

# Nonlinear Estimation for Autonomous Optical Navigation

by

Stefan Hallgrimson

Bachelor of Engineering, Ryerson University (2017)

A thesis

presented to Ryerson University

in partial fulfillment of the

requirements for the degree of

Master of Applied Science

in the program of

Aerospace Engineering

Toronto, Ontario, Canada, 2019

©Stefan Hallgrimson, 2019

## **AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A THESIS**

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my thesis may be made electronically available to the public.

# Nonlinear Estimation for Autonomous Optical Navigation

Stefan HALLGRIMSON

Master of Applied Science, Aerospace Engineering, Ryerson University, Toronto (2019)

## ABSTRACT

Many interplanetary mission concepts can benefit from autonomous orbit estimation, particularly during critical mission phases. Previous studies have examined the feasibility of optical navigation using nanosatellite class instruments. While promising, these techniques are not without drawbacks. Convergence of the navigation estimates are often sensitive to errors in initial state estimates. This thesis compares various methods to perform nonlinear estimation for autonomous optical navigation. These methods include an extended Kalman filter (EKF), an unscented Kalman filter (UKF), a particle filter (PF), a fixed-lag smoother (FLS), and moving horizon estimation (MHE). The EKF, UKF, and PF can be implemented in real time, while the FLS and MHE implement a delay into the estimation process. To compare the performance of each state estimator three initial reference scenarios around Mars were considered: a hyperbolic flyby, an elliptic orbit and a orbital maneuver using observations of Mars and its moons. Parameter estimation was also explored, where the mass of Mars was to be estimated as a reference parameter in both the hyperbolic and elliptical trajectories. One last reference scenario included a low Earth orbit (LEO) using observations of satellites in a geosynchronous equatorial orbit. In each case, the FLS and MHE showed similar or better performance over each state estimator but at the cost of an increased computation time with respect to the reference EKF. Similarly the UKF was able to provide improved results with respect to the EKF. While, the PF provided poor estimates in the Mars trajectories but improvements were seen from the UKF and EKF in the LEO scenario.

## **ACKNOWLEDGEMENTS**

I would like to especially thank my advisor, Dr. John Enright, whose patience and forethought has guided this research in the right direction.

This gratitude also extends to all my fellow lab mates at SAIL for all their help. As well as my family, friends and girlfriend for their love and support.

# Table of Contents

<b>Declaration of Authorship</b>	<b>ii</b>
<b>Abstract</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>iv</b>
<b>List of Tables</b>	<b>vi</b>
<b>List of Figures</b>	<b>vii</b>
<b>Abbreviations</b>	<b>ix</b>
<b>Physical Constants</b>	<b>x</b>
<b>Symbols</b>	<b>xi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 The ST-16 Star Tracker . . . . .	2
<b>2 Background</b>	<b>5</b>
2.1 Mathematical Notations . . . . .	6
2.2 Discretization . . . . .	8
2.3 Extended Kalman Filter . . . . .	8
2.4 Unscented Kalman Filter . . . . .	9
2.5 Particle Filter . . . . .	11
2.6 Fixed-Lag Smoother . . . . .	13
2.7 Moving Horizon Estimation . . . . .	14
2.7.1 Solving the MHE Optimization Problem . . . . .	16
2.8 Chapter Summary . . . . .	17
<b>3 Hyperbolic Mars Approach</b>	<b>19</b>
3.1 Reference Mission . . . . .	20
3.2 State Variables . . . . .	22

3.3	System Dynamics . . . . .	23
3.4	Measurement Models . . . . .	24
3.5	Simulation Results . . . . .	28
3.6	Chapter Summary . . . . .	35
<b>4</b>	<b>Elliptical Orbit around Mars</b>	<b>36</b>
4.1	Elliptical Orbit . . . . .	36
4.1.1	System Dynamics . . . . .	37
4.1.2	Measurement Models . . . . .	39
4.1.3	Simulation Results . . . . .	40
4.2	Orbital Maneuver Timing . . . . .	47
4.2.1	Reference Mission . . . . .	47
4.2.2	Simulation Results . . . . .	49
4.3	Chapter Summary . . . . .	54
<b>5</b>	<b>Other Applications of Autonomous Optical Navigation</b>	<b>55</b>
5.1	Planetary Mass Estimation . . . . .	55
5.1.1	Simulation Results . . . . .	56
5.2	Geostationay Observations . . . . .	58
5.2.1	Reference Mission . . . . .	59
5.2.2	System Dynamics . . . . .	61
5.2.3	Measurement Model . . . . .	62
5.2.4	Simulation Results . . . . .	63
5.3	Chapter Summary . . . . .	67
<b>6</b>	<b>Conclusion and Future Work</b>	<b>68</b>
6.1	Conclusion . . . . .	68
6.2	Future Work . . . . .	70
	<b>Appendices</b>	<b>71</b>
<b>A</b>	<b>Moving Horizon Estimation Extensions</b>	<b>72</b>
<b>B</b>	<b>Mars Measurement Models</b>	<b>74</b>
B.1	Attitude Measurements . . . . .	74
B.2	Direction Measurements . . . . .	75
B.3	Position Measurements . . . . .	77
B.4	Phase Angle Measurements . . . . .	78
B.5	Terminator Measurements . . . . .	79
	<b>Bibliography</b>	<b>80</b>

# List of Tables

1.1	The ST-16RT Star Tracker . . . . .	3
3.1	Hyperbolic Trajectory Frame Identification . . . . .	20
3.2	Hyperbolic Trajectory Scenario Parameters . . . . .	28
3.3	Hyperbolic Trajectory Simulation Results . . . . .	34
4.1	Elliptical Orbit Frame Identification . . . . .	37
4.2	Mars Zonal Harmonic Coefficients [27] . . . . .	38
4.3	Elliptical Orbit Scenario Parameters . . . . .	41
4.4	Orbital Burn Scenario Parameters . . . . .	50
4.5	Apoapsis Error Standard Deviation [30] . . . . .	50
5.1	Earth Orbit Frame Identification . . . . .	59
5.2	Earth Orbit Scenario Parameters . . . . .	63
6.1	Mean RSS Errors in Position and Velocity for each Trajectory . . . . .	69
6.2	Mean Computation Time Relative to EKF . . . . .	70

# List of Figures

1.1	The ST-16 star tracker [7]. . . . .	3
1.2	Pinhole camera model. . . . .	4
2.1	Estimation window. . . . .	5
3.1	Hyperbolic trajectory showing the periapse, body and orbit frame. . . . .	20
3.2	Rotation from Frame- $S$ to Frame- $T$ . . . . .	22
3.3	Attitude measurement, and Phobos and Deimos direction measurements. . . . .	25
3.4	Mars displacement vector. . . . .	26
3.5	The terminator and phase angles. . . . .	26
3.6	Measurement availability for the hyperbolic Mars approach [22]. . . . .	27
3.7	Hyperbolic orbit position error over single run. . . . .	29
3.8	Hyperbolic orbit velocity error over single run. . . . .	30
3.9	Hyperbolic orbit sensor misalignment and attitude error over a single run. . . . .	30
3.10	Mean position error in MC trials. . . . .	32
3.11	Mean velocity error in MC trials. . . . .	32
3.12	Mean attitude and sensor misalignment errors in MC trials. . . . .	33
3.13	CDF of maximum errors in MC trials. . . . .	33
4.1	Elliptical trajectory showing the periapse and orbital frames. . . . .	37
4.2	Measurement availability for an elliptical orbit. . . . .	40
4.3	Elliptical orbit position error over single run. . . . .	42
4.4	Elliptical orbit velocity error over single run. . . . .	42
4.5	Elliptical orbit velocity error over single run. . . . .	43
4.6	Elliptical orbit mean position error over MC trials. . . . .	44
4.7	Elliptical orbit mean velocity error over MC trials. . . . .	44
4.8	Elliptical orbit mean sensor misalignment and attitude error over MC trials. . . . .	45
4.9	Mean position error for varying apoapsis. . . . .	46
4.10	Mean velocity error for varying apoapsis. . . . .	47
4.11	Flow chart used for estimating the time and magnitude of the orbital burn. . . . .	48
4.12	Periapse error after burn. . . . .	52
4.13	Apoapsis error at time of burn from spacecrafts true position. . . . .	52
4.14	Apoapsis error at time of burn using filter estimates. . . . .	53
5.1	Hyperbolic orbit mean fractional mass error. . . . .	57
5.2	Elliptical orbit mean fractional mass error. . . . .	58



5.3	Pointing rule used to point spacecraft at point along GEO with same azimuth angle. . . . .	60
5.4	Equatorial orbit mean position error. . . . .	64
5.5	Equatorial orbit mean velocity error. . . . .	65
5.6	Sun-synchronous orbit mean position error. . . . .	65
5.7	Sun-synchronous orbit mean velocity error. . . . .	66

# Abbreviations

<b>ADCS</b>	<b>A</b> ttitude <b>D</b> etermination and <b>C</b> ontrol <b>S</b> ystem
<b>CDF</b>	<b>C</b> umulative <b>D</b> istribution <b>F</b> unction
<b>EDL</b>	<b>E</b> nter <b>D</b> escent, and <b>L</b> anding
<b>EKF</b>	<b>E</b> xtended <b>K</b> alman <b>F</b> ilter
<b>FLS</b>	<b>F</b> ixed- <b>L</b> ag <b>S</b> moother
<b>FOV</b>	<b>F</b> ield <b>O</b> f <b>V</b> iew
<b>GEO</b>	<b>G</b> eosynchronous <b>E</b> quatorial <b>O</b> rbital
<b>LEO</b>	<b>L</b> ow <b>E</b> arth <b>O</b> rbital
<b>MC</b>	<b>M</b> onte <b>C</b> arlo
<b>MHE</b>	<b>M</b> oving <b>H</b> orizon <b>E</b> stimation
<b>OpNav</b>	<b>O</b> ptical <b>N</b> avigation
<b>PF</b>	<b>P</b> article <b>F</b> ilter
<b>RSS</b>	<b>R</b> oot <b>S</b> um <b>S</b> quare
<b>SOI</b>	<b>S</b> phere <b>O</b> f <b>I</b> nfluence
<b>UKF</b>	<b>U</b> nscented <b>K</b> alman <b>F</b> ilter

# Physical Constants

Gravitational Constant	$G$	$=$	$6.674 \times 10^{-11}$	$m^3 \cdot kg^{-2} \cdot s^{-2}$
Earth Standard Gravitational Parameter	$\mu_{Earth}$	$=$	$3.986 \times 10^5$	$km^3 \cdot s^{-2}$
Jupiter Standard Gravitational Parameter	$\mu_{Jupiter}$	$=$	$1.267 \times 10^8$	$km^3 \cdot s^{-2}$
Mars Standard Gravitational Parameter	$\mu_{Mars}$	$=$	$4.282 \times 10^4$	$km^3 \cdot s^{-2}$
Sun Standard Gravitational Parameter	$\mu_{Sun}$	$=$	$1.327 \times 10^{11}$	$km^3 \cdot s^{-2}$
Earth $J_2$ Coefficient	$J_2$	$=$	$1.08263 \times 10^{-3}$	$[-]$
Mars $J_2$ Coefficient	$J_2$	$=$	$1.95545 \times 10^{-3}$	$[-]$
Mars $J_2$ Coefficient	$J_3$	$=$	$3.14498 \times 10^{-5}$	$[-]$
Mars $J_2$ Coefficient	$J_4$	$=$	$-1.53774 \times 10^{-5}$	$[-]$

# Symbols

$a$	Semi-major Axis	$km$
$\mathbf{a}$	Acceleration	$km \cdot s^{-2}$
$e$	Eccentricity	[-]
$E$	Eccentric Anomaly	[-]
$f$	Focal Length	[-]
$\mathbf{F}$	State Transition Matrix	[-]
$h$	Orbital Momentum	$km^2 \cdot s^{-1}$
$\mathbf{h}$	Estimated Measurements	[-]
$\mathbf{H}$	Measurement Sensitivity Matrix	[-]
$K$	Kalman Gain	[-]
$M$	Mean Anomaly	[-]
$P$	State Covariance Matrix	[-]
$Q$	Continuous Process Noise Covariance Matrix	[-]
$\mathbf{Q}$	Discrete Process Noise Covariance Matrix	[-]
$r$	Range	[-]
$\mathbf{r}$	Cartesian Position Vector	$km$
$\mathbf{R}$	Measurement Noise Covariance Matrix	$km$
$T$	Orbital Period	$s$
$\mathbf{v}$	Cartesian Velocity Vector	$km \cdot s^{-1}$
$\mathbf{w}$	Process Noise	[-]
$\mathbf{x}$	State Vector	[-]
$\tilde{\mathbf{z}}$	Measurements	[-]

$\beta$	Terminator Angle	$rad$
$\delta_{\text{pix}}$	Pixel Spacing	$\mu m$
$\Delta t$	Measurement Update Time	$s$
$\Delta t_{ap}$	Time to Apoapsis	$s$
$\Delta v$	Burn Magnitude	$km s^{-1}$
$\Delta$	Attitude Error Pseudomeasurement	$rad$
$\mu$	Standard Gravitational Parameter	$km^3 s^{-2}$
$\nu$	True Anomaly	$rad$
$\nu$	Measurement Noise	$[-]$
$\rho_P$	Target Position	$km$
$\sigma$	Standard Deviation	$[-]$
$\tau$	Attitude Error Time Constant	$s$
$\phi$	Attitude Error	$rad$
$\chi$	Phase Angle	$rad$
$\psi$	Sensor Misalignment Error	$rad$

*I dedicate this to my friends, family and loved ones.*

# Chapter 1

## Introduction

A small satellite is of smaller mass and size than the conventional satellite, and have become popularized due to their economic benefits. Their small size allows for a reduction in manufacturing and launching costs. One class of small satellite is a nanosatellite, which has a mass ranging from 1 to 10 kg. For a nanosatellite mission a navigation system may be required to locate or guide the spacecraft depending on the mission requirements. One method to perform navigation is by optical navigation (OpNav), where a star tracker can be used to fit the small size and weight requirements that come with nanosatellite missions.

OpNav can provide orbit determination by analyzing captured images. It is a proven concept that has been used in a wide variety of missions. It was first demonstrated in the 1960s by the Mariner 6 and Mariner 7 missions, where Mariner 6 used images of the lit limb of Mars to determine the spacecrafts trajectory as it approached the planet [1]. Recently the New Horizons mission used OpNav for planning the flyby encounter with Pluto in 2015 [2], while the Cassini mission used images of Saturn's moons for trajectory planning [3]. When performing OpNav there are distinctions between the type of observations made. For example, in this thesis horizon based OpNav was performed. This method uses the lit limb of a planet or moon to perform OpNav. Other methods include landmark navigation which uses distinct landmarks, such as observations of craters on the surface of a planet, for orbit determination or entry, descent and landing (EDL) applications. While, star occultation can provide measurements by observing the appearance or disappearance of stars behind other bodies.

As more small satellites are launched ground based tracking can become more challenging. Autonomous OpNav can alleviate some of the challenges by allowing the satellite to perform estimates of its state without communication with Earth. To add it can potentially reduce

navigation and operation costs by allowing less reliance on ground personal and ground based tracking through automating orbit determination or trajectory planning. A autonomous system can also provide faster updates to the spacecraft than would be achievable through ground communication [4]. Autonomous OpNav was first demonstrated on the Deep Space 1 mission and has since seen success on many missions [5].

The effectiveness of autonomous OpNav often depends on the convergence properties of the chosen state estimator, which fuses together known dynamic and measurement models with observations made by the satellite to produce an estimate of the spacecrafts state. An extended Kalman filter (EKF) is one of the most popular methods for nonlinear estimation. However, the EKF can be unreliable if there are large nonlinearities present in the dynamic or measurement models [6]. The aim of this thesis is to explore other forms of nonlinear estimation to improve the navigation accuracy for nanosatellite missions by exploring other algorithms that perform nonlinear estimation. These algorithms include an EKF to be used as a reference solution, an unscented Kalman filter (UKF), a particle filter (PF), a fixed-lag smoother (FLS), and moving horizon estimation (MHE). To compare the performance of each state estimator multiple case studies were used. To perform OpNav a nanosatellite star tracker was used. Section 1.1 provides an introduction to star trackers and the star tracker used for all simulations.

In this thesis, Chapter 2 provides an overview of each state estimator, the various mathematical notations used, and the star tracker used to perform OpNav. Chapter 3 outlines the first reference scenario, a hyperbolic Mars approach using observations of Mars and its moons. Here the scenario’s framework, the dynamic and measurement models, and results from a series of Monte Carlo (MC) trials is presented. Chapter 4 uses the framework seen in Chapter 3 for the next two reference scenarios, a elliptical orbit around Mars, and comparing the performance of each state estimator in timing an orbital maneuver. Chapter 5 looks at applications of OpNav, these include parameter estimation, and another reference mission of a satellite in low Earth orbit (LEO) using observations of geostationary satellites for orbit determination. Lastly, Chapter 6 summarizes the thesis, provides any concluding remarks, and an outline for potential future work.

## 1.1 The ST-16 Star Tracker

A star tracker is typical used to determine the attitude of the spacecraft by capturing stars in the field of view of the sensor and comparing their position to a star catalog. Extending the applications to OpNav can potential reduce the required instruments onboard the nanosatellite



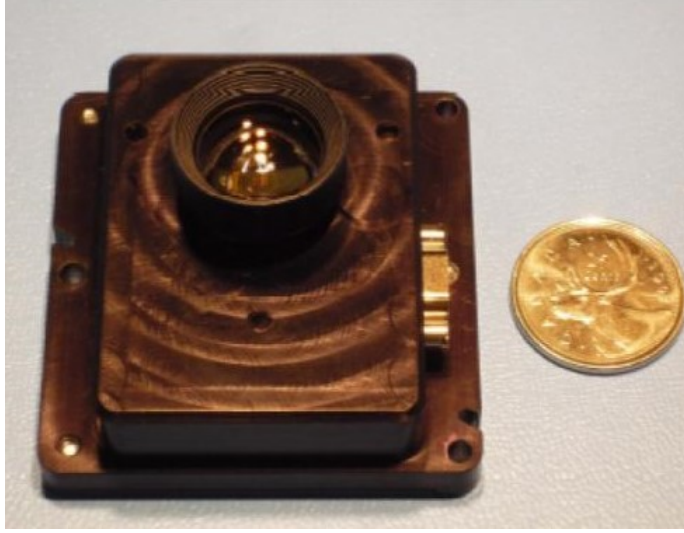


FIGURE 1.1: The ST-16 star tracker [7].

by having the sensor perform many tasks. A miniaturized star tracker could also be fitted onboard a nanosatellite to meet the required size and mass requirements.

To perform optical navigation and model the performance of a typical nanosatellite class star tracker parameters from Sinclair Interplanetarys ST-16RT star tracker were used. The ST-16RT star tracker is a nanosatellite class star tracker with key parameters listed in Table 1.1. Figure 1.1 shows the ST-16 star tracker, the predecessor of the ST-16RT, and its size relative to a loonie.

TABLE 1.1: The ST-16RT Star Tracker

Focal Length	16 mm
Field of View	7.5° half-angle
Pixel Size	2.2 $\mu\text{m}$
Accuracy	< 7 arcsecond cross-boresight (RMS) < 70 arcsecond around boresight (RMS)
Mass	$\approx 90$ g
Size	$59 \times 56 \times 31.5$ mm

In the preceding chapters the observations made by the star tracker were based on a pinhole camera model. A coordinate system was attached to the sensor to describe where it was pointing and the location of a point on the detector plane. Figure 1.2 outlines the sensor frame.

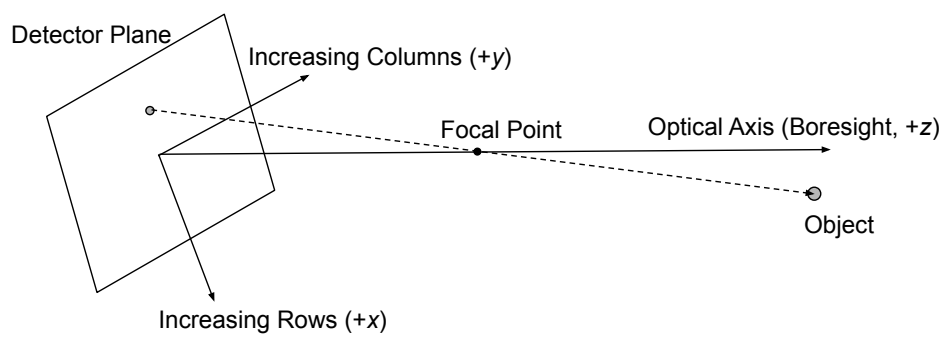


FIGURE 1.2: Pinhole camera model.

## Chapter 2

# Background

To perform nonlinear state estimation, five algorithms were considered: an extended Kalman filter (EKF); an unscented Kalman filter (UKF); a particle filter (PF); a fixed-lag smoother (FLS); and moving horizon estimation (MHE). The EKF, UKF and PF can be implemented in real time, while the FLS and MHE introduce a delay into the estimation process. Throughout this thesis a window length of  $N$  (or delay of  $N-1$ ) contains all discrete state values in the range of  $\mathbf{x}_{k-N+1}$  to  $\mathbf{x}_k$ , refer to Figure 2.1. The notation  $\mathbf{x}_{p|q}$  was used to refer to some estimate  $\mathbf{x}_p$  at time  $t_q$ . For example,  $\mathbf{x}_{k|k-2}$  refers to the estimate of  $\mathbf{x}_k$  two time steps ago. The shorthand notation  $\mathbf{x}_k$  or  $\mathbf{x}_{k-N+1}$  for example was used to represent  $\mathbf{x}_{k|k}$  and  $\mathbf{x}_{k-N+1|k}$  respectively. That is the estimates of  $\mathbf{x}_k$  or  $\mathbf{x}_{k-N+1}$  at the current time step  $t_k$ .

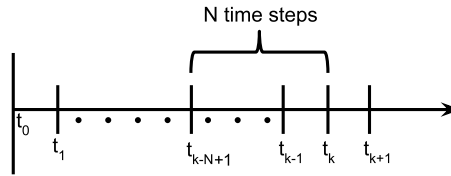


FIGURE 2.1: Estimation window.

A continuous dynamic model was used to model the spacecrafts state,  $\mathbf{x}$ , over time. Thus the state matrix was propagated as

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{w}, t) \quad (2.1)$$

Where,  $\mathbf{w}(t)$  refers to the process noise with continuous process noise covariance matrix  $Q(t)$ , by definition  $Q(t) = E \left\{ \mathbf{w}(t)\mathbf{w}(t)^T \right\}$ . Strictly speaking the process noise is random inputs that excite the dynamic system. It can be used to compensate for some modeling error between the dynamic model and actual system. To add each discrete measurement was assumed to have

additive noise, leading to the measurement model

$$\tilde{\mathbf{z}}_k = \mathbf{h}(\mathbf{x}_k, t_k) + \boldsymbol{\nu}_k \quad (2.2)$$

Here,  $\boldsymbol{\nu}_k$  refers to the measurement noise with covariance matrix  $\mathbf{R}_k$ ,  $\mathbf{R}_k = \mathbb{E} \left\{ \boldsymbol{\nu}_k \boldsymbol{\nu}_k^T \right\}$ , at some time step  $k$ . The measurement noise is used to model the variance seen in the sensor measurements. As needed the dynamic and measurement Jacobians can be evaluated, denoted as  $\mathbf{F}_k$  and  $\mathbf{H}_k$  respectively.

$$\mathbf{F}_k = \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}_k} \quad (2.3)$$

$$\mathbf{H}_k = \left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}_k} \quad (2.4)$$

To initialize each state estimator an initial estimate of the state is provided,  $\hat{\mathbf{x}}_0$ , with covariance matrix  $\mathbf{P}_0 = \mathbb{E} \left\{ (\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T \right\}$ . The initial covariance matrix is based on how accurately the initial state estimate is believed to be.

The rest of this chapter provides background knowledge on the various concepts used throughout this thesis. Section 2.1 presents the mathematical notions used throughout the thesis, Section 2.2 overviews a method to discretize the continuous system, Section 2.3 to Section 2.7 provide a brief overview of each nonlinear state estimator, and lastly, Section 1.1 presents the star tracker that was used for OpNav.

## 2.1 Mathematical Notations

This section provides some overview on the mathematical notations used further in this thesis. To begin, principal axes rotations are denoted by  $\mathbf{C}_1$ ,  $\mathbf{C}_2$ , and  $\mathbf{C}_3$ . Given a rotation angle  $\theta$  each rotation matrices is defined as

$$\mathbf{C}_1(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix} \quad (2.5)$$

$$\mathbf{C}_2(\theta) = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \quad (2.6)$$

$$\mathbf{C}_3(\theta) = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

Given a matrix  $\mathbf{a} = [a_1 \ a_2 \ a_3]^T$  a skew symmetric matrix that describes the cross product is defined as

$$\mathbf{a}^\times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \quad (2.8)$$

A small angle error rotation vector is defined as a series of Euler angle rotations. Given a small angle error rotation vector  $\boldsymbol{\theta}$ , a first-order approximation can be defined as

$$\mathbf{C}(\boldsymbol{\theta}) = \mathbf{C}_1(\theta_1)\mathbf{C}_2(\theta_2)\mathbf{C}_3(\theta_3) \approx (\mathbf{I}_{3 \times 3} - \boldsymbol{\theta}^\times) \quad (2.9)$$

A axis angle rotation of the form  $\boldsymbol{\theta} = \theta \mathbf{a}$ , where  $\mathbf{a}$  is a unit vector, can be expressed by a rotation matrix using

$$\mathbf{C}(\theta) = \cos \theta \mathbf{I}_{3 \times 3} + (1 - \cos \theta) \mathbf{a} \mathbf{a}^T - \sin \theta \mathbf{a}^\times \quad (2.10)$$

Lastly, the notation  $\mathbf{I}_{n \times n}$  refers to an identity matrix of size  $n$ . For example,

$$\mathbf{I}_{3 \times 3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.11)$$

## 2.2 Discretization

For implementation of some of the state estimators a discrete dynamic model is needed. To discretize the continuous system Van Loan's procedure was done at each time step [8]. To begin, the matrix  $M$  is formulated using the continuous state matrix  $\mathbf{F}_k$  at time step  $k$  with continuous covariance matrix  $Q(t)$ .

$$M = \begin{bmatrix} -\mathbf{F}_k & Q(t) \\ 0 & \mathbf{F}_k^T \end{bmatrix} \quad (2.12)$$

The matrix exponential is then taken

$$E = e^{M\Delta t} = \begin{bmatrix} E_{11} & E_{12} \\ 0 & E_{22} \end{bmatrix} \quad (2.13)$$

This leads to the discrete state transition matrix  $\Phi_k$  of the form  $\mathbf{x}_{k+1} = \Phi_k \mathbf{x}_k$  with discrete process noise covariance  $\mathbf{Q}_k$  at time instance  $t_k$ .

$$\Phi_k = E_{22} \quad (2.14)$$

$$\mathbf{Q}_k = E_{12} E_{11}^T \quad (2.15)$$

## 2.3 Extended Kalman Filter

A Kalman Filter (KF) takes sensor measurements with known dynamic and measurement models to produce an estimate of the current state. For linear systems with Gaussian noise the KF is the optimal estimator [9]. An Extended Kalman Filter adapts the standard KF equations to a nonlinear system by linearizing the functions  $f()$  and  $\mathbf{h}()$  around the current estimated state while assuming Gaussian process and measurement noise. For a continuous system with discrete measurements the equations are summarized below, where  $\hat{\mathbf{x}}_k$  refers to an estimate of the state at time step  $k$  with covariance  $\mathbf{P}_k$ . As well in the proceeding equations the notation  $a_k^-$  refers to the pre-update value right after propagation and  $a_k^+$  the post-update value right before propagation. The update step includes

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k) \quad (2.16)$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k [\tilde{\mathbf{z}}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-, t_k)] \quad (2.17)$$

$$\mathbf{P}_k^+ = [\mathbf{I} - \mathbf{K}_k \mathbf{H}_k] \mathbf{P}_k^- \quad (2.18)$$

With the state and covariance propagated as

$$\dot{\hat{\mathbf{x}}} = f(\hat{\mathbf{x}}, t) \quad (2.19)$$

$$\dot{\mathbf{P}}_k = \mathbf{F}_k \mathbf{P}_k + \mathbf{P}_k \mathbf{F}_k^T + \mathbf{Q}(t) \quad (2.20)$$

## 2.4 Unscented Kalman Filter

A UKF achieves state estimation using the Unscented Transform to propagate a set of  $2n + 1$  sigma points through the nonlinear system, with  $n$  being the size of the state matrix. A UKF has been shown to have a lower error than the EKF by representing the state distribution as a set of sample points rather than through linearizations [10]. To add partial derivatives do not need to be found. Using an augmented state matrix,  $\mathbf{x}_k^a = [\hat{\mathbf{x}}_k^T \mathbf{w}_k^T]^T$ , the set of sigma points, partitioned as  $\chi_k = [(\chi_k^x)^T (\chi_k^w)^T]^T$ , at each time step can be found. Here  $\mathbf{w}_k$  refers to the discrete process noise.

$$\chi_k = \begin{bmatrix} \hat{\mathbf{x}}_k^a & \hat{\mathbf{x}}_k^a + \sqrt{(n + \lambda)\mathbf{P}_k^a} & \hat{\mathbf{x}}_k^a - \sqrt{(n + \lambda)\mathbf{P}_k^a} \end{bmatrix} \quad (2.21)$$

where,

$$\hat{\mathbf{x}}_k^a = [\hat{\mathbf{x}}_k^T \ 0_{1 \times m}]^T \quad (2.22)$$

Here,  $\lambda$  is a scaling parameter found as  $\lambda = \alpha^2(n + \kappa) - n$ , where  $\alpha$  determines the spread of the sigma points, and  $\kappa$  is a secondary scaling parameter. Given a process noise covariance matrix of dimension  $m$ , and assuming no correlation between state error and process noise,  $\mathbf{P}_k^a$  refers to an augmented covariance matrix of the form:

$$\mathbf{P}_k^a = \begin{bmatrix} \mathbf{P}_k^+ & \mathbf{0}_{n \times m} \\ \mathbf{0}_{m \times n} & \mathbf{Q}_k \end{bmatrix} \quad (2.23)$$

The estimated states are found as a weighted sum of the sigma points, with  $i = 0$  representing the first sigma point  $\mathbf{x}_k^a$ .

$$\hat{\mathbf{x}}_k^- = \sum_{i=0}^{2L} W_i^{mean} \chi_k(i) \quad (2.24)$$

With covariance matrix

$$\mathbf{P}_k^- = \sum_{i=0}^{2L} W_i^{cov} [\chi_k(i) - \hat{\mathbf{x}}_k^-] [\chi_k(i) - \hat{\mathbf{x}}_k^-]^T \quad (2.25)$$

and mean observation

$$\hat{\mathbf{y}}_k^- = \sum_{i=0}^{2L} W_i^{mean} \gamma_k(i) \quad (2.26)$$

with

$$\gamma_k(i) = \mathbf{h}(\chi_k(i), t_k) \quad (2.27)$$

In Wan and Merwe [11] it is stated that if the measurement noise is purely additive the measurement covariance matrix can simply be added to the output covariance. This also has the advantage of a reduced computation time by reducing the amount of sigma points. This means

$$\mathbf{P}_k^{yy} = \sum_{i=0}^{2L} W_i^{cov} [\gamma_k(i) - \hat{\mathbf{y}}_k^-] [\gamma_k(i) - \hat{\mathbf{y}}_k^-]^T + \mathbf{R}_k \quad (2.28)$$

$$\mathbf{P}_k^{xy} = \sum_{i=0}^{2L} W_i^{cov} [\chi_k(i) - \hat{\mathbf{x}}_k^-] [\gamma_k(i) - \hat{\mathbf{y}}_k^-]^T \quad (2.29)$$

Finally, the state and error covariance can be updated as

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K_k (\tilde{\mathbf{z}}_k - \hat{\mathbf{y}}_k^-) \quad (2.30)$$

$$\mathbf{P}_k^+ = \mathbf{P}_k^- - K_k \mathbf{P}_k^{yy} K_k^T \quad (2.31)$$

where

$$K_k = \mathbf{P}_k^{xy} (\mathbf{P}_k^{yy})^{-1} \quad (2.32)$$

Each sigma point is propagated by integrating Eq. (2.33) over the time span  $t_k$  to  $t_{k+1}$ .

$$\chi_{k+1} = f(\chi_k, t_k) \quad (2.33)$$

Lastly, scalar weights can be defined as  $W_0^{mean} = W_0^{cov} = \frac{\lambda}{n+\lambda}$ , and  $W_i^{mean} = W_i^{cov} = \frac{1}{2(n+\lambda)}$  for  $i = 1, 2, \dots, 2n$ .



## 2.5 Particle Filter

A particle filter uses a set of  $n$  particles to approximate the posterior distribution  $p(\mathbf{x}_k|\tilde{\mathbf{z}}_{0:k})$  of the system without making assumptions about the distribution. In this paper a Bootstrap filter is considered [12]. To begin, each particle is initialized by sampling an initial distribution  $\mathbf{x}_0^{(j)} \sim p(\mathbf{x}_0)$  for  $j = 1, \dots, n$ . At each update step the particles are assigned an importance weight  $w_k^{(j)}$  based on the estimated quality of each particle relative to the true state. With each weight being initialized as  $w_0^{(j)} = \frac{1}{n}$  for  $j = 1, \dots, n$ . For a Markov process the importance weight for each particle is

$$w_k^{(j)} = w_{k-1}^{(j)} \frac{p(\tilde{\mathbf{z}}_k|\mathbf{x}_k^{(j)})p(\mathbf{x}_k^{(j)}|\mathbf{x}_{k-1}^{(j)})}{q(\mathbf{x}_k^{(j)}|\mathbf{x}_{k-1}^{(j)}, \tilde{\mathbf{z}}_k)} \quad (2.34)$$

However in a Bootstrap filter the transition and importance density are assumed equal,  $p(\mathbf{x}_k^{(j)}|\mathbf{x}_{k-1}^{(j)}) = q(\mathbf{x}_k^{(j)}|\mathbf{x}_{k-1}^{(j)}, \tilde{\mathbf{z}}_k)$ . Therefore, each importance weight is a function of the likelihood function.

$$w_k^{(j)} = w_{k-1}^{(j)} p(\tilde{\mathbf{z}}_k|\mathbf{x}_k^{(j)}) \quad (2.35)$$

Once found the weights are normalized so that  $\sum_{j=1}^n w_k^{(j)} = 1$ . From below the notation  $a \leftarrow b$  means  $a$  is set to  $b$ .

$$w_k^{(j)} \leftarrow \frac{w_k^{(j)}}{\sum_{j=1}^n w_k^{(j)}} \quad (2.36)$$

Assuming a Gaussian additive measurement noise the likelihood function at each time step becomes  $p(\tilde{\mathbf{z}}_k - \mathbf{h}(\mathbf{x}_k^{(j)}, t_k))$ , thus from Crassidis and Junkins [6] (pp. 79)

$$p(\tilde{\mathbf{z}}_k - \mathbf{h}(\mathbf{x}_k^{(j)}, t_k)) = \frac{1}{\sqrt{(2\pi)^n |\mathbf{R}_k|}} \exp\left[-\frac{1}{2}(\tilde{\mathbf{z}}_k - \mathbf{h}(\mathbf{x}_k^{(j)}, t_k))^T \mathbf{R}_k^{-1} (\tilde{\mathbf{z}}_k - \mathbf{h}(\mathbf{x}_k^{(j)}, t_k))\right] \quad (2.37)$$

For systems where few particles fall within significant regions of the likelihood function, meaning many weights go to zero or become significantly small, each likelihood value can be raised to a power  $c$ , where  $c \in (0, 1)$ . This has the affect of smoothing the observation model [13]. The value  $c$  can be found through experimentation.

The estimated state and its covariance matrix can then be computed as

$$\hat{\mathbf{x}}_k = \sum_{j=1}^N w_k^{(j)} \mathbf{x}_k^{(j)} \quad (2.38)$$

$$\mathbf{P}_k = \sum_{j=1}^N w_k^{(j)} \bar{\mathbf{x}}_k^{(j)} \bar{\mathbf{x}}_k^{(j)T} \quad (2.39)$$

$$\bar{\mathbf{x}}_k^{(j)} = \mathbf{x}_k^{(j)} - \hat{\mathbf{x}}_k \quad (2.40)$$

As the system progresses over time the distribution of the importance weights can become skewed in such a way that only one particle will have a meaningful weight [14]. Resampling the distribution can help mitigate this problem and maintain an effective population of particles. Resampling replaces the particles with negligible weights with higher weighted particles. After resampling each particle is assigned an equal weight of  $w_k^{(j)} = \frac{1}{n}$  for  $j = 1, \dots, n$ . Before resampling occurs the number of effective particles  $N_{eff}$  can be estimated as  $N_{eff} \approx \frac{1}{\sum_{j=1}^n (w_k^{(j)})^2}$ . In this paper resampling is performed once  $N_{eff}$  falls below a threshold value, unlike in the normal Bootstrap filter where it is performed at every time step. There are many algorithms that perform resampling, Li et al. [15] provides a comparison of various methods from which systematic resampling was chosen. Systematic resampling is one of the more popular techniques due to its ease of implementation and decreased computation time. The systematic resampling algorithm is

1. Generate  $c_i$  as the  $i^{th}$  cumulative sum of  $w^{(j)}$  for  $j = 1 : i$ , done by setting  $c_1 = w^{(1)}$  then

For  $i = 2 : n$

$$c_i = c_{i-1} + w^{(i)}$$

end

2. Draw a single uniform sample  $u_0 \sim U(0, n^{-1})$ , and set  $j = 1$

For  $i = 1 : n$

While  $c_j < u$

$j = j + 1$

end

$$\mathbf{x}^{(i)} \leftarrow \mathbf{x}^{(j)}$$

$$u = u_0 + \frac{i}{n}$$

end

3. Set  $w^{(j)} = \frac{1}{n}$  for  $j = 1, \dots, n$

Resampling can however create many identical particles so a roughening step can be added to provide variance to the particles, this step adds an independent jitter  $c_k^{(j)}$  to each particle [14].

$$\mathbf{x}_k^{(j)} \leftarrow \mathbf{x}_k^{(j)} + c_k^{(j)} \quad (2.41)$$

This jitter is drawn from a Gaussian distribution  $\mathcal{N}(0, h^2 \Sigma_k)$  with  $h$  being a tuning parameter and  $\Sigma_k$  defined below [16]. The jitter is tuned in such a way as to provide variance to the particles but not change the original distribution.

$$\Sigma_k = \frac{1}{n-1} \sum_{j=1}^n \tilde{\mathbf{x}}_k^{(j)} \tilde{\mathbf{x}}_k^{(j)T} \quad (2.42)$$

$$\tilde{\mathbf{x}}_k^{(j)} = \bar{\mathbf{x}}_k^{(j)} - \hat{\mathbf{x}}_k \quad (2.43)$$

$$\hat{\mathbf{x}}_k = \frac{1}{N} \sum_{j=1}^n \mathbf{x}_k^{(j)} \quad (2.44)$$

## 2.6 Fixed-Lag Smoother

A fixed-lag smoother finds improved estimates of the states  $\hat{\mathbf{x}}_{k-N+1|k}$  to  $\hat{\mathbf{x}}_{k|k}$  at the current time step  $k$  given a lag of  $N-1$  and measurements  $\hat{\mathbf{z}}_{k-N+1}$  to  $\hat{\mathbf{z}}_k$ . Through making use of the EKF filtering equations, Eq. (2.16) to Eq. (2.20), and formulating an augmented state vector  $\hat{\mathbf{x}}_a$  the smoothed state estimates can be found. For the linear discrete case this algorithm is presented in Moore [17], which is then extended for the nonlinear case with discrete measurements in Moore [18]. The augmented state vector is formulated as

$$\hat{\mathbf{x}}_k^a = [\hat{\mathbf{x}}_{k|k}^T \quad \hat{\mathbf{x}}_{k-1|k}^T \quad \hat{\mathbf{x}}_{k-2|k}^T \quad \dots \quad \hat{\mathbf{x}}_{k-N+1|k}^T]^T \quad (2.45)$$

For the continuous system the augmented state vector is propagated as

$$\dot{\hat{\mathbf{x}}}_k^a = \begin{bmatrix} \dot{\hat{\mathbf{x}}}_{k|k} \\ \dot{\hat{\mathbf{x}}}_{k-1|k} \\ \dot{\hat{\mathbf{x}}}_{k-2|k} \\ \vdots \\ \dot{\hat{\mathbf{x}}}_{k-N+1|k} \end{bmatrix} = \begin{bmatrix} f(\hat{\mathbf{x}}_{k|k}, t_k) \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{w}_k \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (2.46)$$

with Jacobian

$$\mathbf{F}_k^a = \begin{bmatrix} \mathbf{F}_k & 0 & \cdots & 0 & 0 \\ I & 0 & \cdots & 0 & 0 \\ 0 & I & \ddots & \vdots & \vdots \\ \vdots & & \ddots & 0 & 0 \\ 0 & 0 & \cdots & I & 0 \end{bmatrix} \quad (2.47)$$

and measurement sensitivity matrix

$$\mathbf{H}_k^a = [\mathbf{H}_k \quad 0 \quad \cdots \quad 0] \quad (2.48)$$

Lastly, the augmented covariance matrix can be seen below.  $\mathbf{P}_k^{(i,j)}$  represents the covariance between the state vectors at time  $i$ , and  $j$ . Thus,  $\mathbf{P}_k^{(k-N+1,k-N+1)}$  represents the error covariance of the smoother.

$$\mathbf{P}_k = \begin{bmatrix} \mathbf{P}_k^{(k,k)} & \mathbf{P}_k^{(k,k-1)} & \cdots & \mathbf{P}_k^{(k,k-N+1)} \\ \mathbf{P}_k^{(k-1,k)} & \mathbf{P}_k^{(k-1,k-1)} & \cdots & \mathbf{P}_k^{(k-1,k-N+1)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_k^{(k-N+1,k)} & \mathbf{P}_k^{(k-N+1,k-1)} & \cdots & \mathbf{P}_k^{(k-N+1,k-N+1)} \end{bmatrix} \quad (2.49)$$

## 2.7 Moving Horizon Estimation

Moving horizon estimation involves solving a least squares optimization problem over a sliding window of  $N$  time steps to find the optimal states. MHE has the main advantage that constraints can be added into the filtering problem. Advantages of the unconstrained MHE may be seen from the EKF due to the smoothed state estimates however. Estimates at the current time step are the filtered estimates and all subsequent estimates are smoothed. The MHE cost function is presented below, with  $k$  representing the current time step.

$$\min_{\mathbf{x}_{k-N+1}, \{\mathbf{w}_k\}_{k-N+1}^{k-1}} \sum_{i=k-N+1}^k \|\boldsymbol{\nu}_i\|_{W_Q}^2 + \sum_{i=k-N+1}^{k-1} \|\mathbf{w}_i\|_{W_R}^2 + \Gamma(\mathbf{x}_{k-N+1}) \quad (2.50)$$

subject to

$$\begin{aligned} \mathbf{x}_{i+1} &= \mathbf{f}(\mathbf{x}_i, t_i) + \mathbf{w}_i \\ \tilde{\mathbf{z}}_i &= \mathbf{h}(\mathbf{x}_i, t_i) + \boldsymbol{\nu}_i \end{aligned}$$

Weights can be defined as the inverse of the discrete covariance matrices with process and measurement noise  $\mathbf{w}_k$  and  $\boldsymbol{\nu}_k$  respectively,  $W_{\mathbf{Q}} = \text{diag}([\mathbf{Q}_{k-1}^{-1} \ \mathbf{Q}_{k-2}^{-1} \ \cdots \ \mathbf{Q}_{k-N+1}^{-1}])$  and  $W_{\mathbf{R}} = \text{diag}([\mathbf{R}_k^{-1} \mathbf{R}_{k-1}^{-1} \cdots \mathbf{R}_{k-N+1}^{-1}])$ . In the least squares problem the term  $\Gamma(\mathbf{x}_{k-N+1})$  describes the *arrival cost* which is used to partially summarize all past data that has left the window. Two different arrival costs presented by V. Rao et al. [19] were considered, the filtering and smoothing formulation.

- **Filtering Formulation:** Here estimates are penalized for their deviation from an *a priori* estimate  $\mathbf{x}_{k-N+1|k-N+1}$  of the state  $k-N+1$  time steps ago with covariance  $\mathbf{P}_{k-N+1|k-N+1}$ .

$$\Gamma(\mathbf{x}_{k-N+1}) = \|\mathbf{x}_{k-N+1} - \mathbf{x}_{k-N+1|k-N+1}\|_{\mathbf{P}_{k-N+1|k-N+1}^{-1}}^2 \quad (2.51)$$

This covariance can be found using the covariance update and propagation equations from the EKF, these are Eq. (2.18) and Eq. (2.20).

- **Smoothing Formulation:** In Eq. (2.52) estimates are penalized for their deviation from an estimate  $\mathbf{x}_{k-N+1|k-1}$  of the state at a previous time step  $k-1$ . With the covariance  $\mathbf{P}_{k-N+1|k-1}$  found by solving the backward Riccati equation given estimates  $\mathbf{P}_{k|k}$  and  $\mathbf{P}_{k+1|k}$ . Refer to Appendix A for expansion of unknown terms. The measurements  $\tilde{\mathbf{z}}_{T-N+1:T-1}$  are used previously to find the estimate of the state  $\mathbf{x}_{T-N+1|T-1}$  and are to be used in the MHE problem when finding the estimates in the current window, the second term in this cost function is used to ensure these measurements are not counted twice.

$$\Gamma(\mathbf{x}_{k-N+1}) = \|\mathbf{x}_{k-N+1} - \mathbf{x}_{k-N+1|k-1}\|_{\mathbf{P}_{k-N+1|k-1}^{-1}}^2 - \|\mathbf{Y}_{k-1}^{k-N+1} - \mathbf{O}_{N-1}\mathbf{x}_{k-N+1}\|_{\mathbf{W}_{N-1}^{-1}}^2 \quad (2.52)$$

Using the state transition matrix  $\Phi_i$  the backward Riccati equation is found below starting at  $i = k-1$  and propagating backward until  $k = T-N+1$ .

$$\mathbf{P}_{i|k-2} = \mathbf{P}_{i|i} + \mathbf{P}_{i|i}\Phi_i^T\mathbf{P}_{i+1|k-2}^{-1}(\mathbf{P}_{i+1|k-2} - \mathbf{P}_{i+1|i})\mathbf{P}_{i+1|k}^{-1}\Phi_i\mathbf{P}_{i|i} \quad (2.53)$$

The covariance at time step  $k$  can be found the same as in the filtering formulation.

The filtering formulation can induce oscillations into the system as initial errors in state estimation are propagated through time [19]. The smoothing formulation can resolve these oscillations, however Haseltine and Rawlings [20] states that in the case of local optima the smoothing formulation can create a bias in state estimates, and that for global optimization a uniform prior

with a longer horizon length is recommended. In this thesis the filtering formulation was implemented with a suitable horizon length. This conclusion was made by comparing the performance of each formulation in the to be seen reference scenarios, and noting the increased computation time observed with the smoothing formulation.

### 2.7.1 Solving the MHE Optimization Problem

Lastly, to solve the MHE problem Vandersteen et al. [21] presents an iterative Gauss-Newton method where Eq. (2.50) is linearized around the previous best estimate  $\bar{\mathbf{x}}_i$  for  $i = k - N + 1, k - N + 2, \dots, k$ , and a new estimate  $\mathbf{x}_i$  is found by finding the deviation  $\Delta \mathbf{x}_i$  from this previous estimate,  $\mathbf{x}_i = \bar{\mathbf{x}}_i + \Delta \mathbf{x}_i$ . To begin, each term in Eq. (2.50) is evaluated at  $\bar{\mathbf{x}}_i$ , and partials are taken around the previous estimate. These terms are then separated into matrices  $B_j$  and  $A_j$  respectively for each term in the MHE problem. As well, a term  $\Delta \mathbf{X} = [\Delta \mathbf{x}_{k-N+1}^T, \Delta \mathbf{x}_{k-N+2}^T, \dots, \Delta \mathbf{x}_k^T]^T$  is introduced to determine the deviation over the entire window. Given an equality constraint the linearized cost function becomes

$$\min_{\Delta \mathbf{x}_{k-N+1}} \|A_R \Delta X - B_R\|_{W_R}^2 + \|A_Q \Delta X - B_Q\|_{W_Q}^2 + \|A_A \Delta X - B_A\|_{P_{k-N+1|k-N+1}^{-1}}^2 \quad (2.54)$$

subject to

$$A_{eq} \Delta X - B_{eq} = 0 \quad (2.55)$$

Where, for example

$$A_Q = \begin{bmatrix} \frac{\partial \mathbf{f}(\bar{\mathbf{x}}_{k-N+1}, t_{k-N+1})}{\partial \bar{\mathbf{x}}_{k-N+1}} & 1 & 0 & \dots & 0 \\ 0 & \frac{\partial \mathbf{f}(\bar{\mathbf{x}}_{k-N+2}, t_{k-N+2})}{\partial \bar{\mathbf{x}}_{k-N+2}} & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & & 0 \\ 0 & 0 & \dots & \frac{\partial \mathbf{f}(\bar{\mathbf{x}}_{k-1}, t_{k-1})}{\partial \bar{\mathbf{x}}_{k-1}} & 1 \end{bmatrix} \quad (2.56)$$

and

$$B_Q = \left[ [\bar{\mathbf{x}}_{k-N+2} - \mathbf{f}(\bar{\mathbf{x}}_{k-N+1}, t_{k-N+1})]^T, [\bar{\mathbf{x}}_{k-N+3} - \mathbf{f}(\bar{\mathbf{x}}_{k-N+2}, t_{k-N+2})]^T, \dots, [\bar{\mathbf{x}}_k - \mathbf{f}(\bar{\mathbf{x}}_{k-1}, t_{k-1})]^T \right]^T \quad (2.57)$$

The Karush–Kuhn–Tucker (KKT) matrices are then formulated as

$$J = \begin{bmatrix} \mathcal{H} & A_{eq}^T \\ A_{eq} & \mathbf{0} \end{bmatrix} \quad (2.58)$$

$$\mathbf{F} = \begin{bmatrix} \mathbf{G} \\ B_{eq} \end{bmatrix} \quad (2.59)$$

where,

$$\mathcal{H} = A_R^T W_R A_R + A_Q^T W_Q A_Q + A_A^T \mathbf{P}_{k-N+1|k-N+1}^{-1} A_A \quad (2.60)$$

$$\mathbf{G} = A_R^T W_R B_R + A_Q^T W_Q B_Q + A_A^T \mathbf{P}_{k-N+1|k-N+1}^{-1} B_A \quad (2.61)$$

Lastly, the estimated deviations can be found, where in the presence of only equality constraints requires only one matrix factorization. Here,  $\Lambda = [\lambda_1^T, \lambda_2^T, \dots, \lambda_N^T]$  is a vector of Lagrange multipliers.

$$\begin{bmatrix} \Delta \mathbf{X} \\ \Lambda \end{bmatrix} = J^{-1} \mathbf{F} \quad (2.62)$$

The deviation  $\Delta \mathbf{X}$  from the current estimates can then be found from Eq. (2.62) leading to the new estimated state values. This processes is then repeated until a desirable state convergence or a max number of iterations is had. The optimal estimate  $\mathbf{x}_{k-N+1}$  at the last point in the window is then saved and the estimation window is propagated to the next time step by propagating the state  $\mathbf{x}_k$  and its covariance matrix  $\mathbf{P}_k$  using Eq. (2.19) and Eq. (2.20). Finally, if  $k < N$  the window is expanded by one time step but if  $k \geq N$  the window is slid one time step forward.

## 2.8 Chapter Summary

In this chapter an overview of the various mathematical notations used through this thesis was seen, as well, an overview of Van Loan's procedure used to discretize a continuous system, and an outline of each nonlinear state estimator. The key concepts of each algorithm is presented below:

- **EKF**: Applies the KF equations to a nonlinear system through linearizing the dynamic and measurement models around an estimate of the current state and covariance. The Jacobian matrix of the dynamic and measurement models is required.
- **UKF**: A set number of sigma points is propagated through the nonlinear system to better approximate the true state and its covariance. The algorithm used requires the discrete process noise covariance matrix found from Van Loan's procedure.
- **PF**: A set of particles is used to approximate the posterior distribution without making assumptions about the distribution. A bootstrap filter was implemented.
- **FLS**: A improved estimate of the state is found by using the EKF filtering equations but augmenting the state matrix to include a delay in estimates. The Jacobian matrix of the dynamic and measurement models is required.
- **MHE**: A weighted least squares optimization problem is solved over a sliding window of  $N$  time steps. A term denoted as the *arrival cost* is used to partially summarize all data that has left the window. The discrete process noise covariance matrix is required as a weight in the optimization problem. As well, the method used to solve the weight least squares problem requires the state transition matrix found from Van Loan's procedure.



## Chapter 3

# Hyperbolic Mars Approach

This chapter provides an overview of the hyperbolic Mars approach scenario that was used to compare the EKF, UKF, PF, FLS and MHE to be used for orbit determination. The scenario used observations of Mars and its moons to aid in state estimation. In previous work, Enright et al. [22] studied the feasibility of using an EKF-based navigation system during a hyperbolic Mars approach. The filter fused observations of Mars and its moons to provide orbit determination. Although effective, the EKF often showed an initial divergence due both the sensitive nature of the hyperbolic scenario and the EKF utilizing linearization's that may inadequately describe the dynamic and measurement models.

This work utilizes a previously defined framework derived in Enright et al. [22], but evaluates the convergence and accuracy of various algorithms to perform nonlinear estimation. Other studies have examined similar scenarios, Christian and Lightsey [23], used a EKF for autonomous OpNav in a planetary flyby of Venus. Franzese et al. [24], explored using an EKF for the LUMIO CubeSat mission. While, Paluszek et al. [25], implemented a UKF into a optical navigation system for use in a wide variety of missions.

In this chapter, Section 3.1 to Section 3.3 provide an overview of the operational scenario, state variables, and system dynamics. Section 3.4 overviews the measurements models used. While, Section 3.5 presents the results from a series of Monte Carlo (MC) trials where the performance of each state estimator was looked at. Refer to Enright et al. [22] for a more comprehensive derivation of the reference frames, state variables, dynamic and measurement models.

### 3.1 Reference Mission

Figure 3.1 presents the essential orbital geometry for a spacecraft approaching Mars on a hyperbolic trajectory. The scenario starts when the satellite enters the sphere of influence (SOI) of Mars, and begins on a planeto-centric hyperbolic trajectory. Table 3.1 defines a set of frames that will be used within this chapter. The spacecraft's trajectory was described in the inertial Frame- $P$ . It should be noted that this frame is fixed and based on the initial geometry of the approach.

TABLE 3.1: Hyperbolic Trajectory Frame Identification

Frame	Symbol	Centre	X-Axis	Z-Axis
J2000Ecliptic	$E$	Solar System Barycentre	VE projection into ecliptic	Normal to ecliptic
Periapse	$P$	Mars	Direction of periapse	Angular momentum direction of S/C
Orbit	$O$	Spacecraft	Outward radius	Angular momentum direction of S/C
Body	$B$	Spacecraft	Parallel to orbit normal	Convenient direction for viewing
Sensor	$S$	Spacecraft	Increasing columns	Boresight
Target	$T$	Spacecraft	Normal to boresight and planet vectors	Towards Planet

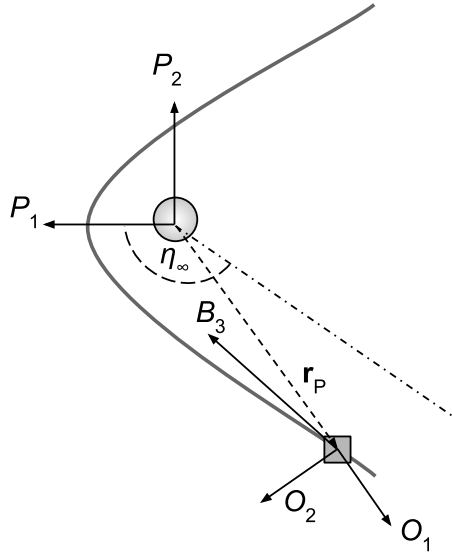


FIGURE 3.1: Hyperbolic trajectory showing the periapse, body and orbit frame.

The rotation matrices describing the transformation between each frame are listed below. For a hyperbolic trajectory the hyperbolic asymptote is found at an angle  $\eta_\infty$  from the  $P_1$  axis, with  $e$  being the eccentricity of the orbit. To ensure the spacecraft lies on the approach leg, the negative solution was taken.

$$\eta_\infty = -\arccos\left(-\frac{1}{e}\right) \quad (3.1)$$

It was assumed the scenario occurs at the aphelion of the heliocentric transfer ellipse, thus

$$\eta_{Sun} = \eta_{\infty} - \frac{\pi}{2} \quad (3.2)$$

Here  $\eta_{Sun}$  is the angle from the  $P_1$  axis to the direction vector of the Sun. This leads to the rotation from Frame- $E$  to Frame- $P$  as

$$\mathbf{C}_{PE} \equiv \mathbf{C}_3(\eta_{Sun} + \pi - \alpha_M) \mathbf{C}_1(\pi) \quad (3.3)$$

Where,  $\alpha_M$  is the right ascension of Mars in Frame- $E$ . The rotation from the periapse to orbital frame  $\mathbf{C}_{OP}$  is constructed from the position  $\mathbf{r}_P$  and velocity  $\mathbf{v}_P$  of the spacecraft. As  $\mathbf{C}_{OP}$  is a function of the spacecraft's position and velocity it does not remain constant over time. From below  $\mathbf{O}_{1P}$  is defined using the outward radius, and  $\mathbf{O}_{3P}$  the angular momentum of the spacecraft.

$$\mathbf{C}_{OP} = \begin{bmatrix} \mathbf{O}_{1P} & \mathbf{O}_{2P} & \mathbf{O}_{3P} \end{bmatrix}^T \quad (3.4)$$

where,

$$\mathbf{O}_{1P} = \frac{\mathbf{r}_P}{r}, \quad \mathbf{O}_{3P} = \frac{\mathbf{r}_P^\times \mathbf{v}_P}{\|\mathbf{r}_P^\times \mathbf{v}_P\|} \quad \text{and} \quad \mathbf{O}_{2P} = \mathbf{O}_{3P}^\times \mathbf{O}_{1P} \quad (3.5)$$

A mission dependent pointing rule was used to describe how the spacecraft was orientated within its orbit. For this study  $\mathbf{C}_{BO}$  represents a Mars pointing rule that keeps the limb of Mars a desired angle  $\xi$  off the center of the sensor boresight.

The angular radius of Mars, as seen from the spacecraft, is  $\rho$ . Thus,

$$\mathbf{C}_{BO} \equiv \mathbf{C}_1(\rho - \xi) \mathbf{C}_2\left(-\frac{\pi}{2}\right) \quad (3.6)$$

Lastly, looking at Figure 3.2 the rotation from the sensor to target frame can be found given that Mars center is at an angle  $\rho_c$  from the sensor boresight and in an azimuthal direction  $\theta_c$ .

$$\mathbf{C}_{TS} = \mathbf{C}_1(\rho_c) \mathbf{C}_3(\theta_c) \quad (3.7)$$

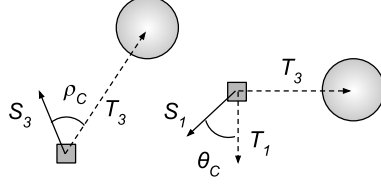


FIGURE 3.2: Rotation from Frame- $S$  to Frame- $T$ .

### 3.2 State Variables

The spacecraft's state is composed of the inertial position,  $\mathbf{r}_p$ , and velocity,  $\mathbf{v}_p$ , of the spacecraft, the sensor's unknown mounting error,  $\psi$ , with respect to a desired orientation, and the spacecraft's attitude error,  $\phi$ , with respect to a nominal reference orientation. Both  $\phi$  and  $\psi$  represent small angle error rotation vectors with rotation matrices  $\mathbf{C}(\phi)$  and  $\mathbf{C}(\psi)$ . This leads to the  $12 \times 1$  state vector:

$$\mathbf{x} = \begin{bmatrix} \mathbf{r}_P \\ \mathbf{v}_P \\ \psi \\ \phi \end{bmatrix} \quad (3.8)$$

The rotation from the body to sensor frame  $\mathbf{C}_{SB}$  is a combination of the reference orientation  $\bar{\mathbf{C}}_{SB}$  and sensor misalignment error.

$$\mathbf{C}_{SB} = \mathbf{C}(\psi) \bar{\mathbf{C}}_{SB} \quad (3.9)$$

The transformation of Frame- $P$  to Frame- $B$  is shown below. Here  $\mathbf{C}_{BO}$  refers to a reference orientation that is found from the pointing rule and derived from the estimate state vector, meaning  $\mathbf{C}_{BO} = \mathbf{C}_{BO}(\hat{\mathbf{x}})$ .

$$\mathbf{C}_{BP} = \mathbf{C}(\phi) \bar{\mathbf{C}}_{BP} = \mathbf{C}(\phi) \mathbf{C}_{BO} \mathbf{C}_{OP} \quad (3.10)$$

This means that the rotation from the periaipse to the sensor frame is equal to

$$\mathbf{C}_{SP} = \mathbf{C}_{SB} \mathbf{C}_{BP} = \mathbf{C}(\psi) \bar{\mathbf{C}}_{SB} \mathbf{C}(\phi) \bar{\mathbf{C}}_{BP} \quad (3.11)$$

### 3.3 System Dynamics

The spacecraft was assumed to undergo unpowered flight throughout the mission, thus accelerations on the satellite are caused by the gravitational force of Mars and third body perturbations. All accelerations were modeled in the periapse frame with the true spacecraft state modeled as:

$$\begin{bmatrix} \dot{\mathbf{r}}_p \\ \dot{\mathbf{v}}_p \end{bmatrix} = \begin{bmatrix} \mathbf{v}_p \\ -\frac{\mu_{Mars}}{r^3} \mathbf{r}_p + \mathbf{a}_{Disturbance} \end{bmatrix} \quad (3.12)$$

The modeled third body perturbations represent the gravitational forces caused by the Sun and Jupiter. Here,  $\Delta \mathbf{r}_{DS}$  is the vector from the disturbing body to the spacecraft, and  $\Delta \mathbf{r}_{DM}$  is from the disturbing body to Mars.

$$\mathbf{a}_{Disturbance} = - \sum \mu_{Disturbing} \left( \frac{\Delta \mathbf{r}_{DS}}{\Delta r_{DS}^3} - \frac{\Delta \mathbf{r}_{DM}}{\Delta r_{DM}^3} \right) \quad (3.13)$$

For the test scenario, the truth simulation will include these third body disturbances, but the onboard filter will assume a two body problem. A process noise  $\mathbf{w}_v$  with covariance  $Q_v$  can be used to partially remedy this worst case modeling error.

To avoid the need to explicitly model an attitude controller, it was assumed the spacecraft could closely track a reference orientation,  $\bar{\mathbf{C}}_{BP}$ , with a relatively small error described by  $\phi$ . The dynamics of the attitude error of the satellite were modeled as a first-order system driven by random noise. Here,  $\mathbf{w}_\phi$  represents a zero-mean Gaussian random variable with known covariance  $Q_\phi$ .

$$\dot{\phi} = -\frac{\phi}{\tau} + \mathbf{w}_\phi \quad (3.14)$$

Lastly, the mounting error remains constant over time, meaning  $\dot{\psi} = 0$ . In the MHE this was considered an equality constraint. This leads to the continuous model to be used in the navigation system

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}, t) + \mathbf{w} = \begin{bmatrix} \mathbf{v}_P \\ -\frac{\mu_{Mars}}{r^3} \mathbf{r}_P \\ \mathbf{0}_{3 \times 1} \\ -\frac{\phi}{\tau} \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix} \begin{bmatrix} \mathbf{w}_v \\ \mathbf{w}_\phi \end{bmatrix} \quad (3.15)$$

The dynamics Jacobian is then given by

$$\mathbf{F}(\mathbf{x}(t)) = \frac{\partial f}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\mathbf{x}(t)} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \frac{\mu}{r^3} \left( \frac{3}{r^2} \mathbf{r}_P \mathbf{r}_P^T - \mathbf{I}_{3 \times 3} \right) & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & -\frac{1}{r} \mathbf{I}_{3 \times 3} \end{bmatrix} \quad (3.16)$$

and the continuous process noise covariance matrix is:

$$Q = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & Q_v & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & Q_\phi \end{bmatrix} \quad (3.17)$$

### 3.4 Measurement Models

The measurement models used were derived in Enright et al. [22], this section provides a brief description of each measurement but refer to Appendix B for each measurement model. The measurements are defined using  $\tilde{\mathbf{z}}$ , and the estimated measurement as  $\hat{\mathbf{z}}$ . During the approach five measurements were considered:

1. The attitude of the spacecraft
2. The direction to Mars moons Phobos and Deimos
3. The displacement vector to Mars
4. The phase angle between the spacecraft, the Sun and Mars
5. The orientation of the terminator ellipse

The attitude of the spacecraft was defined using an attitude error pseudomeasurement, defined as  $\Delta$ , to represent an error rotation vector between the reference and actual attitude. The direction vector,  $\mathbf{s}_S$ , to Mars moons can be found by their coordinates on the star trackers detector plane. Figure 3.3 provides a basic visual representation of these measurements.

A pseudomeasurement was used to represent the displacement vector to Mars,  $\mathbf{r}_s$ , which can be found from the size and position of the Mars disk in the field of view of the sensor, as seen in

Figure 3.4. Due to the setup of the coordinate frames the displacement vector is simply defined as the rotation of spacecrafts position into Frame- $S$ ,  $\mathbf{r}_S = -\mathbf{C}_{SP}\mathbf{r}_P$ .

The phase angle,  $\chi$ , between the spacecraft, the Sun and Mars can in practice be found through the illuminated fraction of Mars. Lastly, the terminator angle,  $\beta$ , is defined as the orientation of the terminator ellipse on the image disk of Mars. These last two measurements can be seen in Figure 3.5.

At each time step the availability of each measurement may change based on the state of the spacecraft, and that of the targets. The measurement availability for the hyperbolic scenario is seen in Figure 3.6. The availability of Phobos and Deimos depends on the scenario start date. As well, the loss of the phase and terminator angles near periapse is caused by the Mars disk in the FOV of the sensor becoming too large to obtain measurements.

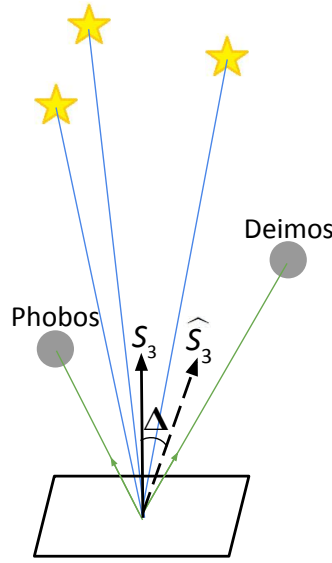


FIGURE 3.3: Attitude measurement, and Phobos and Deimos direction measurements.

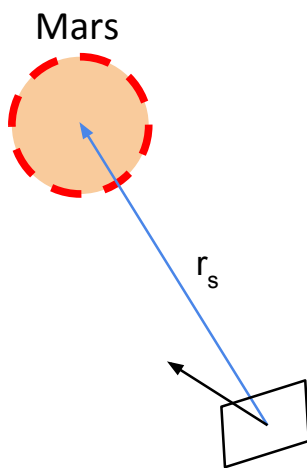


FIGURE 3.4: Mars displacement vector.

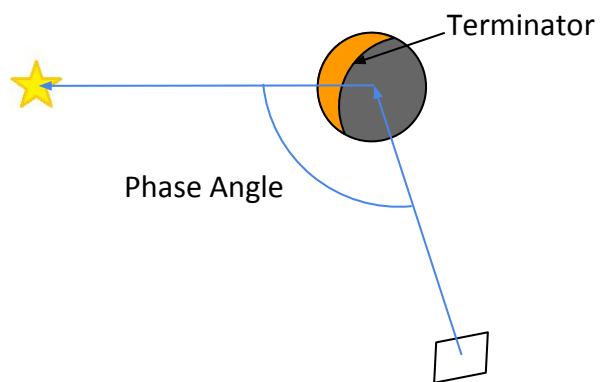


FIGURE 3.5: The terminator and phase angles.



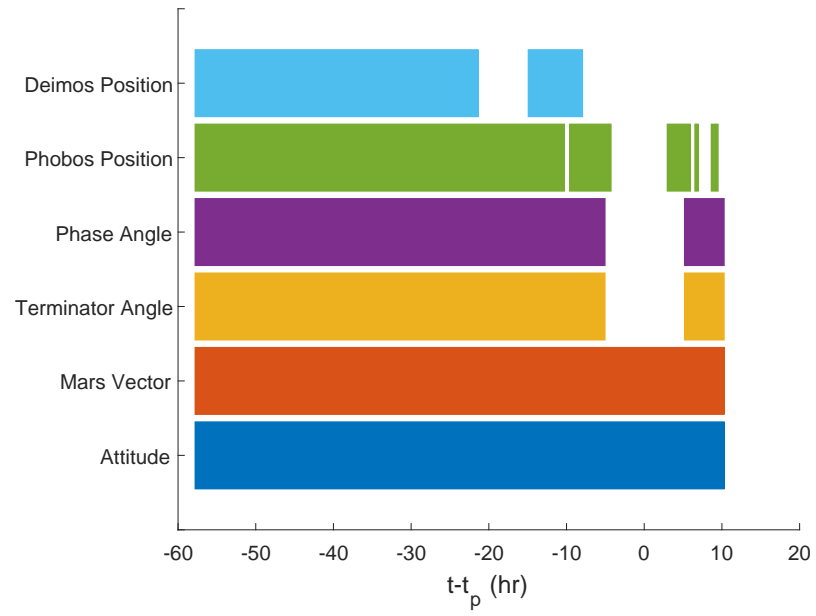


FIGURE 3.6: Measurement availability for the hyperbolic Mars approach [22].

### 3.5 Simulation Results

In this section each state estimator was compared using the hyperbolic Mars approach. Only one trajectory was considered, where the parameters used to initialize the simulation, and the spacecraft's trajectory are outlined in Table 3.2. Assessing various trajectories may provide better insight into the expected performance of each estimator, however without a specific mission in mind it can be impossible to assess the performance over every possible trajectory. However, a Monte Carlo (MC) simulation was used to assess the performance statistics of each estimator over a wide range of initial state errors, where each trials initial state error was a random variable described by the initial covariance matrix.

For the simulation each estimator was tuned by varying each tunable parameter over a series of MC trials until improvements in state errors were no longer seen. From this method the UKF was simulated using  $\alpha = 1$  and  $\kappa = 0$ , and the PF was initialized with 2000 particles. To find the window length for the MHE and FLS a balance needs to be struck between improved estimates, a longer delay, and increased computational time. This balance can be made by both tuning the window length and measurement update time while staying within the limits of the sensor. The limits include the time needed to capture and process the optical images. Through testing a window length of 15 was chosen. Increasing the window length generally reduced errors from a size of 1 to 15 but larger values yielded minimal additional improvements.

TABLE 3.2: Hyperbolic Trajectory Scenario Parameters

Scenario Start Date	2019 Jan 15 01:55:00 (UTC)
Initial Mars-Spacecraft Distance	571000 km
Initial Mars-Centric Speed	2.6694 km/s
Right Ascension of Sun (in $P$ frame)	$-0.5236$ rad
Initial Eccentricity	2.0
Initial Semi-Major Axis	6139.7612 km
Position uncertainty ( $1 - \sigma$ ), each axis	100 km
Velocity uncertainty ( $1 - \sigma$ ), each axis	0.1 km/s
Sensor mounting uncertainty, each axis	0.1 deg.
Attitude error uncertainty, each axis	1 deg.
Cross-axis attitude error ( $1 - \sigma$ )	$1 \times 10^{-3}$ deg.
Roll attitude error ( $1 - \sigma$ )	$1 \times 10^{-2}$ deg.
Measurement update time	400 s
Attitude error time constant, $\tau$	600 s
Attitude error process noise	$\mathbf{Q}_\phi = (10^{-8})I_{3 \times 3} \text{ rad}^2/\text{s}^2$
Velocity process noise (estimator only)	$\mathbf{Q}_v = (10^{-16})I_{3 \times 3} \text{ m}^2/\text{s}^4$

A single run trial was first performed to provide a basic performance analysis. For small errors in initial state estimates the EKF, UKF, FLS, and MHE behaved similarly. The PF was able

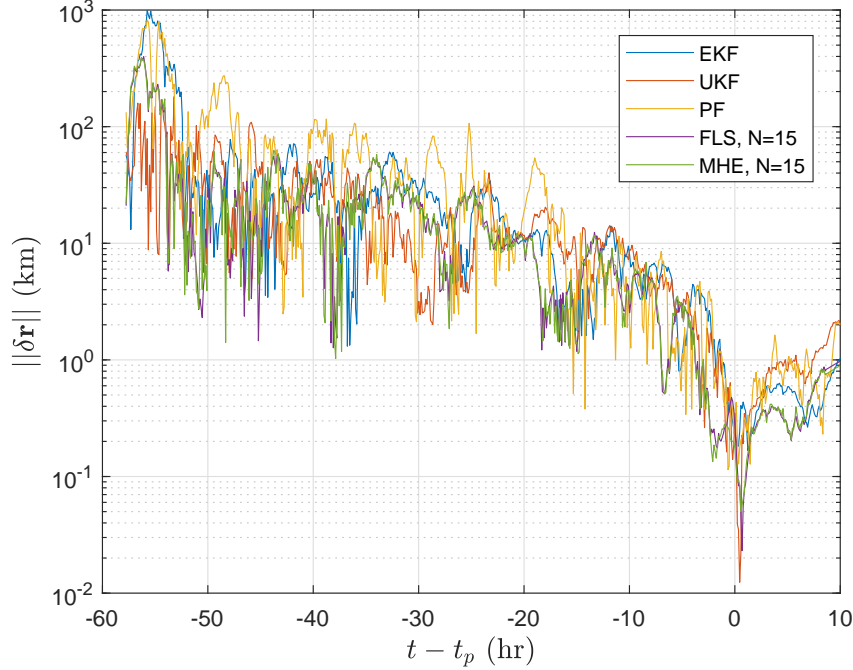


FIGURE 3.7: Hyperbolic orbit position error over single run.

to provide similar errors but the relative performance was inconsistent. As the initial state error was increased a separation between filter performance can be seen. For example, given an arbitrarily chosen initial state error of  $[-145 \ 70 \ -20]^T$  km in position,  $[0.18 \ 0.2 \ -0.15]^T$  km/s in velocity, and an initial estimate of  $0^\circ$  for the attitude and sensor misalignment error along each axis Figures 3.7 to 3.9 show the errors in position, velocity, and the angle errors respectively. For the FLS and MHE the optimal estimate at the last point in the estimation window is plotted.

An initial divergence is seen in position estimates before converging to the minimal value at periapsis. For this specific case the UKF provides the largest reduction in initial divergence and periapsis error. Followed by the FLS and MHE. This is not seen in the velocity estimates where each estimator slowly converges over time. Here the UKF provides a lower initial error, while the FLS and MHE achieve lower errors near periapsis. The PF is able to provide errors on par with the other estimators but the estimation errors are variable. Looking at the sensor misalignment error the EKF, UKF, FLS, and MHE provide similar results for this single case. While, the PF appears to exhibit erratic behavior. This is caused by the sensor misalignment error being a constant value. Lastly, each estimator provides similar results in regards to the attitude error, this is due to the dynamics being dominated by the process noise.

To better assess the system performance a MC simulation of 1000 trials was run to measure the error statistics of each estimator over a wide range of initial state estimates. From the MC

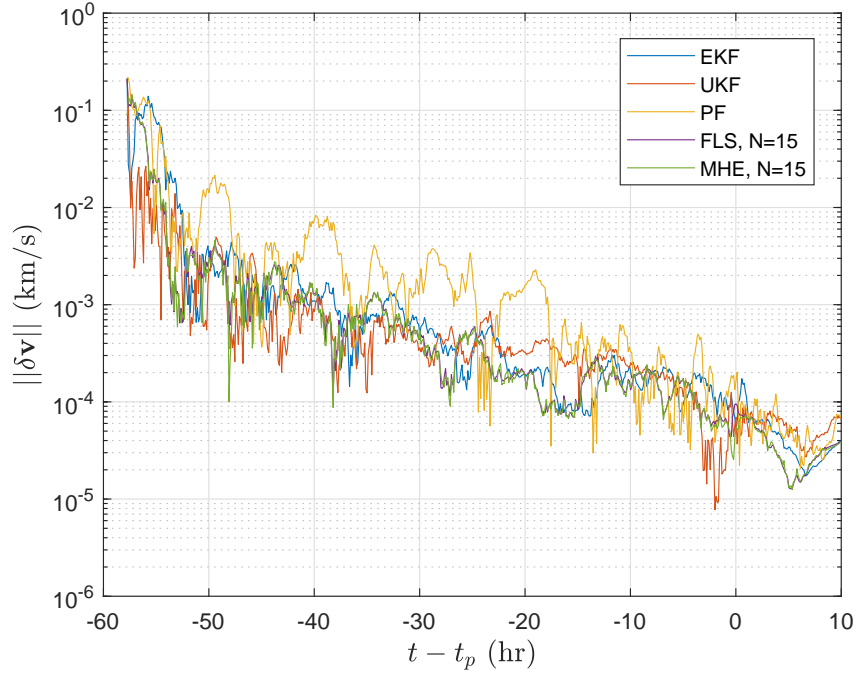


FIGURE 3.8: Hyperbolic orbit velocity error over single run.

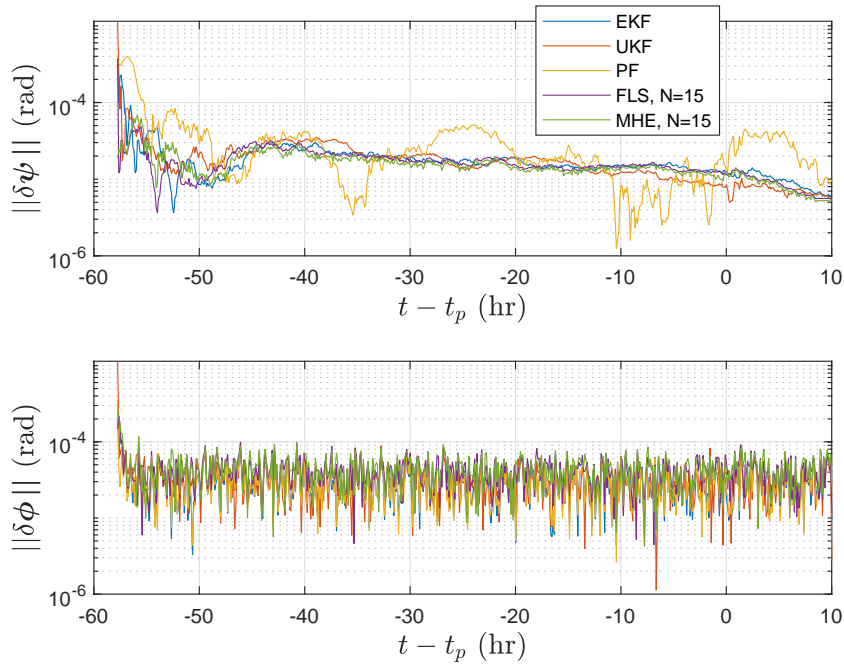


FIGURE 3.9: Hyperbolic orbit sensor misalignment and attitude error over a single run.

simulation Figure 3.10 presents the mean position error over the entire simulation, Figure 3.11 the mean velocity error, and Figure 3.12 the mean sensor misalignment and attitude error. All points represent the mean RSS error. Looking at the position estimates there is an initial divergence in error due to both the sensitivity of the hyperbolic trajectory, and large measurement noises due to the distance from Mars. The sensitive nature of the hyperbolic trajectory implies that small changes in the initial conditions can lead to a very different final trajectory. After the initial divergence, the state estimators slowly converge before a rapid convergence is seen as the spacecraft enters a regime where the measurements improve significantly. Minimum errors are then achieved around periapse.

Comparing each estimator, the MHE and FLS outperform both the EKF, UKF, and PF with respect to the position error, reducing the initial divergence and improving the periapse error. However, the MHE does provide a lower initial and periapse error than the FLS. The EKF and UKF perform similarly, although the UKF does provide a lower error near periapse. The reason for the similar performance at the beginning of the scenario is the sensitive nature of the hyperbolic scenario combined with the initial error in state estimates, and large measurement noises due to the distance from Mars. Lastly, the PF provides the highest estimation errors across the entire time span. To ensure the maximum divergence results are not caused by a few bad trials that skew the end result, Figure 3.13 presents the cumulative distribution function (CDF) of the maximum errors seen throughout the MC trials. Again similar results are seen for the EKF and UKF, and the MHE and FLS, where the latter two estimators are very likely to achieve a lower divergence. The PF again provides the worst performance.

In contrast, the velocity error does not exhibit an initial divergence, however an initial drop in errors is had before a more consistent convergence to periapse. When comparing each filter similar trends are seen as the position error. With the FLS and MHE improving on the estimation error, benefits from the UKF compared to the EKF are seen near periapse and the PF provides the highest estimation errors. Looking at the sensor misalignment the MHE produces a lower error up to periapse, where the FLS overtakes it. The UKF has the worst initial estimates before converging below the EKF near periapse. However, besides the PF each estimator displays similar trends with minimal difference in error between them. This could be due to the sensor misalignment being a constant. Lastly, all filters provided similar results in regards to the attitude error, again this is due to the dynamics being dominated by the process noise.

To add, Table 3.3 presents statistics on key points throughout the trials. All errors represent the RSS error across the trials. The mean errors are the average errors across the trials and each time span. The maximum position error is the max RSS error across all trials and takes a look at the reduction in initial divergence in position. The velocity, attitude and sensor misalignment

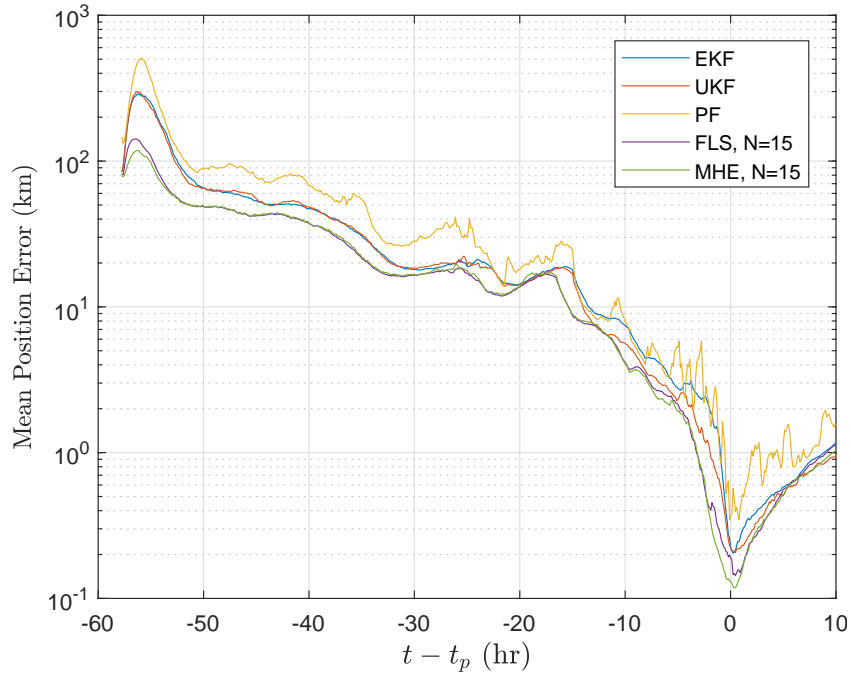


FIGURE 3.10: Mean position error in MC trials.

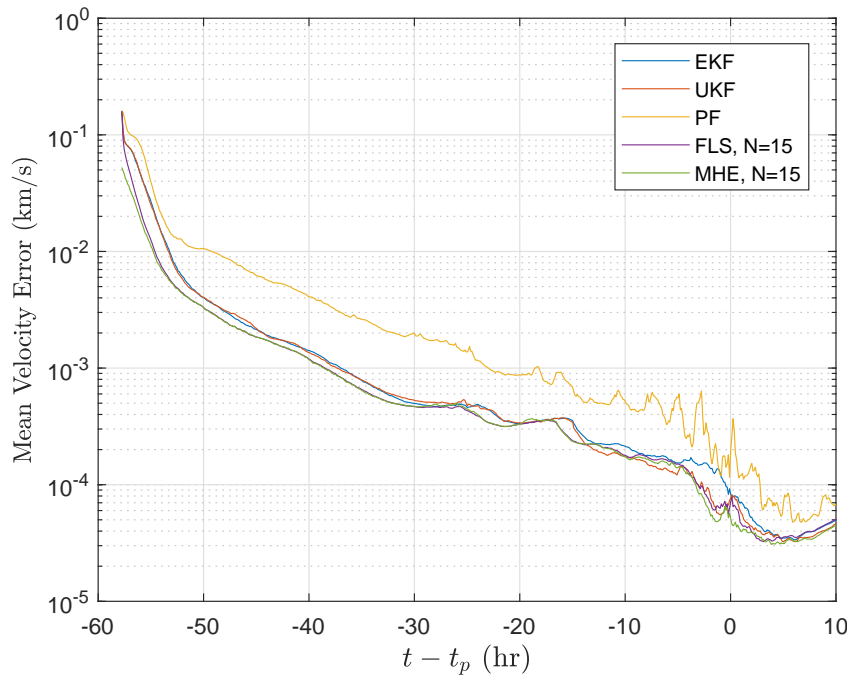


FIGURE 3.11: Mean velocity error in MC trials.

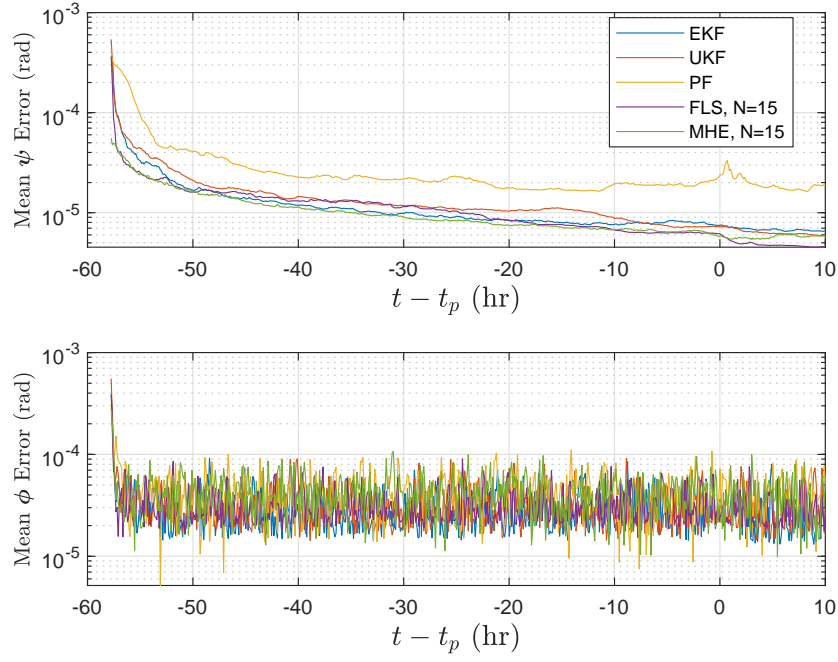


FIGURE 3.12: Mean attitude and sensor misalignment errors in MC trials.

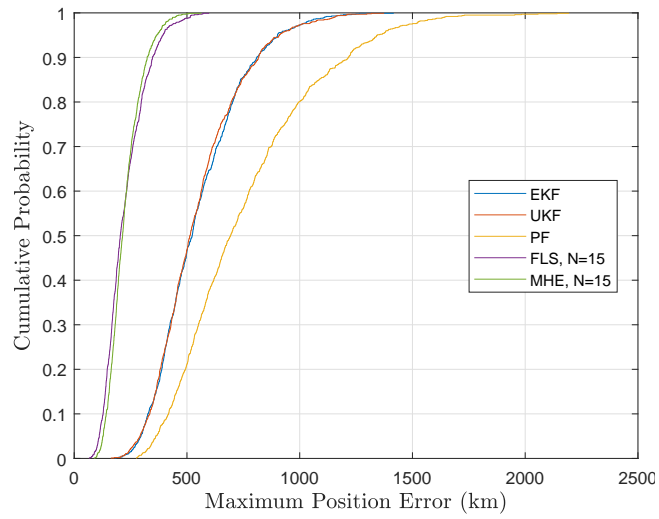


FIGURE 3.13: CDF of maximum errors in MC trials.

TABLE 3.3: Hyperbolic Trajectory Simulation Results

	EKF	UKF	PF	FLS	MHE
Mean Position Error (km)	36.47	35.96	52.00	23.75	23.02
Mean Velocity Error (m/s)	3.89	3.88	6.60	2.61	2.03
Mean Psi Error ( $\times 10^{-5}$ rad)	1.38	1.58	3.25	1.21	1.04
Mean Phi Error ( $\times 10^{-5}$ rad)	3.03	3.33	4.39	3.26	4.33
Mean Periapse Error (km)	0.22	0.22	0.34	0.19	0.13
Max Position Error ( $\times 10^3$ km)	1.42	1.37	2.32	0.60	0.56

maximum error occurs at start of the simulation due to the initial state error and thus, provides no information on estimator performance.

Looking at the table the EKF and UKF provide similar results, while, the PF provides the highest errors for each category. The FLS and MHE improve the position and velocity in the related categories, providing a reduction in initial divergence in position, and lowering the periapse error. Looking at the sensor misalignment error again the FLS and MHE provide the best results, with the PF providing the largest error followed by the UKF. With regards to the attitude error no improvement is seen from the reference EKF.

It is evident that the PF estimates are not comparable with the other state estimators. It was found that the PF was sensitive to the initial distribution, if no particles fall within “good” regions the PF estimates were seen to diverge. More particles could be used to help mitigate the problem but at the cost of a higher computation time. As well, from observing velocity estimates it is thought that as velocity does not appear in the likelihood function these particles slowly converge based on the propagated values of the position estimates rather than direct information on how good the estimate is, leading to a slower convergence. Lastly, there is difficulty estimating the sensor misalignment, as it is a constant value any variation in the particles comes from the roughening step. It was found that this caused erratic behavior in the estimates, seen in Figure 3.9, but the quality of the estimates is again dependent on the initial distribution. Further tuning of the PF may improve estimates however.

Lastly, the mean computation time over the MC trials was looked at. Comparing the computational time of each estimator relative to the EKF the UKF, PF, FLS and MHE required approximately 3.3-, 67.7-, 4.1- and 8.0- times the computation time for this scenario. The improved state estimates from the FLS and MHE thus come at the cost of an increased computation time. As well, less particles could be used to improve the PF’s run time but if not enough particles are used the filters performance severely degrades.



## 3.6 Chapter Summary

In this chapter a hyperbolic Mars approach was used as a reference missions to compare the EKF, UKF, PF, FLS, and MHE. This work used a previously derived scenario and measurement models that can be found in Enright et al. [22]. In the scenario four state variables were considered: the position and velocity of the spacecraft, the sensor's unknown mounting error with respect to a desired orientation, and the attitude error of the spacecraft with respect to a nominal reference orientation. Looking at the dynamics of each state variables the spacecrafts trajectory was modeled using gravitational perturbations from the Sun and Jupiter, while the filter dynamics only consider a two body problem. The sensor misalignment error remains constant over time, and the spacecrafts attitude error was modeled as a first order system driven by random noise. In the simulation five previously derived OpNav observations were considered: the attitude of the spacecraft, the direction to Mars moons Phobos and Deimos, the displacement vector to Mars, the phase angle between the spacecraft, the Sun and Mars, and the orientation of the terminator ellipse. A basic performance analysis of each state estimator was first performed by looking at their performance over a single trial. A series of MC simulations was then used to access each state estimators performance over a wide range of initial conditions. Looking at the position and velocity estimates in the MC trials it was seen that the FLS and MHE improved the mean RSS error over the entire time span. Improving on the initial divergence and periapse error. The UKF had slight improvements over the EKF, while the PF had the worst estimates. Besides the PF each state estimator converged to similar values with respect to the sensor misalignment error, and each state estimator behaved similarly with respect to the attitude error. The mean computation time for each state estimator with respect the EKF was also looked at. It was seen that the improved estimates from the FLS and MHE come at the cost of a higher computation time, with the FLS taking about half the time of the MHE. Besides the EKF, the UKF had the quickest run time and the PF by far the worst, taking about 67.7 times longer than the EKF.

## Chapter 4

# Elliptical Orbit around Mars

In this chapter two reference scenarios are presented to compare each state estimator. Section 4.1 takes a look at an elliptical orbit around Mars using similar dynamic and measurements models as seen in the previous chapter, therefore only deviations from what has already been presented will be seen within this section. Section 4.2 then extends the framework of the elliptical scenario to compare the performance of each estimator in timing an orbital maneuver to raise the periaipse of the spacecraft out of the Martian atmosphere after an aerocapture maneuver. In each scenario a set of MC trials was used to assess the performance of the state estimators.

### 4.1 Elliptical Orbit

For this scenario a spacecraft was placed into an elliptical orbit around Mars, seen in Figure 4.1. Table 4.1 outlines the set of frames that were used for this scenario. Again the spacecraft's trajectory was described in Frame- $P$ . Here Frame- $P$  was considered an inertial frame located at the origin of Mars with the  $P_3$  axis aligned with the planet's spin axis, the  $P_1$  axis along the equatorial plane in the direction of periaipse and the  $P_2$  axis completes the right handed frame. The  $P_1$  and  $P_2$  axes are arbitrary, just as long as the  $P_3$  axis is aligned with Mars spin axis. Frames  $O$ ,  $B$ ,  $S$ , and  $T$  are defined the same as in the hyperbolic scenario. As well, the rotation between each frame has been defined in Section 3.1.

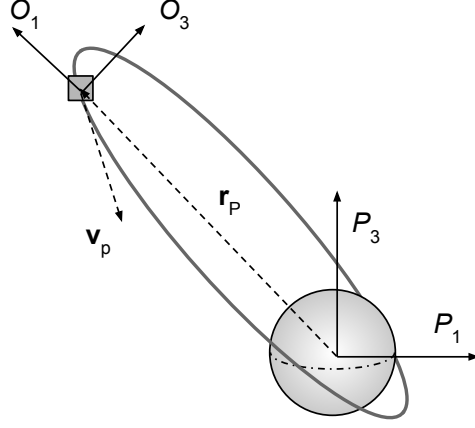


FIGURE 4.1: Elliptical trajectory showing the periapse and orbital frames.

TABLE 4.1: Elliptical Orbit Frame Identification

Frame	Symbol	Centre	X-Axis	Z-Axis
Mars inertial frame	$P$	Mars	Along the equatorial plane and in the direction of periapse	Mars spin axis
Orbit (RTN)	$O$	Spacecraft	Outward radius	Angular momentum direction of S/C in orbit
Body	$B$	Spacecraft	Parallel to orbit normal	Convenient direction for viewing
Sensor	$S$	Spacecraft	Increasing columns	Boresight
Target	$T$	Spacecraft	Normal to boresight and Mars vector	Towards Mars

#### 4.1.1 System Dynamics

In this scenario the spacecraft was again assumed to undergo unpowered flight, with accelerations on the satellite from gravitational perturbations due to Mars mass distribution. The accelerations up to the fourth harmonic were used in the truth model, denoted as  $\mathbf{a}_{J_2}$ ,  $\mathbf{a}_{J_3}$ , and  $\mathbf{a}_{J_4}$ . Therefore the true position and velocity were propagated as

$$\begin{bmatrix} \dot{\mathbf{r}}_P \\ \dot{\mathbf{v}}_P \end{bmatrix} = \begin{bmatrix} \mathbf{v}_P \\ -\frac{\mu_{mars}}{r^3} \mathbf{r}_P + \mathbf{a}_{J_2} + \mathbf{a}_{J_3} + \mathbf{a}_{J_4} \end{bmatrix} \quad (4.1)$$

Due to Mars not being a perfect sphere there are gravitational perturbations due to its actual mass distribution. Using Mars zonal harmonics up to  $J_4$  these perturbations are seen below and were derived in Schaub and Junkins [26]. Mars zonal harmonics shown in Table 4.2. The distances  $r_x$ ,  $r_y$ , and  $r_z$  must be defined with respect to a frame where the  $z$ -axis is that of the

spin axis, therefore they are defined with respect to Frame- $P$  of the elliptical orbit.

$$\mathbf{a}_{J_2} = -\frac{3}{2} \frac{J_2 \mu_{Mars} R_{mars}^2}{r^4} \begin{bmatrix} (1 - 5(\frac{r_z}{r})^2) \frac{r_x}{r} \\ (1 - 5(\frac{r_z}{r})^2) \frac{r_y}{r} \\ (3 - 5(\frac{r_z}{r})^2) \frac{r_z}{r} \end{bmatrix} \quad (4.2)$$

$$\mathbf{a}_{J_3} = -\frac{1}{2} \frac{J_3 \mu_{Mars} R_{mars}^3}{r^5} \begin{bmatrix} 5(7(\frac{r_z}{r})^3 - 3(\frac{r_z}{r})) \frac{r_x}{r} \\ 5(7(\frac{r_z}{r})^3 - 3(\frac{r_z}{r})) \frac{r_y}{r} \\ 3(10(\frac{r_z}{r})^2 - \frac{35}{3}(\frac{r_z}{r})^4 - 1) \end{bmatrix} \quad (4.3)$$

$$\mathbf{a}_{J_4} = -\frac{5}{8} \frac{J_4 \mu_{Mars} R_{mars}^4}{r^6} \begin{bmatrix} (3 - 42(\frac{r_z}{r})^2 + 63(\frac{r_z}{r})^4) \frac{r_x}{r} \\ (3 - 42(\frac{r_z}{r})^2 + 63(\frac{r_z}{r})^4) \frac{r_y}{r} \\ -(15 - 70(\frac{r_z}{r})^2 + 63(\frac{r_z}{r})^4) \frac{r_z}{r} \end{bmatrix} \quad (4.4)$$

TABLE 4.2: Mars Zonal Harmonic Coefficients [27]

Harmonics	Value
$J_2$	$1.955\,45 \times 10^{-3}$
$J_3$	$3.144\,98 \times 10^{-5}$
$J_4$	$-1.537\,74 \times 10^{-5}$

In the navigation system only the perturbations due to the  $J_2$  term will be considered, and a process noise will be added to account for the modeling error.

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}, t) + \mathbf{w} = \begin{bmatrix} \mathbf{v}_P \\ -\frac{\mu_{Mars} \mathbf{r}_P}{r^3} + \mathbf{a}_{J_2} \\ \mathbf{0}_{3 \times 1} \\ \frac{-\phi}{\tau} \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix} \begin{bmatrix} \mathbf{w}_v \\ \mathbf{w}_\phi \end{bmatrix} \quad (4.5)$$

The state Jacobian is then given by

$$\mathbf{F}(\mathbf{x}(t)) = \frac{\partial f}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\mathbf{x}(t)} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \frac{\mu}{r^3} \left( \frac{3}{r^2} \mathbf{r}_P \mathbf{r}_P^T - \mathbf{I}_{3 \times 3} \right) + \frac{\partial \mathbf{a}_{J_2}}{\partial r} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & -\frac{1}{\tau} \mathbf{I}_{3 \times 3} \end{bmatrix} \quad (4.6)$$

where,

$$\frac{\partial \mathbf{a}_{J_2}}{\partial r} = \frac{3J_2 R_{mars}^2 \mu_{Mars}}{2r^9} \times \begin{bmatrix} 4r_x^4 + 3r_x^2 r_y^2 - 27r_x^2 r_z^2 - r_y^2 + 3r_y^2 r_z^2 + 4r_z^4 & 5r_x r_y (r_x^2 + r_y^2 - 6r_z^2) & 5r_x r_z (3r_x^2 + 3r_y^2 - 4r_z^2) \\ 5r_x r_y (r_x^2 + r_y^2 - 6r_z^2) & 3r_x^2 r_y^2 - r_x^4 + 3r_x^2 r_z^2 + 4r_y^4 - 27r_y^2 r_z^2 + 4r_z^4 & 5r_y r_z (3r_x^2 + 3r_y^2 - 4r_z^2) \\ 5(3r_x^2 + 3r_y^2 - 4r_z^2) & 5(3r_x^2 + 3r_y^2 - 4r_z^2) & 3r_x^4 + 6r_x^2 r_y^2 - 24r_x^2 r_z^2 + 3r_y^4 - 24r_y^2 r_z^2 + 8r_z^4 \end{bmatrix} \quad (4.7)$$

Finally, the continuous process noise covariance matrix is equal to

$$Q = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & Q_v & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & Q_\phi \end{bmatrix} \quad (4.8)$$

#### 4.1.2 Measurement Models

The measurement models used are the same as the hyperbolic scenario, refer to Section 3.4. As well, the pointing rule used to orientate the spacecraft was the same as in the hyperbolic scenario, where the aim was to keep the limb of Mars a desired angle off the sensor boresight. The five measurements considered were:

1. The attitude of the spacecraft
2. The direction to Mars moons Phobos and Deimos
3. The displacement vector to Mars
4. The phase angle between the spacecraft, the Sun and Mars
5. The orientation of the terminator ellipse

Although each measurement was initially considered during the simulation the availability of the measurements was dependent on the orbit and time of the simulation. For example, given two orbits both with an inclination of  $75^\circ$ , periapsis of 300 km, and apoapsis of 10,000 and 72,000 km, Figures 4.2a and 4.2b show the measurement availability from a single trial. Further parameters of the simulation can be seen in the next section, Section 4.1.3. Looking at Figure 4.2a due to how close the spacecraft is to Mars only the Mars displacement vector and attitude measurements are seen. The Mars displacement vector is then lost near periapse due to the size

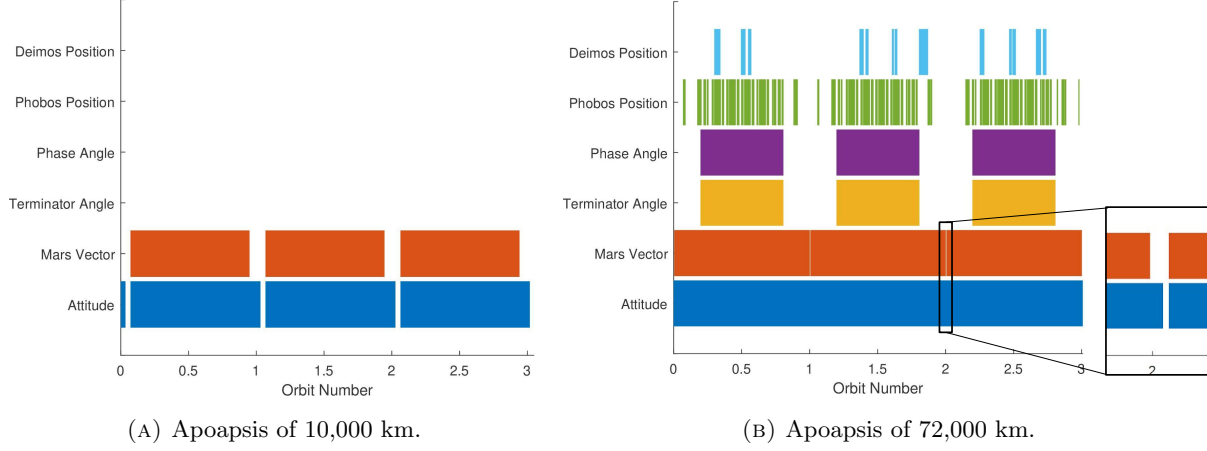


FIGURE 4.2: Measurement availability for an elliptical orbit.

of Mars in the FOV of the sensor. The attitude measurements are lost after periapse due to where the satellite was pointed, at this point the Sun falls within an exclusion angle of  $35^\circ$  off the sensor boresight meaning all measurements are lost. The time of occurrence and duration of being unavailable does differ from orbit to orbit due to the third body perturbations however. Looking at Figure 4.2b as the apoapsis is increased more measurements become available. The phase and terminator angles can be measured as the entire Mars disk is now in the FOV of the sensor. As well, the displacement vector and attitude measurements are lost due to the same reasons as in the previous orbit. Lastly, the availability of Phobos and Deimos depends on the scenario start date and for the most part if the moons lie within the FOV of the sensor. The direction measurements were not seen for the first simulation but changing the orbit parameters, start time or simulation length results in some availability.

#### 4.1.3 Simulation Results

For this scenario Table 4.3 provides an overview of the parameters used. A single run trial was first looked at to provide a basic performance analysis. Given an arbitrarily chosen initial state error of  $[0.8 \ 21 \ 11]^T$  km in position,  $[0.011 \ 0.012 \ -0.004]^T$  km/s in velocity, and an initial estimate of  $0^\circ$  for the attitude and sensor misalignment error along each axis Figure 4.3 presents the position error, Figure 4.4 the velocity error, and Figure 4.5 the attitude and sensor misalignment errors over the single run. The y-axis shows the root sum square (RSS) error, while the x-axis the number of orbits performed. The simulation spans just over three orbits, where at the start of the simulation the spacecraft was initialized at periapse. As seen in the previous section, near periapse no measurements are available and why an increase in uncertainty can be seen at the start of the simulation and once an orbit is complete. Looking

TABLE 4.3: Elliptical Orbit Scenario Parameters

Scenario Start Date	2019 Jan 15 01:55:00 (UTC)
Periapsis	300 km
Apoapsis	10,000 km/s
Inclination	75 deg
RAAN	0 deg
Argument of Periapsis	0 deg
Position uncertainty $(1 - \sigma)$ , each axis	10 km
Velocity uncertainty $(1 - \sigma)$ , each axis	0.01 km/s
Sensor mounting uncertainty, each axis	0.1 deg.
Attitude error uncertainty, each axis	1 deg.
Cross-axis attitude error $(1 - \sigma)$	$1 \times 10^{-3}$ deg.
Roll attitude error $(1 - \sigma)$	$1 \times 10^{-2}$ deg.
Measurement update time	100s
Attitude error time constant, $\tau$	600 s
Attitude error process noise	$\mathbf{Q}_\phi = (10^{-8})\mathbf{I}_{3 \times 3} \text{ rad}^2/\text{s}^2$
Velocity process noise (estimator only)	$\mathbf{Q}_v = (3 \times 10^{-14})\mathbf{I}_{3 \times 3} \text{ m}^2/\text{s}^4$

at the position estimates each estimator converges to similar errors, this was observed for both small and larger initial state errors. However, the MHE does provide the quickest convergence, while the PF the worst. This was also observed for the velocity estimates. Regarding the sensor misalignment errors for this case the FLS convergences to the lowest value, while the MHE, EKF and UKF each converged to similar values at the end of the simulation, however this result tended to be trial dependent. Similar to results seen in the hyperbolic trajectory the PF provided the worst estimates due to the sensor misalignment error being a constant value. For a constant value any variation in the estimates comes from the roughening step, this was seen to cause erratic behavior in the estimates. Lastly, each estimator provided similar errors with respect to the attitude error.

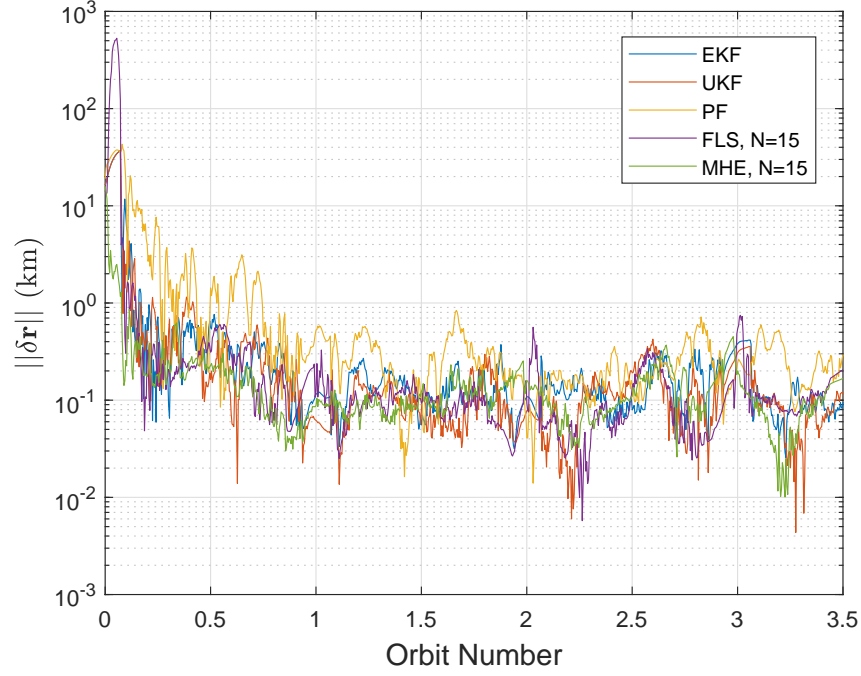


FIGURE 4.3: Elliptical orbit position error over single run.

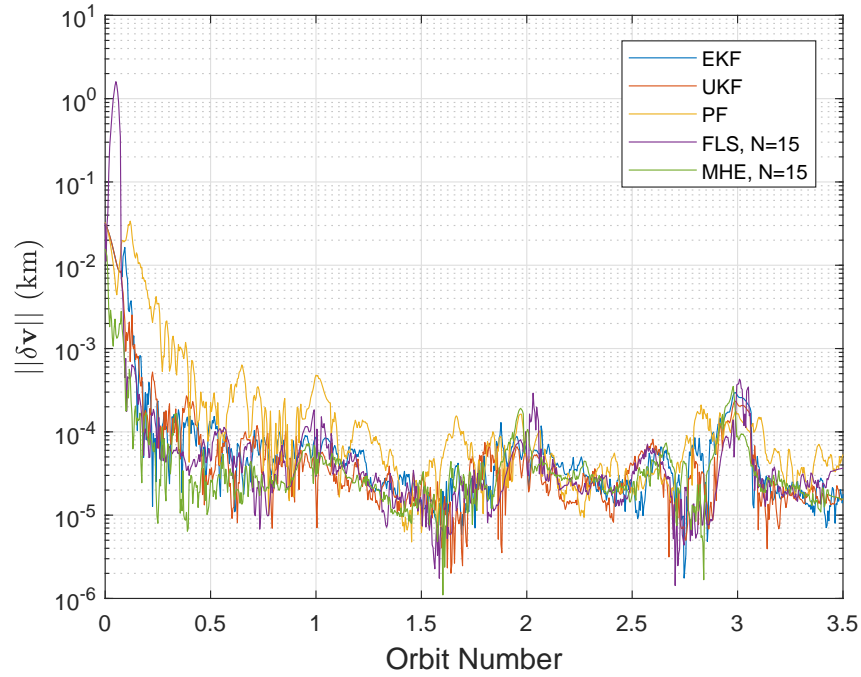


FIGURE 4.4: Elliptical orbit velocity error over single run.



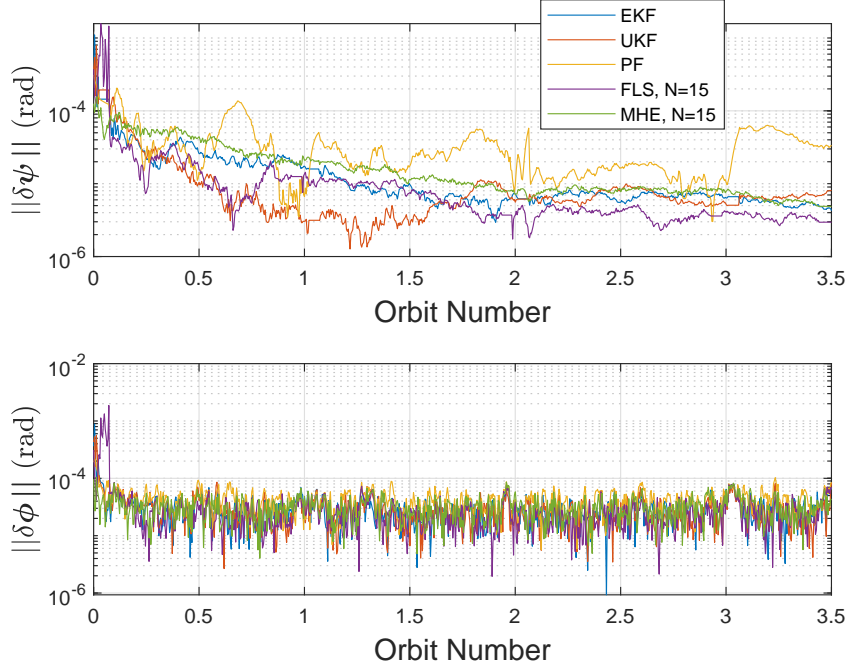


FIGURE 4.5: Elliptical orbit velocity error over single run.

Next a MC simulation of 1000 trials was run, through experimentation 1000 trials was chosen as it provided stable results. Over the MC trials Figure 4.6 looks at the mean position error, Figure 4.7 the mean velocity error, and Figure 4.8 the mean attitude and sensor misalignment error. The relative performance of the estimators is similar to that seen in the hyperbolic scenario, Section 3.5. The FLS and MHE provide the lowest errors, with the MHE providing the quickest convergence. The EKF and UKF provide similar results with the UKF providing slightly lower error along various points throughout the orbit. The PF then provides the highest estimation errors.

Looking at the sensor misalignment errors the MHE provides a slightly lower error than the other estimators, although besides the PF each estimator provides similar errors. All estimators again provide similar results in regards to the attitude error. This was due to the dynamics being dominated by the process noise. Finally, the PF provides poor results compared to the other estimators for reasons that have been mentioned in Section 3.5.

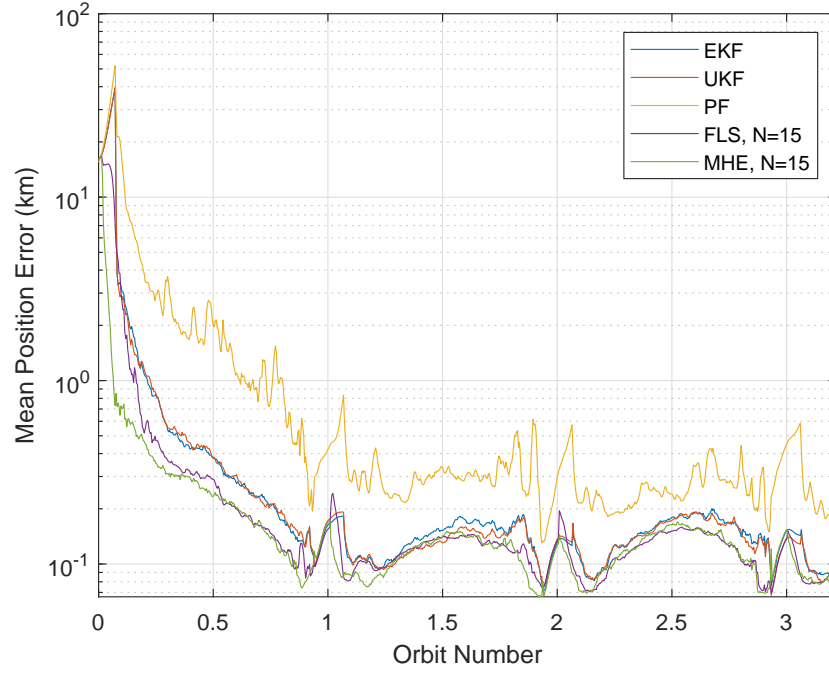


FIGURE 4.6: Elliptical orbit mean position error over MC trials.

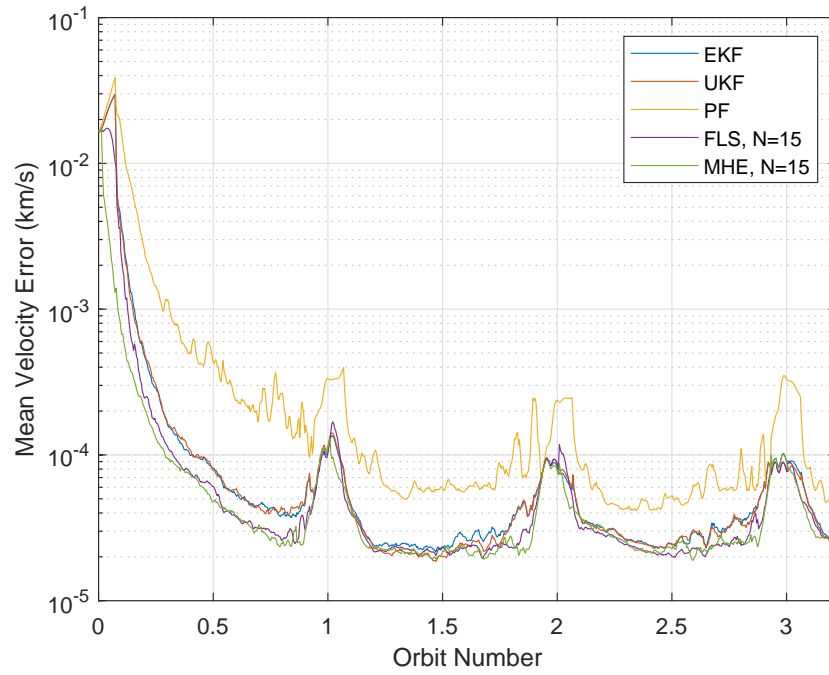


FIGURE 4.7: Elliptical orbit mean velocity error over MC trials.

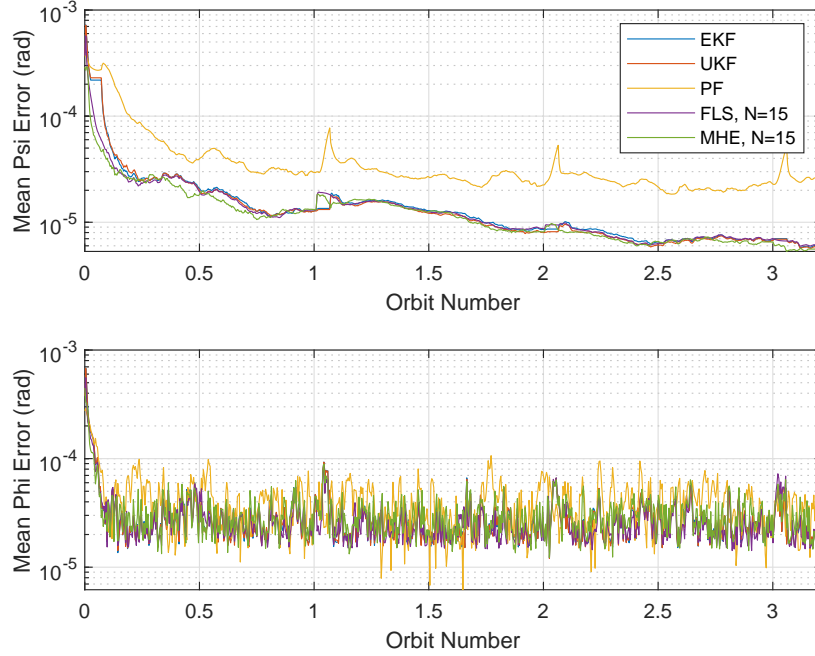


FIGURE 4.8: Elliptical orbit mean sensor misalignment and attitude error over MC trials.

The affect that the apoapsis distance has on the estimates can be further explored by looking at two other orbits with the same initial conditions as found in Table 4.3 but with an apoapsis altitude of 33,000 and 72,000 km. Again 1000 MC trials were run. Figure 4.9 presents the mean position error over the MC trials, and Figure 4.10 the mean velocity error. Looking at each figure the errors seen in each orbit are designated by the line marker, for example, the diamond marker represents the errors from the orbit with an apoapsis of 33,000 km. To reduce clutter only the EKF and MHE estimates are shown, the MHE provided the best estimates in each case, and other than the PF the EKF the worst. Therefore, the performance of the UKF and FLS for the larger orbits can be roughly extrapolated from the figures, where the trends were similar to what has been previously seen in Figures 4.6 and 4.7. The sensor misalignment and attitude errors are not shown as the results were comparable to Figure 4.8. Looking at the magnitude of the position errors it can be seen that as the apoapsis altitude is increased there becomes a larger error in position as apoapsis is approached. This is a similar trend as seen in the hyperbolic trajectory and is caused by the spacecrafts increased distance from Mars causing increased measurement noise. This trend is absent in the velocity estimates. In each orbit the converged value is roughly the same. Although with a larger apoapsis the velocity error converges faster with respect to its position in the orbit as there was more time for the estimates to converge. Comparing the state estimators as the apoapsis is increased in regards to the position estimates the performance of each estimator becomes more similar. This trend is seen in the velocity estimates as well.

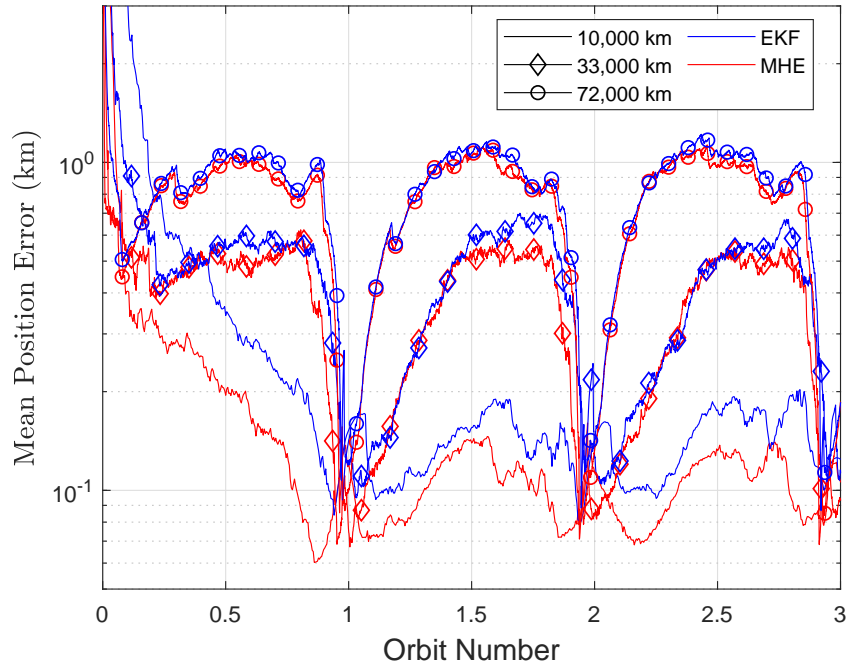


FIGURE 4.9: Mean position error for varying apoapsis.

Lastly, the mean computation time over the MC trials can be looked at. Relative to the EKF the UKF, PF, FLS and MHE took about 1.8-, 32.4-, 2.9- and 5.4- times the computation time. The improved estimates from the FLS and MHE come at the cost of an increased computation time. However, with the reduction in available measurements the computation cost is not as high as in the hyperbolic scenario.

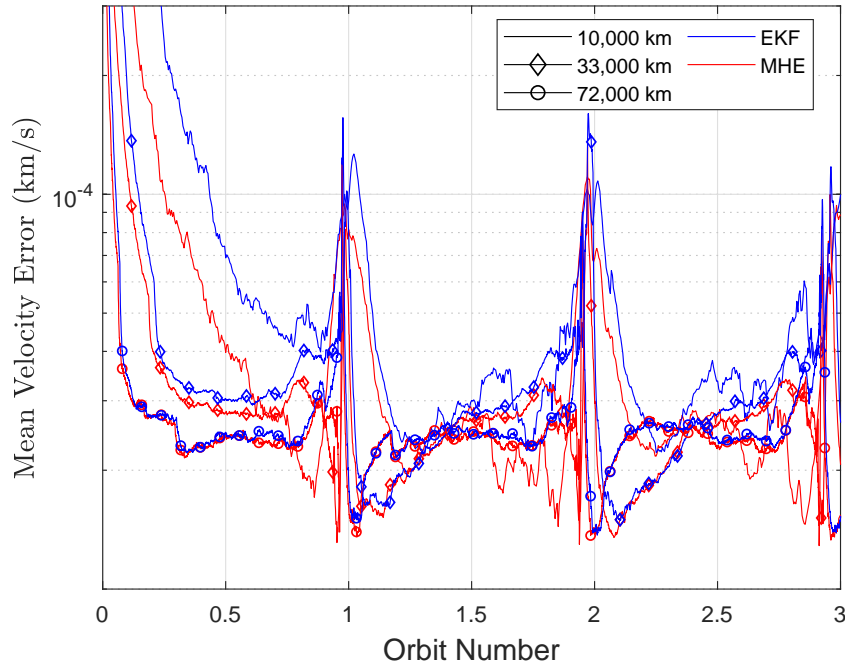


FIGURE 4.10: Mean velocity error for varying apoapsis.

## 4.2 Orbital Maneuver Timing

This section provides a preliminary study on using autonomous OpNav for timing an orbital burn maneuver. It was assumed that a spacecraft was placed into an elliptical orbit around Mars by means of an aerocapture maneuver. The goal will then be to use autonomous OpNav to time the apoapsis burn to raise the periapee out of the Martian atmosphere. Autonomous orbit control is a well established technique with the UoSAT 12 being the first satellite to demonstrate autonomous orbit control in an orbit around the Earth using GPS measurements in 1999 [28]. As well refer to Wertz [29] for a more in depth overview of the topic, here autonomous orbit control was looked at to reduce costs in space missions. This study however examines the performance of the five presented state estimators using a nanosatellite-class star tracker and the previously derived optical measurement models for orbital maneuvers.

### 4.2.1 Reference Mission

Injection into an elliptical orbit from a hyperbolic approach trajectory around Mars can occur by means of a retrograde burn or by use of an aerocapture. An aerocapture makes use of the Martian atmosphere to slow the satellite, allowing orbit insertion to be near propellant less. An aerocapture consists of four basic sequences:

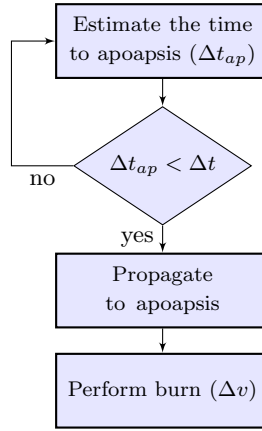


FIGURE 4.11: Flow chart used for estimating the time and magnitude of the orbital burn.

1. Hyperbolic approach trajectory
2. Atmospheric entry and exit, here the hyperbolic trajectory turns into an elliptical orbit
  - Subsequent atmospheric passes will lower the apoapsis if required
3. Apoapsis burn to raise periapse out of the Martian atmosphere
4. If required a periapse burn to produce the desired orbit

Step 3 is the focus of this study as the actual aerocapture maneuver was not modeled but rather it was assumed that the satellite could be placed into an elliptical orbit with the accuracies found in a previous study. Austin et al. [30] looked at the feasibility of delivering a small satellite into an elliptical orbit around Venus through an aerocapture, and provided the theoretical accuracy of the maneuver. Even though this study looks at an elliptical orbit around Mars these accuracies were assumed applicable. To add the atmospheric drag on the satellite was not modeled but rather the dynamic equations and measurement models used can be found in Section 4.1 for an elliptical orbit. The spacecraft was in the Martian atmosphere for a short period of time, thus it was felt that modeling the atmospheric drag would not have an affect on the end performance of each state estimator.

Once the satellite was placed into the elliptical orbit a small burn at apoapsis must occur to raise the periapse and take the satellites trajectory out of Mars' atmosphere. Figure 4.11 outlines the approach used to time and perform the burn. Refer to Curtis [31] for the mathematical equations used. The burn was modeled as an impulse maneuver.

Looking at Figure 4.11 the time to apoapsis was found by:

1. The estimated eccentricity vector was found as

$$- \mathbf{e} = \frac{\hat{\mathbf{v}}_P \times \mathbf{h}}{\mu_{Mars}} - \frac{\hat{\mathbf{r}}_P}{\hat{r}_P}, \text{ where } \mathbf{h} = \hat{\mathbf{r}}_P \times \hat{\mathbf{v}}_P$$

2. Find the estimated true anomaly as

$$- \nu = \arccos\left(\frac{\mathbf{e} \cdot \hat{\mathbf{r}}_P}{e \hat{r}_P}\right)$$

3. The estimated eccentric anomaly is

$$- E = 2 \arctan \frac{\tan \frac{\nu}{2}}{\sqrt{\frac{1+e}{1-e}}}$$

4. Through Keplers equation the estimated mean anomaly is

$$- M = E - e \sin E$$

5. The estimated orbital period is

$$- T = 2\pi \sqrt{\frac{a^3}{\mu_{Mars}}}, \text{ where } a = \frac{1}{\left(\frac{2}{\hat{r}_P} - \frac{\hat{v}_P^2}{\mu_{Mars}}\right)}$$

6. Lastly, the estimated time to apoapsis can be found as

$$- \Delta t_{ap} = \frac{\pi - M}{n}, \text{ where } n = \frac{2\pi}{T}$$

While the magnitude of the burn was found by:

$$1. h = \sqrt{\mu_{Mars} \hat{r}_{ap} (1 - e)}, \text{ where } e = \frac{\hat{r}_{ap} - r_{pe}}{\hat{r}_{ap} + r_{pe}}$$

$$2. \Delta v = \frac{h}{\hat{r}_{ap}} - v_P$$

Undefined terms include  $\Delta v$  as the magnitude of the performed burn,  $n$  is the mean motion of the satellite,  $\hat{r}_{ap}$  is the estimated apoapsis or the distance from Mars that the spacecraft is when the burn maneuver is performed, and  $r_{pe}$  is the desired periapse. All bold letters refer to a vector and non-bold letters the scalar magnitude, for example  $a = ||\mathbf{a}||$ .

#### 4.2.2 Simulation Results

To compare the performance of each estimator a Monte Carlo simulation of 2000 trials was run. The parameters of the simulator can be found in Table 4.4. As the FLS and MHE implement a delay into the estimation process when finding the time to apoapsis and the burn magnitude the first estimate in the estimation window was taken, or in other words from previous notations

TABLE 4.4: Orbital Burn Scenario Parameters

Initial Periapsis	60 km
Target Periapsis	200 km
Inclination	0 deg
RAAN	0 deg
Argument of Periapsis	0 deg
Position uncertainty $(1 - \sigma)$ , each axis	10 km
Velocity uncertainty $(1 - \sigma)$ , each axis	0.01 km/s
Sensor mounting uncertainty, each axis	0.1 deg.
Attitude error uncertainty, each axis	1 deg.
Sensor FOV (half-angle)	7.5 deg.
Cross-axis attitude error $(1 - \sigma)$	$1 \times 10^{-3}$ deg.
Roll attitude error $(1 - \sigma)$	$1 \times 10^{-2}$ deg.
Measurement update time	100s
Attitude error time constant, $\tau$	600 s
Attitude error process noise	$Q_\phi = (10^{-8})\mathbf{I}_{3 \times 3} \text{ rad}^2/\text{s}^2$
Velocity process noise (estimator only)	$Q_v = (3 \times 10^{-14})\mathbf{I}_{3 \times 3} \text{ m}^2/\text{s}^4$

the estimate  $\mathbf{x}_{k|k}$ . It was also attempted to take the optimal estimate in the window, that is  $\mathbf{x}_{k-N+1|k}$ , and propagate it forward to the current time step using the filter state propagation equations but similar results were seen.

Four different orbits with varying apoapses were considered. These include an apoapsis altitude of 5-, 10-, 15-, and  $20 \times 10^3$  km, where the spacecraft was placed into this orbit with the apoapsis accuracy found in Austin et al. [30]. From this paper the spacecraft was placed into an orbit with an apoapsis error described in Table 4.5. Through experimentation it was found that the magnitude of the standard deviation had little impact on the end errors.

TABLE 4.5: Apoapsis Error Standard Deviation [30]

Apoapsis ( $\times 10^3$ km)	Standard Deviation (km)
5	243
10	303
15	425
20	486

From the MC simulation Figure 4.12 presents the periapse error after the burn was performed with respect to an intended trajectory. Here similar results between the EKF, UKF, FLS, and MHE can be seen. The PF can provide lower errors but provides inconsistent results. Comparing the errors at various orbits similar trends are observed but a reduction in errors as the orbit gets larger is had. From previous results seen in Section 4.1.3 as the apoapsis is increased a



reduction in velocity errors for the first orbit is seen. The increased apoapsis allows more time for convergence, meaning the required  $\Delta v$  can be more accurately found.

It was desired to perform the burn at apoapsis, Figure 4.13 presents the radial error between the true apoapsis and where the burn was actually performed due to the state estimates. Here near identical errors are seen between each filter. This is because even though the position errors vary for each estimator, the results are close enough that the estimated time to apoapsis is similar resulting in comparable errors. Comparing the errors for each apoapsis an increase in errors is seen as the apoapsis increases due to increased measurement noises as the spacecraft is further from Mars, resulting in less accurate estimates. This trend was observed in Section 4.1.3 and Section 3.5.

Even though each state estimator provided similar errors in Figure 4.13, Figure 4.14 presents the radial error from each state estimator at the time the burn was performed. Here the variation in estimates is seen. Again similar results are noted between the EKF, UKF, FLS, and MHE, with the PF providing the worst errors. However, the errors are close enough to place the spacecraft into similar orbits. The estimated errors are smaller than the apoapsis error, meaning that the method used to calculate the burn, seen in Figure 4.11, has errors associated and is not perfect. The equations used to find the time to apoapsis and  $\Delta v$  do not take into account the perturbations due to Mars mass distribution which are considered in the simulation. Recalculating  $\Delta t_{ap}$  at each time step was thought to reduce some of these errors but the method is not perfect. This may also explain why the PF was able to provide better results in Figure 4.12 at times. With the variation in PF estimates previously seen some of the estimates may get “lucky” so to speak and provide velocity estimates that better place the spacecraft into the desired orbit.

By looking at the errors from the MC trials the delay in state estimates does not impede on orbital maneuvers, however provides no improvement either. With the FLS and MHE providing errors that are almost imperceptible from the EKF and UKF. As well, the navigation framework is accurate enough to be able to place a nanosatellite into a desired elliptical orbit with a high degree of accuracy.

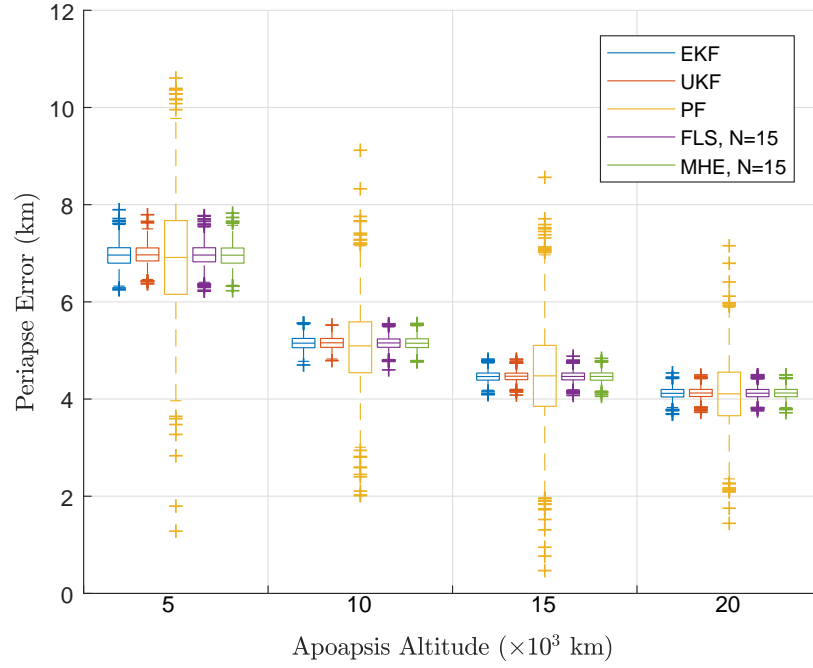


FIGURE 4.12: Periapse error after burn.

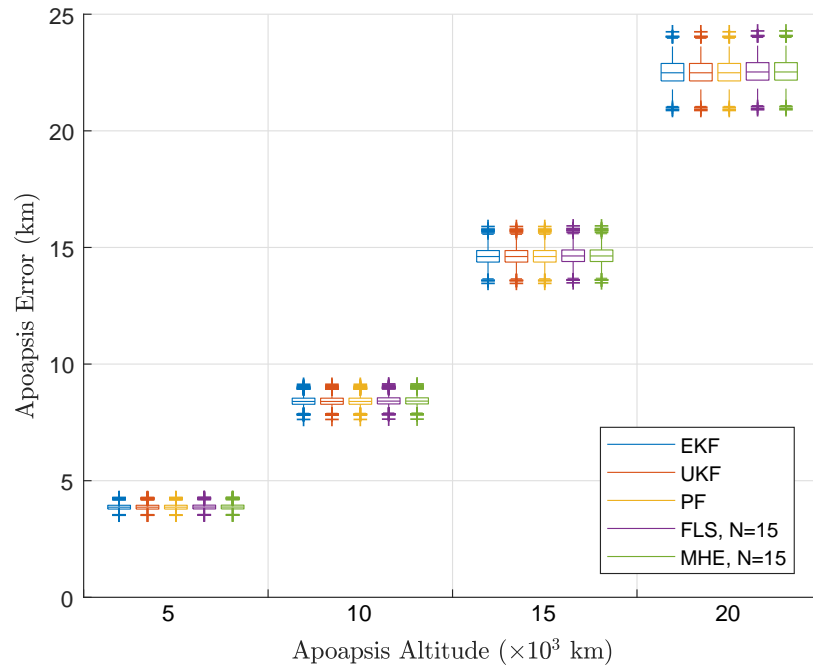


FIGURE 4.13: Apoapsis error at time of burn from spacecrafts true position.

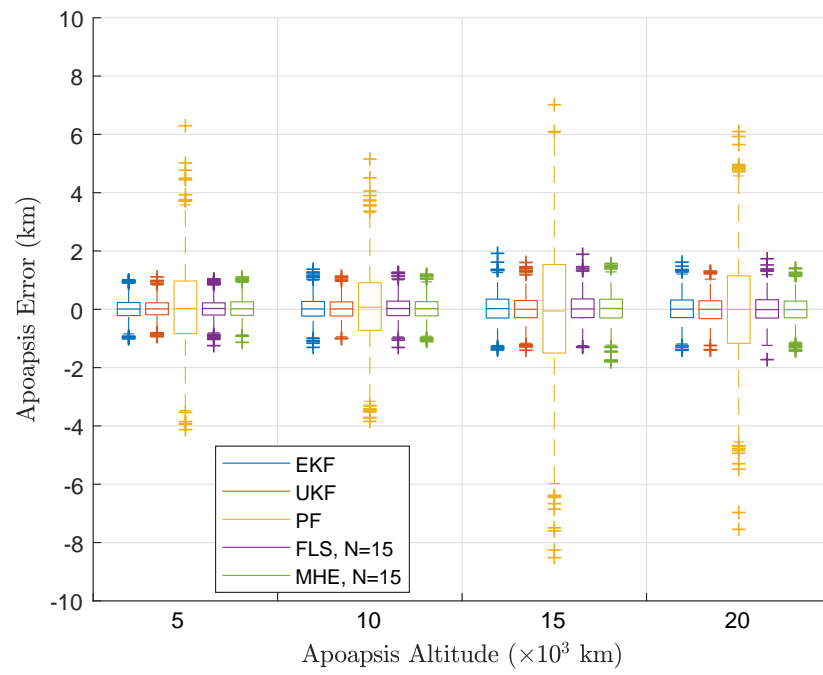


FIGURE 4.14: Apoapsis error at time of burn using filter estimates.

### 4.3 Chapter Summary

In this chapter two reference scenarios were looked at. The first involved an elliptical orbit around Mars. In this scenario the spacecrafts true trajectory includes Mars mass perturbations up to  $J_4$ , while the truth dynamics only include up to  $J_2$ . The sensor misalignment error remained constant over time and the spacecraft's attitude error was modeled as a first order system driven by random noise. Five OpNav observations were again considered for the elliptical trajectory, the outline of each observations can be found in the previous chapter. The availability of each measurement was also explored. For an apoapsis of 10,000 km only really the attitude of the spacecraft and displacement vector of Mars are observed but as the apoapsis is increased more measurements become available. To compare each state estimator a single run trial was first performed followed by a series of MC trials. From the trials it was seen that the FLS and MHE improved the mean RSS position and velocity error. The UKF showed slight benefits over the EKF, and the PF had the highest estimation errors. Besides the PF each sensor showed similar performance in estimating the sensor misalignment error, and each state estimator provided similar results when estimating the attitude error. When comparing the run times with respect to the EKF the PF had the highest computational cost, followed by the MHE, then the FLS and finally the UKF.

The second scenario compared each estimator in timing an orbital maneuver to raise the periapse of the spacecraft out of the Martian atmosphere after an aerocapture maneuver. The same scenario framework as the elliptical trajectory was used, this includes the frames of reference, dynamic and measurement models. As well, the actual aerocapture was not modeled but rather it was assumed that the spacecraft could be placed into an elliptical trajectory with the accuracy seen in a previous study. A series of MC trials was used to assess the performance of each state estimator. When looking at the estimation errors it was found that besides the PF each estimator provides a similar periapse error after the burn was performed. As well, even with the variation in estimates the burn was performed at roughly the same location. This means that the delay in estimates does not have an affect on real time events but no improvement either.

## Chapter 5

# Other Applications of Autonomous Optical Navigation

In this chapter Section 5.1 first explores the performance of the EKF, UKF, FLS, and MHE in estimating the mass of Mars through augmenting the state matrix with the gravitational parameter. The PF was excluded due to poor estimates seen when estimating a constant value, as seen in previous sections. Next, Section 5.2 outlines the final scenario in which each state estimator was compared. Here the spacecraft was placed into a LEO, where the only observations were of satellites in a geostationary orbit (GEO).

### 5.1 Planetary Mass Estimation

Parameter estimation of a system can be done by augmenting the state vector to provide estimates based on the dynamic, observation and estimate evolution over time. This section provides a preliminary study on examining the effectiveness of planetary mass estimation using horizon based OpNav. It is desired to estimate a bodies mass for both scientific reasons, and to improve the dynamic models which in turn will improve future state estimates. A comparison of the EKF, UKF, FLS, and MHE for estimation is also made. The mass of Mars was chosen as it is well known but the same concepts can be applied to other planetary bodies, particularly to small bodies.

A body's mass can be found by observing the affect it has on a satellites trajectory or moon that is within its gravitation pull. For example, the Swarm Flyby Gravimetry mission proposes measuring a small body masses by observing the trajectory of a collection of probes around the

body [32]. While, Rosenblatt et al. [33] estimated the masses of Phobos and Deimos by looking at two years worth of tracking information of the Mars Express satellite. Another method is that given any uncertainty in the mass estimate the state vector can be augmented with the gravitational parameter to find improved estimates of  $\mu$  from which the mass can be found using  $\mu = GM$ , where  $M$  refers to the mass of the planet, and  $G$  the gravitational constant equal to  $6.674 \times 10^{-11} m^3 kg^{-1} s^{-2}$  [31]. This method requires an initial estimate however. Due to the potential magnitude of the mass value compared to the other states it was found that directly augmenting the state vector with the mass lead to instability, and thus it was better to estimate the gravitational parameter. Using the previously defined state vector the augmented vector looks like

$$\mathbf{x} = \begin{bmatrix} \mathbf{r}_P^T & \mathbf{v}_P^T & \psi^T & \phi^T & \mu \end{bmatrix}^T \quad (5.1)$$

The EKF, UKF, FLS, and MHE will be compared using the previously defined elliptical orbit with an apoapsis altitude of 10,000 km and the hyperbolic trajectory around Mars to estimate Mars mass. The PF algorithm used was excluded as it was found too have poor convergence when estimating a constant, as seen in previous chapters when estimating the sensor misalignment error. Estimating a constant can cause particle collapse to a single value due to resampling, it was attempted to use the roughening step to provide variance to the particles but poor estimates were still seen. Further tuning may resolve this issue however.

### 5.1.1 Simulation Results

Using the scenario parameters for the hyperbolic and elliptical scenario defined in Chapter 3 and Chapter 4 two Monte Carlo simulations of 500 trials were run to estimate Mars mass. Figure 5.1, and Figure 5.2 present the fractional error, Eq. (5.2), over the simulation for the hyperbolic and elliptical trajectory respectively. In the hyperbolic scenario a initial uncertainty  $(1 - \sigma)$  of  $1 \times 10^4 \text{ km}^3/s^2$  was used to initialize the gravitational parameter in the MC trials. While, for the elliptical scenario a uncertainty of  $5 \times 10^4 \text{ km}^3/s^2$  was used. It was found that the hyperbolic scenario was more tolerant to initial errors in estimating the gravitational parameter, this may be due to the fact that the state estimators had more time to converge before Mars gravitational force had a large affect on the trajectory compared to the elliptical orbit. This is noted in Figure 5.1 with the lack of convergence for the first few hours of the simulation. It should be noted that for the elliptical orbit the simulation was initialized at periapse, and every time periapse is reached no measurements are available. This is the reason for the zero change or odd behavior in estimating the gravitation parameter. Lastly, the hyperbolic trajectory spans about 58 hours,

while the elliptical trajectory spans about 34 hours.

$$\text{Fractional Error} = \frac{|\mu_{Estimated} - \mu_{True}|}{\mu_{True}} \quad (5.2)$$

Comparing each estimator slight differences can be seen, such as the MHE providing a slightly quicker convergence in the hyperbolic trajectory, however each estimator provides a similar performance. One cause of the similar performance is the fact that the gravitational parameter does not appear in the measurement sensitivity matrix. In the MC trials the estimators are able to provide an accuracy up to about  $\pm 1 \text{ km}^3/\text{s}^2$  when estimating the gravitational parameter in these scenarios. This accuracy was found to be appropriate when looking to provide orbital determination, with errors on par with those seen in Chapter 3 and Chapter 4.

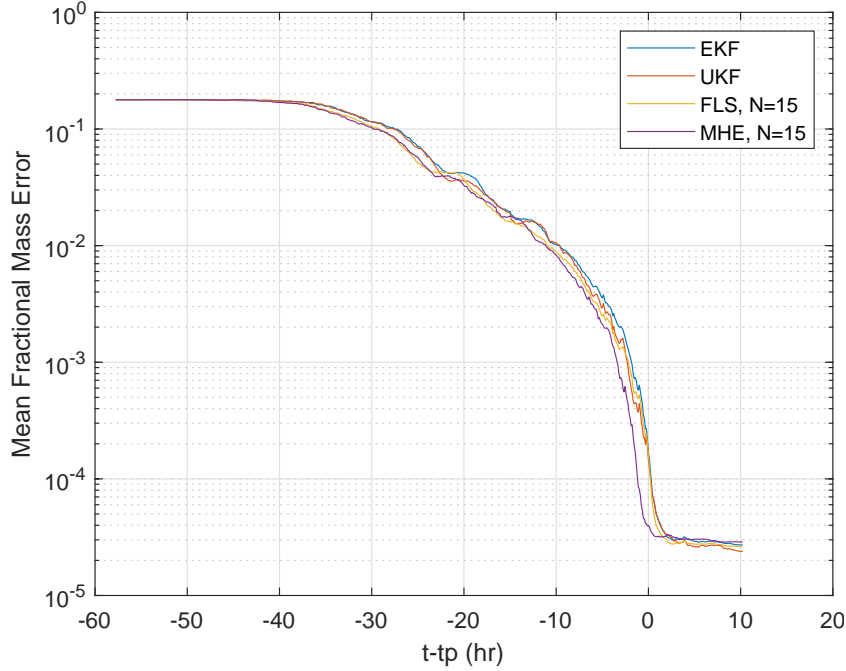


FIGURE 5.1: Hyperbolic orbit mean fractional mass error.

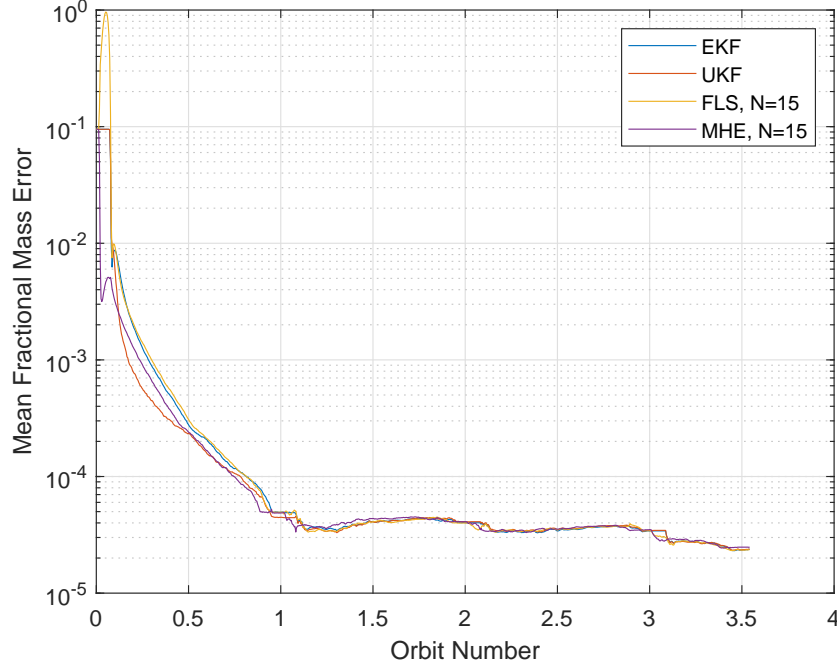


FIGURE 5.2: Elliptical orbit mean fractional mass error.

## 5.2 Geostationay Observations

Fesq et al. [34] describes using the ASTERIA CubeSat to explore using Autonav for orbit determination by passively observing objects in space. One potential set of objects mentioned are satellites in a geostationary orbit (GEO). This section provides a preliminary study on using only observations of GEO satellites for orbit determination through OpNav. This study also provides another scenario to compare the performance of the EKF, UKF, PF, FLS and MHE.

GEO is a geosynchronous circular orbit above Earth's equator with an altitude around 35,786 km. This orbit currently has more than 400 satellites each placed within slots. These slots are two degrees of longitude with each satellite operating within a box of 0.1 degrees of longitude [35]. This positioning means that pointing a camera towards GEO will likely observe multiple satellites. As the locations of these satellites are well known, the spacecraft's position and velocity could potentially be found through observing these satellites. The algorithms used to identify each satellite was not looked at but rather it was assumed that the positions of each satellite could accurately be determined.

Many previous studies have looked at many other forms of observations to provide autonomous OpNav in various planetary Earth orbits. For example, Straub and Christian [36] explored using observed coastlines for orbit determination and an EKF was used for state estimation.



TABLE 5.1: Earth Orbit Frame Identification

Frame	Symbol	Centre	X-Axis	Z-Axis
Inertial	$E$	Earth	Along equatorial plane	Along spin axis
Orbital	$O$	Spacecraft	Angular momentum direction of S/C	Outward radius
Sensor	$S$	Spacecraft	Increasing columns	Boresight

As well, orbit determination through landmark tracking has been heavily researched, such as by Kau [37] or Schlee and Toda [38] where an EKF was used in both studies.

In this chapter, Section 5.2.1 provides an overview of the scenario, reference frames and state variables, Section 5.2.2 and Section 5.2.3 provide an overview of the dynamic and measurement models used, and Section 5.2.4 provides the results from a set of MC trials.

### 5.2.1 Reference Mission

The spacecraft was placed into a planetary orbit around the Earth where two orbits were considered, a equatorial and a sun-synchronous orbit. Table 5.1 outlines the set of frames that will be used throughout this section.

Looking at the table Frame- $E$  represents an inertial frame with its center located at the center of mass of the Earth, its  $E_3$  axis aligned with the spin axis, and the  $E_1$  axis pointing in an arbitrary direction along the equatorial plane. Frame- $E$  is where the spacecrafts Cartesian position  $\mathbf{r}_E$  and velocity  $\mathbf{v}_E$  are defined within. The spacecraft's orientation is defined with respect to an orbital frame denoted  $O$  with its origin located at the spacecraft. The  $O_3$  axis points in the direction of the outward radius, and the  $O_1$  axis in the direction of angular momentum of the spacecraft. The rotation from Frame- $E$  to Frame- $O$  is then found as

$$\mathbf{C}_{OE} = \begin{bmatrix} \mathbf{E}_{1O} & \mathbf{E}_{2O} & \mathbf{E}_{3O} \end{bmatrix}^T \quad (5.3)$$

where

$$\mathbf{E}_{1O} = \frac{\mathbf{r}_E^\times \mathbf{v}_E}{\|\mathbf{r}_E^\times \mathbf{v}_E\|}, \quad \mathbf{E}_{3O} = \frac{\mathbf{r}_E}{\|\mathbf{r}_E\|} \quad \text{and} \quad \mathbf{E}_{2O} = \mathbf{E}_{3O}^\times \mathbf{E}_{1O} \quad (5.4)$$

A sensor frame denoted as  $S$  is used to define where the sensor, and inherently the spacecraft, is pointing. The  $S_3$  axis points along the boresight of the sensor, and the  $S_1$  axis in the direction of increasing columns. A pointing rule was used to determine how the spacecraft was orientated throughout the mission. It was assumed that the spacecraft was always able to

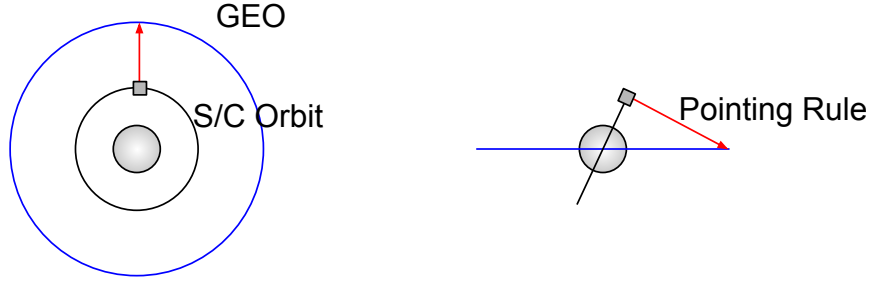


FIGURE 5.3: Pointing rule used to point spacecraft at point along GEO with same azimuth angle.

point at a place along the geosynchronous orbit with a high degree of accuracy. This means that the attitude determination and control system (ADCS) is able to perfectly follow a desired attitude. In actuality there would be some attitude error, however, as it is a preliminary study on examining the feasibility of using satellites in GEO for orbit determination it is thought to be an appropriate assumption. If this assumption is not made then in the case of initial state errors the observations may be lost and the state estimates do not converge as the satellite is unable to reorientate itself without measurements. The pointing rule involves pointing the satellite at a point along the GEO with roughly the azimuth angle. This is illustrated in Figure 5.3.

For both orbits the rotation between Frame- $E$  and the desired orientation of Frame- $S$  can be directly found by using an axis angle rotation of the form  $\boldsymbol{\theta} = \theta \mathbf{a}$ . The axis of rotation  $\mathbf{a}$  is in the direction perpendicular to the point on the GEO from the spacecraft in Frame- $E$  and the  $E_3$  axis. Where,  $\mathbf{g}_E$  is the location of this point in Frame- $E$ .

$$\mathbf{a} = \frac{(\mathbf{g}_E - \mathbf{r}_E)^\times E_3}{\|(\mathbf{g}_E - \mathbf{r}_E)^\times E_3\|} \quad (5.5)$$

Then the desired rotation angle can be described by the angle between both vectors

$$\theta = \arccos\left(\frac{(\mathbf{g}_E - \mathbf{r}_E) \cdot E_3}{\|(\mathbf{g}_E - \mathbf{r}_E)\| \|E_3\|}\right) \quad (5.6)$$

The rotation matrix can then be found using Rodrigues formula.

$$\mathbf{C}_{SE} = \cos \theta I_{3 \times 3} + (1 - \cos \theta) \mathbf{a} \mathbf{a}^T - \sin \theta \mathbf{a}^\times \quad (5.7)$$

Lastly, the state matrix can be composed. As only the spacecrafts position and velocity are to be estimated this matrix is

$$\mathbf{x} = \begin{bmatrix} \mathbf{r}_E \\ \mathbf{v}_E \end{bmatrix} \quad (5.8)$$

### 5.2.2 System Dynamics

The spacecraft was assumed to undergo unpowered flight throughout its orbit, where the only accelerations acting on the spacecraft are those caused by the gravitational force of the Earth. The true position and velocity are propagated using

$$\begin{bmatrix} \dot{\mathbf{r}}_E \\ \dot{\mathbf{v}}_E \end{bmatrix} = \begin{bmatrix} \mathbf{v}_P \\ -\frac{\mu_{Earth}}{r^3} \mathbf{r}_E + \mathbf{a}_{J_2} \end{bmatrix} \quad (5.9)$$

Here,  $\mathbf{a}_{J_2}$  is used to model the perturbations caused by the oblateness of the Earth modeled up to the  $J_2$  zonal harmonic, for the Earth  $J_2 = 0.00108263$ . A common simplified model for these perturbations is seen below [31].

$$\mathbf{a}_{J_2} = -\frac{3 J_2 \mu_{Earth} R_{Earth}^2}{2 r^4} \begin{bmatrix} (1 - 5(\frac{r_3}{r})^2) \frac{r_1}{r} \\ (1 - 5(\frac{r_3}{r})^2) \frac{r_2}{r} \\ (3 - 5(\frac{r_3}{r})^2) \frac{r_3}{r} \end{bmatrix} \quad (5.10)$$

In the onboard estimator only a two body problem was considered. A process noise  $\mathbf{w}_v$  with covariance  $Q_v$  is added to partially account for the modeling error. The dynamic model used in the estimators is then

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), t) + \mathbf{w} = \begin{bmatrix} \mathbf{v}_P \\ -\frac{\mu_{Earth} \mathbf{r}_P}{r^3} \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{w}_v \end{bmatrix} \quad (5.11)$$

With the full process noise covariance matrix equal to

$$Q = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & Q_v \end{bmatrix} \quad (5.12)$$

### 5.2.3 Measurement Model

A set of  $m$  satellites in GEO were uniformly distributed around the Earth and simulated for observation. In this study it was assumed the position of each satellite within the FOV of the sensor could accurately be determined. The position of each satellite  $m$  in Frame- $E$  is known and denoted as  $s_{E,m}$ . The position of each satellite with respect to the spacecraft in the sensor frame can then be found as

$$s_{S,m} = \mathbf{C}_{SE}(s_{E,m} - \mathbf{r}_E) \quad (5.13)$$

For observations a pinhole camera model was used, where the image frame axes align with  $S_1$  and  $S_2$ . Given the locations of each satellite with respect to the spacecraft in Frame- $S$  the measurement model can be found in Eq. (5.14), where the measurements are scaled in terms of pixels given the sensor pixel dimension  $\delta_{\text{pix}}$ . Here  $j$  refers to each observed satellite in view of the sensor.

$$\tilde{\mathbf{z}}_j = \frac{f}{\delta_{\text{pix}}} \begin{bmatrix} s_{S,j,1}/s_{S,j,3} \\ s_{S,j,2}/s_{S,j,3} \end{bmatrix} + \boldsymbol{\nu}_{\mathbf{z}} \quad (5.14)$$

Here,  $f$  refers to the focal length of the sensor, and  $\boldsymbol{\nu}_{\mathbf{z}}$  is the measurement noise. The measurement noise covariance matrix is based on how accurately the location of each satellite in the sensors field of view can be determined. For this study each satellite is considered a point source with a coordinate uncertainty of  $\sigma_{\mathbf{z}} = 0.2$  pixels. The measurement noise covariance matrix is seen below. Here  $r$  is the total number of measurements.

$$\mathbf{R}_{\mathbf{z},k} = \sigma_{\mathbf{z}}^2 I_{2r \times 2r} \quad (5.15)$$

The measurement sensitivity matrix can then be found by applying the chain rule to Eq. (5.14).

$$\mathbf{H}_j = \frac{\partial \mathbf{z}_j}{\partial s_{S,j}} \frac{\partial s_{S,j}}{\partial \mathbf{r}_E} \quad (5.16)$$

where,

$$\frac{\partial \mathbf{z}_j}{\partial s_{S,j}} = \frac{f}{\delta_{\text{pix}}} \begin{bmatrix} \frac{1}{s_{S,j,3}} & 0 & -\frac{s_{S,j,1}}{s_{S,j,3}^2} \\ 0 & \frac{1}{s_{S,j,3}} & -\frac{s_{S,j,2}}{s_{S,j,3}^2} \end{bmatrix} \quad (5.17)$$

$$\frac{\partial s_{S,j}}{\partial \mathbf{r}_E} = -\mathbf{C}_{SE} \begin{bmatrix} I_{3 \times 3} & 0_{3 \times 3} \end{bmatrix} \quad (5.18)$$

TABLE 5.2: Earth Orbit Scenario Parameters

	Equatorial Orbit	Sun-Synchronous Orbit
Initial Inclination	0°	98°
Altitude	800 <i>km</i>	800 <i>km</i>
Starting Position	$[7717.81\ 0\ 0]^T \text{ km}$	$[7717.81\ 0\ 0]^T \text{ km}$
Position uncertainty ( $1 - \sigma$ ), each axis	5 km	5 km
Velocity uncertainty ( $1 - \sigma$ ), each axis	0.01 km/s	0.01 km/s
Measurement update time	100 <i>s</i>	100 <i>s</i>
Velocity process noise (estimator only)	$Q_v = 1 \times (10^{-5})I_{3 \times 3} \text{ km}^2/\text{s}^4$	$Q_v = 1 \times (10^{-5})I_{3 \times 3} \text{ km}^2/\text{s}^4$

#### 5.2.4 Simulation Results

For both the equatorial and sun-synchronous orbit a Monte Carlo simulation of 1000 trials was run. Table 5.2 outlines the parameters for each trial, with the initial state error being derived from the initial covariance matrix. For this simulation a set of 200 satellites in GEO with slightly varying orbital parameters were simulated, leading to on average 7.3 satellites observed per time step. This number was somewhat arbitrarily chosen, and in actuality more satellites would probably be observed. However, as a rather ideal scenario is already made this may not be fully forthcoming of the potential results. As well, increasing the number of observed satellites was seen to have little impact on the current results. The UKF was initialized with  $\alpha = 1$  and  $\kappa = 0$ , the PF with 2000 particles, and the MHE and FLS with a window length of 15.

Figure 5.4 and Figure 5.5 present the mean RSS error in position and velocity for the equatorial orbit in the MC trials. Here the MHE provides the lowest estimation error, followed by the FLS. However, unlike in previous sections the PF provides errors that are better than both the EKF and UKF, and equivalent to the FLS with respect to the velocity estimates. This is unlike what has been seen in previous sections and may be due to either the simplified simulation or its framework. Further tuning of the PF may further improve the results. Lastly, the UKF improves estimates compared to the EKF.

Figure 5.6 and Figure 5.7 presents the mean RSS position and velocity error for the sun-synchronous orbit respectively. In the position errors there are large oscillations due to the two body forces. There is then the smaller oscillations due to the  $J_2$  accelerations on the spacecraft. The larger oscillations occur every quarter of an orbit, while the small ones are seen in between. These oscillations can also be seen in the velocity estimates and the high frequency of them is caused by the  $J_2$  accelerations. To add a large dip in position estimates is seen when the spacecraft is furthest from the equatorial plane. This is thought to be caused by a decrease in the speed of the spacecraft. Comparing each estimator similar results are seen in the equatorial

orbit. However in this case the EKF, UKF, and PF appear to provide estimates that are similar, especially in position. This is unlike in the equatorial orbit where a clear variation in estimates was seen.

Looking at the magnitude of the errors produced by each estimator it can be seen that they are not low enough to provide precise orbit determination, and are not comparable to other methods of orbit determination. For example, Battistini [39] showed that through GPS onboard a Cubesat the autonomous orbit determination system was able to achieve errors in position and velocity  $< 1$  m and  $< 1$  cm/s respectively. However, satellites in GEO could serve as another form of observation. Some of the estimation error does lie in only considering a two-body problem in the onboard estimator, adding the  $J_2$  accelerations to the filter dynamic model still do not produce errors low enough to be comparable with other methods however.

Finally, the computation time of each estimator was compared to the reference EKF. Here the UKF, PF, FLS, and MHE took roughly 2.2-, 37.0-, 1.3-, and 4.1- times that of the EKF. Improved estimates from the MHE come at a higher computational cost. The FLS provides errors close to the MHE but with a large reduction in computation time. The PF has by far the highest run time. Fewer particles could be used to reduce it but at the cost of higher estimation errors.

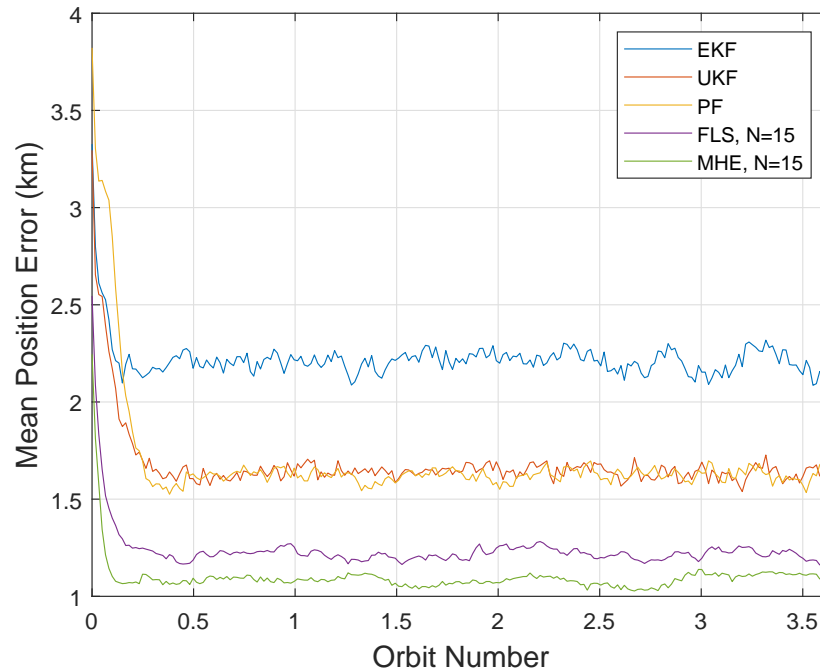


FIGURE 5.4: Equatorial orbit mean position error.

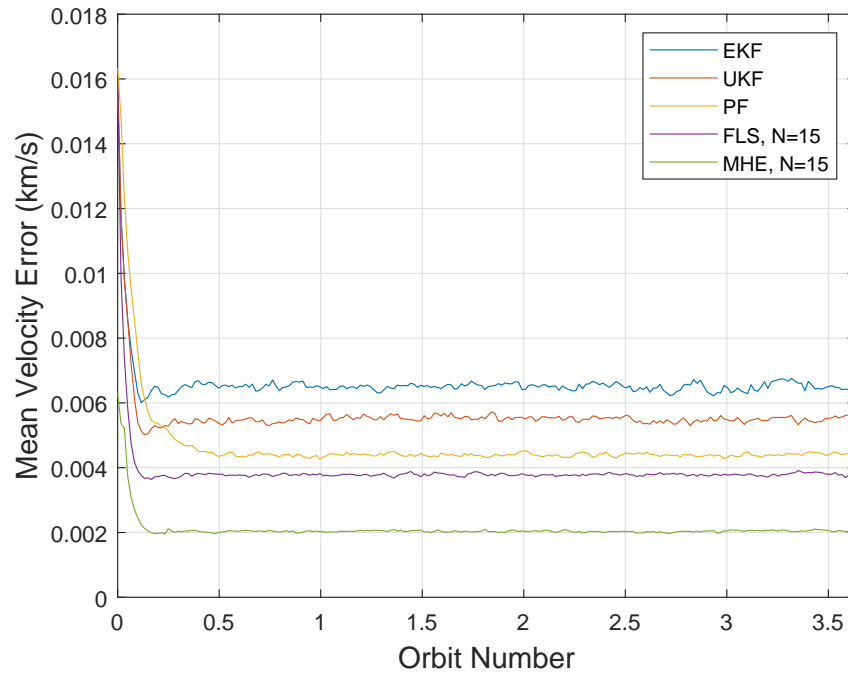


FIGURE 5.5: Equatorial orbit mean velocity error.

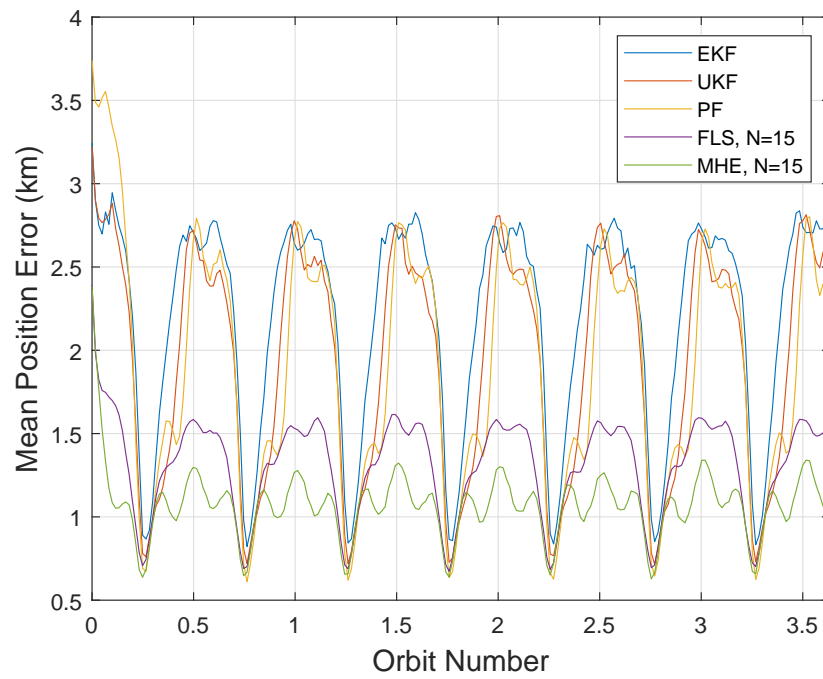


FIGURE 5.6: Sun-synchronous orbit mean position error.

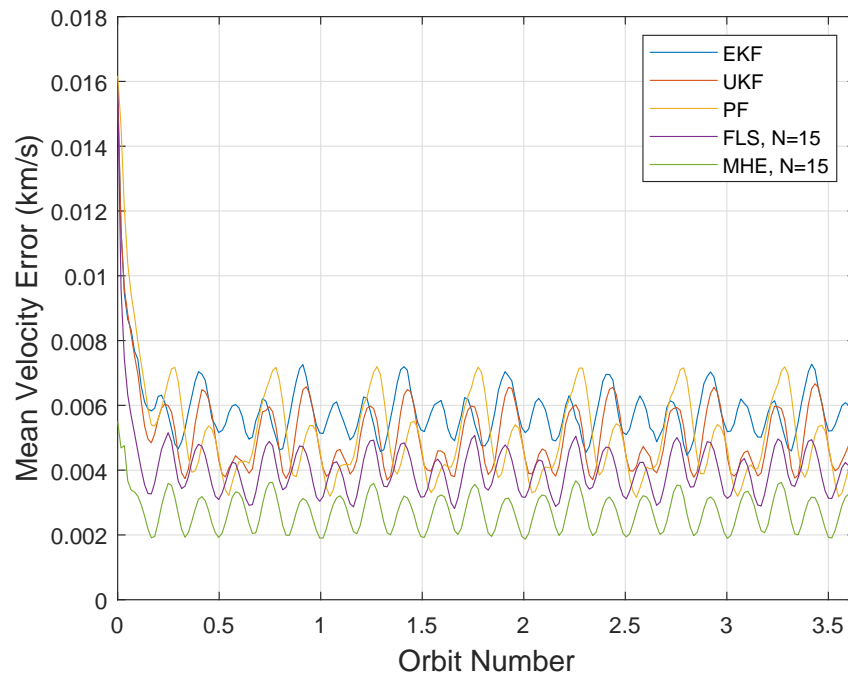


FIGURE 5.7: Sun-synchronous orbit mean velocity error.



### 5.3 Chapter Summary

In this chapter the first section explored estimating the mass of Mars by augmenting the state matrix with the gravitational parameter. The PF was excluded due to poor results seen when estimating a constant. Both the hyperbolic and elliptical orbit were simulated, where a series of MC trials was run to assess the performance of the EKF, UKF, FLS and MHE. Each estimator displayed a similar performance when estimating Mars mass, with only slight differences seen in the mean fractional mass error between each estimator. The accuracy seen was also found to be appropriate for orbital determination, with errors similar to those found in Chapter 3 and Chapter 4 for the hyperbolic and elliptical trajectory.

The next section placed the spacecraft into a LEO, where satellites in GEO were used as observations. In this scenario the spacecrafts state only included its position and velocity, with the true dynamics including Earth's  $J_2$  term and the filter dynamics only considering a two body problem. For the observation model a pin-hole camera model was used. A series of MC trials was run to assess the performance of each state estimator. In the simulation two orbits were considered: an equatorial orbit, and a sun-synchronous orbit. In the equatorial orbit the MHE provided the lowest estimation errors, followed by the FLS. The PF provides equivalent estimation errors as the UKF in regards to the position estimates and improvements when estimating the velocity. Finally, the EKF had the highest estimation errors. In the sun-synchronous orbit the MHE again provided the best estimation errors followed by the FLS. The UKF and PF both provided improvements over the EKF, however the estimation errors were closer than in the equatorial orbit. When looking at the magnitude of the errors it was seen that they are not low enough to provide precise orbit determination, falling short of other methods to perform the same task.

## Chapter 6

# Conclusion and Future Work

### 6.1 Conclusion

This thesis compared several nonlinear state estimators to perform autonomous OpNav using nanosatellite class instruments. The state estimators included: an extended Kalman filter as a reference solution, an unscented Kalman filter, a particle filter, a fixed-lag smoother, and moving horizon estimation. To gauge the performance and convergence properties of each estimator three reference trajectories were considered: a hyperbolic approach and elliptical orbit around Mars using observations of Mars and its moons, and a LEO using observations of only satellites in a GEO. Two further studies were applied to the Mars trajectories. The first looked at timing an orbital maneuver to raise the periapse of a small satellite out of the Martian atmosphere after an aerocapture maneuver. The other involved estimating the mass of Mars through augmenting the state matrix with Mars gravitational parameter. MC simulations were then used to validate the results.

Conclusions can be drawn when directly comparing each estimator. In general when purely looking at orbit determination adding a delay into the estimation process can improve the results, this was especially seen when looking at the position and velocity estimates for the hyperbolic Mars approach and both LEO scenarios. Table 6.1 shows the mean RSS position and velocity errors from both Mars and LEO trajectories in the MC trials. Here the MHE and FLS have the lowest errors, with the MHE having slightly better results. Slight improvements were seen in the elliptical trajectory from implementing the MHE and FLS but as the apoapsis was increased these improvements grew smaller. To add the UKF provided better if not similar results with respect to the reference EKF in the Mars orbits, while lower errors were seen in both LEO scenarios.

TABLE 6.1: Mean RSS Errors in Position and Velocity for each Trajectory

	Hyperbolic Trajectory		Elliptical Trajectory		LEO	
	Pos. (km)	Vel. (m/s)	Pos. (km)	Vel. (m/s)	Pos. (km)	Vel. (m/s)
EKF	36.47	3.89	0.85	0.67	2.21	6.23
UKF	35.96	3.88	0.84	0.67	1.81	5.35
PF	52.00	6.60	1.64	1.17	1.79	4.93
FLS	23.75	2.61	0.55	0.48	1.27	3.98
MHE	23.02	2.03	0.31	0.21	1.08	2.42

The PF provided poor results in both the hyperbolic and elliptical Mars trajectories. The PF provided poor results for a few reason, for one it was found sensitive to the initial distribution, if no particles fell within “good” regions the PF estimates were seen to diverge. As well, the velocity estimates did not appear in the likelihood function which may have lead to a slower convergence as there was no direct information on how good the estimate was. Lastly, there was difficulting in estimating a constant. Any variation in the estimates came from the roughening step which was seen to cause erratic behavior. This was seen in estimating the sensor misalignment error and why the PF was excluded when the estimators were compared for estimating the mass of Mars. The PF did perform well in the LEO simulation albeit. Producing lower errors than the UKF and EKF, and results similar to the FLS in the equatorial Earth orbit. While, in the Sun-synchronous orbit the PF provided estimates similar to the UKF. The reason for the improved performance in the LEO may have been due to its simpler framework and no constant state variables.

It should be noted the effect that the delay in state estimates may have in practice. This was explored in Section 4.2 where each state estimator was compared for timing an orbital maneuver. Here it was seen that the results from the FLS and MHE were comparable to the EKF and UKF, meaning the delay has little impact on orbital maneuvers, however no improvement either. Lastly, when estimating the mass of Mars slight improvements on the time to convergence were seen through the MHE but each estimator converged to similar values. Again it should be noted that the PF was excluded for this trial.

The mean computational time of each estimator relative to the EKF can be seen in Table 6.2. It can be noted that to achieve the benefits of the MHE in each case an increased computation time is needed, almost double the time of both the UKF and FLS. The FLS can provided similar errors in the Mars orbits and slightly higher errors in the LEO’s but with a reduced computational cost. For small satellites the available computational power can be a limiting factor making this an important statistic. It should be noted that these times are also dependent on the measurement and dynamic models used and are not the same for each case.

TABLE 6.2: Mean Computation Time Relative to EKF

	Mars Trajectories	LEO
UKF	2.6	2.1
PF	50.1	37.0
FLS	3.5	1.3
MHE	6.7	4.1

Without a specific mission in mind it can be difficult to assess how each state estimator will perform, and if the increased computational burden is warranted. However, the conducted studies showed an improvement in state estimation compared to the conventional EKF through other methods of nonlinear estimation. These results could be applied to other mission scenarios, sensors and measurements to further enable nanosatellite instruments and missions.

## 6.2 Future Work

This section looks at possible future directions that this research could take. To begin, in regards to state estimation future work would involve exploring the effect the horizon length for the MHE and FLS has on estimates. Studies were performed to find an optimal window length however a more in depth analysis could be made. To add improving the overall computational performance of the MHE, and PF could be looked at. An effort was made into making efficient code, however it is felt it could be improved. For example, for the MHE better sparsity management could further improve the computational performance. As well, different forms of the *arrival cost* could be considered in the MHE to further improve estimates. Lastly, as noted in the previous sections high estimation errors were seen from the PF state estimates, further effort could go into tuning the PF to try and improve estimates or finding better ways or strategies to estimate the sensor misalignment constant, consequently this would also look at parameter estimation.

For each orbit further disturbances could be included to better look at the expected performance of each estimator in practice. Just looking at examples of disturbances with a magnitude that may disturb the spacecrafts trajectory in each orbit, solar radiation pressure could be added to the hyperbolic trajectory. While, for the elliptical trajectory around Mars gravitational perturbations due to the Sun and Jupiter and lastly, for the LEO perturbations due to the Moon and Sun could be considered.

The parameter estimation simulation performed in Chapter 5 could be extended to observe the performance with regards to smaller bodies, such as planets or asteroids, which would more likely want to be estimated using a small satellite.

Lastly, in Chapter 5 a more simplistic simulation was performed. Further work could go into a more in depth analysis of the scenario. As well, further observations could be added into the scenario to consider the affect that observations of satellites in GEO may have as a secondary observation.

## Appendix A

# Moving Horizon Estimation Extensions

In this section the undefined terms in the smoothing formulation of the *arrival cost* seen in Eq. (2.52) are presented. Partial derivatives of the describe dynamic and measurement models are defined as

$$\mathbf{F}_k = \frac{\partial \mathbf{f}}{\partial \mathbf{x}_k}, \quad \mathbf{G}_k = \frac{\partial \mathbf{f}}{\partial \boldsymbol{\omega}_k}, \quad \mathbf{H}_k = \frac{\partial \mathbf{h}}{\partial \mathbf{x}_k} \quad (\text{A.1})$$

The rest of these terms were derived in V. Rao et al. [19]. While they will be represented using the notations seen in Tenny and Rawlings [40].

$$Y_{T-1}^{T-N+1} = \begin{bmatrix} \tilde{y}_{T-N+1} \\ \tilde{y}_{T-N+2} \\ \tilde{y}_{T-N+3} \\ \vdots \\ \tilde{y}_{T-1} \end{bmatrix} \quad (\text{A.2})$$

$$O_{N-1} = \begin{bmatrix} \mathbf{H}_{T-N+1} \\ \mathbf{H}_{T-N+2} \mathbf{F}_{T-N+1} \\ \mathbf{H}_{T-N+3} \mathbf{F}_{T-N+2} \mathbf{F}_{T-N+1} \\ \vdots \\ \mathbf{H}_{T-1} \mathbf{F}_{T-2} \mathbf{F}_{T-3} \cdots \mathbf{F}_{T-N+1} \end{bmatrix} \quad (\text{A.3})$$

$$W_{N-1} = \mathbf{R} + M\mathbf{Q}M^T \quad (\text{A.4})$$

where, for  $i, j \leq T-1$

$$M_{i,j} = \begin{cases} 0 & \text{if } j \geq i \\ \mathbf{H}_i \mathbf{G}_i & \text{if } j = i-1 \\ \mathbf{H}_i \mathbf{F}_{i-1} \mathbf{F}_{i-2} \cdots \mathbf{F}_{j+1} \mathbf{G}_j & \text{otherwise} \end{cases} \quad (\text{A.5})$$

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_{T-N+1} & 0 & \cdots & 0 \\ 0 & \mathbf{R}_{T-N+2} & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ 0 & 0 & \cdots & \mathbf{R}_{T-1} \end{bmatrix} \quad (\text{A.6})$$

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_{T-N+1} & 0 & \cdots & 0 \\ 0 & \mathbf{Q}_{T-N+2} & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ 0 & 0 & \cdots & \mathbf{Q}_{T-1} \end{bmatrix} \quad (\text{A.7})$$

The term  $W_{N-1}$  may be a rather extensive but say for example  $T = 4$  and  $N = 4$  then

$$W_3 = \begin{bmatrix} \mathbf{R}_1 & 0 & 0 \\ 0 & \mathbf{H}_2 \mathbf{G}_1 \mathbf{Q}_1 \mathbf{G}_1^T \mathbf{H}_2^T + \mathbf{R}_2 & \mathbf{H}_2 \mathbf{G}_1 \mathbf{Q}_1 \mathbf{G}_1^T \mathbf{F}_2^T \mathbf{H}_3^T \\ 0 & \mathbf{H}_3 \mathbf{F}_2 \mathbf{G}_1 \mathbf{Q}_1 \mathbf{G}_1^T \mathbf{H}_2^T & \mathbf{H}_2 \mathbf{G}_1 \mathbf{Q}_1 \mathbf{G}_1^T \mathbf{F}_2^T \mathbf{H}_3^T + \mathbf{H}_3 \mathbf{G}_2 \mathbf{Q}_2 \mathbf{G}_2^T \mathbf{H}_3^T + \mathbf{R}_3 \end{bmatrix} \quad (\text{A.8})$$

## Appendix B

# Mars Measurement Models

A overview of each measurement model is outlined below, refer to Enright et al. [22] for a more in depth derivation. To begin, the measurement sensitivity matrix for each measurement is the partial derivative of the measurement model with respect to each state variable.

$$\mathbf{H}_j = \begin{bmatrix} \frac{\partial \mathbf{h}_j}{\partial \mathbf{r}_P} & \frac{\partial \mathbf{h}_j}{\partial \mathbf{v}_P} & \frac{\partial \mathbf{h}_j}{\partial \psi_1} & \dots & \frac{\partial \mathbf{h}_j}{\partial \phi} \end{bmatrix} \quad (\text{B.1})$$

Combining each measurement the final matrix becomes

$$\mathbf{H}_k = \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{H}_2 \\ \vdots \end{bmatrix} \quad (\text{B.2})$$

### B.1 Attitude Measurements

In practice a star tracker can provide an estimate of the attitude of a spacecraft by matching visible stars in the field of view (FOV) of the sensor against their position in a star catalog. For this thesis an attitude error pseudomeasurement,  $\Delta$ , was used to represent the first-order error rotation vector between the reference and actual attitude. The ideal  $\Delta$  vector can be found by solving for  $\Delta$  and then taking the RHS matrix elements:

$$\left(\mathbf{I} - \Delta^\times\right) = \mathbf{C}(\psi) \mathbf{C}(\bar{\mathbf{C}}_{SB}\phi) \approx \left(\mathbf{I}_{3 \times 3} - \psi^\times\right) \mathbf{C}(\phi_S) \quad (\text{B.3})$$



To account for the star tracker accuracy a zero mean Gaussian noise  $\boldsymbol{\nu}_\Delta$  is added to this ideal value to produce the noisy attitude error pseudomeasurment.

$$\tilde{\mathbf{z}}_\Delta = \boldsymbol{\Delta} + \boldsymbol{\nu}_\Delta \quad (\text{B.4})$$

To find the estimated pseudomeasurment  $\hat{\boldsymbol{\Delta}}$  the estimated attitude,  $\hat{\boldsymbol{\phi}}$ , and sensor misalignment,  $\hat{\boldsymbol{\psi}}$ , error are used, where  $\hat{\boldsymbol{\Delta}}$  is solved for and the RHS matrix elements are taken.

$$\left(\mathbf{I} - \hat{\boldsymbol{\Delta}}^\times\right) \approx \left(\mathbf{I}_{3 \times 3} - \hat{\boldsymbol{\psi}}^\times\right) \mathbf{C}\left(\hat{\boldsymbol{\phi}}_S\right) \quad (\text{B.5})$$

To find the partials of the measurement model the pseudomeasurment can be approximated as a function of the sensor misalignment error and the attitude error rotated into the sensor frame. Therefore, it can be shown that

$$\boldsymbol{\Delta} \approx \boldsymbol{\psi} + \boldsymbol{\phi}_S = \boldsymbol{\psi} + \bar{\mathbf{C}}_{SB}\boldsymbol{\phi} \quad (\text{B.6})$$

Leading to

$$\frac{\partial \boldsymbol{\Delta}}{\partial \boldsymbol{\psi}} = \mathbf{I}_{3 \times 3} \quad (\text{B.7})$$

$$\frac{\partial \boldsymbol{\Delta}}{\partial \boldsymbol{\phi}} = \bar{\mathbf{C}}_{SB} \quad (\text{B.8})$$

The attitude error covariance matrix can be seen below, with  $\sigma_{\text{bs}}$  representing the error in the cross-boresight direction, and  $\sigma_{\text{roll}}$  the roll error.

$$\mathbf{R}_\Delta = \begin{bmatrix} \sigma_{\text{bs}}^2 & 0 & 0 \\ 0 & \sigma_{\text{bs}}^2 & 0 \\ 0 & 0 & \sigma_{\text{roll}}^2 \end{bmatrix} \quad (\text{B.9})$$

## B.2 Direction Measurements

The direction vector of Phobos and Deimos can be found by their coordinates on the star tracker detector plane. This measurement was modeled by first finding the true displacement vector  $\mathbf{s}_S$  between the spacecraft and each moon, where  $\boldsymbol{\rho}_P$  is the target's position. JPL's SPICE toolbox was used for all ephemeris calculations [41]. The true displacement vector can be modeled as

$$\mathbf{s}_S = \mathbf{C}_{SP}(\boldsymbol{\rho}_P - \mathbf{r}_P) \quad (\text{B.10})$$

Plugging in the state estimates leads to estimated displacement vector  $\widehat{\mathbf{s}}_S$ .

$$\widehat{\mathbf{s}}_S \equiv \widehat{\mathbf{C}}_{SP} (\boldsymbol{\rho}_P - \widehat{\mathbf{r}}_P) \quad (\text{B.11})$$

Assuming a pinhole model for the camera the direction measurements  $\widetilde{\mathbf{z}}$  can be seen below. With the estimated direction measurement  $\widehat{\mathbf{z}}$  found by using  $\widehat{\mathbf{s}}_S$ .

$$\widetilde{\mathbf{z}}_S = \frac{f}{\delta_{\text{pix}}} \begin{bmatrix} s_{S,x}/s_{S,z} \\ s_{S,y}/s_{S,z} \end{bmatrix} + \boldsymbol{\nu}_z \quad (\text{B.12})$$

The nonzero partials derivatives can be seen as

$$\frac{\partial \mathbf{z}_S}{\partial \mathbf{r}_P} = -\frac{\partial \mathbf{z}}{\partial \mathbf{s}_S} \mathbf{C}_{SP} \quad (\text{B.13})$$

$$\frac{\partial \mathbf{z}_S}{\partial \boldsymbol{\psi}} = \frac{\partial \mathbf{z}}{\partial \mathbf{s}_S} \frac{\partial \mathbf{s}_S}{\partial \boldsymbol{\psi}} \quad (\text{B.14})$$

$$\frac{\partial \mathbf{z}_S}{\partial \phi} = -\frac{\partial \mathbf{z}_S}{\partial \mathbf{s}_S} \mathbf{C}_{SB} \frac{\partial \mathbf{s}_B}{\partial \phi} \quad (\text{B.15})$$

For each moon, Phobos and Deimos, the direction measurements depend on the star trackers ability to estimate the centroid of each moon.

$$\mathbf{R}_{moon} = \sigma_{moon}^2 I_{2 \times 2} \quad (\text{B.16})$$

The value  $\sigma_{moon}$  is approximated using Figure B.1 which shows how the Sinclair Interplanetary ST-16RT star tracker centroid error varies with brightness, the photometric brightness of planetary bodies

$$V = V(1, 0) + 2.5 \log_{10} \left( \frac{r^2 r_{\text{Sun}}^2}{P(\chi)} \right) \quad (\text{B.17})$$

and from Lester et al. [42] the phase function

$$P(\chi) = \frac{2}{3} \left[ \left( 1 - \frac{\chi}{\pi} \right) \cos \chi + \frac{1}{\pi} \sin \chi \right] \quad (\text{B.18})$$

From above  $V(1, 0)$  is the tabulated visual magnitude at opposition and a distance of 1 au, for each moon the values  $V_{Phobos}(1, 0) = 11.8$  and  $V_{Deimos}(1, 0) = 12.89$  were used [43]. As well,  $r_{Sun}$  is the distance between Mars and the Sun, and  $r$  is the estimated distance from Mars.

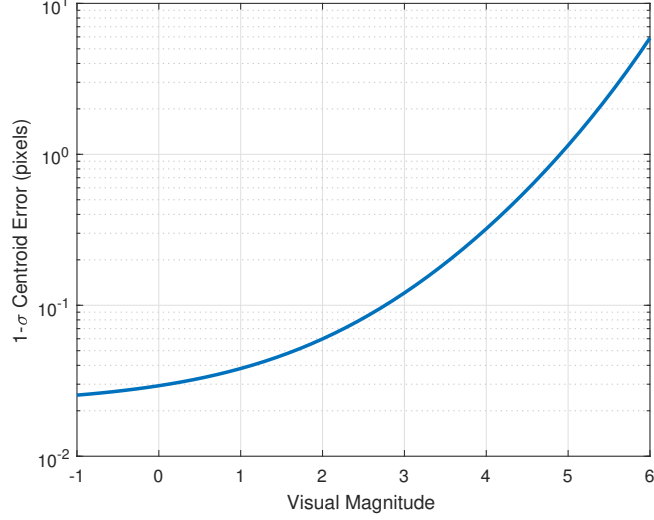


FIGURE B.1: Approximate relationship between visual magnitude and centroid error for the ST-16RT [22].

### B.3 Position Measurements

The Mars displacement vector represents a pseudomeasurement that is derived from the size and position of the Mars disk in the field of view of the sensor. Enright et al. [22] provides a full overview on estimating the displacement vector from the detector-plane observations, however written in terms of the filter state the pseudomeasurement is simply a rotation of the spacecrafts position into Frame- $S$ . The noisy pseudomeasurement is

$$\tilde{\mathbf{z}}_M = -\mathbf{C}_{SP}\mathbf{r}_P + \boldsymbol{\nu}_r \quad (\text{B.19})$$

With the estimated Mars vector being

$$\mathbf{z}_M = -\hat{\mathbf{C}}_{SP}\hat{\mathbf{r}}_P \quad (\text{B.20})$$

with partials

$$\frac{\partial \mathbf{z}_M}{\partial \mathbf{r}_P} = -\mathbf{C}_{SP} \quad (\text{B.21})$$

$$\frac{\partial \mathbf{z}_M}{\partial \psi} = -\frac{\partial \mathbf{r}_S}{\partial \psi} \quad (\text{B.22})$$

$$\frac{\partial \mathbf{z}_M}{\partial \phi} = -\mathbf{C}_{SB} \frac{\partial \mathbf{r}_B}{\partial \phi} \quad (\text{B.23})$$

The Mars position covariance matrix was developed by Hikes et al. [44]. The covariance matrix can be seen as

$$\mathbf{R}_M = R \begin{bmatrix} \frac{4}{2\xi - \sin(2\xi)} & 0 & 0 \\ 0 & \frac{\xi}{D} & \frac{\sqrt{r^2 - r_M^2} \sin \xi}{Dr_M} \\ 0 & \frac{\sqrt{r^2 - r_M^2} \sin \xi}{Dr_M} & \frac{(r^2 - r_M^2)(2\xi + \sin(2\xi))}{4Dr_M^2} \end{bmatrix} \quad (\text{B.24})$$

where,

$$R = \frac{\sigma_{\text{pix}}^2 r^4 \delta_{\text{pix}} \xi}{N_{\text{limb}} f^2 (r^2 - r_M^2)} \quad (\text{B.25})$$

and

$$D = \frac{\xi}{4} (2\xi + \sin(2\xi)) - \sin^2 \xi \quad (\text{B.26})$$

$\mathbf{R}_s$  is originally in terms of a frame aligned with cusp and limb directions but can be brought into the S-frame for use in the filter.

$$\mathbf{R}_{M,S} = \mathbf{C}_{ST} \mathbf{C}_3(\beta) \mathbf{R}_s \mathbf{C}_3^T(\beta) \mathbf{C}_{TS} \quad (\text{B.27})$$

Further terms include  $\delta_{\text{pix}}$  as the pixel spacing,  $f$  the focal length of star tracker,  $\xi$  the half-width angle of the illuminated limb,  $N_{\text{limb}}$  the number of points extracted from the limb curve and found as  $N_{\text{limb}} = \frac{2\xi}{k\delta_{\text{pix}}}$ ,  $r_M$  is the radius of Mars, and  $\sigma_{\text{pix}}$  is the terminator noise parameter.

## B.4 Phase Angle Measurements

The illuminated fraction of Mars can be used to find the phase angle  $\chi$  between the spacecraft, the Sun and Mars [45]. Written in terms of state variables

$$\tilde{\mathbf{z}}_\chi = \arccos \left( \frac{\mathbf{r}_P^T \boldsymbol{\rho}_{\text{Sun},P}}{r_P R_{\text{Sun},P}} \right) + \boldsymbol{\nu}_\chi \quad (\text{B.28})$$

The partial derivative can be taken as

$$\frac{\partial \mathbf{z}_\chi}{\partial \mathbf{r}_P} = - \left( \frac{\boldsymbol{\rho}_{\text{sun},P}^T}{r^3 \sin \chi} \right) \left( r^2 I_{3 \times 3} - \mathbf{r}_P \mathbf{r}_P^T \right) \quad (\text{B.29})$$

The covariance matrix for the phase angle measurements is described in the next section.

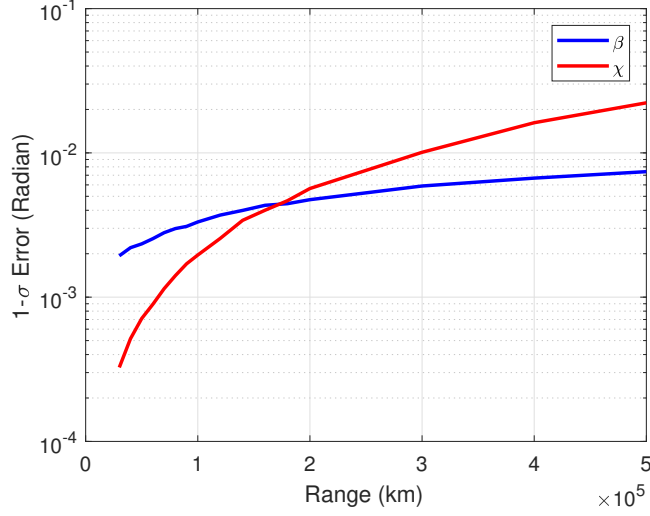


FIGURE B.2: Measurement errors for phase and terminator angles [22].

## B.5 Terminator Measurements

The terminator angle  $\beta$  is defined as the orientation of the terminator ellipse on the imaged disk of Mars in the field of view of the sensor. The major axis of the terminator ellipse is perpendicular to the plane that contains the spacecraft, Sun and Mars, and thus must point in the direction

$$\boldsymbol{\kappa}_P = -\mathbf{r}_P^\times \boldsymbol{\rho}_{\text{Sun},P} \quad (\text{B.30})$$

The observed terminator angle  $\beta$  is caused by the projection of this direction vector into the  $T$ -Frame. While, the estimated terminator angle  $\hat{\beta}$  is simply found using the estimated state vector.

$$\tilde{\mathbf{z}}_\beta = \text{atan}\left(\frac{\kappa_{T,y}}{\kappa_{T,x}}\right) + \boldsymbol{\nu}_\beta \quad (\text{B.31})$$

where,

$$\boldsymbol{\kappa}_T = \mathbf{C}_{TS} \mathbf{C}_{SP} \boldsymbol{\kappa}_P \quad (\text{B.32})$$

The partial derivative is rather extensive, therefore refer to Enright et al. [22] for its derivation. Lastly, the covariance matrix for the phase  $\mathbf{R}_\chi$  and terminator  $\mathbf{R}_\beta$  measurements were found from an empirical method by generating interpolants to estimate the measurement errors under various conditions. Simulated images were generated, then analyzed and error statistics were found from the estimates of the phase and terminator angles to generate these interpolants. Figure B.2 shows the  $1-\sigma$  measurement errors used for both the phase and terminator angles found from the empirical method.

# Bibliography

- [1] Navin Jerath and Hiroshi Ohtakay. Mariner ix optical navigation using mars lit limb. *Journal of Spacecraft and Rockets*, 11(7):505–511, 1974. doi: 10.2514/3.62114.
- [2] William M Owen, Philip J Dumont, and Coralie D Jackman. Optical navigation preparations for new horizons pluto flyby. International Symposium on Space Flight Dynamics, October 2012.
- [3] Simon Nolet, Stephen D Gillam, and Jeremy B Jones. Optical navigation planning process for the cassini solstice mission. American Astronomical Society/American Institute of Aeronautics and Astronautics, February 2011.
- [4] S Desai, D Han, S Bhaskaran, B Kennedy, T McElrath, G.W Null, and J.E Riedel. Autonomous Optical Navigation (AutoNav) DS1 Technology Validation Report. Technical report, Jet Propulsion Laboratory.
- [5] Shyam Bhaskaran. *Autonomous Navigation for Deep Space Missions*. March 2013. doi: 10.2514/6.2012-1267135. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2012-1267135>.
- [6] John Crassidis and John Junkins. *Optimal Estimation of Dynamic Systems*. Number Second Edition. Taylor & Francis Group, 2012. URL <https://www-taylorfrancis-com.ezproxy.lib.ryerson.ca/books/9781439839867>.
- [7] Tom Dzamba, John Enright, David A. Sinclair, Kofi Amankwah, Ronny Votel, Ilija Jovanovic, and Geoffrey R. McVittie. Success by 1000 improvements: Flight qualification of the st-16 star tracker. 2014.
- [8] C. Van Loan. Computing integrals involving the matrix exponential. *IEEE Transactions on Automatic Control*, 23(3):395–404, June 1978. ISSN 0018-9286. doi: 10.1109/TAC.1978.1101743.
- [9] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

- [10] Jeffrey K. Uhlmann Simon J. Julier. New extension of the kalman filter to nonlinear systems, 1997. URL <https://doi.org/10.1117/12.280797>.
- [11] Eric A. Wan and Rudolph van der Merwe. The Unscented Kalman Filter. In *Kalman Filtering and Neural Networks*, pages 221–280. Wiley-Blackwell, March 2002. ISBN 978-0-471-22154-8. doi: 10.1002/0471221546.ch7. URL <http://onlinelibrary.wiley.com/doi/abs/10.1002/0471221546.ch7>.
- [12] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings F - Radar and Signal Processing*, 140(2):107–113, April 1993. ISSN 0956-375X. doi: 10.1049/ip-f-2.1993.0015.
- [13] Drew Bagnell. Particle filters: The good, the bad, the ugly, September 2014. URL [http://www.cs.cmu.edu/~16831-f14/notes/F11/16831\\_lecture04\\_tianyu1.pdf](http://www.cs.cmu.edu/~16831-f14/notes/F11/16831_lecture04_tianyu1.pdf). Accessed: 2018-1-6.
- [14] Arnaud Doucet, Nando de Freitas, and Neil Gordon. *An Introduction to Sequential Monte Carlo Methods*. Springer, New York, NY, 2001.
- [15] T. Li, M. Bolic, and P. M. Djuric. Resampling methods for particle filtering: Classification, implementation, and strategies. *IEEE Signal Processing Magazine*, 32(3):70–86, May 2015. ISSN 1053-5888.
- [16] Yang Cheng and John L. Crassidis. Particle filtering for attitude estimation using a minimal local-error representation. *Journal of Guidance, Control, and Dynamics*, 33(4):1305–1310, 2010. doi: 10.2514/1.47236. URL <https://doi.org/10.2514/1.47236>.
- [17] John B. Moore. Discrete-time fixed-lag smoothing algorithms. *Automatica*, 9(2):163–173, March 1973. ISSN 0005-1098. doi: 10.1016/0005-1098(73)90071-X. URL <http://www.sciencedirect.com/science/article/pii/000510987390071X>.
- [18] J Moore. Fixed-lag smoothing for nonlinear systems with discrete measurements. *Information Sciences*, 6:151–160, 1972.
- [19] Christopher V. Rao, James B. Rawlings, and Jay H. Lee. Constrained linear state estimation—a moving horizon approach. *Automatica*, 37:1619–1628, 10 2001. doi: 10.1016/S0005-1098(01)00115-7.
- [20] Eric L. Haseltine and James B. Rawlings. Critical evaluation of extended kalman filtering and moving-horizon estimation. *Industrial & Engineering Chemistry Research*, 44(8):2451–2460, 2005. doi: 10.1021/ie034308l. URL <https://doi.org/10.1021/ie034308l>.

- [21] Jeroen Vandersteen, Moritz Diehl, Conny Aerts, and Jan Swevers. Spacecraft Attitude Estimation and Sensor Calibration Using Moving Horizon Estimation. *Journal of Guidance, Control, and Dynamics*, 36(3):734–742, 2013. ISSN 0731-5090. doi: 10.2514/1.58805. URL <https://doi.org/10.2514/1.58805>.
- [22] John Enright, Ilija Jovanovic, Laila Kazemi, Harry Zhang, and Tom Dzamba. Autonomous optical navigation using nanosatellite-class instruments: a Mars approach case study. *Celestial Mechanics and Dynamical Astronomy*, 130(2), February 2018. ISSN 0923-2958, 1572-9478. doi: 10.1007/s10569-017-9800-x. URL <http://link.springer.com/10.1007/s10569-017-9800-x>.
- [23] John Christian and E. Glenn Lightsey. *Integrated Performance of an Autonomous Optical Navigation System for Space Exploration*. doi: 10.2514/6.2010-8786. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2010-8786>.
- [24] V. Franzese, P. Di Lizia, and F. Topputo. Autonomous optical navigation for the lunar meteoroid impacts observer. *Journal of Guidance, Control, and Dynamics*, 42(7):1579–1586, 2019. doi: 10.2514/1.G003999. URL <https://doi.org/10.2514/1.G003999>.
- [25] Michael Paluszek, Michael Littman, and Joseph Mueller. *Optical Navigation System*. doi: 10.2514/6.2010-3462. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2010-3462>.
- [26] Hanspeter Schaub and John L. Junkins. *Analytical Mechanics of Aerospace Systems*. American Institute of Aeronautics and Astronautics, 2003.
- [27] Xiaodong Liu, Hexi Baoyin, and Xingrui Ma. Five special types of orbits around mars. *Journal of Guidance, Control, and Dynamics*, 33(4):1294–1301, 2010. ISSN 0731-5090. doi: 10.2514/1.48706. URL <https://doi.org/10.2514/1.48706>.
- [28] J.R. Wertz, J.L. Cloots, J.T. Collins, S.D. Dawson, G Gurevich, B.K. Sato, and L.J. Hansen. Autonomous orbit control: Initial flight results from uosat-12. American Astronomical Society, February 2000.
- [29] James Wertz. Autonomous navigation and autonomous orbit control in planetary orbits as a means of reducing operations cost. International Symposium on Reducing the Cost of Spacecraft Ground Systems and Operations, 2013.
- [30] A. Austin, A. Nelessen, B. Strauss, J. Ravich, M. Jesick, E. Venkatapathy, R. Beck, P. Wercinski, M. Aftosmis, M. Wilder, G. Allen, R. Braun, M. Werner, and E. Roelke. Smallsat aerocapture to enable a new paradigm of planetary missions. In *2019 IEEE Aerospace Conference*, pages 1–20, March 2019. doi: 10.1109/AERO.2019.8742220.



- [31] Howard D. Curtis. Chapter 12 - introduction to orbital perturbations. In Howard D. Curtis, editor, *Orbital Mechanics for Engineering Students (Third Edition)*, pages 651 – 720. Butterworth-Heinemann, Boston, third edition edition, 2014. ISBN 978-0-08-097747-8. doi: <https://doi.org/10.1016/B978-0-08-097747-8.00012-8>.
- [32] Justin Atchison, Ryan Mitch, and Andy Rivkin. Swarm flyby gravimetry. NIAC Phase I Final Report, The Johns Hopkins University Applied Physics Laboratory, April 2015.
- [33] P. Rosenblatt, V. Lainey, S. Le Maistre, J. C. Marty, V. Dehant, M. Pätzold, T. Van Hoolst, and B. Häusler. Accurate mars express orbits to improve the determination of the mass and ephemeris of the martian moons. 56(7):1043 – 1053, 2008. ISSN 0032-0633. doi: <https://doi.org/10.1016/j.pss.2008.02.004>. URL <http://www.sciencedirect.com/science/article/pii/S0032063308000330>.
- [34] Lorraine Fesq, Patricia Beauchamp, Amanda Donner, Rob Bocchino, Brian Kennedy, Faiz Mirza, Swati Mohan, David Sternberg, Matthew W. Smith, Martina Troesch, and Mary Knapp Knapp. Extended mission technology demonstrations using the asteria spacecraft. *Institute of Electrical and Electronics Engineers*, 2019.
- [35] Robert L. Scott. *A Method for Optical Tracking of On-Orbit Servicing Operations in Geostationary Orbit Using Speckle Interferometry*. PhD thesis, Carleton University, 2015.
- [36] Miranda Straub and John A. Christian. *Autonomous Optical Navigation for Earth-Observing Satellites using Coastline Matching*. January 2015. doi: 10.2514/6.2015-1334. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2015-1334>.
- [37] S. P. Kau. Autonomous satellite orbital navigation using known and unknown earth landmarks. St. Petersburg, Fl, August 1975. AIAA, Guidance and Control Conference, Honeywell, Inc.
- [38] F. H. Schlee and N. F. Toda. Autonomous orbital navigation by timatical tracking of unknown landmarks. *Journal of Spacecraft and Rockets*, 4(12):1644–1648, 1967. doi: 10.2514/3.29147. URL <https://doi.org/10.2514/3.29147>.
- [39] Simone Battistini. Autonomous cubesat orbit determination via ukf state estimation. European Cubesat Symposium, January 2012.
- [40] M. J. Tenny and J. B. Rawlings. Efficient moving horizon estimation and nonlinear model predictive control. In *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*, volume 6, pages 4475–4480. Institute of Electrical and Electronics Engineers, May 2002. doi: 10.1109/ACC.2002.1025355.

- [41] Charles H. Acton. Ancillary data services of NASA's Navigation and Ancillary Information Facility. *Planetary and Space Science*, 44(1):65–70, January 1996. ISSN 0032-0633. doi: 10.1016/0032-0633(95)00107-7. URL <http://www.sciencedirect.com/science/article/pii/0032063395001077>.
- [42] T. P. Lester, M. L. McCall, and J. B. Tatum. Theory of planetary photometry. *Journal of the Royal Astronomical Society of Canada*, 73:233–257, October 1979. ISSN 0035-872X. URL <http://adsabs.harvard.edu/abs/1979JRASC..73..233L>.
- [43] David Williams. NASA-Goddard Mars fact sheet, June 2017. URL <https://nssdc.gsfc.nasa.gov/planetary/factsheet/marsfact.html>.
- [44] Jacob Hikes, Andrew J. Liounis, and John A. Christian. Parametric Covariance Model for Horizon-Based Optical Navigation. *Journal of Guidance, Control, and Dynamics*, 40(1):170–178, October 2016. ISSN 0731-5090. doi: 10.2514/1.G000708. URL <http://arc.aiaa.org/doi/10.2514/1.G000708>.
- [45] John Enright. Moon-tracking modes for star trackers. *Journal of Guidance, Control, and Dynamics*, 33(1):171–185, 2010. doi: 10.2514/1.42716. URL <https://doi.org/10.2514/1.42716>.