INVESTIGATION OF THE STAR TRACKER ALGORITHMS AND KALMAN FILTER INTEGRATION

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

BY

SELVA SARGIN GÜÇLÜ

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

FEBRUARY 2019

Approval of the thesis:

INVESTIGATION OF THE STAR TRACKER ALGORITHMS AND KALMAN FILTER INTEGRATION

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

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. İsmail Hakkı Tuncer
Head of Department, **Aerospace Engineering** _____

Assoc. Prof. Dr. İlkay Yavrucuk
Supervisor, **Aerospace Engineering, METU** _____

Examining Committee Members:

Prof. Dr. Ozan Tekinalp
Aerospace Engineering, METU _____

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

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

Prof. Dr. Coşku Kasnakoğlu
Electrical and Electronics Engineering, TOBB ETU _____

Assist. Prof. Dr. Ali Türker Kutay
Aerospace Engineering, METU _____

Date:

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

Name, Surname: Selva Sargın Güçlü

Signature :

ABSTRACT

INVESTIGATION OF THE STAR TRACKER ALGORITHMS AND KALMAN FILTER INTEGRATION

Sargın Güçlü, Selva
M.S., Department of Aerospace Engineering
Supervisor: Assoc. Prof. Dr. İlkay Yavrucuk

February 2019, 79 pages

This research outlines the investigation of star tracker algorithms and Kalman Filter implementation of star tracker aided INS. In the first chapter, the usage areas of the star tracker sensor was investigated. The purposes to use this sensor were also included. Furthermore, the sensor architecture and algorithms were examined. In the second chapter, the three main steps of the star tracker algorithms were investigated. First of all, literature survey about centroiding algorithms was done and two of them were examined in detail. Then, the literature about star matching algorithms was searched. Liebe's triangle method and Astrometry.net's quad method were investigated in detail. Finally, the research about the attitude determination algorithms was made. The studies made about three of them which are SVD, Triad and Quest methods were shown. In the last chapter, the results of an Extended Kalman Filter implementation of star tracker aided INS were demonstrated. For this implementation, a high altitude aircraft was chosen because star tracker sensor gives more accurate results in high altitudes. To simulate a high altitude aircraft, X-15 aircraft simulation was used in the Airlib library of Simulink. Assumptions were made about the accu-

racies of the star tracker sensor and INS according to the sensors that can be used in the high altitude aircrafts. Moreover, the simulation architecture of the Kalman Filter implementation, which makes attitude correction, was investigated.

Keywords: Star Tracker, Star Centroiding Algorithms, Star Matching Algorithms, Attitude Determination Algorithms, Kalman Filter, INS

ÖZ

YILDIZ TAKİP ALGORİTMALARININ İNCELENMESİ VE KALMAN FİLTRE ENTEGRASYONU

Sargın Güçlü, Selva
Yüksek Lisans, Havacılık ve Uzay Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. İlkay Yavrucuk

Şubat 2019 , 79 sayfa

Bu araştırma, yıldız takip sensörü algoritmalarını ve yıldız takip sensörü destekli ANS'nin Kalman Filtre uygulamasının incelenmesini özetlemektedir. İlk bölümde, yıldız takip sensörünün kullanım alanları ve kullanmanın amaçlarından bahsedilmiştir. Ayrıca, sensör mimarisi ve algoritmalar incelenmiştir. İkinci bölümde, yıldız takip algoritmalarının üç ana adımı araştırılmıştır. İlk olarak, merkezleme algoritmaları ile ilgili literatür taraması yapılmış ve bunlardan ikisi ayrıntılı olarak incelenmiştir. Daha sonra yıldız eşleştirme algoritmaları ile ilgili literatür araştırılmıştır. Liebe'nin üçgen metodu ve Astrometry.net'in dörtgen yöntemi detaylı olarak araştırılmıştır. Son olarak, yönelim belirleme algoritmaları ile ilgili araştırma yapılmıştır. Bunlardan SVD, Triad ve Quest yöntemleri ile ilgili detaylı çalışmalar yapılmıştır. Son bölümde ise, yıldız takip sensörü destekli ANS'nin Genişletilmiş Kalman Filtresi uygulamasının sonuçları gösterilmiştir. Bu uygulama için, yüksek irtifa uçağı seçilmiştir çünkü yıldız takip sensörü yüksek irtifalarda daha doğru sonuçlar vermektedir. Yüksek irtifa uçağını simüle etmek için Simulink Airlib kütüphanesinde yer alan X-15 uçak simülasyonu kullanılmıştır. Yüksek irtifa uçaklarında kullanılabilecek sensörlere göre

yıldız takip sensörünün ve ANS'nin doğruluğu hakkında varsayımlar yapılmıştır. Ayrıca, yönelim düzeltmesi yapan Kalman Filtre uygulamasının simülasyon mimarisi incelenmiştir.

Anahtar Kelimeler: Yıldız Takip Sensörü , Yıldız Merkezi Belirleme Algoritmaları, Yıldız Eşleştirme Algoritmaları, Yönelim Belirleme Algoritmaları, Kalman Filtresi, ANS

To my lovely husband Anıl, my baby on the way, my mom, dad and sister for their endless support...

ACKNOWLEDGMENTS

First of all, I would like to thank my advisor, Assoc. Prof. Dr. İlkay Yavrucuk for his guidance and the freedom to explore my interests. He always encouraged me. I would also like to thank my colleagues at ROKETSAN for their help in navigation studies.

I am grateful to my parents, Zuhal and Aydoğan Sargın, for always supporting me and for their endless love. Moreover, I thank my sister Esra Sargın for her cheering and for making me feel that she is behind me.

I would also like to thank my mother-in-law, father-in-law and brother-in-law, Ümit, Yılmaz and Egemen Güçlü, for their support and motivation.

I thankful to my valuable friends, Büşra Timur, Merve Acar Temel, Ayça Bayram for their long-lasting friendships and Tuğba Tunçel for being always open her office's door to me. I would also like to thank my colleagues, Hanife Usta, Ece Alaçakır and Aslı Doğan for their endless cheerfulness and their delicious teas and coffees.

My special thanks to my lovely and cheerful husband Anıl Güçlü for his endless love, support and motivation. I would like to dedicate my thesis to my present and future family.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

(x_c, y_c)	Centroid location
$(C_b^n)_{INS}$	from body frame to navigation frame DCM which is calculated from inertial navigation equations
$(C_b^n)_{measurement}$	from body frame to navigation frame DCM which is calculated from star tracker measurements
$\overrightarrow{V_1}$	Inertial frame vectors
$\overrightarrow{W_1}$	Body frame vectors
h_b	Height of the body frame in the navigation frame
(C_b^n)	Coordinate transformation matrix from body to navigation frame
L_b	Latitude of the body frame in the navigation frame
V_{eb}^n	Velocity vector of body frame with respect to Earth frame in the navigation frame
a_i	Weight set
b_i	Observed stars unit vectors in the camera frame
f_{ib}^n	Specific force of body frame with respect to inertial frame in the navigation frame
$\overrightarrow{q}$	Eigenvector
r_i	Observed stars unit vectors in the inertial frame
α_{ROI}	Size of the region of the interest in pixels
α_k	Weights
λ_b	Longitude of the body frame in the navigation frame
σ_{radius}	Truncation radius of the Gaussian kernel
A	Direction Cosine Matrix
ADU	Analog to Digital Converter

AIM	Attitude Estimates Using Image Matching
APS	Active Pixel Sensor
B	Total brightness in the ROI
CCD	Charge Coupled Device
CMOS	Complementary Metal Oxide Semiconductor
DCM	Direction Cosine Matrix
DEC	Declination
EKF	Extended Kalman Filter
ESOQ	Optimal Quaternion Estimator
f	Focal length
FITS	Flexible Image Transfer System
FOV	Field of View
FPS	Frame Per Second
FWHM	Full Width Half Maximum
g(A)	Gain function
GNSS	Global Navigation Satellite System
HWHM	Half Width Half Maximum
I(i,j)	Light flux in the current pixel
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
J(A)	Cost function
K	Kalman gain
L(A)	Loss (cost) function
LIS	Lost in Space
N	Measurement (star) numbers
P	Error covariance matrix
PSF	Point Spread Function

Q	System noise covariance matrix
QUEST	Quaternion Estimation
R	Measurement noise covariance matrix
RA	Right Ascension
ROI	Region of Interest
S	Singular values diagonal matrix
SVD	Singular Value Decomposition
U and V	Orthogonal matrices
α	Right ascension in J2000 inertial coordinate system
δ	Declination in J2000 inertial coordinate system
λ	Eigenvalue

CHAPTER 1

INTRODUCTION

Attitude determination is a crucial topic in the aerospace industry. Satellites communicate with the Earth have to orient the antennas towards the Earth. Spacecraft which has solar panels have to orient them towards the Sun. Moreover, thrust direction is important to change orbit by Hohmann transfer. Thus, accurate attitude determination for a space vehicle is valuable information. Star tracker is one of the attitude determination sensor. It can be used either alone or with the other attitude sensors in space vehicles. To estimate orientation (attitude), sun sensor, earth sensor, and magnetometer are also available. These sensors which are the parts of the Attitude and Orbit Control Systems [1] measure the orientation by using the direction of the Sun and Earth and the magnetic field of the Earth. Yet, these sensors provide lower attitude accuracy when compared to star trackers.

Nowadays, star trackers are mainly used in satellites because they provide accurate orientation information. Moreover, with the improving sensor technology, it is more compact, more accurate, autonomous and consumes less power. That's why, it is reasonable to use star tracker as the orientation sensor based on the studies to reduce the size of the satellites and make more capable satellites. Moreover, star trackers are affected by the atmospheric effects because stars should be observable to determine the orientation. Atmospheric gases, dust, light pollution and the difference of the day and night observations make less useful in atmospheric use. Despite there are studies for day observations for high altitude atmospheric conditions for scientific balloons, they are under development. Therefore, it can be said that space is good place to use star tracker. Yet, high altitude applications that have minimum atmospheric effect are also reasonable for star tracker observations.

Star tracker has been used for attitude determination for centuries. Previously, star sensors tracks one star and therefore they are called star tracker. Another application is to obtain differenet pictures for celestial sphere, this one is called star mapper [2]. However, now, all of the star sensors are called star trackers. In earlier applications, it has been used as a gimbaled sensor. Nowadays, this sensor is preferred to use as a strapdown system. Technologies used for these sensors have evolved over the years. For example, CCD (Charge Coupled Device) sensor technology was replaced with the CMOS APS (Complementary Metal Oxide Semiconductor Active Pixel Sensor) technology. On the other hand, algorithms used in star trackers were automatized during this time period, i.e. algorithms need not to be initialized. Therefore, Lost-in-space (LIS) mode was born.

Stars are landmarks for the star trackers. Space vehicles/satellites discover their orientation by observing the stars. The sensor matches the stars observed with the onboard database and makes star identification. Afterward, spacecraft orientation is found by using attitude determination algorithms. An example of a star tracker is shown in Figure 1.1 [3]. This navigation sensor does not need a priori attitude information. This mode is called as Lost in Space (LIS) mode for the star tracker. Tracking mode starts once orientation information is obtained. On the other hand, star trackers can be used as an INS aiding sensor. This implementation compensates attitude drift in INS applications [4]. GNSS is also one of the INS aiding sensors. Because GNSS is vulnerable to jamming and GNSS signals are not available in all locations, star tracker can be thought of as an alternative INS aiding navigation sensor [5]. However, INS/GNSS/Star tracker integration is one of the implementation method [6]. While GNSS updates INS position and velocity, star tracker updates INS attitude solution [7]. Yet, in this thesis, star tracker is the only INS aiding sensor. Furthermore, INS/Star tracker integration implementation is investigated for a high altitude aircraft. High altitude aircraft is chosen because atmospheric effects should be reduced as much as possible.

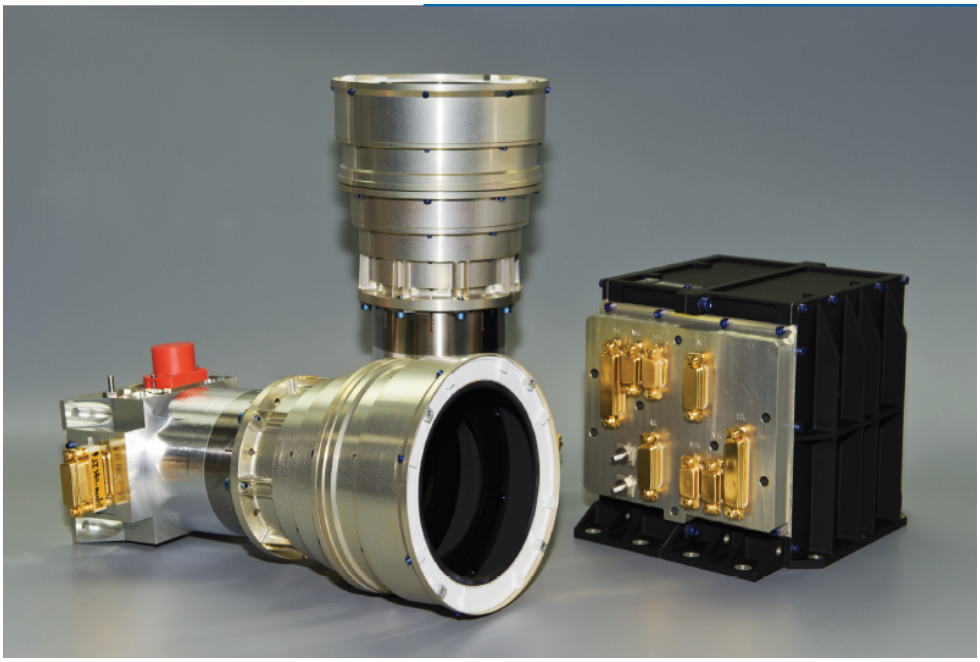


Figure 1.1: Sodern's Hydra Star Tracker

1.1 Sensor Architecture

Star trackers consist of two main parts. These are camera unit and data processing unit. Camera unit provides star images by blocking stray light coming from bright objects like the Sun, the Earth, the Moon and the other planets. To achieve this, star sensor has the structure called as a baffle. This structure is coated with a black material that has very low reflectivity. Some part of the unwanted stray light should be absorbed by the black material and the other part should be reflected many times to minimize stray light entrance. Therefore, baffle geometry is also an important parameter. It has more than one sharp edges to achieve this goal. Almost only the desired starlight comes the objective of star tracker [8]. Star tracker main parts are schematized in Figure 1.2 [9].

Objective is a lens array that takes rays from objects and reflects them to the focal plane array. Some parameters are critical for the optical design of this sensor like Field of View (FOV), F#, focal length, pixel size, and number. FOV determination is an optimization problem. Smaller FOV provides to enter less stray light but causes to be seen less star. In contrast, while wider FOV provides to be seen more star, it causes

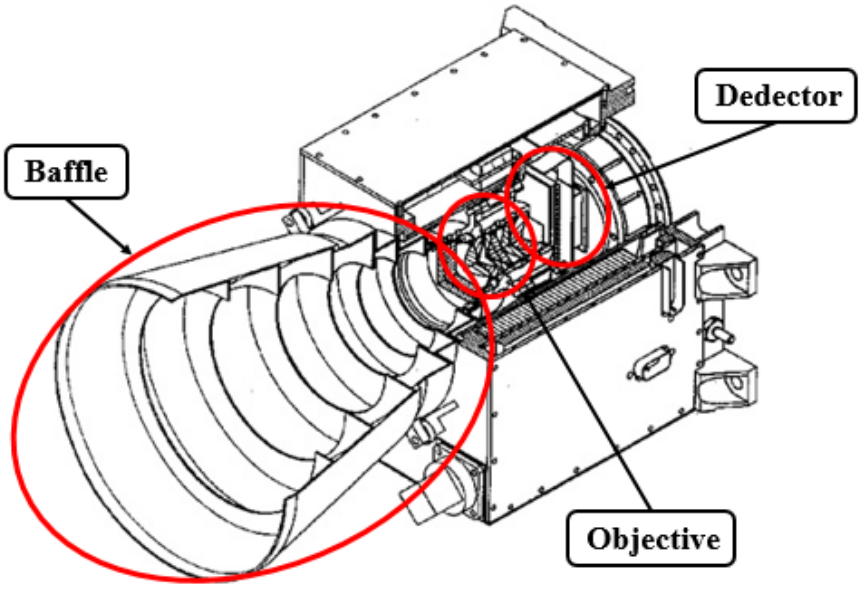


Figure 1.2: Sodern's Hydra Star Tracker

to enter the more stray light. To determine Field of View, star matching algorithm is also important because each algorithm requires different number of star in the FOV. Pixel size and number are also crucial design parameters. Smaller pixel size ensures resolution increase but very small pixel size causes crosstalk. Moreover, every pixel contains information. More information is gathered by having more pixels. Thus, centroiding calculation will be more accurate. Yet, lots of information requires a lot of processing power. Therefore, determination of the pixel size and number is also an optimization problem. Furthermore, star tracker detector topic is crucial because in space radiation hardened devices is deeply needed. CMOS sensors are more radiation tolerant. Additionally, CMOS sensors can have more intelligent designs than CCD sensors and offer substantially simpler system integration. Besides, they consume less power. Nowadays, with the low light CMOS sensor technology, dark current and noise performance are increased. For this reason, CMOS sensor technology is preferred over CCD sensor technology [10].

Data processing unit is composed of the software including the star database and algorithms. Star database forms according to star matching algorithm by using a star catalog. The star catalog is a list for stars that includes position in terms of right ascension and declination, magnitude, color, and position error etc. information.

Numerous catalogs have been developed for different purposes in many years. Many of the modern star catalogs are open source. The more modern catalogs have two aims. Either try to list each of the stars in the sky or try not to list all but to list the selected stars for specific purposes [11]. Hipparcos-Tycho catalog has widespread use in the field of the star tracker sensors. There are three main algorithms used in star tracker. The first algorithm makes centroiding of the stars observed by the camera. The second algorithm makes star identification using star matching algorithms and star database. Finally, the job of the last algorithm is to find the orientation of the space vehicle. These algorithms are summarized in Figure 1.3.

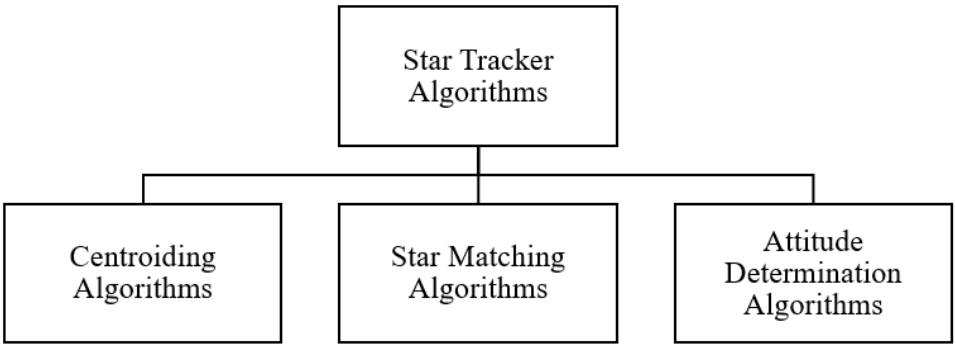


Figure 1.3: Star Tracker Algorithms

1.2 Organization of the Thesis

One of the aims of this thesis is to make literature survey about the algorithms frequently used in star tracker sensors. Moreover, the theory behind some of them will be explained further. The results of these algorithms will be compared in terms of speed, accuracy and robustness. The other goal is to make Extended Kalman Filter implementation of the star tracker aided INS (Inertial Navigation System).

In the second chapter, three types of algorithms in the literature which are mentioned in the Figure 1.3 are listed and explained. Then, some of them are explained much further and compared to each other.

In the third chapter, integrated navigation implementation of high altitude aircraft by using Extended Kalman filter is mentioned. Star tracker observations are reasonable

for space or sparse atmospheric conditions and there are ready-to-use high altitude aircraft 6-DOF simulations to create trajectory and IMU (Inertial Measurement Data) sensor data in Simulink. Therefore, in the second chapter, a high altitude aircraft Extended Kalman Filter implementation of star tracker aided INS topic is investigated.

CHAPTER 2

LITERATURE SURVEY, METHODOLOGY, STUDIES AND RESULTS

2.1 CENTROIDING ALGORITHMS STUDY

Star trackers play a key role in attitude determination of space vehicles. The first step in estimating highly accurate orientation information is to find the location of the stars in the image. In order to have subpixel centroiding accuracy, the optics of star tracker should be defocused. Thus, the light of the star is spread over some neighboring pixels [12]. In this step, there are mainly two different methods. The first one is the image moment analysis and the second one is the point spread function (PSF) fitting analysis.

2.1.1 Image Moment Analysis Method

In the simplest sense, measured starlight value averaging in pixels is made in the image moment analysis method. In this method, firstly, the size of the region of interest is determined. According to this ROI (region of interest), each of pixels in the ROI is investigated with respect to their ADU values. This value represents the number of photons detected in each pixel. After that, the threshold value is set according to the background noise level. This threshold generally equals to the mean of the background noise level plus its three sigma value. The star centroiding operation starts after selecting the pixels above the threshold value. This process is schematized in Figure 2.1 [13].

In image moment analysis method, first of all, background noise level is determined. Then, it is decided how many times the threshold value should be larger than the

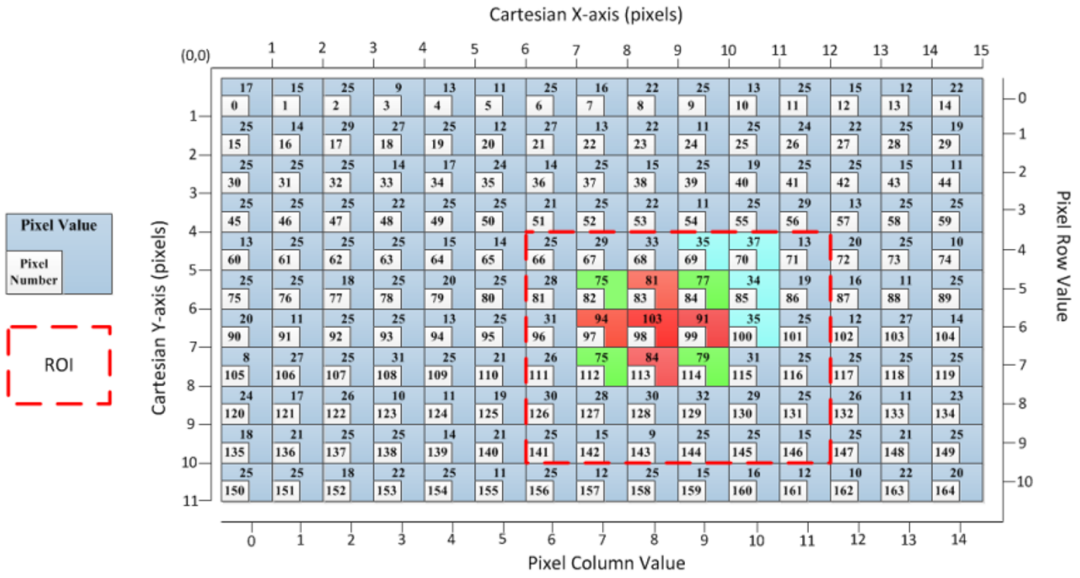


Figure 2.1: Image Moment Analysis Method

background noise level standard deviation value. The threshold is usually 3 times the standard deviation of the background noise level. All the pixels are examined sequentially according to the threshold value. When a pixel above the threshold value is found, neighboring pixels are examined in the region of interest. After that, centroiding calculation starts with all of the pixels in the ROI. Image moment analysis flow diagram architecture is shown in Figure 2.2.

The region of interest is calculated for every pixel as in Equations 2.1-2.4 [14].

$$x_{start} = x - \frac{\alpha_{ROI} - 1}{2} \tag{2.1}$$

$$y_{start} = y - \frac{\alpha_{ROI} - 1}{2} \tag{2.2}$$

$$x_{end} = x_{start} + \alpha_{ROI} \tag{2.3}$$

$$y_{end} = y_{start} + \alpha_{ROI} \tag{2.4}$$

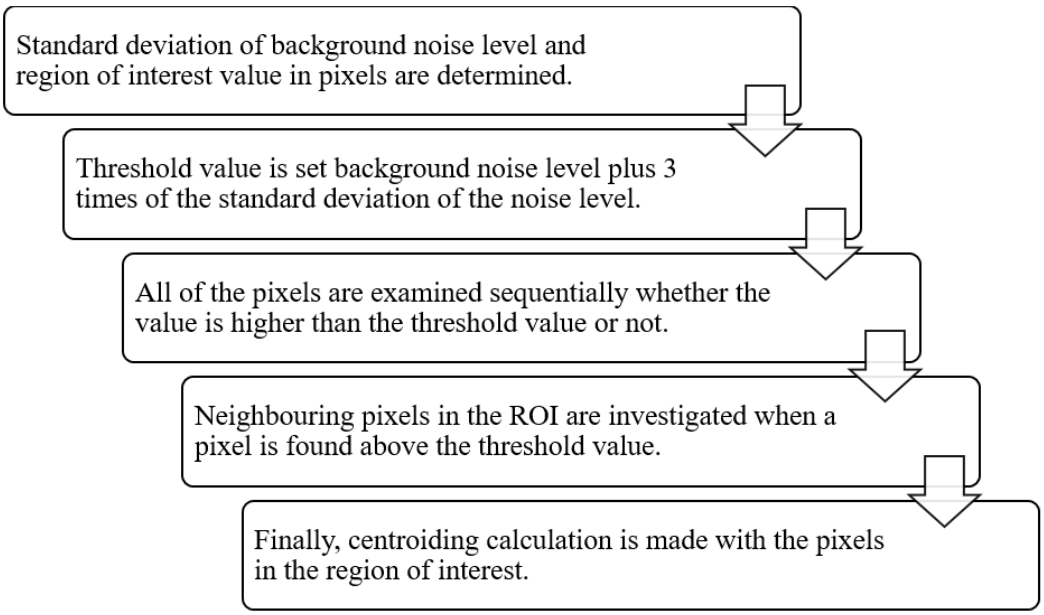


Figure 2.2: Image Moment Analysis Method Flow Diagram Architecture

α_{ROI} : the size of the region of interest in pixels

(x_{start} , y_{start}) : starting pixels in the ROI

(x_{end} , y_{end}) : ending pixels in the ROI

(x , y) : current pixels

After finding a pixel above the threshold value, the centroiding calculation in the ROI is made as in Equations 2.5-2.7[14].

$$B = \sum_{i=x_{start}}^{x_{end}} \sum_{j=y_{start}}^{y_{end}} I(i,j) \tag{2.5}$$

$$x_c = \sum_{i=x_{start}}^{x_{end}} \sum_{j=y_{start}}^{y_{end}} \frac{i \times I(i,j)}{B} \tag{2.6}$$

$$y_c = \sum_{i=x_{start}}^{x_{end}} \sum_{j=y_{start}}^{y_{end}} \frac{j \times I(i,j)}{B} \tag{2.7}$$

B : total brightness in the ROI

$I(i,j)$: light flux in the current pixel

(x_c, y_c) : centroid location

In the image moment analysis method, all objects have brightness value above the threshold are detected as if they are stars. Yet, there are cosmic rays, planets, satellites, and debris etc., so this method finds a lot of stars but at the same time a lot of non-star objects as if they are stars. Therefore, it can be said that this method is not robust.

2.1.2 Point Spread Function Fitting Method

Despite the stars are point sources, starlight on the image plane is not point source because of the defocusing and the other image error sources. The distribution of these sources in the image is described by a mathematical function. The function is called as point spread function [15].
In PSF fitting analysis, the star centroiding information is obtained by making Gaussian curve fitting to star image profiles. To define a 2-D Gaussian curve, FWHM (Full Width Half Maximum) of the profile should be specified in pixels. It is schematized in Figure 2.3 [16].

In Point Spread Function (PSF) fitting method, first of all, pixels above the threshold value are determined as in image moment analysis. The threshold is set generally 3 times the standard deviation of the background noise level. All the pixels are investigated according to this threshold value. Then, a full width half maximum (FWHM) value is determined in pixels. While star profiles are generally similar to the shape of the Gaussian function, the shape of the cosmic rays is generally similar to the impulse function. In PSF fitting analysis, the star location information is obtained by making Gaussian curve fitting to star image profiles. In order to define a 2-D Gaussian curve, FWHM (Full Width Half Maximum) of the profile should be specified in pixels. PSF fitting method flow diagram architecture is shown in Figure 2.4.

There are various different methods to create the Gaussian kernel. DAOFIND [17] is one of the most implemented methods. In photutils package affiliated to the astropy package in Python software language, there is a module named DAOStarFinder [18].

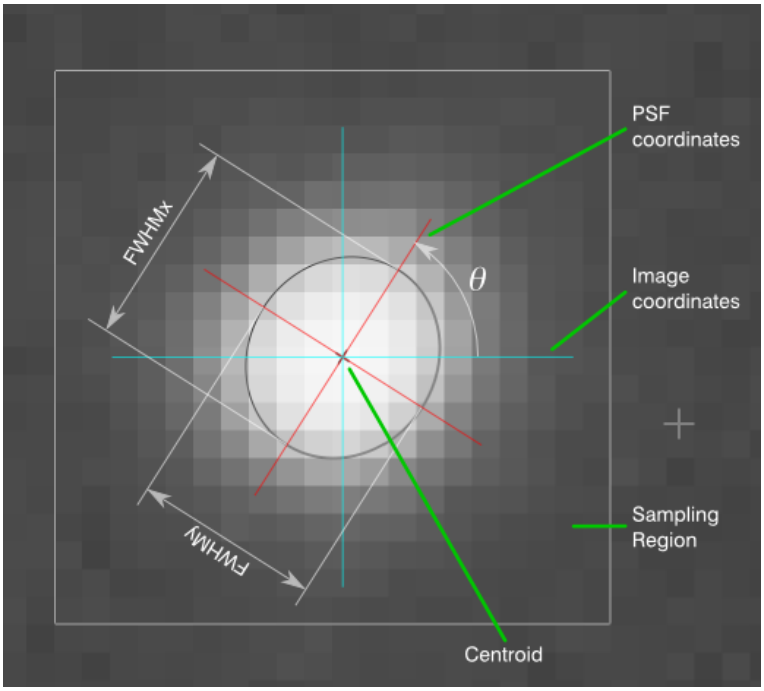


Figure 2.3: PSF fitting analysis - Gaussian curve fitting

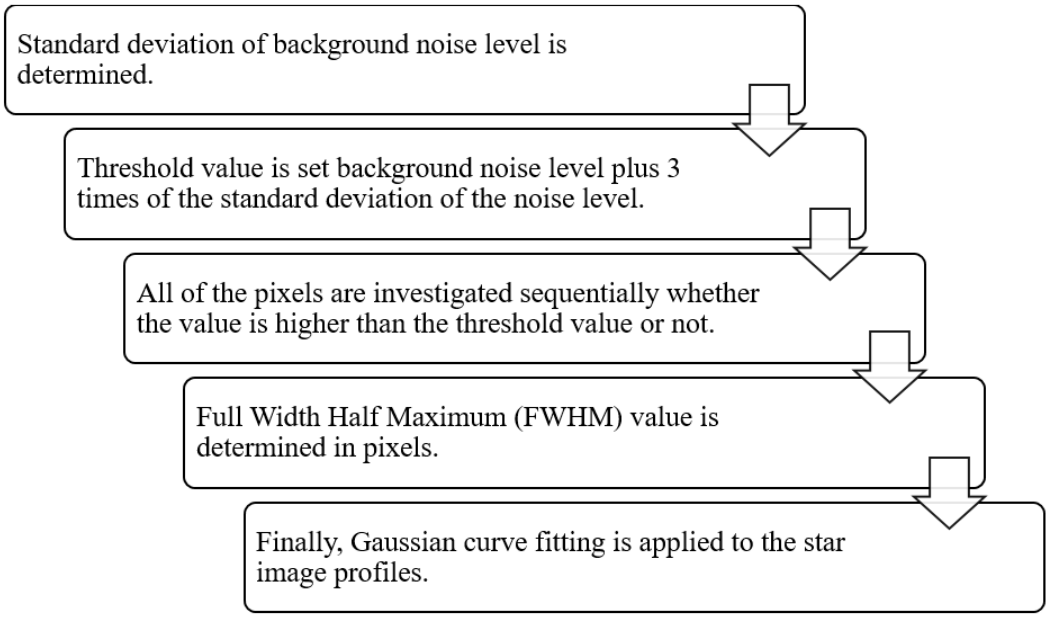


Figure 2.4: PSF Fitting Method Flow Diagram Architecture

The other method is called IRAFStarFinder [19]. While DAOStarFinder module can use a 2D circular or elliptical Gaussian kernel, IRAFStarFinder uses only 2D circular kernel.

There are little differences in parameters used in these methods. For example, DAOStarFinder uses FWHM value, but IRAFStarFinder method uses HWHM value in the algorithm. Moreoever, the truncation radius of the Gaussian kernel (σ_{radius} : input parameter) representation is different in IRAFStarFinder. These differences are represented in Equations 2.8-2.10 [18].

$$fwhm = hwhmpsf \times 2 \tag{2.8}$$

$$\sigma_{radius} = fradius \times \sqrt{2 \times log(2)} \tag{2.9}$$

$$minsep_{fwhm} = 0.5 \times sepmin \tag{2.10}$$

$minsep_{fwhm}$: the minimum separation for detected objects
σ_{radius} : the truncation radius of the Gaussian kernel

Image moment analysis method finds more stars, but it also finds more non-star objects as if they are stars. On the other hand, PSF fitting method finds fewer stars, but it is robust and less computationally expensive, i.e. the duration of finding star centroid with the help of PSF fitting algorithm is less than the image moment analysis.

Table 2.1: Differences between Image Moment Analysis and PSF Fitting Analysis

Properties	**Image Moment Analysis**	**PSF Fitting Analysis**
Number of stars	More	Less
Number of non-star object	More	Less
Duration of centroiding	More	Less

2.2 STAR MATCHING ALGORITHMS

After the calculation of centroids of the stars in the image plane, star identification process is started. First of all, features (parameters) in the image are extracted. Then,

star database which is derived from the star catalog according to the selected star matching algorithm is searched for features. Star database contains features of the stars. When a literature survey is made about the star matching algorithms, it is seen that there are many developments from the past to the present. Briefly, if these developments are explained, first of all, the transition from algorithms that have a limitation of requiring a priori information to algorithms that need not a priori information is a major milestone. In 1992, Liebe simplified the Lost-In-Space problem by bringing the critical link of the feature selection process to database search. Liebe described the use of the nearest two stars to a star chosen as a component of the star pattern, and the angular separations between them as his parameters [20], as shown in Figure 2.7 [21]. In 1996, Quine was reduced database search time dramatically by using a binary search tree approach [20]. Later in 1997, Mortari [22, 23] succeeded to reduce database search time even further by using Search Less Algorithm (SLA) [20]. Mortari retained any pair of stars selection in FOV but he suggested using a "k-vector" to search the database over a period of time, regardless of the size of the database. A few years later, Mortari developed a method called pyramid method. In this method, more features are extracted so it is more robust than the triangle method. Unfortunately, triangle method is faster than the Mortari's pyramid method. Pyramid method, firstly, forms a triangle constituting three stars. This triangle is called primary triangle. After that, a fourth star which is called reference star is selected from the image plane and forms a pyramid. Then, identification parameters are created. They are three angles in the primary triangle and three angles between the reference star and the stars of primary angle vertices. Now, these features are compared with the star database and searches for the solution. If star identification cannot be done, another reference star is selected from the image plane, which is illustrated in Figure 2.5 [21].

Neural networks have started to be used in star identification algorithms around 1989 [24]. Hong suggested usingthe methods of neural network and fuzzy logic for star identification in 2000. Another method called as the quad method is any other geometric method. Yet, it is composed of quads, not triangles. In source extraction, using four sources builds more distinctive features. Therefore, the quad method is the most robust method. It does not give false positives, but it can give no solution case. Firstly, two stars are selected for the basis set that defines the local coordinate sys-

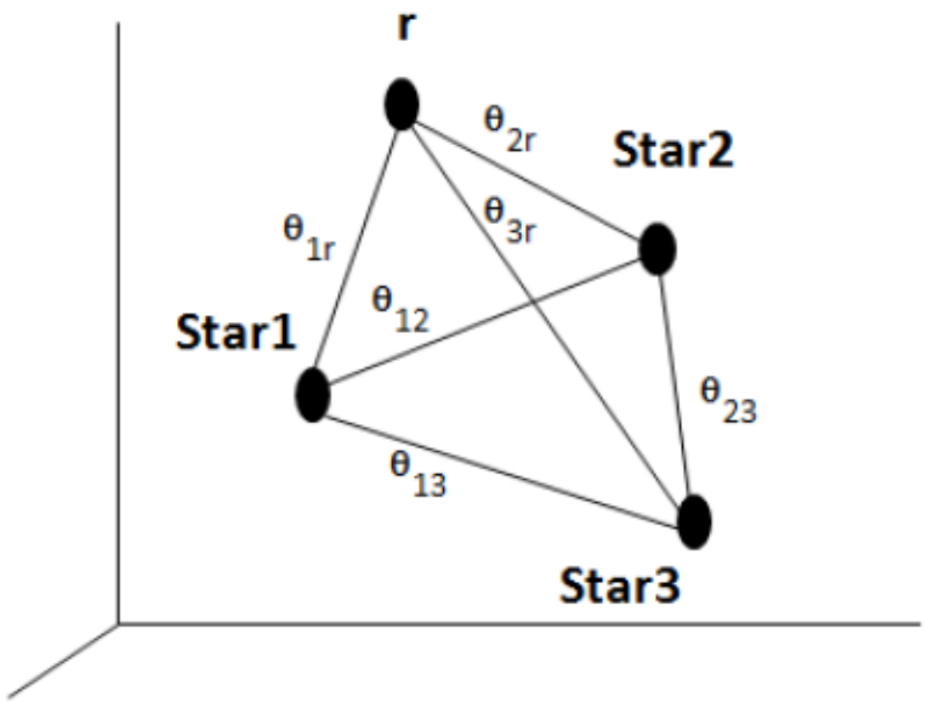


Figure 2.5: Mortari's Pyramid Method

tem. The features are composed of positions of the remaining stars in this coordinate system. The coordinate frame in the quad method is schematized in Figure 2.8 [25]. The locations of the two stars which is located in C and D are hash codes of these two stars. Moreover, hash codes of stars in the catalog are tabulated. Then, the hash codes of the stars in the image plane are compared with the database hash codes. If the star matching occurs, the star identification is completed.

In this thesis, Liebe's triangle method and Astrometry.net's quad method will be investigated further. Triangle method was chosen for its simplicity and quad method was chosen for its robustness. Their implementation theory will be described. Before going on the topic of star matching algorithms deeply, the equatorial coordinate system will be described. This coordinate system is used for astronomical purposes. Astronomers use equatorial coordinate system to describe the position of the sky objects. From now on, in this thesis, this coordinate system will be used to describe the star locations. This is a spherical coordinate system so the position of the stars can be described with two angles. These angles are called as right ascension (RA) and declination (DEC). Right ascension is like longitude and declination is like latitude

for celestial sphere. Right ascension measures eastward along the celestial equator and declination measures perpendicular to the celestial equator [26]. The coordinate system is described in Figure 2.6.

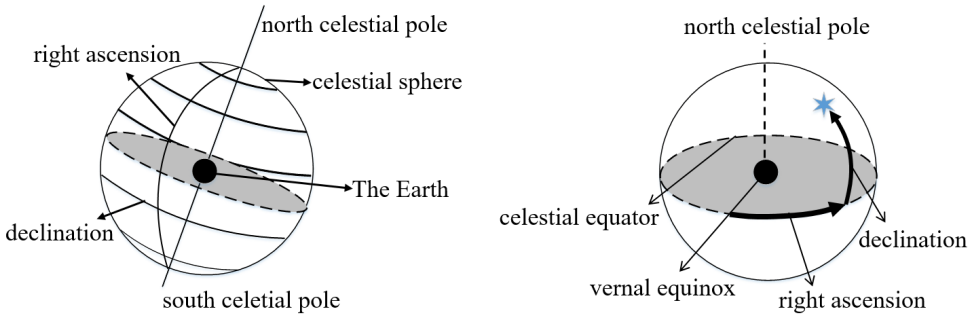


Figure 2.6: Equatorial Coordinate System

2.2.1 Liebe's Triangle Method

The triangle method was first proposed by Liebe in 1995. For this reason, this method is referred to by his name. In this method, three stars in the field of view are determined. When these stars are determined, one of the stars in the image is selected as the central star. After that, the two closest reference stars to the central star are selected. Then, the angular distance between the central star and the reference stars is determined. Moreover, the angle between the lines connecting the central star and the reference stars is also selected as a feature. Finally, these features are compared with the features in the database and star identification process is performed [21].

Before applying the star-matching algorithm, a star centroiding algorithm is applied and the positions of the stars in the image are found. Afterward, the features are extracted from the stars according to the matching algorithm and then matched with the database. After the star identification process is performed, the orientation determination step starts.

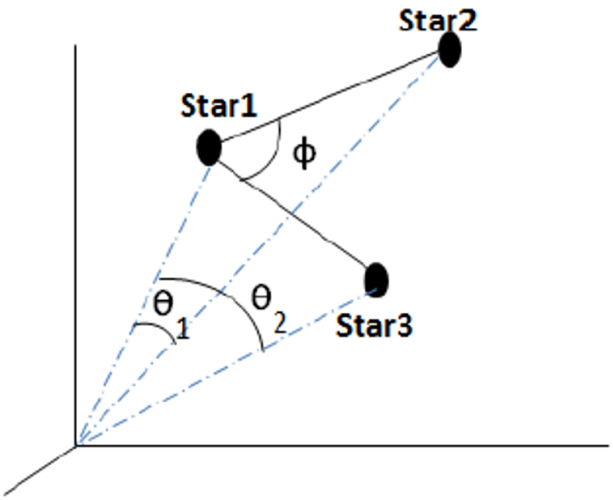


Figure 2.7: Liebe's Triangle Method

2.2.1.1 Database Creation

The stars in the star catalog are filtered according to a certain limiting magnitude value and field of view. Then, each star is selected as the center star and two nearest reference stars are detected. The database is created by calculating the angular distance between the stars and the angle between the lines connecting the central star and the reference stars.

2.2.1.2 Synthetic Image Creation

The stars in the star catalog are filtered according to a certain limiting magnitude value and field of view as in the database creation. In addition, the size of the image to be created and the FWHM information of the stars in the image are fed as an input. Finally, an error is added to the position information of the stars in the catalog, and non-star objects may be added to some images as if they are stars. Also, the midpoint position in the image is randomly selected. According to the determined FOV value, the remaining stars in this area are determined. Then, according to RA and DEC values, the position of the stars is converted from the spherical coordinate system to the Cartesian coordinate system. Finally, the Cartesian coordinate information is transformed into x, y pixel information according to the size of the image to be

generated.

2.2.2 Astrometry.net's Quad Method

Quad method is developed by Dustin Lang et al. using the Astrometry.net platform. This platform has been developed using C and Python languages and it is an open source code. In the quad algorithm, the index files are available for 2Mass and Tycho catalogs. In addition, index files can be created for the desired star map. Another important criterion when creating index files is the field of view [27].

The quad method selects four neighbor stars and the two most distant stars will be corners of the quad. Then, the side lengths of the quad are scaled to 1 and the coordinates of the interior stars are the unique hash codes of the quad method. For example, the location of C is (0.1, 0.1) and the location of the D is (0.7, 0.5), etc.

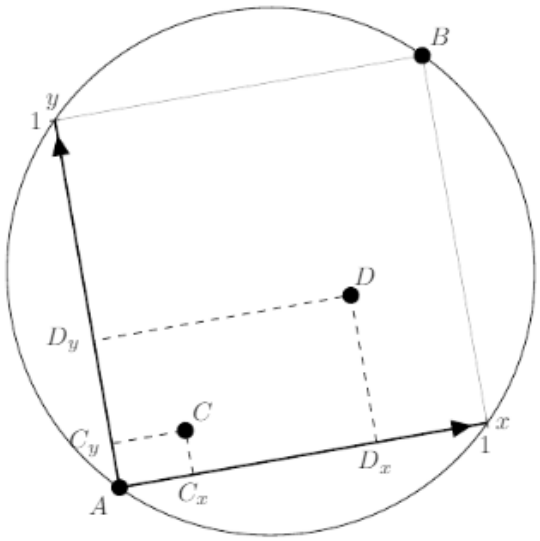


Figure 2.8: Quad Method

The generated hash codes are not affected by the rotation and enlargement/reduction of the image. If the positions of A and B are changed, the coordinates of C and D stars are $(1 - x_C, 1 - y_C)$ and $(1 - x_D, 1 - y_D)$ respectively. Furthermore, by the displacement of the stars C and D, the hash codes are (x_D, y_D) and (x_C, y_C). For this

reason, $x_C \leq x_D$ and $x_C + x_D \leq 1$ must be satisfied in order to get the same code [25].

2.2.2.1 Database Creation

The database created with hash codes is called index. These indices are searched for star groups that match the unique codes in the image. Index files can be created via the "build-astrometry-index" component. Afterward, it is decided how many slices can be divided by "hpsplit" component. According to FOV, index files are divided into slices. The index files are located in the /data folder. In addition, the configuration file that determines the location of the index files and which index files to use is called astrometry.cfg in the /etc folder. Astrometry.net's quad method source code includes PSF centroiding algorithm in itself. A most basic mode of operation is by solve-field component. (x, y) is transferred to the FITS tables called xylist with the xyls extension from the brightest to the faintest, with the centroiding algorithm by taking the input file of the solve-field component image or binary FITS table format and directing it to the related components of the program. This file is passed as an input parameter to the "backend" component. The "backend" looks at the configuration file (astrometry.cfg) to determine which index files to use, how long to match, and how to match according to the parameters specified in the configuration file. It then performs the matching and calls the components that make up the desired output files and passes the output as the input to them [27].

100 synthetic images are created to compare the results of the two star matching algorithms. To identify the stars in these images, the same database which defines the same star catalog, FOV and limiting magnitude is used. Two results will be investigated in this part for two different FOV values. It can be seen that less images are matched with the database for small FOV case because there are fewer stars in the fields that are near the poles and small FOV case is affected much more than bigger FOV. In the synthetic image simulations, the images are created using almost any field in the sky. The results that includes percentage success for two different FOV values are shown in 2.2. Despite the percentage success results of the triangle method are a little bit of better than the quad method, triangle method can give false matches.

Because the quad method's features (hash codes) are unique, it is more robust than the triangle method. Therefore, the standard deviation of the position error in terms of the right ascension and declination in the quad method is much smaller compared to the triangle method.

Table 2.2: Results of the comparison of the two star matching algorithms

Methods	**FOV =10°**	**FOV = 15°**
Triangle Method	97/100	98/100
Quad Method	96/100	97/100

Because the quad method is more robust than the triangle method, this method will be investigated much further. In order to analyze quad method deeply with real images taken from the program simulates the real sky, Stellarium program is used in the thesis. Stellarium is a free open source astronomical software which can be used to observe the realistic sky. By using this software, stars can be simulated at any time of anywhere. Moreover, atmospheric effects, light pollution etc. may be included in this simulation. On the other hand, non-star objects which are planets, satellites, asteroids etc. are simulated in Stellarium program. It simulates stars according to the different star catalog database. Date and time can be set at any time. Furthermore, the location and height where stars are observed can be arranged. Another feature of this program is that the ocular simulation/telescope configuration arrangements can be made. By using telescope configuration, camera and optics parameters can be changed to any meaningful configuration for the star tracker sensor [28]. Stellarium provides lots of useful information such as star locations, magnitudes, non-star objects like planets and satellites, etc. Moreover, it has ability to control the date and time, location and atmospheric effects, etc. Stellarium user interface is shown in Figure 2.9. In this configuration, atmospheric effects are not represented, ground is represented. FOV and available FPS (frame per second) information is also represented in the illustration. Furthermore, this place simulates Ankara at 874 meters height, the date is 2018/06/10 and the time is approximately 21:00.

In the location window of the Stellarium program, there is a wide location list all over the world. Moreover, any location is chosen from the location map or according to the

current latitude, longitude, and height information. Furthermore, anyone can choose any other planet except Earth for observation. In the Date and Time window, any date and any time can be represented within the time limits allowed by the Stellarium program. Optionally, time is set at anything by using Julian Day representation form of the time. Thus, stars in the sky can be simulated and observed at any time of anywhere. In the Sky and Viewing options window, there are options composed of sky, deep sky objects, markings, landscape, and starlore. In this thesis, the most important ones are the sky and deep sky objects parts. In the sky options part, magnitude limitation can be set at the simulation. By using this limitation, a magnitude limit value can be put for the star tracker camera. Moreover, atmospheric effects can be included and light pollution data can be taken from the location database. In the Deep Sky Objects options part, there are catalogs of the deep sky objects. Anyone can select the deep sky object catalog and object type from here.

Figure 2.9: Stellarium User Interface

In the Stellarium program, there is a useful part to visualize the sky according to the star tracker sensor used in the system. In the top right configuration window, there is an options part for the image sensor and optics that are used in the system. In the Figure 2.10, an image with the properties indicated at the right top of the figure is simulated in the program. At the same time, the right ascension and declination information of the middle point is represented according to J2000 coordinate system

in this case.

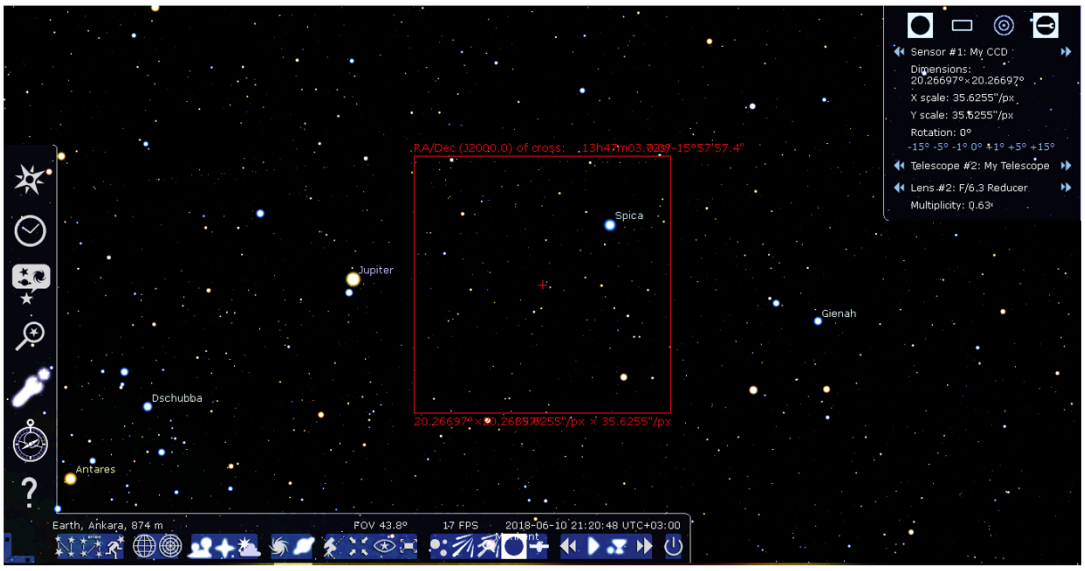


Figure 2.10: Stellarium Image Sensor Frame

With the image sensor configuration window, telescope configuration window and lens configuration window, a star tracker detector, lens configurations can be simulated and then it can be thought of as simulated image is obtained from the star tracker sensor. Then, the middle point right ascension and declination information of this image is found by using the quad method. Finally, Stellarium and quad method right ascension and declination information can be compared as if Stellarium result is correct. In fact, this result also contains errors but this can be thought of as a master information.

In the light of above information, Stellarium program is used to create an image and to obtain master right ascension/declination information.

The image which is represented in the Figure 2.11 is created in the location of Ankara from the ground (actually 874 meters above the sea level) at 2 p.m. approximately. In this image, atmospheric effects are not included.

For the image sensor and optics configuration part, the values represented below are used to visualize this image.

Pixel number: 2048 x 2048

Pixel size: 5.5 μm x 5.5 μm

F#: 4

Focal length: 50 mm

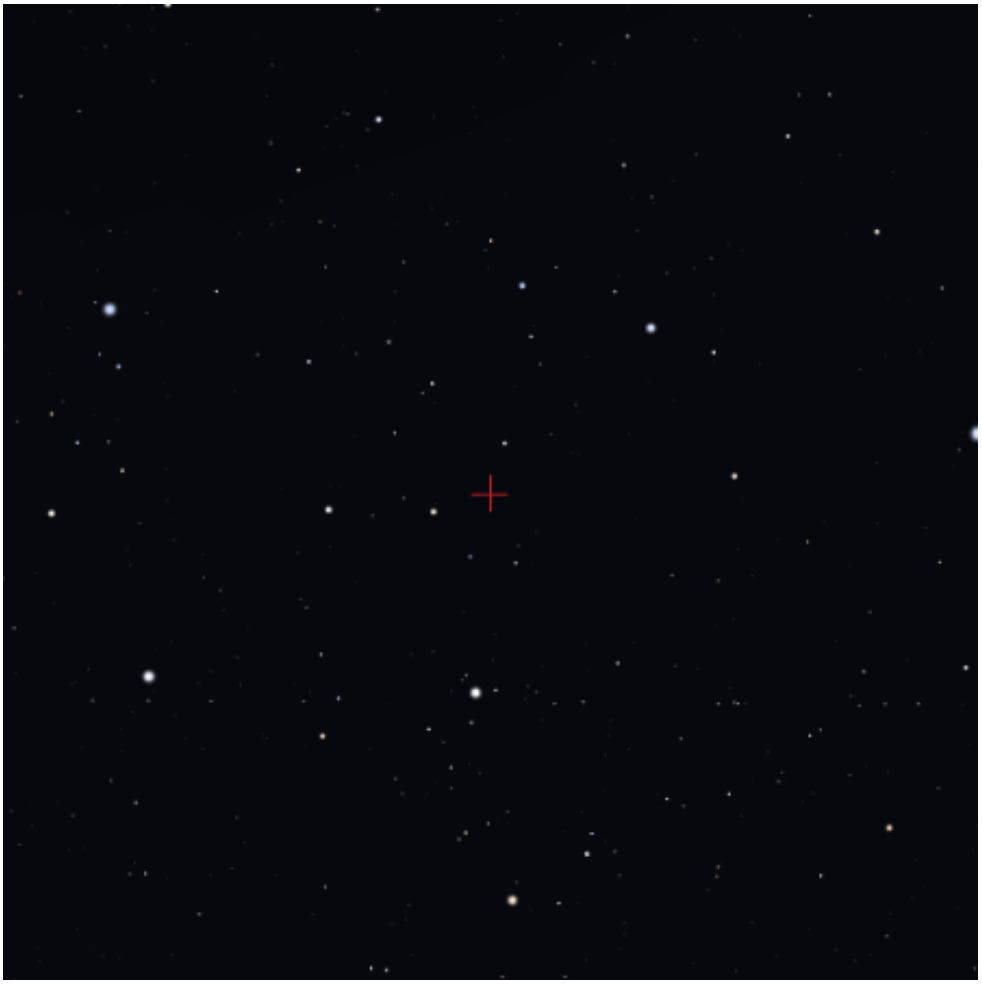


Figure 2.11: A Stellarium simulated image from the ground at 2 pm excluded from the atmospheric effects

Quad method runs on Linux/Mac operating system. In this thesis, Ubuntu operating system was used for this purpose. The quad method simply runs with the solve-field component at the terminal. This can be achieved by writing the command solve-field and the image wanted to be solved. By this way, the image simulated in the Stellarium in the Figure 2.11 was solved and the output in the terminal is like in the Table 2.3. Field center RA/DEC information can be seen from this output. Moreover,

quad algorithm gives some output files to explain and show the result obtained.

Table 2.3: Quad method solve-field component in Ubuntu terminal

Field Center (RA, Dec) (RA H:M:S Dec D:M:S)	**Field Size**	**Stars in the field**
(213.714991, -1.853957) deg (14:14:51.598, -01:51:14.245)	20.0256 x 20.0643 deg	Parts of the constellation of Virgo The star 80Vir The star 82Vir The star 84Vir The star 90Vir The star 92Vir ...

These output files contains pixel location and RA/DEC information of the sources found in the image, description of the quad that provides the image solved and images that show the sources found in the image and index stars, the representation of the quad that ensures matching, and a new FITS (Flexible Image Transport System) image which contains World Coordinate System headers, etc.

Now, the Stellarium RA/DEC result and Quad method RA/DEC result are compared in the Table 2.4.

Table 2.4: The difference between the Stellarium and Quad method result

Stellarium Result	**Quad Method Result**	**Difference**
RA: 14 hr 14 m 55.73 s DEC: -1 °53’ 05.4”	RA: 14 hr 14 m 51.598 s DEC: -1 °51’ 14.245”	RA: 4.132 s DEC: 1’ 51.155”

In the Figure 2.12, Quad algorithm gives the output of the representation of the quad that solves the image.

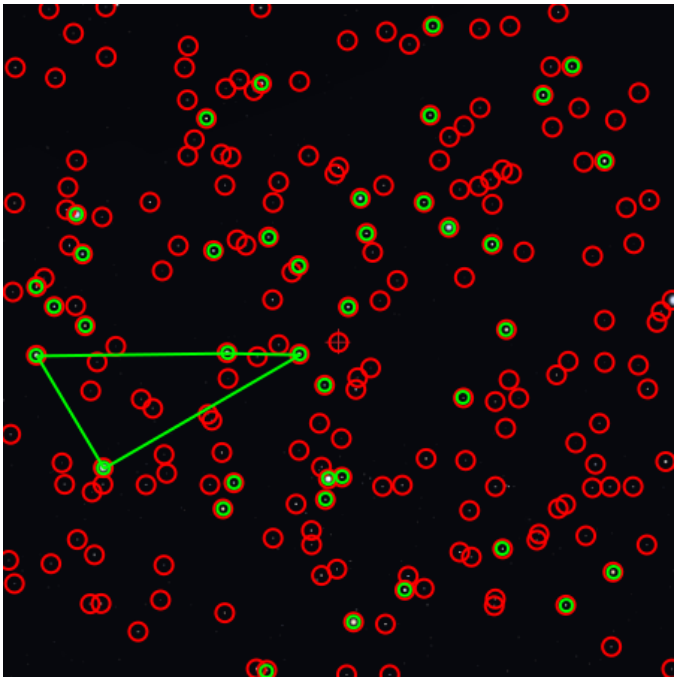


Figure 2.12: The quad used in matching (red ones are sources, green ones are from the index catalog)

The sources (stars) found in the image is represented in the Figure 2.13.

Figure 2.13: Stars found in the image

The image which is represented in the Figure 2.14 is created in the location of Ankara from 50 kilometers above the sea level at 14 p.m. approximately. In this image, atmospheric effects are not included.

For the image sensor and optics configuration part, the values represented below are used to visualize this image.

Pixel number: 2048 x 2048
Pixel size: 5.5 μm x 5.5 μm
F#: 4
Focal length: 50 mm

Figure 2.14: Another Stellarium simulated image from 50 km above at 2 pm excluded from the atmospheric effects

The quad method is simply running with the solve-field component at the terminal. This can be achieved by writing the command solve-field and the image wanted to be solved. By this way, the image simulated in the Stellarium in the Figure 2.14 was solved and the output in the terminal is like in the Table 2.5. Field center RA/DEC information can be seen from this output. Moreover, quad algorithm gives some output files to explain and show the result obtained.

Table 2.5: Quad method solve-field component in Ubuntu terminal

Field Center (RA, Dec) (RA H:M:S Dec D:M:S)	**Field Size**	**Stars in the field**
(121.595304, -9.369343) deg (08:06:22.873, -09:22:09.634)	20.1236 x 20.0816 deg	Parts of the constellation of Monoceros The star 25Mon The star aMon The star 4Pup The star 5Pup The star 6Pup ...

Now, the Stellarium RA/DEC result and Quad method RA/DEC result are compared in the Table 2.6.

Table 2.6: The difference between the Stellarium and Quad method result

Stellarium Result	**Quad Method Result**	**Difference**
RA: 8 hr 06 m 21.22 s DEC: -9°23' 31.2"	RA: 8 hr 06 m 22.873 s DEC: -9°22' 09.634"	RA: 1.653 s DEC: 1' 21.566"

In the Figure 2.15, Quad algorithm gives the output of the representation of the quad that solves the image. The sources (stars) found in the image is represented in the Figure 2.16.

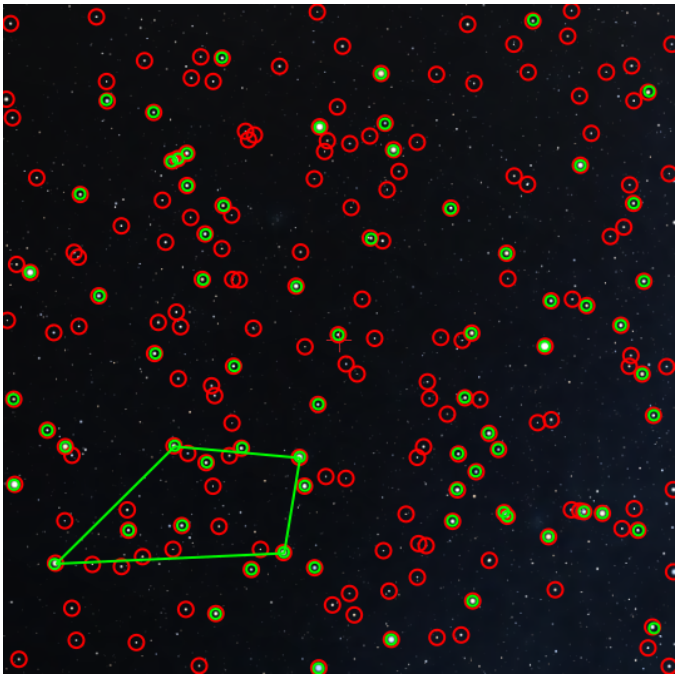


Figure 2.15: The quad used in matching (red ones are sources, green ones are from the index catalog)

Figure 2.16: Stars found in the image

The image which is represented in the Figure 2.17 is created in the location of Ankara

from 50 kilometers above the sea level at 14 p.m. approximately. In this image, atmospheric effects are included.

For the image sensor and optics configuration part, the values represented below are used to visualize this image.

Pixel number: 2048 x 2048
Pixel size: 5.5 μm x 5.5 μm
F#: 4
Focal length: 50 mm

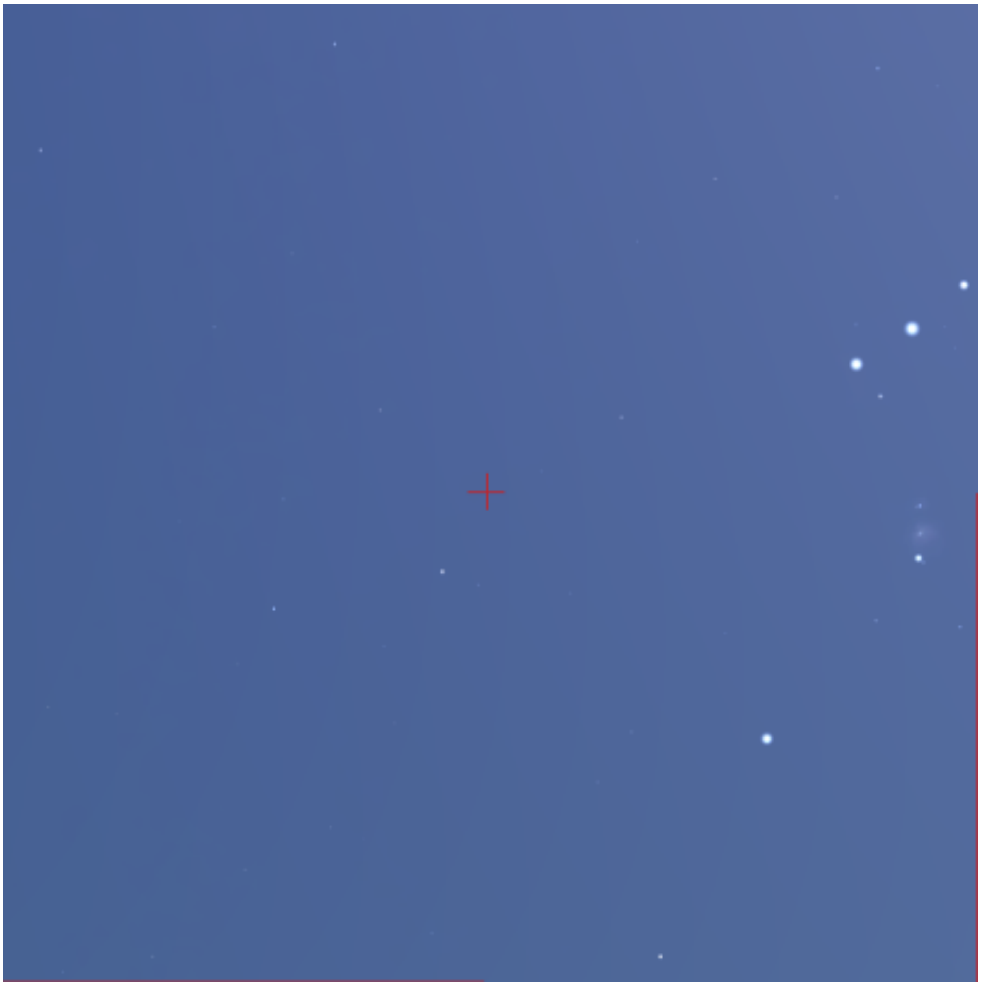


Figure 2.17: Another Stellarium simulated image from 50 km above at 2 pm included from the atmospheric effects

The quad method is simply running with the solve-field component at the terminal. This can be achieved by writing the command solve-field and the image wanted to be solved. By this way, the image simulated in the Stellarium in the Figure 2.17 was solved and the output in the terminal is like in the Table 2.7. Field center RA/DEC information can be seen from this output. Moreover, quad algorithm gives some output files to explain and show the result obtained.

Table 2.7: Quad method solve-field component in Ubuntu terminal

Field Center (RA, Dec) (RA H:M:S Dec D:M:S)	**Field Size**	**Stars in the field**
(92.709035, -4.638511) deg (06:10:50.168, -04:38:18.638)	20.0187 x 20.1052 deg	Parts of the constellation of Monoceros Parts of the constellation of Orion The star Mintaka The star uOri The star 38Ori The star 42Ori ...

Now, the Stellarium RA/DEC result and Quad method RA/DEC result are compared in the Table 2.8.

Table 2.8: The difference between the Stellarium and Quad method result

Stellarium Result	**Quad Method Result**	**Difference**
RA: 6 hr 11 m 13.49 s DEC: -4°36’ 52.4”	RA: 6 hr 10 m 50.168 s DEC: -4°38’ 18.638”	RA: 23.322 s DEC: 1’ 26.238”

In the Figure 2.18, Quad algorithm gives the output of the representation of the quad that solves the image. The sources (stars) found in the image is represented in the Figure 2.19.

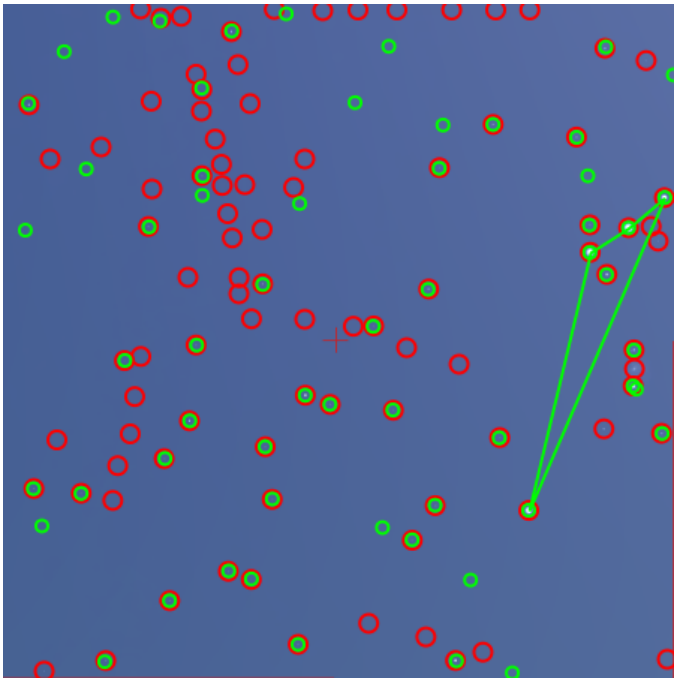


Figure 2.18: The quad used in matching (red ones are sources, green ones are from the index catalog)

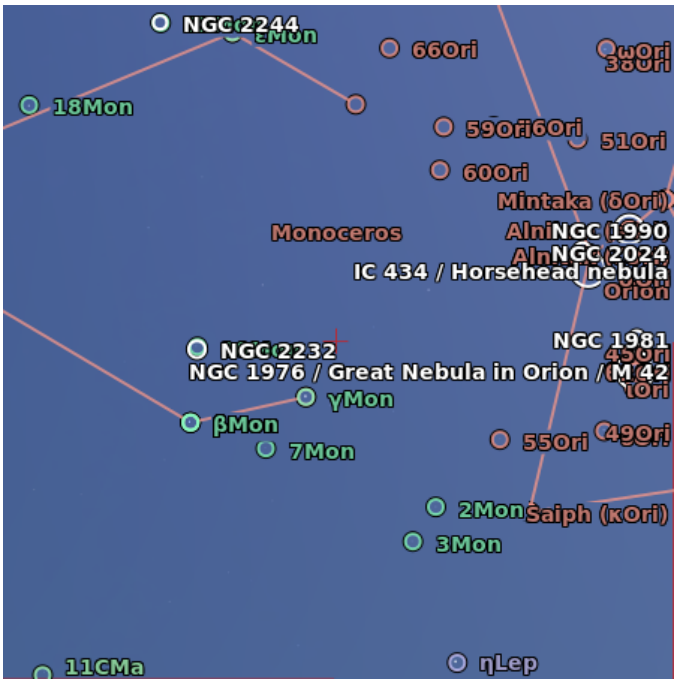


Figure 2.19: Stars found in the image

It is observed that atmospheric attenuation, time and altitude affect the star tracker sensor result. Atmospheric effects cause observing less stars due to atmospheric attenuation. Moreover, day and night observations are different because of atmospheric effects. Also, different sky observations are created by the atmosphere at different altitudes. When the atmospheric effect is excluded from the observations, altitude and day-night observation differences decrease. When above 3 different examples are examined, it is seen that the first and second examples are excluded from the atmospheric effects and the only difference is the altitude difference. The results of these two examples are nearly similar to the each other. When the atmosphere is excluded from the simulation, altitude effect is very small in the results. On the other hand, when the second and third examples are examined, it is concluded that the result of the second one is better than the third one because it includes atmospheric effects. Despite the simulations of the two examples are made at the same altitude, results are different due to atmosphere. In real world, because of the atmosphere is always included, altitude, time of the day, and also latitude/longitude affects the sky observations. Therefore, in order to make good observations at any time of anywhere, the altitude should be as high as possible where the atmospheric effects are too low. Thus, in the last chapter, it is chosen that the altitude of the aircraft in which the star tracker sensor is attached to is high.

2.3 ATTITUDE DETERMINATION ALGORITHMS

Finally, the calculation of attitude determination of the space vehicle is made just after the star identification process. To determine the attitude of the space vehicle, body vectors of stars observed in the image plane and inertial vectors of stars in the catalog are compared. Stars in the catalog are defined in terms of the Right Ascension and Declination. Inertial frame unit vectors are described as $\overrightarrow{V_1}, ..., \overrightarrow{V_n}$ and body frame unit vectors are described as $\overrightarrow{W_1}, ..., \overrightarrow{W_n}$. Then, an orthogonal matrix A is described as follows:

$$A\overrightarrow{V_1} = \overrightarrow{W_1} \tag{2.11}$$

The orthogonal matrix A does not have an exact solution because of measurement errors. To find a rotation matrix A, Wahba suggests the use of direction cosine matrix [29].
Minimization of matrix A is described mathematically as [29]:

$$\sum_{j=1}^{n} \left\| \overrightarrow{W_j} - A\overrightarrow{V_j} \right\|^2 \tag{2.12}$$

First, the 2D position information of the star in the camera coordinate plane is converted into 3-dimensional position information in the camera coordinate system by knowing the focal length of the optical system. This conversion is carried out according to the "pinhole" camera model. This model is shown in the Figure 2.20.

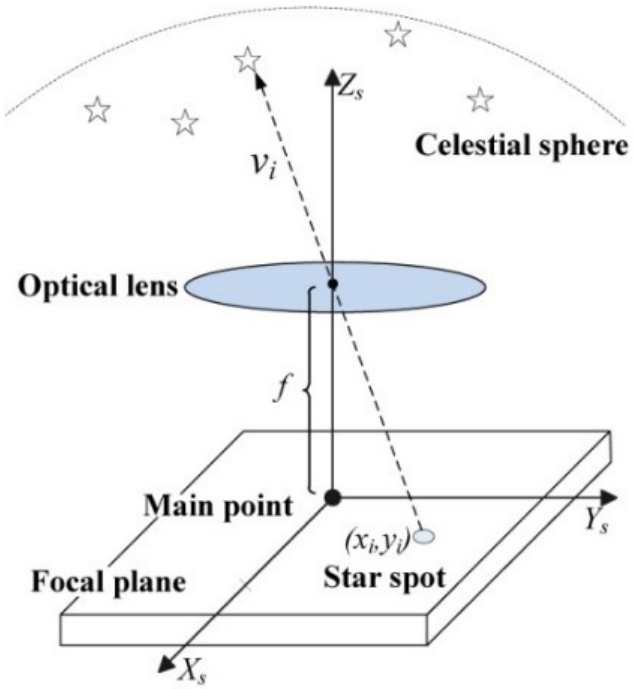


Figure 2.20: Ideal pinhole camera model

With the ideal pinhole camera model shown above, the 2D to 3D conversion equation

looks like this: [30]

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = v_i = \frac{1}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + f^2}} \begin{bmatrix} -(x_i - x_0) \\ -(y_i - y_0) \\ f \end{bmatrix} \tag{2.13}$$

(x_0, y_0) : midpoint of focal plane
(x_i, y_i) : the location of the stars observed in the focal plane
f : focal length

This equation describes the ideal pinhole camera model. However, faults in the system cause it to deviate from the ideal pinhole camera model. These faults are divided into faults that affect the (x_i, y_i) position and faults that affect the focal length (f). Errors affecting the (x_i, y_i) location are the errors in the star centroiding, electronic errors caused by the detector, and atmospheric effects, etc. Errors that affect the focal length are the optical misalignments, coma error and the optical flaws caused by temperature-induced effects, etc. Time-independent errors should be calibrated in the ground, and time-varying errors should be calibrated in real-time to correct the orientation of the system. The other one of the faults is the fault caused by the high dynamics of the vehicle that the system is attached to. Distortions (motion blur effect) arise due to the dynamic nature of the vehicle. Various filters are used to eliminate this error. Wiener and Richardson-Lucy filters are the most commonly used filters. By using these filters, the blurred and noisy images are restored with the calculation of motion PSF (Point Spread Function).

The difference between real and ideal values is as follows: [30]

$$(x_0', y_0') = (x_0 + \Delta x_0, y_0 + \Delta y_0) \tag{2.14}$$

$$f' = f + \Delta f \tag{2.15}$$

$$(x_i', y_i') = (x_i + \Delta x_i, y_i + \Delta y_i) \tag{2.16}$$

$$r_i = \sqrt{x_i^2 + y_i^2} \tag{2.17}$$

$$\Delta x_i = x_i * (k_1 * r_i^2 + k_2 * r_i^4) \tag{2.18}$$

$$\Delta y_i = y_i * (k_1 * r_i^2 + k_2 * r_i^4) \tag{2.19}$$

(k_1, k_2) : coefficients determined by laboratory measurements
$(\Delta x_0, \Delta y_0)$: midpoint error of focal plane
$(\Delta x_i, \Delta y_i)$: the location error of the stars observed in the focal plane
Δf : focal length error

The actual star vector measurement is as follows: [30]

$$v_i' = \frac{1}{\sqrt{(x_i' - x_0')^2 + (y_i' - y_0')^2 + f'^2}} \begin{bmatrix} -(x_i' - x_0') \\ -(y_i' - y_0') \\ f' \end{bmatrix} \tag{2.20}$$

(x_0', y_0') : midpoint of focal plane
(x_i', y_i') : the location of the stars observed in the focal plane
f' : focal length

Once the star matching phase is completed, the orientation calculation phase begins. At this stage, the star's position in the J2000 inertial coordinate system will be known. The two-dimensional position information in the J2000 spherical coordinate system is also moved to the three dimensions using the following formula:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} cos\delta cos\alpha \\ cos\delta sin\alpha \\ sin\delta \end{bmatrix} \tag{2.21}$$

α : Right ascension defined in J2000 inertial coordinate system
δ : Declination defined in J2000 inertial coordinate system

In addition, the position in the camera coordinate system was found through the pin-hole camera model. The calculation of the orientation is generally based on the Wahba problem.

On the basis of this problem lies the finding of matrix A which transforms Vector 1 into Vector 2. In the star tracking concept, this means that the star-tracker sensor is going to find the matrix from the camera coordinate system to the J2000 coordinate system. This matrix corresponds to the smallest value of the following cost function [31].

$$J(A) = \frac{1}{2}\sum_{k=1}^{N} a_k \left\| W_k - A * V_k \right\|^2 \tag{2.22}$$

$J(A)$: cost function
W_k : body (camera) star vectors
V_k : inertial (J2000) star vectors
a_k : weights
N : measurement (star) number

There are three main methods to solve the Wahba problem. The first one is single reference vector method, the second one is double reference vector method and the last one is multiple reference vector. Single vector reference method uses only one observed vector so this method cannot give fully determine attitude. Sun and Earth sensors can give only one vector direction so full attitude vector cannot be obtained. On the other hand, star trackers can have multiple vector observations by sensing multiple stars. Therefore, multiple camera is not a necessity to fully determine attitude. Yet, by having more than one camera head, the accuracy can be increased. Double reference vector method uses two non-collinear vectors. Therefore, fully determine attitude can be obtained. Triad algorithm is an example of this method. Furthermore, in the multiple reference vector method, there are more than two non-collinear vectors in the image so fully determine attitude can be obtained also in this case [32]. Because there are more than two vectors, these vectors are weighted according to their importance/precision.

There are lots of algorithms in order to find the attitude knowledge. These are the Extended Kalman filter, QUEST (Quaternion Estimation), ESOQ (Optimal Quaternion Estimator), Least Squares, q-Davenport, AIM (Attitude Estimates Using Image Match) and TRIAD methods.

2.3.1 Least Squares

Since the star tracker system has noise sources, it will cause measurement error. This error will be modeled as follows, and then the smallest value of the "cost" function will be found [33].

$$\widetilde{e_l} = A\widetilde{r_l} - \widetilde{b_l} \tag{2.23}$$

$$J = \frac{1}{2} e^t e \tag{2.24}$$

$$J = \frac{1}{2}\left(\tilde{b}^T\tilde{b} - 2\tilde{b}^T A\tilde{r} + \tilde{r}^T\tilde{A}^T A\tilde{r}\right) \tag{2.25}$$

$$\nabla_r J = \begin{bmatrix} \frac{\partial J}{\partial \tilde{r}_1} \\ . \\ . \\ . \\ \frac{\partial J}{\partial \tilde{r}_n} \end{bmatrix} = A^T A\tilde{r} - A^T\tilde{b} = 0 \tag{2.26}$$

$$\left(A^T A\right)\tilde{r} = A^T\tilde{b} \tag{2.27}$$

$$\tilde{r} = \left(A^T A\right)^{-1} A^T\tilde{b} \tag{2.28}$$

R is the inertial coordinate system and r is the coordinate system of the star tracker

sensor. A represents the transformation matrix.

$$\begin{bmatrix} R_x \\ R_y \\ R_z \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} \tag{2.29}$$

$$\begin{bmatrix} R_x \\ R_y \\ R_z \end{bmatrix} = \begin{bmatrix} a_{11}r_x & a_{12}r_y & a_{13}r_z \\ a_{21}r_x & a_{22}r_y & a_{23}r_z \\ a_{31}r_x & a_{32}r_y & a_{33}r_z \end{bmatrix} \tag{2.30}$$

$$\begin{bmatrix} R_x \\ R_y \\ R_z \end{bmatrix} = \begin{bmatrix} r_x & r_y & r_z & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & r_x & r_y & r_z & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & r_x & r_y & r_z \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{31} \\ a_{32} \\ a_{33} \end{bmatrix} \tag{2.31}$$

$$\hat{y} = A\hat{x} \tag{2.32}$$

$$\begin{bmatrix} R_x \\ R_y \\ R_z \end{bmatrix} = \hat{y}, \begin{bmatrix} r_x & r_y & r_z & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & r_x & r_y & r_z & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & r_x & r_y & r_z \end{bmatrix} = A, \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{31} \\ a_{32} \\ a_{33} \end{bmatrix} = \hat{x} \tag{2.33}$$

$$\hat{x} = \left(A^T A\right)^{-1} A^T \hat{y} \tag{2.34}$$

2.3.2 TRIAD

In the TRIAD algorithm, two non-parallel unit vectors are obtained in the inertial coordinate system and the coordinate system of the star tracker [33].

$$\begin{aligned} A\hat{V}_1 &= \hat{W}_1 \\ A\hat{V}_2 &= \hat{W}_2 \end{aligned} \tag{2.35}$$

$\hat{V}_1$ & $\hat{V}_2$: Unit vectors defines stars in the inertial frame
$\hat{W}_1$ & $\hat{W}_2$: Unit vectors defines stars in the star tracker body frame (camera frame)

Then the algorithm finds the following triad column matrices:

$$\begin{aligned} \hat{r}_1 &= \hat{V}_1 \\ \hat{r}_2 &= \left(\hat{V}_1 \times \hat{V}_2\right) / \left|\hat{V}_1 \times \hat{V}_2\right| \\ \hat{r}_3 &= \left(\hat{V}_1 \times \left(\hat{V}_1 \times \hat{V}_2\right)\right) / \left|\hat{V}_1 \times \hat{V}_2\right| \end{aligned} \tag{2.36}$$

and

$$\begin{aligned} \hat{s}_1 &= \hat{W}_1 \\ \hat{s}_2 &= \left(\hat{W}_1 \times \hat{W}_2\right) / \left|\hat{W}_1 \times \hat{W}_2\right| \\ \hat{s}_3 &= \left(\hat{W}_1 \times \left(\hat{W}_1 \times \hat{W}_2\right)\right) / \left|\hat{W}_1 \times \hat{W}_2\right| \end{aligned} \tag{2.37}$$

There is a matrix A that provides the following equation.

$$A\hat{r}_i = \hat{s}_i \tag{2.38}$$

This matrix A is defined as:

$$\begin{aligned} A &= \textstyle\sum_{i=1}^{3} \hat{s}_i \hat{r}_i^T \\ M_{ref} &= [\hat{r}_1 : \hat{r}_2 : \hat{r}_3] \\ M_{obs} &= [\hat{s}_1 : \hat{s}_2 : \hat{s}_3] \end{aligned} \tag{2.39}$$

The attitude determination matrix is defined in the following equation:

$$A = M_{obs} M_{ref}^T \tag{2.40}$$

or

$$A = r_1 \cdot s_1^T + r_2 \cdot s_2^T + r_3 \cdot s_3^T \tag{2.41}$$

2.3.3 q-method

This algorithm is based on the Wahba problem and solves this problem by solving the eigenvalue problem [34, 35].
An optimum matrix A provides the smallest value that the "loss (cost)" function can take. Another way to find the matrix A is to find the largest matrix A that provides the "gain" function.

$$\sum_{i=1}^{n} a_i = 1 \tag{2.42}$$

$$g(A) = 1 - L(A) = \sum_{i=1}^{n} a_i \hat{W}_i^T A \hat{V}_i = q^T K q \tag{2.43}$$

$$g(A) = \sum_{i=1}^{n} a_i tr\left[\hat{W}_i^T A \hat{V}_i\right] \tag{2.44}$$

a_i : Weight set

To continue with this algorithm, the K(4x4) matrix needs to be computed.

$$K = \begin{bmatrix} S - \sigma I & Z \\ Z^T & \sigma \end{bmatrix} \tag{2.45}$$

$$S = B + B^T = \sum_{i=1}^{n} a_i \left(\hat{W}_i \hat{V}_i^T + \hat{V}_i \hat{W}_i^T\right) \tag{2.46}$$

$$B = \sum_{i=1}^{n} a_i \hat{W}_i \hat{V}_i^T \tag{2.47}$$

$$Z = \sum_{i=1}^{n} a_i \left(\hat{W}_i \times \hat{V}_i\right) \tag{2.48}$$

$$\sigma = trB = \sum_{i=1}^{n} a_i \hat{W}_i \cdot \hat{V}_i \tag{2.49}$$

In the light of this information, the "gain" function can be edited as follows:

$$g(\tilde{q}) = \left(q^2 - Q \cdot Q\right) trB^T + 2tr\left[QQ^T B^T\right] + 2qtr\left[QB^T\right] \tag{2.50}$$

$$Q = \begin{bmatrix} 0 & Q_3 & -Q_2 \\ -Q_3 & 0 & Q_1 \\ Q_2 & -Q_1 & 0 \end{bmatrix} \tag{2.51}$$

$$g(\tilde{q}) = \tilde{q}^T K \tilde{q} \tag{2.52}$$

A new gain function can be defined as follows:

$$g'(\tilde{q}) = \tilde{q}^T K \tilde{q} - \lambda \tilde{q}^T \tilde{q} \tag{2.53}$$

$$K\tilde{q} = \lambda \tilde{q} \tag{2.54}$$

λ : eigenvalue of the K matrix

$\tilde{q}$: eigenvector of the K matrix

The largest value for $g'(\tilde{q})$ is found by choosing the largest eigenvalue of the K matrix.

$$K\tilde{q}_{opt} = \lambda_{max} \tilde{q}_{opt} \tag{2.55}$$

2.3.4 Quest

Unlike the Extended Kalman filter, the QUEST algorithm does not require initial orientation information. But QUEST cannot cope with complicated dynamic models and needs time history to update the orientation in order to be able to use the measurement data coming from different times [34, 35].

$$K\tilde{q}_{opt} = \lambda_{max} \tilde{q}_{opt} \tag{2.56}$$

The above equation corresponds to the following two equations:

$$[(\lambda_{max} + trB)\, I - S]\, q = q_4 z \tag{2.57}$$

$$(\lambda_{max} - trB)\, q_4 = q^T z \tag{2.58}$$

In the light of above equations:

$$adj\,[(\lambda_{max} + trB)\, I - S] = \alpha I + \beta S + S^2 \tag{2.59}$$

$$\alpha = \lambda_{max}^2 - (trB)^2 + tr(adjS) \tag{2.60}$$

$$\beta = \lambda_{max} - trB \tag{2.61}$$

$$\gamma = det[(\lambda_{max} + trB)I - S] = \alpha(\lambda_{max} + trB) - detS \tag{2.62}$$

$$x = (\alpha I + \beta S + S^2)z \tag{2.63}$$

$$q_{opt} = \frac{1}{\sqrt{\gamma^2 + |x|^2}} \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.64}$$

The characteristic equation is as follows:

$$det(K - \lambda_{max} I) = 0 \tag{2.65}$$

If the optimal "loss" function takes a small value,

$$L(A_{opt}) = \lambda_0 - \lambda_{max} \tag{2.66}$$

λ_{max} takes a value close to the following value.

$$\lambda_0 \sum a_i \tag{2.67}$$

Newton-Raphson is a method to solve λ_{max}. Initially, the iteration starts with λ_0. QUEST method is less robust than the q-Davenport method because characteristic equation is one of the worst ways to find an eigenvalue [34, 35].

2.3.5 ESOQ

This algorithm is based on the Wahba problem and solves this problem by solving the eigenvalue problem. But it avoids the sequential rotations of the QUEST algorithm by computing the maximum eigenvalue value differently [34]

2.3.6 SVD

The matrix B is calculated as follows according to the calculations made with n observed stars from the star identification algorithm [36].

$$B = \sum_{i=1}^{n} b_i r_i^T \tag{2.68}$$

b_i : Observed stars unit vectors in the camera frame
r_i : Observed stars unit vectors in the inertial frame

The singular value decomposition of the matrix B is performed as follows:

$$B = USV^T \tag{2.69}$$

U & V: "Orthogonal" matrices

S : Singular values diagonal matrix

$$U_+ = U \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & detU \end{bmatrix} \tag{2.70}$$

$$V_+ = V \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & detV \end{bmatrix} \tag{2.71}$$

Finally, the matrix A (Direction Cosine Matrix) can be found using the equation:

$$A = U_+ V_+^T \tag{2.72}$$

Using Wahba's solution, to find the attitude of a spacecraft there are several methods such as least squares method, TRIAD, and QUEST (Quaternion Estimation), q-davenport, SVD, ESOQ (Estimator of the Optimal Quaternion), FOAM (Fast Optimal Attitude Matrix) and AIM (Attitude Estimation using Image Matching) algorithms. The computation of attitude with the method of least squares is very expensive [31]. QUEST and TRIAD algorithms need much less computational efforts. TRIAD algorithm generates Direction Cosine Matrix [37]. This algorithm is a deterministic solution to define attitude. DCM calculation needs two observed unit vectors and two reference unit vectors. Another method to solve the Wahba problem is QUEST algorithm. This algorithm gives quaternion output [38]. TRIAD algorithm is nearly two times faster than QUEST method [29]. The most robust predictors to solve Wahba's problem are Davenport's Q method and Single Value Decomposition (SVD) method. FOAM method requires measurement of at least two vectors to determine the DCM. Its output is a quaternion. All deterministic methods, such as TRIAD, QUEST, and FOAM can fail with only one observed vector. ESOQ method is fast but it is not as much accurate as q-Method and SVD method. A summary of different attitude estimation algorithms is given regarding accuracy, speed and output format in the Table 2.9 [39].

Because of the high accuracy of the Singular Value Decomposition (SVD) method

Table 2.9: Evaluation of different attitude determination algorithms in terms of accuracy, speed, and output format

Algorithm	Accuracy	Speed	Output Format
TRIAD	Low	Very high	Rotation Matrix
q-Method	High	Low	Quaternion
SVD	High	Low	Rotation Matrix
QUEST	Low	High	Quaternion
ESOQ / ESOQ 1 / ESOQ 2	Low	High	Quaternion

and the high-speed operation of the Triad and QUEST methods, these methods were given priority. The speed and predicted accuracy of these algorithms have been analyzed. The QUEST, ESOQ, and q-method are based on the problem of finding the greatest eigenvalue. Each of these methods uses different approaches to solve this problem. Through these different approaches, speed or accuracy is increased. Numerical methods used in the QUEST and ESOQ methods accelerate the process but cause the accuracy to decrease. For this reason, numerical methods have not been studied. On the other hand, studies on the Triad method have been carried out because it has the highest speed. Thus, highest speed and highest accuracy algorithms are compared.

With the QUEST method, orientation calculations are made taking into account all the sources (measurements) on the image. This method is more likely to have higher accuracy than the Triad method because it scans all sources in the image. It is expected that the weight of each measurement will be entered as an input since there is a possibility of giving erroneous results. Since this method scans all resources, the time to find orientation is longer than the Triad. Moreover, it can be said that the orientation finding time has changed depending on the number of sources in the image. The Triad method calculates the orientation using any two of the sources in the image. For this reason, the accuracy of selected sources directly affects the accuracy of orientation. The correctness of the method will not be too low as it is not likely to give false-positive results in the quad method. In terms of the speed, it is expected to work faster than QUEST. The SVD (Singular Value Decomposition) method also

takes into account the n-measurements. According to the approach to solve Wahba problem, SVD method is quite accurate than Quest and Triad methods. However, the speed is lower compared to the other methods.

By using a camera which has 2048 x 2048 pixels and visible and near-infrared CMOS sensor, two images have been taken from night time on the ground, then the accuracy comparisons of attitude determination algorithms have been made. The measurement errors that may occur in the images are defined as σ_{tot} parameter of 0.01°. In all algorithms, the same measurement error was used. The specified measurement error parameter will be used to estimate the accuracy of the orientation results determined by the attitude determination algorithms. It will be necessary to identify and model all sources of error so that the measurement error can be determined more clearly. Since this parameter cannot be modeled realistically, angle errors will not reflect the truth perfectly but will allow the comparison of the attitude determination algorithms relatively.

2.3.7 QUEST Error Estimation

$$B_{Quest} = \sum_{i=1}^{n} a_i \widehat{W}_i \widehat{V}_i^T \tag{2.73}$$

$$\sigma_{tot} = 0.01^{\circ} \tag{2.74}$$

$$P_{qq} = 0.25 * \sigma_{tot}^2 * (I - B)^{-1} \tag{2.75}$$

$$P_{\theta\theta} = 4 * P_{qq} \tag{2.76}$$

The square root of the diagonal elements' error values of the above-computed error covariance matrix yield Euler angles' error values $(x, y, z - 1\sigma error)$ [40].

2.3.8 SVD Error Estimation

$$s_1 = S(1,1) \tag{2.77}$$

$$s_2 = S(2,2) \tag{2.78}$$

$$s_3 = |U||V|S(3,3) \tag{2.79}$$

$$A_1 = \frac{1}{s_2 + s_3} \tag{2.80}$$

$$A_2 = \frac{1}{s_1 + s_3} \tag{2.81}$$

$$A_3 = \frac{1}{s_1 + s_2} \tag{2.82}$$

$$\begin{bmatrix} A_1 & 0 & 0 \\ 0 & A_2 & 0 \\ 0 & 0 & A_3 \end{bmatrix} \tag{2.83}$$

$$P = \sigma_{tot}^2 * U * A * U^T \tag{2.84}$$

The square root of the diagonal elements' error values of the above-computed error covariance matrix yield Euler angles' error values $(x, y, z - 1\sigma error)$ [41].

2.3.9 TRIAD Error Estimation

The Triad method involves two random measurements on the image.

$$B_{Triad} = \sum_{i=1}^{n} a_i \widehat{W}_i \widehat{V}_i^T \tag{2.85}$$

$$P_{qq} = 0.25 * \sigma_{tot}^2 * (I - B)^{-1} \tag{2.86}$$

$$P_{\theta\theta} = 4 * P_{qq} \tag{2.87}$$

The square root of the diagonal elements' error values of the above-computed error covariance matrix yield Euler angles' error values $(x, y, z - 1\sigma error)$ [40].

The two images taken by using the camera mentioned above are identified with the quad algorithm. The Astrometry.net's quad method also includes PSF fitting algorithm to find centroid of the stars and then the star identification phase is completed. With the help of this method, the stars on the images are identified and this information is used for the attitude determination. To compare attitude results of these two images, QUEST, SVD and Triad methods are used. When we examine the sources on these 2 images, it is observed that there are more sources (stars) on the first image. For this reason, the processes performed in the first image have been observed to take longer.

When the Table 2.10 and Table 2.11 are examined, it is seen that the x-axis and y-axis errors are less than the z-axis error. Moreover, it can be said that SVD method is the most accurate algorithm compared to the others. On the other hand, the accuracy of the Triad method which uses only two measurements in the image is close to the QUEST because the quad method measurements are quite robust in general. If one or two of the measurements used in the Triad method is false-positive or low accuracy results, then the accuracy of the method would be the low.

Table 2.10: Comparison of the attitude determination algorithms for the first image

Methods	**Accuracy Levels**		
	x (1σ) **(arcsec)**	**y** (1σ) **(arcsec)**	**z** (1σ) **(arcsec)**
QUEST	30.96	30.91	60.27
SVD	6.59	6.59	52.53
TRIAD	36.00	36.82	62.12

Table 2.11: Comparison of the attitude determination algorithms for the second image

Methods	**Accuracy Levels**		
	x (1σ) **(arcsec)**	**y** (1σ) **(arcsec)**	**z** (1σ) **(arcsec)**
QUEST	30.93	31.12	61.49
SVD	6.71	6.73	50.31
TRIAD	36.00	37.60	63.00

CHAPTER 3

EXTENDED KALMAN FILTER IMPLEMENTATION OF STAR TRACKER AIDED INS

In this chapter, Extended Kalman Filter implementation of the star tracker aiding inertial navigation system application is analyzed. This application is applied on a high altitude aircraft because starlight can be seen as clear as possible. Star tracker sensor which observes starlight is affected by atmospheric effects such as refraction, light absorption, and light pollution etc. Therefore, aircraft altitude must be high in order to reduce atmospheric effects. In fact, there is a model to describe the atmospheric attenuation [42]. However, in this thesis, atmospheric effects are ignored. Furthermore, star tracker accuracy also depends on the angular velocity and vibration of the vehicle. However, if the camera has high dynamic range and exposure time is low enough under low vibration and low angular motion, there is little effect of blurring so it can be ignored. Dynamic range of the camera is related to the rate between read noise and saturation capacity. Read noise represents the minimum number of electrons that a pixel can read. Saturation capacity is the maximum number of electrons that a pixel can evaluate. Thus, if this range is high, low exposure time images can be qualified to star centroiding and extraction process. This low exposure time does not allow to be created the motion blur in the image. On the other hand, in the case of the high angular velocity and vibration, there is motion blur on the image. There are numerous filters to reduce the blurring effects such as Richardson-Lucy and Wiener filters etc. They calculate the motion blur PSF (Point Spread Function) and restore the image in the frequency domain. Wiener filter is a linear filter to restore the blurred image and it is modeled with Gaussian noise. On the other hand, Richardson-Lucy filter is a nonlinear filter. It is assumed that the star image has Poisson distributed noise in this method [43]. Yet, in this thesis, image blur effect is not calculated because this

effect comes from the camera parameters as well as the motion parameters. Because the camera parameters are not specified, the blur effect is not considered numerically. Although atmospheric effects and image blur effects are not calculated, their effects can be observed in aircraft applications. Therefore, the accuracy of a star tracker in an aircraft will be lower than the accuracy of the space counterpart. These accuracy values also depend on in-motion time dependent parameters such as temperature effect on optical system etc. and the other stochastic error parameters. The simulated images or real test images including all these error parameters are not included in this thesis. Thus, the star tracker accuracy levels in the simulation are determined according to these considerations.

X-15 aircraft is chosen as a high altitude aircraft. It is a three rocket-powered aircraft and reached the edge of space [44]. Inside the atmosphere, conventional aerodynamic controls are used and in the edge of space, hydrogen peroxide thrust rockets are used in the aircraft [45].

Figure 3.1: X-15 Aircraft

3.1 X-15 AIRCRAFT 6 DEGREES OF FREEDOM SIMULATION

In order to create flight trajectory, X-15 aircraft 6 degrees of freedom flight simulation model is used. This 6-DOF model is found from Simulink Airlib library [46]. It is shown in the Figure 3.2. Moreover, 6-DOF simulation model architecture is described in the Figure 3.3.

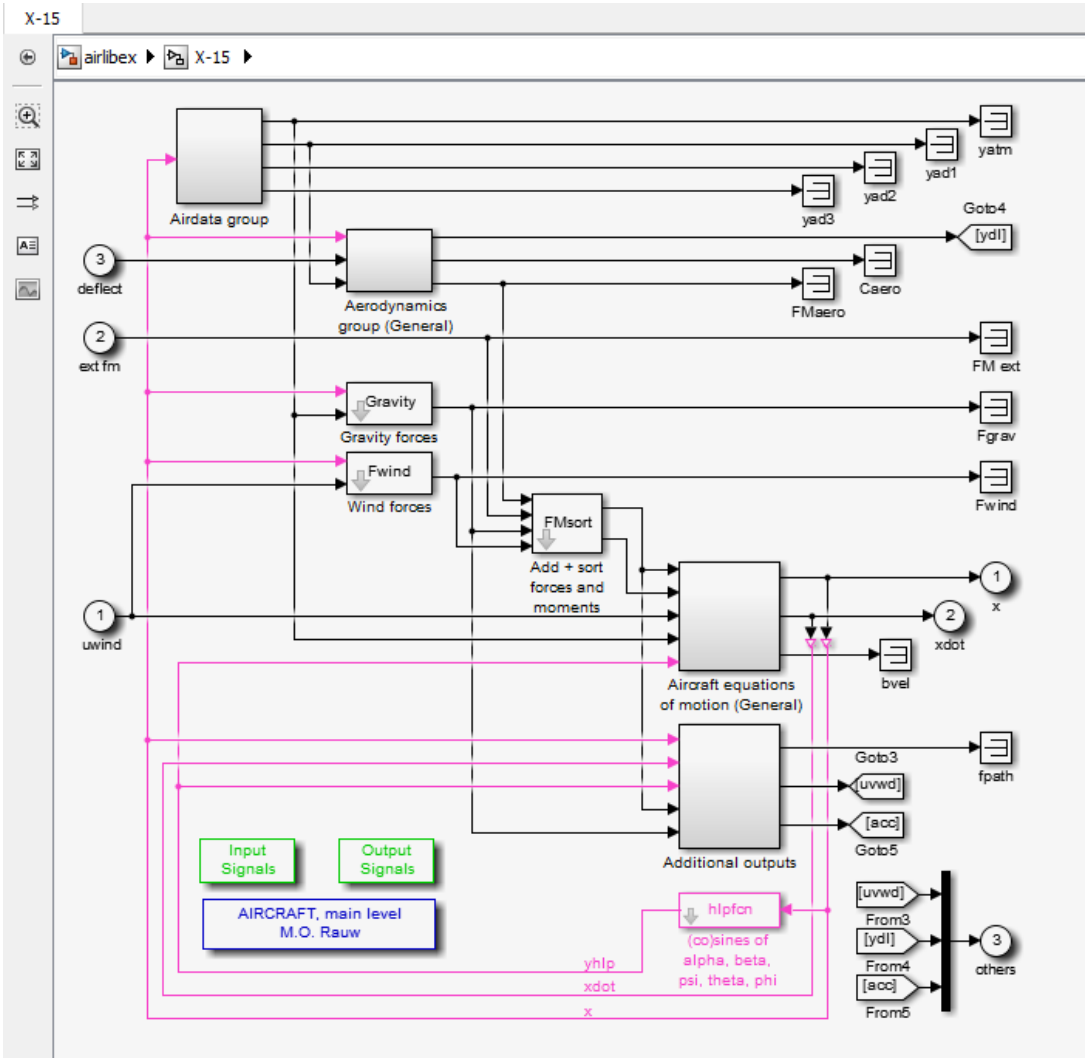


Figure 3.2: X-15 Aircraft 6-DOF Simulink Model

By applying external forces and moments in this simulation, aircraft flight trajectory and IMU sensor data are obtained. Moreover, when wind forces and deflection of control surfaces are requested, they are also fed into the aircraft simulation. To reduce the atmospheric effects, high altitude trajectory is created. As an output, position,

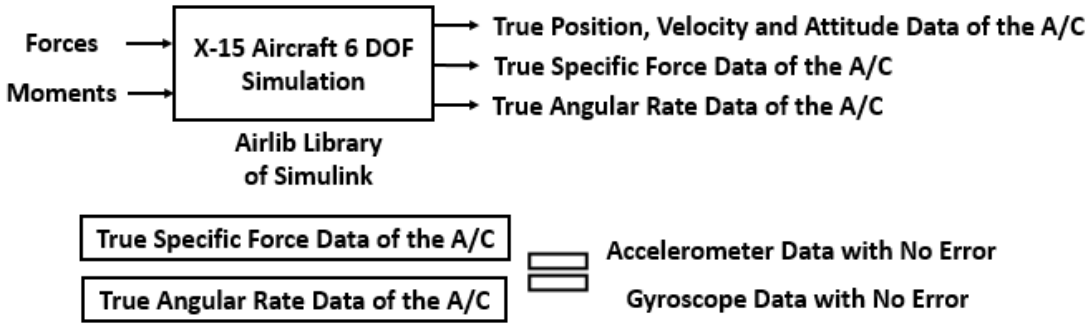


Figure 3.3: X-15 6 Degrees of Freedom Model Architecture

velocity and attitude in the navigation frame, angular rate, and specific force data of the aircraft in body frame can be obtained. From now on, position, velocity and attitude data obtained from 6 DOF flight simulation will be mentioned as true flight trajectory data and angular rate and specific force data from this simulation will be mentioned as IMU data with no error.

After obtaining flight data, Extended Kalman filter implementation of star tracker aiding INS application is created by using Simulink environment. This simulation is composed of inertial navigation system solution (INS mechanization) and Extended Kalman Filter solution. In this way, only INS solution and star tracker aided INS solution is compared.

When these two results are compared, it can be seen that filter makes improvements in the INS solution provided that Kalman filter tuning is made. This improvement level is determined by INS grade and star tracker accuracy level. Yet, it is found that in low specific force flight, this correction is not observed. Because inertial navigation solution is made in the local navigation frame, specific force transformation is made by multiplying accelerometer data with direction cosine matrix which converts body frame to navigation frame. The filter makes attitude correction and new direction cosine matrix is calculated from the corrected Euler angles. This new DCM is multiplied with the specific force for velocity correction. To correct velocity in the navigation frame, specific force is transformed in navigation frame and added to velocity correction equation as an additional element. Therefore, in low specific force flight, inertial navigation solution makes already small error and Kalman filter solution does not improve INS solution so much. Because the position correction is

calculated from the velocity correction, position correction is also small. Inertial navigation equations resolved in local navigation frame is shown in Equations from 3.1 to 3.6.

Position Update:

$$h_b(+) = h_b(-) - \frac{\tau_i}{2}\big(v^n_{eb,D}(-) + v^n_{eb,D}(+)\big) \tag{3.1}$$

$$L_b(+) = L_b(-) + \frac{\tau_i}{2}\big(\frac{v^n_{eb,N}(-)}{R_N(L_b(-)) + h_b(-)} + \frac{v^n_{eb,N}(+)}{R_N(L_b(+)) + h_b(+)}\big) \tag{3.2}$$

$$\lambda_b(+) = \lambda_b(-) + \frac{\tau_i}{2}\big(\frac{v^n_{eb,E}(-)}{(R_E(L_b(-)) + h_b(-)) * cosL_b(-)} + \frac{v^n_{eb,E}(+)}{(R_E(L_b(+)) + h_b(+)) * cosL_b(+)}\big) \tag{3.3}$$

Velocity Update:

$$V^n_{eb}(+) = V^n_{eb}(-) + \big(f^n_{ib} + g^n_b(L_b(-), h_b(-)) - (\Omega^n_{en}(-) + 2\Omega^n_{ie}(-))V^n_{eb}(-)\big)\tau_i \tag{3.4}$$

Specific Force Transformation:

$$f^n_{ib}(t) = C^n_b(t) * f^b_{ib}(t) \tag{3.5}$$

Attitude Update:

$$C^n_b(+) = C^n_b(-)(I_3 + \Omega^b_{ib}\tau_i) - (\Omega^n_{ie}(-) + \Omega^n_{en}(-))C^n_b\tau_i \tag{3.6}$$

Star tracker attitude information accuracy is determined according to the flight conditions. Although there are few atmospheric effects, it has some effect on the star tracker. Moreover, according to the trajectory, the aircraft has high angular velocity. Therefore, the star tracker does not have as high accuracy as recent star trackers used in the satellites. Given these circumstances, two different star tracker accuracies are determined. Star tracker accuracy levels are shown in the Table 3.1. Moreover,

two different tactical grade INS accuracy levels are used for the simulation. These accuracy levels are shown in the Table 3.2.

Table 3.1: Star Tracker Accuracy Levels

Sensor	**Phi Error** (1σ)	**Theta Error** (1σ)	**Psi Error** (1σ)
Star Tracker-1	10	10	15
Star Tracker-2	15	15	25

Table 3.2: INS Accuracy Levels

Sensor	**Acc. Bias** (mg)	**Gyro Bias** (deg/hr)	**VRW** $(m/s/\sqrt{hr})$	**ARW** $(deg/\sqrt{hr})$
INS-1	0.3	0.8	0.007	0.07
INS-2	0.5	1	0.01	0.125

3.2 INERTIAL NAVIGATION AND KALMAN FILTER SIMULATION

In this simulation, Discrete Kalman Filter is used. Because the navigation equations are nonlinear, these equations are linearized. Because of system nonlinearity, Extended Kalman Filter is used. In the filter, error dynamics are selected to find the aircraft system parameters.

Firstly, IMU data with error is created according to the parameters in the Table 3.2. Error model is based on scale factor, misalignment and bias of sensors. Furthermore, according to the angular random walk and velocity random walk parameters, white Gaussian noise is added to the IMU data. On the other hand, star tracker data is fed into the filter as a measurement. This data is obtained from aircraft 6-DOF model attitude information. Star tracker error model is based on the Euler angles' standard deviation information. It is shown in the Table 3.1. Star tracker data with error are created by adding white Gaussian noise according to the predefined standard deviation. Finally, estimated position, velocity and attitude information of the aircraft in navigation frame is compared true system parameters obtained from the 6-DOF model.

Inertial Navigation System and Star Tracker integration architecture are shown in the Figure 3.4 [32] :

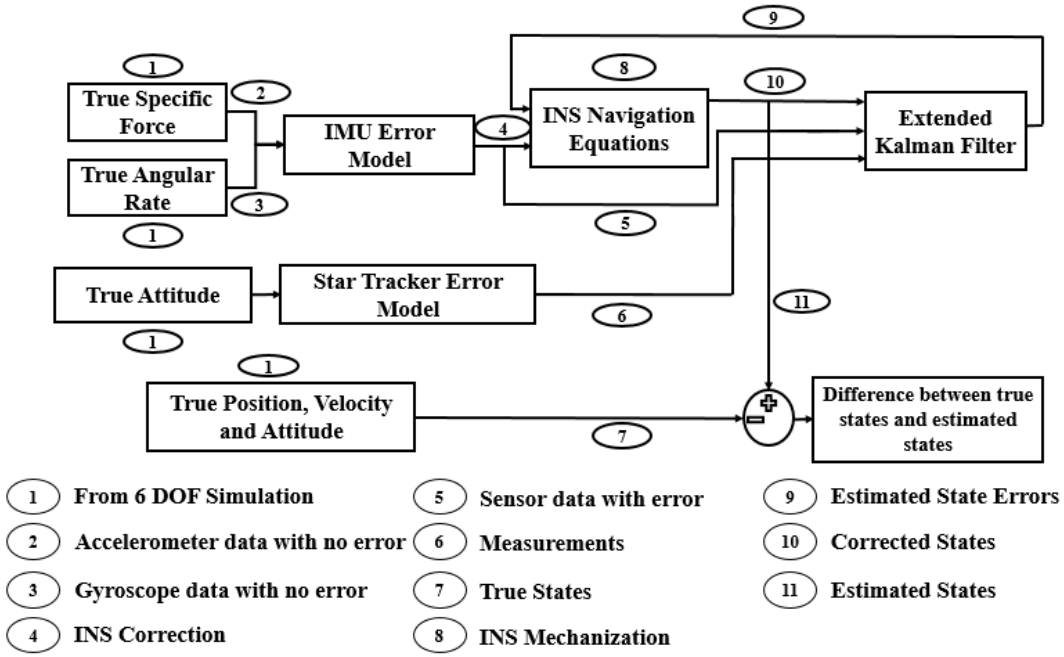


Figure 3.4: INS/Star Tracker Integration Architecture

3.3 KALMAN FILTER EQUATIONS

Discrete Kalman Filter equations are schematized in the Figure 3.5. In this filter, initial state estimates and covariance matrices are fed into the INS mechanization and Kalman Filter model.

Due to the linearization of F and H matrices, Extended Kalman Filter may be affected P matrix initialization and this can cause stability issues. Therefore, P matrix tuning is made [47].

Because the error dynamics are used in the Kalman Filter, measurement error is fed into the filter. Position and velocity measurement can be fed into the filter after the subtraction from INS solution as in GNSS. However, attitude measurement cannot be fed into the filter directly. It is fed into the system after the calculation of error of Euler angles shown in the Figure 3.6 [4].

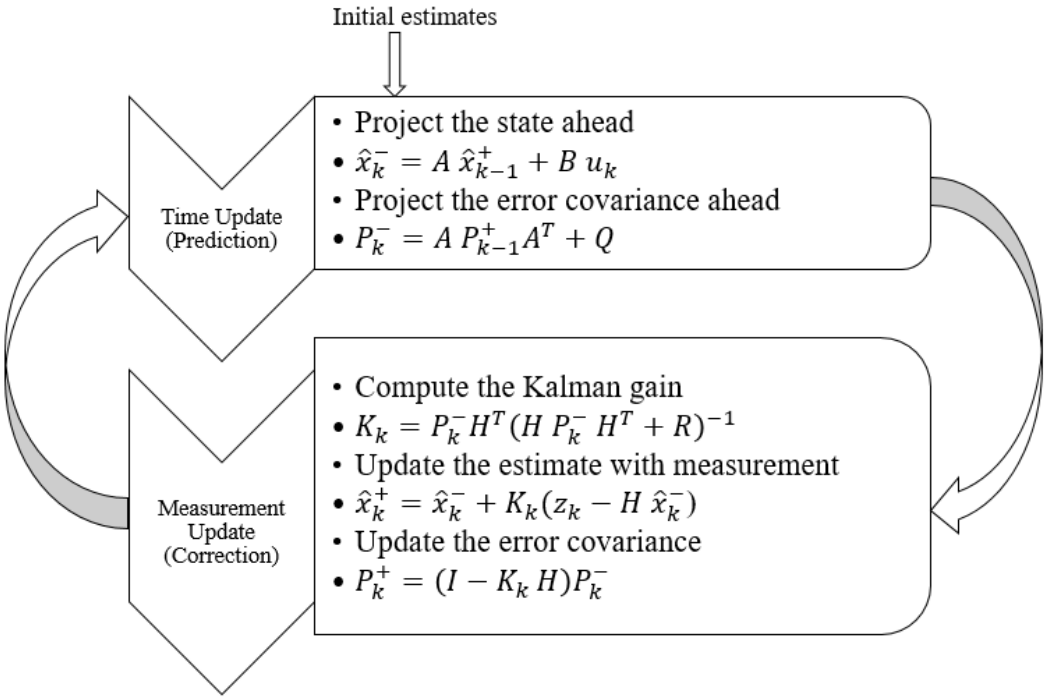


Figure 3.5: Discrete Kalman Filter Equations

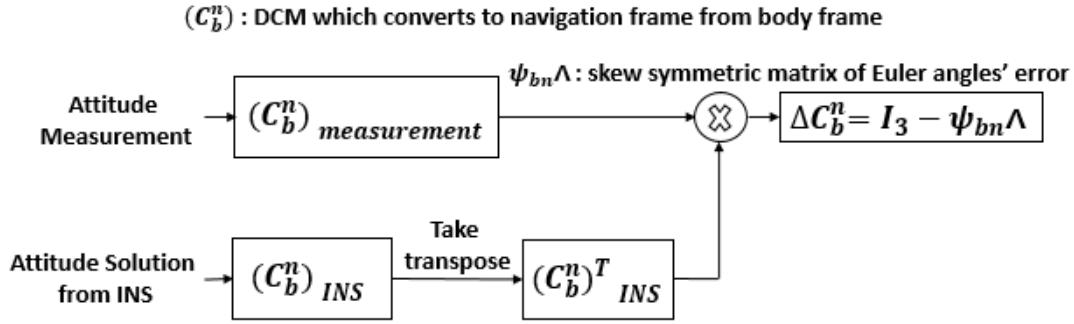


Figure 3.6: Measurement Update Architecture

$$\left(C_b^n\right)_{measurement}\left(\widetilde{C_b^n}\right)_{INS} = \dot{C}_b^n \tag{3.7}$$

$$\dot{C}_b^n = I_{3x3} - \psi_{bn}\Lambda \tag{3.8}$$

$\left(C_b^n\right)_{measurement}$: from body frame to navigation frame DCM which is calculated from star tracker measurements

$\left(\widetilde{C_b^n}\right)_{INS}$: from body frame to navigation frame DCM which is calculated from

inertial navigation equations

High and low specific force data are created to compare Kalman filter performance. Then, two different IMU data with error is created. Moreover, from true PVA data, two different star tracker data with predefined error are created. Nine states (position, velocity, and attitude) of the aircraft in navigation frame and their differences with the true states are found by using 8 different cases shown in the Table 3.3.

Table 3.3: Case Definitions

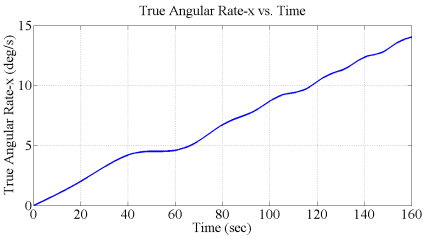
Case-1	INS-1 only solution (1000 Hz)
Case-2	INS-2 only solution (1000 Hz)
Case-3	INS-1 (1000 Hz) + Star Tracker-1 (1 Hz) KF solution
Case-4	INS-2 (1000 Hz) + Star Tracker-1 (1 Hz) KF solution
Case-5	INS-1 (1000 Hz) + Star Tracker-2 (1 Hz) KF solution
Case-6	INS-2 (1000 Hz) + Star Tracker-2 (1 Hz) KF solution
Case-7	INS-1 (1000 Hz) + Star Tracker-1 (10 Hz) KF solution
Case-8	INS-2 (1000 Hz) + Star Tracker-2 (10 Hz) KF solution

3.4 SIMULATION AND RESULTS

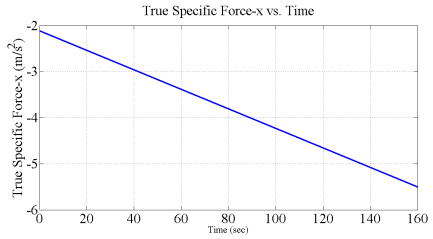
Kalman Filter simulation is made in the Simulink environment. It lasts for 160 seconds. INS mechanization is made by time steps of 0.001 seconds. The attitude measurement update is made at 1 Hz or 10 Hz.

First of all, high specific force flight data is created. With this data, it is observed that INS results are improved by the filter. True angular rate and specific force data are obtained from X-15 aircraft six degrees of freedom Simulink model. This case represents high specific force data. These IMU data with no error are shown in the Figure 3.7. The first three is the gyroscope data with no error, the last three is the accelerometer data with no error.

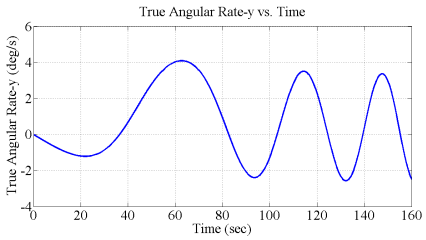
In the low specific force flight, there is little error in INS mechanization results in the navigation frame. Because only the attitude measurement is fed into the filter, only



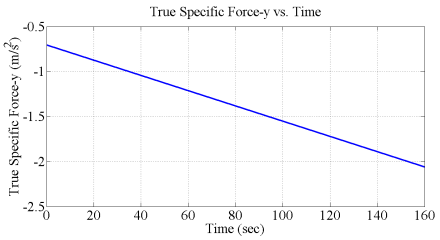
(a) True Angular Rate-x for High Specific Force Flight



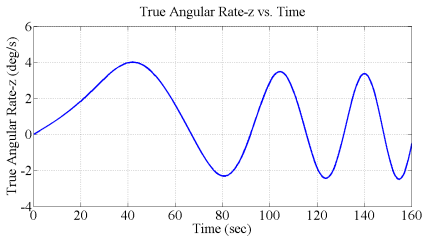
(b) True Specific Force-x for High Specific Force Flight



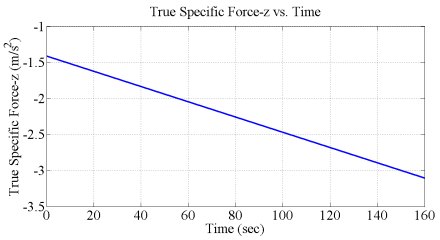
(c) True Angular Rate-y for High Specific Force Flight



(d) True Specific Force-y for High Specific Force Flight



(e) True Angular Rate-z for High Specific Force Flight



(f) True Specific Force-z for High Specific Force Flight

Figure 3.7: True Angular Rate and Specific Force Results for High Specific Force Flight

attitude information is corrected. There are very small corrections in velocity and position states. In low specific force flight, star tracker is important for only attitude correction. However, in high specific force flights, star tracker is crucial for all system states.

Finally, position, velocity and attitude difference between true system states and estimated states are schematized. Aircraft true system states are obtained from 6 degrees of freedom aircraft model. Estimated states are obtained by using different IMU and

star tracker sensor data shown in the Table 3.3 and Simulink model which includes INS mechanization and EKF model. This model is created by using the calculations included in the Figure 3.4, Figure 3.5 and Figure 3.6. Note that, the position error is defined in terms of cartesian coordinates. This is achieved by knowing latitude and longitude values defined in navigation frame.

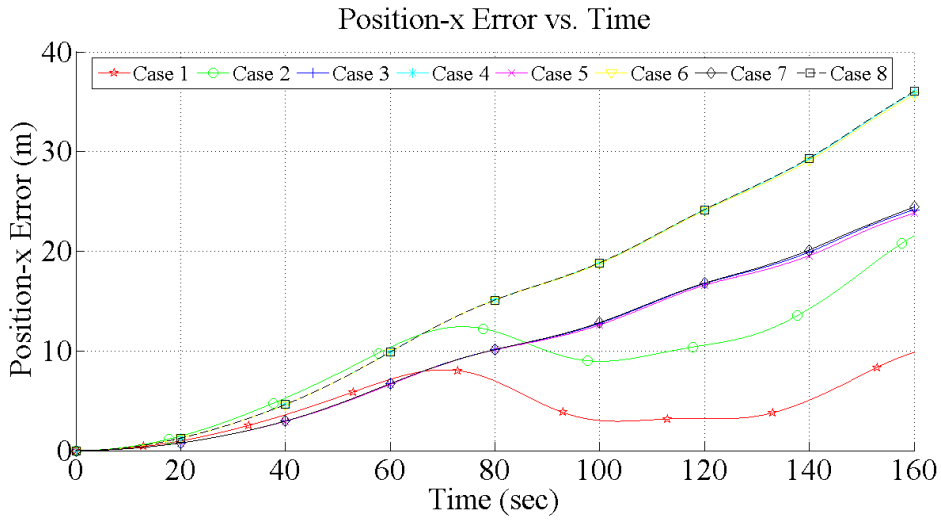


Figure 3.8: Position-x Error for 8 Different Cases for High Specific Force

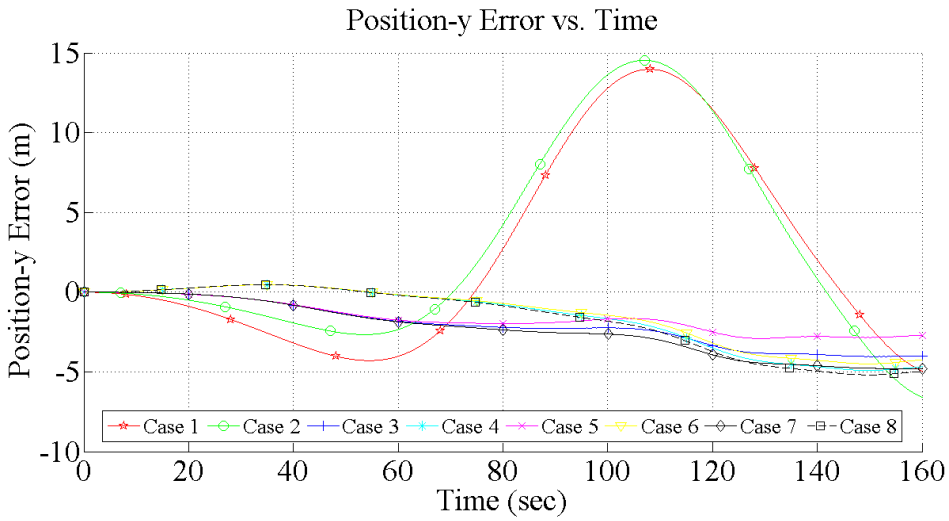


Figure 3.9: Position-y Error for 8 Different Cases for High Specific Force

Because INS-1 is better than INS-2, INS-1 mechanization gives more accurate state results on its own. However, as shown in the difference figures, worst cases are the INS only solutions as compared to the INS/Star Tracker integration complements.

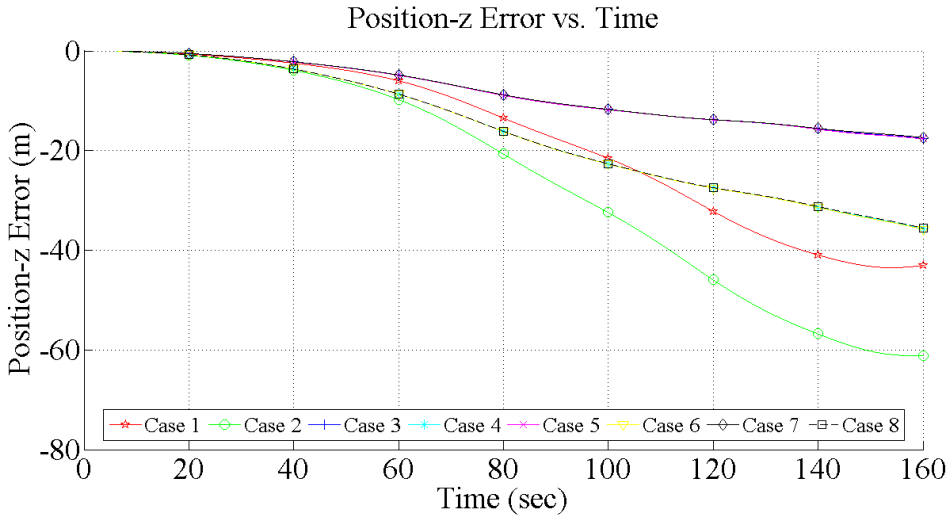


Figure 3.10: Position-z Error for 8 Different Cases for High Specific Force

Case 3, 4, 5 and 6 represent different error levels of sensors. It is shown that how much the different error levels affect the accuracy of the system states. In general, it can be said that Case 6 and Case 8 are the worst cases among the INS/Star Tracker integration implementations because INS and Star Tracker accuracy levels are low compared to the others. On the other hand, Case 3 and Case 7 are the best cases because the sensor accuracy levels are the highest ones. Case 4 is better than Case 5 for attitude states because star tracker accuracy level is higher. Despite the INS-2 accuracy level is lower than the INS-1 level, attitude measurement has more impact on Euler angles. On the other hand, Case 5 is a little bit better than Case 4 for position and velocity states because in this case, INS accuracy level is a little more important than star tracker measurement accuracy level.

Prior to Case 7 and 8, it is assumed that star tracker sensor output frequency is 1 Hz. However, at Case 7 and 8, sensor output frequency is assumed as 10 Hz. It is found that while data sampling frequency is increased, attitude solution is improved which is, in turn, affects the solution of the velocity and position states in a positive manner although it has little effect on velocity and position states. When the measurement data rate is increased, Kalman filter output frequency is increased. Therefore, attitude correction is made more often and this makes estimated attitude more accurate. While the data output frequency has a direct impact on attitude solution, it has an indirect effect on position and velocity solution. These effects can be seen from the Figure

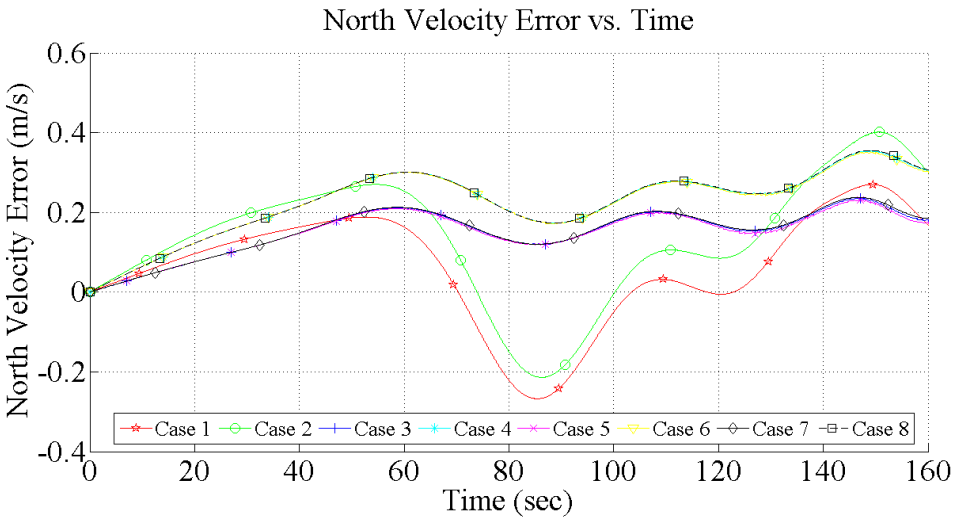


Figure 3.11: North Velocity Error for 8 Different Cases for High Specific Force

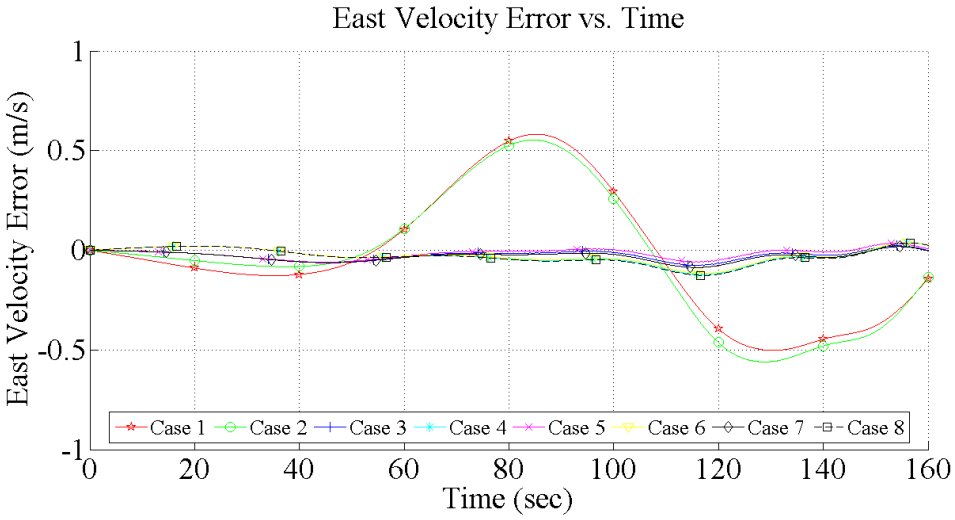


Figure 3.12: East Velocity Error for 8 Different Cases for High Specific Force

3.8 to Figure 3.16.

This simulation also lasts 160 seconds. INS mechanization is made by time steps of 0.001 seconds. The attitude measurement update is made at 1 Hz or 10 Hz. Case 8 and Case 9 represent 10 Hz measurement update cases.

In low specific force flight, only INS solution is enough for position and velocity solutions and it cannot be seen that the filter does not make a meaningful contribution. However, for attitude solution, it is observed that filter makes corrections by using star

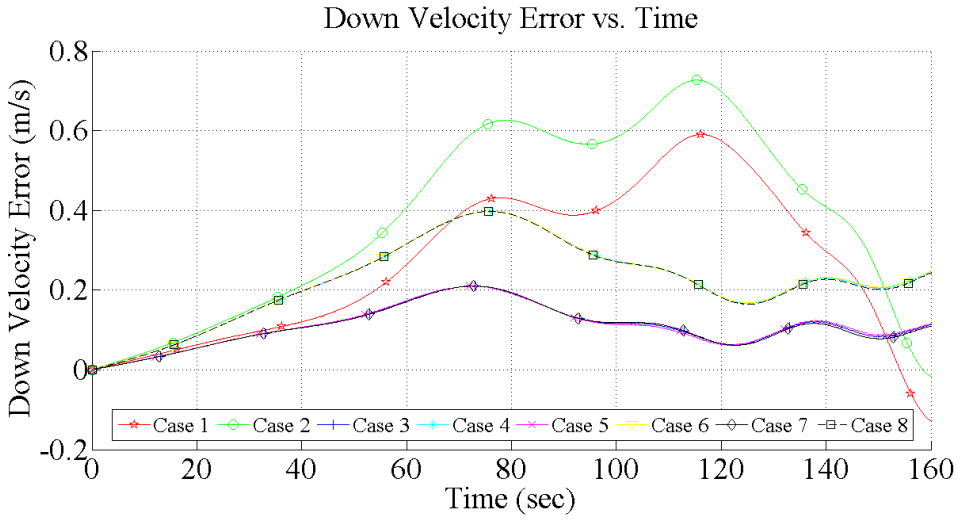


Figure 3.13: Down Velocity Error for 8 Different Cases for High Specific Force

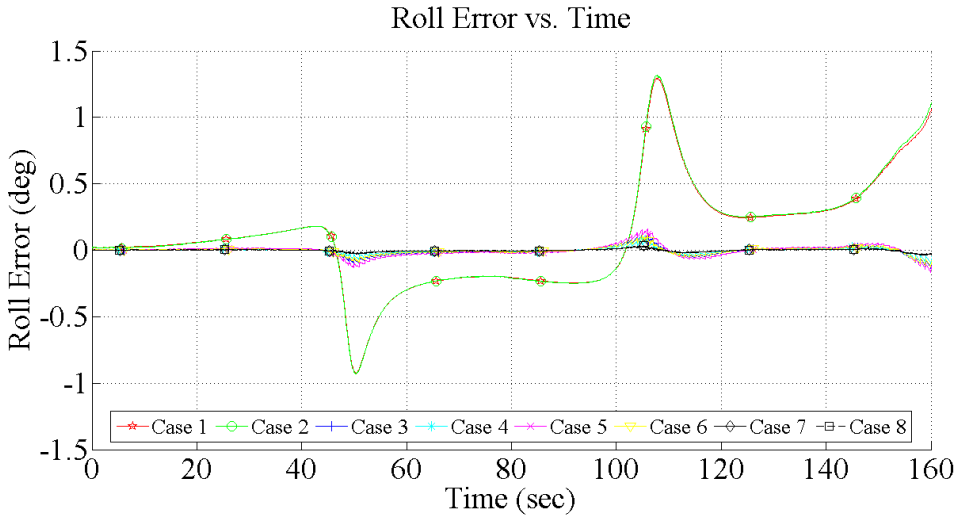


Figure 3.14: Phi Error for 8 Different Cases for High Specific Force

tracker measurements.

In this part, true angular rate and specific force data are also obtained from X-15 aircraft 6-DOF model. This case represents low specific force data. These IMU data with no error are shown in the Figure 3.17. The first three is the gyroscope data with no error, the last three is the accelerometer data with no error.

In low specific force flight, only INS solution and star tracker aiding INS solution for position and velocity states are quite same. INS-1 accuracy level is better than the

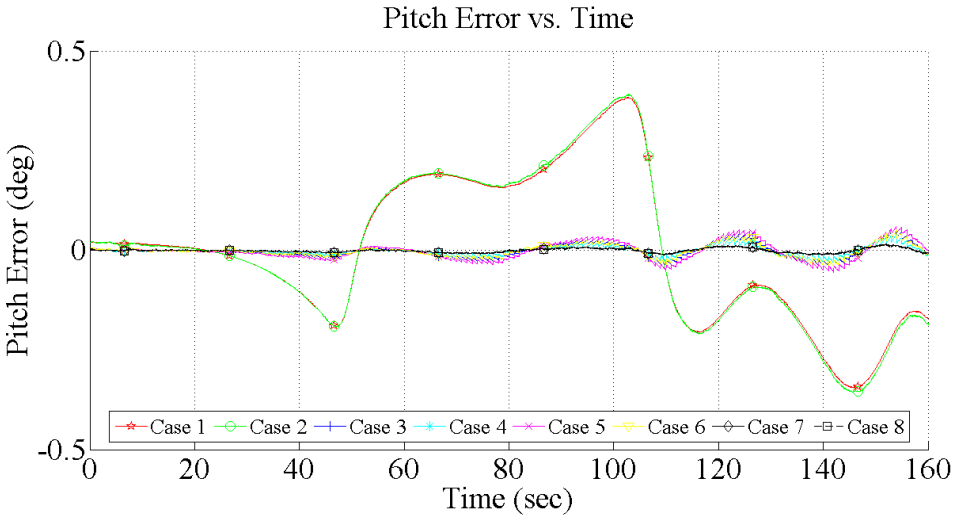


Figure 3.15: Theta Error for 8 Different Cases for High Specific Force

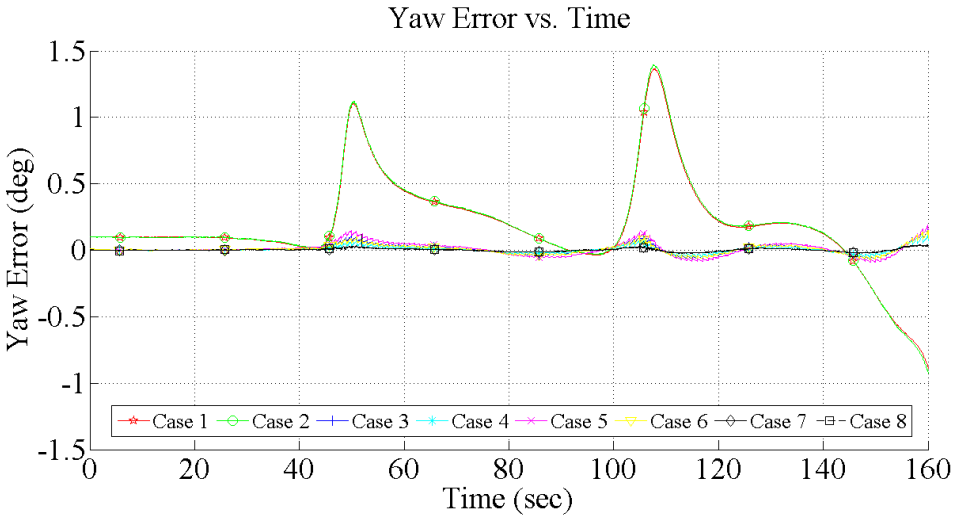
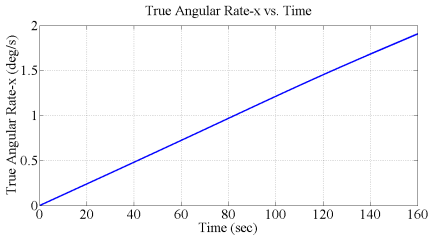


Figure 3.16: Psi Error for 8 Different Cases for High Specific Force

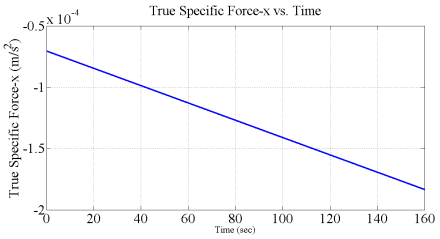
INS-2 accuracy level, so INS-1 only solution is better than the INS-2 only solution.

Case 1, Case 3, Case 5 and Case 7 have INS-1 sensor. The differences between these cases are star tracker accuracy levels and measurement update rates. These cases have pretty same accuracy in position and velocity states.

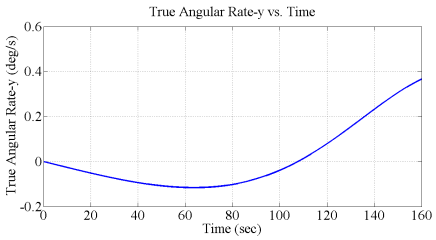
Case 2, Case 4, Case 6 and Case 8 have INS-2 sensor. The differences between these cases are also star tracker accuracy levels and measurement update rates. For position and velocity states of the aircraft is nearly same for these cases.



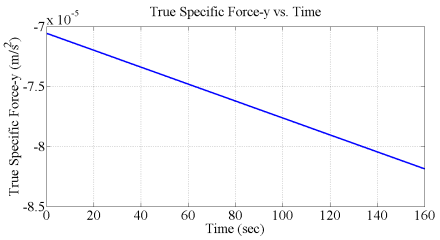
(a) True Angular Rate-x for Low Specific Force Flight



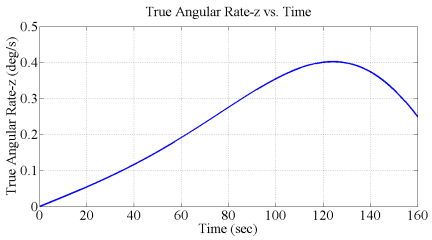
(b) True Specific Force-x for Low Specific Force Flight



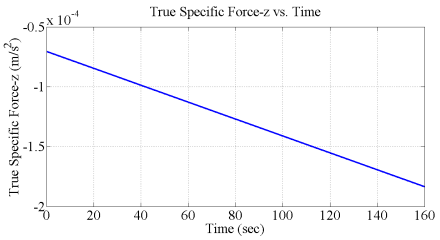
(c) True Angular Rate-y for Low Specific Force Flight



(d) True Specific Force-y for Low Specific Force Flight



(e) True Angular Rate-z for Low Specific Force Flight



(f) True Specific Force-z for Low Specific Force Flight

Figure 3.17: True Angular Rate and Specific Force Results for Low Specific Force Flight

For the attitude states of the aircraft, it is observed that filter makes a correction as in the high specific force case. INS-1 and INS-2 only cases are the worst cases and star tracker aiding INS integration cases are better than the INS only cases according to the star tracker accuracy levels and measurement update rate.

At an aircraft which can make high altitude and high specific force flight, star tracker sensor would be beneficial. Because star trackers provide highly accurate orientation information, when they are used to support inertial navigation system (INS),

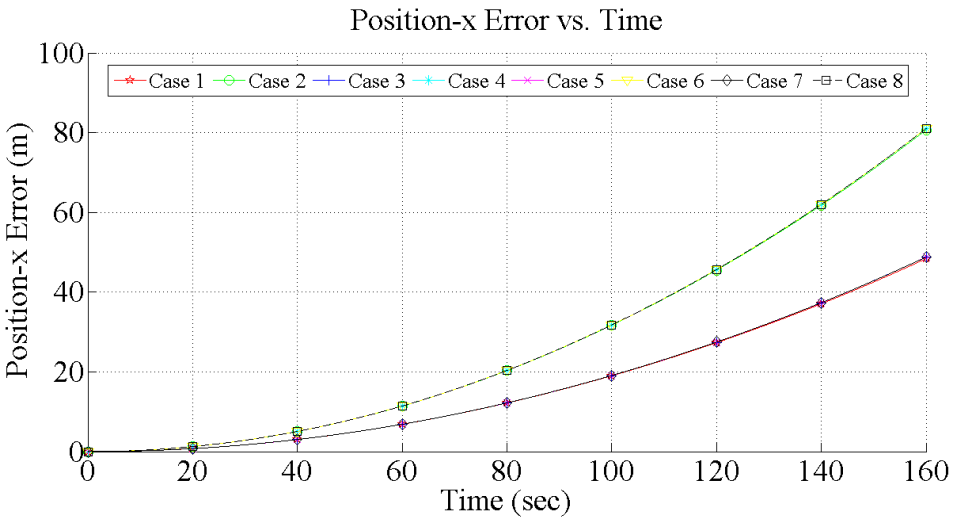


Figure 3.18: Position-x Error for 8 Different Cases for Low Specific Force

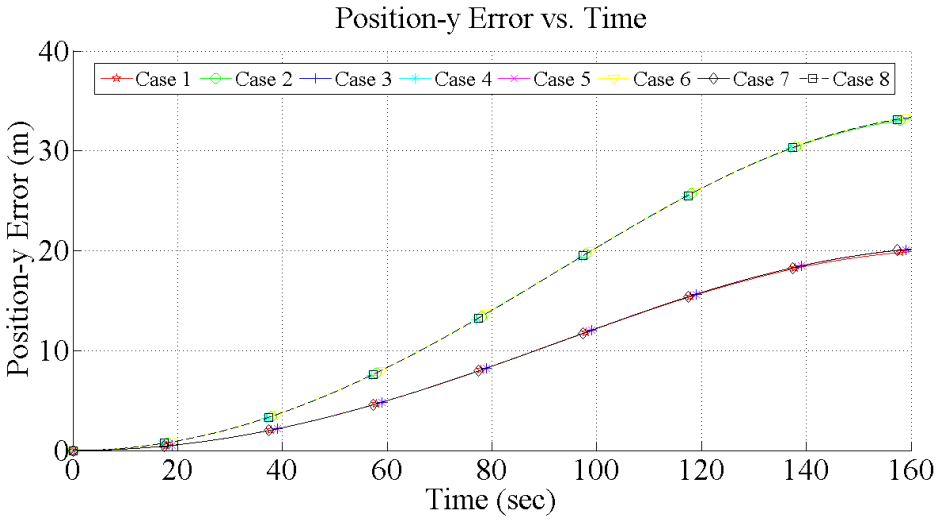


Figure 3.19: Position-y Error for 8 Different Cases for Low Specific Force

the navigation solution improves further in the aircraft which has this type of flight. Therefore, it can be said that star tracker is an alternative INS aiding sensor if there is no GNSS support.

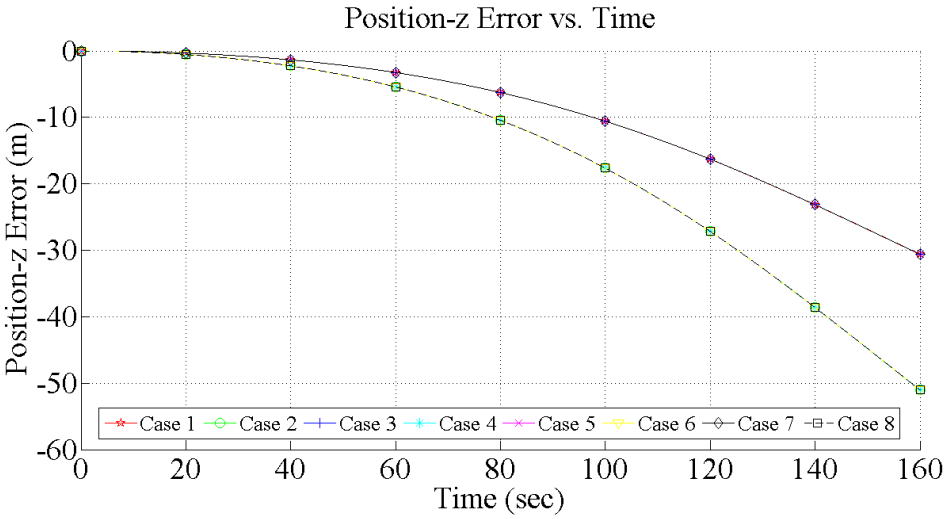


Figure 3.20: Position-z Error for 8 Different Cases for Low Specific Force

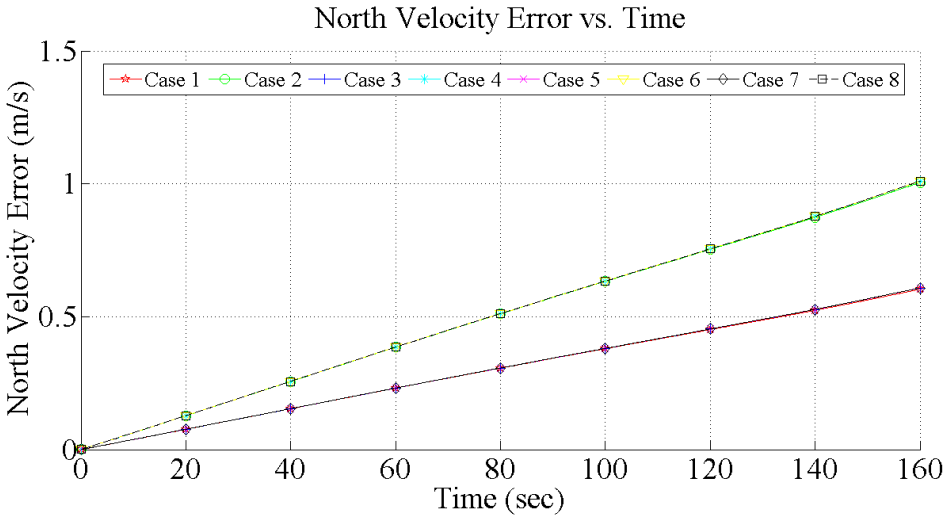


Figure 3.21: North Error for 8 Different Cases for Low Specific Force

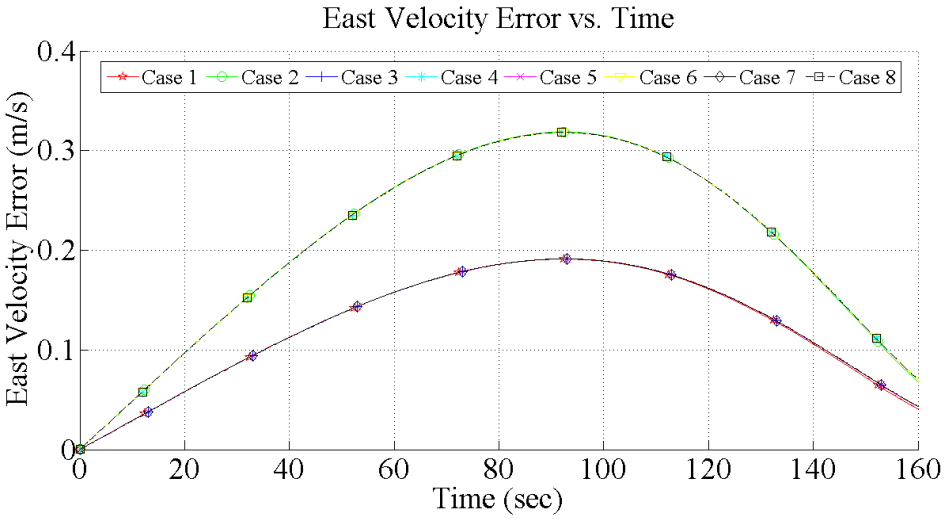


Figure 3.22: East Velocity Error 8 Different Cases for Low Specific Force

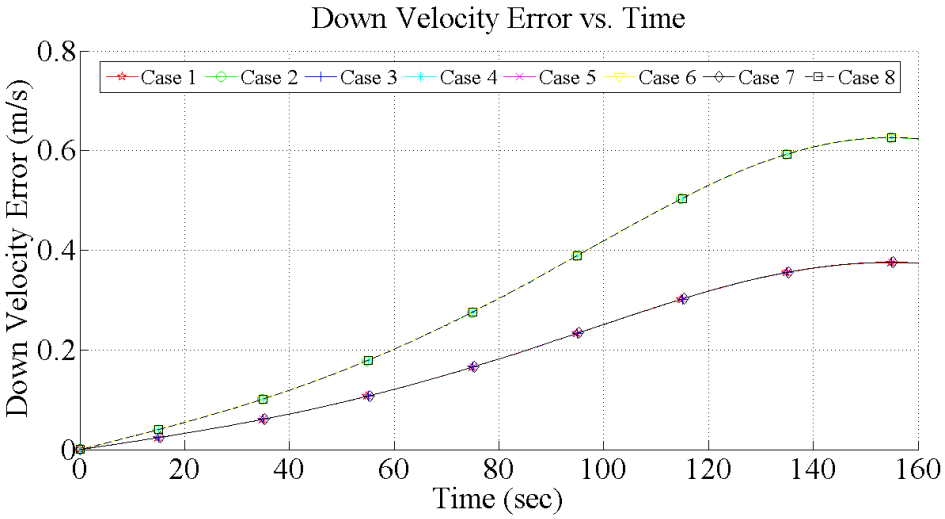


Figure 3.23: Down Velocity Error for 8 Different Cases for Low Specific Force

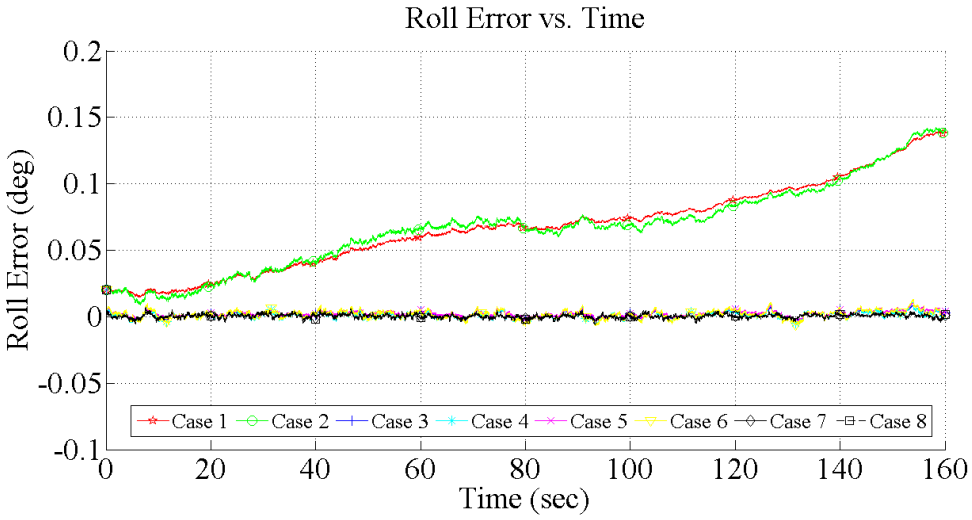


Figure 3.24: Phi Error for 8 Different Cases for Low Specific Force

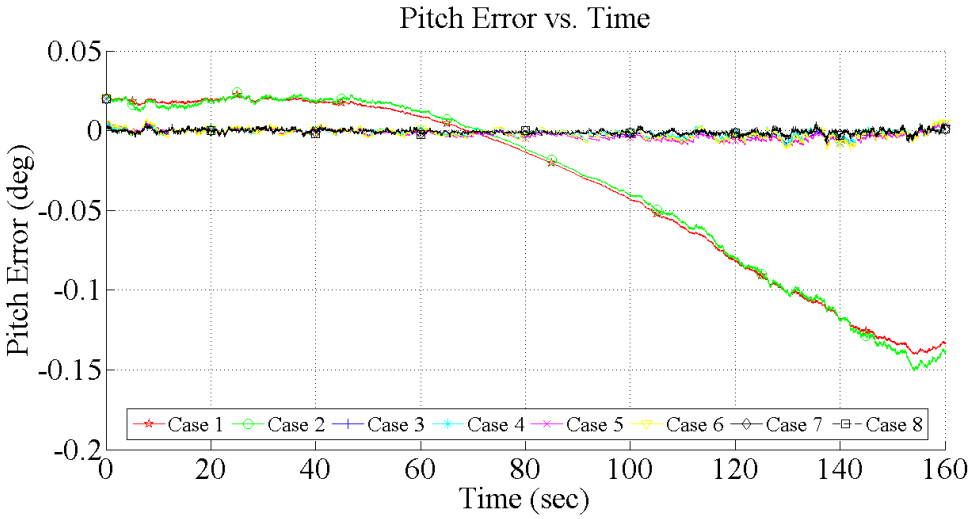


Figure 3.25: Theta Error for 8 Different Cases for Low Specific Force

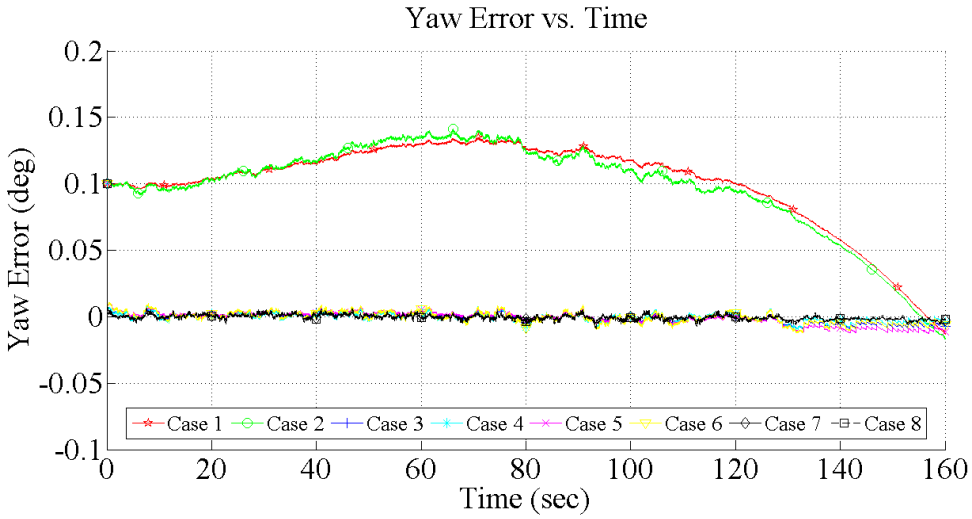


Figure 3.26: Psi Error for 8 Different Cases for Low Specific Force

CHAPTER 4

DISCUSSIONS AND CONCLUSIONS

In this thesis, first of all, the importance of the star tracker sensor is discussed because it gives accurate attitude information. Moreover, the areas of use of the sensor are explained. These sensors have widespread use in space applications, especially in satellites. In this area, this sensor can be used either alone or with other attitude determination sensors like sun sensor, earth sensor vs. High altitude vehicles/systems are another application area for this sensor. Furthermore, this sensor can be an aiding sensor for INS. Then, in the second chapter, star tracker algorithms are described. They are split into three main parts: centroiding, star matching and attitude determination algorithms. Initially, literature survey is given about each of them and then the chosen methods are discussed. Their results are compared in terms of speed, accuracy and robustness issues. Finally, in the last chapter, for X-15 aircraft which is a high altitude aircraft, Extended Kalman Filter implementation of star tracker aided INS is discussed.

First of all, star tracker algorithms are investigated in this thesis. Image moment analysis method and Point Spread Function (PSF) fitting method are analyzed from centroiding algorithms. It is seen that PSF fitting method is more robust than image moment analysis because while in the image moment analysis, non-star objects cannot be recognized, in the PSF fitting analysis method, non-star objects are differentiated. PSF fitting method recognize non-star objects because star and non-star object profiles are different. While star profiles are often like Gaussian distribution, profiles of the non-star objects (cosmic rays, signal spikes vs.) are not like Gaussian. For example, image moment analysis can find at least 2 or 3 times more stars for an image that contains above 50 stars. However, all of those sources may not be stars,

these might be cosmic rays or signal spikes etc. Moreover, for such an image, the process of finding star centroids with the help of PSF fitting analysis is at least 4 or 5 times faster than the image moment analysis because it finds more sources in the image and calculation of star centroids needs more computational time.

Then, literature survey about the star matching algorithms is investigated. After that, triangle method and quad method are discussed in detailed. It is observed that quad method is more robust than the triangle method because its features (hash codes) are unique. Therefore, it does not give false positives but it can give no solution. In the study to evaluate these methods, it can be seen that triangle method solves 97% of the synthetic images. On the other hand, quad method solves 96% of these images for 10° FOV. Moreover, for 15° FOV, while triangle method gives 98% output, quad method gives 97% output. It is observed that for wider field of view, methods can give much more results because there more stars in the scene for wider FOV case in general. Moreover, the accuracy of star locations in terms of (RA, Dec) for quad method is at least 2 times better than the triangle method for 15° FOV.

Finally, the literature survey about the attitude determination algorithms is made. These method are compared in terms of the accuracy, speed and output format. Then, QUEST, SVD method and Triad method are analyzed deeply and compared in terms of accuracy and speed. While SVD method has higher accuracy, QUEST and Triad methods have high-speed operation capability. SVD method takes into account n measurements in the image. Therefore, it is slower but it is more accurate. On the other hand, QUEST method also takes into account n measurements in the image but it calculates eigenvalue by solving characteristic equation. Therefore, it is less accurate than the SVD algorithm. Moreover, this process is slower than the process in the SVD method. On another hand, Triad method uses any of two sources in the image. Therefore, the duration of the attitude calculation is less than the others. However, the accuracy is also less than the other methods. It is observed that for attitude determination algorithms used in this thesis, the accuracy of the SVD algorithm is 4-5 times better than the other algorithms. On the other hand, the process of Triad algorithm is nearly 5 times faster than the SVD algorithm. This duration changes with the sources found in the image.

It can also be said that in tracking mode these sensors give outputs at a higher frequency. While in the lost in space mode, they give the output in 5-10 seconds, but in tracking mode, the output frequency is between 1 and 10 Hz because in tracking mode, the star database can be reduced with the help of previous orientation information.

For the Kalman Filter implementation, a high altitude aircraft was chosen because of reducing the atmospheric effects. Star tracker sensor is affected by day and night differences due to the atmosphere, dust, atmospheric gases, clouds, etc. Therefore, space or high altitude application is more suitable for these sensors. Atmospheric effects should be reduced as much as possible. Moreover, these effects are not modeled in this thesis. Moreover, motion blur is another error source in these sensors. It is caused by rotational motion and vibration. These effects are also not calculated and motion blur is not restored in this thesis because this calculation needs the information of camera and optical parameters. Therefore, it is assumed that this effect is greatly eliminated by hardware. Exposure time, dynamic range, pixel dimensions and focal length are important parameters to reduce the motion blur. Software solutions are not applied. However, the contribution of atmospheric effects and motion blur are taken into consideration when determining of star tracker accuracy levels. Furthermore, there are any other error sources like dark current, read noise, hot/bad pixel, astigmatism vs. in the star tracker sensors. They are also considered to select accuracy levels of star tracker sensor. Yet, any of the error sources are not calculated because hardware of the sensor is not created. In the last chapter, INS/Star tracker implementation is investigated. Moreover, INS/GNSS/Star tracker can also be another option for Kalman Filter implementation. In this chapter, it is seen that the low specific force application by the aid of the star tracker does not give improvement because only INS solution is already pretty good. Yet, in the high specific force implementation, it is observed that the INS solution is improved by the star tracker aid. In order to investigate the results for both high and low specific force flight, 160 seconds simulation was carried out. This simulation shows that for high specific force flight, z-position (altitude) error is nearly decreased by 25 meters compared to the INS-only solutions. This error shows little differences according to the accuracy and sampling frequency rates of star trackers. Moreover, for high specific force flight, it

can be said that attitude errors are nearly decreased by 1°, 0.5° and 1°, for roll, pitch and yaw respectively. Finally, for low specific force flight, for position and velocity states, there are no changes between INS only solution and INS/Star tracker integration approximately. However, attitude states are nearly decreased by 0.15° for low specific force flight.

4.1 Future Work

Many different filter implementations could be applied in the area of INS/Star Tracker integration. For example, Unscented Kalman Filter may be a good choice for non-linear applications. Because this filter does not have linearization steps, it may have more accurate results, but initialization step is important. Particle Filter is one step further. Future of the application of star tracker aided INS solution is filter implementations. Furthermore, INS/GNSS/Star Tracker integration is another hot topic and this implementation is important in aerospace applications.

REFERENCES

[1] ESA, "Geo orbit determination using an aps-based navigation sensor." https://artes.esa.int/projects/geo-orbit-determination-using-aps-based-navigation-sensor. Accessed: 2018.

[2] F. W. Schenkel, "Star tracker/mapper: System design parameters," tech. rep., 1974.

[3] Sodern, "Datasheet hydra tc." http://www.sodern.com/website/docs_wsw/RUB_215/tile_508/Datasheet_HYDRA_TC_2017.pdf, June 2017. Accessed: 2018.

[4] P. G. Savage, "Designing an extended kalman filter for a stellar aided strapdown inertial attitude reference," tech. rep., 2016.

[5] S. Ebcin and M. Veth, "Tightly-coupled image-aided inertial navigation using the unscented kalman filter," tech. rep., 2007.

[6] Y. W. V. Capuano, C. Botteron and J. Tian, "Gnss/ins/star tracker integrated navigation system for earth-moon transfer orbit," *ION*, 2014. Florida.

[7] H. Alkhaldi, "Integration of a star tracker and inertial sensors using an attitude update," 2014.

[8] J. E. M. Marciniak, "On the testing and validation of stray light attenuation for microsatellite star tracker baffles," Master's thesis, Ryerson University, 2014.

[9] Sodern, "The star tracker." http://spot5.cnes.fr/gb/satellite/4831.htm. Accessed: 2018.

[10] J. B. S. Cos, D. Uwaerts and W. Ogiers, "Active pixels for star trackers: Final report," Master's thesis, Mechelen, 2006.

[11] The Editors of Encyclopaedia Britannica, "Star catalog | astronomy | britannica.com." https://www.britannica.com/science/star-catalog. Accessed: 2018.

[12] J. S. T. Delabie and B. Vandenbussche, "An accurate and efficient gaussian fit centroiding algorithm for star trackers," *Space Flight Mechanics Meeting*, Hawaii, 2013.

[13] M. Knutson and D. Miller, "Fast star tracker centroid algorithm for high performance cubesat with air bearing validation," Massachusetts, 2012.

[14] C. R. McBryde, "A star tracker design for cubesats," tech. rep., Texas, 2012.

[15] G. H. a. B. E. C. Fosu, "Determination of centroid of ccd star images," *Department of Geodetic Engineering Commission*, Munich.

[16] "Pixinside reference documentation | dynamicpsf." https://pixinsight.com/doc/tools/DynamicPSF/DynamicPSF.html. Accessed: 2018.

[17] P. B. Stetson, "Daophot - a computer program for crowded-field stellar photometry," tech. rep., 1987.

[18] T. P. Developers, "Daostarfinder — photutils v0.4." http://photutils.readthedocs.io/en/stable/api/photutils.DAOStarFinder.html, October 2017. Accessed: 2018.

[19] T. P. Developers, "Irafstarfinder — photutils v0.4." http://photutils.readthedocs.io/en/stable/api/photutils.IRAFStarFinder.html#photutils.IRAFStarFinder, October 2017. Accessed: 2018.

[20] B. B. Spratling and D. Mortari, "A survey on star identification algorithms," *Algorithms*, pp. 93–107, 2009.

[21] M. S. A. A. R. Toloei and M. Abaszadeh, "A new composite algorithm for identifying the stars in the star tracker," *International Journal of Computer Applications*, 2014.

[22] D. Mortari, "A fast on-board autonomous attitude determination system based on a new star-id technique for a wide fov star tracker," *Adv. Astronaut. Sci.*, no. 93, pp. 893–903, 1997.

[23] D. Mortari, "Search-less algorithm for star pattern recognition," *J. Astronaut. Sci*, no. 45, pp. 179–194, 1997.

[24] P. Alveda, "Neural network star pattern recognition of spacecraft attitude determination and control," *Advances in Neural Information Processing System*, pp. 213–322, 1989.

[25] D. Lang, "Geometric hash codes for large-scale pattern recognition," tech. rep., 2008.

[26] Las Cumbres Observatory, "Cosmic coordinates | las cumbres observatory." `https://lco.global/spacebook/equatorial-coordinate-system/`. Accessed: 2018.

[27] Lang, Hogg, Mierle and Roweis, "Astrometry.net code readme – astrometry.net master documentation." `http://astrometry.net/doc/readme.html`. Accessed: 2018.

[28] Stellarium Developers, "Stellarium astronomy software." `http://stellarium.org/`. Accessed: 2018.

[29] J. A. Tappe, "Development of star tracker system for accurate estimation of spacecraft attitude," tech. rep., Monterey, 2009.

[30] J. L. C. Z. H. Zhang, Y. Niu and Y. Yang, "On-orbit calibration for star sensors without priori information," *Optical Society of America*, 2017.

[31] M. D. Shuster, "The generalized wahba problem," *The Journal of the Astronautical Sciences*, vol. 54, no. 2, pp. 245–259, 2006.

[32] X. G. W. Quan, J. Li and J. Fang, *INS/CNS/GNSS Integrated Navigation Technology*.

[33] A. J. J. Tappe, J. J. Kİm and B. Agrawal, "Star tracker attitude estimation for an indoor ground-based spacecraft simulator," *AIAA Modeling and Simulation Technologies Conference*, Portland, 2011.

[34] F. L. Markley and D. Mortari, "How to estimate attitude from vector observations," NASA's Goddard Space Flight Center, Greenbelt.

[35] F. L. Markley and D. Mortari, "Quaternion attitude estimation using vector observations," NASA's Goddard Space Flight Center, Greenbelt.

[36] C. G. Smith, "Development and implementation of star tracker based attitude determination," tech. rep., Missouri, 2017.

[37] I. Y. Bar-Itzack and R. R. Harman, "Optimized triad algorithm for attitude determination," *Flight Dynamics Support Branch*. Greenbelt, Maryland.

[38] M. and O. S. Shuster, "Three-axis attitude determination from vector observations," *Journal of Guidance and Control*, pp. 70–77, 1981.

[39] M. Z. Hasan and A. Ahmed, "Review on attitude estimation algorithm of attitude determination system," *ARPN Journal of Engineering and Applied Sciences*, vol. 11, no. 7, pp. 4455–4460, 2016.

[40] N. H. J. E. Darling, C. Frueh and K. J. DeMars, "Recursive filtering of star tracker data," *AIAA/AAS Astrodynamics Specialist Conference*, 2016.

[41] D. Mandı, "Simulation of the satellite attitude determination covariance analysis using svd method," Master's thesis, Istanbul Technical University Faculty Of Aeronautics And Astronautics, 2018.

[42] M. G. G. F. A. R. G. Rufino, D. Accardo and U. Tancredi, "Real-time hardware-in-the-loop tests of star tracker algorithms," *International Journal of Aerospace Engineering*, vol. 2013, 2013.

[43] Z. Y. X. W. T. Sun, F. Xing and B. Li, "Smearing model and restoration of star image under conditions of variable angular velocity and long exposure time," *Optical Society of America*, 10 March 2014.

[44] Y. Gibbs, "Nasa armstrong fact sheet: X-15 hypersonic research program | nasa." `https://www.nasa.gov/centers/armstrong/news/FactSheets/FS-052-DFRC.html`, February 2014. Accessed: 2018.

[45] Y. Gibbs, "X-15 rocket-powered aircraft | nasa." https://www.nasa.gov/centers/dryden/multimedia/imagegallery/X-15/E-5251.html, March 2014. Accessed: 2018.

[46] G. Campa, "Airlib - file exchange - matlab central." https://www.mathworks.com/matlabcentral/fileexchange/3019-airlib, March 2004. Accessed: 2018.

[47] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*.