

Spring 4-2023

Autonomous Space Surveillance for Arbitrary Domains

David Zuehlke

Embry-Riddle Aeronautical University, zuehlked@erau.edu

Follow this and additional works at: <https://commons.erau.edu/edt>



Part of the [Astrodynamics Commons](#), and the [Navigation, Guidance, Control and Dynamics Commons](#)

Scholarly Commons Citation

Zuehlke, David, "Autonomous Space Surveillance for Arbitrary Domains" (2023). *Doctoral Dissertations and Master's Theses*. 745.

<https://commons.erau.edu/edt/745>

This Dissertation - Open Access is brought to you for free and open access by Scholarly Commons. It has been accepted for inclusion in Doctoral Dissertations and Master's Theses by an authorized administrator of Scholarly Commons. For more information, please contact commons@erau.edu.

AUTONOMOUS SPACE SURVEILLANCE FOR ARBITRARY DOMAINS

By

David Zuehlke

A Dissertation Submitted to the Faculty of Embry-Riddle Aeronautical University

In Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy in Aerospace Engineering

April 2023

Embry-Riddle Aeronautical University

Daytona Beach, Florida

AUTONOMOUS SPACE SURVEILLANCE FOR ARBITRARY DOMAINS

By

DAVID ZUEHLKE

This Dissertation was prepared under the direction of the candidate's Dissertation Committee Chair, Dr. Troy Henderson, Department of Aerospace Engineering, and has been approved by the members of the Dissertation Committee. It was submitted to the Office of the Senior Vice President for Academic Affairs and Provost, and was accepted in the partial fulfillment of the requirements for the Degree of Doctor of Philosophy in Aerospace Engineering.

DISSERTATION COMMITTEE

Troy Henderson

Digitally signed by Troy Henderson
Date: 2023.04.14 09:42:11 -04'00'

Chair, Dr. Troy Henderson

Richard J. Prazenica

Digitally signed by Richard J. Prazenica
Date: 2023.04.14 09:47:23 -04'00'

Member, Dr. Richard Prazenica

Morad Nazari

Digitally signed by Morad Nazari
Date: 2023.04.14 12:01:58 -04'00'

Member, Dr. Morad Nazari

Theodore von Hippel

Digitally signed by Theodore von Hippel
Date: 2023.04.14 23:09:29 -04'00'

Member, Dr. Ted von Hippel



Digitally signed by SIZEMORE,ALEX,EDWARD.1522418915
Date: 2023.04.13 15:06:09 -06'00'

Member, Dr. Alex Sizemore

Thomas Alan Lovell

Digitally signed by Thomas Alan Lovell
Date: 2023.04.15 10:01:52 -04'00'

Member, Dr. Alan Lovell

Sirish Namilae

Digitally signed by Sirish Namilae
Date: 2023.04.17 12:22:18 -04'00'

Graduate Program Coordinator,
Dr. Sirish Namilae

04/17/2023

Date

Dean of the College of Engineering,
Dr. James W. Gregory

Date

Senior Vice President for Academic
Affairs and Provost,
Dr. Lon Moeller

Date

Dedicated to God from whom all blessings flow. And to my dear wife Kayla, without whom this endeavor could never have been completed. Your faithful love and support made the late nights, countless hours of research, and working through endless equations not just possible, but enjoyable with you by my side.

ACKNOWLEDGMENTS

Psalms 8:5-6 “When I look at Your heavens, the work of Your fingers, the moon and the stars, which you have set in place, what is man that You are mindful of him, and the son of man that You care for him?” You may set out on the journey of pursuing a doctoral degree alone, but you never finish alone. Countless individuals played an indispensable part in providing the instruction, encouragement, and support to help me along the way. As such, I would like to thank all those who supported me in one way or another through this endeavor. A special thank you to my dissertation advisor Dr. Henderson, for the countless meetings and discussions over the past several years. You helped me become a better person through your example of steadfastness and discipline during this time. Thank you to my dissertation committee for your support and feedback along the way. And thank you to all my friends and colleagues at the Space Technologies Laboratory. All the hours of collaboration and late nights on the roof with a telescope will not be soon forgotten. Thank you to my friend Dr. Tiwari, coffee and friendship fuel great works. I would also like to thank the National Defense Science and Engineering Graduate (NDSEG) fellowship program for their generous support during my time as a PhD student at Embry-Riddle Aeronautical University. Thank you to my parents for teaching me to always do my best in anything. And finally thank you to my wife, your encouragement and support made all the difference to help me run this race well.

ABSTRACT

Space is becoming increasingly congested every day and the task of accurately tracking satellites is paramount for the continued safe operation of both manned and unmanned space missions. In addition to new spacecraft launches, satellite break-up events and collisions generate large amounts of orbital debris dramatically increasing the number of orbiting objects with each such event. In order to prevent collisions and protect both life and property in orbit, accurate knowledge of the position of orbiting objects is necessary. Space Domain Awareness (SDA) used interchangeably with Space Situational Awareness (SSA), are the names given to the daunting task of tracking all orbiting objects. In addition to myriad objects in low-earth-orbit (LEO) up to Geostationary (GEO) orbit, there are a growing number of spacecraft in cislunar space expanding the task of cataloguing and tracking space objects to include the whole of the earth-moon system.

This research proposes a series of algorithms to be used in autonomous SSA for earth-orbiting and cislunar objects. The algorithms are autonomous in the sense that once a set of raw measurements (images in this case) are input to the algorithms, no human in the loop input is required to produce an orbit estimate. There are two main components to this research, an image processing and satellite detection component, and a dynamics modeling component for three-body relative motion.

For the image processing component, resident space objects, (commonly referred to as RSOs) which are satellites or orbiting debris are identified in optical images. Two methods of identifying RSOs in a set of images are presented. The first method autonomously builds a template image to match a constellation of satellites and proceeds to match RSOs across a set of images. The second method utilizes optical flow to use the image velocities of objects to differentiate between stars and RSOs. Once RSOs have been detected, measurements are generated from the detected RSO locations to estimate the orbit of the observed object. The orbit determination component includes multiple methods capable of handling both earth-orbiting and cislunar observations. The methods used include batch-least squares and unscented Kalman filtering for earth-orbiting objects. For cislunar objects, a novel application of a particle swarm optimizer (PSO) is used to estimate the

observed satellite orbit. The PSO algorithm ingests a set of measurements and attempts to match a set of virtual particle measurements to the truth measurements. The PSO orbit determination method is tested using both MATLAB and Python implementations.

The second main component of this research develops a novel linear dynamics model of relative motion for satellites in cislunar space. A set of novel linear relative equations of motion are developed with a semi-analytical matrix exponential method. The motion models are tested on various cislunar orbit geometries for both the elliptical restricted three-body problem (ER3BP) and the circular restricted three-body problem (CR3BP) through MATLAB simulations. The linear solution method's accuracy is compared to the non-linear equations of relative motion and are seen to hold to meter level accuracy for deputy position for a variety of orbits and time-spans.

Two applications of the linearized motion models are then developed. The first application defines a differential corrector to compute closed relative motion trajectories in a relative three-body frame. The second application uses the exponential matrix solution for the linearized equations of relative motion to develop a method of initial relative orbit determination (IROD) for the CR3BP.

TABLE OF CONTENTS

ACKNOWLEDGMENTS	i
ABSTRACT	ii
LIST OF FIGURES	xv
LIST OF TABLES	xvii
1 Introduction	1
1.1 Research Goals	1
1.2 Dissertation Overview	3
2 Background	4
2.1 Optical Space Surveillance Background	4
2.1.1 Space Debris a Monumental Problem	4
2.1.2 Optical Space Situational Awareness	5
2.1.3 Template Matching Overview	7
2.2 Three-Body Problem and Relative Motion	9
2.2.1 Relative Motion Background	9
2.2.2 Three-Body Problem Background	10
2.3 Linear System Theory: The Matrix Exponential	13
3 Satellite Tracking and Orbit Estimation	15
3.1 Optical Measurements	15
3.1.1 Tracking Modes	16
3.1.2 Image Processing	18
3.2 Template Matching for RSO Detection	20
3.2.1 Autonomous Template Generation Algorithm	21

3.2.2	Template Matching and RSO Association	25
3.2.3	Template Update Process	27
3.2.4	Experimental Setup	29
3.3	Optical Flow for RSO Detection	30
3.3.1	Image Simulator	32
3.3.2	Autonomous RSO Identification Process	33
3.3.3	Optical Flow Algorithm Summary	36
3.3.4	Simulated Space Image Results	38
3.3.5	Experimental Images Optical Flow Results	39
3.4	Orbit Estimation	40
3.4.1	Initial Orbit Determination	40
3.4.2	Unscented Kalman Filter for Orbit Estimation	43
3.4.3	Autonomous RSO Constellation Orbit Determination Algorithm	47
3.4.4	Batch Orbit Estimation - Differential Correction Algorithm	48
3.5	Orbit Determination Results	51
3.5.1	Template Matching Results	52
3.5.2	UKF Orbit Estimation Results	56
3.5.3	Batch Orbit Estimation Results	59
4	Three-Body Relative Motion Models	62
4.1	Three-Body Dynamics	63
4.1.1	Three-Body Problem Geometry	63
4.1.2	Coordinate Transformation	66
4.1.3	ER3BP Equations of Motion	67
4.1.4	CR3BP Equations of Motion	71
4.1.5	Three-Body Equilibrium Positions (Lagrange Points)	72
4.2	Relative Three-Body Dynamics	74
4.2.1	ER3BP Relative Equations of Motion	74

4.2.2	CR3BP Relative Equations of Motion	79
4.3	Linearized Relative Motion for the Three-Body Problem	80
4.3.1	ER3BP Linear Motion Model	80
4.3.2	CR3BP Linear Motion Model	83
4.3.3	Matrix Exponential Solution	85
4.4	Linearized Relative Motion Results	87
4.4.1	Low Lunar Orbit Example	87
4.4.2	Low-Earth Orbit Example	94
4.4.3	Geostationary Example	96
4.4.4	Lyapunov Orbit Example	102
4.4.5	Southern HALO Orbit Example	108
4.4.6	L2 Vertical Orbit	120
4.5	Stability of the Linear Motion Models	121
4.6	State-Transition Matrix	128
4.7	Linearized Relative Motion Summary	129
5	Applications of Linear Relative Motion for the Three-Body Problem	130
5.1	Natural Closed Relative Motion	130
5.1.1	Differential Correction for Relative Motion	131
5.1.2	Differential Correction Algorithm for Closed Relative Orbits	134
5.2	Numerical Examples of Closed Relative Motion	135
5.2.1	L2 Southern HALO Example	135
5.2.2	L1 Northern HALO Example	140
5.2.3	Resonant Orbit Example	144
5.2.4	Natural Closed Relative Motion Summary	146
5.3	Initial Relative Orbit Determination (IROD)	151
5.3.1	Three-Body IROD	152
5.3.2	MATLAB Symbolic Solution Method	157

5.3.3	IROD Results	158
5.3.4	IROD Future Work and Summary	161
6	Three-Body Orbit Determination Using Particle Swarm Optimization	164
6.1	PSO Background	164
6.2	PSO Scenario CR3BP Dynamics	164
6.2.1	Deputy Chief Scenario for Cislunar IOD	165
6.2.2	Particle Swarm Optimizer	166
6.2.3	PSO Tolerance for Ending Conditions	169
6.2.4	Local Minimization	169
6.3	Python vs. MATLAB	171
6.4	MATLAB PSO Results for CR3BP IOD	172
6.4.1	Scenario 1: HALO Orbiting Deputy and Chief	172
6.4.2	Scenario 2: Chief Axial Orbit	174
6.5	Python Results	182
6.6	PSO for CR3BP IOD Conclusions	186
7	Conclusions and Recommendations	188
7.1	Research Conclusions	188
7.1.1	Satellite Image Processing and Orbit Estimation Conclusions	189
7.1.2	Spacecraft Relative Motion and Applications Conclusions	190
7.2	Recommendations for Future Research	191
7.2.1	Template Matching Improvements	191
7.2.2	Relative Motion Improvements	191
7.2.3	Natural Closed Relative Motion Improvements	191
7.2.4	IROD Improvements	192
7.2.5	PSO IOD Improvements	192
7.3	Concluding Remarks	192

REFERENCES	194
PUBLICATIONS	206

LIST OF FIGURES

Figure	Page
2.1 Orbital debris examples.	6
2.2 Example of template matching from Ref. [1].	8
2.3 Example correlation peaks from Ref. [1].	8
2.4 Illustration of the general three-body problem.	11
3.1 Example optical tracking system during the day.	15
3.2 Example optical tracking system at night.	15
3.3 RSO tracking mode example image. Stars appear as streaks throughout the image and the tracked RSO appears as a point-source.	16
3.4 sidereal tracking mode example image. Stars appear as point-sources while RSOs appear as streaks. Two RSOs are visible as streaks above the center of the image.	17
3.5 Autonomous template generation process.	22
3.6 Raw (left) and processed (right) cropped portions of example image containing four RSOs.	22
3.7 RSO template matching peaks.	23
3.8 Cropped raw image showing matched RSO locations highlighted in red. Detected RSO locations are used to autonomously build a template image for subsequent matching.	24
3.9 Autonomously generated template image example. (Rotated from actual orientation for display purposes).	26
3.10 Simulated space image for “satellite tracking mode”. All stars in the image appear as short streaks and simulated RSOs appear as point sources and are highlighted by red boxes.	34
3.11 RSO identification process block diagram.	35

3.12	Optical flow example for simulated image. Note that stars appear as bloated streaks and an RSO appears as a point source. Blue vectors represent the image velocity of each subset of pixels. Note the higher magnitudes of image velocities on all the stars compared to the RSO near the center right of the image.	36
3.13	Simulated image optical flow velocity magnitude.	38
3.14	Simulated star field.	39
3.15	Optical Flow from simulated star field.	39
3.16	Simulated star field with RSOs highlighted.	39
3.17	Closeup of simulated image optical flow showing identified RSO.	39
3.18	Raw ground-based image of GEO satellite.	41
3.19	Optical flow detection of RSO in real image of GEO satellite.	41
3.20	RSO detection frame 10.	42
3.21	RSO detection frame 30.	42
3.22	RSO detection frame 50.	42
3.23	RSO detection frame 70.	42
3.24	(a) Correlation values in a 1D array with RSOs in red. (b) Correlation 3D map with RSOs in red.	53
3.25	Rotated template image.	53
3.26	Successful template match example. Six RSOs are present in the image and were autonomously labeled by the template matching algorithm.	54
3.27	Template match correlation maps.	55
3.28	Peak correlation value with template updates.	55
3.29	Template images through time.	56
3.30	RSO angular measurements.	57
3.31	UKF measurement comparison RSO#1.	57

3.32	Batch orbit results for satellite position. Three- σ bounds on position are shown as transparent ellipses. Note that the true and estimated position are very close.	60
3.33	Batch orbit results for satellite velocity. Three- σ bounds on velocity are shown as transparent ellipses. Note that the true and estimated velocity are quite close.	61
4.1	General three-body problem geometry.	64
4.2	Geometry for the three-body problem. Coordinate systems shown include the barycentric and Moon-centered synodic rotating systems.	64
4.3	Earth-Moon Lagrange point positions and distances.	73
4.4	Geometry of a deputy and chief scenario in the ER3BP. Position vectors are shown in the rotating barycentric frame. Note the chief-centered relative frame (C -frame) is aligned with the rotating \mathcal{B} -frame.	75
4.5	Low lunar orbit example: deputy and chief 3D positions.	89
4.6	Low lunar orbit example: 3D relative motion plots.	90
4.7	Low lunar orbit example: relative position plots.	91
4.8	Low lunar orbit example: relative position errors.	92
4.9	Low lunar orbit example: relative velocity errors.	93
4.10	Low Earth orbit example: deputy and chief 3D positions.	95
4.11	Low Earth orbit example: 3D relative motion plots.	97
4.12	Low Earth orbit example: 3D relative motion plots.	98
4.13	Low Earth orbit example: relative position errors.	99
4.14	Low Earth orbit example: relative velocity errors.	100
4.15	Low Earth orbit example: range and speed errors.	101
4.16	Geostationary orbit example: deputy and chief 3D positions.	103
4.17	Geostationary orbit example: 3D relative motion plots.	104
4.18	Geostationary orbit example: relative position errors.	105
4.19	Geostationary orbit example: relative velocity errors.	106
4.20	Geostationary orbit example: range and speed errors.	107

4.21	Lyapunov orbit example: deputy and chief 3D positions.	109
4.22	Lyapunov orbit example: 3D relative motion plots.	110
4.23	Lyapunov orbit example: relative position errors.	111
4.24	Lyapunov orbit example: relative velocity errors.	112
4.25	Lyapunov orbit example: range and speed errors.	113
4.26	Southern HALO orbit example: deputy and chief 3D positions.	115
4.27	Southern HALO orbit example: 3D relative motion plots.	116
4.28	Southern HALO orbit example: relative position errors.	117
4.29	Southern HALO orbit example: relative velocity errors.	118
4.30	Southern HALO orbit example: range and speed errors.	119
4.31	L2 vertical orbit example: deputy and chief 3D positions.	122
4.32	L2 vertical orbit example: 3D relative motion plots.	123
4.33	L2 vertical orbit example: relative position errors.	124
4.34	L2 vertical orbit example: relative velocity errors.	125
4.35	L2 vertical orbit example: range and speed errors.	126
4.36	L2 vertical orbit magnitude of first two system eigenvalues.	127
4.37	L2 vertical orbit magnitude of third and fourth system eigenvalues.	127
4.38	L2 vertical orbit magnitude of fifth and sixth system eigenvalues.	128
5.1	Differential correction process for finding closed relative trajectories.	131
5.2	Single-shooting algorithm diagram. An initial deputy position and velocity are updated in an attempt to find a final closed trajectory (shown in red).	134
5.3	L2 southern HALO example: convergence of design variables and tolerance.	137
5.4	L2 southern HALO example: convergence and final closed trajectory.	138
5.5	L2 southern HALO example: family of solutions with increasing TOF.	139
5.6	L2 southern HALO example: “rendezvous” trajectory and chief propagated orbit.	141
5.7	L1 northern HALO example: convergence of design variables and tolerance.	142
5.8	L1 northern HALO example: convergence and final closed trajectory.	143

5.9	L1 northern HALO example: family of solutions with increasing TOF.	145
5.10	Chief orbit propagated for one period for L1 northern HALO orbit.	146
5.11	Resonant orbit example convergence.	147
5.12	Resonant orbit example convergence.	148
5.13	Resonant orbit example relative orbits.	149
5.14	Chief orbit propagated for one period for 3-2 resonant orbit case.	150
5.15	Typical IROD scenario shown in relative frame based at the chief. Three LOS measurements are taken that are parallel to the current deputy position vector.	153
5.16	IROD scenario in L2 Lyapunov orbit showing deputy and chief position in the rotating \mathcal{B} -frame using CR3BP dynamics.	160
5.17	IROD scenario #1 results. Deputy IC estimated by IROD method is plotted near truth, and propagated to show the close alignment.	161
5.18	IROD scenario in a 3-2 resonant orbit showing deputy and chief position in the rotating \mathcal{B} -frame using CR3BP dynamics. Note that the chief orbit period is much longer than the 6.75 day scenario period.	162
5.19	IROD scenario #2 results. The deputy is in a closed periodic trajectory, and the three LOS measurements are shown by red arrows pointing to the position of the deputy at the measurement time. The IROD solution is propagated and shown by a green dashed line.	163
6.1	PSO operation overview. Particle population moves under the influence of current best positions (blue arrows), current motion (red arrows), and current global best position (green arrows). After m iterations the population converges to the global optimum solution.	165
6.2	Deputy chief initial orbit determination scenario. PSO initial states are seeded around a probable deputy position (shown as red circles).	166

6.3	Range and speed error reduction with increasing iterations. After a certain point the PSO error meets the LSQ error. Before that point there is an advantage of using LSQ on the PSO current solution.	171
6.4	Scenario 1: Chief and deputy HALO orbits depicted over 7-day period with PSO deputy orbit overlay.	173
6.5	Scenario 1: Global cost per iteration, decreasing until iteration counter is reached or tolerance is met.	174
6.6	Scenario 1: Initial (red) and final state (blue) particle positions in three axes, swarming to global best.	175
6.7	Scenario 1: Initial (red) and final state (blue) of particle velocity in three axes.	176
6.8	Scenario 1: Position error in each axis per iteration (top) and velocity error in each axis per iteration (bottom).	177
6.9	Scenario 1: Magnitude of position error per iteration (top) and magnitude of velocity error per iteration (bottom).	178
6.10	Scenario 2: Chief and deputy HALO orbits depicted over 7-day period with PSO deputy orbit overlay.	178
6.11	Scenario 2: Global cost per iteration, decreasing until iteration counter is reached or tolerance is met.	179
6.12	Scenario 2: Initial (red) and final state (blue) particle positions in three axes, swarming to global best.	180
6.13	Scenario 2: Initial (red) and final state (blue) of particle velocity in three axes.	180
6.14	Scenario 2: Position error in each axis per iteration (top) and velocity error in each axis per iteration (bottom).	181
6.15	Scenario 2: Magnitude of position error per iteration (top) and magnitude of velocity error per iteration (bottom).	182
6.16	Python PSO scenario results. Deputy and chief positions propagated for duration of scenario.	184

6.17 Python PSO scenario results. Global minimum cost plot.	184
6.18 Python PSO scenario results. Global best estimate of deputy initial position.	185
6.19 Python PSO scenario results. Global best estimate of deputy initial velocity.	185
6.20 Python PSO scenario results. Starting and ending particles for deputy position states in non-dimensional units.	186
6.21 Python PSO scenario results. Starting and ending particles for deputy velocity states in non-dimensional units.	187

LIST OF TABLES

Table	Page
3.1 UKF estimated orbital element error from TLE orbital element data.	58
3.2 Batch least squares error results. Estimated initial state errors are given in position and velocity form as well as orbital element form.	60
4.1 LLO example deputy and chief initial conditions given in non-dimensional units for both the ER3BP and the CR3BP.	88
4.2 Low lunar orbit example. RMS errors for linear motion models of ER3BP and CR3BP.	88
4.3 LEO example deputy and chief initial conditions given in non-dimensional units for both the ER3BP and the CR3BP.	94
4.4 Low Earth orbit example RMS errors for deputy position (in km) and velocity (in km/s) components over duration of scenario.	94
4.5 GEO example deputy and chief initial conditions given in non-dimensional units for both the ER3BP and the CR3BP.	96
4.6 Geostationary orbit RMS errors for deputy position and velocity components.	102
4.7 Lyapunov orbit example deputy and chief initial conditions given in non-dimensional units for both the ER3BP and the CR3BP.	102
4.8 Lyapunov orbit example RMS errors for deputy position and velocity components.	108
4.9 Southern HALO orbit example deputy and chief initial conditions given in non-dimensional units for both the ER3BP and the CR3BP.	114
4.10 Southern HALO example RMS errors for deputy position and velocity components.	120
4.11 L2 vertical orbit deputy and chief initial conditions given in non-dimensional units for both the ER3BP and the CR3BP.	120
4.12 L2 vertical orbit RMS errors of deputy position and velocity components.	121

5.1	Deputy and chief initial conditions for L2 southern HALO example given in non-dimensional units in the barycentric frame. The deputy IC is given for the initial guess, and the final computed closed trajectory IC.	136
5.2	Deputy and chief initial conditions for L1 northern HALO example given in non-dimensional units in the barycentric frame. The deputy IC is given for the initial guess, and the final computed closed trajectory IC.	140
5.3	Deputy and chief initial conditions for 3:2 resonant orbit example given in non-dimensional units in the barycentric frame. The deputy IC is given for the initial guess, and the final computed closed trajectory IC.	144
5.4	Chief and deputy initial conditions for L2 Lyapunov orbit IROD scenario. Note that the initial conditions are given in non-dimensional units in the barycentric frame.	159
5.5	Chief and deputy initial conditions for closed relative motion IROD scenario. Note that, initial conditions are given in non-dimensional units in the barycentric frame.	160
6.1	Summary of PSO and nonlinear least squares (NLSQ) results. In both cases NLSQ produced a lower error after PSO terminated.	172
6.2	Scenario 1 initial conditions given in non-dimensional units in the barycentric frame. The scenario was propagated for 7 days and utilized 35 measurements.	173
6.3	Scenario 2 initial conditions given in non-dimensional units in the barycentric frame. The scenario was propagated for 7 days and utilized 10 measurements.	175
6.4	Python PSO scenario error results for estimating deputy initial condition.	183

1 Introduction

Space is becoming increasingly congested every day and the task of accurately tracking satellites is paramount for the continued safe operation of both manned and unmanned space missions. Low Earth orbit faces the prospect of multiple super constellations of satellites working in close proximity to one another. Already there are more than 2,000 satellites in SpaceX's StarLink network alone, with thousands more planned. In addition to growing numbers of active satellites, the near Earth debris population has also grown, and current estimates put at least 700,000 objects greater than 1 cm in size in Earth orbit [2, 3]. These objects include active satellites, launch generated debris, and debris from satellite break-up events and collisions. While near Earth space surveillance is incredibly important, increased utilization of cislunar space with NASA's upcoming Artemis missions and commercial lunar payload program point to a coming dramatic increase in space missions to the cislunar domain and a subsequent need for accurate cislunar space situational awareness (SSA) [4, 5]. The increase in objects orbiting beyond geostationary orbit (GEO) necessitate the need for accurate SSA efforts in this domain. Thus, the task of cataloguing and tracking space objects governs the whole of the Earth-Moon system. This need for accurate and autonomous methods of space situational awareness (SSA) for all orbital domains is the main motivator for this research.

This research seeks to advance the state of the art in SSA methods by providing a set of algorithms that require no human interaction to produce an orbit estimate. Additionally, this research seeks to advance knowledge of cislunar relative orbital dynamics through the development of a novel set of linear relative equations of motion for cislunar space.

1.1 Research Goals

This research presents a series of algorithms to be used for autonomous space domain awareness for Earth-orbiting and cislunar objects. The algorithms presented in this research are autonomous in the sense that given the proper set of inputs, no human in the loop interaction is needed. For example, for detecting satellites in imagery, raw images are supplied and an orbital estimate of the detected satellites is provided. There are two main components of this dissertation that are developed to advance SSA efforts for both Earth orbiting and cislunar satellites. First measurement

acquisition and processing, and secondly accurate dynamics modeling.

First, to advance the field of autonomous measurement acquisition from unresolved imagery of satellites, this research develops two independent methods of detecting satellites in a set of images. Once detected, measurements are used to produce an orbit estimate of the detected satellite(s). The image processing and measurement component utilizes template matching and optical flow as methods of obtaining measurements of satellites for the orbit determination step [6, 7]. The orbit determination component includes multiple methods capable of handling both Earth-orbiting and cislunar observations. The method used will depend on the number of measurements available and the domain of the observed object. For Earth-orbiting satellites, an orbit estimate is computed from observations using batch-least squares and unscented Kalman filtering. For cislunar orbit determination, a novel application of a particle swarm optimizer (PSO) is developed to estimate the observed object's orbit.

Secondly, for the dynamics modeling component of this research, this work focuses on advancing knowledge for relative motion in cislunar space by the development of a novel set of linearized relative equations of motion for the three-body problem (both the elliptical restricted three-body problem (ER3BP) and the circular restricted three-body problem (CR3BP)). A linearized state-matrix is developed and modeled through integration, and the use of the matrix exponential. The linearized motion models are then utilized for two applications. One application uses the linear dynamics for a differential corrector to compute closed relative trajectories in cislunar space. The second application uses the linearized motion model to develop a method of initial relative orbit determination (IROD) for the CR3BP. The main contributions of this work are summarized as follows.

1. Leverage optical image processing techniques including template matching and optical flow for satellite detection in unresolved space images.
2. Development of linearized three-body equations of relative motion with a semi-analytical matrix exponential solution for cislunar dynamics.
3. Application of linearized relative three-body dynamics for finding closed natural motion

trajectories in the relative three-body problem.

4. Development a of three-body initial relative orbit determination method.
5. Development of a particle swarm optimization method for three-body orbit determination.

1.2 Dissertation Overview

The elements of the above contributions have been successfully demonstrated and published as a combination of conference and journal publications. A full list of the publications resulting from this research can be found at the end of this document. The remainder of this dissertation will be structured as follows. An overview of relevant background material and literature will be presented in Chapter 2. Optical image processing methods as well as several Earth-orbiting orbit estimation techniques are presented in Chapter 3. Chapter 4 presents various three-body relative motion research topics. Including the development of a linearized relative motion model for both ER3BP and the circular CR3BP. Chapter 5 demonstrates two applications of the linearized motion models developed for the three-body problem. The first application developed is a novel method of using differential correction to find closed relative motion trajectories. Secondly, a method of finding an initial relative orbit estimate for the CR3BP is presented. Chapter 6 outlines a novel application of using a particle swarm optimizer for estimating orbits in the CR3BP. Finally, Chapter 7 provides overall conclusions and recommendations for future developments of this research.

2 Background

In this section, background research into the area of space surveillance and relevant orbital dynamics for the three-body problem is presented. There is a wealth of knowledge in SDA and SSA due to the increasing interest of the field in recent years. The basic techniques of determining an orbit go back hundreds of years and were the subject of intense mathematical study by the great minds of mathematicians such as Gauss and Lagrange [8]. Background information on the three-body problem, and linear system theory is also presented with the goal of developing autonomous systems for SSA in cislunar space.

2.1 Optical Space Surveillance Background

Due to the increased number of satellites and debris in orbit, the task of optical space surveillance is more important than ever before. This section outlines why tracking of satellites and space debris is necessary, as well as background material for the development of the autonomous SSA methods discussed in this research.

2.1.1 Space Debris a Monumental Problem

Space is becoming more congested with every passing day. Current estimates put at least 700,000 objects greater than 1 cm in size orbiting the Earth [2, 3]. The vast majority of these objects are uncatalogued space debris objects caused by orbital collisions, satellite breakup events, launch events, and the latest military testing of anti-satellite weaponry [9]. The prospect of mega-constellations such as SpaceX's Starlink satellites, propose to loft thousands of satellites into low-Earth orbit (LEO). To date, more than 3,800 Starlink satellites have already been launched and can be seen crossing the night sky like long strings of pearls. The ever-increasing number of satellites in orbit demands that groups of satellites (constellations or swarms) must cooperate within close proximity to accomplish their missions. Providing accurate orbit estimates of these satellites relative to one another is of paramount importance for preventing on-orbit collisions. As an example of the damage that orbital debris can cause to a satellite, figure 2.1a shows an orbital debris strike on the panel of the NASA Solar Max satellite [10]. Figures 2.1b - 2.1d show a NASA generated map of orbital debris objects from LEO to GEO. Note that the largest portion of debris lies in LEO and

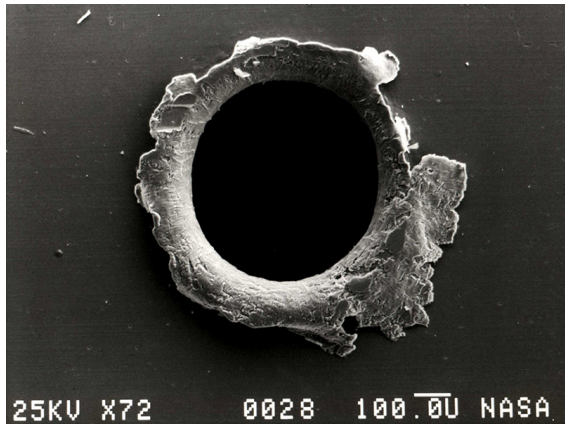
the GEO belt due to the most satellites operating in these regimes. With the current launch cadence, the space debris population will only increase, necessitating accurate SSA data analysis to prevent damage to orbiting assets.

Current methods of optical SSA lack autonomous closed-loop approaches that address resident space object (RSO) detection and orbit determination in one algorithm. This research seeks to provide two approaches to provide measurements for orbit determination without human in the loop intervention. Before delving into the details of RSO detection and orbit determination, a brief overview of optical SSA is presented.

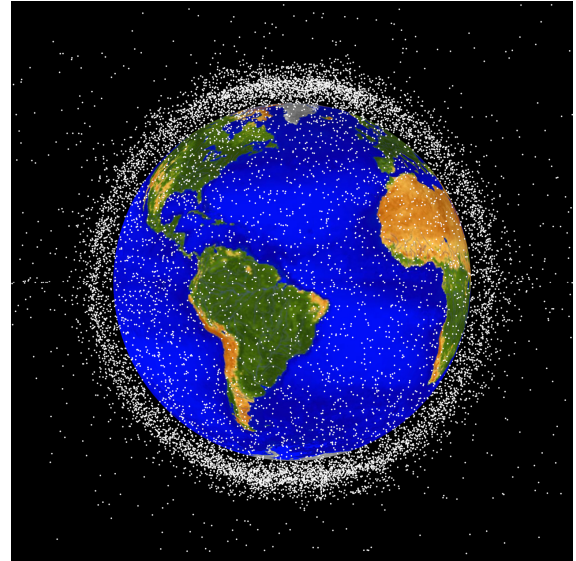
2.1.2 Optical Space Situational Awareness

Space situational awareness tracking data have two main contributing sensor types, active tracking using radar, and passive tracking using optical telescopes. Optical tracking methods have long been used to provide SSA data for performing orbit estimation, and a variety of existing methods provide exceptional results [11]. Optical tracking involves using a ground (or space-based) telescope to capture an image of one or more resident space objects (RSOs) and the background star field.

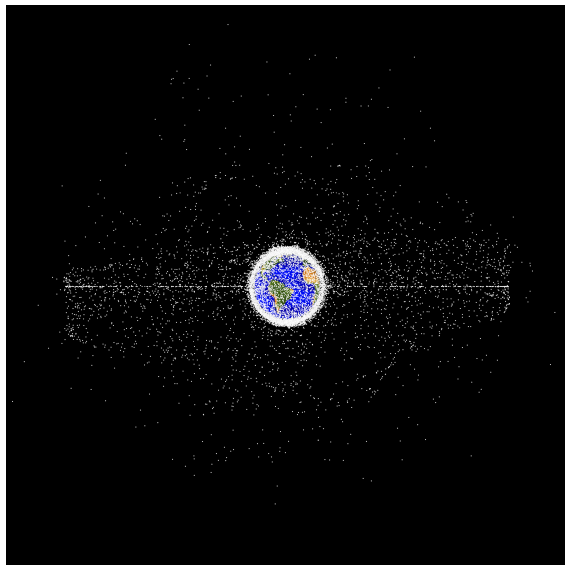
There are a variety of methods to detect which objects in a given image represent stars and which objects RSOs. One approach involves using the gross-motion of objects in the image sequence to discriminate between background objects (stars) and RSOs since the background star field will move differently than any RSOs in the scene [12, 13]. Another method that can be used involves streak detection for images taken using a sidereal (the rate at which stars move) tracking mode so that all stars in the field of view (FOV) remain stationary and RSOs form streaks in the image [14, 15]. Depending on various observation factors, such as exposure length, camera sensitivity, and tracking mode, RSOs will appear either as point-sources or as streaked objects in an image. Another method of RSO detection is to transform object locations (stars and RSO detected points) to an inertial observation frame fixed to the stars. In this inertial frame, RSOs exhibit motion, whereas stars remain stationary. Assuming availability of sufficiently dense measurements, the motion in the inertial frame can be used to determine RSO locations from background stars and was successfully



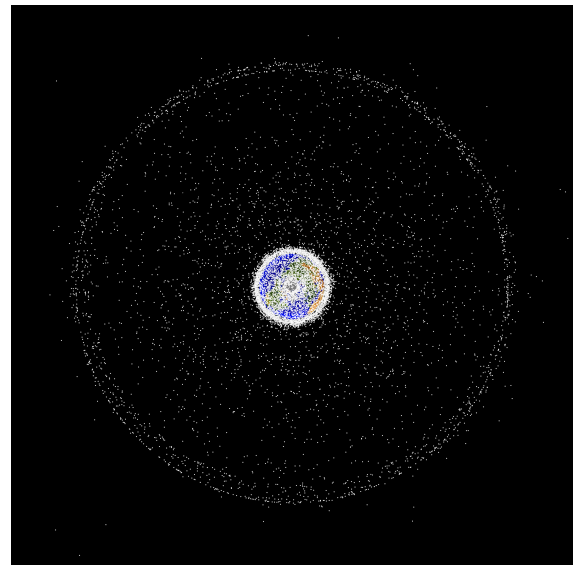
(a) Orbital debris strike on NASA Solar Max satellite panel. Courtesy of NASA [10].



(b) LEO view of orbital debris. Courtesy of NASA [10].



(c) GEO view of orbital debris. Courtesy of NASA [10].



(d) GEO polar view of orbital debris. Courtesy of NASA [10].

Figure 2.1 Orbital debris examples.

demonstrated in Refs. [16, 17]. For the problem of tracking multiple satellites at once, Zuehlke and Henderson [18] demonstrated that template matching provides an effective way for obtaining angular measurements of constellations of geostationary satellites.

2.1.3 Template Matching Overview

Template matching is a well known algorithm for determining whether a query (template) image is contained in a target (base) image. Methods of template matching are varied and include robust methods such as shown by Yang et al. [19] where the authors proposed a method of iteratively searching through multiple possible rotation transformations of a template image. Another method is shown in Ref. [20] where an intelligent search method was used to propose scale and rotation changes. Normalized cross-correlation (NCC) is a well known method of template matching and has been shown to be robust to changes in scale and rotation [21]. Essentially the method works by comparing the template image to all the pixels of the target image to compute a matrix of coefficients representing the degree of correlation [6, 21]. The correlation map can be thought of as a 3D surface where the highest peak corresponds to the location of the template image in the target image. When the correlation coefficients are represented as a 3D surface, the peaks represent areas of close correlation. Assuming a successful match, the maximum peak gives the location of the region matching with the template image. Equation (2.1) is used to find the matrix of coefficients and is the heart of the template matching algorithm. Note that: $\gamma(u, v)$, is the matrix of correlation coefficients, $I(x, y)$ is the input image, $T(\bar{x}, \bar{y})$ is the template image, $\bar{I}_{x,y}$ is the input image mean, and \bar{T} is the template image mean intensity.

$$\gamma(u, v) = \frac{\sum_{x,y} [I(x, y) - \bar{I}_{x,y}] [T(x - u, y - v) - \bar{T}]}{\left[\sum_{x,y} [I(x, y) - \bar{I}_{x,y}]^2 \sum_{x,y} [T(x - u, y - v) - \bar{T}]^2 \right]^{0.5}} \quad (2.1)$$

Normalizing the correlation coefficients such that: $[-1 \leq \gamma(u, v) \leq +1]$, helps to overcome difficulties that plague cross-correlation such as illumination, rotation, and scale differences between the target and template image [6]. The MATLAB implementation of `normxcorr2` was used in this research for NCC. Figures 2.2 and 2.3 show an example of template matching and the associated

peak in the cross correlation surface (courtesy of MATLAB [1]). The highest peak represents the location of the template image in the target image.



Figure 2.2 Example of template matching from Ref. [1].

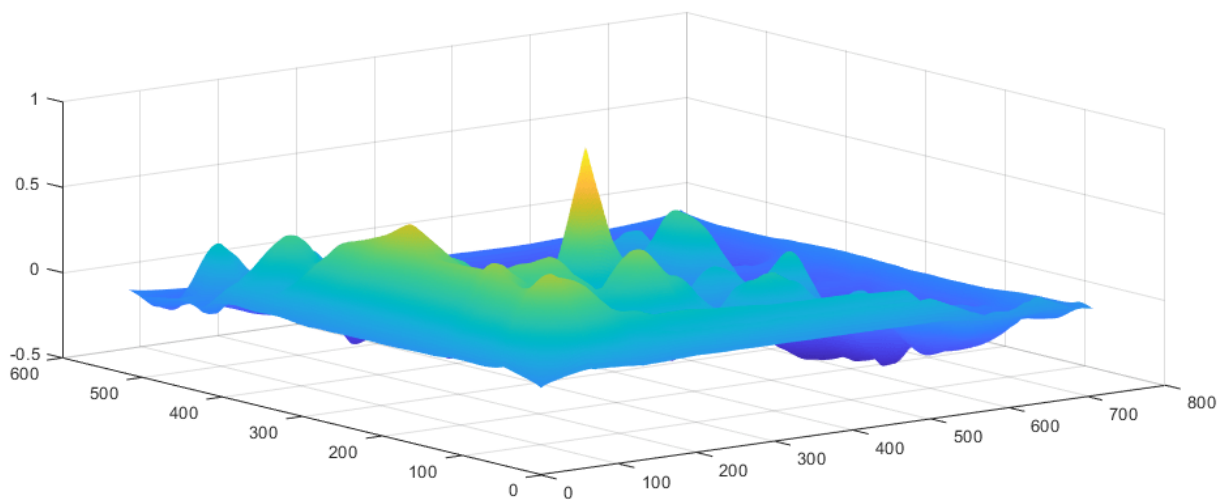


Figure 2.3 Example correlation peaks from Ref. [1].

Given that many satellites operate in groups (constellations), template matching for a constellation solves the data association problem of determining which detected RSO associates across a series of images. However, a significant drawback of the method involved was the need for *a priori* knowledge to provide a template image of the constellation to be tracked. Additionally, if the constellation exhibits relative orbital motion (i.e. the constellation changed with time), then over time the initial template would become invalidated after the satellites changed position relative

to one another. Geostationary constellations change slowly over time given their orbit geometry and long time-scale perturbations; however, over a long enough observation window the motion is still enough to make an initial template inaccurate. Non-continuous measurements also causes a problem for an initial template. For example if measurements were taken separated by days, the initial template configuration would no longer be valid. LEO constellations exhibit rapid relative orbital motion due to the nature of their orbits. Dynamically changing scenes requires a method of updating the template image being matched against a target image. Using the image data itself to update the template image is possible and was demonstrated for tracking a small UAS quad-copter in flight by Zuehlke et al. [22]. The same concept of updating the template applies to the context of tracking a satellite constellation. Regardless of the method used to process optical images of RSOs, the goal is to obtain a set of angular measurements at a moment in time for use in an orbit estimation algorithm.

2.2 Three-Body Problem and Relative Motion

This section outlines relevant background information to the problem of cislunar motion and the three body problem in preparation for the development of cislunar SSA methods.

2.2.1 Relative Motion Background

The problem of relative motion has been of keen interest since the dawn of the space race [23]. The Gemini and Apollo programs perfected rendezvous and docking in Earth orbit and eventually led to the hallmark moment of docking in Lunar orbit. However, it is important to note that the majority of knowledge available for docking and close-proximity operations is based on two-body motion [8]. The now ubiquitous Hill-Clohessy-Wiltshire (HCW) equations of relative motion form the primary basis of current knowledge on relative spacecraft motion. And while the HCW equations are valid in Earth orbit, and under the simplifying assumptions of the two-body model, applying them to a docking scenario in Lunar orbit, especially for higher altitude orbits such as near rectilinear halo orbits (NRHO) and distant retrograde orbits (DROs) presents large errors in their solutions [24]. The limitations of the HCW equations is an area of significant research interest given that NASA's upcoming Artemis missions and Gateway program are set to pave the way for myriad

missions to the cislunar region [4, 5]. There exist several extensions to the HCW equations to deal with perturbations and non-circular orbits, but the base model is still limited to the two-body orbit assumption [25]. A major contribution of this work is the development of a set of linear relative equations analogous to the HCW equations for cislunar dynamics.

To support the Artemis program, NASA desires significant support from commercial partners, and payload delivery through the Commercial Lunar Payload Program which will inevitably lead to the need for a scheme of traffic management and lunar space situational awareness [4]. In addition to NASA, international and commercial space agencies alike have expressed interest in utilizing cislunar space for science, exploration, and resource extraction missions with interested parties including ESA, CNSA, Roscosmos, ISRO, JAXA, Blue Origin, SpaceX and Dynetix [4, 5]. The Air Force Research Laboratory (AFRL) recently released requests for commercial assistance with the development of a mission designed to provide the first space traffic management for cislunar space [26].

Two important considerations arise from the potential large increase in space missions to the cislunar domain: (1) the inevitable proliferation of space debris and need for cislunar SSA. And (2) the ability to autonomously perform rendezvous and proximity operations, including circumnavigation, and docking in cislunar space. Both lunar SSA and proximity operations require an innate understanding of the underlying dynamics of the cislunar domain and the three-body problem.

2.2.2 Three-Body Problem Background

The study of the three-body problem has been a subject of mathematical interest for more than 200 years dating to the time of Gauss and Lagrange. Three body motion can be broadly described as the motion of three bodies in space due to their mutual gravitational attraction. The geometry of the three-body problem is illustrated in Fig. 2.4. The general three-body problem presents no known analytical solution, with only 10 integrals of the motion available. As such a number of simplifications have been studied which offer more tractable analysis. In the restricted three-body problem, the mass of the third body is considered negligible, and thus the gravitational

force between the third body and the primary and secondary bodies can be neglected (i.e. the mass of a satellite in orbit around the Moon does not affect the orbit of the Moon or Earth). The restricted three body problem's most general form is the Elliptical Restricted Three Body Problem (ER3BP) wherein the motion of the primary gravitational bodies is assumed to be elliptical. The ER3BP can be simplified further by assuming a circular orbit between the primary bodies, which describes the Circular Restricted Three Body Problem (CR3BP). Both the CR3BP and the ER3BP have been studied extensively for satellites orbiting in the Earth-Moon system [27–31].

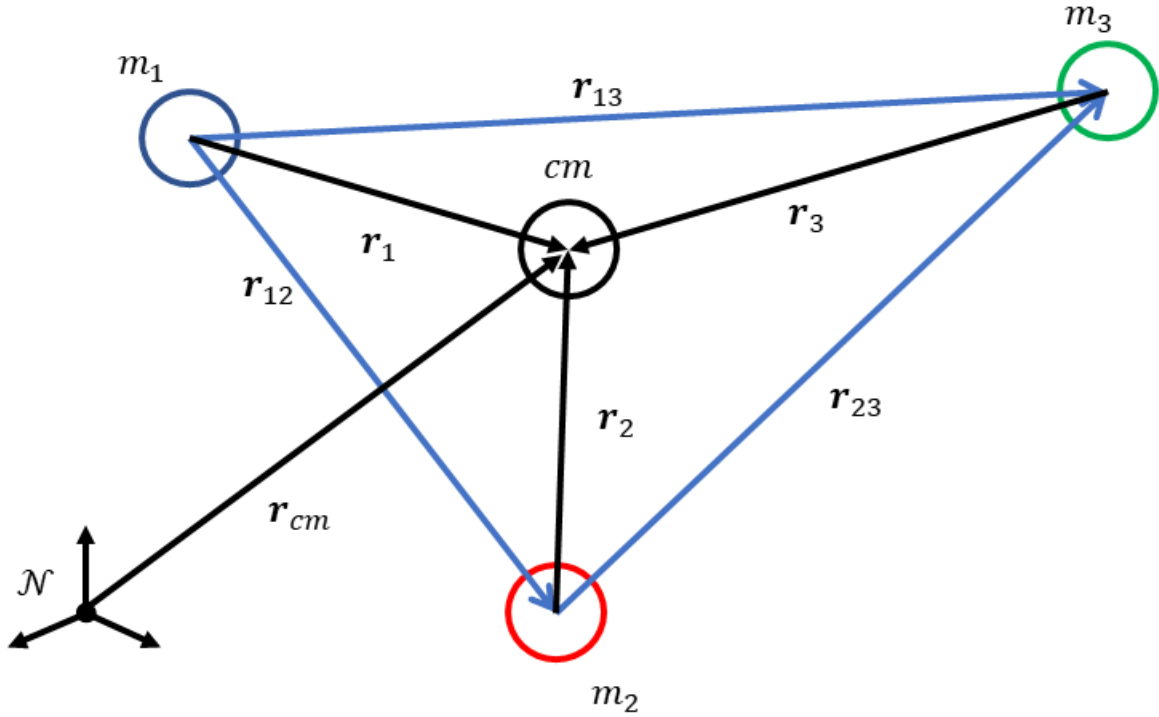


Figure 2.4 Illustration of the general three-body problem.

In the early 1960s, Szebehely compiled all available knowledge on the problem of restricted three-body orbits and published his findings as a reference for future development [27]. His work focused on the CR3BP, but much of the development and the coordinate frame descriptions are useful for the elliptical problem. More recent work includes that of Gordon who studied orbit determination error analysis near libration point orbits, including the use of a linear state-transition

matrix solution to the elliptical three-body problem [28]. Wilson, Gupta, and Grebow analyzed periodic orbits for the circular three-body problem, and proceeded to generate and design periodic orbits in the three-body domain [32–34]. Pernicka studied numerically computing periodic and quasi-periodic orbits for the elliptical restricted problem, using a differential correction algorithm that included a numerically computed state-transition matrix suitable for elliptical motion [35]. Hiday worked on the problem of optimal transfers in the elliptic restricted three-body problem (ER3BP) [36]. Innocenti looked at relative motion in the restricted three body problem in a local-vehicle relative frame, performing error analysis between non-linear, gravitational force linearized, circular three-body problem, and Hill’s equations. It was found that Hill’s equations break down for modeling relative motion in three-body dynamics [37, 38]. The breakdown of Hill’s relative equations for three-body dynamics is a driving motivator for the development of the linearized relative three-body equations of motion in Sections 4.3.1 and 4.3.2.

Innocenti also proposed a simplification to the full non-linear three-body relative motion equations by a linearization of the gravitational force terms via first-order Taylor series and compared the non-linear, linearized gravitational and CR3BP relative solutions. Innocenti’s linearized gravitational form lends itself to future application with analytical solutions for spacecraft proximity operations and control, but still lacks the fidelity given by the full ER3BP [38]. Recent work by Greaves and Scheeres sought to lay out a framework using optical measurements alone for conducting cislunar SSA under the assumption of CR3BP motion. They found that a single space-based sensor placed at the L2 point could provide successful state estimation and maneuver detection for a variety of periodic orbit families, such as NRHO and DRO orbits. However, the optimal control based estimator they proposed required the inclusion of angular rate measurements (or psuedo-rate measurements) to stabilize the filter estimates. [29, 39]. Miller studied a relative navigation problem for spacecraft in NRHOs, and used an Extended Kalman Filter (EKF) to estimate the relative states using a linearized model of the CR3BP which showed promising results; however, the dynamics were limited to the CR3BP [40].

Further research in cislunar SSA includes work by Hall et al. utilizing reachability set theory

for detecting maneuvering objects in cislunar space [41]. The authors conducted extensive Monte Carlo trials for two distinct transfer orbits, one from L1 to L2, and an L2 to GEO maneuver. In both cases, the maneuver was assumed to be bounded, and governed by CR3BP dynamics. LaFarge et al. sought to leverage reinforcement learning for developing station-keeping maneuver strategies and timing. [42]. Once again, though, the analysis is limited to the CR3BP. A common theme among much of the current research on cislunar space limits the dynamical model to the circular problem.

Khoury studied relative motion in the cislunar domain and outlined the relative and non-relative equations of motion for both the CR3BP and the ER3BP. [43]. Further work by Greaves demonstrated that optical observations were sufficient for simultaneous state estimation of both an observer and target spacecraft. [29]. Fowler further studied the problem of cislunar SSA and examined various observer placements ranging from Earth-orbiting observatories to Lagrange point satellites and developed several metrics to aid in the creation of cislunar SSA constellation design. [44]. Recent research by Wilmer et al. has also studied the use of periodic orbits for achieving surveillance of the Earth-Moon Lagrange points [45, 46]. Their study showed that cislunar periodic orbits were highly effective at maintaining continuous surveillance coverage of halo and Lyapunov orbits about the Earth-Moon L_1 and L_2 Lagrange points.

Much of the body of current research outlines relative motion or SSA methods in cislunar space that are limited to the CR3BP. Extension of cislunar relative motion to the ER3BP is highly desirable for developing higher fidelity models of cislunar SSA. This research aims to address these shortcomings by the development of a linearized relative motion model for the ER3BP as well as introduce two orbit determination methods for the CR3BP.

2.3 Linear System Theory: The Matrix Exponential

This section lays out background information necessary for understanding the use of the matrix exponential. For linear time-invariant (LTI) systems, the state-transition matrix (STM) can be computed easily with a matrix exponential of the associated system matrix. The system matrix will be denoted by the matrix “A” for a given LTI system. Assuming a linear dynamics model of the form $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t)$, with system states given by the vector $\mathbf{x}(t)$ completes the LTI model

definition. Furthermore, for an LTI system, the state transition matrix is the matrix exponential of the system matrix [47]. Therefore, the system states can be propagated forward in time with the matrix exponential as: $\mathbf{x}(t) = e^{A\Delta t}\mathbf{x}_0$, where $\Delta t = t - t_0$ gives the difference in time from the initial time t_0 to the current time t , and the exponential mapping moves the states from an initial condition (\mathbf{x}_0) to the current time [47]. There are many methods for computing the matrix exponential; the most straightforward way is by direct computation through a truncation of the infinite Taylor series shown in Eq. (2.2) [48].

$$e^{A\Delta t} = \left[I + A\Delta t + \frac{A^2\Delta t^2}{2!} + \frac{A^3\Delta t^3}{3!} + \dots \right] \quad (2.2)$$

Truncation of the infinite series form requires a method of deciding how many terms to keep. One such way of determining the number of terms to keep is to set a tolerance on the change to the solution between terms. If the change to the solution after a term is below the tolerance, a sufficient number of terms have been kept. As mentioned above, for an LTI system, the matrix exponential is the state-transition matrix and can be used directly to advance system states forward in time as shown in Eq. (2.3).

$$\mathbf{X}(t) = \left[I + A\Delta t + \frac{A^2\Delta t^2}{2!} + \frac{A^3\Delta t^3}{3!} + \dots \right] \mathbf{X}_0 \quad (2.3)$$

When a finite number of terms are used for the matrix exponential, a semi-analytical solution for state evolution can then be written with Eq. (2.3). Analytical solutions for system motion are of great value for linear control and estimation applications, a fact that will be utilized in the development of a relative initial orbit determination technique in Section 5.3.

3 Satellite Tracking and Orbit Estimation

This chapter outlines the first part of autonomous orbit estimation systems. several techniques for satellite tracking and identification from optical measurements are presented. Additionally, two-orbit determination methods for Earth-orbiting satellites (an unscented Kalman filter and batch-least squares) are presented. This chapter will focus on SSA for Earth-orbiting satellites and the two-body problem.

3.1 Optical Measurements

Optical measurements of RSOs are obtained through capturing unresolved images through a ground or space-based optical system. There are many configurations possible for capturing images of satellites. The optical system used during this research is shown in Figures 3.1 and 3.2. The optical tube is a Celestron RASA 1100 telescope [49] mounted on a Software Bisque, Paramount MyT robotic telescope mount. The camera used is an ASI 1600MM Pro cooled monochrome CCD camera.



Figure 3.1 Example optical tracking system during the day.

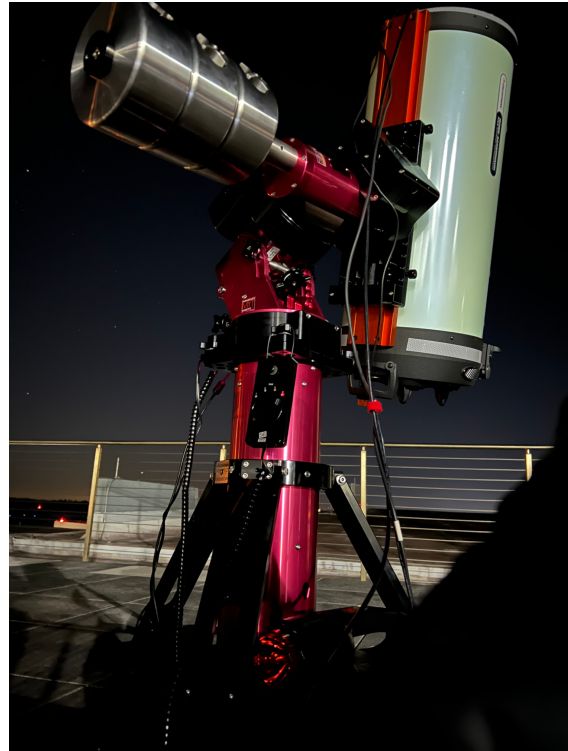


Figure 3.2 Example optical tracking system at night.

3.1.1 Tracking Modes

When imaging RSOs, it will be helpful to define some useful terminology. A “collect” refers to a contiguous set of images of the same object. There are three main tracking modes: inertial tracking, sidereal tracking, and no tracking. Inertial tracking is achieved by using *a priori* knowledge of a spacecraft’s position in the sky by propagation of the objects’ two-line element (TLE) to the observation time. The telescope then uses an open loop tracking process to follow the predicted path of the RSO across the sky. In the resulting images, the RSO being tracked will show up nearly as a point source in images, while stars will form streaks along the direction of motion of the satellite. Additional RSOs that cross the field of view (FOV) will show up as streaks in an image, but not necessarily of the same length or direction as the stars in the image. Figure 3.3 shows an example image of an RSO in a LEO orbit captured in RSO tracking mode. Note that, all stars in the image appear as streaks moving in the same direction, while a single RSO appears as a point-source above the center of the image.



Figure 3.3 RSO tracking mode example image. Stars appear as streaks throughout the image and the tracked RSO appears as a point-source.

In sidereal tracking mode, the telescope tracks at the sidereal rate of the Earth’s rotation. Any

RSOs that appear in the FOV will appear as streaks, while all the stars in the FOV appear as point sources. This mode of tracking is useful if an objects' predicted path is unknown, i.e. for adding objects to a catalog. Figure 3.4 shows an example image captured in sidereal tracking mode. Note that, all stars appear as point-sources in the image and two RSOs appear as streaks above the center of the image.



Figure 3.4 sidereal tracking mode example image. Stars appear as point-sources while RSOs appear as streaks. Two RSOs are visible as streaks above the center of the image.

The final tracking mode is that of fixed pointing, or no tracking. The fixed tracking mode is achieved by turning the telescope motors off. Stars will form streaks in the resulting image. Additionally, RSOs may show up as streaks or point sources in the resulting image. In the case of observing a GEO satellite, the GEO satellite would show up as a point source since the satellite's orbit period is the same as the Earth's rotational period. Each tracking mode has unique applications and advantages. The choice of a mode depends on the observed object and the processing methodology for the images.

3.1.2 Image Processing

Once an image is obtained of a desired RSO, processing steps then must be taken to turn the raw image data into a useful measurement.

Raw images must be processed to reduce noise and increase chances of successfully detecting RSOs [16, 17]. Images are processed using Gaussian filtering, threshold filtering, and intensity based centroiding [50, 51]. Raw images from telescope imaging systems are typically imported as flexible image transport system (FITS) files, a format designed for scientific imaging that preserves raw image data and allows the storage of custom image metadata. Each image is represented mathematically as a matrix of intensities where the x, y coordinates give the pixel location, and the matrix value at that location gives the image intensity. Thus, the intensity map of the raw image is represented as $I_{raw}(x, y)$.

Note that there are many methods to process raw image data to reduce noise. The methodology here represents a basic architecture that provides satisfactory results. The first processing step applied to a raw image is that of Gaussian smoothing to reduce background image noise. A Gaussian kernel ($G(x, y, \sigma)$) is defined based on a given pixel radius, and the kernel is convolved with the raw image to create the smoothed image.

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad (3.1)$$

The smoothed image $I_s(x, y)$ is given by the 2D convolution of the filter kernel with the image [51]:

$$I_s(x, y) = G(x, y, \sigma) * I_{raw}(x, y) \quad (3.2)$$

where the asterisk “*” denotes the convolution operation, and $I_{raw}(x, y)$ is the raw image. After Gaussian smoothing, a threshold filter is applied to remove the background from the image and segment the image background and foreground for centroiding. The threshold filter sets all pixel intensities below a given threshold to zero. The threshold is tuned by η , a user-defined parameter

for the thresholding function shown in Eq. (3.3).

$$I_\tau = \text{mean}(I_s(x, y)) + \eta \times \text{std}(I_s(x, y)), \quad (3.3)$$

$$I_P(x, y) = I_s(x, y) > I_\tau \quad (3.4)$$

where $I_P(x, y)$ gives the final processed image matrix. The threshold filter applied is a global filter for each image, and removes the background for most cases. Adopting a more sophisticated adaptive image threshold would improve the background removal, but is not the focus of this work [52]. After application of all filters, the final processed image is given by $I_P(x, y)$.

In order to generate measurements from an image, another useful processing step is the detection of the centroids of all objects that remain after the threshold filtering. First, a binary mask of the image ($I_B(x, y)$) is created to mark the locations of objects in the foreground of the image. Intensity weighted centroids are then calculated using Eq. (3.5).

$$\mathbf{x}_c = \frac{\sum_{i=1}^N \mathbf{x}_i w_i}{\sum_{i=1}^N w_i}. \quad (3.5)$$

The pixel locations \mathbf{x}_i correspond to pixel locations in $I_B(x, y)$ of the processed image I_P , and $w_i = I_P(x_i, y_i)$ is the intensity of the i^{th} pixel in the processed image. Once all centroids in the camera frame are detected, these x, y coordinates can be transformed into inertial RA, DEC space through a process called plate-solving. Plate-solving is a process very much like spacecraft attitude determination through a lost-in-space algorithm, wherein known star patterns are matched with star patterns in the given image. By knowing the image scale, the process becomes easier to compute possible matches to the detected star patterns. A local installation of the software *Astrometry* is utilized to perform image registration/plate-solving [52]. With an image reduced to image-plane coordinates (x, y) and inertial coordinates (RA, DEC) , the next step to forming a measurement of an observed RSO is the non-trivial process of segmenting any detected RSOs from all other objects in the image. The subject of RSO determination is the focus of the following sections.

3.2 Template Matching for RSO Detection

This section lays out the process for using template matching and autonomous generation of templates for satellite constellation tracking. In order to increase the robustness of RSO identification over time, a method of periodically updating the template image of an RSO constellation is proposed to counter the effects of changing satellite configurations with time. After RSOs have been detected in an image or set of images, RSO locations can be transformed to angular measurements to be used in an appropriate orbit determination method (depending on the number of observations and the domain being observed).

The autonomous template matching method proposed involves matching a model point spread function (PSF) of a single object (RSO) to multiple objects in an image. Finding the location of all RSOs present in a constellation allows the autonomous generation of a template image containing the configuration of the observed group of RSOs. Subsequent images are then matched to the template containing all RSOs in order to find angular measurements to all RSOs and track their locations across each image. Once the constellation has been matched, an initial orbit estimate can be formed using any number of multi-measurement orbit determination (OD) methods. There exist N-measurement forms of Gauss and Gooding's methods which could be used for performing the OD step [11]. However, filtering and batch approaches provide the best orbit estimate given large numbers of observations. In Park et al. [53] a batch filter using the Unscented Transform was utilized to form an orbit estimate from range and angular measurements. Methods utilized in this research will be a sequential Unscented Kalman Filter (UKF) and a batch least squares (BLSQ) approach. There has been previous work done using angles-only methods of non-linear filtering for spacecraft navigation (Ref. [54]) and the concepts applied are very similar to the problem of sequential filtering for angles-only orbit determination [55]. For this research, it will be assumed that an initial orbit estimate to start the filter is available from prior knowledge, or can be generated using a three-measurement implementation of Gooding's method or a batch orbit estimation method [56, 57].

The main goal of the template matching algorithm is to provide optical (angular) measurements

from a series of optical images for use in an initial orbit determination (IOD) or precise orbit determination (POD) method.

Equation (3.6) is used to find the matrix of coefficients and is the heart of the template matching algorithm and is repeated here for convenience. Note that: $\gamma(u, v)$, is the matrix of correlation coefficients, $I(x, y)$ is the input image, $T(\bar{x}, \bar{y})$ is the template image, $\bar{I}_{x,y}$ is the input image mean, and \bar{T} is the template image mean intensity.

$$\gamma(u, v) = \frac{\sum_{x,y} [I(x, y) - \bar{I}_{x,y}] [T(x - u, y - v) - \bar{T}]}{\left[\sum_{x,y} [I(x, y) - \bar{I}_{x,y}]^2 \sum_{x,y} [T(x - u, y - v) - \bar{T}]^2 \right]^{1/2}} \quad (3.6)$$

3.2.1 Autonomous Template Generation Algorithm

In this section the autonomous template generation process will be outlined. The basic idea is to use the template image of a single object (RSO) to create a correlation map in an image containing multiple RSOs. A threshold on matching value is set and all objects above the threshold are considered RSOs. From those detected locations, a template image preserving the location of all RSOs is created. Note that theoretically one could use the template of a single RSO to match for all images. However, by creating a template image from a constellation of $M > 1$ RSO objects, there are more feature points to correlate and matching results are improved thereby, providing more robust matching compared to using the single RSO template. Furthermore, once a template containing multiple objects is formed then the data association problem of determining which RSO associates across images is solved. Otherwise, detected RSOs must be associated across frames, which in itself is a non-trivial process [18]. Figure 3.5 shows the outline of the autonomous template generation process. Further details are included below and summarized in algorithm 1. The first step is to load a raw image that contains the constellation that is desired to be tracked across a series of images. Figure 3.6 shows an example raw and processed image of a constellation of GEO satellites. A cropped portion is shown for clarity to highlight the visible RSOs in the raw image (left) and the processed image (right).

Before attempting to find the RSOs in a raw image to form the template image, the image is

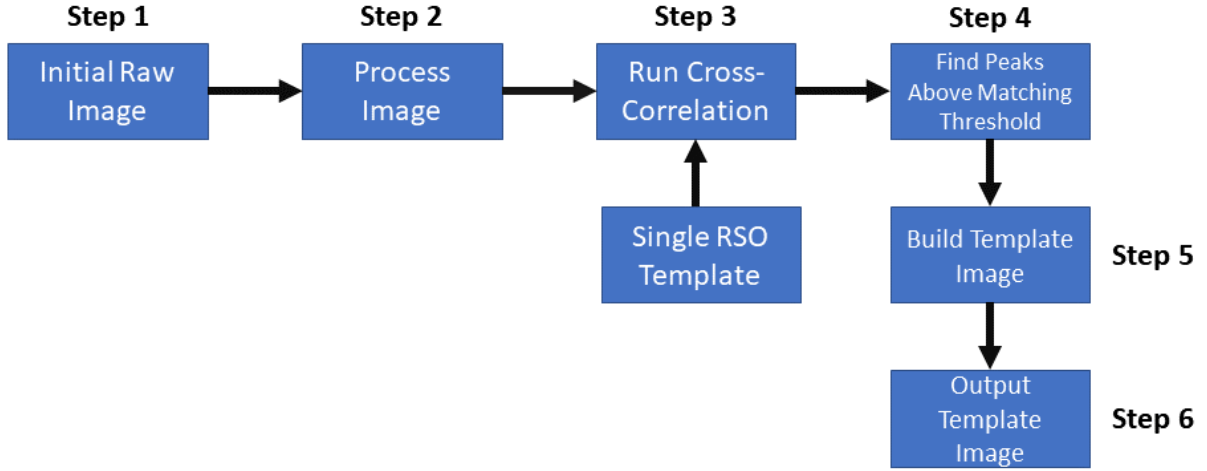


Figure 3.5 Autonomous template generation process.

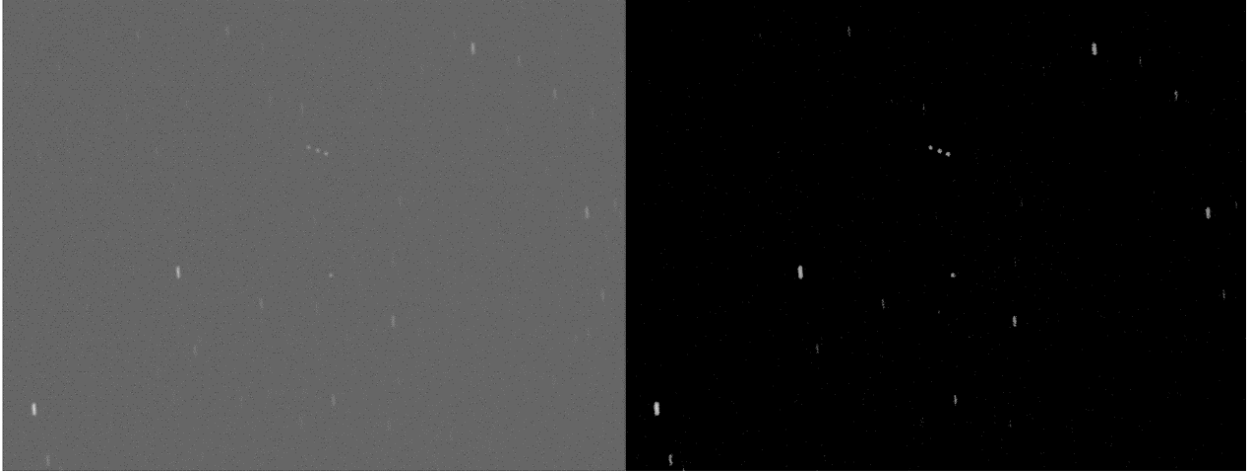


Figure 3.6 Raw (left) and processed (right) cropped portions of example image containing four RSOs.

processed to reduce noise. Gaussian smoothing, and a threshold filter are applied to smooth and then remove the background image noise. Further details in the space image processing can be found in references [17, 21]. The processed image is then matched with an initial template, T_i of a single RSO. The initial RSO template is assumed to be a single PSF that follows a continuous Gaussian distribution.

Step 3 in the process is to run image cross-correlation on a template image of a single RSO. If the initial image $I(x, y)$ is captured such that the stars are streaking through the image and the RSOs

remain stationary, then RSOs will appear as point sources in the image. Therefore cross-correlating a single-point source with the image will give strong peaks in the correlation map at all locations in the image with an RSO. The matrix of coefficients is calculated using Eq. (3.6) and is the heart of the cross-correlation algorithm [21]. When plotted in three dimensions, the correlation coefficients $\gamma(u, v)$ form a surface where height corresponds to the level of correlation between the base image $I(x, y)$ and the template image T . Areas of high correlation correspond to peaks in the surface and give the location of RSOs in the image. Figure 3.7 shows an example correlation map with peaks highlighted by red boxes and red arrows to show the locations of the single RSO template in the target image.

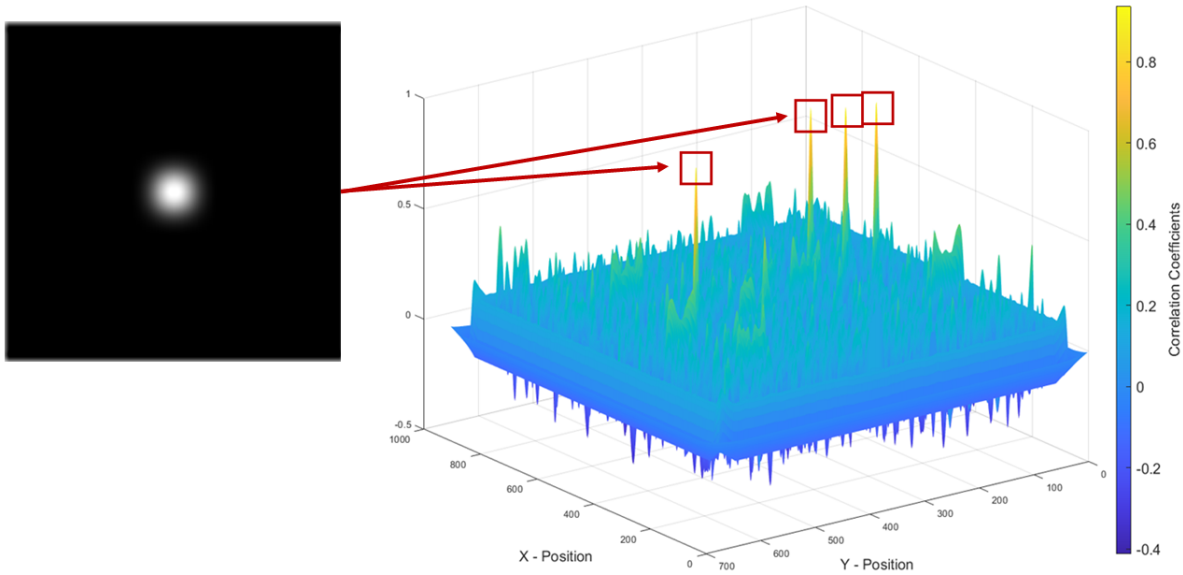


Figure 3.7 RSO template matching peaks.

Once the correlation coefficients have been calculated, step 4 of the algorithm is to find the set of peaks that are above a user-defined threshold. The threshold is set such that strong peaks for RSOs are kept as locations of RSOs and all other coefficients are ignored. Figure 3.7 shows the resulting coefficient matrix from performing cross-correlation between the processed and template images. The four highest peaks (highlighted in red) show the matched locations of the RSOs in the original image. In order to create a template image keeping the location of the RSOs, the peaks in the correlation map are found by centroiding the peak locations in $\gamma(u, v)$. Figure 3.8 shows the

matched locations on a cropped portion of an example raw image.

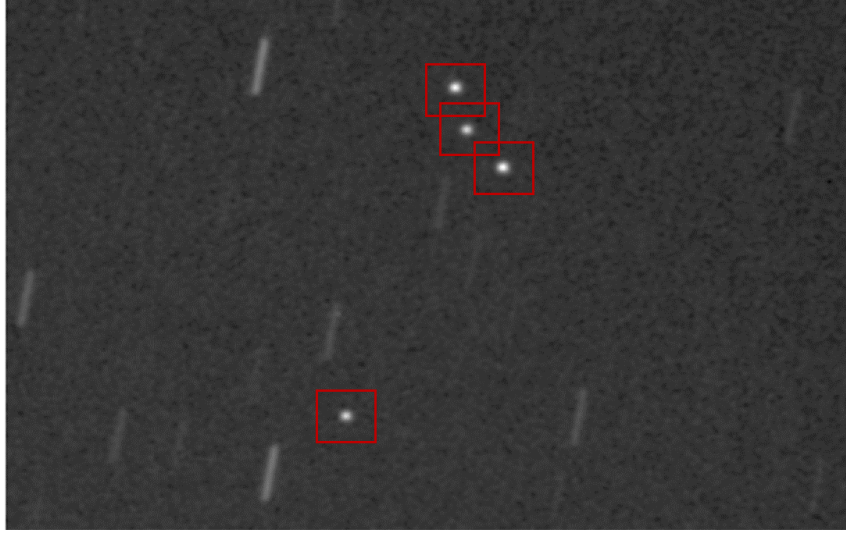


Figure 3.8 Cropped raw image showing matched RSO locations highlighted in red. Detected RSO locations are used to autonomously build a template image for subsequent matching.

Now that the points of maximum correlation have been identified, the next step (step-5) is to create a template image from these matched locations. A template image is generated by first creating a matrix of all zeros with dimensions defined by a rectangular region subtended by the matched locations in the original image (with some padding pixels added around the edges to account for each RSO's PSF). The final step is to create Gaussian point sources in the template image that preserve the original detected RSO locations. An RSO's Gaussian PSF is a continuous function that is approximated by discretizing the PSF into x, y pixel coordinates. Equation (3.7) gives the discrete Gaussian equation where $T(x, y)$ is the template image, B is a brightness scaling factor that in practice is set to 1, σ is the Gaussian standard deviation that determines the width of the PSF, and x_c, y_c are the centroid locations of each RSO. Note that x_c, y_c are the sub-pixel coordinates corresponding to the peak locations from the correlation map $\gamma(u, v)$. Using sub-pixel coordinates for the PSF preserves the true RSO locations in the template image, and better approximates the true continuous RSO PSF in the template image.

$$T(x, y) = \frac{B}{2\pi\sigma^2} \exp\left(-\frac{(x - x_c)^2 + (y - y_c)^2}{2\sigma^2}\right) \quad (3.7)$$

For each RSO in the template image, Eq. (3.7) is used to add the RSO's PSF contribution to the template image. For example in Fig. 3.8 with four RSOs detected, Eq. (3.7) is used four times to add the contribution from each RSO to the template image. Note that, in practice, the Gaussian distribution in Eq. (3.7) falls off fairly quickly, and a maximum radius can be used to define a cutoff distance from the RSO centroid. Algorithm 1 summarizes the steps used to autonomously generate a template image given an input image, and a base template containing a single RSO.

Algorithm 1 Generate Template Image

Require: Input base image $I(x, y)$ initial template of single RSO T_i , and matching threshold ϵ_m

- 1: Process base image to improve detection results (Gaussian smoothing, threshold filtering)
- 2: $I_p \leftarrow \text{process_image}(I(x, y))$
- 3: Run normalized cross correlation on processed image with initial template using Eq. (3.6)
- 4: $\gamma(u, v) \leftarrow \text{NCC}(I_p, T_i)$
- 5: Find peaks above the matching threshold and set these peaks as the locations of M RSOs in the template image
- 6: $\mathbf{X}_{RSO} \leftarrow \gamma(u, v) > \epsilon_m$
- 7: Create template image using dimensions that encompass all elements of \mathbf{X}_{RSO}
- 8: Loop over the number of RSOs (M) and add the PSF contribution from each to the template
- 9: **for** $j = 1$ to M **do**
- 10: Add RSO i to template image using Eq. (3.7)
- 11: **end for**
- 12: **return** Output template image T

An example template image generated autonomously with algorithm 1 is shown in Fig. 3.9. Note that the template image was rotated 90 degrees purely for display purposes. Now that a template image has been created, the template matching process via NCC can be applied to all subsequent images.

3.2.2 Template Matching and RSO Association

Once a template image T has been created using algorithm 1, the template image can then be matched against target image $I(x, y)$ and detected RSOs can be associated across image frames by their respective positions in the template image. Algorithm 2 shows the process of finding a matching template location $\mathbf{X}_{matched}$ in a target image $I(x, y)$ using normalized cross-correlation (NCC).



Figure 3.9 Autonomously generated template image example. (Rotated from actual orientation for display purposes).

Algorithm 2 Template Match Image

Require: Input image $I(x, y)$, template image T , matching threshold ϵ_m

- 1: Process base image $I(x, y)$ to improve detection results
- 2: $I_p \leftarrow \text{process_image}(I(x, y))$
- 3: Run normalized cross correlation on processed image with current template image using Eq. (3.6)
- 4: $\gamma(u, v) \leftarrow \text{NCC}(I_p, T)$
- 5: Find template location from maximum peak
- 6: $\mathbf{X}_t \leftarrow \max(\gamma(u, v) > \epsilon_m)$
- 7: **return** $\mathbf{X}_t = [x_t, y_t]^T$ where x_m, y_m denotes the coordinates of the template in image I

Ensure: Located template at \mathbf{X}_t lies within image $I(x, y)$

Finding the center of the template is only the first step in determining RSO locations and associating them across images. Once the template has been successfully located in target image $I(x, y)$, the next step is to pull out a sub image I_s centered at the detected template location. The sub image is then operated on to associate RSOs from the template image to RSOs in the actual image. Processing the smaller sub-image area corresponding to the detected template location helps to reduce the overall computational burden of the algorithm. Note that, from the generated template, the number of RSOs expected (M) is a known quantity. RSO locations are compared in the template image and matched sub-image area via nearest neighbor for centroids detected. Centroids are calculated using an intensity weighted centroiding method. The Euclidean distance

between centroids is used to determine if a point is a nearest neighbor. For two centroid locations $\mathbf{X}_i, \mathbf{X}_j$ given by x, y coordinates, the distance is given by the Euclidean norm.

$$d = \|\mathbf{X}_i - \mathbf{X}_j\| \quad (3.8)$$

Two points are considered nearest neighbors if $d < \epsilon$ for a given threshold in pixel distance ϵ . Given a set of centroids consisting of x, y pairs in pixel space, finding nearest neighbors reduces to finding the minimum distance pairs between two sets of points. Centroiding the target sub-image centered on the template location gives one set of centroids \mathbf{X}_s , where the s subscript denotes the sub-image. Centroids from the template image are denoted \mathbf{X}_T and can be calculated one time when the template image is created. In order to provide consistent labeling of RSOs in the template image, RSO centroids are labeled with an index number starting from the top left of the image and proceeding to the bottom right of the image in pixel space. Once both target and template image centroids have been found, nearest neighbors are found with the minimum distance pairs below a threshold level (the threshold prevents a non-detection of an RSO from associating incorrectly to a star or noise point that was the closest point but clearly not the RSO). To associate RSOs loop through number of RSOs in the template image (M), and find the nearest neighbor matching to each RSO. If the j^{th} centroid $\mathbf{X}_{T_j} = [x_{s_j}, y_{s_j}]$ is the nearest neighbor given by:

$$NN = \min(d(\mathbf{X}_s - \mathbf{X}_{T_j})) \quad (3.9)$$

then the j^{th} matched RSO location is associated successfully, i.e. $\mathbf{X}_{matched_j} = \mathbf{X}_s$. If no nearest neighbor is found, then the j^{th} RSO was not matched in this image, and the loop continues for the next RSO. The process outlined above for associating RSOs is summarized in algorithm 3.

3.2.3 Template Update Process

Over time the relative orbital motion (or maneuvering) of constellation agents can cause the original template image to become outdated. In order to overcome this difficulty, a process of updating the original template image based on the current detected location of RSOs is implemented.

Algorithm 3 Associate RSOs

Require: Template matched location \mathbf{X}_t , processed image $I_p(x, y)$, template image T , and pixel matching threshold for RSOs ϵ_d

- 1: Pull out sub-image $I_s(x, y)$ from $I(x, y)$ centered at $\mathbf{X}_{matched}$ with dimensions $\mathbb{R}^{p \times q}$ the dimensions of T
- 2: Detect centroids in $I_s(x, y)$ and T
- 3: $\mathbf{X}_s \leftarrow \text{centroids}(I_s(x, y))$
- 4: $\mathbf{X}_T \leftarrow \text{centroids}(T)$
- 5: **for** $j = 1$ to M **do**
- 6: Find nearest neighbor to RSO locations and associate that centroid as RSO_j
- 7: $\mathbf{X}_{T_j} \leftarrow x, y$ centroid of RSO_j
- 8: **if** $\min(d(\mathbf{X}_{ks} - \mathbf{X}_{T_j}) < \epsilon_d)$ **then**
- 9: Nearest neighbor is associated successfully
- 10: $[x_{s_j}, y_{s_j}] = \mathbf{X}_{T_j}$
- 11: $\mathbf{X}_{matched_j} = [x_{s_j}, y_{s_j}]$
- 12: **else**
- 13: $\mathbf{X}_{T_j} \leftarrow$ Not matched for current image
- 14: **end if**
- 15: **end for**
- 16: **return** $\mathbf{X}_{matched_j}$ All matched centroid locations

Zuehlke et al. [22] proposed a method of updating template images in dynamic video scenes. The process is similar to the case of a changing satellite constellation. In the above mentioned paper, the template image of a small UAS quadcopter was updated to track the object in flight. Large changes in illumination, orientation, and size prompted the need for a template update. With satellite constellations, the changes will presumably be on a much slower scale, but the same concept applies.

There are two criteria for determining if a template update is necessary. The first criteria compares the peak correlation value $\gamma_{max} = \max(\gamma(u, v))$ to a defined matching threshold ϵ_m . Note that the matching threshold depends on several factors including image noise. For this research a value of $\epsilon_m = 0.35$ provided a good balance between update frequency and matching accuracy and was determined experimentally. If $\gamma_{max} < \epsilon_m$, then we consider the match to be of insufficient quality to attempt association of RSOs and the template image is updated from the previously successful matching image. Once the template has been updated, a second attempt is made to match the current image $I_k(x, y)$ to the new template. The second criteria for updating the template image is if association of RSOs fails. Failed association could occur for a number of reasons, including image

noise, poor image quality, or failure to find the inertial coordinates of the RSOs from background image stars. If association does fail, the template image is updated from the previous successful matching image and a second attempt is made to match the image and associate RSOs. Once criteria have been met to update the template image, the same process (Algorithm 1) used to generate the initial template T is used to generate a new template $T_{updated}$ from the last image that successfully matched. Algorithm 4 outlines the process of determining if a template update is necessary, and then the process of updating the template image.

Algorithm 4 Template Update Process

Require: Current matched correlation value γ_{max} , matching threshold ϵ_m

- 1: **if** $\gamma_{max} \geq \epsilon_m$ **then**
 - 2: Update template using algorithm 1
 - 3: $T_{updated} \leftarrow \text{generate_template}(I_{t_{k-1}}, T_i)$
 - 4: **else if** $\gamma_{max} < \epsilon_m$ but no RSOs associated **then**
 - 5: Update template using algorithm 1
 - 6: $T_{updated} \leftarrow \text{generate_template}(I_{t_{k-1}}, T_i)$
 - 7: **end if**
-

At this point, all processes necessary to the optical tracking via template matching and association of RSOs are complete. The end goal of the tracking is to provide angular measurements for use in an appropriate orbit estimation method. Angular measurements are computed from optical images through a process known as plate-solving by matching the background stars with a known catalog. Inertial (right ascension (RA) and declination (DEC)) coordinates of all objects in an image are then returned. A customized MATLAB implementation of the program *astrometry.net* is utilized for plate-solving images [52].

3.2.4 Experimental Setup

The experimental setup used for this research is pictured in Fig. 3.2. The imaging system consisted of a Celestron RASA 11 inch telescope with an ASI 1600m monochrome cooled camera mounted on a Celestron CGE Pro mount. The Celestron RASA 11 is an optically fast ($f/2.2$) and short focal length scope (focal length is 620mm) [49]. These characteristics give the telescope a wide field of view (FOV) compared to other 11 inch telescopes and, with the paired camera,

covers an area of sky approximately 1.25×1.65 degrees. The fast and wide optics are ideal for capturing the short exposure images needed to capture RSOs in various orbital domains (LEO, HEO, GEO). Additionally, imaging such a large area of sky allows the capture of multiple RSOs in each image frame. All images were processed on a desktop computer with an AMD Ryzen 5 2600 CPU with 32 Gb of RAM and a 1Tb SSD. The average solution time per image was approximately 10 seconds using non-optimized MATLAB code. Run time could be improved by the use of GPU image processing functions, and parallel processing when feasible.

3.3 Optical Flow for RSO Detection

In this section an alternative to the template matching method for RSO detection is developed. Other methods exist as well but are beyond the scope of this research. The problem of object tracking has been addressed extensively by the computer vision community. Applying ready built tools for object tracking to the specific satellite tracking problem (i.e. picking out moving satellites in a static star-field) holds promise.

A relatively new method used for motion tracking in computer vision research today is called optical flow [58]. Essentially optical flow seeks to find the image plane velocity field caused by the motion of the scene or the observer in a sequence of images. Optical flow has been applied to object tracking in dense object scenes for tracking vehicle motion by Xiang et al. [59] and robot obstacle avoidance and path planning [60]. In relation to spacecraft, optical flow has been applied to space-based ground imaging to detect sandstorms, and other large scale weather phenomenon as observed from Earth orbiting satellites [58]. Optical flow has also been used to track the movement of ground objects (such as ships in a harbor) from space by Du et al. [61] and to estimate the angular velocity of a spacecraft using images from a star tracker camera [62]. The star tracker problem is similar to the satellite tracking problem as the overall motion of the star-field provided a method of determining the inertial spacecraft angular velocity without any star catalog using only image motion. However, none of the mentioned research applies optical flow to the problem of SSA.

This research implements an optical flow algorithm based on the Lucas-Kanade (LK) method to estimate the camera frame velocities of all objects contained in an unresolved space image [63].

A space image can be loosely defined as an optical image taken of the night sky that contains stars, astronomical objects, and possibly RSOs. Given that stars make up the bulk of objects in any space image, it is hypothesized that the overall image velocity provides a way of identifying which objects are stars in the image without the use of a star catalog or plate solving with an application such as Astrometry.net [52]. The velocity estimate for the entire image is taken to be the average velocity of all objects, with the effects of outliers removed. Objects seen to be moving with velocity profiles (magnitude and/or direction) distinct from the overall image motion are classified as potential RSOs. Once all potential RSOs have been identified, the next step in the SSA process is to transform the pixel coordinates to inertial coordinates (angles such as azimuth and elevation or right-ascension (RA) and declination (DEC)) for use in an angles-only orbit determination method. In order to thoroughly test the RSO identification algorithm, a star and RSO image simulator was developed. Simulated imagery provides a testable “ground-truth” scenario for finding RSOs in an image sequence. In addition to the simulated imagery, the algorithm is further tested by processing several actual images captured with the ground-based telescope described in section 3.2.4.

Some preliminary definitions are necessary before delving into the mechanics of optical flow for satellite detection. First, all images can be represented mathematically by a matrix of intensities written as $I(x, y)$, where I is the intensity at the point (x, y) in the image. The optical flow between two images can be thought of as the 2D vector field representing the apparent motion between two consecutive images [64]. As a vector field, optical flow allows the calculation of the displacement and velocities of detected objects in a series of images. There are many methods of computing optical flow, but two of the most common are the methods of Lucas-Kanade and Horn-Schunck [63, 65]. In this work, the Lucas-Kanade method will be followed. Optical flow assumes the following.

1. The pixel intensities of objects are constant between consecutive image frames.
2. The pixels in a neighborhood all follow similar paths of motion in consecutive image frames.

Given these assumptions, a pixel at a time t is represented by $I(x, y, t)$. After a short time, the same pixel will be displaced a small amount (dx, dy) when time dt has passed between images. Thus the

pixel must obey the relationship given in Eq. (3.10).

$$I(x, y, t) = I(x + dx, y + dy, t + dt) \quad (3.10)$$

The optical flow constraint equation is obtained by taking a Taylor series expansion of the right hand side of Eq. (3.10) [58]. Performing the Taylor series expansion, we obtain the following shown in Eq. (3.11),

$$I_x u + I_y v + I_t = 0. \quad (3.11)$$

where $I_x = \frac{\partial I}{\partial x}$ and $I_y = \frac{\partial I}{\partial y}$ are the spatial image derivatives (image gradients), and $I_t = \frac{\partial I}{\partial t}$ is the temporal image brightness derivative. Furthermore, $u = \frac{dx}{dt}$ is defined as the horizontal optical flow, and finally $v = \frac{dy}{dt}$ is defined as the vertical optical flow. In order to solve Eq. (3.11) the LK method takes the original image and divides it into small sections while assuming a constant velocity in the sections. A weighted least-squares equation is then solved to obtain the optical flow fit. The final solution of the least-squares solution is given by Eq. (3.12) [66].

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum_i I_{x_i}^2 & \sum_i I_{x_i} I_{y_i} \\ \sum_i I_{x_i} I_{y_i} & \sum_i I_{y_i}^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i I_{x_i} I_{t_i} \\ -\sum_i I_{y_i} I_{t_i} \end{bmatrix} \quad (3.12)$$

With the optical flow solution, the velocities of each object detected in consecutive images can be found using Eq. (3.12).

3.3.1 Image Simulator

In order to test the optical flow tracking method on space images, a star and satellite image simulator was developed. The simulator assumes that the camera frame is tracking at a constant velocity. Note that this camera frame velocity can be defined to be the rate of star motion, the rate of RSO motion for a given orbit regime, or simply a constant angular rate. For the purposes of this simulator, actual star positions were not important as it is desired to simulate the appearance of stars without having to incorporate actual positions from a star catalog. Star positions are seeded

randomly across the image frame, with star brightness levels also set by a random number generator to some bound. To simulate camera motion, each image frame is propagated forward in time for a user defined number of samples. During each sample, the image is convolved with a Gaussian kernel to simulate the effect of an optical system's PSF that would blur the image and cause objects to streak if they are moving differently than the camera. Image intensities are scaled according to a user specified camera bit depth. Intensities that fall above the bit depth are automatically set to the bit depth to simulate saturated pixels.

In order to simulate satellites, RSO positions are seeded randomly across the image. RSO velocities are set to be different from the image velocity for stars (as is the case in real images). Noise is added to each image frame as white Gaussian noise with a mean level set by a user defined percentage of the bit depth. Figure 3.10 shows an example simulated frame with stars that streak slightly in the image and RSOs that show up as point sources. In order to provide more realistic images, Gaussian white noise was added to corrupt the pristine simulated image. White noise was set to be at a value of 5% of the maximum intensity of the image.

3.3.2 Autonomous RSO Identification Process

Optical flow was utilized to develop an autonomous RSO identification algorithm. The basic outline of the process is shown in Fig. 3.11. Given a sequence of consecutive space images, the algorithm begins by loading the image at time t and the image at $t + dt$, where $t + dt$ is the time of the next image in the sequence. Both images are then processed to reduce noise and improve detection results. The image processing applied involves using a Gaussian smoothing filter to reduce the effects of random image noise, and a threshold filter to remove the background noise level. The image processing methods used are outlined in Section 3.1.2.

Application of a Gaussian smoothing filter suppresses high-frequency noise content of the image. Gaussian smoothing works by blurring the original image slightly by convolution with a Gaussian Kernel, such as that shown by Eq. (3.1). Each smoothed pixel is a weighted average of the neighboring pixels, with the mean weighted towards the central pixels, thus keeping the image in a Gaussian distribution. A typical Gaussian Kernel is given in Eq. (3.1), where (x, y) represent

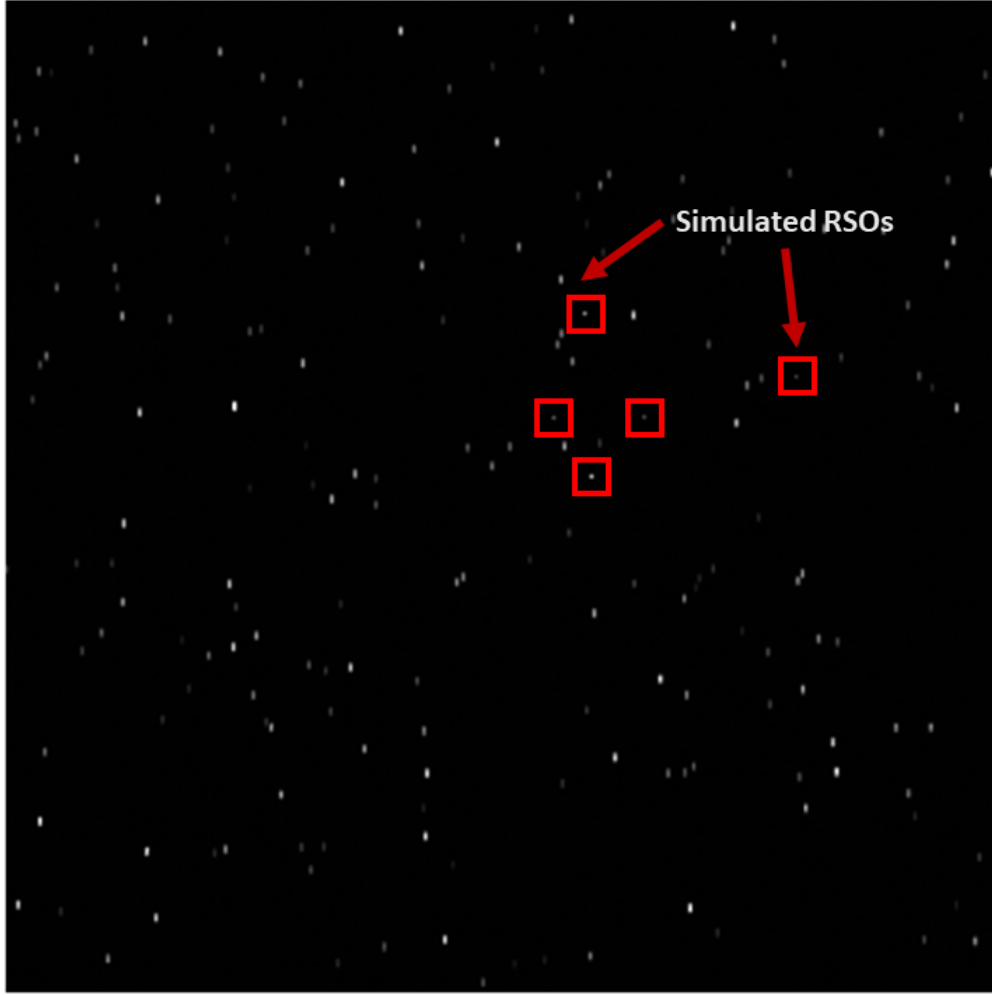


Figure 3.10 Simulated space image for “satellite tracking mode”. All stars in the image appear as short streaks and simulated RSOs appear as point sources and are highlighted by red boxes.

pixel coordinates and σ is the standard deviation for the Gaussian distribution that is desired. A higher σ value results in an image that is smoother.

Applying a threshold on image intensity based on the mean background intensity level provides a simple method of removing any remaining noise after the Gaussian smoothing. All pixels below the chosen threshold level are set to zero intensity, with all remaining illuminated pixels then representing the stars and possible RSOs in an image. The threshold utilized is shown in Eq. (3.4). The final processed image, denoted $I_f(x, y)$, is the result of applying the threshold to the image as shown by Eq. (3.4), where only pixel values above the threshold remain in the final processed image.

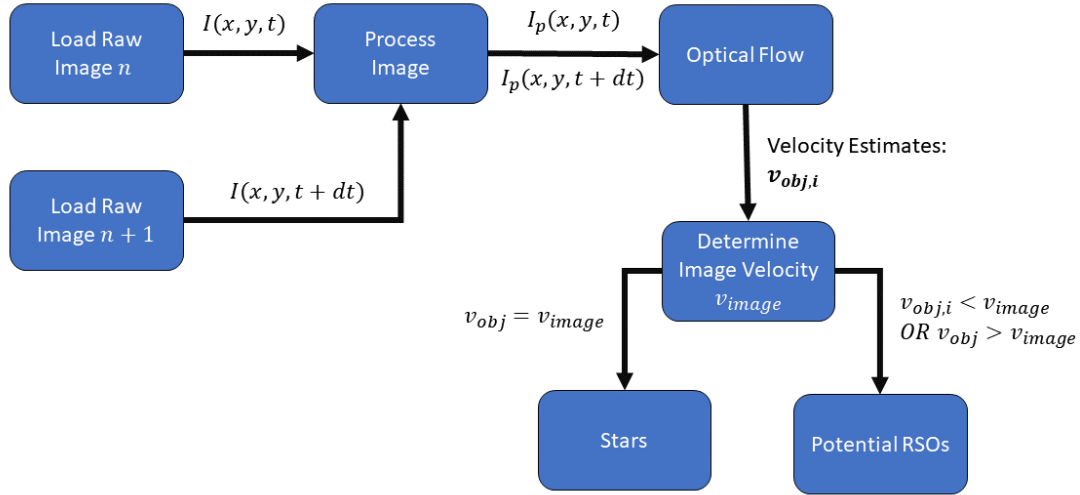


Figure 3.11 RSO identification process block diagram.

Processed images are then passed to the next step in the algorithm to find the optical flow between the two images and estimate the image velocities used for the optical flow step.

Once the processing is applied to both images at each time-step, $I(x, y, t)$ and $I(x, y, t + dt)$, the optical flow between the images is calculated using the (LK) method. The result is a velocity field estimating the motion of all objects detected in the images. A cropped image example of the optical flow between two simulated images is shown in Fig. 3.12. Now, with the velocity field calculated, the next step in the algorithm is to determine the overall image velocity based on the motion of the most prevalent objects in the image (i.e. stars). This is accomplished by looking at the magnitudes of the image velocity for all objects detected in the image and finding the average while rejecting outliers. Calculating this mean image velocity allows the differentiation for objects moving differently than the majority of objects in the image. Since it is posited that RSOs will move different than the majority of objects (stars) in an image, then the overall image velocity will allow the detection and tracking of RSOs.

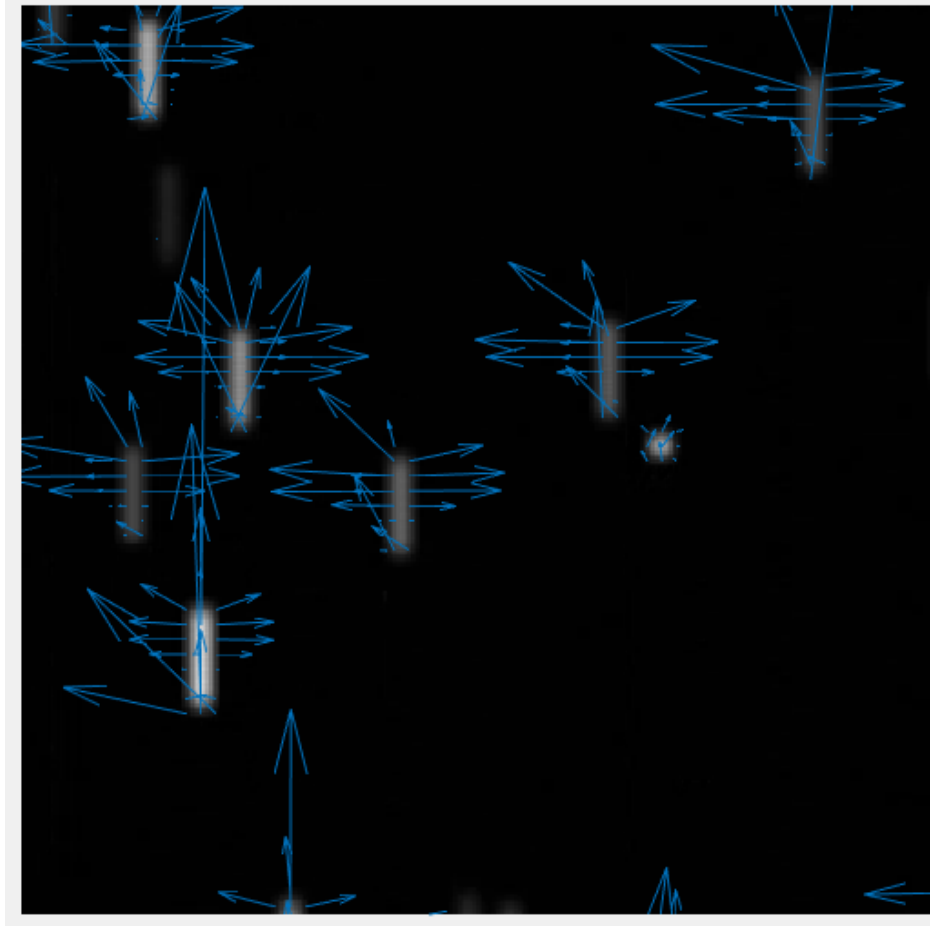


Figure 3.12 Optical flow example for simulated image. Note that stars appear as bloated streaks and an RSO appears as a point source. Blue vectors represent the image velocity of each subset of pixels. Note the higher magnitudes of image velocities on all the stars compared to the RSO near the center right of the image.

3.3.3 Optical Flow Algorithm Summary

Once the optical flow between two consecutive images has been found, the overall image velocity is calculated based on the magnitude of velocities for all image objects. Objects determined to be moving with velocities near the overall image velocity are classified as stars, while objects that move with velocities that are different from the overall image velocity are classified as potential RSOs.

MATLAB includes a direct function to estimate optical flow based on different methods including the LK algorithm. These methods include the use of pre-built functions listed below [67].

- `opticalFlowLK`: This function creates an object to estimate the direction and speed of a moving feature.
- `estimateFlow`: This function takes the `opticalFlowLK` object and estimates optical flow between two consecutive video frames. The method runs recursively as the frames in the video advance.

Utilizing the built-in functions, the velocity profile of the image is found. Given the velocity magnitudes estimated between consecutive images, a “magnitude map” can be formed by forming a matrix of the velocity magnitudes for each pixel location. The image magnitudes can be visualized as an image whose intensities consist of the individual pixel object velocities. The magnitude matrix can then be centroided like a traditional image to find the magnitude and direction of each detected object (star or RSO). MATLAB’s computation of Optical Flow uses a fixed object size and contains one parameter for a noise threshold to determine what is considered a pixel object. By looking at the entire magnitude map then the actual full “object” velocity is found rather than just the velocity of a small portion of the object. An example of the resulting velocity magnitude (cropped version) from a simulated image is shown in Fig. 3.13 with a false color heat map scale applied to visualize the velocity magnitudes. Velocities of each object are computed as the sum of the “intensity” for the detected object in the velocity magnitude matrix. With the velocity of all the image objects, now the maximum, minimum, mean, and median velocity objects can be found. It was found that, given a sufficient number of stars to set the mean velocity, then by looking at the standard deviation of the velocities, RSOs can be picked out as shown in Eq. 3.13.

$$v_{rsok} = v_{mean} \pm 2 \sigma v_i \quad i = 1, 2, \dots, N \quad (3.13)$$

The velocity of the k^{th} RSO detected is given by v_{rsok} and the index i is the object detected in the current velocity magnitude map with a total of N objects detected.

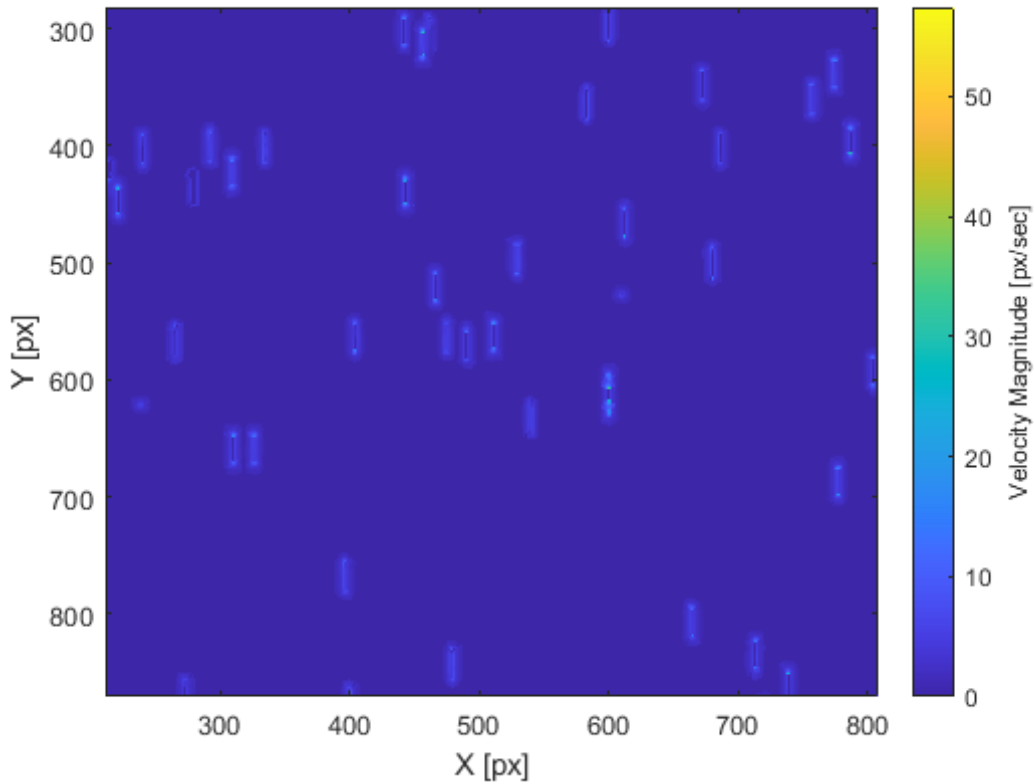


Figure 3.13 Simulated image optical flow velocity magnitude.

3.3.4 Simulated Space Image Results

The MATLAB implementation of the optical flow tracker is shown to be able to successfully track the motion of stars across a set of images for both simulated and experimental images. Figure 3.14 shows a simulated image star field with a group of 5 stationary RSOs. This simulates a GEO constellation of satellites being tracked. Next, Figure 3.15 shows the optical flow for the simulated star field image (computed with the consecutive frame). Blue arrows represent the velocity magnitude and direction estimated for the local pixel regions. Note that the magnitude scale (length of the vectors) is exaggerated for the sake of visualization. Figure 3.16 shows the simulated star field with RSOs highlighted. A cropped result of running the optical flow and RSO identification routine is shown in Figure 3.17. Note that the stars all have large velocity magnitudes while the detected RSO velocity vectors are much smaller. The algorithm was able to successfully differentiate the stars from the RSOs for the simulated image set.



Figure 3.14 Simulated star field.

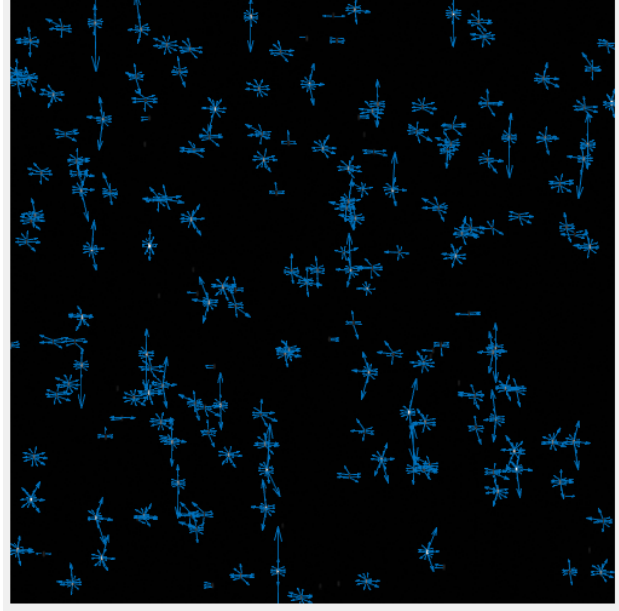


Figure 3.15 Optical Flow from simulated star field.

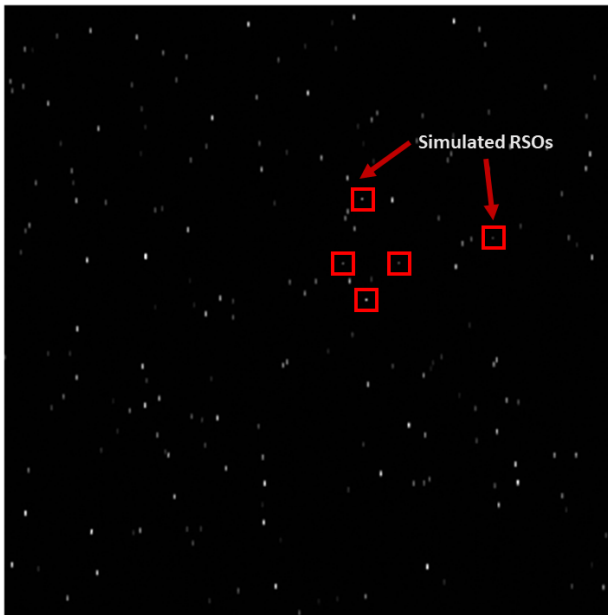


Figure 3.16 Simulated star field with RSOs highlighted.

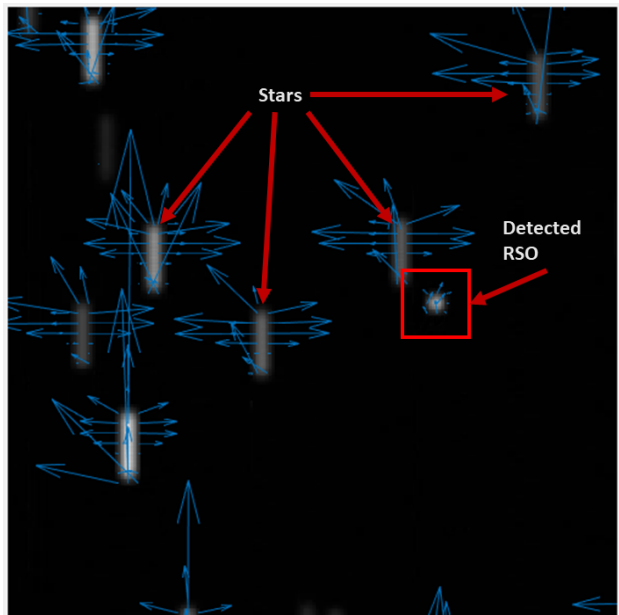


Figure 3.17 Closeup of simulated image optical flow showing identified RSO.

3.3.5 Experimental Images Optical Flow Results

In this section, results from running the optical flow algorithm on ground-based imagery are presented. Figure 3.18 shows a raw space image and Fig. 3.19 shows the resulting optical flow

calculation and RSO detection step. Blue arrows represent the optical flow velocities estimated for various objects detected in the image, and the RSO position is denoted by a red asterisk. This set of images was taken in sidereal tracking mode wherein star positions remain constant while the RSO is seen moving through the frame with a much higher image velocity. The algorithm was able to autonomously label the correct object in the image as an RSO. Figures 3.20 - 3.23 show the results for a series of 70 images of the same object. A single RSO was detected moving from the right center of the image towards the center of the image. Note that detected objects are shown in red, the RSO in green, and the flow velocity vectors in blue.

3.4 Orbit Estimation

Regardless of the method used to attain measurements from a set of images, the end product of a set of inertial angular (RA/DEC) measurements can then be used for orbit estimation of the observed satellite. There are numerous methods for orbit estimation given a minimal set of three pairs of angular measurements using classical methods such as those of Gauss, Lagrange, Laplace, or Escobal [68, 69]. Additionally, for the N -measurement case, there are similarly many methods that could be applied, such as batch-least squares, or N-Gooding [11]. The goal of this section is not to derive multiple methods of orbit determination, but simply to present a selection of implementations of methods that work, while making no claims to be the newest or best method available. The goal is to pair the orbit estimation technique with an RSO identification method from images to provide an orbit estimation result.

3.4.1 Initial Orbit Determination

The end goal of obtaining optical images of RSOs is to find an orbit estimate for the observed line-of-sight (LOS) vectors of an observed RSO. In order to accomplish this, the camera frame centroid (i.e. x, y location) of an RSO must be transformed into an inertial LOS vector from the observer's location to the RSO. When capturing images of RSOs, the background star field is used to obtain the inertial LOS vector through plate-solving with `Astrometry.net`. [52]. Once RA, DEC coordinates are available for all detected RSOs, then it is a simple matter to convert the angles

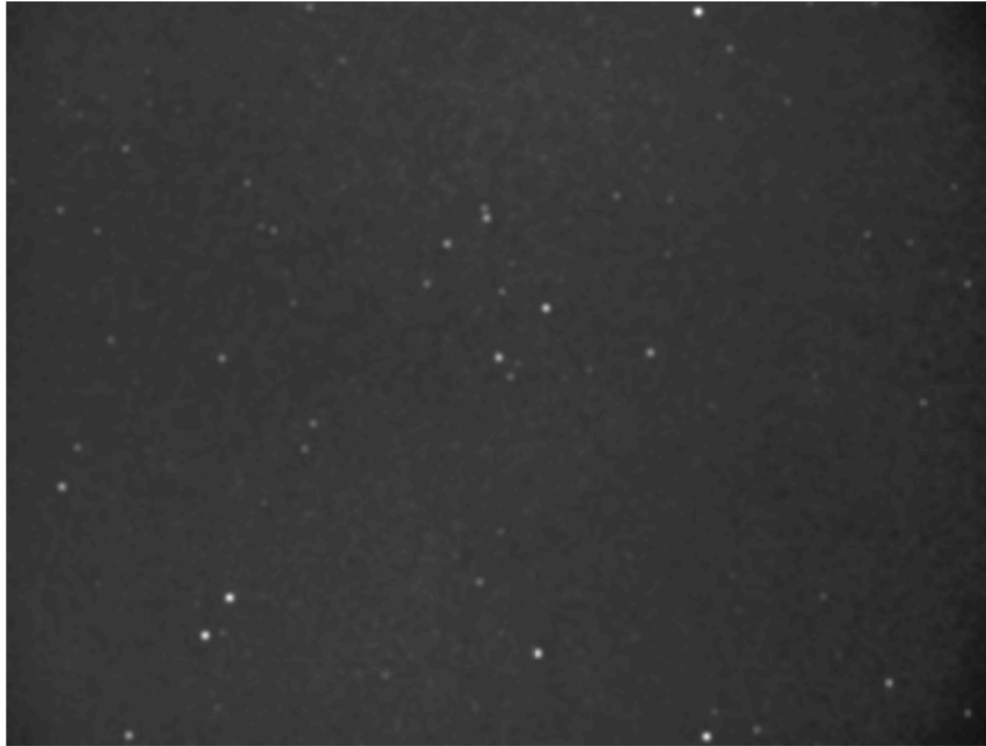


Figure 3.18 Raw ground-based image of GEO satellite.

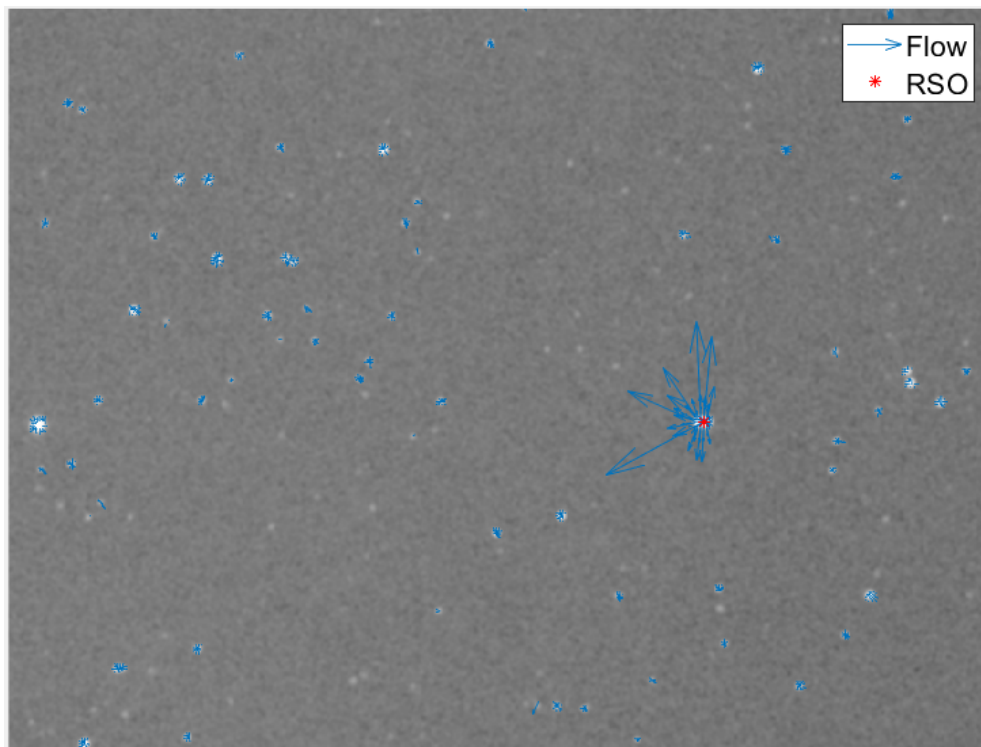


Figure 3.19 Optical flow detection of RSO in real image of GEO satellite.

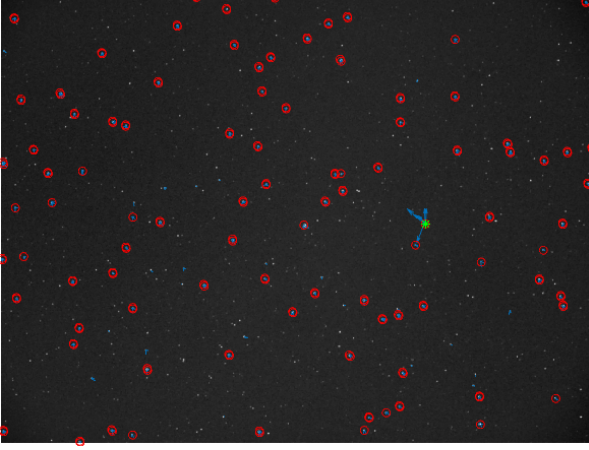


Figure 3.20 RSO detection frame 10.

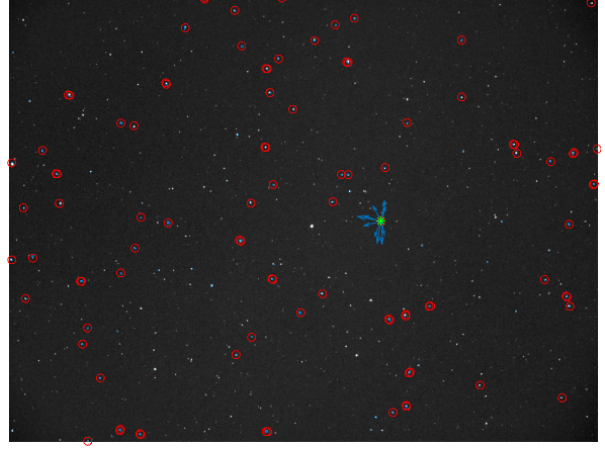


Figure 3.21 RSO detection frame 30.

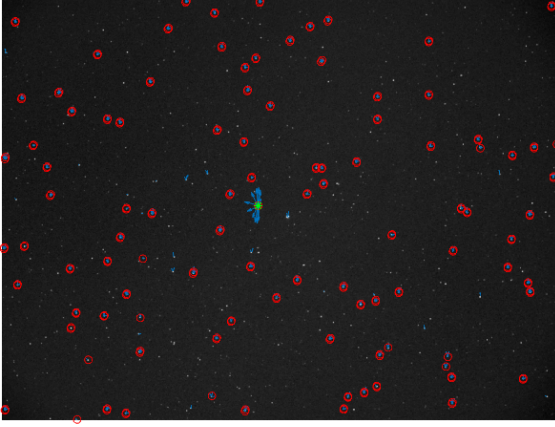


Figure 3.22 RSO detection frame 50.

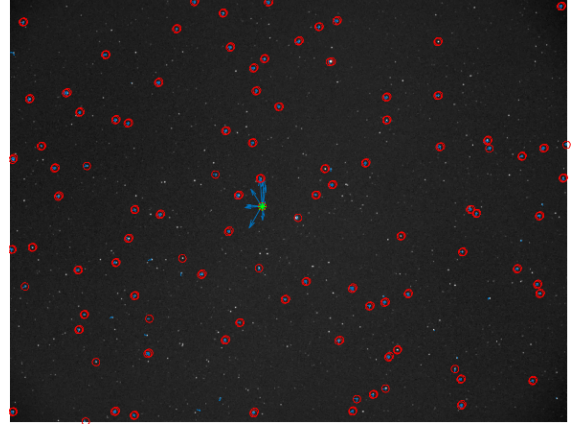


Figure 3.23 RSO detection frame 70.

into LOS vectors using Eq. (3.14), [68].

$$\widehat{\mathbf{L}}_k = \begin{bmatrix} \cos(\delta_{t_k}) \cos(\alpha_{t_k}) \\ \cos(\delta_{t_k}) \sin(\alpha_{t_k}) \\ \sin(\delta_{t_k}) \end{bmatrix} \quad (3.14)$$

where $\widehat{\mathbf{L}}_k$ is the k^{th} LOS vector at time t_k and α_{t_k} and δ_{t_k} are the corresponding RA and DEC angular measurements.

Given the time of observation, LOS vectors, and observer position, the problem is now to fit the observations to an orbit. Classically the problem is solved given observations at three times because the measurement consists of two angles at each time which corresponds to the minimum number of

observations to find an orbiting object's six orbital states [68]. Gooding's method of IOD will only be summarized here; for further details of the implementation of Gooding's method see references [51, 57]. Gooding assumes that measurements are available at three times, and also requires a guess for the range to the satellite at each time. Then Gooding assumes that the range estimates to the first and last measurements are perfect, placing all the "error" on the middle measurement. The orbit is then estimated from the first and last measurements using a Lambert solver and a Newton Raphson method that corrects all three ranges until convergence criteria are met. The result is the orbital position and velocity at the time of the first observation. This initial orbit estimate can be used as the starting point for the UKF orbit estimation method presented in the following section. Note that Gooding's method provides no measure of the uncertainty of the IOD estimate.

3.4.2 Unscented Kalman Filter for Orbit Estimation

Given a time series of angular observations (the output of the template matching algorithm) a UKF is then used to estimate the orbit of each member of the detected constellation at each time step. The UKF is a commonly used non-linear sequential filtering method that begins with an initial state and covariance estimate. First, define the desired state vector of the satellite as the satellite's position and velocity vectors $\mathbf{X} \in \mathbb{R}^6$.

$$\mathbf{X} = \begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T \quad (3.15)$$

The UKF works by taking a set of statistically important points called Sigma Points (denoted by χ_i) that preserve the mean and covariance of an initial distribution of points. In order to preserve the mean and covariance, a minimum of $2N$ sigma points are required where N is the dimension of the state vector ($N = 6$ for the position and velocity states of a satellite). Assumed errors in the states are used to generate the initial covariance and are set as $\epsilon_r = 10$ km and $\epsilon_v = 1$ m/s as the position and velocity *a priori* error estimates. In order to propagate the sigma points forward in time, the nonlinear dynamics of the system are defined as the derivative of the state vector, where μ is Earth's

gravitational parameter and $r = \|\mathbf{r}\|$ is the magnitude of the position vector given by $\mathbf{r} = [x \ y \ z]^T$.

$$\mathbf{f}(\mathbf{X}) = \dot{\mathbf{X}} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} & -\frac{\mu x}{r^3} & -\frac{\mu y}{r^3} & -\frac{\mu z}{r^3} \end{bmatrix}^T \quad (3.16)$$

The UKF requires a set of $2N$ statistically important points to approximate the first two moments of the given distribution, where $N = 6$ is the number of states for a satellite [70]. The selection of the sigma points can be accomplished by choosing a set of $2N$ points such that the mean and covariance of the original states are maintained. The desired statistical distribution of the sigma points is accomplished with weighting parameters tuned by a constant parameter α . A value of $\alpha = 0.95$ was chosen for this research. Begin by setting $L = 2N$ to be the number of sigma points. Then define auxiliary constants $\kappa = 3 - L$, and $\lambda = \alpha^2 (L + \kappa) - L$, as scaling factors to define the spread of the distribution around the mean. Note that, the tuning parameter α typically lies in the range $0 < \alpha < 2$. With the above definitions, the i th sigma point, denoted by χ_i , is calculated as follows.

$$\chi^i = \mathbf{X}_k \pm \sqrt{(L + \lambda)P_k}_i^T \quad (3.17)$$

The distribution mean is given by \mathbf{X}_k at time t_k . The covariance is given by P_k at time t_k and $\sqrt{(L + \lambda)P_k}_i^T$ denotes the i th column of the matrix square root (which in practice is the Cholesky decomposition). The dimensions of P_k are $\mathbb{R}^{6 \times 6}$ and to form $2N$ points each column is taken on either side of the mean of the state. Once a set of sigma points is established at the initial time t_0 , the sequential part of the filter can begin. The initial state X_0 is used to initialize the covariance at time t_0 as P_0 . Once the initial covariance and state are set, the sigma points are propagated forward in time to the next measurement time through the nonlinear function $\mathbf{f}(\mathbf{X})$, defined by the state propagation model given in Eq. (3.16).

$$\chi_k^i = \mathbf{f}(\chi_{k-1}^i) \quad (3.18)$$

Once the sigma points are propagated to the next measurement time, the mean of the propagated states ($\widehat{\chi}$) is found by averaging the propagated sigma points as:

$$\widehat{\chi}_k^- = \frac{1}{2N} \sum_{i=1}^{2N} \chi_k^i \quad (3.19)$$

where the minus superscript denotes that this mean is calculated after the propagation step, but before the measurement update. Furthermore, the i^{th} superscript denotes the i^{th} sigma point as stated above. From the propagated states and new mean, the covariance before the measurement also needs to be calculated and is found through equation (3.20).

$$P_k^- = \frac{1}{2N} \sum_{i=1}^{2N} [\chi_k^i - \widehat{\chi}_k^-] [\chi_k^i - \widehat{\chi}_k^-]^T + Q \quad (3.20)$$

Once the mean and covariance at the current measurement time t_k are found, new sigma points incorporating the updated covariance are computed to find the predicted measurement at the current time-step. The measurement model is given by the calculation of topocentric RA and DEC angles. First, the relative vector between the site and the predicted position of the RSO is found as $\rho_k = \chi_k^i - \mathbf{r}_{site_k}$. Next, calculate the estimated LOS vector as $\widehat{\mathbf{L}}_k = \frac{\rho_k}{\|\rho_k\|}$. From the LOS vector $\widehat{\mathbf{L}}_k$, the topocentric RA and DEC denoted α_k and δ_k are calculated with Eqs. (3.21) - (3.23).

$$\alpha_k = \text{atan2}(\widehat{\mathbf{L}}_k(2), \widehat{\mathbf{L}}_k(1)) \quad (3.21)$$

$$\delta_k = \text{asin}(\widehat{\mathbf{L}}_k(3)) \quad (3.22)$$

$$\mathbf{h}(\mathbf{X}_k, t_k) = \begin{bmatrix} \alpha_k & \delta_k \end{bmatrix}^T \quad (3.23)$$

The four quadrant inverse tangent function is given by $\text{atan2}()$. The inverse sine function is given by $\text{asin}()$ and $\widehat{\mathbf{L}}_k(i)$ $i = 1, 2, 3$ give the x, y, z components of the estimated LOS vector. Finally, the vector \mathbf{h} represents the nonlinear measurement function utilized by the UKF.

Now that the measurement function has been defined, the next measurement update step of the UKF can be accomplished. First measurements $\hat{\mathbf{y}}_k^i$ are calculated for all sigma points through the

measurement function $\mathbf{h}(\chi_k^{-i}, t_k)$, followed by finding the mean measurement $\bar{\mathbf{y}}_k$.

$$\chi_k^{-i} = \widehat{\chi}_k^- \pm \sqrt{(L + \lambda)P_k^-}_i^T \quad (3.24)$$

$$\hat{\mathbf{y}}_k^i = \mathbf{h}(\chi_k^{-i}, t_k) \quad (3.25)$$

$$\bar{\mathbf{y}}_k = \frac{1}{2N} \sum_{i=1}^{2N} [\hat{\mathbf{y}}_k^i] \quad (3.26)$$

$$(3.27)$$

The measurement covariance P_y is then calculated given the measurement noise R_k (Eq. (3.28)). Next the measurement and state cross covariance matrix P_{xy} is calculated using the sigma points and their mean, and the estimated measurements and their mean.

$$P_y = \sum_{i=1}^{2N} [\hat{\mathbf{y}}_k^i - \bar{\mathbf{y}}_k][\hat{\mathbf{y}}_k^i - \bar{\mathbf{y}}_k]^T + R_k \quad (3.28)$$

$$(3.29)$$

In order to update the state and covariance with the measurement information, the Kalman gain must first be calculated.

$$K = P_{xy}P_y^{-1} \quad (3.30)$$

After the measurement step, the state is updated using the Kalman gain, and the difference in the true measurement $\tilde{\mathbf{y}}_k$, and the mean estimated measurement $\bar{\mathbf{y}}_k$. The state covariance is also updated with the measurement data using the Kalman gain as shown in Eq. (3.32).

$$\widehat{\chi}_k^+ = \widehat{\chi}_k^- + K[\tilde{\mathbf{y}}_k - \bar{\mathbf{y}}_k] \quad (3.31)$$

$$P_k^+ = P_k^- + KP_yK^T \quad (3.32)$$

Note that the “+” superscript denotes the state and covariance after the measurement update has been applied. The entire measurement and update step encompasses equations (3.24) - (3.32).

After the update step, the new mean and covariance ($\widehat{\mathbf{X}}_k$ and P_k^+) are used to calculate an updated set of sigma points which are then propagated to the next measurement time, with the process continuing as before. The UKF process is summarized in algorithm 5. The output of running the UKF algorithm on a set of RA, DEC measurements of an RSO is the state and covariance of the RSO at each measurement time.

Algorithm 5 Unscented Kalman Filter for Orbit Estimation

Require: Initial state \mathbf{X}_0 , covariance P_0 , and set of measurements $\tilde{\mathbf{y}}_k$

```

1: for  $k = 1$  to the number of measurements ( $N_{images}$ ) do
2:    $\mathbf{X}_{k-1}^i \leftarrow \widehat{\mathbf{X}}_{k-1}$  Calculate  $2N$  sigma points at time  $t_{k-1}$  using Eq. (3.17)
3:    $\mathbf{X}_k^i \leftarrow$  Propagate sigma points  $\mathbf{X}_{k-1}^i$  to measurement time  $t_k$  using Eq. (3.18)
4:    $\widehat{\mathbf{X}}_k^-, P_k^- \leftarrow$  Find new mean and covariance from propagated states using equations (3.19) - (3.20)
5:    $\hat{\mathbf{y}}^i \leftarrow$  Generate measurements for each sigma point using equations (3.23) - (3.26)
6:    $K \leftarrow P_{xy} P_y^{-1}$  Find Kalman gain using equations (3.28) - (3.30)
7:    $\widehat{\mathbf{X}}_k^+, P_k^+ \leftarrow$  Update state and covariance using the Kalman gain and actual measurements using equations (3.31) - (3.32).
8:   Set new mean and covariance to calculate sigma points for next iteration.
9:    $\widehat{\mathbf{X}}_{k-1} \leftarrow \widehat{\mathbf{X}}_k^+$  and  $P_{k-1} \leftarrow P_k^+$ 
10:  return Mean orbital states and covariance from each iteration
11:   $\mathbf{X}_{out} \leftarrow \widehat{\mathbf{X}}^+$ 
12:   $P_{out} \leftarrow P^+$ 
13:  Continue to next iteration  $k = k + 1$ 
14: end for
```

3.4.3 Autonomous RSO Constellation Orbit Determination Algorithm

Now that all the individual pieces have been discussed, the full algorithm from raw image input to a final orbit estimate is presented. The end goal of the method is to provide an orbit estimate for all RSOs in a given image set. The algorithm is general in that the user need only input an image set and appropriate thresholds and the rest will be performed autonomously. Note that images are assumed to be in the “.fit” format with an appropriate data header that contains image metadata such as image sensor pixel size, exposure times, and image capture times. Also, note that the current implementation of the algorithm will process all images through the template matching portion, and then proceed to the UKF orbit estimation process. These steps can be easily combined into a single loop such that the UKF estimate is updated as each image is processed. The overall process

to generate an orbit estimate given an input set of images is outlined in algorithm 6.

Algorithm 6 Autonomous Template Matching with UKF Orbit Estimation

Require: Input image set with N images, template image of single RSO T_i , matching threshold ϵ_m , association distance threshold ϵ_d

- 1: Generate template image from first image using algorithm 1
- 2: $T \leftarrow \text{generate_template}(I_1(x, y), T_i)$
- 3: **for** $k = 1$ to N **do**
- 4: Solve template match of image I_k with NCC using algorithm 2
- 5: $\mathbf{X}_t, \gamma_{max} \leftarrow \text{template_match}(I_k, T, \epsilon_m)$
- 6: **if** $\gamma_{max} < \epsilon_m$ **then**
- 7: Update template image using algorithm 4
- 8: $T \leftarrow \text{update_template}(I_{k-1}, T_i)$
- 9: **end if**
- 10: Associate RSOs from template image T to matched location \mathbf{X}_t using algorithm 3
- 11: $\mathbf{X}_{matched} \leftarrow \text{associate_RSOs}(\mathbf{X}_t, I_k, T, \epsilon_d)$
- 12: **if** Association Failed **then**
- 13: Update template image using algorithm 4
- 14: $T \leftarrow \text{update_template}(I_{k-1}, T_i)$
- 15: **end if**
- 16: **end for**
- 17: **return** Output the associated measurement sets $\tilde{\mathbf{y}}_j$ for each detected RSO
- 18: Run UKF on measurements of each RSO
- 19: **for** $j = 1$ to M **do**
- 20: Generate initial orbit estimate using Gooding's method
- 21: $\mathbf{X}_0 \leftarrow \text{gooding}(\dots)$
- 22: Run UKF using algorithm 5 for RSO $_j$
- 23: $\mathbf{X}_{out_j}, P_{out_j} \leftarrow \text{UKF}(\mathbf{X}_0, P_0, \tilde{\mathbf{y}}_j)$
- 24: **end for**
- 25: **return** Orbital state information for all RSOs

3.4.4 Batch Orbit Estimation - Differential Correction Algorithm

Batch least-squares is a common estimation approach for solving a non-square system of equations for system parameters. In this case, solving for the orbital parameters of an observed satellite (6 states given either by an initial position and velocity, or the classical orbital elements of the satellite.)

This section presents the batch orbit estimation algorithm for Earth-orbiting satellites. The development of Vallado in Chapter 10 of reference [68] is followed. The process of using a batch least squares estimator to find an orbit is referred to as differential correction. An initial state vector

guess, often supplied by a classical IOD method, is iteratively corrected until either an iteration limit is reached or an update tolerance is met.

A number of preliminaries are necessary before laying out the equations for the differential corrector. Observations are assumed to be angular observations (RA and DEC) obtained from optical images. The differential corrector works by propagating a possible initial state forward in time and comparing estimated measurements to true measurements. The measurement function is given by finding the angles given by the unit vector to the observed satellite, where the relative vector from the site to the satellite is given by: $\boldsymbol{\rho}_k = \mathbf{r}_{nom_k} - \mathbf{r}_{site_k}$. The unit vector is then found with: $\hat{\boldsymbol{\rho}}_k = \frac{\boldsymbol{\rho}_k}{\|\boldsymbol{\rho}_k\|}$. The measurement function finds the azimuth and elevation angles to the observed satellite using Eq. (3.33), which is a function of the satellite and observer positions only.

$$\mathbf{h} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \text{atan2}(\hat{\rho}_y, \hat{\rho}_x) \\ \text{asin}(\hat{\rho}_z) \end{bmatrix} \quad (3.33)$$

$$\hat{\mathbf{y}}_k = \mathbf{h}(\mathbf{r}_k, \mathbf{r}_{site_k}) \quad (3.34)$$

The subscripts of the relative vector ρ correspond to the x, y, z components of the observed LOS unit vector respectively. Thus the measurement function is purely dependent on the current position of the observer and the observed object. Measurement residuals are calculated as the difference between the true and estimated measurements at a given time t_k with Eq. (3.35).

$$\tilde{\mathbf{b}} = \mathbf{y}_k - \hat{\mathbf{y}}_k \quad (3.35)$$

A key component of least squares is the solve-for vector, or nominal state. For the application of orbit determination, the solve-for vector will be defined as a satellites initial position and velocity vector given by: $\hat{\mathbf{x}}_0 = \begin{bmatrix} \mathbf{r}_0^T & \mathbf{v}_0^T \end{bmatrix}$. The core of least squares estimation is the use of the Gauss normal equations. For batch least squares, the normal equations can be written to update the state estimate in one step, or in the case of large numbers of measurements, a loop form can be used. Given the standard deviations of the elements of the desired solve-for vector, an *a priori* guess for the system

covariance $P_{\Delta x_0} = \text{diag}\left(\left[\sigma_1^{-2} \quad \sigma_2^{-2} \quad \dots \quad \sigma_n^{-2}\right]\right)^{-1}$ can be formed. Note that σ_i = the standard deviation of the i th element of the desired solve for vector given by \mathbf{x}_0 . Note that a weighting matrix W is often used to scale the impact of noise on different solve-for states. The Gauss normal equation to update the state estimate can then be written as shown in Eq. (3.36).

$$\delta \hat{\mathbf{x}} = \left[A^T W A\right]^{-1} A^T W \tilde{\mathbf{b}} = P A^T W \tilde{\mathbf{b}} \quad (3.36)$$

Once the update term to the state is found, the solve-for (often referred to as the “nominal”) state \mathbf{x}_0 is updated by adding the $\delta \hat{\mathbf{x}}$ term to the previous value of \mathbf{x}_0 .

$$\hat{\mathbf{x}}_0 = \hat{\mathbf{x}}_0 + \delta \hat{\mathbf{x}} \quad (3.37)$$

The system matrix A can be defined by finite differencing, or by analytical derivation of the equations of motion. Using an analytical approach, then A is the matrix relating the observation vector to the solve-for vector. The partial derivative can be expanded by the chain rule to be the partial of the observation matrix with the states and the states with respect to the initial states (which is the state-transition matrix).

$$A = \frac{\partial \mathbf{h}}{\partial \hat{\mathbf{x}}_0} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \hat{\mathbf{x}}_0} = H \Phi \quad (3.38)$$

The observation partial matrix (H) is the partial of the measurement function $\mathbf{h}(\mathbf{x})$ with respect to the satellite position and velocity states $\mathbf{x} = \begin{bmatrix} \mathbf{r}^T & \mathbf{v}^T \end{bmatrix}$.

$$H = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial \alpha}{\partial \mathbf{r}} & \frac{\partial \alpha}{\partial \mathbf{v}} \\ \frac{\partial \beta}{\partial \mathbf{r}} & \frac{\partial \beta}{\partial \mathbf{v}} \end{bmatrix} \quad (3.39)$$

However, since the measurement equations have no dependence on the spacecraft velocity, the right column terms of Eq. (3.39) will be zero, making only the partials with respect to position non-zero. Note that the measurement vector is defined as in Eq. (3.33) with the relative vector between the

spacecraft position and the site defined as: $\boldsymbol{\rho} = \mathbf{r} - \mathbf{r}_{site}$. The magnitude of the distance between the site and the satellite is then given by $\rho = \|\boldsymbol{\rho}\|$.

$$H = \begin{bmatrix} -\frac{\rho_y}{\rho_x^2 + \rho_y^2} & \frac{\rho_x}{\rho_x^2 + \rho_y^2} & 0 & 0 & 0 & 0 \\ -\frac{\rho_x \rho_y}{\rho^3} & -\frac{\rho_y \rho_z}{\rho^3} & \frac{\rho_x^2 + \rho_y^2}{\rho^3} & 0 & 0 & 0 \end{bmatrix} \quad (3.40)$$

The observation partial matrix H is commonly referred to as the sensitivity matrix in the literature [68]. The two-body STM matrix Φ can be obtained from chapter 10 of Ref. [68]. With all necessary terms defined, the batch least squares algorithm for orbit determination can be described.

The process begins by inputting a set of observations $\mathbf{y}(t_k)$ and an initial guess to the nominal state \mathbf{x}_0 . Note that the initial state guess need not be close to the truth. While the BLSQ algorithm is not converged, the nominal state is propagated to all measurement times and a set of estimated measurements $\hat{\mathbf{y}}_k$ is generated using the measurement function defined in Eq. (3.33). The measurement residuals vector $\tilde{\mathbf{b}}$ is formed for each measurement. The system A matrix is then calculated via finite differencing or analytical means. For large numbers of measurements, rather than finding the matrix products $A^T W A$ and $A^T W \tilde{\mathbf{b}}$ for all N measurements at once, the matrices can be calculated at each measurement time and then summed to find the total state correction using Eq. (3.36). The root mean square (RMS) of the residual vector can then be evaluated for convergence after the state update.

$$RMS = \sqrt{\frac{\tilde{\mathbf{b}}^T W \tilde{\mathbf{b}}}{N}} \quad (3.41)$$

If the RMS is less than a given tolerance ϵ , then the BLSQ has converged: $RMS < \epsilon$. If not converged then the nominal state vector $\hat{\mathbf{x}}_0$ is updated using Eq. (3.37). The batch least squares process for orbit determination is summarized in Algorithm 7.

3.5 Orbit Determination Results

The full autonomous template matching algorithm was tested on a set of ground-based imagery of the geostationary object AMAZONAS-2 (NORAD ID 35942) and several surrounding satellites. A total of 1095 images were captured over a 2 hour period, where 745 images were processed

Algorithm 7 Batch Least Squares for Orbit Estimation

Require: Measurement vector \mathbf{y}_k , initial covariance $P_{\Delta x_0}$, an initial guess for the satellite nominal state $\hat{\mathbf{x}}_0$, and a convergence tolerance ϵ

- 1: **while** $RMS < \epsilon$ **do**
- 2: **for** $k = 1$ to N the number of measurements **do**
- 3: Propagate the nominal state to measurement time t_k and find estimated observations.
- 4: $\mathbf{y}(\hat{t}_k) \leftarrow \mathbf{h}(\hat{\mathbf{x}}_0)$
- 5: Compute the measurement residual vector \tilde{b}_k using Eq. (3.35).
- 6: Compute A_k matrix for time-step t_k and find compute quantities $A_k^T W A_k$ and $A_k^T W \tilde{b}_k$
- 7: **end for**
- 8: Update the nominal state using the accumulated $A_k^T W A_k$ and $A_k^T W \tilde{b}_k$ quantities to calculate the state correction.
- 9: $A^T W A \leftarrow \sum_{k=1}^N A_k^T W A_k$
- 10: $A^T W \tilde{b} \leftarrow \sum_{k=1}^N A_k^T W \tilde{b}_k$
- 11: $\delta \hat{\mathbf{x}}_0 = [A^T W A]^{-1} A^T W \tilde{b} = P A^T W \tilde{b}$
- 12: $\hat{\mathbf{x}}_0 \leftarrow \hat{\mathbf{x}}_0 + \delta \hat{\mathbf{x}}_0$
- 13: Check for convergence using Eq. (3.41).
- 14: **end while**
- 15: Batch least squares estimation is complete. Output nominal state $\hat{\mathbf{x}}_0$

through the template matching algorithm (the remaining images were discarded due to clouds blocking the telescope field of view).

3.5.1 Template Matching Results

Algorithm 6 was utilized to process all images. An initial template image was autonomously generated from the first image in the data set. The correlation map $\gamma(u, v)$ used to generate the initial template image is shown in Fig. 3.24. Figure 3.24 (a) shows all of the correlation map values plotted as a 1D array, while Fig. 3.24 (b) shows the 3D peaks of the correlation map. The peaks of the six RSOs contained in the image are highlighted by red circles in both the 1D and 3D representations. The x, y coordinates of these peaks (shown in Fig. 3.24 (b)) are the points used to build the initial template image. Note that the two views are shown to illustrate the advantage of building a template using multiple points rather than using only a template with a single RSO PSF (further details given below, see Fig. 3.27).

The template image that was autonomously generated is shown in Fig. 3.25. Note that, for display purposes only, the template has been rotated 90° from its actual orientation. This template

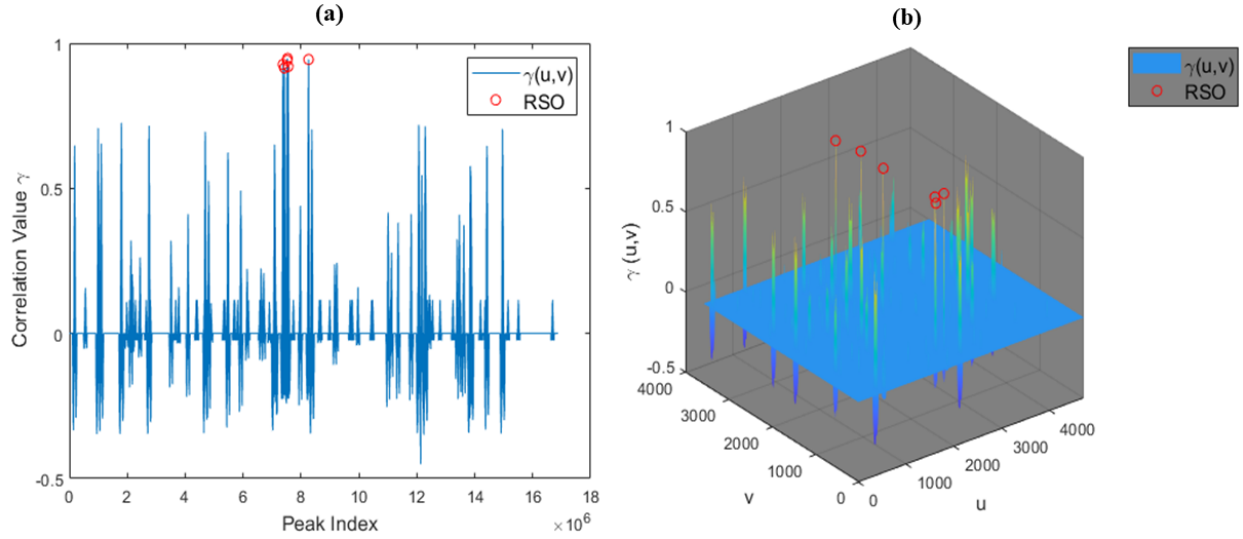


Figure 3.24 (a) Correlation values in a 1D array with RSOs in red. (b) Correlation 3D map with RSOs in red.

image was then matched to the proceeding in the data set until criteria were met to update the template image. One example of successful matching is shown in Fig. 3.26. The associated positions



Figure 3.25 Rotated template image.

of RSOs in the image are marked with white arrows and text labels. Note that the template RSO locations shown in Fig. 3.25 correspond directly to the associated positions in the matched image. The corresponding correlation map $\gamma(u,v)$ in 1D and 3D formats are shown in Fig. 3.27. When the template containing multiple RSOs is used to match against the image, note the clear single peak corresponding to the template location unlike the multiple peaks for the case of generating the template image. This dominant peak increases the accuracy of finding the template location in the target image and is the chief advantage of using a template image containing multiple points.

In order to show the process of when template updates occur, the peak correlation value matched for each image is shown plotted against time in Fig. 3.28. The red vertical lines denote points in time where criteria were met to update the template image. Once the template was updated,

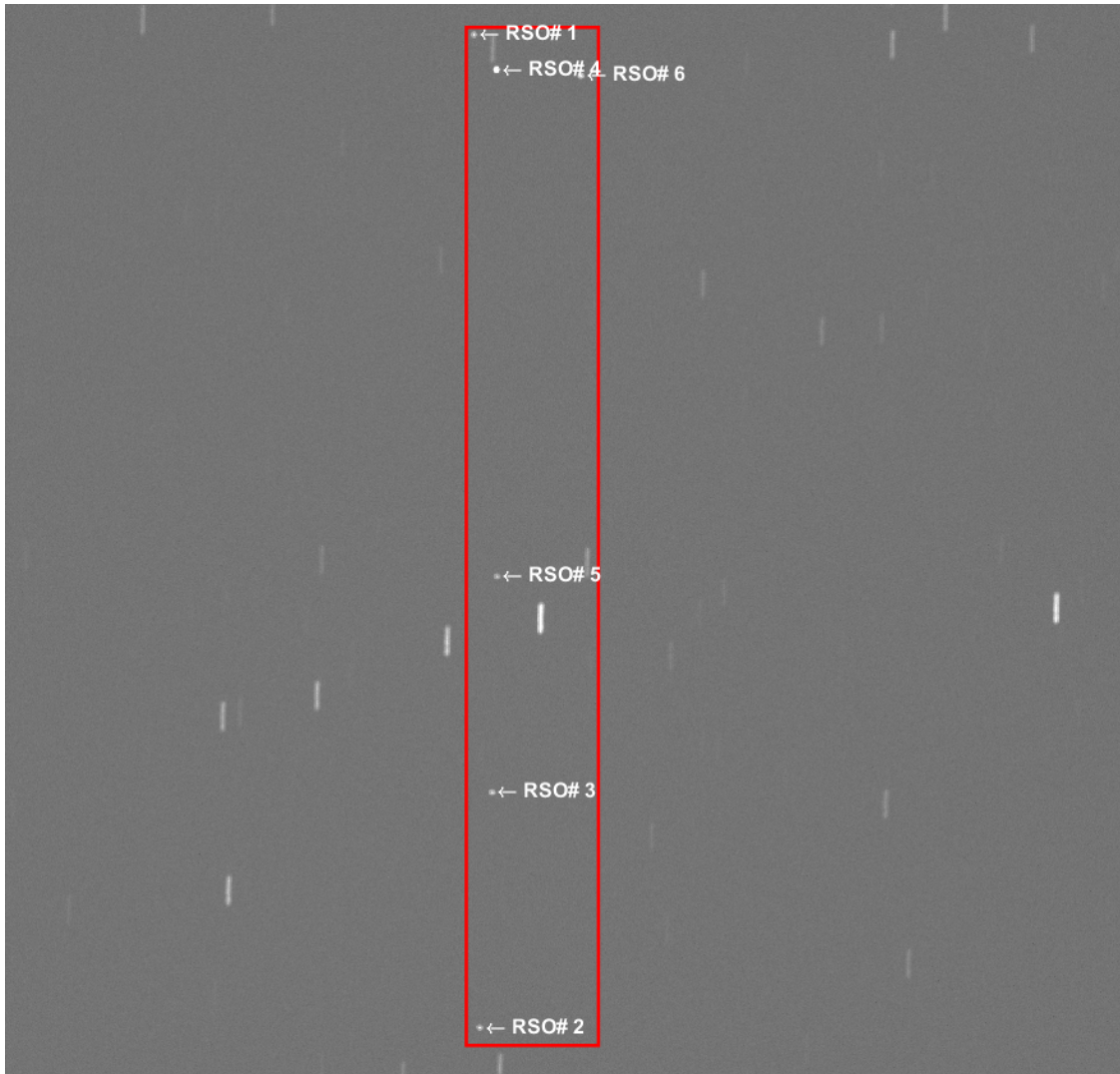


Figure 3.26 Successful template match example. Six RSOs are present in the image and were autonomously labeled by the template matching algorithm.

the image that failed to match before was then matched with the updated template. Note that by recursively attempting to resolve an image after the template was updated, a successful match was achieved for nearly all images. The exception was a handful of images where clouds blocked the field of view. At the points where template updates occur, there is a corresponding increase in the peak correlation value. The decrease in the peak correlation value as time increases results from motion of the satellites with respect to one another. Figure 3.29 shows the top three satellites in the template image at four different times to illustrate the change in position of satellites in the template image. In the span of two hours during data collection, there is a clear difference in the

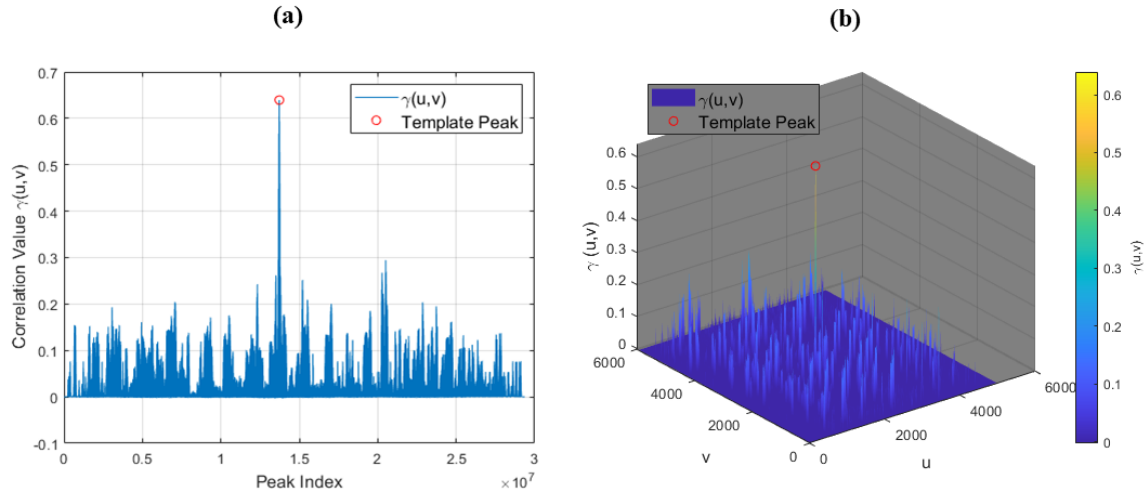


Figure 3.27 Template match correlation maps.

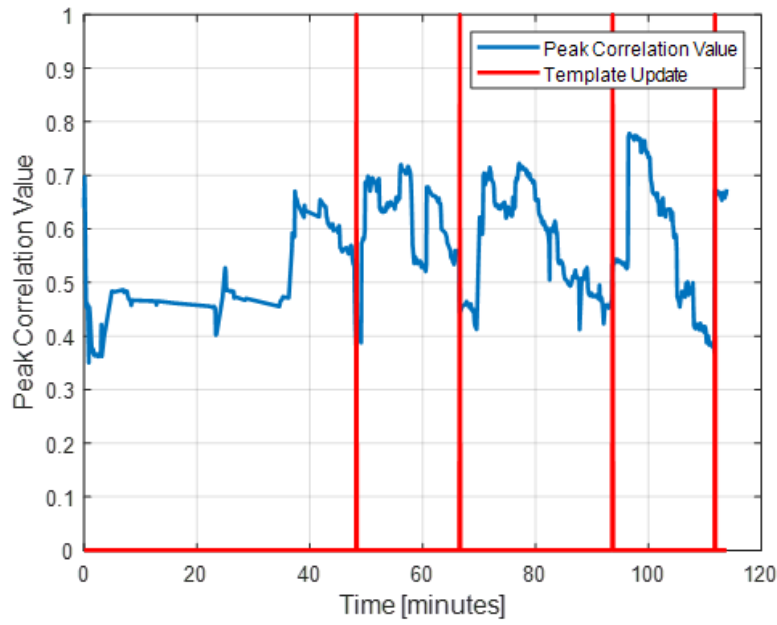


Figure 3.28 Peak correlation value with template updates.

RSO locations in this portion of the template, which showcases the need for periodic updating of the template image. Figure 3.29 also illustrates the successful autonomous update process for the tested image data set.

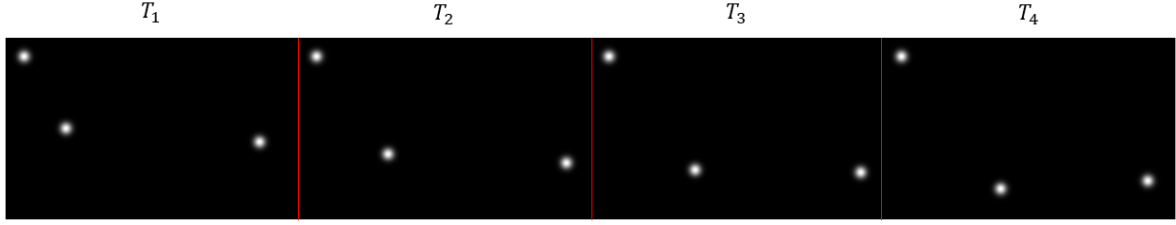


Figure 3.29 Template images through time.

3.5.2 UKF Orbit Estimation Results

In order to run the orbit determination algorithm, the camera frame (x, y) coordinates of all RSOs must be converted to inertial angular observations. After a successful template match the image x, y coordinates for the RSOs are converted to RA and DEC observation angles through plate-solving using a local installation of `Astrometry.net` [51, 52]. Converting all associated RSO points to RA, DEC angular observations gives the full set of measurements for orbit determination.

For the GEO objects observed over a two hour window, these observations form a set of quasi-linear trails or “tracklets”. Figure 3.30 shows the RA, DEC coordinates of the six RSOs detected across the image set. Each of the six RSOs was successfully associated across the image set and is shown in a unique color. The gaps in the RA, DEC trails are the result of only 745 out of 1095 of the total images captured being processed through the template matching algorithm due to poor image quality caused by cloud cover. However, the non-continuous data provide a more realistic observation scenario than simulated continuous measurements. The RA/DEC measurements shown and their respective time values are used for the orbit estimation step. Fig. 3.31 shows for RSO #1 the comparison of the actual measurements and UKF predicted measurements with their respective errors. Note that observation errors for DEC stay, on average, constant near 0.001° and for RA the error increases for some time and then levels out. The error in observation and predicted measurements could be caused by several factors. One possible factor is the use of only two-body propagation in the UKF, thus ignoring any perturbations affecting the actual satellite. Another possibility for the discrepancy in the predicted and actual measurements could be because the observation window is only 2 hours, corresponding to observing only $1/12$ of the actual orbit. Given spacing between observations, the filter prediction may converge more closely to the exact solution.

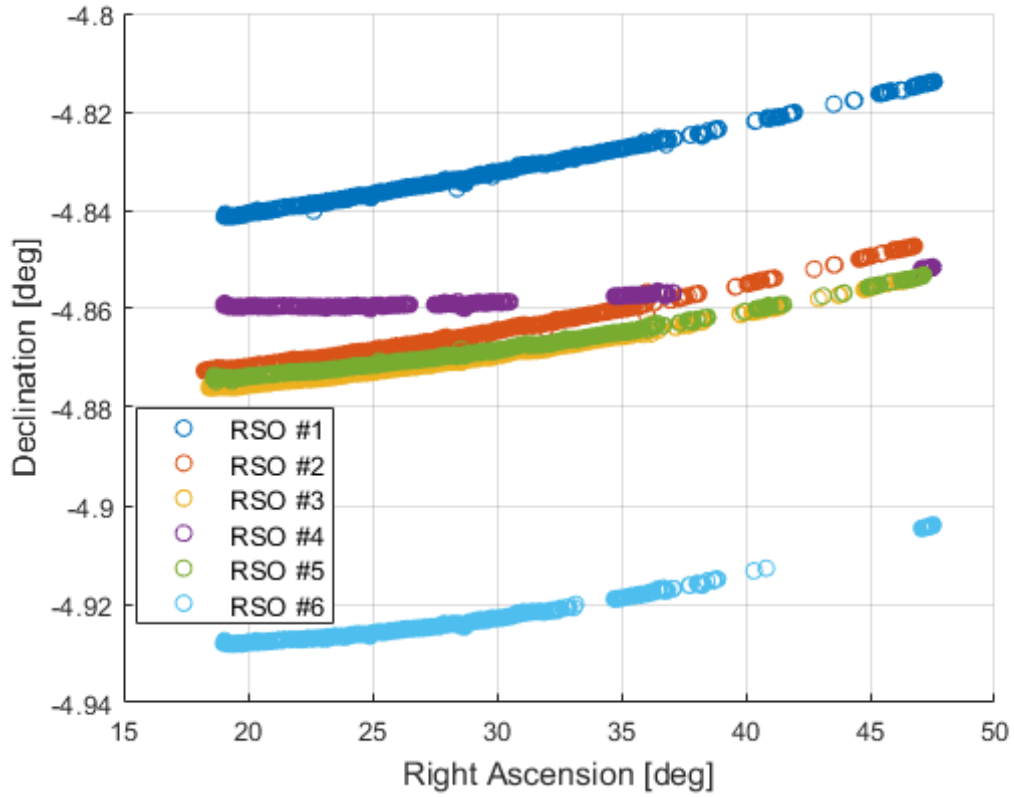


Figure 3.30 RSO angular measurements.

GEO observability is inherently low due to the fact that the orbits are nearly circular, equatorial orbits and present challenging observational geometry.

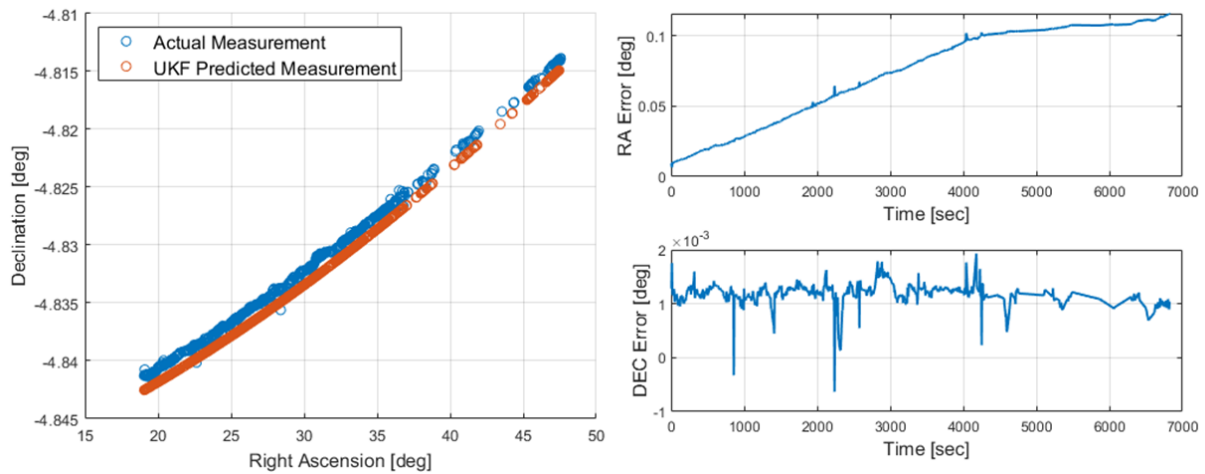


Figure 3.31 UKF measurement comparison RSO#1.

The UKF outlined in algorithm 5 was used to process the angular measurement data for each RSO detected in all images. In order to compare to a "truth" value the orbital elements at the time of the final measurement were calculated to compare with the corresponding orbital elements taken from TLE data. Note that all state vectors from TLEs must first be converted from the True Equator Mean Equinox (TEME) frame to the J2000 Earth Centered Inertial (ECI) frame before comparing orbital element values [68, 71]. Two Line Element data for each RSO detected were downloaded from space-track.org to provide the truth comparison [72]. Space-Track TLE data corresponding as close to the image epoch time as possible were used to minimize propagation errors for finding the "truth" comparison orbital elements.

A MATLAB implementation of the Special General Perturbations 4 (SGP4) software was used to propagate the TLEs to the image epoch time and then state vectors were converted from the TEME frame to ECI state vectors at the appropriate times [73]. When converting from the TEME frame to ECI, rotations accounting for the Earth's nutation and precession must be accounted for; see Vallado chapter 3 for more details [68]. The epoch time of the final measurement was 2021-11-03 04:12:46.805 UTC. Orbital elements were calculated for all RSOs using the UKF mean states at the final time and compared to the orbital elements resulting from TLE data. The resulting difference in orbital elements is shown in Table 3.1. The NORAD ID for each RSO is given in the first two columns of Table 3.1 for reference. Note that, all angular differences are given in degrees. For the

Table 3.1 UKF estimated orbital element error from TLE orbital element data.

RSO #	NORAD ID	a [km]	e	i [deg]	Ω [deg]
1	42934	0.00952727	-1.32E-05	2.15E-06	-0.05188064
2	39008	0.001100883	-3.72E-06	9.90E-06	-0.083774096
3	43562	0.126258476	2.84E-05	2.26E-05	-0.084483232
4	36792	0.063955954	7.73E-06	-2.95E-07	-0.041614312
5	41592	0.015283389	-2.98E-06	-3.29E-07	-0.095538326
6	35942	0.00992726	6.66E-06	-2.15E-05	0.018696663

first four elements (a, e, i, Ω), the difference in the orbital elements is very small as these describe the shape and orientation of the orbit. Because all the RSOs are GEO objects in very near equatorial circular orbits, the comparison of the argument of perigee and true anomaly is not particularly

helpful and thus these orbital elements are omitted from the comparison. The largest difference in semi-major axis is less than 0.13 km. The UKF estimated the size and shape of each orbit very well. Adding observations over a longer window should improve the estimation of the RAAN as well as the argument of perigee and true anomaly.

Average processing time for each image was 10 seconds. However, no attempt was made to optimize the MATLAB code to improve run times. Given an observational scenario where measurements are taken at this frequency, real time operation could be achieved.

3.5.3 Batch Orbit Estimation Results

This section shows the results of running the batch least squares orbit estimation routine described in Section 3.4.4. In order to provide the angular measurement data for the BLSQ algorithm, the same data resulting from running the autonomous template matching method in Sections 3.4.4 and 3.5.2 is used. The measurement data used for batch estimation are shown in Fig. 3.30. The orbit estimation was performed on RSO #1 which is identified as NORAD ID 42934. The estimation results are summarized in Table 3.2. Both the position and velocity as well as the orbital element errors are shown. The position was estimated to within 6.67 km in range and 3.62×10^{-4} km/s in speed. Overall, the batch LSQ method resulted in similar error levels to the UKF orbit estimation.

The resulting position with 3σ error bounds is shown in Fig. 3.32. The nominal state calculated by the BLSQ algorithm is shown as a blue circle, and the true position is shown by a red diamond. Note that the “true” position is taken to be the position predicted at the measurement time by propagation of the satellites TLE to the measurement time. The estimated velocity with 3σ error bounds is shown in Fig. 3.33. Note that, the nominal state value and the true state values lie very close to one another.

Table 3.2 Batch least squares error results. Estimated initial state errors are given in position and velocity form as well as orbital element form.

State Errors		
x_0 Error	-4.8286	km
y_0 Error	4.4945	km
z_0 Error	1.0307	km
\dot{x}_0 Error	-3.2103E-04	km/s
\dot{y}_0 Error	-1.5802E-04	km/s
\dot{z}_0 Error	-5.4835E-05	km/s
Orbital Element Errors		
Semi-major axis (a) error	8.5206	km
Eccentricity (e) error	-9.1423E-05	-
Inclination (i) error	2.9688E-05	deg
RAAN (Ω) error	-3.9628E-03	deg
Argument of Perigee (ω) error	2.5618E-01	deg
True Anomaly (ν) error	-2.5235E-01	deg

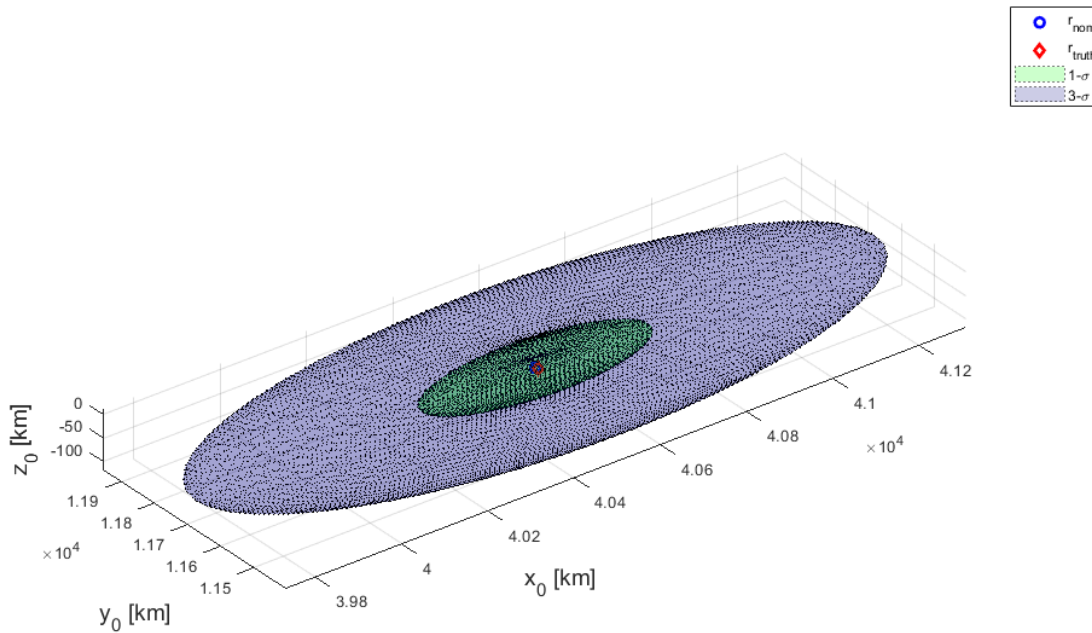


Figure 3.32 Batch orbit results for satellite position. Three- σ bounds on position are shown as transparent ellipses. Note that the true and estimated position are very close.

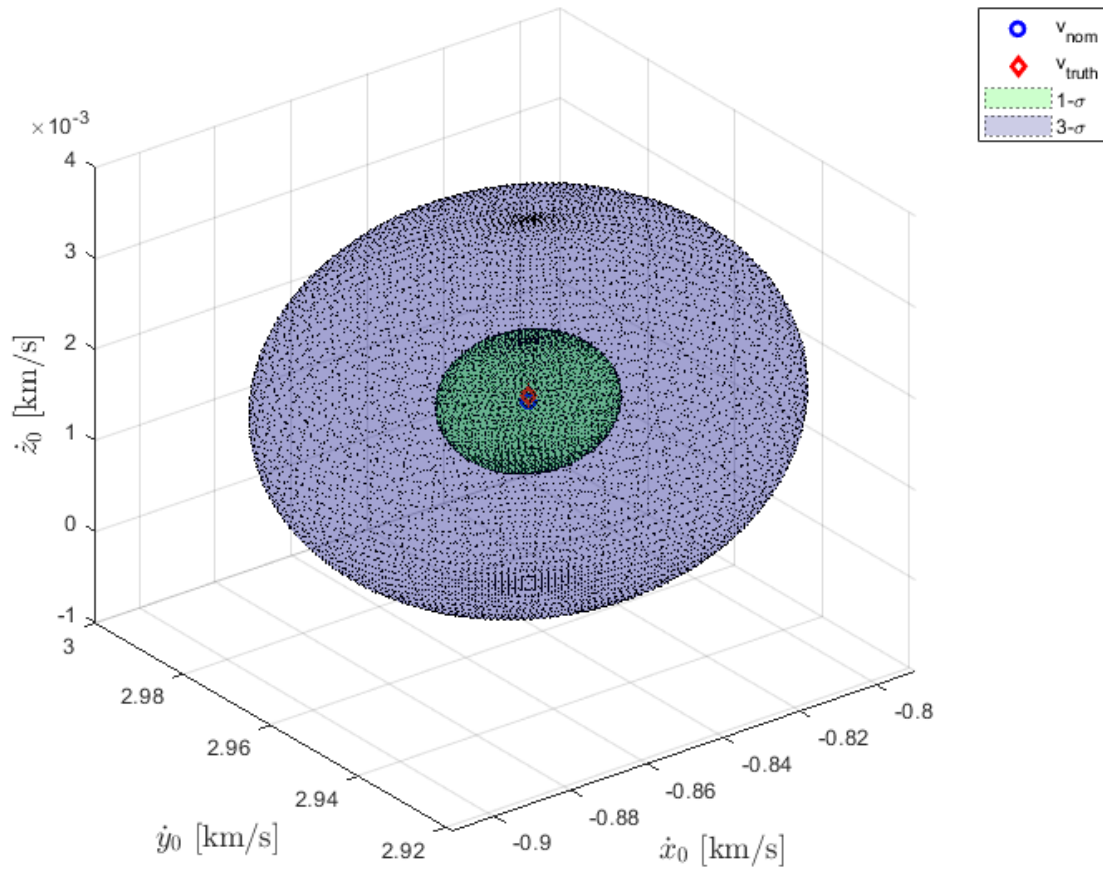


Figure 3.33 Batch orbit results for satellite velocity. Three- σ bounds on velocity are shown as transparent ellipses. Note that the true and estimated velocity are quite close.

4 Three-Body Relative Motion Models

The second main focus of this research lies in extending understanding of relative motion in the three-body problem. Linear relative motion solutions are well-known in the two-body problem in the form of the HCW equations. A similar framework is sought for the three-body problem through the development of a general purpose linear motion model for the ER3BP and CR3BP. Linear motion solutions for the three-body problem have been well characterized near the Lagrange equilibrium points, but comparatively little work has been done for a general linearized solution [27, 37]. Franzini et al. developed a non-linear relative motion model for the ER3BP and linearized the gravitational terms leading to a somewhat more simplified equation set [74]. The frame used by Franzini was a local vertical local horizontal (LVLH) frame that required the computation of the deputy spacecraft jerk and angular acceleration terms. This research presents an alternative linearized relative motion framework for the ER3BP and the CR3BP that is independent of the spacecraft orientation.

This chapter is structured as follows. First, the general three-body problem dynamics are summarized. Secondly, a chief and deputy spacecraft scenario set in cislunar space is used to develop the full non-linear model for spacecraft relative motion in the ER3BP and CR3BP in a rotating barycentric frame. The non-linear model is then linearized about the chief spacecraft trajectory by a first-order Taylor series method. A semi-analytical solution for relative motion is then developed using the exponential matrix solution and direct propagation of the linearized equations of motion. Propagated spacecraft trajectories for the non-linear model, linear model, and matrix exponential solution are compared via MATLAB simulation. The linearized relative motion models assume that the primary bodies (Earth-Moon) move under Keplerian elliptical motion for the ER3BP and in a circular orbit for the CR3BP. The position state of the chief spacecraft is assumed to be known at all times.

The contribution of this chapter is the development of the linearized relative equations of motion for a deputy spacecraft in the rotating barycentric synodic frame and the development of a semi-analytical solution to the linearized equations of motion through a matrix exponential method.

4.1 Three-Body Dynamics

This section describes the dynamics of the general three-body problem. The geometry of the general form of the three-body problem is shown in Fig. 4.1. An arbitrary inertial frame is denoted as the N -frame which may, without loss of generality, lie at the center of mass (barycenter) for the system. Three massive bodies are assumed to be moving under the respective gravitational fields of the other two bodies. The general three-body problem (3BP) provides a framework for understanding the chaotic nature of orbits in this domain. The term “chaotic” is used in the sense that a small shift in initial conditions leads to radically different motion of the bodies. In order to perform meaningful analysis, several simplifying assumptions can be made for a spacecraft in cislunar space. First, the mass of the spacecraft can be considered negligible compared to the mass of the Earth and the Moon. Thus the gravitational effect of the satellite on the Earth and Moon’s motion can be ignored. This simplification is the restricted three-body problem (R3BP).

The most general form of the R3BP is the elliptical restricted three-body problem (ER3BP). In the ER3BP the motion of the two massive bodies m_1 and m_2 is governed by Keplerian two-body dynamics. For the Earth-Moon system, this means that the Moon orbits the Earth in an elliptical orbit. As a consequence of the changing distance between the Earth and the Moon, the Lagrange equilibrium points are non-stationary.

4.1.1 Three-Body Problem Geometry

Choosing an appropriate coordinate system often simplifies the expression of the dynamics of complex systems. In the case of a satellite in cislunar space, a rotating coordinate system fixed to either the barycenter or the Moon are the most commonly used systems. Choice of the appropriate origin helps to simplify the expression of the equations of motion of a satellite. Fig. 4.2 shows the geometry of a typical cislunar scenario. A satellite orbits in cislunar space, and its position can be measured relative to the barycenter by the vector \mathbf{R}_i .

Two-coordinate systems are of particular interest in this figure, the Moon-centered synodic system denoted: \mathcal{M} , and the barycentric synodic system denoted: \mathcal{B} . Both systems are rotating coordinate systems with the axis of rotation assumed to be along the Earth-Moon angular momentum

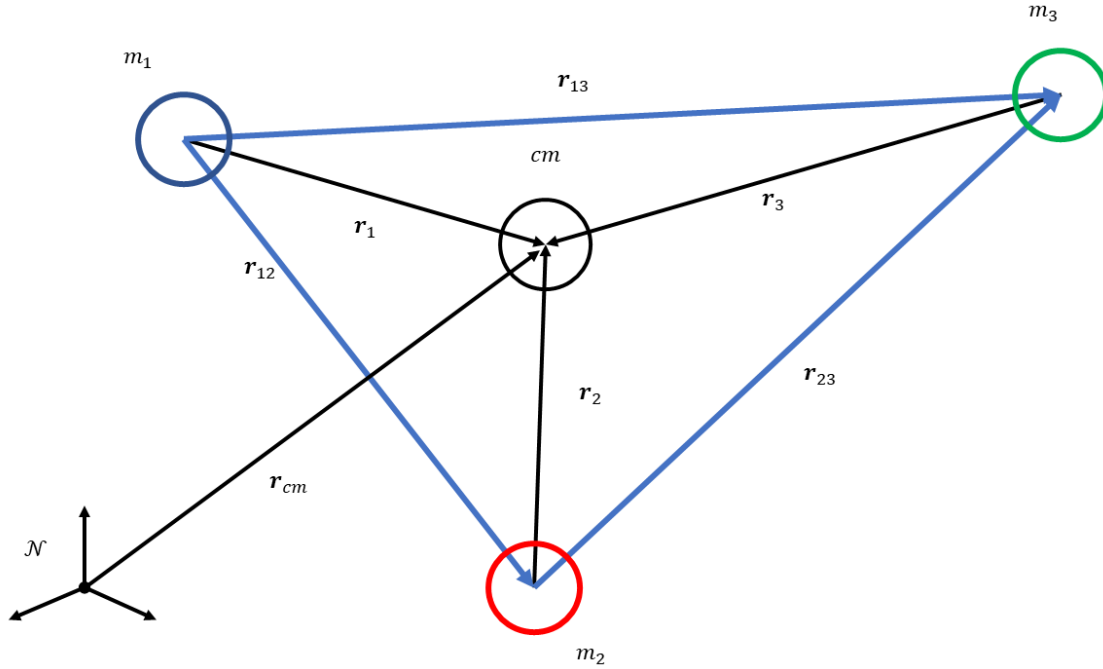


Figure 4.1 General three-body problem geometry.

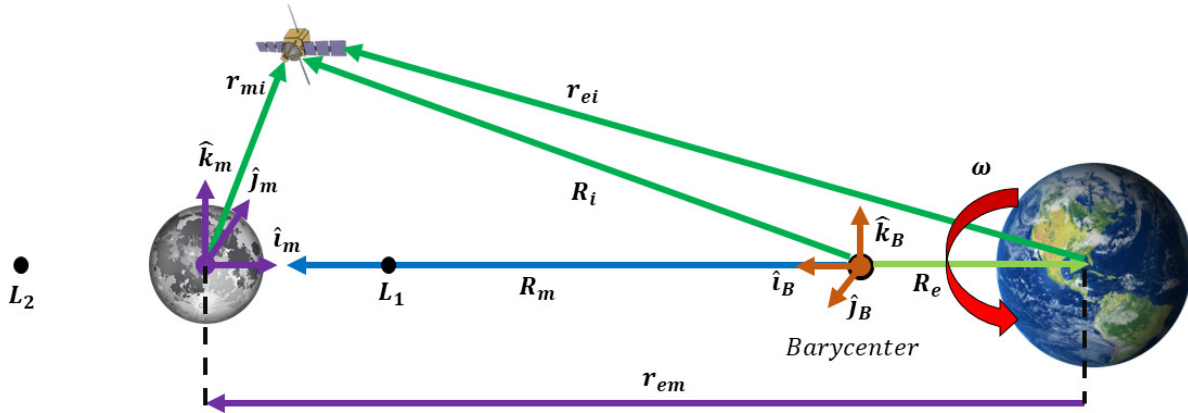


Figure 4.2 Geometry for the three-body problem. Coordinate systems shown include the barycentric and Moon-centered synodic rotating systems.

direction. The \mathcal{B} -frame is located at the Earth-Moon barycenter, with the x -axis oriented along the vector from the Earth to the Moon. The z -axis is in the Earth-Moon angular momentum direction, and the y -axis completes the right-handed set. The unit vectors are defined formally with respect to

an inertial frame residing at the barycenter as shown in Eq. (4.1).

$$\begin{aligned}\hat{\mathbf{i}}_{\mathcal{B}} &= \frac{\mathbf{r}_{em}}{r_{em}} \\ \hat{\mathbf{j}}_{\mathcal{B}} &= \hat{\mathbf{k}}_{\mathcal{B}} \times \hat{\mathbf{i}}_{\mathcal{B}} \\ \hat{\mathbf{k}}_{\mathcal{B}} &= \frac{\mathbf{r}_{em} \times \dot{\mathbf{r}}_{em}}{\|\mathbf{r}_{em} \times \dot{\mathbf{r}}_{em}\|}\end{aligned}\tag{4.1}$$

Without loss of generality the inertial frame can be assumed to be co-located at the barycenter. Then the \mathcal{B} -frame is fully defined with respect to the inertial frame as below.

$$\mathcal{B} : \{\mathbf{0}, \hat{\mathbf{i}}_{\mathcal{B}}, \hat{\mathbf{j}}_{\mathcal{B}}, \hat{\mathbf{k}}_{\mathcal{B}}\}\tag{4.2}$$

The inertial frame will be denoted as the \mathcal{I} -frame and will have axes aligned with the \mathcal{B} -frame when the Moon is at periapsis. Note that, in the ER3BP, the rotation of the \mathcal{B} -frame is not constant and is related to the Moon's true anomaly rate. And further note that a conversion from the \mathcal{B} -frame to the \mathcal{I} -frame is simply a rotation about the $\hat{\mathbf{k}}_{\mathcal{B}}$ axis and will be the subject of section 4.1.2.

Defining a few of the parameters of the Moon's orbit will be useful for developing the ER3BP EOMs. The distance between the Earth and the Moon, denoted as R_{em} can be found through the orbit equation using two-body dynamics. Also needed is the Moon's angular momentum h .

$$R_{em} = \frac{p}{1 + e \cos(\nu)}\tag{4.3}$$

$$p = \frac{h^2}{\mu_e}\tag{4.4}$$

$$h = \|\mathbf{R}_{em} \times \dot{\mathbf{R}}_{em}\|\tag{4.5}$$

The Moon's semi-parameter is given by p , the eccentricity of the Moon's orbit is e , the Moon's true anomaly is ν , and the Earth's gravitational parameter is given by μ_e .

The angular velocity and angular acceleration of the Moon around the Earth can also be written using purely two-body quantities. Note that ω and $\dot{\omega}$ will be required to relate the rotating frame

and the inertial frame.

$$\omega = \frac{\sqrt{\mu}}{(a(1-e^2))^{3/2}} (1 + e \cos(\nu))^2 \quad (4.6)$$

$$\dot{\omega} = \frac{\sqrt{\mu}}{(a(1-e^2))^3} (1 + e \cos(\nu))^2 (-2e \sin(\nu) - 2e^2 \sin(\nu) \cos(\nu)) \quad (4.7)$$

The magnitude of the angular velocity (ω) is found through the true-anomaly rate of the Moon as in Eq. (4.6) [68], where a is the Earth-Moon semi-major axis, e is eccentricity, and E is the current eccentric anomaly of the Moon. Additionally, the velocity of the Earth-Moon distance can be found. Note that, in the rotating frames, the Earth-Moon vector, and velocity act only along the x -axis. And $\dot{\mathbf{R}}_m = \dot{R}_m \hat{\mathbf{i}}_{\mathcal{B}}$, where the radial rate can be found purely from the Moon's orbital element information. Once again the development of these two-body expressions can be found in ref [68].

$$\dot{R}_m = \frac{R_m \dot{\nu} e \sin(\nu)}{1 + e \cos(\nu)} \quad (4.8)$$

$$\dot{\nu} = \frac{na^2}{R_m^2} \sqrt{1 - e^2} \quad (4.9)$$

4.1.2 Coordinate Transformation

Sometimes it will be necessary to perform coordinate transformations between the inertial and rotating frames. To transform from one frame to the other will only involve a rotational component since both frames are co-located at the Earth-Moon barycenter. In many cases the initial conditions for spacecraft in periodic trajectories are given in the \mathcal{B} -frame; therefore it will be useful to have a conversion to the \mathcal{I} -frame for comparison purposes.

Consider the general position vector $\mathbf{r}_{\mathcal{B}} = (x\hat{\mathbf{i}}_{\mathcal{B}} + y\hat{\mathbf{j}}_{\mathcal{B}} + z\hat{\mathbf{k}}_{\mathcal{B}})$ written in the barycentric rotating frame. The goal is to transform this vector to the barycentric inertial frame. In general, any vector can be rotated to a new frame by pre-multiplication of a rotation matrix corresponding to the appropriate axes [75]. The transformation of a vector from the \mathcal{B} -frame to the \mathcal{I} -frame is shown in Eq. (4.10).

$$\mathbf{r}_I = C_{I\mathcal{B}}\mathbf{r}_{\mathcal{B}} \quad (4.10)$$

Note that, the transformation matrix $C_{I\mathcal{B}}$ is time-varying in nature due to the non-constant angular velocity of the Moon on its orbit. The rotation matrix is given by a simple rotation about the $\hat{\mathbf{k}}_{\mathcal{B}}$ axis:

$$C_{I\mathcal{B}} = \begin{bmatrix} \cos(v_{moon}) & -\sin(v_{moon}) & 0 \\ \sin(v_{moon}) & \cos(v_{moon}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.11)$$

where v_{moon} is the true anomaly of the Moon along its orbit, which is governed by two-body dynamics (see ref. [68] for more details).

4.1.3 ER3BP Equations of Motion

In the general three-body problem (3BP), all bodies are assumed to be modelled as point masses, and the geometry of the problem is shown in Fig. 4.2, where a spacecraft is assumed to be orbiting in a regime affected by the gravitational force of both the Earth and the Moon. Considering the mass of a spacecraft ($m_3 \ll m_1, m_2$) to be negligible, the equation of motion of the spacecraft under the gravitational influence of two primary bodies (m_1 and m_2) can be defined by a summation of the gravitational forces acting on m_3 . The assumptions of the ER3BP are summarized as:

1. The mass of the satellite is negligible: $m_i \ll m_E, m_m$.
2. The Moon orbits the Earth in an elliptical orbit: $e_{moon} = 0.05490$ [76].
3. The distance between the primary (Earth and Moon) bodies is non-constant and can be found using Eq. (4.3).
4. Rotation of the Moon about the Earth is non-constant with angular rate given by Eq. (4.6).

Referring to the geometry in Fig. 4.2, and assuming without loss of generality that the inertial frame coincides with the origin of the \mathcal{B} -frame, the EOM of the satellite and each primary mass can be

written by summing all forces acting on each body and using Newton's universal law of gravitation.

$$m_i \ddot{\mathbf{R}}_i = -GM_e m_i \frac{\mathbf{r}_{ei}}{r_{ei}^3} - GM_m m_i \frac{\mathbf{r}_{mi}}{r_{mi}^3} \quad (4.12)$$

$$M_e \ddot{\mathbf{R}}_e = GM_m M_e \frac{\mathbf{r}_{em}}{r_{em}^3} + GM_e m_i \frac{\mathbf{r}_{ei}}{r_{ei}^3} \quad (4.13)$$

$$M_m \ddot{\mathbf{R}}_m = -GM_e M_m \frac{\mathbf{r}_{em}}{r_{em}^3} + GM_m m_i \frac{\mathbf{r}_{mi}}{r_{mi}^3} \quad (4.14)$$

The mass of the Earth, Moon, and satellite respectively are given by M_e, M_m, m_i , and G is the universal gravitational constant. Note that the relative vector $\mathbf{r}_{mi} = \mathbf{R}_i - \mathbf{R}_m$ gives the position of the satellite with respect to the Moon and $\mathbf{r}_{ei} = \mathbf{R}_i - \mathbf{R}_e$ gives the position of the satellite with respect to the Earth. The accelerations written in Eqs. (4.12) - (4.14) are written with respect to the inertial \mathcal{I} -frame. Note that, as written, these equations have no known analytical solution to model the motion of the satellite mass m_i when it is affected by the gravitational forces of M_e and M_m .

In order to make the problem more tractable, the restricted three-body problem assumption must be applied. The mass of the satellite m_i is assumed to be negligible when compared to the other two massive bodies (in this case the Earth and the Moon). Thus, for the restricted three-body problem (R3BP) $m_i \ll M_e$ and $m_i \ll M_m$ [27]. Introducing the gravitational parameters for the Earth and the Moon simplifies notation a bit further with: $\mu_e = GM_e$ and $\mu_m = GM_m$. Assuming that the Moon orbits the Earth in a Keplerian ellipse completes the definition of the elliptical restricted three-body problem assumptions. The desired equation of motion is that of the satellite with respect to the barycenter.

$$\ddot{\mathbf{R}}_i = -\mu_e \frac{\mathbf{r}_{ei}}{r_{ei}^3} - \mu_m \frac{\mathbf{r}_{mi}}{r_{mi}^3} \quad (4.15)$$

The relative vectors between the Earth and the satellite \mathbf{r}_{ei} and the Moon and the satellite \mathbf{r}_{mi} can be re-written using $\mathbf{R}_i, \mathbf{R}_e, \mathbf{R}_m$ as follows.

$$\mathbf{r}_{ei} = \mathbf{R}_i - \mathbf{R}_e \quad (4.16)$$

$$\mathbf{r}_{mi} = \mathbf{R}_i - \mathbf{R}_m \quad (4.17)$$

$$\mathbf{R}_e = -\mu R_{em} \hat{\mathbf{i}}_{\mathcal{B}} \quad (4.18)$$

$$\mathbf{R}_m = (1 - \mu) R_{em} \hat{\mathbf{i}}_{\mathcal{B}} \quad (4.19)$$

$$\mu = \frac{\mu_m}{\mu_m + \mu_e} \quad (4.20)$$

The combined gravitational parameter for the Earth-Moon system is given by μ . Substituting for \mathbf{r}_{ei} and \mathbf{r}_{mi} in Eq. (4.15), a more useful version of the EOM can be found.

$$\ddot{\mathbf{R}}_i = -\mu_e \frac{\mathbf{R}_i - \mathbf{R}_e}{\|\mathbf{R}_i - \mathbf{R}_e\|^3} - \mu_m \frac{\mathbf{R}_i - \mathbf{R}_m}{\|\mathbf{R}_i - \mathbf{R}_m\|^3} \quad (4.21)$$

Equation (4.21) gives the inertial acceleration of the satellite written in the \mathcal{I} -frame. While numerical propagation of Eq. (4.21) is possible, the equations of motion are easily cast into the rotating \mathcal{B} -frame by use of the transport theorem [75].

The derivation of the \mathcal{B} -frame EOMs starts with writing the vector from the barycenter to the satellite \mathbf{R}_i in the rotating \mathcal{B} -frame. Let the rotation between the inertial \mathcal{I} -frame and the rotating \mathcal{B} -frame be defined by $\omega_{\mathcal{B}/\mathcal{I}} = \omega \hat{\mathbf{k}}_{\mathcal{B}}$. Taking two derivatives with respect to the inertial frame, and applying the transport theorem, we arrive at Eq. (4.24).

$$[\mathbf{R}_i]_{\mathcal{B}} = x\hat{\mathbf{i}}_{\mathcal{B}} + y\hat{\mathbf{j}}_{\mathcal{B}} + \hat{\mathbf{k}}_{\mathcal{B}} \quad (4.22)$$

$$\left[\dot{\mathbf{R}}_i\right]_{\mathcal{I}} = \left[\dot{\mathbf{R}}_i\right]_{\mathcal{B}} + \omega_{\mathcal{B}/\mathcal{I}} \times [\mathbf{R}_i]_{\mathcal{B}} \quad (4.23)$$

$$\begin{aligned} \left[\ddot{\mathbf{R}}_i\right]_{\mathcal{I}} &= \left[\ddot{\mathbf{R}}_i\right]_{\mathcal{B}} + 2\omega_{\mathcal{B}/\mathcal{I}} \times \left[\dot{\mathbf{R}}_i\right]_{\mathcal{B}} + \dot{\omega}_{\mathcal{B}/\mathcal{I}} \times [\mathbf{R}_i]_{\mathcal{B}} \\ &\quad + \omega_{\mathcal{B}/\mathcal{I}} \times (\omega_{\mathcal{B}/\mathcal{I}} \times [\mathbf{R}_i]_{\mathcal{B}}) \end{aligned} \quad (4.24)$$

The subscript on the bracket notation for a vector denotes the frame a vector is written in; for example, $[*]_{\mathcal{B}}$ denotes a vector written in the rotating \mathcal{B} -frame. The inertial acceleration of \mathbf{R}_i was previously found in Eq. (4.21). Thus substituting this value in for $\left[\ddot{\mathbf{R}}_i\right]_{\mathcal{I}}$, the barycentric velocity

term $[\ddot{\mathbf{R}}_i]_{\mathcal{B}}$ is easily solved for and is shown in Eq. (4.25).

$$[\ddot{\mathbf{R}}_i]_{\mathcal{B}} = -2\omega_{\mathcal{B}/I} \times [\dot{\mathbf{R}}_i]_{\mathcal{B}} - \dot{\omega}_{\mathcal{B}/I} \times [\mathbf{R}_i]_{\mathcal{B}} - \omega_{\mathcal{B}/I} \times (\omega_{\mathcal{B}/I} \times [\mathbf{R}_i]_{\mathcal{B}}) - \mu_e \frac{[\mathbf{R}_i]_{\mathcal{B}} - \mathbf{R}_e}{\|[\mathbf{R}_i]_{\mathcal{B}} - \mathbf{R}_e\|^3} - \mu_m \frac{[\mathbf{R}_i]_{\mathcal{B}} - \mathbf{R}_m}{\|[\mathbf{R}_i]_{\mathcal{B}} - \mathbf{R}_m\|^3} \quad (4.25)$$

Since all terms are written in the \mathcal{B} -frame at this point, the bracket notation will be dropped.

$$\ddot{\mathbf{R}}_i = -2\omega_{\mathcal{B}/I} \times \dot{\mathbf{R}}_i - \dot{\omega}_{\mathcal{B}/I} \times \mathbf{R}_i - \omega_{\mathcal{B}/I} \times (\omega_{\mathcal{B}/I} \times \mathbf{R}_i) - \mu_e \frac{\mathbf{R}_i - \mathbf{R}_e}{\|\mathbf{R}_i - \mathbf{R}_e\|^3} - \mu_m \frac{\mathbf{R}_i - \mathbf{R}_m}{\|\mathbf{R}_i - \mathbf{R}_m\|^3} \quad (4.26)$$

Working out all cross product terms and noting that $\mathbf{r}_{ei} = \mathbf{R}_i - \mathbf{R}_e$ and $\mathbf{r}_{mi} = \mathbf{R}_i - \mathbf{R}_m$, we arrive at the scalar ER3BP equations of motion in the \mathcal{B} -frame. Note that it will be helpful to have the scalar form of the EOMs for development of the linearized relative equations.

$$\ddot{x}_{\mathcal{B}} = 2\omega\dot{y}_{\mathcal{B}} + \dot{\omega}y_{\mathcal{B}} + \omega^2x_{\mathcal{B}} - \mu_e \frac{(x_{\mathcal{B}} + R_e)}{r_{ei}^3} - \mu_m \frac{(x_{\mathcal{B}} - R_m)}{r_{mi}^3} \quad (4.27)$$

$$\ddot{y}_{\mathcal{B}} = -2\omega\dot{x}_{\mathcal{B}} - \dot{\omega}x_{\mathcal{B}} + \omega^2y_{\mathcal{B}} - \mu_e \frac{y_{\mathcal{B}}}{r_{ei}^3} - \mu_m \frac{y_{\mathcal{B}}}{r_{mi}^3} \quad (4.28)$$

$$\ddot{z}_{\mathcal{B}} = -\mu_e \frac{z_{\mathcal{B}}}{r_{ei}^3} - \mu_m \frac{z_{\mathcal{B}}}{r_{mi}^3} \quad (4.29)$$

The magnitudes of the position vectors from the Earth and Moon to the satellite (r_{ei} and r_{mi}) are found simply as the norm of the appropriate vector quantities.

$$r_{ei} = \sqrt{(x + R_e)^2 + y^2 + z^2} \quad (4.30)$$

$$r_{mi} = \sqrt{(x - R_m)^2 + y^2 + z^2} \quad (4.31)$$

$$R_e = \mu R_{em} \quad (4.32)$$

$$R_m = (1 - \mu)R_{em} \quad (4.33)$$

Note that R_{em} is the Earth-Moon distance and can be found using Eq. (4.3). Note that these equations are all shown in their dimensional form, but if desired could be easily cast into non-dimensional form by simple substitutions [74].

4.1.4 CR3BP Equations of Motion

The dynamics of the CR3BP are very similar to that of the ER3BP. The key difference between the equation sets arises from the assumption in the CR3BP that the Moon's orbit around the Earth is circular rather than elliptical. As a consequence of the Moon's orbit being assumed to be circular, the Earth Moon distance is constant (i.e. $R_{em} = a_{moon}$, where a_{moon} is the semi major axis of the Moon), and the angular rate of the \mathcal{B} -frame is constant and equal to the Moon's mean motion ($\omega = n_{moon}$, and $\dot{\omega} = 0$). The CR3BP assumptions can be summarized as follows.

1. The mass of the satellite is negligible: $m_i \ll m_E, m_m$.
2. The Moon orbits the Earth in a circular orbit: $e_{moon} = 0$.
3. Earth-Moon distance is constant: $R_{em} = a_{moon}$.
4. Rotation of the Moon about the Earth is constant: $\omega = n_{moon}$, $\dot{\omega} = 0$.

Making the substitutions for the Earth-Moon distance and angular rate, the CR3BP equations of motion can be directly written from Eq. (4.26).

$$\begin{aligned} \ddot{\mathbf{R}}_i = & -2\omega_{\mathcal{B}/I} \times \dot{\mathbf{R}}_i - \omega_{\mathcal{B}/I} \times (\omega_{\mathcal{B}/I} \times \mathbf{R}_i) \\ & - \mu_e \frac{\mathbf{R}_i - \mathbf{R}_e}{\|\mathbf{R}_i - \mathbf{R}_e\|^3} - \mu_m \frac{\mathbf{R}_i - \mathbf{R}_m}{\|\mathbf{R}_i - \mathbf{R}_m\|^3} \end{aligned} \quad (4.34)$$

And similarly the scalar EOMs can be written as shown in Eqs. (4.35) - (4.37).

$$\ddot{x}_{\mathcal{B}} = 2\omega\dot{y}_{\mathcal{B}} + \omega^2 x_{\mathcal{B}} - \mu_e \frac{(x_{\mathcal{B}} + R_e)}{r_{ei}^3} - \mu_m \frac{(x_{\mathcal{B}} - R_m)}{r_{mi}^3} \quad (4.35)$$

$$\ddot{y}_{\mathcal{B}} = -2\omega\dot{x}_{\mathcal{B}} + \omega^2 y_{\mathcal{B}} - \mu_e \frac{y_{\mathcal{B}}}{r_{ei}^3} - \mu_m \frac{y_{\mathcal{B}}}{r_{mi}^3} \quad (4.36)$$

$$\ddot{z}_{\mathcal{B}} = -\mu_e \frac{z_{\mathcal{B}}}{r_{ei}^3} - \mu_m \frac{z_{\mathcal{B}}}{r_{mi}^3} \quad (4.37)$$

The magnitudes of the vectors from the Earth and Moon to the satellite (r_{ei} and r_{mi}) are easily found by taking the norms of the appropriate vector quantities. The key difference between the ER3BP and the CR3BP here lies in the constant distance between the Earth and the Moon.

$$r_{ei} = \sqrt{(x + R_e)^2 + y^2 + z^2} \quad (4.38)$$

$$r_{mi} = \sqrt{(x - R_m)^2 + y^2 + z^2} \quad (4.39)$$

$$R_e = \mu a_{moon} \quad (4.40)$$

$$R_m = (1 - \mu)a_{moon} \quad (4.41)$$

4.1.5 Three-Body Equilibrium Positions (Lagrange Points)

Before developing the relative equations of motion in the ER3BP it is important to understand the equilibrium positions of the three-body problem. In the circular restricted three-body problem, there exist five static equilibrium points known as the Lagrange or libration points (so named after famous mathematician Joseph Louis Lagrange). There are three equilibrium points that are co-linear with the Earth and Moon, and two that exist at equilateral triangular points in the Moon's orbital plane. In the circular three-body problem, the equilibrium points are stationary and fixed in the rotating synodic coordinate system [28, 36].

When the more general elliptical three-body problem is considered, the equilibrium points are no longer stationary in the synodic frame, but pulsate along the $\hat{\mathbf{i}}_m$ -axis near the static positions corresponding to the CR3BP. Of particular interest are the two co-linear equilibrium points L1 and L2 that lie between the Earth and the Moon, and on the far side of the Moon respectively. Figure 4.3 shows the geometry of the co-linear Lagrange points. For the CR3BP, the Earth-Moon radius R is constant and equal to the semi-major axis of the Moon's orbit. For the ER3BP, the radius of the Moon's orbit changes according to the two-body relationship: $R = a(1 - e \cos(E))$ [68]. Note that, a is the semi-major axis, e is the eccentricity, and E is the eccentric anomaly of the Moon's orbit. Scenarios in the ER3BP then must consider the motion of the Moon around the Earth with time in order to specify initial conditions. In order to find the position of the co-linear

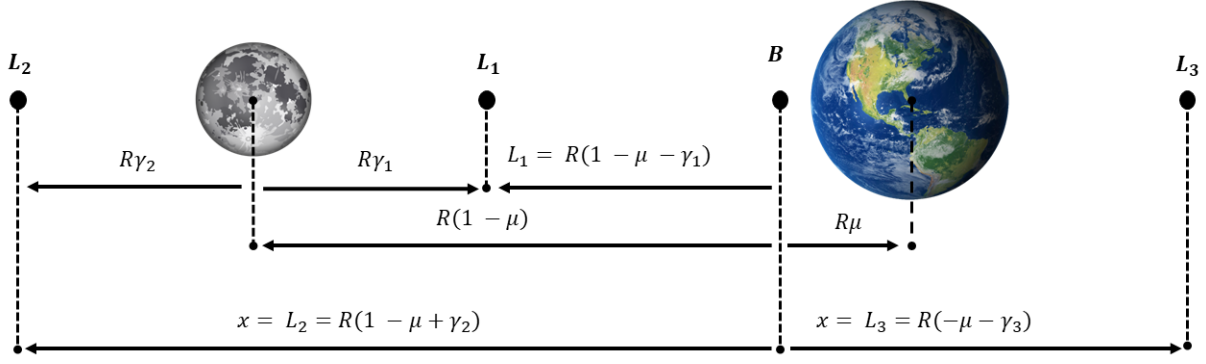


Figure 4.3 Earth-Moon Lagrange point positions and distances.

Lagrange points, a quintic polynomial in the variable γ_i , $i = 1, 2, 3$ must be solved where i refers to the corresponding Lagrange point [27, 28]. The polynomial equations result from considering the equilibrium condition for a particle constrained to move along the x -axis in the barycentric rotating frame with: $y, z, \dot{y}, \dot{z}, \ddot{y}, \ddot{z} = 0$. Considering a spacecraft orbiting the Moon in the barycentric rotating frame, the equations of motion can be solved for a relationship between the distance from the barycenter x and the Earth-Moon distance $R_{em} = R$. The ratio of $\frac{x}{R}$ is constant through an orbit and for the three co-linear points has the following functional, and derivative form [28].

$$f(x) = \frac{x}{R^3} - \frac{1-\mu}{(x+\mu R)^2} + \frac{\mu}{(x-(1-\mu)R)^2} = 0 \quad (4.42)$$

$$f'(x) = \frac{1}{R^3} + \frac{2(1-\mu)}{(x+\mu R)^3} - \frac{2\mu}{(x-(1-\mu)R)^3} = 0 \quad (4.43)$$

To find the location of any of the co-linear Lagrange points, simply substitute the appropriate distance from the barycenter corresponding to the chosen Lagrange point into Eq. (4.42). The result will be a fifth order polynomial in γ_i . The solution for the Lagrange point is then found by numerically solving Eq. (4.42) via a Newton Raphson method. The appropriate distances for each Lagrange point are listed below, and are also found from the geometry shown in Fig. 4.3. The

combined Earth-Moon gravitational parameter μ is repeated here in Eq. (4.44) for convenience.

$$\mu = \frac{\mu_{moon}}{\mu_{moon} + \mu_{earth}} \quad (4.44)$$

$$x_1 = L_1 = R(1 - \mu - \gamma_1) \quad (4.45)$$

$$x_2 = L_2 = R(1 - \mu + \gamma_2) \quad (4.46)$$

$$x_3 = L_3 = R(-\mu - \gamma_3) \quad (4.47)$$

Once a solution for γ_i is available, the instantaneous location of the Lagrange point in the barycentric frame is then found by substituting the calculated value for γ_i into Eqs. (4.45) - (4.47).

4.2 Relative Three-Body Dynamics

The scenarios of interest are relative trajectories with a deputy and chief satellite orbiting in the three-body dynamics domain. Figure 4.4 [77] shows the geometry of one such scenario. For measuring relative vectors, a coordinate frame with the same axes as the \mathcal{B} -frame is placed at the position of the chief satellite and is formally defined as the chief-centered barycentric rotating frame: $C : \{\mathbf{R}_c, \hat{\mathbf{i}}_{\mathcal{B}}, \hat{\mathbf{j}}_{\mathcal{B}}, \hat{\mathbf{k}}_{\mathcal{B}}\}$. Thus all relative position vectors: $\boldsymbol{\rho}_{\mathcal{B}} = \mathbf{R}_d - \mathbf{R}_c$ are written in the chief-centered rotating frame (C-frame).

4.2.1 ER3BP Relative Equations of Motion

In this section, the non-linear ER3BP relative equations of motion for the \mathcal{B} -frame are developed. The scenario geometry is shown in Fig. 4.4. The starting point for developing the relative EOMS is defining the relative position and velocity vectors from the chief to the deputy satellite as:

$$\boldsymbol{\rho}_{\mathcal{B}} = \begin{bmatrix} \rho_x & \rho_y & \rho_z \end{bmatrix}^T \quad (4.48)$$

$$\dot{\boldsymbol{\rho}}_{\mathcal{B}} = \begin{bmatrix} \dot{\rho}_x & \dot{\rho}_y & \dot{\rho}_z \end{bmatrix}^T \quad (4.49)$$

where the vectors are written in \mathcal{B} -frame coordinates. The position vector of the chief satellite is defined by $\mathbf{R}_c = \begin{bmatrix} x_c & y_c & z_c \end{bmatrix}^T$. At this point, it will be useful to define several scalar quantities relating the distances between the chief satellite to the Earth and the Moon. Let r_{ec} be the distance

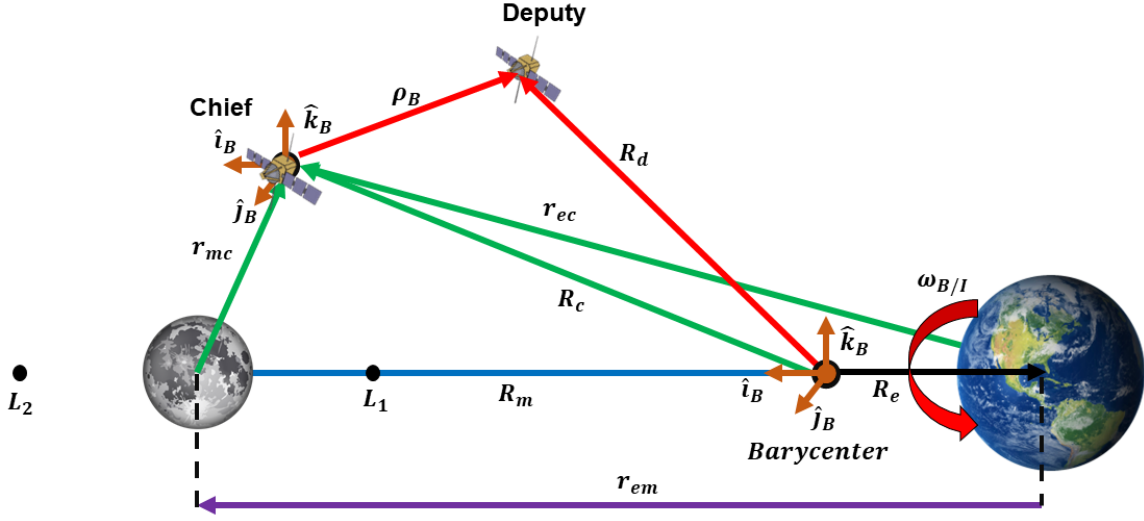


Figure 4.4 Geometry of a deputy and chief scenario in the ER3BP. Position vectors are shown in the rotating barycentric frame. Note the chief-centered relative frame (C-frame) is aligned with the rotating \mathcal{B} -frame.

from the Earth to the chief satellite, and r_{mc} be the distance from the Moon to the chief, with R_e the current Earth distance from the barycenter, and R_m the current Moon distance from the barycenter. (Also note that R_{em} is the current Earth-Moon distance computed with Eq. (4.3), and μ is the combined gravitational parameter for the Earth-Moon system as mentioned beforehand). To simplify notation, several vector norms that are used repeatedly will be defined.

$$r_{ec} = \sqrt{(x_c + R_e)^2 + y_c^2 + z_c^2} \quad (4.50)$$

$$r_{mc} = \sqrt{(x_c - R_m)^2 + y_c^2 + z_c^2} \quad (4.51)$$

$$R_e = \mu R_{em} \quad (4.52)$$

$$R_m = (1 - \mu)R_{em} \quad (4.53)$$

Given a deputy and chief satellite in orbit in cislunar space, the relative vector (ρ_B) from the chief to the deputy can be written in terms of the deputy (\mathbf{R}_d) and chief (\mathbf{R}_c) positions as follows.

$$\rho_B = \mathbf{R}_d - \mathbf{R}_c \quad (4.54)$$

To arrive at the relative EOM, two derivatives of Eq. (4.54) are taken with respect to time in the inertial frame using the transport theorem[75]. Noting that the rotating between the \mathcal{I} and \mathcal{B} frames is given by $\omega_{\mathcal{B}/\mathcal{I}}$, the inertial derivatives are written as follows.

$$\frac{d}{dt} [\boldsymbol{\rho}]_{\mathcal{I}} = \frac{d}{dt} [\boldsymbol{\rho}]_{\mathcal{B}} + \omega_{\mathcal{B}/\mathcal{I}} \times \boldsymbol{\rho}_{\mathcal{B}} \quad (4.55)$$

$$\frac{d}{dt} [\dot{\boldsymbol{\rho}}]_{\mathcal{I}} = \frac{d}{dt} \left[\frac{d}{dt} [\boldsymbol{\rho}]_{\mathcal{B}} + \omega_{\mathcal{B}/\mathcal{I}} \times \boldsymbol{\rho}_{\mathcal{B}} \right]_{\mathcal{B}} + \omega_{\mathcal{B}/\mathcal{I}} \times \left(\frac{d}{dt} [\boldsymbol{\rho}]_{\mathcal{B}} + \omega_{\mathcal{B}/\mathcal{I}} \times \boldsymbol{\rho}_{\mathcal{B}} \right) \quad (4.56)$$

$$[\ddot{\boldsymbol{\rho}}]_{\mathcal{I}} = \ddot{\boldsymbol{\rho}}_{\mathcal{B}} + 2\omega_{\mathcal{B}/\mathcal{I}} \times \dot{\boldsymbol{\rho}}_{\mathcal{B}} + \dot{\omega}_{\mathcal{B}/\mathcal{I}} \times \boldsymbol{\rho}_{\mathcal{B}} + \omega_{\mathcal{B}/\mathcal{I}} \times (\omega_{\mathcal{B}/\mathcal{I}} \times \boldsymbol{\rho}_{\mathcal{B}}) \quad (4.57)$$

Defining the inertial acceleration vector as a relative vector in terms of the inertial deputy and chief accelerations will allow further simplification.

$$[\ddot{\boldsymbol{\rho}}]_{\mathcal{I}} = [\ddot{\mathbf{R}}_d]_{\mathcal{I}} - [\ddot{\mathbf{R}}_c]_{\mathcal{I}} \quad (4.58)$$

The inertial acceleration of the deputy and chief, $[\ddot{\mathbf{R}}_d]_{\mathcal{I}}$ and $[\ddot{\mathbf{R}}_c]_{\mathcal{I}}$ are readily found using the inertial EOM given in Eq. (4.15).

$$[\ddot{\mathbf{R}}_d]_{\mathcal{I}} = -\mu_e \frac{\mathbf{r}_{ed}}{r_{ed}^3} - \mu_m \frac{\mathbf{r}_{md}}{r_{md}^3} \quad (4.59)$$

$$[\ddot{\mathbf{R}}_c]_{\mathcal{I}} = -\mu_e \frac{\mathbf{r}_{ec}}{r_{ec}^3} - \mu_m \frac{\mathbf{r}_{mc}}{r_{mc}^3} \quad (4.60)$$

Note that, $\mathbf{r}_{md} = \mathbf{r}_{mc} + \boldsymbol{\rho}_{\mathcal{B}}$ gives the deputy position with respect to the Moon, and $\mathbf{r}_{ed} = \mathbf{r}_{ec} + \boldsymbol{\rho}_{\mathcal{B}}$ gives the deputy position with respect to the Earth. Solving Eq. (4.57) for $\ddot{\boldsymbol{\rho}}_{\mathcal{B}}$, and substituting in for the deputy and chief accelerations using Eqs. (4.59) and (4.60), we arrive at the relative acceleration equation in Eq. (4.61).

$$\begin{aligned} \ddot{\boldsymbol{\rho}}_{\mathcal{B}} = & -2\omega_{\mathcal{B}/\mathcal{I}} \times \dot{\boldsymbol{\rho}}_{\mathcal{B}} - \dot{\omega}_{\mathcal{B}/\mathcal{I}} \times \boldsymbol{\rho}_{\mathcal{B}} - \omega_{\mathcal{B}/\mathcal{I}} \times (\omega_{\mathcal{B}/\mathcal{I}} \times \boldsymbol{\rho}_{\mathcal{B}}) \\ & + \mu_e \left(\frac{\mathbf{r}_{ec}}{r_{ec}^3} - \frac{\mathbf{r}_{ec} + \boldsymbol{\rho}_{\mathcal{B}}}{\|\mathbf{r}_{ec} + \boldsymbol{\rho}_{\mathcal{B}}\|^3} \right) + \mu_m \left(\frac{\mathbf{r}_{mc}}{r_{mc}^3} - \frac{\mathbf{r}_{mc} + \boldsymbol{\rho}_{\mathcal{B}}}{\|\mathbf{r}_{mc} + \boldsymbol{\rho}_{\mathcal{B}}\|^3} \right) \end{aligned} \quad (4.61)$$

The expression can be simplified by realizing that the angular velocity and angular acceleration terms $\omega_{\mathcal{B}/I}$ and $\dot{\omega}_{\mathcal{B}/I}$ act only along the $\hat{\mathbf{k}}_{\mathcal{B}}$ direction with magnitudes given by two-body motion (Eqs. (4.6) and (4.7)). Defining the skew-symmetric matrix $\left[\omega^\times\right]$ allows for the cross product terms to be simplified more easily.

$$\left[\omega^\times\right] = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \quad (4.62)$$

Using the definition of $\left[\omega^\times\right]$ in Eq. (4.62) allows for some simplification along with the gathering of terms (note that the definition for $\left[\dot{\omega}^\times\right]$ follows that of $\left[\omega^\times\right]$).

$$\begin{aligned} \ddot{\rho}_{\mathcal{B}} = & -2\left[\omega^\times\right]\dot{\rho}_{\mathcal{B}} - \left[\dot{\omega}^\times\right]\rho_{\mathcal{B}} - \left[\omega^\times\right]\left(\left[\omega^\times\right]\rho_{\mathcal{B}}\right) \\ & + \mu_e \left(\frac{\mathbf{r}_{ec}}{r_{ec}^3} - \frac{\mathbf{r}_{ec} + \rho_{\mathcal{B}}}{\|\mathbf{r}_{ec} + \rho_{\mathcal{B}}\|^3} \right) + \mu_m \left(\frac{\mathbf{r}_{mc}}{r_{mc}^3} - \frac{\mathbf{r}_{mc} + \rho_{\mathcal{B}}}{\|\mathbf{r}_{mc} + \rho_{\mathcal{B}}\|^3} \right) \end{aligned} \quad (4.63)$$

$$\begin{aligned} \ddot{\rho}_{\mathcal{B}} = & -2\left[\omega^\times\right]\dot{\rho}_{\mathcal{B}} - \left(\left[\dot{\omega}^\times\right] - \left[\omega^\times\right]^2\right)\rho_{\mathcal{B}} \\ & + \mu_e \left(\frac{\mathbf{r}_{ec}}{r_{ec}^3} - \frac{\mathbf{r}_{ec} + \rho_{\mathcal{B}}}{\|\mathbf{r}_{ec} + \rho_{\mathcal{B}}\|^3} \right) + \mu_m \left(\frac{\mathbf{r}_{mc}}{r_{mc}^3} - \frac{\mathbf{r}_{mc} + \rho_{\mathcal{B}}}{\|\mathbf{r}_{mc} + \rho_{\mathcal{B}}\|^3} \right) \end{aligned} \quad (4.64)$$

Further reduction of terms is realized by substitution for the chief position vectors with respect to the Earth and Moon, as positions with respect to the barycenter.

$$\mathbf{r}_{ec} = \mathbf{R}_c - \mathbf{R}_e \quad (4.65)$$

$$\mathbf{r}_{mc} = \mathbf{R}_c - \mathbf{R}_m \quad (4.66)$$

$$\mathbf{R}_e = -\mu R_{em} \hat{\mathbf{i}}_{\mathcal{B}} \quad (4.67)$$

$$\mathbf{R}_m = (1 - \mu) R_{em} \hat{\mathbf{i}}_{\mathcal{B}} \quad (4.68)$$

The Earth's position is given by \mathbf{R}_e and \mathbf{R}_m gives the Moon's position with respect to the barycenter. Additionally, let the distance of the chief and deputy satellites to the Earth and Moon be defined as

follows.

$$d_{ec} = \|\mathbf{R}_c - \mathbf{R}_e\| \quad (4.69)$$

$$d_{ed} = \|\mathbf{R}_c - \mathbf{R}_e + \boldsymbol{\rho}_{\mathcal{B}}\| \quad (4.70)$$

$$d_{mc} = \|\mathbf{R}_c - \mathbf{R}_m\| \quad (4.71)$$

$$d_{md} = \|\mathbf{R}_c - \mathbf{R}_m + \boldsymbol{\rho}_{\mathcal{B}}\| \quad (4.72)$$

Note that, the angular velocity vector $\boldsymbol{\omega}_{\mathcal{B}/I} = \omega \hat{k}_{\mathcal{B}}$, so that the skew-symmetric vector $\left[\boldsymbol{\omega}^\times\right]$ only has the z -terms populated. Substituting Eqs. (4.69) - (4.72) into Eq. (4.64) yields an easier to use set of equations:

$$\begin{aligned} \ddot{\boldsymbol{\rho}}_{\mathcal{B}} = & -2 \left[\boldsymbol{\omega}^\times \right] \dot{\boldsymbol{\rho}}_{\mathcal{B}} - \left(\left[\dot{\boldsymbol{\omega}}^\times \right] - \left[\boldsymbol{\omega}^\times \right]^2 \right) \boldsymbol{\rho}_{\mathcal{B}} \\ & + \mu_e \left(\frac{\mathbf{R}_c - \mathbf{R}_e}{d_{ec}^3} - \frac{\mathbf{R}_c - \mathbf{R}_e + \boldsymbol{\rho}_{\mathcal{B}}}{d_{ed}^3} \right) \\ & + \mu_m \left(\frac{\mathbf{R}_c - \mathbf{R}_m}{d_{mc}^3} - \frac{\mathbf{R}_c - \mathbf{R}_m + \boldsymbol{\rho}_{\mathcal{B}}}{d_{md}^3} \right) \end{aligned} \quad (4.73)$$

which only depends on the chief position, deputy relative position and velocity, and the Earth-Moon two-body motion. The scalar non-linear relative equations of motion in the \mathcal{B} are written by working out the cross product and vector terms in Eq. (4.73). Expanding all vectors and working out cross-product gives us the scalar non-linear relative equations of motion for the ER3BP in the \mathcal{B} -frame.

$$\begin{aligned} \ddot{\rho}_x = & 2\omega\dot{\rho}_y + \dot{\omega}\rho_y - \omega^2\rho_x + \frac{\mu_e}{d_{ec}^3}(x_c + R_e) - \frac{\mu_e}{d_{ed}^3}(x_c + R_e + \rho_x) \\ & + \frac{\mu_m}{d_{mc}^3}(x_c - R_m) - \frac{\mu_m}{d_{md}^3}(x_c - R_m + \rho_x) \end{aligned} \quad (4.74)$$

$$\begin{aligned} \ddot{\rho}_y = & -2\omega\dot{\rho}_x - \dot{\omega}\rho_x - \omega^2\rho_y + \frac{\mu_e}{d_{ec}^3}y_c - \frac{\mu_e}{d_{ed}^3}(y_c + \rho_y) \\ & + \frac{\mu_m}{d_{mc}^3}y_c - \frac{\mu_m}{d_{md}^3}(y_c + \rho_y) \end{aligned} \quad (4.75)$$

$$\ddot{\rho}_z = \frac{\mu_e}{d_{ec}^3} z_c - \frac{\mu_e}{d_{ed}^3} (z_c + \rho_z) + \frac{\mu_m}{d_{mc}^3} z_c - \frac{\mu_m}{d_{md}^3} (z_c + \rho_z) \quad (4.76)$$

Equations (4.74) - (4.76) will be referred to as the non-linear equations of relative motion for the ER3BP (NLERM-E) model. The component form equations will be useful for developing a linearized relative motion model in the coming sections. First, however, the non-linear equations will be adapted to the CR3BP.

4.2.2 CR3BP Relative Equations of Motion

Similar to the absolute EOMs, the relative EOMs for a deputy satellite in the ER3BP can be adapted to the CR3BP motion model. The coordinate frames are the same, but a few quantities need to be changed to reflect the CR3BP motion model. The non-linear equations of relative motion for the CR3BP are given in vector form by:

$$\begin{aligned} \ddot{\rho}_{\mathcal{B}} = & -2 \left[\omega^\times \right] \dot{\rho}_{\mathcal{B}} - \left[\omega^\times \right]^2 \rho_{\mathcal{B}} + \mu_e \left(\frac{\mathbf{R}_c - \mathbf{R}_e}{d_{ec}^3} - \frac{\mathbf{R}_c - \mathbf{R}_e + \rho_{\mathcal{B}}}{d_{ed}^3} \right) \\ & + \mu_m \left(\frac{\mathbf{R}_c - \mathbf{R}_m}{d_{mc}^3} - \frac{\mathbf{R}_c - \mathbf{R}_m + \rho_{\mathcal{B}}}{d_{md}^3} \right) \end{aligned} \quad (4.77)$$

The EOMs can also be written in scalar form by the working out of all vector terms using the definitions for the CR3BP.

$$\begin{aligned} \ddot{\rho}_x = & 2\omega\dot{\rho}_y - \omega^2\rho_x + \frac{\mu_e}{d_{ec}^3} (x_c + R_e) - \frac{\mu_e}{d_{ed}^3} (x_c + R_e + \rho_x) \\ & + \frac{\mu_m}{d_{mc}^3} (x_c - R_m) - \frac{\mu_m}{d_{md}^3} (x_c - R_m + \rho_x) \end{aligned} \quad (4.78)$$

$$\begin{aligned} \ddot{\rho}_y = & -2\omega\dot{\rho}_x - \omega^2\rho_y + \frac{\mu_e}{d_{ec}^3} y_c - \frac{\mu_e}{d_{ed}^3} (y_c + \rho_y) \\ & + \frac{\mu_m}{d_{mc}^3} y_c - \frac{\mu_m}{d_{md}^3} (y_c + \rho_y) \end{aligned} \quad (4.79)$$

$$\ddot{\rho}_z = \frac{\mu_e}{d_{ec}^3} z_c - \frac{\mu_e}{d_{ed}^3} (z_c + \rho_z) + \frac{\mu_m}{d_{mc}^3} z_c - \frac{\mu_m}{d_{md}^3} (z_c + \rho_z) \quad (4.80)$$

Equations (4.78) - (4.80) will be referred to as the non-linear equations of relative motion for the CR3BP (NLERM-C).

4.3 Linearized Relative Motion for the Three-Body Problem

The non-linear three-body motion models are useful for propagation of trajectories, but admit little in the way of analytical solutions (apart from the Lagrange equilibrium points). The development of a linear set of equations is highly desirable and has application for differential correction for periodic orbit calculation, as well as spacecraft dynamics and control [30, 38, 78, 79]. To date there are no known general analytical solutions to the ER3BP or its associated relative motion model. From the linearized set of relative equations it has been proposed to form a semi-analytical solution for relative motion using a finite exponential matrix solution for short time-spans [77, 79, 80]. The original derivation of the linearized dynamics and exponential solution is expanded here to include the barycentric frame of motion as well as the CR3BP. In both the ER3BP and CR3BP cases, the linearized EOMs are obtained by taking a first-order Taylor series of the non-linear relative motion equations. In order to perform the linearization, the chief's position states will be assumed to be known and serve as the trajectory that is linearized about. At this point no assumptions have been made other than those for the ER3BP and CR3BP models.

4.3.1 ER3BP Linear Motion Model

The linear motion model for the ER3BP is derived by forming the non-linear relative equations of motion (NLERM-B) as a state-space derivative vector and then finding the Jacobian of the non-linear function with respect to the relative states. The ER3BP relative states and derivative vectors are given by the following equations.

$$\mathbf{X} = \begin{bmatrix} \rho_x & \rho_y & \rho_z & \dot{\rho}_x & \dot{\rho}_y & \dot{\rho}_z \end{bmatrix}^T \quad (4.81)$$

$$\dot{\mathbf{X}} = \mathbf{f}(t) = \begin{bmatrix} \dot{\rho}_x & \dot{\rho}_y & \dot{\rho}_z & \ddot{\rho}_x & \ddot{\rho}_y & \ddot{\rho}_z \end{bmatrix}^T \quad (4.82)$$

The acceleration terms $\ddot{\rho}_x$, $\ddot{\rho}_y$, and $\ddot{\rho}_z$ are given by the \mathcal{B} -frame non-linear acceleration terms in Eqs. (4.74) - (4.76).

Now the linear model is formed by finding the Jacobian matrix of the non-linear function $\mathbf{f}(t)$. The goal is to find a linear system of equations that takes the standard (uncontrolled) linear time-varying state-space form shown below.

$$\dot{\mathbf{X}} = A(t)\mathbf{X}_0 \quad (4.83)$$

The matrix $A(t)$ gives the linearized state matrix and is implicitly time dependent due to both the deputy and chief motion. The initial condition for the relative states is given by \mathbf{X}_0 . The state matrix $A(t)$ has a block matrix structure composed of four 3×3 sub matrices of varying complexity. The upper left, and upper right blocks are zero and identity blocks respectively. The lower left-block ($A_\rho(t)$) is fully populated and results from the Jacobian terms with respect to the relative position of the deputy, and the lower right block (A_ρ) has only two terms and results from the Jacobian with respect to the relative velocity.

$$A(t)_{i,j} = \frac{\partial f_i(t)}{\partial \mathbf{X}_j} |_{\mathbf{x}, \mathbf{x}_c} \quad (4.84)$$

$$A(t) = \begin{bmatrix} 0_{3 \times 3} & I_{3 \times 3} \\ A_\rho(t) & A_\rho \end{bmatrix} \quad (4.85)$$

The i, j component of $A(t)_{i,j}$ is given by the appropriate partial derivative in Eq. (4.84), and the partial derivatives are evaluated at the current deputy and chief positions. The principal terms of interest are in the fully populated $A_\rho(t)$ sub matrix, and are found by using Eq. (4.84) for $i = 4, 5, 6$ and $j = 1, 2, 3$. The partials of interest can be written in Jacobian form as:

$$A_\rho(t) = \begin{bmatrix} \frac{\partial \ddot{\rho}_x}{\partial \rho_x} & \frac{\partial \ddot{\rho}_x}{\partial \rho_y} & \frac{\partial \ddot{\rho}_x}{\partial \rho_z} \\ \frac{\partial \ddot{\rho}_y}{\partial \rho_x} & \frac{\partial \ddot{\rho}_y}{\partial \rho_y} & \frac{\partial \ddot{\rho}_y}{\partial \rho_z} \\ \frac{\partial \ddot{\rho}_z}{\partial \rho_x} & \frac{\partial \ddot{\rho}_z}{\partial \rho_y} & \frac{\partial \ddot{\rho}_z}{\partial \rho_z} \end{bmatrix} \quad (4.86)$$

The relative acceleration terms ($\ddot{\rho}_x, \ddot{\rho}_y$, and $\ddot{\rho}_z$) are given by their barycentric forms in Eqs. (4.74)

- (4.76). The members of the fully populated Jacobean sub-matrix A_ρ are shown in Eqs. (4.87) - (4.95).

$$A_\rho(1, 1) = \frac{\partial \ddot{\rho}_x}{\partial \rho_x} = -\omega^2 - \frac{\mu_e}{d_{ed}^3} \left(\frac{3(x_c + R_e + \rho_x)^2}{d_{ed}^2} - 1 \right) + \frac{\mu_m}{d_{md}^3} \left(\frac{3(x_c - R_m + \rho_x)^2}{d_{md}^2} - 1 \right) \quad (4.87)$$

$$A_\rho(1, 2) = \frac{\partial \ddot{\rho}_x}{\partial \rho_y} = \dot{\omega} + \frac{3\mu_e(y_c + \rho_y)(x_c + R_e + \rho_x)}{d_{ed}^5} + \frac{3\mu_m(y_c + \rho_y)(x_c - R_m + \rho_x)}{d_{md}^5} \quad (4.88)$$

$$A_\rho(1, 3) = \frac{\partial \ddot{\rho}_x}{\partial \rho_z} = \frac{3\mu_e(z_c + \rho_z)(x_c + R_e + \rho_x)}{d_{ed}^5} + \frac{3\mu_m(z_c + \rho_z)(x_c - R_m + \rho_x)}{d_{md}^5} \quad (4.89)$$

$$A_\rho(2, 1) = \frac{\partial \ddot{\rho}_y}{\partial \rho_x} = -\dot{\omega} + \frac{3\mu_e(x_c + R_e + \rho_x)(y_c + \rho_y)}{d_{ed}^5} + \frac{3\mu_m(x_c - R_m + \rho_x)(y_c + \rho_y)}{d_{md}^5} \quad (4.90)$$

$$A_\rho(2, 2) = \frac{\partial \ddot{\rho}_y}{\partial \rho_y} = -\omega^2 + \frac{\mu_e}{d_{ed}^3} \left(\frac{3(y_c + \rho_y)^2}{d_{ed}^2} - 1 \right) + \frac{\mu_m}{d_{md}^3} \left(\frac{3(y_c + \rho_y)^2}{d_{md}^2} - 1 \right) \quad (4.91)$$

$$A_\rho(2, 3) = \frac{\partial \ddot{\rho}_y}{\partial \rho_z} = \frac{3\mu_e(z_c + \rho_z)(y_c + \rho_y)}{d_{ed}^5} + \frac{3\mu_m(z_c + \rho_z)(y_c + \rho_y)}{d_{md}^5} \quad (4.92)$$

$$A_\rho(3, 1) = \frac{\partial \ddot{\rho}_z}{\partial \rho_x} = \frac{3\mu_e(x_c + R_e + \rho_x)(z_c + \rho_z)}{d_{ed}^5} + \frac{3\mu_m(x_c - R_m + \rho_x)(z_c + \rho_z)}{d_{md}^5} \quad (4.93)$$

$$A_\rho(3, 2) = \frac{\partial \ddot{\rho}_z}{\partial \rho_y} = \frac{3\mu_e(y_c + \rho_y)(z_c + \rho_z)}{d_{ed}^5} + \frac{3\mu_m(y_c + \rho_y)(z_c + \rho_z)}{d_{md}^5} \quad (4.94)$$

$$A_\rho(3, 3) = \frac{\partial \ddot{\rho}_z}{\partial \rho_z} = \frac{\mu_e}{d_{ed}^3} \left(\frac{3(z_c + \rho_z)^2}{d_{ed}^2} - 1 \right) + \frac{\mu_m}{d_{md}^3} \left(\frac{3(z_c + \rho_z)^2}{d_{md}^2} - 1 \right) \quad (4.95)$$

Because the acceleration equations only have two terms with velocity components, the sub matrix A_ρ will only have two non-zero entries.

$$\frac{\partial \ddot{x}}{\partial \dot{y}} = 2\omega \quad (4.96)$$

$$\frac{\partial \ddot{y}}{\partial \dot{x}} = -2\omega \quad (4.97)$$

Thus, the matrix A_ρ can be simplified by pulling out the factor 2ω and defining a new matrix Ω as

follows.

$$A_{\dot{\rho}} = \begin{bmatrix} 0 & 2\omega & 0 \\ -2\omega & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = 2\omega \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (4.98)$$

$$\Omega = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (4.99)$$

Now the bottom right block matrix of $A(t)$ can be written compactly as $A_{\rho} = 2\omega\Omega$. Forming the block structured matrices together, the \mathcal{B} -frame linearized equations can be formed in standard state space form with initial conditions \mathbf{X}_0 .

$$\dot{\mathbf{X}} = A(t)\mathbf{X}_0 \quad (4.100)$$

$$\dot{\mathbf{X}} = \begin{bmatrix} 0_{3 \times 3} & I_{3 \times 3} \\ A_{\rho} & 2\omega\Omega \end{bmatrix} \mathbf{X}_0 \quad (4.101)$$

The state-space model for the linear equations of relative motion for the ER3BP (LERM-E) is now complete. The LERM-E model requires knowledge of the chief satellite's position, and the Earth-Moon two-body motion, as well as a relative initial condition. The LERM-E model can be propagated numerically using Eqs. (4.101) for the state-space model with terms defined by Eqs. (4.87) - (4.95). The Earth-Moon distance is modeled by solving Kepler's equation to find the true anomaly and solving for the current Earth-Moon distance using equation (4.3) and the Earth-Moon angular velocity and acceleration are found using Eqs. (4.6) and (4.7).

4.3.2 CR3BP Linear Motion Model

Although it is advantageous to model the Earth-Moon system dynamics more accurately through the ER3BP, much research in the literature for three-body dynamics focuses on the circular restricted three-body problem. The linearized dynamics model (LERM-E) can be easily transitioned to the

CR3BP by substituting for $\omega = n_{moon}$, $\dot{\omega} = 0$, and $R_{em} = a_{moon}$ in the same way that the non-linear EOMs were transitioned to the CR3BP. Recall that n_{moon} is the mean motion of the Moon, and represents the constant angular velocity of the CR3BP for the Earth-Moon system, and a_{moon} is the Earth-Moon semi-major axis, and represents the constant separation of the Earth and Moon in the CR3BP. Applying the CR3BP simplification is useful for a first check on the dynamical models developed as numerous periodic spacecraft initial conditions are readily available in the CR3BP. Note that simulation initial conditions were downloaded from NASA's Jet Propulsion Laboratory Solar System Dynamics website [81].

The linearized motion models for the CR3BP exactly follow the derivations for the ER3BP case, with the Earth-Moon system parameters being constant. The system state-vectors are defined the same:

$$\mathbf{X} = \begin{bmatrix} \rho_x & \rho_y & \rho_z & \dot{\rho}_x & \dot{\rho}_y & \dot{\rho}_z \end{bmatrix}^T \quad (4.102)$$

$$\dot{\mathbf{X}} = \mathbf{f}(t) = \begin{bmatrix} \dot{\rho}_x & \dot{\rho}_y & \dot{\rho}_z & \ddot{\rho}_x & \ddot{\rho}_y & \ddot{\rho}_z \end{bmatrix}^T \quad (4.103)$$

and the process of finding the Jacobian matrix to form the linear system mirrors the process for the ER3BP:

$$A(t)_{i,j} = \frac{\partial f_i(t)}{\partial \mathbf{X}_j} \big|_{\mathbf{x}, \mathbf{x}_c} \quad (4.104)$$

$$A(t) = \begin{bmatrix} 0_{3 \times 3} & I_{3 \times 3} \\ A_\rho(t) & A_{\dot{\rho}} \end{bmatrix} \quad (4.105)$$

$$\dot{\mathbf{X}} = A(t)\mathbf{X}_0 \quad (4.106)$$

$$\dot{\mathbf{X}} = \begin{bmatrix} 0_{3 \times 3} & I_{3 \times 3} \\ A_\rho(t) & 2\omega\Omega \end{bmatrix} \mathbf{X}_0 \quad (4.107)$$

The key changes in the linear state-matrix occur in the dropping of the $\dot{\omega}$ term, and the Earth-Moon distance being constant. The fully populated $A_\rho(t)$ matrix is given by Eqs. (4.108) - (4.116).

$$A_\rho(1, 1) = \frac{\partial \ddot{\rho}_x}{\partial \rho_x} = -\omega^2 - \frac{\mu_e}{d_{ed}^3} \left(\frac{3(x_c + R_e + \rho_x)^2}{d_{ed}^2 - 1} \right) + \frac{\mu_m}{d_{md}^3} \left(\frac{3(x_c - R_m + \rho_x)^2}{d_{md}^2} - 1 \right) \quad (4.108)$$

$$A_\rho(1, 2) = \frac{\partial \ddot{\rho}_x}{\partial \rho_y} = \frac{3\mu_e(y_c + \rho_y)(x_c + R_e + \rho_x)}{d_{ed}^5} + \frac{3\mu_m(y_c + \rho_y)(x_c - R_m + \rho_x)}{d_{md}^5} \quad (4.109)$$

$$A_\rho(1, 3) = \frac{\partial \ddot{\rho}_x}{\partial \rho_z} = \frac{3\mu_e(z_c + \rho_z)(x_c + R_e + \rho_x)}{d_{ed}^5} + \frac{3\mu_m(z_c + \rho_z)(x_c - R_m + \rho_x)}{d_{md}^5} \quad (4.110)$$

$$A_\rho(2, 1) = \frac{\partial \ddot{\rho}_y}{\partial \rho_x} = \frac{3\mu_e(x_c + R_e + \rho_x)(y_c + \rho_y)}{d_{ed}^5} + \frac{3\mu_m(x_c - R_m + \rho_x)(y_c + \rho_y)}{d_{md}^5} \quad (4.111)$$

$$A_\rho(2, 2) = \frac{\partial \ddot{\rho}_y}{\partial \rho_y} = -\omega^2 + \frac{\mu_e}{d_{ed}^3} \left(\frac{3(y_c + \rho_y)^2}{d_{ed}^2} - 1 \right) + \frac{\mu_m}{d_{md}^3} \left(\frac{3(y_c + \rho_y)^2}{d_{md}^2} - 1 \right) \quad (4.112)$$

$$A_\rho(2, 3) = \frac{\partial \ddot{\rho}_y}{\partial \rho_z} = \frac{3\mu_e(z_c + \rho_z)(y_c + \rho_y)}{d_{ed}^5} + \frac{3\mu_m(z_c + \rho_z)(y_c + \rho_y)}{d_{md}^5} \quad (4.113)$$

$$A_\rho(3, 1) = \frac{\partial \ddot{\rho}_z}{\partial \rho_x} = \frac{3\mu_e(x_c + R_e + \rho_x)(z_c + \rho_z)}{d_{ed}^5} + \frac{3\mu_m(x_c - R_m + \rho_x)(z_c + \rho_z)}{d_{md}^5} \quad (4.114)$$

$$A_\rho(3, 2) = \frac{\partial \ddot{\rho}_z}{\partial \rho_y} = \frac{3\mu_e(y_c + \rho_y)(z_c + \rho_z)}{d_{ed}^5} + \frac{3\mu_m(y_c + \rho_y)(z_c + \rho_z)}{d_{md}^5} \quad (4.115)$$

$$A_\rho(3, 3) = \frac{\partial \ddot{\rho}_z}{\partial \rho_z} = \frac{\mu_e}{d_{ed}^3} \left(\frac{3(z_c + \rho_z)^2}{d_{ed}^2} - 1 \right) + \frac{\mu_m}{d_{md}^3} \left(\frac{3(z_c + \rho_z)^2}{d_{md}^2} - 1 \right) \quad (4.116)$$

The distances d_{ij} for the deputy and chief satellites are defined as for the ER3BP in Eqs. (4.69) - (4.72). The CR3BP linear motion model will be referred to as the linear equations of relative motion for the CR3BP (LERM-C). The LERM-C motion model is defined by Eq. (4.107) with A_ρ defined by Eqs. (4.108) - (4.116).

4.3.3 Matrix Exponential Solution

While propagating the linearized relative motion models numerically can be helpful for trajectory generation, the numerical solution still defies any analytical analysis methods. Here it is proposed to use the matrix exponential to develop a semi-analytical solution for the linearized relative equations of motion. For a linear time invariant (LTI) system, the matrix exponential is the state-transition matrix and can be used as an analytical expression for the propagation of states forward in time [47]. The linearized models of relative motion for the three-body problem (LERM-E and LERM-C),

however, are in fact linear time-varying (LTV) systems since their state matrices change with time (due to the chief and deputy motion). Suppose the state matrix $A(t_k)$ is discretized over a short time-span $\Delta t = t_{k+1} - t_k$, during which $A(t_k)$ can be considered constant. Then a matrix exponential solution of the standard form shown in Eq. (4.117) can be found.

$$\mathbf{X}(t_k) = e^{A(t_k)\Delta t} \mathbf{X}(t_{k-1}) \quad (4.117)$$

$$e^{A(t_k)\Delta t} = \left[I + A(t_k)\Delta t + \frac{A(t_k)^2 \Delta t^2}{2!} + \dots + \frac{A(t_k)^n \Delta t^n}{n!} \right] \quad (4.118)$$

The number of terms taken for the infinite series controls to some extent the desired accuracy of the approximation. The state matrix for the linearized relative motion model $A(t_k)$ is defined either by Eq. (4.101) or Eq. (4.107) for the ER3BP or CR3BP system respectively.

Due to the time-varying nature of $A(t_k)$ for either system, the system matrix must be stepped forward in time with the states in order to propagate the relative states forward in time. This stepping can be accomplished using Eq. (4.119) with $A(t_k)$ computed at time t_k from the appropriate linear state matrix. This advances the states to t_{k+1} , and then the process continues for each time step where the state-matrix is recomputed and the states are stepped forward in time.

$$\mathbf{X}(t_{k+1}) = e^{A(t_k)\Delta t} \mathbf{X}(t_k) \quad (4.119)$$

This propagation method will be referred to as the linear exponential matrix model (LEXPM), where for the ER3BP the method will be referred to as LEXPM-E, and for the CR3BP as LEXPM-C.

There are numerous methods to compute the matrix exponential, with perhaps the most straight forward being the series expansion taken to a desired number of terms. Comparison between computation methods showed little difference in the final resulting simulations [48]. The key assumption of the LEXPM models remains that during the short time-step Δt the state-matrix $A(t_k)$ can be approximated as constant. A key advantage of the LEXPM models for future development is that an analytical solution for the states is available when a finite number of terms for $e^{A(t_k)\Delta t}$ are computed. Having an analytical solution for the motion is especially advantageous for dynamics

and control applications, as well as for space situational awareness. In two-body dynamics relative spacecraft state-estimation has been shown to work for second-order CW equations in a process called initial relative orbit determination (IROD) [78]. Using a similar framework it may be possible to use the LEXPM motion models to derive an initial relative orbit estimate and is a subject of future research.

4.4 Linearized Relative Motion Results

Representative scenarios were run in MATLAB to showcase the applicability and compare the accuracy of the various relative motion models. Note that, in all cases, the non-linear relative motion models are taken as the truth. Example scenarios are included for comparison cases in low-lunar orbit (LLO), low-Earth orbit (LEO), and geostationary orbit (GEO). Additionally several cases of three-body orbit geometry such as HALO, Lyapunov, resonant, and axial orbits are presented. Note that, for all scenarios the initial conditions for the deputy and chief are provided in non-dimensional units in a table accompanying the scenario. The initial conditions are all written in the \mathcal{B} -frame. Additionally, the non-dimensional distance unit (DU) and time unit (TU) are given by: $DU = 389703$ km and $TU = 382981$ seconds.

4.4.1 Low Lunar Orbit Example

The first example scenario is a stress case of the linear equations of relative motion with the deputy and chief placed in a low-lunar orbit. The rapidly changing orbit geometry of a LLO orbit offers a difficult case for linearized dynamics to model. The initial conditions for the deputy were set by back-propagation of the chief for 0.15 seconds and then numerically differencing the back-propagated state with the chief initial condition. This gives a “leader-follower” type geometry to the relative orbital dynamics and allows a straightforward comparison to the behavior we would expect from other linear models such as the CW equations. The deputy and chief initial conditions are given for both the ER3BP and the CR3BP scenarios in non-dimensional units in Table 4.1.

Figures 4.5a and 4.5b show the orbit geometry of the scenario around the Moon. The deputy and chief are in an inclined nearly circular orbit at an average altitude of 500 km above the Lunar surface. In order to achieve satisfactory results for the linear approximations for this rapidly changing orbit,

Table 4.1 LLO example deputy and chief initial conditions given in non-dimensional units for both the ER3BP and the CR3BP.

Initial Condition	Chief ER3BP	Deputy ER3BP	Chief CR3BP	Deputy CR3BP
x_0 [DU]	0.939069981	0.939069946	0.993302914	0.993302879
y_0 [DU]	0	1.63573E-05	0	1.63573E-05
z_0 [DU]	0	1.14535E-05	0	1.14535E-05
\dot{x}_0 [DU/TU]	0	-0.005296621	0	-0.005300904
\dot{y}_0 [DU/TU]	1.252910912	1.252902989	1.252910912	1.252902982
\dot{z}_0 [DU/TU]	0.877297665	0.877292064	0.877297665	0.877292064

a small time step (1 second) was required. Initial separation is about 1km from the chief satellite. Table 4.2 gives a summary of the root mean squared errors (RMSE) of each component of the deputy's position and velocity for each of the motion models for the scenario.

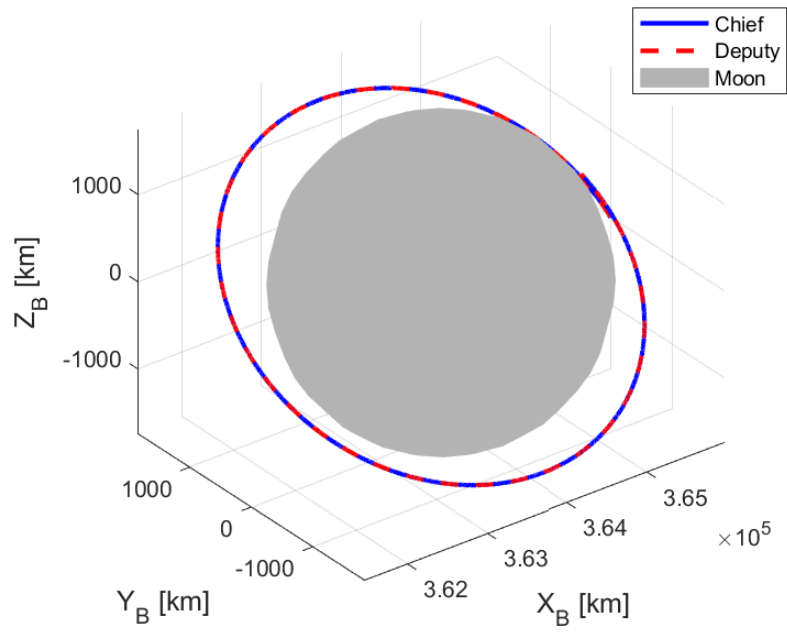
Table 4.2 Low lunar orbit example. RMS errors for linear motion models of ER3BP and CR3BP.

	LERM-E	LEXPM-E	LERM-C	LEXPM-C
RMSE: ρ_x [km]	2.700E-01	2.160E-01	2.692E-01	3.273E-01
RMSE: ρ_y [km]	2.270E-01	1.816E-01	2.278E-01	2.771E-01
RMSE: ρ_z [km]	1.591E-01	1.273E-01	1.596E-01	1.942E-01
RMSE: $\dot{\rho}_x$ [km/s]	1.844E-04	1.475E-04	1.853E-04	2.254E-04
RMSE: $\dot{\rho}_y$ [km/s]	1.337E-04	1.069E-04	1.334E-04	1.626E-04
RMSE: $\dot{\rho}_z$ [km/s]	9.298E-05	7.439E-05	9.289E-05	1.132E-04

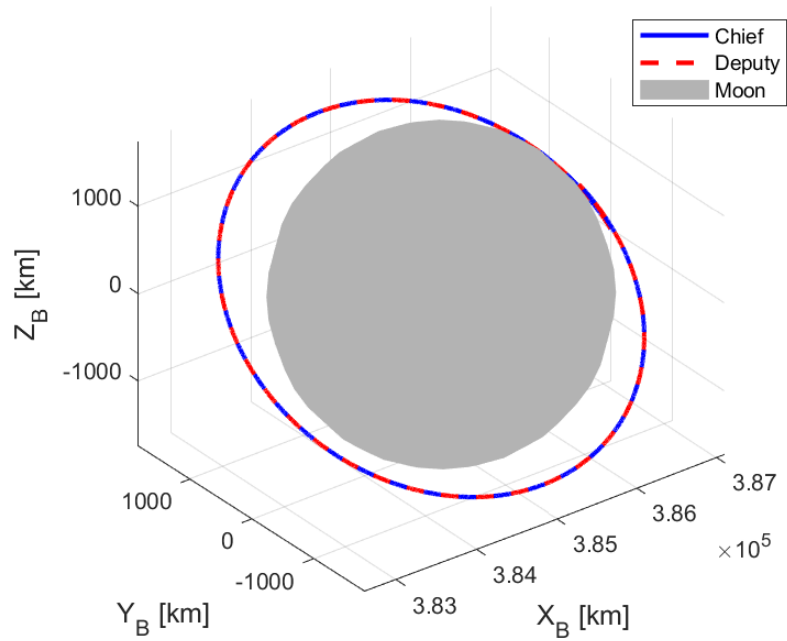
The relative motion in the “leader-follower” formation is shown in Figs. 4.6a and 4.6b for both the ER3BP and the CR3BP. Note that the relative frames are fixed to the chief and rotating at the rate of the Moon's angular rotation about the Earth.

The position evolution is better seen in 2D along each axis and is shown in Figs. 4.7a and 4.7b. Note the small separations of the models, indicating low errors. The errors for the propagation are shown along each axis in Figs. 4.8a and 4.8b. The models for the ER3BP and the CR3BP performed almost equally, and had peak errors of around 40 m in any single axis over the propagation period. The errors in velocity are shown in Figs. 4.9a and 4.9b with maximum error on the order of 20 cm/s.

Overall the linearized motion models in both frames held up well for the LLO case. However, this comes at the expense of a small time-step and small initial separation. An interesting note is that the exponential matrix models (LEXPM-E, LEXPM-C) outperformed the integrated linearized

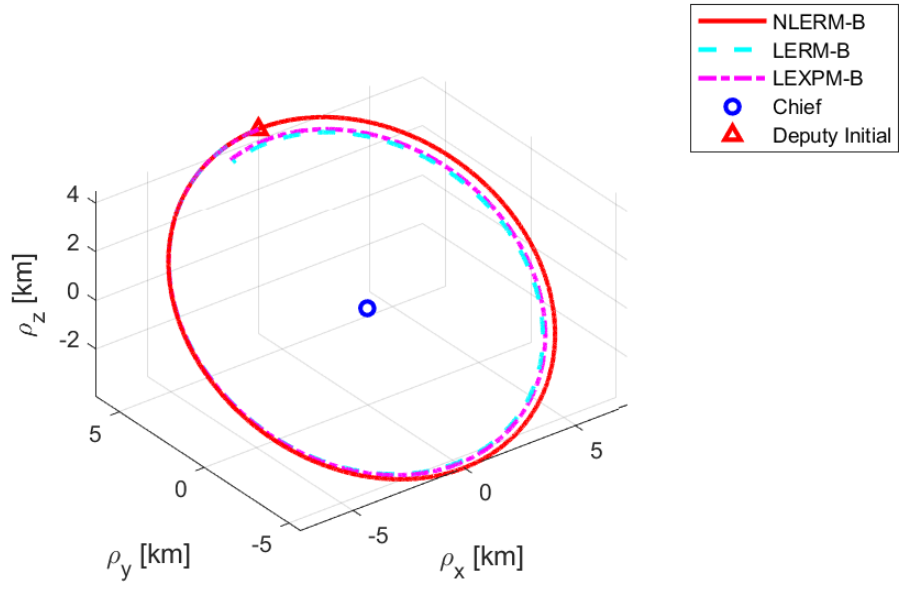


(a) Deputy and chief 3D position in the ER3BP

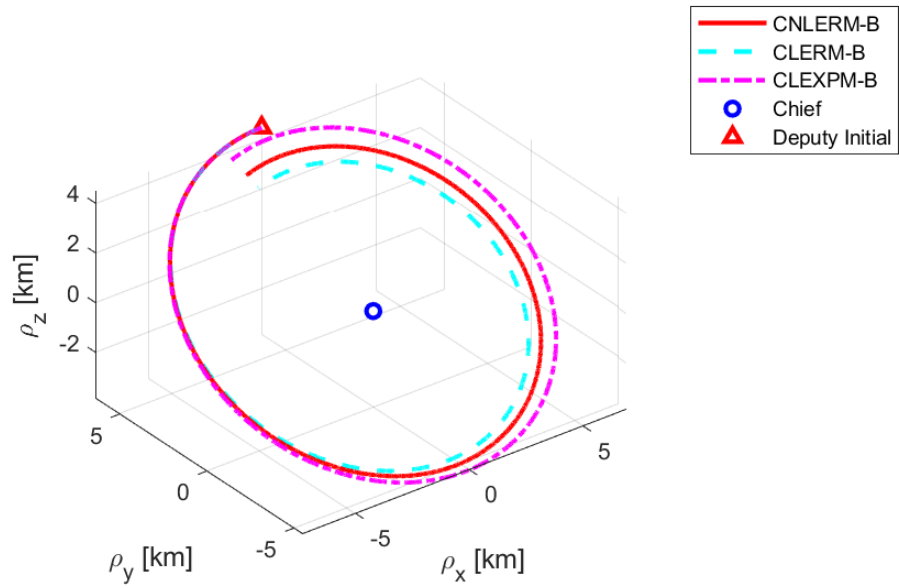


(b) Deputy and chief 3D position in the CR3BP

Figure 4.5 Low lunar orbit example: deputy and chief 3D positions.

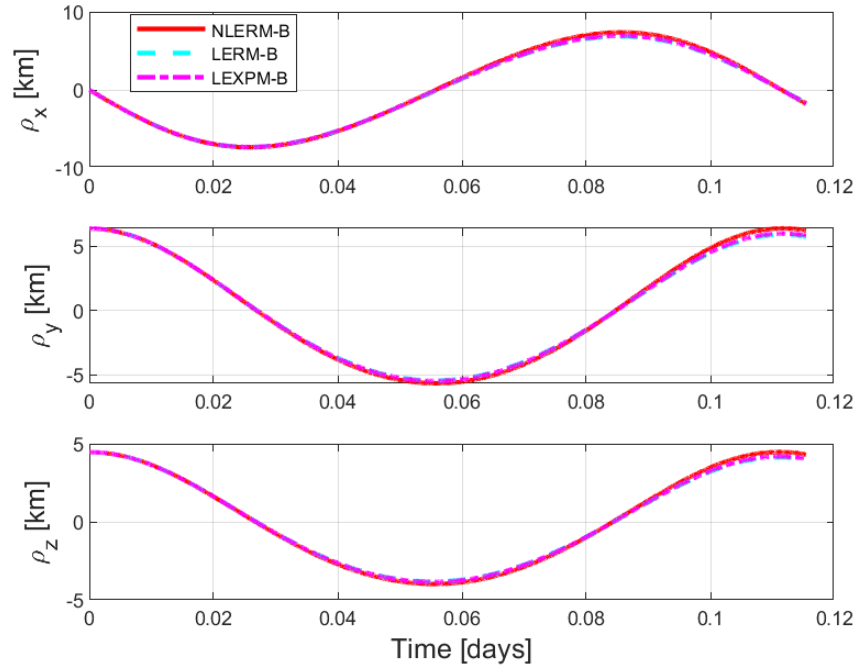


(a) Deputy and chief 3D position in the relative ER3BP.

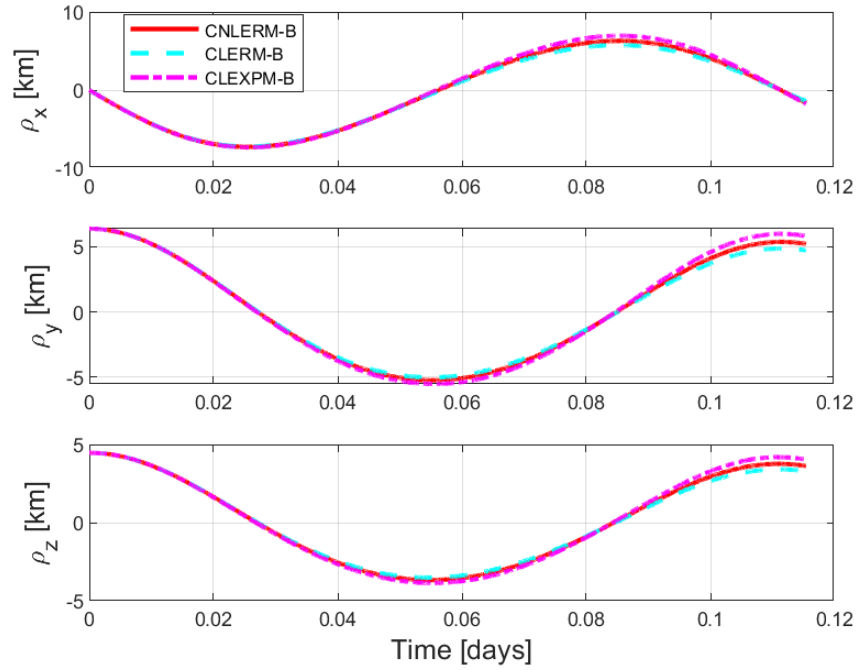


(b) Deputy and chief 3D position in the relative CR3BP.

Figure 4.6 Low lunar orbit example: 3D relative motion plots.

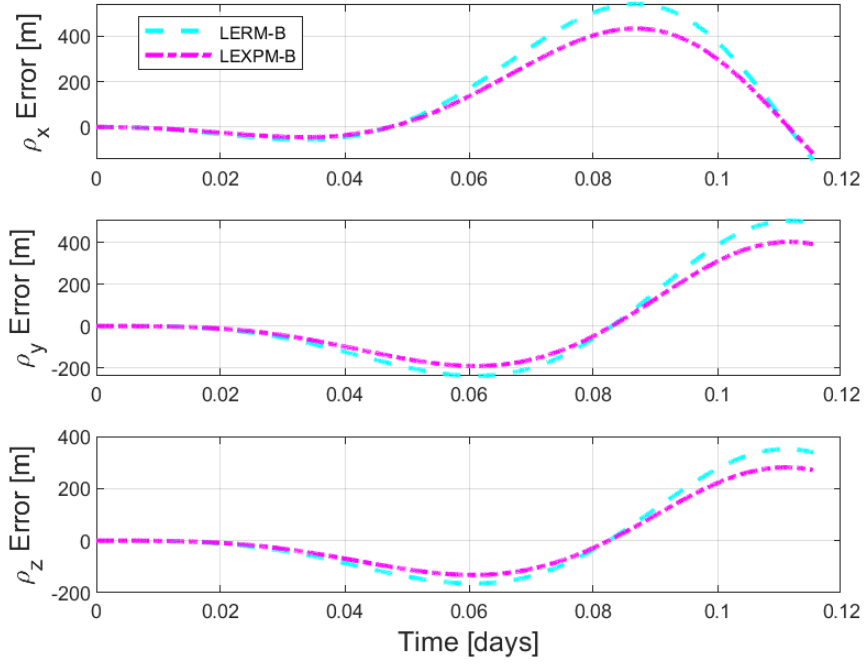


(a) Relative position in the ER3BP. Note that each motion model's position in each axes lies very close to the truth.

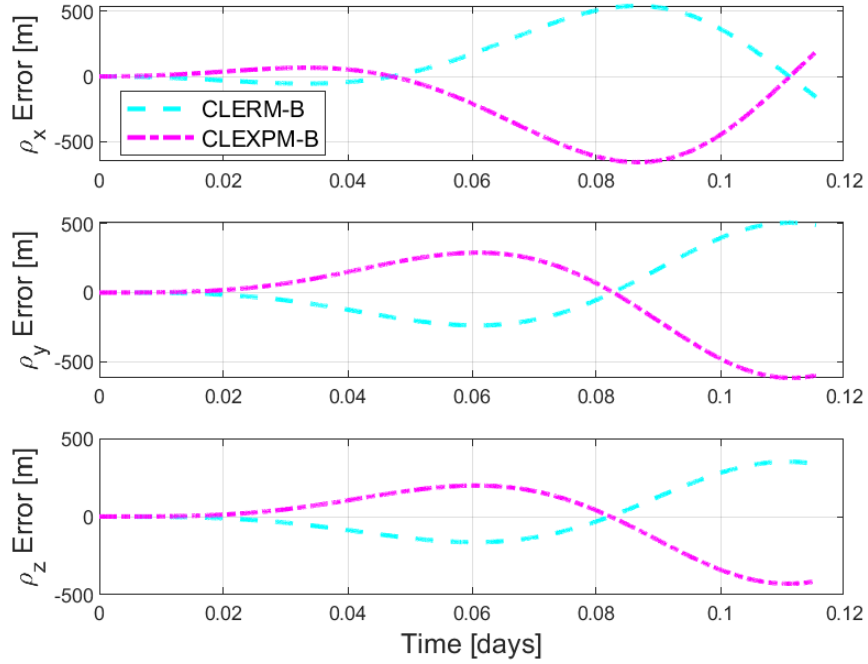


(b) Relative position in the CR3BP. Note that each motion model's position in each axes lies very close to the truth.

Figure 4.7 Low lunar orbit example: relative position plots.

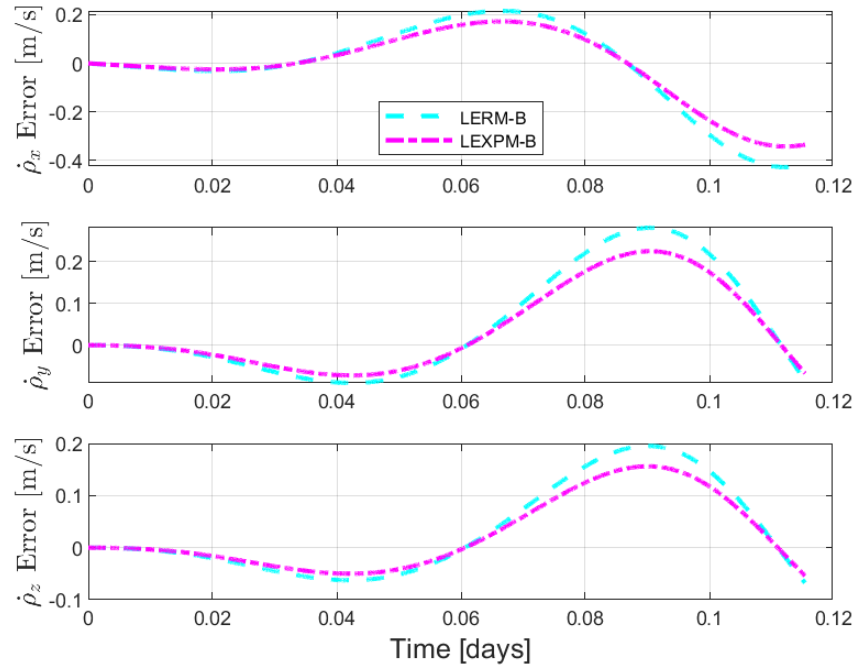


(a) ER3BP relative position error.

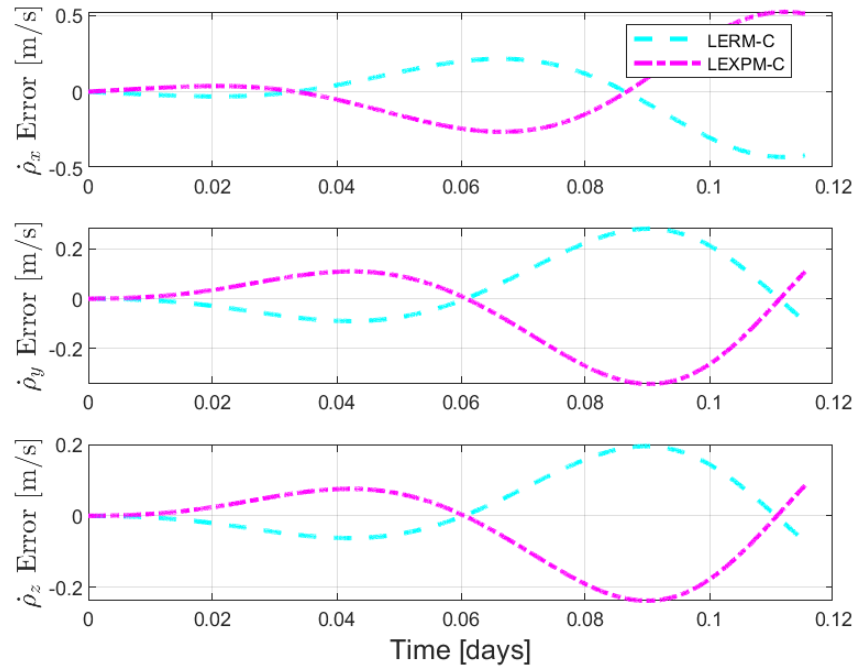


(b) CR3BP relative position error.

Figure 4.8 Low lunar orbit example: relative position errors.



(a) Relative velocity errors in the ER3BP.



(b) Relative velocity errors in the CR3BP.

Figure 4.9 Low lunar orbit example: relative velocity errors.

models (LERM-E, LERM-C);

4.4.2 Low-Earth Orbit Example

The next example considered is a case that places the chief satellite in an orbit roughly equivalent in shape to the international space station's (ISS) orbit in LEO, with classical orbital elements given by a semi-major axis of 6738 km, eccentricity of 0.0004215 and an inclination of 51.6427 degrees. The argument of perigee, right ascension of the ascending node (RAAN), and true anomaly were set to zero. The deputy initial position was then set by back propagation of the chief state by 1.5 seconds to give an initial separation close to 10 km. The deputy and chief initial conditions are given in non-dimensional units for both the ER3BP and the CR3BP in Table 4.3.

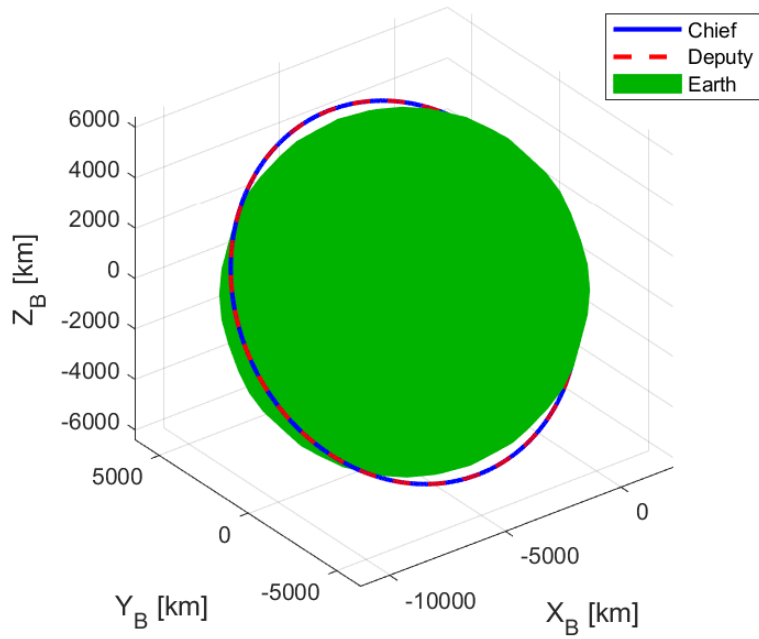
Table 4.3 LEO example deputy and chief initial conditions given in non-dimensional units for both the ER3BP and the CR3BP.

Initial Condition	Chief ER3BP	Deputy ER3BP	Chief CR3BP	Deputy CR3BP
x_0 [DU]	0.005799284	0.005799259	0.005132217	0.005132191
y_0 [DU]	0	1.83793E-05	0	1.83793E-05
z_0 [DU]	0	2.32245E-05	0	2.32245E-05
\dot{x}_0 [DU/TU]	0	-0.012912264	0	-0.01291636
\dot{y}_0 [DU/TU]	4.692628635	4.692621804	4.692628635	4.692621798
\dot{z}_0 [DU/TU]	5.929696621	5.929687917	5.929696621	5.929687917

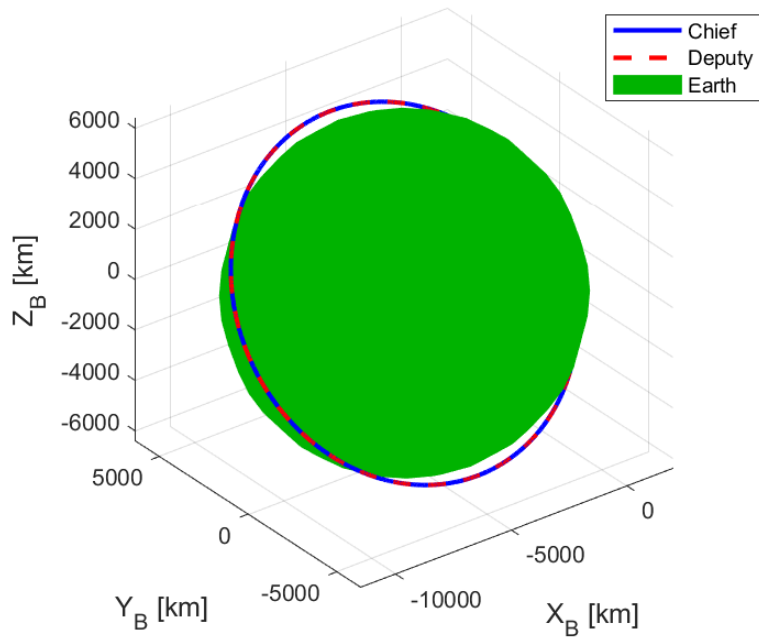
The orbital geometry is illustrated in Fig. 4.10. Note that for the LEXPM models to keep up with the rapidly changing orbit geometry, a time-step of 1 second was used. Table 4.4 shows the RMS errors for the deputy satellite position and velocity and gives a good overview of the linear model performance.

Table 4.4 Low Earth orbit example RMS errors for deputy position (in km) and velocity (in km/s) components over duration of scenario.

	LERM-E	LEXPM-E	LERM-C	LEXPM-C
RMSE: ρ_x [km]	1.736E-01	5.794E-02	1.733E-01	2.905E-01
RMSE: ρ_y [km]	1.000E-01	3.338E-02	1.002E-01	1.679E-01
RMSE: ρ_z [km]	1.267E-01	4.227E-02	1.269E-01	2.126E-01
RMSE: $\dot{\rho}_x$ [km/s]	1.823E-04	6.084E-05	1.828E-04	3.062E-04
RMSE: $\dot{\rho}_y$ [km/s]	1.127E-04	3.762E-05	1.125E-04	1.888E-04
RMSE: $\dot{\rho}_z$ [km/s]	1.416E-04	4.727E-05	1.415E-04	2.374E-04



(a) Deputy and chief 3D position in the ER3BP.



(b) Deputy and chief 3D position in the CR3BP.

Figure 4.10 Low Earth orbit example: deputy and chief 3D positions.

The scenario is roughly a leader-follower formation, with the deputy circumnavigating the chief using only natural dynamics. The relative frame geometry of the example is shown in Fig. 4.11. Both the ER3BP and CR3BP simulations performed very similarly.

The relative position for both motion models is shown in Fig. 4.12, and the errors in position are shown in Fig. 4.13. The LEXPM models performed better in tracking the relative position than the LERM models. This can also be seen in the velocity errors shown in Fig. 4.14. The difference is also clearly visualized when looking at the range and speed errors (Fig. 4.15). Overall the performance of the linear motion errors begin to decline and errors began to increase significantly after half an orbit period of the chief.

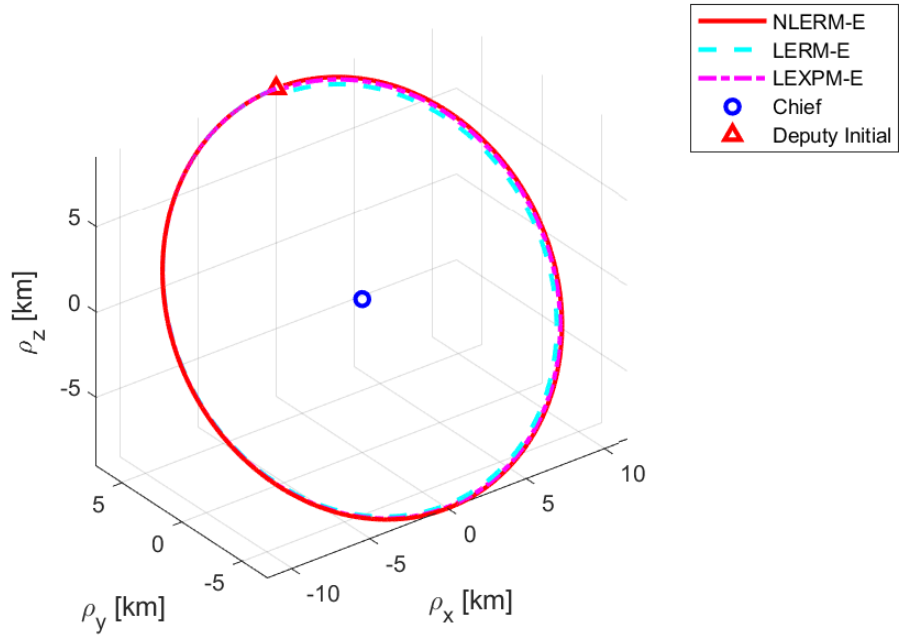
4.4.3 Geostationary Example

The next example scenario is for a geostationary (GEO) orbit of the chief satellite. The deputy initial condition is set by back propagation of the chief for a short time, until a relative separation of around 20 km is achieved. The GEO scenario also closely resembles a relative circumnavigation case, with the circumnavigation almost complete at the end of the propagation time. The deputy and chief initial conditions in non-dimensional units are shown for both the ER3BP and the CR3BP in Table 4.5.

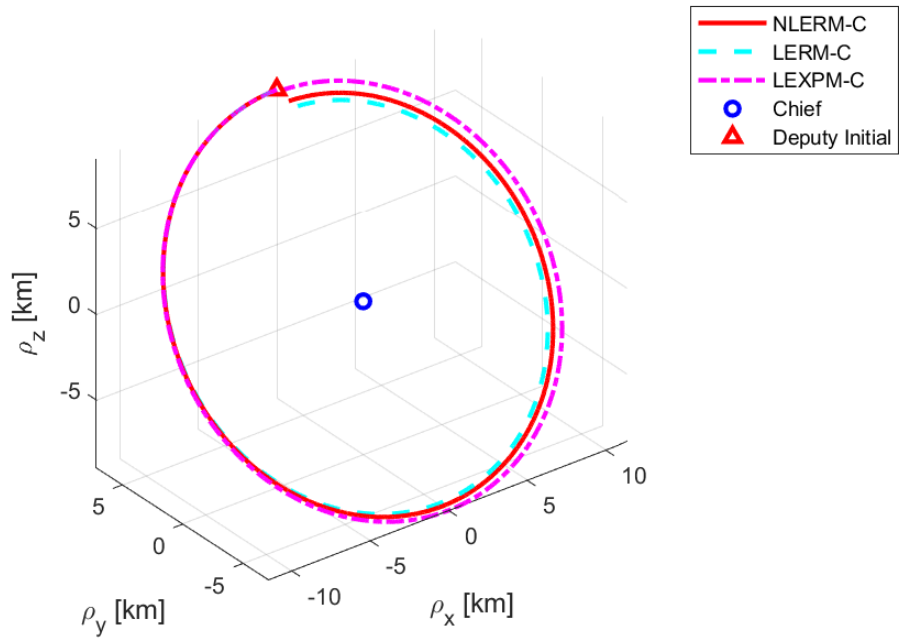
Table 4.5 GEO example deputy and chief initial conditions given in non-dimensional units for both the ER3BP and the CR3BP.

Initial Condition	Chief ER3BP	Deputy ER3BP	Chief CR3BP	Deputy CR3BP
x_0 [DU]	0.097868988	0.097868343	0.097201921	0.097201269
y_0 [DU]	0	0.000392395	0	0.000392395
z_0 [DU]	0	0	0	0
\dot{x}_0 [DU/TU]	0	-0.00989514	0	-0.009985647
\dot{y}_0 [DU/TU]	3.005599579	3.005581695	3.005599579	3.005581557
\dot{z}_0 [DU/TU]	0	0	0	0

The geometry of the scenario can be seen in Fig. 4.16, where both the ER3BP and CR3BP simulations performed very similarly once again. Overall the errors in the ER3BP models were slightly smaller than those incurred by the CR3BP models. The deputy motion in the relative frame is shown in Fig. 4.17. Table 4.6 shows the RMS errors for the deputy position and velocity for each

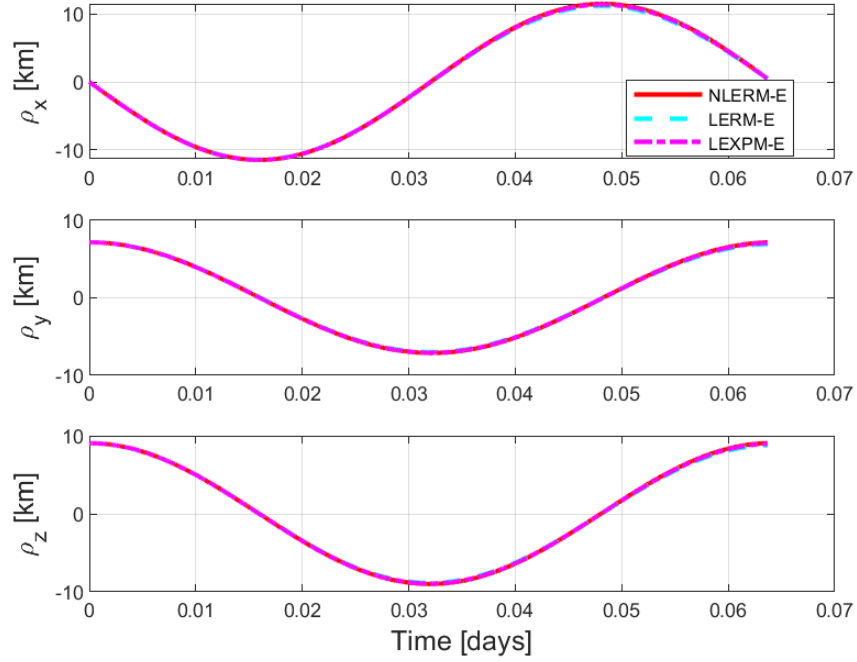


(a) Deputy and chief 3D position in the relative ER3BP

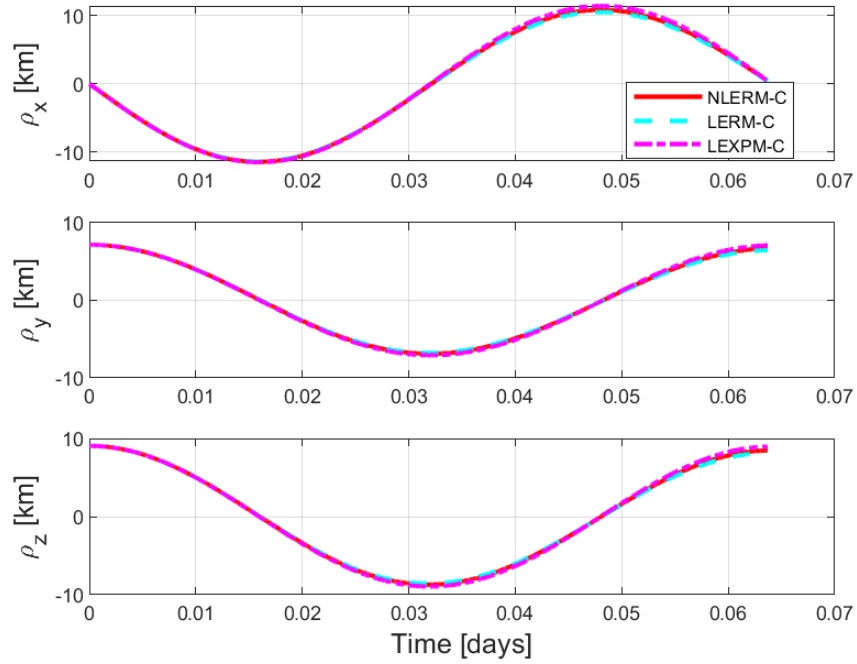


(b) Deputy and chief 3D position in the relative CR3BP.

Figure 4.11 Low Earth orbit example: 3D relative motion plots.

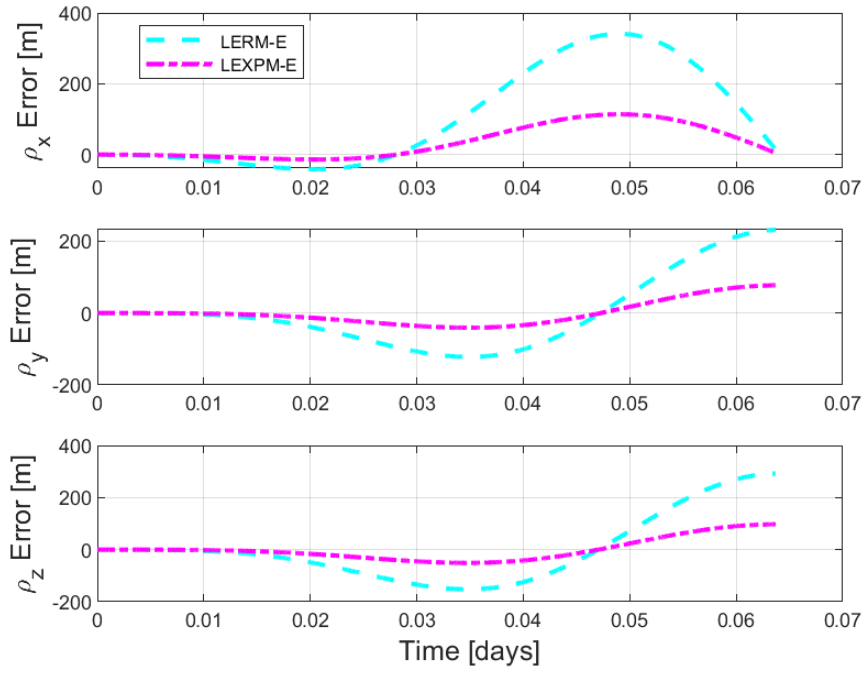


(a) Relative position in the ER3BP. Note that each motion model's position in each axes lies very close to the truth.

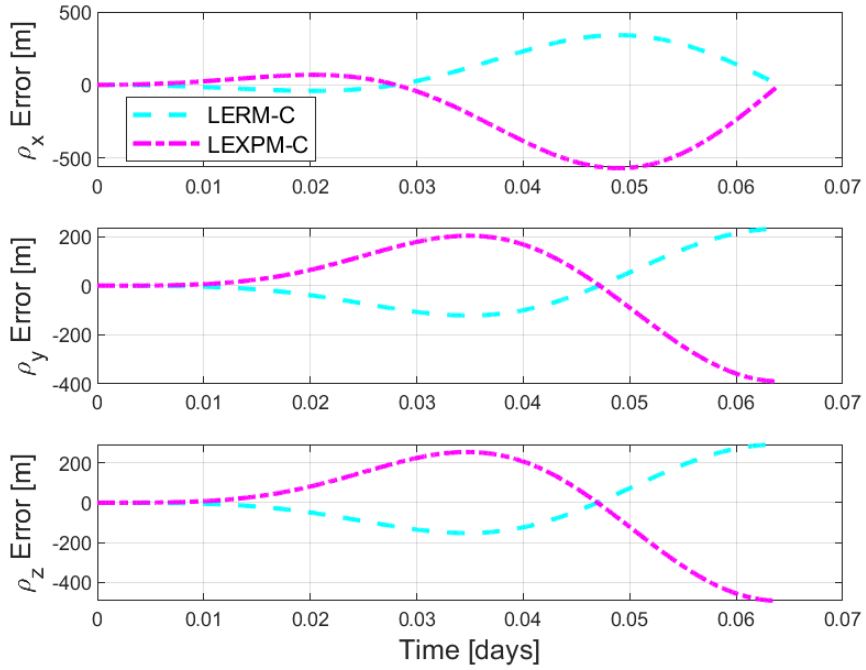


(b) Relative position in the CR3BP. Note that each motion model's position in each axes lies very close to the truth.

Figure 4.12 Low Earth orbit example: 3D relative motion plots.

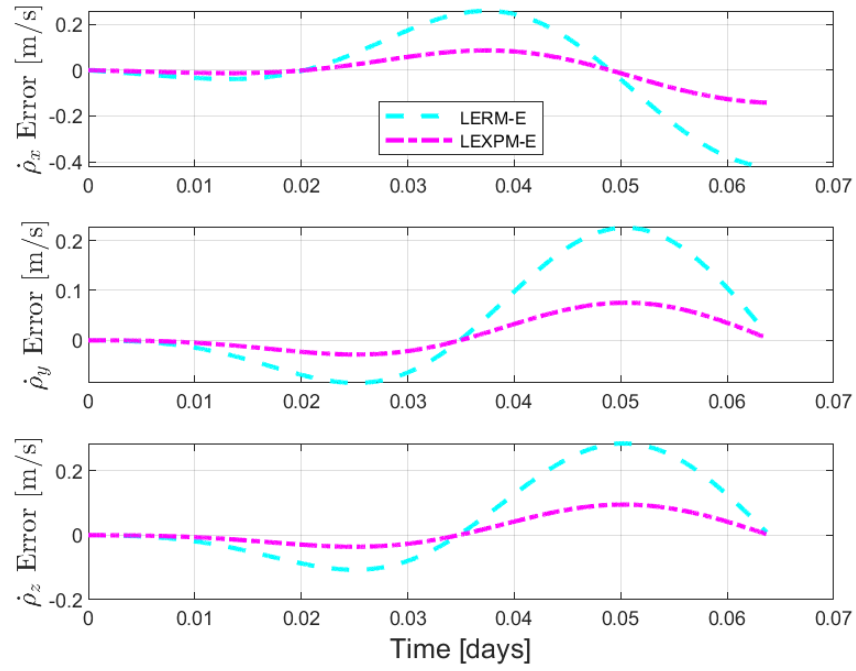


(a) Relative position errors in the ER3BP.

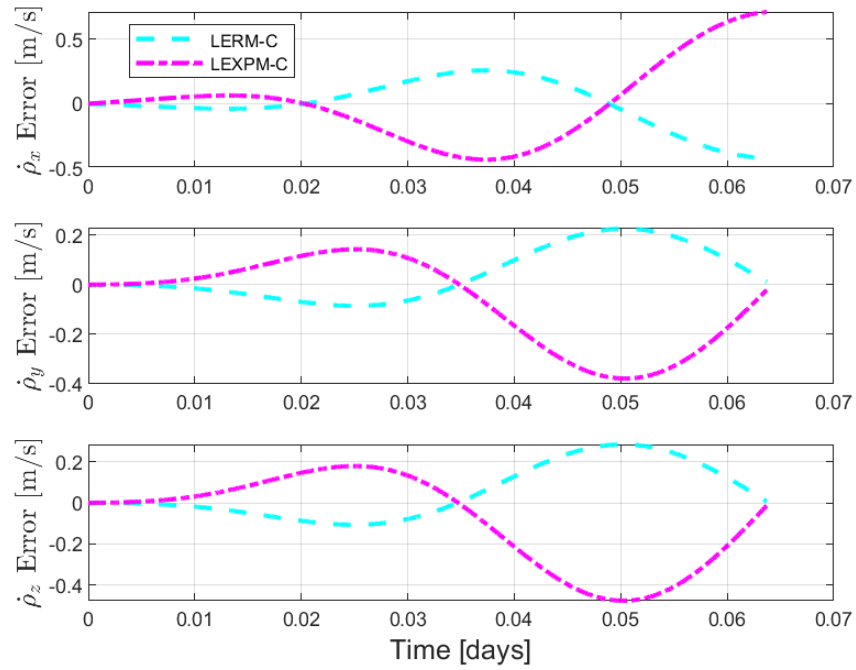


(b) CR3BP relative position errors.

Figure 4.13 Low Earth orbit example: relative position errors.

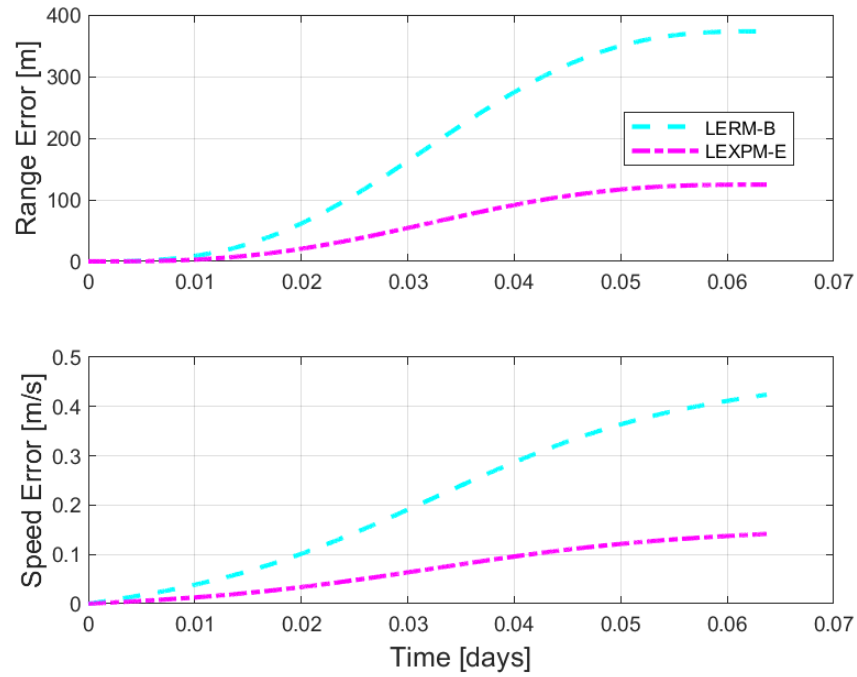


(a) Relative velocity errors in the ER3BP

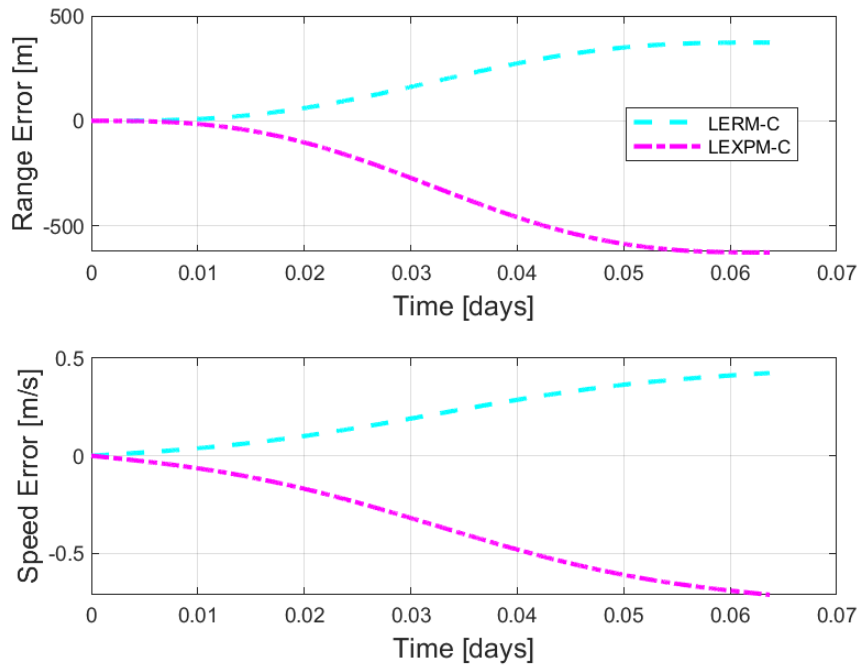


(b) Relative velocity errors in the CR3BP.

Figure 4.14 Low Earth orbit example: relative velocity errors.



(a) Relative range and speed errors in the ER3BP.



(b) Relative range and speed errors in the CR3BP.

Figure 4.15 Low Earth orbit example: range and speed errors.

linear motion model over the course of the scenario. Overall the CR3BP motion models held much closer to the non-linear truth model than the ER3BP models.

Table 4.6 Geostationary orbit RMS errors for deputy position and velocity components.

	LERM-E	LEXPM-E	LERM-C	LEXPM-C
RMSE: ρ_x [km]	5.521	4.969	5.550	6.180
RMSE: ρ_y [km]	3.640	3.276	3.646	4.053
RMSE: ρ_z [km]	0	0	0	0
RMSE: $\dot{\rho}_x$ [km/s]	2.213E-04	1.992E-04	2.246E-04	2.503E-04
RMSE: $\dot{\rho}_y$ [km/s]	2.987E-04	2.688E-04	3.095E-04	3.454E-04
RMSE: $\dot{\rho}_z$ [km/s]	0	0	0	0

The relative position errors for both models are shown in Fig 4.18. Once again, the LEXPM models outperformed the LERM linear propagation models in terms of error in the ρ_x and ρ_y axes. A similar result can be seen for the velocity error in Fig. 4.19. For both position and velocity, errors start to increase dramatically around half of the orbit period of the chief. Similar trends can be seen in the range and speed errors shown in Fig. 4.20. Of note, it appears that the LEXPM models “undershoot” on the range and speed error while the LERM models “overshoot” as seen in Fig. 4.20.

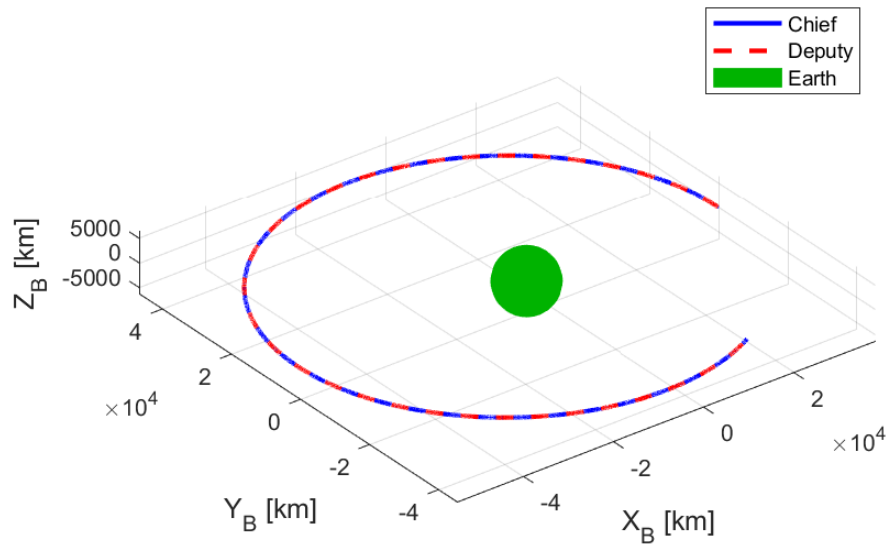
4.4.4 Lyapunov Orbit Example

This section contains a Lyapunov type orbit example. The orbit initial condition is taken from NASA’s JPL Horizon’s three-body periodic orbit database [81]. The deputy and chief initial conditions in non-dimensional units are given for both the ER3BP and the CR3BP in Table 4.7.

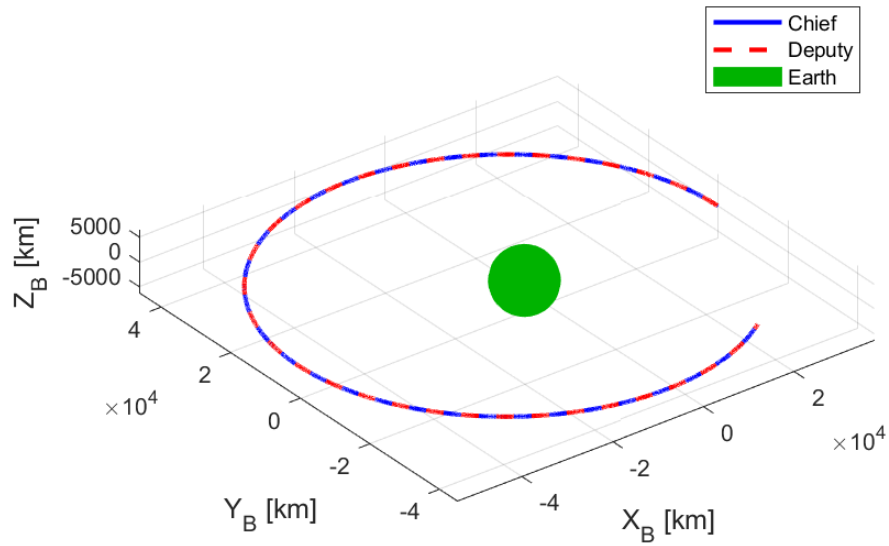
Table 4.7 Lyapunov orbit example deputy and chief initial conditions given in non-dimensional units for both the ER3BP and the CR3BP.

Initial Condition	Chief ER3BP	Deputy ER3BP	Chief CR3BP	Deputy CR3BP
x_0 [DU]	0.518491814	0.518491814	0.518491814	0.518491814
y_0 [DU]	0	2.87454E-05	0	2.87454E-05
z_0 [DU]	0	0	0	0
\dot{x}_0 [DU/TU]	1.89539E-13	-9.40519E-06	1.89539E-13	-1.91337E-05
\dot{y}_0 [DU/TU]	1.100895303	1.100895301	1.100895303	1.100895301
\dot{z}_0 [DU/TU]	0	0	0	0

Table 4.8 shows the RMS errors for the deputy position and velocity components for the scenario

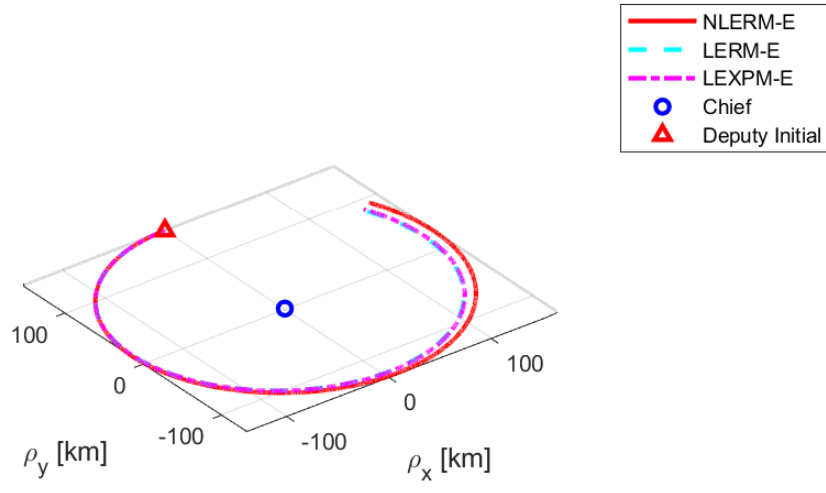


(a) Deputy and chief 3D position in the ER3BP.

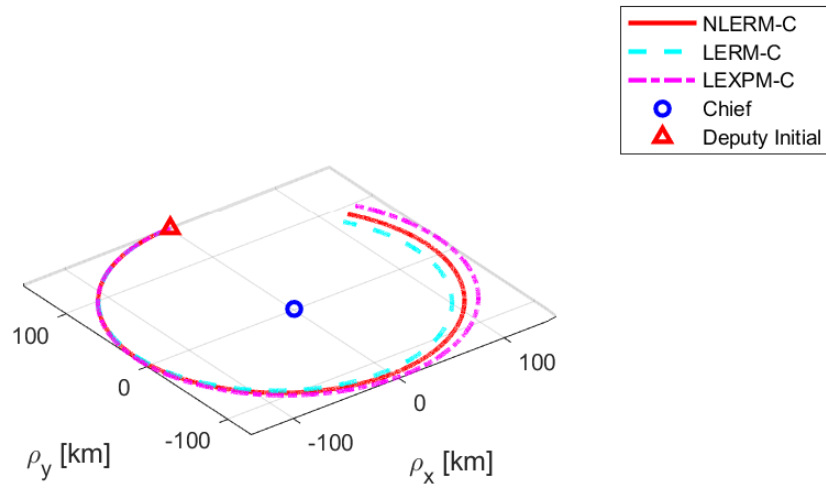


(b) Deputy and chief 3D position in the CR3BP.

Figure 4.16 Geostationary orbit example: deputy and chief 3D positions.

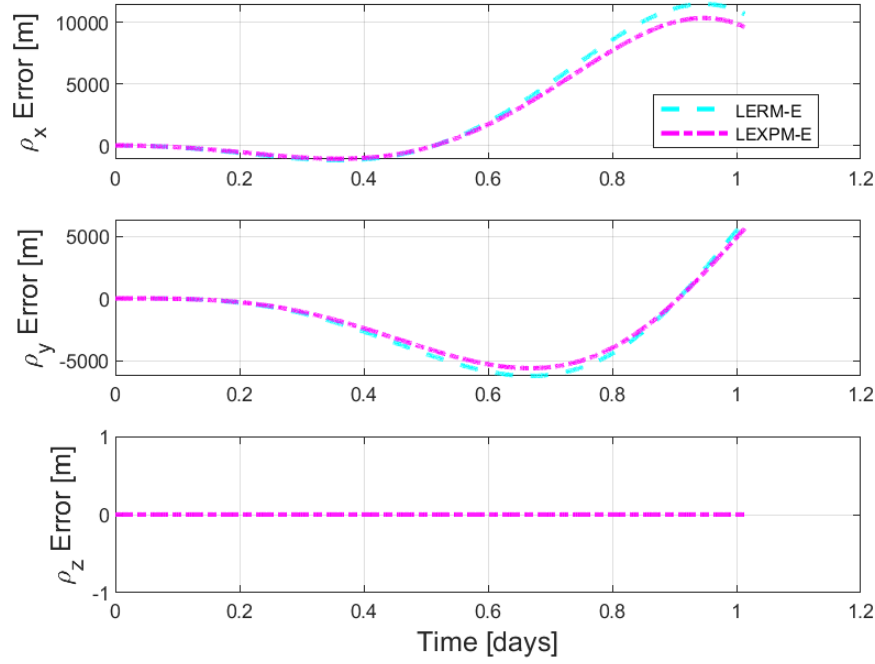


(a) Deputy and chief 3D position in the relative ER3BP.

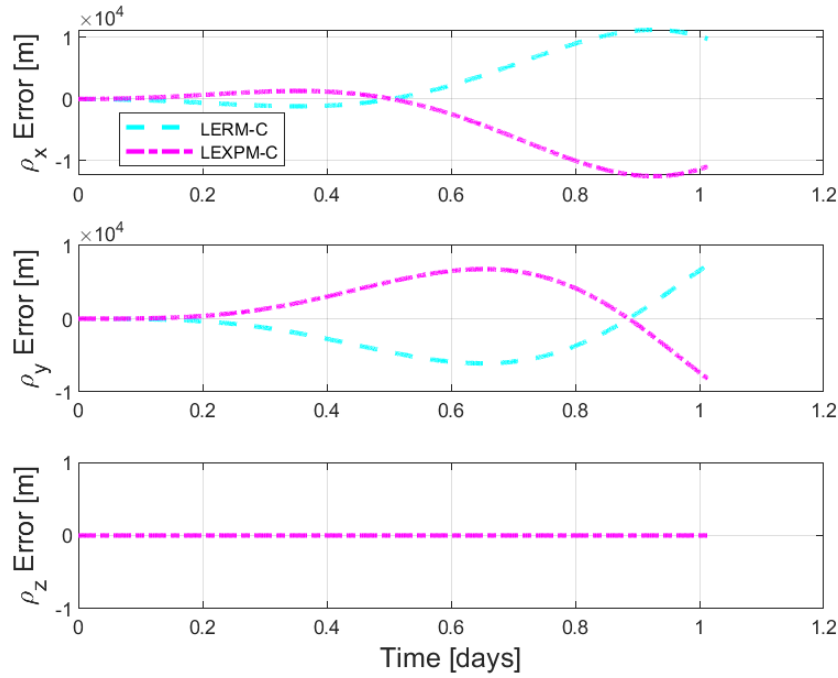


(b) Deputy and chief 3D position in the relative CR3BP.

Figure 4.17 Geostationary orbit example: 3D relative motion plots.

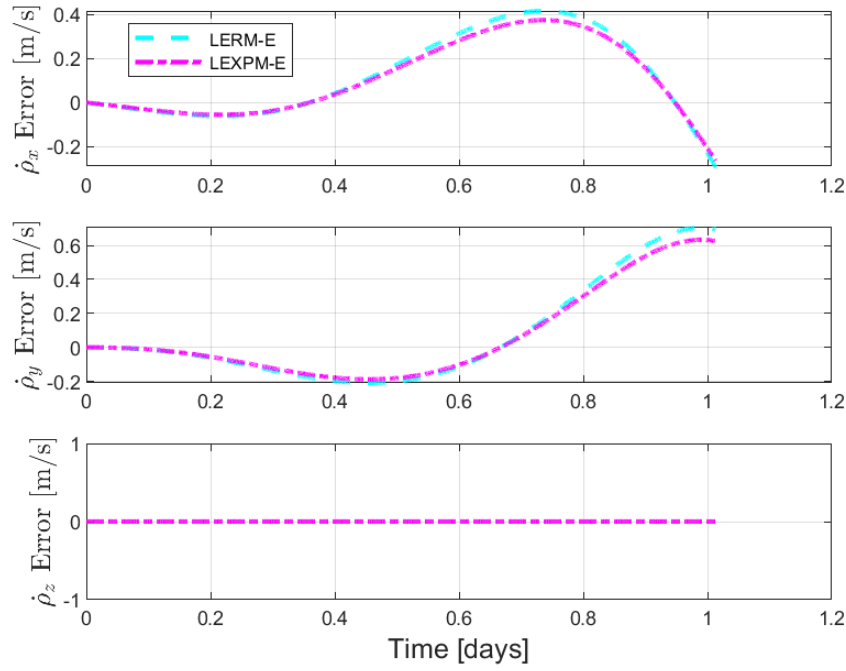


(a) ER3BP relative position error.

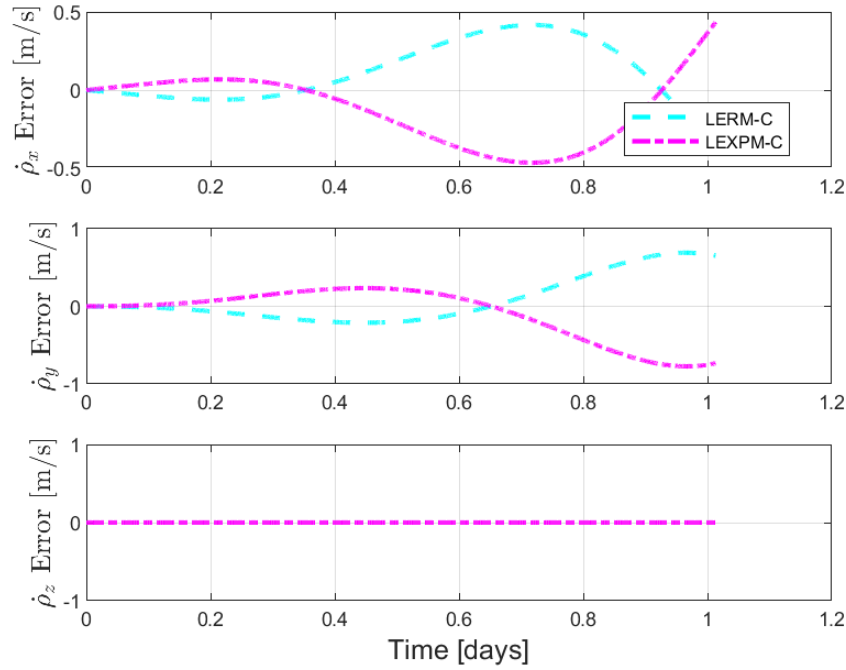


(b) CR3BP relative position error.

Figure 4.18 Geostationary orbit example: relative position errors.

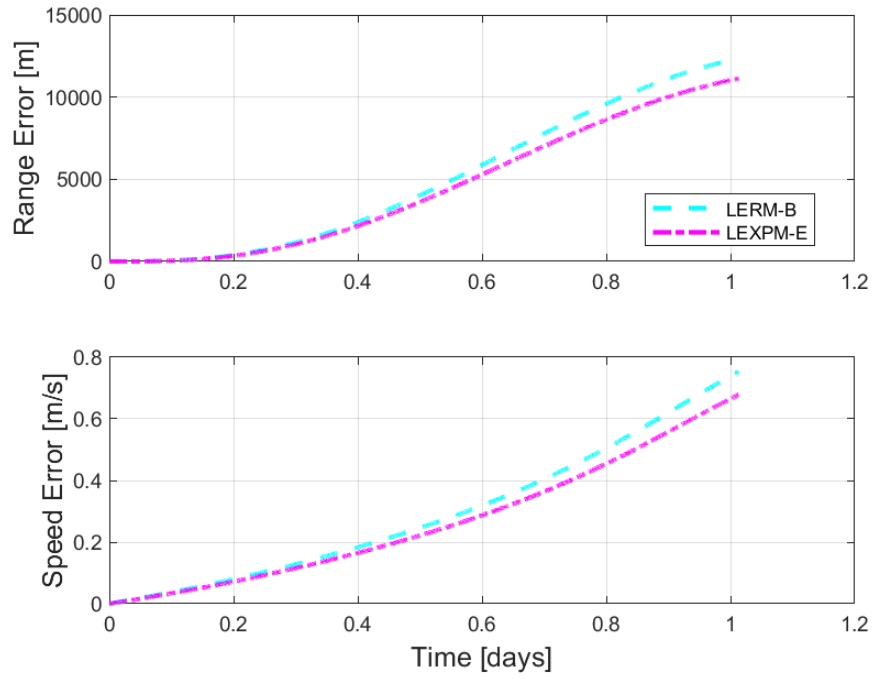


(a) ER3BP relative velocity error.

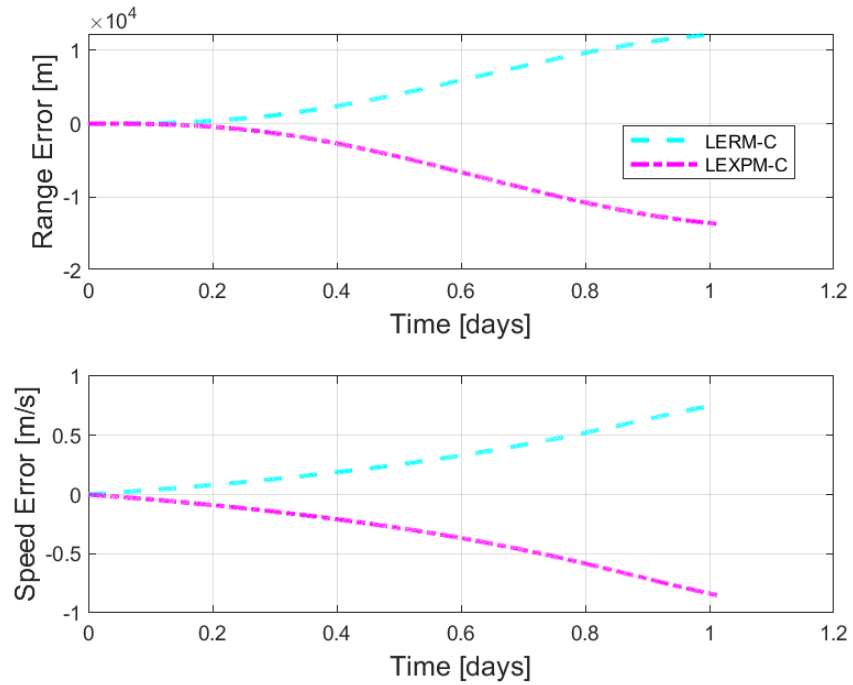


(b) CR3BP relative velocity error.

Figure 4.19 Geostationary orbit example: relative velocity errors.



(a) Relative range and speed errors in the ER3BP.



(b) Relative range and speed errors in the CR3BP.

Figure 4.20 Geostationary orbit example: range and speed errors.

for each motion model. The tabular summary gives a good indication of the closeness that the respective linear models held to the truth non-linear models. The Lyapunov orbit geometry is clearly seen in the CR3BP shown in Fig. 4.21b; notably, some of the orbit geometry is preserved in the ER3BP as shown in Fig. 4.21a. The orbit was propagated for one orbit of the chief satellite, which was roughly for 32 days with a time-step of 100 seconds. The relative position of the deputy satellite is shown for both ER3BP and CR3BP in Fig. 4.22. The position errors are shown in Fig. 4.23, where the ER3BP model has significantly lower error throughout the simulation. Most likely this is caused by the periapsis that occurs in the CR3BP model and increases the model error significantly after 16 days. Note that, until periapsis, the CR3BP linear models followed the truth model closely.

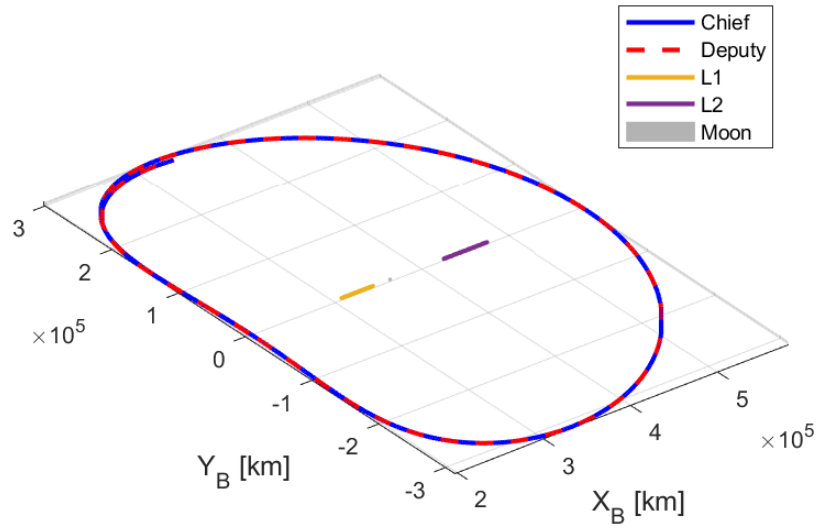
The velocity errors are shown in Fig. 4.24 which show the same effect of the periapsis on the CR3BP models increasing the velocity errors drastically around the day 16 mark of the simulation. Note that velocity errors for the ER3BP linear models remain at sub cm/s levels as shown in Fig. 4.24a. The range and speed errors for the simulation are shown in Fig. 4.25 and provide an easy visual comparison as to the difference in error magnitudes for the ER3BP linear models and the CR3BP linear models for this scenario. Overall the ER3BP models outperformed the CR3BP models by a large margin. The CR3BP models for this scenario might benefit from a variable time-step that allowed for much smaller integration steps near the lunar periapsis.

Table 4.8 Lyapunov orbit example RMS errors for deputy position and velocity components.

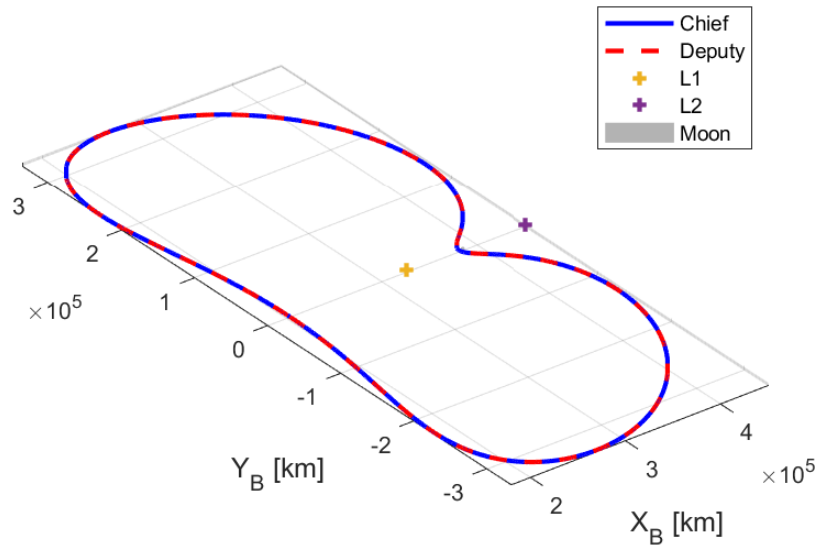
	LERM-E	LEXPM-E	LERM-C	LEXPM-C
RMSE: ρ_x [km]	3.585E-03	6.206E-03	5.485E-01	4.353E-01
RMSE: ρ_y [km]	1.180E-03	1.740E-03	1.090	8.370E-01
RMSE: ρ_z [km]	0	0	0	0
RMSE: $\dot{\rho}_x$ [km/s]	9.567E-09	1.117E-08	3.949E-06	3.355E-06
RMSE: $\dot{\rho}_y$ [km/s]	3.750E-09	1.023E-08	4.783E-06	3.897E-06
RMSE: $\dot{\rho}_z$ [km/s]	0	0	0	0

4.4.5 Southern HALO Orbit Example

The next example to consider is a southern HALO type orbit. The initial condition for the chief was set from a periodic L3 southern HALO orbit. The deputy initial condition was set by back-propagation of the chief satellite for a few seconds until a relative separation of around 20 km

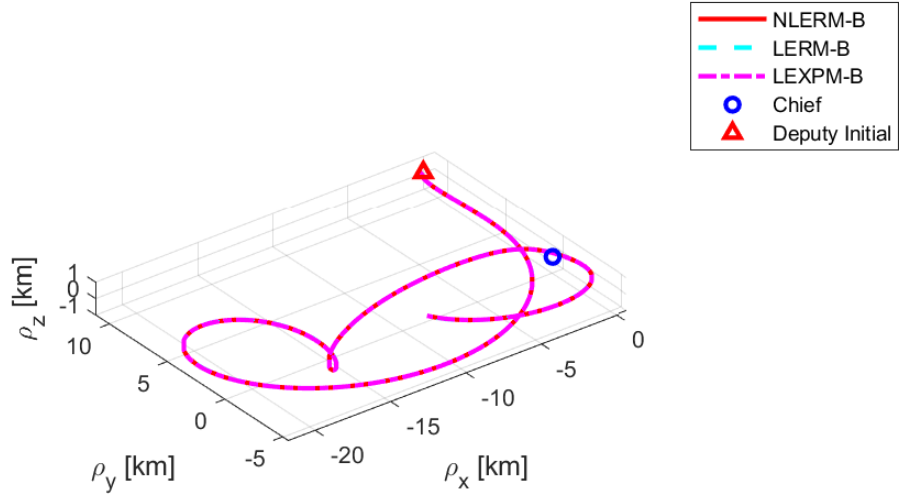


(a) Deputy and chief 3D position ER3BP \mathcal{B} -frame.

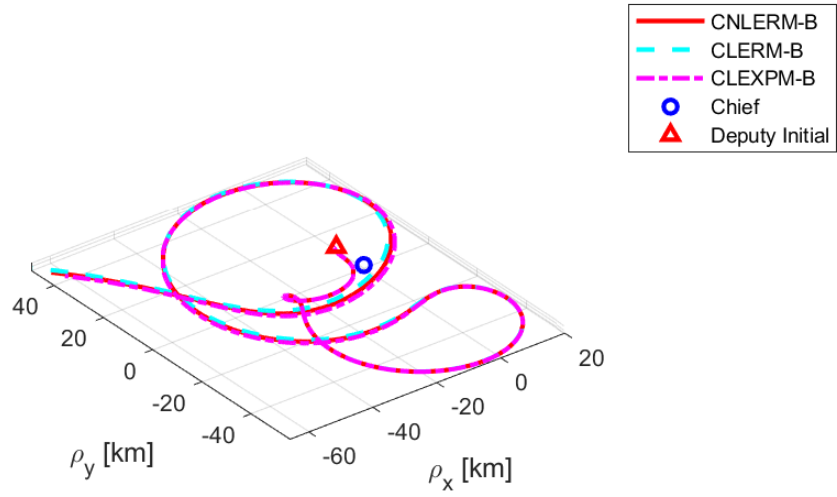


(b) Deputy and chief 3D position in the CR3BP \mathcal{B} -frame.

Figure 4.21 Lyapunov orbit example: deputy and chief 3D positions.

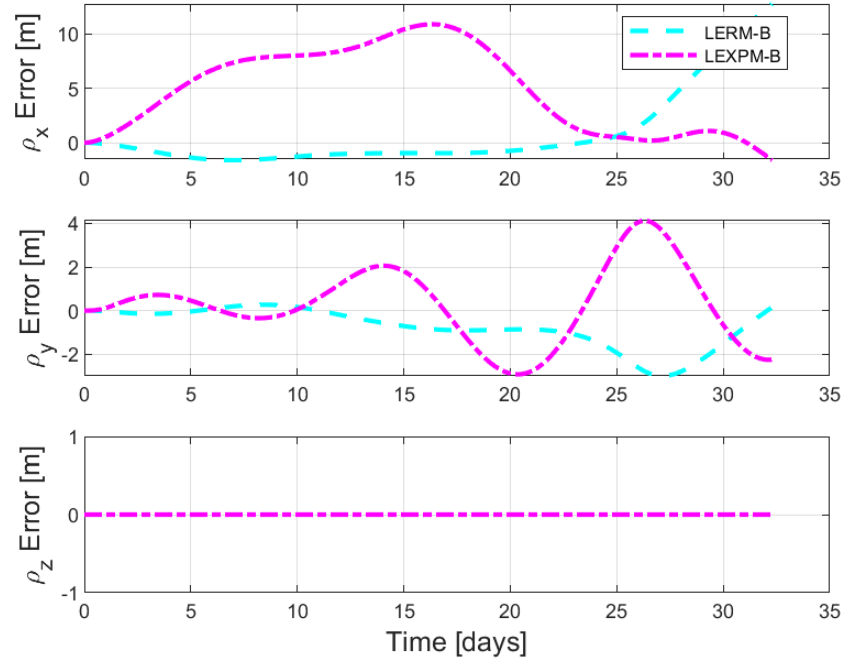


(a) Deputy and chief 3D position in the ER3BP relative C -frame.

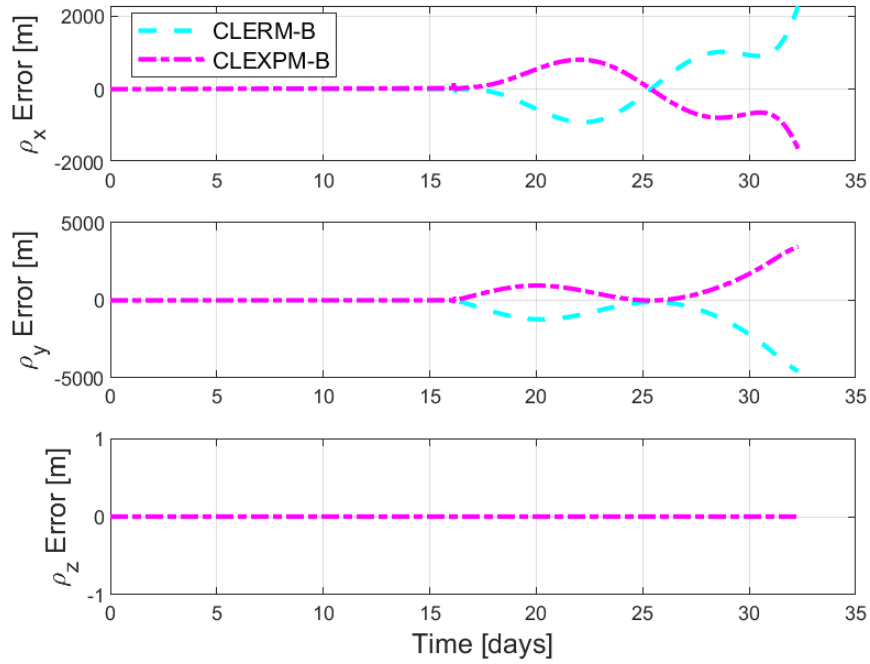


(b) Deputy and chief 3D position in the CR3BP relative C -frame.

Figure 4.22 Lyapunov orbit example: 3D relative motion plots.

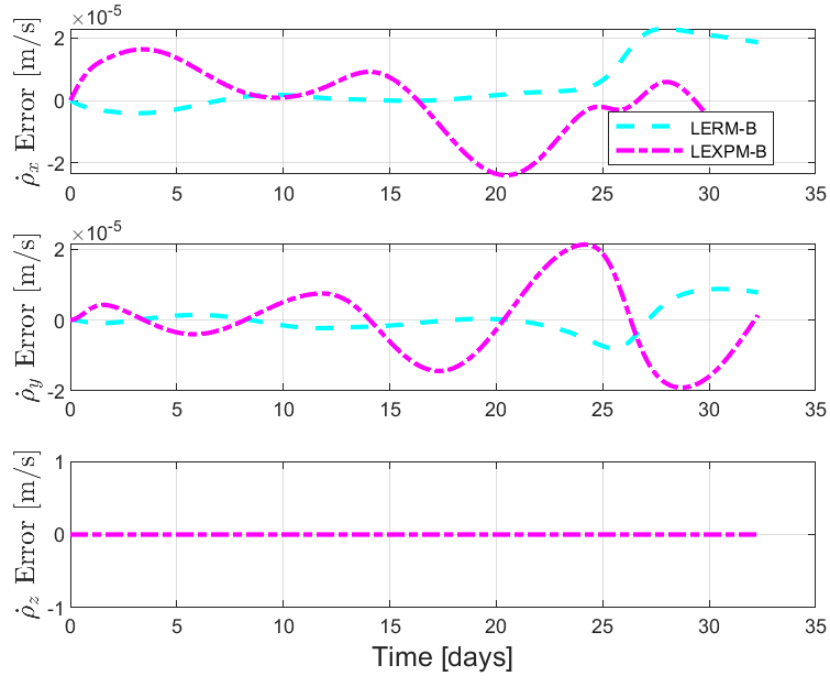


(a) Relative position errors in the ER3BP C-frame.

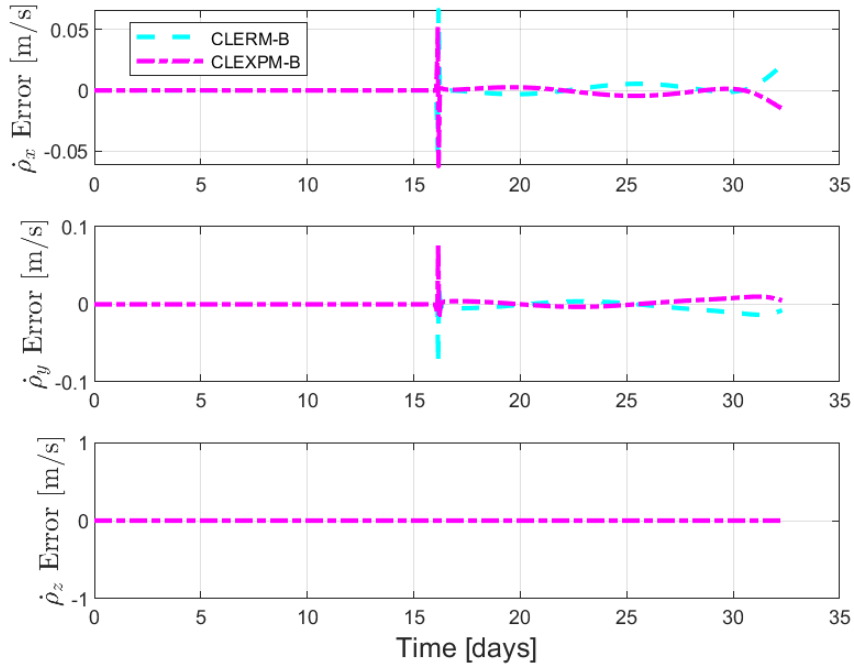


(b) Relative position errors in the CR3BP C-frame.

Figure 4.23 Lyapunov orbit example: relative position errors.

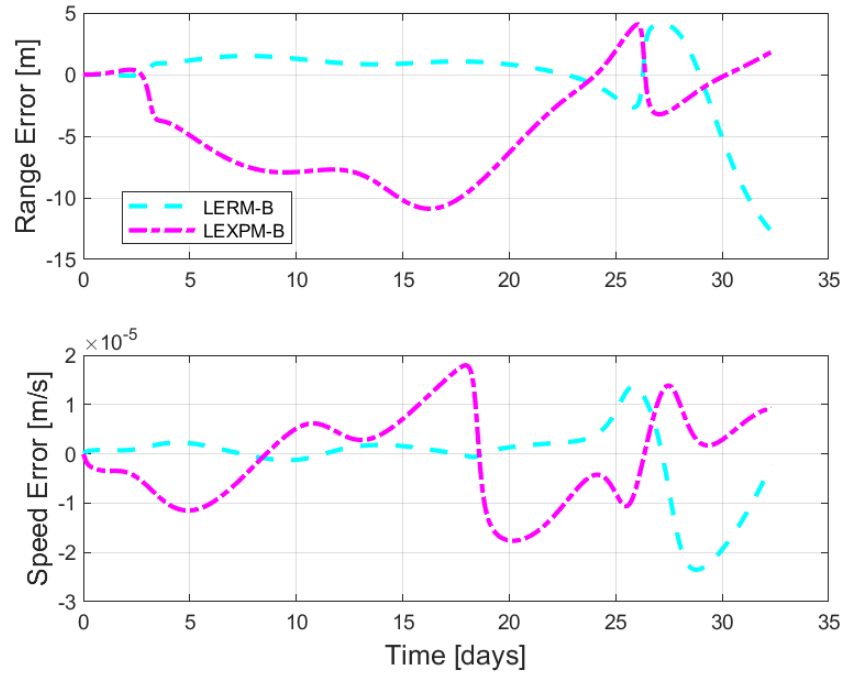


(a) Relative velocity errors in the ER3BP C-frame.

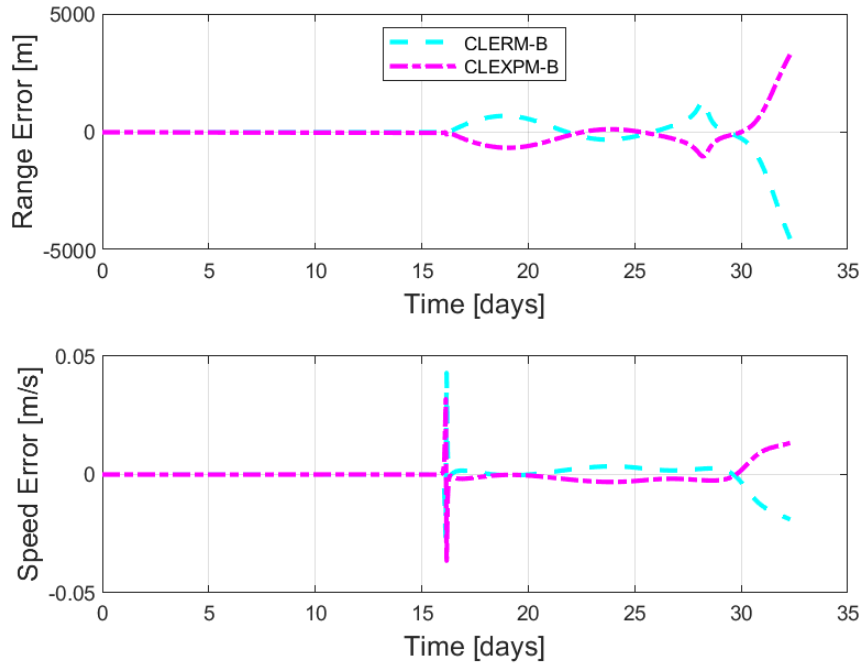


(b) Relative velocity errors in the CR3BP C-frame.

Figure 4.24 Lyapunov orbit example: relative velocity errors.



(a) Relative range and speed errors in the ER3BP C-frame.



(b) Relative range and speed errors in the CR3BPC-frame.

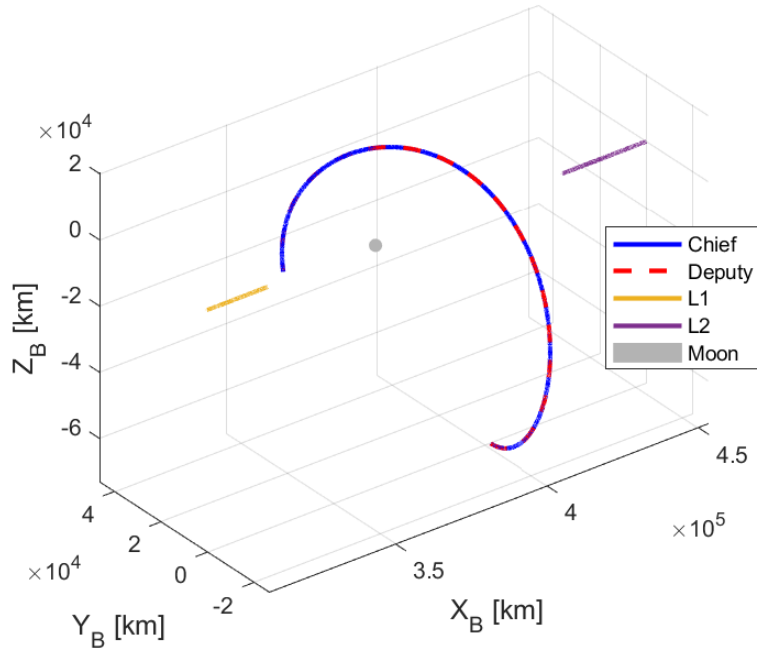
Figure 4.25 Lyapunov orbit example: range and speed errors.

was achieved. The scenario was propagated for one period of the chief satellite in the CR3BP. The deputy and chief initial conditions for both the ER3BP and the CR3BP are given in non-dimensional units in Table 4.9.

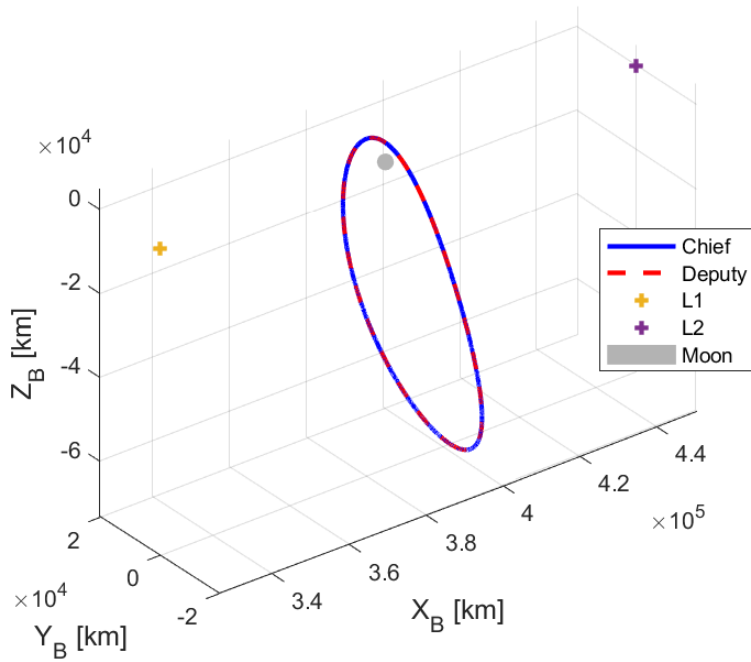
Table 4.9 Southern HALO orbit example deputy and chief initial conditions given in non-dimensional units for both the ER3BP and the CR3BP.

Initial Condition	Chief ER3BP	Deputy ER3BP	Chief CR3BP	Deputy CR3BP
x_0 [DU]	1.031111241	1.031111243	1.031111241	1.031111208
y_0 [DU]	0	-7.97856E-05	0	-7.97856E-05
z_0 [DU]	-0.187722875	-0.18772279	-0.187722875	-0.187722774
\dot{x}_0 [DU/TU]	-6.04188E-15	6.37818E-06	-6.04188E-15	-9.93032E-05
\dot{y}_0 [DU/TU]	-0.122225451	-0.122225464	-0.122225451	-0.122225346
\dot{z}_0 [DU/TU]	-2.30133E-13	0.000259133	-2.30133E-13	0.000309897

Table 4.10 shows the RMS errors for deputy position and velocity components over the scenario for all linear motion models. The table gives a good summary of the relative “closeness” the linear models achieve to the non-linear truth models. Figure 4.26 shows the 3D view of the chief and deputy trajectories in both the ER3BP and the CR3BP. The relative positions of the deputy satellite are shown in Fig. 4.27. The relative position errors for each of the linear propagation models is shown in Fig. 4.28. The errors for the ER3BP peaked at 150 m in any axis; however for the CR3BP, the errors spiked after four days of propagation to much higher levels. As with the Lyapunov scenario, the closer perilune passage in the CR3BP geometry most likely contributes to this increase in error. The relative velocity errors for the propagation models are shown in Fig. 4.29 and follow the trend of the position errors. A spike in error occurs near perilune passage, for velocity just like for position in the CR3BP models. Note that, for the ER3BP linear motion models, the LERM-E and LEXPM-E models followed the truth very closely for about 4 days before an appreciable increase in velocity error (see Fig. 4.29a). The range and speed errors for the scenario are shown in Fig. 4.30 where once again the ER3BP models outperformed the CR3BP models significantly. Note that, for the shorter time-span before perilune passage, both the ER3BP and CR3BP linear models closely mirrored the non-linear models. To increase accuracy, the integration time-step could be decreased proportionally to the distance from perilune passage.

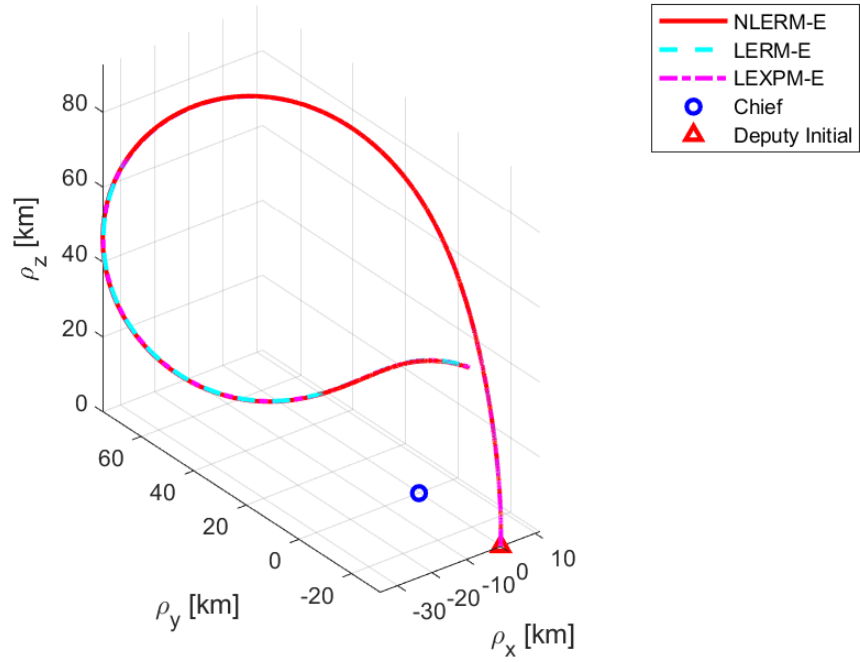


(a) Deputy and chief 3D position ER3BP \mathcal{B} -frame.

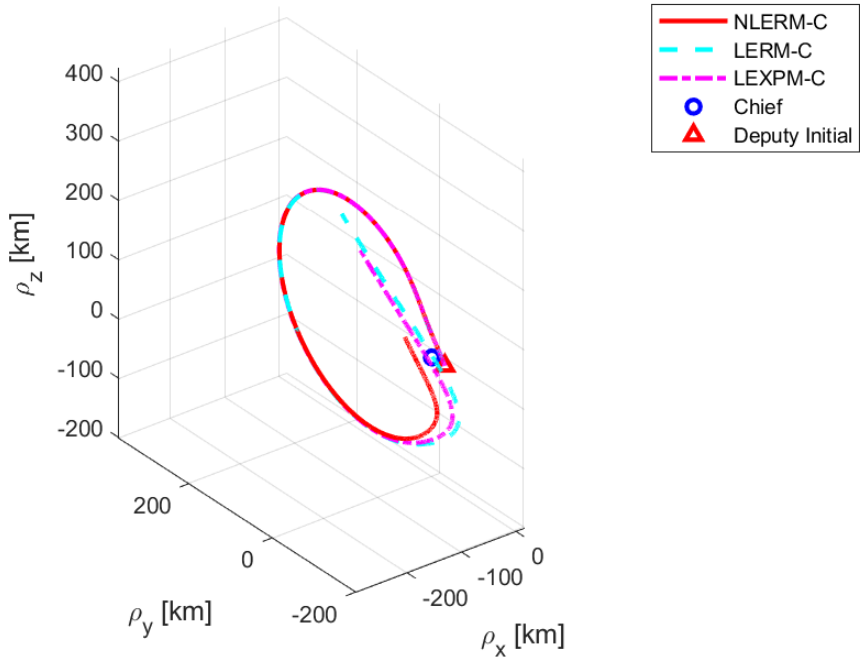


(b) Deputy and chief 3D position in the CR3BP \mathcal{B} -frame.

Figure 4.26 Southern HALO orbit example: deputy and chief 3D positions.

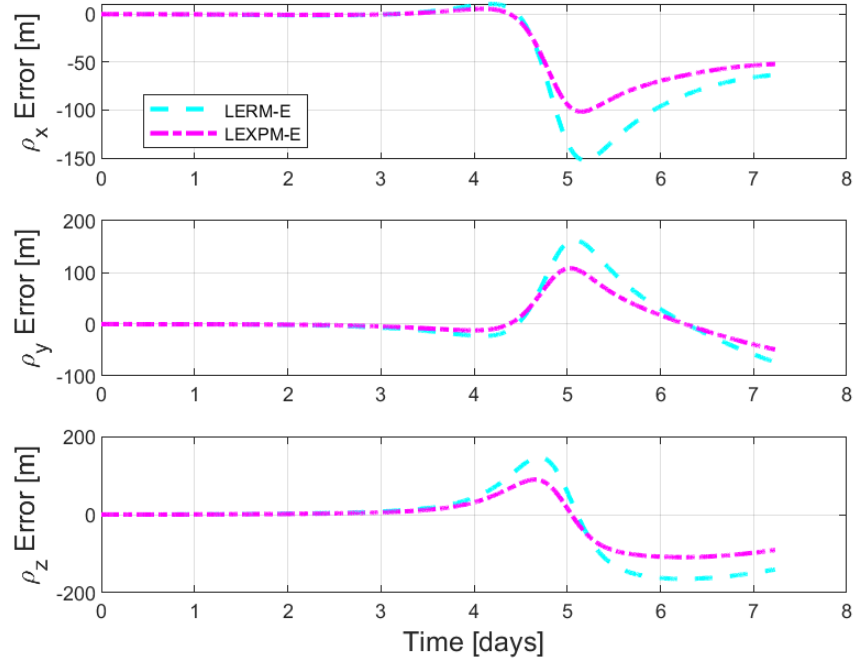


(a) Deputy and chief 3D position in the ER3BP relative C-frame.

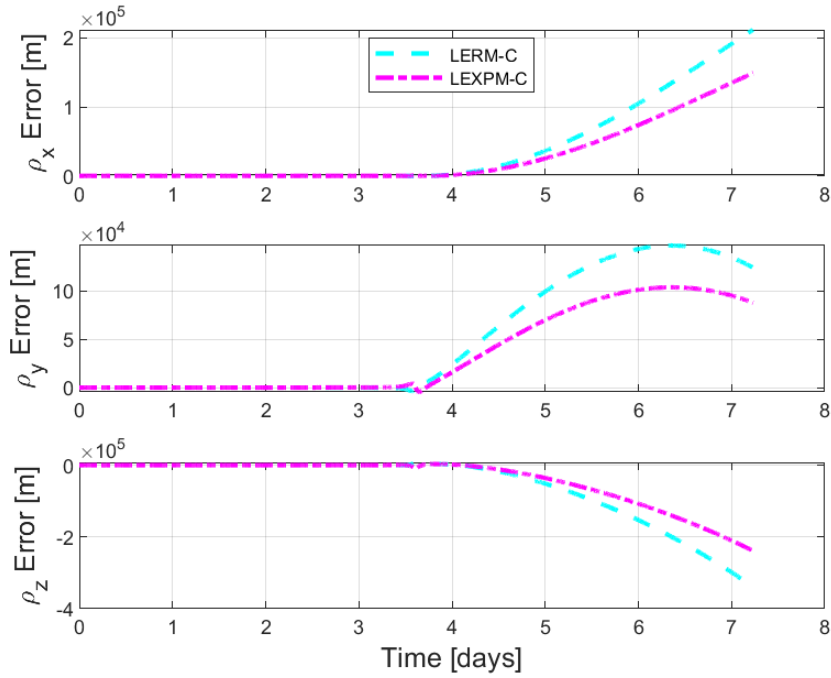


(b) Deputy and chief 3D position in the CR3BP relative C-frame.

Figure 4.27 Southern HALO orbit example: 3D relative motion plots.

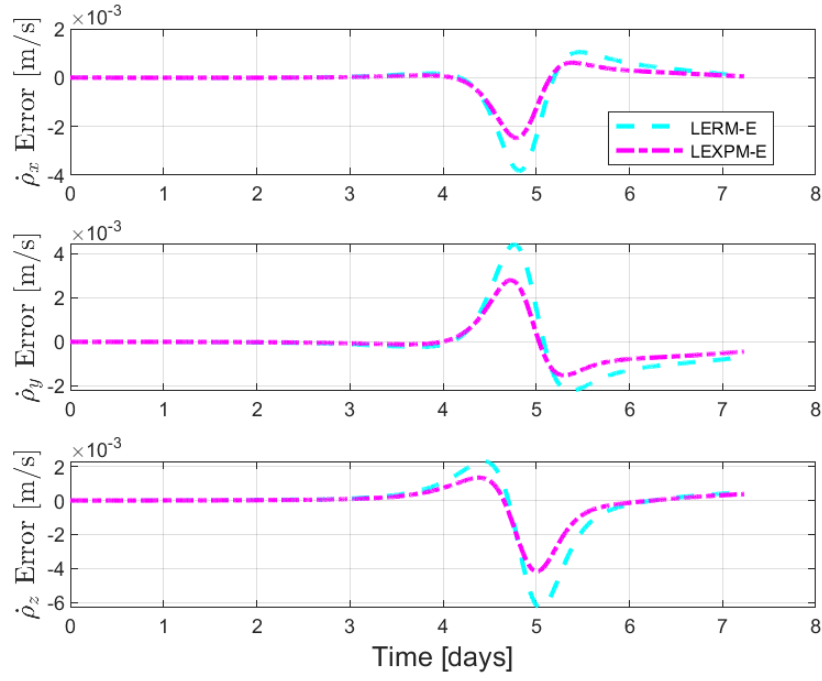


(a) Relative position errors in the ER3BP C-frame.

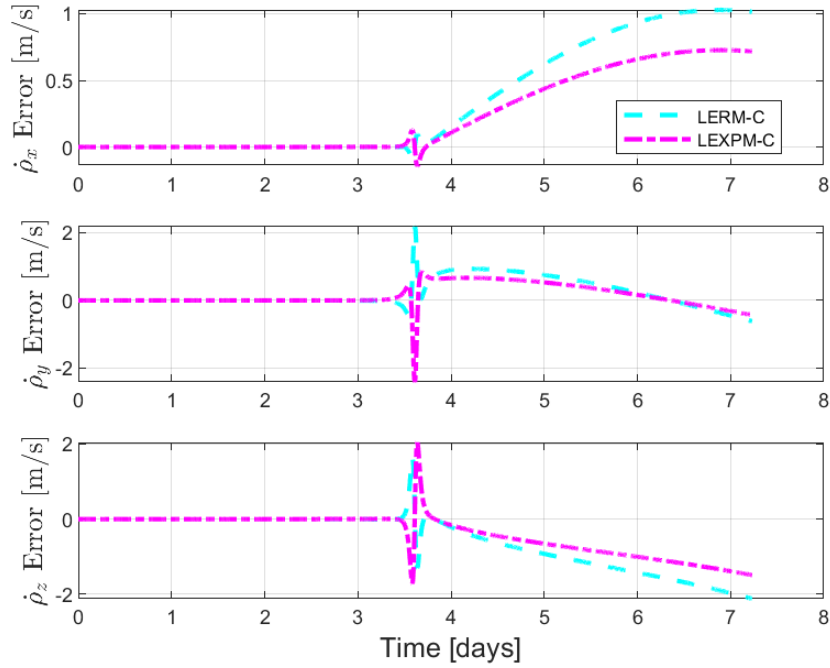


(b) Relative position errors in the CR3BP C-frame.

Figure 4.28 Southern HALO orbit example: relative position errors.

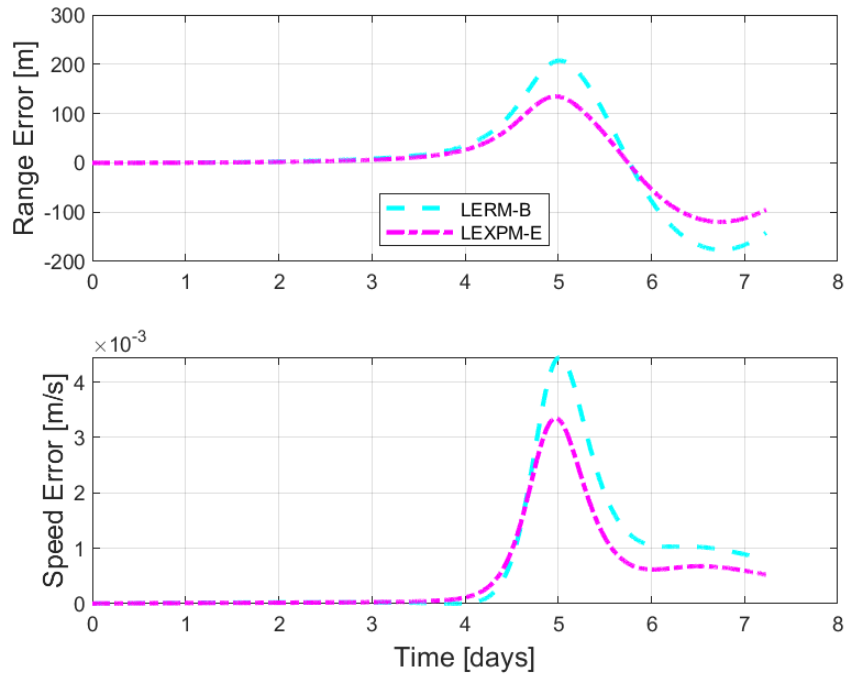


(a) Relative velocity errors in the ER3BP C-frame.

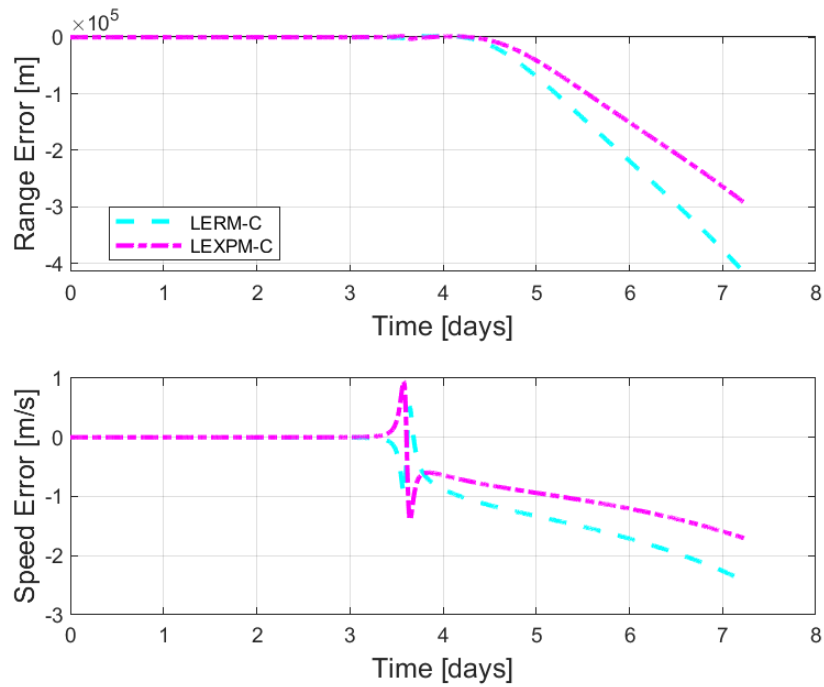


(b) Relative velocity errors in the CR3BP C-frame.

Figure 4.29 Southern HALO orbit example: relative velocity errors.



(a) Relative range and speed errors in the ER3BP *C*-frame.



(b) Relative range and speed errors in the CR3BP *C*-frame.

Figure 4.30 Southern HALO orbit example: range and speed errors.

Table 4.10 Southern HALO example RMS errors for deputy position and velocity components.

	LERM-E	LEXPM-E	LERM-C	LEXPM-C
RMSE: ρ_x [km]	6.084E-02	4.330E-02	7.171E+01	5.046E+01
RMSE: ρ_y [km]	5.133E-02	3.397E-02	7.902E+01	5.568E+01
RMSE: ρ_z [km]	8.873E-02	5.870E-02	1.089E+02	7.656E+01
RMSE: $\dot{\rho}_x$ [km/s]	9.075E-07	5.806E-07	5.340E-04	3.768E-04
RMSE: $\dot{\rho}_y$ [km/s]	1.228E-06	7.930E-07	4.563E-04	3.618E-04
RMSE: $\dot{\rho}_z$ [km/s]	1.585E-06	1.016E-06	8.961E-04	6.539E-04

4.4.6 L2 Vertical Orbit

The final orbital scenario considered is that of an L2 “vertical” family orbit with a period of 27.8 days. As with previous three-body scenarios the periodic family geometry is better preserved in the CR3BP dynamics due to the fact that initial conditions are designed for the CR3BP system. Note that a differential corrector could be used to adjust the initial conditions to achieve quasi-periodic behavior for a similar trajectory. Much research has been performed on transitioning orbital trajectories to models with higher fidelity [82]. The deputy initial condition is set by back propagation of the chief satellite for 10 seconds. The deputy and chief initial conditions are given in non-dimensional units for both the ER3BP and the CR3BP in Table 4.11.

Table 4.11 L2 vertical orbit deputy and chief initial conditions given in non-dimensional units for both the ER3BP and the CR3BP.

Initial Condition	Chief ER3BP	Deputy ER3BP	Chief CR3BP	Deputy CR3BP
x_0 [DU]	1.063021041	1.06302104	1.063021041	1.063021039
y_0 [DU]	0	-4.45129E-05	0	-4.45129E-05
z_0 [DU]	-1.30107E-13	2.1296E-05	-1.30107E-13	2.1296E-05
\dot{x}_0 [DU/TU]	-9.98388E-14	-0.000105936	-9.98388E-14	-0.000139727
\dot{y}_0 [DU/TU]	-1.704759171	-1.704759165	-1.704759171	-1.704759151
\dot{z}_0 [DU/TU]	0.815596256	0.815596254	0.815596256	0.815596248

Table 4.12 shows the RMS errors for the deputy position and velocity components. The orbit geometry of the scenario is shown in Fig. 4.31. The relative orbit geometry of the scenario is shown in Fig. 4.32. The relative position errors for the ER3BP and CR3BP linear models are shown in Fig. 4.33. Note that the ER3BP model errors begin to increase after 10 days, with the LEXPM and LERM models diverging at that point as shown in Fig 4.33a. For the CR3BP, the model error

increases significantly after 20 days of propagation as shown in Fig. 4.33b. The velocity errors for the scenario are shown in Fig. 4.34. The ER3BP linear models outperformed the CR3BP models in terms of maximum error from the truth. The same increase in errors is seen for the ER3BP models around 10 days where the LEXPM-E and LERM-E models separate from one another as shown in Fig. 4.34a. For the CR3BP, a significant increase in velocity error for the linear models LERM-C and LEXPM-C occurs near the end of the scenario. The range and speed errors for the scenario are shown in Fig. 4.35 and show the same error trends as discussed above.

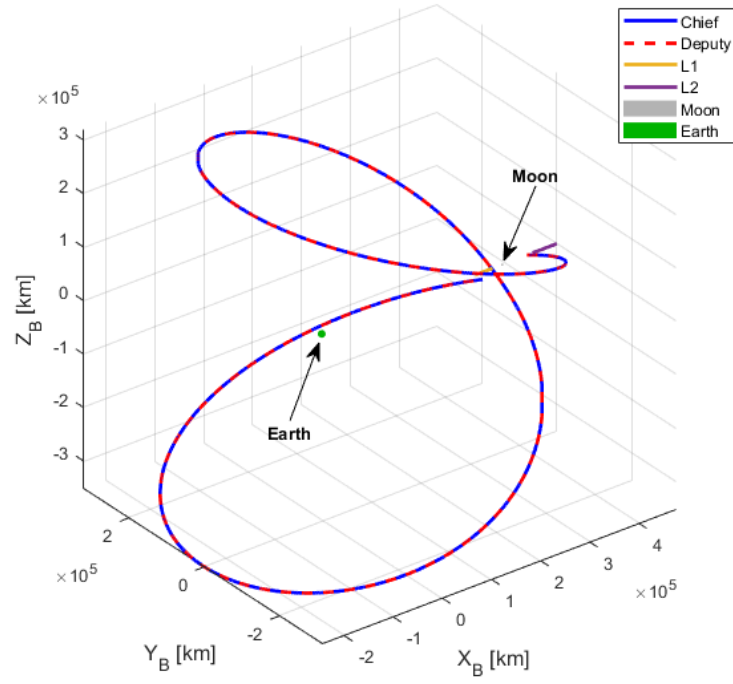
Table 4.12 L2 vertical orbit RMS errors of deputy position and velocity components.

	LERM-E	LEXPM-E	LERM-C	LEXPM-C
RMSE: ρ_x [km]	1.950E-02	4.048E-02	3.906E-01	1.071
RMSE: ρ_y [km]	3.578E-02	5.538E-02	4.218E-01	1.171
RMSE: ρ_z [km]	4.573E-02	1.129E-01	5.532E-01	2.180
RMSE: $\dot{\rho}_x$ [km/s]	1.806E-07	3.273E-07	8.619E-06	1.247E-05
RMSE: $\dot{\rho}_y$ [km/s]	1.718E-07	1.734E-07	7.220E-06	8.353E-06
RMSE: $\dot{\rho}_z$ [km/s]	7.197E-08	2.952E-07	8.855E-06	9.502E-06

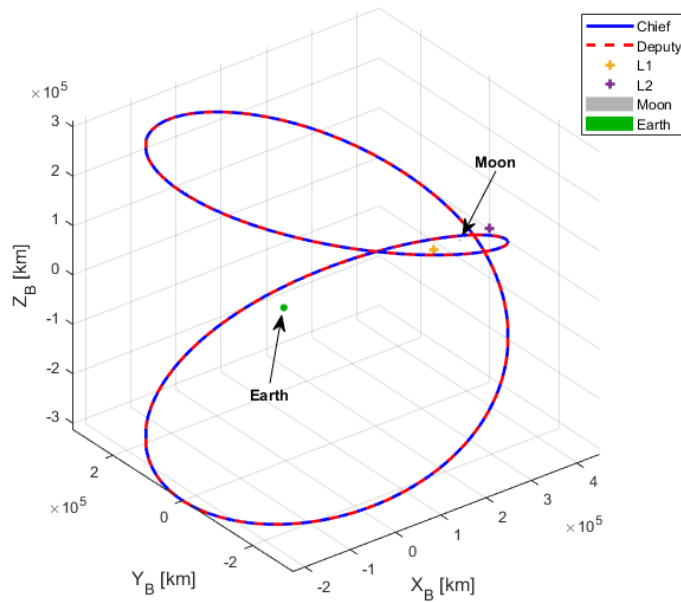
4.5 Stability of the Linear Motion Models

For an LTI system, stability analysis is easily done by inspection of the system eigenvalues [47]. However, for linear time-varying (LTV) systems, eigenvalue analysis does not provide an easy snapshot into system stability. Some information can be garnered about system stability by inspection of the magnitude of the LTV system eigenvalues as time advances. The STM is computed using the system $A(t)$ matrix (note that the system STM satisfies the equation $\dot{\Phi} = A(t)\Phi$) with initial conditions for the STM set by $\Phi_0 = I_{6 \times 6}$, where I is the identity matrix.

Once the STM is formed, it should be computed and propagated out to the period of the system (in this case the orbital period of the chief satellite). The matrix that results is known as the monodromy matrix, the eigenvalues of which should be inside the unit circle for stability of the system. For the L2 vertical orbit example, Figs. 4.36 - 4.38 show the magnitude of the six system eigenvalues for the duration of the scenario. The system eigenvalues are all of very small magnitudes (1×10^{-6}) which ensures that all eigenvalues stay within the unit circle. Future work should investigate the monodromy matrix properties for a wide variety of three-body orbits.

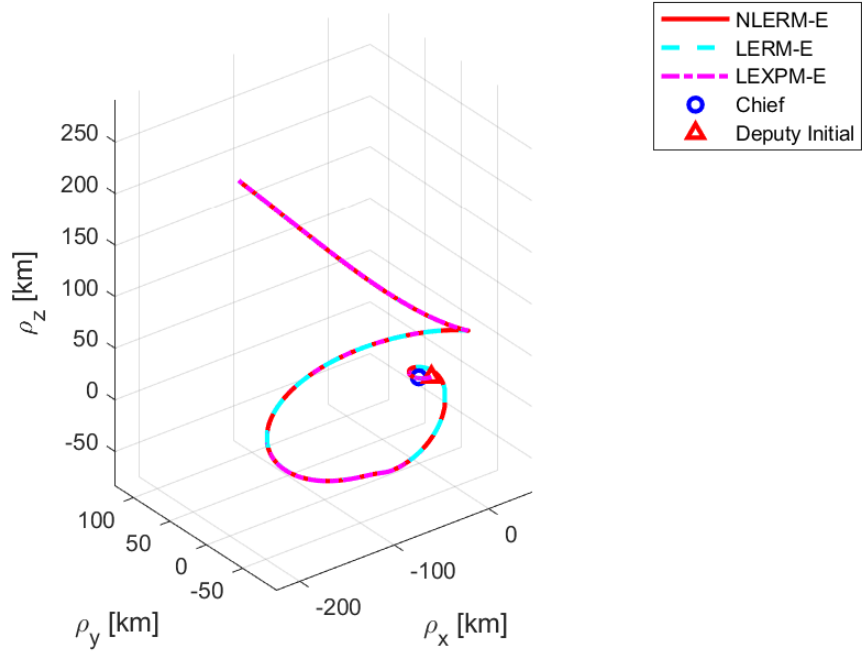


(a) Deputy and chief 3D position ER3BP \mathcal{B} -frame.

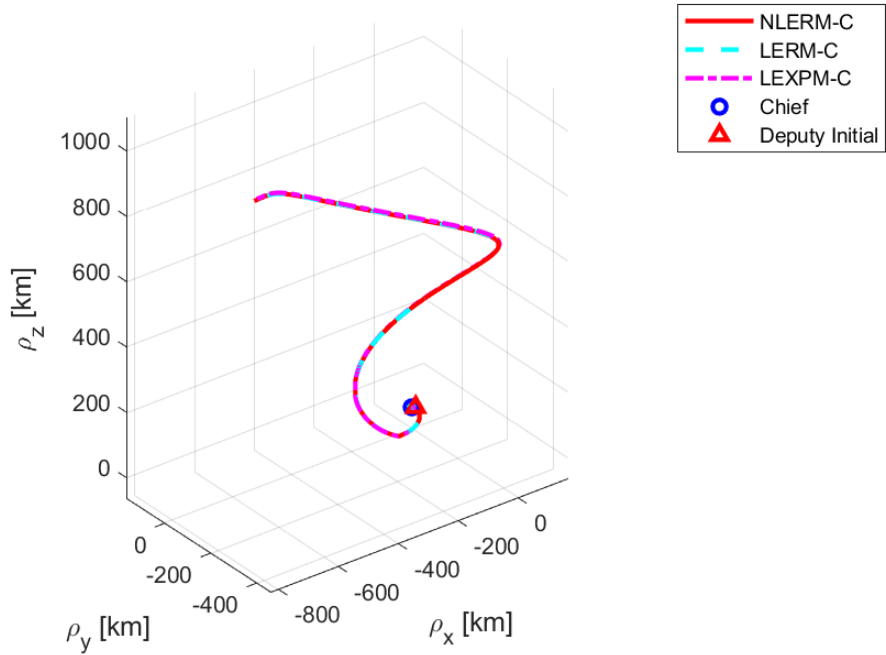


(b) Deputy and chief 3D position in the CR3BP \mathcal{B} -frame.

Figure 4.31 L2 vertical orbit example: deputy and chief 3D positions.

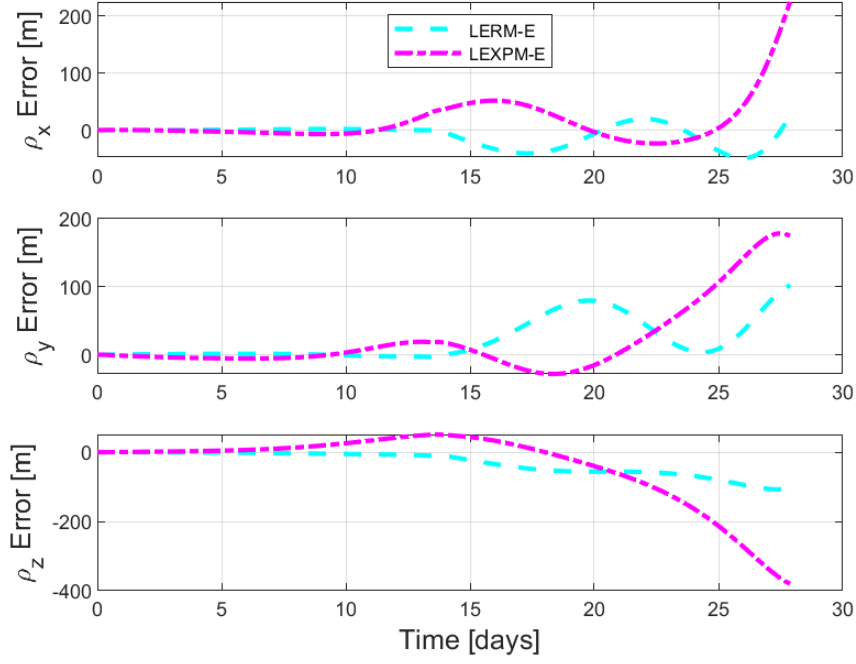


(a) Deputy and chief 3D position in the ER3BP relative C-frame.

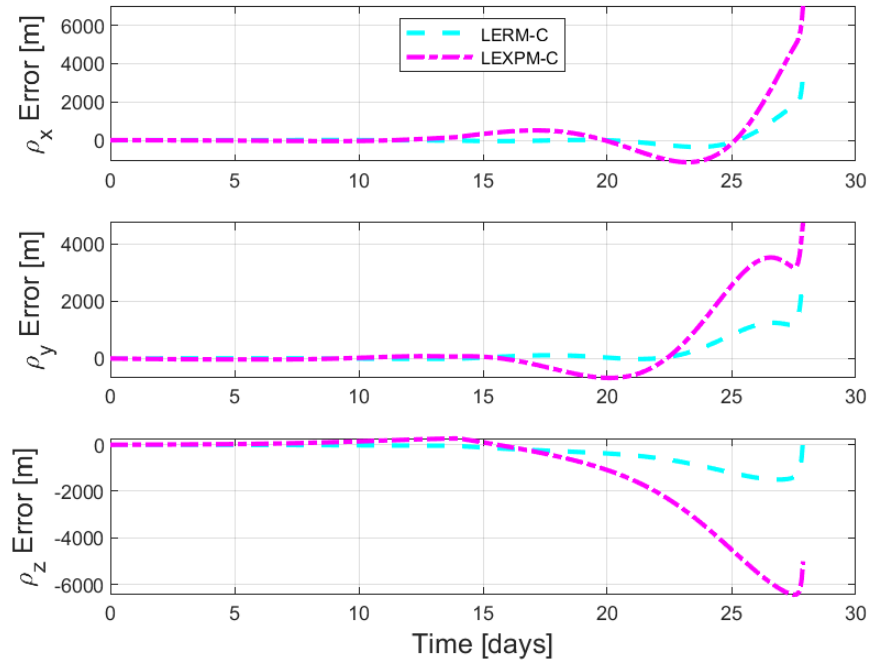


(b) Deputy and chief 3D position in the CR3BP relative C-frame.

Figure 4.32 L2 vertical orbit example: 3D relative motion plots.

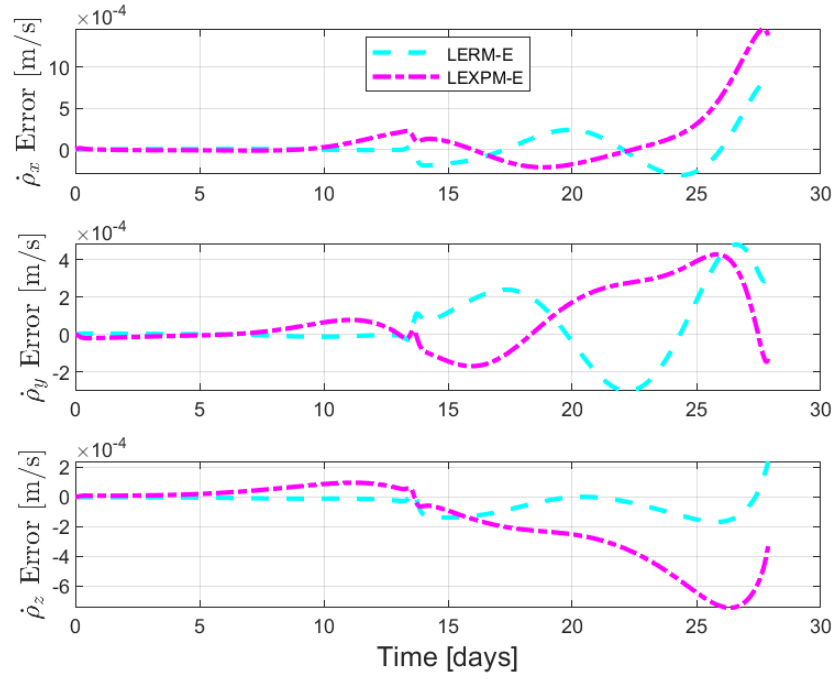


(a) Relative position errors in the ER3BP C-frame.

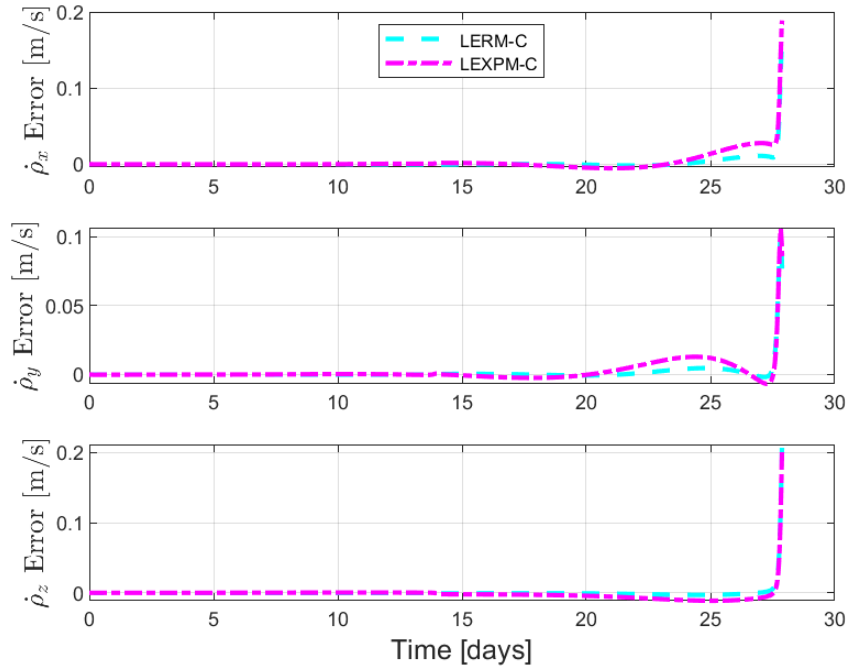


(b) Relative position errors in the CR3BP C-frame.

Figure 4.33 L2 vertical orbit example: relative position errors.

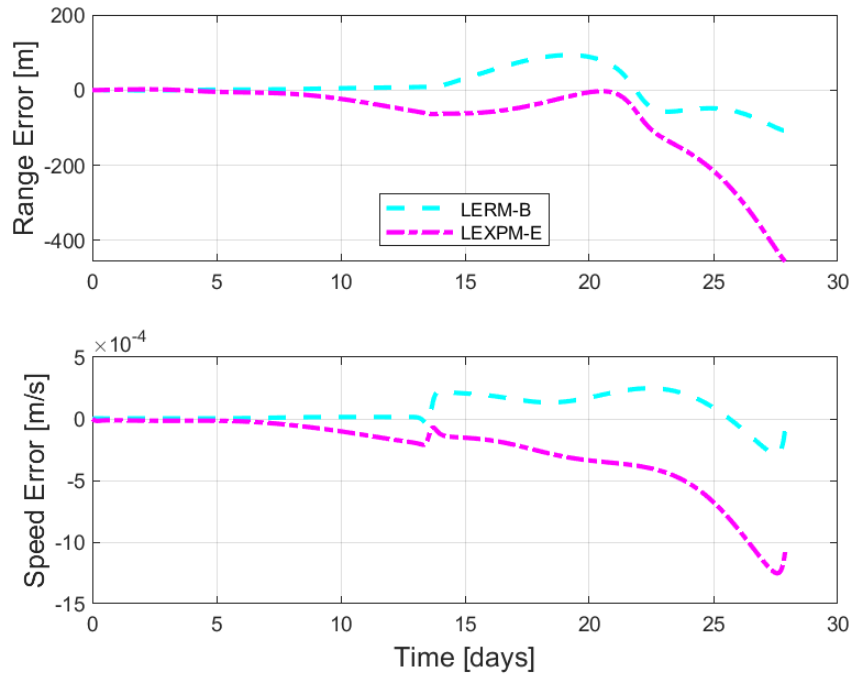


(a) Relative velocity errors in the ER3BP C-frame.

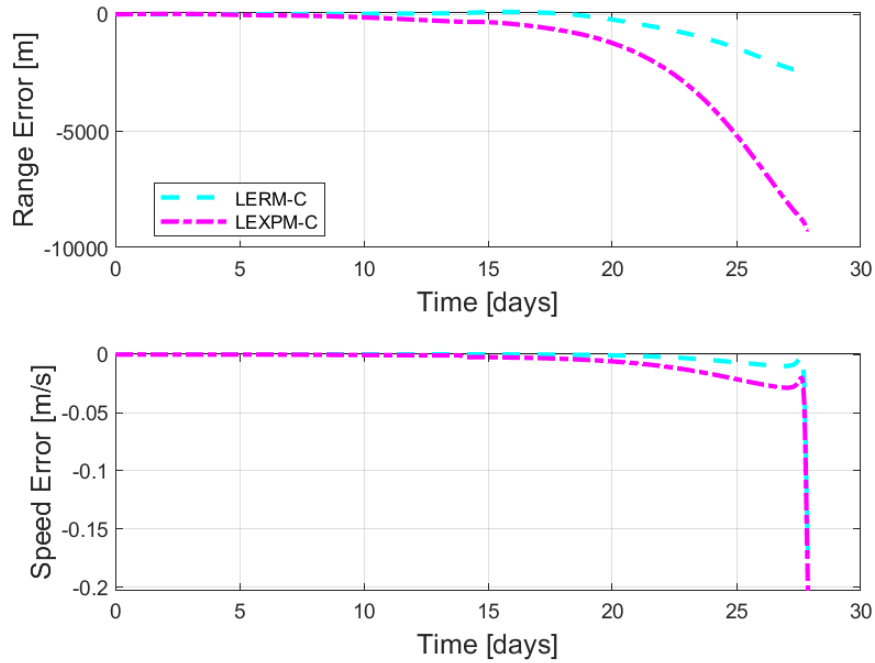


(b) Relative velocity errors in the CR3BP C-frame.

Figure 4.34 L2 vertical orbit example: relative velocity errors.



(a) Relative range and speed errors in the ER3BP C -frame.



(b) Relative range and speed errors in the CR3BP C -frame.

Figure 4.35 L2 vertical orbit example: range and speed errors.

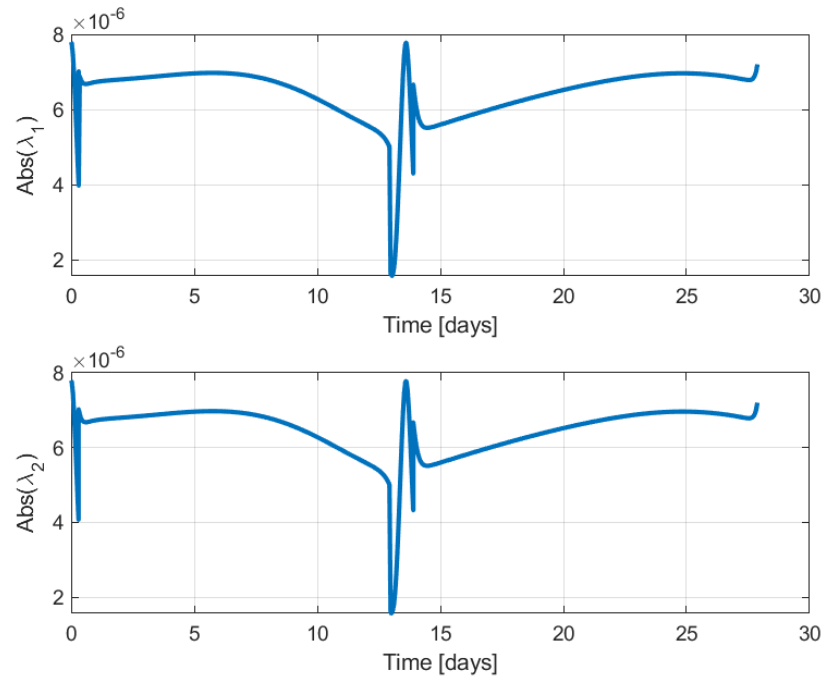


Figure 4.36 L2 vertical orbit magnitude of first two system eigenvalues.

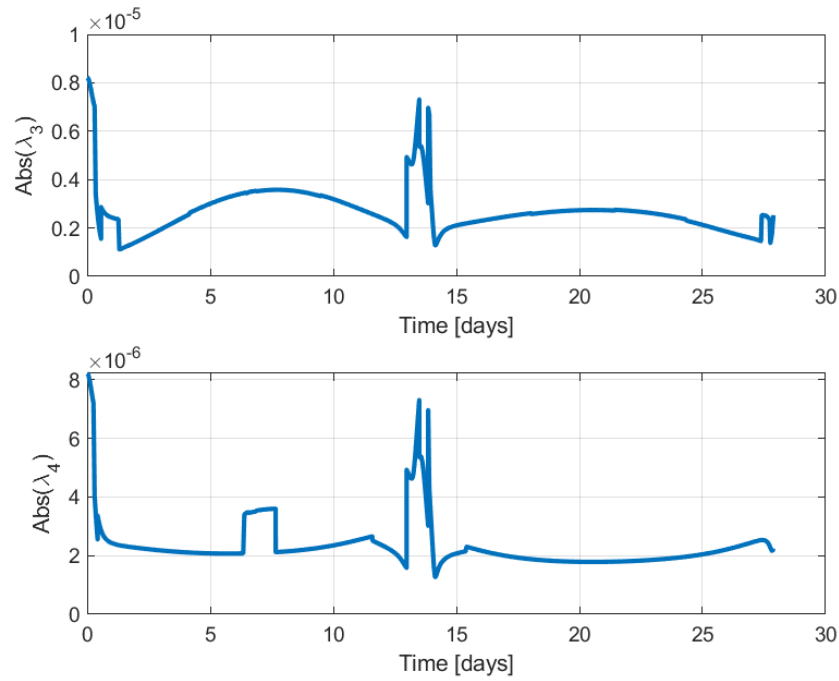


Figure 4.37 L2 vertical orbit magnitude of third and fourth system eigenvalues.

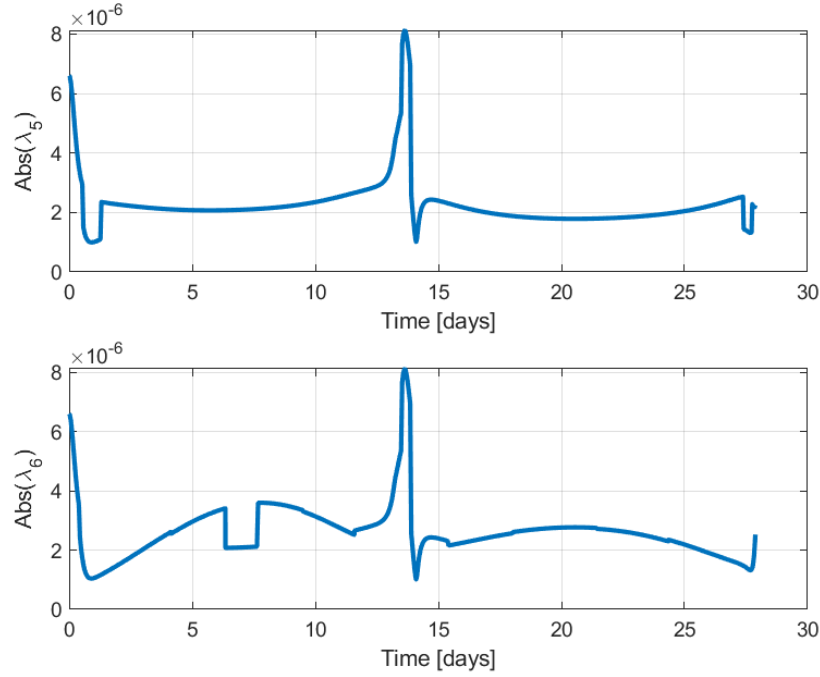


Figure 4.38 L2 vertical orbit magnitude of fifth and sixth system eigenvalues.

4.6 State-Transition Matrix

The next logical development in forming analytical solutions for the LER3BP involves computing a state-transition matrix (STM). Given the time-varying nature of the LERM and LEXPM EOMs, a new value for the system $A(t)$ matrix will need to be computed for each time-step that the equations are propagated for. The state-transition solution assumes a solution of the form:

$$\mathbf{X}(t) = \Phi(t, t_0)\mathbf{X}(t_0) \quad (4.120)$$

$$\dot{\Phi}(t) = A(t)\Phi(t, t_0) \quad (4.121)$$

$$\Phi(t_0, t_0) = I_{n \times n} \quad (4.122)$$

The initial condition for the STM is the identity matrix. In order to calculate the STM at each time step, the chief position vector in the synodic frame is needed. The state-transition matrix for the non-relative ER3BP has been studied and is used for differential correction schemes to generate periodic orbit solutions for higher fidelity ephemeris models [32, 35]. The matrix differential

equation given in Eq. (4.121) must be computed numerically while simultaneously integrating the equations of motion for the deputy and chief satellites. This corresponds to a total of 48 first-order differential equations to be simultaneously numerically integrated to form the elements of the STM. Once the STM is computed along a trajectory, then the time-varying elements of $\Phi(t, t_0)$ can be used to move the deputy relative states from any initial time to any point along the trajectory at time t by the matrix multiplication shown by Eq. (4.120). The STM for relative motion will be used in the next chapter for an application of differential corrections for finding periodic orbit solutions in the relative frame.

4.7 Linearized Relative Motion Summary

A semi-analytical solution for spacecraft relative motion in the ER3BP and CR3BP dynamics models was developed using a discretized matrix exponential. The linearized state-matrices were also developed and can be integrated as a set of linear differential equations (LERM motion models). The linear motion models were shown to maintain accuracy to acceptable levels in three-body orbits for relatively long periods of time, and for shorter periods of time in low orbits such as LEO and LLO. The linear state-matrices will be used in the following chapter to develop two applications involving relative motion.

5 Applications of Linear Relative Motion for the Three-Body Problem

Applications of the linearized relative motion models developed in Chapter 4 will be discussed in this chapter. The first application considered is using the linearized state-matrix and STM to find closed relative periodic orbits. The second application considered is the problem of initial relative orbit determination (IROD) for the three-body problem. Using the linearized LEXPM-C motion model, a method of finding an initial relative orbit estimate from a set of three LOS measurements is proposed.

5.1 Natural Closed Relative Motion

One application of the linear motion models developed in Chapter 4 lies in differential corrections. Relative motion trajectories are of great interest in general, and this research investigates the problem of finding closed relative natural motion trajectories in the CR3BP. Natural motion trajectories are of great interest since they exploit the dynamics of the system to the fullest. To that end, the problem of finding closed relative trajectories in the CR3BP is studied.

This research proposes to use a single-shooting differential correction method for finding closed relative solutions in the CR3BP. The natural dynamics of the CR3BP admit multiple families of periodic orbits, including HALO, NRHO, Lyapunov, long-period, short-period, and axial orbit families. Periodic orbits exhibit certain features including Jacobi constants and stability indices that help define them uniquely. A deputy satellite initial condition will be seeded near a given periodic orbit. A single shooting differential corrector then adjusts the deputy initial conditions such that the deputy returns to the initial position to a given tolerance, thus forming a closed path in the relative frame. An initial periodic reference orbit will be assumed to be known for the chief satellite.

Single and multiple shooting methods are a common way to establish initial conditions for periodic orbits in the CR3BP [42, 83]. A similar approach is followed for establishing closed relative trajectories. The relative orbital dynamics of the CR3BP are utilized for the search space rather than the chief's dynamics. A single-shooting differential corrector is applied to an initial guess on a deputy state and iterated upon until a set maximum number of iterations, or the convergence criteria of the deputy returning to its initial position is met to a given tolerance.

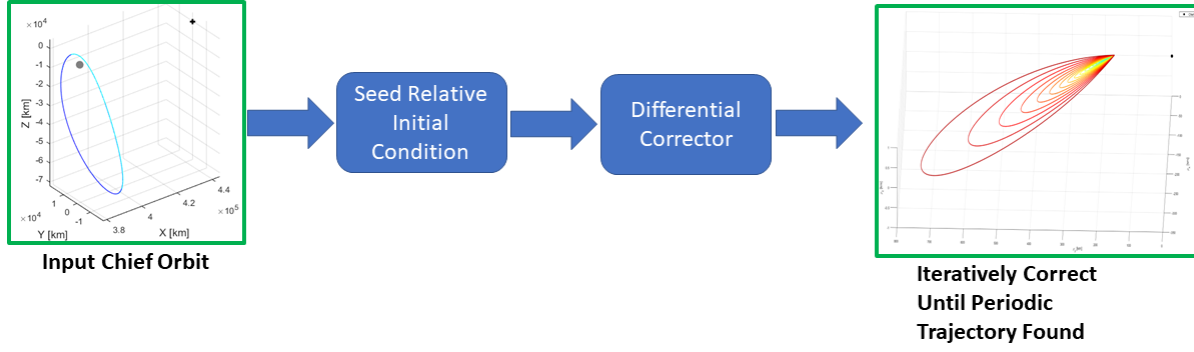


Figure 5.1 Differential correction process for finding closed relative trajectories.

5.1.1 Differential Correction for Relative Motion

Differential correction offers an iterative approach to alter a set of design variables to satisfy a set of given constraints. This approach is often used for the CR3BP to generate periodic orbit solutions either by single-shooting or multiple shooting algorithms [43, 68, 84]. This research employs a single-shooting algorithm to generate a closed relative trajectory.

The first step in building a differential corrector is to define a set of m constraint variables for the problem's success criteria. The Euclidean norm of the constraint vector is typically evaluated to test for convergence. Let the constraint vector be defined as $\mathbf{F}(\mathbf{x})$ which is a vector function of the n design states (\mathbf{x}^k). The constraint is defined such that the initial relative position should be equal to the final relative position of the deputy. Thus $\mathbf{F}(\mathbf{x})$ is defined as the final relative position of the deputy ($\rho_B(t_0 + T)$) minus the initial relative position ($\rho_B(t_0)$).

$$\mathbf{F}(\mathbf{x}^k) = \mathbf{0} \quad (5.1)$$

$$\mathbf{F}(\mathbf{x}^k) = \begin{bmatrix} \rho_x(t_0 + T) - \rho_x(t_0) \\ \rho_y(t_0 + T) - \rho_y(t_0) \\ \rho_z(t_0 + T) - \rho_z(t_0) \end{bmatrix} \quad (5.2)$$

The magnitude of the constraint vector $\epsilon = \|\mathbf{F}(\mathbf{x}^k)\|$ will be used to define convergence. Casting the differential correction process as an optimization problem, the objective becomes finding \mathbf{x}^{k*} such that $\mathbf{F}(\mathbf{x}^{k*}) = \mathbf{0}$. Furthermore, the constraint equation can be written as a Taylor series expansion

about the initial condition for the states.

$$\mathbf{0} = \mathbf{F}(\mathbf{x}) = \mathbf{F}(\mathbf{x}_0) + \left[\frac{\partial \mathbf{F}(\mathbf{x}_0)}{\partial \mathbf{x}_0} \right] (\mathbf{x} - \mathbf{x}_0) + \frac{1}{2!} (\mathbf{x} - \mathbf{x}_0)^T \left[\frac{\partial^2 \mathbf{F}(\mathbf{x}_0)}{\partial \mathbf{x}_0^2} \right] (\mathbf{x} - \mathbf{x}_0) + \dots \quad (5.3)$$

Keeping only the linear terms in the expansion simplifies the expression greatly and allows the expression to be written in an iterative update form.

$$\mathbf{0} = \mathbf{F}(\mathbf{x}^k) + \left[\frac{\partial \mathbf{F}(\mathbf{x}^k)}{\partial \mathbf{x}_0} \right] (\mathbf{x}^{k+1} - \mathbf{x}^k) \quad (5.4)$$

The iterative update form is found by solving Eq. (5.4) for the desired state at the next iteration \mathbf{x}^{k+1} . However, in order to solve for the iterative update form, the Jacobian matrix of $\mathbf{F}(\mathbf{x})$ must be invertible, which is the case if the Jacobian is square. If $\mathbf{F}(\mathbf{x})$ is not a square matrix, a pseudo inverse must be used instead (note that a square Jacobian matrix only results when the number of constraints is equal to the number of design variables (i.e. $m = n$). The general iterative update form for a non-square Jacobian is given by:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \left[\frac{\partial \mathbf{F}(\mathbf{x}^k)}{\partial \mathbf{x}_0} \right]^T \left(\left[\frac{\partial \mathbf{F}(\mathbf{x}^k)}{\partial \mathbf{x}_0} \right] \left[\frac{\partial \mathbf{F}(\mathbf{x}^k)}{\partial \mathbf{x}_0} \right]^T \right)^{-1} \mathbf{F}(\mathbf{x}^k) \quad (5.5)$$

Given an initial guess for the design variables, iteration on the design variables using Eq. (5.5) continues until a desired tolerance is met (or a maximum number of iterations has been reached). The tolerance check is performed by seeking the minimum norm solution that satisfies the constraint vector [43].

$$\|\mathbf{F}(\mathbf{x}^{k+1})\| < \tau \quad (5.6)$$

For the problem of finding a closed relative trajectory in CR3BP dynamics, let the design variables be set equal to the deputy initial position and velocity:

$$\mathbf{x}(t_0) = \begin{bmatrix} \rho_x(t_0) & \rho_y(t_0) & \rho_z(t_0) & \dot{\rho}_x(t_0) & \dot{\rho}_y(t_0) & \dot{\rho}_z(t_0) \end{bmatrix}^T \quad (5.7)$$

where t_0 specifies the initial time. The constraint vector defined as in Eq. (5.2) is evaluated by propagation of the deputy states forward from time $t_0 \rightarrow T$ using the NLERM-C equations (Eqs. (4.78) - (4.80)), where the final deputy position is given by $\boldsymbol{\rho}_{\mathcal{B}}(t_0 + T)$. Once the states are propagated forward to the final time, the constraint is checked for convergence using Eq. (5.6). If not converged the design states are updated using Eq. (5.5) and the process begins again until the desired convergence tolerance τ is met.

In order to perform the update to the design variables, the elements of the Jacobian matrix $\left[\frac{\partial \mathbf{F}(\mathbf{x})}{\partial \mathbf{x}_0} \right]$ are required. The Jacobian gives the relationship between the states at time t_k to the states at the initial time t_0 , which by definition is the STM for the relative CR3BP denoted as: Φ_c . Note that, since only the initial position is part of the constraint vector, only the top three rows of the STM are required for the update equation. The Jacobian matrix terms are shown in Eqs. (5.8) and (5.9).

$$\left[\frac{\partial \mathbf{F}(\mathbf{X})}{\partial \mathbf{x}_0} \right] = \begin{bmatrix} \frac{\partial \rho_x}{\partial \rho_x(t_0)} & \frac{\partial \rho_x}{\partial \rho_y(t_0)} & \frac{\partial \rho_x}{\partial \rho_z(t_0)} & \frac{\partial \rho_x}{\partial \dot{\rho}_x(t_0)} & \frac{\partial \rho_x}{\partial \dot{\rho}_y(t_0)} & \frac{\partial \rho_x}{\partial \dot{\rho}_z(t_0)} \\ \frac{\partial \rho_y}{\partial \rho_x(t_0)} & \frac{\partial \rho_y}{\partial \rho_y(t_0)} & \frac{\partial \rho_y}{\partial \rho_z(t_0)} & \frac{\partial \rho_y}{\partial \dot{\rho}_x(t_0)} & \frac{\partial \rho_y}{\partial \dot{\rho}_y(t_0)} & \frac{\partial \rho_y}{\partial \dot{\rho}_z(t_0)} \\ \frac{\partial \rho_z}{\partial \rho_x(t_0)} & \frac{\partial \rho_z}{\partial \rho_y(t_0)} & \frac{\partial \rho_z}{\partial \rho_z(t_0)} & \frac{\partial \rho_z}{\partial \dot{\rho}_x(t_0)} & \frac{\partial \rho_z}{\partial \dot{\rho}_y(t_0)} & \frac{\partial \rho_z}{\partial \dot{\rho}_z(t_0)} \end{bmatrix}_{|t=t_0+T} \quad (5.8)$$

$$\left[\frac{\partial \mathbf{F}(\mathbf{X})}{\partial \mathbf{x}_0} \right] = \begin{bmatrix} \phi_{11} & \phi_{12} & \phi_{13} & \phi_{14} & \phi_{15} & \phi_{16} \\ \phi_{21} & \phi_{22} & \phi_{23} & \phi_{24} & \phi_{25} & \phi_{26} \\ \phi_{31} & \phi_{32} & \phi_{33} & \phi_{34} & \phi_{35} & \phi_{36} \end{bmatrix}_{|t=t_0+T} \quad (5.9)$$

The variables ϕ_{ij} represent individual elements of the STM evaluated at the final time $t = t_0 + T$.

The STM is propagated with the matrix differential equation:

$$\dot{\Phi}(t, t_0) = A(t)\Phi_c(t, t_0) \quad (5.10)$$

$$\Phi(t_0, t_0) = I_{6 \times 6} \quad (5.11)$$

where the initial condition for $\Phi(t_0, t_0)$ is the identity matrix and the system matrix $A(t)$ is the Jacobian matrix of the CR3BP relative state-vector evaluated at the current time, as developed in Section 4.3.2 [43].

5.1.2 Differential Correction Algorithm for Closed Relative Orbits

With all appropriate terms and equations defined, the differential correction algorithm can now be summarized. The process begins by initializing the design variables \mathbf{x}^k . There are a number of ways to initialize the design variable \mathbf{x}^k (deputy initial position and velocity). One method used by this research is to back-propagate the chief initial condition a short amount of time, and then take the numerical difference between the back-propagated states relative to the chief initial condition (IC) as the deputy starting point. Note that a random seed in position and velocity from the chief's initial position and velocity could also be used. Once initialized, the differential corrector begins a loop to iteratively correct the deputy initial states until either convergence criteria are met, or the maximum number of iterations have passed. The convergence tolerance was chosen to be $\tau = 1 \times 10^{-10}$. A visual depiction of the single-shooting process is provided in Fig. 5.2 and an algorithmic psuedo-code is shown in Alg. 8.

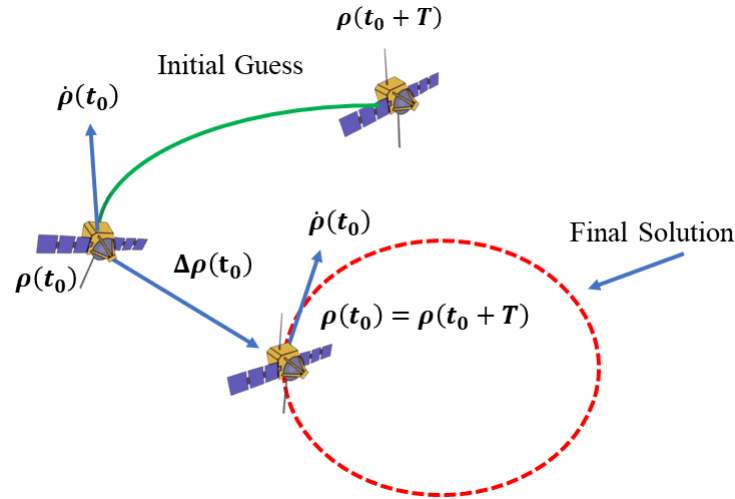


Figure 5.2 Single-shooting algorithm diagram. An initial deputy position and velocity are updated in an attempt to find a final closed trajectory (shown in red).

Algorithm 8 Differential Correction Algorithm for Closed Relative Orbits

```
1: Initialize design variable  $\mathbf{x}^k$ 
2: while  $\|\epsilon\| > \tau$  AND  $k < \text{max iterations}$  do
3:   Propagate relative states and STM to time  $t = T$ 
4:   Evaluate constraint vector  $\mathbf{F}(\mathbf{x})$  (Eq. (5.2))
5:   Check for convergence
6:   if  $\|\mathbf{F}(\mathbf{x}^{k+1})\| = \|\epsilon\| > \tau$  then
7:     Update design variables
8:     
$$\mathbf{x}^{k+1} = \mathbf{x}^k - \left[ \frac{\partial \mathbf{F}(\mathbf{x}^k)}{\partial \mathbf{x}_0} \right]^T \left( \left[ \frac{\partial \mathbf{F}(\mathbf{x}^k)}{\partial \mathbf{x}_0} \right] \left[ \frac{\partial \mathbf{F}(\mathbf{x}^k)}{\partial \mathbf{x}_0} \right]^T \right)^{-1} \mathbf{F}(\mathbf{x}^k)$$

9:     (Eq. (5.5))
10:   else
11:     Tolerance met, closed trajectory IC found:
12:      $\mathbf{x}_0 = \mathbf{x}^k$ 
13:   end if
14: end while
```

5.2 Numerical Examples of Closed Relative Motion

Initial results of using the differential correcting algorithm to find closed orbits are presented in this section. The definition for a “closed” relative trajectory used is that the deputy satellite returns to its initial position to within a set numerical tolerance. Note that these trajectories may not repeat more than once. A variety of periodic orbits in the CR3BP were chosen for the chief satellite. Periodic initial conditions for the chief are obtained from NASA Jet Propulsion Laboratory’s three-body periodic orbits web page which contains a database of periodic CR3BP initial conditions [81]. In order to start the differential correction process, a first guess for the deputy initial conditions is set by back-propagation of the chief satellite for $t = -0.5$ seconds to $t = -300$ seconds. The differential correction algorithm uses a fixed time of flight to compute a solution. However, the use of multiple times of flight (TOFs) is investigated to determine “families” of relative orbits which may be compared to the type of families of relative orbits admitted in the 2-body problem using Clohessy-Wiltshire dynamics.

5.2.1 L2 Southern HALO Example

The first example considered is a scenario with the chief satellite in an L2 Southern HALO orbit. The chief and deputy initial conditions are given in Table 5.1. The initial guess for the

deputy is given as well as a final computed initial condition. Note that, the initial conditions are in non-dimensional units and set in the \mathcal{B} -frame.

Table 5.1 Deputy and chief initial conditions for L2 southern HALO example given in non-dimensional units in the barycentric frame. The deputy IC is given for the initial guess, and the final computed closed trajectory IC.

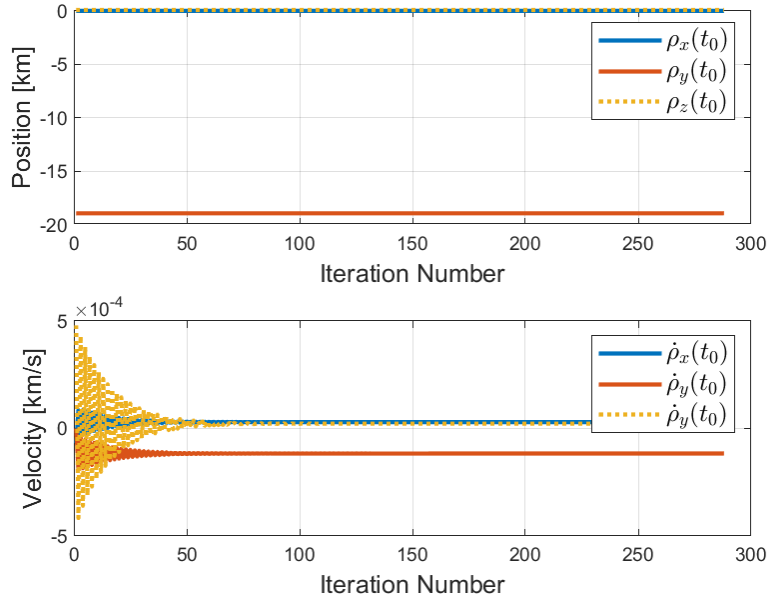
Initial Condition	Chief IC	Deputy First Guess IC	Deputy Computed IC
x_0 [DU]	1.004500661	1.004500634	1.004500634
y_0 [DU]	0	-4.91836E-05	-4.91836E-05
z_0 [DU]	-0.165719603	-0.165719416	-0.165719416
\dot{x}_0 [DU/TU]	-6.78809E-13	-6.70629E-05	2.89754E-05
\dot{y}_0 [DU/TU]	-0.061594336	-0.061594232	-0.061708084
\dot{z}_0 [DU/TU]	-9.13309E-12	0.00046888	2.18959E-05

Figure 5.3a shows the evolution of the design variables as iterations proceed. Note the trend of the velocity values converging to a solution, and note that the position values remain relatively unchanged. Figure 5.3b shows the magnitude of the constraint vector decreasing with iterations, in both regular and a logarithmic scale. The logarithmic scale shows the steady decrease in the magnitude of the constraint vector as convergence is achieved.

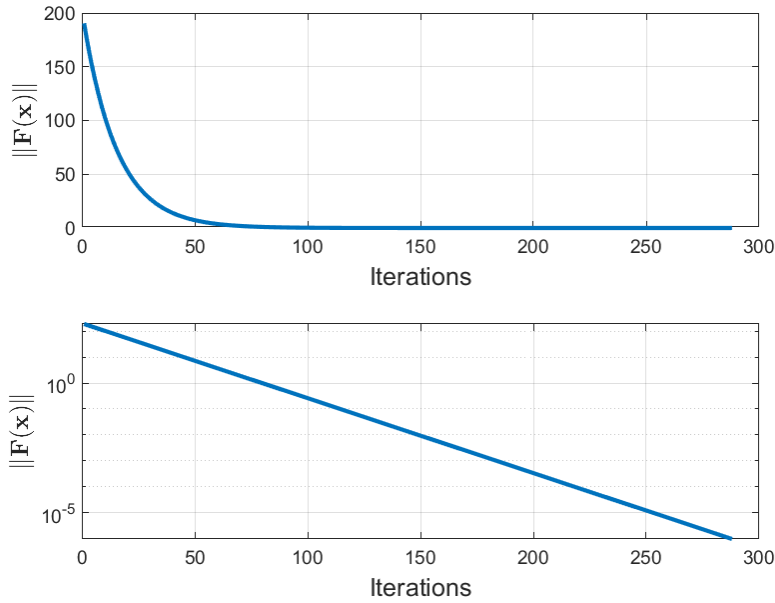
The progress of the differential corrector can also be visualized by the propagation of the deputy initial states given by the design variables at each iteration and is shown in Fig. 5.4a. The propagated trajectories for each iteration are shown in blue, while the final deputy trajectory is shown in red. Each blue trajectory can be seen to be approaching the final red closed trajectory. For clarity the final closed relative trajectory is also shown in Fig. 5.4b. Note that the deputy is essentially in a leader-follower formation and returns to its initial position at the final time.

Increasing the time of flight for the deputy leads to a “family” of closed trajectories. Figure 5.5a shows the results of increasing the TOF on the deputy trajectories. Note the increasing TOFs start to move farther from the chief satellite. Figure 5.5b shows the family of closed trajectories resulting from increasing TOFs along the ρ_x , ρ_y , and ρ_z axes. The color of the lines indicates the deputy TOF returns to its beginning position with blue being the shortest, and red being the longest TOF.

By increasing the TOF for the deputy, a bifurcation in the type of relative orbit occurs. The close-up closed relative orbit transitions to a relative orbit that resembles a “rendezvous” with the

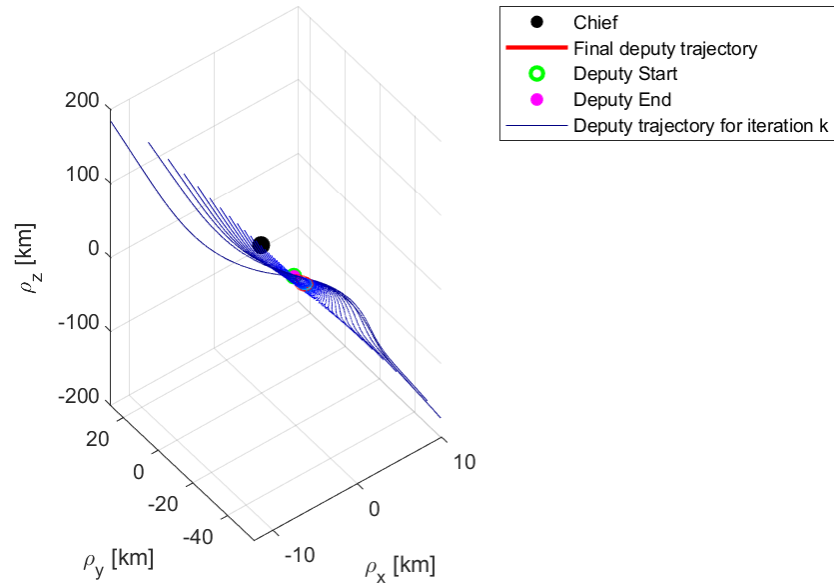


(a) Figure shows the design variable updates with increasing differential corrector iterations. The top plot shows the updates to the initial relative position and bottom plot shows the initial relative velocity updates.

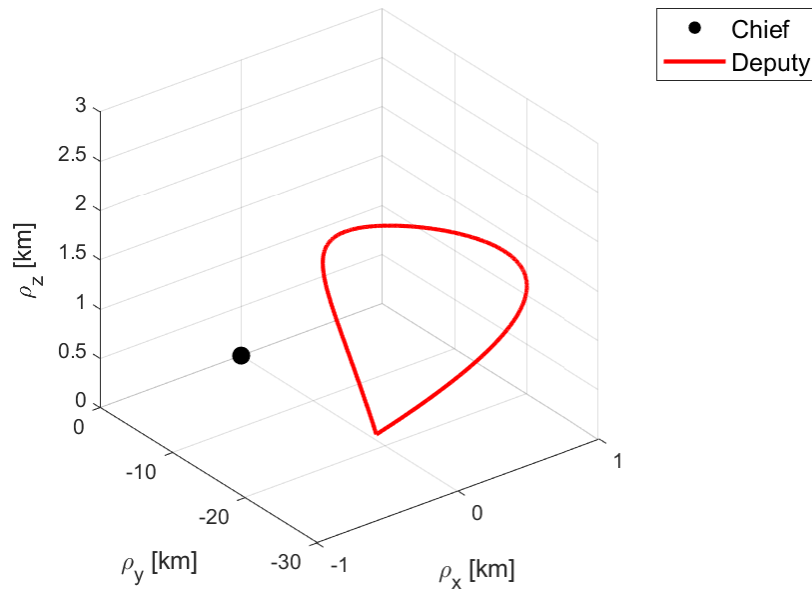


(b) Convergence of constraint vector to tolerance value. Top plot shows the magnitude of the constraint, bottom plot shows the magnitude of the constraint but plotted against a logarithmic scale to better show the final tolerance value is reached.

Figure 5.3 L2 southern HALO example: convergence of design variables and tolerance.

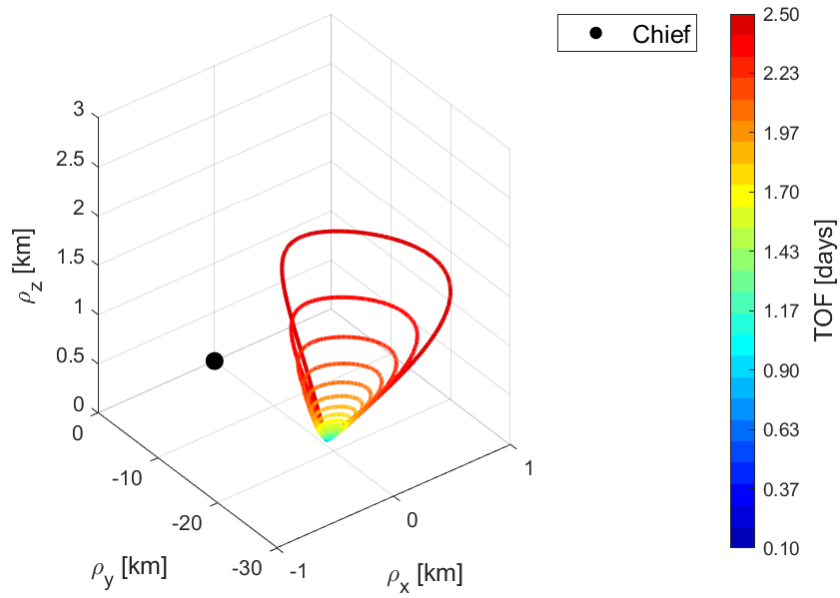


(a) Results of propagating the deputy initial position guess as the differential corrector proceeds. Each blue trajectory shows the resulting deputy relative position propagated to the final time for each iteration of the differential corrector. Final deputy closed trajectory shown in red.

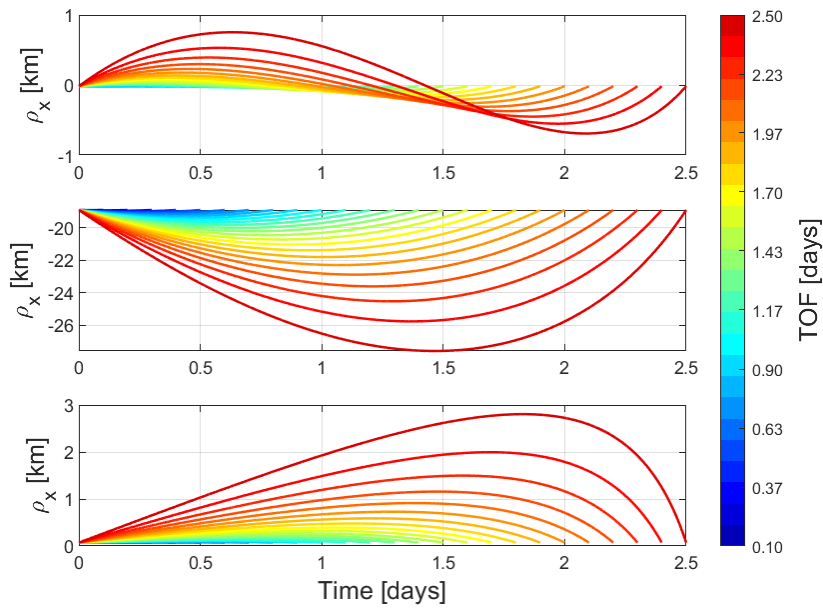


(b) Final deputy closed trajectory shown in red in the relative frame.

Figure 5.4 L2 southern HALO example: convergence and final closed trajectory.



(a) Deputy closed trajectories with increasing time of flight (TOF).



(b) Deputy closed trajectories for each axis with increasing time of flight (TOF).

Figure 5.5 L2 southern HALO example: family of solutions with increasing TOF.

same relative position to the chief. An example of this type of trajectory is shown in Fig. 5.6a which depicts the “rendezvous” like trajectory for the deputy. Essentially the deputy satellite leaves an initially close position to the chief and at a later point in the chief’s orbit returns to its initial position. The symmetry of the CR3BP most likely is what allows this type of relative orbit. Further investigation of these type of relative orbits is a subject of future work. The chief trajectory is visualized by propagation for one orbit period, as shown in Fig. 5.6b.

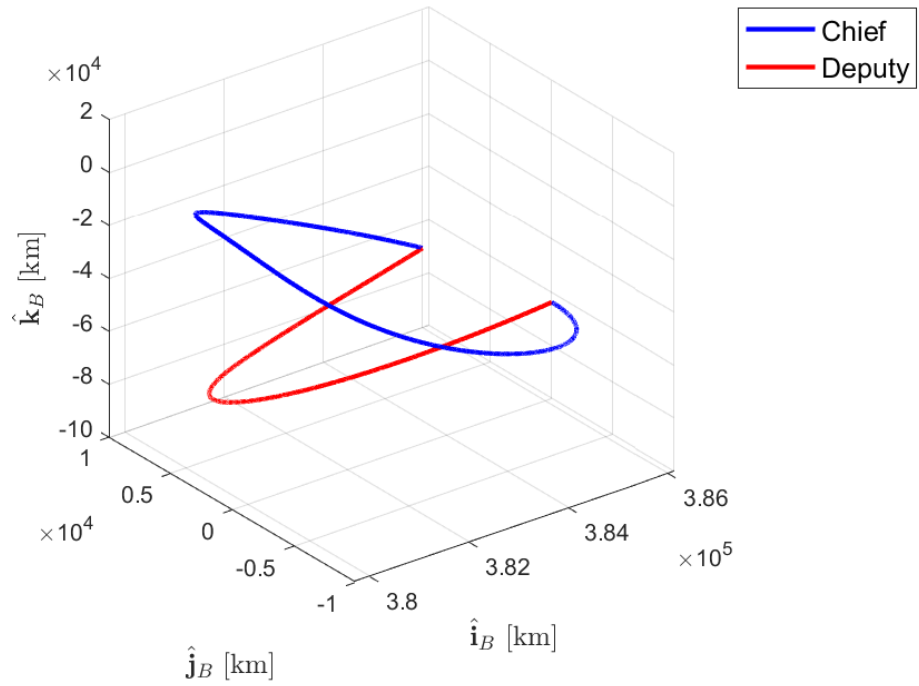
5.2.2 L1 Northern HALO Example

The next scenario shows an example with the chief satellite in a periodic L1 Northern HALO orbit. The method of back-propagating the chief state to initialize the deputy satellite first guess for the differential corrector is the same as above. Table 5.2 gives the chief and deputy initial conditions in non-dimensional units in the \mathcal{B} -frame. Note that, the deputy initial condition is given for both the first guess for the differential correction process and for an example final computed closed relative trajectory.

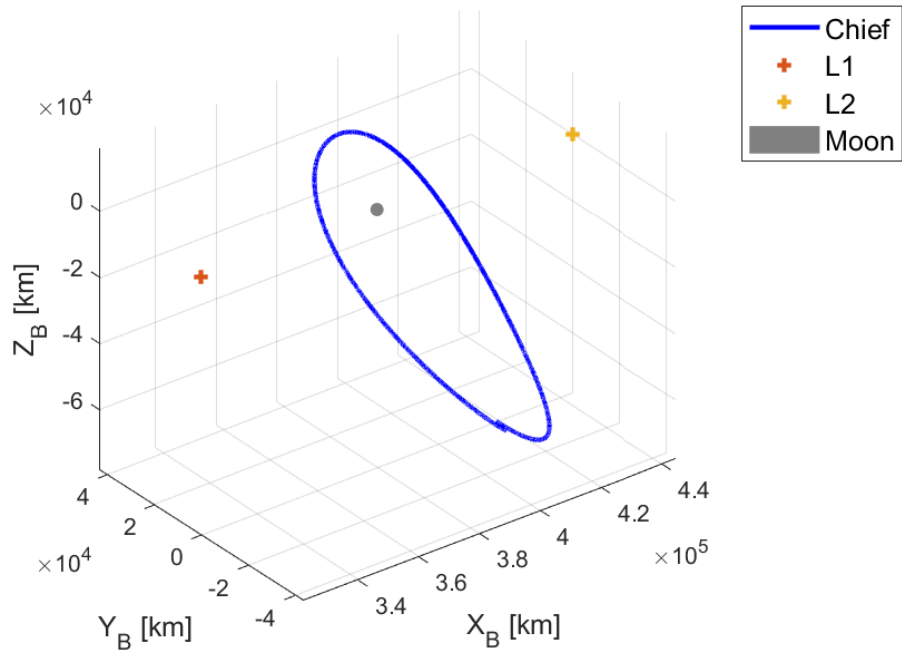
Table 5.2 Deputy and chief initial conditions for L1 northern HALO example given in non-dimensional units in the barycentric frame. The deputy IC is given for the initial guess, and the final computed closed trajectory IC.

Initial Condition	Chief IC	Deputy First Guess IC	Deputy Computed IC
x_0 [DU]	0.896599395	0.896599481	0.896599481
y_0 [DU]	0	0.000152826	0.000152826
z_0 [DU]	0.19928822	0.199288069	0.199288069
\dot{x}_0 [DU/TU]	-5.23356E-14	0.000213289	-4.714E-05
\dot{y}_0 [DU/TU]	0.191388835	0.19138858	0.191676123
\dot{z}_0 [DU/TU]	2.22247E-13	-0.000379869	-0.000115507

Figure 5.7a shows the design variables converging to their final value that give the closed trajectory shown in Fig. 5.8b. Figure 5.7b shows the magnitude of the constraint vector with each iteration; the bottom shows the same on a logarithmic y-axis. Figure 5.8a shows the differential corrector converging to the closed trajectory shown in red. The final trajectory is shown in Fig. 5.8b and gives a closed trajectory that “orbits” within a few hundred km of the chief. Note that an entire family of solutions exist, with several orbits staying closer to the chief with a shorter total TOF.

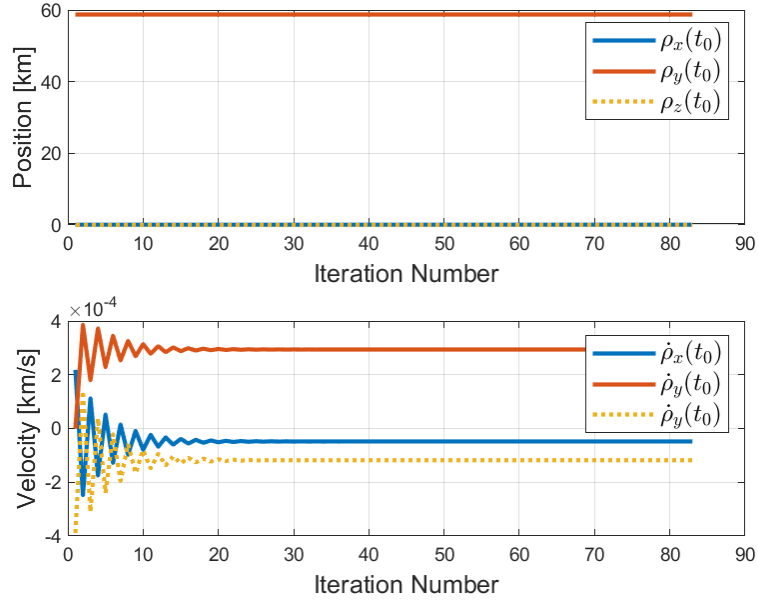


(a) Deputy and chief position in the rotating \mathcal{B} -frame. The deputy satellite intersects the chief orbit at two points and returns to its initial relative position.

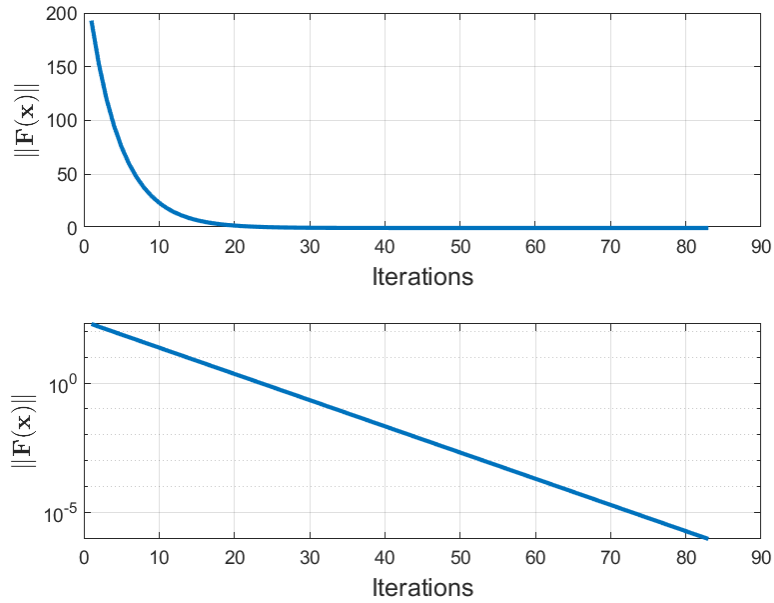


(b) Chief trajectory propagated for one period.

Figure 5.6 L2 southern HALO example: “rendezvous” trajectory and chief propagated orbit.

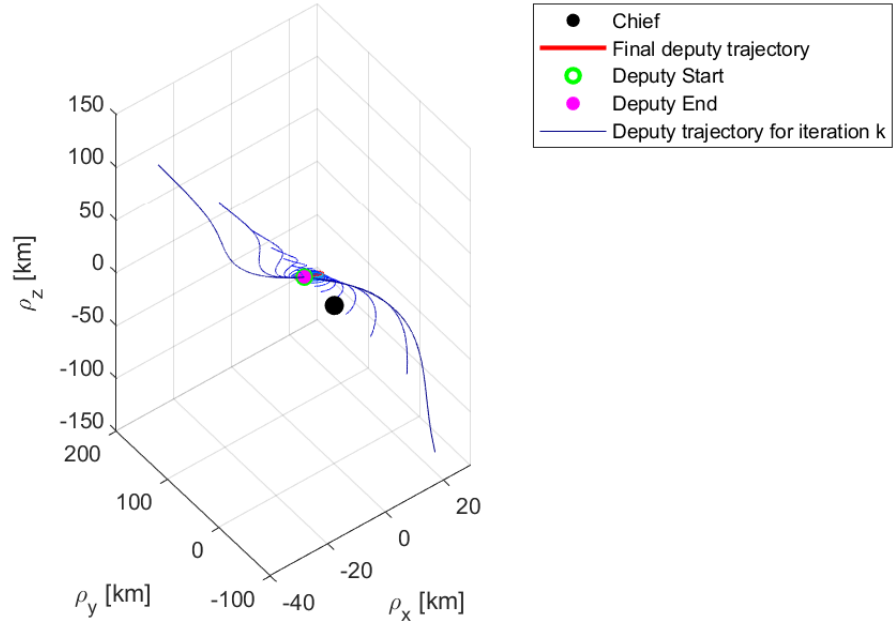


(a) L1 North HALO design variable updates with iterations. Top plot shows the initial relative position and bottom plot shows the initial relative velocity for the deputy.

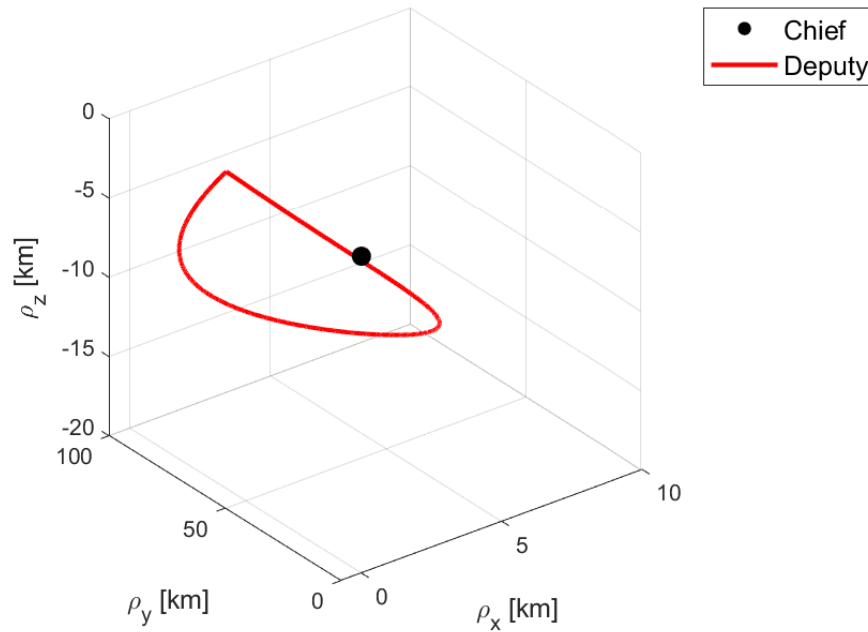


(b) L1 North HALO convergence of constraint vector to tolerance value. Top plot shows the magnitude of the constraint, bottom plot shows the magnitude of the constraint but plotted against a logarithmic scale to better show the final tolerance value is reached.

Figure 5.7 L1 northern HALO example: convergence of design variables and tolerance.



(a) Results of propagating the deputy initial position guess as the differential corrector proceeds. Each blue trajectory shows the resulting deputy relative position propagated to the final time for each iteration of the differential corrector. The final deputy trajectory is shown in red.



(b) Final deputy closed trajectory shown in in the relative frame.

Figure 5.8 L1 northern HALO example: convergence and final closed trajectory.

By increasing the TOF, a family of relative orbits is generated. Figure 5.9a shows the family of trajectories that were found to exist for TOFs ranging from 0.5 to 3.75 days. Another family of the “rendezvous” type trajectories exists for longer TOFs. The evolution of the family of orbits is more clearly seen by looking at the values of the relative position in each axis as shown in Fig. 5.9b. The chief trajectory propagated for one orbit period is shown in Fig. 5.10 to give a better view of the expected orbit geometry.

5.2.3 Resonant Orbit Example

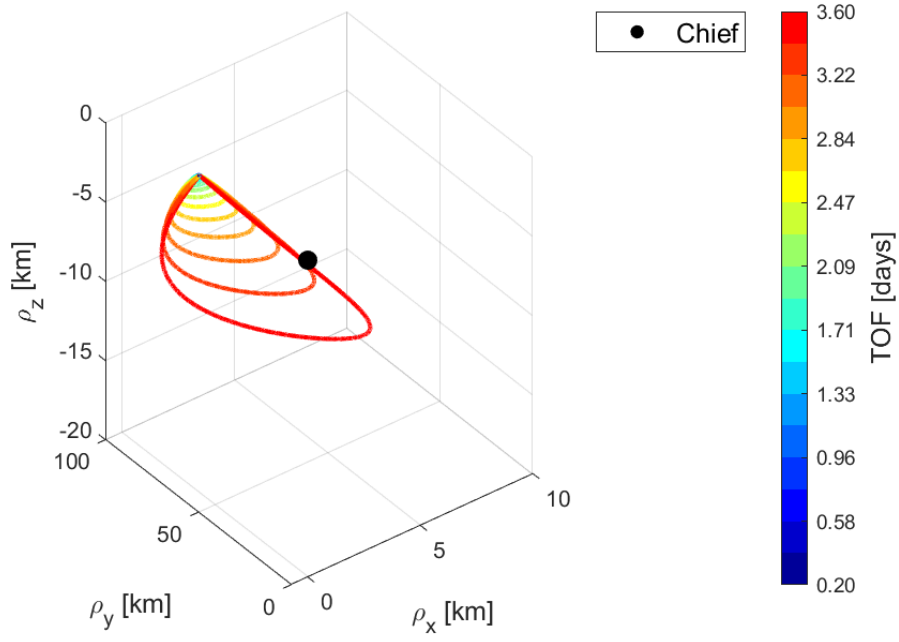
The next scenario example places the chief satellite in a 3-2 resonant orbit. The chief satellite initial condition is again backward propagated for a few seconds, and then subtracted from the chief IC to setup an initial guess for the deputy state. Similar to the HALO orbit examples, two distinct types of solutions exist, a family of closed relative orbits near the chief, and a family of orbits that essentially rendezvous with the deputy’s initial position at a later point in the chief’s orbit. Table 5.3 gives the initial conditions for the chief and deputy in non-dimensional units in the barycentric frame. Note that, the deputy IC is given for both the first guess for the differential correction process and the final computed closed-trajectory.

Table 5.3 Deputy and chief initial conditions for 3:2 resonant orbit example given in non-dimensional units in the barycentric frame. The deputy IC is given for the initial guess, and the final computed closed trajectory IC.

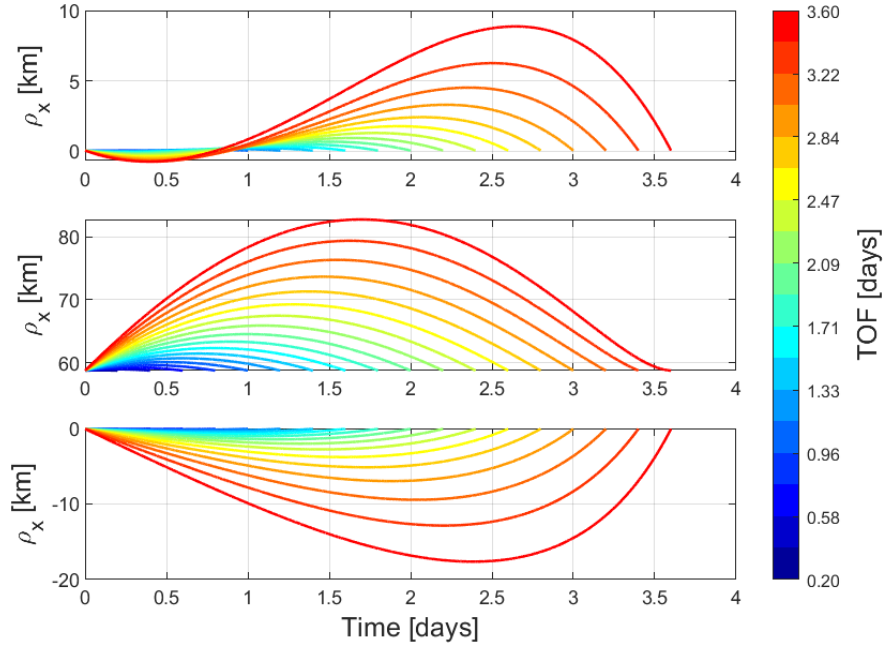
Initial Condition	Chief IC	Deputy First Guess IC	Deputy Computed IC
x_0 [DU]	0.689923223	0.689923205	0.689923205
y_0 [DU]	0	0.000449088	0.000449088
z_0 [DU]	0	0	0
\dot{x}_0 [DU/TU]	-9.18057E-16	-4.5972E-05	-0.002602586
\dot{y}_0 [DU/TU]	0.562407482	0.562407102	0.564686794
\dot{z}_0 [DU/TU]	0	0	0

Figure 5.11a shows the design variables (initial position and velocity) for the deputy satellite converge as iterations increase for the differential corrector. And Fig. 5.11b shows the convergence of the constraint to the selected tolerance value.

The convergence of the differential corrector can also be seen by propagating to the final time the design variable (deputy IC) at each iteration of the differential corrector. Figure 5.12a shows the



(a) Deputy closed trajectories with increasing time of flight.



(b) Deputy closed trajectories for each axis with increasing time of flight.

Figure 5.9 L1 northern HALO example: family of solutions with increasing TOF.

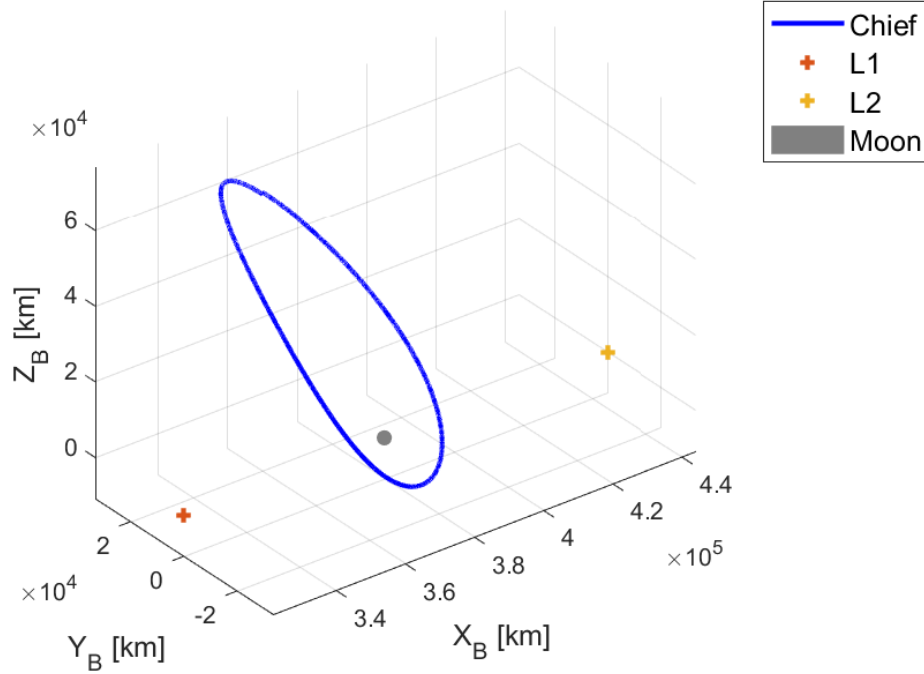
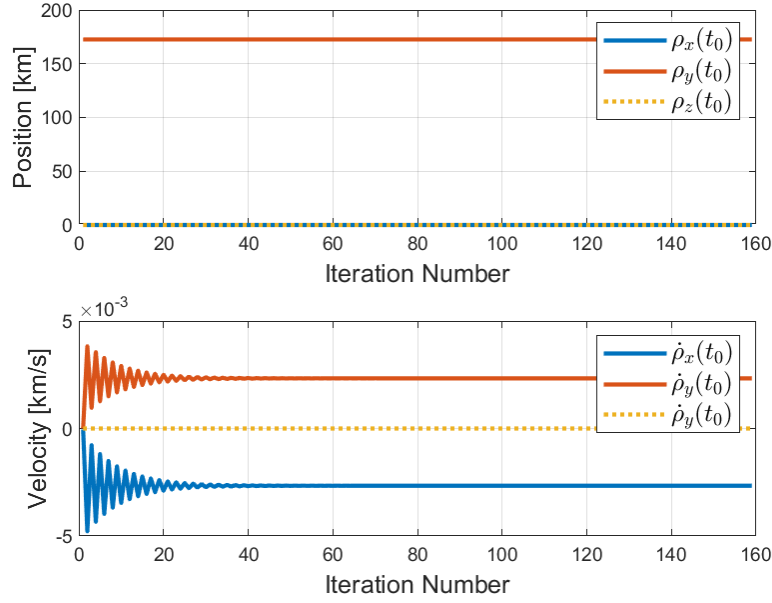


Figure 5.10 Chief orbit propagated for one period for L1 northern HALO orbit.

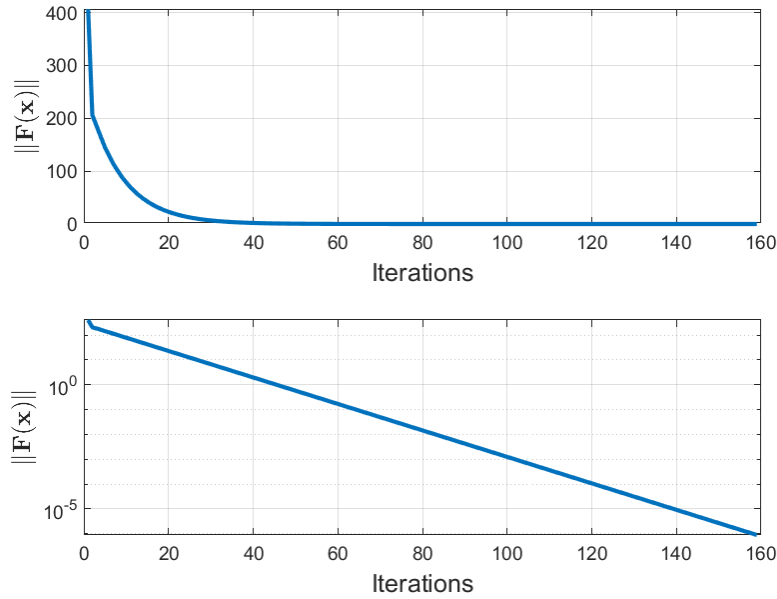
deputy relative orbits converging to the closed path found on the final iteration plotted in red. Figure 5.12b shows just the final closed trajectory of the deputy relative to the chief satellite. Again, the effect of increasing TOF is illustrated in the finding of a family of relative orbits. In this case, for the resonant orbit, the relative orbits travel to much greater distances than for either of the HALO orbit examples. Figure 5.13a shows the family of orbits in the relative frame. And Fig. 5.13b shows the relative positions for the family of relative orbits in each axis. Note that these are planar orbits. The chief's orbit for one period is shown in Figure 5.14 for context.

5.2.4 Natural Closed Relative Motion Summary

One application of the linearized relative motion dynamics was shown in utilizing a differential corrector to find closed relative trajectories in the CR3BP. The algorithm was tested on several different cislunar orbital geometries and shows promise for finding closed trajectories for a variety of time of flight values. Two families of trajectories were observed, a family of closed trajectories near the chief, and a set of solutions that essentially become rendezvous trajectories with the deputy initial relative position.

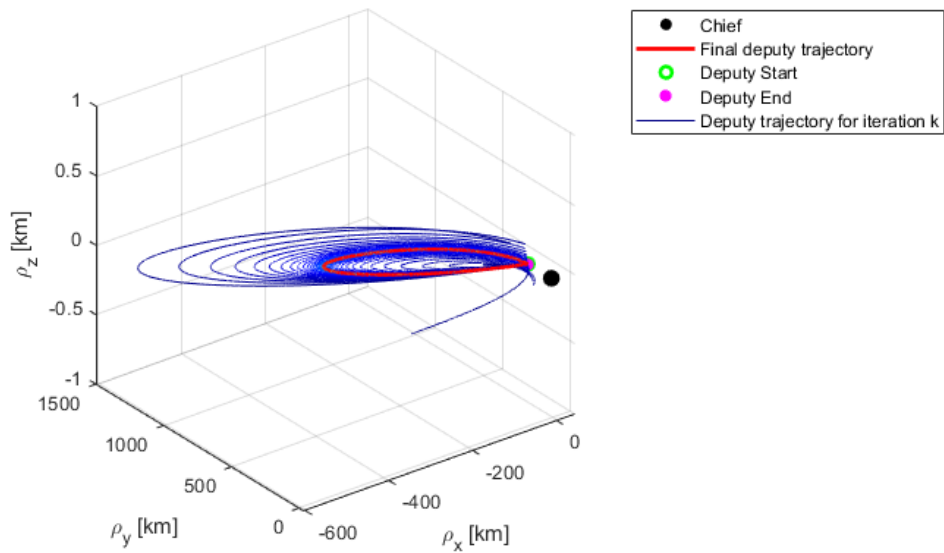


(a) Design variable updates with iterations. Top plot shows the initial relative position and bottom plot shows the initial relative velocity for the deputy.

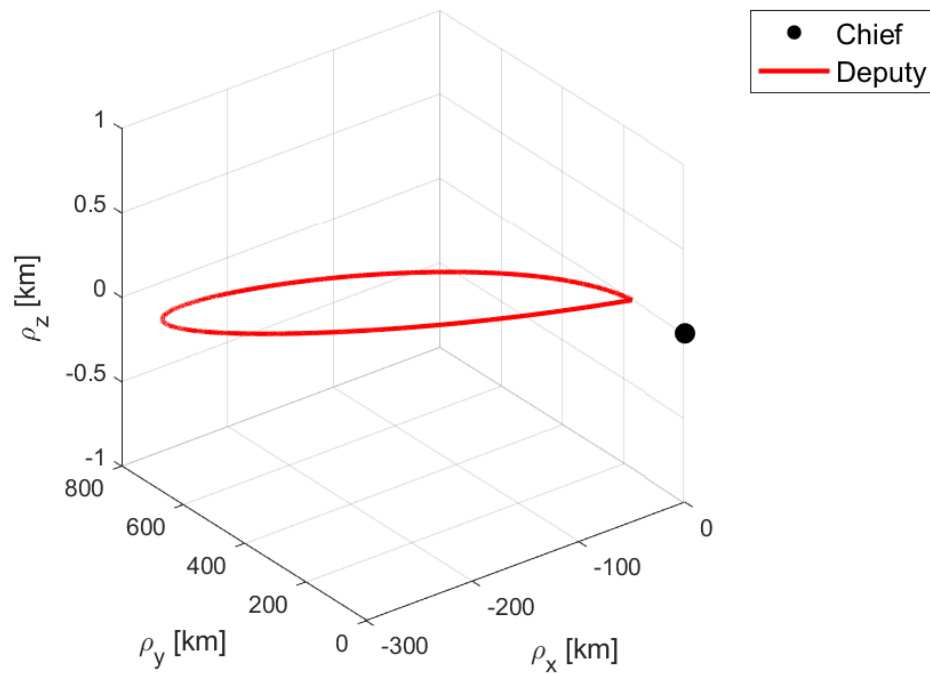


(b) Convergence of constraint vector to tolerance value. Top plot shows the magnitude of the constraint, bottom plot shows the magnitude of the constraint but plotted against a logarithmic scale to better show the final tolerance value is reached.

Figure 5.11 Resonant orbit example convergence.

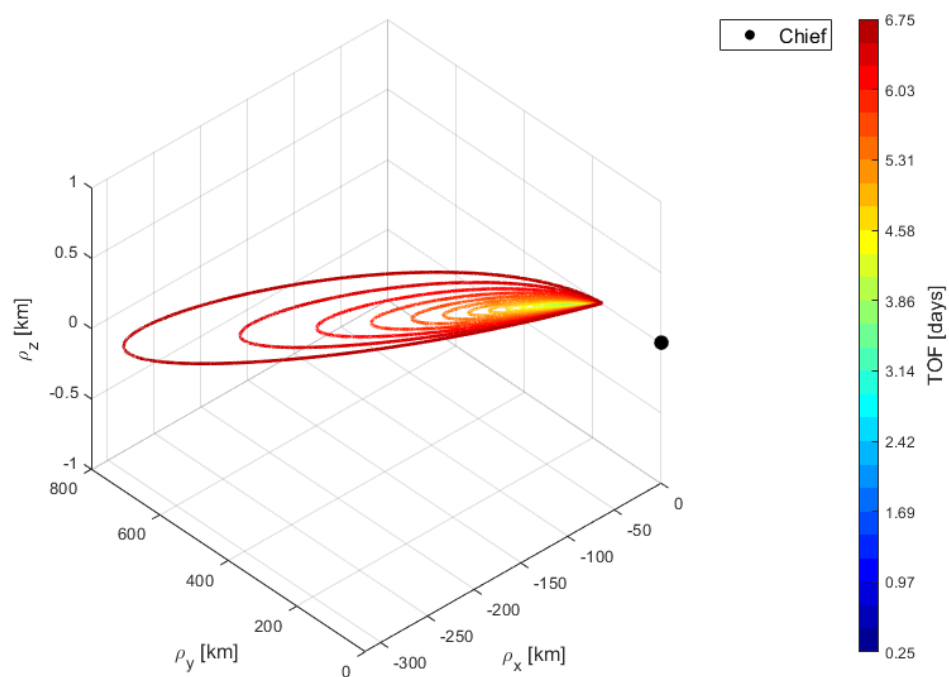


(a) Results of propagating the deputy initial position guess as the differential corrector proceeds. Each blue trajectory shows the resulting deputy relative position propagated to the final time for each iteration of the differential corrector. Final deputy closed trajectory shown in red.

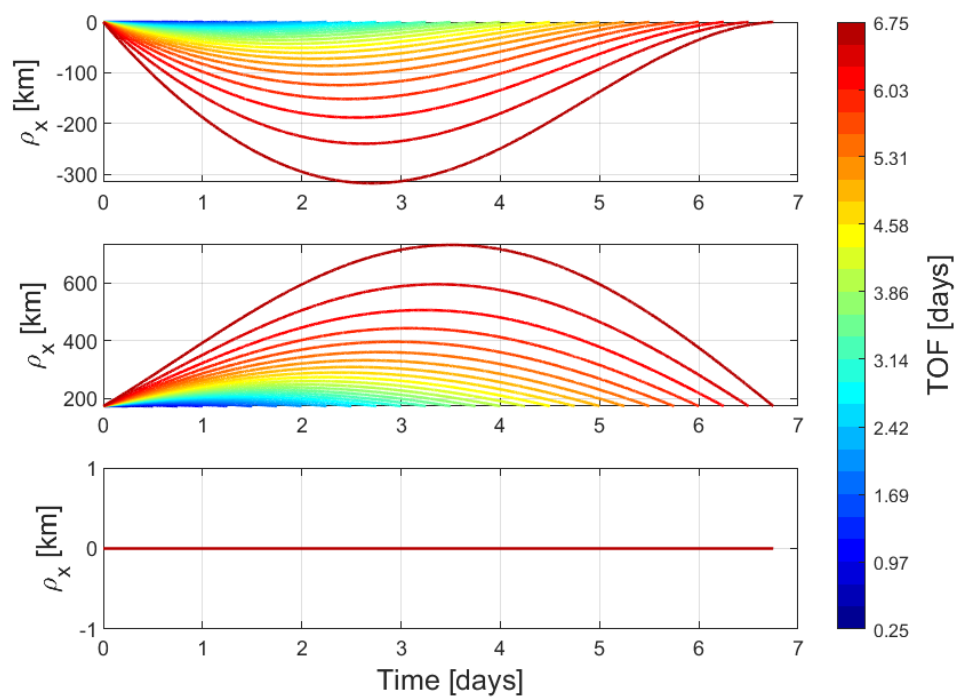


(b) Final deputy closed trajectory shown in red in the relative frame.

Figure 5.12 Resonant orbit example convergence.



(a) Deputy trajectories with increasing time of flight.



(b) Deputy closed trajectories for each axis with increasing time of flight.

Figure 5.13 Resonant orbit example relative orbits.

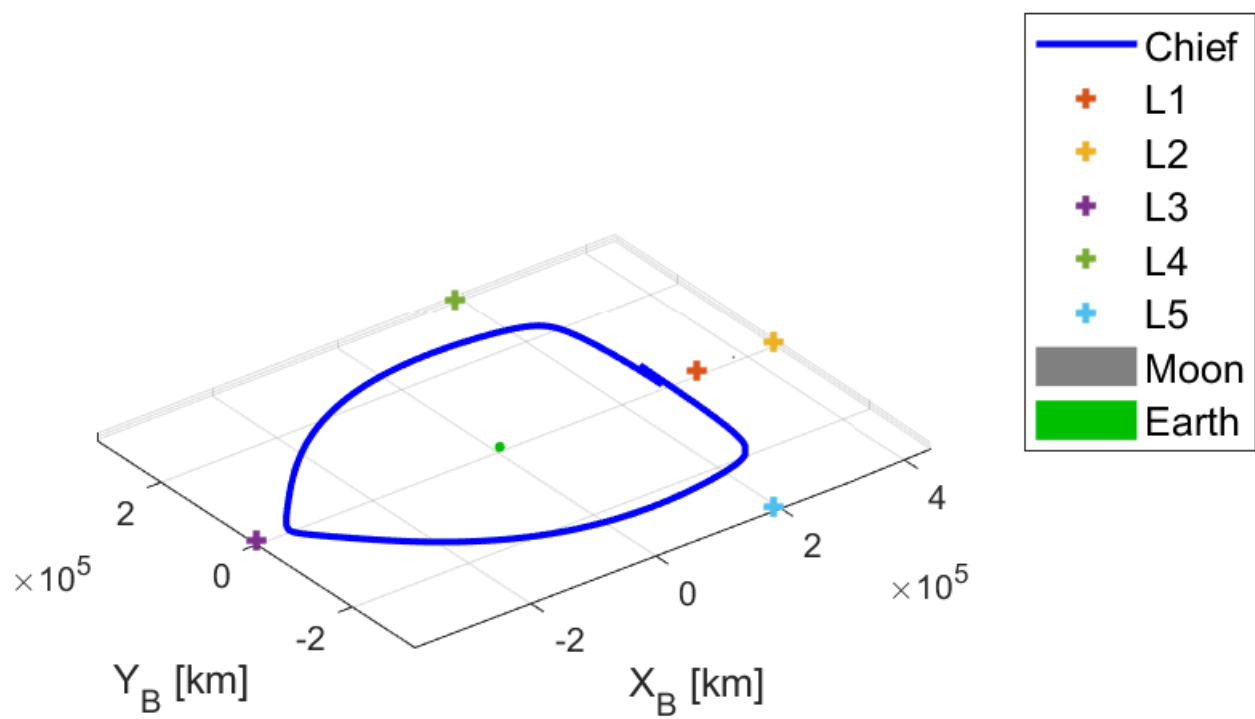


Figure 5.14 Chief orbit propagated for one period for 3-2 resonant orbit case.

5.3 Initial Relative Orbit Determination (IROD)

An additional application of the linear exponential motion models (LEXPM-E, LEXPM-C) lies in forming a semi-analytical solution for the spacecraft motion in terms of the initial conditions. Spacecraft relative state-estimation is of great importance for applications such as docking. Filtering approaches have been developed to facilitate relative navigation in cislunar space; however all navigation filters require an initial guess to kick-start the filter [40]. One such approach is proposed here with a method called initial relative orbit determination (IROD). Forming an initial guess using an IROD method has been shown to work for two-body dynamics using first and second order Clohessy Wiltshire (CW) relative dynamics [78, 85]. The IROD problem is cast as a polynomial solution problem wherein the deputy initial condition can be recovered from a set of three line of sight measurements. It will be assumed that the chief's motion is known and well characterized, and that LOS measurements are available at three separate times. It is also assumed that the relative vector from the chief to the deputy ($\boldsymbol{\rho}$) is co-linear with any LOS measurements. Since the relative vector and LOS measurement are co-linear, then their cross product will be zero. Therefore, let the IROD measurement equation be defined as shown in Eq. (5.12).

$$\boldsymbol{\rho}(t_k) \times \hat{\mathbf{u}}(t_k) = \mathbf{0}_{3 \times 1} \quad (5.12)$$

For convenience the necessary assumptions for the IROD problem are summarized below.

1. Chief spacecraft position is known: \mathbf{R}_c .
2. Three LOS measurements $\hat{\mathbf{u}}(t_k)$ are available.
3. The LOS measurements are sufficiently spaced to provide observability.
4. An analytical solution for deputy motion is available.
5. The deputy position $\boldsymbol{\rho}(t_k)$ and LOS vectors ($\hat{\mathbf{u}}(t_k)$) are co-linear: $\boldsymbol{\rho}(t_k) \times \hat{\mathbf{u}}(t_k) = \mathbf{0}_{3 \times 1}$.
6. The measurement equations (Eq. (5.12)) can be written in polynomial form of the deputy initial conditions: $\boldsymbol{\rho}(t_0)$, $\dot{\boldsymbol{\rho}}(t_0)$.

For the two-body IROD problem, a common limitation was found to be range ambiguity

between measurements for a set of relative orbits. This problem was overcome by using higher order approximations for the linear motion of the CW equations [78]. For the three-body problem the orbital dynamics and linear model should not need more than the single-order linearization to avoid the range ambiguity problem. The deputy motion will be assumed to be operating using the LEXPM-C motion model developed in in Section 4.3.3.

5.3.1 Three-Body IROD

To form the IROD problem for three-body dynamics, first we will assume the same type of general three-body relative scenario in cis-lunar space depicted in Fig. 4.4 that was used to develop the motion models of the previous sections. Given the scenario geometry shown in Fig. 4.4, define the LOS unit vector measurement at time t_k as:

$$\hat{\mathbf{u}}(t_k) = \begin{bmatrix} u_x(t_k) \\ u_y(t_k) \\ u_z(t_k) \end{bmatrix} \quad (5.13)$$

where the measured LOS vector is assumed to be parallel with the relative position of the deputy satellite:

$$\boldsymbol{\rho}(t_k) = \mathbf{R}_d(t_k) - \mathbf{R}_c(t_k) \quad (5.14)$$

$$\hat{\mathbf{u}}(t_k) = \frac{\boldsymbol{\rho}(t_k)}{\|\boldsymbol{\rho}(t_k)\|} \quad (5.15)$$

Figure 5.15 shows the geometry of the IROD scenario being discussed. Note that the deputy's position is not restricted to any particular trajectory. The chief satellite takes three LOS measurements (shown in green) that are parallel to the deputy's current position vector. The goal is to recover the deputy satellite's initial state (\mathbf{X}_0) using only knowledge of the chief's position, three LOS measurements, and a closed form solution for the motion of the deputy satellite.

Because the LOS vector and the relative vector are assumed to be co-linear, then the cross-

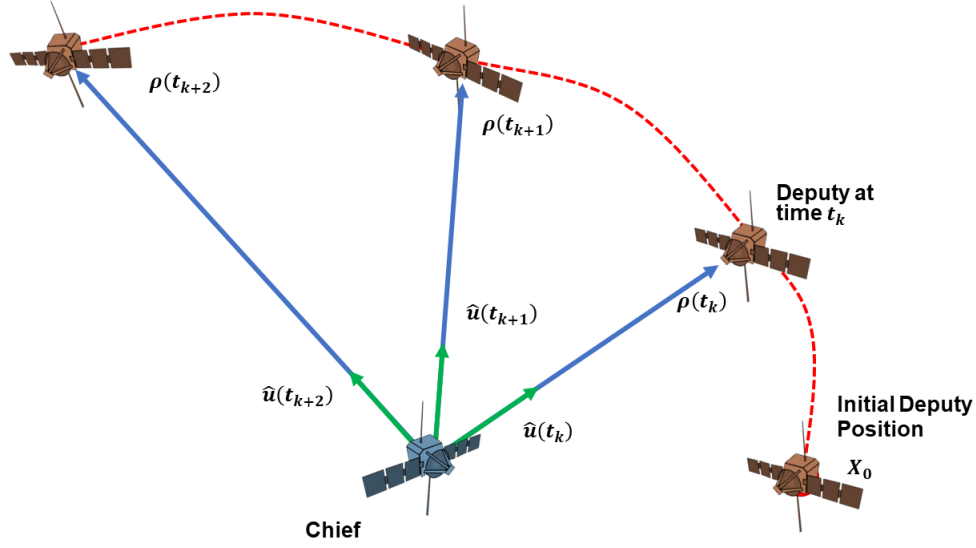


Figure 5.15 Typical IROD scenario shown in relative frame based at the chief. Three LOS measurements are taken that are parallel to the current deputy position vector.

product between these two vectors will be zero.

$$\hat{\mathbf{u}}(t_k) \times \boldsymbol{\rho}(t_k) = \mathbf{0} \quad (5.16)$$

The skew-matrix form of the cross-product gives a matrix vector form for Eq. (5.16).

$$[\hat{\mathbf{u}}(t_k)]^\times = \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix} \quad (5.17)$$

$$\hat{\mathbf{u}}(t_k) \times \boldsymbol{\rho}(t_k) = \mathbf{0} \quad (5.18)$$

$$[\hat{\mathbf{u}}(t_k)]^\times \boldsymbol{\rho}(t_k) = \mathbf{0} \quad (5.19)$$

Equation (5.19) will be referred to as the “IROD constraint” since it constrains our LOS measurements to be parallel to the deputy relative position vectors. At this point, it is desired to write the deputy state at time t_k in terms of it’s STM and a deputy initial condition. Let the deputy’s initial

condition be written as the vector \mathbf{X}_0 of the deputy initial position and velocity:

$$\mathbf{X}_0 = \begin{bmatrix} \boldsymbol{\rho}(t_0) \\ \dot{\boldsymbol{\rho}}(t_0) \end{bmatrix} \quad (5.20)$$

For a linear system, a later state can easily be written in terms of the STM and an initial condition. Furthermore, for an LTI system the STM is the exponential matrix of the system state-matrix. Therefore, assume that the deputy dynamics can adequately be represented by the LEXPM equations of motion, with all of the appropriate assumptions as discussed in Sec. 4.3.3. Thus the deputy state at time t_k can be written as the matrix product of the STM between the times multiplied with the initial state:

$$\mathbf{X}(t_k) = \Phi(t_k, t_0)\mathbf{X}_0 \quad (5.21)$$

For the IROD constraint equation, only the deputy position is required and not the full state. The position state of the deputy can be extracted from Eq. (5.21) by pre-multiplication of the STM by a 3×6 matrix composed of a block 3×3 identity and 3×3 matrix of zeros. The matrix to extract the position states will be denoted by $[C_{3 \times 6}]$ and is defined in Eq. (5.22).

$$[C_{3 \times 6}] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (5.22)$$

Furthermore, we will assume that the STM can be written as the exponential matrix given by Eq. (4.118) as discussed in Section 4.3.3. Thus the deputy's position can be written in terms of an initial states using the LEXPM-C solution. Note that, the STM expansion to the matrix exponential uses the system state matrix $A(t_0)$ fixed at the initial time since the deputy states at t_0 are the desired

states to be computed.

$$\boldsymbol{\rho}(t_k) = [C_{3 \times 6}] \Phi(t_k, t_0) \mathbf{X}_0 \quad (5.23)$$

$$\boldsymbol{\rho}(t_k) = [C_{3 \times 6}] e^{A(t_0)\Delta t_k} \mathbf{X}_0 \quad (5.24)$$

$$\boldsymbol{\rho}(t_k) = [C_{3 \times 6}] \left[I_{6 \times 6} + A(t_0)\Delta t_k + \frac{A(t_0)^2 \Delta t_k^2}{2!} + \dots + \frac{A(t_0)^n \Delta t_k^n}{n!} \right] \mathbf{X}_0 \quad (5.25)$$

By substitution of the deputy relative position in terms of the deputy initial position (Eq. (5.25)) into the IROD measurement equation, we arrive at an expression that relates the LOS measurements to the deputy initial states.

$$[\hat{\mathbf{u}}(t_k)]^\times \boldsymbol{\rho}(t_k) = [\hat{\mathbf{u}}(t_k)]^\times [C_{3 \times 6}] \left[I_{6 \times 6} + A(t_0)\Delta t_k + \frac{A(t_0)^2 \Delta t_k^2}{2!} + \dots + \frac{A(t_0)^n \Delta t_k^n}{n!} \right] \mathbf{X}_0 = \mathbf{0} \quad (5.26)$$

To ease notation, let the STM notation be used for the exponential matrix thus:

$$[\hat{\mathbf{u}}(t_k)]^\times \boldsymbol{\rho}(t_k) = [\hat{\mathbf{u}}(t_k)]^\times [C_{3 \times 6}] \Phi(t_k, t_0) \mathbf{X}_0 = \mathbf{0} \quad (5.27)$$

Define the vector $\mathbf{b}(t_k)$ as:

$$\mathbf{b}(t_k) = \begin{bmatrix} b_1(t_k) \\ b_2(t_k) \\ b_3(t_k) \end{bmatrix} = [C_{3 \times 6}] \left[I_{6 \times 6} + A(t_0)\Delta t_k + \frac{A(t_0)^2 \Delta t_k^2}{2!} + \dots + \frac{A(t_0)^n \Delta t_k^n}{n!} \right] \mathbf{X}_0 \quad (5.28)$$

$$\mathbf{b}(t_k) = [C_{3 \times 6}] \Phi(t_k, t_0) \mathbf{X}_0 \quad (5.29)$$

$$\mathbf{b}(t_k) = \boldsymbol{\rho}(t_k) \quad (5.30)$$

Note that the elements of $\mathbf{b}(t_k)$ correspond to the ρ_x, ρ_y, ρ_z components of the deputy's position at time t_k . Using the definition of $\mathbf{b}(t_k)$ in the IROD constraint Eq. (5.19), and working out the matrix vector multiplication, we arrive at a set of three equations, of which only two are linearly

independent.

$$[\hat{\mathbf{u}}(t_k)]^\times \boldsymbol{\rho}(t_k) = [\hat{\mathbf{u}}()]^\times b(t_k) = \mathbf{0} \quad (5.31)$$

$$\begin{bmatrix} 0 & -u_z(t_k) & u_y(t_k) \\ u_z(t_k) & 0 & -u_x(t_k) \\ -u_y(t_k) & u_x(t_k) & 0 \end{bmatrix} \begin{bmatrix} b_1(t_k) \\ b_2(t_k) \\ b_3(t_k) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (5.32)$$

$$-u_z(t_k)b_2(t_k) + u_y(t_k)b_3(t_k) = 0 \quad (5.33)$$

$$u_z(t_k)b_1(t_k) - u_x(t_k)b_3(t_k) = 0 \quad (5.34)$$

$$-u_y(t_k)b_1(t_k) + u_x(t_k)b_2(t_k) = 0 \quad (5.35)$$

Note that in Eqs. (5.33) - (5.35), the LOS measurement components of $\hat{\mathbf{u}}(t_k)$ are known; however, the relative positions of the deputy given by $\mathbf{b}(t_k) = \boldsymbol{\rho}(t_k)$ are currently unknown. Further note that though, we have three sets of equations, only two-are linearly independent. Thus if three measurements are taken at times t_k , t_{k+1} , and t_{k+2} , then at each time step a set of two independent equations can be formed which will give a total of six equations and six unknowns. Thus, at three measurement times, the IROD constraint equation can be written in the following compact form.

$$[\hat{\mathbf{u}}(t_k)]^\times \mathbf{b}(t_k) = \mathbf{0} \quad (5.36)$$

$$[\hat{\mathbf{u}}(t_{k+1})]^\times \mathbf{b}(t_{k+1}) = \mathbf{0} \quad (5.37)$$

$$[\hat{\mathbf{u}}(t_{k+2})]^\times \mathbf{b}(t_{k+2}) = \mathbf{0} \quad (5.38)$$

The task at hand now is to relate the deputy initial state to the deputy state at each measurement time. Using the definition of the $\mathbf{b}(t_k)$ vector this can be accomplished by utilizing the property of superposition that for a linear system the STM can be written between two different times as the multiplication of the STM for each time-span.

$$\Phi(t_{k+1}, t_0)\mathbf{X}_0 = \Phi(t_{k+1}, t_k)\Phi(t_k, t_0)\mathbf{X}_0 \quad (5.39)$$

Thus the vector $\mathbf{b}(t_k)$ can be written for each measurement time in terms of only the initial deputy state as the unknown variables.

$$\mathbf{b}(t_k) = [C_{3 \times 6}] \Phi(t_k, t_0) \mathbf{X}_0 \quad (5.40)$$

$$\mathbf{b}(t_{k+1}) = [C_{3 \times 6}] \Phi(t_{k+1}, t_0) \mathbf{X}_0 \quad (5.41)$$

$$\mathbf{b}(t_{k+2}) = [C_{3 \times 6}] \Phi(t_{k+2}, t_0) \mathbf{X}_0 \quad (5.42)$$

Substituting in the expressions for $\mathbf{b}(t_k)$ into Eqs. (5.36) - (5.38) yields the IROD constraint equation for each measurement time, where the only unknown elements are the deputy initial conditions.

$$[\hat{\mathbf{u}}(t_k)]^\times [C_{3 \times 6}] \Phi(t_k, t_0) \mathbf{X}_0 = \mathbf{0} \quad (5.43)$$

$$[\hat{\mathbf{u}}(t_{k+1})]^\times [C_{3 \times 6}] \Phi(t_{k+1}, t_0) \mathbf{X}_0 = \mathbf{0} \quad (5.44)$$

$$[\hat{\mathbf{u}}(t_{k+2})]^\times [C_{3 \times 6}] \Phi(t_{k+2}, t_0) \mathbf{X}_0 = \mathbf{0} \quad (5.45)$$

Equations (5.43) - (5.45) form a set of nine equations with six unknowns to be solved for; however only six of the equations are actually linearly independent. By the use of a finite number of terms for the exponential matrix solution for the STM, the equations become a polynomial system of highly non-linear equations to solve for the initial deputy state \mathbf{X}_0 .

5.3.2 MATLAB Symbolic Solution Method

Rather than attempt to write out by hand the matrix exponential expansions implicit in Eqs. (5.43) - (5.45), the symbolic math capabilities of MATLAB will be leveraged. Since it is assumed the chief position vectors and the LOS measurements are all available, the only unknowns will be the six initial states of the deputy satellite. For a first test of the method the CR3BP dynamics will be used, and thus the LEXPM-C EOMs utilized for the STM. In principle, the higher the number of terms used for the exponential matrix expansion for the system STM, the more accurate the state propagation will be. However, beyond a few terms, Eq. (5.45) becomes intractable even for the symbolic math capabilities of MATLAB to solve due to the high matrix powers of terms involved.

For a given number of desired terms N , then the matrix exponential terms for the STM can be expanded in terms of the current chief position $\mathbf{R}_c(\mathbf{t}_k)$ and the symbolic deputy initial position and velocity given by \mathbf{X}_0 . Note once again that, the system state matrix $A(t_0)$ is held fixed at the initial time in order to recover the deputy initial state.

$$\Phi(t_k, t_0) = \sum_{n=0}^N \left[\frac{A(t_0)^n \Delta t_k^n}{n!} \right] \quad (5.46)$$

The system matrix $A(t)$ is defined by Eq. (4.106). The deputy state initial state is taken to be the initial condition \mathbf{X}_0 each time that $A(t_0)$ is used to compute the STM. Using the STM propagated to each measurement time ensures that the states at later times map back to the initial deputy states. Given a set of three LOS measurements, and the chief satellite position states at each measurement time, Eqs. (5.43) - (5.45) can now be solved numerically to recover the deputy initial condition. MATLAB's `vpasolve` function is used to numerically attempt to solve the IROD equations using a basic Newton root solver [86]. Note that more advanced polynomial solution methods certainly could be used to obtain a solution and are suggested to be pursued by individuals interested in furthering this line of research.

5.3.3 IROD Results

Two short examples proving the feasibility of IROD with linearized three-body dynamics are presented in this section. In order to provide future researchers the opportunity of replicating these results, the initial conditions for the deputy and chief satellites are provided in a table for each scenario. Initial conditions are provided in the \mathcal{B} -frame using non-dimensional units with a distance and time-unit given by: $DU = 384400$ km and $TU = 3.756998819175271 \times 10^5$.

IROD Example: Non-Periodic Deputy

The first example places the chief satellite in an L2-Lyapunov orbit and the deputy satellite at an initial separation along the y-axis. The overall orbit geometry of the deputy and chief satellites is shown in Fig. 5.16. Note that this is a planar scenario to simplify the problem slightly. The deputy is initially separated only along the y-axis and then spirals away from the chief over the course of

the scenario. The initial conditions for the chief and deputy are provided in Table 5.4. Note that, the initial conditions are given in non-dimensional units in the barycentric rotating frame.

Table 5.4 Chief and deputy initial conditions for L2 Lyapunov orbit IROD scenario. Note that the initial conditions are given in non-dimensional units in the barycentric frame.

Initial Condition	Chief CR3BP IC	Deputy CR3BP IC
x_0 [DU]	1	0.999989148
y_0 [DU]	0	0.000689121
z_0 [DU]	0	0
\dot{x}_0 [DU/TU]	-1.67E-14	-0.041538566
\dot{y}_0 [DU/TU]	1.32	1.318804057
\dot{z}_0 [DU/TU]	0	0

In order to show that the IC recovered can be at an arbitrary time-step, the first measurement time was referenced to a time-step 2.89 days after the scenario began, with measurements at 4.34, 5.50, and 6.37 days. Each measurement for now is assumed to be noise-free for a proof of concept that the IROD equations can be solved for a three-body scenario. The results of the scenario are shown in Fig. 5.17. The chief initial position is shown as a blue circle, with the deputy initial position as a red circle. The IROD estimate for the deputy is shown as a green triangle. The LOS measurements are shown as red arrows pointing from the chief to the deputy position at the time of the measurement. The IROD calculated IC for the deputy satellite was then propagated for the duration of the scenario and is plotted as a dashed green line. MATLAB's `vpasolve` function was used to numerically solve the IROD equations for the scenario, with a starting bound that was 2% away from the true IC. Overall, the range error for the deputy IC was 3.613 km and the speed error was 4.238×10^{-6} km/s. Note that further investigation into the resulting equations and a more advanced polynomial solution method may provide better results and allow the use of a much wider initial guess to numerically solve the IROD constraint equations. The main point of these initial results was to prove that the IROD equations could be solved.

IROD Example: Periodic Deputy

The second IROD example presented uses a deputy scenario computed from the natural periodic motion method described in Section 5.1. The chief orbit in this case is in a 3-2 resonant orbit. The

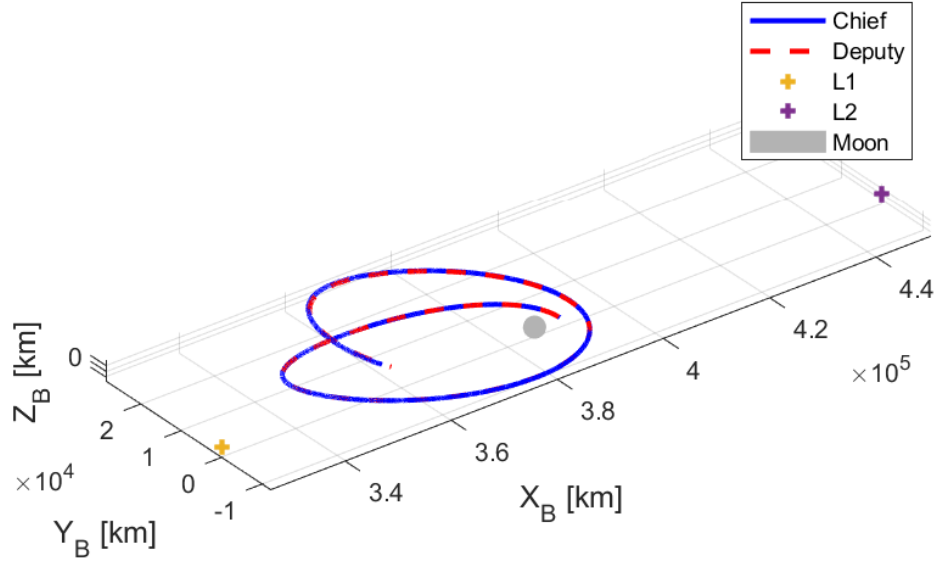


Figure 5.16 IROD scenario in L2 Lyapunov orbit showing deputy and chief position in the rotating \mathcal{B} -frame using CR3BP dynamics.

scenario is propagated for 6.75 days and the overall orbit geometry can be seen in Fig. 5.18. The initial scenario time is taken to be the time referenced for the deputy initial condition. Measurements are taken at 0.58, 2.89, and 6.02 days respectively. Table 5.5 gives the deputy and chief initial conditions in non-dimensional units in the \mathcal{B} -frame.

Table 5.5 Chief and deputy initial conditions for closed relative motion IROD scenario. Note that, initial conditions are given in non-dimensional units in the barycentric frame.

Initial Condition	Chief CR3BP IC	Deputy CR3BP IC
x_0 [DU]	0.689923223	0.689923205
y_0 [DU]	0	0.000449088
z_0 [DU]	0	0
\dot{x}_0 [DU/TU]	-9.18057E-16	-0.002602586
\dot{y}_0 [DU/TU]	0.562407482	0.564686794
\dot{z}_0 [DU/TU]	0	0

The scenario results are summarized by Fig. 5.19. Note the closed relative trajectory of the

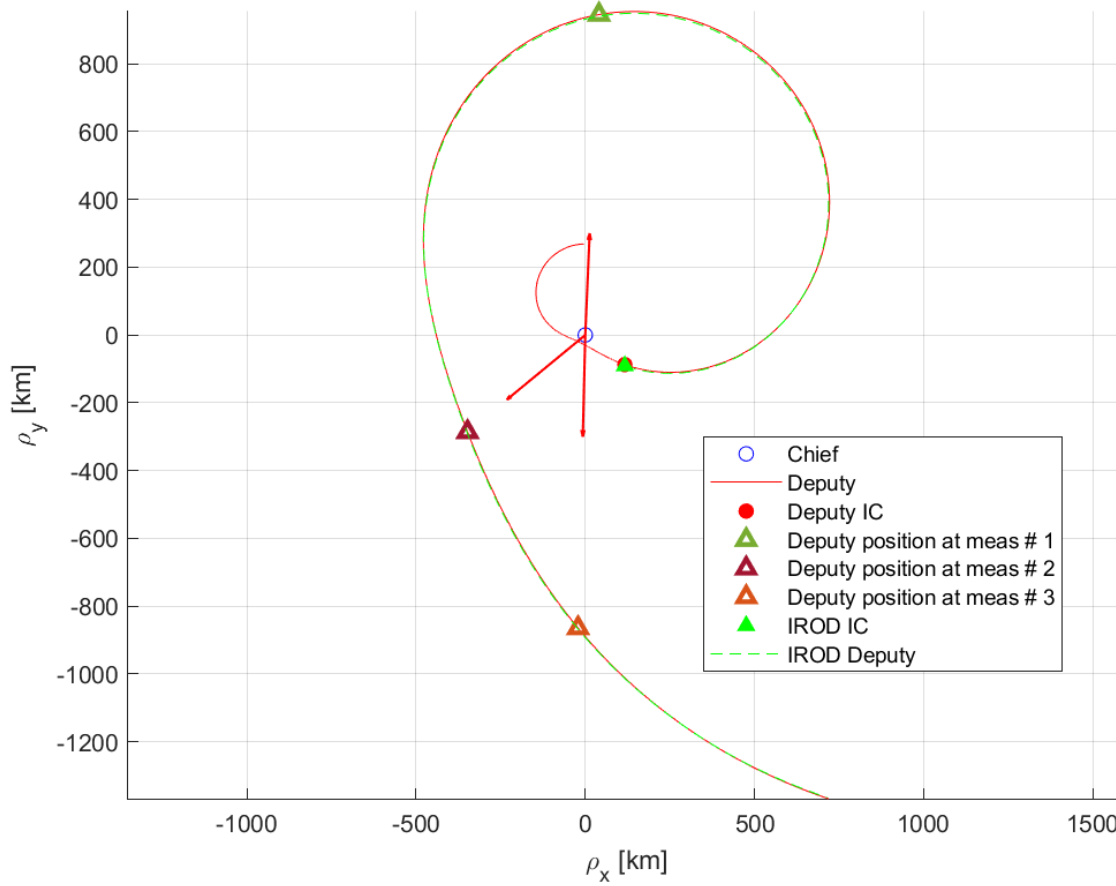


Figure 5.17 IROD scenario #1 results. Deputy IC estimated by IROD method is plotted near truth, and propagated to show the close alignment.

deputy, which returns to its initial position at the end of the scenario. The LOS measurements are shown as red arrows pointing to the deputy position at the time the measurement was taken. Note that this also is a 2D scenario. The IROD equations were solved with an initial guess range again of 2% from the true deputy IC. The IROD estimated IC had a range error of 0.0578 km and a speed error of 3.0429×10^{-5} km/s.

5.3.4 IROD Future Work and Summary

Several suggestions are made for future work on the IROD problem sketched out in the above sections. First the IROD constraint equations for the CR3BP are very poorly numerically scaled. Elements in the equations range greatly in magnitude from 1×10^1 to 1×10^{14} . One solution for this

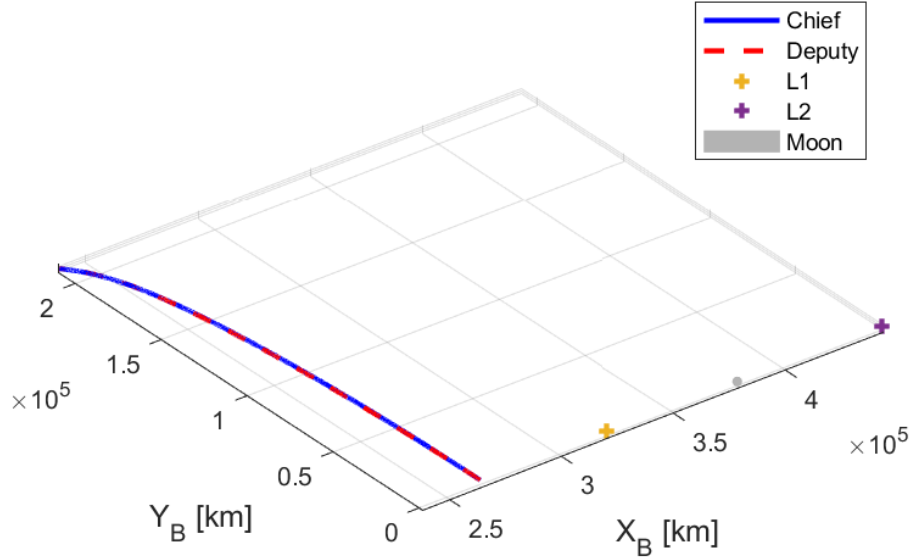


Figure 5.18 IROD scenario in a 3-2 resonant orbit showing deputy and chief position in the rotating \mathcal{B} -frame using CR3BP dynamics. Note that the chief orbit period is much longer than the 6.75 day scenario period.

may be numerical re-scaling of the equations. Another solution would be to use the non-dimensional form of CR3BP dynamics to develop the IROD equations. Another issue that must be addressed is the solution method for the polynomial equations to be solved. MATLAB's `vpasolve` only utilizes a standard Newton method and convergence is not guaranteed unless a relatively close guess to the true deputy IC is provided. A more robust equation solver should be applied to the IROD equations, ideally a method that does not require a close initial guess. Overall, it was shown that for the relative CR3BP the IROD problem can be solved. Some key advantages of the method are that it is based on analytical solution methods, and only requires three LOS measurements to work.

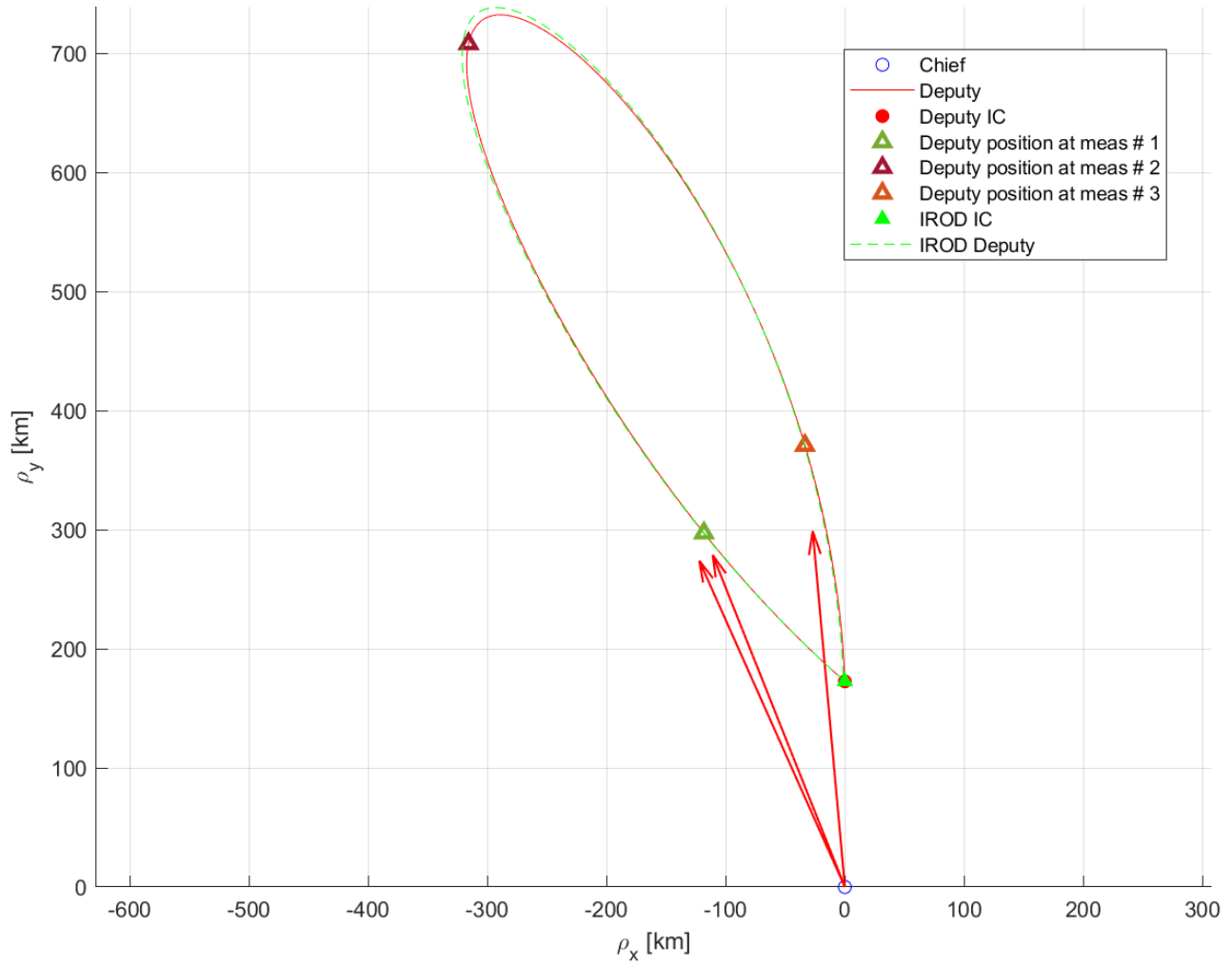


Figure 5.19 IROD scenario #2 results. The deputy is in a closed periodic trajectory, and the three LOS measurements are shown by red arrows pointing to the position of the deputy at the measurement time. The IROD solution is propagated and shown by a green dashed line.

6 Three-Body Orbit Determination Using Particle Swarm Optimization

This work seeks to provide a novel method of cislunar IOD using particle swarm optimization (PSO). Deputy and chief satellites are simulated under CR3BP dynamics for a variety of observer and target orbit geometries. Then a PSO is used to fit a set of observations to particle observations computed from propagating initial particle states forward to measurement times. It is shown that the PSO converges to an accurate initial state estimate for the deputy satellite. Parallel processing and GPU processing methods are utilized to speed computation time.

6.1 PSO Background

The Particle Swarm Optimization algorithm is a stochastic optimization method that is inspired by the societal and formation behavior of birds. The PSO method works by sharing information among members or “particles” of the swarm through a concept called “collective intelligence” [87–89]. A population based algorithm, the PSO works by initializing a set of M particles with an N dimensional position that represents potential state values. Each of these particles also has an associated velocity in the search space which controls the update of the particle positions. Additionally, each particle has an associated cost given its current position in the N dimensional search space. Note that, for the given application of cislunar IOD, each particle’s $N = 6$ dimensional “position” corresponds to a possible deputy initial position and velocity vector. The particles are weighted to balance exploration between their current search direction, their current “best” local position, and being pulled towards the current global optimal position. Each particle’s position update also includes a randomized search factor that adds to the stochastic nature of the PSO algorithm.

6.2 PSO Scenario CR3BP Dynamics

As a proof of concept that the PSO algorithm can be used for cislunar IOD, a scenario is defined using the non-dimensional CR3BP motion model assumptions. Figure 6.2 shows the geometric relationships of the barycentric synodic frame used to develop the CR3BP equations of motion. The system EOMs written in the \mathcal{B} -frame are given by Eqs. 6.1 - 6.3. Note that the non-dimensional

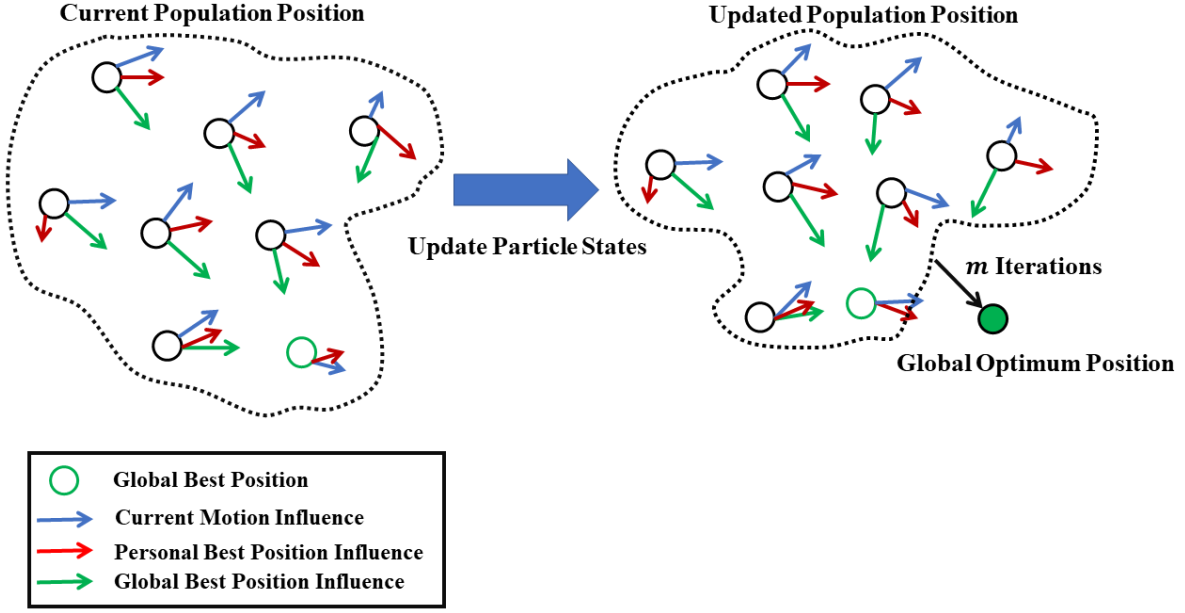


Figure 6.1 PSO operation overview. Particle population moves under the influence of current best positions (blue arrows), current motion (red arrows), and current global best position (green arrows). After m iterations the population converges to the global optimum solution.

form of the equations are shown here with the combined gravitational parameter $\mu = \frac{\mu_m}{\mu_m + \mu_e}$ where μ_m and μ_e are the gravitational parameters of the Moon and the Earth respectively.

$$\ddot{x} = 2\dot{y} + x - (1 - \mu) \frac{x + \mu}{((x + \mu)^2 + y^2 + z^2)^{3/2}} - \mu \frac{(x - (1 - \mu))}{((x - (1 - \mu))^2 + y^2 + z^2)^{3/2}} \quad (6.1)$$

$$\ddot{y} = -2\dot{x} + y - (1 - \mu) \frac{y}{((x + \mu)^2 + y^2 + z^2)^{3/2}} - \mu \frac{y}{((x - (1 - \mu))^2 + y^2 + z^2)^{3/2}} \quad (6.2)$$

$$\ddot{z} = -(1 - \mu) \frac{z}{((x + \mu)^2 + y^2 + z^2)^{3/2}} - \mu \frac{z}{((x - (1 - \mu))^2 + y^2 + z^2)^{3/2}} \quad (6.3)$$

6.2.1 Deputy Chief Scenario for Cislunar IOD

The scenario of interest is a deputy chief scenario in cislunar space. The chief's states are assumed to be known and available for the orbit determination process. The geometry of the scenario is shown in Fig. 6.2. The chief is assumed to be equipped with a sensor capable of capturing both range and angular measurements. In order for the PSO to work, an initial population of possible deputy states are required, and are seeded around the probable guess, shown by the red circles in

Algorithm 9 Particle Swarm Optimization Algorithm Pseudo-code

```
1: Initialize particle states:  $\mathbf{x}_i$  and  $\mathbf{v}_i$  for  $i = 1, \dots, m$ 
2:  $\hat{\mathbf{x}}_i \leftarrow \mathbf{x}_i$  and  $\hat{\mathbf{g}} = \min J(\mathbf{x}_i)$  for  $i = 1, \dots, m$ 
3: for  $n = 1$  to max iterations  $N$  do
4:   for  $i = 1$  to number particles  $m$  do
5:      $J_i \leftarrow J(\mathbf{x}_i)$  Find current cost of particle:
6:     Check if personal best cost:
7:     if  $J_i < J_{best_i}$  then
8:        $J_{best_i} \leftarrow J_i$ 
9:        $\mathbf{x}_{best_i} \leftarrow \mathbf{x}_i$  Update personal best state
10:    end if
11:    Randomly generate  $r_1, r_2 \in U[0, 1]$  for particle velocity update:
12:    Update particle velocity:  $\mathbf{v}_i \leftarrow \omega \mathbf{v}_i + \eta_1 r_1 (\mathbf{x}_{best_i} - \mathbf{x}_i) + \eta_2 r_2 (\hat{\mathbf{g}} - \mathbf{x}_i)$ 
13:    Update particle position:  $\mathbf{x}_i \leftarrow \mathbf{x}_i + \mathbf{v}_i$ 
14:  end for
15:  Check if new global best:
16:  if  $\min(J_i(\mathbf{x}_i)) < J(\hat{\mathbf{g}}_i)$  then
17:     $\hat{\mathbf{g}}_i \leftarrow \mathbf{x}_i$ 
18:  end if
19:  Check if tolerance met for convergence
20:  if  $g_{error} < \tau$  then
21:    break
22:  end if
23: end for
```

ρ , α , and β respectively. Particle states consist of a six-dimensional state vector containing a possible deputy initial position and velocity $\mathbf{x}_i = \begin{bmatrix} \mathbf{X}_0^T & \mathbf{V}_0^T \end{bmatrix}$. Thus each particle position is given by: $\mathbf{x}_i \in \mathbb{R}^6$. At each iteration of the PSO, each particle state is propagated forward in time using the CR3BP equations of motion (Eqs. (6.1) - (6.3)) to all measurement times (t_k). Next the relative position vector from the known chief position to the propagated particle deputy position is calculated as well as the LOS vector.

$$\boldsymbol{\rho}(t_k) = \mathbf{r}_{d_i}(t_k) - \mathbf{r}_c(t_k) \quad (6.4)$$

$$\mathbf{L}_{t_k} = \frac{\boldsymbol{\rho}(t_k)}{\|\boldsymbol{\rho}(t_k)\|} \quad (6.5)$$

Note that the vector $\mathbf{r}_{d_i}(t_k)$ denotes the position vector of the i th particle at time t_k . Measurements are then computed for each time-step as range azimuth, and elevation angles. The range is simply the norm of the relative position vector $\rho(t_k) = \|\boldsymbol{\rho}(t_k)\|$, and the angular measurements are calculated

from the LOS vector components as shown below.

$$\alpha(t_k) = \text{atan2} \left(\frac{\mathbf{L}_{t_k}(2)}{\mathbf{L}_{t_k}(1)} \right) \quad (6.6)$$

$$\beta(t_k) = \arcsin(\mathbf{L}_{t_k}(3)) \quad (6.7)$$

The parenthetic argument terms (1, 2, 3) denote the x, y, z component of the line of sight vector \mathbf{L}_{t_k} respectively. With all the elements defined, the measurement function is given by equation (6.8).

$$\mathbf{y}(t_k) = \begin{bmatrix} \rho(t_k) \\ \alpha(t_k) \\ \beta(t_k) \end{bmatrix} \quad (6.8)$$

For each time step, the measurement residuals \mathbf{b}_{t_k} are computed as the difference from the true measurement $\tilde{y}(t_k)$ and the particle predicted measurement $y(t_k)$ as shown below.

$$\mathbf{b}(t_k) = \tilde{y}(t_k) - y(t_k) \quad (6.9)$$

The measurement residuals for each time step are then squared and weighted by the expected sensor noise level and formed into a row vector. The measurement weight factors are defined as the inverse square of the sensor expected noise value $w_j = \frac{1}{\sigma_j^2}$, where $j = 1, 2, 3$ denotes range, azimuth, and elevation noise levels. Thus a row vector of residuals is formed as shown in Eq. (6.10).

$$\mathbf{b}_{total} = \left[\mathbf{b}(t_1)^T W \mathbf{b}(t_1) \quad \mathbf{b}(t_2)^T W \mathbf{b}(t_2) \quad \dots \quad \mathbf{b}(t_k)^T W \mathbf{b}(t_k) \right] \quad (6.10)$$

$$W = \begin{bmatrix} w_1 & 0 & 0 \\ 0 & w_2 & 0 \\ 0 & 0 & w_3 \end{bmatrix} \quad (6.11)$$

Once all measurement residuals are formed into a row vector, the final scalar cost from the

current particle is calculated as the the square of the row vector as:

$$J = \mathbf{b}_{total}^T \mathbf{b}_{total} \quad (6.12)$$

Once the cost is calculated for a given particle, the current cost is compared to the particles personal best (lowest) cost. If the current cost is lower than the personal best, then the particle's personal best state is updated and used for the update in the particle states as outlined in algorithm 9.

6.2.3 PSO Tolerance for Ending Conditions

In order to reduce run-time an ending condition tolerance was implemented. The best global cost is formed into a vector J_n where the global best cost is appended to the vector after each iteration of the PSO. After p iterations, the elements of J_n are compared to determine if the global best cost has changed over the last p iterations. Experimental testing found that a value of $p = 5$ to $p = 10$ provided good results while balancing run-time and PSO performance in finding the optimal state. The user must choose a specific tolerance τ_{stop} to define how much of a change in global cost is desired to prematurely stop the PSO algorithm.

Algorithm 10 Tolerance Algorithm Pseudo-code

```

1: Initialize global best matrix  $J_n$ 
2: for  $n = 1$  to max iterations  $N$  do
3:   Append current global best to global best matrix
4:   if  $n > p$  minimum runs for algorithm then
5:     Determine error between current global cost and global cost 5 iterations prior:
6:      $K = |(J_n - J_{n-p})/J_n|$ 
7:   end if
8:   if  $K < \tau_{stop}$  is met then
9:     Run terminates
10:  else  $\tau_{stop}$  not met
11:    Run continues
12:  end if
13: end for

```

6.2.4 Local Minimization

The PSO is a metaheuristic global optimizer and is very good at discovering minimum in a global sense. For multi-minimum problems however, particles can get “stuck” in a local minimum.

A common method to increase accuracy is to use a local minimizer after a global optimization has been run to get the states as close as possible to the desired states. The method chosen for local minimization is non-linear least squares using the MATLAB implementation of the Levenberg-Marquardt (LM) method [90–92]. The non-linear least-squares problem is setup to minimize the difference between true and predicted measurements from the estimated state of the deputy satellite. The LM method seeks to minimize a function of the form:

$$\min_{\mathbf{x}} f(\mathbf{x}) = \|\mathbf{F}(\mathbf{x})\|_2^2 = \sum_i^N F_i^2(\mathbf{x}) \quad (6.13)$$

where the vector \mathbf{F} is given as the residuals for a set of measurements, which are subsequently squared in order to minimize the error between the truth measurements and the estimated measurements. The objective function is a slightly modified version of the cost function employed for the PSO, where the residuals are formed as as a row vector in the following form.

$$\mathbf{F} = \begin{bmatrix} J_1 & J_2 & \dots & J_k \end{bmatrix} \quad (6.14)$$

The elements of \mathbf{F} are given by a slightly modified version of the cost function employed for the PSO. Instead of summing all measurement residuals and then computing the weighted square, each measurement residual is computed and squared with the measurement weights. Each individual cost element then takes the form shown below.

$$J_k = \mathbf{b}^T(t_k) W \mathbf{b}(t_k) \quad (6.15)$$

The measurement residual is given by: $\mathbf{b}(t_k) = \tilde{y}(t_k) - y(t_k)$ at time t_k . The output of the non-linear least squares minimization is an initial state for the deputy that has now moved closer to the truth than the result of the PSO. Simulation results showed that using the non-linear optimization after running the PSO was shown to reduce the final cost significantly and the error in the IOD estimate. Figure 6.3 shows the effect of increasing iterations on the accuracy of the solution, and the advantage

of using LSQ to improve an initial PSO solution. After a certain point in iterations, the PSO solution approaches the LSQ solution found. However, LSQ can be computationally more efficient, so to reach an acceptable error level sooner, the best PSO solution after some amount of iterations can then be passed to the LM LSQ algorithm for improvement without incurring the added run-time of more PSO iterations.

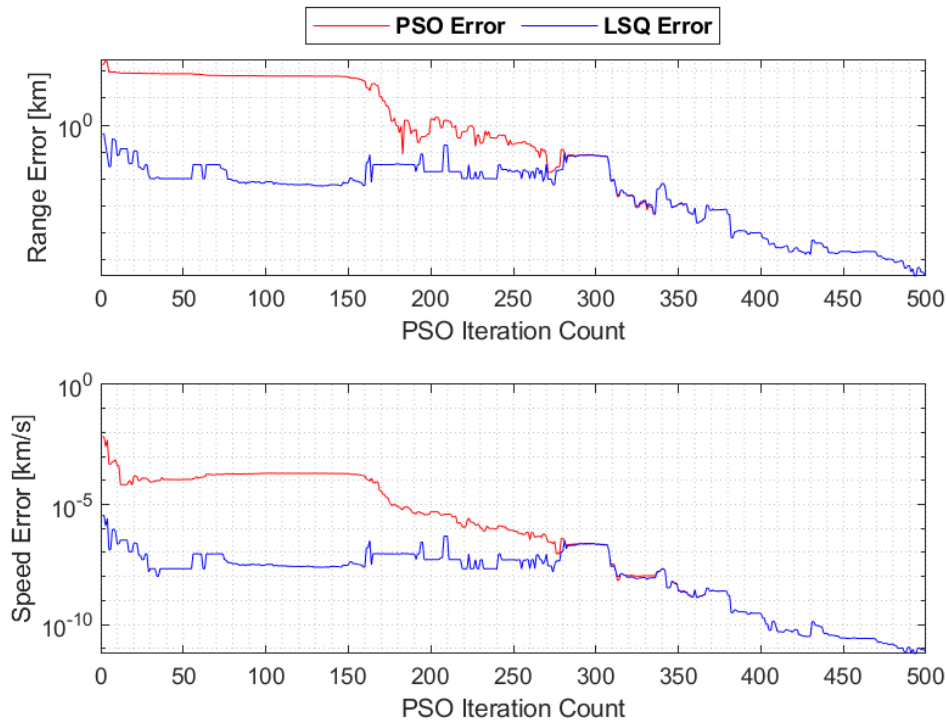


Figure 6.3 Range and speed error reduction with increasing iterations. After a certain point the PSO error meets the LSQ error. Before that point there is an advantage of using LSQ on the PSO current solution.

6.3 Python vs. MATLAB

The PSO algorithm for cislunar IOD defined in Sec. 6.2 was implemented in both MATLAB and Python environments to make use of both the computational power of MATLAB, and the open source nature of Python code. In terms of accuracy, a difference was not noted for the PSO. A difference in accuracy was observed in the non-linear least-squares local minimization, most likely because the NLLS algorithm used in Python is not the same as the LM method used in MATLAB.

6.4 MATLAB PSO Results for CR3BP IOD

Results are shown for running the IOD method for the CR3BP on two scenarios for deputy and chief satellites operating on L2 periodic orbits of both HALO and axial kinds. The initial particle positions and velocities for the deputy were bounded by a Gaussian sphere of 250 km and 0.1 km/s respectively. Initial particles were randomly seeded around the true deputy position using MATLAB's built-in `rand` function and scaled appropriately by the limits in position and velocity.

The results of both scenarios are summarized in Table 6.1. Both scenarios had the deputy in an L2 southern HALO orbit with a period of 7.15 days. The range error for scenario 1 was sub-kilometer for the PSO results. Note that the non-linear least squares (NLSQ) local optimizer was able to achieve near meter level accuracy for range and sub m/s speed accuracy.

Table 6.1 Summary of PSO and nonlinear least squares (NLSQ) results. In both cases NLSQ produced a lower error after PSO terminated.

	Scenario 1		Scenario 2	
	PSO	NLSQ	PSO	NLSQ
Range Error [km]	0.3276	0.0304	6.072	0.0177
Speed Error [km/s]	2.84E-06	3.40E-08	3.27E-05	5.77E-08
Minimum Cost	0.0493	8.56E-11	1.667	7.34E-11

6.4.1 Scenario 1: HALO Orbiting Deputy and Chief

Scenario 1 is a cislunar SDA scenario where the deputy is in a HALO orbit with a period of 7.15 days, and the chief is placed in a HALO orbit with a period of 13.8-days. The initial conditions used for the scenario are shown in Table 6.2. The scenario was propagated for 7 days, and a total of 35 measurements were taken, equally spaced through the total simulation time-span.

Figure 6.4 shows the scenario propagated for 7-days. The chief orbit is shown, and the deputy orbit and the orbit computed by the PSO is also plotted. Note the close agreement and the position overlapping of the PSO and truth comparison positions.

Figure 6.5 shows the cost value for all iterations that the PSO ran through. Note the distinctive extended staircase pattern as the optimizer seeks the global minimum, and that the maximum number of iterations was hit, indicating that the tolerance was not met.

Table 6.2 Scenario 1 initial conditions given in non-dimensional units in the barycentric frame. The scenario was propagated for 7 days and utilized 35 measurements.

	Chief IC	Deputy IC
x_0 [LU]	1.140135389	1.029726968
y_0 [LU]	0	0
z_0 [LU]	-1.63176653574390E-01	-1.869397163946E-01
\dot{x}_0 [LU/TU]	6.13321115086310E-15	-5.585615805585E-14
\dot{y}_0 [LU/TU]	-0.223383154	-0.119441863
\dot{z}_0 [LU/TU]	1.78644826151404E-15	-9.803996218373E-13
Jacobi Constant [LU ² /TU ²]	3.06	3.04
Period [days]	13.8	7.15

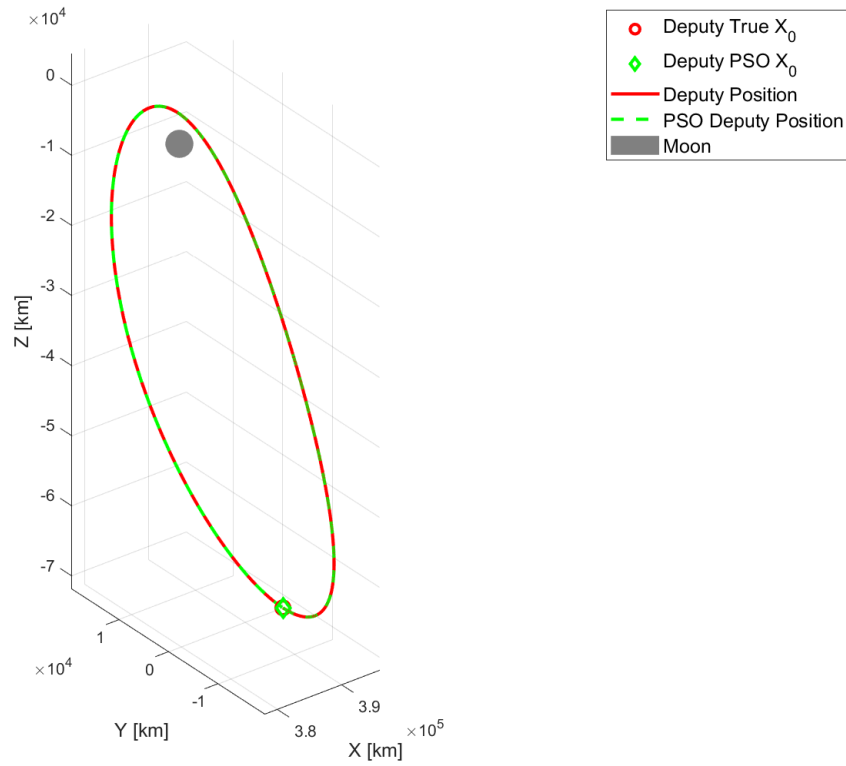


Figure 6.4 Scenario 1: Chief and deputy HALO orbits depicted over 7-day period with PSO deputy orbit overlay.

Figures 6.6 and 6.7 show the initial distribution of particles in position and velocity. Note that a majority of particles converge around the true initial position and velocity states, and that the local minimizer lies extremely close to the true state.

Figure 6.8 shows the position and velocity error of deputy initial conditions as the PSO converges

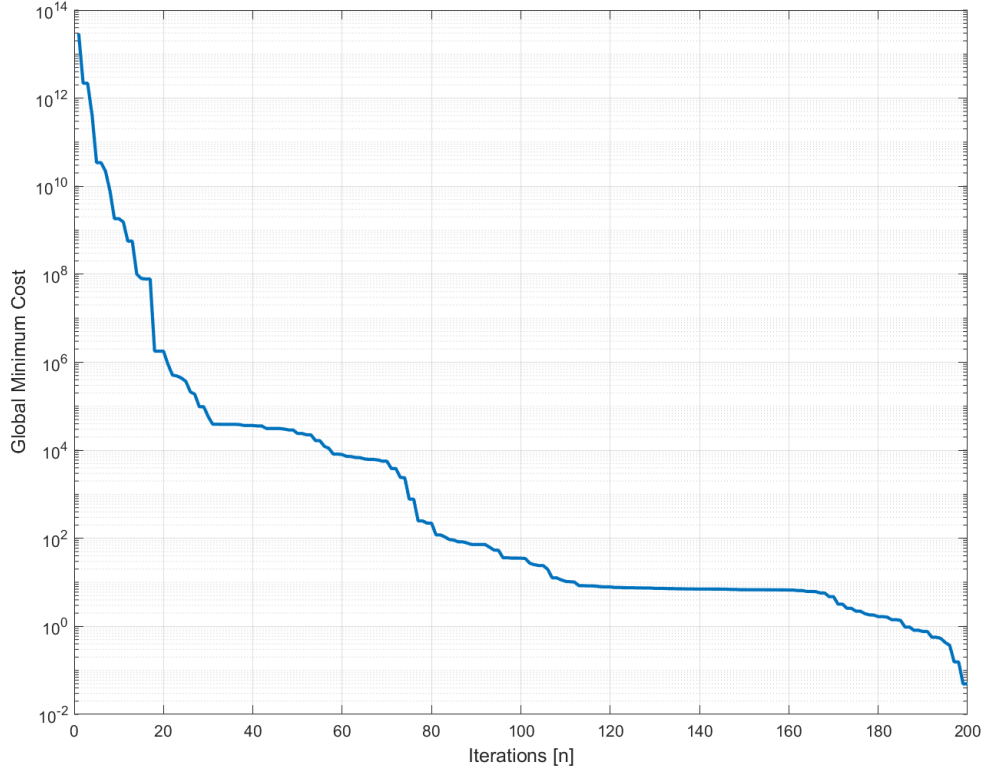


Figure 6.5 Scenario 1: Global cost per iteration, decreasing until iteration counter is reached or tolerance is met.

towards the truth. Note that, for the first few iterations, there are large fluctuations in the error before a general decreasing trend is observed. The errors in position are summarily larger than the errors in velocity. Similarly, Fig. 6.9 shows the range and speed error of the initial state estimate as the PSO converges.

Overall, it can be seen that the PSO was able to determine a relatively close state estimate, and the local non-linear least squares optimizer was able to further refine that state error to within a few meters in position and sub-meter per second in velocity.

6.4.2 Scenario 2: Chief Axial Orbit

The second scenario is another cislunar SDA type application. The deputy is in the same orbit as scenario 1, but this time the chief is in an L2 Axial orbit with a period of 19.1 days. Much sparser measurements were utilized, with a total of 10 measurements taken over the 7 day propagation period. The PSO and NLSQ errors were both very low, with the PSO achieving a range error of < 7

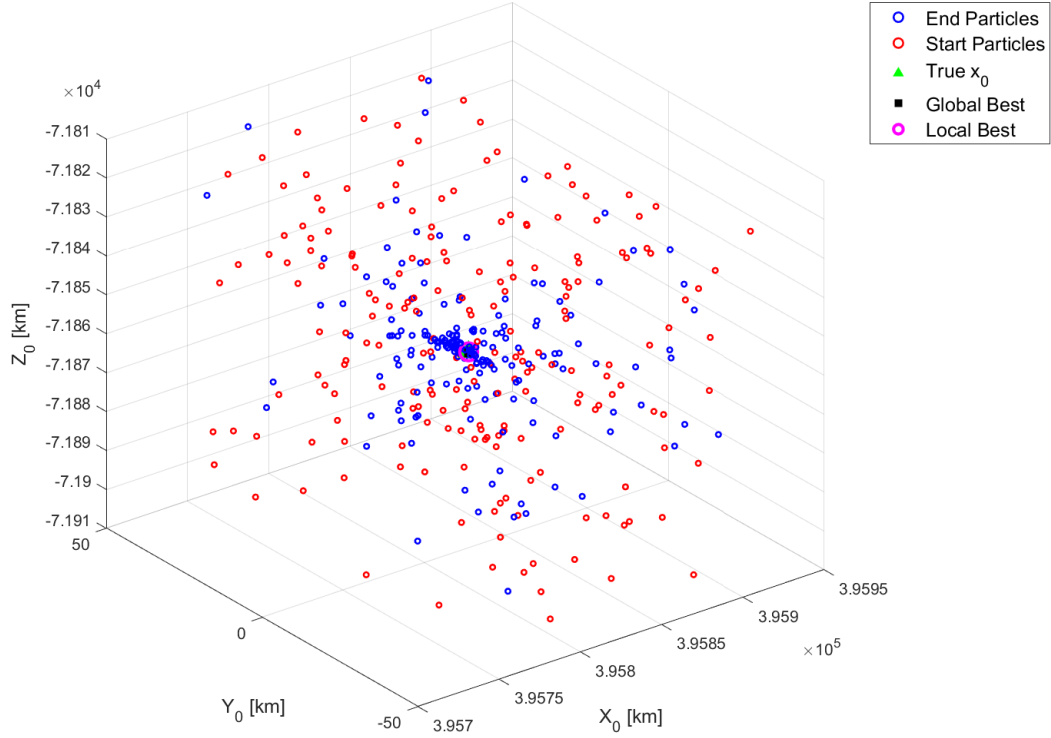


Figure 6.6 Scenario 1: Initial (red) and final state (blue) particle positions in three axes, swarming to global best.

km and a speed error of < 0.05 m/s. The initial particles for PSO were seeded with error bounds of 250 km in position and 0.1 km/s in velocity from the true deputy position. Overall results are very similar to scenario 1, and follow the same trends. Figure 6.10 shows the results of the simulation,

Table 6.3 Scenario 2 initial conditions given in non-dimensional units in the barycentric frame. The scenario was propagated for 7 days and utilized 10 measurements.

	Chief	Deputy
x_0 [LU]	1.21996614837886	1.02972696832499
y_0 [LU]	0	0
z_0 [LU]	-2.49454877925228E-16	-1.86939716394581E-01
\dot{x}_0 [LU/TU]	1.10859609358602E-15	-5.6E-14
\dot{y}_0 [LU/TU]	-4.27475884185211E-01	-1.19441862578254E-01
\dot{z}_0 [LU/TU]	4.09809301627323E-03	4.09809301627323E-03
Jacobi Constant [LU ² /TU ²]	3.01	3.04
Period [days]	19.11	7.15

with the propagated PSO solution for the deputy overlaid on the true deputy position. The chief's

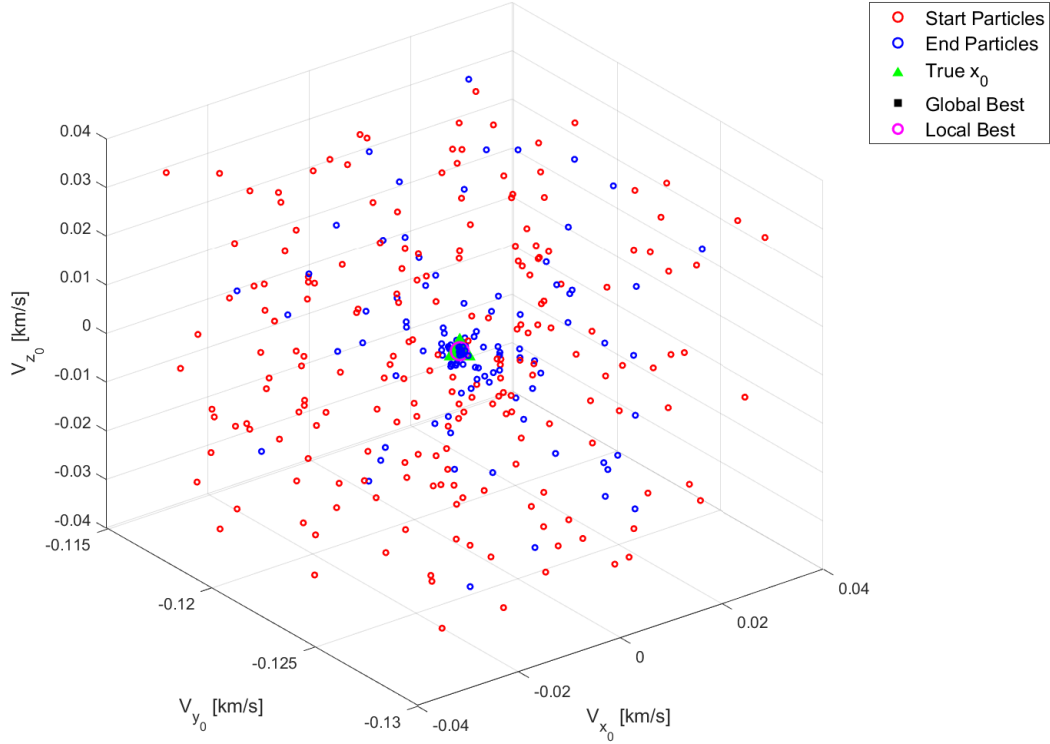


Figure 6.7 Scenario 1: Initial (red) and final state (blue) of particle velocity in three axes.

orbit is a wide, planar orbit, and the deputy is in a periodic HALO orbit. Note that the remaining figures for the second scenario are located in Appendix A for compactness. Overall the results are analogous to scenario 1.

Figure 6.11 shows the global minimum cost as the PSO population moves towards the optimal solution. Note that at times the cost appears to only be moving downwards slowly, but with the logarithmic scale the progress is faster than it appears. Figure 6.12 shows the initial and final particle distributions for the initial position of the deputy. Due to scale, the global best and true solution at the center are difficult to see, but there is a collection of particles clustered on the truth. Figure 6.13 shows the initial and final distributions of velocity particles. Figure 6.14 shows the error in initial position and velocity as the PSO converges to the truth. Velocities converge more quickly than the position states. Figure 6.15 shows the range and speed errors as the PSO converged.

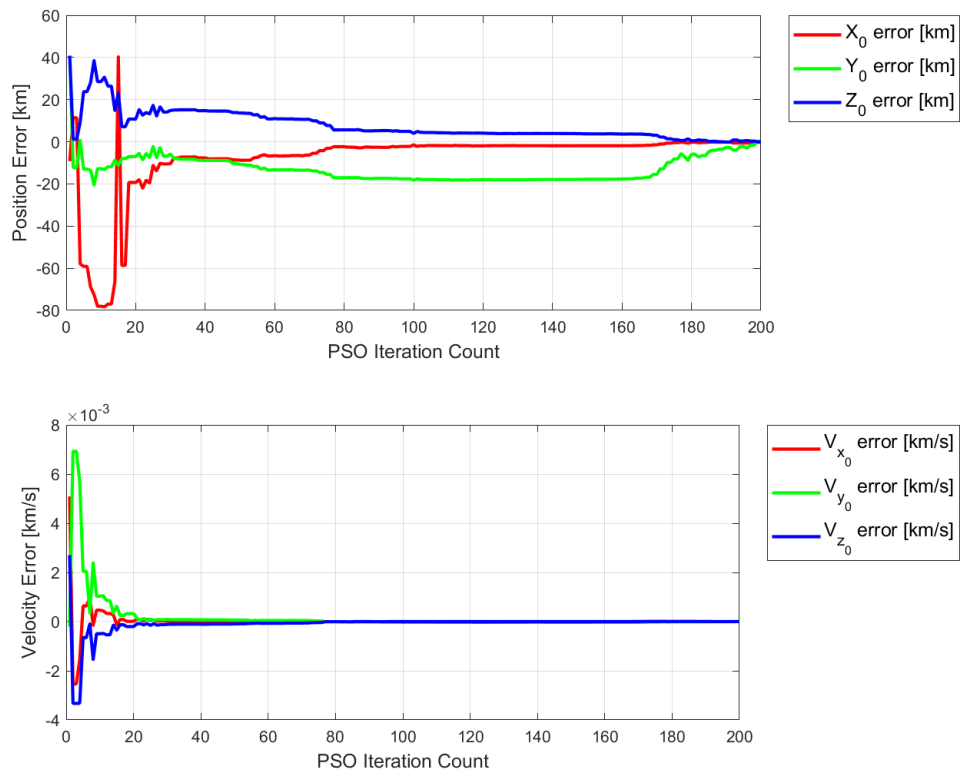


Figure 6.8 Scenario 1: Position error in each axis per iteration (top) and velocity error in each axis per iteration (bottom).

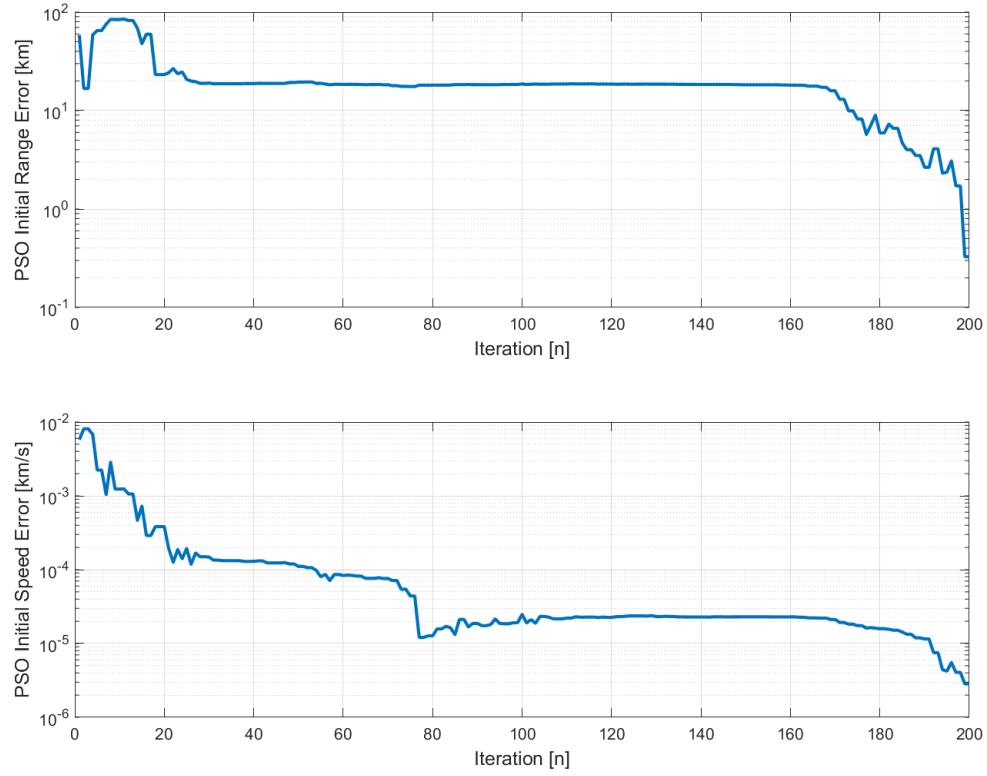


Figure 6.9 Scenario 1: Magnitude of position error per iteration (top) and magnitude of velocity error per iteration (bottom).

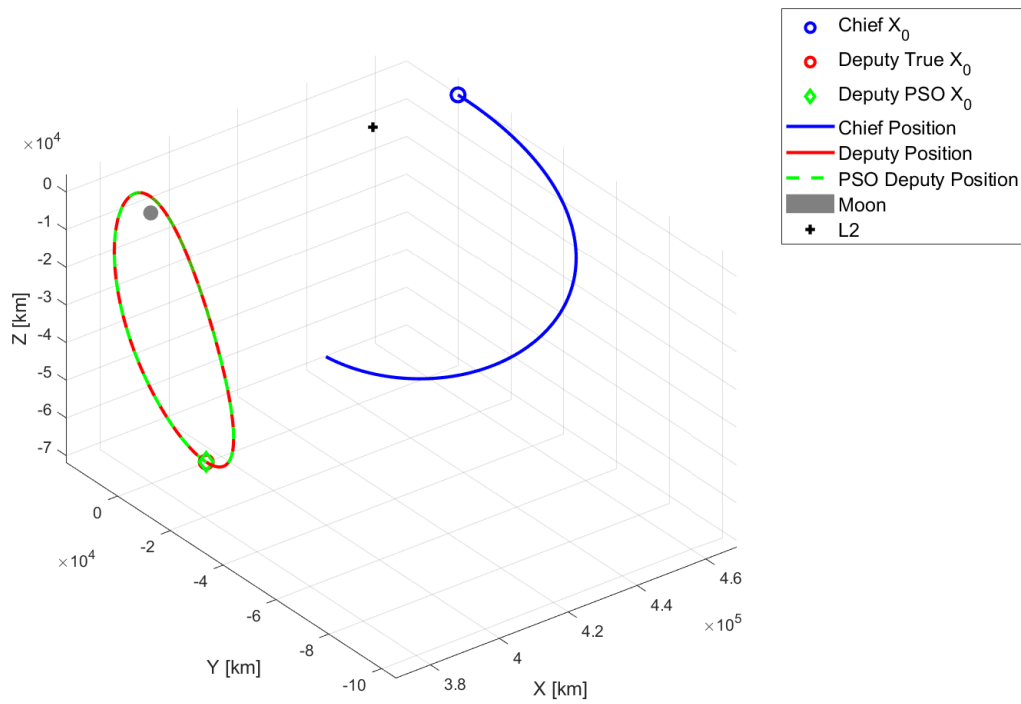


Figure 6.10 Scenario 2: Chief and deputy HALO orbits depicted over 7-day period with PSO deputy orbit overlay.

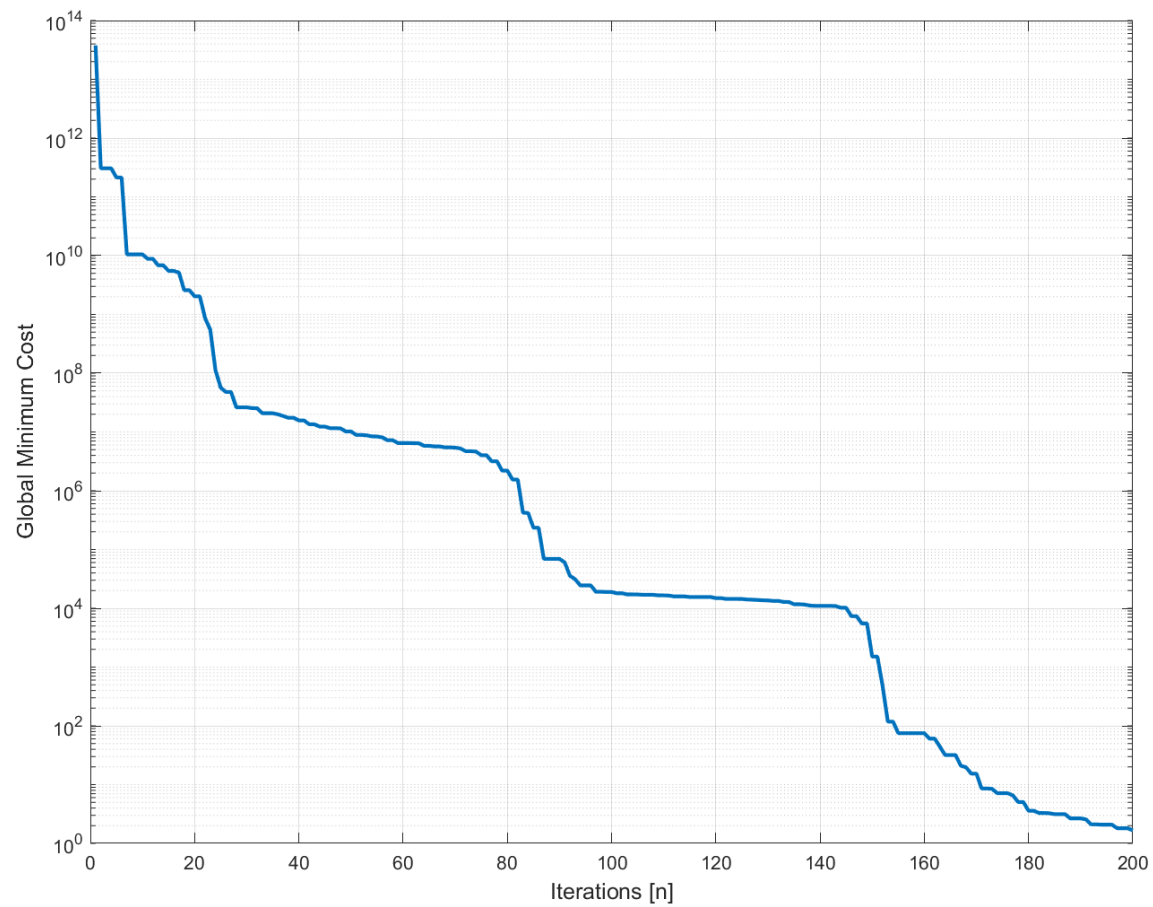


Figure 6.11 Scenario 2: Global cost per iteration, decreasing until iteration counter is reached or tolerance is met.

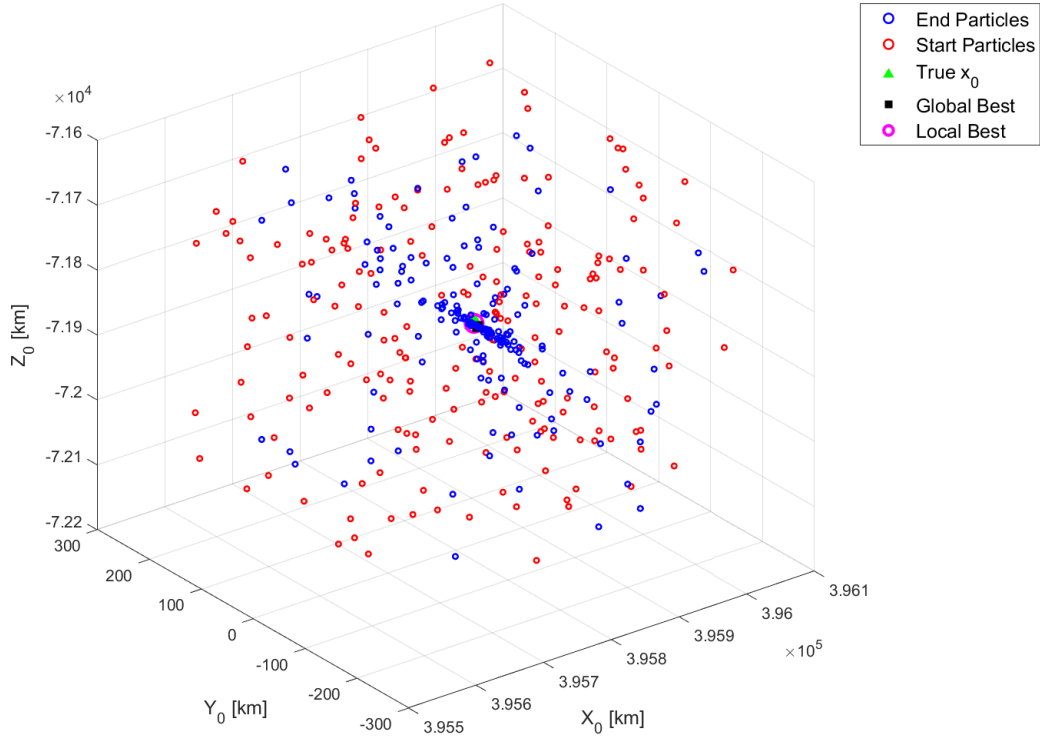


Figure 6.12 Scenario 2: Initial (red) and final state (blue) particle positions in three axes, swarming to global best.

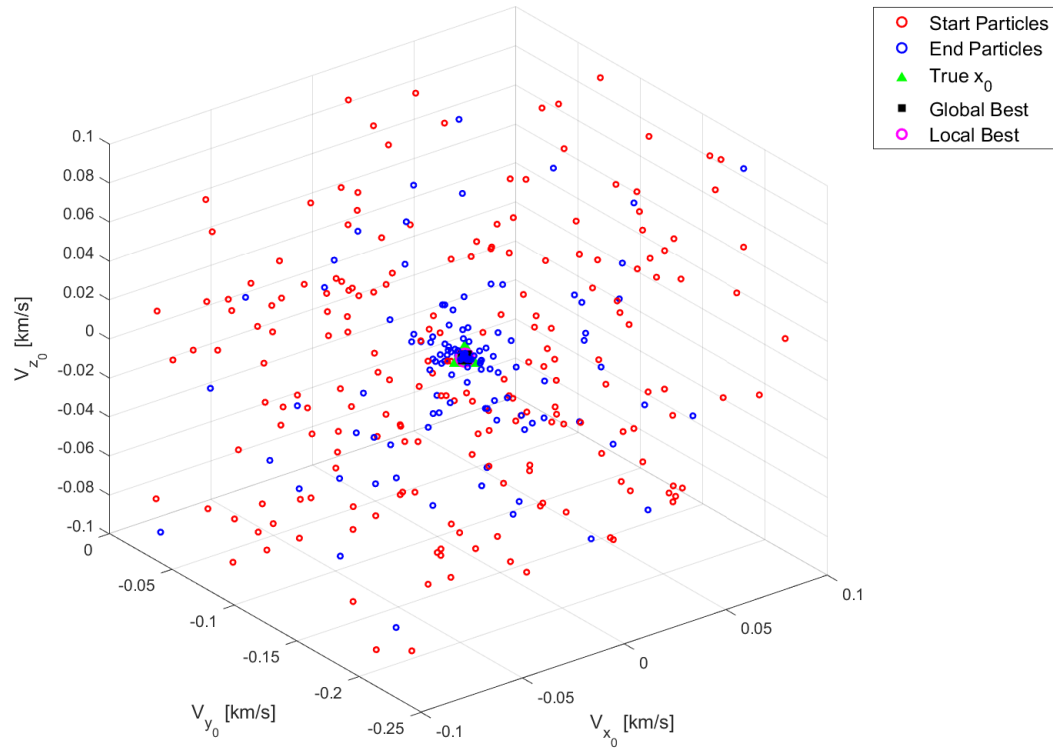


Figure 6.13 Scenario 2: Initial (red) and final state (blue) of particle velocity in three axes.

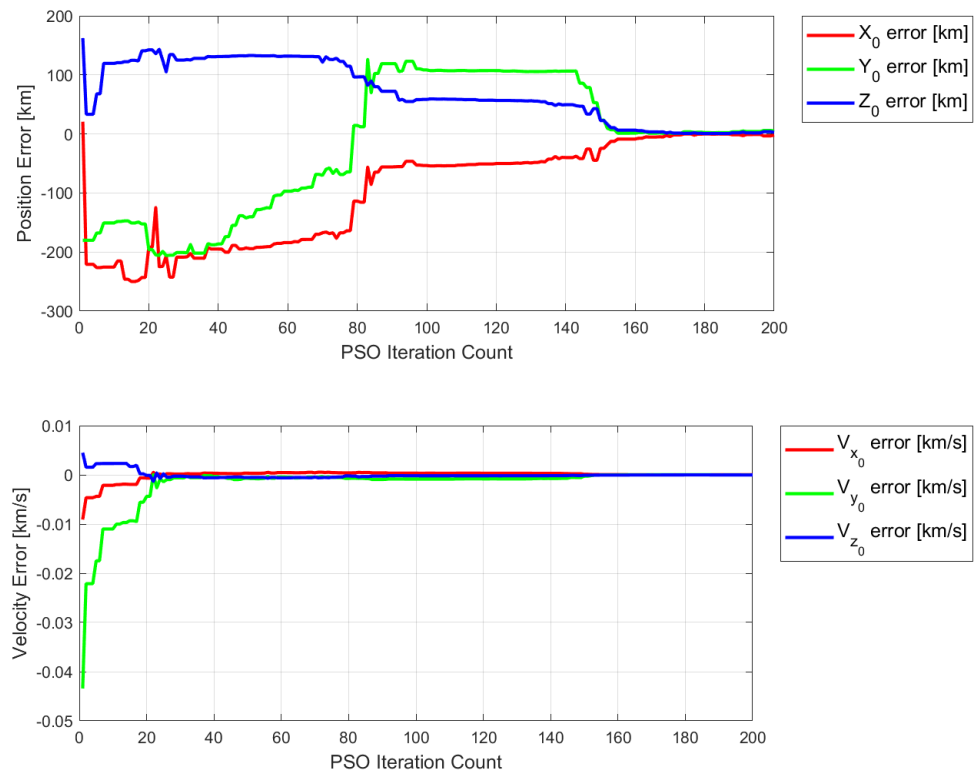


Figure 6.14 Scenario 2: Position error in each axis per iteration (top) and velocity error in each axis per iteration (bottom).

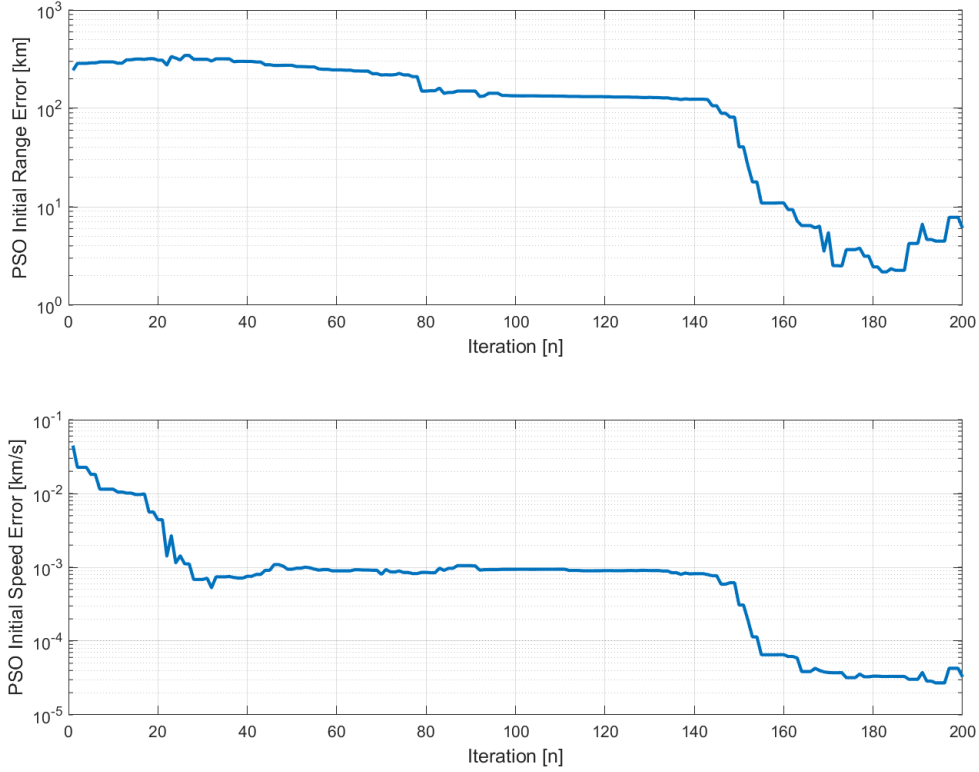


Figure 6.15 Scenario 2: Magnitude of position error per iteration (top) and magnitude of velocity error per iteration (bottom).

6.5 Python Results

The Python implementation of the PSO algorithm outlined in Section 6.2 was used to test an angles only IOD simulation in CR3BP dynamics. The PSO algorithm was initialized with bounds on position of 100km in any direction and an error on initial velocity of 0.1 km/s. Angular measurements were simulated with Gaussian white noise added with a standard deviation of 4 arcseconds. The point of this scenario was to prove two things; first that the PSO algorithm could function using angles only measurements, and secondly that the PSO algorithm could still converge in the face of realistically noisy measurements. The noise value of 4 arcseconds matches observed maximum measurement errors on the optical system described in Section 3.2.4. The deputy was placed in an L2 southern HALO orbit with a period of 7.2 days, and the chief was placed in a southern L2 HALO orbit with a period of roughly 14 days. The scenario was propagated for 7 days and measurements were assumed to be available roughly every 500 seconds for a total of 1200 measurements. The PSO was initialized with 500 particles and ran a maximum of 2500

iterations. After the PSO iterations were completed, the best PSO state was then processed through the non-linear least squares algorithm. The PSO and LSQ final errors are summarized in 6.4. Notably the LSQ algorithm reduced the range and speed errors by an order of magnitude over the PSO.

Table 6.4 Python PSO scenario error results for estimating deputy initial condition.

Error	Error Amount	Units
Range Error	5.2141E+01	km
Speed Error	2.2961E-04	km/s
x_0 Error	-1.5796	km
y_0 Error	-4.42366e+01	km
z_0 Error	-2.7556E+01	km
\dot{x}_0 Error	-1.6397E-04	km/s
\dot{y}_0 Error	6.9542E-05	km/s
\dot{z}_0 Error	1.4490E-04	km/s
LSQ Range Error	8.5057	km
LSQ Speed Error	2.6052E-05	km/s

The overall geometry of the scenario is shown in Fig. 6.16, where the true deputy and chief propagated positions are shown as well as the PSO best estimated position propagated for the scenario duration. Because of the close alignment with the truth, the truth deputy and PSO estimated deputy position lie on top of each other, with the yellow solid line representing the true deputy position and the dashed green line representing the propagated PSO best estimate. The global minimum cost as iterations increase is shown in Fig. 6.17. The cost decreases dramatically for the first 1200 iterations of the PSO algorithm and then levels off for the rest of the scenario. The PSO best estimate of the initial X_0, Y_0, Z_0 positions is plotted in Fig. 6.18. As the PSO algorithm searches the entire initial bounds, the best estimate oscillates around the truth until it settles down to the relatively close values shown in the plot. Figure 6.19 shows the PSO best estimate for the initial velocity states in each axis as the algorithm iterates. The oscillations in the velocity estimates are still present but less pronounced than for the position states.

Figure 6.20 shows a 3D view of what the PSO search space for the position states looks like at the beginning and the end of the PSO algorithm's run-time. The PSO initial population is shown in

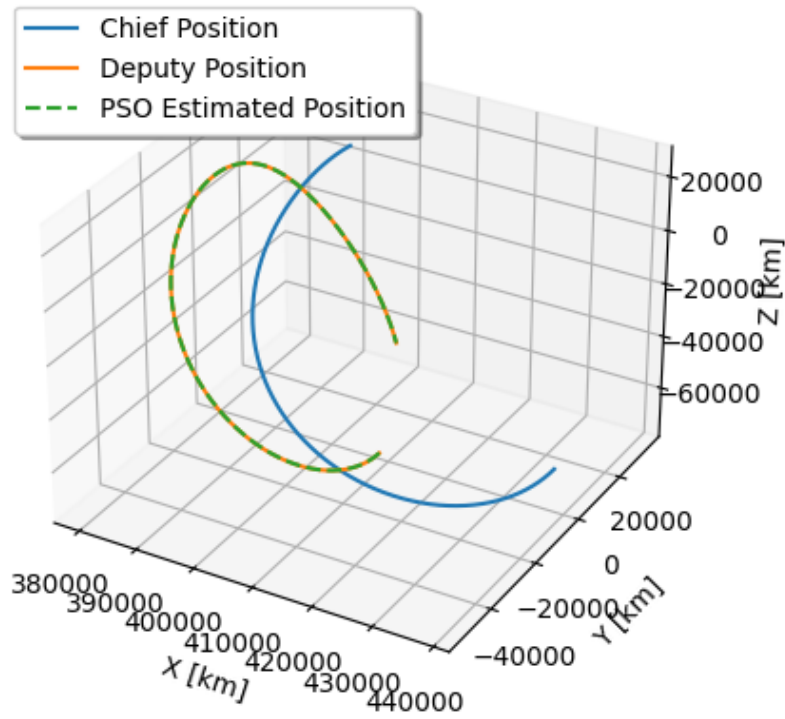


Figure 6.16 Python PSO scenario results. Deputy and chief positions propagated for duration of scenario.

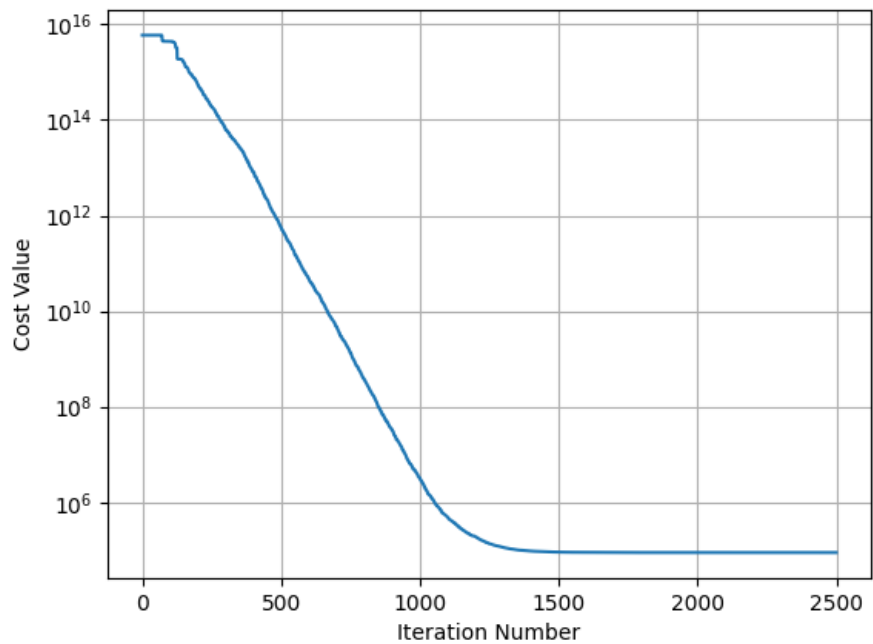


Figure 6.17 Python PSO scenario results. Global minimum cost plot.

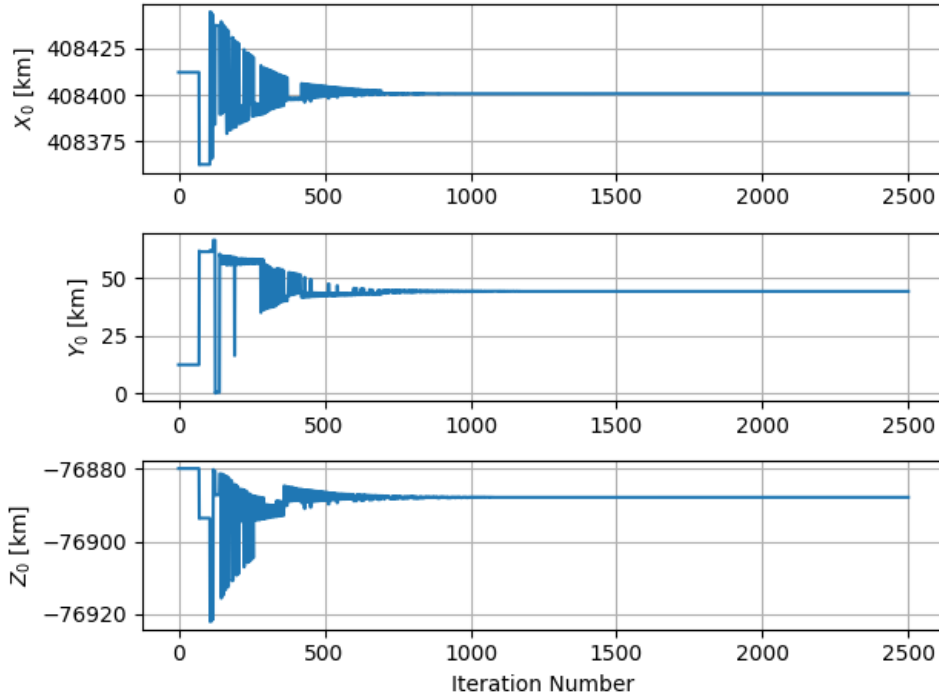


Figure 6.18 Python PSO scenario results. Global best estimate of deputy initial position.

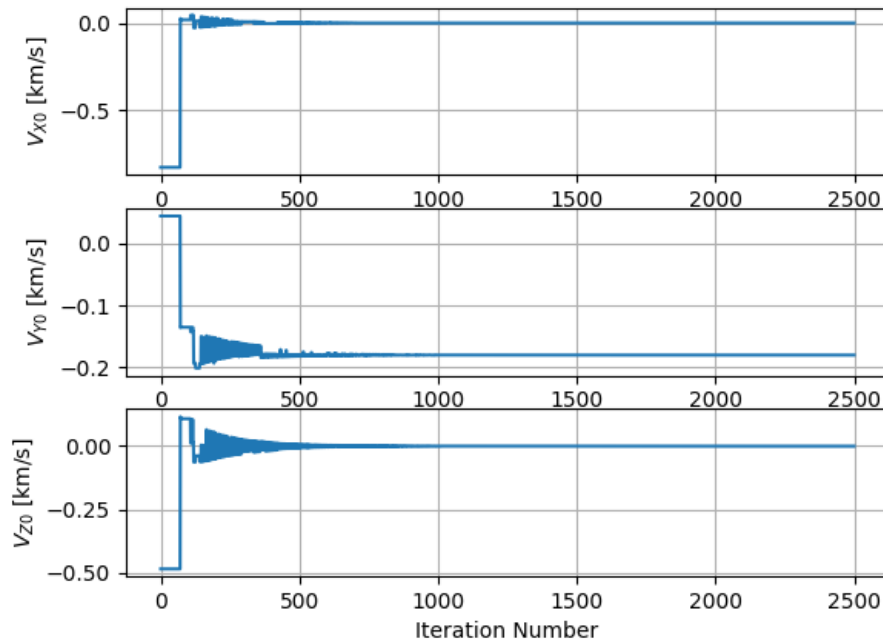


Figure 6.19 Python PSO scenario results. Global best estimate of deputy initial velocity.

blue, and the final population in red. The true and PSO best states are contained under the mass of converging red symbols for the final population. A similar situation is shown in Fig. 6.21 for the

velocity search space.

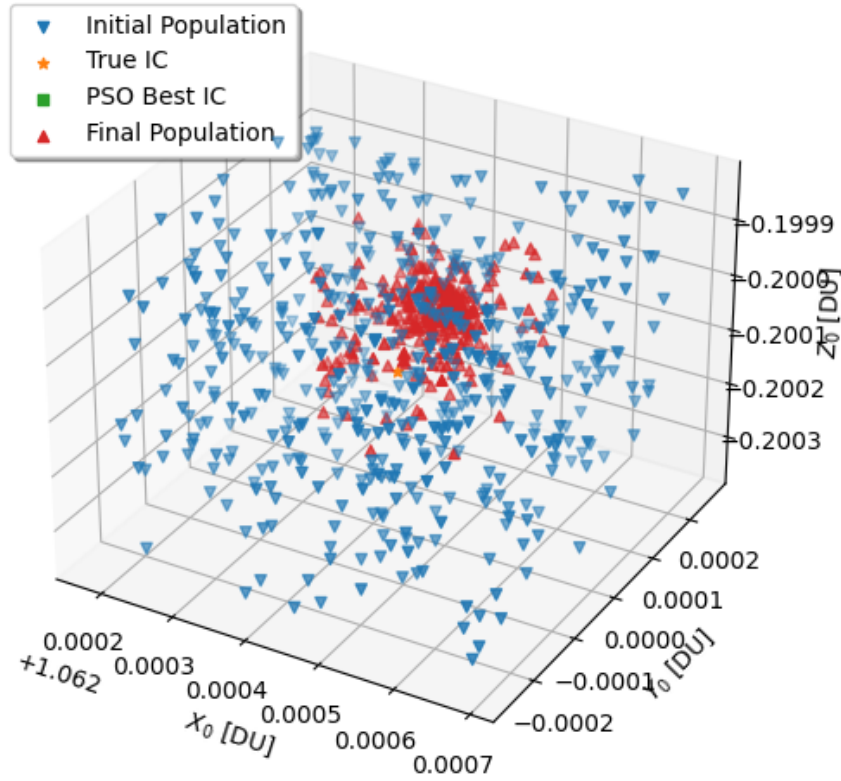


Figure 6.20 Python PSO scenario results. Starting and ending particles for deputy position states in non-dimensional units.

6.6 PSO for CR3BP IOD Conclusions

Particle swarm optimization shows promise for providing IOD results for cislunar SDA applications. Scenarios tested were representative of possible SDA tasks in cislunar space with the goal of obtaining a usable initial state guess for the deputy satellite to initiate proximity operations or enter a satellite into a catalog of cislunar objects for further tracking. PSO global optimization was able to get an initial state estimate that was close enough to seed a local optimizer. Further optimization through nonlinear least squares then found the initial state with much greater accuracy. Further improvement in the initial condition for PSO, and investigating varying weighting parameters is a subject of future investigation to improve the IOD results.

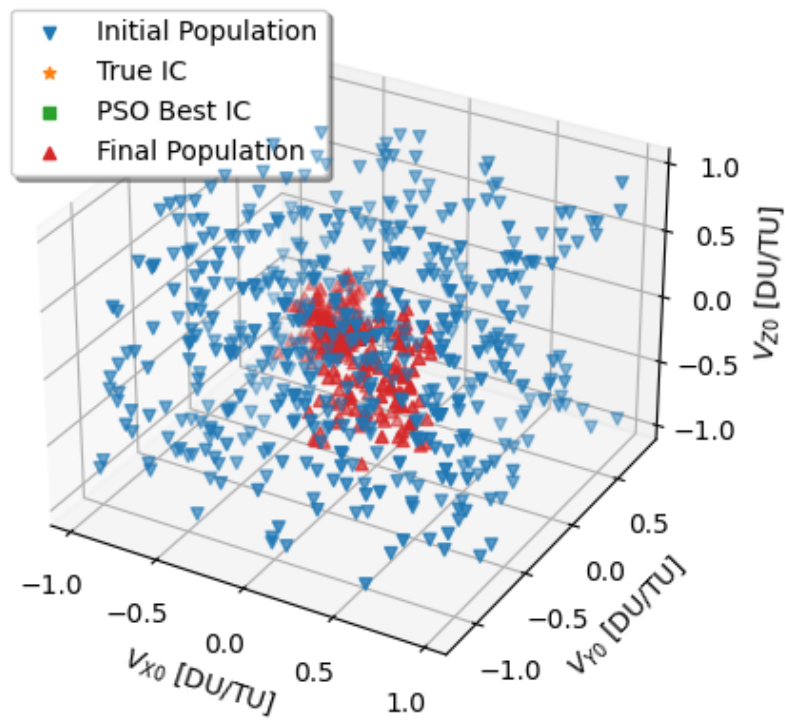


Figure 6.21 Python PSO scenario results. Starting and ending particles for deputy velocity states in non-dimensional units.

7 Conclusions and Recommendations

A summary of the conclusions and recommendations for future work resulting from this body of research are presented in this chapter. The goal of this dissertation was to enhance knowledge towards achieving truly autonomous SSA capabilities for both Earth orbiting and cislunar orbiting satellites. In order to achieve true autonomous SSA for any domain a number of specific capabilities are required and outlined below.

1. The ability to process measurements of spacecraft such as optical images.
2. The ability to use measurements to form an initial orbit solution.
3. The ability to process batches of measurements for a precise orbit solution.
4. The ability to accurately model the dynamics involved for any orbit determination scenario.

The algorithms outlined in this dissertation seek to meet the stated capabilities and have been shown to achieve these goals for a variety of Earth-orbiting and cislunar scenarios.

7.1 Research Conclusions

The specific contributions developed in this research are summarized in this section with the list given below.

1. Developed template matching and optical flow algorithms for satellite detection in unresolved space images (Sections 3.2 and 3.3).
2. Developed a set of linearized relative three-body equations of motion with a matrix exponential solution (Chapter 4).
3. Developed application of linearized relative three-body dynamics for finding closed natural motion trajectories in the relative CR3BP (Section 5.1).
4. Developed and proved feasibility of performing three-body initial relative orbit determination (Section 5.3).
5. Developed a particle swarm optimization method for CR3BP orbit estimation (Chapter 6).

The research contributions meet the objectives defined for this research. The ability to process images and detect RSOs was shown using the template matching and optical flow algorithms. The ability to use measurements for forming an initial orbit estimate was demonstrated for both Earth-

orbiting and cislunar scenarios. For Earth-orbiting scenarios a UKF and a batch least squares filter were used to find an accurate orbit estimate of an observed satellite. For the cislunar case, the PSO algorithm presented in Chapter 6 was used to find IOD solutions for a variety of cislunar scenarios. An IOD solution in the cislunar domain was also developed using the IROD method in Section 5.3. The ability to accurately model relative motion dynamics was shown in the development of the linear LERM and LEXPM motion models for the ER3BP and the CR3BP. The motion models were shown to maintain accuracy for a variety of scenarios ranging from LLO to HALO orbits.

7.1.1 Satellite Image Processing and Orbit Estimation Conclusions

Testing of the template matching and generation algorithms presented show the viability of autonomous template image generation and RSO association across a set of image frames. Note that the image frames need not be taken at a constant time-step, a fact demonstrated by the loss of data near the end of the scenario. The ability to automatically update the template image when the template changes significantly, or matching failed was proven and provided excellent template matching results for the images tested. The algorithm was tested on a large data set containing multiple RSOs with enough relative motion to require updating the template in order to successfully match across all frames. Pairing the template matching results with a UKF for initial orbit determination results shows the capability of the algorithm to accept large amounts of data and process them to provide an orbit estimate given only the input of an image set (and associated image metadata such as image capture time).

The autonomous RSO detection algorithm using optical flow was used to successfully track the gross motion of stars and RSO objects through various simulated and real image sets. Overall image-plane velocity, given by star-motion proved to be a successful method of determining candidate RSO objects. The next step for any RSO detection method is the use of the detected RSO positions for an orbit estimation method to complete the SSA process.

Orbit determination capabilities were shown to be effective using both a UKF and batch least squares approach. The choice of estimation method depends on the specific scenario. Given a large number of measurements, batch estimation provided an accurate orbit solution. For determining an

orbit with sequential filtering the UKF algorithm was shown to provide accurate results.

7.1.2 Spacecraft Relative Motion and Applications Conclusions

A semi-analytical solution for spacecraft relative motion in the ER3BP and CR3BP dynamics models was developed using a discretized matrix exponential. The linearized state-matrices were also developed and can be integrated as a set of linear differential equations. The linear motion models were shown to hold up in three-body orbits for relatively long periods of time, and for shorter periods of time in low orbits such as LEO and LLO. In order to increase the accuracy of simulations, it is suggested that an adaptive time-step for integration with the LERM equations, or the LEXPM equations be adopted. Perilune passage was seen to affect the accuracy of the deputy propagation significantly. Future work should investigate the error incurred by increased deputy separations. Further research into applying the linear EOMs to a controlled scenario should be investigated. Possible scenarios include relative circumnavigation, docking, or observation for SSA.

One application of the linearized relative motion dynamics was shown in utilizing a differential corrector to find closed relative trajectories in the CR3BP. The algorithm was tested on several different cislunar orbital geometries and shows promise for finding closed trajectories for a variety of time of flight values. Two families of trajectories were observed, a family of closed trajectories near the chief, and a set of solutions that essentially become rendezvous trajectories with the deputy initial relative position.

The linearized relative motion system matrices developed in Chap. 4 were successfully used to develop an initial implementation of a three-body IROD method. The method as it stands is limited due to the poor conditioning of the equations, and MATLAB's use of a simple Newton numerical solver. Improving the IROD method will provide an effective means of finding a first relative orbit estimate.

Particle swarm optimization shows promise for providing IOD results for cislunar SSA applications. Scenarios tested were representative of possible SSA tasks in cislunar space with the goal of obtaining a usable initial state guess for the deputy satellite to initiate proximity operations or enter a satellite into a catalog of cislunar objects for further tracking. PSO global optimization was able

to get an initial state estimate that was close enough to seed a local optimizer. Further optimization through nonlinear least squares then found the initial state with much greater accuracy. Further improvement in the initial condition for PSO, and looking at varying weighting parameters is a subject of future investigation to improve the IOD results.

7.2 Recommendations for Future Research

Recommendations for future research and improvements to the algorithms and methods presented in this dissertation are summarized in this section.

7.2.1 Template Matching Improvements

Further research needs to be done examining the bounding cases of “what makes a template”, single RSO templates, and application to further data sets. Investigation should be performed into the capability of template matching to identify a single RSO in an image containing star streaks. A comparative study between different RSO identification methods should be performed for a variety of observation scenarios to establish a baseline of the best method to use for a given scenario. The comparative study should also be paired with examining the best orbit determination method to use for each scenario.

7.2.2 Relative Motion Improvements

The LERM and LEXPM motion models show promise for describing relative motion scenarios in the ER3BP and the CR3BP. A more robust theoretical stability analysis of the linearized equations of motion should be performed. Further research should also be performed to find the limits on the distance from the chief satellite that still provide accurate orbit propagation using the linear motion models.

7.2.3 Natural Closed Relative Motion Improvements

Further investigation into the search space for the corrector and the initial guess for solution is needed. Some additional improvements to the algorithm include changing the corrector to a variable time corrector, and defining the constraint vector such that the deputy would return to its initial position and velocity. Attempting to match initial position and velocity may provide the first step to

finding multi-revolution solutions. Multi-revolution solutions also might be found by establishing a series of “patch points” to return to the same initial position a number of times similar to how multi-revolution periodic solutions are found for the non-relative three-body problem. Another important improvement will be to place constraints on the feasible search space for initial conditions (especially velocity) as it was observed for the longer TOF solutions very large initial velocities were required that may not be feasible based on propellant requirements of spacecraft.

7.2.4 IROD Improvements

First the IROD constraint equations for the CR3BP are very poorly numerically scaled. Elements in the equations range greatly in magnitude from 10^1 to 10^{14} . One solution for this problem may be numerical re-scaling of the equations. Another solution would be to use the non-dimensional form of CR3BP dynamics to develop the IROD equations. Another issue that must be addressed is the solution method for the polynomial equations to be solved. MATLAB’s `vpasolve` only utilizes a standard Newton method and convergence is not guaranteed unless a relatively close guess to the true deputy initial condition. A more robust equation solver should be applied to the IROD equations, ideally a method that does not require a close initial guess. Overall it was shown that for the relative CR3BP, the IROD problem can be solved. Some key advantages of the method are that it is based on analytical solution methods, and only requires three LOS measurements to work.

7.2.5 PSO IOD Improvements

Additional improvements to the PSO IOD method can be made by utilizing GPU core processing. Extension of the method to the ER3BP should also be done. Extension to the relative frame should also be investigated as well as the use of the LEXPM and LERM models for the PSO dynamics model.

7.3 Concluding Remarks

The goal of this dissertation has been to improve knowledge in the realm of autonomous space surveillance for both Earth-orbiting and cislunar domains. The individual algorithms developed herein provide a set of tools that with further development could be transformed into a complete

autonomous SSA package. The theoretical developments for linearized relative three-body motion provide a novel method of propagating spacecraft states using a linear system that is applicable to onboard implementation for a spacecraft to reduce computational burden compared to full non-linear integration. It is hoped that the individual algorithms developed herein provide a useful reference and starting point for the development of a complete autonomous SSA processing package for unresolved space images. Additionally, the linearized motion models for cislunar dynamics provide a foundation for linear closed-loop applications of cislunar satellite estimation and control.

REFERENCES

- [1] “Normalized 2-D cross-correlation - MATLAB normxcorr2,” <https://www.mathworks.com/help/images/ref/normxcorr2.html>, 2021. Accessed: 2021-12-3.
- [2] Virtanen, J., Poikonen, J., Sääntti, T., Komulainen, T., Torppa, J., Granvik, M., Muinonen, K., Pentikäinen, H., Martikainen, J., Näränen, J., Lehti, J., and Flohrer, T., “Streak detection and analysis pipeline for space-debris optical images,” *Adv. Space Res.*, Vol. 57, No. 8, 2016, pp. 1607–1623.
- [3] Schildknecht, T., “Optical surveys for space debris,” *Astron. Astrophys. Rev.*, Vol. 14, No. 1, 2007, pp. 41–111.
- [4] Smith, M., Craig, D., Herrmann, N., Mahoney, E., Krezel, J., McIntyre, N., and Goodliff, K., “The Artemis Program: An Overview of NASA’s Activities to Return Humans to the Moon,” *2020 IEEE Aerospace Conference*, 2020, pp. 1–10.
- [5] Batcha, A., Govinda, C., Hume, S., Kowalski, A., Meszaros, A., Meyer, A., Pino, P., Poliaček, M., Salmeri, A., and Shah, J., “All for one and one for all: Recommendations for Sustainable International Lunar Base Utilisation and Exploration Approaches,” *Proceedings of the 71st International Astronautical Congress (IAC) – The CyberSpace Edition*, the International Astronautical Federation (IAF), 2020.
- [6] Lewis, J. P., “Fast template matching,” *Vision interface*, 1995.
- [7] Tomasi, C., and Manduchi, R., “Bilateral filtering for gray and color images,” *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, IEEE, 1998, pp. 839–846. <https://doi.org/10.1109/ICCV.1998.710815>.
- [8] Vallado, D. A., *Fundamentals of Astrodynamics and Applications*, Space technology library, Vol. 21, Springer New York, 2007.

- [9] Knutson, J., “Axios: U.S. says russian anti-satellite test created more than 1,500 piece of space debris.” , 2021. URL <https://www.proquest.com/docview/2597529777?accountid=27203&forcedol=true>.
- [10] Harrington, N., and Ramji, J., “ARES Orbital Debris Program Office Photo Gallery,” <https://orbitaldebris.jsc.nasa.gov/photo-gallery/>, 2023. Accessed: 2022-4-14.
- [11] Henderson, T. A., Mortari, D., and Davis, J., “Modifications to the Gooding Algorithm for Angles-only Initial Orbit Determination,” *20TH AAS/AIAA Space Flight Mechanics Meeting*, 2010.
- [12] Brad Sease, B. F., “GEODETICA: A general software platform for processing continuous space-based imagery,” *25th AAS/AIAA Space Flight Mechanics Meeting*, 2015.
- [13] Koblick, D., Goldsmith, A., Klug, M., Mangus, P., Flewelling, B., Jah, M., Shanks, J., Piña, R., Stauch, J., and Baldwin, J., “Ground optical signal processing architecture for contributing space-based SSA sensor data,” Tech. rep., AIR FORCE RESEARCH LAB KIRTLAND AFB NM, 2014.
- [14] Guy Nir, Barak Zachay, and Eran O. Ofek, “Optimal and Efficient Streak Detection in Astronomical Images,” *Astronomical Journal*, Vol. 156, No. 5, 2018, p. 229.
- [15] Sease, B., and Flewelling, B., “Polar and spherical image transformations for star localization and RSO discrimination,” *25th AAS/AIAA Space Flight Mechanics Meeting*, 2015.
- [16] Alan Lovell, T., Zuehlke, D., and Henderson, T., “Processing of Space Object Data from Optical Observers for Space Domain Awareness,” *2021 IEEE Aerospace Conference (50100)*, IEEE, 2021, pp. 1–11.
- [17] Zuehlke, D., Henderson, T., Alan Lovell, T., and Sizemore, A., “An End-to-End Process for Local Space Situational Awareness from Optical Observers,” *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2020, pp. 1547–1555.

- [18] Zuehlke, D., and Henderson, T., “Geostationary Satellite Constellation Tracking and Identification Using Normalized Cross Correlation,” *AIAA Scitech 2020 Forum*, AIAA SciTech Forum, American Institute of Aeronautics and Astronautics, 2020.
- [19] Yang, Y., Chen, Z., Li, X., Guan, W., Zhong, D., and Xu, M., “Robust template matching with large angle localization,” *Neurocomputing*, Vol. 398, 2020, pp. 495–504.
- [20] Kim, H. Y., and de Araújo, S. A., “Grayscale Template-Matching Invariant to Rotation, Scale, Translation, Brightness and Contrast,” *Advances in Image and Video Technology*, Springer Berlin Heidelberg, 2007, pp. 100–113.
- [21] Shi, J., and Tomasi, C., “Good features to track,” *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, WA, 1994, pp. 593–600. <https://doi.org/10.1109/CVPR.1994.323794>.
- [22] Zuehlke, D., Henderson, T. A., and McMullen, S., “Machine learning using template matching applied to object tracking in video data,” *Proceedings of the SPIE, Volume 11006, id. 110061S* 7 pp., 2019.
- [23] Battin, R., “An Introduction to the Mathematics and Methods of Astrodynamics,” *New York, American Institute of Aeronautics and Astronautics, Inc., 1987*, 824 p., 1987. <https://doi.org/10.2514/4.861543>.
- [24] Franzini, G., and Innocenti, M., “Relative motion equations in the local-vertical local-horizon frame for rendezvous in lunar orbits,” *2017 AAS/AIAA Astrodynamics Specialist Conference*, AIAA, 2017.
- [25] Khoury, F., and Howell, K. C., “Orbital rendezvous and spacecraft loitering in the earth-moon system,” *Orbital rendezvous and spacecraft loitering in the earth-moon system*, AAS, 2021.
- [26] Erwin, S., “Industry proposals sought for ‘cislunar highway patrol’ satellite,”

<https://spacenews.com/industry-proposals-sought-for-cislunar-highway-patrol-satellite/>, Mar. 2022. Accessed: 2022-3-23.

- [27] Szebehely, V. G., *Theory of Oribts: The Restricted Problem of Three Bodies*, Academic Press, Yale University, New York, 1967.
- [28] Gordon, S. C., “Orbit determination error analysis and station-keeping for libration point trajectories,” Ph.D. thesis, Purdue University, Ann Arbor, United States, 1991.
- [29] Jesse A Greaves, Scheeres, Daniel, “Relative Estimation in the Cislunar Regime using Optical Sensors Jesse A. Greaves,” *Advanced Maui Optical and Space Surveillance Technologies Conference. 22ND 2021.*, 2021.
- [30] Franzini, G., and Innocenti, M., “Relative Motion Dynamics with Arbitrary Perturbations in the Local-Vertical Local-Horizon Reference Frame,” *The Journal of the Astronautical Sciences*, Vol. 67, No. 1, 2020, pp. 98–112.
- [31] Galullo, M., Bucchioni, G., Franzini, G., and Innocenti, M., “Closed Loop Guidance During Close Range Rendezvous in a Three Body Problem,” *The Journal of the Astronautical Sciences*, 2022.
- [32] Wilson, R. S., “A Design Tool for Constructing Multiple Lunar Swingby Trajectories,” Ph.D. thesis, Purdue University, 1993.
- [33] Gupta, M., “Finding order in chaos: Resonant orbits and Poincare sections,” Ph.D. thesis, Purdue University, May 2020.
- [34] Grebow, D. J., “Generating periodic orbits in the circular restricted three-body problem with applications to lunar south pole coverage,” Ph.D. thesis, Purdue University, May 2006.
- [35] Pernicka, H. J., “The numerical determination of nominal libration point trajectories and development of a station-keeping strategy,” Ph.D. thesis, Purdue University, Ann Arbor, United States, 1990.

- [36] Hiday, L. A., “Optimal transfers between libration-point orbits in the elliptic restricted three-body problem,” Ph.D. thesis, Purdue University, Ann Arbor, United States, 1992.
- [37] Franzini, G., and Innocenti, M., “Relative Motion Dynamics in the Restricted Three-Body Problem,” *J. Spacecr. Rockets*, Vol. 56, No. 5, 2019, pp. 1322–1337.
- [38] Giordana Bucchioni, M. I., “Rendezvous in Cis-Lunar Space near Rectilinear Halo Orbit: Dynamics and Control Issues,” *Aerospace*, Vol. 8, No. 3, 2021, p. 68.
- [39] Greaves, J. A., and Scheeres, D. J., “Observation and Maneuver Detection for Cislunar Vehicles,” *The Journal of the Astronautical Sciences*, Vol. 68, No. 4, 2021, pp. 826–854.
- [40] Miller, L. J., “Relative Navigation for Spacecraft in Nearly Rectilinear Halo Orbits,” *AIAA SCITECH 2022 Forum*, AIAA SciTech Forum, American Institute of Aeronautics and Astronautics, 2021.
- [41] Hall, Z., Schwab, D., Eapen, R., and Singla, P., “Reachability-Based Approach for Search and Detection of Maneuvering Cislunar Objects,” *AIAA SCITECH 2022 Forum*, American Institute of Aeronautics and Astronautics, 2021.
- [42] LaFarge, N. B., Howell, K. C., and Folta, D. C., “An Autonomous Stationkeeping Strategy for Multi-Body Orbits Leveraging Reinforcement Learning,” *AIAA SCITECH 2022 Forum*, AIAA SciTech Forum, American Institute of Aeronautics and Astronautics, 2021.
- [43] Khoury, F., “Orbital Rendezvous and Spacecraft Loitering in the Earth-Moon System,” Ph.D. thesis, Purdue University, Dec. 2020.
- [44] Fowler, E., Paley, D., and Hurtt, S., “Observability Metrics for Space-Based Cislunar Domain Awareness,” *31st AIAA/AAS Space Flight Mechanics Meeting*, 2021.
- [45] Wilmer, A. P., “Space Domain Awareness Assessment of Cislunar Periodic Orbits for Lagrange Point Surveillance,” Ph.D. thesis, Air Force Institute of Technology, Dec. 2021.

- [46] Wilmer, A. P., Bettinger, R. A., and Little, B. D., “Cislunar Periodic Orbits for Earth–Moon L1 and L2 Lagrange Point Surveillance,” *J. Spacecr. Rockets*, Vol. 59, No. 6, 2022, pp. 1809–1820. <https://doi.org/10.2514/1.A35337>.
- [47] Brogan, W. L., *Modern control theory*, Prentice Hall, 1991.
- [48] Moler, C., and Van Loan, C., “Nineteen Dubious Ways to Compute the Exponential of a Matrix,” *SIAM Review*, Vol. 20, No. 4, 1978, pp. 801–836. <https://doi.org/10.1137/1020098>, URL <https://doi.org/10.1137/1020098>.
- [49] Richard Berry and the Celestron Engineering Team, “Big! Fast! Wide! Sharp! The Story of the Rowe-Ackermann Schmidt Astrograph,” Online, Apr. 2016.
- [50] Lee, E., Park, S.-Y., Hwang, H., Choi, J., Cho, S., and Jo, J. H., “Initial orbit association and long-term orbit prediction for low earth space objects using optical tracking data,” *Acta Astronaut.*, Vol. 176, 2020, pp. 247–261.
- [51] Zuehlke, D., “Space Image Processing and Orbit Estimation Using Small Aperture Optical Systems,” Master’s thesis, Embry-Riddle Aeronautical University, 2019.
- [52] Lang, D., Hogg, D. W., Mierle, K., Blanton, M., and Roweis, S., “Astrometry.net: Blind Astrometric Calibration Of Arbitrary Astronomical Images,” *The Astronomical Journal*, Vol. 139, No. 5, 2010, p. 1782–1800. <https://doi.org/10.1088/0004-6256/139/5/1782>.
- [53] Park, E.-S., Park, S.-Y., Roh, K.-M., and Choi, K.-H., “Satellite orbit determination using a batch filter based on the unscented transformation,” *Aerosp. Sci. Technol.*, Vol. 14, No. 6, 2010, pp. 387–396.
- [54] Sullivan, J., Lovell, T. A., and D’Amico, S., “Angles-only navigation for autonomous on-orbit space situational awareness applications,” *AAS/AIAA Astrodynamics Specialist Conference, Snowbird, UT*, 2018.

- [55] Tonc, L., and Richards, G., “Orbit Estimation from Angles-Only Observations Using Nonlinear Filtering Schemes,” *Utah Space Grant Consortium*, 2019.
- [56] Gooding, R. H., “A new procedure for orbit determination based on three lines of sight (angles only),” Tech. rep., DEFENCE RESEARCH AGENCY FARNBOROUGH (UNITED KINGDOM), 1993.
- [57] Gooding, R. H., “A new procedure for the solution of the classical problem of minimal orbit determination from three lines of sight,” *Celest. Mech. Dyn. Astron.*, Vol. 66, No. 4, 1996, pp. 387–423.
- [58] Fortun, D., Bouthemy, P., and Kervrann, C., “Optical flow modeling and computation: A survey,” *Comput. Vis. Image Underst.*, Vol. 134, 2015, pp. 1–21.
- [59] Xiang, X., Zhai, M., Lv, N., and El Saddik, A., “Vehicle counting based on vehicle detection and tracking from aerial videos,” *Sensors*, Vol. 18, No. 8, 2018, p. 2560.
- [60] Dur, E., “Optical flow-based obstacle detection and avoidance behaviors for mobile robots used in unmaned planetary exploration,” *2009 4th international conference on recent advances in space technologies*, IEEE, 2009, pp. 638–647.
- [61] Du, B., Cai, S., and Wu, C., “Object Tracking in Satellite Videos Based on a Multiframe Optical Flow Tracker,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, Vol. 12, No. 8, 2019, pp. 3043–3055.
- [62] Fasano, G., Rufino, G., Accardo, D., and Grassi, M., “Satellite angular velocity estimation based on star images and optical flow techniques,” *Sensors*, Vol. 13, No. 10, 2013, pp. 12771–12793.
- [63] Lucas, B. D., and Kanade, T., “An iterative image registration technique with an application to stereo vision,” *IJCAI’81: 7th international joint conference on Artificial intelligence*, Vol. 2, Vancouver, British Columbia, 1981, pp. 674–679.

- [64] Girosi, F., Verri, A., and Torre, V., “Constraints for the computation of optical flow,” *IEEE Workshop on Visual Motion*, 1989. <https://doi.org/10.1109/WVM.1989.47101>.
- [65] Horn, B. K. P., and Schunck, B. G., “Determining Optical Flow,” *Techniques and Applications of Image Understanding*, Vol. 0281, International Society for Optics and Photonics, 1981, pp. 319–331.
- [66] Barron, J. L., Fleet, D. J., and Beauchemin, S. S., “Performance of optical flow techniques,” *Int. J. Comput. Vis.*, Vol. 12, No. 1, 1994, pp. 43–77. <https://doi.org/10.1007/BF01420984>.
- [67] “Optical Flow,” Mathworks, 2023. Available at <https://www.mathworks.com/help/vision/ref/opticalflow.html#bqi5zaf-1>.
- [68] Vallado, D. A., and McClain, W. D., *Fundamentals of astrodynamics and applications*, Microcosm Press, 2013.
- [69] Escobal, P. R., *Methods of Orbit Determination*, Krieger Publishing Company, 1976.
- [70] Julier, S. J., Uhlmann, J. K., and Durrant-Whyte, H. F., “A new approach for filtering nonlinear systems,” *Proceedings of 1995 American Control Conference - ACC’95*, Vol. 3, 1995, pp. 1628–1632 vol.3.
- [71] Seago, J., and Vallado, D., “Coordinate frames of the U.S. Space Object Catalogs,” *Astrodynamics Specialist Conference*, American Institute of Aeronautics and Astronautics, 2000, pp. 10–115.
- [72] “Space-Track.Org,” , 2021. URL <https://www.space-track.org>.
- [73] Vallado, D., and Crawford, P., “SGP4 Orbit Determination,” *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Guidance, Navigation, and Control and Co-located Conferences, American Institute of Aeronautics and Astronautics, 2008.

- [74] Franzini, G., “Relative Motion Dynamics and Control in the Two-Body and in the Restricted Three-Body Problems,” Ph.D. thesis, University of Pisa, May 2018.
- [75] Schaub, H., and Junkins, J. L., *Analytical mechanics of space systems*, Reston, VA: American Institute of Aeronautics and Astronautics, Inc., 2018.
- [76] Roncoli, R. B., “Lunar Constants and Models Document,” https://ssd.jpl.nasa.gov/doc/lunar_constants_and_models.html, sep 2005. Accessed: 2023-4-21.
- [77] David Zuehlke, Alex Sizemore, and Troy Henderson, “Regions of Application for Linearized Relative Motion in the Restricted Three Body Problem,” *Proceedings of the 33rd AAS/AIAA Space Flight Mechanics Meeting*, AAS/AIAA, 2023.
- [78] Lovell, T. A., Sinclair, A. J., and Newman, B., “Angles Only Initial Orbit Determination: Comparison of Relative Dynamics and Inertial Dynamics Approaches with Error Analysis,” *2018 Space Flight Mechanics Meeting*, American Institute of Aeronautics and Astronautics, 2018.
- [79] David Zuehlke, Alex Sizemore, and Troy Henderson, “Periodic Relative Natural Motion in the Circular Restricted Three-Body Problem,” *Proceedings of the 33rd AAS/AIAA Space Flight Mechanics Meeting*, AAS/AIAA, 2023.
- [80] Zuehlke, D., Sizemore, A., Henderson, T., and Langford, A., “Relative Motion Models for the Elliptical Restricted Three Body Problem,” *Proceedings of the 2022 AAS/AIAA Astrodynamics Specialist Conference*, AIAA, 2022.
- [81] Ryan Park, A. B. C., “Solar System Dynamics: Three-Body Periodic Orbits,” https://ssd.jpl.nasa.gov/tools/periodic_orbits.html#/periodic, 2023. Accessed: 2022-11-25.
- [82] Park, B., and Howell, K. C., “Leveraging Intermediate Dynamical Models for Transitioning from the Circular Restricted Three-Body Problem to an Ephemeris Model,” *Proceedings of the 2022 AAS/AIAA Astrodynamics Specialist Conference*, AAS/AIAA, 2022.

- [83] Howell, K., and Breakwell, J., “Almost rectilinear halo orbits,” *20th Aerospace Sciences Meeting*, Aerospace Sciences Meetings, American Institute of Aeronautics and Astronautics, 1982.
- [84] Howell, K. A. C., “Three-Dimensional, Periodic HALO Orbits in the Restricted Three-Body Problem,” Ph.D. thesis, Stanford University, Ann Arbor, United States, 1983.
- [85] Perez, A. C., Geller, D. K., and Lovell, T. A., “Non-iterative angles-only initial relative orbit determination with J2 perturbations,” *Acta Astronaut.*, Vol. 151, 2018, pp. 146–159.
- [86] “MathWorks vpasolve,” https://www.mathworks.com/help/symbolic/sym.vpasolve.html?searchHighlight=vpasolve&s_tid=srchtitle_vpasolve_1, 2023. Accessed: 2023-3-14.
- [87] Kennedy, J., and Eberhart, R., “Particle swarm optimization,” *Proceedings of ICNN’95 - International Conference on Neural Networks*, Vol. 4, 1995, pp. 1942–1948 vol.4. <https://doi.org/10.1109/ICNN.1995.488968>.
- [88] Poli, R., Kennedy, J., and Blackwell, T., “Particle swarm optimization,” *Swarm Intelligence*, Vol. 1, 2007, p. 33–57.
- [89] Clerc, M., and Kennedy, J., “The particle swarm - explosion, stability, and convergence in a multidimensional complex space,” *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 1, 2002, pp. 58–73. <https://doi.org/10.1109/4235.985692>.
- [90] Kenneth Levenberg, F. A., “A Method for the Solution of Certain Non-Linear Problems in Least Squares,” *American Mathematical Society*, 1944.
- [91] Marquardt, D. W., “An Algorithm for Least-Squares Estimation of Nonlinear Parameters,” *Journal of the Society for Industrial and Applied Mathematics*, Vol. 11, No. 2, 1963, pp. 431–441.
- [92] Moré, J. J., “The Levenberg-Marquardt algorithm: Implementation and theory,” *Numerical Analysis*, Springer Berlin Heidelberg, 1978, pp. 105–116.

- [93] Jarred A. Jordan, Daniel Posada, Matthew Gillette, David Zuehlke, Troy Henderson, “Quasi Real-Time Autonomous Satellite Detection and Orbit Estimation,” *Proceedings of the SPIE Defense+ Commercial Sensing Conference Real-Time Image Processing and Deep Learning 2023*, SPIE, 2023, pp. Paper 12528–1.
- [94] Joseph D. Anderson, Annika J. Anderson, David A. Zuehlke, David Canales, and Thomas A. Lovell, “Resident Space Object Identification in Arbitrary Unresolved Space Images,” *Proceedings of the 33rd AAS/AIAA Space Flight Mechanics Meeting*, AAS/AIAA, 2023.
- [95] Jordan, J., Posada, D., Zuehlke, D., Nocerino, A., Fontdegloria, P., Malik, S. J. A., Bevilacqua, R., and Henderson, T., “Estimation of Uncooperative Satellite Inertia Parameters for Space Debris Removal Using PSO,” *Proceedings of the 2023 IEEE Aerospace Conference*, IEEE, 2023.
- [96] David Zuehlke, Taylor Yow, Daniel Posada, Joseph Nicolich, Christopher W. Hays, Arysland Malk, and Troy Henderson, “Initial Orbit Determination for the CR3BP using Particle Swarm Optimization,” *Proceedings of the 2022 AAS/AIAA Astrodynamics Specialist Conference*, AAS/AIAA, 2022.
- [97] Jarred Jordan, Daniel Posada, David Zuehlke, Angelica Radulovic, Arysland Malk, and Troy Henderson, “Satellite Detection in Unresolved Space Imagery for Space Domain Awareness Using Neural Networks,” *Proceedings of the 2022 AAS/AIAA Astrodynamics Specialist Conference*, AAS/AIAA, 2022.
- [98] Arysland Malik, David Canales, Taylor Yow, Daniel Posada, Christopher W. Hays, David Zuehlke, and Troy Henderson, “Computation of Relative Orbital Motion using Product of Exponentials Mapping,” *Proceedings of the 2022 AAS/AIAA Astrodynamics Specialist Conference*, AAS/AIAA, 2022.
- [99] Zuehlke, D., Tiwari, M., and Henderson, T., “Autonomous Template Generation and Matching

for Satellite Constellation Tracking,” *AIAA SCITECH 2022 Forum*, American Institute of Aeronautics and Astronautics, 2021.

- [100] Tiwari, M., Zuehlke, D., Henderson, T., and Prazenica, R. J., “Autonomous Spacecraft Obstacle Avoidance and Trajectory Tracking In Dense Debris Field,” *Proceedings of the 2021 AAS/AIAA Space Flight Mechanics Conference*, Univelt, Inc., 2021.
- [101] Zuehlke, D., Posada, D., Tiwari, M., and Henderson, T., “Autonomous Satellite Detection and Tracking using Optical Flow,” *Proceedings of the 31st AAS/AIAA Space Flight Mechanics Meeting, Virtual*, AAS/AIAA, 2021.
- [102] Henry, Z. W., Vavala, P. R., Zuehlke, D., and Henderson, T. A., “Investigations of Improved Orbit and Attitude Estimation from Simultaneous Dual-Latitude Observations,” *Proceedings of the 2020 AAS/AIAA Astrodynamics Specialist Conferenec*, AAS/AIAA, 2020.

PUBLICATIONS

This chapter includes a list of all publications resulting from the research performed for this dissertation. The list is given in reverse chronological order with the newest items placed first. The publication title is given with a citation reference to the bibliography section.

1. *Quasi Real-Time Autonomous Satellite Detection and Orbit Estimation* [93].
2. *Periodic Relative Natural Motion in the Circular Restricted Three-Body Problem* [79].
3. *Regions of Application for Linearized Relative Motion in the Restricted Three-Body Problem* [77]
4. *Resident Space Object Identification in Arbitrary Unresolved Space Images* [94].
5. *Estimation of Uncooperative Satellite Inertia Parameters for Space Debris Removal Using PSO”* [95].
6. *Relative Motion Models for the Elliptical Restricted Three Body Problem* [80].
7. *Initial Orbit Determination for the CR3BP Using Particle Swarm Optimization* [96]
8. *Satellite Detection in Unresolved Space Imagery for Space Domain Awareness Using Neural Networks* [97]
9. *Computation of Relative Orbital Motion using Product of Exponentials Mapping* [98]
10. *Autonomous Template Generation and Matching for Satellite Constellation Tracking* [99]
11. *Autonomous Spacecraft Obstacle Avoidance and Trajectory Tracking in Dense Debris Field* [100]
12. *Autonomous Satellite Detection and Tracking using Optical Flow* [101]
13. *Processing of Space Object Data from Optical Observers for Space Domain Awareness* [16]
14. *An End-to-End Process for Local Space Situational Awareness from Optical Observers* [17]
15. *Investigations of Improved Orbit and Attitude Estimation from Simultaneous Dual-Latitude Observations* [102]
16. *Geostationary Satellite Constellation Tracking and Identification Using Normalized Cross Correlation* [18]