Analysis of Asteroid Landing Capabilities Using Autonomous Optical Navigation

by

Corwin Grant Olson, B.S.

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

The University of Texas at Austin

May 2009

Analysis of Asteroid Landing Capabilities Using Autonomous Optical Navigation

Approved by
Supervising Committee:

Bob E. Schutz

Wallace T. Fowler

Dedication

For Princess.

Acknowledgements

The Small-body Autonomous Navigation Toolset (SANT) was initially developed at the National Aeronautics and Space Administration (NASA) Jet Propulsion Laboratory (JPL) during the summer of 2008. Much of the code development both before and during my involvement was accomplished by Dr. Shyam Bhaskaran at JPL. I worked with Dr. Bhaskaran at JPL as I started work on this tool and his regular guidance was invaluable as we developed the tool further. I would like to thank Dr. Bhaskaran for giving me the opportunity to work at JPL, for allowing me to continue to work on this topic for my thesis from Austin, and for his help on this thesis. Additionally, I thank Dr. Bhaskaran and JPL for providing funding to me for this work.

I thank my advisor Dr. Bob Schutz for giving me the opportunity to obtain my Masters degree in Aerospace Engineering and for the opportunity to work for the Center for Space Research (CSR) during my time here at the University of Texas (UT) as a graduate student. I also thank Dr. Wallace Fowler for seven years of guidance and teaching and for his help on this thesis as my reader. I would also like to thank the many other professors I have had at the University of Texas, and in particular those in the Aerospace Engineering Department, for all of the knowledge and guidance I have received as both an undergraduate and graduate student here at UT.

I would like to deeply thank my family for the excellent support and love I received from them throughout this endeavor, as well as the amazing job they did raising me. I would also like to thank my Dad for proofreading this thesis. Also I thank my many amazing friends here at UT and the many friends I made in Houston, TX, Greenbelt, MD, and Pasadena, CA while working in those cities. And last but certainly not least, I would like to thank Lindsey for providing non-stop support throughout this entire process and in general making my life a lot better than it would have been otherwise.

May 6th, 2009

Abstract

Analysis of Asteroid Landing Capabilities Using Autonomous Optical Navigation

Corwin Olson , M.S.E.

The University of Texas at Austin, 2009

Supervisor: Bob Schutz

Autonomous Optical Navigation (AON) for landing on small bodies such as asteroids is a major new field of research of importance for future missions to small bodies. Before the development of the Small-body Autonomous Navigation Toolset (SANT) used in this study, there was no tool available to the aerospace industry for measuring the impact of navigational errors on the ability of a spacecraft to land within a certain distance from a target on the surface of an asteroid. Using SANT to perform Monte Carlo analysis of spacecraft landing error on a small asteroid, navigation error sources were varied in an attempt to learn what aspects of the spacecraft landing footprint on the small-body surface were sensitive to these errors. It was discovered by reducing by 50% and increasing by 200% each of the major error sources that the size and shape of the footprint were most sensitive to maneuver execution error, with the average landing

target error magnitude decreasing by 39% to 20 meters with 50% maneuver execution error and increasing by 80% to 58 meters with 200% maneuver execution error. The number of outliers was most sensitive to the initial state error, with the number of trajectories completely missing the asteroid decreasing by 91% to 3 trajectories with 50% initial state error and increasing by 133% to 77 trajectories with 200% initial state error.

Table of Contents

List of Tables xiii

List of Figures xiv

List of Illustrations xvii

CHAPTER 1 **1**

Introduction 1

1.1 Historical contributions 2

1.2 Focus and Objectives of Thesis 3

1.3 Outline of Thesis 4

CHAPTER 2 **5**

Background Theory 5

2.1 Orbital Forces and Dynamics 5

2.1.1 General Motion 5

2.1.2 Perturbed Motion 6

2.1.2.1 Third Body Perturbations 6

2.1.2.2 Solar Radiation Pressure 7

2.1.2.3 Gravitational Harmonics 8

2.2 Targeting 8

2.3 Reference Frames 9

2.3.1 Inertial body-centered reference frame 9

2.3.2 Body-fixed reference frame 10

2.3.3 Spherical reference frame 11

2.3.4 View2 reference frame 11

2.3.5 Topocentric reference frame 12

2.3.6 Geodetic reference frame 12

2.3.6 Camera-centered-Camera-fixed reference frame 14

CHAPTER 3 **15**

Small-body Autonomous Navigation Toolset 15
3.1 Tool Requirements and General Assumptions 15
3.2 Models.....15
3.3 Tool Operations 16
3.3.1 Executive Script.....16
3.3.2 Propagation and Event Detection Function 19
3.3.3 Sliding Window Function.....19
3.4 Landmarks.....20
3.4.1 Landmark Creation Function 21
3.4.2 Landmarks Selection Function 22
3.4.3 Processing Pictures of Landmarks Function.....22
3.4.4 Generate Landmark Observations Function 23
3.4.5 Pixel/Line Calculator Function.....24
3.5 Kinematic Filter 26
3.5.1 Kinematic Position Partial Derivatives Function.....28
3.5.2 Kinematic Pointing Partial Derivatives Function 30
3.6 Dynamical Filter 30
3.7 Maneuvers.....32
3.7.1 Maneuver implementation 32
3.7.2 Compute Maneuver Function 33
3.7.3 Orbit Determination Cut-off 34
3.8 Pointing Calculation Function 35
3.9 Errors.....36
3.9.1 Initial State Error.....37
3.9.2 Observation Error.....37
3.9.3 Execution Error in Maneuver.....38
3.9.4 Modeling Error.....39
3.9.4.1 Third Body Perturbations.....39
3.9.4.2 Solar Radiation Pressure Perturbations.....40
3.9.4.3 Gravitational Harmonics Perturbations.....41

3.9.5 Attitude Error 41
3.10 Monte Carlo 43
3.10.1 Monte Carlo versus Covariance Analysis 43
3.10.2 Monte Carlo Function 44

CHAPTER 4 **46**

Simulation Setup 46

CHAPTER 5 **51**

Simulation Results 51
5.1 Baseline Profile 52
5.2 50% Maneuver Execution errors 62
5.3 200% Maneuver Execution errors 66
5.4 50% Observation errors 69
5.5 200% Observation errors 72
5.6 50% Initial State Errors 75
5.7 200% Initial State Errors 78
5.8 No Attitude Error 82
5.9 Adding all three perturbations to nominal model 85
5.10 Adding only gravity harmonics perturbation to nominal model 89
5.11 Adding only solar radiation pressure perturbation to nominal model 93
5.12 Adding only third body perturbations to nominal model 96
5.13 Adding one maneuver (2 total): 100
5.14 Adding two maneuvers (3 Total): 103
5.15 Setting OD Cutoff to 5 Hours 107
5.16 No onboard navigation 111

CHAPTER 6 **116**

Summary 116
6.1 Conclusions 116
6.1 Recommendations for Future Work 118

Appendix A: Batch Filter 121

Appendix B: Baseline Profile Settings 124

Appendix C: Additional Results 130

C.1 Maneuver Time ahead one hour 130

C.2 Maneuver Time behind one hour 133

C.3 Setting OD-Cutoff to 10 Hours 136

C.4 Setting OD-Cutoff to 21 hours 139

C.5 Adding observations every 10 min in the hour before the maneuver 142

C.6 Turning off Sliding window 145

C.7 Testing only one error source at a time 148

C.7.1 Adding: Initial State Error Only 148

C.7.2 Adding: Observation Error Only 151

C.7.3 Adding: Maneuver Execution Error Only 154

C.7.4 Adding: Attitude Error Only 157

C.7.5 Adding: No perturbations on the nominal model only 160

List of acronyms 164

Bibliography 165

VITA 168

List of Tables

Table 5.1: Baseline Monte Carlo Statistics 56
Table 5.2: Monte Carlo Statistics - 50% maneuver execution errors 63
Table 5.3: Monte Carlo Statistics - 200% maneuver execution errors 66
Table 5.4: Monte Carlo Statistics – 50% observation errors 69
Table 5.5: Monte Carlo Statistics – 200% observation errors 72
Table 5.6: Monte Carlo Statistics – 50% initial state errors 75
Table 5.7: Monte Carlo Statistics – 200% initial state errors 79
Table 5.8: Monte Carlo Statistics – No Attitude Error 82
Table 5.9: Monte Carlo Statistics – All 3 Perturbations on Nominal Model 85
Table 5.10: Monte Carlo Statistics – Only Gravity Harmonics Perturbations 89
Table 5.11: Monte Carlo Statistics – Only SRP Perturbation 93
Table 5.12: Monte Carlo Statistics – Only Third Body Perturbations 97
Table 5.13: Monte Carlo Statistics – Adding Two Maneuvers 100
Table 5.14: Monte Carlo Statistics – Adding Two Maneuvers 104
Table 5.15: Monte Carlo Statistics – Setting OD cutoff to 5 hours 108
Table 5.16: Monte Carlo Statistics – No Onboard Navigation 111

List of Figures

Figure 5.1: Baseline Profile Body-fixed Trajectory 53
Figure 5.2: First seed Monte Carlo Inertial Trajectory 55
Figure 5.3: Baseline True Target Error Scatter Plot 57
Figure 5.4: Baseline Nominal Target Error Scatter Plot 58
Figure 5.5: Baseline Knowledge Error Scatter Plot 59
Figure 5.6: Baseline Final Positions 60
Figure 5.7: Baseline Final Positions - Rotated 61
Figure 5.8: Baseline Final Positions – Rotated and Zoomed 62
Figure 5.9: True Target Error Scatter Plot - 50% maneuver execution errors 64
Figure 5.10: Knowledge Error Scatter Plot - 50% maneuver execution errors 65
Figure 5.11: Final Positions Scatter Plot - 50% maneuver execution errors 66
Figure 5.12: True Target Error Scatter Plot - 200% maneuver execution errors 67
Figure 5.13: Knowledge Error Scatter Plot - 200% maneuver execution errors 68
Figure 5.14: Final Positions Scatter Plot - 200% maneuver execution errors 69
Figure 5.15: True Target Error Scatter Plot - 50% observation errors 70
Figure 5.16: Knowledge Error Scatter Plot - 50% observation errors 71
Figure 5.17: Final Positions Plot - 50% observation errors 72
Figure 5.18: True Target Error Scatter Plot - 200% observation errors 73
Figure 5.19: Knowledge Error Scatter Plot - 200% observation errors 74
Figure 5.20: Final Positions Plot - 200% observation errors 75
Figure 5.21: True Target Error Scatter Plot - 50% initial state errors 76
Figure 5.22: Knowledge Error Scatter Plot - 50% initial state errors 77
Figure 5.23: Final Positions Plot - 50% initial state errors 78
Figure 5.24: True Target Error Scatter Plot - 200% initial state errors 80
Figure 5.25: Knowledge Error Scatter Plot - 200% initial state errors 81
Figure 5.26: Landing Positions Plot - 200% initial state errors 82
Figure 5.27: True Target Error Scatter Plot – No Attitude Errors 83
Figure 5.28: Knowledge Error Scatter Plot – No Attitude Errors 84
Figure 5.29: Final Positions Plot – No Attitude Errors 85
Figure 5.30: True Target Error Scatter Plot – All 3 Perturbations on Nominal Model 86
Figure 5.31: Knowledge Error Scatter Plot – All 3 Perturbations on Nominal Model 87
Figure 5.32: Final Positions Plot – All 3 Perturbations on Nominal Model 88
Figure 5.33: True Target Error Scatter Plot – Only Gravity Harmonics Perturbations on Nominal Model 90
Figure 5.34: Knowledge Error Scatter Plot – Only Gravity Harmonics Perturbations on Nominal Model 91
Figure 5.35: Final Positions Plot – Only Gravity Harmonics Perturbations on Nominal Model 92
Figure 5.36: True Target Error Scatter Plot – Only SRP Perturbation on Nominal Model 94
Figure 5.37: Knowledge Error Scatter Plot – Only SRP Perturbation on Nominal Model 95

Figure 5.38: Final Positions Plot – Only SRP Perturbation on Nominal Model 96
Figure 5.39: True Target Error Scatter Plot – Only Third Body Perturbations on Nominal Model 98
Figure 5.40: Knowledge Error Scatter Plot – Only Third Body Perturbations on Nominal Model 99
Figure 5.41: Final Positions Plot – Only Third Body Perturbations on Nominal Model 100
Figure 5.42: True Target Error Scatter Plot – Adding One Maneuver 101
Figure 5.43: Knowledge Error Scatter Plot – Adding One Maneuver 102
Figure 5.44: Final Positions Plot – Adding One Maneuver 103
Figure 5.45: True Target Error Scatter Plot – Adding Two Maneuvers 105
Figure 5.46: Knowledge Error Scatter Plot – Adding Two Maneuvers 106
Figure 5.47: Final Positions Plot – Adding Two Maneuvers 107
Figure 5.48: True Target Error Scatter Plot – Setting OD cutoff to 5 hours 109
Figure 5.49: Knowledge Error Scatter Plot – Setting OD cutoff to 5 hours 110
Figure 5.50: Final Positions Plot – Setting OD cutoff to 5 hours 111
Figure 5.51: True Target Error Scatter Plot – No Onboard Navigation 112
Figure 5.52: Nominal Target Error Scatter Plot – No Onboard Navigation 113
Figure 5.53: Knowledge Error Scatter Plot – No Onboard Navigation 114
Figure 5.54: Final Positions Plot – No Onboard Navigation 115
Figure C.1: True Target Error Scatter Plot - Maneuver Time Ahead One Hour 131
Figure C.2: Knowledge Error Scatter Plot - Maneuver Time Ahead One Hour 132
Figure C.3: Final Positions Scatter Plot - Maneuver Time Ahead One Hour 133
Figure C.4: True Target Error Scatter Plot - Maneuver Time Behind One Hour 134
Figure C.5: Knowledge Error Scatter Plot - Maneuver Time Behind One Hour 135
Figure C.6: Final Positions Scatter Plot - Maneuver Time Behind One Hour 136
Figure C.7: True Target Error Scatter Plot – Setting OD-Cutoff to 10 hours 137
Figure C.8: Knowledge Error Scatter Plot – Setting OD-Cutoff to 10 hours 138
Figure C.9: Final Positions Scatter Plot – Setting OD-Cutoff to 10 hours 139
Figure C.10: True Target Error Scatter Plot – Setting OD-Cutoff to 21 Hours 140
Figure C.11: Knowledge Error Scatter Plot – Setting OD-Cutoff to 21 Hours 141
Figure C.12: Final Positions Scatter Plot – Setting OD-Cutoff to 21 Hours 142
Figure C.13: True Target Error Scatter Plot – Observations Added Every 10 Minutes in the Hour Before Maneuver 143
Figure C.14: Knowledge Error Scatter Plot – Observations Added Every 10 Minutes in the Hour Before Maneuver 144
Figure C.15: Final Positions Scatter Plot – Observations Added Every 10 Minutes in the Hour Before Maneuver 145
Figure C.16: True Target Error Scatter Plot – Sliding Window Off 146
Figure C.17: Knowledge Error Scatter Plot – Sliding Window Off 147
Figure C.18: Final Positions Scatter Plot – Sliding Window Off 148
Figure C.19: True Target Error Scatter Plot – Initial State Error Only 149
Figure C.20: Knowledge Error Scatter Plot – Initial State Error Only 150
Figure C.21: Final Positions Scatter Plot – Initial State Error Only 151
Figure C.22: True Target Error Scatter Plot – Observation Error Only 152
Figure C.23: Knowledge Error Scatter Plot – Observation Error Only 153

Figure C.24: Final Positions Scatter Plot – Observation Error Only 154
Figure C.25: True Target Error Scatter Plot – Maneuver Execution Error Only 155
Figure C.26: Knowledge Error Scatter Plot – Maneuver Execution Error Only 156
Figure C.27: Final Positions Scatter Plot – Maneuver Execution Error Only 157
Figure C.28: True Target Error Scatter Plot – Attitude Error Only 158
Figure C.29: Knowledge Error Scatter Plot – Attitude Error Only 159
Figure C.30: Final Positions Scatter Plot – Attitude Error Only 160
Figure C.31: True Target Error Scatter Plot – No Perturbations on the Nominal Model 161
Figure C.32: Knowledge Error Scatter Plot – No Perturbations on the Nominal Model 162
Figure C.33: Final Positions Scatter Plot – No Perturbations on the Nominal Model ... 163

List of Illustrations

Illustration 5.1: Missing the asteroid 108

CHAPTER 1

Introduction

For the majority of the history of mankind, we have known asteroids and comets only as the dim points of light that occasionally cross the night sky. Discovery of the true nature of these small bodies within the solar system in the Nineteenth Century [27] was quickly followed by the knowledge that they could pose a potent threat to all life on Earth. Later there were more optimistic realizations, such as how these bodies could serve as treasure troves of scientific data about the initial formation of the solar system as well as a source of raw materials [28].

Mankind's recently-acquired ability to escape the gravity of this planet allows us to explore these bodies in far greater detail via man-made spacecraft missions. These missions are critical to garnering greater understanding of this dual threat and resource. We will need this knowledge if we are to develop methods to deflect the bodies from impacting Earth and also harvest the maximum amount of scientific data and natural resources that exist within them.

In order to explore these bodies however, the ability to safely and accurately navigate the spacecraft must be established. Navigation serves as the foundation for spacecraft operations, since without an accurate understanding of the state of the spacecraft very few other aspects of the mission can be accomplished. The primary challenge for navigating around these bodies results from their typically significant distance from the Earth within the solar system, so communication with the Earth contains significant time delays. Due to this delay and the time-sensitive nature of some spacecraft operations, autonomous navigation will likely be needed. The need for

effective autonomous navigation for a spacecraft landing on an asteroid or comet will be tested in this study.

To eliminate the time delay problem imposed by Earth-based tracking observations, autonomous navigation requires a different source of measurements. One possibility is optical measurements of the small-body, as used in this study. Using images taken by a near-by spacecraft of a small-body such as an asteroid or comet, along with knowledge of the landmarks that appear in the images, Autonomous Optical Navigation (AON) can be attempted for spacecraft missions operating near small-bodies.

Throughout this study, the terms "small-body" and "asteroid" will be used interchangeably. The focus of this study is landing on an asteroid, so the term "asteroid" may be used in many cases where the same information will likely apply to other small-bodies as well, such as comets, small moons, and any other small solar system bodies with distinct surface landmarks and no atmosphere.

1.1 HISTORICAL CONTRIBUTIONS

To date, the use of AON has been mostly restricted to small-body fly-bys and impacts. The first utilization of AON onboard a spacecraft was the technology demonstration mission Deep Space 1 as it flew by the asteroid Braille in July, 1999 and the comet Borrelly in September, 2001 [2]. Other spacecraft which have successfully used this Fly-by AON technology include STARDUST as it flew by the comet Wild 2 in January, 2004 [7], and the Deep Impact fly-by vehicle as it monitored the Impactor colliding with the comet 9P/Tempel in July, 2005 [15]. The Deep Impact Impactor was the first and only spacecraft to use AON to impact a small-body.

The Japanese Aerospace Exploration Agency (JAXA) created and operated a spacecraft called Hayabusa which was intended to land on a near-Earth asteroid named 25143 Itokawa using AON and a Light Detection and Ranging (LIDAR) altimeter [13].

The spacecraft experienced a number of problems while attempting to land, though it is believed to have landed and collected some of the swirling dust in its sample container. The spacecraft is due to return to Earth in 2010.

Even though the Hayabusa mission was designed to land on the asteroid using AON, and there are software tools to analyze landings on small bodies, there is not a tool available to the aerospace industry which can be quickly modified for different scenarios, and which has fast Monte Carlo capabilities. These abilities allow the user to quickly perform simulations which can characterize the accuracy of potential small-body landings using only AON with landmarks on the surface. This tool will provide a better understanding of the impact on landing accuracy of various potential error sources and invaluable information for mission planners of future small-body landing missions.

There have been other studies involving AON, as is seen in [1] and [3] for example, though none for landing on a small-body using landmarks as is done in this study. The study in reference [4] discusses the tracking algorithms and flight results of the Autonomous Navigation (autonav) system for the three times it had been used successfully (up to 2004): the Deep Space 1 flyby of comet Borrelly, and the STARDUST flybys of asteroid Annefrank and comet Wild 2.

1.2 FOCUS AND OBJECTIVES OF THESIS

With the success of the flyby AON systems on recent spacecraft, the next step is to attempt the use of this technology for orbiting and landing on small bodies such as asteroids. The current focus of this work is conducting simulations to study the accuracy with which a spacecraft can land on an asteroid using AON. Thus the Small-Body Autonomous Navigation Toolset (SANT) was developed to perform Monte Carlo analysis of spacecraft landing error on a small asteroid, given a set of spacecraft and asteroid properties, and is described in this document. SANT will likely be needed for

future potential small-body missions as mission designers analyze possibilities for landing on an asteroid or other small bodies such as comets and small moons. The primary objectives are the scatter of the landing footprint via Monte Carlo analysis for a variety of mission profiles, information about the relationship between the shape and size of the landing footprint and navigation error sources, and any appropriate statistics.

1.3 OUTLINE OF THESIS

In the following sections a brief summary of the theory necessary for this study is described, including considerations of dynamics, targeting, and reference frames. With this background, SANT and all of its major components are presented. Chapter 4 explains how SANT was used to simulate the various mission scenarios and generate the results using different simulation characteristics. The simulation results are summarized in chapter 5 along with some basic analysis. The final chapter summarizes the conclusions drawn from the testing and provides recommendations for future endeavors with this area of research.

CHAPTER 2

Background Theory

2.1 ORBITAL FORCES AND DYNAMICS

2.1.1 General Motion

The two-body general motion of a spacecraft in orbit around a small-body can be described by the following second order differential equation:

$$\ddot{\bar{r}} = -\frac{G(m_{sc} + m_{ast})\bar{r}}{|\bar{r}|^3} \tag{2.1}$$

where $\bar{r}$ is the body to spacecraft position vector, G is the universal gravitational constant, and m_{sc} and m_{ast} are the masses of the spacecraft vehicle and the small-body, respectively. This expression can also be written as an array of first order differential equations:

$$\begin{bmatrix} \dot{\bar{r}} \\ \dot{\bar{v}} \end{bmatrix} = \begin{bmatrix} \bar{v} \\ -\frac{G(m_{sc} + m_{ast})\bar{r}}{|\bar{r}|^3} \end{bmatrix} \tag{2.2}$$

where $\bar{v}$ is the velocity vector of the spacecraft. SANT uses this array of first order differential equations format. However, because it is reasonable to assume that the mass of the asteroid far exceeds the mass of the spacecraft, the spacecraft mass can be treated as negligible and thus the equations can now be written as:

$$\begin{bmatrix} \dot{\bar{r}} \\ \dot{\bar{v}} \end{bmatrix} = \begin{bmatrix} \bar{v} \\ -\frac{\mu\bar{r}}{|\bar{r}|^3} \end{bmatrix} \tag{2.3}$$

where $\mu = G\,m_{ast}$.

2.1.2 Perturbed Motion

In addition to the standard two-body gravitational motion described above, a number of perturbations can affect the motion of the spacecraft. The perturbations are forces that are accounted for in the spacecraft model as accelerations added to the standard two-body gravitational force, as described by the equation:

$$\bar{a}_{total} = \bar{a}_{2B} + \bar{a}_{TBP} + \bar{a}_{SRP} + \bar{a}_{grav} \tag{2.4}$$

where $\bar{a}_{total}$ is the total imparted force per unit mass on the spacecraft, $\bar{a}_{2B}$ is the general two-body gravitational force, and $\bar{a}_{TBP}$ is the sum of the third body gravitational forces of other celestial bodies (referred to as third bodies), $\bar{a}_{SRP}$ is the Solar Radiation Pressure (SRP) force, and $\bar{a}_{grav}$ is the force on the spacecraft produced by the non-spherical character. It is assumed that there are no other significant perturbations on the spacecraft besides those listed above.

2.1.2.1 Third Body Perturbations

Beyond the two-body system of the spacecraft and the small-body serving as the central body, other large masses in the solar system and medium-to-small masses in the immediate vicinity of the spacecraft can introduce significant gravitational perturbations on the spacecraft trajectory. These forces are referred to as "third-body" perturbations even though there may be multiple bodies that contribute. The force due to third body perturbations is described by the equation [26]:

$$a_{TBP} = \sum_{TB} GM_{TB}\left(\frac{\bar{r}_{sc-TB}}{\left|\bar{r}_{sc-TB}\right|^3} - \frac{\bar{r}_{SB-TB}}{\left|\bar{r}_{SB-TB}\right|^3}\right) \tag{2.5}$$

where a_{TPB} is the total third-body acceleration on the spacecraft, G is the universal gravitational constant, M_{TB} is the mass of the particular third-body, $\bar{r}_{sc-TB}$ is the spacecraft to third-body position vector, and $\bar{r}_{SB-TB}$ is the small-body (which the spacecraft is

orbiting) to third-body position vector. The right hand side of the equation is summed over all "third bodies" being considered.

2.1.2.2 Solar Radiation Pressure

The solar radiation emanating from the Sun can introduce a significant perturbation to the spacecraft motion. A flat-plate model of the spacecraft is used to estimate the SRP on the spacecraft. The force due to the SRP on the spacecraft is described by the equation [26]:

$$\bar{a}_{SRP} = -\frac{p_{SR} c_R A_{proj}}{M_{sc}} \frac{\bar{r}_{sc-sun}}{|\bar{r}_{sc-sun}|} \tag{2.6}$$

where p_{SR} is the solar-radiation pressure, c_R is the coefficient of reflectivity of the spacecraft, A_{proj} is the exposed spacecraft area to the Sun, or projected area, M_{sc} is the spacecraft mass, and $\bar{r}_{sc-sun}$ is the spacecraft-to-Sun vector. The coefficient of reflectivity is zero if the spacecraft is perfectly transparent, one if it is a pure black body where all energy is absorbed and all force is transmitted, and two if it is a perfect mirror. The SRP is

$$p_{SR} = \frac{SF_{sc}}{c} \tag{2.7}$$

where SF_{sc} is the solar flux at the spacecraft location in the solar system and c is the speed of light in a vacuum. The solar flux at the spacecraft position is

$$SF_{sc} = SF_{earth} \left(\frac{|\bar{r}_{earth-sun}|}{|\bar{r}_{sc-sun}|} \right)^2 \tag{2.8}$$

where SF_{earth} is the solar flux at the Earth, and $\bar{r}_{earth-sun}$ is the Earth-to-Sun vector. The spacecraft projected area and mass can vary with different spacecraft orientations and temporal variation resulting from final fuel expenditure.

2.1.2.3 Gravitational Harmonics

A non-uniform gravity field, which is a result of non-uniform mass distribution within the small-body and non-spherical shape of the small-body, can produce perturbations on the spacecraft two-body motion. Using zonal, tesseral, and sectorial harmonic gravitational coefficients to define a gravitational field for the small-body, the perturbation on the spacecraft at a given position with respect to the body-fixed frame of the asteroid can be calculated using the non-singular method developed by Pines [20].

2.2 TARGETING

In order to land at a particular location on the asteroid, the spacecraft must be able to accurately target this position and correctly implement a maneuver that will bring the spacecraft to the target. Minimum energy Lambert targeting [26], specified initial and final time Lambert Targeting [26], K-matrix targeting, and many other targeting schemes are all potential methods for this targeting requirement. Because of asteroid rotation, a fixed final time targeting algorithm was needed. Of the fixed final time methods available, K-matrix targeting proved to be more accurate than Lambert Targeting, and thus this method was utilized.

Originally developed at NASA JPL, the K-matrix targeting method is a linear targeting strategy previously employed for interplanetary trajectories. Starting with the current inertial spacecraft state, final inertial target position, current time, and time for the spacecraft to reach the target, the K-matrix targeting algorithm calculates the delta-v required at the current time to reach the target position at the specified time. The iterative targeting algorithm is described in the following discussion.

The targeting delta-v that must be applied to the spacecraft at the current time to reach the target position at the target time is calculated using the equation

$$\Delta\bar{v} = \left(K_{part}\right)^{-1}\left(\bar{r}_{t\arg}(t_{t\arg}) - \bar{r}_{sc}(t_{t\arg})\right) \tag{2.9}$$

where t_{targ} is the target time, or time at which it is desired for the spacecraft to reach the target position, $\bar{r}_{targ}$ is the desired position of the spacecraft at the target time, and $\bar{r}_{sc}$ is the actual spacecraft state at the target time projected from the current best known state of the spacecraft. The K-matrix contains numerical partial derivatives of the position components at the target time, i.e. targeting parameters, with respect to the initial velocity components is calculated using finite differencing, and has the following form:

$$K_{part} = \begin{bmatrix} \frac{\partial r_x(t_{targ})}{\partial v_x} & \frac{\partial r_x(t_{targ})}{\partial v_y} & \frac{\partial r_x(t_{targ})}{\partial v_z} \\ \frac{\partial r_y(t_{targ})}{\partial v_x} & \frac{\partial r_y(t_{targ})}{\partial v_y} & \frac{\partial r_y(t_{targ})}{\partial v_z} \\ \frac{\partial r_z(t_{targ})}{\partial v_x} & \frac{\partial r_z(t_{targ})}{\partial v_y} & \frac{\partial r_z(t_{targ})}{\partial v_z} \end{bmatrix} \tag{2.10}$$

This delta-v is added to initial state and delta-v's from previous iterations, and the targeting algorithm will iterate until convergence (the miss distance is less than a user defined tolerance). This is described by the equation

$$\left| \bar{r}_{targ}(t_{targ}) - \bar{r}_{sc}(t_{targ}) \right| \leq tolerance \tag{2.11}$$

where the *tolerance* value is set by the user.

2.3 REFERENCE FRAMES

The state of a spacecraft can be described in a number of different reference frames, each useful for different reasons. The below Sections 2.3.1 through 2.3.6 describe the reference frames used in this particular study.

2.3.1 Inertial body-centered reference frame

The origin is located at the center of mass of the small-body. The frame does not rotate with the body but instead the orientation is fixed with respect to inertial space. The frame is not truly inertial, but this is accounted for in the differential equations of the spacecraft motion.

2.3.2 Body-fixed reference frame

The origin is located at the center of mass of the small-body. The frame is fixed in the body and rotates with it at the same rotational rate. The transformation matrix from the inertial frame to the body-fixed frame is constructed according to the following steps. Because the inertial and body-fixed reference frames share the same origin, the transformation is a rotation matrix.

The asteroid body-fixed pole's right ascension α, declination δ, and the asteroid prime meridian ω at the current time are calculated by the following equations:

$$\begin{aligned} \alpha &= \dot{\alpha}\Delta t + \alpha_0 \\ \delta &= \dot{\delta}\Delta t + \delta_0 \\ \omega &= \dot{\omega}\Delta t + \omega_0 \end{aligned} \tag{2.12}$$

where $\dot{\alpha}$, $\dot{\delta}$, and $\dot{\omega}$ are the given associated rates for these values, Δt is the difference in time between the current time and epoch time, and α_0, δ_0, and ω_0, are the initial right ascension, declination, and prime meridian location at the epoch time.

The transformation matrix is calculated according to the equations

$$R_3(\omega) = \begin{bmatrix} \cos(\omega) & \sin(\omega) & 0 \\ -\sin(\omega) & \cos(\omega) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.13}$$

which is the pure rotation matrix about the third axis for the prime meridian angle ω,

$$R_1(\tfrac{\pi}{2} - \delta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\tfrac{\pi}{2} - \delta) & \sin(\tfrac{\pi}{2} - \delta) \\ 0 & -\sin(\tfrac{\pi}{2} - \delta) & \cos(\tfrac{\pi}{2} - \delta) \end{bmatrix} \tag{2.14}$$

which is the pure rotation matrix about the first axis for the declination angle δ subtracted from $\frac{\pi}{2}$, and

$$R_3(\alpha + \tfrac{\pi}{2}) = \begin{bmatrix} \cos(\alpha + \tfrac{\pi}{2}) & \sin(\alpha + \tfrac{\pi}{2}) & 0 \\ -\sin(\alpha + \tfrac{\pi}{2}) & \cos(\alpha + \tfrac{\pi}{2}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.15}$$

which is the pure rotation matrix about the third axis for the right ascension angle α plus $\frac{\pi}{2}$.

The total rotation matrix $R_{i\to bf}$ is calculated by combining the rotation matrices:

$$R_{i\to bf} = R_3(\omega)R_1(\tfrac{\pi}{2}-\delta)R_3(\alpha+\tfrac{\pi}{2}) \tag{2.16}$$

These equations are formatted according to the International Astronomical Union's (IAU) convention [9].

2.3.3 Spherical reference frame

Spherical coordinates can also be used to represent the spacecraft state. These coordinates include the radius, right ascension, declination of the spacecraft with respect to the small-body. Converting a given spacecraft radius, right ascension, and declination to a Cartesian position vector is defined by the equation

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r_{sc}\cos(\delta_{sc})\cos(\alpha_{sc}) \\ r_{sc}\cos(\delta_{sc})\sin(\alpha_{sc}) \\ r_{sc}\sin(\delta_{sc}) \end{bmatrix} \tag{2.17}$$

where r_{sc} is the radius magnitude, δ_{sc} is the declination, and α_{sc} is the right ascension.

2.3.4 View2 reference frame

The origin is located at the center of mass of the spacecraft. The coordinate system is a radial-transverse-normal (RTN) type frame where the X axis is along the velocity vector, the Y axis is along the spacecraft to central-body center of mass position vector, and the Z axis is normal to both the X and Y axes and thus parallel to the satellite orbital angular momentum vector.

The transformation matrix of the inertial frame to the View2 frame has three rows, with the first row aligned with the velocity direction, the second row aligned perpendicular to the orbit plane, and the third row is in-plane and perpendicular to the velocity. These rows are calculated using the equation

$$R_{i\to v2} = \begin{bmatrix} \bar{v}/|\bar{v}| \\ \bar{r}\times\bar{v} \\ \left(\bar{v}/|\bar{v}|\right)\times\left(\bar{r}\times\bar{v}\right) \end{bmatrix} \tag{2.17}$$

where $R_{i\to v2}$ is the Inertial-to-View2 transformation matrix, $\bar{r}$ is the position vector, and $\bar{v}$ is the velocity vector.

2.3.5 Topocentric reference frame

The origin is located at the target position on the small-body surface. The X axis is positive East (as dictated by the rotation of the asteroid), the Y axis is positive South, and the Z axis is normal to both the X and Y axes and thus positive down into the small-body. The transformation matrix from the inertial frame to the Topocentric reference frame is built according to the equations

$$\bar{z} = \frac{\bar{r}}{|\bar{r}|} \tag{2.18}$$

$$\bar{x} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}\times\bar{z} \tag{2.19}$$

$$\bar{y} = \bar{z}\times\bar{x} \tag{2.20}$$

$$M = \begin{bmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \end{bmatrix} \tag{2.21}$$

where $\bar{r}$ is the Cartesian position in the inertial frame and M is the Cartesian-to-Topocentric transformation matrix.

2.3.6 Geodetic reference frame

The origin is located at the center of mass of the small-body. The frame is fixed in the body and rotates with it at the same rotational rate. The coordinates consist of geodetic latitude, longitude and altitude above the small-body surface.

The process to convert a body-fixed Cartesian position vector to a geodetic latitude, longitude, and altitude, given the triaxial ellipsoid minimum, middle, and

maximum radius values [18], is shown below. The geodetic longitude is the same as the spherical longitude, and is calculated by the equation

$$\lambda_{geo} = \tan^{-1}\left(\frac{y}{x}\right) \tag{2.22}$$

where x and y are part of the Cartesian position vector (x, y, z). The geodetic latitude is calculated using Equations (2.23), (2.24), (2.25), and (2.26) below. First a couple of intermediate values are calculated according to the equations

$$\begin{aligned} a &= 1 \Big/ \sqrt{\left(\frac{\cos(\lambda_{geo})}{A}\right)^2 + \left(\frac{\sin(\lambda_{geo})}{B}\right)^2} \\ b &= C \end{aligned} \tag{2.23}$$

where A, B, and C are the given maximum, middle, and minimum radius values of the triaxial ellipsoid. These a and b values are then used in the equations

$$\begin{aligned} f &= b - a \\ e &= \sqrt{1 - (1 - f)^2} \\ N &= a \\ error &= dz = e^2 z \end{aligned} \tag{2.24}$$

where f is known as "flattening". If the error value *error* is equal to 0, it is set equal to 1. Then the following equations are iterated in the same order until the *error* value is less than a specified tolerance value, e.g. 1×10^{-6}:

$$\begin{aligned} \sin(\varphi_{geo}) &= \frac{z + dz}{\sqrt{x^2 + y^2 + (z + dz)^2}} \\ N &= a \Big/ \sqrt{1 - \left(e \sin(\varphi_{geo})\right)^2} \\ dz_{new} &= Ne^2 \sin(\varphi_{geo}) \\ error &= dz_{new} - dz \\ dz &= dz_{new} \end{aligned} \tag{2.25}$$

Once the *error* value is less than the specified tolerance, the geodetic latitude and altitude are calculated using the equations

$$\varphi_{geo} = \tan^{-1}\left(\frac{z + dz}{\sqrt{x^2 + y^2}}\right)$$
$$h_{geo} = \sqrt{x^2 + y^2 + (z + dz)^2} - N \tag{2.26}$$

where h_{geo} is the height above the triaxial ellipsoid surface.

2.3.6 Camera-centered-Camera-fixed reference frame

The origin is located at the geometric center of camera. The frame is fixed in the camera body and rotates with it at the same rotational rate. The third axis is defined by the vector going directly from the origin through the center of the camera lens, perpendicular to the lens. The other two axes are normal to the third axis and to each other.

CHAPTER 3

Small-body Autonomous Navigation Toolset

3.1 TOOL REQUIREMENTS AND GENERAL ASSUMPTIONS

The Small-body Autonomous Navigation Toolset (SANT) was developed for future orbiting, hovering, and landing small-body missions. It was designed to meet the following requirements:

- Assess navigational and landing accuracy achievable for small-body landings (given specific spacecraft and asteroid parameters) using landmark-tracking AON
- Implement necessary maneuvers designed for a soft touchdown
- Measure landing errors and knowledge errors

Some general assumptions that SANT makes include the following:

- The state of the spacecraft includes the position and velocity of the spacecraft.
- The camera faces toward the asteroid all the way to landing, as the engines are canted out and thus allow the camera to be located in the middle of the spacecraft side that is facing the asteroid.

3.2 MODELS

There are two models used in SANT. The first is called the nominal model, which represents what is known to the simulated spacecraft. The nominal state is the onboard spacecraft estimate of the spacecraft state. The second model is the truth model, which describes the true trajectory of the simulated spacecraft and is used to generate realistic observables. The truth model is sampled from error statistics on various

parameters, such as the initial state, execution errors on maneuvers, and other navigation error sources.

3.3 TOOL OPERATIONS

The following subsections describe the basic operations of SANT.

3.3.1 Executive Script

The executive script is responsible for executing all of the calculations needed for a single run of the simulation, which includes everything from the spacecraft starting at the initial state to the spacecraft landing on the asteroid or until the simulation end time is reached. The executive script first launches the initialization script, as described in Chapter 4, which initializes most of the simulation variables. Next the Picture Sequence Generation Function generates the array of times that pictures will be taken by the spacecraft of the asteroid. The Attitude Error Function, as described in Section 3.9.5, then produces the attitude error for every time in the picture sequence. Initial position and velocity standard deviation values, i.e. sigma's, are defined in an RTN frame and rotated into a Cartesian frame to define the a priori state covariance matrix.

At each time step, simulated images are taken and an orbit fit computed until the simulation has determined spacecraft nominal state has arrived at the asteroid surface or the specified simulation end time is reached. The nominal state is propagated from the epoch time, or time of the first observation, to the current picture time. This is achieved with the Propagation and Event Detection (rk45driver) Function, as described in Section 3.3.2, which generates the time and nominal state of the spacecraft at the current picture time or when the spacecraft has landed, provided rk45driver determines that the spacecraft has reached the asteroid surface before the current picture time.

If rk45driver determines that the nominal state has reached the surface, the truth state is propagated from the epoch time to the pre-set simulation final time using rk45driver to determine if the truth state also lands on the surface. If the truth trajectory does reach the surface, rk45driver outputs the truth state and time at the landing and the simulation is ended. If the truth trajectory does not reach the surface, a message to the user is generated that states the spacecraft did not land and the simulation is ended.

If rk45driver determines that the nominal state has not reached the surface, the truth state is propagated from the epoch time to the current picture time. Then the initial covariance matrix for the Kinematic Filter, as described in Section 3.5, is constructed by placing the variance values defined in the initialization file along the diagonal of the initial covariance matrix. There are six of these variance values, one for each Cartesian direction and the three attitude values right ascension, declination, and twist.

For each simulated image, landmark positions are gathered from the Landmarks Function, as described in Section 3.4. If filtering is active, the Processing Pictures of Landmarks Function, as described in Section 3.4.3, is used to generate the following:

- a position fix for the current picture time
- a use value which dictates if the position fix should be used
- a weight matrix which allows the simulation to place more or less trust in the accuracy of each component of the position fix
- the residuals of the three position components

At this point in the executive file, the Dynamical Filter segment will be entered if all of the following conditions are met:

- the current picture time is greater than or equal to the specified first allowable filter time
- the current picture time is less than or equal to the last allowable filter time

- filtering is active
- the number of position fixes is greater than the minimum allowable for the Dynamical Filter

In the following step, the Sliding Window Function, which is further described in Section 3.3.3, adjusts the observables available to the Dynamical Filter to the latest specified amount, assuming the sliding window feature is on. The function generates the latest specified number of position fixes; the use, time, and weight matrix values associated with those positions fixes; the new epoch time, epoch nominal and truth states; and the new inertial-to-RTN coordinate transformation matrix and initial covariance for the Dynamical Filter.

The Dynamical Filter, as described in Section 3.6, is executed next. Using the position fixes as observables, along with the use vector, time vector, epoch time, epoch nominal state, weight matrix, and initial Dynamical Filter covariance, the Dynamical Filter outputs an updated best estimate of the nominal state at the epoch time along with the covariance at the epoch time.

The new nominal epoch state obtained from Dynamical Filter is then propagated forward in time using rk45driver to the current picture time. If the Dynamical Filter is not executed (e.g. if the minimum number of observations has not been met), then the original nominal epoch state is propagated to current picture time. This nominal state at the current picture time is then used in the maneuver segment, as described in Section 3.7.

After exiting the simulation due to one of the conditions described above, the executive script assembles some statistics and generates plots of the spacecraft trajectory in the body-fixed frame and inertial frame, if specified in the initialization file. Examples of these plots can be seen in Chapter 5.

3.3.2 Propagation and Event Detection Function

The Propagation and Event Detection Function (rk45driver) serves to propagate an input initial state to a given final time or to the time when an event is triggered. In SANT, this event is the landing of the spacecraft on the asteroid surface, as described below. The rk45driver Function outputs the final state and time of the propagation, as well as a flag indicating if an event was triggered. The function is given the model that it should propagate, i.e. the nominal or truth model, the start and end time for the integration, and the state at the start time.

The rk45driver Function utilizes The MathWorks MATLAB propagator ode45, which is based on an explicit Runge-Kutta Dormand-Prince pair (4,5) formula [17]. It is a single step integrator: in computing a solution for a particular time y(t), it needs only the solution at the immediately preceding time point y(t-1). The rk45driver Function provides to ode45 the appropriate model, beginning and end integration times, the initial state, and the integration options as set by the user in the initialization file. These options include the relative and absolute tolerances, as well as the events file that ode45 should use for event detection.

The events file used within ode45 utilizes the algorithm in Section 2.3.6 to convert from Cartesian to Geodetic Coordinates. An event is registered when the Geodetic altitude is less than the target altitude, indicating that the spacecraft has reached the surface.

3.3.3 Sliding Window Function

The Sliding Window Function serves to limit the number of observations that the Dynamical Filter must process in order to prevent unnecessarily long computation cycles. If the number of observations within the executive script is greater than the preset

maximum amount of observations, which for example could be 16, the Sliding Window Function is called in order to set the latest 16 observations as those the filter will utilize.

3.4 LANDMARKS

Landmarks as defined for SANT are unique features on the asteroid surface which are uniquely distinguishable from all other points on that surface. The landmarks in the spacecraft camera field of view are considered optical measurements and are utilized to perform autonomous navigation.

Another variety of optical measurement is determination of the center of brightness of the asteroid in the camera field of view. This method was utilized extensively when first developing SANT. For simulations of spacecraft that are far from the small-body where it would be difficult for the spacecraft to detect and distinguish between landmarks, center of brightness measurements would be more effective. However, when the landmark measurement capability became available, they became the exclusive source of optical measurements. This change was due to the greater navigational accuracy achievable with landmarks over center-of-brightness and in view of the fact that the spacecraft distance from the small-body in this simulation is never large enough to require center of brightness measurements. It is also necessary to use landmarks over center-of-brightness measurements when the spacecraft is close enough to the small-body that the camera cannot image the entire body, which is certainly the case when attempting to land on the small-body.

For this simulation-based study, the landmarks are uniformly distributed across the surface of the triaxial ellipsoid representing the small body, and there are also a denser set of uniformly spaced landmarks around the target position on the surface. In order to generate these landmarks, the Landmark Creation Function is utilized before any simulations are run, as described in the next section.

3.4.1 Landmark Creation Function

Landmarks must first be generated at known points on the asteroid surface before any optical navigation can be accomplished in the simulation. Landmarks are produced once using the Landmark Creation Function and then are loaded in the initialization file each time the simulation is run.

The Landmark Creation Function utilizes the small-body triaxial ellipsoid shape model, as well as the number of angular degrees separating each landmark, to generate a uniform spread of landmarks between the user-specified minimum and maximum latitude and longitude values. This can be done for the entire surface of the asteroid, or around only a particular position on the surface such as the target position.

For each unique latitude and longitude combination, a landmark body-fixed Cartesian position is calculated. In order to generate this position, the radius of the defined triaxial ellipsoid at the given latitude and longitude must be calculated, as described by

$$r = \sqrt{\frac{1}{\left(\cos(\lambda)\cos(\varphi)/A\right)^2 + \left(\sin(\lambda)\cos(\varphi)/B\right)^2 + \left(\sin(\varphi)/C\right)^2}} \tag{3.1}$$

where r is the radius, λ is the specified longitude, φ is the specified latitude, and A, B, and C are the small-body maximum, middle, and minimum radius values. The radius value from Equation (3.1), along with the latitude and longitude values, are converted from spherical coordinates to Cartesian coordinates to generate the landmark body-fixed Cartesian position, according to the equations

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r\cos(\lambda)\cos(\varphi) \\ r\sin(\lambda)\cos(\varphi) \\ r\sin(\varphi) \end{bmatrix} \tag{3.2}$$

Once generated, the body-fixed reference frame positions of the landmarks are saved in a file and simply loaded every time the simulation is run, as opposed to

generating the landmarks for every simulation run, which would be an inefficient use of time and computer resources.

3.4.2 Landmarks Selection Function

The Landmarks Selection Function determines which set of landmarks the simulation needs at a particular time in the simulation. The function calculates the distance of the nominal spacecraft state from the target position, and if greater than a user-specified amount, the function will return an array of landmarks uniformly distributed over the entire surface of the asteroid. If the nominal state is closer to the target than a specified tolerance, the function will output a denser landmarks array that is centered at the target location.

3.4.3 Processing Pictures of Landmarks Function

The Processing Pictures of Landmarks Function computes a Cartesian position fix for the spacecraft at the current picture time using landmark information. The function encompasses two important sub-functions: the Generate Landmark Observations Function, as described in Section 3.4.4, and the Kinematic Filter which is explained in Section 3.5.

As a first step, the true and nominal spacecraft states are rotated from the inertial frame to the body-fixed frame because the landmark locations are in the body fixed frame. The transformation matrix is calculated using the algorithm from Section 2.3.2.

In the second step, the Generate Landmark Observations Function, as described in Section 3.4.4, is utilized to produce the following data for each visible landmark:

- the observed pixel and line values in the camera field of view
- the body-fixed coordinates
- the use value (i.e. a flag indicating whether or not a landmark should be used)

- the weight values for the pixel and line values

If there are more than the user-specified minimum number of useable landmarks, the Kinematic Filter, as described in Section 3.5, is then utilized. The pixel and line values for the visible landmarks serve as the observables for the Kinematic Filter. Using the pixel and line observables, the weights for each of those pixel and line values, the Kinematic apriori covariance matrix, the body-fixed Cartesian positions of the visible landmarks, the nominal pointing direction of the camera, and the current nominal position of the spacecraft, the Kinematic Filter outputs a position fix estimate for the spacecraft at the current time. The Kinematic Filter also outputs for the current time a new estimate of the camera pointing, the post-fit covariance, the pre-fit residuals, and the post-fit residuals.

After the Kinematic Filter is called, the post-fit position fix estimate from the filter is rotated back into an inertial frame in order to calculate the residuals for each position component. If the magnitude of the pre-fit residuals is less than the magnitude of the post-fit residuals, a warning message is displayed informing the user that the Kinematic Filter has likely diverged, and the position fix is not used. The output weight matrix for each position fix is constructed from the square root of the diagonal values of the covariance matrix output by the Kinematic Filter.

3.4.4 Generate Landmark Observations Function

The landmark positions within the camera field of view are represented as pixel and line values. The Generate Landmark Observations Function (make_lndmk_obs) is responsible for producing the following:

- a vector of these pixel and line values in the camera field of view for all visible landmarks
- the associated weights for those pixel and line values

- the nominal camera pointing vector
- the body-fixed Cartesian components of all visible landmarks

These values are calculated for a given time, nominal and truth state, and also require the body-fixed positions of all the landmarks loaded by the Landmarks Function in Section 3.4.2.

A solar system ephemeris, as described in Chapter 4, is referenced to calculate the spacecraft-to-Sun position vector in the small-body centered inertial reference frame. The unit vector of the spacecraft-to-Sun vector along with the truth and nominal spacecraft to small-body position vectors are used along with the current time to execute the Pointing Calculation Function as described in Section 3.8. This function generates the nominal and truth pointing vectors from the spacecraft to the small-body, which consist of the right ascension, declination and twist of the camera for the nominal and truth states. This step assumes that the spacecraft camera has been commanded to point at the origin of the small-body centered coordinate system of the onboard nominal state at every picture time. This process is not directly modeled in the simulation.

In the next step, the landmarks on the far side of the asteroid from the spacecraft are removed from the list of available landmarks. Then the Pixel/Line Calculator, as described in Section 3.4.5, is used to determine the pixel and line values for the remaining landmarks, and only those landmarks in the camera field of view are retained.

If observation errors are utilized, they are added to the pixel and line values for each visible landmark, as more fully described in the Observation Error Section 3.9.2. The weight values for the pixel and line values are user-defined sigma values.

3.4.5 Pixel/Line Calculator Function

The Pixel/Line Calculator Function (intopl) converts the inertial spacecraft-to-landmark position vector, along with the true or nominal pointing of the spacecraft

camera, to pixel and line values in the camera field of view. Equations in this section originate from reference [19].

In the first step, the rotation matrix from the inertial frame to the camera frame is calculated using the right ascension, declination and twist of the camera pointing. The transformation matrix is calculated using the equations below.

$$R_3(\phi) = \begin{bmatrix} \cos(\phi) & \sin(\phi) & 0 \\ -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

which is the pure rotation matrix about the third axis of a camera-centered-camera-fixed reference frame for the camera twist angle ϕ.

$$R_2(\tfrac{\pi}{2} - \delta) = \begin{bmatrix} \cos(\tfrac{\pi}{2} - \delta) & 0 & -\sin(\tfrac{\pi}{2} - \delta) \\ 0 & 1 & 0 \\ \sin(\tfrac{\pi}{2} - \delta) & 0 & \cos(\tfrac{\pi}{2} - \delta) \end{bmatrix} \tag{3.4}$$

which is the pure rotation matrix about the second axis for the declination angle δ subtracted from $\frac{\pi}{2}$.

$$R_3(\alpha) = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.5}$$

which is the pure rotation matrix about the third axis for the camera right ascension angle α.

The total inertial-to-camera transformation matrix is calculated by multiplying the above three rotation matrices together, as shown in the equation

$$R_{i \to cam} = R_3(\phi) R_2(\tfrac{\pi}{2} - \delta) R_3(\alpha) \tag{3.6}$$

In the second step, this transformation matrix is applied to the inertial spacecraft-to-landmark position unit vector, and this new vector is unitized to ensure that it is also a unit vector, as shown in the equations

$$\hat{r}_{sc-LM} = \frac{\bar{r}_{sc-LM}}{|\bar{r}_{sc-LM}|} \tag{3.7}$$

where $\bar{r}_{sc-LM}$ is the inertial spacecraft-to-landmark vector.

$$\bar{p} = R_{i\to cam}\ \hat{r}_{sc-LM} \tag{3.8}$$

where $\bar{p}$ is spacecraft-to-landmark vector in the camera frame.

$$\hat{p} = \frac{\bar{p}}{|\bar{p}|} \tag{3.9}$$

which is the unit vector of the spacecraft-to-landmark vector in the camera frame.

In the third step, the geometric transformation from a sphere to a plane, or gnomonic projection, is applied to the $\hat{p}$ vector using the camera focal length as defined in the initialization script. The rectangular x and y coordinates are calculated using the equations

$$x = f\frac{\hat{p}(1)}{\hat{p}(3)} \tag{3.10}$$

$$y = f\frac{\hat{p}(2)}{\hat{p}(3)} \tag{3.12}$$

where f is the focal length of the camera, and $\hat{p}(1)$, $\hat{p}(2)$, and $\hat{p}(3)$ are the first, second, and third components of the $\hat{p}$ vector.

In the fourth step, these rectangular coordinates are converted to pixel and line values using the camera K-matrix and the central pixel and line values for the camera, both defined in the initialization file. The camera K-matrix is not to be confused with the K-matrix described in Section 2.2 that is used for K-matrix targeting.

$$\begin{bmatrix} pixel \\ line \end{bmatrix} = \begin{bmatrix} K_x & K_{xy} \\ K_{yx} & K_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} pixel_{central} \\ line_{central} \end{bmatrix} \tag{3.13}$$

Pixel values increase to the right in the camera field of view, and line values increase downward in the camera field of view.

3.5 KINEMATIC FILTER

The Kinematic Filter utilizes the batch estimation algorithm, i.e. “batch filter”, from reference [25], as described in Appendix A, to obtain a position fix estimate for the

spacecraft at a particular time. This section will discuss the unique aspects of the Kinematic Filter, but for the exact operations of the batch filter see Appendix A. The Kinematic Filter requires at least one visible landmark to operate.

The Kinematic Filter requires the following:

- The pixel and line values of the visible landmarks in the camera field of view, which serve as the observables for the Kinematic Filter
- The weights for each of the pixel and line values, which are manually set by the user
- The Kinematic Filter apriori, or initial, covariance matrix. This covariance matrix is constructed in the Executive Function from values defined in the initialization file.
- The body-fixed Cartesian positions of the visible landmarks
- The nominal pointing direction of the camera
- The current nominal position of the spacecraft

The Kinematic Filter generates the following:

- The position fix estimate for the spacecraft at a particular time
- A new estimate of the camera pointing
- The post-fit covariance
- The pre-fit residuals
- The post-fit residuals

To utilize the batch filter described in Appendix A, the partial derivatives of the pixel and line values, i.e. the observables for the Kinematic Filter, with respect to the

spacecraft-to-landmark vector and with respect to the nominal pointing attitude of the camera are needed. These partial derivatives are calculated using the Kinematic Position Partials Function, as described in Section 3.5.1, and the Kinematic Pointing Partials Function, in Section 3.5.2, respectively.

A visual way of understanding the Kinematic Filter is as a collection of unit vectors from the spacecraft to each landmark. With the current estimate of the spacecraft position and attitude, along with the location of each landmark in the camera field of view, unit vectors are calculated to each landmark. With at least two unit vectors originating from the same point (the spacecraft), the position of the spacecraft should correspond with the location where the unit vectors intersect. However, due to many error sources, the unit vectors are unlikely to exactly meet at one point in space. With more than two unit vectors (i.e. more than two landmarks observed), the calculation becomes over-determined, and just as with Global Positioning System (GPS) position determination, a Kinematic Estimation Filter is used to converge on a best estimate of the spacecraft position. To find this best estimate of the spacecraft position, the batch filter adjusts the nominal spacecraft position and attitude until the intersection of all the unit vectors is as tight as the tolerance of the filter dictates. The resulting spacecraft position and attitude are the new best estimate of the position and attitude, and this position fix is then output from the filter to be used as an observable for the Dynamical Filter, as described in Section 3.6.

3.5.1 Kinematic Position Partial Derivatives Function

The Kinematic Position Partials Derivatives Function calculates the analytic geometric partials of the pixel and line values of a landmark with respect to the spacecraft-to-landmark vector, given the spacecraft-to-landmark vector and the nominal pointing vector of the spacecraft camera, as described in reference [19].

In the first step, the rotation matrix from the inertial frame to the camera frame is calculated using the right ascension, declination and twist of the nominal camera pointing vector, as performed in the Pixel/Line Calculator Function in Section 3.4.5. The Equations (3.3) through (3.6) are utilized to calculate the transformation matrix $R_{i\to cam}$.

In the second step, this transformation matrix is applied to the spacecraft-to-landmark vector $\bar{r}_{sc-LM}$ to get the vector into the camera frame, which is labeled $\bar{p}$. To obtain the partial derivatives of the pixel and line values with respect to the spacecraft-to-landmark vector, the following equations must be executed for each of the three columns of the partial matrix:

$$\begin{bmatrix} \partial x / \partial \bar{r}_{sc-LM}(i) \\ \partial y / \partial \bar{r}_{sc-LM}(i) \end{bmatrix} = \frac{f}{\bar{p}(3)} \begin{bmatrix} \partial \bar{p}(1) / \partial \bar{r}_{sc-LM}(i) \\ \partial \bar{p}(2) / \partial \bar{r}_{sc-LM}(i) \end{bmatrix} - \frac{f}{(\bar{p}(3))^2} \begin{bmatrix} \bar{p}(1) \\ \bar{p}(2) \end{bmatrix} \frac{\partial \bar{p}(3)}{\partial \bar{r}_{sc-LM}(i)} \tag{3.14}$$

where i is the particular column of the partial derivatives matrix, x and y are the rectangular coordinates of the camera frame, f is the focal length of the camera, and those values in parenthesis indicate which of the specific components of the vectors are being used. The partial derivatives on the right side of the equation are equivalent to values in the $R_{i\to cam}$ transformation matrix, as described in the equations

$$\partial \bar{p}(1) / \partial \bar{r}_{sc-LM}(i) = R_{i\to cam}(1,i) \tag{3.15}$$

$$\partial \bar{p}(2) / \partial \bar{r}_{sc-LM}(i) = R_{i\to cam}(2,i) \tag{3.16}$$

$$\partial \bar{p}(3) / \partial \bar{r}_{sc-LM}(i) = R_{i\to cam}(3,i) \tag{3.17}$$

In the third step, the camera K-matrix is applied to the above partial derivatives of the rectangular coordinates to get the pixel and line partial derivatives with respect to the spacecraft-to-landmark vector for the particular column I of the partial derivatives matrix.

$$\begin{bmatrix} \partial pixel / \partial \bar{r}_{sc-LM}(i) \\ \partial line / \partial \bar{r}_{sc-LM}(i) \end{bmatrix} = \begin{bmatrix} K_x & K_{xy} \\ K_{yx} & K_y \end{bmatrix} \begin{bmatrix} \partial x / \partial \bar{r}_{sc-LM}(i) \\ \partial y / \partial \bar{r}_{sc-LM}(i) \end{bmatrix} \tag{3.18}$$

The total partial derivatives matrix is shown in the equation

$$\frac{d(pixel, line)}{d\bar{r}_{sc-LM}} = \begin{bmatrix} \partial pixel / \partial \bar{r}_{sc-LM}(1) & \partial pixel / \partial \bar{r}_{sc-LM}(2) & \partial pixel / \partial \bar{r}_{sc-LM}(3) \\ \partial line / \partial \bar{r}_{sc-LM}(1) & \partial line / \partial \bar{r}_{sc-LM}(2) & \partial line / \partial \bar{r}_{sc-LM}(3) \end{bmatrix} \tag{3.19}$$

3.5.2 Kinematic Pointing Partial Derivatives Function

The geometric partial derivatives of the pixel and line values of a landmark with respect to the spacecraft camera pointing vector are calculated by the Kinematic Pointing Partial Derivatives Function given the spacecraft-to-landmark vector and the nominal pointing vector of the spacecraft camera.

This function utilizes finite differencing to calculate the partial derivatives matrix, which is displayed in the equation

$$\frac{d(pixel, line)}{d(pointing)} = \begin{bmatrix} \partial pixel / \partial \alpha & \partial pixel / \partial \delta & \partial pixel / \partial \phi \\ \partial line / \partial \alpha & \partial line / \partial \delta & \partial line / \partial \phi \end{bmatrix} \tag{3.20}$$

3.6 DYNAMICAL FILTER

Like the Kinematical Filter, the Dynamical Filter utilizes a batch estimation algorithm, as described in Appendix A. However, instead of obtaining a position fix estimate for the spacecraft at a particular time as the Kinematic Filter does, the Dynamical Filter calculates a full state estimate for the spacecraft at the epoch time. This section will discuss the unique aspects of the Dynamical Filter, and for the exact operations of the batch filter see Appendix A. The Dynamical Filter requires at least two

Kinematic position fixes to operate, though this minimum amount can be modified by the user.

The Dynamical Filter requires the following:

- Position fixes from the Kinematic Filter, which serve as the observables for the Dynamical Filter
- A use vector specifying which position fixes can or cannot be used by the Dynamical Filter.
- A time vector which specifies the times of each of the position fixes.
- A weight array consisting of the weights associated with each position fix. The number of rows is equal to the number of position fixes and the number of columns is three. The weight array is created in the executive file by stacking the rows of weights output by the Kinematic Filter. The Kinematic Filter calculates the weight rows by placing the square root of the diagonal values of the Kinematic post-fit covariance matrix into a 1x3 row.
- The epoch time and nominal epoch state
- The Dynamical Filter apriori, or initial, covariance matrix. This covariance matrix is constructed according to the equation

$$P_i = \begin{bmatrix} \sigma_x^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_y^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_z^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{vx}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{vy}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{vz}^2 \end{bmatrix} \tag{3.21}$$

where the sigma values are the initial standard deviations for each position and velocity component.

The Dynamical Filter generates the following:

- A new estimate of the epoch state. Note that this is not an update of the current state, though an update of the current state can be obtained by propagating the updated epoch state forward to the current time.
- A new estimate of the epoch covariance.

The partial derivatives of the position fixes with respect to the state variables are needed for the batch filter algorithm in Appendix A. Because the observables are position values, the partial derivatives are very simple. The partial derivative of the position fix with respect to the spacecraft position vector is the identity matrix. The partial derivative of the position fix with respect to the spacecraft velocity vector is the zero matrix. These partials are multiplied by the state transition matrix to map the partials back to the epoch state.

3.7 MANEUVERS

3.7.1 Maneuver implementation

Maneuvers are computed at user-defined times. The Compute Maneuver Function, as described in Section 3.7.2, is provided the current nominal state and the maneuver time. The function outputs a change in velocity, or delta-v. If desired, maneuver execution errors are added to the truth model. The nominal delta-v is added to the current nominal state and the truth delta-v is added to the current truth state.

After a maneuver has been executed, all observations gathered before the maneuver are no longer available to the filter. Hence the filter is reinitialized for every maneuver, and thus the filter must wait until there are again a minimum number of observations after a maneuver before estimating the state.

3.7.2 Compute Maneuver Function

The Compute Maneuver Function outputs a change in velocity, or delta-v, given a state, current time, and epoch time. If asteroid rotation is activated, the target position is rotated from body-fixed coordinates to inertial coordinates at the target time. The K-matrix Targeting algorithm, as described in Section 2.2, is used to compute the change in velocity, or delta-v, needed to reach the target position at the target time.

Two other potential targeting algorithms are minimum-energy Lambert targeting and specified-initial-and-final-time Lambert targeting. The inclusion of asteroid rotation, however, dictated that a fixed final time targeting algorithm was needed. This necessity arises because the only inputs to the Lambert minimum energy algorithm are the spacecraft position, target position, and gravitational constant of the small-body. Thus, the travel time of the resulting trajectory can not be determined before implementing the algorithm, and thus the inertial target position at the time the spacecraft reaches the surface can not be determined. An iteration scheme might be achievable, though any future attempts to accomplish this must be careful to avoid the spacecraft final location "chasing" the target around the asteroid. If there is little or no asteroid rotation, however, the minimum targeting algorithm could be used.

Utilization of the Lambert Targeting algorithm was attempted, but this method was never able to achieve the desired accuracy for landing at the target. This deficiency is possibly due to the uniform gravitational field that the Lambert Targeting algorithm assumes, though this hypothesis was never proven. Thus, the K-matrix targeting algorithm, which is also a fixed final time method, was selected after its accuracy was confirmed.

3.7.3 Orbit Determination Cut-off

The Orbit Determination Cutoff value (OD-cutoff) is used to dictate how long before a maneuver the navigational filtering process must cease. Nominally this is set such that only the last picture time normally occurring before the maneuver is less than or equal to the OD-cutoff amount of time before the maneuver, thus allowing for an filtered spacecraft state as close as possible to the maneuver time. However, this is only valid for a fully autonomous system with no processes that take lengthy preparation time before a maneuver can be executed. If there are any such processes, the OD-cutoff must be set accordingly to simulate the time it would take for the processes to be executed.

A very important application of the OD-cutoff value is simulating the delay in communications with the ground if the user wishes to test the spacecraft's ability to land accurately without autonomous navigation. For example, the OD-cutoff might be set to five hours which would be about the amount of time required to send a signal from the asteroid location in the solar system to the Earth, have Earth-based Orbit Determination systems, i.e. "the ground", calculate a navigation solution and a maneuver, and finally to send a return signal from the Earth to the spacecraft. This is an excellent test to see if navigation does not need to be autonomous to land with sufficient accuracy at the target, which is of great interest to mission planners because of the many benefits of ground-based navigation over autonomous navigation. One of these benefits is the dramatically greater amount of heritage for ground-based navigation over autonomous navigation spacecraft systems. Another important benefit is that a less expensive, simpler navigation system would be required for the spacecraft, thus lowering the cost of the mission and increasing it's reliability and probability of successfully meeting mission goals. However, if one of the mission goals is to land within a specified distance of a target on a small-body, no amount of reduced mission costs and greater heritage for the ground-

based system will make it worthwhile if it cannot accomplish this task with sufficiently high probability. This test is conducted for the default profile as described in the Simulation Setup Section 4.0 and the results are given in the Simulation Results Section 5.0.

Multiple OD-cutoff values can be specified, with one for each maneuver. If the time intervals between the maneuvers are different, care must be taken when setting the maneuvers if the user wishes to let the last picture time before the maneuver be included in the navigation solution. A good default value for each OD-cutoff value is the time between pictures before the maneuver.

3.8 POINTING CALCULATION FUNCTION

The nominal and true spacecraft positions are utilized in the Pointing Calculation Function to calculate the nominal and true attitudes of the spacecraft. The nominal pointing, which includes the twist, declination, and right ascension, is calculated using the equations given below.

In the first step, the unit vector of the spacecraft-to-small-body position vector is cross multiplied with the unit vector of the spacecraft-to-Sun vector:

$$\hat{y} = \frac{\vec{r}_{sc-SB}}{\left|\vec{r}_{sc-SB}\right|} \times \frac{\vec{r}_{sc-sun}}{\left|\vec{r}_{sc-sun}\right|} \tag{3.22}$$

where $\vec{r}_{sc-SB}$ is the spacecraft-to-small-body position vector, and $\vec{r}_{sc-sun}$ is the spacecraft-to-Sun vector. In the second step, the cross product of $\hat{y}$ and the spacecraft-to-small-body position unit vector results in a third unit vector:

$$\hat{x} = \hat{y} \times \frac{\vec{r}_{sc-SB}}{\left|\vec{r}_{sc-SB}\right|} \tag{3.23}$$

The twist of the nominal pointing is calculated from the $\hat{x}$ and $\hat{y}$ unit vectors, as seen in the equation

$$\phi = \tan^{-1}\left(\frac{\hat{y}(3)}{-\hat{x}(3)}\right) \tag{3.24}$$

where (3) denotes the component of the vector being used. Note that because the twist is arbitrary, vectors other than the spacecraft-to-Sun vector can be utilized for the above calculations. The declination and right ascension angles are calculated as described in the equations

$$\delta = \sin^{-1}\left(\frac{\vec{r}_{sc-SB}(3)}{|\vec{r}_{sc-SB}|}\right) \tag{3.25}$$

$$\alpha = \tan^{-1}\left(\frac{\vec{r}_{sc-SB}(2)}{\vec{r}_{sc-SB}(1)}\right) \tag{3.26}$$

The true pointing includes the attitude error calculated for each time step by the Attitude Error Function described in Section 3.9.5. Inertial-to-body-fixed coordinates rotation matrices are created about each axis, with rotation equal to the corresponding attitude error for the current time. The three rotation matrices are multiplied by the nominal inertial-to-body-fixed transformation matrix, which consists of the stacked row vectors equal to the $\hat{x}$, $\hat{y}$, and spacecraft-to-small-body position unit vectors described above to produce the truth inertial-to-body-fixed transformation matrix. Finally the twist, declination, and right ascension are calculated using Equations (3.24) through (3.26), though the truth inertial-to-body-fixed transformation matrix is used instead of the $\hat{x}$, $\hat{y}$, and spacecraft-to-small-body position unit vectors.

3.9 ERRORS

There are a variety of errors that can be introduced to the nominal model which simulate potential errors that a spacecraft could experience while attempting to navigate autonomously. These different error sources are detailed below. All of these errors are controlled from the initialization file, as described in Section 4.0.

The various error coefficients described below are multiplied by a number that is drawn from a Gaussian, i.e. normal, distributed random variable with mean equal to zero and standard deviation equal to one. The MATLAB randn function generates this random number.

3.9.1 Initial State Error

The initial state error is the difference between the initial nominal state and initial truth state. From the initial nominal state, an initial state error is calculated and added to the initial nominal state to get the initial truth state. This initial state error calculation is explained below.

The user-set uncertainty values, or variances, in each axis of the View2 frame are multiplied by the pseudorandom output from the MATLAB randn function. These errors in the View2 frame are calculated for all three position and velocity components, and then both the position and velocity errors are rotated from the View2 frame to the inertial frame. These inertial errors are added to the nominal inertial position and velocity to get the true inertial position and velocity.

The error values can also be set manually by the user, which is useful when a particular error value is desired. The manual error setting also allows for the error to be the same for every run, though that can also be accomplished by setting the random number seed for the randn function, as is described in the Monte Carlo Section 3.10.

3.9.2 Observation Error

The observation errors in the pixel and line values in the camera field of view are calculated by multiplying a user-set pixel error coefficient and line error coefficient by the pseudorandom output from the MATLAB randn function. The observation errors in

the pixel and line values are added to the observed pixel and line values before being used by the Kinematic Filter.

3.9.3 Execution Error in Maneuver

The true maneuver that is applied to the spacecraft will not be the same as the nominal maneuver that the spacecraft assumes it has executed. Thus, some error must be added to the nominal maneuver to obtain the true maneuver. The Maneuver Execution Error Function performs this task by generating a true change in velocity (delta-v) given the nominal delta-v. The formulation for the error calculated is a standard Gates model [11] using uncertainty values (sigmas) in fixed magnitude and direction, and proportional magnitude and direction [8].

In the first step, the input nominal delta-v is used to calculate the rotation matrix from the Cartesian frame to the Topocentric South-East-Up (SEZ) frame. This calculation is done using the algorithm from Section 2.3.5. In the second step, the Maneuver Execution Error Function samples for errors both perpendicular to and along the velocity direction, as described in the equations

$$\begin{aligned} \Delta v_1 &= \sigma_{fixdir} R_{1\times1} + \sigma_{propdir} |\Delta v| R_{1\times1} \\ \Delta v_2 &= \sigma_{fixdir} R_{1\times1} + \sigma_{propdir} |\Delta v| R_{1\times1} \\ \Delta v_3 &= \sigma_{fixmag} R_{1\times1} + \sigma_{propmag} |\Delta v| R_{1\times1} \end{aligned} \tag{3.27}$$

where σ_{fixdir} is the fixed direction error coefficient, σ_{fixmag} is the fixed magnitude error coefficient, $\sigma_{propdir}$ is the proportional direction error coefficient, and $\sigma_{propmag}$ is the proportional magnitude error coefficient. Each $R_{1\times1}$ value is a single random number generated by the MATLAB randn function, and Δv is the input delta-v.

These delta-v error values in the Topocentric South-East-Up (SEZ) frame are then rotated into the inertial frame using the transpose of Cartesian-to-Topocentric rotation matrix calculated above, and then added to the nominal delta-v to get the truth delta-v. If

additional maneuvers are introduced, more maneuver error will also be added. However, retargeting at each maneuver works to remove some of the error resulting from previous maneuvers.

3.9.4 Modeling Error

Modeling error can be introduced to the simulation by not including some or all of the perturbations in the dynamical nominal model for the spacecraft motion that are being applied to the truth model. The third body perturbations, SRP perturbations, and gravitational harmonics perturbations can all be activated or deactivated in both the truth and nominal models. The general theory behind each of these perturbations on the spacecraft motion can be found in Section 2.1.2. More detailed information about how each of these perturbations is implemented in SANT is given in the sections below.

3.9.4.1 Third Body Perturbations

The third body perturbations are a result of gravitational forces from bodies outside the two-body system of the spacecraft and central body, which alter the two-body motion of the spacecraft. It is not necessary to include every other body in the solar system for such calculations in this simulation, so only those third bodies which can introduce significant perturbations on a spacecraft orbiting an asteroid in the asteroid belt are considered. Thus, in the initialization file the dominant effects from the Sun, Jupiter and Earth are "turned on" and all other bodies are "turned off".

In order to avoid calling an ephemeris at every time in the simulation, which would dramatically increase the amount of time required to run the simulation, the states for every third body are called from the ephemeris in the initialization file and these states are propagated assuming two-body motion around the Sun, except for the Moon

which is propagated around Earth. Equation (2.5) from Section 2.1.2.1 is then used to calculate the perturbation acceleration on the spacecraft.

3.9.4.2 Solar Radiation Pressure Perturbations

The SRP perturbation is a result of the solar radiation that reaches the spacecraft and alters the spacecraft motion. As with the third body perturbations, the Sun's position relative to the asteroid is initially determined in the initialization file, and the asteroid is propagated around the Sun to the current picture time. Thus the Sun's position relative to the spacecraft can be determined if the spacecraft's position relative to the asteroid is known, and Equations (2.6), (2.7), and (2.8) from Section 2.1.2.2 can be used to calculate perturbation acceleration on the spacecraft. For Equations (2.6), (2.7), and (2.8), the solar flux at the Earth is considered a constant of 1367 Watts per meter squared, and the distance between Earth and Sun is also considered a constant of 149597887.5 kilometers. The spacecraft coefficient of reflection is set by the user.

The spacecraft mass and projected area used in Equations (2.6), (2.7), and (2.8) are different for the truth and nominal models. The difference in the true spacecraft mass and user-set nominal spacecraft mass is calculated by multiplying a mass error coefficient by a single random number generated by the MATLAB randn function. This difference is added to the nominal spacecraft mass to obtain the true mass. The same process is performed for the projected area using a projected area error coefficient. Thus, the simulation generates two different forces based on the type of model being used. For the nominal model, the user-set nominal spacecraft projected area and mass are used, and the truth model uses the calculated true spacecraft projected area and mass.

3.9.4.3 Gravitational Harmonics Perturbations

The gravitational harmonics perturbations introduced on the spacecraft motion from the varying gravitational field of the asteroid are calculated using the Gravitational Harmonics Perturbations Function. The gravitational field of the asteroid is not a constant due to the non-spherical shape of the body varying densities of mass throughout the body. This function calculates the perturbation force on the spacecraft at a given time given the spacecraft state and time.

In the first step, the spacecraft inertial position is rotated into the body-fixed reference frame. In the second step, the gravitational harmonics coefficients for the small-body are provided, normally from an external text file provided by the user. These gravitational harmonics coefficients include the zonal, tesseral, and sectorial coefficients.

Using the small-body gravitational constant, the zonal, tesseral, and sectorial gravitational harmonics coefficients, and the body-fixed position of the spacecraft, the method developed in reference [20] is used to calculate the body-fixed force on the spacecraft. This force is rotated back into the inertial frame before being applied to the spacecraft.

3.9.5 Attitude Error

Given the estimated spacecraft state relative to the asteroid center, the attitude control system, which is not modeled in this simulation, commands the spacecraft camera to point directly at the origin of the asteroid-centered coordinate system (the asteroid center of mass) at every picture time. The attitude attained, however, is not quite the commanded attitude. Thus, the spacecraft has an estimate of its attitude (known attitude), and there is also a truth attitude. The difference between the known attitude and truth attitude is the attitude knowledge error, which is of greater concern for navigation than the difference between the truth attitude and the commanded attitude. The attitude

knowledge error, hereafter known simply as the "attitude error", is calculated in the Attitude Error Function.

Attitude errors of a typical spacecraft gyroscope are used to estimate the attitude error in this simulation. While a star tracker will usually have an initial error and random biases, gyroscopes will have initial error, correlated biases, drift and random walk. SANT must account for all four of these values.

Using the vector of picture times, and the user-set standard deviations (one-sigma) of a typical onboard gyroscope attitude noise, drift rate, initial attitude error, and angle random walk, the Attitude Error Function generates a Gaussian random attitude error about the x, y, and z axes. Before any attitude error values are generated, a temporary time vector is generated at small time increments from the simulation start time to the simulation end time. The below attitude error values will correspond to the times in this time vector.

The initial error is calculated using the equation

$$E_0 = R_{1\times1}\sigma_{init} \tag{3.28}$$

where E_0 is the initial error, R_{1x1} is the single value random number generated by the MATLAB randn function, and σ_{init} is the standard deviation of the onboard gyroscope initial attitude error. The attitude noise error is calculated by the equation

$$\overline{E}_{noise} = R_{n\times1}\sigma_{noise} \tag{3.29}$$

where $\overline{E}_{noise}$ is the attitude noise error vector, $R_{n\times1}$ is a vector of n random number values generated by the MATLAB randn function, n is the number of times in the time vector, and σ_{noise} is the standard deviation of the onboard gyroscope attitude noise. The attitude drift error is determined according to the equation

$$\overline{E}_{drift} = \bar{t}_{rel}\sigma_{drift}R_{1x1} \tag{3.30}$$

where $\overline{E}_{drift}$ is the attitude drift error vector, $\bar{t}_{rel}$ is the vector of time values relative to the first value in the time vector, σ_{drift} is the standard deviation of the onboard gyroscope drift rate, and R_{1x1} is the single value random number generated by the MATLAB randn function. The random walk attitude error is calculated using the equation

$$E_{RW} = \sum_{n} RW\sqrt{dt}R_{nx1} \tag{3.31}$$

where E_{RW} is the random walk attitude error, RW is the user-set random walk coefficient, and *dt* is the time between the first two time values in the time vector. The product inside the summation is then summed over all time values to determine the total random walk attitude error. The total attitude error for a particular axis for every time in the time vector is then given by the expression

$$\overline{E}_{total} = E_0 + \overline{E}_{noise} + \overline{E}_{drift} + E_{RW} \tag{3.32}$$

where the constant values E_0 and E_{RW} are added to every entry in the vector.

The attitude error values at every time in the temporary time vector are linearly interpolated across the vector of picture times to get the attitude error at every time in the picture time vector. This attitude error is utilized in the Pointing Calculation Function as described in Section 3.8.

3.10 MONTE CARLO

Monte Carlo analysis, which involves repeated random sampling of the various error sources to study the nature of a system, is utilized to generate surface scatter plots of the landing error for the spacecraft as it attempts to land as close as possible to the target position.

3.10.1 Monte Carlo versus Covariance Analysis

In an ideal situation, a covariance ellipse that is projected to the surface would perfectly match the landing footprint, with the 3-sigma covariance ellipse containing 99%

of the landing positions. However, this matching can only occur if all the dynamics of the system are modeled perfectly. In a real scenario, the onboard spacecraft computer is not going to be able to perfectly model all of the various forces at work; thus, since the covariance is likely an optimistic representation of the error, Monte Carlo analysis must be conducted. A user could generate an error ellipse that encompasses 99% of the landing positions, but this would not necessarily be equivalent to the 3-sigma covariance ellipse projected to the surface.

Another obstacle to finding the covariance at the surface is the need to estimate the maneuvers, i.e. fit through the maneuvers. In the present configuration of the tool, the filter is simply reinitialized after each maneuver due to the complexity of attempting to estimate the maneuver. Performing all of the maneuvers long before the landing, however, might also lead to a reasonable, though not exact, covariance at the surface.

As a result of utilizing Monte Carlo instead of covariance analysis, the statistics such as the percentage of landings that make it within a certain distance of the target, or landed on the asteroid at all, take on added importance. These values are investigated in the Simulation Results Chapter 5.

3.10.2 Monte Carlo Function

A useful aspect of the Monte Carlo Function is the ability to set the random number seed for the MATLAB randn function, which allows the user to directly compare two different configurations of the tool by providing the same random numbers for the error sources. By setting the seed sequentially for multiple cases of the simulation, Monte Carlo analysis can be conducted and the results from each simulation profile can be directly compared.

The Monte Carlo Function begins by setting the initial random number seed and the number of simulations that are to be run, i.e. the number of times that the executive

script is to be run. The seed is set equal to the current number of the Executive file run, allowing the user to return to a particular run to investigate if needed. If the Monte Carlo process is interrupted before completing all cases, the seed can be set to one more than the last run successfully completed and thus the results from previous cases are not lost.

If the starting seed is 1, the various saved parameter arrays are initialized. These parameters include the following:

- final spacecraft body-fixed position
- final true targeting error
- final nominal targeting error
- final true targeting error magnitude
- final knowledge error
- final knowledge error in Topocentric coordinates
- total truth and nominal delta-v produced by the maneuvers

If the starting seed is not 1, the values from previous cases are loaded by MATLAB. The executive script is run for every random seed value, and the above saved parameters for each run are collected in arrays. These values will be utilized for various plots and other statistical analysis.

CHAPTER 4

Simulation Setup

The setup of the simulation is almost entirely done in the initialization script. The details of this script and how each simulation is set up to conduct a spacecraft landing scenario are described in the sections below. A full listing of every initialization value and switch for the baseline scenario can be found in Appendix B. The values described below represent the baseline scenario values, i.e. default values.

Nominally the initial state errors are calculated using initial state perturbation multipliers, which for the position are 0.0025 kilometers in each View2 axis, and for the velocity are 2.5×10^{-6} kilometers per second in each View2 axis.

The targeting algorithm chosen is K-matrix Targeting. The fixed magnitude coefficient is 2×10^{-6} km/s, the proportional magnitude coefficient is 0.0002, the fixed direction coefficient is 4×10^{-7} kilometers per second, and finally the proportional direction coefficient is 0.0002, or 0.2 milli-radians pointing error.

The gyroscope properties affecting the attitude error are an initial attitude error coefficient of 0.1°, an attitude noise coefficient of 3.3 micro-radians, an initial gyro drift rate of 0.01/3 degrees per hour, and a gyro random walk coefficient of 0.025 degrees per square root of hours.

The pixel and line error multipliers are both 0.25 pixels. The truth model perturbations of gravity harmonics, SRP, and third body perturbations are all turned on. The same perturbations for the nominal model are all turned off.

The surface target is set at a longitude of 270° and a latitude of -2°, with an altitude of 4 meters above the surface. This target location was chosen in an attempt to

minimize the number of times that the spacecraft would miss the asteroid when aiming for the target, as it is relatively close to center of mass of the asteroid compared to other surface positions. The initial time for the simulation is November 24, 2017, at 9:00 am. The maneuver time is November 25, 2017, at 5:50 am. The target time, or time at which the targeting algorithm will attempt to make the spacecraft reach the target, is November 25, 2017, at 11:00 am.

The spacecraft nominal mass is 500 kilograms, with an error multiplier of 20 kilograms. The spacecraft nominal projected area, used for SRP perturbations, is 12 meters squared, with an error multiplier of 0.5 meters squared. The spacecraft coefficient of reflectivity is 1.1, where a value of 0 is equivalent to a perfectly transparent spacecraft, 1 is equivalent to a perfectly black spacecraft that absorbs all radiation, and 2 is equivalent to a perfectly reflecting mirror. These values for the spacecraft mass and projected area are estimates based on the mass and surface area of the solar arrays of the Hayabusa mission spacecraft that landed on the asteroid Itokawa. Hayabusa was about 510 kg and the solar arrays had a combined area of about 12 square meters [12].

The asteroid gravitational constant is 3.62×10^{-8} kilometers cubed per second squared. Thus the asteroid mass is roughly nine orders of magnitude greater than the spacecraft mass, allowing the spacecraft to be considered negligible when calculating the 2-body gravitational acceleration. The asteroid is modeled as a triaxial ellipsoid, with a maximum radius of 0.71646 kilometers, middle radius of 0.64984 kilometers, and minimum radius of 0.52361 kilometers. In the body-fixed reference frame, these values coincide with the x, y, and z axes, respectively. The initial orientation is defined by the initial right ascension of the asteroid pole at 30°, the initial declination of the pole at 40°, and the initial longitude of the prime meridian at 50°. The rotation rates of the asteroid are the right ascension rate of the pole at 2° per Julian century, which is 100 Julian years

or 36,525 days, the declination rate of the pole at 3° per Julian century, and finally an asteroid rotation rate of 30° per day.

The Kinematic Filter properties that are settable by the user include the position convergence tolerance, which is set at 1 meter, and the attitude convergence tolerance, which is set at 0.005°. For the initial covariance of the Kinematic Filter, the standard deviations for the position are set at 5 kilometers for every axis, and the standard deviations for the attitude are $1x10^{-5}$ degrees for the three pointing vector values: right ascension, declination, and twist. The minimum number of Kinematic position fixes that are required before the Dynamical Filter is allowed to operate is two; the maximum number of iterations allowed for the Dynamical Filter is ten; and the convergence tolerance is $1x10^{-5}$ kilometers. For the initial covariance of the Dynamical Filter, the standard deviations for the position are set at 5 kilometers for every axis, and the standard deviations for the velocity are $1x10^{-2}$ kilometers per second degrees for every axis.

The propagation options for the ode45 integrator used by SANT are a relative and absolute tolerance of $1x10^{-9}$.

The camera properties are set as follows:

- the focal length is set to 10 millimeters
- the K-matrix conversion matrix is set to $\begin{bmatrix} 83.3330 & 0 \\ 0 & 83.8333 \end{bmatrix}$ pix/mm
- the central pixel is set as [256 256] pixels
- the camera field of view dimensions are set as [512 512] pixels

The picture sequence properties include the initial time between pictures of one hour, and the time between pictures after the first maneuver of 10 minutes.

The third bodies that are turned on for the third body perturbations include the Earth, Jupiter, and the Sun. All other third bodies are turned off.

The spacecraft at the beginning of the simulation is directly above the target position and is in a circular orbit. This orbit was chosen because it is assumed for a mission that is attempting to land on an asteroid, that mission operators will want detailed photographs of the target location on the asteroid surface. Thus, the initial orbit will bring the spacecraft directly over the target to maximize the time available to image the location and also minimize the distance to the target, assuming the orbit is circular.

The initial orbit radius is set to three times the asteroid radius at the target position. This initial radius is chosen to ensure the spacecraft does not get too close to the surface, as the target position is located at the minimum radius axis. A circular orbit with this radius also conveniently has an orbital period of approximately one day for this asteroid. The maneuver time was chosen at the time given in the above section because the spacecraft is approximately 75% of the way through the circular orbit at that time. The simulation end time, assuming the spacecraft does not reach the surface before this time, is the target time plus two hours. The orbit determination cutoff time is set equal to the time between the pictures immediately before the maneuver. The target Cartesian coordinates are calculated using the algorithm in Section 2.3.3, given the target radius, target longitude, and target latitude.

The nominal state must be initialized next. The nominal position is calculated by multiplying the target unit vector by three times the target radius. The nominal velocity of the approximately circular orbit is then calculated according to the equation

$$v_{nom} = \sqrt{\frac{\mu}{|\bar{r}_{nom}|}} \frac{[1 \quad 0 \quad 0] \times \bar{r}_{nom}}{|[1 \quad 0 \quad 0] \times \bar{r}_{nom}|} \tag{4.1}$$

where v_{nom} is the nominal velocity, μ is the gravitational constant of the asteroid, and $\bar{r}_{nom}$ is the nominal radius vector, or nominal position. Other vectors besides [1 0 0] could be chosen for the cross products in Equation (4.1), as long as they are not parallel

to the nominal radius vector. The nominal position and velocity are then combined to obtain the initial nominal state. The orbital period is then calculated by the equation

$$T_p = 2\pi\sqrt{\frac{\left(\bar{r}_{nom}\right)^3}{\mu}} \tag{4.2}$$

The error in the initial nominal position and velocity that is needed to determine the truth initial nominal position and velocity is calculated according to the Initial State Error Section 3.9.1.

If the gravity harmonics perturbations are turned on in either the truth or nominal model, the text file with gravity coefficients described in the Gravitational Harmonics Perturbations Section 3.9.4.3 is read by a text parser which sorts all of the gravity coefficients provided in the text file into appropriate arrays. These arrays are then placed in a global structure which is used in the calculation of the gravitational harmonics perturbations.

The asteroid and third body states with respect to the Sun, or Earth as is the case for the Moon, are determined in the initialization file for the simulation start time if the third body perturbations for either the truth or nominal model are turned on. These states can be propagated to any future time in the simulation, as is described in the Third Body Perturbations Sections 3.9.4.1 and 2.1.2.1. This approximation does not introduce any significant error. The ephemeris used is the 46P/Wirtanen small-body ephemeris published by JPL, which is available online through JPL's HORIZONS system. This ephemeris includes the DE405 ephemeris data for all the planetary bodies in the solar system.

CHAPTER 5

Simulation Results

In order to study the conditions, and in particular to identify the error sources, that affect the size and shape of the landing footprint for a spacecraft attempting to land at a particular target on a small-body using AON, Monte Carlo simulations were run. These simulations revealed not only how accurately a spacecraft can land on the surface, but also the regions in which the spacecraft is more likely to land.

Throughout the results below, a number of errors are determined, which include the following:

- Landing Knowledge error = Estimated – Truth state at Landing
- Target error = Truth Target error = Truth position at landing – Target Position
- Nominal Target error = Estimated position at landing – Target Position

Since the study was based on a simulation and the truth was known, the actual errors could be determined. The plots given below display these errors in the Topocentric SEZ frame, which is easily accomplished by rotating the body-fixed target position, truth position at landing, and estimated position at landing into the Topocentric frame using the algorithm in Section 2.3.5.

The primary goal of this experimentation was to determine the sensitivities of the final target error and knowledge error to the various navigation error sources that exist and can be introduced in SANT. In order to determine these sensitivities, the simulation conditions were varied. By studying how the input navigation errors and other altered parameters affected the shape of the landing footprint, the source of particular features could be ascertained. Thus different parameters of the simulation profile were varied in a

consistent fashion, such as reducing a navigational error source by 50% and then increasing it by 200% from the baseline profile values defined in Chapter 4 and Appendix B. For each subsection below, 500 cases in the Monte Carlo simulation were conducted. Before any other experimentation was done, a baseline profile was established and the results for this case are displayed in the subsection below.

5.1 BASELINE PROFILE

The baseline profile is fully listed in Appendix B. This section describes a single case in detail, which is followed by the results from a 500 case Monte Carlo Simulation. The trajectory results for the single case are displayed below. The trajectory shown in Figure 5.1 is in the body-fixed coordinate system.

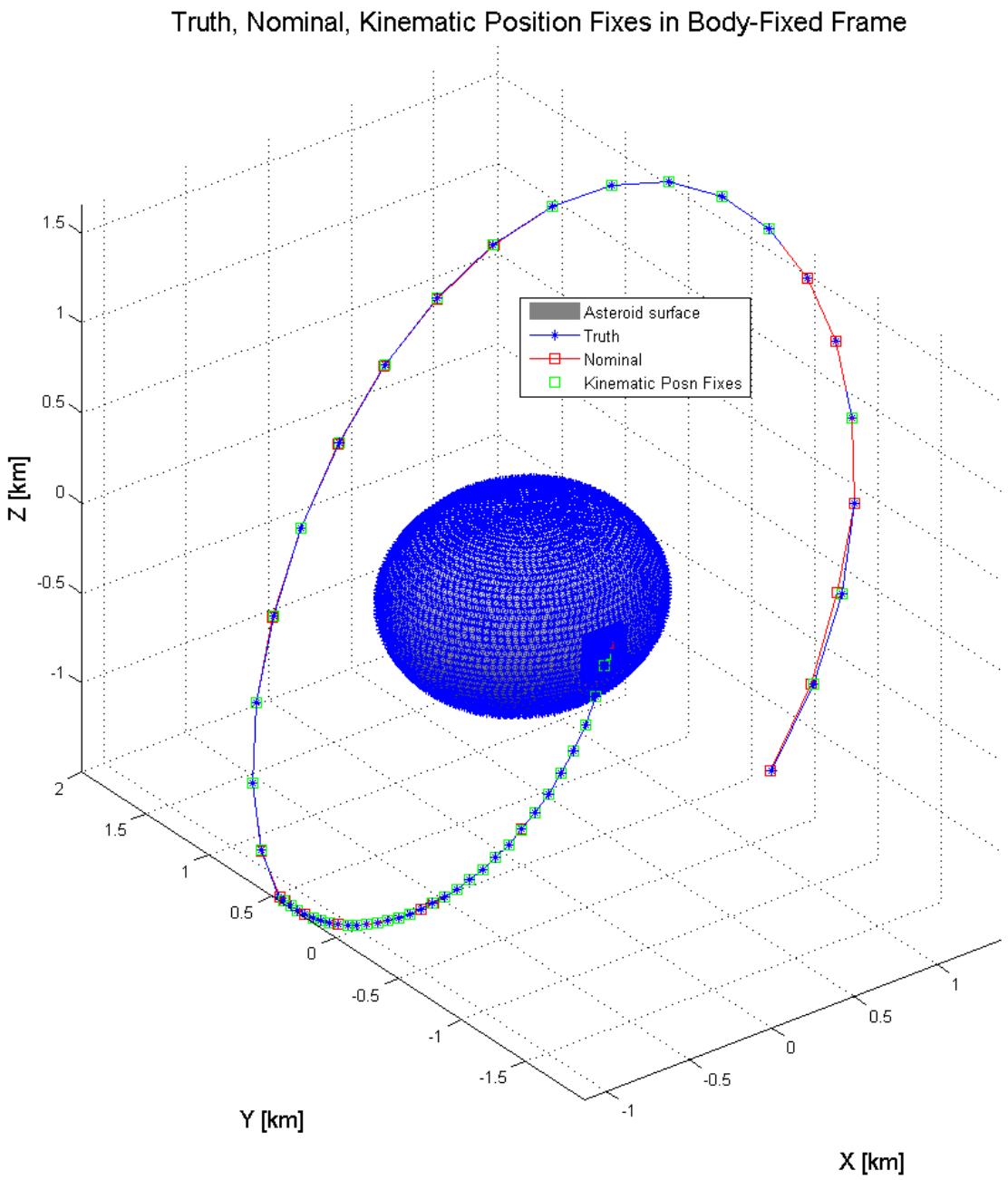

Figure 5.1: Baseline Profile Body-fixed Trajectory

From the trajectory in Figure 5.1, the spacecraft initial position can be seen to start directly above the target position on the asteroid surface. It is also seen that

immediately after the maneuver point (which occurs approximately 75% of the way around the orbit), the number of navigation solutions increases from one every hour to one every ten minutes. After the maneuver, the spacecraft trajectory is then seen to move from the position at the maneuver to the target position on the surface.

The red squares with a red line through them indicate the estimated states, the blue asterisks with a blue line through them indicate the truth states, and the green squares indicate Kinematic position fixes at each picture time. At almost every point in the trajectory, these three values are almost the same due to the excellent performance of the navigation filters. The one area that they visibly diverge is in the beginning of the trajectory, and in particular the third position marker in the trajectory. This divergence arises because the Dynamical Filter is not able to produce a solution and thus allow the estimated trajectory to "lock on" to the truth trajectory until at least two Kinematic position fixes are available. The estimated state does not align with the truth state in the beginning of the trajectory due to this requirement, and thus the estimated state fed into the Kinematic Filter is not as close to the truth state as it normally would be. Despite this larger than normal knowledge error, the Kinematic position fixes are still seen to match very closely to the truth states, which indicates that the Kinematic Filter is performing well. The blue points on the gray surface are optical landmarks that the spacecraft can use as observables in the Kinematic Filter. The target region has an even greater density of landmarks, as seen by the denser set of blue points in a square centered at the target position.

This same trajectory is shown in the inertial reference frame in Figure 5.2. The asteroid body is not displayed in Figure 5.2 because it is rotating in this frame and thus would have a different orientation at every time in the trajectory.

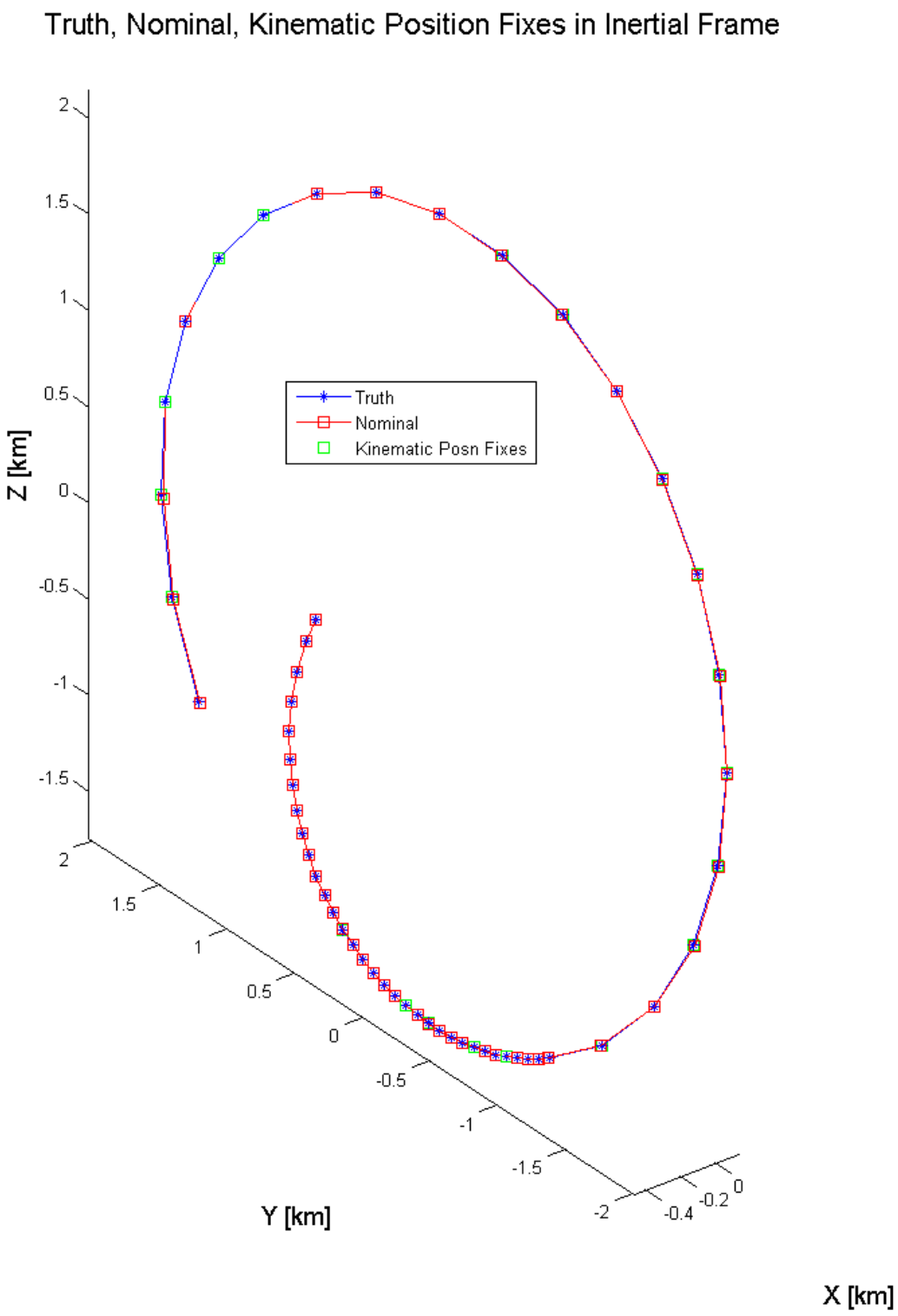

Figure 5.2: First seed Monte Carlo Inertial Trajectory

The 500 case Monte Carlo simulation was executed for the baseline profile. Several important statistics such as the number of times the spacecraft did not land on the surface by two hours after the target time (the desired landing time at the target for the spacecraft), i.e. the number of misses, and the average and standard deviation of the target error magnitudes for every case are presented in Table 5.1.

Table 5.1: Baseline Monte Carlo Statistics

Number of misses out of 500 cases	33 (6.6%)
Average target error magnitude	32.01 meters
Target error magnitude standard deviation	58.35 meters

The true target error as defined at the beginning of Chapter 5 was calculated for the final true position of every case. These values are plotted in Figure 5.3.

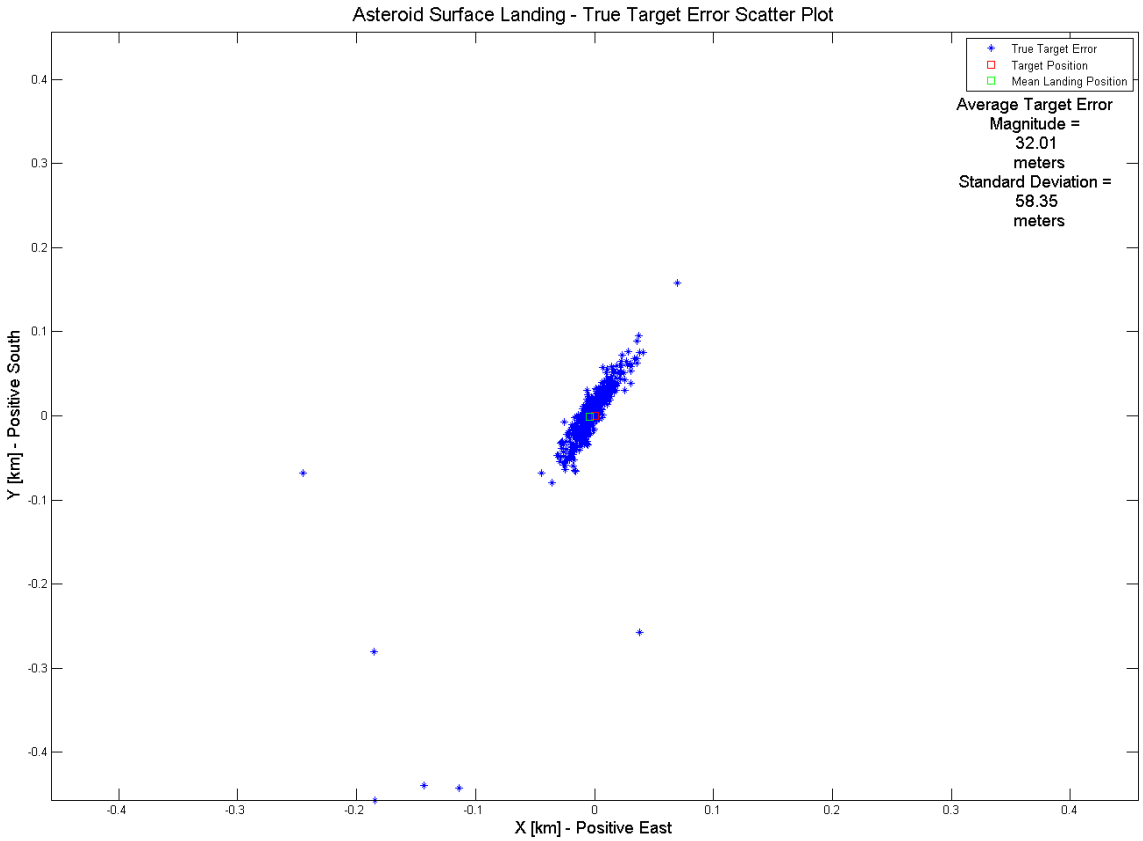

Figure 5.3: Baseline True Target Error Scatter Plot

The landing points are represented by blue asterisk points in Figure 5.3. The target position is represented by a red square located at the origin of the plot. The green square is the average position of all the landing positions. Note that the cases which result in no landing by two hours after the target time are not included in this average, a fact that will be important later. The true target error scatter plot shows that most landings are within a rectangular region centered about the target position, with some outliers as well. The mean landing position is very close to the target position. The orientation of the strip is in the same general direction of the spacecraft's motion near the surface, otherwise known as the along-track direction. The nominal target error is calculated for every landing and this is shown in Figure 5.4.

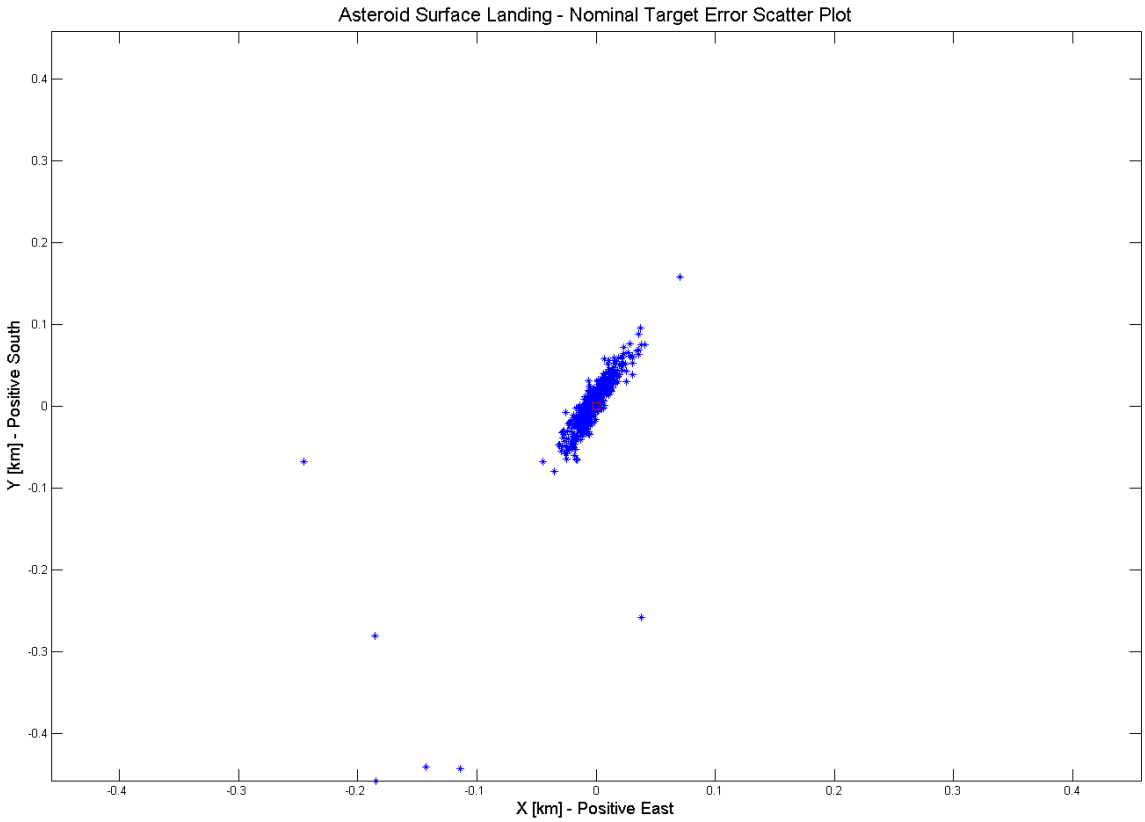

Figure 5.4: Baseline Nominal Target Error Scatter Plot

The nominal target error scatter plot in Figure 5.4 is visually almost identical to the truth target error scatter plot. This similarity is a result of very small knowledge error, which is shown in Figure 5.5.

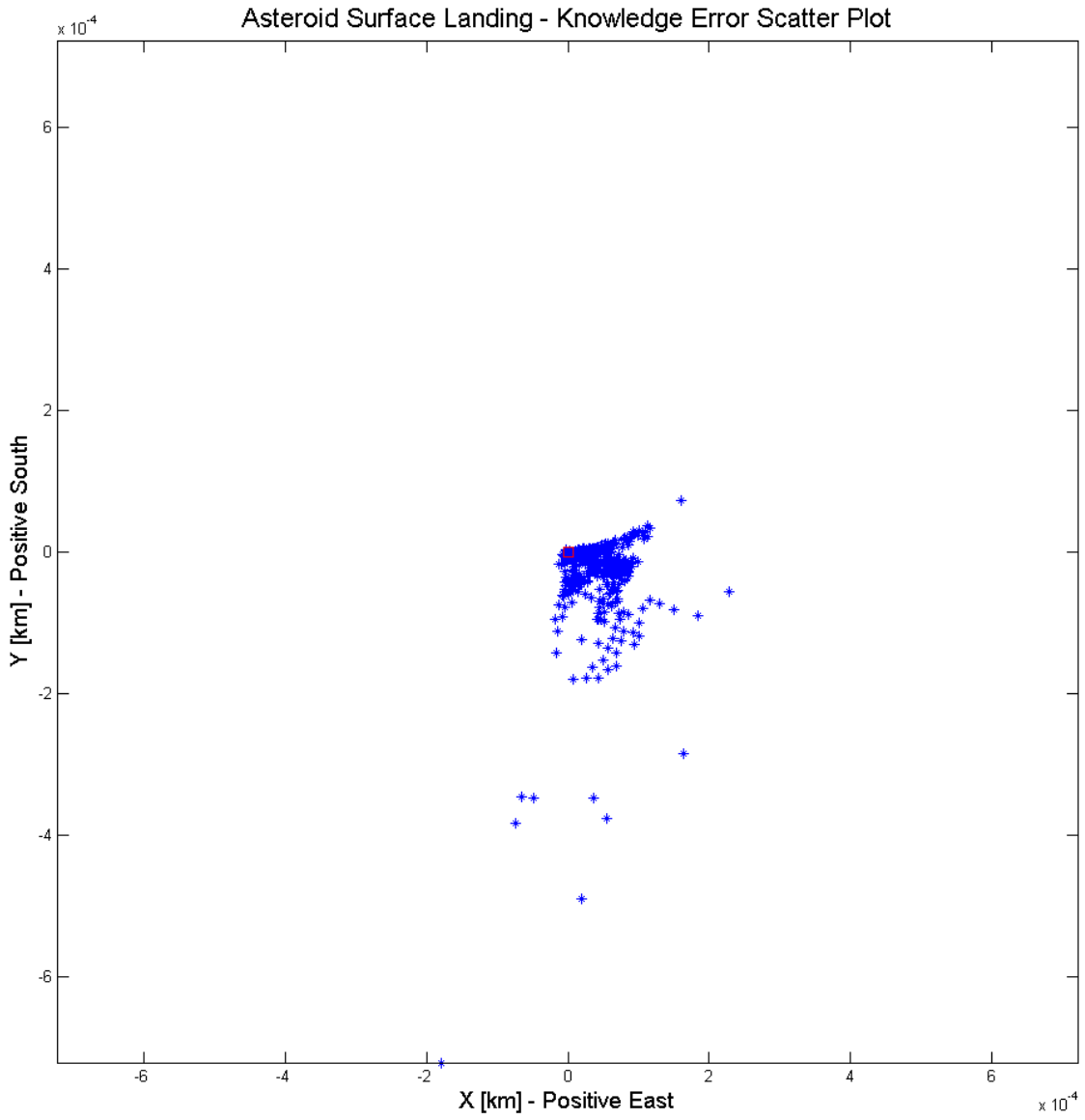

Figure 5.5: Baseline Knowledge Error Scatter Plot

The knowledge error is very small, with multipliers on the above axes of tens of centimeters. This smallness is a result of the estimated final states being very close to the truth final states, which is due to the excellent performance of the filters. The majority of the points are located within a slightly greater than 90 degree angular area around the origin, in a fan-like shape, with greater density as the points approach the origin and the

edges of the fan-like shape. This shape suggests that the spacecraft state estimated on board the spacecraft is consistently different from the true landings within a certain angular area, which indicates a bias is being introduced somewhere in the filter estimation process or perhaps by the very nature of the system dynamics.

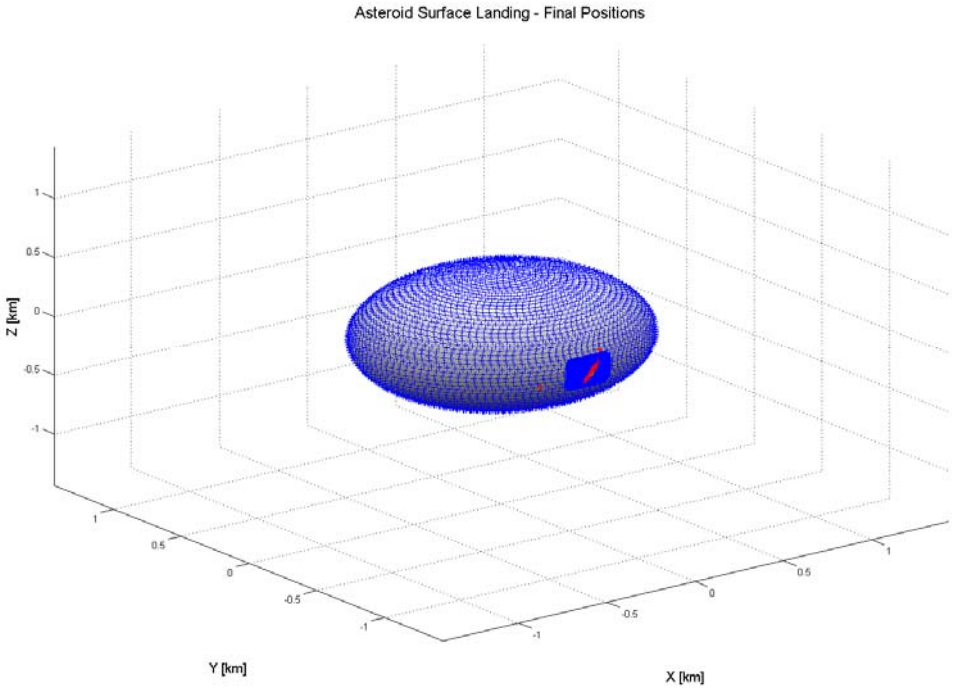

Figure 5.6: Baseline Final Positions

The final positions at landing are plotted on the asteroid body in Figure 5.6 above. Unfortunately from this angle, the scatter of the landing footprint is not easily viewed. Thus the above three dimensional plot is rotated to show the X-Z plot, and this plot is shown in Figure 5.7.

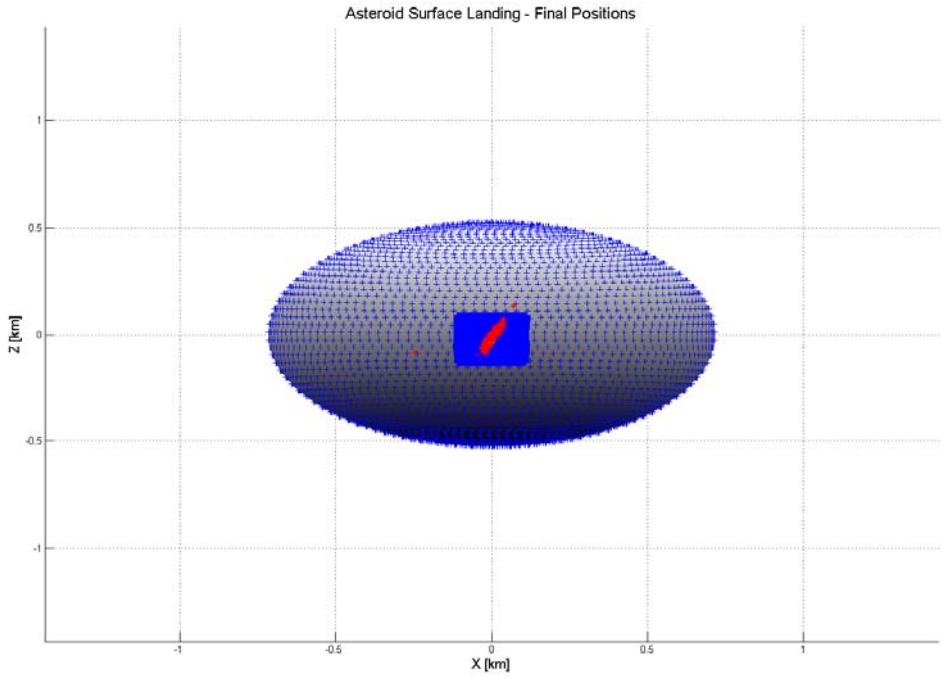

Figure 5.7: Baseline Final Positions - Rotated

While this image is better than the view shown in Figure 5.6, the plot is still quite small and difficult to see. So the plot is zoomed in and centered at the target position. This improved image is shown in Figure 5.8.

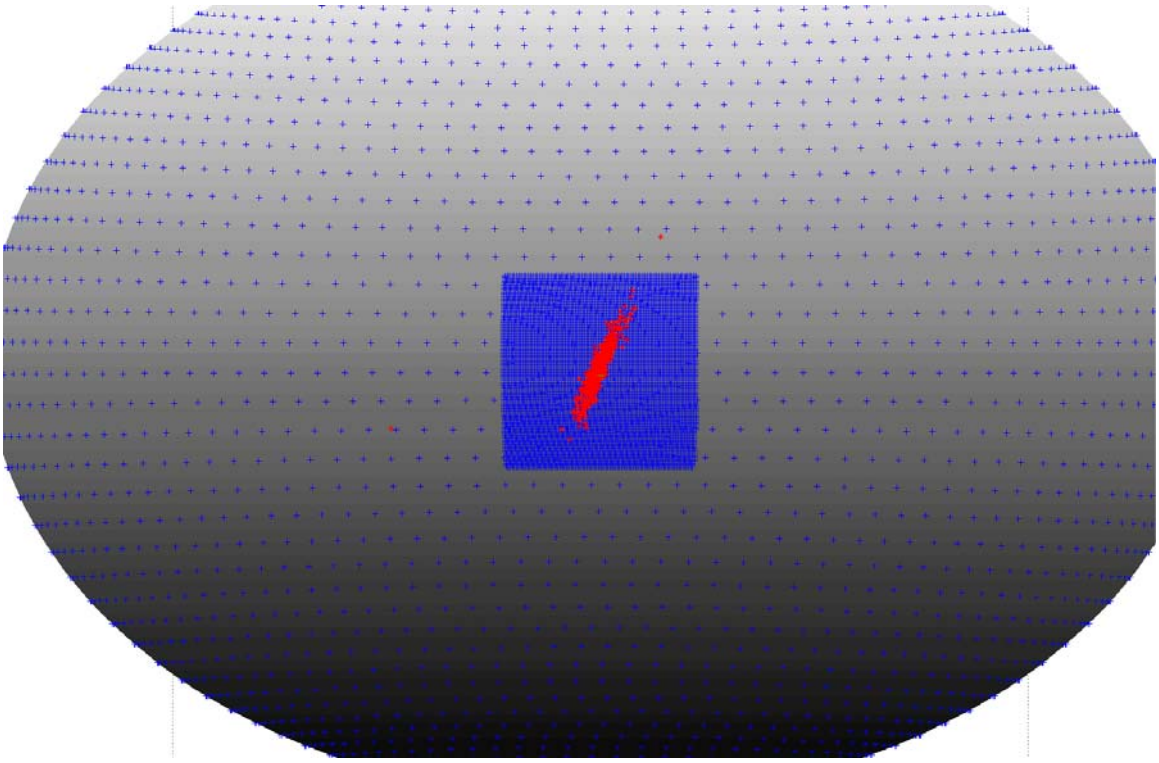

Figure 5.8: Baseline Final Positions – Rotated and Zoomed

The scatter plot, or footprint, in Figure 5.8 is identical to the footprint shown in the Baseline True Target Error Scatter Plot of Figure 5.3, but in this plot one can see how the size of the footprint compares with the size of the asteroid and the area of greater density landmarks, which is the blue square in the middle of the above plot.

5.2 50% MANEUVER EXECUTION ERRORS

Now the maneuver execution errors are reduced by 50%. Thus the following values are reduced by 50%:

- Fixed magnitude = 2e-6 km/s
- Proportion/fraction of magnitude = 0.0002
- Fixed Direction = 4e-7 km/s
- Proportion/fraction of direction = 0.0002 = 0.2 mrad ptg error

The same landing statistics from Table 5.1 are presented in Table 5.2.

Table 5.2: Monte Carlo Statistics - 50% maneuver execution errors

Number of misses out of 500 cases	33 (6.6%)
Average target error magnitude	19.61 meters
Target error magnitude standard deviation	56.61 meters

In the case of 50% maneuver execution errors, the exact same cases missed the asteroid as in the baseline profile, suggesting that maneuver execution errors do not have a large influence on the outliers and cases that result in no landings. However, the average target error magnitude dropped significantly, suggesting that those landings which were reasonably close to the target in the baseline were now even closer to the target. The standard deviation was not reduced significantly, suggesting that while the footprint became smaller, the density near the target did not drop dramatically, perhaps because another error source is more influential closer to the target.

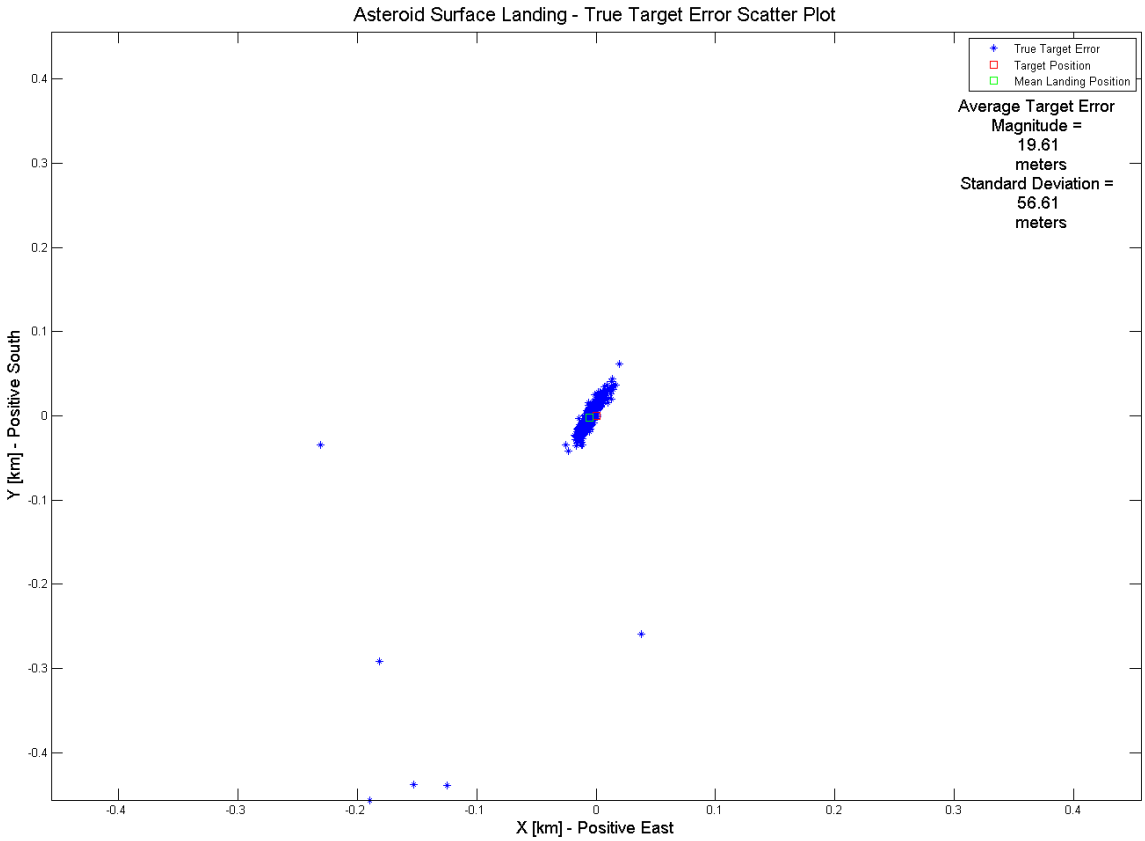

Figure 5.9: True Target Error Scatter Plot - 50% maneuver execution errors

When compared to the baseline true target error scatter plot, one can see the footprint in Figure 5.9 is significantly smaller, though of the same general shape with the same general orientation. As in the baseline scenario, the mean landing position is very close to the target position. The strip of landing positions also appears to be slightly less centered on the target position than the baseline case, though this may simply be a result of the footprint size reduction. All of the outliers occur at the same location, again suggesting the maneuver execution error has very little or no influence on outliers and non-landings.

The nominal target error scatter plot is not presented in the following sections except in the cases where it is significantly different from the true target error scatter plot.

The nominal and true target error scatter plots are almost identical in every case due to the relatively much smaller magnitude of the knowledge error, which is the difference between the true and nominal plots.

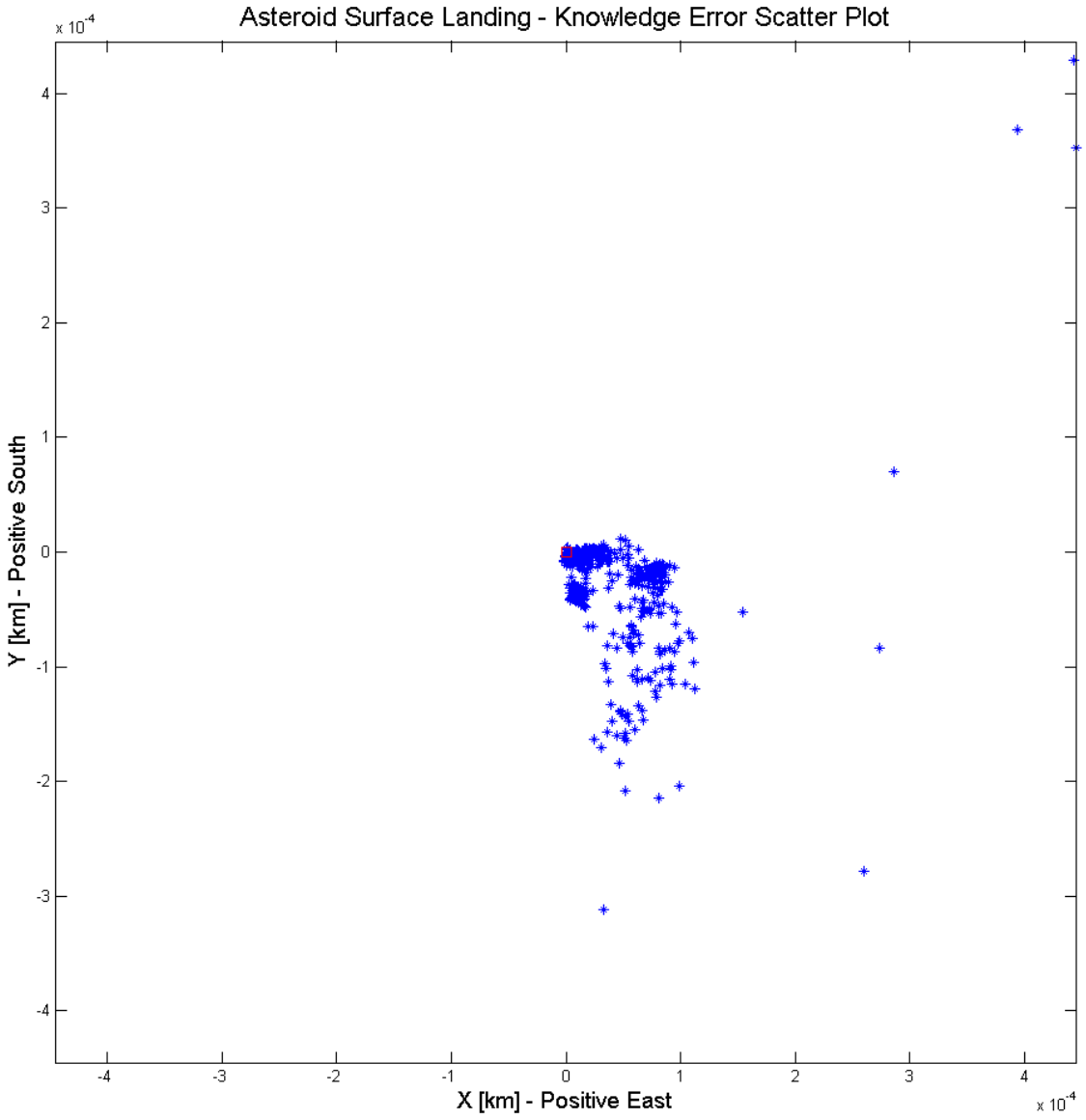

Figure 5.10: Knowledge Error Scatter Plot - 50% maneuver execution errors

While the knowledge error plot in Figure 5.10 looks slightly different than the baseline plot, with a more narrow fan-like shape than the baseline, the scale is similarly

very small. The final positions for this 50% maneuver execution error scenario are shown in Figure 5.11 below.

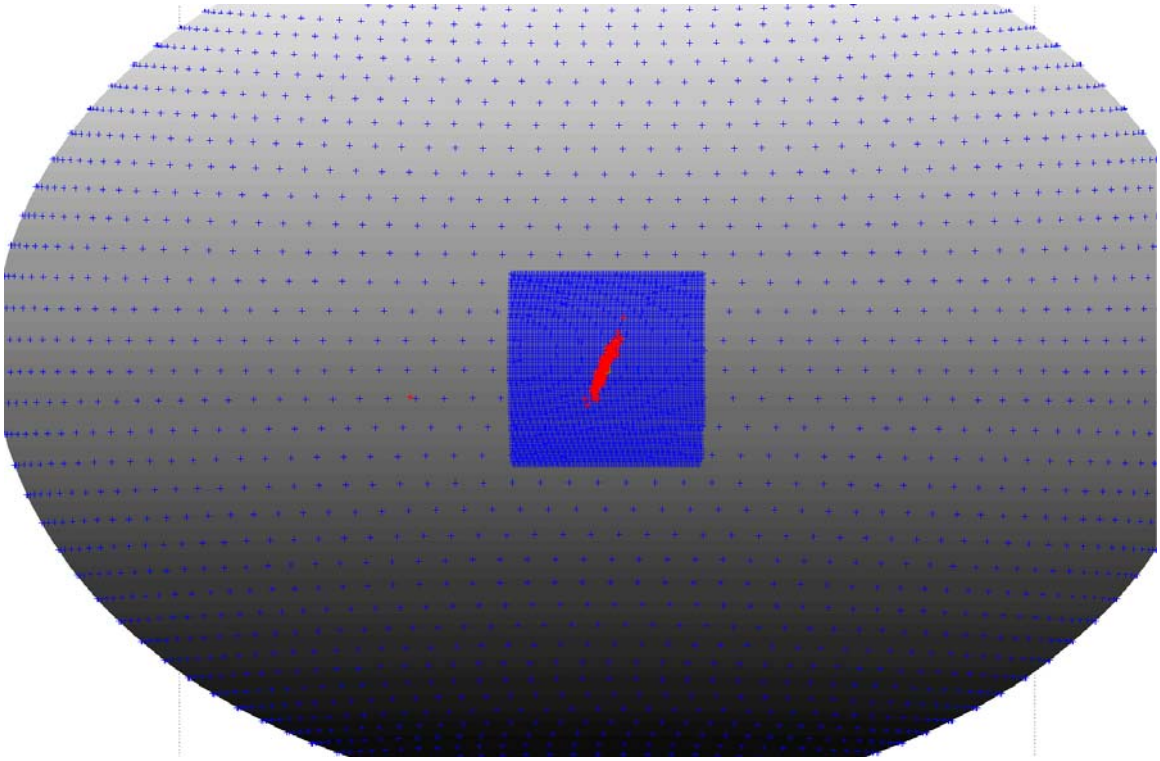

Figure 5.11: Final Positions Scatter Plot - 50% maneuver execution errors

5.3 200% MANEUVER EXECUTION ERRORS

The maneuver execution errors are increased by 200% from the baseline. See Section 5.2 for the values that are increased by 200%.

Table 5.3: Monte Carlo Statistics - 200% maneuver execution errors

Number of misses out of 500 cases	34 (6.8%)
Average target error magnitude	57.89 meters
Target error magnitude standard deviation	66.73 meters

The number of misses in Table 5.3 is almost exactly the same as the baseline scenario. The average target error magnitude increased dramatically, just as in the 50% case it decreased dramatically. These large changes in average target error magnitude suggest that the average target error is highly sensitive to the maneuver execution error.

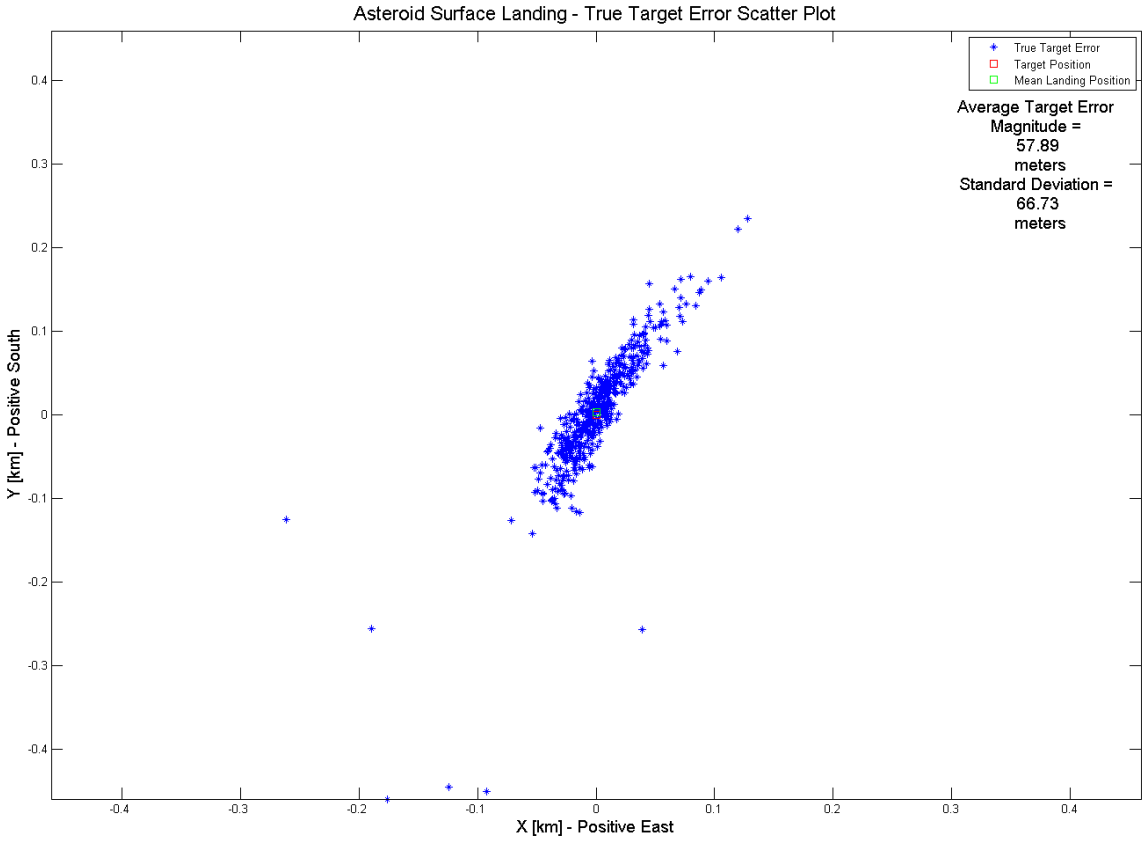

Figure 5.12: True Target Error Scatter Plot - 200% maneuver execution errors

The size of the landing footprint in Figure 5.12 increases dramatically over the baseline scenario, just as it decreased dramatically for 50% maneuver execution errors. As with the 50% maneuver execution errors, the outliers remained much the same.

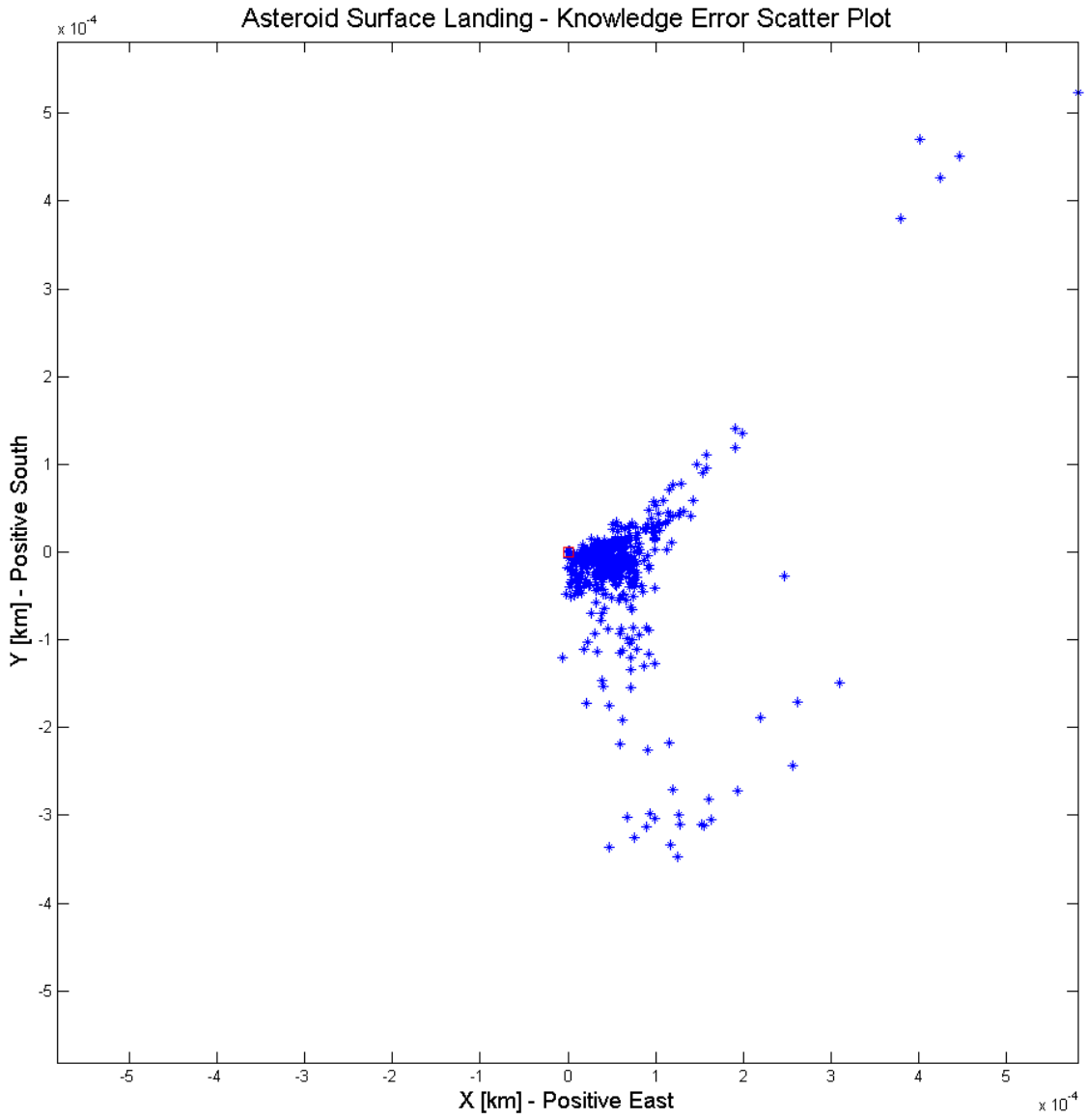

Figure 5.13: Knowledge Error Scatter Plot - 200% maneuver execution errors

The knowledge error plot in Figure 5.13 is closer in shape to the baseline scenario than the 50% maneuver execution errors case, but the edges are not as densely packed with landmarks, and there are more points further away from the origin. The scale is the same as the baseline case, in the sub-meter range.

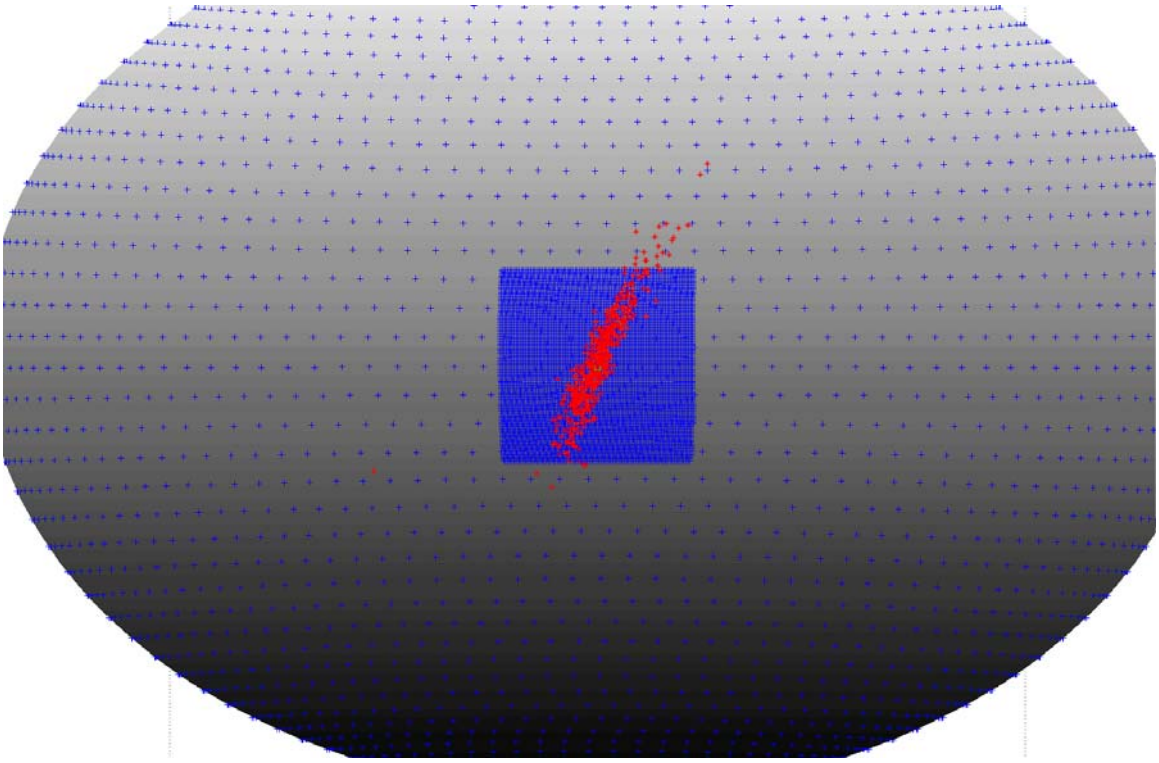

Figure 5.14: Final Positions Scatter Plot - 200% maneuver execution errors

5.4 50% OBSERVATION ERRORS

The observation errors are decreased by 50% from the baseline. Thus the pixel and line error multiplier values are reduced from 0.25 pixels to 0.125 pixels.

Table 5.4: Monte Carlo Statistics – 50% observation errors

Number of misses out of 500 cases	31 (6.2%)
Average target error magnitude	31.72 meters
Target error magnitude standard deviation	65.88 meters

Again the number of misses in Table 5.4 is almost exactly the same as the baseline scenario. While the average target error magnitude is almost identical to the baseline value, the standard deviation increased over the baseline, which is the opposite

of what was expected with a decrease in observation error. This unexpected change in the standard deviation is conjectured to be caused by the slightly higher number of cases that land on the asteroid, as they are still far from the target, thus increasing the standard deviation.

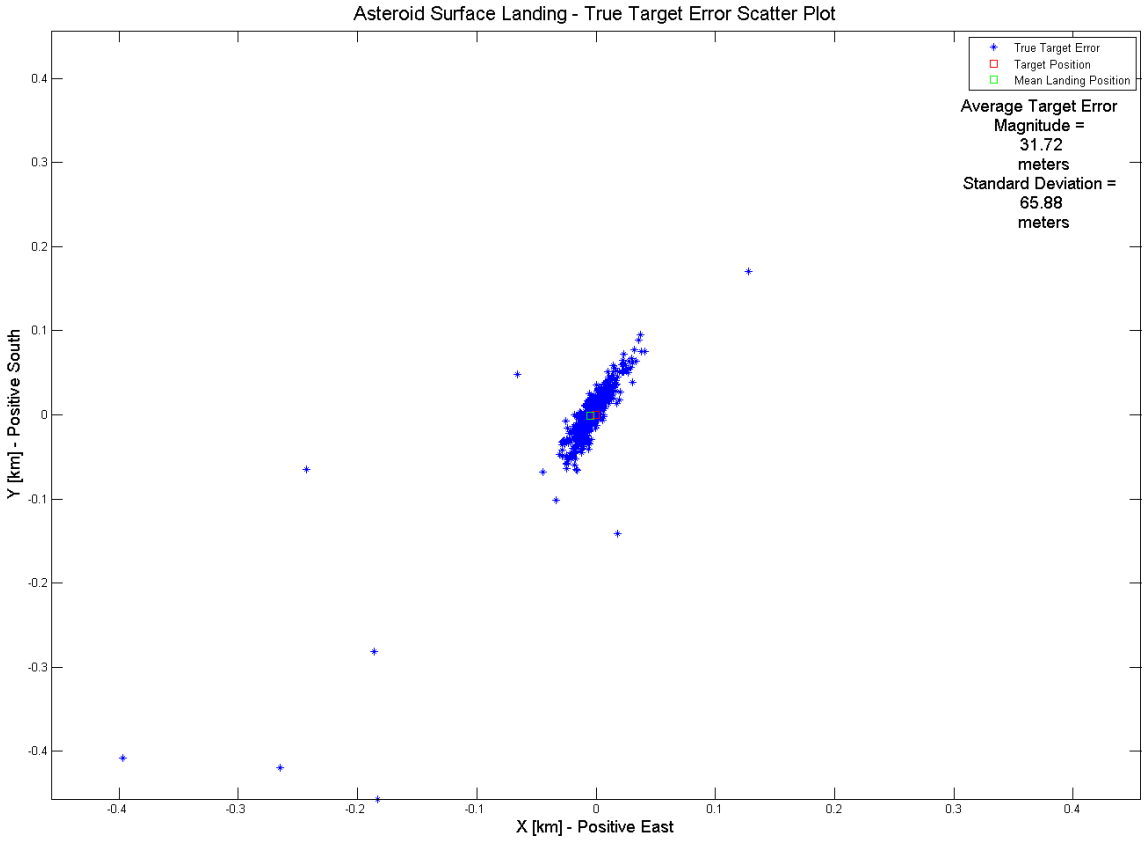

Figure 5.15: True Target Error Scatter Plot - 50% observation errors

The true target errors plot in Figure 5.15 appears only slightly different from the baseline plot. While a few of the outliers are different, the footprint for the most part is similar is shape and size to the baseline. This similarity suggests that while the observation error does have some affect on the outliers and non-landings, it is minimal and not the primary source of these relatively larger errors. The plot also suggests that the observation errors do not have a significant affect on the size and shape of the

footprint. The angular region that contains the scatter points in Figure 5.16 below is slightly narrower than in the baseline knowledge error scatter plot, though the main features and scale are the same.

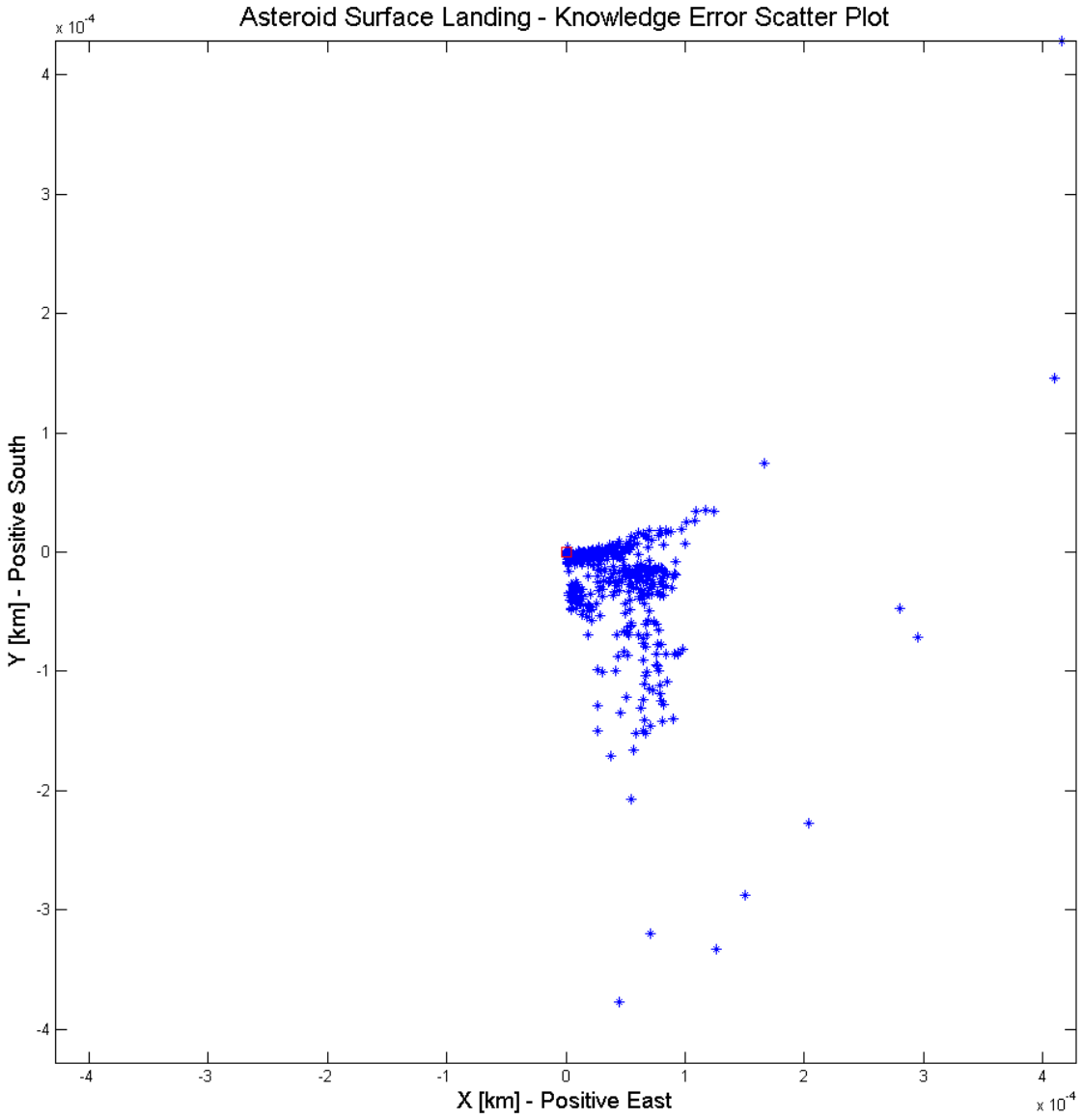

Figure 5.16: Knowledge Error Scatter Plot - 50% observation errors

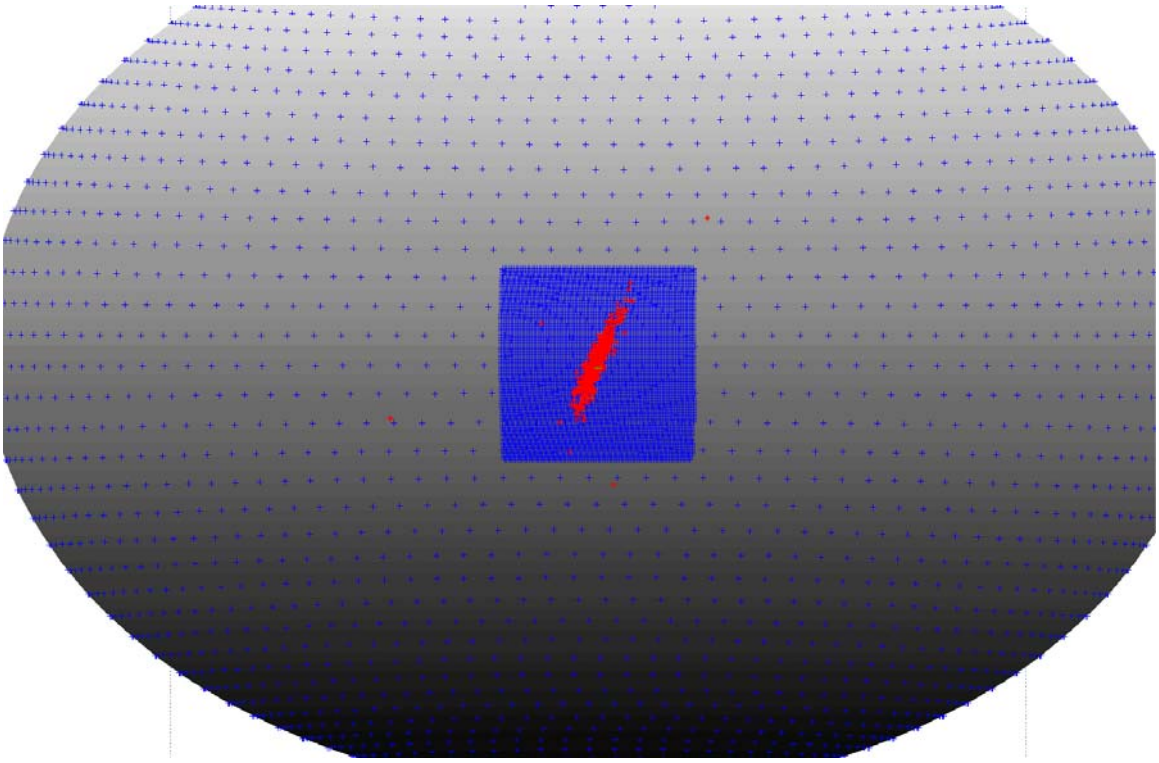

Figure 5.17: Final Positions Plot - 50% observation errors

5.5 200% OBSERVATION ERRORS

The observation errors are increased by 200% from the baseline. Thus the pixel and line error multiplier values are increased from 0.25 pixels to 0.5 pixels.

Table 5.5: Monte Carlo Statistics – 200% observation errors

Number of misses out of 500 cases	32 (6.4%)
Average target error magnitude	35.39 meters
Target error magnitude standard deviation	77.47 meters

The number of misses in Table 5.5 is almost exactly the same as the baseline scenario. The average target error magnitude increased slightly over the baseline value, which when paired with the even smaller change in the 50% observation error case,

further confirms the lack of sensitivity in the average target error magnitude to the observation error. Unlike in the 50% observation error case, the change in the target error magnitude standard deviation was expected: a significant increase in the standard deviation as a result of the increase in observation error. This increase is likely due to larger observation errors, which act to disburse those landing positions which are within the primary footprint away from the target in a relatively consistent fashion. This dispersion results in a greater standard deviation even though the average target error magnitude does not increase by much.

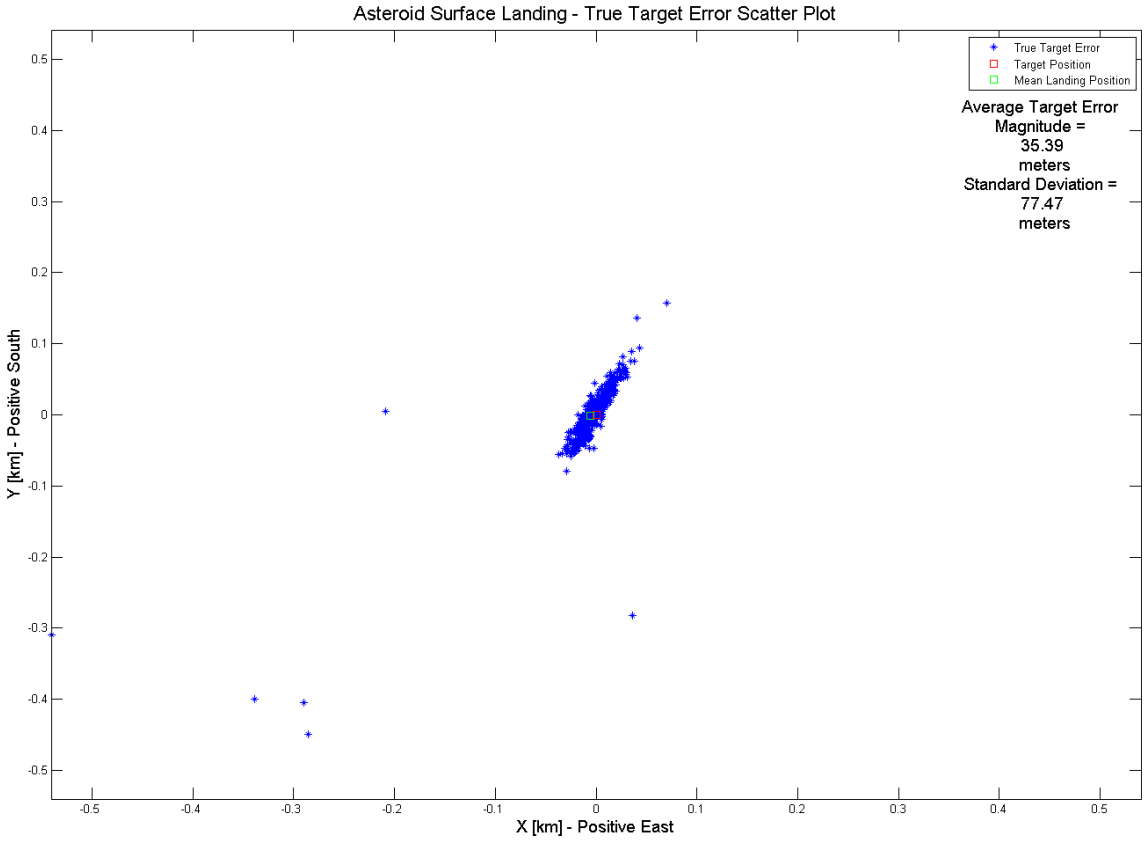

Figure 5.18: True Target Error Scatter Plot - 200% observation errors

The footprint in Figure 5.18 appears to be slightly smaller than baseline. However, the increase in average target error magnitude and standard deviation suggests

that the density at the center of the footprint is actually decreasing, with larger numbers of points moving to the edges of the footprint.

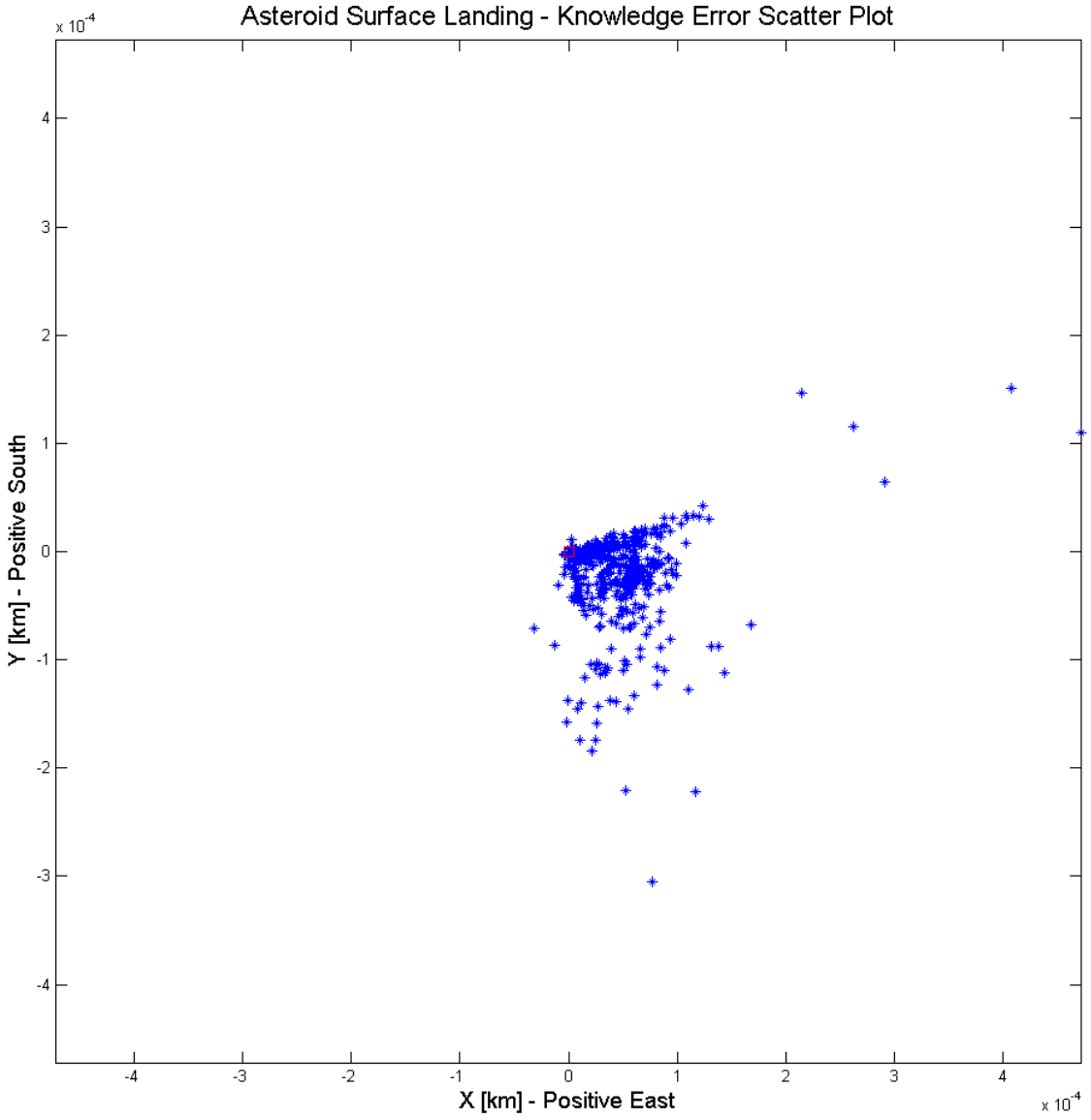

Figure 5.19: Knowledge Error Scatter Plot - 200% observation errors

The angular area of the knowledge error points in Figure 5.19 is almost identical to the baseline scenario, although the number of points outside of this area is slightly greater. The density of points is also slightly greater further away from the origin.

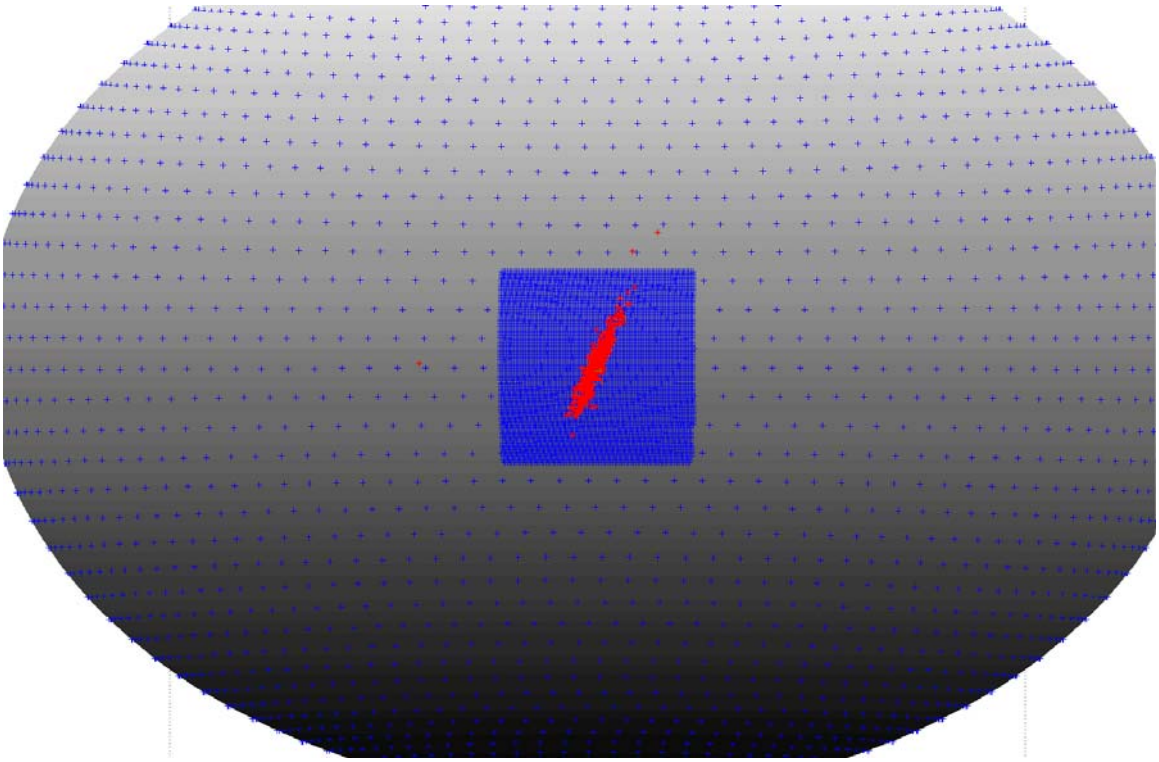

Figure 5.20: Final Positions Plot - 200% observation errors

5.6 50% INITIAL STATE ERRORS

The initial state errors are decreased by 50% from the baseline. Thus the initial position error multipliers for each axis are decreased from 2.5 meters to 1.25 meters, and the initial velocity error multipliers for each axis are decreased from 2.5 millimeters per second to 1.25 millimeters per second.

Table 5.6: Monte Carlo Statistics – 50% initial state errors

Number of misses out of 500 cases	3 (0.6%)
Average target error magnitude	27.51 meters
Target error magnitude standard deviation	21.16 meters

The number of misses in Table 5.6 is dramatically lower than the baseline scenario value of 33. By reducing the initial state error by just 50%, the outliers are almost eliminated. This suggests that the initial state error is the most likely source of the outliers. The average target error magnitude also drops compared to the baseline, though not nearly as precipitously as the number of misses and outliers, suggesting that the landing points within the primary landing footprint are not nearly as sensitive to the initial state error as the outliers.

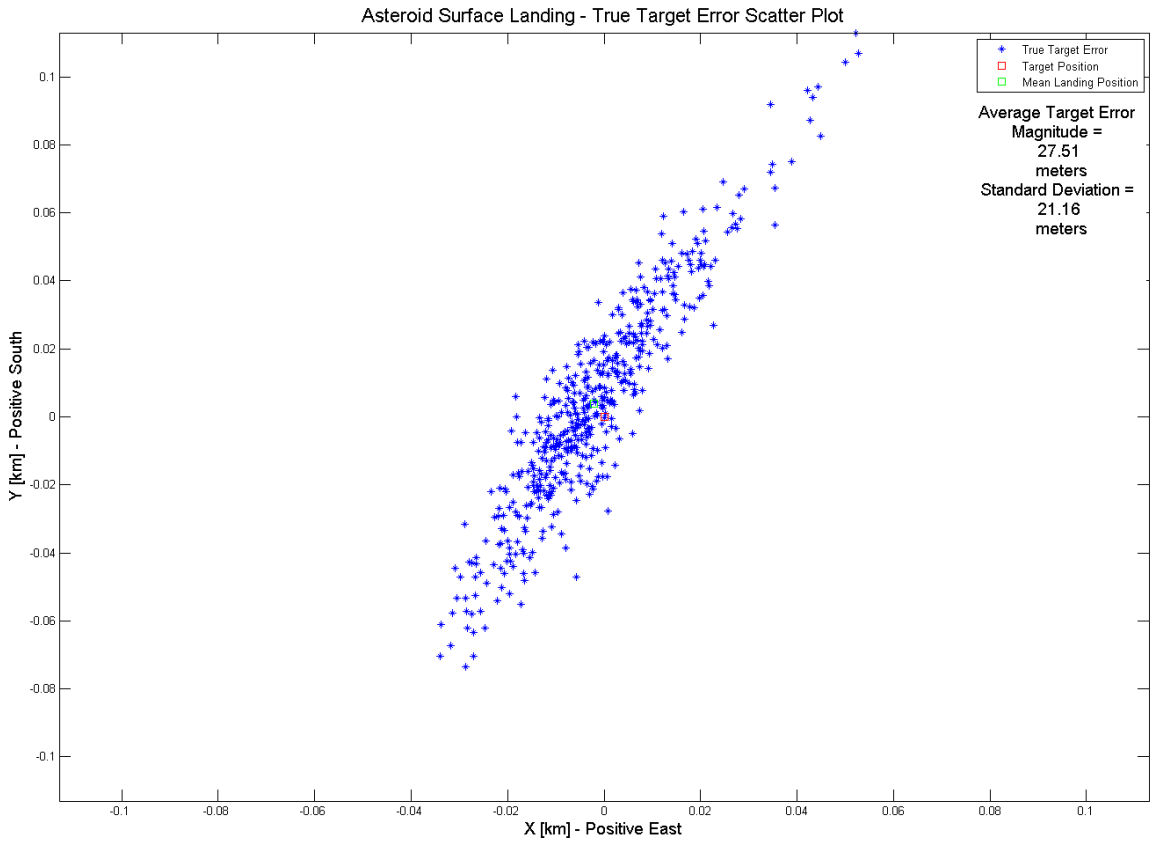

Figure 5.21: True Target Error Scatter Plot - 50% initial state errors

The landing footprint fills a significantly greater portion of Figure 5.21 compared to the baseline due to the total lack of large outliers present. Thus, because the scale is so much smaller than the baseline plot of the true target error scatter, the final positions plots

in Figure 5.23 and Figure 5.8 must be studied to obtain a direct comparison in the size of the footprints.

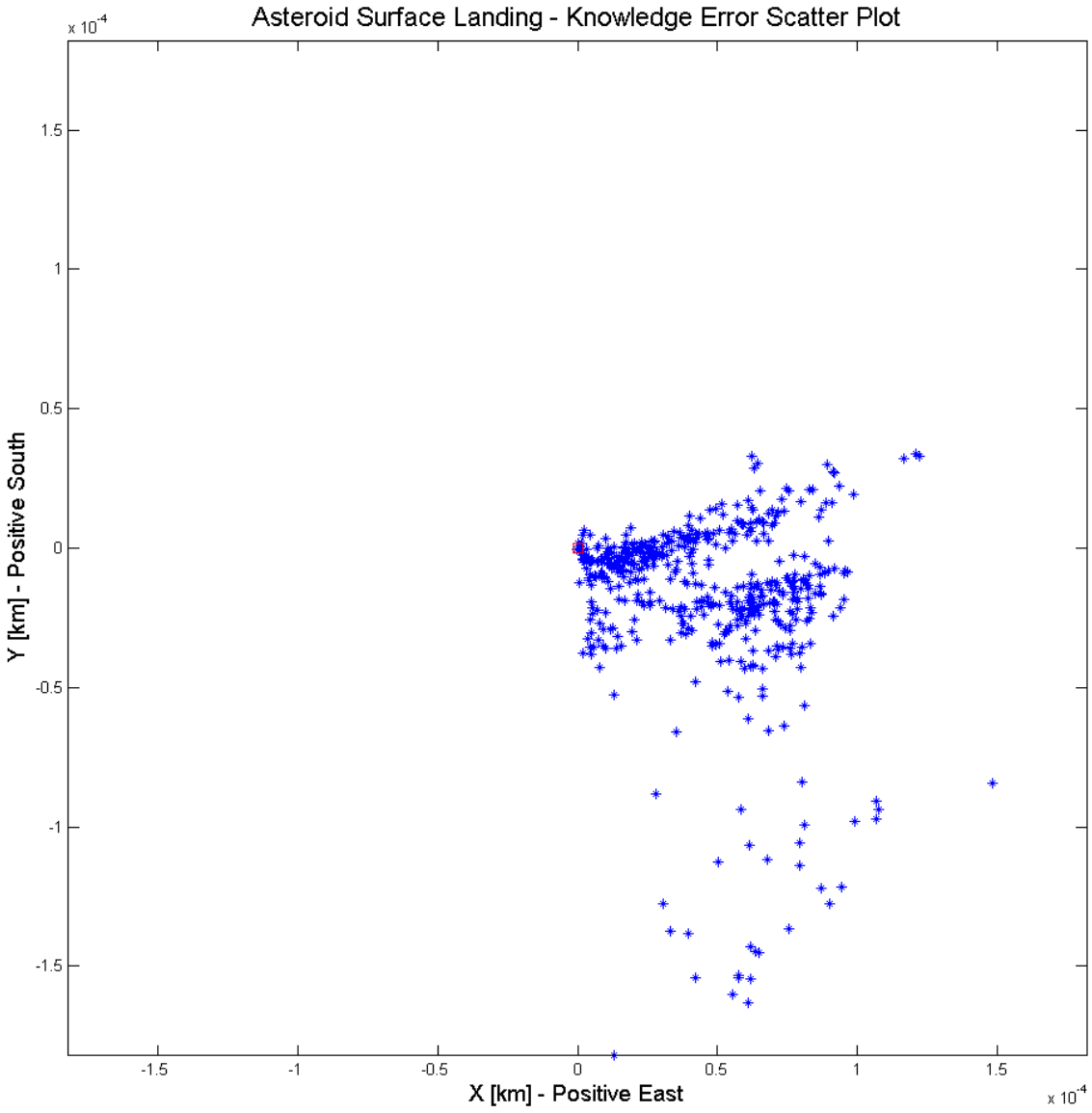

Figure 5.22: Knowledge Error Scatter Plot - 50% initial state errors

The knowledge error plot in Figure 5.22 is significantly smaller than the baseline plot, with many less outlying points and the majority of points appearing closer to the

origin. While the size is smaller, the shape of the plot appears to remain consistent with the baseline.

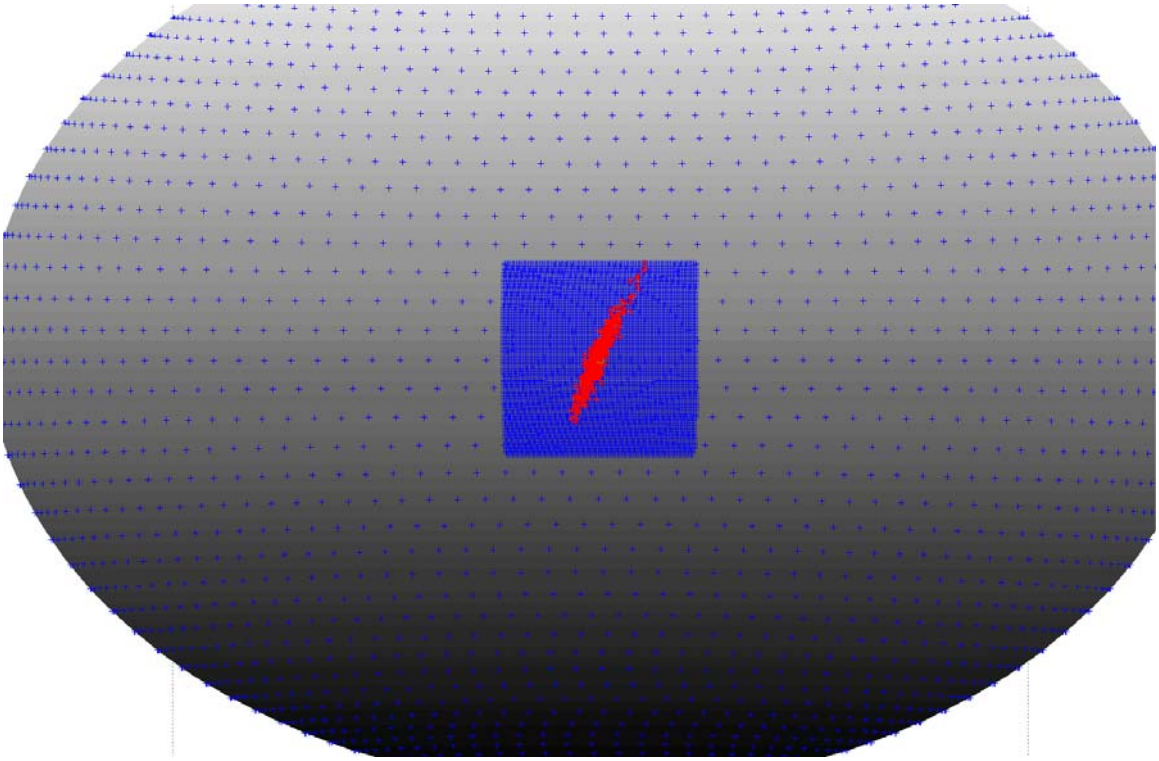

Figure 5.23: Final Positions Plot - 50% initial state errors

Comparing the landing footprint in Figure 5.23 against the baseline, it appears to be approximately the same size as the baseline footprint. The only visible differences are the lack of outliers outside the region of denser landmarks and the footprint appears to be slightly thinner further away from the target than in the baseline, giving it a more elliptical appearance over the rectangular shape of the baseline footprint.

5.7 200% INITIAL STATE ERRORS

The initial state errors are increased by 200% from the baseline. Thus the initial position error multipliers for each axis are increased from 2.5 meters to 5 meters, and the

initial velocity error multipliers for each axis are increased from 2.5 millimeters per second to 5 millimeters per second.

Table 5.7: Monte Carlo Statistics – 200% initial state errors

Number of misses out of 500 cases	77 (15.4%)
Average target error magnitude	73.76 meters
Target error magnitude standard deviation	198.78 meters

The number of misses in Table 5.7 is considerably greater than in the baseline case, with an increase of more than 233% over the baseline case. This increase along with the similarly steep decrease in outliers in the 50% initial error case strongly suggests that the outliers are highly sensitive to initial state errors. Both the average target error magnitude and standard deviation also increased markedly, suggesting there is a reasonable sensitivity in these values to the initial state error.

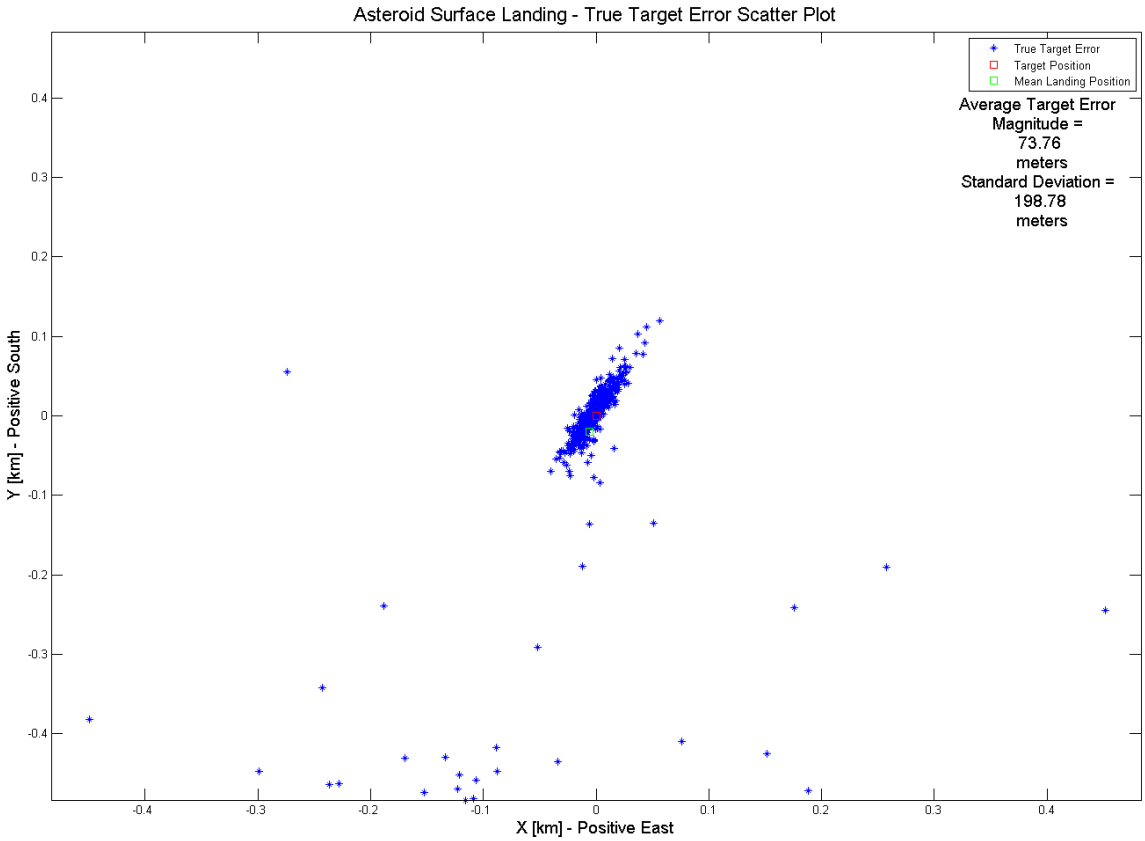

Figure 5.24: True Target Error Scatter Plot - 200% initial state errors

The primary difference between Figure 5.24 and the baseline plot is the large increase in outliers visible. The scale, however, is the same as the baseline plot. The landing footprint also appears thinner, although this is likely due to the large number of landings that shifted from landing at the edges of the primary footprint to being outliers.

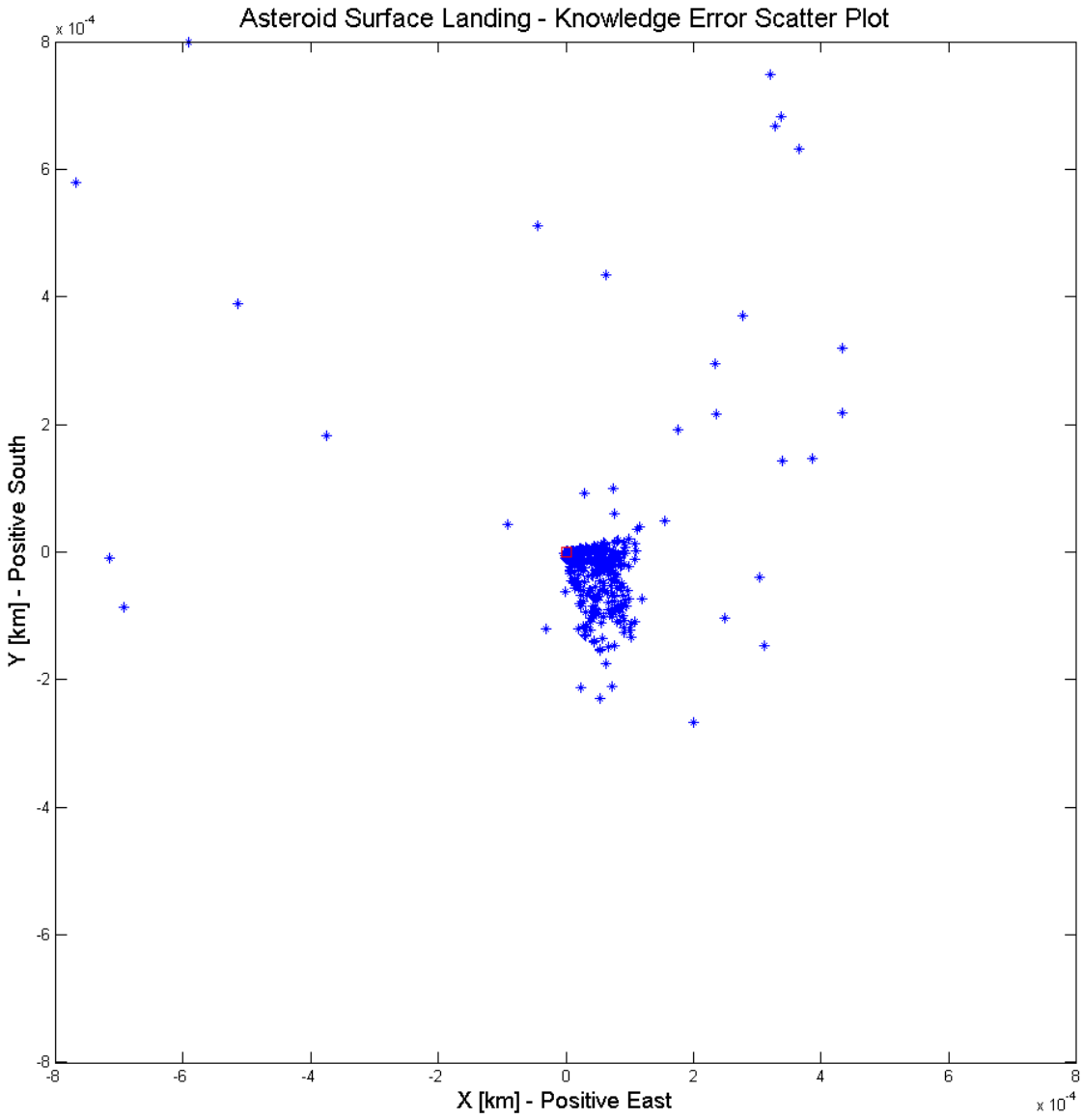

Figure 5.25: Knowledge Error Scatter Plot - 200% initial state errors

As in Figure 5.24, the knowledge error scatter plot in Figure 5.25 shows a large increase in the number of outliers over the baseline case, and for the first time a significant number of outliers now appear in the top of the plot above the origin. The general shape of the footprint, however, remains the same fan-like shape as the baseline.

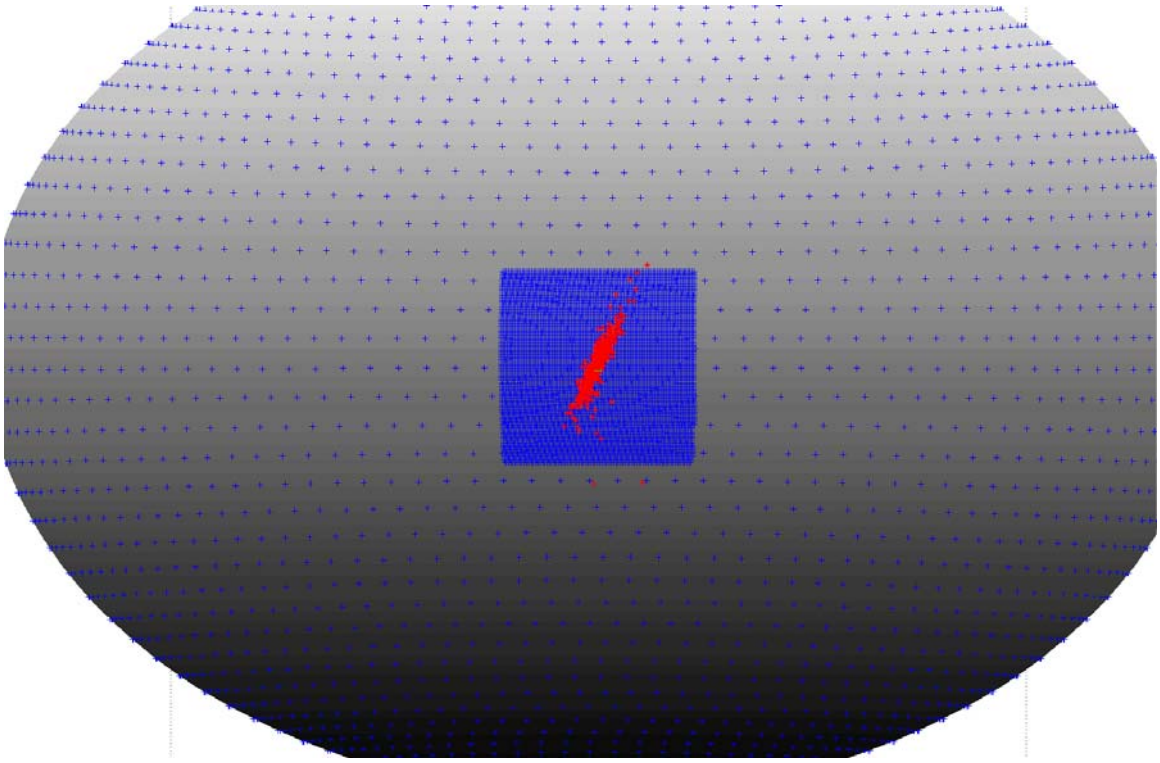

Figure 5.26: Landing Positions Plot - 200% initial state errors

5.8 NO ATTITUDE ERROR

The attitude error in the simulation is completely turned off in the initialization file for the analysis in this section.

Table 5.8: Monte Carlo Statistics – No Attitude Error

Number of misses out of 500 cases	33 (6.6%)
Average target error magnitude	32.02 meters
Target error magnitude standard deviation	58.39 meters

The number of misses in Table 5.8 is identical to the baseline scenario, just as the average target error magnitude and standard are almost the same as the baseline values. This similarity indicates that the simulation as a whole has very little sensitivity to

attitude error. This low sensitivity is thought to be due to the Kinematic Filter accurately estimating the attitude for every position fix. If the attitude was not being estimated, as it was not in previous versions of SANT, the attitude error would likely play a more prominent role.

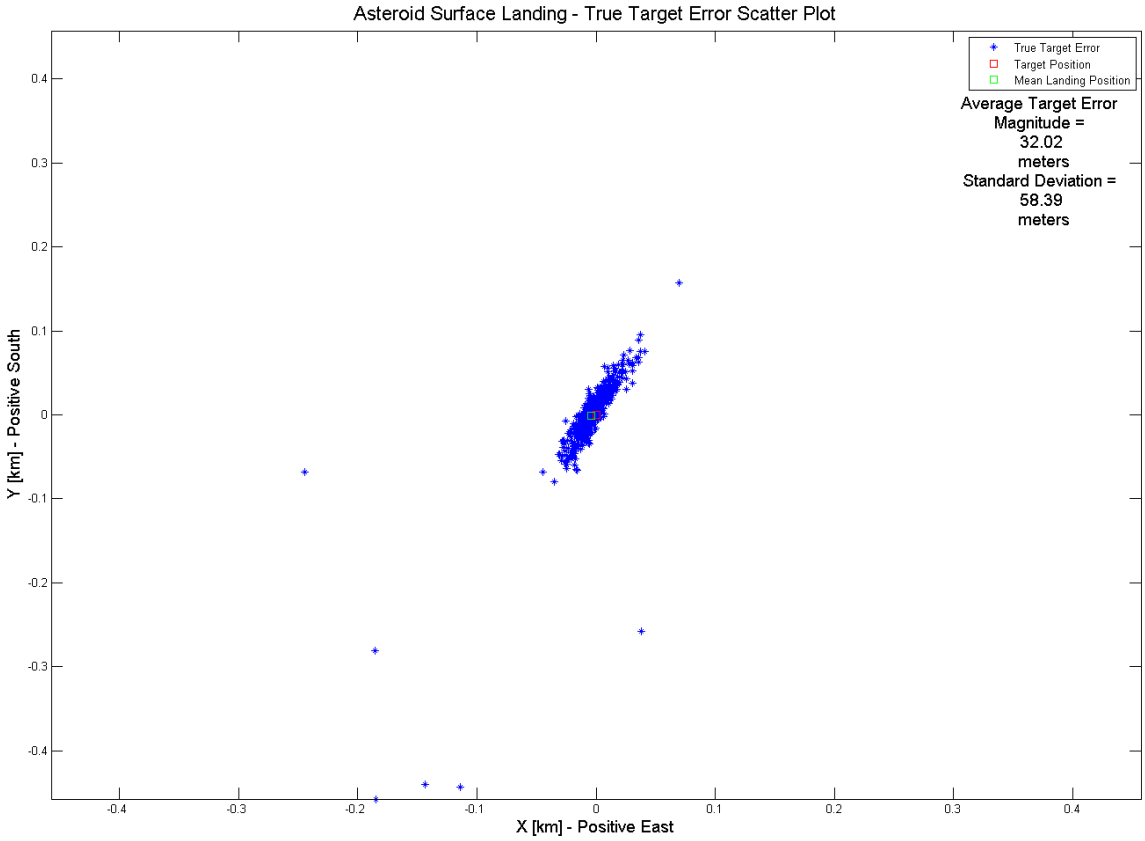

Figure 5.27: True Target Error Scatter Plot – No Attitude Errors

The true target error scatter plot in Figure 5.27 is almost identical to the baseline plot. This similarity reinforces the conclusion drawn above concerning the lack of sensitivity to attitude error in the simulation.

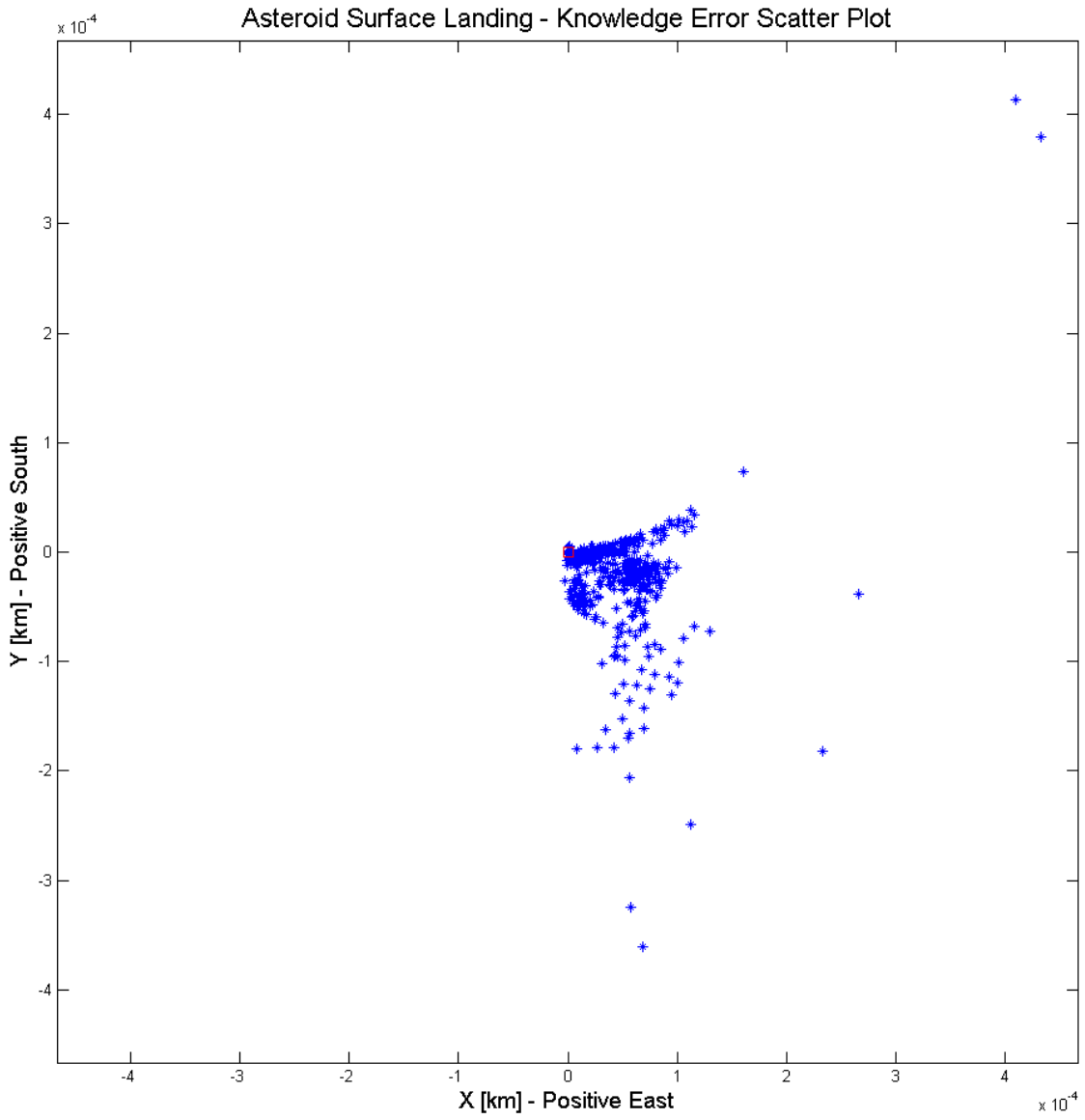

Figure 5.28: Knowledge Error Scatter Plot – No Attitude Errors

Although there is a more significant difference in the knowledge error scatter plot of Figure 5.28 compared to the baseline than for the true target error scatter plot, the difference is still very small. Some of the outliers appear to be in slightly different positions, and a small portion of the main footprint is different than the baseline plot.

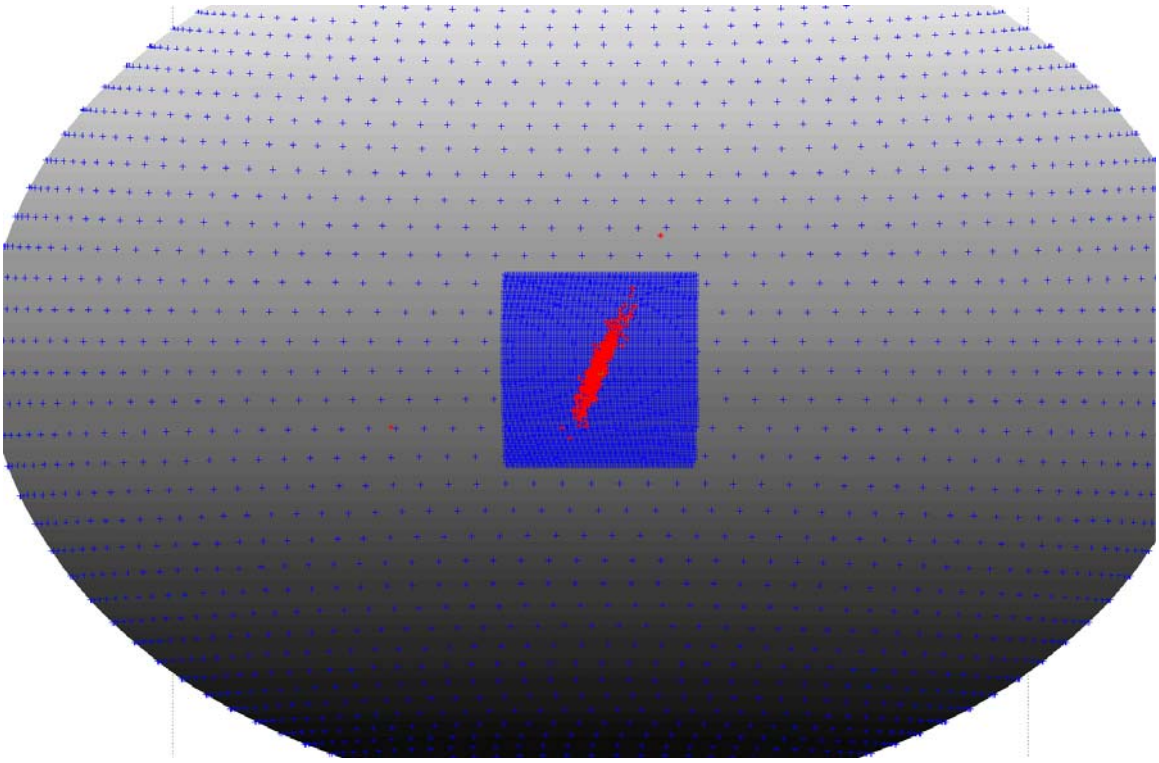

Figure 5.29: Final Positions Plot – No Attitude Errors

5.9 ADDING ALL THREE PERTURBATIONS TO NOMINAL MODEL

In this section, all three of the perturbations as defined in Sections 2.1.2 and 3.9.4, which include the SRP perturbations, full gravitational harmonics perturbations, and third body gravitational perturbations, were added to the nominal model. In the baseline case, none of these perturbations were in the nominal model.

Table 5.9: Monte Carlo Statistics – All 3 Perturbations on Nominal Model

Number of misses out of 500 cases	26 (5.2%)
Average target error magnitude	40.01 meters
Target error magnitude standard deviation	97.17 meters

The number of misses in Table 5.9 is seven less than the baseline case, which is a significant, if not dramatic, reduction in the number of outliers for the Monte Carlo simulation. It appears that by having better knowledge of the forces acting on the spacecraft, the estimated state more closely reflects the truth state and thus the maneuvers to land on the surface are more accurate.

The average target error magnitude and target error magnitude standard deviation are both greater than the baseline value. This unexpected result, however, is likely due to the greater number of cases which are reaching the surface but are still outliers. These additional outliers result in higher average target error magnitude and target error magnitude standard deviation.

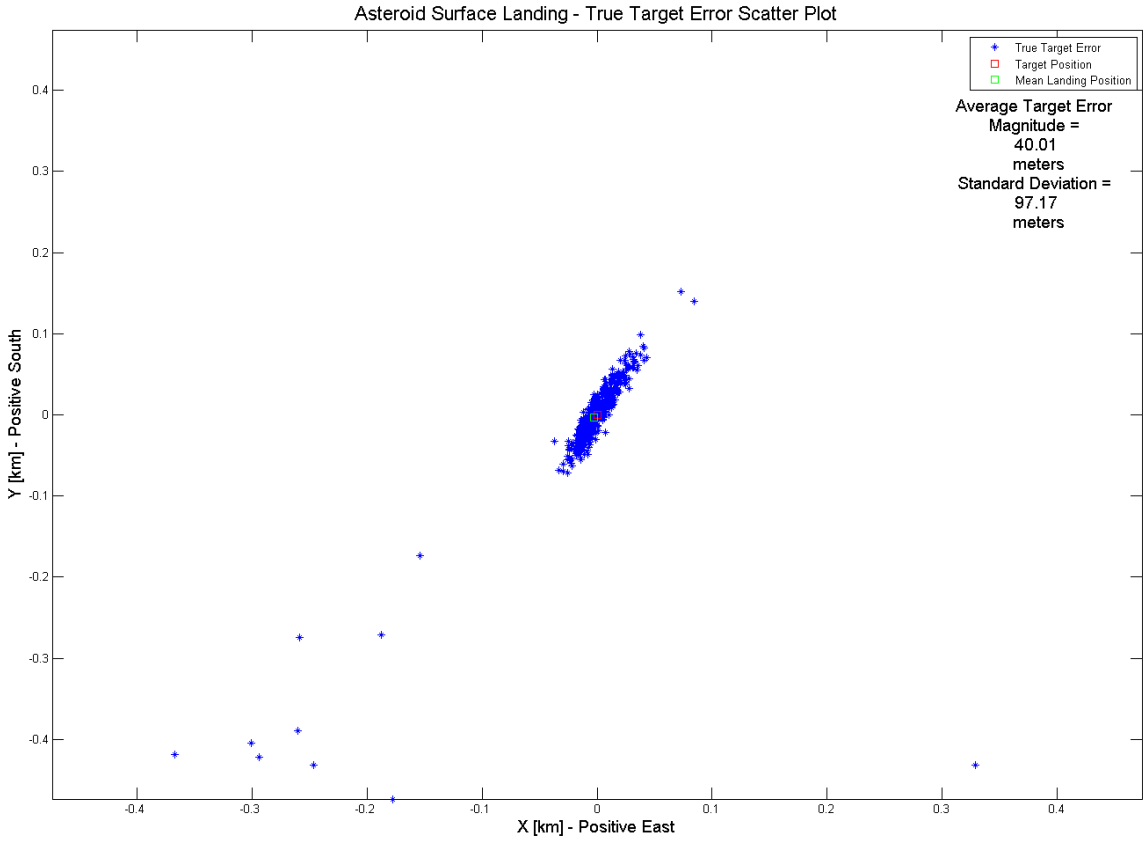

Figure 5.30: True Target Error Scatter Plot – All 3 Perturbations on Nominal Model

In Figure 5.30, at least five more outliers than the baseline plot contains can be seen, but otherwise the shape and size are similar to the baseline scenarios. The outliers seen in Figure 5.30 experienced larger initial state error, with those relatively closer to one another experiencing approximately the same amount of initial state error.

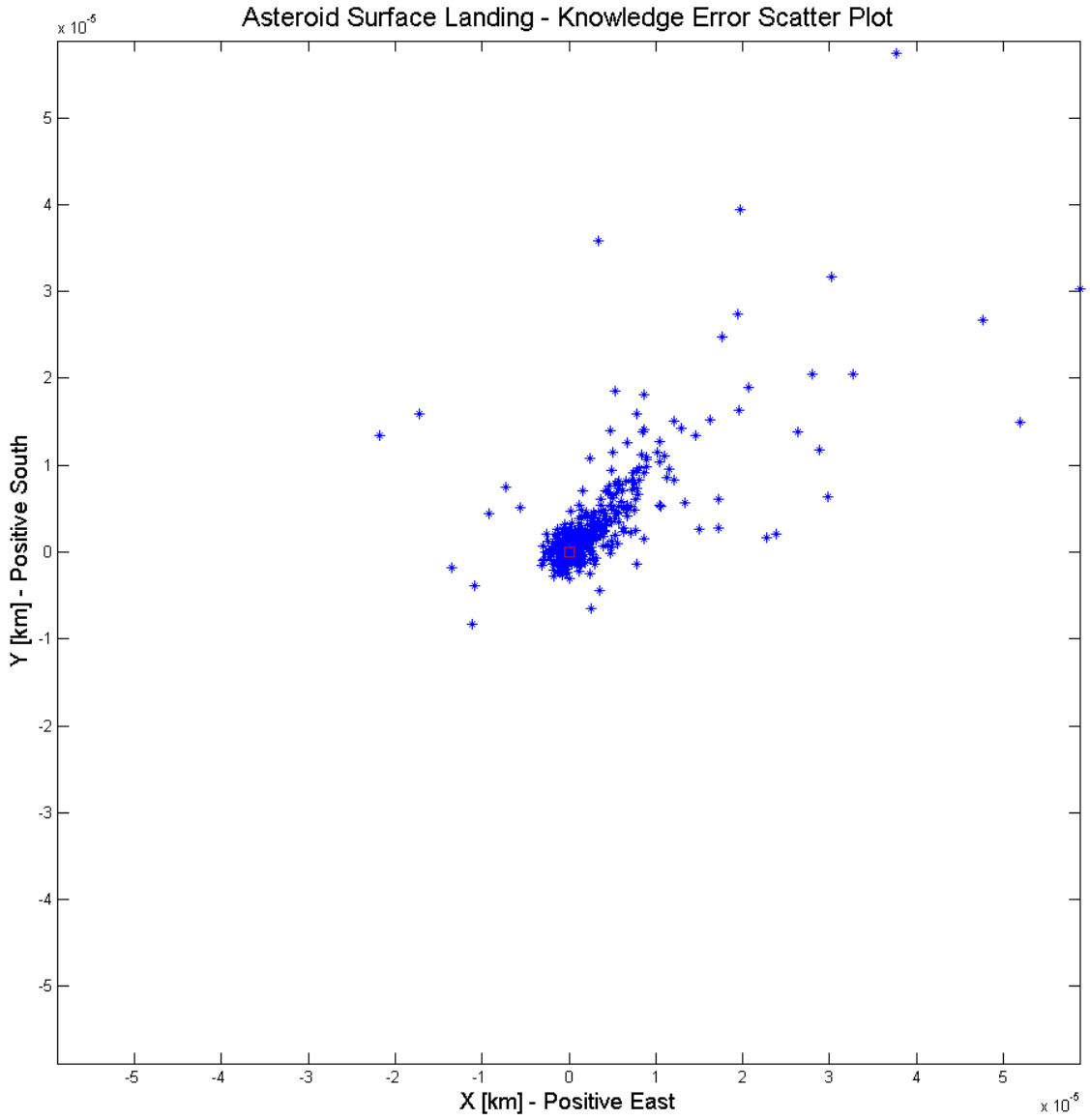

Figure 5.31: Knowledge Error Scatter Plot – All 3 Perturbations on Nominal Model

The knowledge error scatter plot in Figure 5.31 is different than the baseline in two significant ways. First, the scale of the plot has been reduced by an order of magnitude, from tens of centimeters to centimeters, indicating a significantly smaller level of knowledge error. Second, the scatter plot is now much more evenly distributed about the origin, implying a bias source was removed by adding the perturbations to the nominal model. There remains an apparent bias however, as there seem to be more points in the top half of the plot compared to the bottom half.

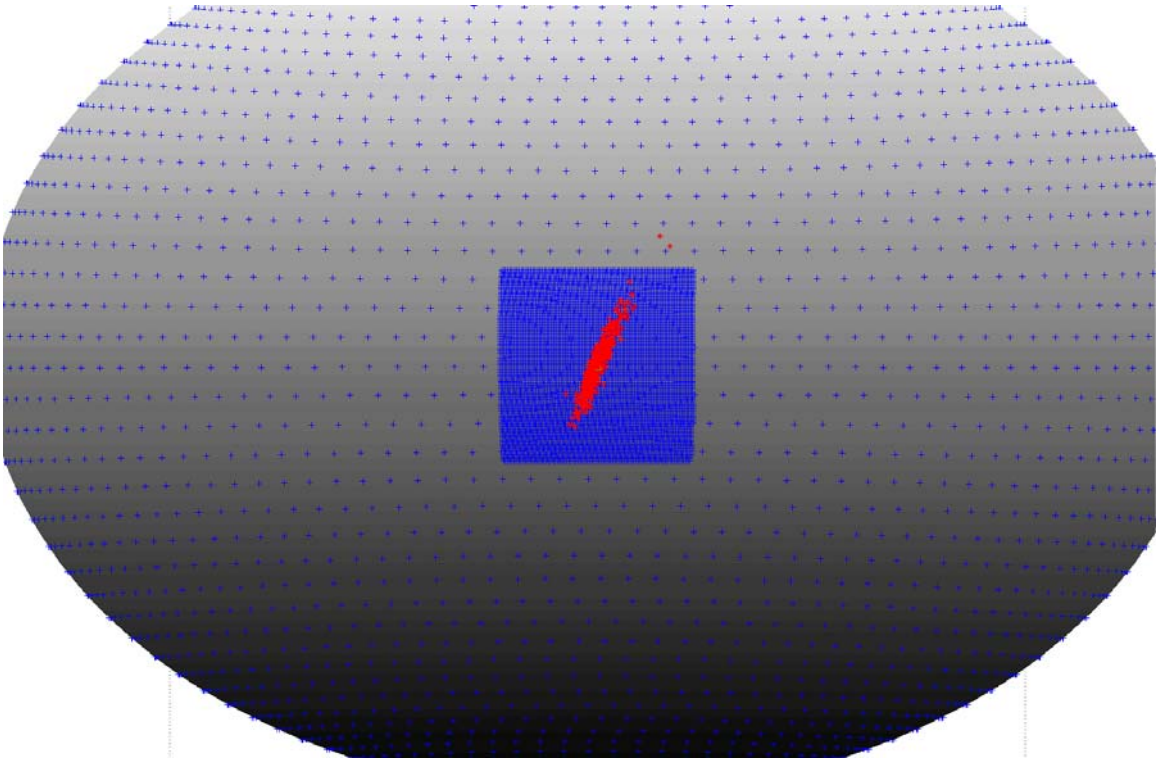

Figure 5.32: Final Positions Plot – All 3 Perturbations on Nominal Model

To better understand the change that occurred by adding all three perturbations to the nominal model, and in particular to see if any one perturbation is most responsible for the change, each perturbation is added individually and the results are detailed below.

5.10 ADDING ONLY GRAVITY HARMONICS PERTURBATION TO NOMINAL MODEL

The gravitational harmonics perturbation was added to the nominal model and the other two perturbations were turned off in the nominal model. In the baseline case none of these perturbations were in the nominal model.

Table 5.10: Monte Carlo Statistics – Only Gravity Harmonics Perturbations

Number of misses out of 500 cases	27 (5.4%)
Average target error magnitude	38.29 meters
Target error magnitude standard deviation	95.76 meters

The number of misses in Table 5.10 is almost identical to the case where all three perturbations were turned on in the nominal model. This similarity is also the case for the average target error magnitude and target error magnitude standard deviation. These results indicate that the addition of the gravity harmonics perturbations to the nominal model was most responsible for the improvement in performance seen in the case where all three perturbations were added to the nominal model. Further evidence for this hypothesis is seen in the true target error scatter plot in Figure 5.33 below, which is significantly closer in appearance to the all-three-perturbations-added scenario plot in Figure 5.30 than the same plot for the baseline scenario.

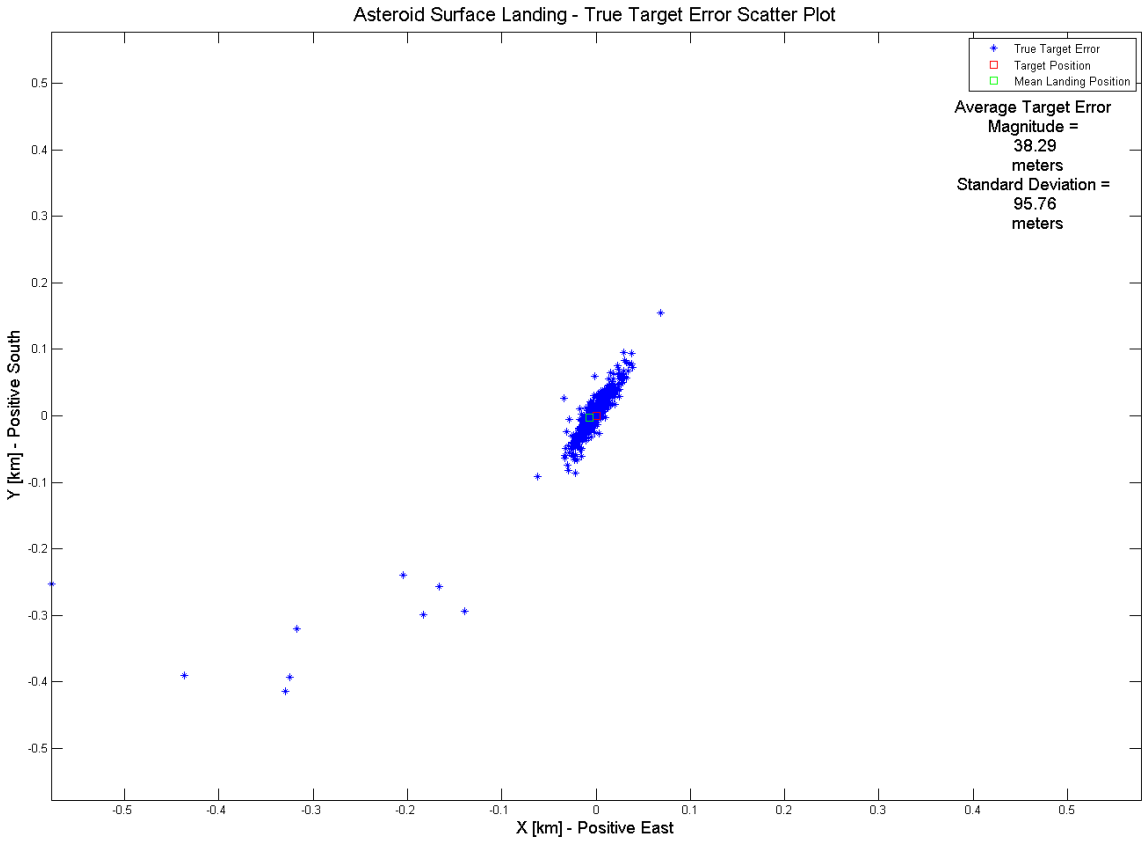

Figure 5.33: True Target Error Scatter Plot – Only Gravity Harmonics Perturbations on Nominal Model

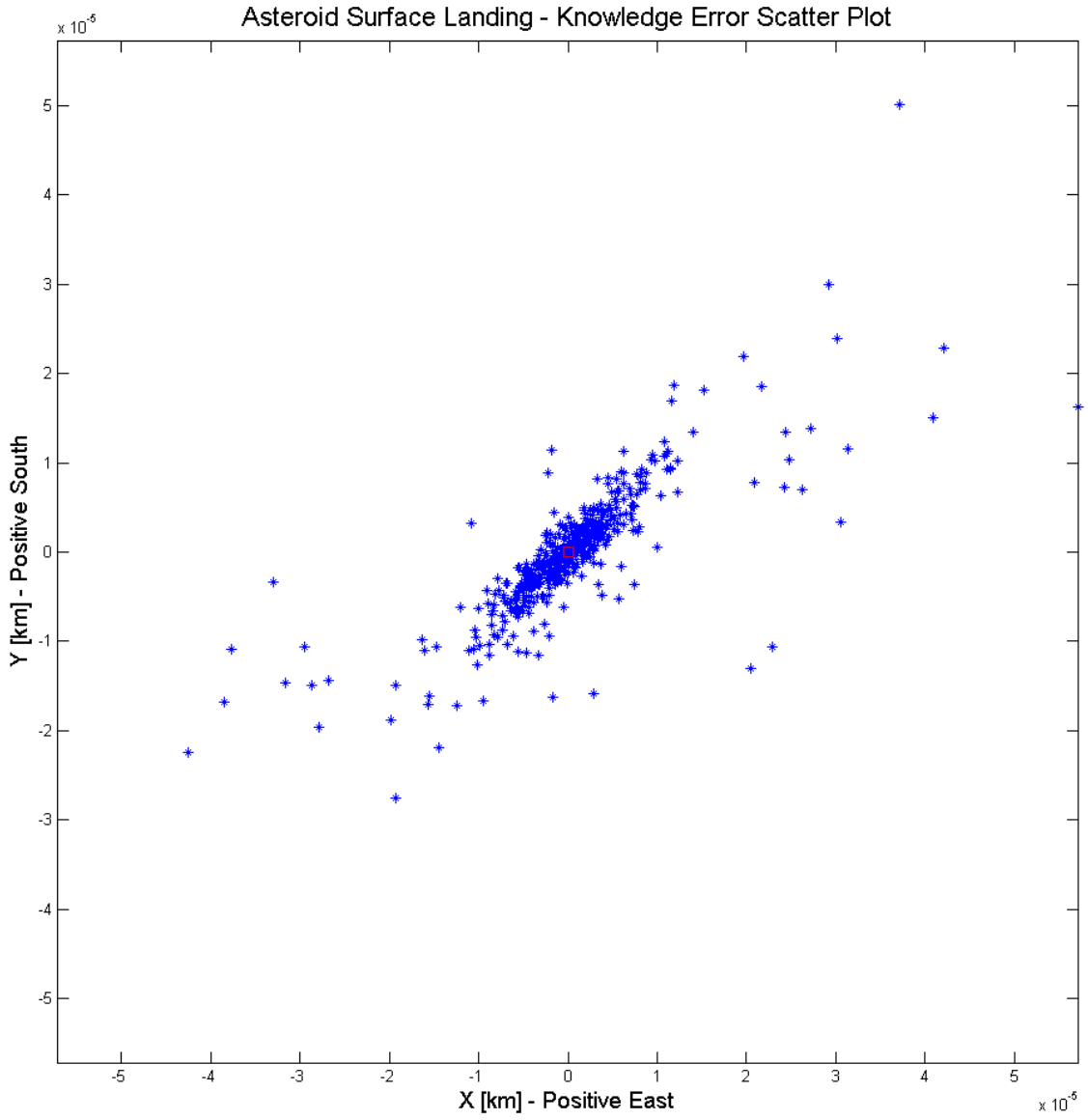

Figure 5.34: Knowledge Error Scatter Plot – Only Gravity Harmonics Perturbations on Nominal Model

The knowledge error scatter plot in Figure 5.34 is by far the least biased knowledge error plot seen to this point. This result, along with a bias seen in the next two sections that is similar to the baseline case, strongly suggests the lack of the gravity harmonics perturbations on the nominal model was the primary source of the bias in the

knowledge error scatter plot of the baseline case. The scale, like in the all-three-perturbations-added case from Section 5.9, is an order of magnitude smaller than the baseline scenario. Thus the lack of gravity harmonics perturbations on the nominal model appears to be the primary factor behind the order of magnitude increase in knowledge errors. All of the above results indicate that the addition of the gravity harmonics perturbations to the nominal model was by far most responsible for the improvement in knowledge error.

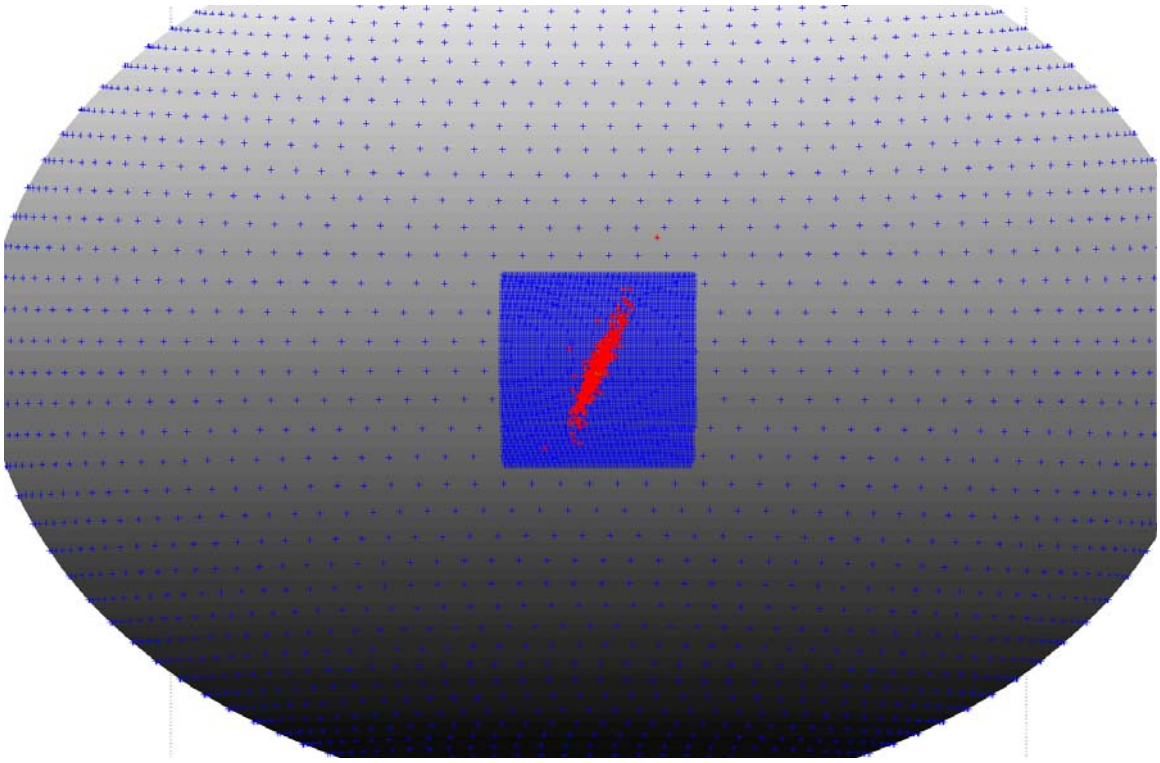

Figure 5.35: Final Positions Plot – Only Gravity Harmonics Perturbations on Nominal Model

5.11 ADDING ONLY SOLAR RADIATION PRESSURE PERTURBATION TO NOMINAL MODEL

The SRP perturbation was added to the nominal model and the other two perturbations were turned off in the nominal model. In the baseline case none of these perturbations were in the nominal model.

Table 5.11: Monte Carlo Statistics – Only SRP Perturbation

Number of misses out of 500 cases	35 (7%)
Average target error magnitude	37.00 meters
Target error magnitude standard deviation	92.21 meters

The number of misses in Table 5.11 is two more than the baseline case. And the average target error magnitude and the target error magnitude standard deviation are both much closer to the all-three-perturbations-added case than the baseline case. These results were the opposite of what was expected, as the adding of a perturbation to the nominal model that is acting on the truth model should make the results either better or at worst not less accurate than without the perturbation on the nominal model.

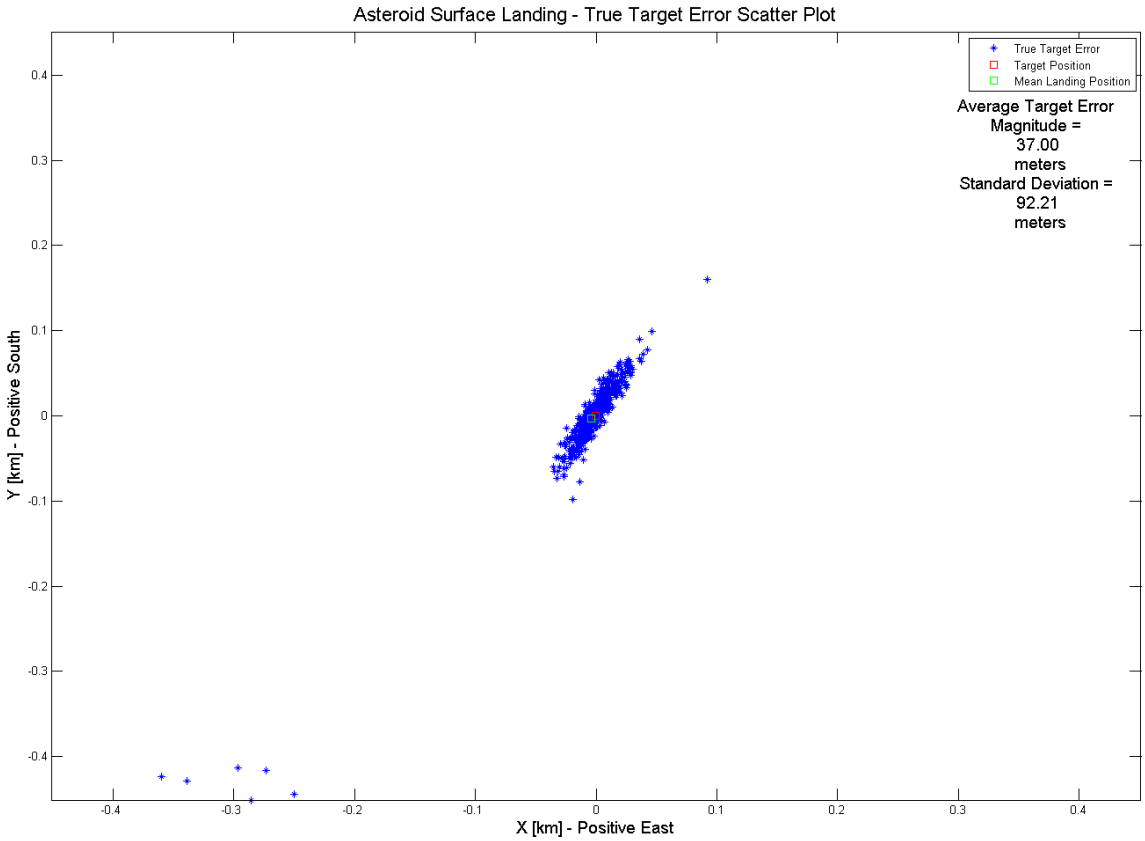

Figure 5.36: True Target Error Scatter Plot – Only SRP Perturbation on Nominal Model

The true target error scatter plot in Figure 5.36 features approximately seven outliers, just as in the baseline plot. Unlike in the baseline plot, however, all but one of the outliers are collected in a single small area that is further away from the target than any in the baseline plot. This fact, along with two less points being considered because two less cases reached the surface, served to increase the average target error magnitude and standard deviation. The outliers seen in Figure 5.36 experienced larger initial state error, with those relatively closer to one another experiencing approximately the same amount of initial state error.

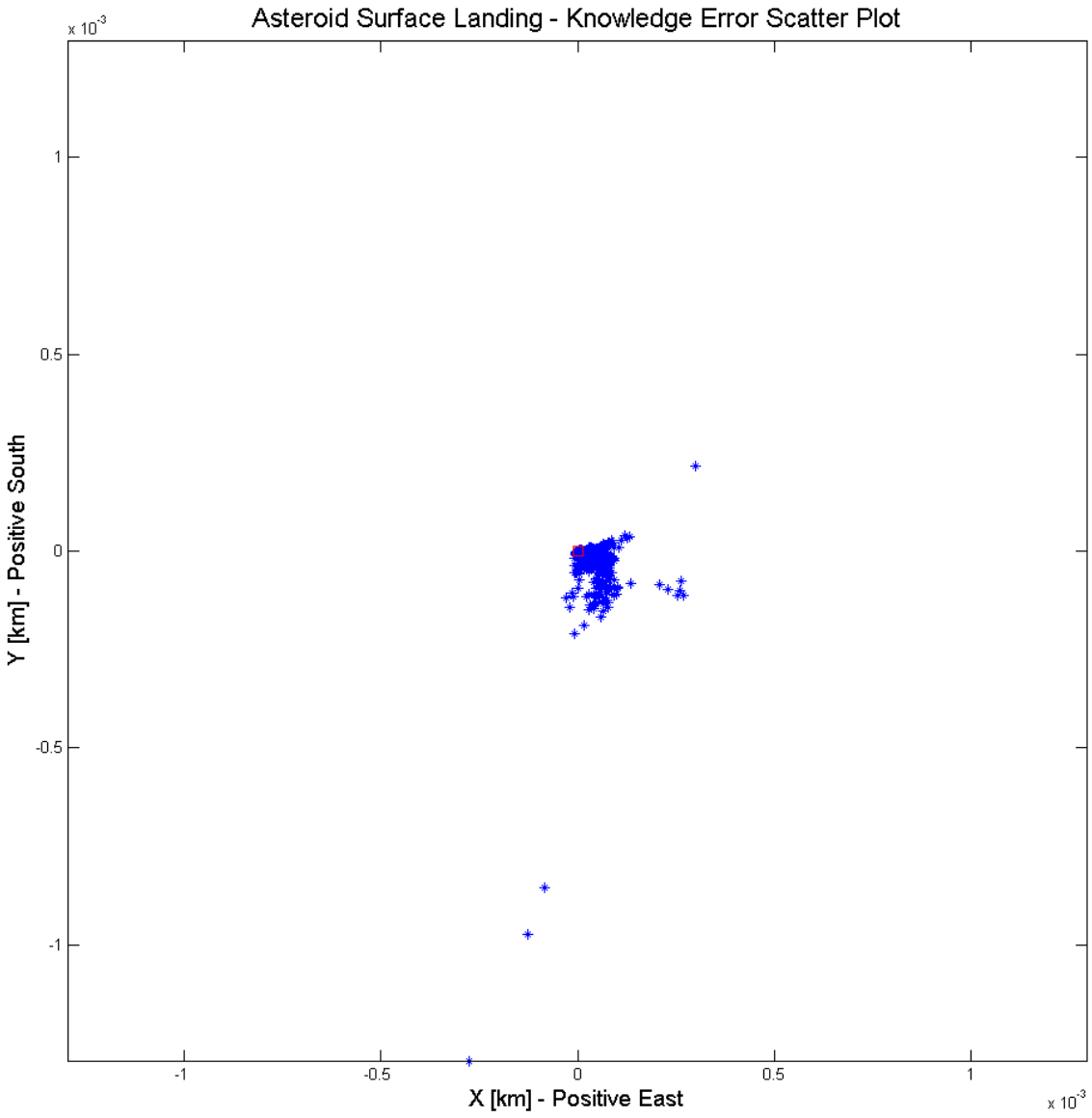

Figure 5.37: Knowledge Error Scatter Plot – Only SRP Perturbation on Nominal Model

The knowledge error scatter plot in Figure 5.37 is, for the most part, the same size and shape of the baseline. However, while it appears there are few outliers, they appear to be an order of magnitude greater than the outliers in the baseline case. Perhaps the knowledge error bias that resulted from not accounting for the gravity harmonics perturbations in the nominal model lead to SRP perturbation differences acting on the

truth and estimated states that actually increased the knowledge error, at least in the case of the outliers. This increase in knowledge error in the outlier cases at the time of the maneuver led to greater true target error in the outliers, as is seen in Figure 5.36. Another possible explanation for this counter-intuitive result is given in the Section 5.15: "OD cutoff equal to 5 hours".

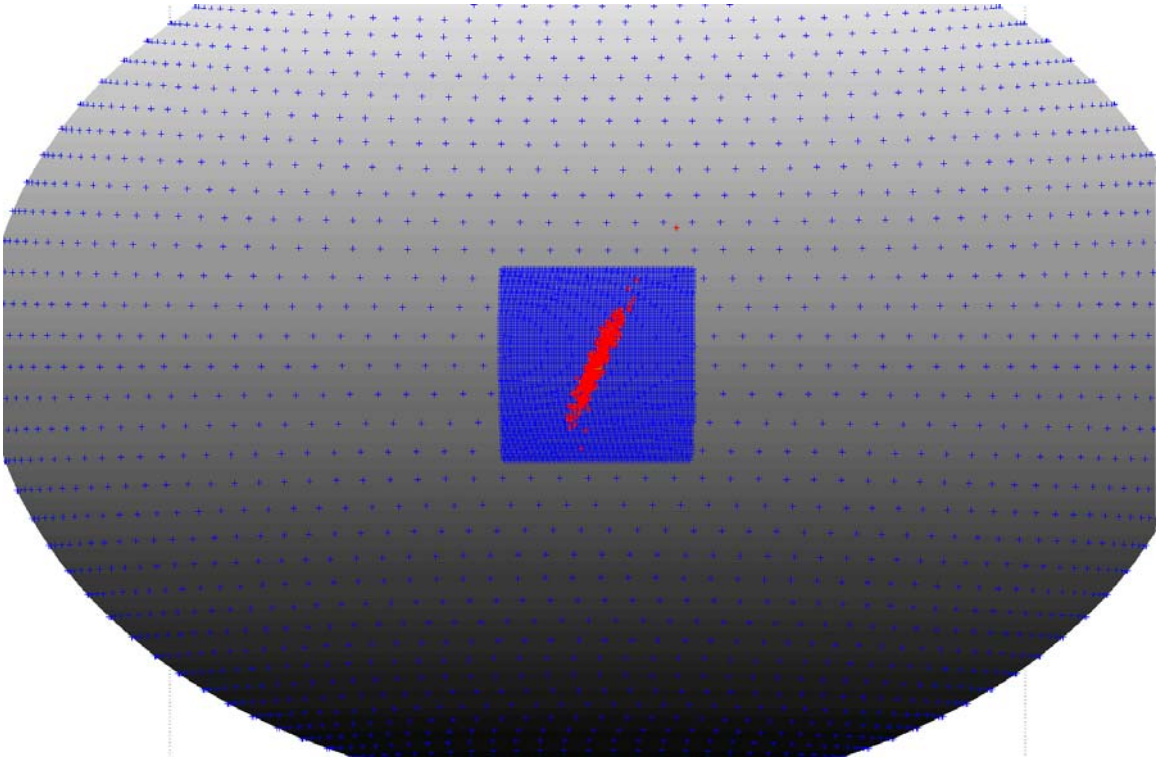

Figure 5.38: Final Positions Plot – Only SRP Perturbation on Nominal Model

5.12 ADDING ONLY THIRD BODY PERTURBATIONS TO NOMINAL MODEL

In this section, third body perturbations were added to the nominal model and the other two perturbations were turned off in the nominal model. In the baseline case, none of these perturbations were in the nominal model.

Table 5.12: Monte Carlo Statistics – Only Third Body Perturbations

Number of misses out of 500 cases	33 (6.6%)
Average target error magnitude	31.84 meters
Target error magnitude standard deviation	61.73 meters

The number of misses in Table 5.12 is identical to the baseline case, and the average target error magnitude and target error magnitude standard deviation are very close to the same values for the baseline case. These results suggest the third body perturbations are very small and have very little impact on the spacecraft trajectory. Further evidence of this minute impact is seen in the true target error scatter plot in Figure 5.39 and the knowledge error scatter plot in Figure 5.40 below, both of which are very similar to the baseline plots.

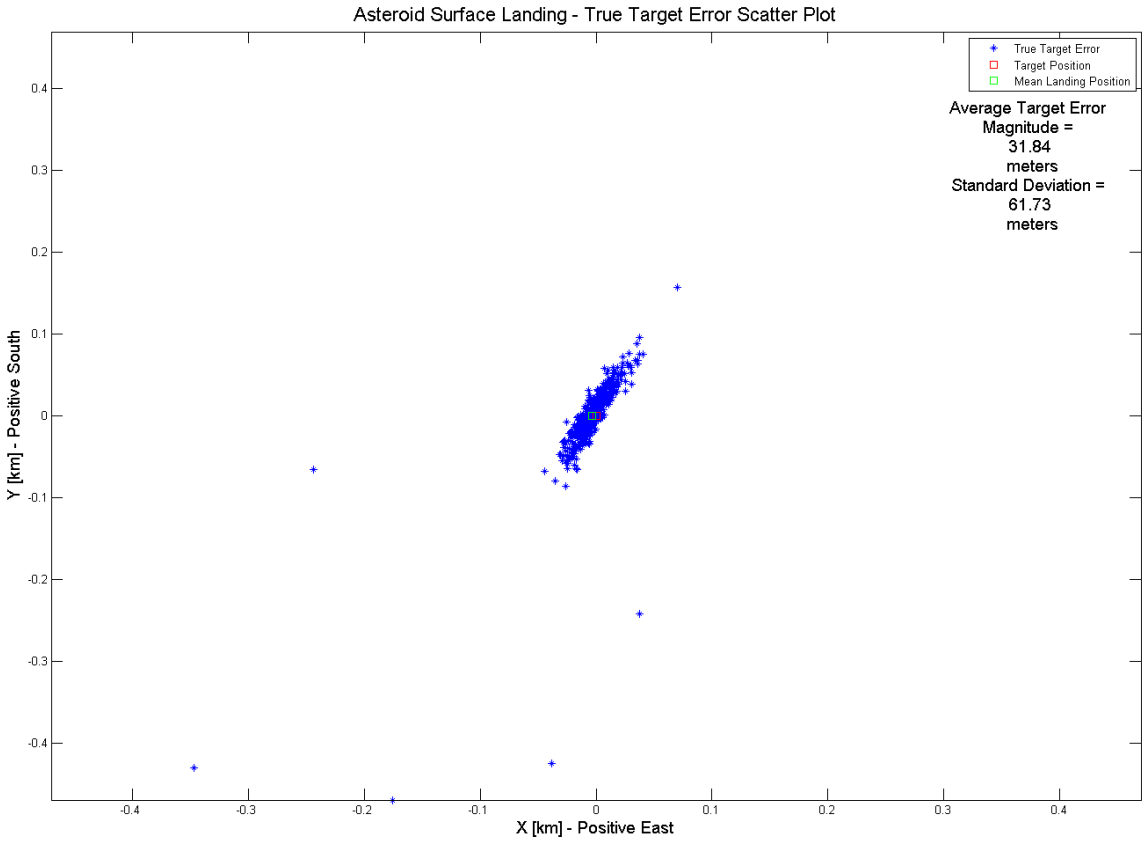

Figure 5.39: True Target Error Scatter Plot – Only Third Body Perturbations on Nominal Model

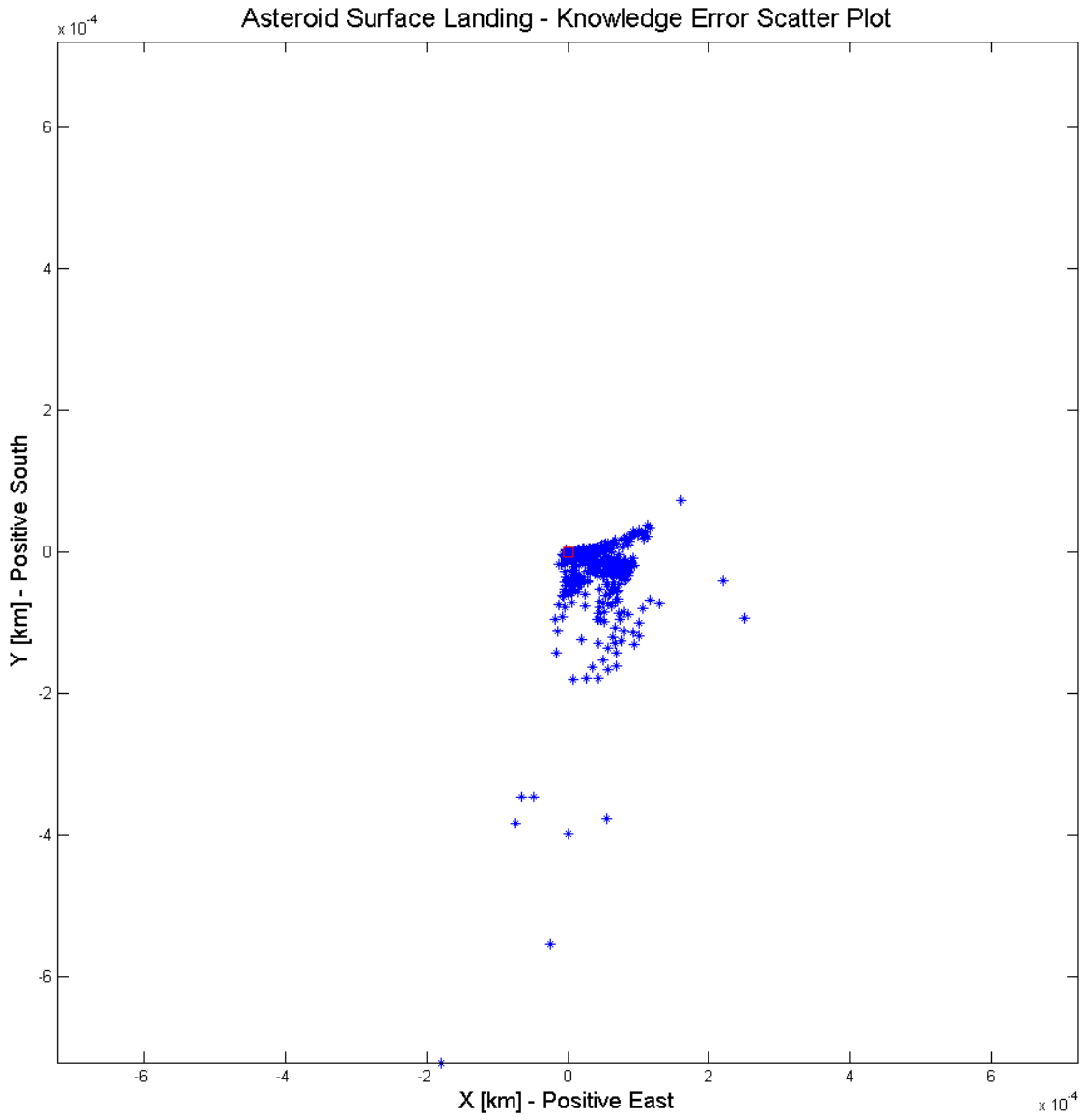

Figure 5.40: Knowledge Error Scatter Plot – Only Third Body Perturbations on Nominal Model

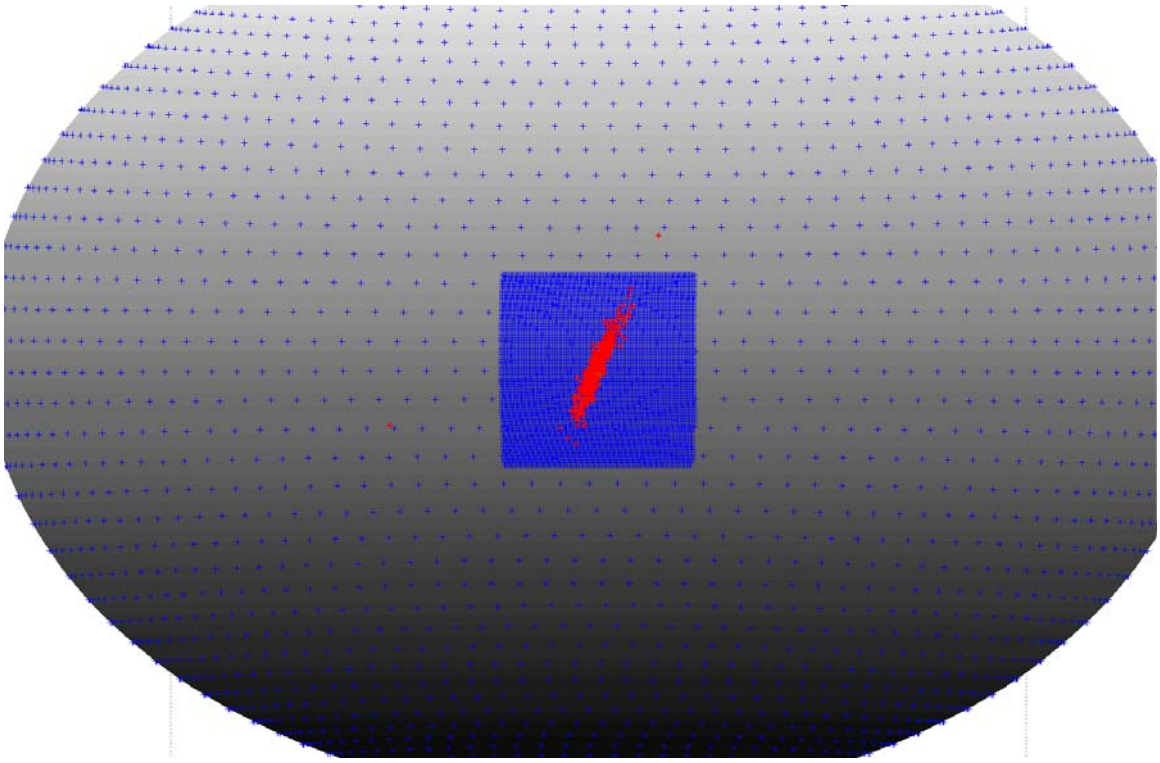

Figure 5.41: Final Positions Plot – Only Third Body Perturbations on Nominal Model

5.13 ADDING ONE MANEUVER (2 TOTAL):

An additional maneuver was added to the baseline profile simulation in an effort to determine if it would increase the accuracy of landing. This maneuver was added approximately halfway between the original maneuver time in the baseline scenario and the target time. The maneuver times were November 25, 2017 at 5:50 am and November 25, 2017 at 8:25 am. The desired landing time is November 25, 2017 at 11:00 am.

Table 5.13: Monte Carlo Statistics – Adding Two Maneuvers

Number of misses out of 500 cases	7 (1.4%)
Average target error magnitude	25.23 meters
Target error magnitude standard deviation	55.28 meters

The number of misses in Table 5.13 is seven, which is dramatically less than the baseline scenario. The reduction in the average target error magnitude and target error magnitude standard deviation as compared to the baseline values is more modest than the drop in outliers, but still significant. A similar drop in outliers was also seen when reducing the initial state errors. But unlike with initial state errors, which can probably be reduced only so much due to the distance between the Earth and spacecraft, planning more maneuvers which the spacecraft will execute autonomously can be a more feasible option for mission operators.

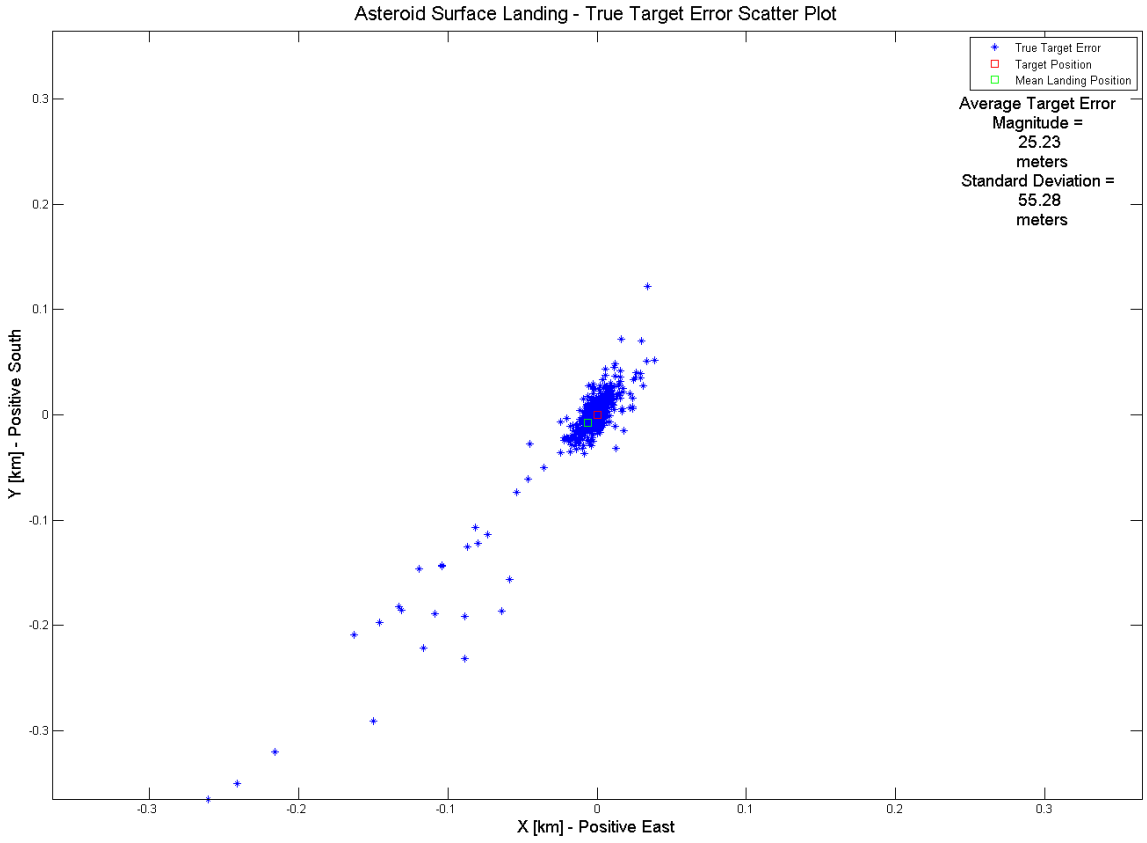

Figure 5.42: True Target Error Scatter Plot – Adding One Maneuver

The true target error scatter plot in Figure 5.42 has many more visible outliers than the baseline plot. This increase results from the fact that many of these outliers were

previously not landing at all. These additional outliers are now part of the average target error magnitude and yet it is still smaller than the baseline value because the primary footprint is smaller than the baseline footprint.

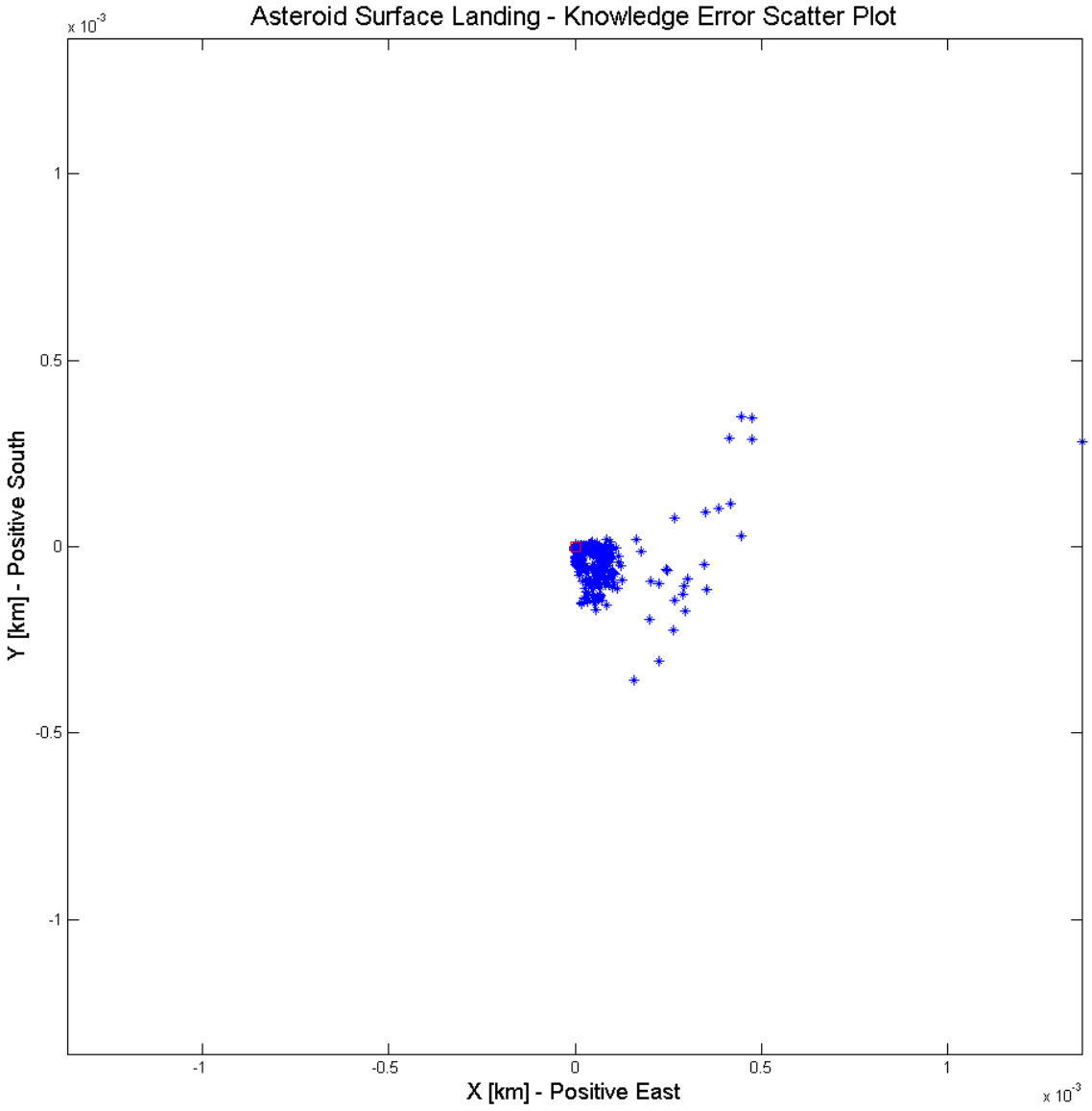

Figure 5.43: Knowledge Error Scatter Plot – Adding One Maneuver

The knowledge error scatter plot in Figure 5.43 appears to be approximately the same size and shape as the baseline plot, although many outliers appear to be further out.

One outlier in particular is twice as far out than the most distant outlier from the baseline scenario. These larger outliers are likely due to larger amounts of maneuver execution error, which results from having two maneuvers rather than one.

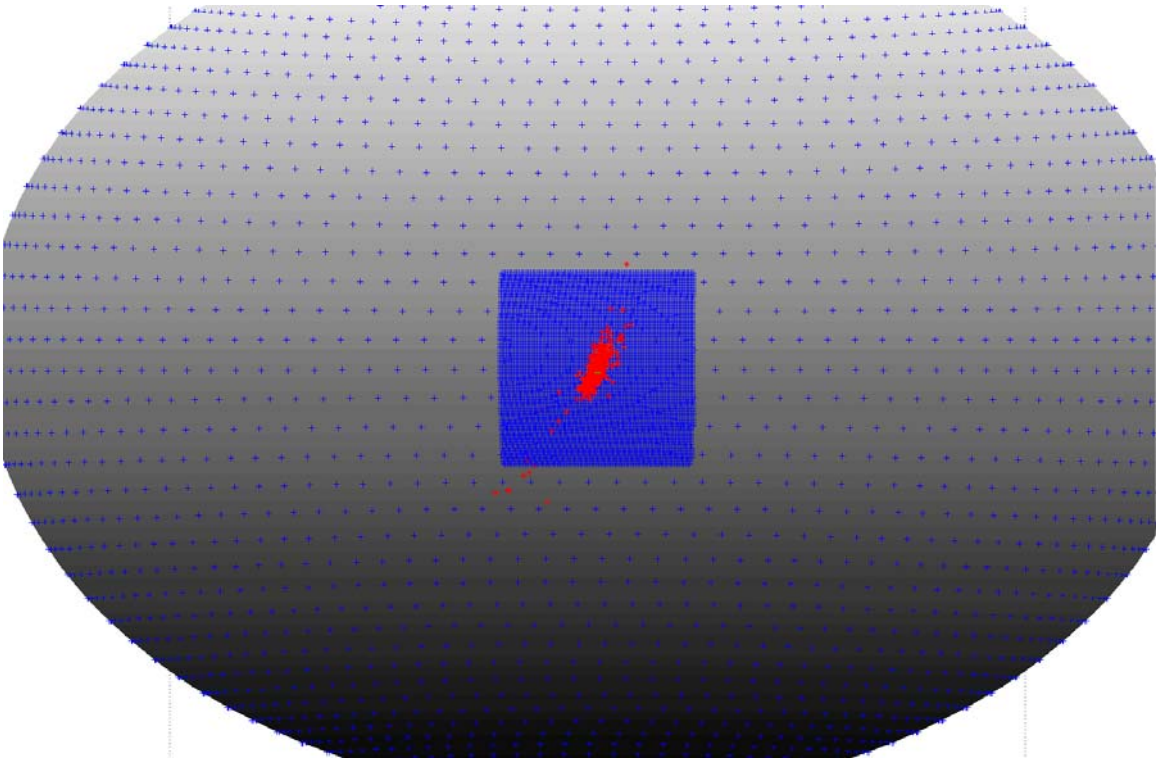

Figure 5.44: Final Positions Plot – Adding One Maneuver

5.14 ADDING TWO MANEUVERS (3 TOTAL):

Two additional maneuvers were added to the baseline profile simulation in an effort to determine if this would increase the accuracy of landing. One maneuver was added approximately halfway between the original maneuver time from the baseline scenario and the target time, and the second maneuver was added halfway between the last maneuver added and the target time. Thus, the maneuver times for this simulation

were November 25, 2017 at 5:50 am, November 25, 2017 at 8:25 am, and November 25, 2017 at 9:45 am. The desired landing time is November 25, 2017 at 11:00 am.

Table 5.14: Monte Carlo Statistics – Adding Two Maneuvers

Number of misses out of 500 cases	2 (0.4%)
Average target error magnitude	19.09 meters
Target error magnitude standard deviation	56.95 meters

The number of misses in Table 5.14 is the lowest number of misses for any variation of the baseline profile that is discussed in these results. With only two cases missing the asteroid out of 500, this scenario is even more accurate than the 50% initial error scenario, and meets the 99% criteria by having less than 1% of the cases missing the asteroid. The average target error magnitude is also reduced significantly from the scenario where one maneuver was added. The true target error scatter plot in Figure 5.45 below has approximately the same number of outliers as the scenario where one maneuver was added, but the primary footprint is significantly smaller.

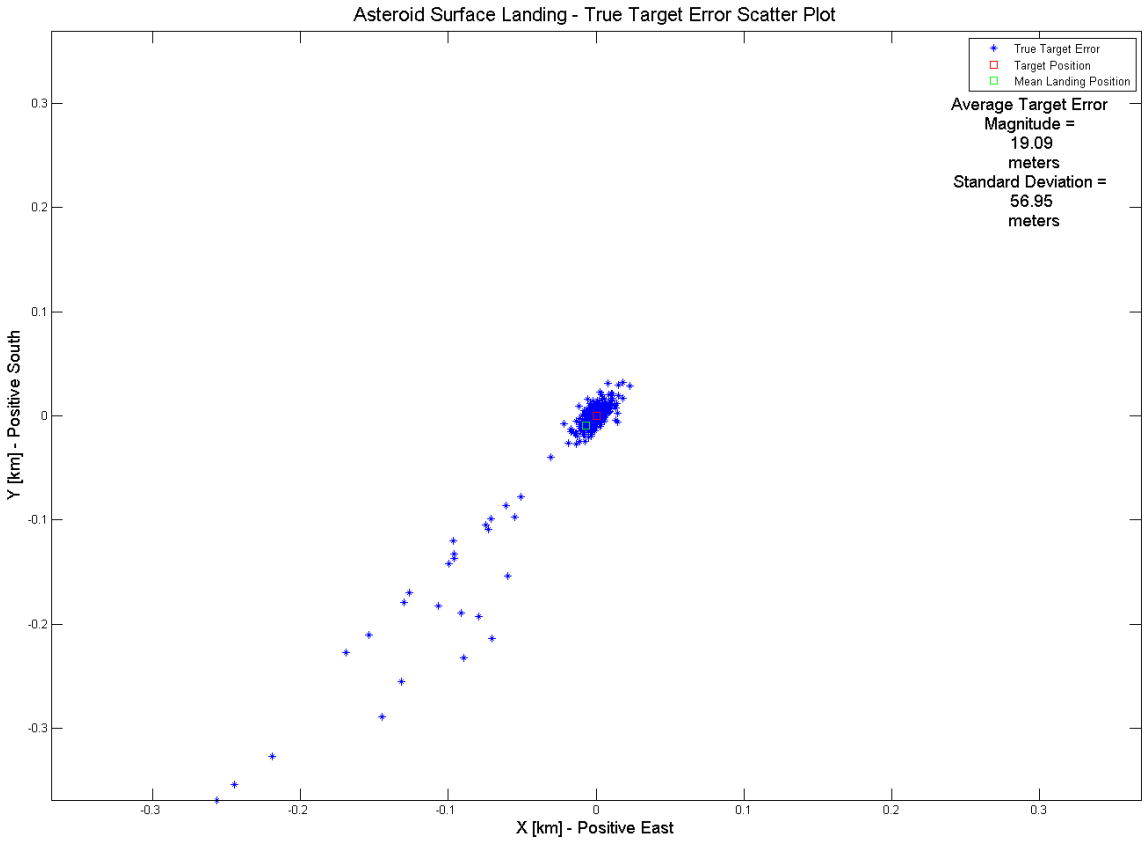

Figure 5.45: True Target Error Scatter Plot – Adding Two Maneuvers

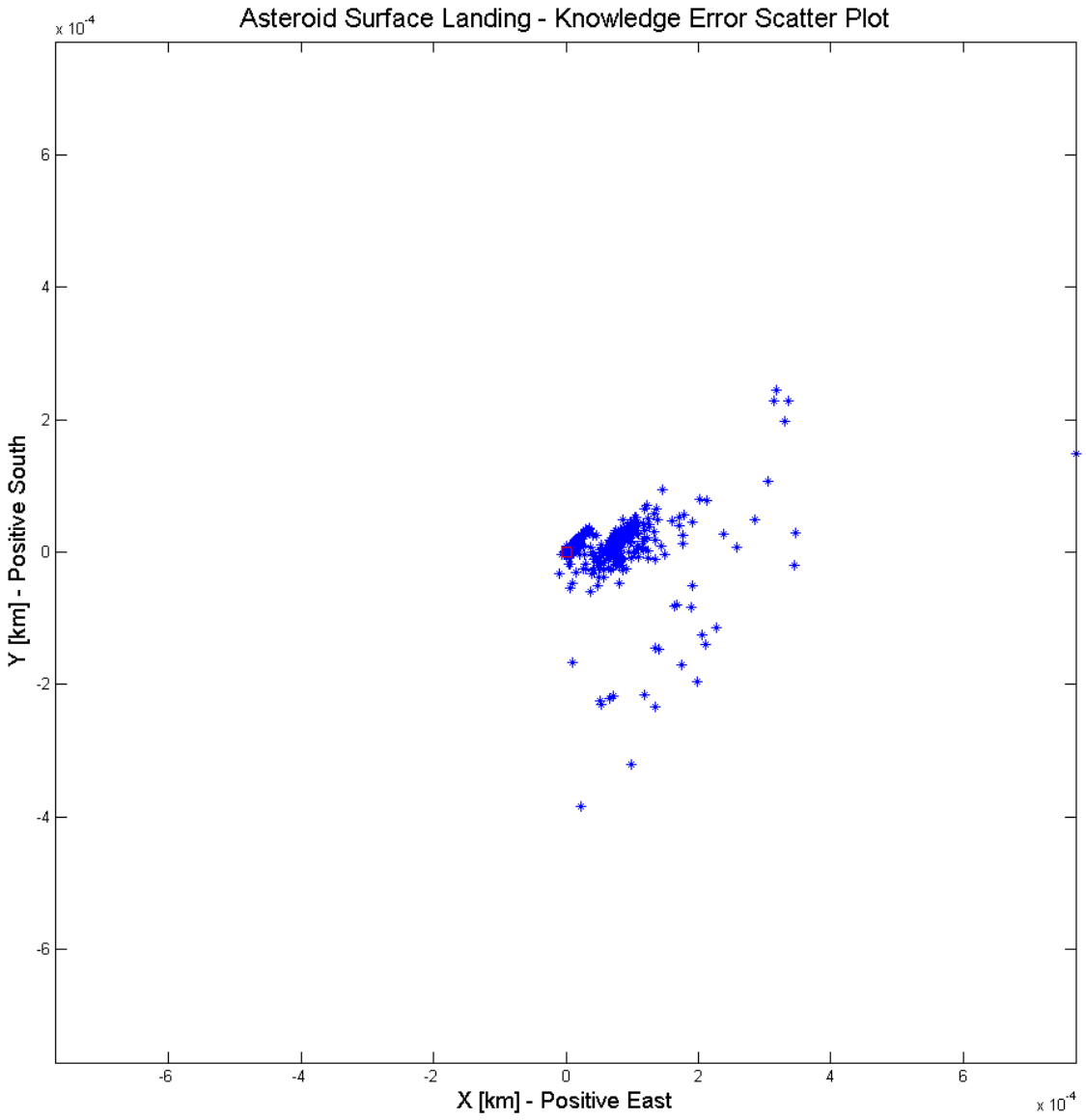

Figure 5.46: Knowledge Error Scatter Plot – Adding Two Maneuvers

The knowledge error scatter plot in Figure 5.46 appears to be approximately the same size and shape as the baseline plot. However, it appears to be a little angularly wider and slightly shifted counter-clockwise about the origin.

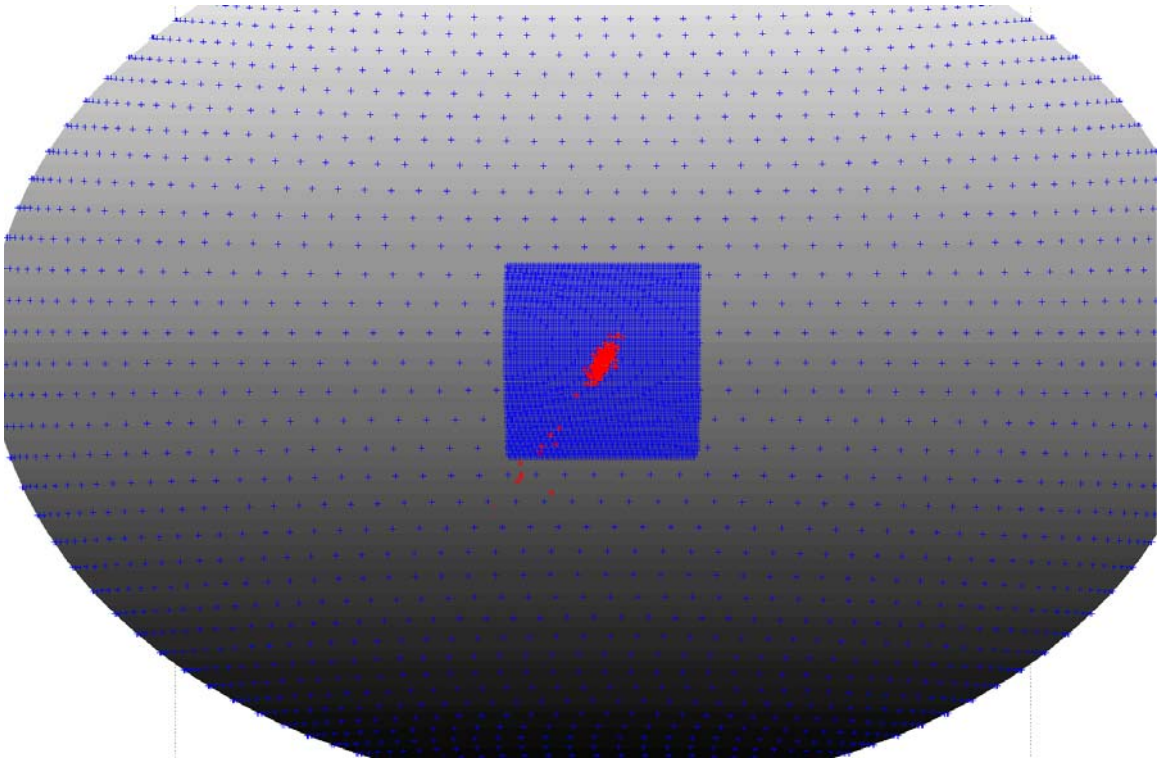

Figure 5.47: Final Positions Plot – Adding Two Maneuvers

5.15 SETTING OD CUTOFF TO 5 HOURS

The orbit determination cutoff value (OD cutoff), discussed in Section 3.7.5, was increased to five hours to simulate the delay in communications with the ground for a ground-based control scenario. Five hours is an estimate of the time that would be required to send a signal from the asteroid location in the solar system to the Earth, have the ground-based system calculate a navigation solution and a maneuver, and send the signal back to the spacecraft from the Earth.

Table 5.15: Monte Carlo Statistics – Setting OD cutoff to 5 hours

Number of misses out of 500 cases	28 (5.6%)
Average target error magnitude	48.37 meters
Target error magnitude standard deviation	135.89 meters

The number of misses in Table 5.15 is less than the 33 that occur for the baseline case, which was not expected. With what should be significantly greater knowledge error at the time of the maneuver, the number of cases that should not reach the surface should increase dramatically.

One potential explanation for this behavior is the following. The knowledge error bias introduced by not including the gravity harmonics perturbations in the nominal model, which is amplified by this five-hour propagation with no navigation before the maneuver, results in the spacecraft estimated state, which the targeter utilizes for trajectory calculation, being further along in the approximately circular orbit than the truth state at the maneuver time. This condition produces trajectories that reach the surface more consistently, though with greater landing error. A simple example of this idea is shown in Illustration 5.1 below.

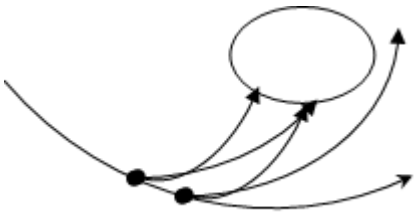

Illustration 5.1: Missing the asteroid

In the above Illustration 5.1, the point earlier in the orbital trajectory, i.e. the point on the left, represents the truth state at the maneuver time. The right point later in the

orbital trajectory represents the estimated state at the maneuver time that the targeter will use. Two trajectories representing the extreme cases of perfectly reaching the target and of missing the asteroid completely are shown from this initial estimated state. The same two trajectories from the initial truth state will both land on the surface, though the landing error will be greater. Any trajectory that will miss the asteroid will do so in the same fashion as in Illustration 5.1; thus an initial state of the landing trajectory which occurs earlier than expected in the orbit will be more likely to reach the surface. The results from Figure 5.48 below reinforce this explanation, as there are greater numbers of outliers and the average target error magnitude is greater than in the baseline case.

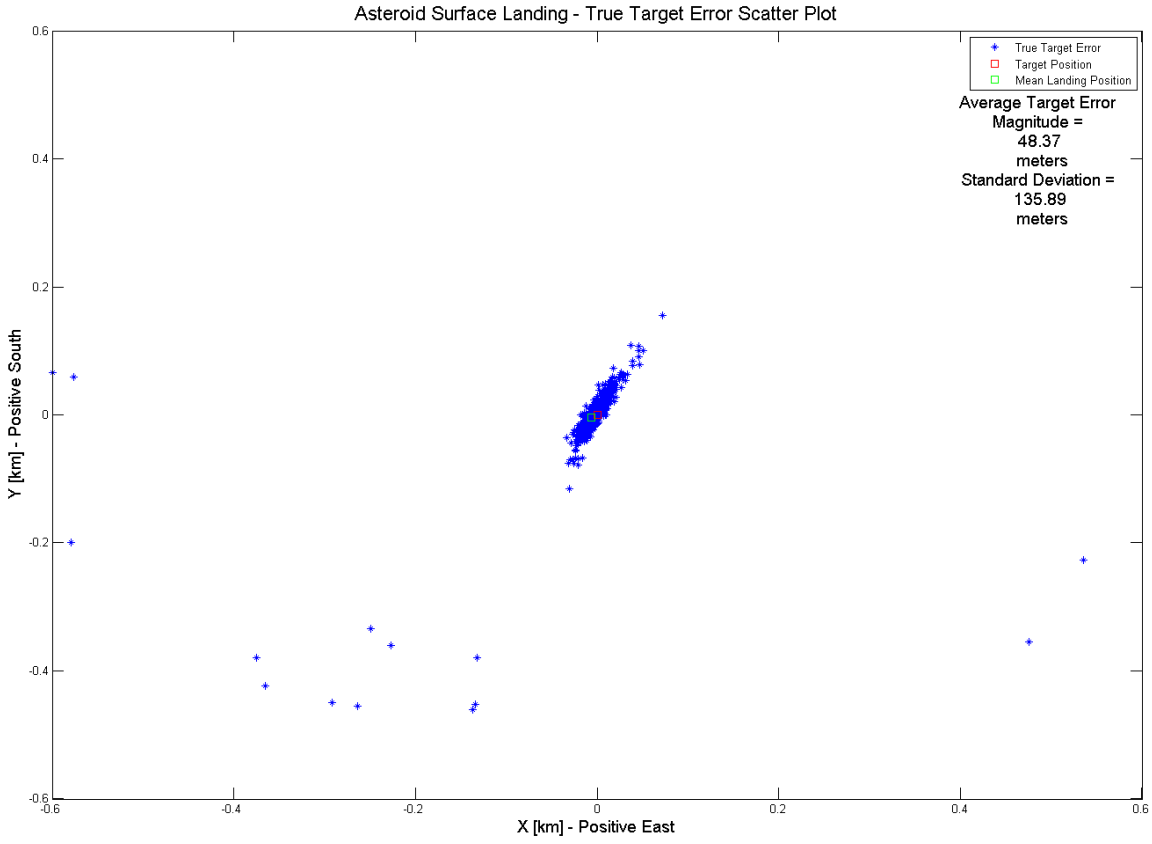

Figure 5.48: True Target Error Scatter Plot – Setting OD cutoff to 5 hours

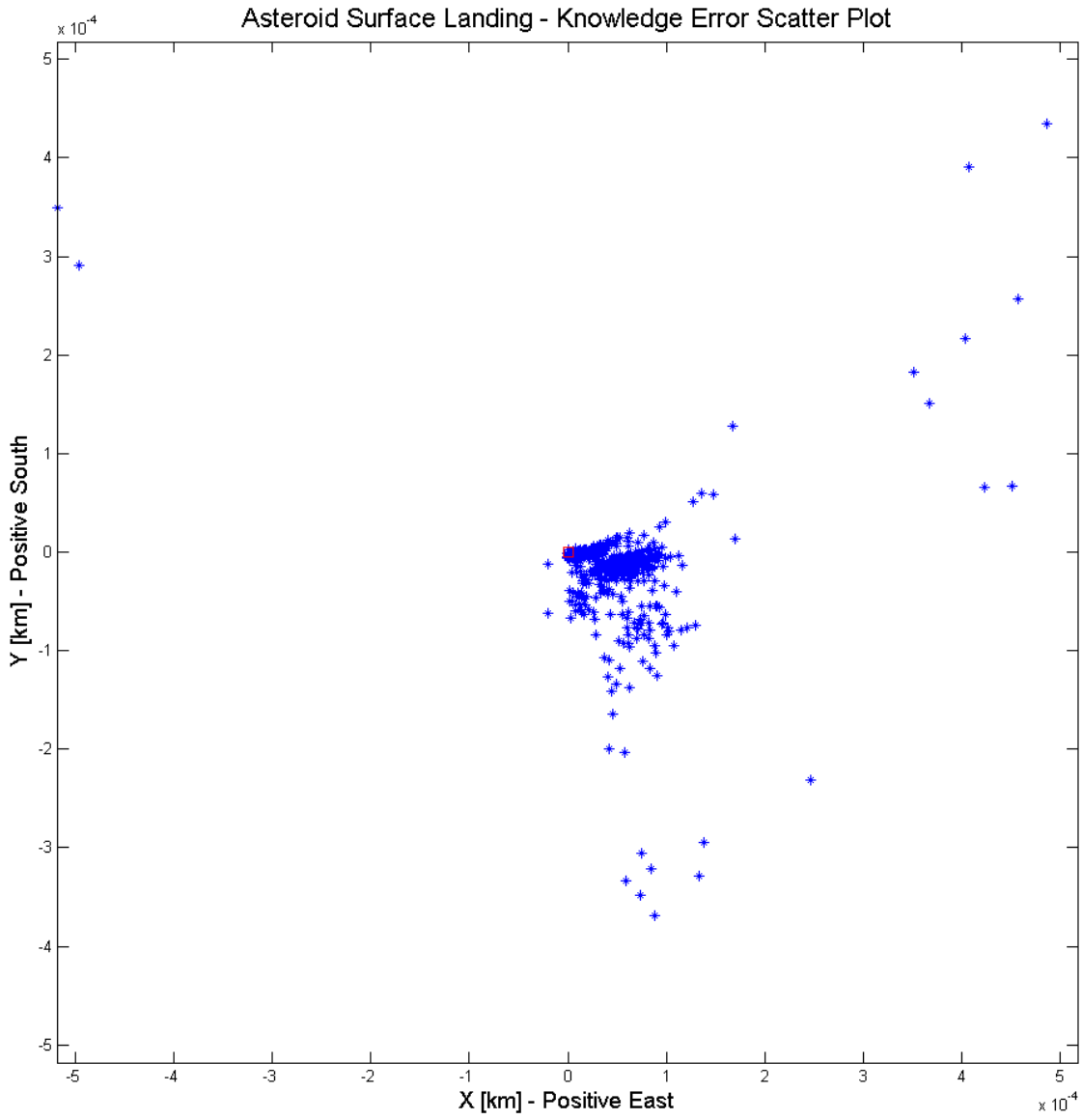

Figure 5.49: Knowledge Error Scatter Plot – Setting OD cutoff to 5 hours

The knowledge error scatter plot in Figure 5.49 is very similar to the baseline. This is because after the maneuver the filtering process resumes, thus reducing the knowledge errors at landing to the baseline values.

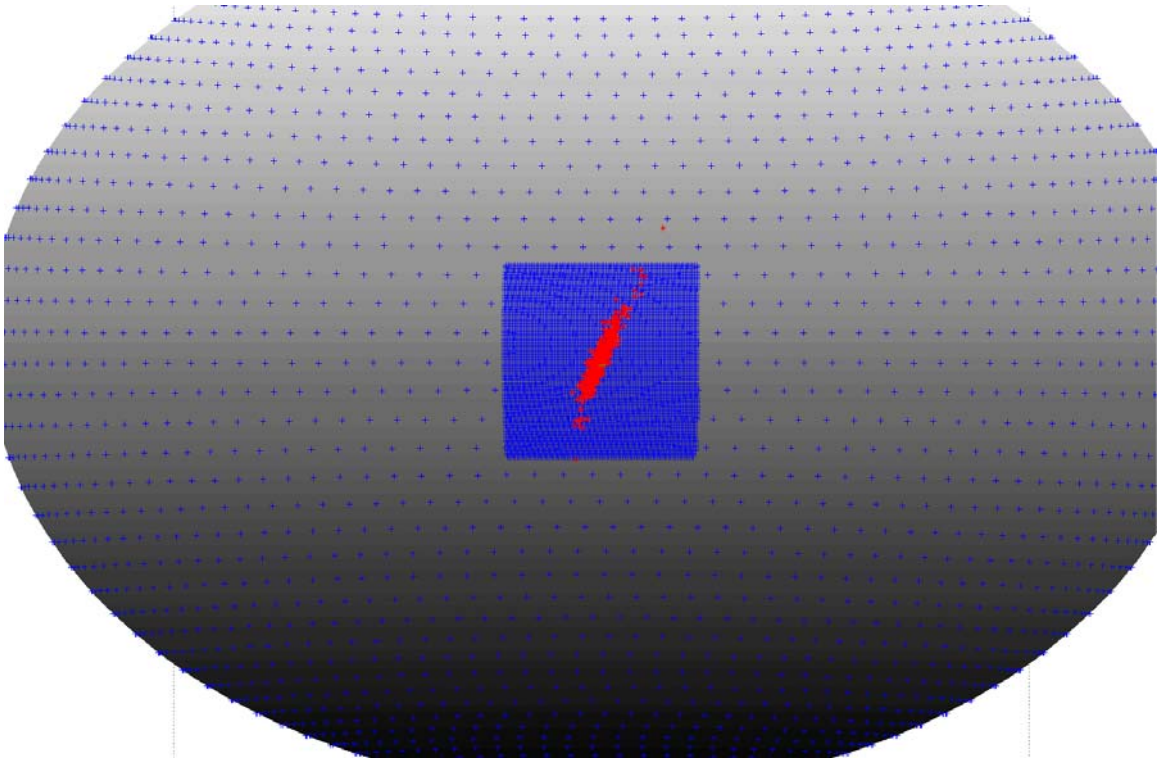

Figure 5.50: Final Positions Plot – Setting OD cutoff to 5 hours

5.16 NO ONBOARD NAVIGATION

In this scenario, the onboard autonomous navigation was completely shut off. The results are shown below.

Table 5.16: Monte Carlo Statistics – No Onboard Navigation

Number of misses out of 500 cases	61 (12.2%)
Average target error magnitude	86.71 meters
Target error magnitude standard deviation	94.74 meters

The number of misses in Table 5.16 is almost double the number of misses in the baseline case, and the average target error and target error magnitude standard deviation

are both much larger. All of these results were expected due to the considerably larger knowledge error at the maneuver time that adversely affected the maneuver calculations.

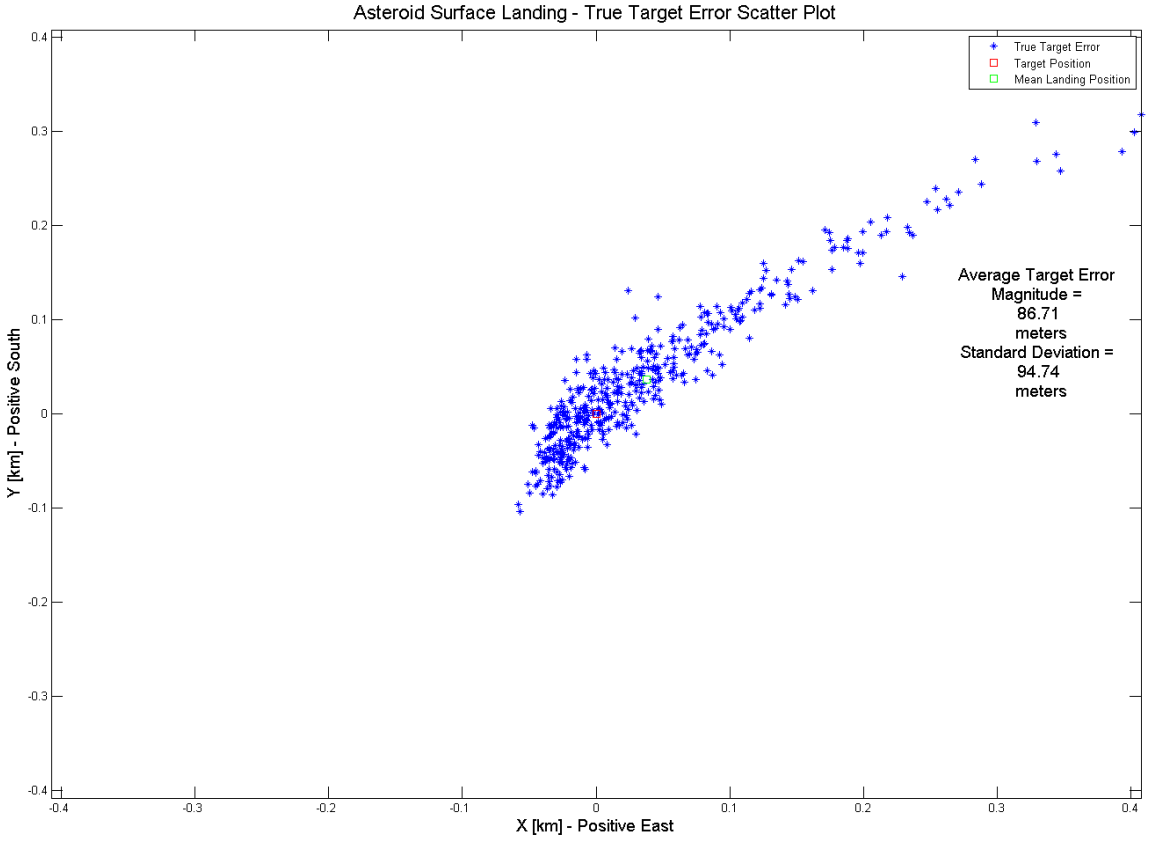

Figure 5.51: True Target Error Scatter Plot – No Onboard Navigation

The true target error scatter plot in Figure 5.51 is significantly larger than the baseline landing footprint. An interesting feature is that there are no outliers beyond the primary footprint, unless those on the top right edge of the footprint on the right side of the plot are considered outliers. This fact suggests that the filtering plays a role in producing the outliers seen far from the primary footprint.

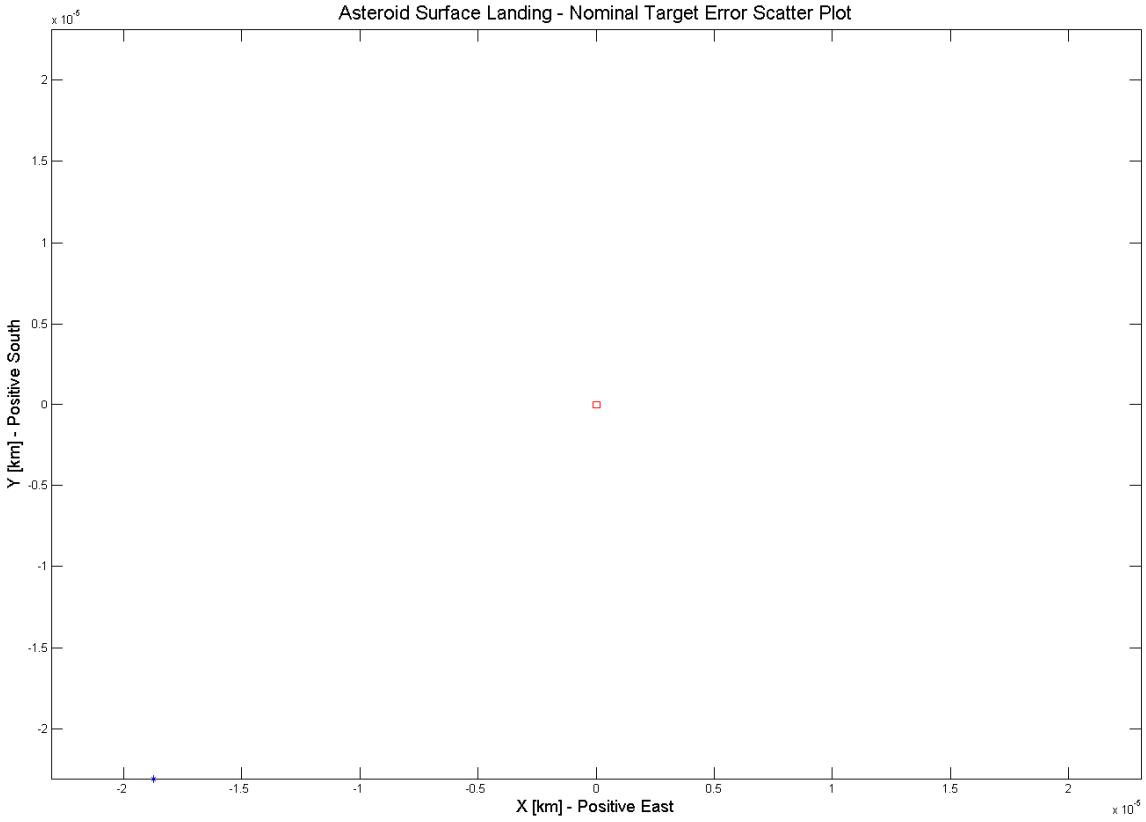

Figure 5.52: Nominal Target Error Scatter Plot – No Onboard Navigation

The nominal target error scatter plot in Figure 5.52 contains only one point, and this point is very close to the target. This lack of other points is due to the fact that the nominal model and trajectory are exactly the same in every case since the spacecraft is not receiving any new information. Thus, the estimated state is now simply a perfect propagation to the target on the surface.

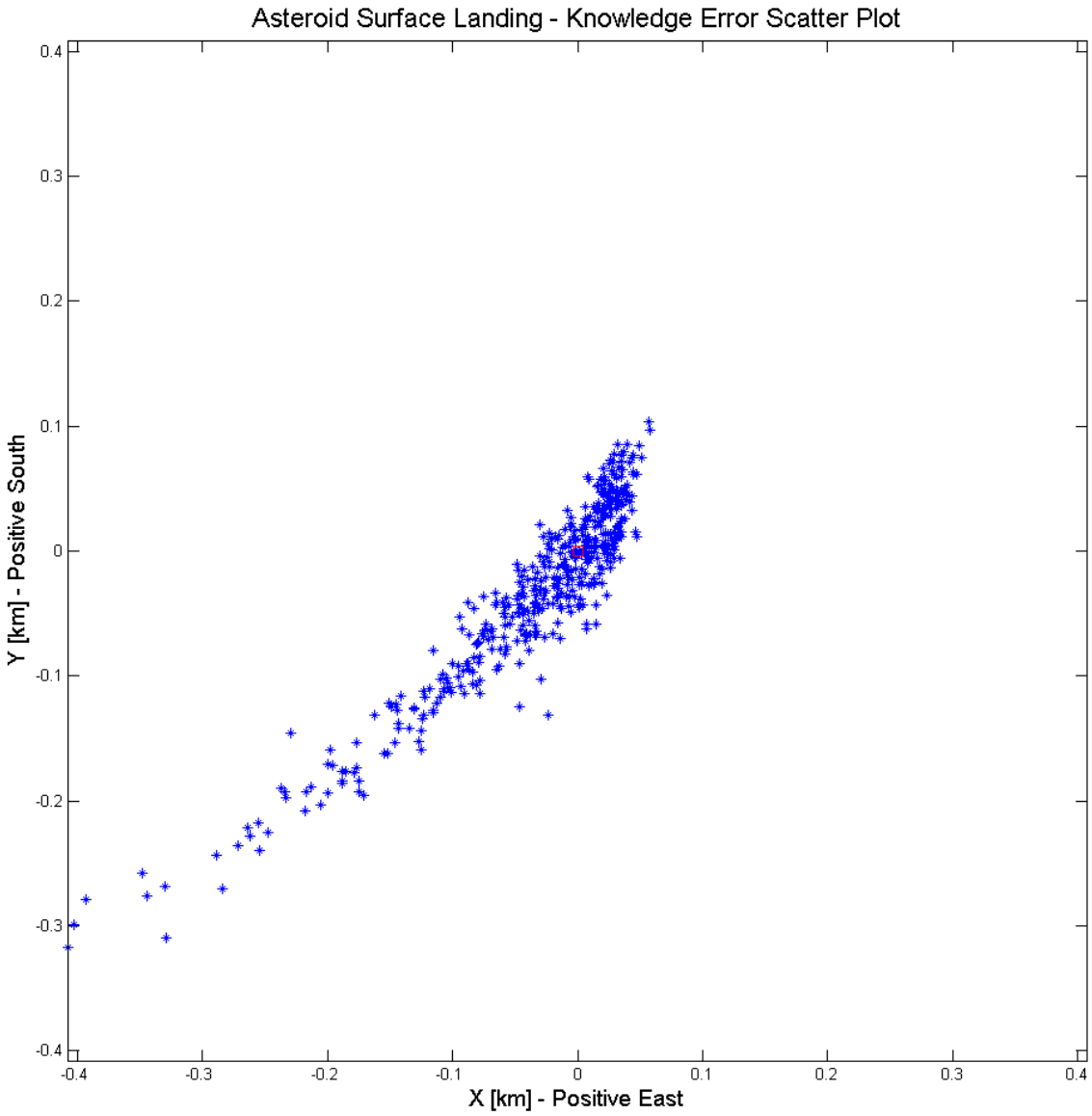

Figure 5.53: Knowledge Error Scatter Plot – No Onboard Navigation

The knowledge error scatter plot in Figure 5.53 is the mirror image of the true target error scatter plot in Figure 5.51. The knowledge error is calculated by subtracting the truth state from the estimated state, but when the estimated state is always effectively at the origin, every point in the knowledge error scatter plot becomes the negative value of the points in the true target error scatter plot.

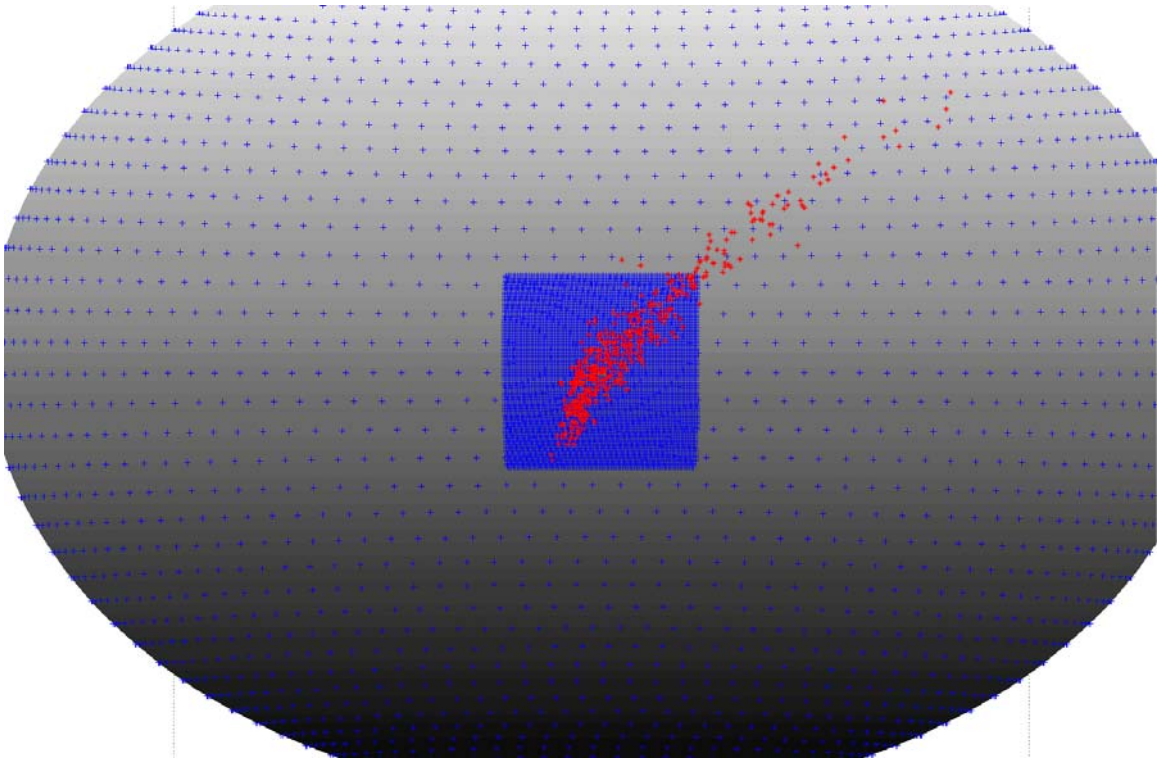

Figure 5.54: Final Positions Plot – No Onboard Navigation

CHAPTER 6

Summary

6.1 CONCLUSIONS

SANT was created and used to generate Monte Carlo analysis of spacecraft landing error on a small asteroid. Navigation error sources were varied in an attempt to investigate which aspects of the spacecraft landing footprint on the small-body surface were sensitive to these errors. Various conclusions about the nature of this small-body landing autonomous navigation problem were discovered during the study.

The utilization of landmarks for AON appeared to perform well, with consistently low knowledge error on the order of tens of centimeters. The result suggests excellent performance by the filters in estimating the state of the spacecraft.

The average target error magnitude and the size of the landing footprint appeared to be quite sensitive to maneuver execution errors, but the outliers were affected very little by variation in this error source. Varying the observation error did not change the average target error magnitude or primary footprint a significant amount, but a small difference in the number of outliers was observed, suggesting a slight correlation between observation error and the number of outliers. However it was concluded that observation error is clearly not the primary source of these outliers.

The initial state error produced the largest impact on the number of outliers, with a fifty percent decrease in initial errors almost eliminating the outliers and a 200% increase in initial errors more than doubling the number of outliers. Thus, the outliers are highly sensitive to the initial state error, and this error source appears to be the primary cause of these outliers. The average target error magnitude and footprint also appear to

change significantly with varying initial state error, but they are not as sensitive to this error source as the outliers.

The attitude error appeared to have little to no impact on any results of the simulation. This result is likely due to the estimation of the attitude error in the Kinematic Filter.

Adding all three perturbations that are in the truth model to the nominal model led to fewer outliers and a somewhat smaller landing footprint, as the better knowledge of the forces acting on the spacecraft led to better maneuver performance. The most dramatic change was in the knowledge error plot, where a large amount of the bias was removed and the scale of the scatter plot dropped by an order of magnitude. It was discovered by adding each perturbation individually that the gravity harmonics were most responsible for this bias and size difference, with the SRP having a minor contribution and the third body gravitational perturbations introducing almost no error at all.

Increasing the number of maneuvers in the simulation reduced the number of outliers by an order of magnitude and also reduced the size of the landing footprint. This result is of great importance because this option might be more feasible for mission operators than attempting to reduce the initial state error.

Increasing the orbit determination cut-off value to five hours in an attempt to simulate control from the Earth for the mission did not lead to the expected results of overall worse performance. There were actually fewer cases that missed the asteroid. However, for those cases that did reach the surface, the landing error was greater. Because of this larger landing error, it is likely that landing within a reasonable distance from the target would be very difficult to accomplish with high certainty using ground control. This suggests that autonomous navigation is probably needed for an accurate landing on a small-body in a different part of the solar system than the Earth.

When onboard navigation was completely turned off, the number of misses almost doubled and the landing error increased dramatically, as expected. But when the onboard navigation was disabled, there were no outliers as seen in previous simulations, which suggest that the filtering process plays a role in producing these outliers that appear far from the primary footprint.

6.1 RECOMMENDATIONS FOR FUTURE WORK

Many other variations of this simulation can be investigated. A number of ideas that were considered but were unable to be included in this study due to time constraints are given below:

- Use different target positions on the asteroid
- Account for the darkness of asteroid side not facing Sun to simulate fewer visible landmarks
- Account for lack of SRP if on the opposite side of asteroid from the Sun
- Account for asteroid features obstructing certain landmarks
- Vary the density and total number of landmarks on the entire asteroid or just around the target, or both, to observe the sensitivity of the landing error to the number of landmarks available at different times of the simulation
- Utilize landmarks with non-uniform placement on the surface, using either random placement or landmark placement based on what might actually occur for a mission
- Vary the asteroid rotation rates
- Vary the asteroid shape
- Vary the gravitational field of the asteroid
- Vary the minimum and maximum amounts of data available to the filters
- Alter the filter stopping conditions, or tolerances

- Attempt different numbers of maneuvers at different times
- Attempt many more different OD cutoff times than those performed in this study
- Evaluate the impact of different initial states and initial orbits
- Evaluate the effect of different initial covariances for both the Kinematic Filter and Dynamical Filter
- Evaluate different spacecraft trajectory propagation options
- Evaluate other variations for the picture times, perhaps placing them in a non-uniform set of times
- Vary the focal length and other parameters for the camera and spacecraft optics
- Study the influence of dust getting dispersed by the spacecraft approaching the surface on the spacecraft imager's ability to see the landmarks and thus the spacecraft's ability to navigate. Consider a scenario in which the spacecraft is performing touch-and-go operations that involve the spacecraft touching the surface and immediately lifting off again to travel further along the surface.
- Study how accurately and quickly the spacecraft can estimate the state after launching off the surface in order to confirm that it is moving away from the surface.
- Develop an iterative scheme that would allow minimum energy Lambert targeting to be using with asteroid rotation. Minimum energy Lambert targeting was not utilized in this study because the travel time of the resulting trajectory, and thus final position of the target on the rotating body at the time of landing, can not be determined before using the targeter. The iterative scheme should avoid "chasing" the target around the asteroid.
- Calculate the covariance at the surface by estimating and fitting through the maneuvers.

- Create an extended Kalman filter to perform the Dynamical Filtering.
- Study the cases in which the spacecraft did not reach the surface and analyze the common features of these cases.
- Change the time duration of the simulation, such as doubling the length of the simulation to see if the outliers are still as sensitive to the initial error.
- The results from the Appendix Section C.1 suggest that the closer the spacecraft is to the simulation starting position, when it is directly above the target position, the smaller the landing errors will be. The disadvantage of performing the maneuver directly over the target is the increased fuel cost, but because of the very low gravity of the small-body, this increase might be very small and thus worth it. Study the cost/benefit features of this trade-off.
- Vary each of the maneuver execution error parameters to better understand the characteristics of the maneuver execution error and their affect on the landing accuracy – and which characteristics are most responsible for the changes seen in the 50% and 200% maneuver execution error cases in Chapter 5.

Appendix A: Batch Filter

A batch navigation filter is used to estimate the state parameters within SANT, which can be either the position alone, the position and velocity, or the position, velocity, and pointing bias and drift. The batch filter is based on the accumulation technique detailed in reference [25]. The batch estimation algorithm is given in the following procedure:

- Set any needed constants
- (A) Initialize for this iteration
  - Set $i = 1$ (the observation count)
  - Set $t_{i-1} = t_0$ (the initial time, or epoch time, or time of first obs)
  - Set $X^*(t_{i-1}) = X_0^*$ (initial guess at the nominal state)
  - Set $\Phi(t_{i-1}, t_0) = \Phi(t_0, t_0) = I$ (STM set to Identity)
  - If there is an *a priori* estimate, set $\Lambda = \overline{P}_0^{-1}$, and set $N = \overline{P}_0^{-1}\overline{x}_0$ (where $\overline{x}_0$ is usually zero, so N usually zero)
  - If there is no *a priori* estimate, set $\Lambda = 0$ and $N = 0$
- (B) Read the next observation: t_i (time of observation), Y_i (observable), R_i (weight derived from covariance matrix)
- Integrate reference trajectory and STM from previous time (t_{i-1}) to current time (t_i)
  - $\dot{X}^* = F(X^*(t), t)$ with initial conditions $X^*(t_{i-1})$ (where the asterisk indicates nominal)
  - $A(t) = \left[\dfrac{\partial F(X,t)}{\partial X}\right]^*$, where the asterisk indicates it is evaluated on the nominal trajectory
  - $\dot{\Phi}(t, t_0) = A(t)\Phi(t, t_0)$ with initial conditions $\Phi(t_{i-1}, t_0)$

- This gives $X^*(t_i)$, $\Phi(t_i, t_0)$.

- Accumulate current observation
  - $\widetilde{H}_i = \left[\frac{\partial G(X,t)}{\partial X}\right]^*$, where $G(X, t)$ is the system model
  - $y_i = Y_i - G(X_i^*, t_i)$, where $G(X_i^*, t_i)$ is the system model evaluated on the nominal trajectory. This is the residual, which is the difference between the actual observation and the observation predicted by the model.
  - $H_i = \widetilde{H}_i \Phi(t_i, t_0)$
  - $\Lambda = \Lambda + H_i^T R_i^{-1} H_i$ where Λ is accumulated for each observation, and R_i^{-1} is the inverse of the R matrix, which serves as the weight matrix
  - $N = N + H_i^T R_i^{-1} y_i$, where N is accumulated for each observation
- If $t_i < t_{final}$:
  - $i = i + 1$. Thus:
  - t_i becomes t_{i-1}
  - $X^*(t_i)$ becomes $X^*(t_{i-1})$
  - $\Phi(t_i, t_0)$ becomes $\Phi(t_{i-1}, t_0)$
  - Go to (B) and read next observation.
- If $t_i \geq t_{final}$:
  - Go to (C) and solve normal equations.
- (C) Solve Normal Equations
  - The normal equations are $\Lambda \hat{x}_0 = N$. The solution is $\hat{x}_0 = \Lambda^{-1} N$. The covariance is $P_0 = \Lambda^{-1}$.
- Update the nominal trajectory: Replace X_0^* with $X_0^* + \hat{x}_0$.
- Determine if the process has converged (for example, if the root mean square of the residuals changes less than a certain preset amount)

- If it has converged: Stop
- If it hasn't converged: Iterate. Shift $\bar{x}_0 = \bar{x}_0 - \hat{x}_0$. Use the original $\overline{P}_0$. Return to (A) with new values for X_0^* and $\bar{x}_0$.

Appendix B: Baseline Profile Settings

These switches and values, which are set in the initialization file by the user, correspond to the baseline profile, or default settings:

Most Important User Set Switches/Values

- Open Loop or Filter: Filter
- Manual (user input) Initial/Epoch State errors: Off
    - Error in the Initial Position in the View2 Reference Frame: [0 0 0] km
    - Error in the Initial Velocity in the View2 Reference Frame: [0 0 0] km
- Calculated Initial/Epoch State Perturbation Multipliers (if Manual initial state errors off)
    - Position
        - View2 Downtrack (Along Velocity Vector) Axis: 0.0025 km
        - View2 Cross1 (Normal to Pos and Vel) Axis: 0.0025 km
        - View2 Cross2 (Normal to Vel and Cross1) Axis: 0.0025 km
    - Velocity
        - View2 Downtrack (Along Velocity Vector) Axis: 2.5e-6 km/s
        - View2 Cross1 (Normal to Pos and Vel) Axis: 2.5e-6 km/s
        - View2 Cross2 (Normal to Vel and Cross1) Axis: 2.5e-6 km/s
- Maneuvers: On
- Maneuver Targeting: K-matrix Targeting
- Maneuver execution error: On
    - Fixed magnitude = 2e-6 km/s
    - Proportion/fraction of magnitude = 0.0002
    - Fixed Direction = 4e-7 km/s

    - Proportion/fraction of direction = 0.0002 = 0.2 mrad ptg error
- Attitude Error: On
- Gyroscope Properties affecting Attitude Error
    - Initial Attitude Error: 0.1°
    - Attitude Noise: 3.3 micro-rad
    - Initial Gyro Drift Rate: 0.01/3 °/hr
    - Gyro Random Walk: 0.025 °/√(hr)
- Observation Error: On
    - Pixel Error Multiplier: 0.25 pixels
    - Line Error Multiplier: 0.25 pixels
- True Model Perturbations
    - Gravity Harmonics Perturbation = On
    - Solar Radiation Pressure Perturbation = On
    - Third Body Perturbations = On
- Nominal Model Perturbations
    - Gravity Harmonics Perturbation = Off
    - Solar Radiation Pressure Perturbation = Off
    - Third Body Perturbations = Off
- Sliding Window: On
    - Maximum Number of Observations: 16
- Surface Target
    - Longitude: 270°
    - Latitude = -2°
    - Altitude above surface: 0.004 km
- Initial/Epoch Time: 24-Nov-2017 09:00:00.0000

- Maneuver Times: 25-NOV-2017 05:50:00.0
- Target Time: 25-NOV-2017 11:00:00.0

Less Important User Set Switches/Values (Changed far less often)

- Spacecraft Mass: 500 kg
- Spacecraft Mass Error Multiplier: 20 kg
- Spacecraft Projected Area (for SRP): 12 m^2
- Spacecraft Projected Area Error Multiplier: 0.5 m^2
- Spacecraft Coefficient of Reflectivity (for SRP): 1.1 {0=transparent, 1=black body (entirely absorbed, all force transmitted), 2 = mirror}
- Asteroid GM: 3.62e-08 km^3/s^2
- Asteroid Maximum radius: 0.71646 km
- Asteroid Middle radius: 0.64984 km
- Asteroid Minimum radius: 0.52361 km
- Asteroid Rotation: On
    - Initial Right Ascension of Pole: 30°
    - Right Ascension Rate of Pole: 2°/(Julian Century)
    - Initial Declination of Pole: 40°
    - Declination Rate of Pole: 3°/(Julian Century)
    - Initial Longitude of Prime Meridian: 50°
    - Asteroid Rotation Rate: 30°/day
- Kinematic Filter
    - Position Convergence Tolerance: 0.001 km
    - Attitude Convergence Tolerance: 0.005°
    - Initial Covariance:

        - Position
            - "X" Sigma-1: 5 km
            - "Y" Sigma-2: 5 km
            - "Z" Sigma-3: 5 km
        - Attitude
            - Right Ascension Sigma: 1e-5 degrees
            - Declination Sigma: 1e-5 degrees
            - Twist Sigma: 1e-5 degrees
- ODE45 Propagation Options:
    - Relative Tolerance: 1e-9
    - Absolute Tolerance: 1e-9
- Dynamical Filter
    - Minimum Number of Kinematic Position Fixes before Dynamical Filter Update: 2
    - Maximum Number of Iterations for Dynamical Filter: 10
    - Convergent Tolerance: 1e-5 km
    - Initial Covariance
        - Position
            - Sigma-1: 5 km
            - Sigma-2: 5 km
            - Sigma-3: 5 km
        - Velocity
            - Sigma-1: 1e-2 km/s
            - Sigma-2: 1e-2 km/s
            - Sigma-3: 1e-2 km/s

- Debug Switch: Off
- Batch or Extended Kalman Filter: Batch
- Scaling the State Transition Matrix: Off
- Plotting
    - Plot the Pre-fit Residuals: Off
    - Plot the Post-fit Residuals: Off
    - Plot the error: Off
    - Plot the trajectories: Off
- Camera Properties
    - Focal Length: 10 mm
    - K-matrix Conversion: [83.3330 0;0 83.83330] x/mm
    - Central Pixel Value: [256 256] pixels
    - Camera Field of View Dimensions: [512 512] pixels
    - PSI, CHI, OMEGA Scan Platform Angles, used in inertial to camera reference frame transformation: 0°
    - Electromagnetic and Optical Distortion Parameters: [0 0 0 0 0 0]
- Picture Sequence Times
    - Initial time between pictures: 1 Hour
    - Time between pictures after first maneuvers: 10 Minutes
- Third Body Perturbations
    - Mercury: Off
    - Venus: Off
    - Earth: On
    - Mars: Off
    - Jupiter: On

    - Saturn: Off
    - Uranus: Off
    - Neptune: Off
    - Pluto: Off
    - Sun: On
    - Moon: Off
- Initial orbit radius: three times asteroid radius at the target position
- Simulation end time (if the spacecraft does not reach the surface before): target time plus two hours
- OD cutoff time: time between pictures immediately before the maneuver (deltpic)

Appendix C: Additional Results

Additional results are presented in this Appendix. These results were considered of lower importance compared to those presented in Chapter 5, given space and time constraints of this thesis.

C.1 Maneuver Time ahead one hour

In this section, the maneuver time from the baseline profile was moved to an hour before the default maneuver time. There were ten cases that did not land on the asteroid within two hours of the user-defined target time, and half of those cases did not agree with those cases from the baseline case. Figures C.1, C.2, and C.3 display the True Target Error Scatter Plot, Knowledge Error Scatter Plot, and the Final Positions Scatter Plot.

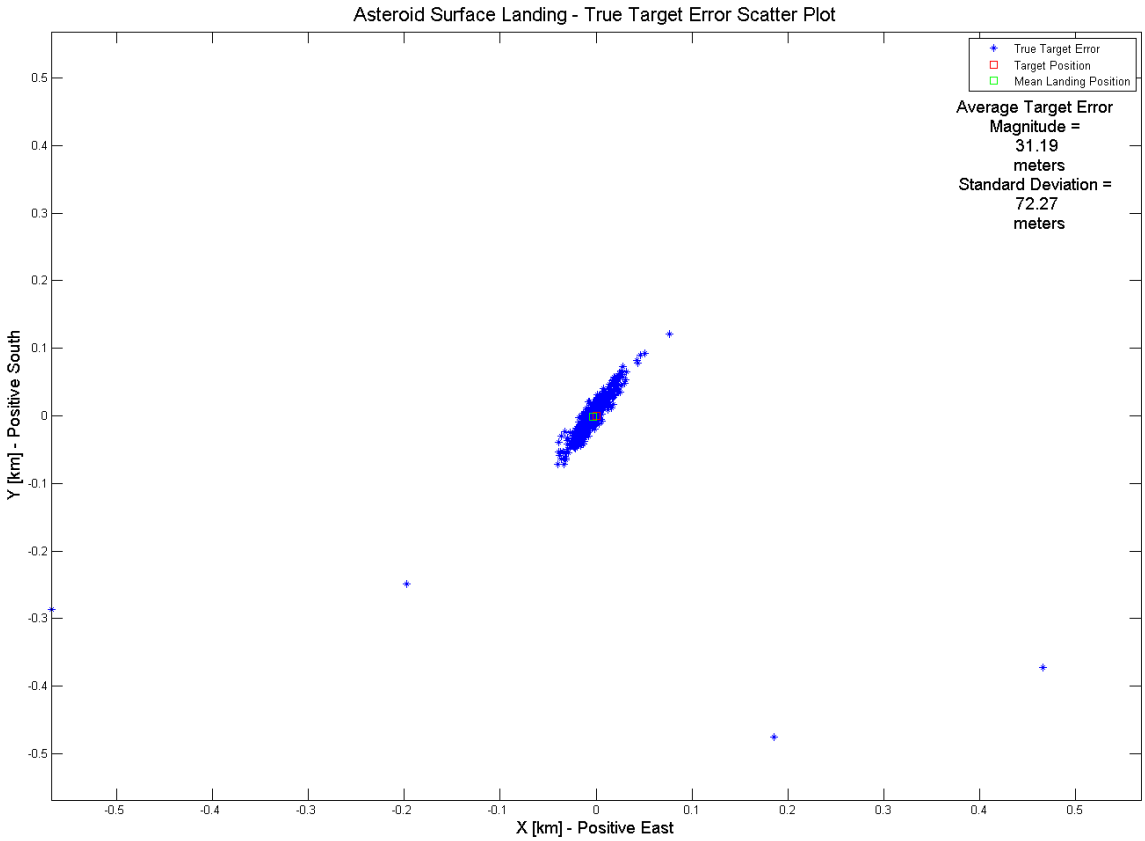

Figure C.1: True Target Error Scatter Plot - Maneuver Time Ahead One Hour

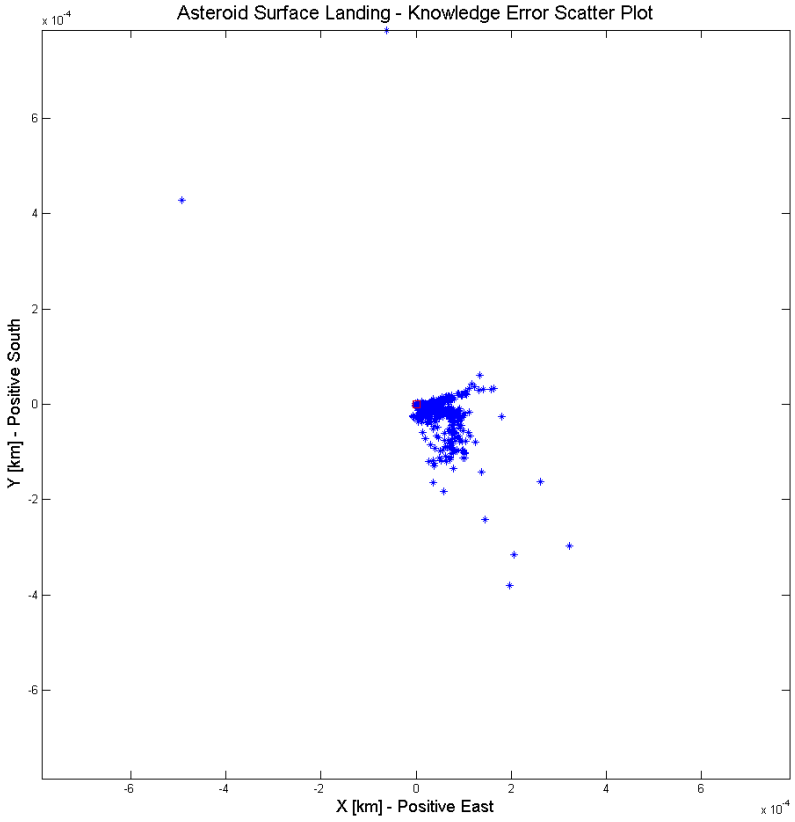

Figure C.2: Knowledge Error Scatter Plot - Maneuver Time Ahead One Hour

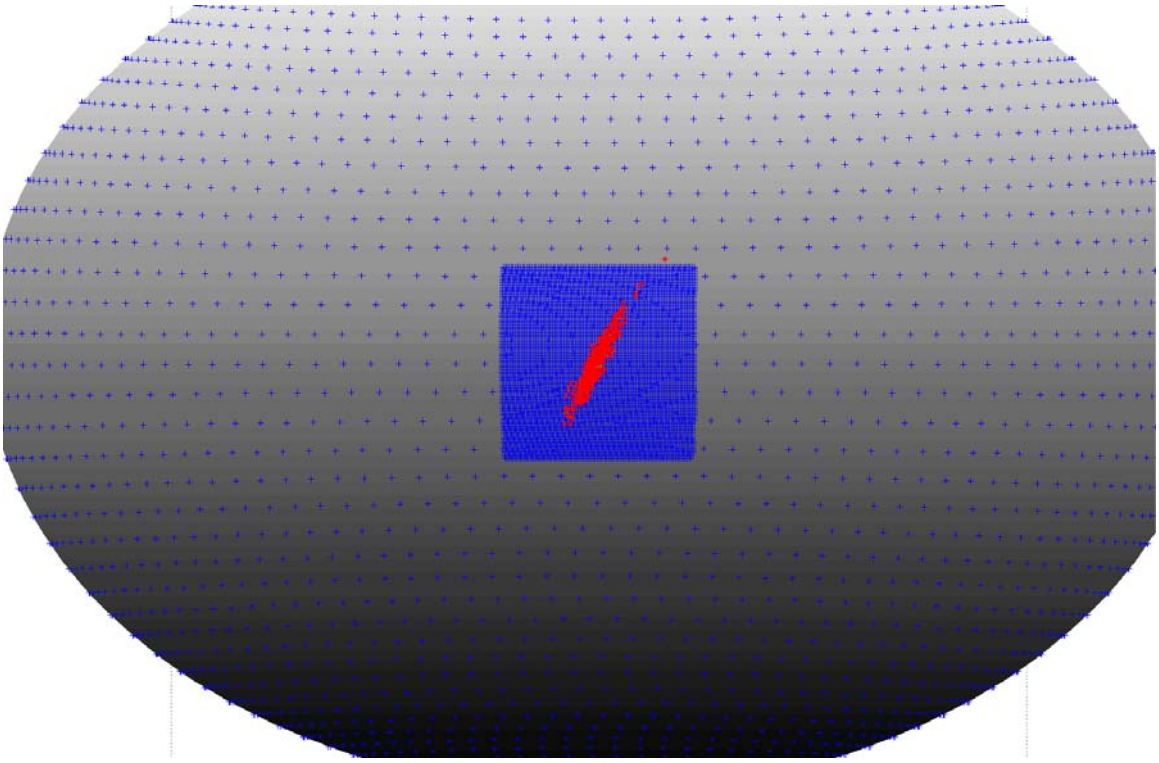

Figure C.3: Final Positions Scatter Plot - Maneuver Time Ahead One Hour

The results in this section suggest that the closer the spacecraft is to the simulation starting position, when it is directly above the target position, the smaller the landing errors will be. The disadvantage of performing the maneuver directly over the target is the increased fuel cost, but because of the very low gravity of the small-body, this increase might be very small and thus worth the greater accuracy for landing.

C.2 MANEUVER TIME BEHIND ONE HOUR

In this section, the maneuver time from the baseline profile was moved to an hour after the default maneuver time. There were 33 cases that did not land on the asteroid within two hours of the user-defined target time, and 14 of those cases did not agree with those cases from the baseline case. Figures C.4, C.5, and C.6 display the True Target Error Scatter Plot, Knowledge Error Scatter Plot, and the Final Positions Scatter Plot.

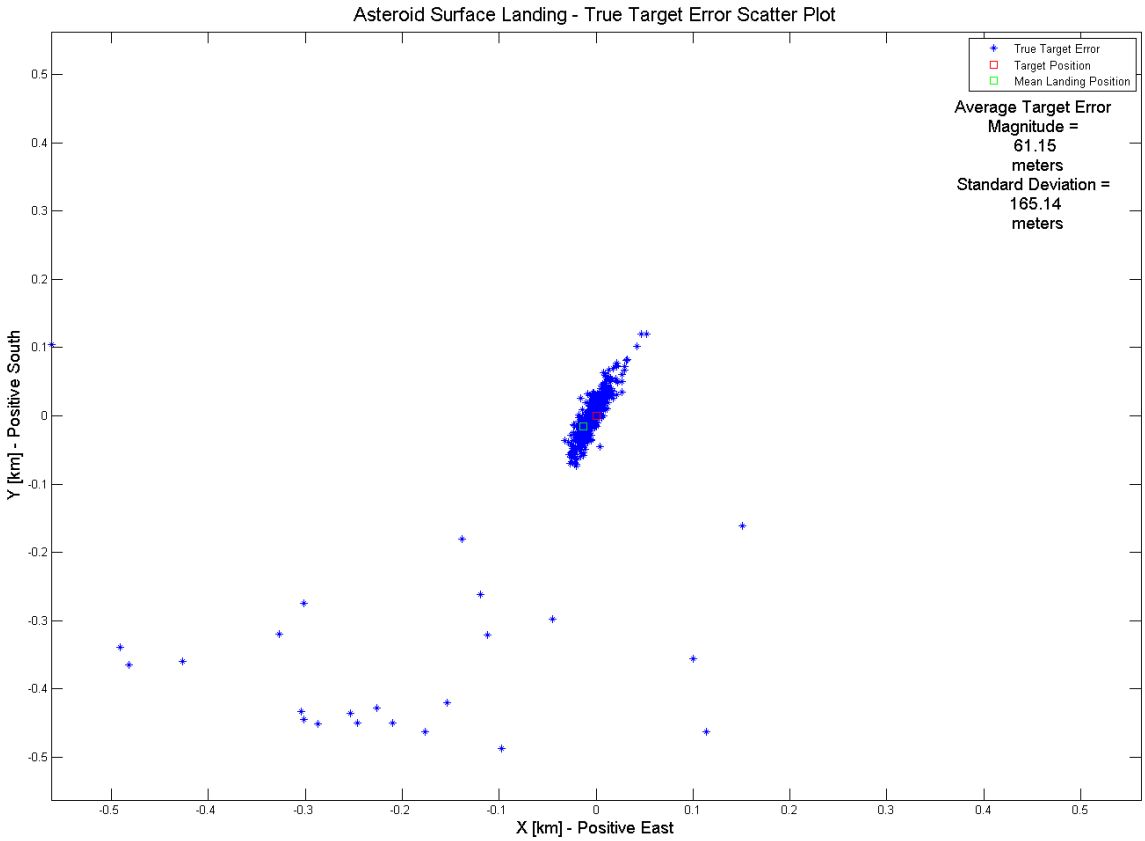

Figure C.4: True Target Error Scatter Plot - Maneuver Time Behind One Hour

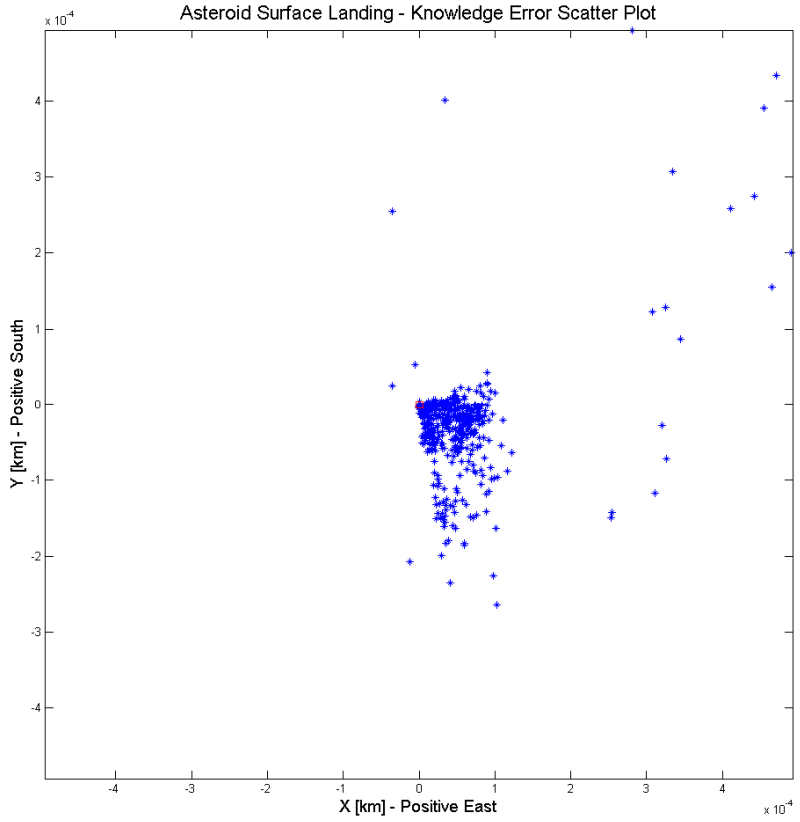

Figure C.5: Knowledge Error Scatter Plot - Maneuver Time Behind One Hour

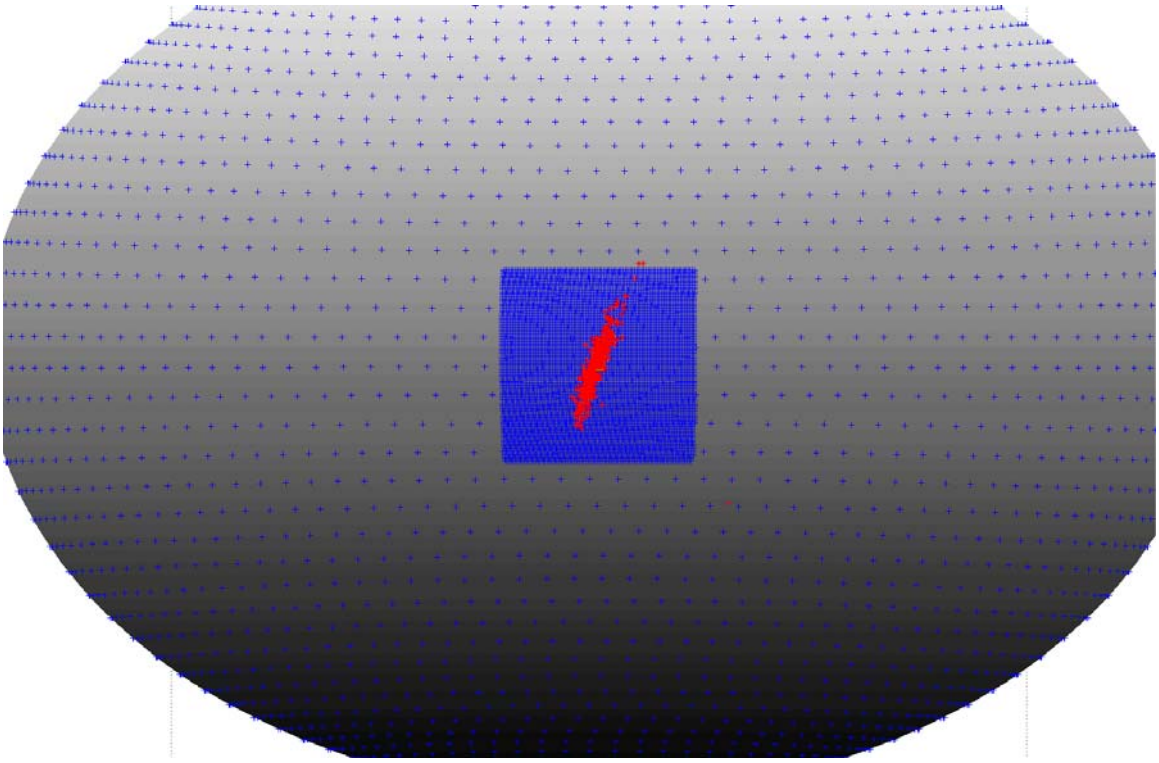

Figure C.6: Final Positions Scatter Plot - Maneuver Time Behind One Hour

C.3 SETTING OD-CUTOFF TO 10 HOURS

In this section, the OD-cutoff time was set to 10 hours. There were 25 cases that did not land on the asteroid within two hours of the user-defined target time, and 5 of those cases did not agree with those cases from the baseline case. Figures C.7, C.8, and C.9 display the True Target Error Scatter Plot, Knowledge Error Scatter Plot, and the Final Positions Scatter Plot.

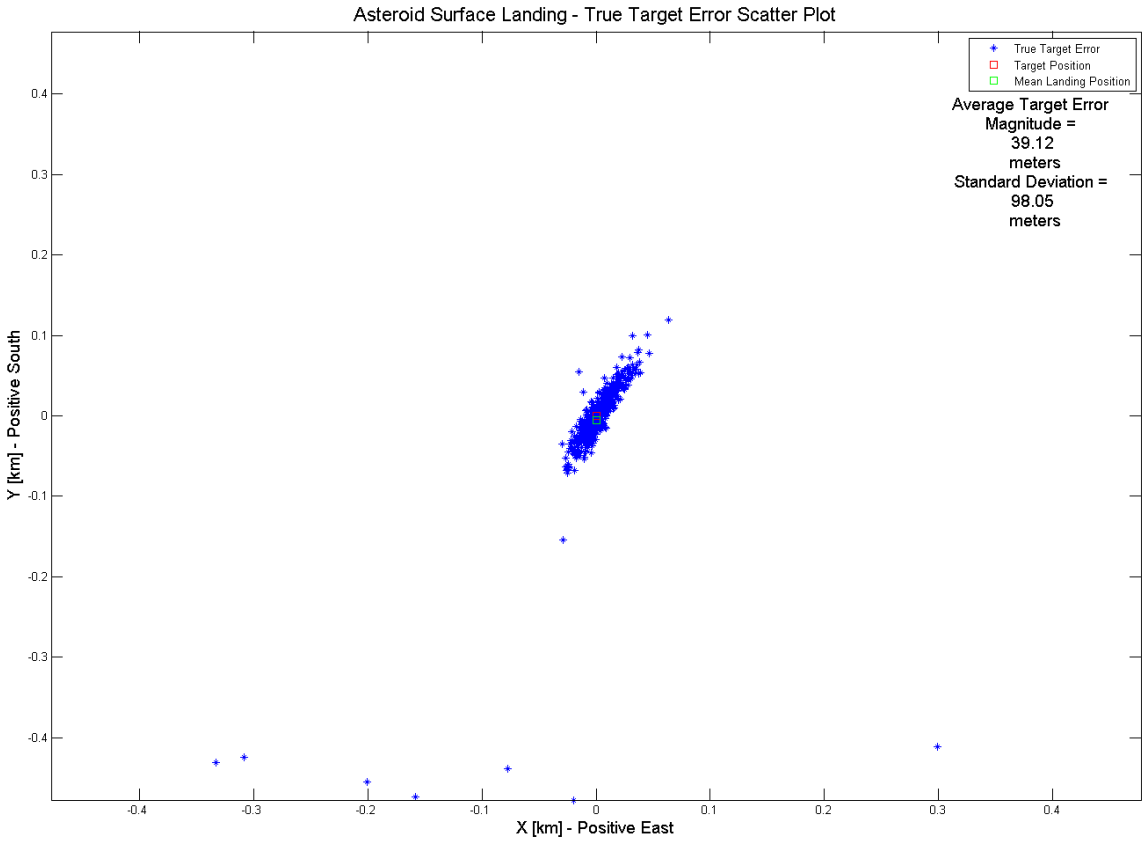

Figure C.7: True Target Error Scatter Plot – Setting OD-Cutoff to 10 hours

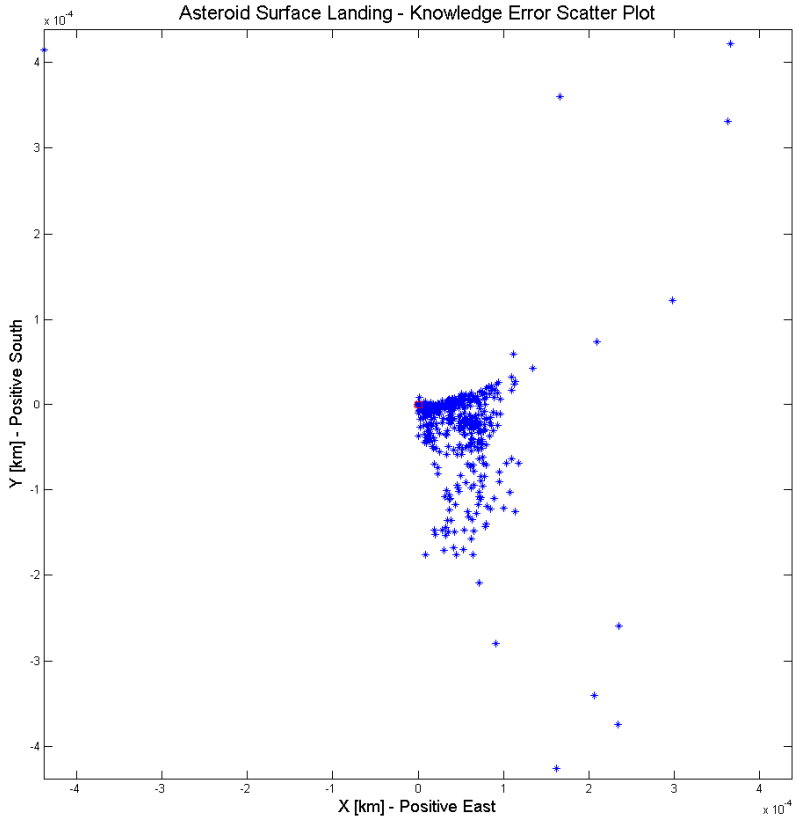

Figure C.8: Knowledge Error Scatter Plot – Setting OD-Cutoff to 10 hours

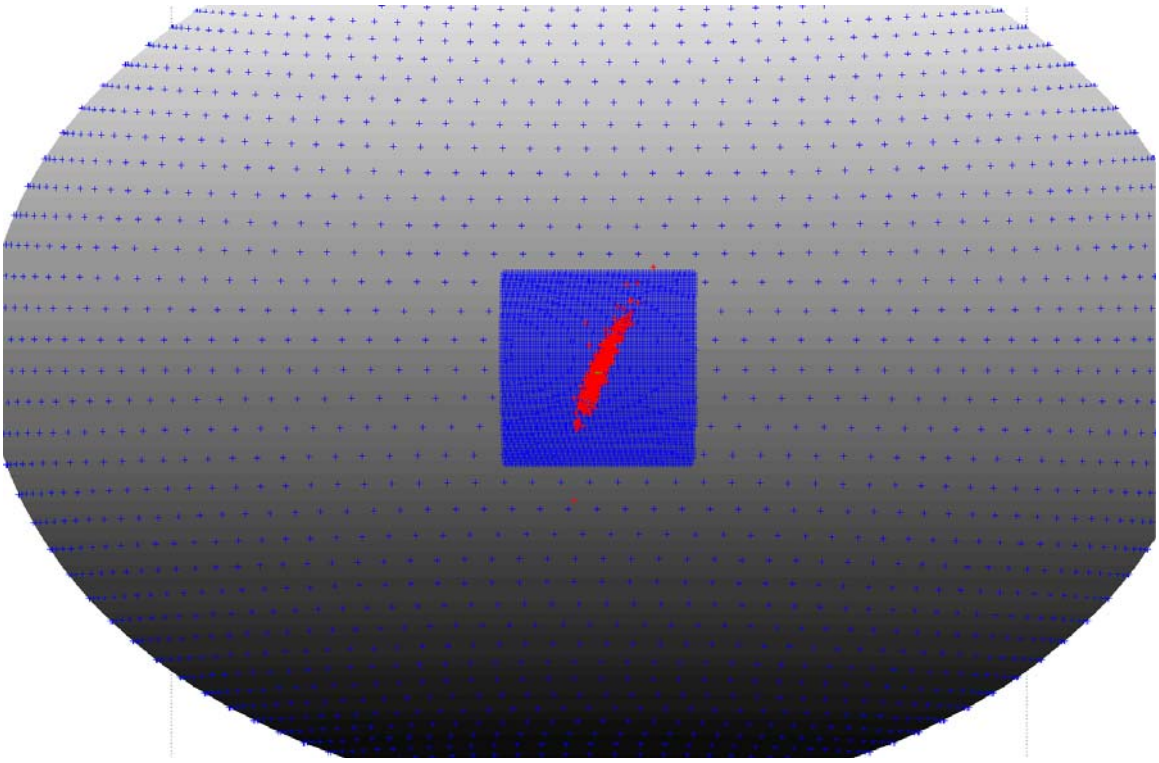

Figure C.9: Final Positions Scatter Plot – Setting OD-Cutoff to 10 hours

C.4 SETTING OD-CUTOFF TO 21 HOURS

In this section, the OD-cutoff time was set to 21 hours. There were 65 cases that did not land on the asteroid within two hours of the user-defined target time. Figures C.10, C.11, and C.12 display the True Target Error Scatter Plot, Knowledge Error Scatter Plot, and the Final Positions Scatter Plot.

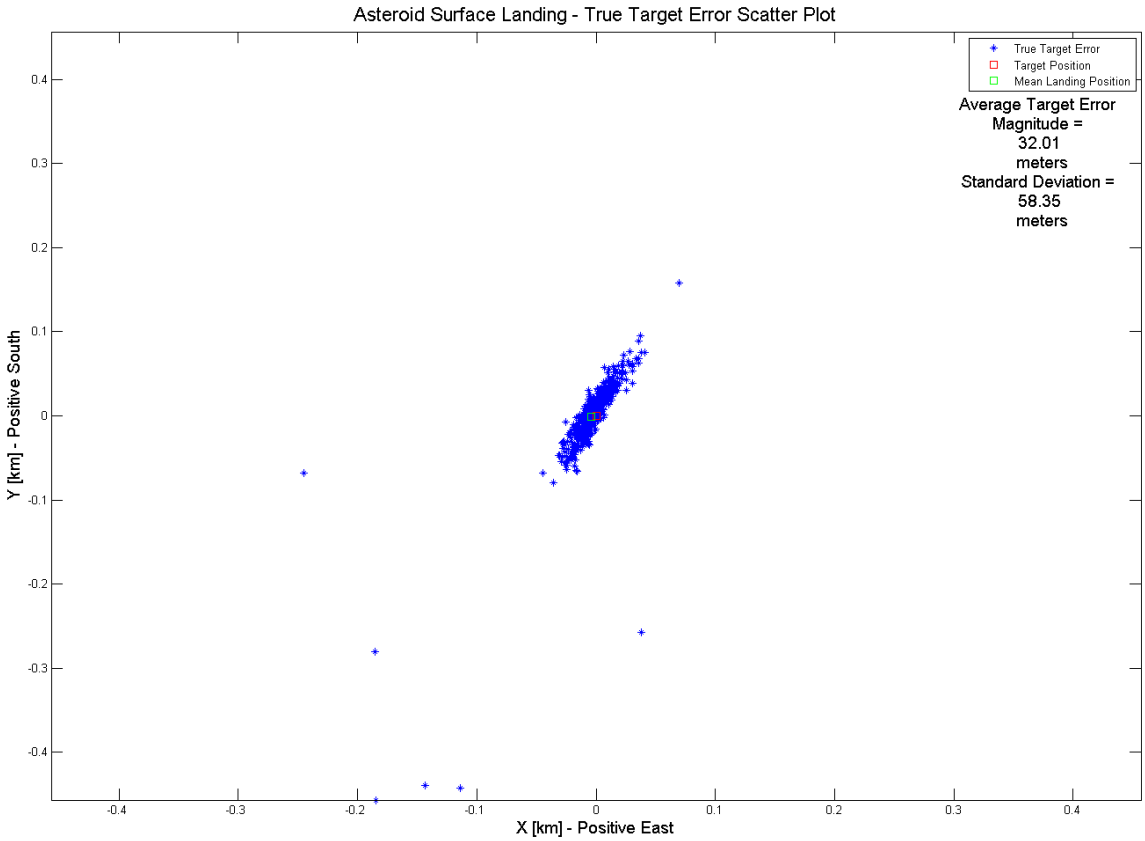

Figure C.10: True Target Error Scatter Plot – Setting OD-Cutoff to 21 Hours

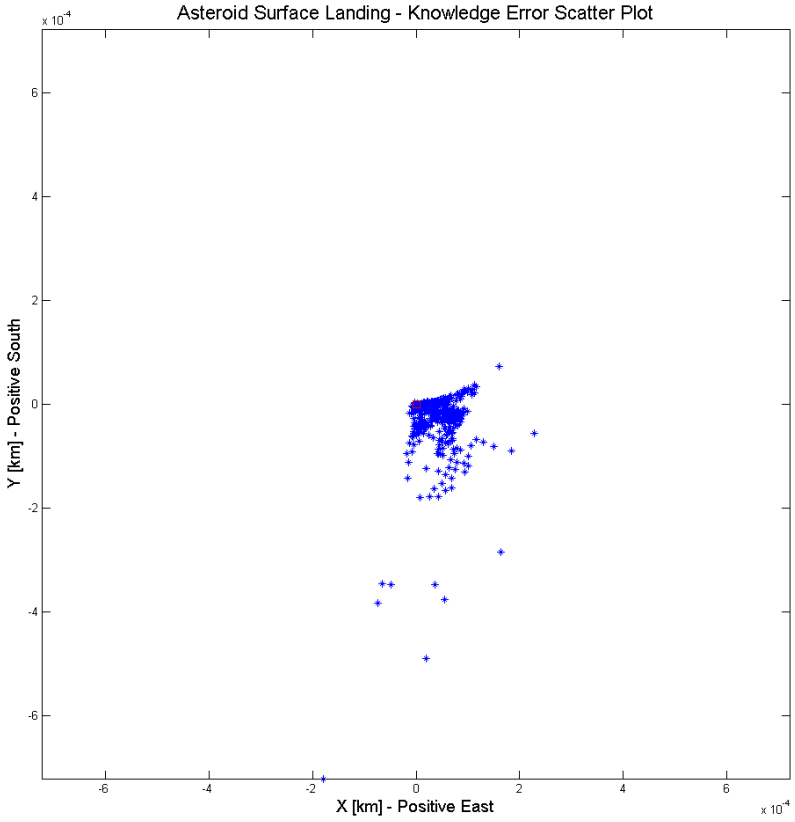

Figure C.11: Knowledge Error Scatter Plot – Setting OD-Cutoff to 21 Hours

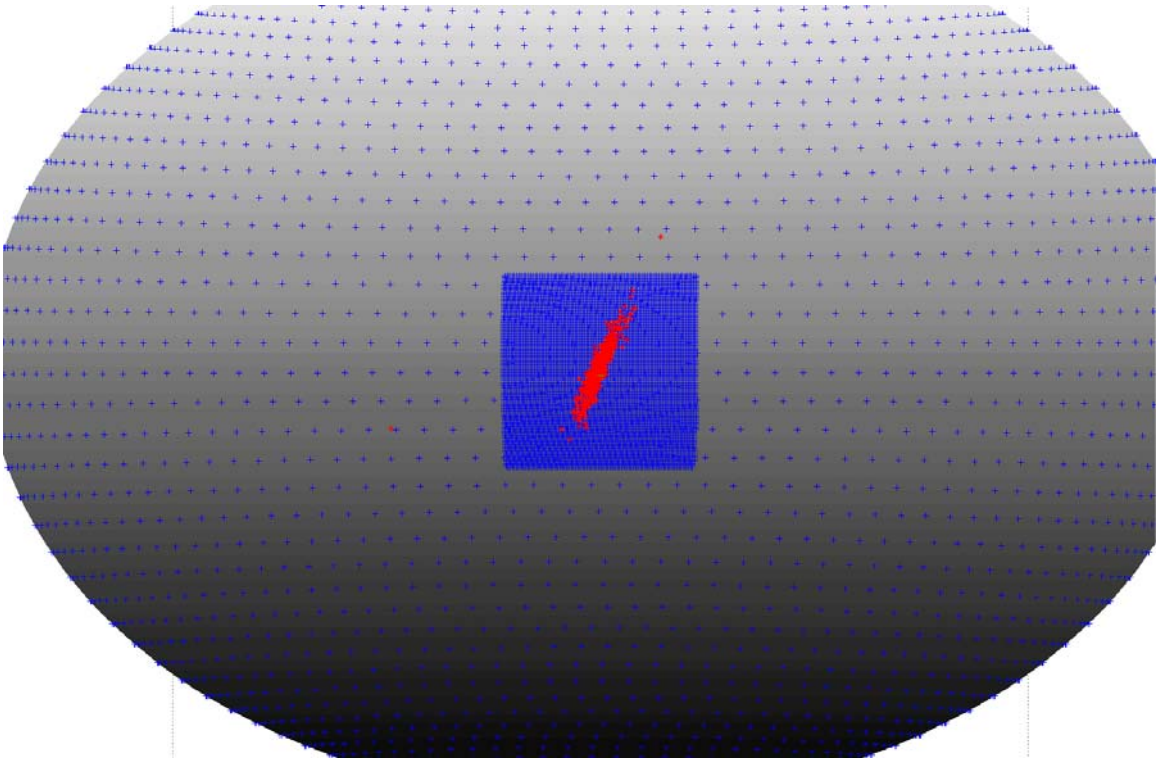

Figure C.12: Final Positions Scatter Plot – Setting OD-Cutoff to 21 Hours

C.5 ADDING OBSERVATIONS EVERY 10 MIN IN THE HOUR BEFORE THE MANEUVER

In this section, observations were added every ten minutes in the hour before the maneuver time. There were 36 cases that did not land on the asteroid within two hours of the user-defined target time. Figures C.13, C.14, and C.15 display the True Target Error Scatter Plot, Knowledge Error Scatter Plot, and the Final Positions Scatter Plot.

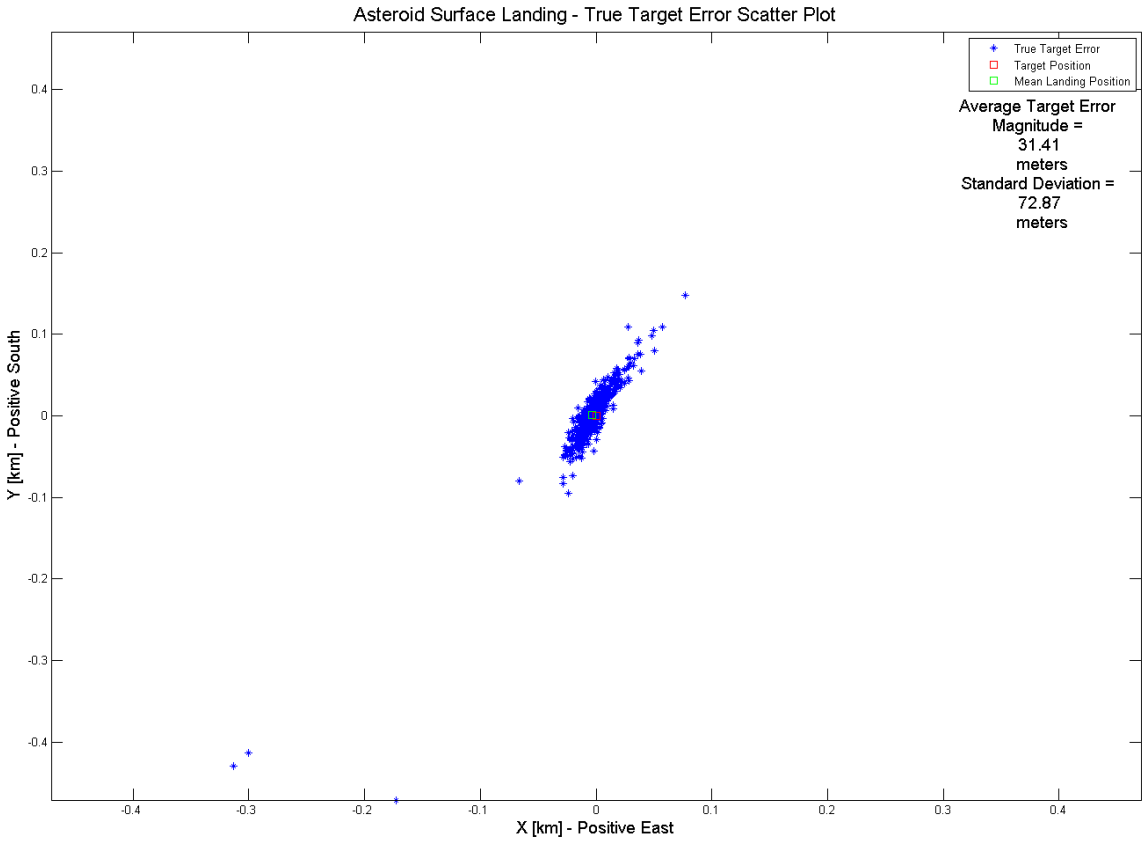

Figure C.13: True Target Error Scatter Plot – Observations Added Every 10 Minutes in the Hour Before Maneuver

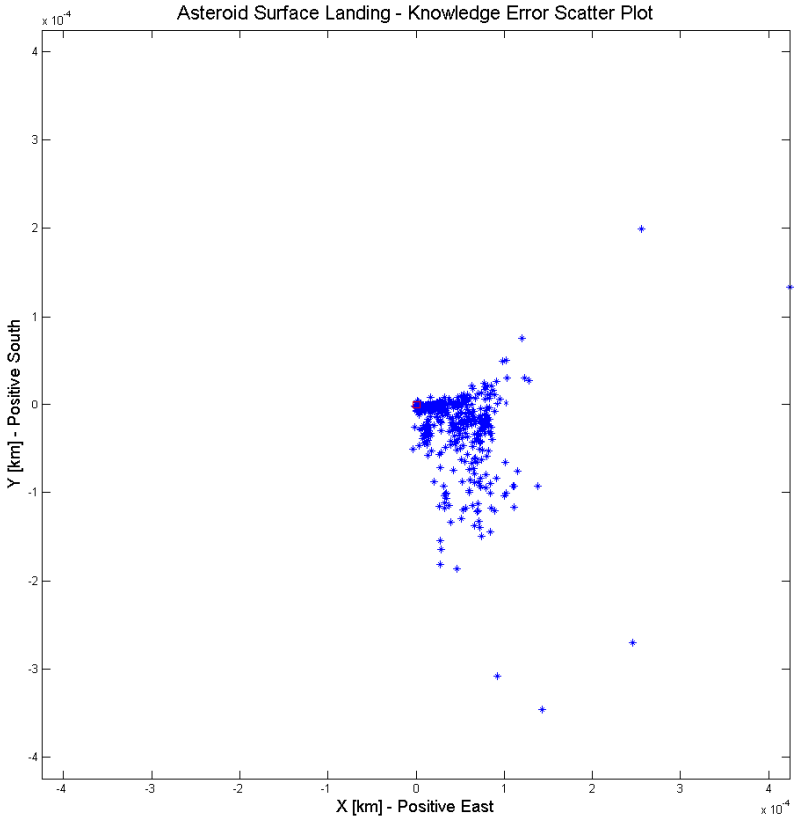

Figure C.14: Knowledge Error Scatter Plot – Observations Added Every 10 Minutes in the Hour Before Maneuver

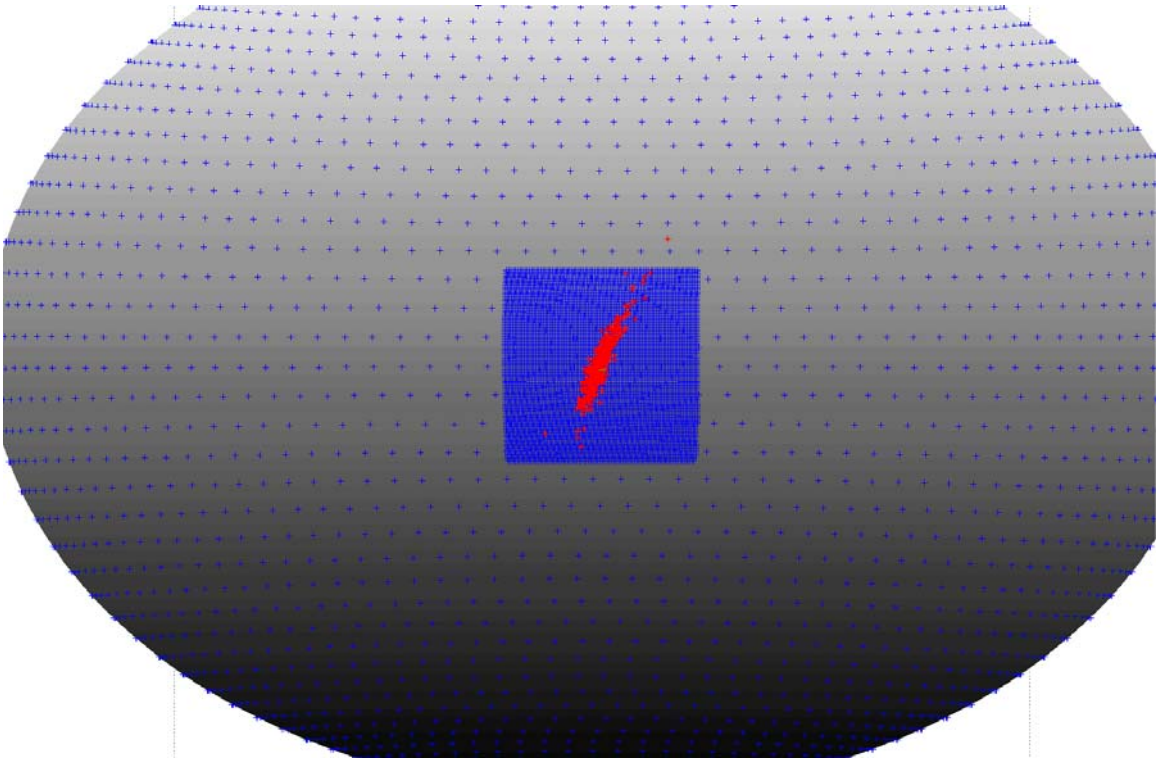

Figure C.15: Final Positions Scatter Plot – Observations Added Every 10 Minutes in the Hour Before Maneuver

C.6 TURNING OFF SLIDING WINDOW

In this section, the sliding window feature was turned off. There were 41 cases that did not land on the asteroid within two hours of the user-defined target time. Figures C.16, C.17, and C.18 display the True Target Error Scatter Plot, Knowledge Error Scatter Plot, and the Final Positions Scatter Plot.

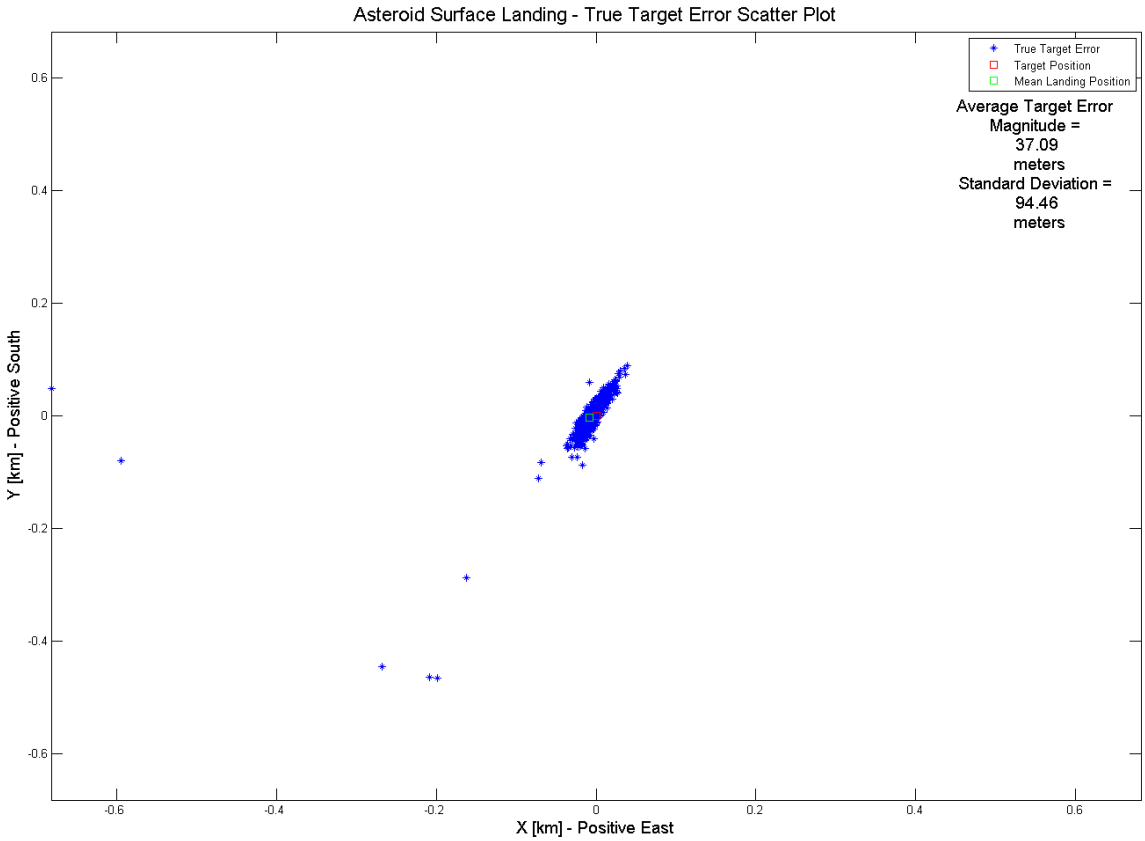

Figure C.16: True Target Error Scatter Plot – Sliding Window Off

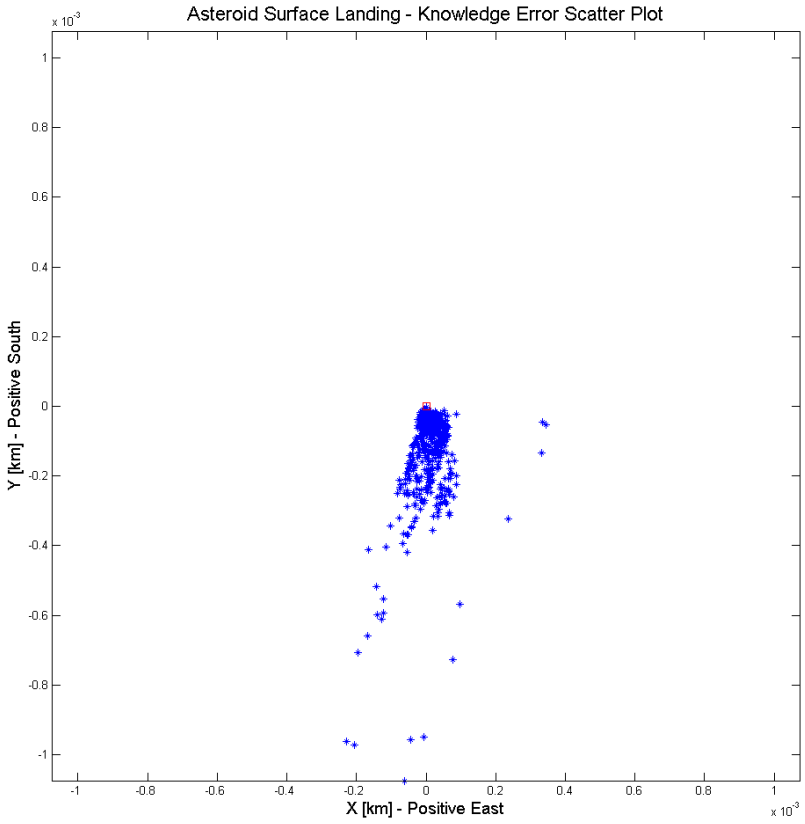

Figure C.17: Knowledge Error Scatter Plot – Sliding Window Off

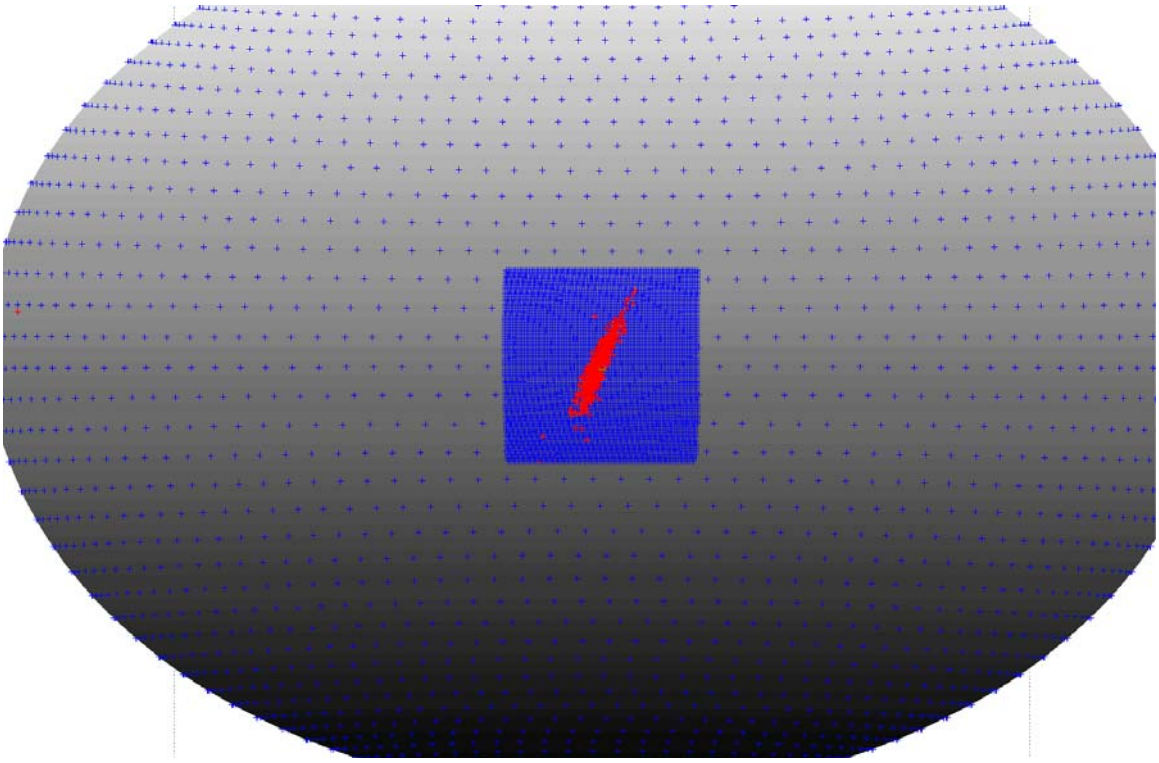

Figure C.18: Final Positions Scatter Plot – Sliding Window Off

C.7 TESTING ONLY ONE ERROR SOURCE AT A TIME

Only 50 cases were computed for each of the following Monte Carlo simulations. Each of the following cases had all error sources removed except those specified in the title of the section.

C.7.1 Adding: Initial State Error Only

In this section, only the initial state error was implemented in the simulation. Thus attitude error, observation error, and maneuver execution error were removed, and the three perturbations in the nominal model were turned on. There was one case that did not land on the asteroid within two hours of the user-defined target time. Figures C.19, C.20, and C.21 display the True Target Error Scatter Plot, Knowledge Error Scatter Plot, and the Final Positions Scatter Plot.

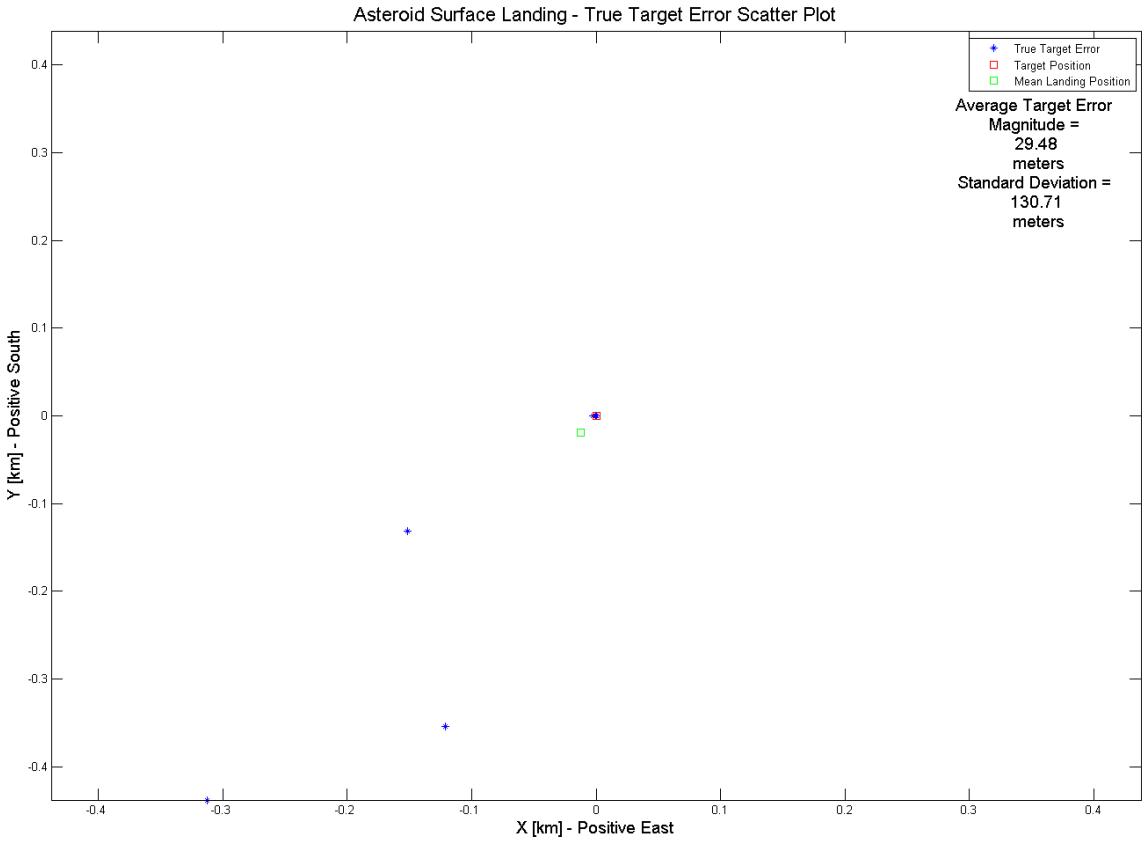

Figure C.19: True Target Error Scatter Plot – Initial State Error Only

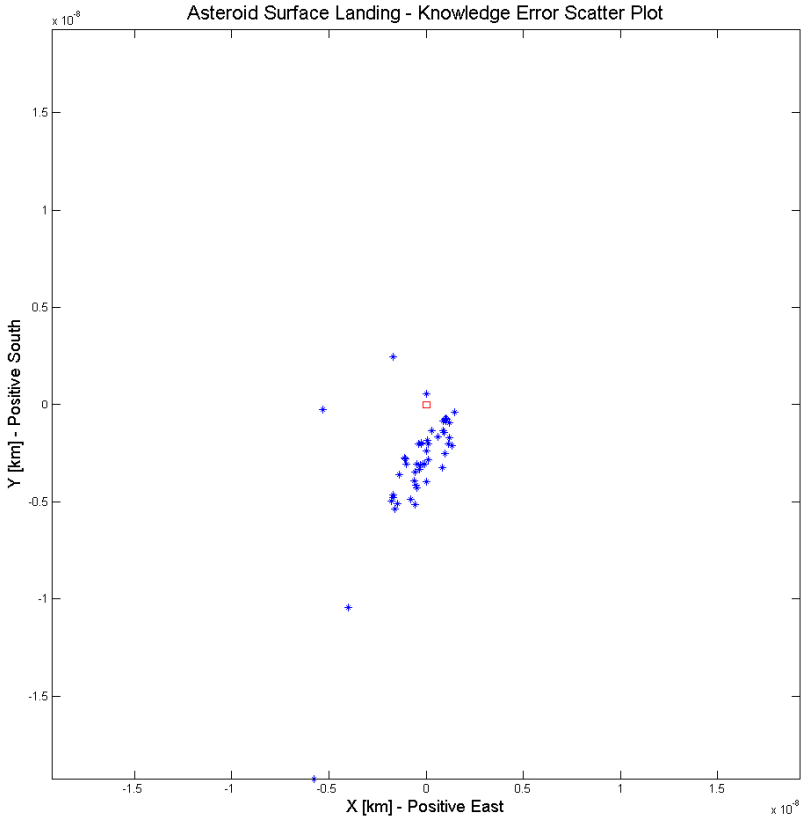

Figure C.20: Knowledge Error Scatter Plot – Initial State Error Only

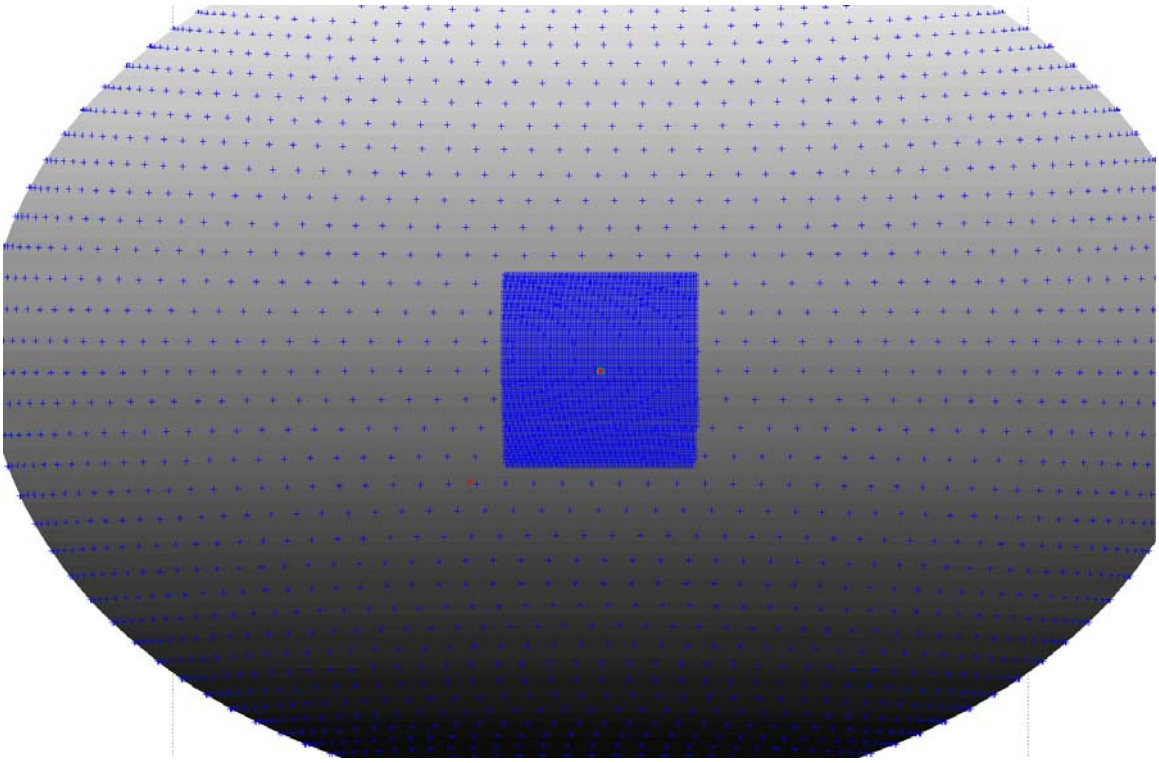

Figure C.21: Final Positions Scatter Plot – Initial State Error Only

C.7.2 Adding: Observation Error Only

In this section, only the observation error was implemented in the simulation. Thus attitude error, initial state error, and maneuver execution error were removed, and the three perturbations in the nominal model were turned on. There were no cases that did not land on the asteroid within two hours of the user-defined target time. Figures C.22, C.23, and C.24 display the True Target Error Scatter Plot, Knowledge Error Scatter Plot, and the Final Positions Scatter Plot.

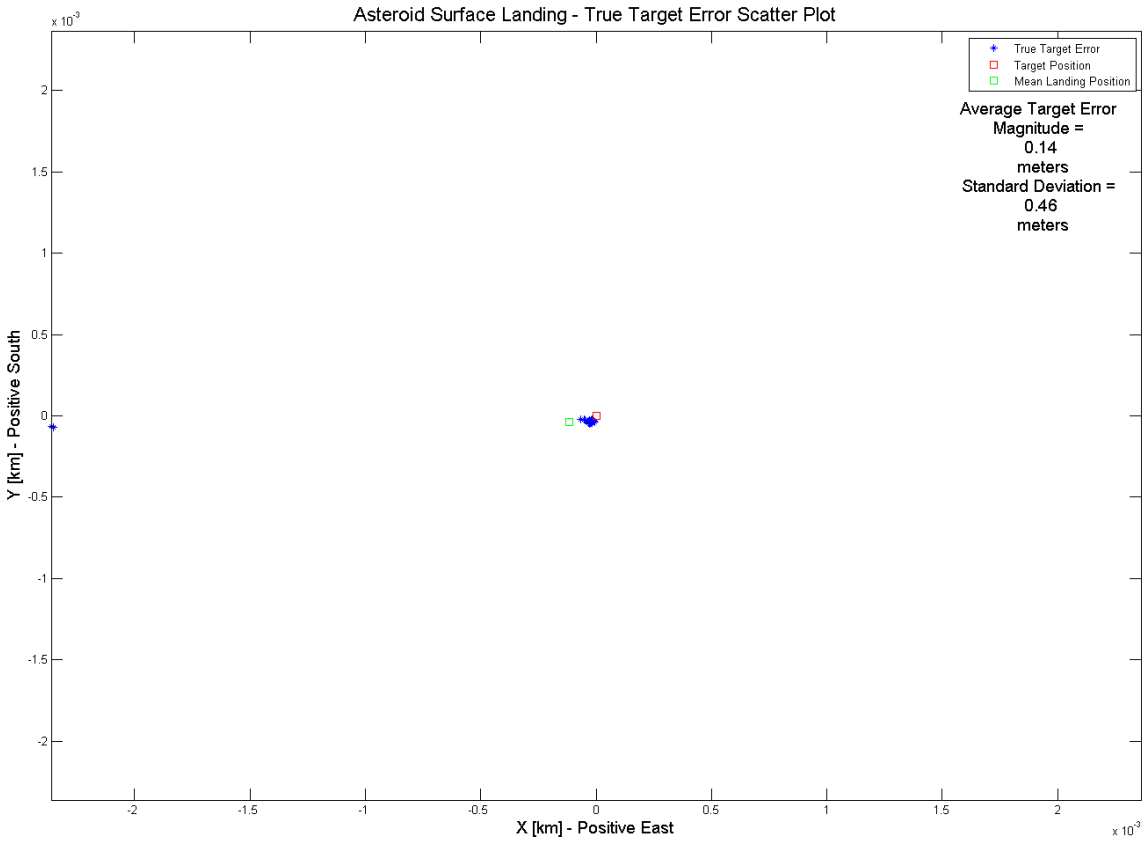

Figure C.22: True Target Error Scatter Plot – Observation Error Only

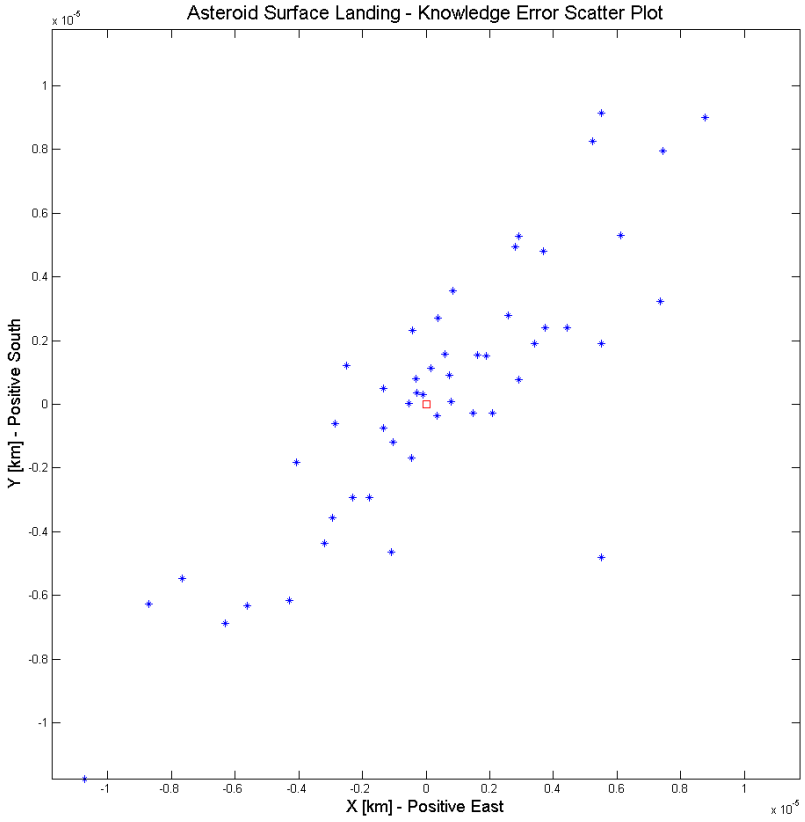

Figure C.23: Knowledge Error Scatter Plot – Observation Error Only

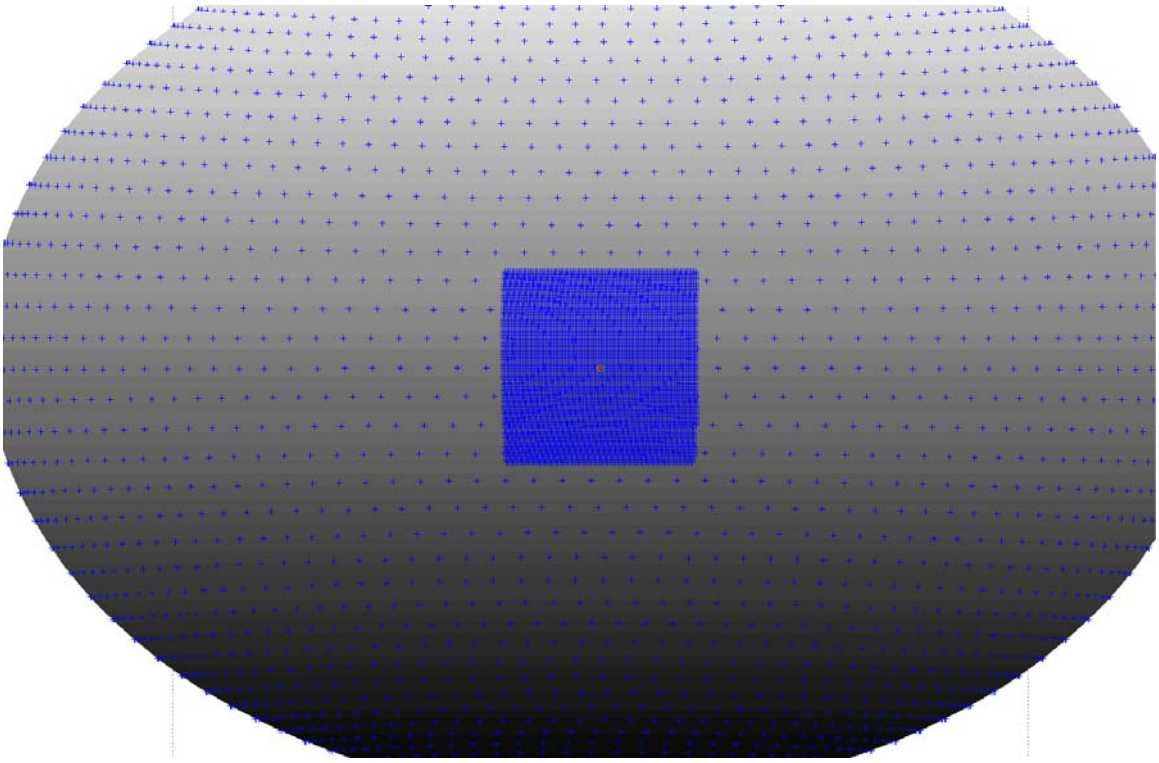

Figure C.24: Final Positions Scatter Plot – Observation Error Only

C.7.3 Adding: Maneuver Execution Error Only

In this section, only the maneuver execution error was implemented in the simulation. Thus attitude error, initial state error, and observation error were removed, and the three perturbations in the nominal model were turned on. There were no cases that did not land on the asteroid within two hours of the user-defined target time. Figures C.25, C.26, and C.27 display the True Target Error Scatter Plot, Knowledge Error Scatter Plot, and the Final Positions Scatter Plot.

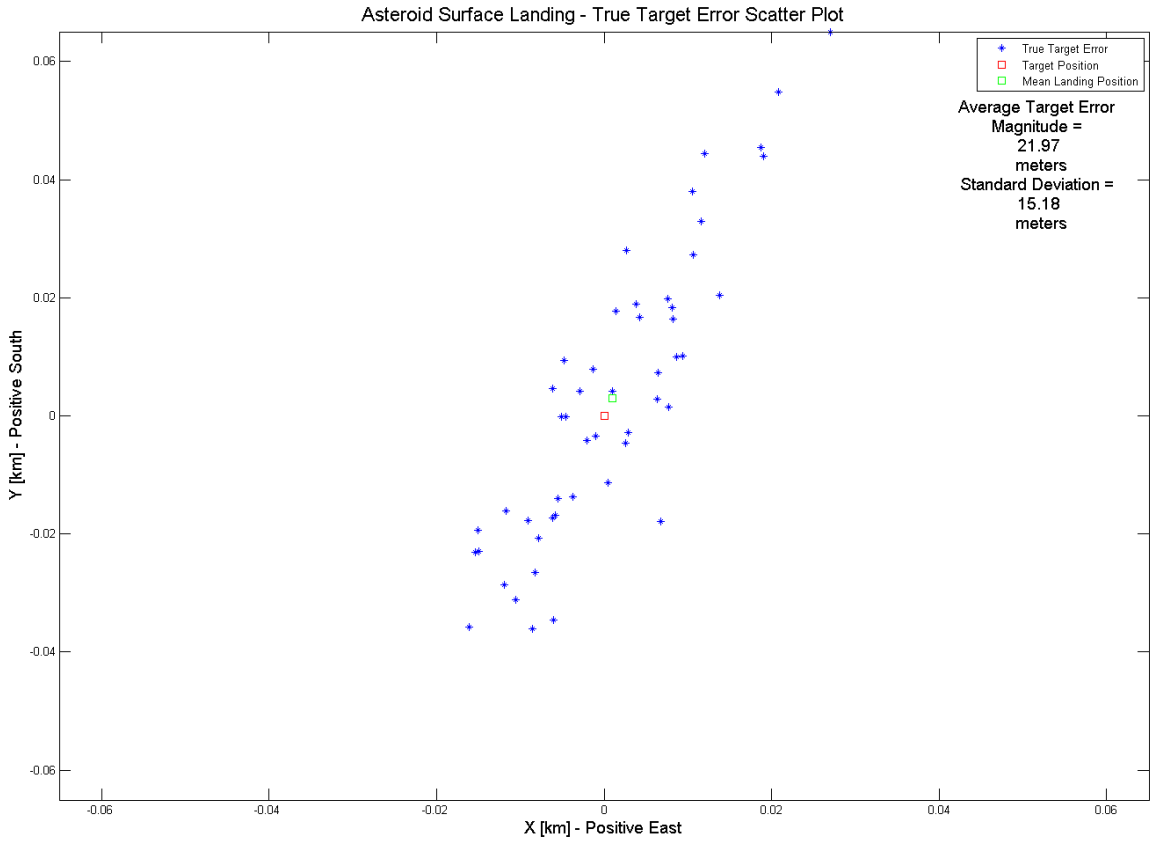

Figure C.25: True Target Error Scatter Plot – Maneuver Execution Error Only

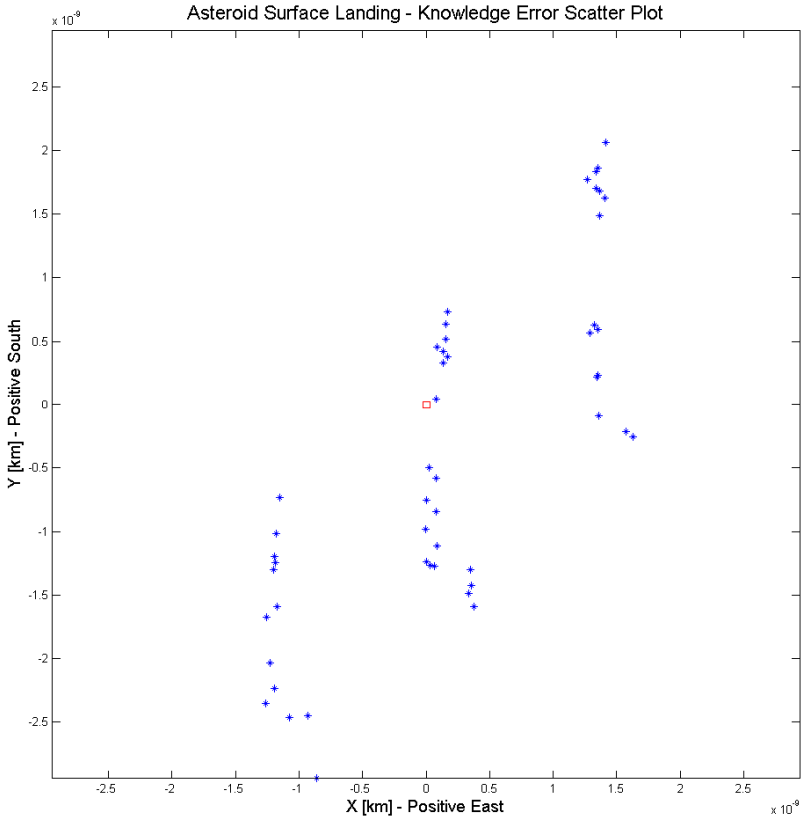

Figure C.26: Knowledge Error Scatter Plot – Maneuver Execution Error Only

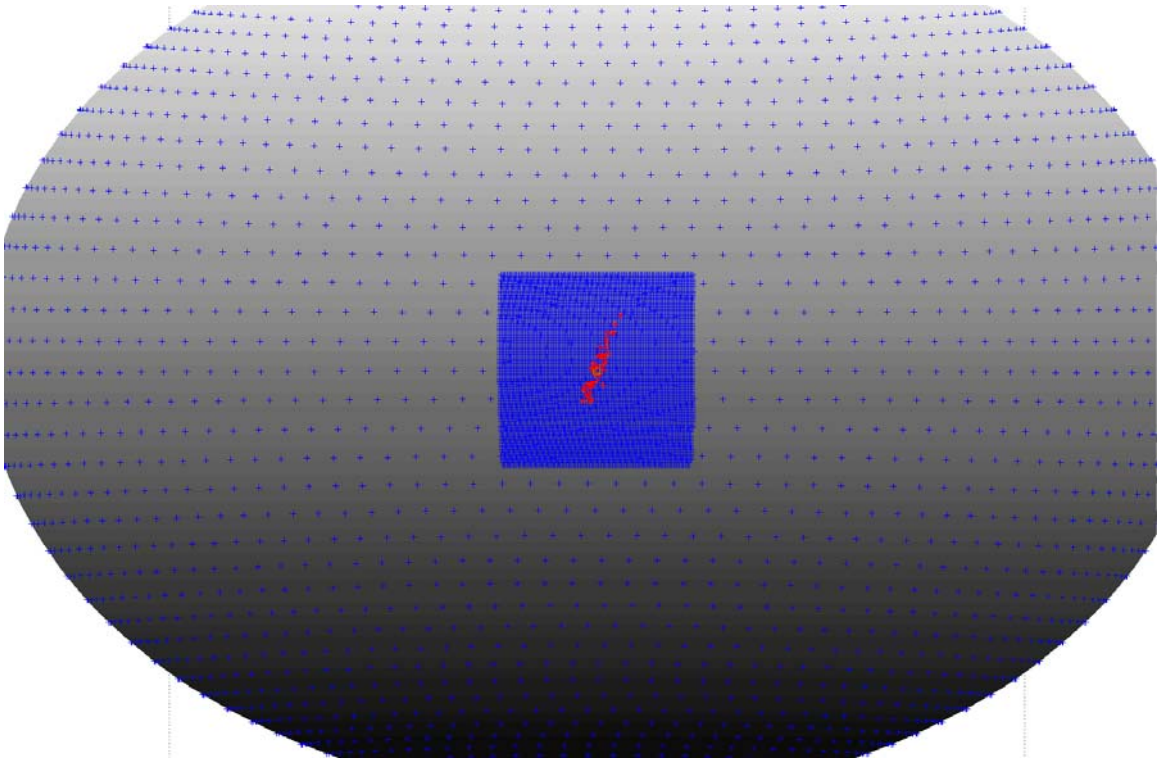

Figure C.27: Final Positions Scatter Plot – Maneuver Execution Error Only

C.7.4 Adding: Attitude Error Only

In this section, only the attitude error was implemented in the simulation. Thus maneuver execution error, initial state error, and observation error were removed, and the three perturbations in the nominal model were turned on. There were no cases that did not land on the asteroid within two hours of the user-defined target time. Figures C.28, C.29, and C.30 display the True Target Error Scatter Plot, Knowledge Error Scatter Plot, and the Final Positions Scatter Plot.

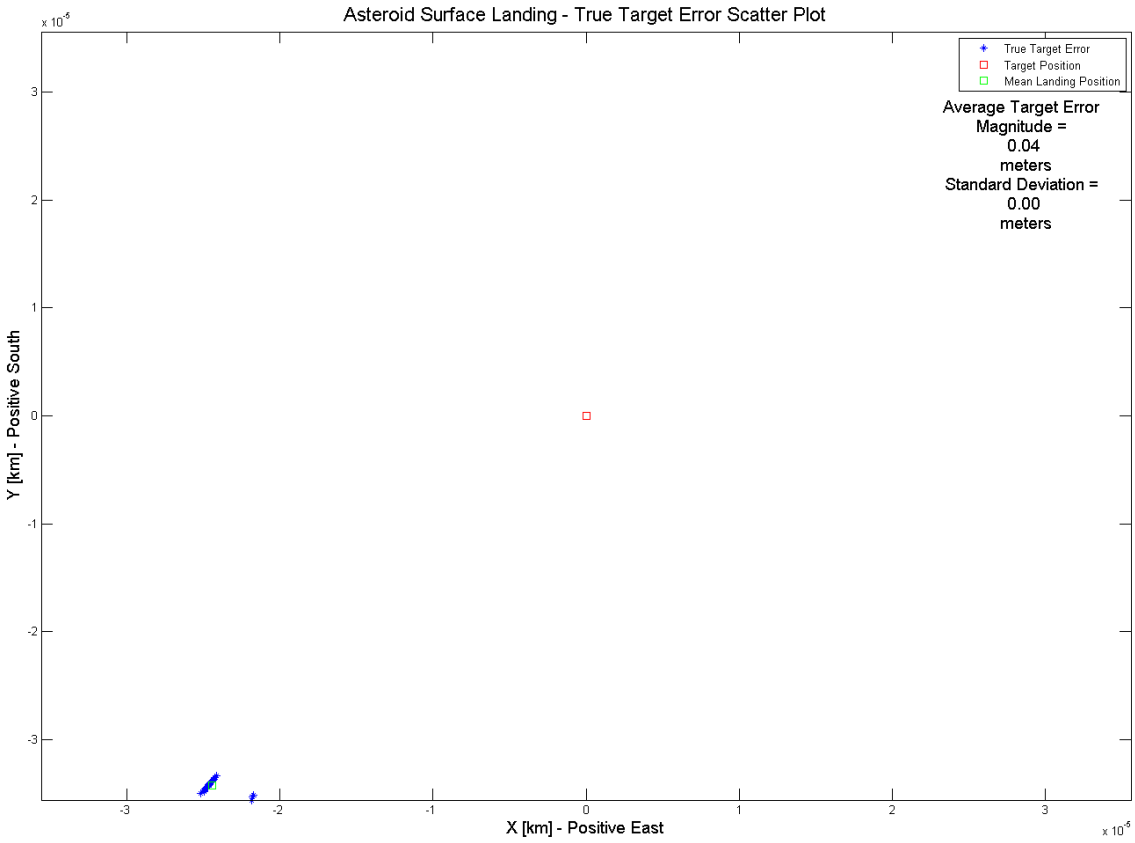

Figure C.28: True Target Error Scatter Plot – Attitude Error Only

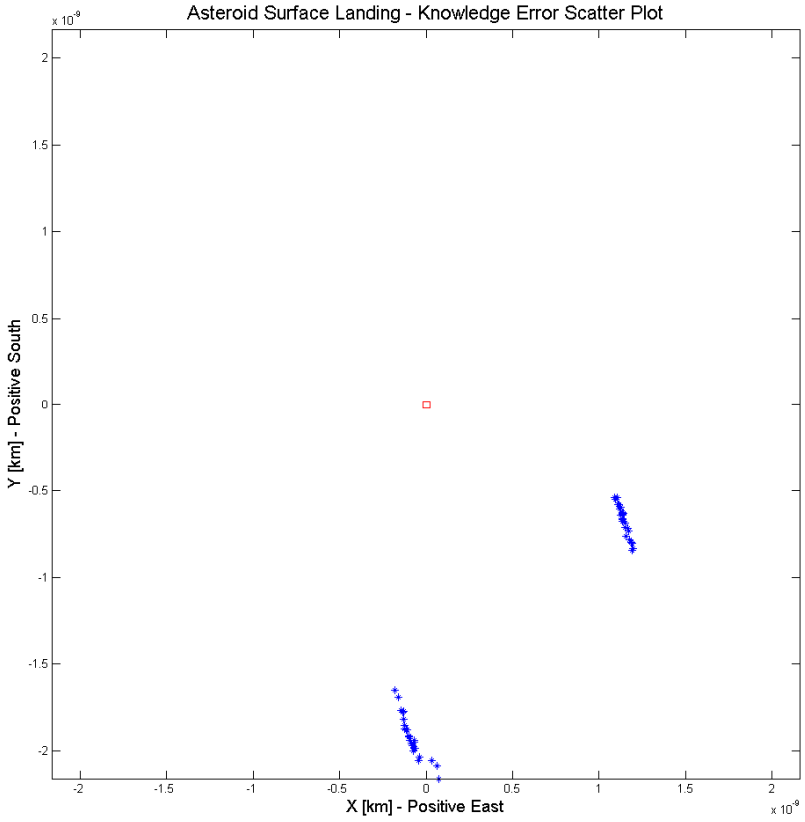

Figure C.29: Knowledge Error Scatter Plot – Attitude Error Only

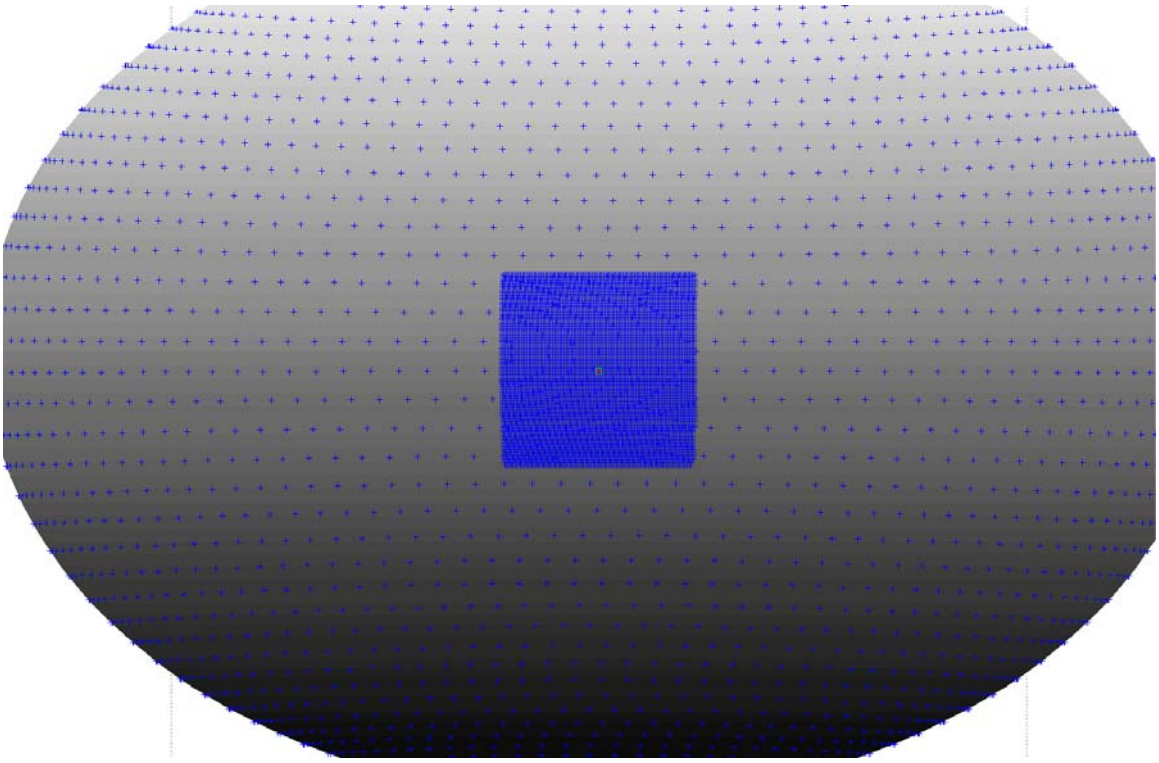

Figure C.30: Final Positions Scatter Plot – Attitude Error Only

C.7.5 Adding: No perturbations on the nominal model only

In this section, the three perturbations in the nominal model were turned off. Thus the maneuver execution error, initial state error, attitude error, and observation error were removed. There were no cases that did not land on the asteroid within two hours of the user-defined target time. Figures C.31, C.32, and C.33 display the True Target Error Scatter Plot, Knowledge Error Scatter Plot, and the Final Positions Scatter Plot. All of the points in the True Target Error Scatter Plot seem to be in the same region, an interesting feature that suggests a bias.

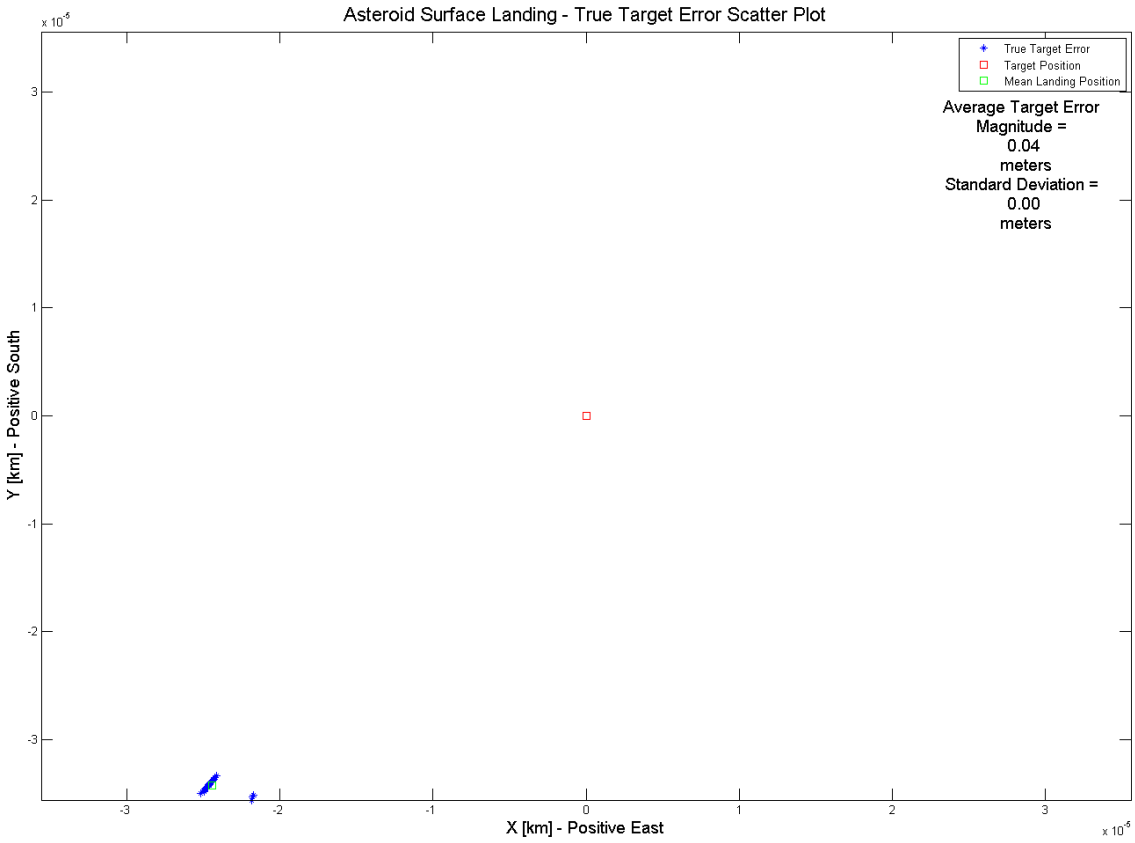

Figure C.31: True Target Error Scatter Plot – No Perturbations on the Nominal Model

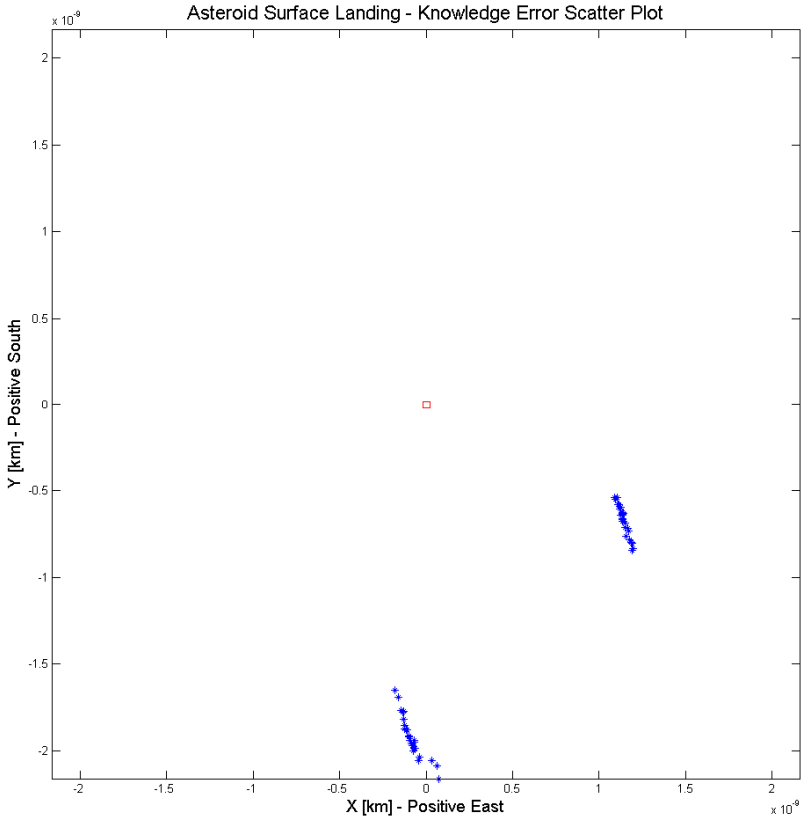

Figure C.32: Knowledge Error Scatter Plot – No Perturbations on the Nominal Model

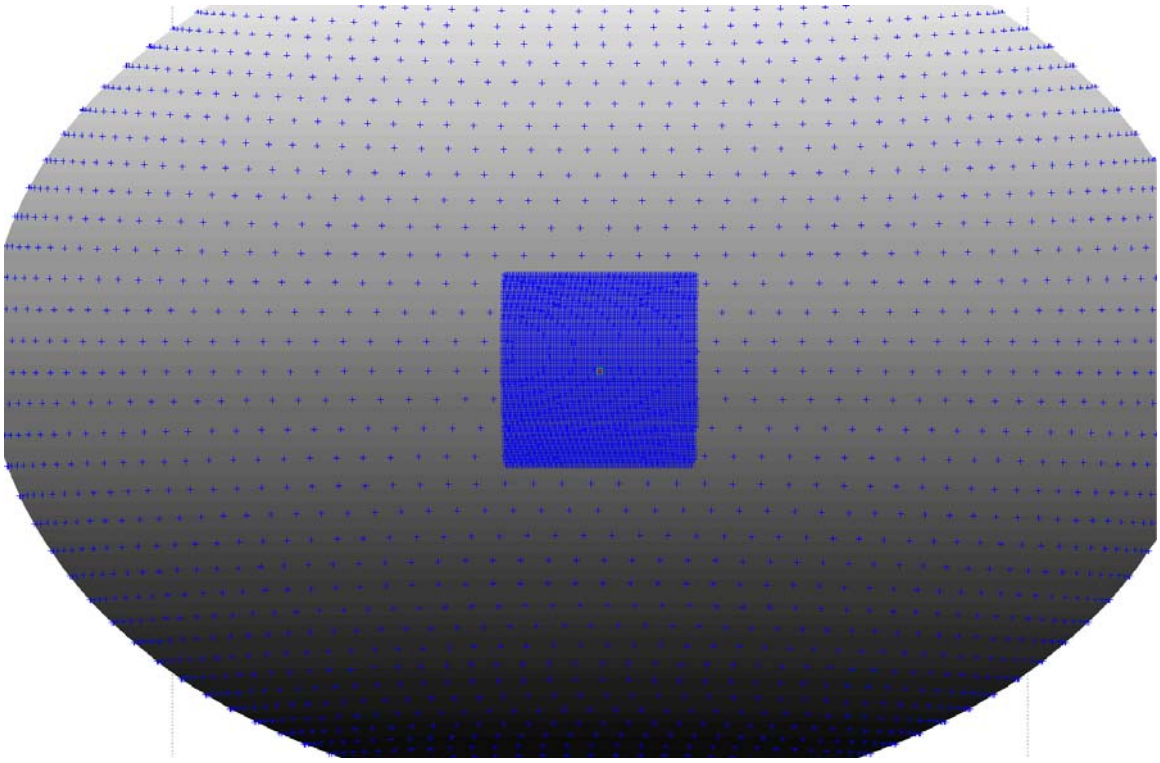

Figure C.33: Final Positions Scatter Plot – No Perturbations on the Nominal Model

List of acronyms

AON	Autonomous Optical Navigation
CSR	Center for Space Research
GPS	Global Positioning System
IAU	International Astronomical Union
JAXA	Japanese Aerospace Exploration Agency
JPL	NASA Jet Propulsion Laboratory
LIDAR	LIght Detection And Ranging
NASA	National Aeronautics and Space Administration
RMS	Root-Mean-Square
RTN	Radial-Transverse-Normal
SANT	Small-Body Autonomous Navigation Toolset
SEZ	South-East-Up
SRP	Solar Radiation Pressure
STM	State Transition Matrix
UT	University of Texas

Bibliography

[1] Bhaskaran, S., Chesley, S, Park, R., Petropoulos, A., Sims, J., Yeomans, D., Haw R., "Navigation Challenges of a Kinetic Energy Asteroid Deflection Spacecraft," *AIAA 2008-6254*, AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Honolulu, Hawaii, August 18-21, 2008.

[2] Bhaskaran, S., Desai, S. D., Dumont, P. J., Kennedy, B. M., Null, G. W., Owen, W. M., Riedel, J. E., Synnott, S. P., and Werner, R.A., "Orbit Determination Performance Evaluation of the Deep Space 1 Autonomous Navigation System," *AAS Paper 98-193*, AAS/AIAA Spaceflight Mechanics Meeting, Monterrey, CA, February 1998.

[3] Bhaskaran, S., Riedel, J. E., Synnott, S. P., "Autonomous Optical Navigation for Interplanetary Missions," *Space Sciencecraft Control and Tracking in the New Millennium*, E. Kane Casani, Mark A. Vander Does, Editors, Proc. SPIE 2810, pages 32-43, 1996.

[4] Bhaskaran, S., Riedel, J. E., Synnott, S. P., "Autonomous Target Tracking of Small Bodies During Flybys," *AAS 04-236*, AAS/AIAA Spaceflight Mechanics Conference, Maui, HI, February 8-12, 2004.

[5] Bhaskaran, S., Riedel, J. E., Synnott, S. P., "Demonstration of Autonomous Orbit Determination Around Small Bodies," *AAS 95-387*, AAS/AIAA Astrodynamics Specialist Conference, Halifax, Nova Scotia, Canada, August 14-17, 1995.

[6] Bhaskaran, S., Riedel, J. E., Synnott, S. P., Wang, T. C., "The Deep Space 1 Autonomous Navigation System: A Post-flight analysis," *AIAA 2000-3935*, AIAA/AAS Astrodynamics Specialist Conference, Denver, CO, August 14-17, 2000.

[7] Bhaskaran, S., Riedel, J. E., Synnott, S. P., "Autonomous Nucleus Tracking for Comet/Asteroid Encounters: The STARDUST Example," *AAS 97-628*, AAS/AIAA Astrodynamics Conference, Sun Valley, ID, August 1997.

[8] Chadwick, C., Miller, L. J., "An Overview of the ADAM maneuver Analysis System," *AAS Paper 83-411*, pages 20-22, AAS/AIAA Astrodynamics Specialist Conference, Lake Placid, New York, August 22-25,1983.

[9] Davies, M. E., et. al., "Report of the IAU/IAG/COSPAR Working Group on Cartographic Coordinates and Rotational Elements of the Planets and Satellites: 1994", *Celestial Mechanics and Dynamical Astronomy*, vol 63, pp 127-148, 1996.

[10] Gaskell, R., Barnouin-Jha, O., Scheeres, D. , Mukai, T. , Hirata, N. , Abe, S., Saito, J., Ishiguro, M., Kubota, T., Hashimoto, T., Kawaguchi, J., Yoshikawa, M., Shirakawa, K., Kominato, T., "Landmark Navigation Studies and Target Characterization in the Hayabusa Encounter with Itokawa," *Paper AAS 06-6660*, AAS/AIAA Astrodynamics Specialists Conf., Keystone, CO, Aug 2006.

[11] Gates, C.R., "A Simplified Model of Midcourse Maneuver Execution Errors," *JPL Technical Report 32-504*, JPL, Pasadena, CA, October 15, 1963.

[12] Hayabusa Spacecraft description. National Aeronautics and Space Administration Planetary Data system. Retrieved 2009 from website: http://starbrite.jpl.nasa.gov/pds/viewHostProfile.jsp?INSTRUMENT_HOST_ID=HAY

[13] Kawaguchi, J., Hashimoto, T., Kubota, T., Sawai, S., Fujii, G., "Autonomous optical guidance and navigation strategy around a small body," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 5, pages 1010-1017, September - October, 1997.

[14] Kominato, T., Matsuoka, M., Uo, M., Hashimoto, T., Kawaguchi, J., "Optical Hybrid Navigation in Hayabusa – Approach, Station Keeping & Hovering," *Paper AAS 06-209*, AAS/AIAA Spaceflight Mechanics Conference, Tampa, FL, January 22-26, 2006.

[15] Kubitschek, D., et. al., "Deep Impact Autonomous Navigation: The Trials of Targeting the Unknown," *AAS 06-081*, Advances in the Astronomical Sciences, Proceedings of the Annual AAS Guidance and Control Conference, 2006.

[16] Mastrodemos, N., Kubitschek, D., Werner, R., Kennedy, B., Synnott, S., Null, G., Riedel, J., Bhaskaran, S., Vaughan, A., "Autonomous Navigation for Deep Impact," *Paper AAS 06-177*, AAS/AIAA Spaceflight Mechanics Meeting, Tampa, FL, January 22-26, 2006.

[17] Mathworks MATLAB help document; ode23, ode45, ode113, ode15s, ode23s, ode23t, ode23tb. Retreived 2009 from website: http://www.mathworks.com/access/helpdesk/help/techdoc/index.html?/access/helpdesk/help/techdoc/ref/ode45.html

[18] Montenbruck, O., Gill, E., *Satellite Orbits*, Springer-Verlag Publishing, pages 187-188, 2005. (Ryan S. Park at JPL wrote the function using the algorithm presented in this source).

[19] Owen, W. M., Jr., Vaughan, R. M., "Optical Navigation Program Mathematical Models," *JPL Engineering Memorandum 314-513*, JPL, Pasadena, CA, August 9, 1991.

[20] Pines, Samuel; "Uniform Representation of the Gravitational Potential and its Derivatives", *AIAA Journal*, Vol. 11, No. 11, November, 1973, pp. 1508-1511.

[21] Riedel, J. E., Bhaskaran, S. , Synnott, S. P., Bollman, W. E., Null, G. W., "An autonomous optical navigation and control system for interplanetary exploration missions," Presented at the Second IAA International Conference on Low-Cost Planetary Missions, International Academy of Astronautics, 1996; also preprint IAA-L-0506, Applied Physics Lab., The Johns Hopkins University, Baltimore, MD, Apr. 1996.

[22] Riedel, J. E., Bhaskaran, S., Synnott, S. P., Desai, S. D., Bollman, W. E., Dumont, P. J., Halsell, C. A., Han, D., Kennedy, B. M., Null, G. W., Owen, W. M. Jr., Werner, R. A., Williams, B. G., "Navigation for the new millennium: Autonomous navigation for Deep-Space 1," Proceedings of the 12th International Symposium, Darmstadt, Germany, June 2-6, 1997.

[23] Scheeres, D., "Satellite Dynamics About Asteroids," *AAS 94-112*, AAS/AIAA Spaceflight Mechanics Meeting, Cocoa Beach FL, February 14-16, 1994.

[24] Stastny, N. B., Geller, D. K., "Autonomous Optical Navigation at Jupiter: A Linear Covariance Analysis," *Journal of Spacecraft and Rockets*, Vol. 45, No. 2, March–April 2008.

[25] Tapley, B.D., Schutz, B.E., Born, G.H.; *Statistical of Orbit Determination,* Academic Press, 2004.

[26] Vallado, David A., *Fundamentals of Astrodynamics and Applications*, 2nd ed. El Segundo: Microcosm Press, 2004.

[27] Williams. G. V., Murdin, P., "Asteroid Discovery in History," *Encyclopedia of Astronomy and Astrophysics,* Article 2747, Bristol: Institute of Physics Publishing, 2001.

[28] Yeomans, D.K., *Comets: A Chronological History of Observation, Science, Myth, and Folklore*, New York: Wiley Publishing, 1991; Barnes-Svarney, P., *Asteroid: Earth Destroyer or New Frontier?*, Basic Books, 2003.

VITA

Corwin Olson was born on July 11, 1984 in Dallas, Texas. He is the son of Delaine Fast and Rob Olson. After graduating from South Garland High School in Garland, Texas in May, 2002, he entered the Aerospace Engineering program at the University of Texas at Austin in August, 2002. After obtaining a Bachelor of Science degree in Aerospace Engineering in May, 2007, he began graduate school at the University of Texas at Austin in August, 2007 with the goal of obtaining a Master of Science degree in Aerospace Engineering.

Corwin has also worked at three different NASA centers. He was a certified NASA instructor in the Training Division at NASA Johnson Space Canter in 2005 and 2006, an intern in the Flight Dynamics Analysis Branch at NASA Goddard Space Flight Center during the summer of 2007, and an intern in the Outer Planets Navigation Group of the Guidance Navigation and Control Section at NASA Jet Propulsion Laboratory during the summer of 2008. He was also an instructor of Tae Kwon Do in Dallas before moving to Austin for school in 2002.

Permanent address: 522 Santa Cruz Dr., Garland, TX 75043

This thesis was typed by the author.