Prof. Dr. Çağatay Candan _____
Electrical and Electronics Engineering Department, METU

Assist. Prof. Dr. Emre Özkan _____
Electrical and Electronics Engineering Department, METU

Assist. Prof. Dr. Yakup Özkazanç _____
Electrical and Electronics Engg. Dept., Hacettepe University

Date: 02.02.2017

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: MEHMET İPEK

Signature :

ABSTRACT

SATELLITE ORBIT ESTIMATION USING KALMAN FILTERS

İpek, Mehmet

M.S., Department of Electrical and Electronics Engineering

Supervisor : Assoc. Prof. Dr. Umut Orguner

February 2017, 134 pages

In this thesis, satellite orbit determination problem is investigated using various types of Kalman filters based on several ground and space based measurement types. First, the performance of the Gauss' method for initial orbit determination is evaluated and the measurement time separation values required for it to give sufficiently good estimates are determined. Second, filter initialization methods based on a few measurements are proposed for angles only (AE), angles and range (AER), angles and Doppler (AED) and full state observation cases and their performances are investigated. Third, comparison of the relative performances of extended Kalman filter (EKF), unscented Kalman filter (UKF) and continuous-discrete EKF (CD-EKF) is carried out for the sample scenarios assuming that AER and full state measurements are available. It is shown that continuous-discrete EKF running with a reasonable prediction step size, gives the best results in terms of root mean square (RMS) estimation error and computation time. The performance of the CD-EKF is further examined by comparing its RMS errors with posterior Cramer Rao Lower Bound (PCRLB) for all measurement types and it is shown that CD-EKF is almost efficient. In addition to these, relative orbit determination performances obtained using AE, AER and AED measurements are compared. It is shown that the use of the Doppler measurement in addition to angles only measurements significantly improves the estimation performance in terms of position and velocity RMSE's.

Furthermore, usefulness of a second observing station (located sufficiently apart from first) is studied. It is found that using a second station provides a significant improvement in the estimation performance especially when the range and Doppler data are not available.

Keywords: Orbit determination, low earth orbit, satellite tracking, extended Kalman filter, unscented Kalman filter, continuous-discrete Kalman filter, RTS smoother, angles only, angles and range, angles and Doppler, initial orbit, Gauss' method, posterior Cramer-Rao lower bound.

ÖZ

KALMAN SÜZGEÇLERİ KULLANARAK UYDU YÖRÜNGE KESTİRİMİ

İpek, Mehmet

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi : Doç. Dr. Umut Orguner

Şubat 2017 , 134 sayfa

Bu tezde, uydu yörünge belirleme problemi, yer ve uzay kaynaklı değişik ölçümlere dayalı çeşitli tiplerde Kalman süzgeçleri kullanarak incelenmiştir. İlk olarak, Gauss başlangıç yörünge belirleme metodunun performansı değerlendirilmiştir ve yeterince iyi kestirimler için gerekli ölçüm zaman aralığı değerleri belirlenmiştir. İkinci olarak, birkaç ölçüme dayanan filtre başlatma yöntemleri, sadece açılar, açılar ve mesafe, açılar ve Doppler ve tüm durum vektörü gözlem durumları için önerilmiş ve bunların performansları incelenmiştir. Üçüncü olarak, açılar ve mesafe ve tüm durum vektörü ölçümlerinin mevcut olduğu varsayılan örnek senaryolar için, genişletilmiş Kalman süzgeci (EKF), kokusuz Kalman süzgeci (UKF) ve sürekli-ayrık Kalman süzgecinin (CD-EKF) göreceli performans karşılaştırması yapılmıştır. Makul bir tahmin adımı ile çalıştırılan sürekli-ayrık Kalman süzgecinin ortalama karekök kestirim hatası ve hesaplama zamanı açısından en iyi sonucu verdiği gösterilmiştir. Sürekli-ayrık Kalman süzgecinin performansı, ortalama karekök hatası ile sonsal Cramer-Rao alt sınırı (PCRLB) karşılaştırılarak her ölçüm tipi için incelenmiştir ve sürekli-ayrık Kalman süzgecinin neredeyse etkin olduğu gösterilmiştir. Bunlara ek olarak, sadece açılar, açılar ve Doppler, açılar ve mesafe gözlem durumları için elde edilen göreceli yörünge belirleme performansları karşılaştırılmıştır. Sadece açılar ölçümünün yanında Doppler kullanılmasıyla pozisyon ve hız ortalama karekök hataları açısından kestirim performansını hayli geliştirdiği gösterilmiştir. Ayrıca, ikinci

bir gözlem istasyonunun (ilk istasyondan yeterince uzağa yerleştirilmiş) faydalılığı üzerinde çalışılmıştır. İkinci bir istasyon kullanmanın, özellikle mesafe ve Doppler verisi mevcut olmadığında kestirim performansında kayda değer bir iyileştirme sağladığı bulunmuştur.

Anahtar Kelimeler: Yörünge belirleme, genişletilmiş Kalman süzgeci, kokusuz Kalman süzgeci, sürekli-ayrık genişletilmiş Kalman süzgeci, RTS düzleştirici, sadece açılar, açılar ve mesafe, açılar ve Doppler, başlanıç yörüngesi, Gauss metodu, sonsal Cramer-Rao alt sınırı.

To people who are interested in this work.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to Assoc. Prof. Umut Orguner for his constant support, guidance and valuable suggestions. His excellent comments and explanations have shaped my understanding of and my approach to estimation problems. Furthermore, it is a great honor to work with him.

I would like to acknowledge, Dr. Egemen İmre for his guidance on orbit dynamics and dynamic modelling, Dr. Murat Gökçe for his endless but also valuable discussions. I would also like to thank my colleagues who supported me throughout this work.

Besides, I would also like to thank my examining committee members for sharing their times and for their insightful comments and contributions.

TABLE OF CONTENTS

ABSTRACT	v
ÖZ	vii
ACKNOWLEDGMENTS	x
TABLE OF CONTENTS	xi
LIST OF TABLES	xv
LIST OF FIGURES	xvi
LIST OF ABBREVIATIONS	xix
CHAPTERS	
1 INTRODUCTION	1
1.1 Literature Review	4
1.2 Organization of the Thesis	6
2 BACKGROUND	9
2.1 State Models	9
2.1.1 Two Body Orbit Model	10
2.1.2 Perturbed Orbit Model	11
2.1.3 Numerical Integration	19

2.1.4	Classical (Keplerian) Orbital Elements	20
2.1.4.1	Two Line Element Set	23
2.2	Measurement Models	24
2.2.1	Full State Observation	27
2.2.2	Angles Only Observation	28
2.2.3	Angles and Doppler Observation	29
2.2.4	Angles and Range Observation	30
2.3	Reference Coordinate Systems and Transformations	30
2.3.1	Earth Centric Inertial Frame	32
2.3.2	Earth Centric Earth Fixed Frame	32
2.3.3	Topocentric Coordinate Frames	33
2.3.4	General Transformations	34
2.3.5	ECI - ECEF Transformation	36
2.3.5.1	Time Expressions and Their Relations	39
2.3.6	Lat-Lon-Alt to ECEF Transformation	42
2.3.7	ECEF to AER Transformation	43
2.3.8	Az-El to Topocentric Ra-Dec Transformation	47
2.4	State Estimation and Smoothing	48
2.4.1	Extended Kalman Filter	51
2.4.2	Continuous-Discrete Extended Kalman Filter	54
2.4.3	Unscented Kalman Filter	56

2.4.4	Filter Predictions : A Toy Example	59
2.4.5	Nonlinear Smoothing	63
2.5	Cramer-Rao Lower Bound	65
3	INITIAL ORBIT DETERMINATION	71
3.1	Gauss' Method of Initial Orbit Determination	73
3.2	Performance Evaluation of the Gauss' Method	78
4	ORBIT ESTIMATION USING DIFFERENT STATE ESTIMATORS	85
4.1	Data Generation and Necessary Parameters	85
4.1.1	True Data Generation	85
4.1.2	Noisy Data Generation	86
4.1.3	Filter Parameters	87
4.1.4	Error Metrics	88
4.2	Filter Initialization	89
4.2.1	Full State Observation Case	89
4.2.2	AER Observation Case	90
4.2.2.1	Two-Point Initialization	90
4.2.2.2	One-Point Initialization	92
4.2.2.3	One-Point Initialization with Velocity Direction Information	93
4.2.2.4	Initialization Method Comparison	94
4.2.3	AE Observation Case	95

4.2.3.1	One Point Initialization	96
4.2.3.2	Middle Point Initialization	98
4.2.3.3	Smoother Initialization	99
4.2.3.4	Initialization Method Comparison	100
4.2.4	AED Observation Case	102
4.3	Estimator Comparison	102
4.3.1	Relative Filter Performances	103
4.3.2	Effects of the Initial Uncertainties	109
4.4	Estimator Performance with Respect to PCRLB	113
4.4.1	Absolute Filter Performances	117
5	CONCLUSION AND FUTURE WORK	127
	REFERENCES	129

LIST OF TABLES

TABLES

Table 2.1.1 Reference values of exponential density model [1].	14
Table 2.1.2 Relation between the eccentricity and the shape	21
Table 2.4.1 Equations for standard Kalman filter.	51
Table 2.4.2 Equations for CD-EKF.	55
Table 2.4.3 Prediction performance comparison of the filters using RK4 method.	62
Table 2.4.4 Prediction performance comparison of the filters using Euler's method.	63
Table 4.2.1 Comparison of initialization techniques for the AE case by time averaged RMSE's using last 50 estimates.	102
Table 4.3.1 Filter computation times.	106
Table 4.3.2 Full state observation case: Time averaged RMSE perfor- mances of last 50 estimates.	108
Table 4.3.3 AER observation case: Time averaged RMSE performances of last 50 estimates.	108
Table 4.3.4 Effects of the initial uncertainty on the RMSE performance.	113
Table 4.4.1 Time averaged RMSE and PCRLB comparison for the last 20 samples.	125

LIST OF FIGURES

FIGURES

Figure 1.0.1 Earth orbiting objects [4].	2
Figure 1.0.2 Earth orbiting satellites with different altitudes [7].	2
Figure 2.1.1 Basic forces explaining orbital motion [7].	10
Figure 2.1.2 Illustration of the gravitational forces.	10
Figure 2.1.3 Magnitudes of the perturbing accelerations versus altitude [20].	12
Figure 2.1.4 Gravity potential of an arbitrary object. ϕ is latitude and λ is longitude of point A	15
Figure 2.1.5 Earth's gravitational field representation [27].	16
Figure 2.1.6 Tesseral and zonal harmonics [23].	16
Figure 2.1.7 First few zonal harmonics [24].	18
Figure 2.1.8 Representation of the orbit.	21
Figure 2.1.9 Representation of the right ascension and the argument of perigee.	22
Figure 2.1.10 Two Line Element format and its explanation for the NOAA6 [35].	23
Figure 2.2.1 One way and two way radio tracking of satellites.	25
Figure 2.2.2 Basic principles of ranging via measuring phase shifts [3]. . .	25
Figure 2.2.3 Satellite tracking telescope and symbolic camera image. . .	26
Figure 2.2.4 Satellite laser ranging system [24].	26
Figure 2.2.5 Azimuth and elevation angles on local coordinate system. . .	28
Figure 2.3.1 ECI Frame [1].	32

Figure 2.3.2 Rotation of Cartesian coordinate system [24].	35
Figure 2.3.3 Coordinate system translation.	36
Figure 2.3.4 ECI and ECEF representation.	37
Figure 2.3.5 Illustration of precession and nutation [2].	39
Figure 2.3.6 Geodetic latitude ϕ and geocentric latitude ϕ_g on reference ellipsoid.	43
Figure 2.3.7 Visualization of ECEF, ENZ and geodetic coordinates [24].	44
Figure 2.3.8 Vector representation of observer and satellite in Earth fixed frame.	45
Figure 2.3.9 Visualization right ascension α and declination δ with ECI frame.	47
Figure 2.4.1 Update and prediction mechanism of Kalman filtering. . . .	50
Figure 3.1.1 Scenario for the Gauss' method.	73
Figure 3.2.1 Minimum time separation to achieve a specified position un- certainty.	81
Figure 3.2.2 Minimum time separation to achieve a specified velocity un- certainty.	82
Figure 3.2.3 Minimum time separation to achieve a specified initial range uncertainty.	83
Figure 4.2.1 Illustration of orbit ellipse, its tangent and satellite position vector.	93
Figure 4.2.2 A sample AER trajectory.	96
Figure 4.2.3 RMSE performances of EKFs using different initialization methods (the first 14 samples).	97
Figure 4.2.4 Effect of initialization method to estimator performance for AE measurement case.	101
Figure 4.3.1 Sample full state measurements.	104
Figure 4.3.2 RMSE performance of filters with full state measurements. .	109
Figure 4.3.3 Zoomed version of Figure 4.3.2.	110

Figure 4.3.4 RMSE performance of filters with AER measurements. . . .	111
Figure 4.3.5 Zoomed version of Figure 4.3.4.	112
Figure 4.3.6 RMSE performances of the filters with initial covariance P_0^3	113
Figure 4.4.1 Full state case: Comparison of the proposed estimator with PCRLB.	117
Figure 4.4.2 AER case: Comparison of the proposed estimator with PCRLB.	118
Figure 4.4.3 AER with two observing station: Comparison of the proposed estimator with PCRLB.	119
Figure 4.4.4 AE case: Comparison of the proposed estimator with PCRLB.	120
Figure 4.4.5 AE with two observing station: Comparison of the proposed estimator with PCRLB.	121
Figure 4.4.6 AED case: Comparison of the proposed estimator with PCRLB.	122
Figure 4.4.7 AED with two observing stations: Comparison of the proposed estimator with PCRLB.	123
Figure 4.4.8 RMSE performance comparison of the AE and AED measurement cases.	124
Figure 4.4.9 RMSE performances of the EKF and the UKF with respect to the discrete PCRLB.	126

LIST OF ABBREVIATIONS

AE	Azimuth-Elevation
AED	Azimuth-Elevation-Doppler
AER	Azimuth-Elevation-Range
AGI	Analytical Graphics Incorporated
Alt	Altitude
Az	Azimuth
BC	Before Christ
CD-EKF	Continuous-Discrete Extended Kalman Filter
CIRA	Cospar International Reference Atmosphere
CRLB	Cramer-Rao Lower Bound
Cov	Covariance
Dec	Declination
ECEF	Earth Centric Earth Fixed
ECI	Earth Centric Inertial
EKF	Extended Kalman Filter
El	Elevation
ENZ	East-North-Zenith
ESA	European Space Agency
DOP	Dilution of Precision
GEO	Geosynchronous Orbit
GAST	Greenwich Apparent Sidereal Time
GMST	Greenwich Mean Sidereal Time
GMT	Greenwich Mean Time
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GSFC	Goddard Space Flight Center
GST	Greenwich Sidereal Time
ISS	International Space Station

JD	Julian Day
JGM	Joint Gravity Model
KF	Kalman Filter
Lat	Latitude
LEO	Low Earth Orbit
LMMSE	Linear Minimum Mean Square Error
Lon	Longitude
LS	Least Squares
LST	Local Sidereal Time
MEO	Medium Earth Orbit
MSE	Mean Square Error
NASA	National Aeronautics and Space Administration
NOAA	National Oceanic and Atmospheric Administration
NORAD	North American Aerospace Defence Command
PCRLB	Posterior Cramer-Rao Lower Bound
PF	Particle Filter
Pos	Position
Ra	Right Ascension
RK4	Fourth Order Runge-Kutta
RMS	Root Mean Square
RMSE	Root Mean Square Error
SEZ	South-East-Zenith
SLR	Satellite Laser Ranging
SNR	Signal to Noise Ratio
STK	Systems Tool Kit
TLE	Two Line Element
UKF	Unscented Kalman Filter
UT	Universal Time
UTC	Coordinated Universal Time
Vel	Velocity
WGS	World Geodetic System
3-D	Three Dimensional

CHAPTER 1

INTRODUCTION

A satellite is a natural or an artificial object that revolves around a celestial body and an orbit is the path followed by the satellite periodically. Orbital motion of comets, natural satellites and planets have been studied since ancient times [1]. However, the most significant contributions were made in the last several hundred years. Kepler and Newton are two of the greatest contributors of this area. Kepler described the orbital motion laws and Newton explained Kepler's ideas mathematically [2].

After the launch of first man made satellite [3] Sputnik in 1957, the period of artificial satellites began and the number of Earth orbiting satellites has been increasing every year since then. Now there are several thousands of man made satellites or satellite related objects around the Earth (see Figure 1.0.1). Orbits of the Earth orbiting satellites can be classified according to their altitude and several other properties such as inclination (angle between orbital plane and equatorial plane), ellipticity, mission as well [5, 6]. These classification criteria do not completely distinguish one type of orbit from another. Low Earth orbit (LEO) for altitude of under 1,500 km, medium Earth orbit (MEO) for altitude of $\sim 20,000$ km and geosynchronous orbit (GEO) for altitude of $\sim 36,000$ km are examples of classification with respect to altitude (see Figure 1.0.2). Since LEO satellites are the main interest of this thesis, other types of orbits will not be investigated in detail. Further information about orbit types and classifications can be found in [3, 5, 6, 7, 8]. LEO satellites generally revolve in circular orbits at altitudes between 300 and 1,500 km [3]. Altitudes below 300 km are not

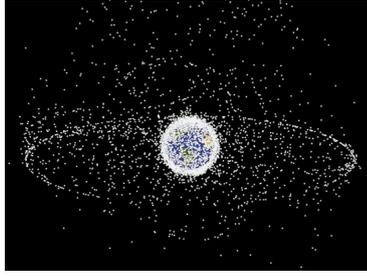


Figure 1.0.1: Earth orbiting objects [4].

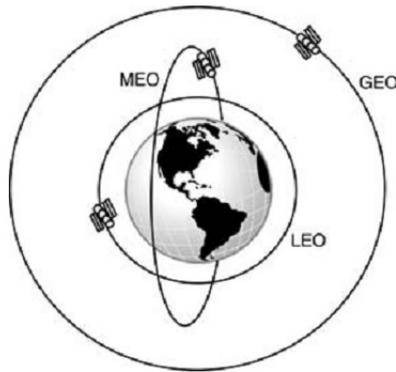


Figure 1.0.2: Earth orbiting satellites with different altitudes [7].

preferred due to rapid orbital decay because of the atmospheric drag. Orbital period changes from approximately 90 to 115 minutes for orbits at 300 and 1,500 km respectively. Due to their short period of revolution, a ground based sensor can track the satellite for only about ten minutes while the satellite is visible with respect to sensor. Moreover, signals transmitted from navigation satellites are available for LEO satellites.

Orbit of a satellite is chosen or designed according to its mission which can be communication, remote sensing, meteorological, scientific, navigational, intelligence etc. [5]. For example, remote sensing satellites generally revolve around polar, circular low Earth orbit because of coverage, sensor resolution and communication power requirements. Communication satellites are generally launched into geostationary orbit (geosynchronous orbit with approximately zero inclination) due to its large and non-changing coverage. Navigational satellites (GPS, Glonass etc.) are launched into MEO in order to satisfy the required minimum number of continuously visible satellites with using fewer total number of

satellites.

Satellite operators or users would like to know the satellite position, velocity and their uncertainties for different purposes. For example an Earth observation satellite operator should know the position of the satellite with a few meters of uncertainty in order to match the coordinates of the image (taken by the satellite) with coordinates of the Earth correctly (geolocation). Furthermore, a laser ranging station should know the satellite position with an accuracy of tens of meters in order to strike the satellite with its very narrow laser beam. Moreover, an Earth station that is used for satellite communication should accurately (required accuracy depends on the antenna beamwidth) know the pointing angles of the antenna for signal quality. In addition, collision avoidance maneuvers and collision probability calculations also need accurate orbit information [9].

Orbit determination has been studied for more than two hundred years but after the launch of the first artificial satellite, orbit determination became much more important. Kepler, Newton, Laplace and Gauss were four of early contributors of this area [1, 2]. Gauss calculated the orbit of asteroid Ceres and played a major role in the rediscovery of it [2]. He is also known as the inventor of the Least Square (LS) Method for fitting observations to a best possible orbit [2],[10].

Orbit determination is the use of mathematical techniques to calculate the position and velocity of a satellite or an orbiting object using dynamical equations of motion and the noisy data coming from different kinds of sensors [11]. Uncertainty reduction for the orbit is also a responsibility of the orbit determination process. The data to be used is generally taken from onboard sensors such as Global Navigation Satellite System (GNSS) receivers or ground based systems such as radar, laser tracking stations or optical telescopes. Optical telescopes were the only sensor type to observe space objects until mid 1900s. In addition to observations, a dynamic model describing the satellite motion and the related mathematical tools for orbit determination should be considered. Dynamic models can be constructed using Newton's laws of motion. In this context accurate force modelling and taking these forces into account are key parameters

for precise orbit determination [12].

There exist different mathematical techniques for orbit determination. These techniques can be divided into two parts: classical and modern orbit determination. In classical (deterministic) approach for orbit determination, measurement and modelling errors are not taken into account. Unlike the classical approach, possible errors are taken into account in the modern approach [12]. While initial orbit determination techniques are examples of the classical approach, batch and sequential estimation are commonly used examples of the modern approach [11]. Initial orbit determination is generally made in order to find a reasonable initial condition for the recursive estimation techniques by using only few measurements from the ground based tracking sensors [13]. Estimation techniques try to reduce the observation and/or model based errors by means of statistical methods such as batch least squares and Kalman filtering. A batch estimator uses a set of measurements in order to refine the orbit at a certain epoch. On the other hand, a sequential estimator uses the current measurement in order refine (update) the orbit at the current epoch, uses the dynamic model in order to propagate the orbit until a new measurement arrives, then makes a refinement and it continues this procedure [1, 3, 11, 12]. Although, batch and sequential estimators have been applied to orbit determination problem successfully and both are powerful techniques [3], batch estimators are slowly being replaced by sequential estimators [11]. In this thesis only sequential estimation techniques (Kalman filters) with various types of measurements for orbit determination will be investigated.

1.1 Literature Review

In the literature, sequential orbit estimation concepts with measurements coming from different types of sensors are mentioned in several books such as [1, 3, 12]. Precise orbit determination using GNSS measurements, range measurements coming from laser ranging systems are investigated widely with both sequential (Kalman filters) and batch estimation techniques. These precision orbit determination studies rely on very accurate force modelling and include the es-

timization of the gravity field and drag parameters. GEO satellite tracking using sampled data Kalman filters using angles and range measurements produced by several LEO satellites is investigated in [14]. In [15], range only measurements coming from three radar stations are employed in the Kalman filter in order to determine the orbit. In [11], evolution of the orbit determination concepts are summarized in terms of dynamic models, common observation systems, types of measurements and their accuracies. For the initialization procedures, in [16], uncertainty calculation methodology based on the unscented transform is investigated for the Gauss' method. Relative error performances of the extended Kalman filter (EKF) and sigma point filters using angles and range measurements, are investigated in [17]. In [17], Herrick-Gibbs method is proposed for filter initialization. In [18], the problem of tracking a GEO satellite by a single LEO satellite is considered and the error performances of EKF, unscented KF (UKF), particle filter (PF) and the linear minimum mean square error (LMMSE) filter are compared both with each other and with posterior Cramer Rao lower bound.

Compared to the aforementioned previous literature, the contributions of this thesis can be described as follows.

- Reliability region of the Gauss' method of initial orbit determination is studied numerically for a LEO satellite case.
- As a sequential estimator, continuous-discrete extended Kalman filter (CD-EKF), which is rarely used in the literature, is utilized for orbit estimation in addition to EKF and UKF.
- Cramer-Rao lower bound (CRLB) is calculated for the continuous-discrete orbit model.
- Evaluation of the absolute performance of the CD-EKF is carried out by comparing the root mean square error of the estimates with the CRLB. It should be mentioned that the CRLB is not a well-known estimator performance criterion in orbit determination community.

Satellite orbits can be determined by using various types of sensors with certain

accuracies. Future orbit (i.e., the state of a satellite) can be found by propagating the current state according to the dynamic model. In this context, it should be mentioned that the orbit determination errors increase as the propagation time increases. In order to reduce the error caused by the propagation, space or ground based measurements are utilized. Error reduction depends on the number of measurements and their accuracies. For example, when one have a precise orbit information of a satellite, ground based angles only measurements can be useless since their accuracies are not very good.

In this thesis, filter initialization and orbit estimation of a satellite is carried out assuming there is no prior orbit information except the coarse knowledge of its altitude. Moreover, physical properties (geometry and mass) of the satellite are also assumed to be known. It should be noted that, in order to track the satellite with ground based sensors, one should know the pointing angles of the ground based sensors (radar, telescope and etc.) with a certain level of accuracy. This is because satellite should lie in the field of view of the sensor. It is also assumed that the satellite is initially captured and then tracked by the ground based sensors.

1.2 Organization of the Thesis

Chapter 2 introduces the necessary models, mathematical relations and tools for satellite orbit determination. State estimation and smoothing concepts using Kalman filters are presented. Cramer Rao lower bound used for the performance evaluation of Kalman filters is also described.

In Chapter 3, a brief review of deterministic initial orbit determination is given. In addition, Gauss' algorithm that is used for initial orbit determination purposes is explained. A reliable region is defined for the Gauss' algorithm where the algorithm works well with reasonable position and velocity errors. The necessary angular measurement error standard deviation and measurement time separation for the Gauss' algorithm to remain in this reliable region are investigated.

In Chapter 4, filter initialization methods based on a few measurements are proposed for angles only (AE), angles and range (AER), angles and Doppler (AED) and full state observation cases and their performances are investigated. Comparison of the relative performances of extended Kalman filter (EKF), unscented Kalman filter (UKF) and continuous-discrete EKF (CD-EKF) is carried out for the sample scenarios assuming that AER and full state measurements are available. The performance of the CD-EKF is further examined by comparing its RMS errors with posterior Cramer Rao Lower Bound (PCRLB) for all measurement types. In addition to these, relative orbit determination performances obtained using AE, AER and AED measurements are compared. Furthermore, usefulness of a second observing station (located sufficiently apart from first) is studied.

In Chapter 5, conclusions of this study are drawn and possible future approaches that can be used in order to improve the orbit determination performance further are listed.

CHAPTER 2

BACKGROUND

In this chapter, necessary models and tools for satellite orbit determination are explained. Orbit dynamic model and measurement models for different types of observations are derived. Furthermore, needed coordinate systems and transformations between these coordinate systems are also presented. Moreover, state and parameter estimation methods for satellite orbit determination are investigated. Finally, Posterior Cramer Rao Lower Bound (PCRLB) for measuring the quality of sequential Bayesian estimators is also introduced.

2.1 State Models

In order to explain orbital motion of satellites, it is essential to express that this motion is basically governed by two forces. First one is centripetal force due to gravitational attraction and the second one is centrifugal force due to circular motion (see Figure 2.1.1). The former one is not really acting on the satellite, it just comes from Newton's third law namely action-reaction principle [7].

Combining these forces with Newton's laws, one can express this orbital motion mathematically with differential equations and the resulting expression gives the state model that will be used for orbit determination.

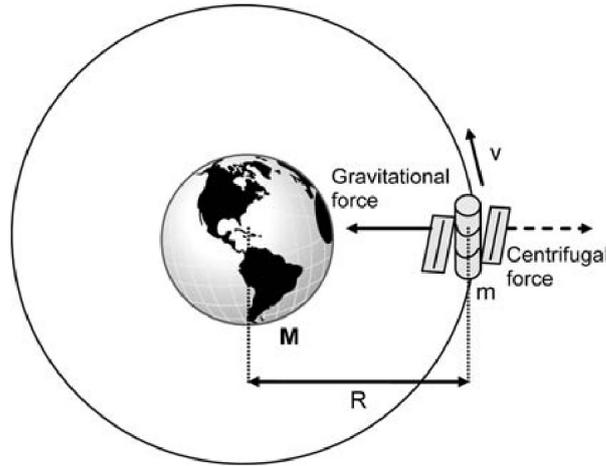


Figure 2.1.1: Basic forces explaining orbital motion [7].

2.1.1 Two Body Orbit Model

In this problem, it is assumed that there exists only two bodies with spherically symmetric mass distribution and mass of one of them is negligible with respect to other [1]. Using this assumption and Newton's law of universal gravity, the magnitude of the forces acting on these masses is

$$F = \frac{Gm_1m_2}{r^2} \quad (2.1.1)$$

where G is the universal constant of gravity, m_1 and m_2 are the masses of two bodies in space and finally r is the distance between these two bodies [2].

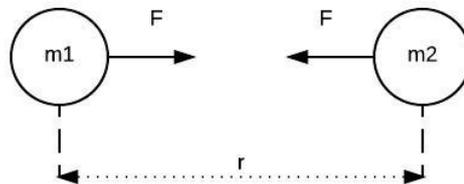


Figure 2.1.2: Illustration of the gravitational forces.

Furthermore, Newton's first and second laws of motion state that the total force acting on a mass m is the multiplication of this mass with its acceleration \vec{a}

which is represented by the following equation [1].

$$\vec{F} = m \vec{a}. \quad (2.1.2)$$

It should be noted that Newton's second law requires the concept of inertial frame. Inertial frame is a coordinate system which is non-rotating and unaccelerated [2]. Combining Newton's law of gravitation and his first two laws of motion for the earth and one of its satellites yields the following second order differential equation as orbit dynamic model [19].

$$\ddot{\vec{r}} = -\frac{\mu}{\|\vec{r}\|^3} \vec{r} \quad (2.1.3)$$

where μ is the multiplication of gravitational constant and the mass of the Earth, which is approximately $398,600.44 \frac{km^3}{s^2}$ [1]. \vec{r} is the position vector of the satellite in inertial frame and it can be expressed in vector form as follows

$$\vec{r} = \begin{bmatrix} x & y & z \end{bmatrix}^T.$$

Similarly the velocity vector is

$$\dot{\vec{r}} = \begin{bmatrix} v_x & v_y & v_z \end{bmatrix}^T.$$

2.1.2 Perturbed Orbit Model

In real life, the presence of other bodies and other forces (perturbing forces) acting on the satellite, two body problem does not represent the reality [8]. These forces are gravitational forces, drag forces, radiation pressure, third body effects, tidal effects, relativistic effects etc. [2]. Including all of these forces in orbit dynamic model, increases the accuracy of the orbit determination. However, magnitudes of the perturbing forces depend on the orbit types (see Figure 2.1.3). As an example, atmospheric drag is not very effective at GEO but it is one of the major forces at LEO. After taking these disturbing forces into account, the two body model becomes

$$\ddot{\vec{r}} = -\frac{\mu}{\|\vec{r}\|^3} \vec{r} + \vec{a}_{grav} + \vec{a}_{drag} + \vec{a}_{other}. \quad (2.1.4)$$

In the above equation, \vec{a}_{grav} and \vec{a}_{drag} represent the accelerations caused by non-uniform gravity field of the Earth and the atmospheric drag respectively. Here, \vec{a}_{other} is the acceleration component coming from all other perturbing forces. Since the main interest of this thesis is LEO satellites, only gravitational and drag based perturbations will be taken into account and the remaining forces will be considered as a process noise in Kalman filtering operations. In Fig-

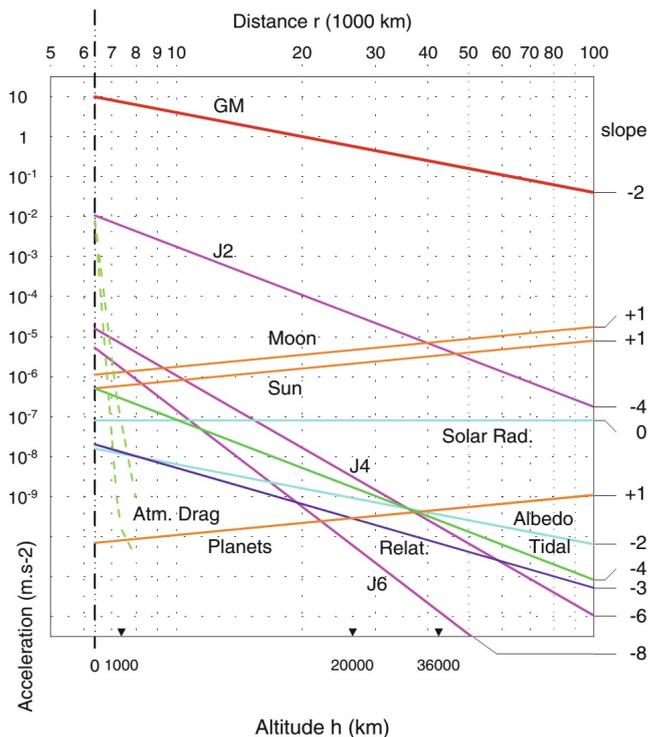


Figure 2.1.3: Magnitudes of the perturbing accelerations versus altitude [20].

Figure 2.1.3, GM represents the gravitational acceleration of the spherical Earth with uniform mass distribution. J2, J4 and J6 are the zonal gravitational accelerations caused by the spherical harmonics and these are described later in this section. Moon, Sun and Planets represent the accelerations due to third body effects. Other perturbing accelerations seen in the aforementioned figure are solar radiation pressure (caused by the photons), photons reflected from the Earth (albedo), relativistic effects and Earth tides.

For near Earth satellites, drag force is one of the largest disturbing effect (see Figure 2.1.3). It is an energy dissipating effect and causes orbital decay. Accel-

eration due to drag force can be represented as follows.

$$\vec{a}_{drag} = \frac{1}{2} \rho C_d \frac{A}{m} \|\vec{v}\| \vec{v} \quad (2.1.5)$$

where ρ is the air density in $\frac{kg}{km^3}$, C_d is the drag coefficient related with the satellite geometry, A in km^2 is the satellite cross sectional area in the direction of motion, m is the mass of satellite in kg and finally $\vec{v} = [v_x \ v_y \ v_z]^T$ is the velocity with respect to air in $\frac{km}{s}$. In (2.1.5), acceleration depends on atmospheric density, satellite geometry and velocity. Modelling of atmospheric density is very difficult due to its changing nature. It changes depending on the solar activity and the Earth's magnetic field variations. The simplest density model for altitudes up to 1,000 km is the Exponential Model and it is given by the following equation [1]

$$\rho = \rho_0 e^{-\frac{h_{ellp} - h_0}{H}} \quad (2.1.6)$$

where ρ_0 is the air density and H is the scaling factor at certain reference altitude h_0 . Moreover, h_{ellp} is the actual altitude defined as satellite height above Earth's surface. Let the norm of satellite position vector be $r = \sqrt{x^2 + y^2 + z^2}$ then actual altitude will be

$$h_{ellp} = r - R_e$$

where R_e shows the Earth equatorial radius with a value of 6,378.137 km [3]. Nominal values for ρ_0 and H can be acquired from Table 2.1.1 for different reference altitudes. For example, if we consider a satellite at altitude of 650 km then the air density can be calculated by using the model given in (2.1.6) and values given in Table 2.1.1.

$$\rho = 1.454 \times 10^{-13} e^{-\frac{(650 - 600)}{71.835}}, \left(\frac{km}{m^3}\right).$$

There are much more complex atmospheric models both depending on time and position. Solar activities, thermal and magnetic variations are taken into account in these complex models. CIRA (Cospar International Reference Atmosphere), Harris-Priester and Jachia-Roberts are the examples. Further information about atmospheric modelling can be obtained from [1, 3, 21]. Exponential model will be used in following sections since the consistency of the air density model is beyond the scope of this thesis.

Table 2.1.1: Reference values of exponential density model [1].

Altitude h_{ellp} (km)	Base Altitude h_o (km)	Nominal Density ρ_o (kg/m ³)	Scale Height H (km)	Altitude h_{ellp} (km)	Base Altitude h_o (km)	Nominal Density ρ_o (kg/m ³)	Scale Height H (km)
0–25	0	1.225	7.249	150–180	150	2.070×10^{-9}	22.523
25–30	25	3.899×10^{-2}	6.349	180–200	180	5.464×10^{-10}	29.740
30–40	30	1.774×10^{-2}	6.682	200–250	200	2.789×10^{-10}	37.105
40–50	40	3.972×10^{-3}	7.554	250–300	250	7.248×10^{-11}	45.546
50–60	50	1.057×10^{-3}	8.382	300–350	300	2.418×10^{-11}	53.628
60–70	60	3.206×10^{-4}	7.714	350–400	350	9.158×10^{-12}	53.298
70–80	70	8.770×10^{-5}	6.549	400–450	400	3.725×10^{-12}	58.515
80–90	80	1.905×10^{-5}	5.799	450–500	450	1.585×10^{-12}	60.828
90–100	90	3.396×10^{-6}	5.382	500–600	500	6.967×10^{-13}	63.822
100–110	100	5.297×10^{-7}	5.877	600–700	600	1.454×10^{-13}	71.835
110–120	110	9.661×10^{-8}	7.263	700–800	700	3.614×10^{-14}	88.667
120–130	120	2.438×10^{-8}	9.473	800–900	800	1.170×10^{-14}	124.64
130–140	130	8.484×10^{-9}	12.636	900–1000	900	5.245×10^{-15}	181.05
140–150	140	3.845×10^{-9}	16.149	1000–	1000	3.019×10^{-15}	268.00

Gravitational disturbance acceleration due to the non-uniform mass distribution of the Earth is another substantial effect and should be taken into account for LEO satellites. It can be derived from geopotential of an arbitrary shaped body by taking the gradient of geopotential [1]. It is better to start with potential (u) of point mass m at distance r which is given by the following equation [3, 22].

$$u = G \frac{m}{r} \quad (2.1.7)$$

Then, gravity potential for an arbitrary shaped body (see Figure 2.1.4) with mass M at a point A with position vector \vec{r} , can be found by integrating the potential caused by infinitesimal mass element dm at position \vec{s} over the entire body as follows [23].

$$U = G \int \frac{dm}{\|\vec{r} - \vec{s}\|}. \quad (2.1.8)$$

Let the norm (i.e., the magnitude) of the vectors \vec{r} and \vec{s} be r and s respectively. Inverse of the distance between mass element and point A can be represented by the series expansion of Legendre polynomials [3, 23, 24] as

$$\frac{1}{\|\vec{r} - \vec{s}\|} = \frac{1}{r} \sum_{n=1}^{\infty} \left(\frac{s}{r}\right)^n P_n(\cos \psi) \quad (2.1.9)$$

where, $P_n(\cdot)$ shows the Legendre polynomials [25] of degree n and ψ is the angle between vectors \vec{r} and \vec{s} . Combining equations 2.1.8 and 2.1.9 yields

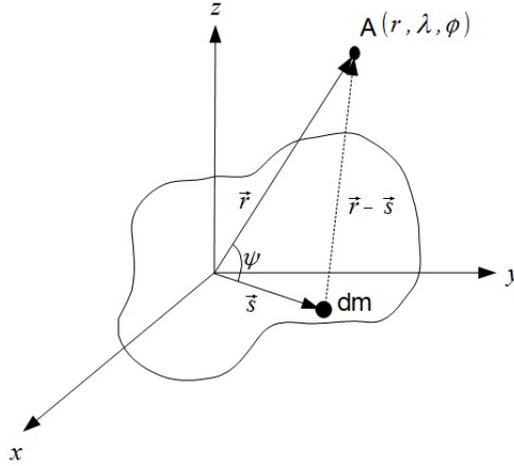


Figure 2.1.4: Gravity potential of an arbitrary object. ϕ is latitude and λ is longitude of point A .

$$U = G \int \frac{dm}{r} \sum_{n=1}^{\infty} \left(\frac{s}{r}\right)^n P_n(\cos \psi). \quad (2.1.10)$$

Instead of an arbitrary body, if Earth is considered and if angle ψ is replaced by the function of angles ϕ and λ which are the latitude and longitude of the point A respectively then (2.1.10) becomes

$$U = \frac{GM_e}{r} \left(1 + \sum_{n=1}^{\infty} \sum_{m=0}^n \left(\frac{R_e}{r}\right)^n P_{nm}(\sin \phi) (C_{nm} \cos m\lambda + S_{nm} \sin m\lambda) \right) \quad (2.1.11)$$

where M_e and R_e are mass and equatorial radius of the Earth respectively. Moreover, $P_{nm}(\cdot)$ shows the associated Legendre polynomials of degree n and order m . Lastly, C_{nm} and S_{nm} are gravitational coefficients also called as spherical harmonics [23] related with the mass distribution of the Earth and should be somehow determined. Determination of these coefficients is the subject of Geodesy and generally done by using orbit information of LEO satellites (see Figure 2.1.5). Briefly, if the orbit of a satellite is known then these coefficients can be estimated. Coefficients are directly associated with satellite position around the Earth. Hence, in order to calculate the gravitational acceleration, using a coordinate system that is fixed to Earth is necessary. Further information about gravitational models can be found in [1, 3, 23, 24, 26].

Spherical harmonics can be divided into three groups such as zonal, sectorial and tesseral depending on the segmentation type. Zonal and sectorial harmon-

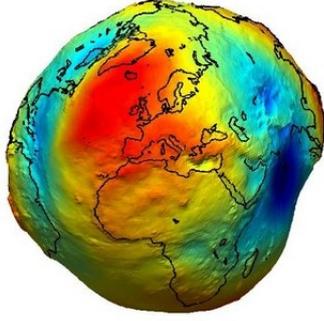


Figure 2.1.5: Earth's gravitational field representation [27].

ics divide the Earth in latitudinal and longitudinal directions respectively. In addition, tesseral harmonics split in both directions and the number of divisions depend on the numbers n and m . For tesseral harmonics $n \neq m \neq 0$. For example, $n = 9$ and $m = 6$ corresponds $n - m = 3$ latitudinal and $2m = 12$ longitudinal parts as shown in Figure 2.1.6. If it is assumed that the mass

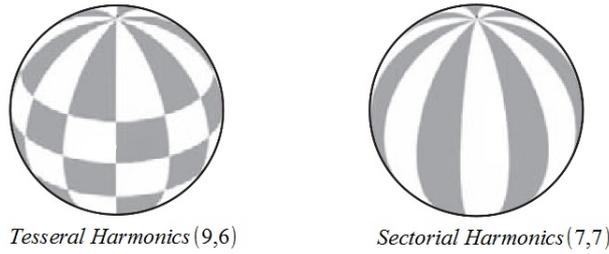


Figure 2.1.6: Tesseral and zonal harmonics [23].

distribution of the Earth is symmetric with respect to its rotation axis namely $m = 0$ then coefficients of sectorial harmonics (i.e., S_{nm} 's) will be zero and there exists only zonal harmonics (i.e., C_{n0} 's) and potential equation becomes

$$U = \frac{GM_e}{r} \left(1 - \sum_{n=2}^{\infty} J_n \left(\frac{R_e}{r} \right)^n P_{n0}(\sin \phi) \right) \quad (2.1.12)$$

where $-C_{n0} = J_n$. Visualization of the first few zonal harmonics can be seen in Figure 2.1.7. Furthermore, after C_{00} , gravitational effect of C_{20} is the greatest one [3] and it is approximately 1000 times higher than the other components [24]. Neglecting other zonal components yields the following equation [26].

$$U = \frac{GM_e}{r} \left(1 - J_2 \left(\frac{R_e}{r} \right)^2 \frac{1}{2} (3 \sin^2 \phi - 1) \right) \quad (2.1.13)$$

Rewriting above equation and representing U as $U = U_0 + U_1$ gives the following equation.

$$U = \underbrace{\frac{\mu}{r}}_{U_0} - \underbrace{\frac{\mu J_2 R_e^2}{2r^3} (3 \sin^2 \phi - 1)}_{U_1}.$$

In Section 2.1.1 it was mentioned that GM_e equals to μ . Moreover, U_0 shows the gravity potential of a sphere that was already taken into account in two body model and U_1 represents the perturbing gravitational potential due to the Earth oblateness. Finally, the unitless coefficient J_2 is given as ~ 0.00108263 [1] which is taken from the Joint Gravity Model 2 (JGM2) of NASA. Now, in order to find the perturbing acceleration caused by the oblateness, gradient of U_1 is needed. Let the gradient of U_1 be ∇U_1 then it can be expressed in vector form as follows.

$$\nabla U_1 = \vec{a}_{grav} = \begin{bmatrix} \frac{\partial U_1}{\partial x} \\ \frac{\partial U_1}{\partial y} \\ \frac{\partial U_1}{\partial z} \end{bmatrix}.$$

It is known that $z = r \sin \phi$ and $r = \sqrt{x^2 + y^2 + z^2}$ from the relations between spherical coordinates and Cartesian coordinates of point A (see Figure 2.1.4). Substituting these relations in expression of U_1 and taking the partial derivatives with respect to x , y and z give the following acceleration vector [1].

$$\vec{a}_{grav} = \begin{bmatrix} \frac{-3J_2\mu R_e^2}{2r^5} \left(1 - \frac{5z^2}{r^2}\right) x \\ \frac{-3J_2\mu R_e^2}{2r^5} \left(1 - \frac{5z^2}{r^2}\right) y \\ \frac{-3J_2\mu R_e^2}{2r^5} \left(3 - \frac{5z^2}{r^2}\right) z \end{bmatrix}. \quad (2.1.14)$$

It should be noted that above vector is in Earth fixed coordinates and can not be used in dynamic equation directly since it is not represented in inertial frame. Using the information that, a second order differential equation can be represented by two first order differential equations [2]. Taking \vec{r} and $\dot{\vec{r}}$ as

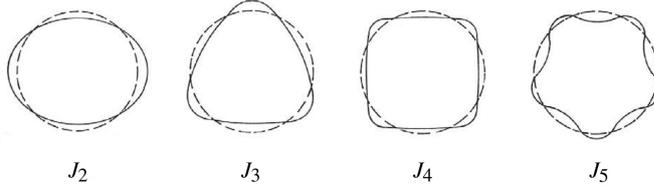


Figure 2.1.7: First few zonal harmonics [24].

states, our dynamic model will be

$$\underbrace{\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{v}_x \\ \dot{v}_y \\ \dot{v}_z \end{bmatrix}}_{\dot{X}} = \begin{bmatrix} v_x \\ v_y \\ v_z \\ \frac{-\mu x}{\sqrt{(x^2 + y^2 + z^2)^3}} + a_{drag}^x + a_{grav}^x \\ \frac{-\mu y}{\sqrt{(x^2 + y^2 + z^2)^3}} + a_{drag}^y + a_{grav}^y \\ \frac{-\mu z}{\sqrt{(x^2 + y^2 + z^2)^3}} + a_{drag}^z + a_{grav}^z \end{bmatrix} + \omega$$

$f_c(X)$

where a_{drag}^n and a_{grav}^n for $n = \{x, y, z\}$ show the components of drag and gravity based accelerations. Furthermore, ω is the 6x1 process noise vector represents mismodelling. Above equation can be rewritten in a more compact form as follows

$$\dot{X} = f_c(X) + \omega \quad (2.1.15)$$

where $f_c(\cdot)$ represents a vector consisting nonlinear functions and subscript c denotes that the function is in continuous time. Moreover, noise component should be included in our dynamic model because the force modelling used is actually a simplified version of the real case. In this model only the most significant forces for LEO satellites are represented and the rest is taken as a process noise. It should be noted that even if a precise force modelling is used, process noise is still needed because of the remaining mismatch between the model and reality.

2.1.3 Numerical Integration

In order to find the solution of the differential equation which represents the orbit dynamics, analytical or numerical techniques can be used. In the presence of perturbation accelerations, use of analytical methods is relatively complex [28]. Numerical integration methods are widely used since the computational effort takes a minor role with the performance of today's computers [24]. Use of numerical integration can also be considered as a discretization method for dynamic models and this provides convenience for computer based estimation techniques. There are several types of numerical integrators used for orbit determination purposes such as Runge-Kutta, Cowell, Encke, Gauss-Jackson and Adams-Bashforth and many others [3, 24]. These methods belong to single step and multi step integration classes. In single step methods, every integration step can be considered individually i.e., they do not require the results found in previous steps contrary to multi step methods and this makes single step methods more compact and easy to use [3]. In order to choose a numerical integration method, one should consider its accuracy, speed, complexity and storage requirements [2]. Furthermore, type of the orbit and the perturbations also affect the choice of integration type and the required step size. In addition, Runge-Kutta type methods (an example of single step integration) are usually used in wide variety of problems. For instance, the simplest version of Runge-Kutta method is based on the first order approximation of Taylor series expansion and this is also known as Euler's method [3].

Fourth order Runge-Kutta (RK4) algorithm is used in this thesis due to its simple implementation, moderate error performance [29] and computational load. Detailed information about numerical integration with applications to orbit determination can be found in [1, 2, 3, 24, 30].

For a continuous time dynamic system represented by a first order differential equation $\dot{x} = f(x)$, the solution with RK4 integration will be

$$x_{k+1} = x_k + \frac{\delta}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (2.1.16)$$

$$\begin{aligned} k_1 &= f(x_k) \\ k_2 &= f(x_k + k_1\delta/2) \\ k_3 &= f(x_k + k_2\delta/2) \\ k_4 &= f(x_k + k_3\delta) \end{aligned} \quad (2.1.17)$$

where δ is the integration step size, k_i 's for $i = 1, 2, 3, 4$ are intermediate calculations and x_k is the state at time $t_0 + \delta k$. Note that smaller step sizes give more accurate solutions but take more time to propagate. Moreover, in order to use a numerical integration method, one needs a starting point x_0 for time t_0 .

After implementing RK4 integration method for the continuous time orbit dynamic model given in (2.1.15) with an integration step size of δ , we obtain the following discrete time model for $k = 0, 1, 2, \dots$

$$X_{k+1} = f_d(X_k) + \omega_d \quad (2.1.18)$$

where X_k represents the state containing position and velocity of satellite at time $t_0 + \delta k$ assuming X_0 is the state at time t_0 , $f_d(\cdot)$ is the discrete dynamic model resulting from RK4 integration and ω_d is the corresponding noise component in discrete time.

2.1.4 Classical (Keplerian) Orbital Elements

Classical or Keplerian orbital elements are the commonly used parameterization in order to define the orbit of a satellite. One can easily visualize the orbit by knowing the orbital elements. These elements describe the size, shape and the orientation of the orbit in space and the location of the satellite in the orbit [1, 31]. Orbital elements are given and explained as follows [1, 2, 31, 32].

- Semi-major axis (a) specifies the size of the orbit. It is the half of the distance between the farthest points of the orbit as seen in Figure 2.1.8.

Table 2.1.2: Relation between the eccentricity and the shape

Eccentricity	Orbit Shape
$e = 0$	Circular
$0 < e < 1$	Elliptical
$e = 1$	Parabolic
$e > 1$	Hyperbolic

- Eccentricity (e) defines the shape of the orbit and it can be given in a simple form the relation below [31].

$$e = \frac{c}{a} \quad (2.1.19)$$

where c is the half of the distance between two foci and a is the semi-major axis (see Figure 2.1.8). The relation between the orbital shape and the eccentricity can be seen in Table 2.1.2. Note that the parabolic and the hyperbolic shaped orbits are open [1], in other words, they are not periodic.

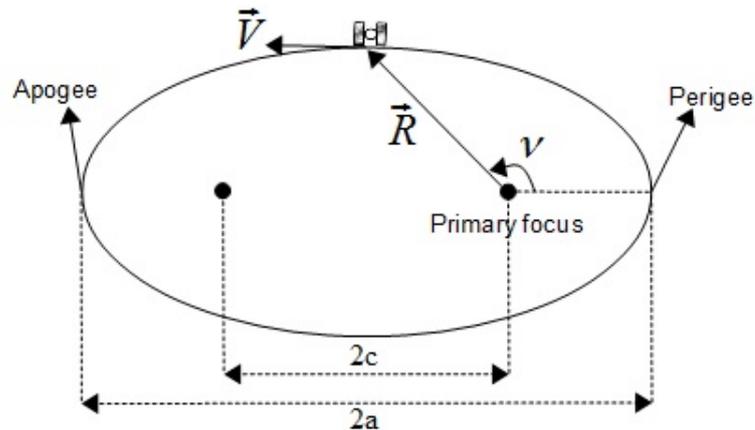


Figure 2.1.8: Representation of the orbit.

- Inclination (i) is the angle between the equatorial plane and the orbital plane. If the motion is in the direction of the Earth's rotation then the

inclination obeys $0^\circ < i < 90^\circ$ and $90^\circ < i < 180^\circ$ for the reverse case. It should be mentioned that the inclination is either 0° or 180° when the orbital plane and the equatorial planes are coincident. Inclination is 90° when the orbital plane is orthogonal to the equatorial plane.

- Right ascension of the ascending node (Ω) defines the orientation of the orbital plane in the space. Specifically, it is the eastward angle between the vernal equinox and the ascending node. Ascending node is the intersection point of the orbit with the equatorial plane which satellite crosses from south to north (see Figure 2.1.9). For vernal equinox one can refer to [1, 2, 3, 12, 20, 33, 34].
- Argument of perigee (ω) indicates the orientation of the orbit in the orbital plane. The angle between the ascending node and the perigee point in the direction of satellite's motion. It should be noted that the perigee is defined as the closest point of the orbit to the prime focus of the ellipse (see Figures 2.1.8 and 2.1.9).
- True anomaly (ν) specifies the location of the satellite in the orbit. It is the angle between the perigee and the position of the satellite in the direction of the satellite's motion. Illustration of this angle is given in Figure 2.1.8. ν takes values from 0° to 360° .

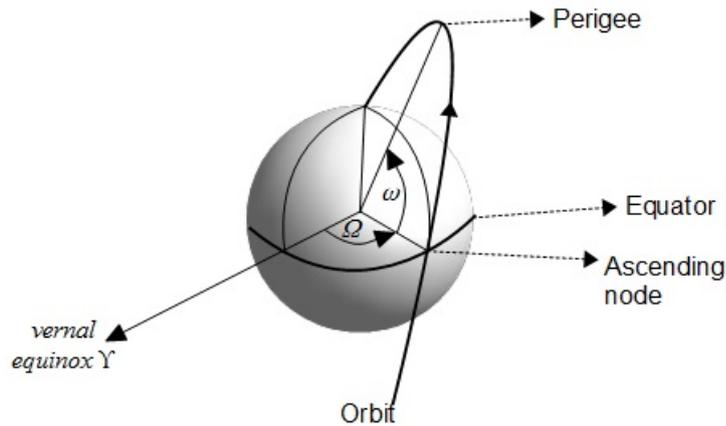


Figure 2.1.9: Representation of the right ascension and the argument of perigee.

Orbit determination algorithms used in this thesis are based on the three dimensional position and velocity vectors but they can be utilized in order to calculate

the orbital elements. Transformation between the orbital elements and the state vector used in thesis is beyond the scope of the thesis. For further information one can refer to [1, 2, 20, 24].

2.1.4.1 Two Line Element Set

Space objects up to altitudes approximately 8,000 km which have a radar cross section area of above a certain level are continuously tracked by the space surveillance network of the North American Aerospace Defence Command (NORAD) [20]. The resulting orbit information is distributed through the internet in Two Line Element (TLE) form . The information provided by the TLE is directly related (but not identical) to the classical orbital elements[1]. In addition to the orbital elements, some other information such as the time instant (i.e., the epoch which the given parameters belong to), launch date, mean motion, rate of change of the mean motion, Bstar term (drag-like parameter) etc. are also included in the TLE. A sample TLE format which is taken from the [35] for the weather satellite NOAA6 is represented in Figure 2.1.10. In Figure 2.1.10, left-

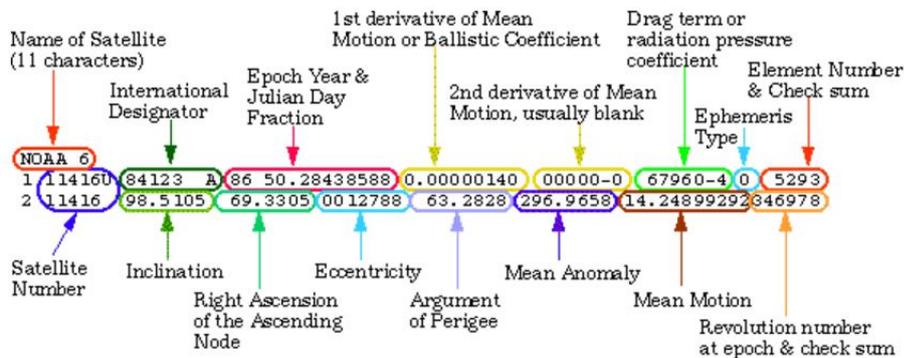


Figure 2.1.10: Two Line Element format and its explanation for the NOAA6 [35].

most 1 and 2 shows the line numbers. Satellite number part contains NORAD catalog number of the satellite and the classification indicator (U: unclassified, S: classified). International designator part consists of the parameters of the last two digits of the launch year, launch number and a character for the piece of the launch. Epoch part shows the time information of the given orbit parameters

and it includes last two digits of the year, day of the year and its fractions. The following part shows the first time derivative of the half of the mean motion in revolutions per day. In the 2nd derivative part, second time derivative of the mean motion divided by six is given. Bstar term (drag-like parameter) is given in the next part. Ephemeris type is related with the orbit model (i.e., the propagator). However in all of the distributed TLE's ephemeris type is 0 [36]. Next, the element number shows the count for the TLE. In the second line, some of the orbital elements are given in degrees. Instead of true anomaly, mean anomaly (circular orbit assumption) is given. The following part shows the mean motion (in revolutions per day) using circular orbit assumption. Revolution number is the count for the revolutions and finally the checksum digits for both lines are calculated using the numbers given in each line and it helps to check errors [36]. Explanations given above and further information about TLE format can be found in [1, 20, 36].

Since TLE is constructed in a certain way by removing the variations in the orbital elements, one should take this removed variations into account in order to make consistent predictions [37]. For this purpose, there are specific orbit propagators which are consistent with the TLE information such as Simplified General Perturbations (SGP) for near Earth orbiting objects and Simplified Deep Space Perturbations (SDP) for the objects having a revolution period of greater than a certain level [1, 37]. Using other orbit propagators with the TLE would probably yield bad results. For further information about the conversion between the TLE and orbital elements and implementation of the aforementioned TLE specific propagators, one can refer to [1, 20, 37].

It should be also mentioned that the TLE data does not contain accuracy information and is not appropriate for precision orbit determination purposes [1, 20].

2.2 Measurement Models

Position or position related data about satellite orbits can be acquired from different kinds of sensors such as spaceborne GNSS receivers, active or passive

tracking radar systems, optical telescopes, satellite laser ranging (SLR) systems with different order of accuracies. First of all, GNSS receivers provide both position and velocity data using satellite to satellite range measurements. In order to find the position of the receiver and the time offset (between a GNSS satellite and the receiver), at least four satellites of GNSS constellation must be visible by the receiver [20]. Global Positioning System (GPS) and GLONASS are examples of GNSS. Furthermore, radar systems usually supply range and Doppler

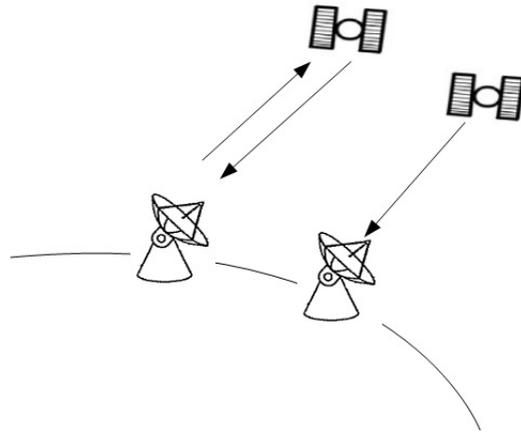


Figure 2.2.1: One way and two way radio tracking of satellites.

shift and they will also provide pointing angles (e.g., azimuth and elevation) if they have tracking capabilities. Range information is extracted from round trip time of the radio wave by looking at the phase shift between transmitted and received signal in general (see Figure 2.2.2). Moreover, signal tracking is done

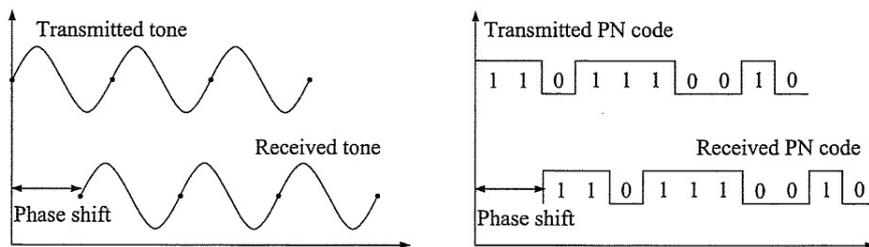


Figure 2.2.2: Basic principles of ranging via measuring phase shifts [3].

with lobe comparison techniques [38, 39]. If the tracking system is a receive only system i.e., it does not transmit radio waves (e.g., passive radar) then it will produce only angular and Doppler data (see Figure 2.2.1). The angular accuracy of radio based tracking depends on the antenna beamwidth (namely, diameter and frequency for reflector antennas) and signal to noise ratio (SNR)

of received signal [38, 39]. Usually, the accuracy of angular measurements are worse than the signal based measurements [1] since they are affected by physical issues such as calibration defects, thermal and mechanical distortions and loads [1, 3]. Also, angular measurements are less sensitive to small position changes. It is required to have a transmitter with stable (constant) frequency source in order to use Doppler techniques [24].

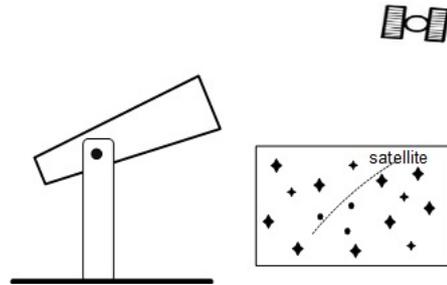


Figure 2.2.3: Satellite tracking telescope and symbolic camera image.

Optical telescopes provides only high precision angular data (and also the angular rates in some cases). Use of optical telescopes has a major role in satellite or celestial object tracking in history. Pointing angles with accuracies in the order of a few arcseconds ($1 \text{ arcsec} = 1/3600^\circ$) are provided by today's telescopes (actually telescope aided digital cameras). Telescope observations can not be obtained during daylight and they are also prone to weather conditions such as clouds and fogs. A sample illustration of an optical telescope can be seen in Figure 2.2.3.

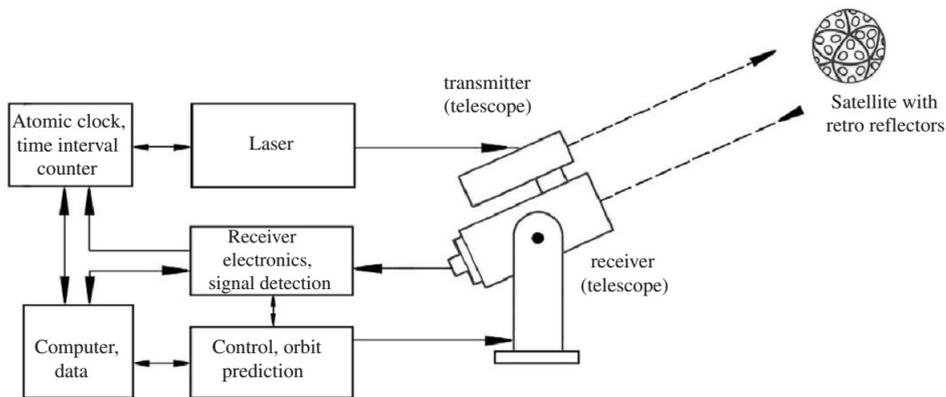


Figure 2.2.4: Satellite laser ranging system [24].

Finally, laser ranging systems give accurate range data using travelling time of photons produced by laser transmitter. This system requires the target satellite to be equipped with laser retro-reflectors in order to reflect incident photons back to the station [24]. They are also weather dependent [1]. SLR systems do not have autotracking mechanism as radars so they need high accuracy prior orbit information in order to correctly direct the laser transmitter beam to the satellite (see Figure 2.2.4 for laser ranging mechanism).

It should be emphasized that the ground based observation of LEO satellites with a limited number of observing stations is problematic since visibility durations are very short [3].

In the following sections, measurement equations are investigated in the following form.

$$Y_k = h(X_k) + \nu_k \quad (2.2.1)$$

where Y_k is the measurement vector corresponding state X_k , function $h(\cdot)$ is the mathematical relation between the states and measurements and ν_k is a random vector that shows the measurement noise. Depending on the observation type different $h(\cdot)$'s are utilized in measurement equations.

2.2.1 Full State Observation

If the satellite is equipped with GNSS receiver then all elements of the state can be acquired. GNSS sensors give the state in Earth fixed coordinate system and it should be converted to the inertial coordinate system since our dynamic equations were expressed in the inertial coordinate system. It should be mentioned that different navigation systems use different reference frames (e.g., WGS84 is used by GPS and PZ-90 is used by GLONASS [7]). Transformation between these reference frames will be investigated in Section 2.3. Assuming H_{full} is the mapping from inertial to fixed frame, measurement equation will be

$$Y_k = H_{full}X_k + \nu_k \quad (2.2.2)$$

where ν_k is the measurement noise of the GNSS receiver containing position and velocity noises. For example, a usual GPS receiver for a spacecraft provides position and velocity with a few tens of meters and a few tens of centimeters per second uncertainty respectively [24]. This kind of sensors also supply the quality of data with Dilution of Precision (DOP) values. One can use this quality measures as a weighting factor for measurement noise.

2.2.2 Angles Only Observation

In general angle observations are obtained as azimuth and elevation. Azimuth angle is defined as the clockwise angle from the true north and elevation is the angle between pointing vector and local tangential plane (see Figure 2.2.5). To

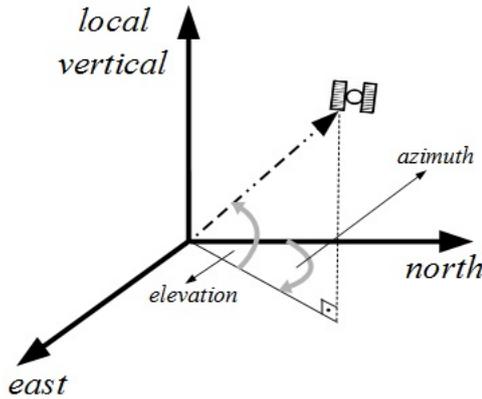


Figure 2.2.5: Azimuth and elevation angles on local coordinate system.

construct measurement equations, position components of the state vector (satellite position) should be converted to azimuth and elevation (see Section 2.3.3).

$$Y_k = h_{ae}(X_k) + \nu_k \quad (2.2.3)$$

where Y_k is the 2×1 measurement vector which consists of azimuth and elevation, namely $Y_k = [Az \quad El]^T$. Moreover, $h_{ae}(\cdot)$ is the vector valued function containing transformation functions from ECI to azimuth and elevation. Lastly the ν_k is the parameter representing angular measurement noise. For example, a 10 meter K_u band (14 GHz) antenna with monopulse tracking capability can measure pointing angles with an accuracy of approximately 3 milidegrees [3].

2.2.3 Angles and Doppler Observation

A receive only radio based tracking system provides angular and Doppler data related with the orbit of the satellite. Angle measurements are explained in Section 2.2.2. One way Doppler shift, ignoring the relativistic effects, caused by the relative motion of radio wave transmitter and receiver can be given as

$$\Delta f = -\frac{f_t}{c} \frac{dR}{dt}, \quad (2.2.4)$$

where Δf is Doppler frequency shift, f_t is the transmitted signal frequency, c is the speed of light ($2.99792458 \times 10^5 \frac{km}{s}$ [1]) and R is the distance between transmitter and receiver. If two way tracking system is used then Doppler shift equation should be modified and following equation will be obtained

$$\Delta f = -\frac{2f_t}{c} \frac{dR}{dt}. \quad (2.2.5)$$

In order to convert (2.2.4) into a useful form which can be used in a measurement equation, dR/dt should be written in terms of states (i.e. satellite position and velocity components). The distance R can be written as follows [12]

$$R = \sqrt{(x - x_g)^2 + (y - y_g)^2 + (z - z_g)^2} \quad (2.2.6)$$

then, time derivative of R will be

$$\frac{dR}{dt} = \frac{(x - x_g)(v_x - v_x^g) + (y - y_g)(v_y - v_y^g) + (z - z_g)(v_z - v_z^g)}{\sqrt{(x - x_g)^2 + (y - y_g)^2 + (z - z_g)^2}}, \quad (2.2.7)$$

where x, y, z, v_x, v_y, v_z are the components of the state vector in inertial frame and $x_g, y_g, z_g, v_x^g, v_y^g, v_z^g$ show the position and velocity components of ground station in inertial frame. Combining (2.2.4) and (2.2.7) gives us the final form of Doppler measurement that can be directly used in the measurement model.

$$\Delta f = -\frac{f_t [(x - x_g)(v_x - v_x^g) + (y - y_g)(v_y - v_y^g) + (z - z_g)(v_z - v_z^g)]}{c\sqrt{(x - x_g)^2 + (y - y_g)^2 + (z - z_g)^2}} \quad (2.2.8)$$

Combining angles only measurement equations with the above Doppler equation yields

$$Y_k = h_{aed}(X_k) + \nu_k \quad (2.2.9)$$

where $Y_k = [Az \quad El \quad \Delta f]^T$, $h_{aed}(\cdot)$ is the mapping from states to azimuth, elevation and Doppler shift and ν_k is the measurement noise.

Angular uncertainties have already been mentioned in the previous sections. In order to express the meaning of Doppler measurement uncertainty it is worth to use a sample scenario. 1 Hz of Doppler measurement accuracy at 10 GHz means an approximate range rate accuracy of 3 centimeters per second.

2.2.4 Angles and Range Observation

Angles only measurement equations are mentioned in Section 2.2.2 and the range measurement equation is given in (2.2.6). If we combine these angles only and range measurement equations, the resulting equation will be in the following form.

$$Y_k = h_{aer}(X_k) + \nu_k \quad (2.2.10)$$

where $Y_k = [Az \quad El \quad R]^T$. In the equation above, $h_{aer}(\cdot)$ is the mapping from states to azimuth, elevation and range and ν_k is the measurement noise. This kind of measurements are in general acquired from two way radars. Uncertainty levels for angle measurements can be taken into account as given in the previous sections. Range measurement uncertainties are in the order of a few tens of meters by conventional techniques (tone ranging, code ranging).

2.3 Reference Coordinate Systems and Transformations

A coordinate system is used to describe the location of a point or to define a vector in space by using numbers. In order to characterize a coordinate system its center, fundamental plane ($x - y$ plane), principal direction (direction of x axis) and direction of z axis should be specified. Positive direction for a coordinate system should also be defined [1, 2]. Right handed systems are used in general.

Reference system and frame concepts differ from each other. While, reference system is a conceptual definition of how to construct a coordinate system such as defining its origin, axes and fundamental planes, reference frame is a particular

realization of corresponding frame with defining coordinates of specific points that are directly accessible by observations [24, 40].

Cartesian (rectangular), spherical or oblate spheroidal coordinate systems are used in orbit determination problem in general [41]. Since the motion of the satellite of interest is primarily around the Earth, usually the origin of the desired coordinate system is the center of mass (geocenter) of the Earth [24] excluding topocentric coordinates. Cartesian coordinate system consists of an origin and three axes which are perpendicular to each other. Secondly, the spherical coordinate system can be defined as a three dimensional orthogonal coordinate system where a point is specified by a distance from origin and two angles. Finally, oblate spheroidal coordinate system is also an orthogonal coordinate system constructed by rotating an ellipse around its nonfocal axis [42].

In satellite orbit determination usually three reference coordinate frames are widely used [33]. First one is the Earth centric inertial (ECI) frame which is fixed in space. It is used in order to represent the orbit and it allows us the direct use of the Newton's laws. Second one is the Earth Centric Earth Fixed (ECEF) which is rotating with the Earth and used for defining the orbit in terms of geographic coordinates. The last one is the topocentric coordinate frame which is centered at the location of the observer and basically used to define the position of the satellite with respect to the observer.

It should be noted that ECI is not a true inertial frame and ECEF is not a true fixed frame. This is because, inertial frames are defined as non-rotating and unaccelerated. However, the motion of the Earth (also the motion of ECI system) about the sun yields a centripetal acceleration and as a result this reference frame is called "almost" or "quasi" or "approximate" inertial reference frame [2, 3, 12, 43]. Additionally, for fixed coordinates, due to mass deformation of Earth, movement of tectonic plates and tides, coordinates on Earth surface may change. Consequently, ECEF is not an actual fixed frame [12, 24].

2.3.1 Earth Centric Inertial Frame

ECI is an almost inertial frame and assumed to be fixed in space. Its origin is at the center of mass of Earth and its fundamental plane coincides with equatorial plane. Principal direction for this coordinate system is directed from origin to vernal equinox Υ . More information about vernal equinox can be found in [1, 2, 3, 12, 20, 33, 34]. The z_i axis which does not belong to the fundamental plane is along the rotation axis of the Earth. The last axis, y_i is located along 90° eastward of the x_i axis. Here the axes are named as x_i, y_i, z_i respectively. Illustration of ECI frame can be seen in Figure 2.3.1.

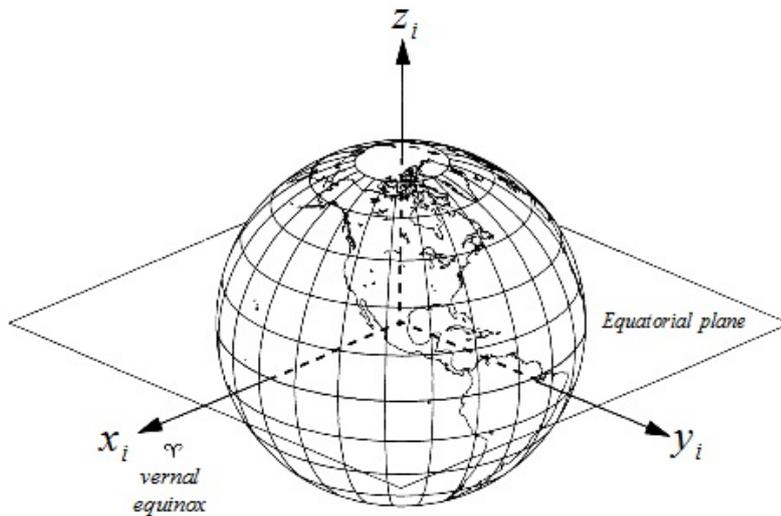


Figure 2.3.1: ECI Frame [1].

2.3.2 Earth Centric Earth Fixed Frame

ECEF is fixed to Earth and rotates. This frame is geocentric i.e., its origin is fixed to the geocenter (center of mass of the Earth). The second descriptive feature of this frame i.e., the fundamental plane is overlapped with again equatorial plane as it is in the ECI frame. However, its principal axis x_e is directed towards the zero degree meridian (i.e., the Greenwich meridian). Its z_e axis coincides with the rotation axis of Earth. Finally, the direction of y_e axis is chosen to make the system right handed. Furthermore, the coordinate system should be defined for a certain epoch since it is rotating [1].

2.3.3 Topocentric Coordinate Frames

This coordinate system is fixed to Earth and its origin is defined as the location of the observer or sensor. Fundamental plane of the topocentric coordinate system is formed by the local horizon [1] namely the tangential plane at the observer's location. It is useful to define both rectangular and spherical coordinate systems based on this origin and fundamental plane.

Firstly, let us define the rectangular coordinates. Although it is possible to choose different axes directions, East-North-Zenit (ENZ) and South-East-Zenith (SEZ) is commonly used. Here, only the ENZ coordinates will be explained but conversion is simple, the only thing to be done is to change the sign and order of the axes. Origin and fundamental plane for a topocentric coordinate system have already defined in previous paragraph. Moreover, x axis is directed towards the East, y axis points north and consequently z axis points to zenith (upwards from the local horizon) directions for ENZ coordinates. Let the axes names for ENZ coordinate system be x_t, y_t, z_t respectively for north, east and zenith. This coordinate system is an intermediate step in order to relate the observer based look angles and range with the geocentric Earth fixed frame.

Secondly, another useful topocentric system is a kind of spherical coordinate system with the same origin and the same fundamental plane which allows us to specify satellite position in terms of look angles and distance from observer's site. In general, satellite position is represented in terms of azimuth and elevation angles and distance between observer and satellite.

Finally, representation of observer coordinates is another important issue for Earth based observation case. The observer's location is commonly expressed by geodetic latitude ϕ , longitude λ and altitude h . If one knows the vector from the geocenter to observer and the vector from the observer to satellite then the vector from the geocenter to satellite would easily be constructed by adding the aforementioned vectors (vectors should be represented in the same coordinate system).

2.3.4 General Transformations

Consider a Cartesian coordinate system with origin O and axes x, y, z and a vector v representing point P in this coordinate system (see Figure 2.3.2). If we would like to represent this point in another coordinate system as v' with axes x', y', z' and with same origin O , we should apply series of rotations to this vector about its axes. Let the vector v be

$$v = \begin{bmatrix} x_p & y_p & z_p \end{bmatrix}^T$$

and let the transformed vector be

$$v' = \begin{bmatrix} x'_p & y'_p & z'_p \end{bmatrix}^T$$

and let the angles α, β, γ be the required rotations about x, y, z axes respectively. Then the relation between these vectors can be given by the equation below [2, 24, 32, 44].

$$\begin{bmatrix} x'_p \\ y'_p \\ z'_p \end{bmatrix} = R_1(\alpha)R_2(\beta)R_3(\gamma) \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} \quad (2.3.1)$$

where the rotations about x, y and z axis are given by the $R_1(\alpha), R_2(\beta)$ and $R_3(\gamma)$ matrices which are defined as follows [24].

$$R_1(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha \\ 0 & -\sin \alpha & \cos \alpha \end{bmatrix}, \quad R_2(\beta) = \begin{bmatrix} \cos \beta & 0 & -\sin \beta \\ 0 & 1 & 0 \\ \sin \beta & 0 & \cos \beta \end{bmatrix},$$

$$R_3(\gamma) = \begin{bmatrix} \cos \gamma & \sin \gamma & 0 \\ -\sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

It should be noted that the rotation order is important. Considering this issue from vector mathematics point of view, matrix multiplication is not commutative that is

$$AB \neq BA$$

except for some special conditions on A and B matrices [2]. Moreover, rotation about reverse directions can be carried out by simply changing the angle term with its negative and this corresponds to taking the transpose of the rotation matrix (i.e., $R_i(-angle) = R_i^T(angle)$ for $i = 1, 2, 3$). In addition to rota-

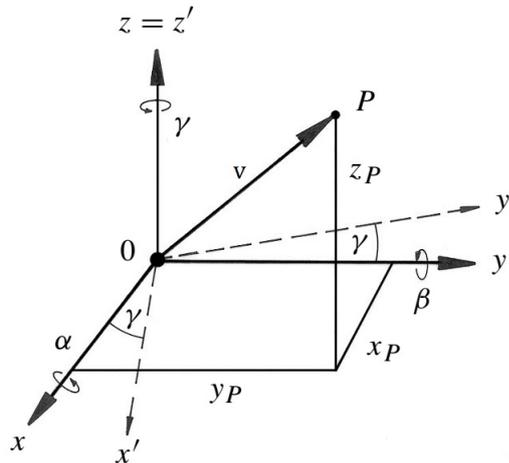


Figure 2.3.2: Rotation of Cartesian coordinate system [24].

tion, there is also an operation called translation. It can be simply interpreted as changing the origin of the coordinate system without any rotation (see Figure 2.3.3). This concept is relatively more straightforward than the rotation case [34]. In order to translate a coordinate system, the only thing to do is shifting the origin with the required amount. Assume, we have a general Cartesian coordinate system represented by its origin and axes by $(O; x, y, z)$. Let (x_p, y_p, z_p) shows the position of a point P in this coordinate system. If it is needed to represent this point in a new coordinate system with origin O' at (x_1, y_1, z_1) then the result will be $((x_p - x_1), (y_p - y_1), (z_p - z_1))$.

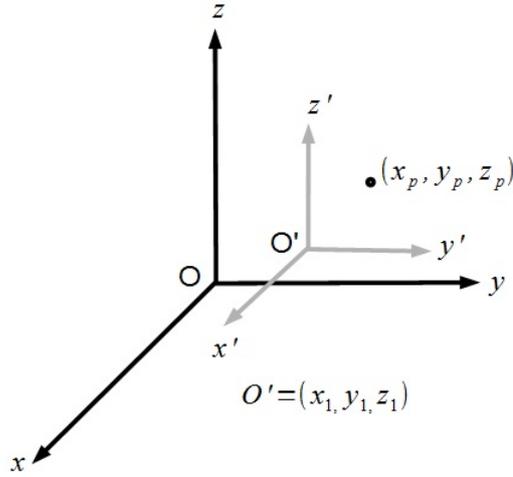


Figure 2.3.3: Coordinate system translation.

2.3.5 ECI - ECEF Transformation

In order to use measurement equations, conversion from ECI to ECEF is necessary since the state vector is represented in ECI and measurements are represented in Earth fixed coordinates. Moreover, in order to calculate gravitational perturbation acceleration which depends on the satellite location above the Earth, both ECI to ECEF and ECEF to ECI conversion are required. This is because the acceleration expression requires the satellite position to be represented in ECEF and the equations of motion are represented in ECI.

For the simplified case, ignoring all rotational variations (precession, nutation and polar motion that will be briefly explained later in this section) and assuming that the only difference between these frames is the rotation about their common axis z , transformation is simply represented by the following equation [1]

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{ecef} = T(\theta) \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{eci} \quad (2.3.2)$$

where the matrix $T(\theta)$ shows the rotation about z axis and is given by

$$T(\theta) = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and θ is the angle (at certain epoch) between the vernal equinox vector and the vector passing through Greenwich Meridian, in other words, it is the angle between x axes of two frames which are x_i and x_e (see Figure 2.3.4). This angle is called as Greenwich Apparent Sidereal Time (GAST). Furthermore, the inverse rotation (i.e., the rotation from ECEF to ECI) is simply carried out by the matrix multiplication by changing the sign of the angle θ in the transformation matrix.

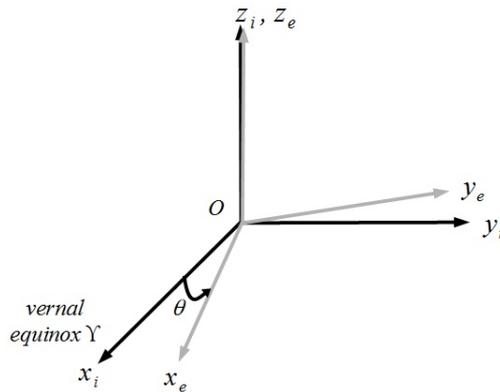


Figure 2.3.4: ECI and ECEF representation.

It is also required to convert velocity vectors. However, for this case (2.3.2) can not be directly used simply by replacing the position vectors with velocity vectors. There are extra terms coming from the differentiation of a position vector in a rotating frame.

Time derivative of a position vector r with respect to inertial frame can be written as the sum of two components. First one is the time derivative of r with respect to rotating frame and the other is the velocity term caused by the rotation. Assuming that the rotating frame revolves with an angular velocity of $\omega = [\omega_x \ \omega_y \ \omega_z]^T$ where its elements are angular velocities about x, y, z axes,

we can write the result as follows [2].

$$\left. \frac{dr}{dt} \right|_i = v_i = v_r + \omega \times r \quad (2.3.3)$$

where v_i and v_r are the velocity vectors represented in inertial and rotating frames respectively. Let the position vector be r_{eci} and r_{ecef} in inertial and rotating frames respectively and let the relation between these vectors be

$$\begin{aligned} r_{ecef} &= T(\theta) r_{eci} \\ r_{eci} &= T(-\theta) r_{ecef} \end{aligned}$$

then the velocity vectors in ECI and ECEF can be represented as follows.

$$\begin{aligned} v_{ecef} &= T(\theta) v_{eci} - \omega_e \times T(\theta) r_{eci} \\ v_{eci} &= T(-\theta) v_{ecef} + \omega_e \times T(-\theta) r_{ecef} \end{aligned} \quad (2.3.4)$$

in (2.3.4), the term $-\omega_e \times T(\theta)$ comes from the time derivative of $T(\theta)$ where ω_e is the angular velocity vector of the Earth (about its axis) and v_{eci}, v_{ecef} are velocities in ECI and ECEF respectively. Simply, the angular velocity of Earth is about z axis and the value for this angular velocity is $\sim 7.292115 \times 10^{-5} \frac{rad}{s}$ [24]. In vector form, we have

$$\omega_e = \begin{bmatrix} 0 & 0 & 7.292115 \times 10^{-5} \end{bmatrix}^T$$

and the cross product operation for a vector $v = \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix}^T$ is defined as

$$v \times = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_3 & 0 \end{bmatrix}.$$

In practice, the topic of ECI - ECEF conversion does not only include the regular Earth rotation about its z axis. There are also other effects called precession, nutation and polar motion. Due to equatorial bulge of the Earth, the Sun and the Moon produce a gravitational torque and this causes luni-solar precession (see Figure 2.3.5) with a period of $\sim 26,000$ years [1]. Moreover, the Moon also causes a torque resulting in small variations superimposed on the precession effect called nutation with a period of ~ 18.6 years [1]. Finally, the polar motion is

explained as the motion of the Earth's spin axis with respect to Earth fixed frame [23]. In order to do a precise transformation these effects should be included. However, in this thesis only rotation about z axis will be considered since the most significant change is caused by the Earth rotation [1]. In order to visualize the effect of precession, nutation and polar motion one can refer to [45] and more information about these effects can be found in [1, 2, 3, 23, 24, 40].

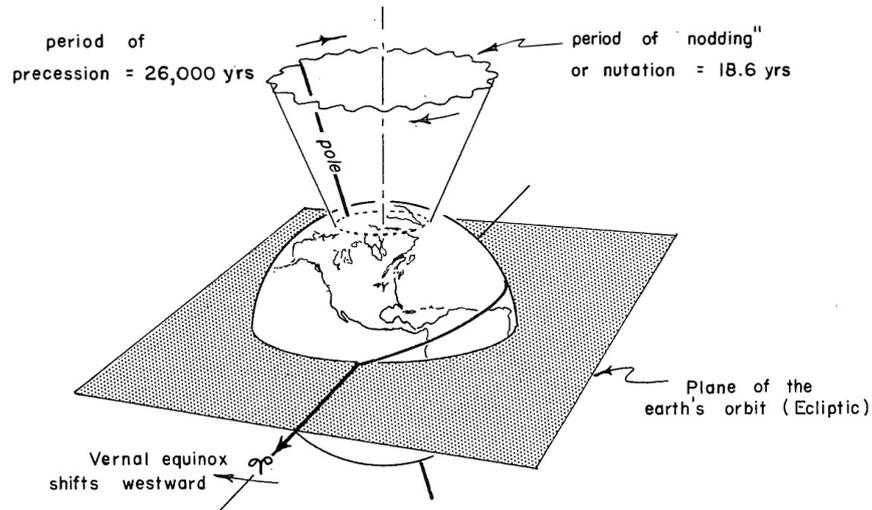


Figure 2.3.5: Illustration of precession and nutation [2].

2.3.5.1 Time Expressions and Their Relations

In order to find the transformation matrix between ECI and ECEF, GAST is our main interest but it is useful and also necessary to mention about the time systems and their relations.

For satellite orbit determination (especially, precise orbit determination), accurate time tagging of measurements is required. This is because, the velocity of a satellite is generally in the order of a few kilometers per second and 1 millisecond of timing uncertainty causes a few meters of position uncertainty. Today, different time systems are used for different purposes and for different regions. Hence, awareness of the utilized time system (i.e., the system which measurement time tags are given in) is required and one should know the relation between these time systems. Another reason that necessitates the relation

between time expressions is the need for coordinate transformations. As we mentioned earlier in Section 2.3.5 some coordinate systems are fixed while some others are rotating in space. Transformation between these kind of coordinate systems requires the knowledge of absolute time starting from a certain reference. This is because, the orientation of a rotational frame is known at certain epoch and it is desired to find the orientation of this frame at given epoch. If the timing is incorrect then the resulting coordinates will also incorrect.

A commonly used definition for time is solar day and it is defined as the required time of successive passes of the Sun over a reference meridian [1]. It varies due to the apparent motion of the Sun. A less variant time definition is sidereal day and it is defined as the time for Earth to rotate on its axis once relative to stars [2]. In other words, sidereal time is a system of keeping the time with respect to stars [46]. Sidereal day is approximately 4 minutes shorter than the solar day [43]. If a local meridian is assumed as a reference it will be named as Local Sidereal Time (LST) and if this local meridian is zero degree meridian then it is called Greenwich Sidereal Time (GST). In this context, time refers to the angle between a reference meridian and the vernal equinox. Since apparent location of the stars changes due to Earth's rotational variations (precession, nutation and polar motion), sidereal time changes a little. Considering this condition, apparent and mean sidereal time concepts arises. While, the first one takes true location of vernal equinox, second one uses mean location of vernal equinox [23]. They are called Greenwich Apparent Sidereal Time (GAST) and Greenwich Mean Sidereal Time (GMST) respectively if the observation point is assumed to be the zero degree meridian. Another time representation system is Universal Time (UT) which can be explained as the mean solar time of Greenwich Meridian. For astronomical and navigational use, more accurate form of the Universal Time is needed and it is obtained by astronomical observations which is called UT1. Moreover, commonly used time system in everyday life is Coordinated Universal Time (UTC). UTC is kept by several laboratories in the world with precise atomic clocks and one second is defined by the atomic transition of cesium element in certain conditions [47]. UT1 is completely identified by the rotation of the Earth but UTC is a human made system and the

difference between them is not allowed to exceed 0.9 seconds [47]. If one do not need an accuracy better than one second then UTC and UT1 are replacable. Another common time representation technique (used in ECI-ECEF coordinate transformation) is Julian Days (JD). It is the count of the days and fractions for the Greenwich meridian starting at 12:00, January 1st, 4713 BC [40]. A Julian year contains exactly 365.25 Julian days by definition [23]. Let Y , M and D be the numbers representing year, month and day with fractions of a certain calendar date respectively. Then the Julian Day for this calendar date can be calculated by using the following algorithm [48].

- If $M \leq 2$ replace $Y = Y - 1$ and $M = M + 12$, otherwise Y and M remains unchanged
- In Gregorian Calendar $A = \lfloor \frac{Y}{100} \rfloor$ and $B = 2 - A + \lfloor \frac{A}{4} \rfloor$, in Julian Calendar $B = 0$ (generally Gregorian Calendar is used)
- Day's fraction: $\frac{hours \times 1440 + minutes \times 60 + seconds}{86400}$

Then, corresponding Julian Days will be calculated as

$$JD = \lfloor 365.25(Y + 4713) \rfloor + \lfloor 30.6001(M + 1) \rfloor + D + B - 1524.5 \quad (2.3.5)$$

where $\lfloor x \rfloor$ is the floor function that gives the greatest integer that is less than or equal to x . After calculation of Julian Days, GMST should be calculated. If GMST is known for a particular time t_0 (say $GMST(t_0)$) then any future GMST at time t can be calculated by the following approach since we know the Earth's approximate rotation velocity [2].

$$GMST(t) = GMST(t_0) + \Omega \times 360 \times D \quad (^\circ) \quad (2.3.6)$$

where $\Omega = 1.0027279093$ is the number of rotations in a solar day and D is the time difference in terms of days and fractions (i.e., $t - t_0$). $GMST(t_0)$ can be calculated by the following expression if and only if t_0 is in the form of 0^hUT .

$$GMST(t_0) = 100.460618137 + 36000.770053608 T + 0.000387903 T^2 - T^3/38710000 \quad (^\circ) \quad (2.3.7)$$

where T is given by the following relation (JD should be calculated for t_0).

$$T = \frac{JD - 2451545.0}{36525}.$$

Moreover, GMST at any given UT is further calculated by the following formula regardless of reference time [48].

$$\begin{aligned} GMST(t) = & 280.46061837 + 360.98564736629(JD - 2451545.0) \\ & + 0.000387933T^2 - T^3/38710000 \quad (^\circ). \end{aligned} \quad (2.3.8)$$

In addition, more accurate calculations can be carried out by taking the precession, nutation and polar motion into account. If all are used, the required angle turns out to be GAST and can be calculated by the following formula [1, 3, 48].

$$GAST = GMST + \Delta\psi \cos \epsilon \quad (2.3.9)$$

where $\Delta\psi$ is the nutation in longitude and ϵ is the true obliquity of the ecliptic [48]. Since we have ignored all these effects, GMST is used instead of GAST in our calculations.

2.3.6 Lat-Lon-Alt to ECEF Transformation

In order to find an expression for the range between the satellite and the observer it is needed to transform observer coordinates which are usually represented by geodetic latitude (ϕ), longitude (λ) and altitude (h) to the ECI coordinates. This transformation can be divided into two steps. First one is the conversion from geodetic coordinates (ϕ, λ, h) to ECEF since they are both Earth fixed frames and secondly, the transformation from ECEF to ECI which is mentioned in Section 2.3.5.

According to [23, 24, 40, 43] the transformation between geodetic coordinates and ECEF is given by the following equation,

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{ecef} = \begin{bmatrix} (N + h) \cos \phi \cos \lambda \\ (N + h) \cos \phi \sin \lambda \\ [N(1 - e^2) + h] \sin \phi \end{bmatrix} \quad (2.3.10)$$

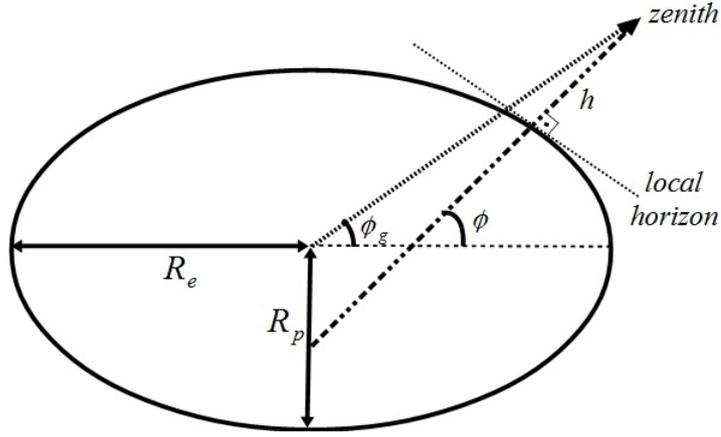


Figure 2.3.6: Geodetic latitude ϕ and geocentric latitude ϕ_g on reference ellipsoid.

where N is the radius of the curvature in the prime vertical [24] and it is given as follows [3].

$$N = \frac{R_e}{\sqrt{1 - e^2 \sin^2 \phi}}$$

In the equation above, e is the first eccentricity and R_e is the semimajor axis of the reference ellipsoid (i.e., the equatorial radius of the Earth). Furthermore, the first eccentricity can be expressed in terms of flattening f as $e = \sqrt{f(2 - f)}$ [12, 24, 43]. Finally, flattening of the reference ellipsoid is given by the ratio of difference between semimajor axis R_e and semiminor axis R_p (see Figure 2.3.6) to R_e which is given by the below relation [43].

$$f = \frac{R_e - R_p}{R_e}.$$

One of the commonly used ellipsoid is World Geodetic System 1984 (in short WGS84) model and it gives the values for required the parameters as $R_e = 6378.137 \text{ km}$ and $f \approx 1/298.257$ [40, 43].

2.3.7 ECEF to AER Transformation

Ground based measurement equations use satellite position and velocity (i.e., the state vector) as an input and find look angles, range, range rate or Doppler shift. It is necessary to find a transformation expression which converts satellite position from ECI to azimuth (Az), elevation (El) and range (R). Furthermore,

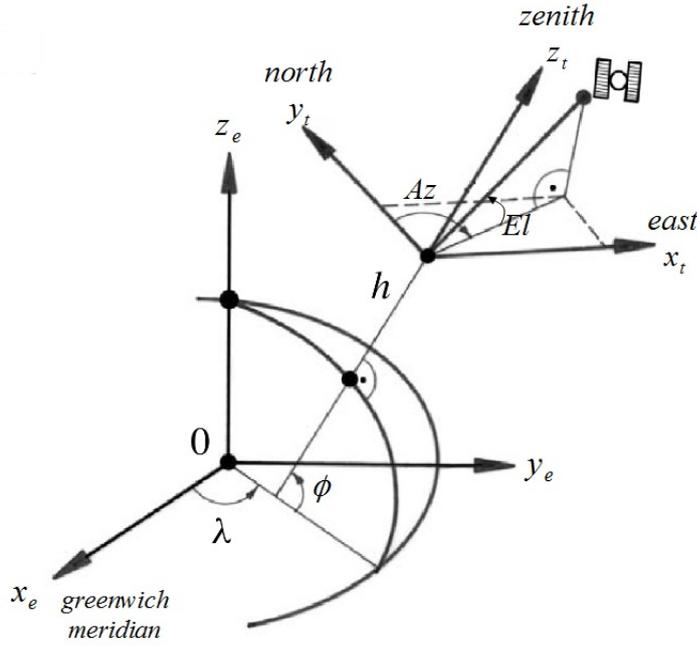


Figure 2.3.7: Visualization of ECEF, ENZ and geodetic coordinates [24].

in order to initialize the estimator, first few AER (azimuth-elevation-range) measurements should be converted to a state vector form in ECI. Therefore, AER to ECEF transformation is also needed.

Since look angles and distance are defined in an Earth fixed frame, the first thing to be done is to convert satellite position from ECI to ECEF which has been explained earlier. After representing the position of the satellite in ECEF, observer coordinates usually given in terms of geodetic latitude, longitude and altitude should also be expressed in ECEF frame. This operation was also clarified in the previous section. Afterwards, it is possible to use vector mathematics since all vectors are represented in ECEF frame. Let the observer's position vector be $\vec{\rho}_o$ and satellite position vector be $\vec{\rho}_s$ (see Figures 2.3.7 and 2.3.8). Then the satellite position vector with respect to observer can be written as below.

$$\vec{R} = \vec{\rho}_s - \vec{\rho}_o. \quad (2.3.11)$$

In order to relate \vec{R} with azimuth and elevation we should apply a transformation from ECEF to ENZ frame as follows.

$$\vec{R}_{ENZ} = T_{ECEF}^{ENZ} \vec{R} \quad (2.3.12)$$

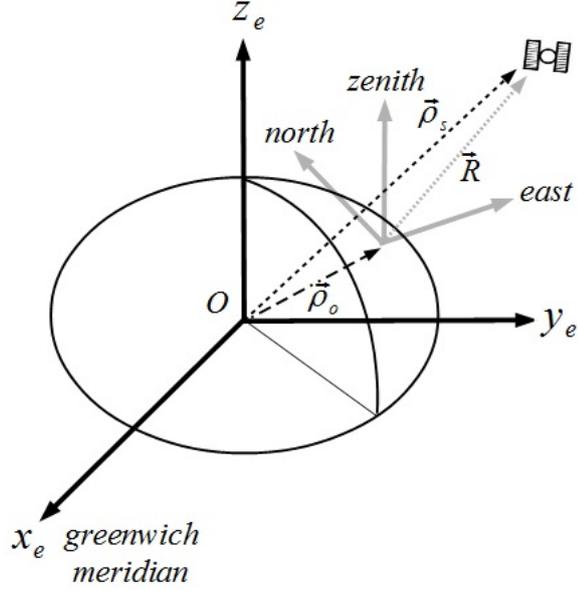


Figure 2.3.8: Vector representation of observer and satellite in Earth fixed frame.

where T_{ECEF}^{ENZ} is the rotation matrix from ECEF to ENZ and it is explained as first rotation about z axis by $90 + \lambda$ degrees where λ is the observer's longitude and second rotation about x axis by $-\phi$ degrees where ϕ is the observer's latitude then change the sign of y axis and interchange the rows corresponding y and z axes. This can be given by the following matrix operation.

$$T_{ECEF}^{ENZ} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \sin(-\phi) & \cos(-\phi) \\ 0 & -\cos(-\phi) & \sin(-\phi) \end{bmatrix} \begin{bmatrix} \cos(90 + \lambda) & \sin(90 + \lambda) & 0 \\ -\sin(90 + \lambda) & \cos(90 + \lambda) & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Carrying out the matrix multiplication shown above gives the result in closed form [49] as

$$T_{ECEF}^{ENZ} = \begin{bmatrix} -\sin \lambda & \cos \lambda & 0 \\ -\sin \phi \cos \lambda & -\sin \phi \sin \lambda & \cos \phi \\ \cos \phi \cos \lambda & \cos \phi \sin \lambda & \sin \phi \end{bmatrix}. \quad (2.3.13)$$

As a result, one can find the coordinate of the satellite in ENZ frame by the

below equation letting $\begin{bmatrix} x_t & y_t & z_t \end{bmatrix}^T$ represents the vector form of the \vec{R}_{ENZ}

$$\vec{R}_{ENZ} = \begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix} = \begin{bmatrix} -\sin \lambda & \cos \lambda & 0 \\ -\sin \phi \cos \lambda & -\sin \phi \sin \lambda & \cos \phi \\ \cos \phi \cos \lambda & \cos \phi \sin \lambda & \sin \phi \end{bmatrix} \begin{bmatrix} x_s - x_o \\ y_s - y_o \\ z_s - z_o \end{bmatrix} \quad (2.3.14)$$

where $\begin{bmatrix} x_s & y_s & z_s \end{bmatrix}^T$ is the vector representation of the satellite position in ECEF ($\vec{\rho}_s$) and $\begin{bmatrix} x_o & y_o & z_o \end{bmatrix}^T$ is the vector representation of the observer position in ECEF ($\vec{\rho}_o$).

Now, the satellite position can be represented in terms of observer look angles and R by the following equation simply by using the spherical to rectangular coordinate transformation, assuming R is the magnitudes of the vectors \vec{R}_{ENZ} and \vec{R} . The relation between ENZ coordinates, look angles and distance will be

$$\vec{R}_{ENZ} = \begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix} = \begin{bmatrix} R \cos El \sin Az \\ R \cos El \cos Az \\ R \sin El \end{bmatrix}, \quad (2.3.15)$$

hence the following expression will be obtained.

$$\begin{bmatrix} \cos El \sin Az \\ \cos El \cos Az \\ \sin El \end{bmatrix} = \begin{bmatrix} \frac{x_t}{R} \\ \frac{y_t}{R} \\ \frac{z_t}{R} \end{bmatrix}. \quad (2.3.16)$$

After making some trigonometric manipulations, formulation for azimuth and elevation angles will be as follows.

$$Az = \tan^{-1} \left(\frac{x_t}{y_t} \right), \quad (2.3.17)$$

$$El = \sin^{-1} \left(\frac{z_t}{R} \right).$$

AER to ECEF conversion is simply the reverse of the above processes. First,

AER to ENZ conversion is needed and this can be done by using right hand side of (2.3.15). Then, multiplication of the position vector represented in ENZ with the transpose of T_{ECEF}^{ENZ} given in (2.3.13) from the left side will give the position vector in ECEF.

2.3.8 Az-El to Topocentric Ra-Dec Transformation

A convenient way to represent satellite position with respect to the observer is to use topocentric right ascension (Ra) and declination (Dec). Origin of this coordinate system is at observer's location and it uses set of axes fixed in space and coincide with the axes of ECI, namely it can be treated as translated version of ECI. $Ra(\alpha)$ and $Dec(\delta)$ is shown in Figure 2.3.9. It is useful since the vector operations require vectors to be represented in the same coordinate system. Let ϕ and λ be the latitude and longitude of the observer's location and θ_{LST}

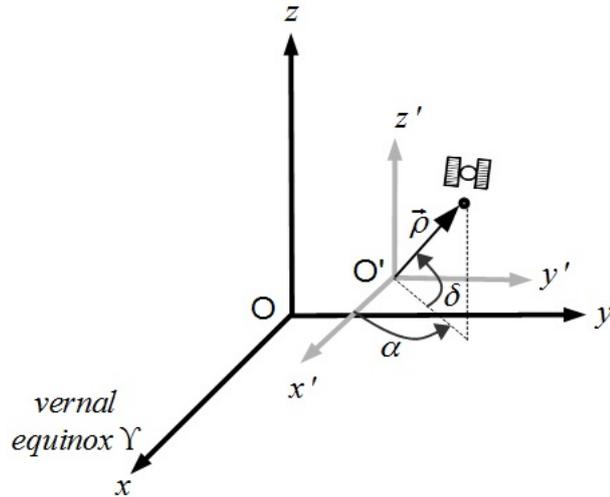


Figure 2.3.9: Visualization right ascension α and declination δ with ECI frame.

be the local sidereal time at the observer's longitude. θ_{LST} can be found by the following relation.

$$\theta_{LST} = \theta_{GAST} + \lambda \quad (2.3.18)$$

where θ_{GAST} is the Greenwich apparent sidereal time has been investigated in Section 2.3.5. Moreover, in order to convert Az and El to topocentric Ra and Dec , firstly, azimuth and elevation should be projected into ENZ frame. Applying series of multiplication by rotation matrices, the relation between them

is constructed by the following equation assuming $\theta = \theta_{LST}$ [32].

$$\begin{bmatrix} \cos \alpha \cos \delta \\ \sin \alpha \cos \delta \\ \sin \delta \end{bmatrix} = \begin{bmatrix} -\sin \theta & -\cos \theta \sin \phi & \cos \theta \cos \phi \\ \cos \theta & -\sin \theta \sin \phi & \sin \theta \cos \phi \\ 0 & \cos \phi & \sin \phi \end{bmatrix} \begin{bmatrix} \sin Az \cos El \\ \cos Az \cos El \\ \sin El \end{bmatrix}. \quad (2.3.19)$$

After carrying out the matrix multiplication and some trigonometric manipulations right ascension and declination will be found as

$$\begin{aligned} \alpha &= \theta - \theta_h, \\ \delta &= \sin^{-1} (\cos \phi \cos Az \cos El + \sin \phi \sin El), \end{aligned} \quad (2.3.20)$$

where θ_h is hour angle and defined as follows.

$$\theta_h = \begin{cases} 360^\circ - \cos^{-1} \left(\frac{\cos \phi \sin El - \sin \phi \cos Az \cos El}{\cos \delta} \right), & 0^\circ < Az < 180^\circ \\ \cos^{-1} \left(\frac{\cos \phi \sin El - \sin \phi \cos Az \cos El}{\cos \delta} \right), & 180^\circ \leq Az \leq 360^\circ \end{cases}.$$

2.4 State Estimation and Smoothing

In this section, Kalman filters for discrete time systems represented by nonlinear equations will be considered. Moreover, an estimation algorithm for dynamic systems which are represented by continuous time state equation and discrete time measurement equation will also be mentioned. Finally, the smoothing concept for nonlinear estimation problems will be investigated.

Estimation is the procedure of deducing the value of a variable or variables that we would like to know by using indirect and uncertain measurements [50]. An estimator is a function of measurements which gives the values of the desired variables. Large class of dynamic systems are described by differential equations resulting from the use of physical laws and they can be represented in state space form. If our interest is to find the state of a dynamic system then the procedure is called state estimation. State estimation is carried out sequentially in general and state can be considered as a random parameter to be estimated since it is affected by random noise components. As a result, state estimation scheme is said to be Bayesian.

It is always desired to find an optimal estimator. Optimal estimators use the measurements, models and their statistical properties in an ideal way in some sense. Usually, a cost function based on estimation error is desired to be minimized. Mean square estimation error is a common optimality criterion. Moreover, if an optimal estimator does not exist or can not be implementable, suboptimal solutions are applied [51]. The optimal estimator in the sense of minimum mean square error (MMSE) for the Bayesian case is the mean of the conditional probability density function given all the available information.

A widely used state estimator is Kalman filter. It is based on the state space representation of linear dynamic systems and it provides optimal estimate in the sense of mean square error in a recursive way [52]. In order to guarantee the optimality of Kalman filter, several conditions on the dynamic system and noise components must be satisfied. First, the system of interest must be linear. Then, the noise components and the initial estimate must have Gaussian distribution and they must be uncorrelated. If the Gaussianity assumption is not satisfied, Kalman filter is still the best linear estimator. Namely, it gives the best mean square error performance when compared to other linear estimators [50, 53]. Although Kalman filter was originally developed for linear problems, it can be applied to nonlinear problems by linear approximations [54]. Nonlinear versions of Kalman filter are examples of suboptimal estimation algorithms.

Kalman filter utilizes both measurements and dynamic model in an optimal way. Namely, if the measurement is bad (i.e., measurement noise is high) then the filter assigns more importance to the dynamic model by decreasing the Kalman gain and in the reverse situation (i.e., modelling uncertainties are high) it increases the gain. It can also be concluded that Kalman gain determines how much of the new information (innovation) will be used.

Let us consider the following linear, time invariant dynamic system.

$$\begin{aligned} x_{k+1} &= F x_k + \omega_k \\ y_k &= H x_k + \nu_k \end{aligned} \tag{2.4.1}$$

where x_k and y_k are $n \times 1$ state and $m \times 1$ measurement vectors respectively. ω_k and ν_k are zero mean, white Gaussian noise components with covariance matrices Q and R . These covariance matrices are defined as follows

$$E \{ \omega_j \omega_k^T \} = \begin{cases} Q, & \text{if } j = k \\ 0, & \text{otherwise} \end{cases}$$

$$E \{ \nu_j \nu_k^T \} = \begin{cases} R, & \text{if } j = k \\ 0, & \text{otherwise} \end{cases}.$$

Let the superscripts "–" and "+" represent respectively the predicted and updated forms of state vector and covariance matrix. Kalman filter equations are summarized in Table 2.4.1 and its update mechanism is described in Figure 2.4.1.

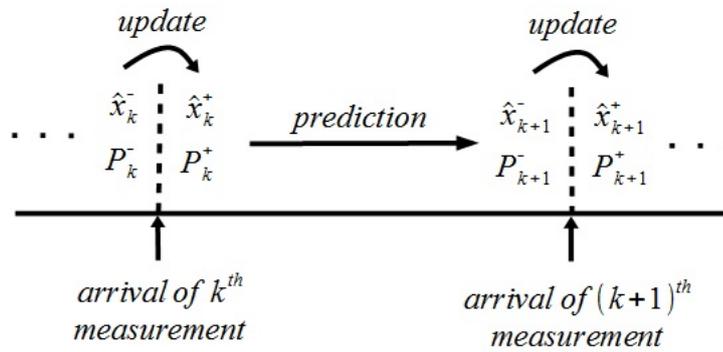


Figure 2.4.1: Update and prediction mechanism of Kalman filtering.

In general, tuning of a Kalman filter (selection of appropriate Q and R matrices) is difficult and time consuming [19]. Usually, filter designer have much more information about R due to the known statistical characteristics of the sensors. However, in general there is little information about Q and a good choice of process noise covariance depends on the filter designers experience and knowledge about the system dynamics [19]. After choosing a considerable Q and running the filter several times with different choices, one may decide a good value for Q by trial and error.

Table 2.4.1: Equations for standard Kalman filter.

Initialization	\hat{x}_0^- and P_0^-
Innovations covariance	$S_k = H P_k^- H^T + R$
Kalman gain	$K_k = P_k^- H^T S_k^{-1}$
State update	$\hat{x}_k^+ = \hat{x}_k^- + K_k (y_k - H \hat{x}_k^-)$
Covariance update	$P_k^+ = (I - K_k H) P_k^-$
State prediction	$\hat{x}_{k+1}^- = F \hat{x}_k^+$
Covariance prediction	$P_{k+1}^- = F P_k^+ F^T + Q$

Finally, different implementations of Kalman filter is used in the literature such as information filter, square root filter due to computational issues. They make computations cheaper and/or precise respectively. For example, information filter utilizes the inverse of covariance matrix (P^{-1}) and reduces the computational load under certain conditions [53]. Moreover, square root filter decomposes covariance matrix using a factorization method and uses the resulting components instead of the covariance matrix. This makes Kalman filter algorithm more robust and less sensitive to numerical errors. For more information about these implementation methods one can refer to [50, 53, 54, 55, 56].

2.4.1 Extended Kalman Filter

When the state and/or measurement equations are nonlinear, Kalman filter given in Table 2.4.1 can not be used directly. Instead, one can approximate these nonlinear equations using first order linearization around the estimated states and then uses standart Kalman filter. This procedure is called Extended Kalman Filtering (EKF). Many dynamic systems and sensors with practical value have nonlinear characteristics but they are approximately linear for small

perturbations around the state [54].

EKF has remained the most popular method for nonlinear estimation problems and other estimator forms are investigated only when the EKF does not give satisfactory results [19].

$$\begin{aligned}x_{k+1} &= f(x_k) + \omega_k \\y_k &= h(x_k) + \nu_k\end{aligned}\tag{2.4.2}$$

For the nonlinear dynamic system described in (2.4.2), EKF can be implemented by using the following algorithm.

1. Initialize the filter with prior information about the state by

$$\hat{x}_0, \quad P_0 = E \{ \tilde{x}_0 \tilde{x}_0^T \}$$

where \tilde{x}_0 is the initial estimation error (i.e., $\tilde{x}_0 = x_0 - \hat{x}_0$). Initial conditions can be considered as either updated (\hat{x}_0^+) or predicted (\hat{x}_0^-) states depending on how these values are obtained and for which time interval they are valid. If they are considered as updated states, filtering will start with prediction. Otherwise, the first thing to start the filter is updating the state and covariance matrix. Here, initial conditions are assumed as predicted.

2. Calculate Kalman gain according to

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R)^{-1}\tag{2.4.3}$$

where H_k is the Jacobian of the vector valued measurement function $h(x)$ evaluated at $x = \hat{x}_k^-$. Numerical calculation of Jacobians will be mentioned later in this section.

3. Carry out state and covariance updates by

$$\begin{aligned}\hat{x}_k^+ &= \hat{x}_k^- + K_k (y_k - h(\hat{x}_k^-)) \\P_k^+ &= (I - K_k H_k) P_k^-\end{aligned}\tag{2.4.4}$$

4. Propagate state and covariance matrix by the following equations respectively until a new measurement (y_{k+1}) arrives.

$$\begin{aligned}\hat{x}_{k+1}^- &= f(\hat{x}_k^+) \\ P_{k+1}^- &= F_k P_k^+ F_k^T + Q\end{aligned}\tag{2.4.5}$$

where F_k is the Jacobian of state equations vector $f(x)$ evaluated at $x = \hat{x}_k^+$.

Since propagation and update equations need F_k and H_k , namely the Jacobians of $f(x)$ and $h(x)$ around the estimated and predicted state vector respectively, partial derivatives of them are required. However, calculation of partial derivatives in closed form is problematic if the elements of $f(\cdot)$ and $h(\cdot)$ are complex. Instead of finding partial derivatives analytically, numerical methods can be used. For an $n \times 1$ vector valued function $\begin{bmatrix} f_1(x) & f_2(x) & \dots & f_n(x) \end{bmatrix}^T$, numerical partial derivative for one of its element $f_1(x)$, assuming x is scalar, can be approximated as follows [57]

$$\frac{\partial f_1(x)}{\partial x} \approx \frac{f_1(x + \epsilon) - f_1(x - \epsilon)}{2\epsilon}\tag{2.4.6}$$

where ϵ is a small scalar. If x is an n dimensional vector (i.e., $x = \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}^T$) then the partial derivative of f_1 with respect to x_i , namely the i^{th} element of x can be approximated as follows.

$$\frac{\partial f_1(x)}{\partial x_i} \approx \frac{f_1\left(\begin{bmatrix} x_1 \\ \vdots \\ x_i + \epsilon \\ \vdots \\ x_n \end{bmatrix}\right) - f_1\left(\begin{bmatrix} x_1 \\ \vdots \\ x_i - \epsilon \\ \vdots \\ x_n \end{bmatrix}\right)}{2\epsilon}.\tag{2.4.7}$$

It should be noted that ϵ is added to or subtracted from only the variable of interest, namely x_i and other components of x remain unchanged. In (2.4.6) and (2.4.7) central finite differencing method is used. For further information about numerical differentiation schemes one can refer to [57, 58, 59]. In this thesis, all the simulations (which require to take Jacobian) are carried out by using numerical Jacobians as given in (2.4.7).

2.4.2 Continuous-Discrete Extended Kalman Filter

State equations of most dynamical systems are originally expressed in continuous time since they are representing real world systems which obey physical laws. However, measurements are taken discretely depending on the sampling time of the digital processors. In continuous-discrete EKF (CD-EKF) formulation, state and covariance matrix prediction expressions are in differential equation form and implementing this filter by digital processors requires numerical integration of these differential equations with sufficiently small integration step size. For the following nonlinear and time invariant dynamic system and knowing that $x = x(t)$ and $x_k = x(t_k)$

$$\begin{aligned}\dot{x} &= f(x) + \omega \\ y_k &= h(x_k) + \nu_k\end{aligned}\tag{2.4.8}$$

Kalman filtering prediction equations for the state and covariance matrix are given as follows [19, 60, 61].

$$\begin{aligned}\dot{\hat{x}} &= f(\hat{x}) \\ \dot{P} &= F_k P + P F_k^T + Q_t\end{aligned}\tag{2.4.9}$$

where \hat{x} is the state estimate, F_k is the Jacobian of $f(\cdot)$ evaluated at current estimate (at time k), P is the covariance matrix of estimate and Q_t is the covariance matrix for the continuous time noise component ω . Summary of CD-EKF equations are given in Table 2.4.2.

For this type of EKF, state and covariance matrix predictions requires numerical integration with sufficiently small integration step sizes (in order to ensure that the integration is stable) until a new measurement arrives [61, 62]. Using RK4 numerical integration scheme state prediction becomes the same as in (2.1.16). Covariance matrix prediction is also carried out by the same approach and the

Table 2.4.2: Equations for CD-EKF.

Initialization	\hat{x}_0^- and P_0^-
Innovation covariance	$S_k = H_k P_k^- H_k^T + R$
Kalman gain	$K_k = P_k^- H_k^T S_k^{-1}$
State update	$\hat{x}_k^+ = \hat{x}_k^- + K_k (y_k - h(\hat{x}_k^-))$
Covariance update	$P_k^+ = (I - K_k H_k) P_k^-$
State prediction	$\hat{x} = f(\hat{x})$
Covariance prediction	$\dot{P} = F_k P + P F_k^T + Q_t$

equations are given as

$$\begin{aligned}
 K_1 &= F_k P_k + P_k F_k^T + Q_d \\
 K_2 &= F_k (P_k + 0.5\delta K_1) + (P_k + 0.5\delta K_1) F_k^T + Q_d \\
 K_3 &= F_k (P_k + 0.5\delta K_2) + (P_k + 0.5\delta K_2) F_k^T + Q_d \\
 K_4 &= F_k (P_k + \delta K_3) + (P_k + \delta K_3) F_k^T + Q_d \\
 P_{k+1} &= P_k + \frac{\delta}{6} (K_1 + 2K_2 + 2K_3 + K_4)
 \end{aligned} \tag{2.4.10}$$

where δ is the integration step size and K_i 's are the intermediate calculations of the covariance matrix. Moreover Q_d is the discrete counterpart of Q_t . The relation between the process noises of continuous time and corresponding discrete time systems can be given as

$$Q_d = Q_t \delta. \tag{2.4.11}$$

For further information about the relation between continuous and discrete time covariance matrices, one can refer to [50, 54].

Continuous-discrete Kalman filters are well suited to the problems with the following properties [60, 63].

- High process noise; smaller prediction step size reduces the effects of the

process noise.

- High nonlinearity; more frequent linearization makes the nonlinearities easy to handle.
- Rare measurements.

2.4.3 Unscented Kalman Filter

Since most of the dynamic system representations are expressed in nonlinear form, optimal (in the sense of MMSE) estimate of the state requires computation of conditional probability density function of the state given all the available information. Hence, standard Kalman filter can not be directly used. In Section 2.4.1 use of EKF was presented. Although it is not optimal, EKF gives satisfactory results when the first order linearization around estimate adequately approximates the conditional probability density function of the state [19]. This local linear approximations may not work in problems with significant nonlinearities [64] and may lead to large estimation errors or sometimes divergence of the filter [52, 65]. In order to overcome this kind of problems Julier and Uhlmann proposed a method called Unscented Kalman Filter (UKF) in [66].

In UKF, the distribution of the state is represented by carefully selected sample points and these sample points completely express the mean and covariance of the distribution [65]. This procedure looks like Monte Carlo type sampling methods but the main difference between them is that UKF selects samples deterministically [64]. Main component of UKF is "unscented transform" which describes the propagation of the distribution of the state represented by sample points through a nonlinear system.

Consider the discrete time nonlinear system given in (2.4.2). One can implement UKF for this system by the following algorithm [19, 52, 53, 65, 66].

1. Initialize filter with state estimate and covariance matrix as follows

$$\hat{x}_0, \quad P_0 = E \{ \tilde{x}_0 \tilde{x}_0^T \}$$

where \hat{x}_0 is the initial state estimate and \tilde{x}_0 is the initial estimation error.

2. Generate sigma points (actually vectors) using the current state estimate and covariance matrix.

$$\begin{aligned}\chi_0 &= \hat{x}_k^+ \\ \chi_i &= \hat{x}_k^+ + \left\{ \sqrt{(n + \lambda) P_k^+} \right\}_i, \quad i = 1, \dots, n \\ \chi_i &= \hat{x}_k^+ - \left\{ \sqrt{(n + \lambda) P_k^+} \right\}_i, \quad i = n + 1, \dots, 2n\end{aligned}$$

where n is the dimension of state vector and λ is a scaling parameter given below. Moreover, the notation $\{.\}_i$ denotes the i^{th} column of the matrix inside.

$$\lambda = \alpha^2(n + \kappa) - n.$$

In the equation above, α is usually chosen in the interval $[10^{-4}, 1]$, determines the spread of sigma points around the state estimate [52] and κ is another scaling parameter and usually set to $(3 - n)$ [67]. In order to take the square root of a symmetric positive definite matrix one may use Cholesky or modified Cholesky decompositions [54]. Let us assume A is a symmetric positive definite matrix. Then it can be factorized as

$$A = LDL^T$$

where L is a lower triangular matrix and D is a diagonal matrix with positive diagonal elements. It can be concluded that square root of the matrix A is $A^{1/2} = LD^{1/2}$ and the square root of a diagonal matrix is simply taken by taking the scalar square roots of diagonal elements. For more information about matrix decompositions and square root algorithms, one can refer to [54, 68].

3. Propagate the sigma points through the nonlinear system dynamics and find the mean and covariance matrix of the transformed sigma points by

using weights W_i^m and W_i^c for i^{th} sigma point respectively.

$$\begin{aligned} W_0^m &= \frac{\lambda}{n + \lambda} \\ W_0^c &= \frac{\lambda}{n + \lambda} + (1 - \alpha^2 + \beta) \\ W_i^m &= W_i^c = \frac{0.5}{n + \lambda}, \quad i = 1, \dots, 2n \end{aligned} \tag{2.4.12}$$

where β is a parameter used for incorporating the prior distribution of the state and for Gaussian distribution $\beta = 2$ is a good choice [19, 65]. After adjusting the weights, state and covariance prediction is calculated as the weighted sum of the transformed sigma points as follows.

$$\begin{aligned} \hat{x}_{k+1}^- &= \sum_{i=0}^{2n+1} W_i^m f(\chi_i) \\ P_{k+1}^- &= \sum_{i=0}^{2n+1} W_i^c [f(\chi_i) - \hat{x}_{k+1}^-] [f(\chi_i) - \hat{x}_{k+1}^-]^T \end{aligned}$$

It should be noted that this step corresponds prediction of the state and covariance matrix.

4. Generate new set of sigma points using predicted state and covariance matrix.

$$\begin{aligned} \chi_0^p &= \hat{x}_{k+1}^- \\ \chi_i^p &= \hat{x}_{k+1}^- + \left\{ \sqrt{(n + \lambda) P_{k+1}^-} \right\}_i, \quad i = 1, \dots, n \\ \chi_i^p &= \hat{x}_{k+1}^- - \left\{ \sqrt{(n + \lambda) P_{k+1}^-} \right\}_i, \quad i = n + 1, \dots, 2n \end{aligned}$$

5. Calculate the observation prediction and its covariance matrix according to equations below.

$$\begin{aligned} \hat{y}_{k+1}^- &= \sum_{i=0}^{2n+1} W_i^m h(\chi_i^p) \\ P_{k+1}^{yy} &= \sum_{i=0}^{2n+1} W_i^c [h(\chi_i^p) - \hat{y}_{k+1}^-] [h(\chi_i^p) - \hat{y}_{k+1}^-]^T. \end{aligned}$$

6. Calculate the cross correlation between the predicted state (\hat{x}_{k+1}^-) and the predicted observation (\hat{y}_{k+1}^-).

$$P_{k+1}^{xy} = \sum_{i=0}^{2n+1} W_i^c [f(\chi_i) - \hat{x}_{k+1}^-] [h(\chi_i) - \hat{y}_{k+1}^-]^T.$$

7. Calculate Kalman gain.

$$K_{k+1} = P_{k+1}^{xy} S_{k+1}^{-1}$$

where $S_{k+1} = P_{k+1}^{yy}$ is the innovations covariance at time $k + 1$ [19].

8. Carry out state and covariance update.

$$\hat{x}_{k+1}^+ = \hat{x}_{k+1}^- + K_{k+1} (y_{k+1} - \hat{y}_{k+1}^-)$$

$$P_{k+1}^+ = P_{k+1}^- - K_{k+1} S_{k+1} K_{k+1}^T$$

9. Iterate.

For this algorithm, the selection of sigma points and their weights may cause the covariance matrix to be non-positive definite. In order to fix this issue, instead of assigning several variables (α, κ, β) related with the distribution of the state, one can choose the first weights of the mean and covariance as $W_0^m = W_0^c \in (0, 1)$ and remaining can be chosen equally to make the sum of all weights one.

There are other unscented transforms which use reduced number of sigma points such as simplex and spherical unscented transforms. They reduce the computational complexity while sacrificing the accuracy of estimation and/or numerical stability. More information about these unscented transforms can be found in [53].

2.4.4 Filter Predictions : A Toy Example

In this section, single prediction (i.e., the time update) performances of the EKF, the UKF and the CD-EKF with different prediction step sizes are compared in terms of position and velocity prediction errors. Errors in the predicted

covariances for the position and the velocity are also investigated. In addition, computation times of the predictions are compared.

The following continuous time dynamic model is used in this section for the comparison.

$$\dot{X} = f_c(X) + w$$

where $X = [x \ y \ z \ v_x \ v_y \ v_z]^T$ is the state vector, $w \sim \mathcal{N}(0_{6 \times 6}, Q)$ is the process noise and $f_c(\cdot)$ is the two body orbit model given as follows.

$$f_c(X) = \begin{bmatrix} v_x \\ v_y \\ v_z \\ \frac{-\mu x}{\sqrt{(x^2 + y^2 + z^2)^3}} \\ \frac{-\mu y}{\sqrt{(x^2 + y^2 + z^2)^3}} \\ \frac{-\mu z}{\sqrt{(x^2 + y^2 + z^2)^3}} \end{bmatrix} \quad (2.4.13)$$

For the state and covariance predictions, RK4 and Euler's methods are utilized. RK4 method is presented in Section 2.1.3. Assuming discrete time model (discretized with the RK4) is represented by $f_d^{RK4}(\cdot)$, it is known that the calculation of the $f_d^{RK4}(\cdot)$ requires 4 evaluations of the $f_c(\cdot)$.

In order to see the effects of different numerical integration schemes, the simplest form of numerical integration (i.e., the Euler's method) is used. Euler's method can be given as follows for the dynamic equation given in (2.4.13).

$$X_{k+1} = f_d^{Euler}(X_k) = X_k + \delta f_c(X_k) \quad (2.4.14)$$

where δ is the integration step size and $f_d^{Euler}(\cdot)$ shows the resulting discrete time model. This method is also used to calculate the covariance prediction for the CD-EKF. It should be noted that the covariance prediction for the CD-EKF is originally given by the differential equation given in Table 2.4.2. This differential equation can be discretized using the Euler's method as follows.

$$P_{k+1} = P_k + \delta(F_k P_k + P_k F_k^T + Q_d) \quad (2.4.15)$$

where F_k is the Jacobian of the $f_c(\cdot)$ evaluated at X_k and $Q_d = \delta Q$ is the discrete counterpart of the continuous time process noise covariance matrix.

For the analysis, given the state estimate (X_k^+) and its covariance (P_k^+) one second prediction is done by using prediction relations of the aforementioned Kalman filters. The following parameters are utilized.

- True predictions are obtained by using CD-EKF (with RK4) with prediction step size of 1 ms.
- EKF and UKF prediction step sizes are 1 second (i.e., $\delta = 1$).
- $\delta = 0.5$ s and $\delta = 0.05$ s are used for CD-EKF.
- Initial estimate (in ECI) and its covariance used are given below. It should be noted that position components are in km and velocity components are in km/s.

$$X_k^+ = \begin{bmatrix} 7000 & 0 & 0 & 0.0 & 2.8 & 7.0 \end{bmatrix}^T,$$

$$P_k^+ = \begin{bmatrix} I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 10^{-4} I_{3 \times 3} \end{bmatrix}.$$

- Process noise covariance for the continuous time model is given as follows. It should be noted that the process noise covariance matrix should be scaled with the δ when different step sizes are used.

$$Q = \begin{bmatrix} 10^{-6} I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 10^{-8} I_{3 \times 3} \end{bmatrix}$$

Prediction errors for the position and the velocity are calculated as follows.

$$\text{pos. error} = \sqrt{(x_{k+1}^- - x_{k+1})^2 + (y_{k+1}^- - y_{k+1})^2 + (z_{k+1}^- - z_{k+1})^2},$$

$$\text{vel. error} = \sqrt{(v_{x_{k+1}}^- - v_{x_{k+1}})^2 + (v_{y_{k+1}}^- - v_{y_{k+1}})^2 + (v_{z_{k+1}}^- - v_{z_{k+1}})^2}.$$

where $\begin{bmatrix} x_{k+1}^- & y_{k+1}^- & z_{k+1}^- & v_{x_{k+1}}^- & v_{y_{k+1}}^- & v_{z_{k+1}}^- \end{bmatrix}^T$ is the predicted state vector and $\begin{bmatrix} x_{k+1} & y_{k+1} & z_{k+1} & v_{x_{k+1}} & v_{y_{k+1}} & v_{z_{k+1}} \end{bmatrix}^T$ is the true predicted state vector.

For the covariance errors, Frobenius norm can be used. For an $n \times n$ symmetric matrix A , Frobenius norm is defined as follows [19].

$$\|A\| \triangleq \sqrt{\text{trace}(A^2)} \quad (2.4.16)$$

where $\text{trace}(B)$ represents the sum of diagonal elements of the square matrix B . Let P_{k+1}^- and P_{k+1} be the predicted covariance and the true covariance respectively. In order to find the covariance error for the position and velocity one can use the following equations.

$$\text{pos. covariance error} = \sqrt{\text{trace}(A_p^2)},$$

$$\text{vel. covariance error} = \sqrt{\text{trace}(A_v^2)}.$$

where the 3×3 matrix A_p shows the difference between the position related parts of the predicted and the true covariance matrices and similarly A_v (3×3) is the matrix found by subtraction of the velocity related parts of the predicted and the true covariance matrices.

In order to compare the prediction times, prediction equations are run 1000 times and the average computation times are calculated.

Table 2.4.3: Prediction performance comparison of the filters using RK4 method.

	EKF (1 s)	UKF (1 s)	CD-EKF (0.5 s)	CD-EKF (0.05 s)
Pos. error (km)	1.82×10^{-11}	1.73×10^{-11}	1.73×10^{-11}	1.64×10^{-11}
Vel. error (km/s)	2.70×10^{-15}	2.81×10^{-15}	2.70×10^{-15}	1.78×10^{-15}
Pos. cov. error (km) ²	2.25×10^{-5}	1.73×10^{-6}	8.67×10^{-7}	8.54×10^{-8}
Vel. cov. error (km/s) ²	1.73×10^{-8}	1.73×10^{-8}	8.64×10^{-9}	8.49×10^{-10}
Number of $f_c(\cdot)$ eval.	52	52	32	320
Avg. comp. time (ms)	0.460	0.541	0.343	3.248

The results are shown in Tables 2.4.3 and 2.4.4 for the RK4 and the Euler cases respectively. As we expected, prediction errors for the Euler discretization are higher and it takes less time to compute the prediction equations since this

Table 2.4.4: Prediction performance comparison of the filters using Euler’s method.

	EKF (1 s)	UKF (1 s)	CD-EKF (0.5 s)	CD-EKF (0.05 s)
Pos. error (km)	4.07×10^{-3}	4.07×10^{-3}	2.03×10^{-3}	2.03×10^{-4}
Vel. error (km/s)	4.38×10^{-6}	4.38×10^{-6}	2.19×10^{-6}	2.19×10^{-7}
Pos. cov. error (km) ²	2.27×10^{-5}	3.33×10^{-6}	8.58×10^{-5}	8.58×10^{-6}
Vel. cov. error (km/s) ²	1.73×10^{-8}	1.73×10^{-8}	8.64×10^{-9}	8.49×10^{-10}
Number of $f_c(\cdot)$ eval.	13	13	26	260
Avg. comp. time (ms)	0.133	0.196	0.245	2.392

type of discretization is less complex. The computation time of the CD-EKF predictions are dominated by the calculation of the dynamic equation as seen in the tables. The difference between the evaluation times is basically caused by the multiplications and the additions which are present in the prediction equations when the number of dynamic model evaluations are the same. It can be also said that using a smaller prediction step size reduces the effect of the process noise since we scale it with the step size. It should be mentioned that, when the dynamic model used is simple then the prediction computation times are dominated by the number of additions and multiplications. Therefore, using CD-EKF with a simple dynamic model would probably be slower than the other algorithms depending on the discretization step size.

2.4.5 Nonlinear Smoothing

In filtering, all the information coming from successive measurements up to current time is used in order to calculate the estimate of the current state. Consequently, only the last estimate uses all the data. In order to refine the previous estimates, one can use all the existing data. This procedure is called smoothing. In general, the accuracy of a smoother is better than the filter. For linear stable systems smoothing improves the filtering performance in the sense of MSE at

most by a factor of two [54].

Mathematically, it is known that the optimal estimate in the sense of MMSE is the mean of the conditional distribution given all the available information. Let us assume that x is the random parameter to be estimated and y is the observations. Then the optimal estimate can be written by the following conditional expectation.

$$\hat{x} = E \{x|y\}. \quad (2.4.17)$$

For the state estimation case, at time k , let the available measurements be y_1, \dots, y_k . Then the filtered estimate is given by the equation below.

$$\hat{x}_k = E \{x_k|y_1, \dots, y_k\}. \quad (2.4.18)$$

If it is desired to estimate the state at time $r < k$ by using all the available information at time k then the optimal estimator will be

$$\hat{x}_r = E \{x_r|y_1, \dots, y_k\}. \quad (2.4.19)$$

This fact is the main point of the smoothing idea and this kind of estimators are called smoothers. In literature, smoothers are grouped into three categories [19] which are fixed-interval, fixed-lag and fixed-point smoothers. They are classified according to which set of measurements are used in order to estimate a state or a set of states. In Fixed-interval smoothing, all available data is utilized to estimate all states for certain time of interest. Trajectory fitting can be an example of fixed-interval smoothing. In fixed-lag smoothing, estimate at time k is generated from measurements up to time $k + L$. Here $L > 0$ is the lag time. Usually this types of smoothers run in real time but gives delayed estimates [54]. Finally, in fixed-point smoothing, estimate for the state at certain time is generated using all set of measurements up to current time [54]. Fixed-point smoothers are convenient when the state at only a certain time instant is of interest [54].

A smoother can be described as forward filter followed by a backward filter. A smoothing algorithm proposed by Rauch, Tung and Striebel called RTS smoother is probably the most widely used smoothing algorithm [19, 53]. It

gives a closed form solution to linear Gaussian smoothing problem, i.e., it is the smoother counterpart of Kalman filter [64].

Although there are various types of smoothing algorithm for nonlinear applications, here only the extended version of RTS smoother will be presented. Further information about smoothers can be found in [19, 53, 54, 64, 69]. Extended RTS smoother uses the first order linear approximation of nonlinear systems so it can be said that it corresponds to EKF for filtering case [64]. For the nonlinear model given in (2.4.2), the extended RTS smoother is given by the following algorithm [64].

1. Run EKF normally up to time N . Let the last updated state estimate and covariance matrix be \hat{x}_N^+ and P_N^+ . It should be mentioned that storage of estimates \hat{x}_k^+ , \hat{x}_k^- covariance matrices P_k^+ , P_k^- and Jacobians of the state transition function $f(\cdot)$ (namely, F_k) for each time k is required for the backward recursion.
2. Initialize the smoother as follows assuming smoothed state is \hat{s} and the corresponding covariance matrix is S .

$$\begin{aligned}\hat{s}_N &= \hat{x}_N^+, \\ S_N &= P_N^+.\end{aligned}\tag{2.4.20}$$

3. Calculate smoother gain G_k for time k .

$$G_k = P_k^+ F_k^T (P_{k+1}^-)^{-1}.\tag{2.4.21}$$

4. Iterate smoother equation backward until reaching the desired state.

$$\begin{aligned}\hat{s}_k &= \hat{x}_k^+ + G_k [\hat{s}_{k+1}^- - \hat{x}_{k+1}^-], \\ S_k &= P_k^+ - G_k [P_{k+1}^- - S_{k+1}] G_k^T.\end{aligned}\tag{2.4.22}$$

2.5 Cramer-Rao Lower Bound

In estimation theory, it is always desired to have an optimal (in some sense) estimator. However, usually this requirement is not satisfied especially for the

nonlinear case [56]. Consequently, one of these suboptimal estimators should be chosen. A commonly accepted performance criterion is the estimation error variance and it is desired to be minimized. In order to assess the performance of proposed estimator in terms of Mean Square Error (MSE), Cramer-Rao Lower Bound (CRLB) is widely used. CRLB provides the theoretical lower limit on MSE [70]. CRLB is first developed for classical estimation case (i.e. parameter to be estimated is deterministic). Then it was extended to the Bayesian estimation case [70], namely the estimation of random parameters, by Van Trees in [71]. CRLB for the Bayesian case is called PCRLB. The recursive form of PCRLB for discrete nonlinear filtering problems was proposed in [72] which will be used in this thesis.

Let x be the random parameter vector to be estimated, y be the measurement vector and $\hat{x} = g(y)$ be the estimator based on measurements. PCRLB states the following inequality [71]

$$E \left\{ (x - \hat{x})(x - \hat{x})^T \right\} \geq J^{-1} \triangleq PCRLB \quad (2.5.1)$$

where J is the Fisher information matrix consisting of the below elements at its i^{th} row and j^{th} column

$$J_{ij} = E \left\{ -\frac{\partial^2 \ln p(x, y)}{\partial x_i \partial x_j} \right\} \quad (2.5.2)$$

where $p(x, y)$ denotes the joint probability density function of x and y .

Let us have the following discrete nonlinear system with time dependent nonlinear vector valued functions $f_k(\cdot)$ and $h_k(\cdot)$,

$$\begin{aligned} x_{k+1} &= f_k(x_k, \omega_k) \\ y_k &= h_k(x_k, \nu_k) \end{aligned} \quad (2.5.3)$$

where x_k and y_k represents state ($n \times 1$) and measurement ($m \times 1$) vectors respectively and ω_k and ν_k are noise components. Then the recursive calculation of the Fisher information matrix is given as follows [72]

$$J_{k+1} = D_k^{22} - D_k^{21} (J_k + D_k^{11})^{-1} D_k^{12} \quad (2.5.4)$$

where J_k is the $n \times n$ Fisher information matrix at time k and other matrices involved in (2.5.4) are defined as

$$\begin{aligned}
D_k^{11} &= -E \left\{ \Delta_{x_k}^{x_k} \ln p(x_{k+1}|x_k) \right\} \\
D_k^{12} &= -E \left\{ \Delta_{x_k}^{x_{k+1}} \ln p(x_{k+1}|x_k) \right\} \\
D_k^{21} &= [D_k^{21}]^T \\
D_k^{22} &= -E \left\{ \Delta_{x_{k+1}}^{x_{k+1}} \ln p(x_{k+1}|x_k) \right\} - E \left\{ \Delta_{x_{k+1}}^{x_{k+1}} \ln p(y_{k+1}|x_{k+1}) \right\}
\end{aligned} \tag{2.5.5}$$

For the matrices $A = [a_1 \ a_2 \ \dots \ a_n]^T$ and $B = [b_1 \ b_2 \ \dots \ b_m]^T$, the operator Δ_A^B is defined as

$$\Delta_A^B = \begin{bmatrix} \frac{\partial^2}{\partial a_1 \partial b_1} & \frac{\partial^2}{\partial a_1 \partial b_2} & \cdots & \frac{\partial^2}{\partial a_1 \partial b_m} \\ \frac{\partial^2}{\partial a_2 \partial b_1} & \frac{\partial^2}{\partial a_2 \partial b_2} & \cdots & \frac{\partial^2}{\partial a_2 \partial b_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2}{\partial a_n \partial b_1} & \frac{\partial^2}{\partial a_n \partial b_2} & \cdots & \frac{\partial^2}{\partial a_n \partial b_m} \end{bmatrix}. \tag{2.5.6}$$

If the noise components of above nonlinear system in (2.5.4) is assumed to be additive Gaussian with zero mean and independent then the D matrices in (2.5.5) which contains Δ operator can be calculated easily due to the simple form of the natural logarithm of Gaussian probability density function. The resulting D matrices become

$$\begin{aligned}
D_k^{11} &= E \left\{ [\nabla_{x_k} f_k(x_k)]^T Q_k^{-1} [\nabla_{x_k} f_k(x_k)] \right\} \\
D_k^{12} &= -E \left\{ [\nabla_{x_k} f_k(x_k)]^T \right\} Q_k^{-1} \\
D_k^{21} &= [D_k^{21}]^T \\
D_k^{22} &= Q_k^{-1} + E \left\{ [\nabla_{x_{k+1}} h_{k+1}(x_{k+1})]^T R_{k+1}^{-1} [\nabla_{x_{k+1}} h_{k+1}(x_{k+1})] \right\}
\end{aligned} \tag{2.5.7}$$

where Q_k and R_k represents covariance matrices for ω_k and ν_k respectively. Moreover, ∇_x denotes the Jacobian and for an $m \times 1$ vector valued function $f(\cdot)$

with elements $f_i, i = 1, \dots, m$ it is defined as follows [19].

$$\nabla_x f = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \quad (2.5.8)$$

where $x = [x_1 \ x_2 \ \dots \ x_n]^T$. If it is further assumed that the system is linear with additive, zero mean and independent Gaussian noise components then PCRLB recursion given above reduces to Kalman filter covariance updates [72].

Calculation of the PCRLB recursively by using (2.5.4) requires taking the expectations of matrices involving nonlinear functions of random vectors. Taking this expectations explicitly is impossible most of the times. Instead, the use of Monte Carlo methods is more convenient. This methodology is based on producing samples from the random vector according to its distribution and passing them through nonlinear functions and then taking the expectation by averaging the results. PCRLB algorithm for a time invariant nonlinear system for Gaussian noise case with prior informations x_0 and P_0 can be given as follows where M is the number of Monte Carlo runs.

1. Initialize $J_0 = P_0^{-1}$.
2. Produce M initial states according to its distribution as $x_0^i, i = 1, 2, \dots, M$.
3. Propagate all the sample states according to the following state model starting from x_0^i

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

4. Calculate Jacobians for all sample states as follows

$$F_i = \nabla_x f, \text{ evaluated at } x = x_k^i.$$

and

$$H_i = \nabla_x h \text{ evaluated at } x = x_{k+1}^i.$$

5. Calculate D matrices according to (2.5.7)

$$D_k^{11} = \frac{1}{M} \sum_{i=1}^M F_i^T Q^{-1} F_i,$$

$$D_k^{12} = -\frac{1}{M} \sum_{i=1}^M F_i^T Q^{-1},$$

$$D_k^{21} = [D_k^{12}]^T,$$

$$D_k^{22} = Q^{-1} + \frac{1}{M} \sum_{i=1}^M H_i^T R^{-1} H_i.$$

6. Calculate Fisher information J_{k+1} matrix using (2.5.4).

7. Calculate PCRLB for $(k+1)^{th}$ step by

$$PCRLB_{k+1} = J_{k+1}^{-1}.$$

8. Diagonal elements of PCRLB corresponds to the lower limit of MSE for each state.

9. Return to Step 4 and iterate.

If continuous time state model is used as in the CD-EKF then the above relations for the PCRLB can not be directly used. In [70], for the continuous time state equation and discrete measurement equation case such as in (2.4.8), another form of PCRLB is proposed. Fisher information matrix (J) is propagated according to following differential equation until a new measurement arrives [70].

$$\dot{J} = -JF - F^T J - JQJ. \quad (2.5.9)$$

In the equation above, F is the Jacobian of $f(\cdot)$ at the current state and Q is the process noise covariance matrix for continuous time. When a new measurement arrives then the Fisher information matrix is corrected using the relation below [70].

$$J_k^+ = J_k^- + E \{ H_k^T R^{-1} H_k \} \quad (2.5.10)$$

where J_k^- and J_k^+ shows the predicted and corrected Fisher information matrices at time k respectively. H_k is the Jacobian of $h(\cdot)$ evaluated at state $x(t_k) = x_k$ namely, $H_k = \nabla_x h(x)|_{x=x_k}$ which was defined earlier in this section.

In order to calculate Fisher information matrix recursively by using (2.5.9) and (2.5.10), (2.5.9) should be discretized by using numerical integration techniques (RK4 is also applicable for this case). Utilizing (2.1.16) gives us the propagation (or prediction) of J as follows.

$$\begin{aligned}
K_1 &= -J_k^- F - F^T J_k^- - J_k^- Q_d J_k^- \\
K_2 &= -(J_k^- + 0.5\delta K_1) F - F^T (J_k^- + 0.5\delta K_1) \\
&\quad - (J_k^- + 0.5\delta K_1) Q_d (J_k^- + 0.5\delta K_1) \\
K_3 &= -(J_k^- + 0.5\delta K_2) F - F^T (J_k^- + 0.5\delta K_2) \\
&\quad - (J_k^- + 0.5\delta K_2) Q_d (J_k^- + 0.5\delta K_2) \\
K_4 &= -(J_k^- + \delta K_3) F - F^T (J_k^- + \delta K_3) \\
&\quad - (J_k^- + \delta K_3) Q_d (J_k^- + \delta K_3) \\
J_{k+1}^- &= J_k^- + \frac{\delta}{6}(K_1 + 2K_2 + 2K_3 + K_4)
\end{aligned} \tag{2.5.11}$$

where δ is the step size of the numerical integration and Q_d is the discrete counterpart of Q which was defined in Section 2.4.2.

Detailed information, derivations and proofs about both Bayesian and non-Bayesian CRLB can be found in [50, 51, 70, 71, 72, 73, 74].

CHAPTER 3

INITIAL ORBIT DETERMINATION

In this chapter, orbit determination using limited number of ground based measurements is reviewed. Gauss' angles only initial orbit determination algorithm is presented. Performance of the Gauss' algorithm (in terms of resulting accuracy levels) is investigated.

In order to completely represent the orbit, one needs six quantities [32], these are generally three position and corresponding three velocity components so six independent observations are required to completely specify the orbit. However, for example, single set of angles only measurement gives only two angles (usually azimuth and elevation). Hence, at least three sets of independent angles only measurements are required. If range data is available as well as angles, one needs at least two set of angles and range (corresponds position) measurements. If there is little or no knowledge about the orbit of a satellite then results coming from initial orbit determination (IOD) methods will provide a good starting point for further analysis (starting point for statistical estimation techniques). It should also be mentioned that Earth based sensors produce data in terms of pointing angles, range and possibly their rates and these measurements should be converted to state vector form for statistical analysis.

The first method of orbit determination from three observation is given by Newton in his Principia [2]. Historically, initial orbit determination has been carried out by using angles only observations since telescopes do not provide range or range rate data and they were the only available sensor type in those days. To-

day, in the presence of radar systems the problem of orbit determination is much more simpler with the aid of range information [2]. Basically, IOD calculations are established on the following available observation types.

- Three sets of angles
- Two position vectors
- Three position vectors

Aforementioned methods are originally developed for determining the orbits of planets or asteroids. These methods require time information that is either the absolute time of measurements or the time between measurements. Orbit determination using three sets of angular measurements is first proposed by Laplace [2]. Although his method is not robust for near Earth orbits with rare measurements, it is valid for interplanetary orbits [1]. For this problem, Gauss also proposed a solution that is more robust than the Laplace's method but has bad performance for measurements with high separation. In order to overcome this separation problem of observations, Escobal suggested a method called Double-r iteration. This method can handle observations that are days apart from each other [1]. The second case of initial orbit determination from two position vectors and time of flight between them is usually called Lambert's problem since the solution of this particular problem is first introduced by him [1]. Apart from initial orbit determination, this method can also be utilized in orbital transfer and interplanetary orbit problems [2, 32]. The last method which utilizes three successive position vectors and proposes a solution based on geometric approach is known as Gibbs method [1]. At first glance three position vectors seem to provide nine independent quantities about the orbit but this is not true since the position vectors lie in the same plane (i.e., they are linearly dependent) [2]. This method suffers from too close observations and there exists an improved version called Herrick-Gibbs that handles close observations but it is not robust as Gibbs method. Detailed information about the mentioned methods of initial orbit determination can be obtained from [1, 2, 32].

3.1 Gauss' Method of Initial Orbit Determination

Gauss' method of initial orbit determination from three angles only measurements is developed under certain assumptions. Firstly, all the mentioned IOD methods give solution for two body equations of motion. Moreover, ground based measurement vectors are supposed to lie in the same plane [1]. These assumptions are not perfectly true in practice. Since there exist perturbations (see Section 2.1.2), two body assumption is violated. Furthermore, real world observations may not lie in the same plane. However, these assumptions simplify the problem and make the analytical solution less complex. Gauss' method works well for near Earth satellites and when the measurements are separated by 10° or less, it gives fairly good results [1]. Gauss' method uses topocentric right ascension and declination instead of azimuth and elevation since they can be directly acquired from background stars on telescope camera image [2]. Also they represent the satellite position vector in inertial frame. Moreover, if measurements are obtained as azimuth and elevation, conversion (presented in Section 2.3.8) to topocentric right ascension and declination will be required. In this section it is assumed that topocentric right ascension (α) and declination (δ) measurements are directly available. Consider the scenario in Figure 3.1.1, where t_1 , t_2 and

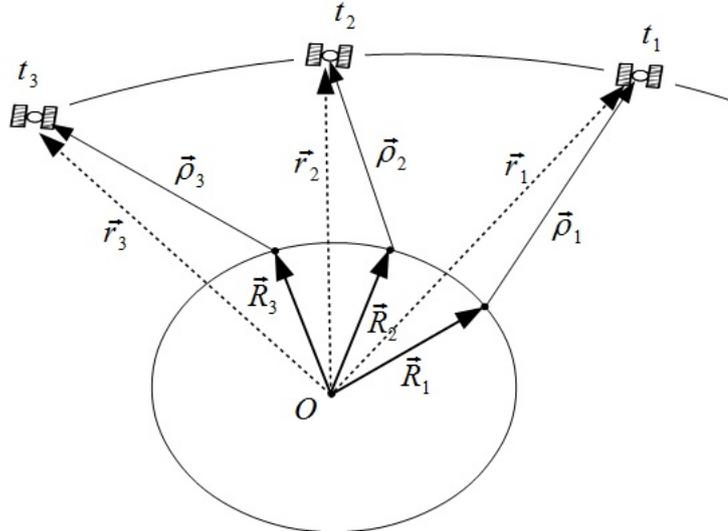


Figure 3.1.1: Scenario for the Gauss' method.

t_3 are successive measurement times and \vec{R}_i 's show the observer position vectors in inertial frame (specifically ECI) corresponding to the observation times, $\vec{\rho}_i$'s

and \vec{r}_i 's represent the satellite position vectors in inertial frame with respect to observer and geocenter respectively. It should be noted that observer location is changing in space not on Earth. The relation between the previously mentioned vectors can be given as

$$\vec{r}_i = \vec{R}_i + \vec{\rho}_i \quad i = 1, 2, 3. \quad (3.1.1)$$

Let r_i , R_i and ρ_i be the magnitudes of the vectors used in (3.1.1). Then (3.1.1) can be written as follows

$$\vec{r}_i = \vec{R}_i + L_i \rho_i \quad i = 1, 2, 3. \quad (3.1.2)$$

where L_i is the unit vector for i^{th} observation with the corresponding right ascension and declination (α_i, δ_i) values and it can be given as

$$L_i = \begin{bmatrix} \cos \alpha_i \cos \delta_i \\ \sin \alpha_i \cos \delta_i \\ \sin \delta_i \end{bmatrix} \quad i = 1, 2, 3.$$

Gauss algorithm can be implemented as follows [32].

1. Calculate the time intervals in seconds between the measurements.

$$\tau_1 = t_1 - t_2$$

$$\tau_3 = t_3 - t_2$$

$$\tau = t_3 - t_1.$$

2. Calculate the following cross product terms.

$$p_1 = L_2 \times L_3$$

$$p_2 = L_1 \times L_3$$

$$p_3 = L_1 \times L_2.$$

3. Calculate the dot product $D_0 = L_1 \cdot p_1$.

4. Calculate the following scalars.

$$\begin{aligned} D_{11} &= \vec{R}_1 \cdot p_1 & D_{12} &= \vec{R}_1 \cdot p_2 & D_{13} &= \vec{R}_1 \cdot p_3 \\ D_{21} &= \vec{R}_2 \cdot p_1 & D_{22} &= \vec{R}_2 \cdot p_2 & D_{23} &= \vec{R}_2 \cdot p_3 \\ D_{31} &= \vec{R}_3 \cdot p_1 & D_{32} &= \vec{R}_3 \cdot p_2 & D_{33} &= \vec{R}_3 \cdot p_3. \end{aligned}$$

5. Calculate the following parameters.

$$\begin{aligned} A &= \frac{1}{D_0} \left[-D_{12} \frac{\tau_3}{\tau} + D_{22} + D_{32} \frac{\tau_1}{\tau} \right] \\ B &= \frac{1}{6D_0} \left[D_{12} (\tau_3^2 - \tau^2) \frac{\tau_3}{\tau} + D_{32} (\tau^2 - \tau_1^2) \frac{\tau_1}{\tau} \right] \\ E &= \vec{R}_2 \cdot L_2 \\ a &= - (A^2 + 2AE + R_2^2) \\ b &= -2\mu B (A + E), \quad (\mu \approx 398,600.44 \frac{km^3}{s^2}) \\ c &= -\mu^2 B^2 \end{aligned}$$

6. Find the roots of the following eighth order polynomial. Discard the negative and complex valued roots and choose one of the most reasonable root as r_2 (magnitude of the satellite position with respect to geocenter at time t_2).

$$x^8 + ax^6 + bx^3 + c = 0. \quad (3.1.3)$$

Newton-Raphson iteration can be utilized in order to find the roots of (3.1.3). Since the performance of the proposed algorithm depends on the initial condition, a reasonable starting point should be chosen (if the starting point is not good then the iteration may not converge or may converge to wrong place). For LEO satellites $\sim 7,000 km$ can be a good initial condition since the Earth radius is approximately $6,378 km$ at the equator. Newton-Raphson method for the equation $f(x) = 0$ is expressed by the following iteration [57].

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}. \quad (3.1.4)$$

7. By using the r_2 found above, calculate the corresponding slant ranges between observer and satellite for t_1, t_2 and t_3 as follows.

$$\rho_1 = \frac{1}{D_0} \left[\frac{6 \left(D_{31} \frac{\tau_1}{\tau_3} + D_{21} \frac{\tau}{\tau_3} \right) r_2^3 + \mu D_{31} (\tau^2 - \tau_1^2) \frac{\tau_1}{\tau_3}}{6r_2^3 + \mu (\tau^2 - \tau_3^2)} - D_{11} \right]$$

$$\rho_2 = A + \mu \frac{B}{r_2^3}$$

$$\rho_3 = \frac{1}{D_0} \left[\frac{6 \left(D_{13} \frac{\tau_3}{\tau_1} + D_{23} \frac{\tau}{\tau_1} \right) r_2^3 + \mu D_{13} (\tau^2 - \tau_3^2) \frac{\tau_3}{\tau_1}}{6r_2^3 + \mu (\tau^2 - \tau_1^2)} - D_{33} \right]$$

8. Calculate the satellite position vectors (\vec{r}_i) for each time by using (3.1.2) and newly found ρ_i 's ($i = 1, 2, 3$).
9. Calculate satellite's velocity vector \vec{v}_2 by the following relation. It should be noted that $\|\vec{v}_2\| = v_2$.

$$\vec{v}_2 = \frac{-f_3 \vec{r}_1 + f_1 \vec{r}_3}{f_1 g_3 - f_3 g_1} \quad (3.1.5)$$

where f_1, f_3 and g_1, g_3 are Lagrange coefficients approximately given by

$$f_1 \approx 1 - \mu \frac{\tau_1^2}{2r_2^3} \quad g_1 \approx \tau_1 - \mu \frac{\tau_1^3}{6r_2^3}$$

$$f_3 \approx 1 - \mu \frac{\tau_3^2}{2r_2^3} \quad g_3 \approx \tau_3 - \mu \frac{\tau_3^3}{6r_2^3}.$$

Further improvement on the calculated orbit can be obtained by applying the subsequent statements.

10. Calculate the reciprocal of the semimajor axis as follows.

$$\alpha = \frac{2}{r_2} - \frac{v_2^2}{\mu}$$

Semimajor axis is the half of the longest diameter of an ellipse.

11. Calculate the magnitude of the radial component of \vec{v}_2 . Define v_{r2} by the following dot product.

$$v_{r2} = \vec{v}_2 \cdot \frac{\vec{r}_2}{\|\vec{r}_2\|}$$

12. Solve the following equations for the universal anomalies χ_1 and χ_3 by using the Newton-Raphson iteration given in (3.1.4).

$$\sqrt{\mu}\tau_1 = \frac{r_2 v_{r2}}{\sqrt{\mu}} \chi_1^2 C(\alpha\chi_1^2) + (1 - \alpha r_2) \chi_1^3 S(\alpha\chi_1^2) + r_2 \chi_1 \quad (3.1.6)$$

$$\sqrt{\mu}\tau_3 = \frac{r_2 v_{r2}}{\sqrt{\mu}} \chi_3^2 C(\alpha\chi_3^2) + (1 - \alpha r_2) \chi_3^3 S(\alpha\chi_3^2) + r_2 \chi_3 \quad (3.1.7)$$

where for elliptical and circular orbits $\alpha\chi^2 > 0$. Moreover, the functions $C(\cdot)$ and $S(\cdot)$ are defined as

$$C(x) = \begin{cases} \frac{1 - \cos(\sqrt{x})}{2}, & x > 0 \\ 0.5, & x = 0, \\ \frac{\cosh(\sqrt{-x}) - 1}{-x}, & x < 0 \end{cases}$$

$$S(x) = \begin{cases} \frac{\sqrt{x} - \sin(\sqrt{x})}{(\sqrt{x})^3}, & x > 0 \\ \frac{1}{6}, & x = 0. \\ \frac{\sinh(\sqrt{-x}) - \sqrt{-x}}{(\sqrt{-x})^3}, & x < 0 \end{cases}$$

In order to solve equations (3.1.6) and (3.1.7), Newton-Raphson iteration can be started using the following initial conditions respectively [32].

$$\chi_1^{start} = \sqrt{\mu}|\alpha|\tau_1,$$

$$\chi_3^{start} = \sqrt{\mu}|\alpha|\tau_3.$$

13. Recalculate Lagrange coefficients by using the values χ_1 and χ_3 found above as follows.

$$\begin{aligned} f_1 &= 1 - \frac{\chi_1^2}{r_2} C(\alpha\chi_1^2) & g_1 &= \tau_1 - \frac{1}{\sqrt{\mu}} \chi_1^3 S(\alpha\chi_1^2) \\ f_3 &= 1 - \frac{\chi_3^2}{r_2} C(\alpha\chi_3^2) & g_3 &= \tau_3 - \frac{1}{\sqrt{\mu}} \chi_3^3 S(\alpha\chi_3^2) \end{aligned} \quad (3.1.8)$$

14. Recalculate the slant ranges between the satellite and the observer ρ_1, ρ_2

and ρ_3 as

$$\rho_1 = \frac{1}{D_0} \left(-D_{11} + \frac{D_{21}}{c_1} - \frac{c_3}{c_1} D_{31} \right)$$

$$\rho_2 = \frac{1}{D_0} (-c_1 D_{12} + D_{22} - c_3 D_{32})$$

$$\rho_3 = \frac{1}{D_0} \left(-\frac{c_1}{c_3} D_{13} + \frac{D_{23}}{c_3} - D_{33} \right)$$

where c_1 and c_2 are defined as follows.

$$c_1 = \frac{g_3}{f_1 g_3 - f_3 g_1}$$

$$c_3 = \frac{-g_1}{f_1 g_3 - f_3 g_1}$$

15. Recalculate the satellite position vectors (\vec{r}_i) by using (3.1.2).
16. Recalculate the satellite velocity vector \vec{v}_2 as in (3.1.5) by using newly found Lagrange coefficients in (3.1.8).
17. Go to Step 11 and iterate until the changes in the slant ranges (ρ_1, ρ_2, ρ_3) fall below a certain level of precision.

To conclude, given the three sets of angles only measurements for times t_1, t_2 and t_3 , Gauss' algorithm gives the corresponding slant ranges between the satellite and the observer. It also provides position and velocity vectors (i.e, the state vector) corresponding to the middle measurement. Furthermore, it should be mentioned that Gauss' method consists of completely analytical techniques (i.e., does not use a statistical approach). It requires a reasonable guess of r_2 in order to find the roots of 8th degree polynomial given in (3.1.3) by the Newton-Raphson iteration. For more information about Gauss' method one can refer to [1, 32, 75, 76].

3.2 Performance Evaluation of the Gauss' Method

In this section, performance evaluation of the Gauss' algorithm (given in Section 3.1) is carried out for a satellite orbit estimation scenario which is going to be described fully in Section 4.1. As mentioned above, Gauss' algorithm takes three

measurements and it calculates the position and the velocity estimates corresponding to the second measurement and the range estimates corresponding to all three measurements. In this section, the uncertainty (i.e., the standard deviation) of the estimates given by the Gauss' algorithm are examined with respect to the time separation between the measurements and the angular measurement accuracy. The aim of this analysis is to find the required time separation and angle measurement accuracy for obtaining a reasonable performance with the Gauss' algorithm.

It is reasonable to assume that the uncertainties in the estimates provided by the Gauss' algorithm are monotone functions of time separation. In other words, as the time separation between the measurements increases, the accuracy in the estimates provided by the Gauss' algorithm increases. Using this observation, in this section, we use bisection algorithm [57] to determine the minimum time separation which results in a certain estimate accuracy with Gauss' algorithm at an angle measurement uncertainty. A total of 20 different angle noise standard deviation values, which are logarithmically uniformly spaced between 10^{-4} and 10^{-1} degrees, are used. For each angle noise standard deviation, the required time separation between the measurements for obtaining a desired performance level from the Gauss' algorithm is found using the bisection algorithm. For this purpose, we need to be able to calculate the uncertainty in the estimates of the Gauss' algorithm given a specific measurement time separation. This is done as follows.

Given a measurement time separation Δ , three true measurements, i.e., $Y_{t_1}^{true}$, $Y_{t_2}^{true}$ and $Y_{t_3}^{true}$ where $t_2 - t_1 = t_3 - t_2 = \Delta$, are selected from a data set which is produced for a satellite visibility duration of 700 seconds with a period of 1 second by using Systems Tool Kit (STK). Each of $Y_{t_i}^{true}$ is composed of true azimuth and elevation values. Then 200 noise corrupted angular measurements

are generated by adding Gaussian noise to $Y_{t_1}^{true}$, $Y_{t_2}^{true}$ and $Y_{t_3}^{true}$ as follows.

$$Y_{t_1}^i = Y_{t_1}^{true} + v_{t_1}^i$$

$$Y_{t_2}^i = Y_{t_2}^{true} + v_{t_2}^i$$

$$Y_{t_3}^i = Y_{t_3}^{true} + v_{t_3}^i$$

for $i = 1, \dots, 200$ where $v_{t_j}^i$ ($j = 1, 2, 3$) denote zero-mean Gaussian measurement noise with a specified standard deviation. Generated triples of noisy azimuth and elevation values are passed through the Gauss' algorithm and covariance of the resulting estimates is calculated (calculation method is explained in Section 4.2.3.4). Using the procedure given above, the required time separation Δ which gives the desired estimate uncertainty is found using the bisection algorithm for each angle measurement noise standard deviation.

Minimum time separations required to obtain position estimates with 3-D accuracies of 50 km, 10 km, 5 km and 1 km can be seen in Figure 3.2.1. Minimum time separations required to obtain velocity estimates with 3-D accuracies of 0.5 km/s, 0.1 km/s, 0.01 km/s and 0.005 km/s can be seen in Figure 3.2.2. Minimum time separations required to obtain range estimates with accuracies 20 km, 10 km, 2 km and 0.5 km can be seen in Figure 3.2.3. It can be concluded from the figures that achieving a position accuracy of better than 1 kilometer requires measurement separations larger than 350 seconds, which means a satellite visibility duration greater than 700 seconds, with an angular measurement uncertainty of 0.01 degrees. In terms of velocity accuracy, performance of the Gauss' algorithm seems rather good since achieving an accuracy of 5 m/s is possible with an accurate sensor system and with a high visibility duration.

Time separation of three sets of azimuth and elevation measurements which are used in the Gauss' method is important in order to achieve certain performance level for the initial estimate (finding consistent state and covariance matrix) for sequential estimation algorithms. Since LEO satellites are our main interest in this work, it should be noted that the maximum visibility duration of the satellite for a ground based sensor is about 800 seconds (depending on the altitude, the maximum elevation angle and the sensor type). Therefore, the time

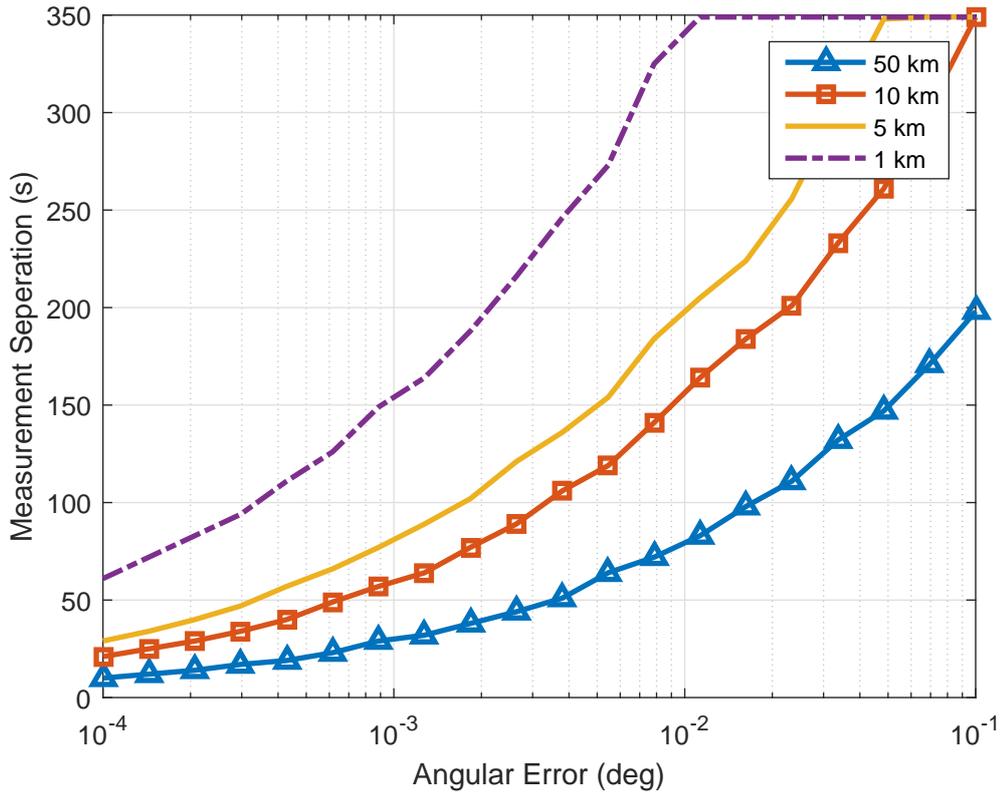


Figure 3.2.1: Minimum time separation to achieve a specified position uncertainty.

separation between three measurements which are used in the Gauss' algorithm can not exceed 400 seconds (assuming equally spaced three measurements are used) in our work. In addition, if optical telescopes are used then the visibility duration is shorter. In order to initialize sequential filters when angles only measurements are available, range information might also be used for some applications. Therefore, the uncertainty level of the initial range calculated using the Gauss' method (i.e., ρ_1) should also be considered.

It should also be mentioned that the radio wave based and optical satellite tracking systems provide angular measurements with a few tens of millidegrees and a few millidegrees of accuracies respectively. Combining this information with the approximate visibility durations, one can achieve a 3-D position accuracy of 1 km and velocity accuracy of 5 m/s for both optical and radio wave based sensors. In addition, range can be estimated with a standard deviation of 2 km for both tracking systems.

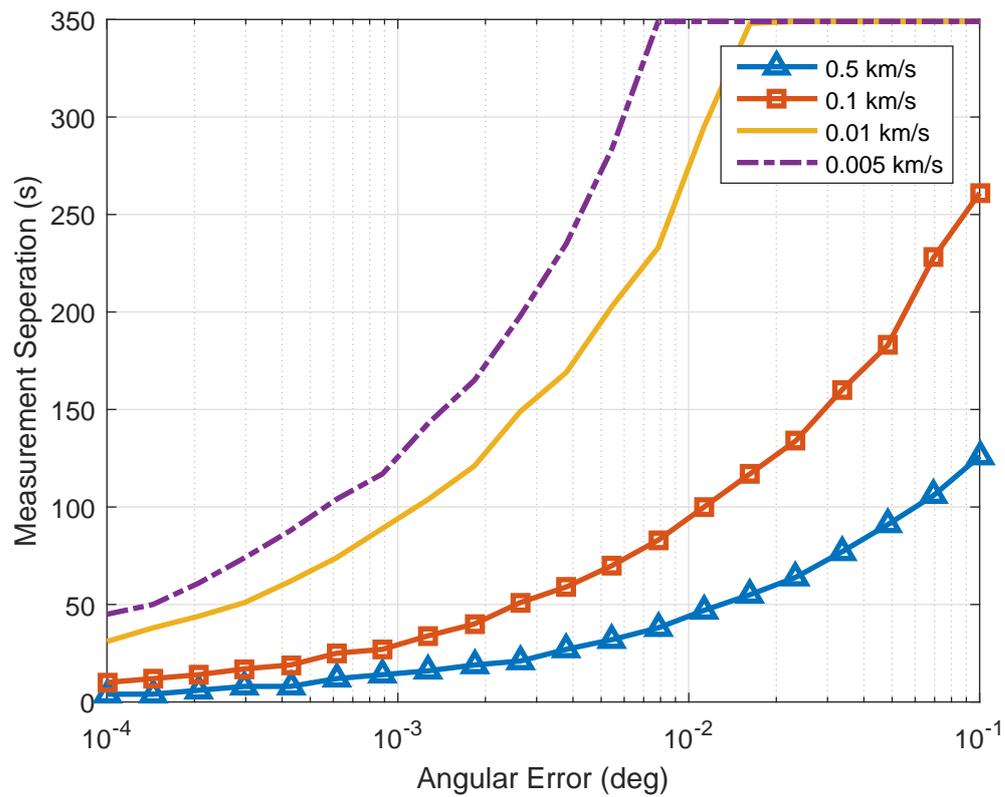


Figure 3.2.2: Minimum time separation to achieve a specified velocity uncertainty.

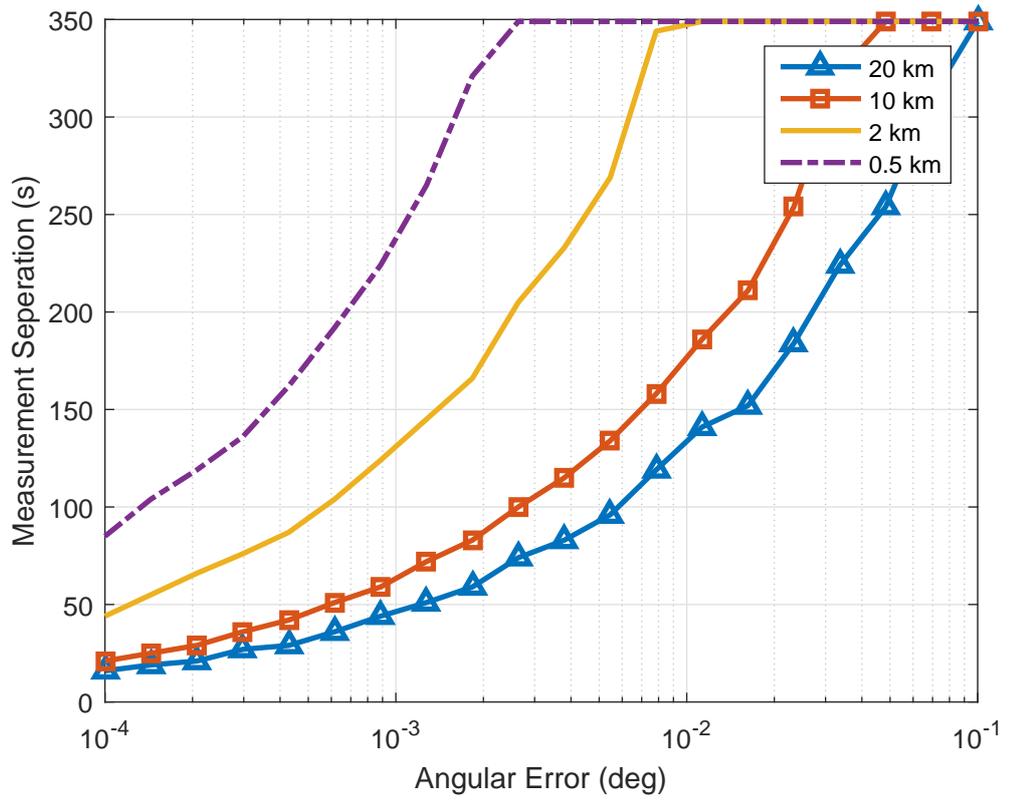


Figure 3.2.3: Minimum time separation to achieve a specified initial range uncertainty.

CHAPTER 4

ORBIT ESTIMATION USING DIFFERENT STATE ESTIMATORS

In this chapter, the problem of orbit estimation with sequential Bayesian estimators is studied. In Section 4.1 the details of the simulation scenario used in this section are given. Initialization of sequential estimators is investigated in Section 4.2. Comparison of estimators that are given in Sections 2.4.1, 2.4.2 and 2.4.3 will be made in terms of computational load and root mean square error (RMSE) performances in Section 4.3. Based on the performance and computational load of the filters, one of the algorithms will be proposed for further study. Then, for all different measurement types, the performance of the proposed estimator will be compared with the PCRLB.

4.1 Data Generation and Necessary Parameters

4.1.1 True Data Generation

For orbital simulations and data generation AGI's Systems Tool Kit (STK 8) is used. The following scenario parameters are used in STK:

- A LEO satellite at 700 km altitude and with the following physical properties is considered.
 - Dimensions: Cube with a side length of 1 m.
 - Mass: 100 kg

- Orbit data is generated using STK's High Precision Orbit Propagator with the following force modelling.
 - RK4 integrator with one second step size
 - 20×20 gravity model
 - Jacchia-Roberts atmospheric model (with a drag coefficient of 2.2)
 - Solar radiation pressure with a coefficient of $C_r = 1$ and dual cone shadow model.
 - Earth and ocean tides
 - Third body effects (Sun, Moon, Jupiter)
- Position and velocity samples (used as a reference to determine the RMSE) are generated in ECI frame.
- For the Doppler data, a transmitter is placed on the satellite with a carrier frequency of 8.15 GHz.
- Ground sensors are placed on the following coordinates in STK.
 - METU campus: Lat: 39.89° Lon: 32.77° Alt: 900 m
 - Gokceada: Lat: 40.23° Lon: 25.91° Alt: 200 m
- Measurements are taken in every second for all observation cases.
- For multiple station (two stations) case, it is assumed that there is perfect time synchronization between the stations. Only the measurements when the satellite is in the line of sight of both stations are utilized.

4.1.2 Noisy Data Generation

The noisy measurements for the simulations are generated in Matlab using the following parameters.

- Zero mean, white Gaussian noise is generated and added to measurements with the following standard deviations.
 - Full state observations: 10 m in position and 20 cm/s in velocity.

- Azimuth and Elevation observations: 10 millidegrees for Az and El .
- Range measurements: 50 m.
- Doppler measurements: 5 Hz.

4.1.3 Filter Parameters

Dynamic model which is used in the filters and PCRLB calculation contains two body model and perturbing accelerations coming from the Earth oblateness (i.e., J_2 effect) and drag with exponential air density described in Sections 2.1.1 and 2.1.2. Discrete form of the dynamic model is obtained using RK4 algorithm presented in Section 2.1.3.

Measurement models used in this chapter are given below. The measurement equations are constructed as described in Section 2.2 by using coordinate transformations presented in Section 2.3.

- Full State Observation
- AER Observation
- AE Observation
- AED Observation

Jacobians are taken numerically by using central differencing method given in (2.4.7). Small scalar ϵ is chosen as 10^{-8} km and 10^{-8} km/s for the states corresponding to the position and the velocity respectively.

It should also be mentioned that, it is necessary to use consistent measurement and process noise covariance matrices (i.e., R and Q) for Kalman filters to work properly. R matrices for each observation case can easily be constructed by using the aforementioned standard deviations. Determination of Q matrix is relatively complex since we do not exactly know the accuracy of the dynamic model used. However, in Figure 2.1.3, the order of magnitudes of the perturbing accelerations due to each force acting on the satellite can be seen. Let us consider

the following dynamic model.

$$X_{k+1} = f(X_k) + \underbrace{w_k}_{G\alpha_k} \quad (4.1.1)$$

where α_k represents the noise due to the perturbing accelerations with distribution $\mathcal{N}(0_{3 \times 1}, Q_\alpha)$ and G is the 6×3 coupling matrix between the state and the process noise. Mean of this distribution is just an assumption and the covariance matrix is chosen as $Q_\alpha = 10^{-8} I_{3 \times 3}$ by taking the effects of unmodeled forces into account. Since the state is in form $X_k = [x_k \ y_k \ z_k \ v_{x_k} \ v_{y_k} \ v_{z_k}]^T$, the coupling matrix G can be written as follows.

$$G = \begin{bmatrix} \frac{\delta^2}{2} & 0 & 0 \\ 0 & \frac{\delta^2}{2} & 0 \\ 0 & 0 & \frac{\delta^2}{2} \\ \delta & 0 & 0 \\ 0 & \delta & 0 \\ 0 & 0 & \delta \end{bmatrix}. \quad (4.1.2)$$

It should be noted that δ is the discretization time. Finally, the covariance matrix Q_w for the w_k can be found as follows.

$$Q_w = E \{w_k w_k^T\} = G Q_\alpha G^T. \quad (4.1.3)$$

4.1.4 Error Metrics

In order to measure the performance of the estimators Monte Carlo runs are used and RMS errors for the position and the velocities are calculated for each time k as follows.

$$\sqrt{\frac{1}{M} \sum_{i=1}^M \left\{ (\hat{x}_k^+(i) - x_k(i))^2 + (\hat{y}_k^+(i) - y_k(i))^2 + (\hat{z}_k^+(i) - z_k(i))^2 \right\}} \quad (4.1.4)$$

$$\sqrt{\frac{1}{M} \sum_{i=1}^M \left\{ (\hat{v}_{x_k}^+(i) - v_{x_k}(i))^2 + (\hat{v}_{y_k}^+(i) - v_{y_k}(i))^2 + (\hat{v}_{z_k}^+(i) - v_{z_k}(i))^2 \right\}} \quad (4.1.5)$$

where M is the number of the Monte Carlo runs; $[\hat{x}_k^+(i), \hat{y}_k^+(i), \hat{z}_k^+(i)]^T$ and $[x_k(i), y_k(i), z_k(i)]$ show the k^{th} position estimate and the k^{th} true position of

the i^{th} Monte Carlo run respectively. Similarly $[\hat{v}_{x_k}^+(i), \hat{v}_{y_k}^+(i), \hat{v}_{z_k}^+(i)]^T$ and $[v_{x_k}(i), v_{y_k}(i), v_{z_k}(i)]^T$ show the k^{th} velocity estimate and the k^{th} true velocity of the i^{th} Monte Carlo run respectively. It should be noted that, if the random trajectories are generated by only adding noise to the measurements then all the state elements are same for all i 's.

4.2 Filter Initialization

In Bayesian estimation, parameter to be estimated is treated as a random variable and prior knowledge about the parameter is required. As a result, in order to start a sequential filter (which is also a Bayesian scheme), initial values for the state and its covariance matrix based on our prior knowledge are required. Choosing accurate initial conditions is important for convergence issues especially if full state observations are not available or they can not be deduced from the available information.

4.2.1 Full State Observation Case

Filter initialization procedure is simplest in the full state observation case since the measurements contain all of the information (position, velocity and their uncertainties) to start the filter even if they are the transformed versions of the state and covariance matrix. Initial conditions can simply be obtained by transforming the first measurement and its covariance matrix. ECEF to ECI conversion, which is given in Section 2.3.5, is needed. This transformation is a simple matrix multiplication (actually construction of this matrix is not simple). Let the relation between the measurement and the state be

$$Y = HX \tag{4.2.1}$$

where Y is the 6×1 measurement vector and X is the 6×1 state vector. H is the transformation matrix from ECI to ECEF. It should be noted that the dimension of the state is 6 since it contains three position and three velocity components. Assuming that the first measurement is Y_0 and its covariance matrix is R , the

initial conditions for the filter can be found as follows.

$$\hat{X}_0 = H^T Y_0 \quad (4.2.2)$$

$$\begin{aligned} P_0 &= E \left\{ (X_0 - \hat{X}_0)(X_0 - \hat{X}_0)^T \right\} \\ &= H^T R H \end{aligned} \quad (4.2.3)$$

where X_0 is the initial state with estimate \hat{X}_0 and P_0 is the corresponding initial covariance matrix. It should be mentioned that since H is a rotation matrix, H^T shows the inverse rotation. Note that this measurement type is usually the most accurate one and it is in general used as a reference in order to calculate RMSE. This is because, in real life, true data is unavailable and one needs a ground truth in order to assess the performance of the designed filter or sensor.

4.2.2 AER Observation Case

This observation case is more complex than the full state observation case since velocity information is not available in the measurements. Position part of the state vector and its covariance matrix can be deduced by converting the first AER measurement to ECI coordinates. For the velocity part, there are several approaches for obtaining the initial velocity estimate and its covariance. Initialization methods for tracking applications can be found in [50]. The initialization methods proposed for AER case are given in the following subsections.

4.2.2.1 Two-Point Initialization

Two point differencing is one of the methods for velocity initialization and it uses successive position measurements and the time between them to compute an estimate of the initial velocity [50]. Another method for velocity initialization is one point initialization. Sometimes, filter designer have knowledge about the bounds on the velocity of the target and he/she can use the upper and lower bounds on velocity to form the distribution of the initial velocity estimate.

Let the initial estimate and the covariance be \hat{X}_0 and P_0 respectively. They can

be written as

$$\begin{aligned} \hat{X}_0 &= [\hat{x}_0 \quad \hat{y}_0 \quad \hat{z}_0 \quad \hat{v}_x^0 \quad \hat{v}_y^0 \quad \hat{v}_z^0]^T \\ P_0 &= \left[\begin{array}{c|c} P_0^p & 0_{3 \times 3} \\ \hline 0_{3 \times 3} & P_0^v \end{array} \right]_{6 \times 6} \end{aligned} \quad (4.2.4)$$

where P_0^p and P_0^v show the position and velocity part of the covariance matrix of the initial estimate. Let us say

$$p\hat{s}_0 = [\hat{x}_0 \quad \hat{y}_0 \quad \hat{z}_0]^T$$

and

$$v\hat{e}l_0 = [\hat{v}_x^0 \quad \hat{v}_y^0 \quad \hat{v}_z^0]^T.$$

In order to find the initial estimate for the position part, uncertainty transformation is needed, namely distribution of the AER measurements should be propagated through the nonlinear AER to ECI transformation. This can be done by representing the distribution of the first measurement with randomly chosen sample points and passing them through the nonlinear function individually. Let the transformation from AER to ECI be $h(\cdot)$ and assume that the first AER measurement is Y_0 with the covariance matrix $R_{3 \times 3}$. Then, producing M random measurements assuming that the first measurement has a Gaussian distribution with mean Y_0 and covariance R gives us the sampled form of its distribution. Passing these samples through the transformation

$$pos_0^i = h(Y_0^i), \quad i = 1, 2, \dots, M \quad (4.2.5)$$

and then calculating the mean and the covariance matrix of the transformed samples (pos_0^i) gives us the position related part of the initial conditions as follows.

$$\begin{aligned} p\hat{s}_0 &= \frac{1}{M} \sum_{i=1}^M pos_0^i \\ P_0^p &= \frac{1}{M} \sum_{i=1}^M (pos_0^i - p\hat{s}_0) (pos_0^i - p\hat{s}_0)^T. \end{aligned} \quad (4.2.6)$$

If two point differencing method is used, the first and second AER measurements are utilized in order to calculate the velocity estimate and its covariance matrix.

Since the velocity vector should be represented in ECI, nonlinear transform of the sampled distribution methodology explained above for position case should be used. Let the first and second AER measurements be Y_0 and Y_1 with same covariance matrix R then the initial velocity estimate and the covariance matrix can be calculated as follows.

$$\hat{vel}_0 = \frac{1}{M} \sum_{i=1}^M \left(\frac{h(Y_1^i) - h(Y_0^i)}{\delta} \right) \quad (4.2.7)$$

$$\begin{aligned} P_0^v &= \frac{1}{\delta^2} (P_0^p + P_1^p) \\ &= \frac{2H^T R H}{\delta^2} \end{aligned} \quad (4.2.8)$$

where δ is the time between two measurements Y_0 and Y_1 . The equation for P_0^v is directly written by using the assumption of conditionally independent measurements and using

$$\hat{vel}_0 = \frac{p\hat{s}_1 - p\hat{s}_0}{\delta} \quad (4.2.9)$$

and

$$p\hat{s}_1 = \frac{1}{M} \sum_{i=1}^M h(Y_1^i). \quad (4.2.10)$$

4.2.2.2 One-Point Initialization

One point initialization method uses the same uncertainty transformation technique (as introduced earlier) to construct the position part of the initial estimate and the covariance matrix. However, for the velocity part, this method uses the knowledge of the filter designer about the bounds on velocity. Since we are interested in LEO satellites in this thesis, we can say that the magnitude of the velocity of a LEO satellite can not exceed 8 km/s even for an altitude of 100 km. As a result, elements of the velocity vector take values in the interval of $(-8, 8)$ km/s. Let us construct a Gaussian distribution for a component of the velocity estimate with zero mean and standard deviation of $8/3$, namely $\mathcal{N}(0, (8/3)^2)$. Then it is guaranteed that the magnitudes of velocity components lie in between -8 and 8 with probability of 0.997. In conclusion, velocity part of the covariance matrix of the initial estimate for one point initialization case can be given as

follows.

$$\begin{aligned} \hat{v}e_l_0 &= \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \\ P_0^v &= \left(\frac{8}{3}\right)^2 I_{3 \times 3} \end{aligned} \tag{4.2.11}$$

where $I_{3 \times 3}$ is the identity matrix of size 3×3 .

4.2.2.3 One-Point Initialization with Velocity Direction Information

For one point initialization of AER measurement case, we also have the direction information of the satellite velocity. It is known that the velocity vector is tangential to the orbit ellipse (see Figure 4.2.1) and its uncertainty in the direction of \vec{r} is much smaller (can be assumed as ten times smaller) than the uncertainties lying on the T plane.

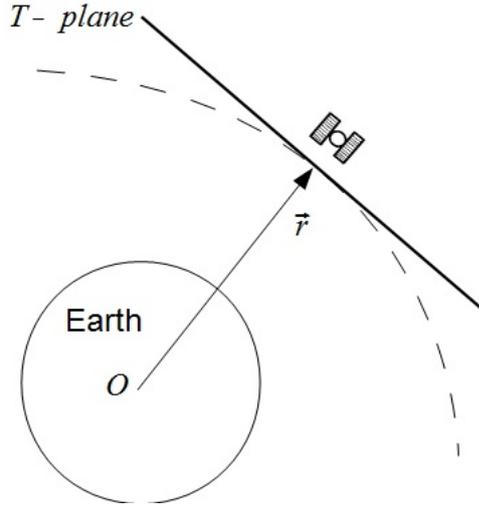


Figure 4.2.1: Illustration of orbit ellipse, its tangent and satellite position vector.

In order to construct the velocity part of the initial covariance matrix (i.e., P_0^v) containing aforementioned velocity direction information, matrix operations and some properties of diagonal matrices and orthogonal vectors can be used. Let e_1, e_2 and e_3 be three orthogonal unit vectors where e_1 is in the direction of \vec{r} , and e_2 and e_3 are lying on the T plane. Then we can construct the covariance matrix as follows.

$$P = \sum_{i=1}^3 \lambda_i e_i e_i^T \tag{4.2.12}$$

where λ_i 's are corresponding to the velocity uncertainties (variances) in e_1, e_2, e_3 directions. For orthogonal unit vectors it is also known that

$$\sum_{i=1}^3 e_i e_i^T = I_{3 \times 3}. \quad (4.2.13)$$

Letting $\lambda_1 = \left(\frac{8}{30}\right)^2$ and $\lambda_2 = \lambda_3 = \left(\frac{8}{3}\right)^2$ (4.2.12) becomes

$$P = \left(\frac{8}{30}\right)^2 e_1 e_1^T + \left(\frac{8}{3}\right)^2 (I_{3 \times 3} - e_1 e_1^T). \quad (4.2.14)$$

4.2.2.4 Initialization Method Comparison

In this section, performances of the two point initialization, one point initialization and one point initialization with direction information are compared. For this purpose, the results of these initialization methods are used as initial conditions to start EKF's and the performance of the corresponding EKF's are compared. For the comparison, first, a measurement sequence of 300 true AER measurements is generated with the parameters given in Section 4.1 for the observing station located at METU Campus. A sample true trajectory is given in Figure 4.2.2. In order to carry out Monte Carlo simulations, 100 random trajectories are generated by adding zero mean white Gaussian noise to the generated true AER trajectory as follows.

$$Y_k^i = Y_k^{true} + v_k^i, \text{ for } k = 1, \dots, 300 \text{ and } i = 1, \dots, 100 \quad (4.2.15)$$

where Y_k^i is the k^{th} noisy AER measurement corresponding i^{th} Monte Carlo run and Y_k^{true} is the true AER measurement at time k . Y_k^i and Y_k^{true} are vectors of size 3×1 . v_k^i is the measurement noise with distribution $\mathcal{N}(0_{3 \times 1}, R)$ where $0_{3 \times 1}$ is a vector of size 3×1 with all its elements equal to zero and R is the measurement covariance matrix formed with standard deviations given in Section 4.1, i.e., we have

$$R = \begin{bmatrix} (0.01)^2 & 0 & 0 \\ 0 & (0.01)^2 & 0 \\ 0 & 0 & (0.05)^2 \end{bmatrix}. \quad (4.2.16)$$

The number of random samples generated for initializing the position states is 100, i.e., $M = 100$ in (4.2.6). The EKFs run with a prediction step size of 1 second. The process noise covariance is used as it is described in (4.1.2) with a discretization step size of 1 second (i.e., $\delta = 1$).

In Figure 4.2.3, the performance of the aforementioned three initialization techniques for AER measurements is compared. Except the first few estimates, the RMSE performances of the corresponding EKFs are approximately the same. The difference in the velocity RMSE's seen in Figure 4.2.3 is completely caused by the first estimate since the initial velocity estimate and its covariance are not very good for one point initialization cases. For one point initialization case, taking the direction information of the satellite velocity into account improves the initial performance of the estimator. Although the initial performances of all three initialization method differ, they all give satisfactory results. This is because, both position and velocity information can be deduced from the successive measurements of AER. In other words, there is no observability problem. As seen in the figure, the RMS errors of all filters converge to the same values.

One can apply on-line filtering when AER measurements are available. Initialization can be done by using the first two measurements. When the measurement noise is high, initial position estimates is not very good for all of the aforementioned initialization techniques. For the two point initialization method, velocity estimate can be very bad since it is found by differencing the positions. It should be noted that differencing amplifies the high frequency noise. Therefore, if the measurement noise is low then the use two point initialization is reasonable and if the noise is high then using one point initialization with velocity direction information is more convenient.

4.2.3 AE Observation Case

Initialization of the filter when angles only (i.e., azimuth and elevation) measurements are available is more problematic than the AER observation case. This is because, measurements give only the direction of the line of sight vector. Hence,

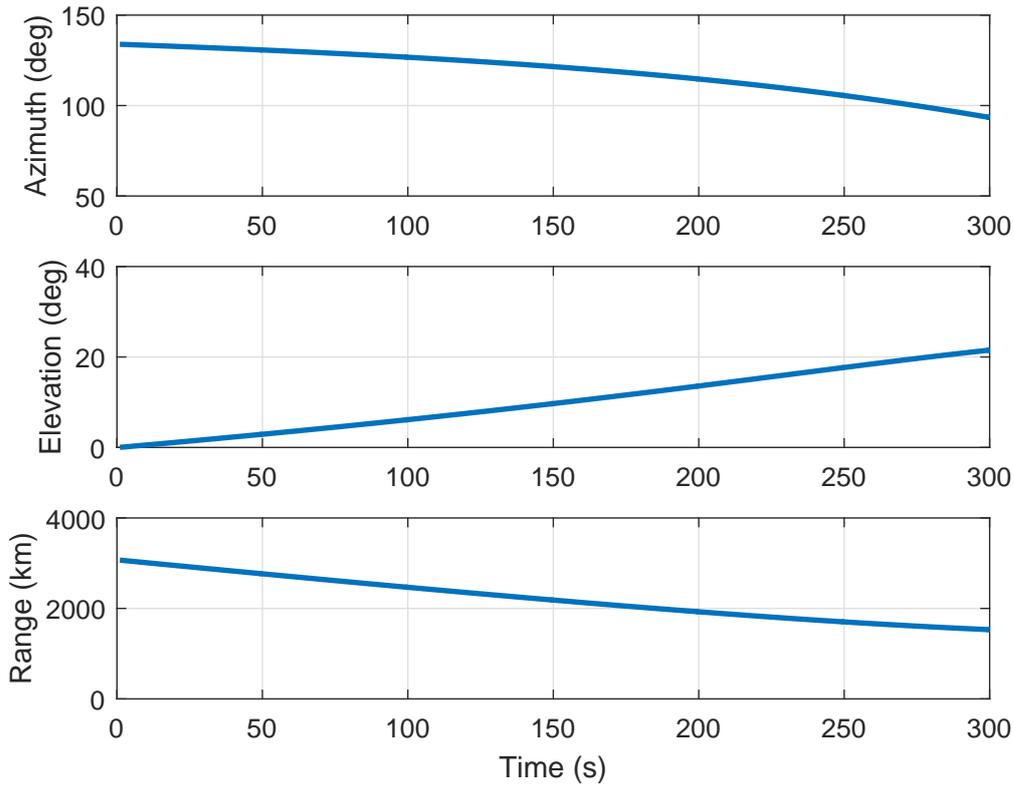


Figure 4.2.2: A sample AER trajectory.

we should extract range data by using a few samples of azimuth and elevation angles. The initialization methods proposed for this case are described in the following subsections.

4.2.3.1 One Point Initialization

This method is the same as the one given in Section 4.2.2.2 except that the range information is obtained using Gauss' method described in Section 3.1. Let us have a sequence of angles only measurement vectors $\{Y_k\}_{k=0}^{2L}$ where Y_k is the k^{th} measurement vector consisting of azimuth and elevation. In order to find the range corresponding to Y_0 , we need to select three of the measurements. It is better to choose the first, the middle and the last measurements (i.e., Y_0 , Y_L and Y_{2L}) since the higher separation times yield more accurate results as seen in Section 3.2. In order to find the range estimate ρ_0 and its covariance, we

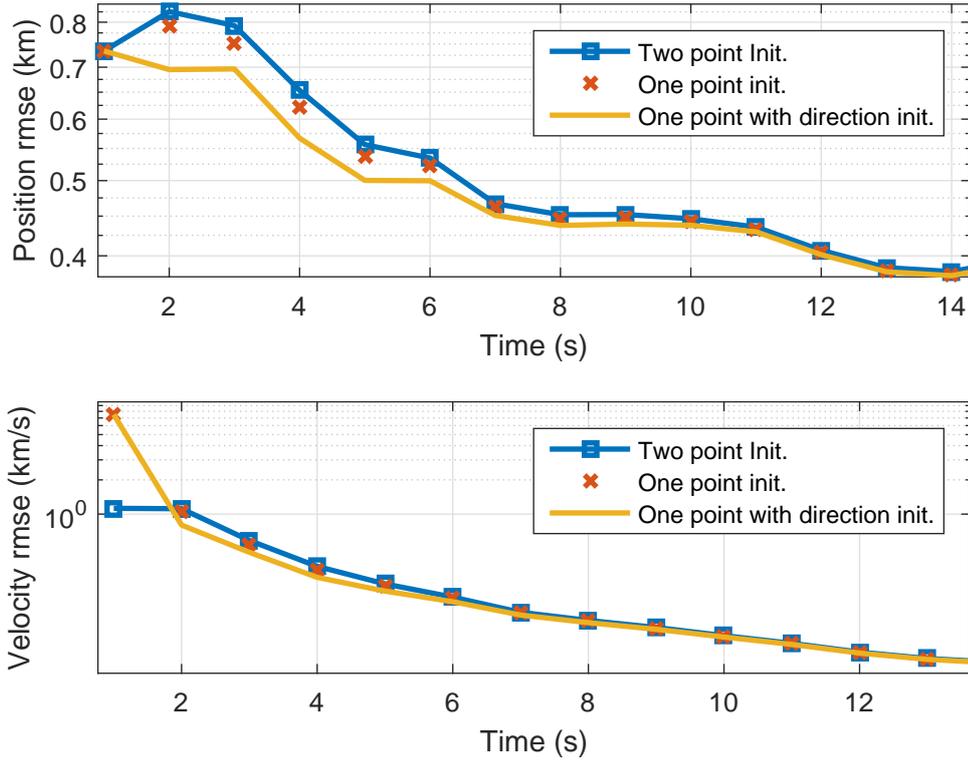


Figure 4.2.3: RMSE performances of EKFs using different initialization methods (the first 14 samples).

generate random samples Y_0^i, Y_L^i, Y_{2L}^i , $i = 1, \dots, M$, distributed as follows.

$$\begin{aligned}
 Y_0^i &\sim \mathcal{N}(Y_0, R), \\
 Y_L^i &\sim \mathcal{N}(Y_L, R), \\
 Y_{2L}^i &\sim \mathcal{N}(Y_{2L}, R),
 \end{aligned} \tag{4.2.17}$$

where R denotes the measurement noise covariance for the azimuth and elevation measurements. Each of the random samples $Z^i \triangleq \{Y_0^i, Y_L^i, Y_{2L}^i\}$ are then passed through the Gauss' algorithm as follows.

$$\begin{bmatrix} X_L^i \\ \rho_0^i \\ \rho_L^i \\ \rho_{2L}^i \end{bmatrix} = G(Z^i, t_0, t_L, t_{2L}, OS) \tag{4.2.18}$$

for $i = 1, \dots, M$ where

- $G(\cdot)$ is the function corresponding to Gauss' algorithm;
- X_L^i is the 6×1 state vector corresponding to time t_L containing the position and the velocity estimates;
- ρ_0^i , ρ_L^i and ρ_{2L}^i are ranges between the satellite and the observing station at times t_0 , t_L and t_{2L} respectively;
- OS is the Earth fixed coordinates (geodetic latitude, longitude and altitude) of the observing station.

After obtaining $\{\rho_0^i\}_{i=1}^M$, the estimate for ρ_0 and its covariance P_{ρ_0} can be calculated as follows.

$$\hat{\rho}_0 = \frac{1}{M} \sum_{i=1}^M \rho_0^i, \quad (4.2.19)$$

$$P_{\rho_0} = \frac{1}{M} \sum_{i=1}^M (\rho_0^i - \hat{\rho}_0)^2. \quad (4.2.20)$$

After finding the estimate $\hat{\rho}_0$ and the covariance P_{ρ_0} , the initialization problem turns into initialization with a single AER measurement as described in Section 4.2.2.4. The velocity estimate and its covariance are chosen as in (4.2.11).

4.2.3.2 Middle Point Initialization

This method only applies Gauss' method of initial orbit determination using three sets of sufficiently separated angles. Since the Gauss' method supplies the state estimate for the middle measurement of the selected azimuth and elevation triple, the initialization is made for the middle measurement, i.e., the time stamp of the resulting initial state corresponds to the middle measurement. Hence we call this method as the middle point initialization. In order to find the initial estimate and its covariance, the same type of samples ($\{Y_0^i, Y_L^i, Y_{2L}^i\}_{i=1}^M$) are generated from Y_0 , Y_L and Y_{2L} as given in (4.2.17). The initial state \hat{X}_L and

its covariance P_L are then calculated as

$$\hat{X}_L = \frac{1}{M} \sum_{i=1}^M X_L^i \quad (4.2.21)$$

$$P_L = \frac{1}{M} \sum_{i=1}^M \left(X_L^i - \hat{X}_L \right) \left(X_L^i - \hat{X}_L \right)^T \quad (4.2.22)$$

where X_L^i is defined in (4.2.18). Afterwards, the resulting estimate and its covariance is used to initialize the filters. When the angular measurement errors and the time separation are good, it is seen that the Gauss' method gives the estimate with 3-D position and velocity standard deviations of a kilometer and a few meters per second. For such a small covariance matrix coming from the uncertainty transformation given in (4.2.22), scaling of the resulting covariance can be useful. It should be mentioned that in this method, the measurements up to the middle measurement (i.e., Y_0, \dots, Y_{L-1}) are not used in the filtering algorithms and this can reduce the estimation performance.

4.2.3.3 Smoother Initialization

Smoothed version of the initial estimate can also be used as an initial condition to start the filter. In order to find the smoothed estimate and its covariance, the RTS smoother algorithm (described in Section 2.4.5) is applied. The RTS smoother is initialized using the initial condition obtained from the one point initialization method given in Section 4.2.3.1. The smoothed estimate is used to initialize the filter. The initial covariance should be scaled to that the the 3-D standard deviations for the position and the velocity should be on the order of a kilometer and a few tens of meters per second respectively. The reason for not using the smoother covariance directly as the initial covariance is that the covariance of the smoothed estimate can be very small and the filter might not care about the future measurements if this covariance is used as the initial covariance.

4.2.3.4 Initialization Method Comparison

In this section, performances of the one point initialization, middle point initialization and smoother initialization are compared. For this purpose, the results of these initialization methods are used as initial conditions to start EKFs and the performance of the corresponding EKFs are compared. For this purpose, a measurement sequence of 601 true AE measurements ($2L = 600$), which corresponds to satellite visibility duration of 601 seconds and sampling period of 1 second, is generated. The noisy measurements (Y_0, Y_1, \dots, Y_{600}) are generated by adding zero mean white Gaussian noise with a standard deviation of 0.01 degrees for both azimuth and elevation as described in Section 4.2.3.1. A total of 100 Monte Carlo simulations are carried out by changing the measurement noise realizations in each run. For each Monte Carlo run, 200 samples (i.e., $M = 200$ in (4.2.18)) are generated while applying the initialization methods.

In the simulations, the initial covariance matrix scaling factors for the one point (γ_{op}), the middle point (γ_{mp}) and the smoother initialization (γ_{sm}) cases are given below respectively.

$$\gamma_{op} = 5, \quad \gamma_{mp} = 20, \quad \gamma_{sm} = 100.$$

It should be noted that the scaling factor, for the one point case, is used only for scaling the position covariance since the worst case covariance is chosen for the velocity. Other scaling factors are applied to both the position and the velocity.

Relative performances of the initialization methods can be seen in Figure 4.2.4. In the Figure 4.2.4, it is seen that although the middle point initialization has smaller initial error (than the other two initialization methods), the RMS errors of the EKF initialized using this initialization technique get smaller quite slowly. There is no considerable difference between the estimation performances of the one point initialization and smoother initialization. However, by changing the the scaling factors one can obtain different results. Table 4.2.1 shows the time averaged RMS estimation error performances of the three initialization techniques, where similar observations can be made.

For the AE case initialization, collection of a long sequence of azimuth and ele-

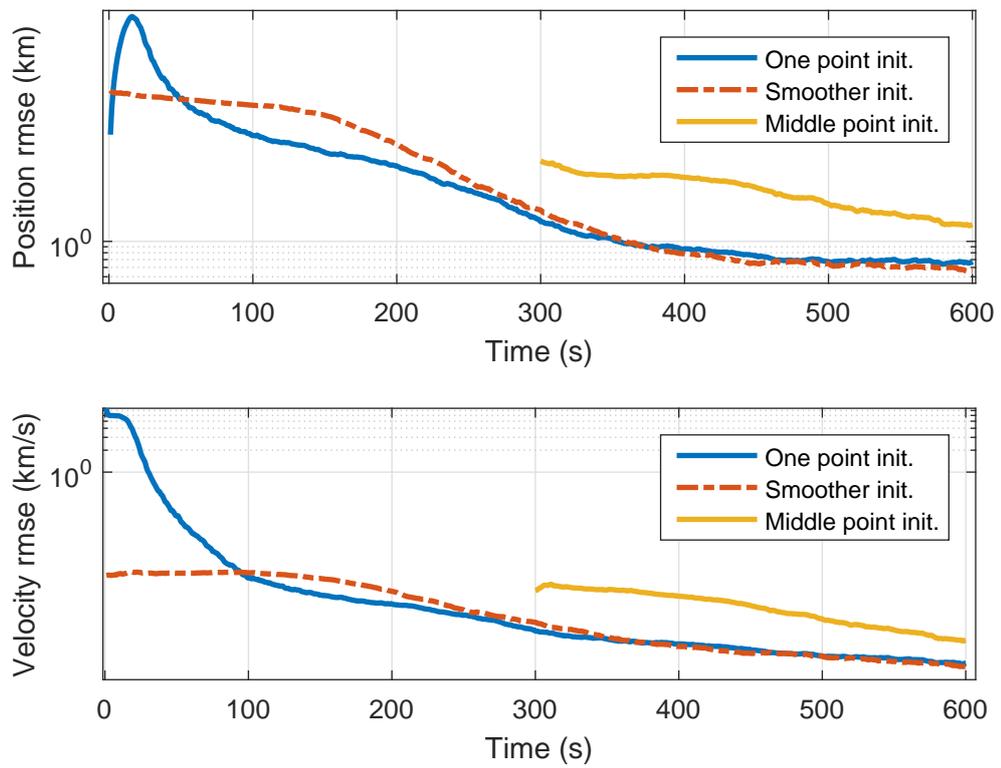


Figure 4.2.4: Effect of initialization method to estimator performance for AE measurement case.

Table 4.2.1: Comparison of initialization techniques for the AE case by time averaged RMSE's using last 50 estimates.

Initialization method	RMSE	
	Position (km)	Velocity (m/s)
One point init.	0.68	2.4
Middle point init.	1.43	5.2
Smoother init.	0.60	1.3

vation measurements is needed since all of the proposed initialization techniques use Gauss' method to find the estimates and their covariances. Therefore, on-line filtering is not applicable if we do not have reasonable prior information. Among the three initialization methods proposed, smoother initialization seems to be the most promising one.

4.2.4 AED Observation Case

For the azimuth-elevation-Doppler (AED) measurement case, existence of Doppler data gives extra information about the velocity of the satellite (i.e., the rate of change of the range between the satellite and the observing station). Therefore, initialization techniques used for AE measurement case can also be used and work better for the AED case.

4.3 Estimator Comparison

In this section, the recursive estimation methods EKF, UKF and CD-EKF are used for the nonlinear orbit estimation problem. Their computational complexities and error performances are compared and one of them is proposed for further analysis. In addition, the effects of initial uncertainties on the RMS error

performances are also investigated.

4.3.1 Relative Filter Performances

There are many observation types for satellite orbit determination such as full state observation, angles only observation, angles and range observation etc. and we know that (from Sections 2.2 and 2.3) the measurement equations are nonlinear except the full state observation case. In order to compare filter performances in terms of computation times and RMSE's for both linear and nonlinear measurement equation cases, full state and AER observations cases are chosen as sample scenarios.

For the full state observation case, a sequence of 200 true measurements with 1 second sampling period is generated in ECI frame by using STK as described in Section 4.1. This corresponds to the following measurement equation.

$$Y_k^{true} = X_k^{true} \quad (4.3.1)$$

where X_k^{true} is the 6×1 state vector consisting of position and velocity components and Y_k^{true} is the true measurement vector. It should be mentioned that the measurement matrix (usually represented by H) is the identity matrix of size 6×6 for this case since the measurements are generated directly in ECI frame. 200 Monte Carlo simulations are carried out for each filtering method in order to obtain the statistical properties of the estimation errors. In order to make the Monte Carlo runs, 200 noisy measurement sequences are generated as follows.

$$Y_k^i = Y_k^{true} + v_k^i \quad (4.3.2)$$

for $i = 1, \dots, 200$ and $k = 1, \dots, 200$ where Y_k^i is the k^{th} noisy measurement vector of size 6×1 and v_k^i is the measurement noise vector for the k^{th} measurement and i^{th} Monte Carlo run. It should be noted that

$$v_k^i \sim \mathcal{N}(0_{6 \times 1}, R) \quad (4.3.3)$$

where R is the measurement covariance matrix selected as

$$R = \begin{bmatrix} 10^{-4} I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 4 \times 10^{-8} I_{3 \times 3} \end{bmatrix}. \quad (4.3.4)$$

In Figure 4.3.1, a sample sequence of true measurements of x-position and x-velocity (in ECI frame) is shown. For AER measurement case, a sequence of 200

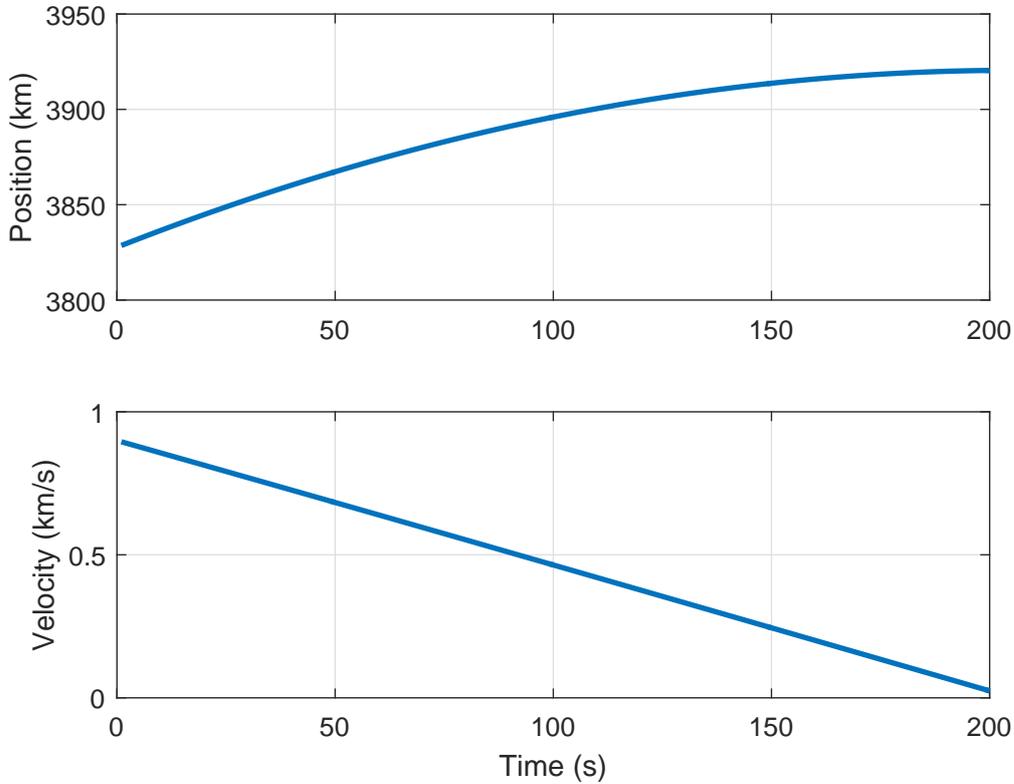


Figure 4.3.1: Sample full state measurements.

true AER measurements are obtained for the same true states as in the full state observation case (in ECI frame) are generated by using the STK with 1 second sampling period. Similar to the full state observation case, 200 random sequences of AER measurements are obtained by adding white noise to measurements as in (4.3.2). In this case, the dimension of the measurement and the noise vectors

are 3×1 and the noise component has the following distribution.

$$v_k^i \sim \mathcal{N} \left(0_{3 \times 1}, \begin{bmatrix} (0.01)^2 & 0 & 0 \\ 0 & (0.01)^2 & 0 \\ 0 & 0 & (0.05)^2 \end{bmatrix} \right) \quad (4.3.5)$$

where the units of the diagonal elements of the covariance matrix are *degree*² (for the first and second elements) and *km*² (for the third element).

For both measurement cases, the initial estimates \hat{X}_0 of the filters are chosen randomly. For each Monte Carlo run the initial states of the filters are distributed as

$$\hat{X}_0 \sim \mathcal{N}(X_0^{true}, P_0) \quad (4.3.6)$$

where X_0^{true} is the true initial state and P_0 is the covariance matrix of initial estimate which is given as follows.

$$P_0 \triangleq \begin{bmatrix} (2.5)^2 I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & (0.05)^2 I_{3 \times 3} \end{bmatrix}. \quad (4.3.7)$$

For UKF, the parameters which determine the weights of each sigma point used for the calculation of the mean and covariance matrices are chosen as follows.

$$\begin{aligned} \alpha &= 1.15, \\ \beta &= 0.3225, \\ \kappa &= 0. \end{aligned} \quad (4.3.8)$$

The parameters above and their meanings are explained in Section 2.4.3.

In Table 4.3.1, filter performances in terms of computation times of 200 Monte Carlo runs are compared. In the table, the step size shows the time between successive predictions in seconds. It should be noted that the measurement updates are carried out in at every second when the measurements arrive. The following explanations can be given for the results in Table 4.3.1:

- The measurement equations of the full state measurement case are linear while the AER measurement equations are nonlinear. Therefore, the measurement prediction for the AER case takes more time to compute. For

Table 4.3.1: Filter computation times.

		Computation time (s)	
Filter type	Step size	Full state	AER
EKF	1 s	119	3111
UKF	1 s	130	5317
CD-EKF	1 s	56	2864
	0.5 s	108	2945
	0.1 s	511	3474

the computation times of the measurement updates, our primary interest is the computation of the measurement functions. In the linear measurement case, computation times of the state and the covariance updates are similar for all of the filters. In the AER measurement case, computation of ECI to AER transformation is nonlinear and takes more time as expected.

- Let us assume that the continuous time dynamic system is represented by $f_c(\cdot)$ and its discretized version (with RK4) is represented by $f_d(\cdot)$. In CD-EKF, continuous time dynamic model is used and this actually reduces the computational effort of CD-EKF compared to EKF. This is because in EKF in order to discretize the dynamic model (i.e., in order to obtain $f_d(\cdot)$), RK4 is used and this requires the computation of system function ($f_c(\cdot)$) four times in each prediction step. On the other hand, in CD-EKF $f_c(\cdot)$ is computed only once for each prediction step which makes the EKF slower than the CD-EKF for each prediction step.

To clarify numerically, let us assume that in the CD-EKF, prediction is completed in two steps while in the EKF, it is done in a single step.

- In EKF,

- * State prediction: $f_d(\cdot)$ is evaluated once (4 evaluations of $f_c(\cdot)$). Prediction equations are given in (2.4.5).
- * Covariance prediction requires numerical Jacobian of $f_d(\cdot)$. This, for a 6th order system requires 12 evaluations of $f_d(\cdot)$ (48 evaluations of $f_c(\cdot)$).
- In CD-EKF (prediction step size is half of the step size used in EKF),
 - * State prediction: $f_d(\cdot)$ is evaluated twice (8 evaluations of $f_c(\cdot)$).
 - * Covariance prediction: $f_c(\cdot)$ is evaluated 24 times for the numerical Jacobian.

Hence, for this example, prediction operations require 32 and 52 evaluations of $f_c(\cdot)$ in CD-EKF and EKF respectively.

Finally, it can be said that, UKF is the slowest algorithm for the AER observation case, since it propagates $2n + 1$ ($n = 6$ for our case) sigma points through the nonlinear state and measurement functions in every measurement and prediction update.

Filter performances in terms of time averaged RMSE's can be seen in Tables 4.3.2 and 4.3.3. RMSE plots for the position and velocity estimates are given in Figures 4.3.2 and 4.3.4 respectively. Since the RMSE difference between the filters can not be seen easily from Figures 4.3.2 and 4.3.4, their zoomed versions are also given in Figures 4.3.3 and 4.3.5 respectively.

It can be concluded from the figures that if the estimators have the same prediction step sizes then they have approximately the same RMSE for both position and velocity. Furthermore, it can be said that decreasing the prediction step size also decreases the RMSE by looking at the Figures 4.3.5 and 4.3.3. Hence, if the computation time is not a problem (i.e., there is no time limitation) and if we need more accurate estimates, using CD-EKF with lower prediction step sizes may be advantageous. However, computation time is limited for many practical cases. As a compromise, we can say that CD-EKF with 0.5 second prediction step size is preferable in terms of both computation time and RMSE performance. As a conclusion of this work, we propose to use CD-EKF as the

Table 4.3.2: Full state observation case: Time averaged RMSE performances of last 50 estimates.

		RMSE	
Filter type	Step size	Position (m)	Velocity (cm/s)
EKF	1 s	2.17	2.68
UKF	1 s	2.17	2.68
CD-EKF	1 s	2.17	2.68
	0.5 s	2.13	2.29
	0.1 s	2.09	2.05

Table 4.3.3: AER observation case: Time averaged RMSE performances of last 50 estimates.

		RMSE	
Filter type	Step size	Position (m)	Velocity (m/s)
EKF	1 s	71.4	0.70
UKF	1 s	71.4	0.70
CD-EKF	1 s	71.4	0.70
	0.5 s	68.6	0.57
	0.1 s	64.5	0.43

preferred method of choice for the orbit determination problem. Selection of the prediction step size in general has to be made by considering both the available computation resources and allowable RMSE performance.

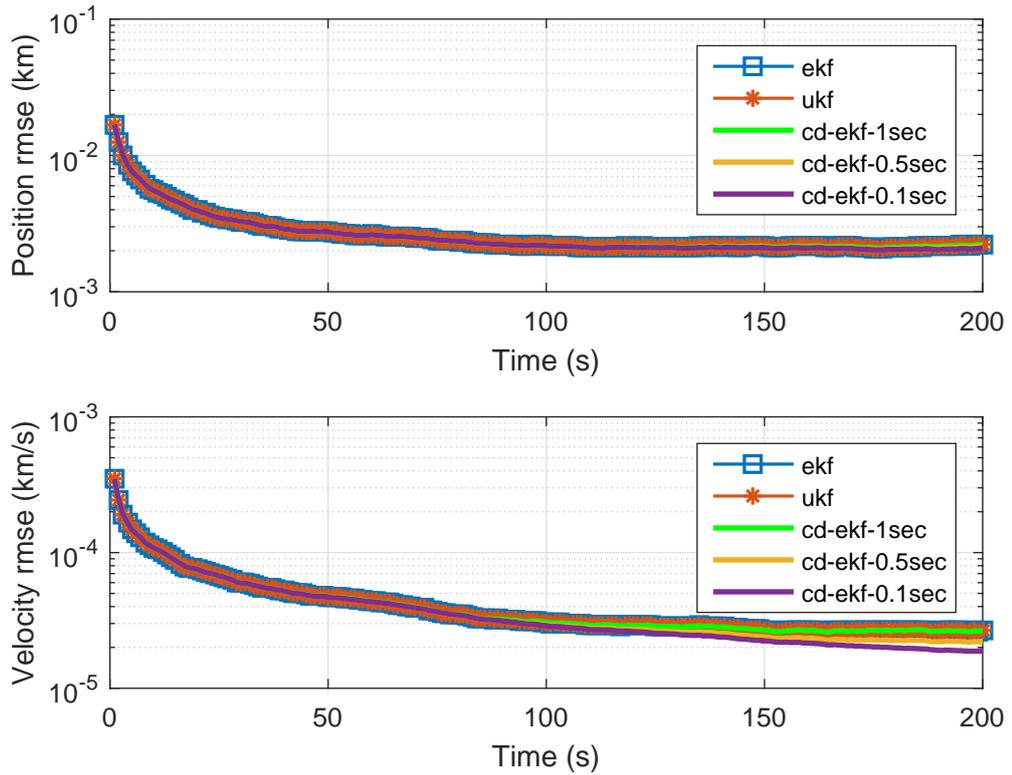


Figure 4.3.2: RMSE performance of filters with full state measurements.

4.3.2 Effects of the Initial Uncertainties

In order to investigate the effects of the initial uncertainties on the estimator performance, in this subsection, EKF and UKF are used with different initial uncertainties for the angles only (i.e., AE) measurement case. For the full state measurement case, AER and AED observation cases, performances of EKF and UKF are expected to be similar since the initial estimates are relatively good for these cases. Therefore, in this section, only AE observation case is considered since the angles only measurements give less information than the other types of measurements mentioned in this thesis. Since the behaviour and the mechanism of the CD-EKF resembles EKF, only the error performance of the EKF is compared with the UKF in this section.

A sequence of 300 true AE measurements and true states (in ECI coordinates) are generated by using STK, which corresponds to the satellite visibility duration of 300 seconds with 1 second sampling time. A total of 100 Monte Carlo runs are

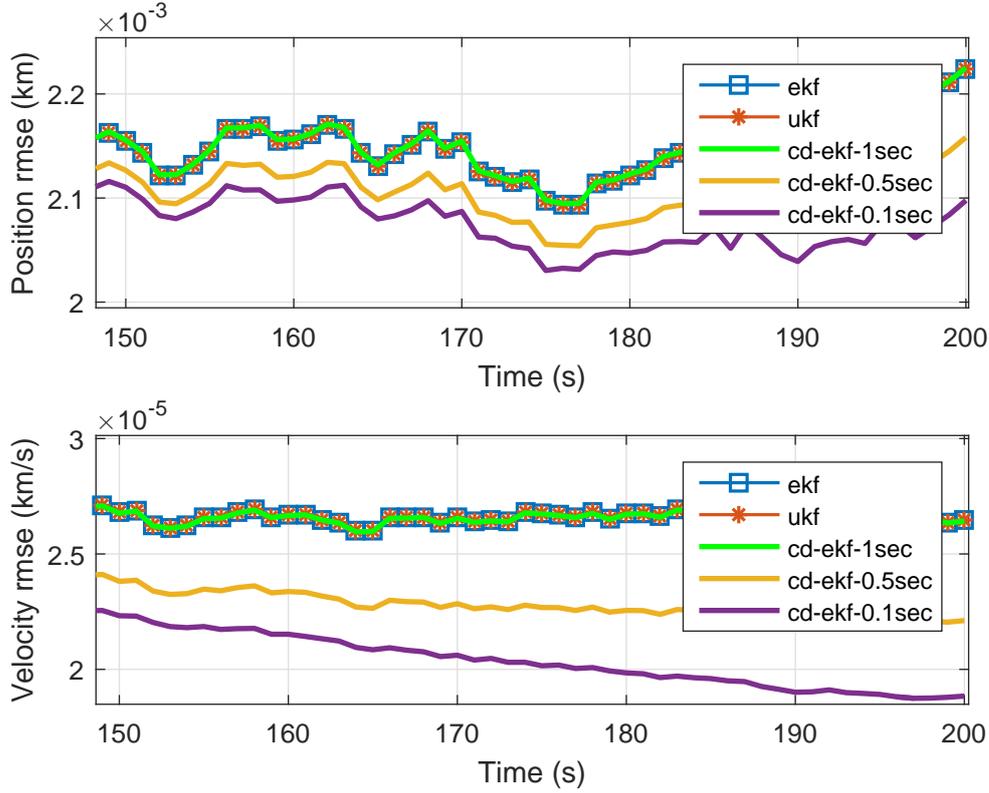


Figure 4.3.3: Zoomed version of Figure 4.3.2.

made with the filters. For this purpose, 100 random sequences of measurements are produced by adding zero mean white Gaussian noise (with standard deviation of 0.01 degrees for both azimuth and elevation) to the true measurements. For each Monte Carlo run, a random initial estimate which is distributed with the following distribution is used.

$$\hat{X}_0^i \sim \mathcal{N}(X_0^{true}, P_0^1) \quad (4.3.9)$$

where \hat{X}_0^i shows the initial estimate for the i^{th} Monte Carlo run; X_0^{true} is the true initial state and P_0^1 is the covariance matrix of the initial estimates. In order to see the effects of the different quality initial estimates, the same Monte Carlo procedure is carried out two more times for the initial estimate distributions $\mathcal{N}(X_0^{true}, P_0^2)$ and $\mathcal{N}(X_0^{true}, P_0^3)$. The covariance matrices P_0^1 , P_0^2 and P_0^3 are

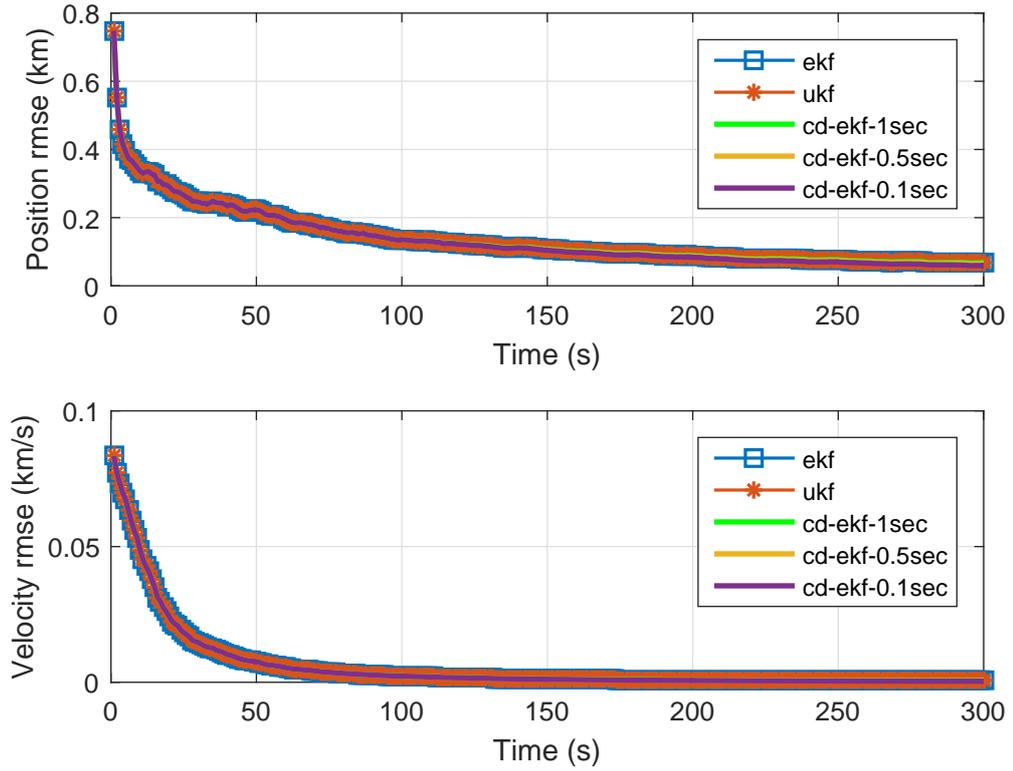


Figure 4.3.4: RMSE performance of filters with AER measurements.

given as

$$P_0^1 = \begin{bmatrix} (20)^2 I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & (3)^2 I_{3 \times 3} \end{bmatrix}, \quad P_0^2 = \begin{bmatrix} (50)^2 I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & (3)^2 I_{3 \times 3} \end{bmatrix},$$

$$P_0^3 = \begin{bmatrix} (150)^2 I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & (3)^2 I_{3 \times 3} \end{bmatrix}$$

For an Earth orbiting satellite, we know that the magnitude of the velocity can not exceed 8 km/s. In Section 4.2.2, it is mentioned that the velocity part of the initial estimate can be chosen according to the following distribution.

$$\mathcal{N} \left(0_{3 \times 1}, \left(\frac{8}{3} \right)^2 I_{3 \times 3} \right).$$

Therefore, choosing a velocity uncertainty of 3 km/s is reasonable and it also corresponds to the worst case for the velocity estimate. In addition, initial position

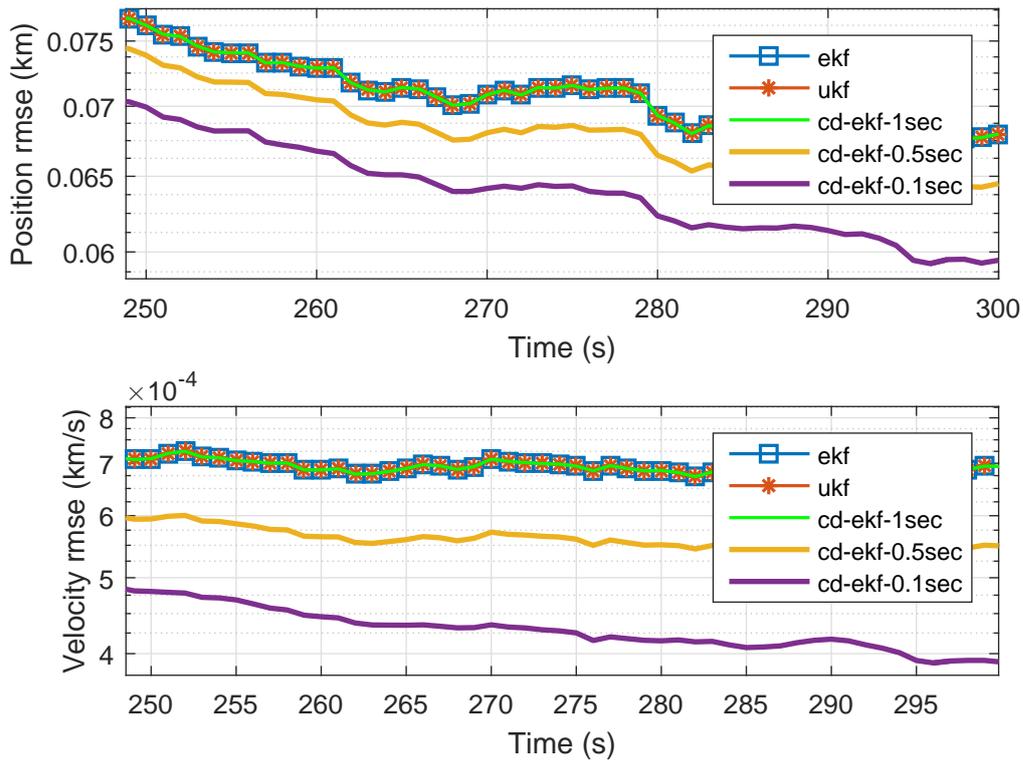


Figure 4.3.5: Zoomed version of Figure 4.3.4.

uncertainties used in this analysis are also much greater than the uncertainties coming from the Gauss' method.

In Table 4.3.4, time averaged RMSE's of the EKF and UKF for the last 100 estimates are shown. It can be concluded from the table that if the position and velocity uncertainties of the initial estimate are less than 20 km and 3 km/s then the error performances of the EKF and UKF will be approximately the same. However, in the case of high initial position uncertainty (corresponding to P_0^3), UKF significantly improves the RMSE when compared to the EKF. RMSE performances of the filters starting from the initial estimate with covariance matrix P_0^3 can be seen in Figure 4.3.6. Similar observations were made before in the literature in [17].

Table 4.3.4: Effects of the initial uncertainty on the RMSE performance.

Initial Cov.	RMSE			
	EKF		UKF	
	pos. (km)	vel. (km/s)	pos. (km)	vel. (km/s)
P_0^1	2.800	0.011	2.793	0.011
P_0^2	4.074	0.015	3.390	0.013
P_0^3	31.140	0.119	6.646	0.025

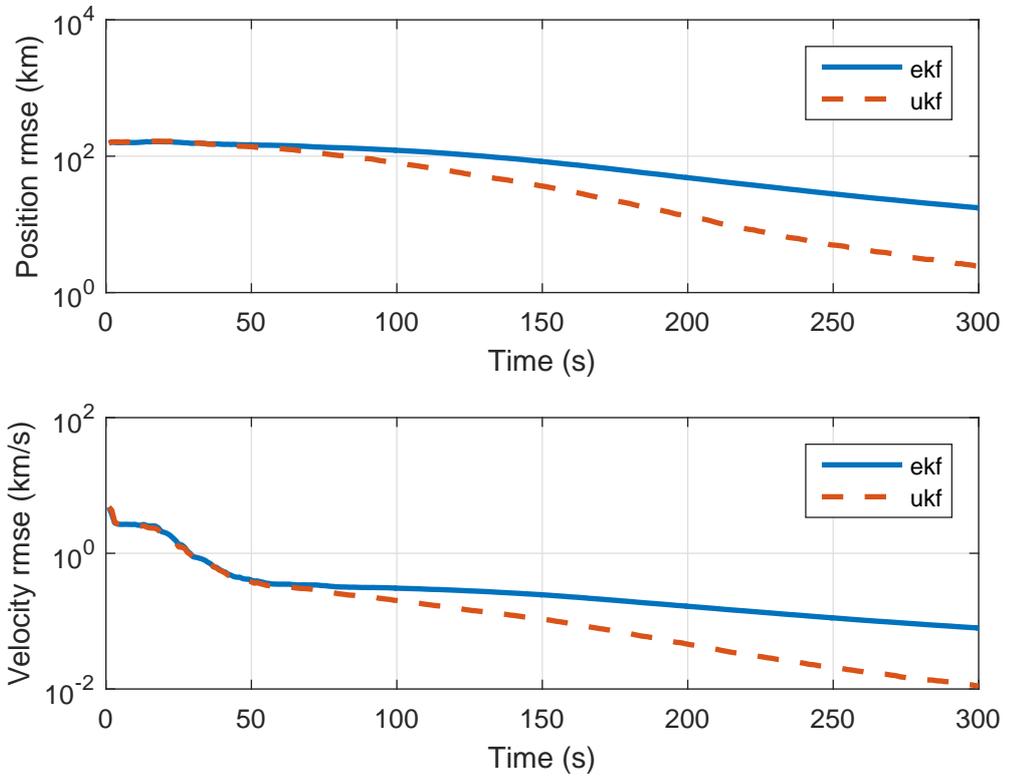


Figure 4.3.6: RMSE performances of the filters with initial covariance P_0^3 .

4.4 Estimator Performance with Respect to PCRLB

In Section 4.3.1 relative error performances of various types of estimators were compared and CD-EKF which had good RMSE performance with relatively low

computational complexity was proposed as the preferred estimator for the orbit determination problem. In this section, absolute performance (comparison of the error performance with the PCRLB) of CD-EKF will be investigated for different measurement types. In addition to this analysis, for the AE, AED and AER measurement cases, the estimation performance gain coming from a second observing station is also investigated.

For the PCRLB calculation, M random trajectories and M measurement realizations are generated for all of the measurement cases according to following state and measurement equations using the discretized version of the dynamic model given in (2.1.15). It should be noted that the states are generated by discrete model but in two steps as in the CD-EKF prediction.

$$X_{k+1} = f(X_k) + w_k \quad (4.4.1)$$

$$Y_k = h(X_k) + v_k \quad (4.4.2)$$

for $k = 0, \dots, 499$ where w_k and v_k are zero mean white Gaussian noises with covariances Q and R respectively. $h(X_k)$ represent the measurement function which depends on the observation type. For the simulations which contain two observing stations, measurement equations are constructed as follows assuming that there are perfect time synchronization between the stations.

$$Y_k = \begin{bmatrix} h^1(X_k) \\ h^2(X_k) \end{bmatrix} + \begin{bmatrix} v_k^1 \\ v_k^2 \end{bmatrix} \quad (4.4.3)$$

where superscripts 1 and 2 denote the identity of the observing station. For the two station case, measurement noise covariance is assumed to be in the block diagonal form. Data generation starts from a random initial condition $X_0 \sim \mathcal{N}(X_0^{true}, P_0)$ where X_0^{true} is the true initial state (belonging to the UTC

time of 18th November 2014 at 20 : 19 : 05) given as

$$X_0^{true} = \begin{bmatrix} 3852.37071681372 \\ 5415.98695792525 \\ 2433.89094306353 \\ -0.552232478547296 \\ -2.74067444416030 \\ 6.96295480567795 \end{bmatrix}. \quad (4.4.4)$$

Parameters used in the simulations in this section for different measurement types are summarized below.

- The covariance matrices of the initial states for the full state and angles based observation cases are given below respectively.

$$P_0 = \begin{bmatrix} (0.1)^2 I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & (0.001)^2 I_{3 \times 3} \end{bmatrix}, P_0 = \begin{bmatrix} (1)^2 I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & (0.05)^2 I_{3 \times 3} \end{bmatrix}. \quad (4.4.5)$$

- Full state case:

- Number of Monte Carlo runs, $M = 500$.
- Process noise covariance matrix:

$$Q = \begin{bmatrix} 10^{-10} I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 10^{-12} I_{3 \times 3} \end{bmatrix}. \quad (4.4.6)$$

- Measurement noise covariance matrix:

$$R = \begin{bmatrix} (0.01)^2 I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & (0.0002)^2 I_{3 \times 3} \end{bmatrix}. \quad (4.4.7)$$

- AER case:

- Number of Monte Carlo runs, $M = 150$.

- Process noise covariance matrix:

$$Q = \begin{bmatrix} 10^{-8} I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 4 \times 10^{-10} I_{3 \times 3} \end{bmatrix}. \quad (4.4.8)$$

- Measurement noise covariance matrix:

$$R = \begin{bmatrix} (0.01)^2 I_{2 \times 2} & 0_{2 \times 1} \\ 0_{1 \times 2} & (0.05)^2 \end{bmatrix}. \quad (4.4.9)$$

- AE case:

- Number of Monte Carlo runs, $M = 100$.

- Process noise covariance matrix:

$$Q = \begin{bmatrix} 10^{-8} I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 4 \times 10^{-10} I_{3 \times 3} \end{bmatrix}. \quad (4.4.10)$$

- Measurement noise covariance matrix:

$$R = (0.01)^2 I_{2 \times 2}. \quad (4.4.11)$$

- AED case:

- Number of Monte Carlo runs, $M = 100$.

- Process noise covariance matrix:

$$Q = \begin{bmatrix} 10^{-8} I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 4 \times 10^{-10} I_{3 \times 3} \end{bmatrix}. \quad (4.4.12)$$

- Measurement noise covariance:

$$R = \begin{bmatrix} (0.01)^2 I_{2 \times 2} & 0_{2 \times 1} \\ 0_{1 \times 2} & 25 \end{bmatrix}. \quad (4.4.13)$$

- For the two observing station cases, measurement noise covariance can be constructed as follows assuming the noises are uncorrelated and the noise characteristics of the stations are identical.

$$R_{two} = \begin{bmatrix} R & 0_{n \times n} \\ 0_{n \times n} & R \end{bmatrix} \quad (4.4.14)$$

where n is the dimension of the measurement vector.

In the following subsection, the resulting RMSE's of the CD-EKF with 0.5 seconds of prediction step size is compared with the PCRLB calculated using the parameters presented above. Note that, in CD-EKF, Q matrices given above should be scaled with 0.5 since they are valid when the prediction step size is 1 second.

4.4.1 Absolute Filter Performances

In Figure 4.4.1 the absolute performance of the proposed estimator for the full state measurements case can be seen. As observed in the figure, in terms of position errors, CD-EKF is almost efficient (i.e., its RMSE reaches PCRLB). On the other hand, in terms of velocity estimation, there is still a gap for improvement since CD-EKF cannot reach PCRLB.

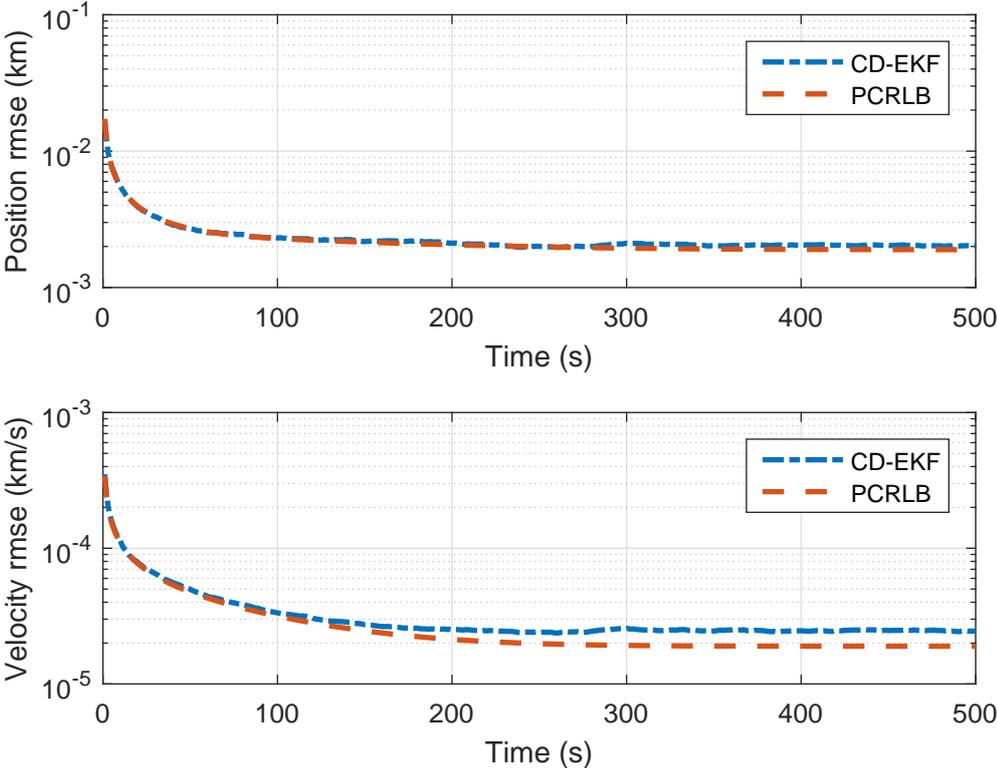


Figure 4.4.1: Full state case: Comparison of the proposed estimator with PCRLB.

Among different measurement types, the second most informative measurement

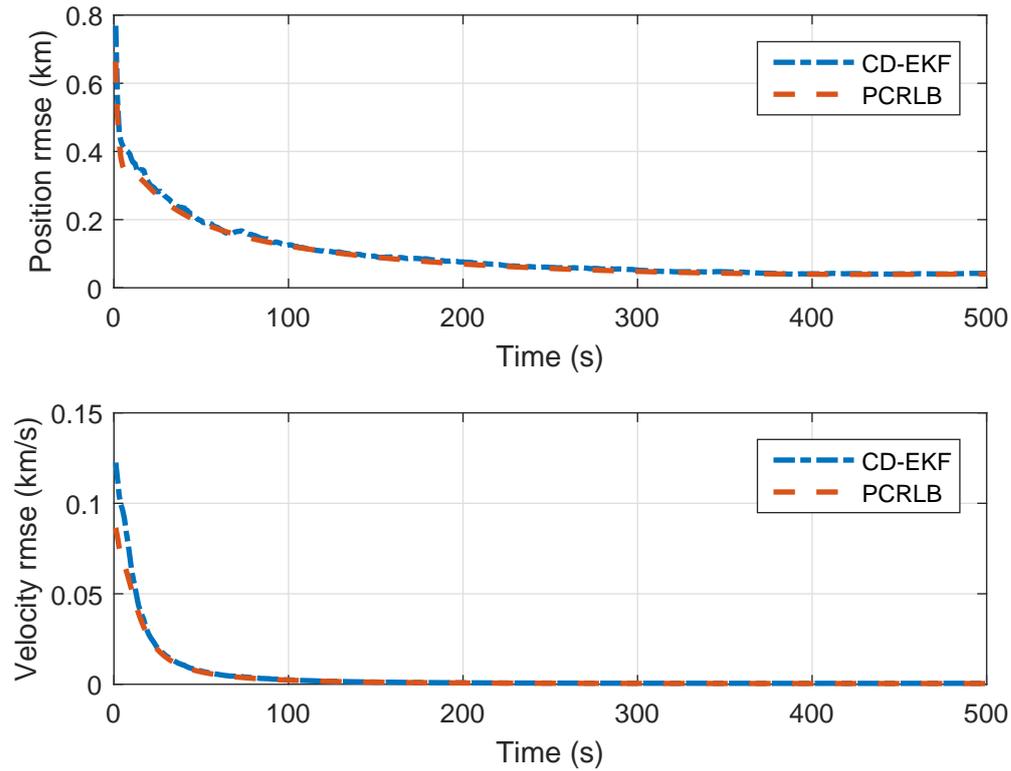


Figure 4.4.2: AER case: Comparison of the proposed estimator with PCRLB.

type is AER. This is because, AER completely defines the position of the satellite and velocity information can be extracted by using successive AER measurements. The absolute performance of the estimator with AER measurements for single station and two station cases is shown in Figures 4.4.2 and 4.4.3 respectively. As observed in the figures, CD-EKF reaches PCRLB asymptotically for both position and velocity estimation. Hence we can say that CD-EKF is asymptotically efficient for the AER case. It is seen that the availability of the second station provides approximately 20% reduction in the position RMS errors. However, the reduction in the velocity errors is much smaller. In our simulations we have seen that this small reduction in the velocity errors might be the result of the high sampling rate used in our study. In addition, we have noticed that the errors are highly sensitive to the number of Monte Carlo runs. Therefore, the reason for the small reduction in the velocity case might be due to the relatively small number of Monte Carlo runs we used in this case.

When the angles only data data (i.e., azimuth and elevation) is available, the

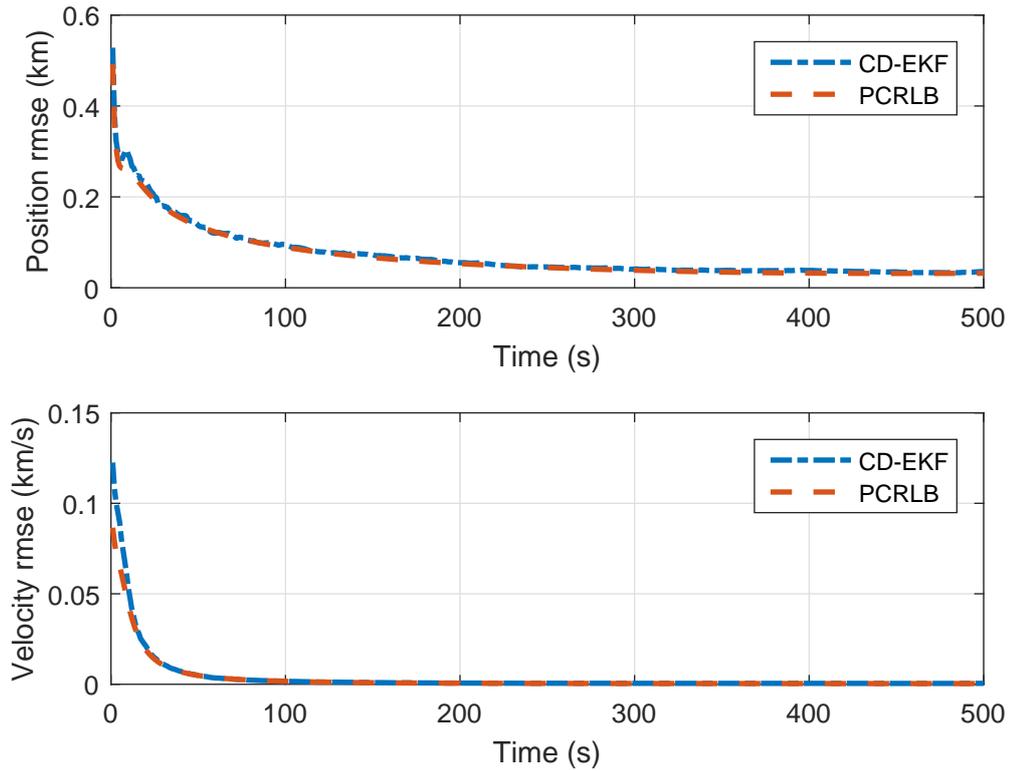


Figure 4.4.3: AER with two observing station: Comparison of the proposed estimator with PCRLB.

only information is the direction of satellite with respect to observing station. For this case, RMSE performance is expected to be worse than the AER case (see Figure 4.4.4). The performance of the estimator for this case can be seen in Figure 4.4.5. In the AE case, we see that similar results to AER case are obtained in terms of PCRLB comparison. On the other hand, in this case, we see that the improvement obtained by using a second observing station is significant for both position and velocity estimation. This result is expected because, when a second observing station is used then the range information can be found by using the intersection of line of sight vectors of the two stations.

In the absence of range data, Doppler shift may be added to angles only measurements and it can improve the estimator performance. This extra information also lowers PCRLB. Results are shown in Figures 4.4.6 and 4.4.7 for a single station and two station cases.

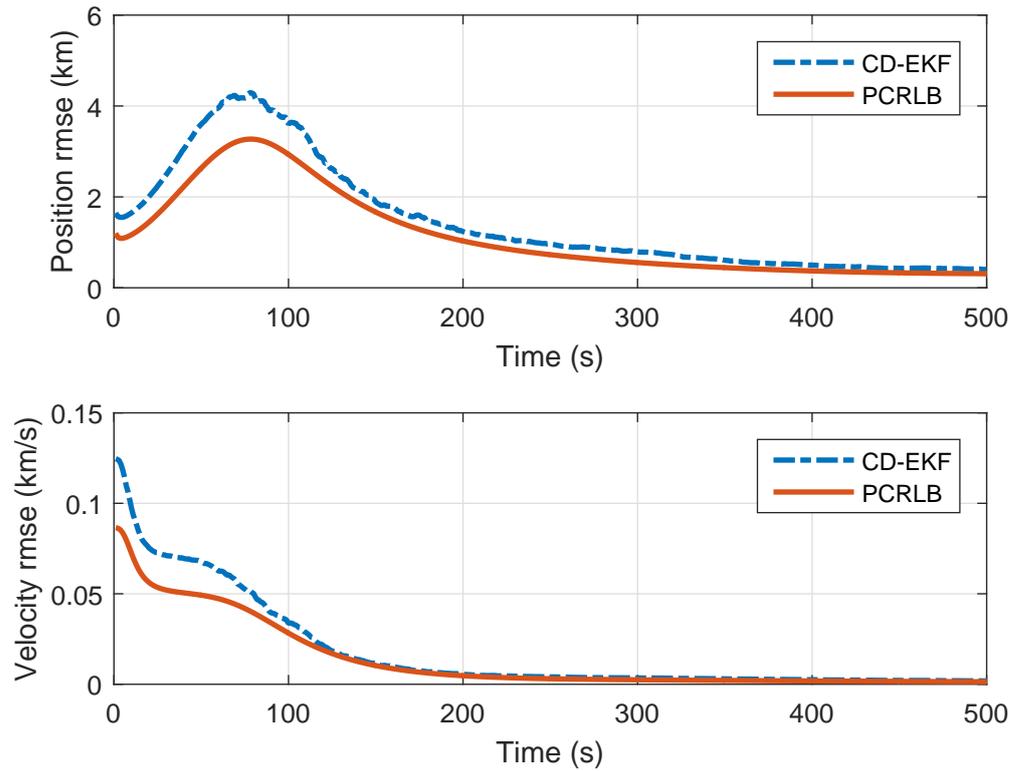


Figure 4.4.4: AE case: Comparison of the proposed estimator with PCRLB.

In order to clearly see how much improvement Doppler information provides us, results obtained using one and two station AE cases and single station AED case are compared in Figure 4.4.8. It can be seen that Doppler measurements can reduce the estimation errors of the AE case by a factor of approximately 4 and can obtain almost half the estimation errors of two station AE case for both position and velocity.

In order to see the summary of the relative performances for different measurement types with respect to each other and with respect to PCRLB, one can refer to Table 4.4.1.

The RMS estimation errors of the EKF and the UKF are compared with the D-PCRLB (posterior Cramer-Rao lower bound calculated by using the discrete formulation) for the AE observation case. Simulations are carried out by using the parameters given above in this section. According to the Figure 4.4.9, performances of the EKF and the UKF seem to be almost efficient asymptotically

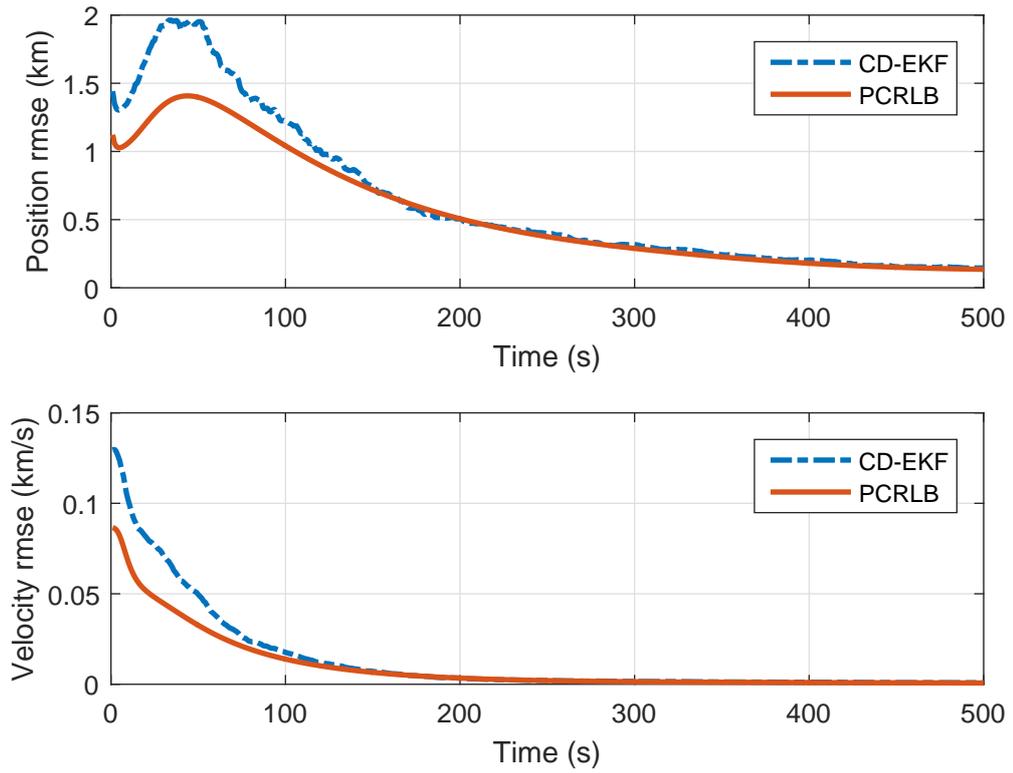


Figure 4.4.5: AE with two observing station: Comparison of the proposed estimator with PCRLB.

as well. It should be noted that since the D-PCRLB corresponds to the estimation problem where true data is generated with the discrete model, comparison of the D-PCRLB and PCRLB (which is calculated for the continuous time case) is not meaningful scientifically.

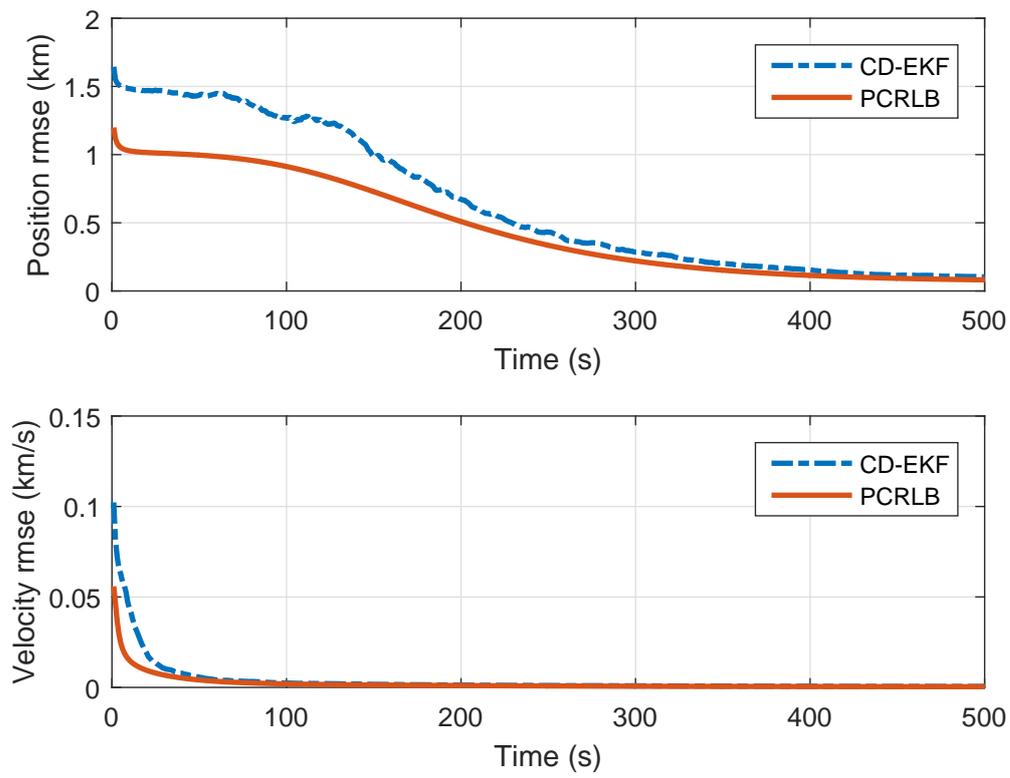


Figure 4.4.6: AED case: Comparison of the proposed estimator with PCRLB.

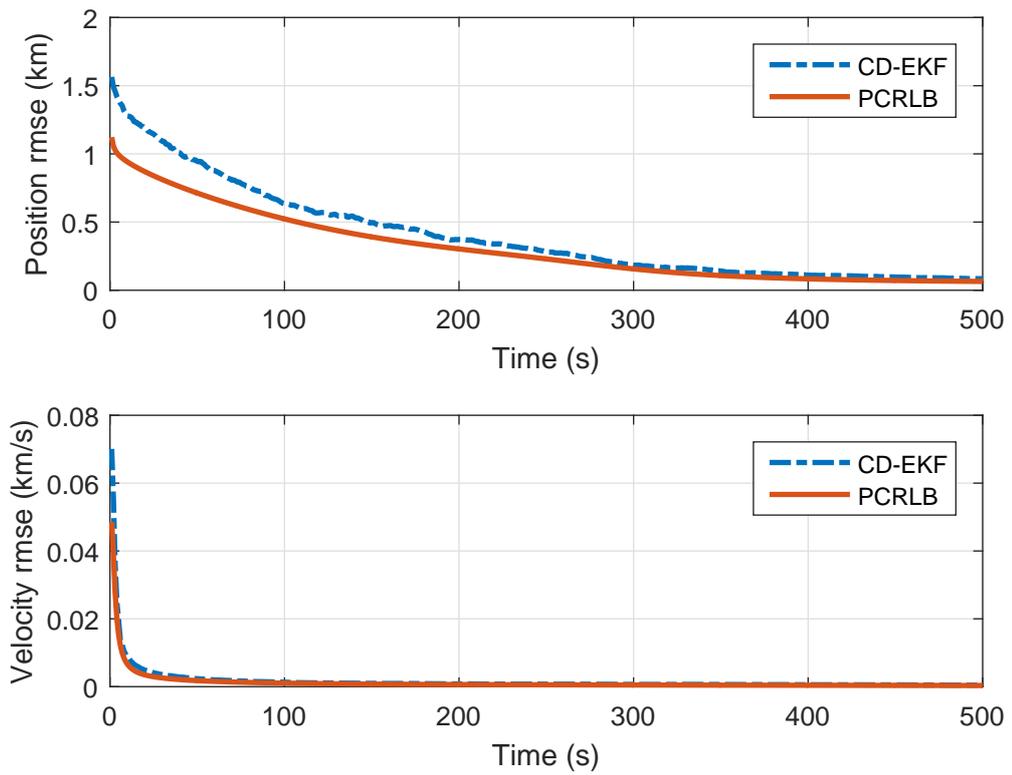


Figure 4.4.7: AED with two observing stations: Comparison of the proposed estimator with PCRLB.

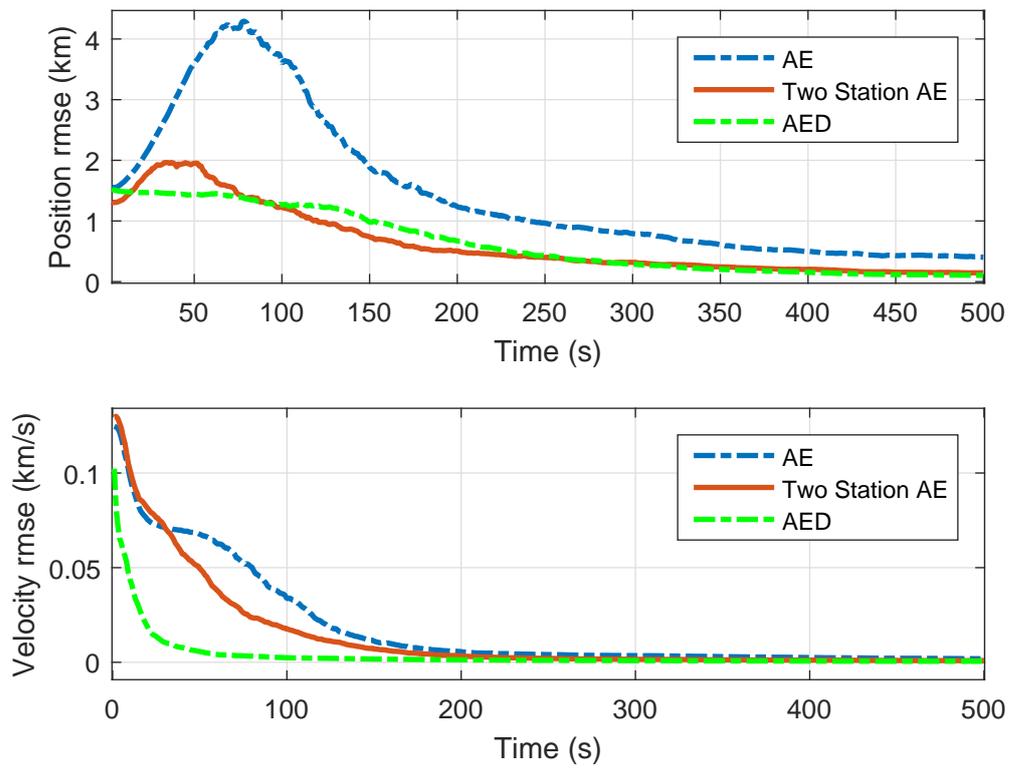


Figure 4.4.8: RMSE performance comparison of the AE and AED measurement cases.

Table 4.4.1: Time averaged RMSE and PCRLB comparison for the last 20 samples.

Measurement	RMSE		PCRLB	
	pos. (m)	vel. (cm/s)	pos. (m)	vel. (cm/s)
AE	425.2	195.9	320.7	145.2
AE (two station)	151.1	86.9	141.2	73.6
AED	108.8	54.4	85.2	42.4
AED (two station)	88.6	50.2	67.2	38.1
AER	42.9	52.1	41.3	40.7
AER (two station)	35.2	49.3	32.7	37.4
Full state	2.1	2.5	1.9	1.9

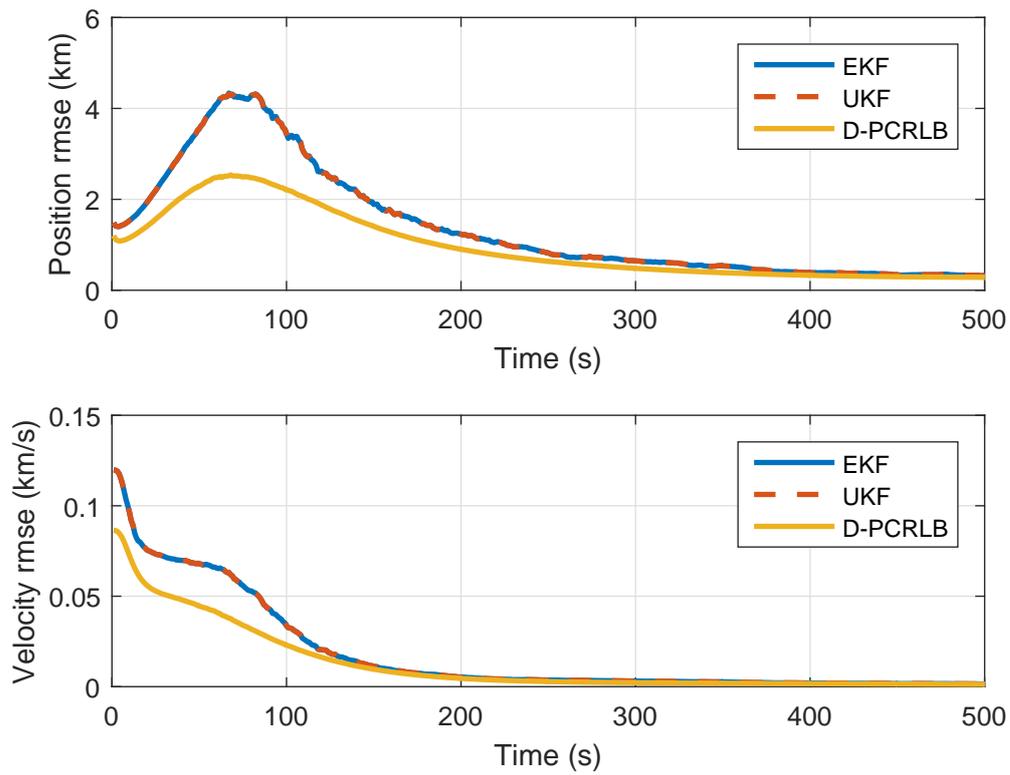


Figure 4.4.9: RMSE performances of the EKF and the UKF with respect to the discrete PCRLB.

CHAPTER 5

CONCLUSION AND FUTURE WORK

In this thesis, satellite orbit determination with statistical methods is investigated using a simplified version of the orbit dynamic model and various combinations of measurements. For this purpose, different types of Kalman filters are utilized and their performances are studied. Gauss' method (an example of a deterministic method) is also presented in order to initialize Kalman filters when satellite position can not be measured directly. Performance of Gauss' method is examined in terms of position and velocity accuracies with respect to angular measurement errors and time separation between measurements. It is shown that with a tracking equipment (should not be very precise) initial state can be determined with a position uncertainty in the order of tens of kilometers and with a velocity uncertainty of less than a kilometer per second.

Performances of EKF, UKF and CD-EKF are examined in terms of their RMSE's and computational load. It has been shown that CD-EKF (with a reasonable choice of prediction step size) gives better error characteristics with a relatively low computational load. It is also shown that if the initial position uncertainty is much greater than 20 km then UKF gives better estimates.

The absolute performance of the CD-EKF was investigated by comparing its RMSE with PCRLB. Since the estimator is not completely in a discrete form, a modified version of the discrete PCRLB was also mentioned. Except velocity estimate of the full state observation case, it has seen that, CD-EKF gives efficient results. In other words, after a few hundred (depending on the observation

type) measurement updates, RMSE of the filter asymptotically reaches PCRLB.

It has been shown that AED measurements can reduce the estimation errors of the AE case by a factor of 3 and can obtain half the estimation errors of two station AE case for both position and velocity. It has also been seen that, for AER and AED measurement cases, using a second observing station does not provide a considerable gain especially for velocity estimates.

The use of 500 seconds of observation duration (which was used in our PCRLB simulations) was a reasonable choice for a LEO satellite tracking system utilizing radio waves. However, if optical tracking systems are used then it is very difficult (if not impossible) to achieve such a long measurement duration. This is because, optical visibility depends on the geometry and orientation of the satellite with respect to the observer and the sun. Weather conditions also affect the satellite visibility for optical systems. Therefore, our results on this sense apply only to LEO satellite tracking systems utilizing radio waves.

For future studies, especially for precise orbit determination purposes, more realistic orbit dynamic models (accurate atmospheric model, higher order gravity model etc.) can be used. Accurate timing and coordinate conversion can also be used for this purpose. Time varying parameters (drag coefficient, radiation pressure coefficient etc.) which appear in either the dynamic model or the measurement model can be estimated adaptively by adding these parameters to the state vector, which is known as state augmentation. Moreover, for the multi station observation case, time tagging errors and fusion of measurements coming from these observing sites are also possible future subjects of research.

REFERENCES

- [1] D. A. Vallado, *Fundamentals of Astrodynamics and Applications*. Springer Science & Business Media, 2001, vol. 12.
- [2] R. R. Bate, D. D. Mueller, and J. E. White, *Fundamentals of Astrodynamics*. Courier Corporation, 1971.
- [3] O. Montenbruck and E. Gill, *Satellite Orbits: Models, Methods and Applications*. Springer Science & Business Media, 2000.
- [4] NASA, GSFC, “The impact of flares,” (accessed:17-July-2016). [Online]. Available: <http://hesperia.gsfc.nasa.gov/rhessi3/mission/science/the-impact-of-flares/index.html>
- [5] M. Capderou, *Satellites: Orbits and missions*. Springer Science & Business Media, 2006.
- [6] Wikipedia, “List of orbits,” (accessed:17-July-2016). [Online]. Available: https://en.wikipedia.org/wiki/List_of_orbits
- [7] A. K. Maini and V. Agrawal, *Satellite technology: Principles and applications*. John Wiley & Sons, 2011.
- [8] Air University, “Au-18 space primer,” (accessed:17-July-2016). [Online]. Available: http://space.au.af.mil/au-18-2009/au-18_chap06.pdf
- [9] Wikipedia, “Orbit determination,” (accessed:17-July-2016). [Online]. Available: https://en.wikipedia.org/wiki/Orbit_determination
- [10] S. M. Stigler, “Gauss and the invention of least squares,” *The Annals of Statistics*, pp. 465–474, 1981.
- [11] J. R. Vetter, “Pfifty years of orbit determination,” *Johns Hopkins APL technical digest*, vol. 27, no. 3, p. 239, 2007.
- [12] B. Schutz, B. Tapley, and G. H. Born, *Statistical orbit determination*. Academic Press, 2004.
- [13] J. Wright *et al.*, “Optimal orbit determination,” 2002.
- [14] B. O. Teixeira, M. A. Santillo, R. S. Erwin, and D. S. Bernstein, “Spacecraft tracking using sampled-data Kalman filters,” *IEEE Control Systems*, vol. 28, no. 4, pp. 78–94, 2008.

- [15] M. E. Hough, "Precise orbit determination using satellite radar ranging," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 4, pp. 1048–1058, 2012.
- [16] I. I. Hussein, C. W. Roscoe, M. P. Wilkins, and P. W. Schumacher Jr, "Probabilistic admissibility in angles-only initial orbit determination," in *Proceedings of the 24th International Symposium on Space Flight Dynamics*, 2014, pp. 5–9.
- [17] D.-J. Lee and K. T. Alfriend, "Sigma point filtering for sequential orbit estimation and prediction," *Journal of Spacecraft and Rockets*, vol. 44, no. 2, pp. 388–398, 2007.
- [18] H. Chen, G. Chen, E. Blasch, and K. Pham, "Comparison of several space target tracking filters," in *Proc. SPIE*, vol. 7330, 2009, p. 73300I.
- [19] J. L. Crassidis and J. L. Junkins, *Optimal estimation of dynamic systems*. CRC press, 2011.
- [20] M. Capderou, *Handbook of Satellite Orbits: From Kepler to GPS*. Springer International Publishing, 2014.
- [21] E. M. Gaposchkin and A. J. Coster, "Analysis of satellite drag," Massachusetts Inst. of Tech., Lexington (USA). Lincoln Lab., Tech. Rep., 1988.
- [22] Purdue University, "Gravitational potential," (accessed:26-July-2016). [Online]. Available: <http://web.ics.purdue.edu/~ecalais/teaching/eas450/Gravity2.pdf>
- [23] W. Torge and J. Müller, *Geodesy*. Walter de Gruyter, 2012.
- [24] G. Seeber, *Satellite geodesy: Foundations, methods, and applications*. Walter de Gruyter, 2003.
- [25] Wikipedia, "Legendre polynomials," (accessed:27-July-2016). [Online]. Available: https://en.wikipedia.org/wiki/Legendre_polynomials
- [26] M. Solarić, *Satellite Motion*. Sciyo, 2010.
- [27] ESA, "Space in images," (accessed:25-July-2016). [Online]. Available: http://www.esa.int/spaceinimages/Images/2004/10/The_Earth_s_gravity_field_geoid_as_it_will_be_seen_by_GOCE
- [28] T. D. Papanikolaou and D. Tsoulis, "Assessment of numerical integration methods in the context of low earth orbits and inter-satellite observation analysis," *Acta Geodaetica et Geophysica*, pp. 1–23, 2016.
- [29] B. Somodi and L. Földváry, "Application of numerical integration techniques for orbit determination of state-of-the-art LEO satellites," *Periodica Polytechnica. Civil Engineering*, vol. 55, no. 2, p. 99, 2011.

- [30] E. Fehlberg, “New one-step integration methods of high-order accuracy applied to some problems in celestial mechanics,” 1966.
- [31] Federal Aviation Administration, “Describing orbits,” (accessed:19-February-2017). [Online]. Available: https://www.faa.gov/other_visit/aviation_industry/designees_delegations/designee_types/ame/media/Section%20III.4.1.4%20Describing%20Orbits.pdf
- [32] H. Curtis, *Orbital Mechanics for Engineering Students*. Butterworth-Heinemann, 2013.
- [33] I. Ali, P. G. Bonanni, N. Al-Dhahir, and J. E. Hershey, *Doppler applications in LEO satellite communication systems*. Springer Science & Business Media, 2006, vol. 656.
- [34] R. H. Battin, *An introduction to the mathematics and methods of astrodynamics*. AIAA, 1999.
- [35] NASA, “Definition of two-line element set coordinate system,” (accessed:22-February-2017). [Online]. Available: https://spaceflight.nasa.gov/realdata/sightings/SSapplications/Post/JavaSSOP/SSOP_Help/tle_def.html
- [36] T. S. Kelso, “Frequently asked questions: Two-line element set format,” (accessed:20-February-2017). [Online]. Available: <https://www.celestrak.com/columns/v04n03/>
- [37] F. R. Hoots and R. L. Roehrich, “Models for propagation of NORAD element sets,” DTIC Document, Tech. Rep., 1980.
- [38] M. Skolnik, *Radar Handbook, Third Edition*. McGraw-Hill Education, 2008.
- [39] R. Johnson and H. Jasik, *Antenna Engineering Handbook*. McGraw-Hill, 1993.
- [40] C. Jekeli, “Geometric reference systems in geodesy,” *Report, Division of Geodesy and Geospatial Science, School of Earth Sciences, Ohio State University*, 2006.
- [41] J. P. Vinti, *Orbital and celestial mechanics*. AIAA, 1998, vol. 177.
- [42] Wikipedia, “Oblate spheroidal coordinates,” (accessed:2-July-2016). [Online]. Available: https://en.wikipedia.org/wiki/Oblate_spheroidal_coordinates
- [43] F. L. Markley and J. L. Crassidis, *Fundamentals of spacecraft attitude determination and control*. Springer, 2014, vol. 33.
- [44] P. C. Hughes, *Spacecraft attitude dynamics*. Courier Corporation, 2012.

- [45] NASA, “Space geodesy project,” (accessed:2-August-2016). [Online]. Available: <http://space-geodesy.nasa.gov/multimedia/EarthOrientationAnimations/EOAnimations.html>
- [46] United States Naval Observatory, “Approximate sidereal time,” (accessed:7-August-2016). [Online]. Available: <http://aa.usno.navy.mil/faq/docs/GAST.php>
- [47] —, “Universal time,” (accessed:7-August-2016). [Online]. Available: <http://aa.usno.navy.mil/faq/docs/UT.php>
- [48] J. H. Meeus, *Astronomical algorithms*. Willmann-Bell, Incorporated, 1991.
- [49] S. P. Drake, “Converting GPS coordinates [phi, lambda, h] to navigation coordinates (ENU),” 2002.
- [50] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: Theory algorithms and software*. John Wiley & Sons, 2004.
- [51] S. M. Kay, *Fundamentals of Statistical Signal Processing, volume I: Estimation theory*. Prentice Hall, 1993.
- [52] S. S. Haykin *et al.*, *Kalman filtering and neural networks*. Wiley Online Library, 2001.
- [53] D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.
- [54] M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice with MATLAB*, 4th ed. Wiley-IEEE Press, 2014.
- [55] G. J. Bierman, *Factorization methods for discrete sequential estimation*. Courier Corporation, 2006.
- [56] B. D. Anderson and J. B. Moore, *Optimal filtering*. Courier Corporation, 2012.
- [57] S. C. Chapra and R. P. Canale, *Numerical methods for engineers*. McGraw-Hill New York, 2012, vol. 2.
- [58] Wikipedia, “Numerical differentiation,” (accessed:14-August-2016). [Online]. Available: https://en.wikipedia.org/wiki/Numerical_differentiation
- [59] University of Oslo, “Numerical differentiation,” (accessed:14-August-2016). [Online]. Available: <http://www.uio.no/studier/emner/matnat/math/MAT-INF1100/h08/kompendiet/diffint.pdf>
- [60] A. H. Jazwinski, *Stochastic processes and filtering theory*, ser. Mathematics in science and engineering. Academic press, 1970.

- [61] P. Frogerais, J. J. Bellanger, and L. Senhadji, “Various ways to compute the continuous-discrete extended Kalman filter,” *IEEE Transactions on Automatic Control*, vol. 57, no. 4, pp. 1000–1004, 2012.
- [62] T. Mazzoni, “Computational aspects of continuous–discrete extended Kalman-filtering,” *Computational Statistics*, vol. 23, no. 4, pp. 519–539, 2008.
- [63] G. Y. Kulikov and M. V. Kulikova, “Accurate numerical implementation of the continuous-discrete extended Kalman filter,” *IEEE Transactions on Automatic Control*, vol. 59, no. 1, pp. 273–279, 2014.
- [64] S. Särkkä, *Bayesian filtering and smoothing*. Cambridge University Press, 2013, vol. 3.
- [65] E. A. Wan and R. Van Der Merwe, “The unscented Kalman filter for nonlinear estimation,” in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*. Ieee, 2000, pp. 153–158.
- [66] S. J. Julier and J. K. Uhlmann, “New extension of the Kalman filter to nonlinear systems,” in *AeroSense’97*. International Society for Optics and Photonics, 1997, pp. 182–193.
- [67] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, “A new approach for filtering nonlinear systems,” in *American Control Conference, Proceedings of the 1995*, vol. 3. IEEE, 1995, pp. 1628–1632.
- [68] G. H. Golub and C. F. Van Loan, *Matrix computations*. Johns Hopkins University Press, 1996.
- [69] J. Hartikainen, A. Solin, and S. Särkkä, “Optimal filtering with Kalman filters and smoothers,” *Department of Biomedical Engineering and Computational Sciences, Aalto University School of Science: Greater Helsinki, Finland*, vol. 16, 2011.
- [70] H. L. V. Trees and K. L. Bell, *Bayesian Bounds for Parameter Estimation and Nonlinear Filtering/Tracking*. Wiley-IEEE Press, 2007.
- [71] H. L. Van Trees, *Detection, estimation, and modulation theory*. John Wiley & Sons, 2004.
- [72] P. Tichavsky, C. H. Muravchik, and A. Nehorai, “Posterior Cramér-Rao bounds for discrete-time nonlinear filtering,” *IEEE Transactions on signal processing*, vol. 46, no. 5, pp. 1386–1396, 1998.
- [73] K. L. Bell and H. L. Van Trees, “Posterior Cramer-Rao bound for tracking target bearing,” in *13th Annual Workshop on Adaptive Sensor Array Process*, 2005.

- [74] N. Bergman, “Posterior Cramér-Rao bounds for sequential estimation,” in *Sequential Monte Carlo methods in practice*. Springer, 2001, pp. 321–338.
- [75] A. Milani and G. Gronchi, *Theory of orbit determination*. Cambridge University Press, 2010.
- [76] Wikipedia, “Gauss’ method,” (accessed:22-August-2016). [Online]. Available: https://en.wikipedia.org/wiki/Gauss'_Method