Autonomous Trajectory Planning for Satellite RPO and Safety of Flight Using Convex Optimization

Nicholas G. Ortolano
Utah State University

Follow this and additional works at: https://digitalcommons.usu.edu/etd
Part of the Aerospace Engineering Commons

Recommended Citation
All Graduate Theses and Dissertations. 7291.
https://digitalcommons.usu.edu/etd/7291
AUTONOMOUS TRAJECTORY PLANNING FOR SATELLITE RPO AND SAFETY OF FLIGHT USING CONVEX OPTIMIZATION

by

Nicholas G. Ortolano

A dissertation submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Aerospace Engineering

Approved:

David K. Geller, Ph.D.
Major Professor

Geordie Richards, Ph.D.
Committee Member

Stephen A. Whitmore, Ph.D.
Committee Member

Robert E. Spall, Ph.D.
Committee Member

Jacob Gunther, Ph.D.
Committee Member

Laurens H. Smith, Ph.D.
Interim Vice President for Research and Dean of the School of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah
2018
ABSTRACT

Autonomous Trajectory Planning for Satellite RPO and Safety of Flight using Convex Optimization

by

Nicholas G. Ortolano, Doctor of Philosophy
Utah State University, 2018

Major Professor: David K. Geller, Ph.D.
Department: Mechanical and Aerospace Engineering

New solutions to the optimal autonomous trajectory planning problem for rendezvous and proximity operations of two satellites are developed using convex optimization theory. Convex optimization algorithms have deterministic convergence properties, are self-starting, i.e., they do not require an initial guess, and have been tested in real-time environments. Traditional spacecraft rendezvous, inspection, and final approach trajectories with respect to a circular chief reference orbit are considered, as well as new approach scenarios which ensure the safety of flight of two satellites. A variety of linear dynamics models are investigated: Hill Clohessy-Wiltshire dynamics to describe the relative motion in a local-horizontal local-vertical frame, relative orbital motion dynamics relative to a spinning or uncontrolled spacecraft, and a new formulation that implements relative orbital elements. Optional trajectory constraints that are considered include maximum thrust acceleration levels, approach corridors, spherical keep-out zones, and passive safety of flight constraints. Passive safety of flight constraints for trajectory design are imposed as a means for ensuring zero probability of collision in the event of a passive failure on the deputy satellite, such as power loss, computer shutdown/reboot, or a suspension of normal activities due to mission/vehicle anomalies. This involves constraints that ensure the deputy is in a passive abort safety
ellipse at all times, to avoid collision with the chief satellite in the event of a passive failure. In all cases, an algorithm based on a second-order cone program is developed and used to generate minimum-fuel rendezvous and proximity operation trajectories. In the event that nonconvex constraints are required, a method of sequential convex programming is adopted, whereby all nonconvex constraints are convexified via linearized approximations, and a convex program is iteratively solved. Data on algorithm CPU and memory requirements for a variety of scenarios is collected. Reference trajectory results from the algorithm are presented for several typical scenarios. These scenarios are implemented in a nonlinear orbital simulation, using optimal trajectory following methods, to better understand the efficiency and practicality of convex rendezvous and proximity operations trajectory planning.
PUBLIC ABSTRACT

Autonomous Trajectory Planning for Satellite RPO and Safety of Flight using Convex Optimization

Nicholas G. Ortolano

Optimal trajectory planning methods that implement convex optimization techniques are applied to the area of satellite rendezvous and proximity operations. This involves the development of linearized relative orbital motion dynamics and constraints for two satellites, where one maintains a near-circular reference orbit. The result is formulated as a convex optimization problem, where the objective is to minimize the amount of fuel required to transfer from a given initial condition to the desired final conditions. A traditional rendezvous and proximity operations scenario is analyzed, which includes examples of initial approach, inspection, final approach, and docking trajectories. This scenario may include trajectory constraints such as maximum allowable control acceleration levels, approach corridors, and spherical keep-out zones. A second scenario that ensures passive safety of flight is also developed, where constraints are imposed to guarantee passive safety, in the event of control failures on the maneuvering satellite. The convex optimization problem is ultimately formulated as a second-order cone program. Algorithm CPU and memory requirements are analyzed. Several examples of resulting optimal trajectories are presented for both scenarios, and these trajectories are implemented in a nonlinear simulation.
For my wife, father, mother, and sister.
My primary impetus for this work.
ACKNOWLEDGMENTS

Throughout my work and progress in the field of satellite guidance, navigation, and control, there have entered many individuals to whom I owe my absolute gratitude. These individuals deserve the foremost recognition for aiding in the advancement of my research, education, and career. Their actions and influence have guided me in my growth as an engineer, and I am grateful to include them as a part of my story. I cherish all of the friends I have made while at Utah State University (USU), and I hope to preserve the memories we’ve made. It is evident that without their help, I would not have been able to achieve my goals. Therefore, I wish to personally acknowledge each, as a testament of my appreciation.

First and foremost, I must thank my major professor, Dr. David Geller. His teachings, insight, and guidance throughout our time together have provided me with immeasurable understanding and opportunity. He is recognized as a quintessential engineering figure; someone who every engineer can look up to. His level of expertise, dedication, and precision in the field of aerospace engineering is seemingly insurmountable. At times he praised me for my efforts (the feeling I will never forget)...and a few times he had to ‘keep me on track,’ telling me things that I needed to hear so that I would remain focused. I must also thank him for acquiring funding for this research from Space Dynamics Laboratory, especially considering how difficult finding funding can often be. For these reasons, he is a dominant authority in my education, and it is an honor for me to show my respect and admiration.

Secondly, I would like to thank my family. My wife, Aarica (whom I met at USU), has always been a great source of inspiration. She continually gives me reliable advice, and I appreciate her honesty with many matters. Her intelligence and understanding always impresses me, and proves to be valuable in every situation. Aarica, I love you, and I am thankful for your companionship. My mother and father, Dee and Mike, have been the utmost driving force behind all of my education endeavors. Their devotion as parents is highly admirable, and has paved the path for many of my accomplishments. Much of my character can be attributed to the way they raised me, the life lessons they’ve taught, and
the love they share for each other. My sister, Ari, will always be a dear friend of mine. For years she has provided the sense of humor, adventure, and leisure that are required to manage the everyday stress of being a graduate student. I value her resolve in her education, as it has helped reinforced my own. I personally thank all of my family, for the love they have displayed to me and each other.

Next, I would like to acknowledge the friends I have made at USU and Space Dynamics Laboratory (SDL). For the efforts of Aaron Avery: I thank him for first bringing the idea of this research to the good folks at SDL. I also greatly appreciate all of his continued input as a member of our autonomous planner group. It has been great to have him in a role where I can ask questions and receive his helpful responses. I would like to thank Daela Huff for her technical writing skills in editing so many of our journal and conference papers. Her level of accuracy is unmatched, therefore, I thank her for bringing our rough drafts up to journal-level quality. I also thank the SDL community as a whole for their funding and interest in this research, without whom, none of this would have been possible. They have provided a world-class experience as a graduate student, and for this I am eternally grateful.

Finally, I would like to thank the members of our annual AAS-RMS GN&C Conference group. My fellow graduate students, Alex Perez, Simon Shuster, Rachit Bhatia, and Ahkter Nafi have been exceedingly supportive and helpful in the process of attending and presenting at conferences. It has been a pleasure to work with them on several projects, and to collaborate and discuss our different areas of research. I thank Simon Shuster personally for his aid in getting my convex optimization code to run more efficiently. Thanks to each of them, and also to all those present at the weekly GN&C TIM. I greatly value the opportunity to share our ideas, and the friendships that I have made.

Nicholas G. Ortolano
CONTENTS

ABSTRACT ...................................................................................... iii
PUBLIC ABSTRACT ....................................................................... v
ACKNOWLEDGMENTS ...................................................................... vii
LIST OF TABLES ............................................................................... xiv
LIST OF FIGURES ............................................................................. xviii
LIST OF ALGORITHMS ..................................................................... xxviii
ACRONYMS ...................................................................................... xxix
1 INTRODUCTION ............................................................................. 1
  1.1 Methodology ............................................................................. 2
2 LITERATURE SURVEY ..................................................................... 10
  2.1 Autonomy, Optimal Control, and Convex Optimization: Aerospace Applications ........................................... 10
  2.2 Relative Satellite Motion Models .................................................. 13
  2.3 Safety of Flight .......................................................................... 16
  2.4 Current Optimal Guidance Techniques .......................................... 19
    2.4.1 Model Predictive Control ......................................................... 19
    2.4.2 Nonlinear Optimization ............................................................ 20
    2.4.3 Mixed Integer Linear Programming ......................................... 21
    2.4.4 Feedback Control ................................................................. 21
    2.4.5 Convex Optimization ............................................................. 21
3 CONVEX OPTIMIZATION AND THE SOCP .................................... 23
  3.1 Convex Optimization ................................................................... 23
    3.1.1 Duality and Convergence ....................................................... 24
    3.1.2 General Form of a Convex Optimization Problem .................... 25
  3.2 Common Forms for the Second-order Cone Program ......................... 25
    3.2.1 Smoothness and Convexity of SOCP Constraint Functions ........ 30
  3.3 Conditions of Optimality for the SOCP ......................................... 32
    3.3.1 Lagrangian ......................................................................... 33
    3.3.2 KKT Conditions ................................................................ 33
  3.4 Dual Problem for the Second-order Cone Program .......................... 34
  3.5 Solving the General Convex Problem ............................................. 35
    3.5.1 Barrier Method .................................................................. 36
    3.5.2 Primal-dual Interior-point Method ......................................... 38
3.5.3 Extended Self-dual Embedding Method ........................................... 39
3.6 Newton’s Method ................................................................................. 41
3.7 Sequential Convex Optimization ........................................................... 42

4 THESIS STATEMENT AND RESEARCH OVERVIEW ............................ 44
4.1 Thesis Statement .................................................................................. 44
4.2 Research Overview .............................................................................. 44
  4.2.1 Dissertation Outline ....................................................................... 47

5 RELATIVE ORBITAL MOTION, THE HCW EQUATIONS ..................... 50
5.1 The Hill, Clohessy-Wiltshire Equations ............................................... 51
  5.1.1 The Cartesian LVLH Frame ............................................................ 52
  5.1.2 The HCW Equations in Spherical Coordinates ................................ 55
5.2 Relative Orbital Elements ...................................................................... 58
  5.2.1 ROE Differential Equations ............................................................ 60
  5.2.2 Properties of the ROE Differential Equations ................................. 61
5.3 Linear Relative Orbital Elements ........................................................... 61
5.4 Pseudo-chief Reference Frame .............................................................. 65
5.5 Common Relative Motion Trajectories .................................................. 66

6 THE GENERALIZED HCW EQUATIONS ............................................. 72
6.1 The Generalized HCW Equations .......................................................... 74
  6.1.1 Linearized Equations of Motion for Constant Angular Rotation Vector 74
  6.1.2 Linearized Equations of Motion for Time-Varying Angular Rotation Vector ........................................... 77
6.2 Reduction to HCW Equations ............................................................... 78
6.3 Approximate Analytic Solution When \( \Omega^2 \ll \omega^2 \) ........................... 80
  6.3.1 Inertial Targeting Algorithm ............................................................ 82

7 RPO PLANNING RELATIVE TO AN LVLH REFERENCE FRAME ........ 84
7.1 Problem Formulation ............................................................................ 84
  7.1.1 Relative Motion Dynamics with Respect to an LVLH Frame .......... 85
  7.1.2 Objective Function ......................................................................... 85
  7.1.3 Constraints ...................................................................................... 87
7.2 Solution Method .................................................................................... 90
  7.2.1 Second-order Cone Problem Statement ........................................ 90
  7.2.2 Solution to the Spherical Keep-out Zone ....................................... 91
7.3 Planner Results .................................................................................... 93
  7.3.1 Initial Approach with Maximum Thrust Constraint ....................... 93
  7.3.2 Way Point Following/Inspection .................................................... 96
  7.3.3 Way Point Following/Inspection with Keep-out Zone ................... 98
  7.3.4 Final Approach with Approach Corridor ..................................... 99
7.4 Summary ............................................................................................. 105
<table>
<thead>
<tr>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.6.1 Barrier Method Formulation</td>
</tr>
<tr>
<td>10.6.2 Barrier Method Pseudo code</td>
</tr>
<tr>
<td>10.7 RPO Planning Algorithm Using Primal-dual Interior-point Method</td>
</tr>
<tr>
<td>10.7.1 Primal-dual Interior-point Method Formulation</td>
</tr>
<tr>
<td>10.7.2 Primal-dual Interior Point Method Pseudo code</td>
</tr>
<tr>
<td>11 RPO PLANNER CPU AND MEMORY REQUIREMENTS</td>
</tr>
<tr>
<td>11.1 Hardware</td>
</tr>
<tr>
<td>11.2 Software</td>
</tr>
<tr>
<td>11.3 Solver Comparison</td>
</tr>
<tr>
<td>11.4 Important Metrics</td>
</tr>
<tr>
<td>11.5 CPU Requirements Tables</td>
</tr>
<tr>
<td>11.6 Memory Requirements Tables</td>
</tr>
<tr>
<td>11.7 Custom Solver Comparison</td>
</tr>
<tr>
<td>11.8 Conclusions</td>
</tr>
<tr>
<td>12 NONLINEAR SIMULATION</td>
</tr>
<tr>
<td>12.1 Introduction</td>
</tr>
<tr>
<td>12.2 Nonlinear Simulation Model</td>
</tr>
<tr>
<td>12.2.1 Inertial Chief States and Relative Deputy States</td>
</tr>
<tr>
<td>12.2.2 Inertial Chief States and Inertial Deputy States</td>
</tr>
<tr>
<td>12.2.3 Differential Equations Summary</td>
</tr>
<tr>
<td>12.3 Reference Trajectory</td>
</tr>
<tr>
<td>12.4 Optimal LQR Control Design</td>
</tr>
<tr>
<td>12.5 Cartesian to Spherical Conversion of Reference Trajectory</td>
</tr>
<tr>
<td>12.5.1 Spherical Feed-forward Control</td>
</tr>
<tr>
<td>12.6 Navigation and Control Assumptions</td>
</tr>
<tr>
<td>13 PERFORMANCE ANALYSIS I, RENDEZVOUS, INSPECTION, AND FINAL APPROACH - SCENARIO 1</td>
</tr>
<tr>
<td>13.1 Scenario 1 Methodology</td>
</tr>
<tr>
<td>13.2 Scenario 1 Analysis Setup</td>
</tr>
<tr>
<td>13.2.1 Simulation Parameters</td>
</tr>
<tr>
<td>13.2.2 Initial Conditions</td>
</tr>
<tr>
<td>13.3 Case 1 - Initial Approach</td>
</tr>
<tr>
<td>13.4 Case 2 - Way-point Following</td>
</tr>
<tr>
<td>13.5 Case 3 - Final Approach</td>
</tr>
<tr>
<td>13.6 Conclusions</td>
</tr>
<tr>
<td>14 PERFORMANCE ANALYSIS II, SAFE RENDEZVOUS, TRAVELING ELLIPSE, AND SAFE FINAL APPROACH - SCENARIO 2</td>
</tr>
<tr>
<td>14.1 Scenario 2 Methodology</td>
</tr>
<tr>
<td>14.2 Scenario 2 Analysis Setup</td>
</tr>
<tr>
<td>14.2.1 Simulation Parameters</td>
</tr>
<tr>
<td>14.2.2 Initial Conditions</td>
</tr>
<tr>
<td>14.3 Case 1 - Safe Initial Approach</td>
</tr>
<tr>
<td>14.4 Case 2 - Safe Traveling Ellipse</td>
</tr>
<tr>
<td>Section</td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td>14.5</td>
</tr>
<tr>
<td>14.6</td>
</tr>
<tr>
<td>15</td>
</tr>
<tr>
<td>15.1</td>
</tr>
<tr>
<td>15.2</td>
</tr>
<tr>
<td>15.3</td>
</tr>
<tr>
<td>15.3.1</td>
</tr>
<tr>
<td>15.3.2</td>
</tr>
<tr>
<td>15.3.3</td>
</tr>
<tr>
<td>15.4</td>
</tr>
<tr>
<td>15.5</td>
</tr>
<tr>
<td>15.5.1</td>
</tr>
<tr>
<td>15.5.2</td>
</tr>
<tr>
<td>15.5.3</td>
</tr>
<tr>
<td>15.6</td>
</tr>
<tr>
<td>16</td>
</tr>
<tr>
<td>16.1</td>
</tr>
<tr>
<td>16.2</td>
</tr>
<tr>
<td>REFERENCES</td>
</tr>
<tr>
<td>CURRICULUM VITAE</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1</td>
<td>94</td>
</tr>
<tr>
<td>7.2</td>
<td>97</td>
</tr>
<tr>
<td>7.3</td>
<td>101</td>
</tr>
<tr>
<td>8.1</td>
<td>118</td>
</tr>
<tr>
<td>9.1</td>
<td>157</td>
</tr>
<tr>
<td>9.2</td>
<td>159</td>
</tr>
<tr>
<td>9.3</td>
<td>161</td>
</tr>
<tr>
<td>9.4</td>
<td>163</td>
</tr>
<tr>
<td>9.5</td>
<td>166</td>
</tr>
<tr>
<td>9.6</td>
<td>169</td>
</tr>
<tr>
<td>11.1</td>
<td>206</td>
</tr>
<tr>
<td>11.2</td>
<td>209</td>
</tr>
<tr>
<td>11.3</td>
<td>209</td>
</tr>
<tr>
<td>11.4</td>
<td>210</td>
</tr>
<tr>
<td>11.5</td>
<td>210</td>
</tr>
<tr>
<td>11.6</td>
<td>210</td>
</tr>
<tr>
<td>11.7</td>
<td>214</td>
</tr>
<tr>
<td>11.8</td>
<td>214</td>
</tr>
</tbody>
</table>
11.9 Scenario 1, without approach corridor, problem results for CPU requirements
11.10 Scenario 2, with approach corridor, problem results for CPU requirements
11.11 Scenario 1, without approach corridor, problem results for memory requirements
11.12 Scenario 2, with approach corridor, problem results for memory requirements
11.13 Scenario 1, without approach corridor, cases tested for custom solver comparison
11.14 Scenario 2, with approach corridor, cases tested for custom solver comparison
11.15 Scenario 1, without approach corridor, problem results for custom solver comparison
11.16 Scenario 2, with approach corridor, problem results for custom solver comparison
13.1 Chief and deputy simulation parameters
13.2 Chief initial orbital elements (* in some cases, chief orbit eccentricity is varied)
13.3 Deputy initial traditional relative orbital elements and final relative position/velocity for Scenario 1, Case 1
13.4 Scenario 1, Case 1 with circular chief and without perturbations, $\Delta v$ from planner and simulation for reference trajectory in Cartesian and spherical coordinates
13.5 Scenario 1, Case 1 with circular chief and perturbations, $\Delta v$ from planner and simulation for reference trajectory in Cartesian and spherical coordinates
13.6 Scenario 1, Case 1 with varying chief ellipticity and perturbations, $\Delta v$ from planner and simulation for reference trajectory in Cartesian and spherical coordinates
13.7 Deputy initial and final relative position/velocity for Scenario 1, Case 2
13.8 Scenario 1, Case 2 with circular chief and perturbations, $\Delta v$ from planner and simulation with Cartesian model
13.9 Scenario 1, Case 2 with varying chief ellipticity and perturbations, $\Delta v$ from planner and simulation with Cartesian model
13.10 Deputy initial and final relative position/velocity for Scenario 1, Case 3
<table>
<thead>
<tr>
<th>Section</th>
<th>Scenario/Case</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>13.11</td>
<td>Scenario 1, Case 3</td>
<td>Scenario 1, Case 3 with circular chief and perturbations, $\Delta v$ from planner and simulation with Cartesian model</td>
<td>254</td>
</tr>
<tr>
<td>13.12</td>
<td>Scenario 1, Case 3</td>
<td>Scenario 1, Case 3 with varying chief ellipticity and perturbations, $\Delta v$ from planner and simulation with Cartesian model</td>
<td>256</td>
</tr>
<tr>
<td>14.1</td>
<td>Deputy initial traditional</td>
<td>Deputy initial traditional relative orbital elements and final HCW relative orbital elements for Scenario 2, Case 1</td>
<td>260</td>
</tr>
<tr>
<td>14.2</td>
<td>Scenario 2, Case 1</td>
<td>Scenario 2, Case 1 with circular chief and without perturbations, $\Delta v$ from planner and simulation for Cartesian and spherical coordinates</td>
<td>263</td>
</tr>
<tr>
<td>14.3</td>
<td>Scenario 2, Case 1</td>
<td>Scenario 2, Case 1 with circular chief and perturbations, $\Delta v$ from planner and simulation for Cartesian and spherical coordinates</td>
<td>265</td>
</tr>
<tr>
<td>14.4</td>
<td>Scenario 2, Case 1</td>
<td>Scenario 2, Case 1 with varying chief ellipticity and perturbations, $\Delta v$ from planner and simulation with spherical model</td>
<td>266</td>
</tr>
<tr>
<td>14.5</td>
<td>Deputy initial and final</td>
<td>Deputy initial and final ROEs for Scenario 2, Case 2</td>
<td>267</td>
</tr>
<tr>
<td>14.6</td>
<td>Scenario 2, Case 2</td>
<td>Scenario 2, Case 2 with circular chief and perturbations, $\Delta v$ from planner and simulation, and SCP iterations, for Cartesian and spherical examples</td>
<td>270</td>
</tr>
<tr>
<td>14.7</td>
<td>Scenario 2, Case 2</td>
<td>Scenario 2, Case 2 with varying chief ellipticity and perturbations, $\Delta v$ from planner and simulation with spherical model</td>
<td>271</td>
</tr>
<tr>
<td>14.8</td>
<td>Deputy initial and final</td>
<td>Deputy initial and final ROEs for Scenario 2, Case 2</td>
<td>272</td>
</tr>
<tr>
<td>14.9</td>
<td>Scenario 2, Case 3</td>
<td>Scenario 2, Case 3 with circular chief and perturbations, $\Delta v$ from planner and simulation with Cartesian model</td>
<td>274</td>
</tr>
<tr>
<td>14.10</td>
<td>Scenario 2, Case 3</td>
<td>Scenario 2, Case 3 with varying chief ellipticity and perturbations, $\Delta v$ from planner and simulation with Cartesian model</td>
<td>276</td>
</tr>
<tr>
<td>15.1</td>
<td>Monte Carlo</td>
<td>Monte Carlo $\Delta v$ for Scenario 1, Case 1, Example 4 with $p_r = 5%$ dispersions</td>
<td>281</td>
</tr>
<tr>
<td>15.2</td>
<td>Monte Carlo</td>
<td>Monte Carlo $\Delta v$ for Scenario 1, Case 1, Example 1 with $p_r = 1%$ dispersions</td>
<td>283</td>
</tr>
<tr>
<td>15.3</td>
<td>Monte Carlo</td>
<td>Monte Carlo $\Delta v$ for Scenario 1, Case 1, Example 4 with $p_r = 5%$ dispersions</td>
<td>285</td>
</tr>
<tr>
<td>15.4</td>
<td>Monte Carlo</td>
<td>Monte Carlo $\Delta v$ for Scenario 1, Case 1, Example 4 with $p_r = 1%$ dispersions</td>
<td>286</td>
</tr>
<tr>
<td>15.5</td>
<td>Monte Carlo</td>
<td>Monte Carlo $\Delta v$ for Scenario 1, Case 1, Example 4 with $p_r = 0.5%$ dispersions</td>
<td>288</td>
</tr>
<tr>
<td>15.6</td>
<td>Monte Carlo</td>
<td>Monte Carlo $\Delta v$ for Scenario 1, Case 2, Example 4 with $p_r = 10%$ dispersions</td>
<td>290</td>
</tr>
<tr>
<td>15.7</td>
<td>Monte Carlo</td>
<td>Monte Carlo $\Delta v$ for Scenario 1, Case 2, Example 4 with $p_r = 5%$ dispersions</td>
<td>291</td>
</tr>
</tbody>
</table>
15.8 Monte Carlo $\Delta v$ for Scenario 1, Case 3, Example 3 with $p_r = 10\%$ dispersions 293
15.9 Monte Carlo $\Delta v$ for Scenario 1, Case 3, Example 3 with $p_r = 5\%$ dispersions 295
15.10 Monte Carlo $\Delta v$ for Scenario 2, Case 1, Example 1 with $p_{SE} = 10\%$ dispersions 298
15.11 Monte Carlo $\Delta v$ for Scenario 2, Case 1, Example 1 with $p_{SE} = 5\%$ dispersions 300
15.12 Monte Carlo $\Delta v$ for Scenario 2, Case 2, Example 2 with $p_{SE} = 10\%$ dispersions 302
15.13 Monte Carlo $\Delta v$ for Scenario 2, Case 2, Example 2 with $p_{SE} = 5\%$ dispersions 303
15.14 Monte Carlo $\Delta v$ for Scenario 2, Case 2, Example 4 with $p_{SE} = 10\%$ dispersions 305
15.15 Monte Carlo $\Delta v$ for Scenario 2, Case 2, Example 3 with $p_{SE} = 5\%$ dispersions 306
15.16 Monte Carlo $\Delta v$ for Scenario 2, Case 3, Example 4 with $p_r = 10\%$ dispersions 308
15.17 Monte Carlo $\Delta v$ for Scenario 2, Case 3, Example 4 with $p_r = 5\%$ dispersions 310
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>18</td>
</tr>
<tr>
<td>2.2</td>
<td>19</td>
</tr>
<tr>
<td>3.1</td>
<td>23</td>
</tr>
<tr>
<td>5.1</td>
<td>53</td>
</tr>
<tr>
<td>5.2</td>
<td>53</td>
</tr>
<tr>
<td>5.3</td>
<td>54</td>
</tr>
<tr>
<td>5.4</td>
<td>55</td>
</tr>
<tr>
<td>5.5</td>
<td>56</td>
</tr>
<tr>
<td>5.6</td>
<td>57</td>
</tr>
<tr>
<td>5.7</td>
<td>59</td>
</tr>
<tr>
<td>5.8</td>
<td>64</td>
</tr>
<tr>
<td>5.9</td>
<td>65</td>
</tr>
<tr>
<td>5.10</td>
<td>66</td>
</tr>
<tr>
<td>5.11</td>
<td>67</td>
</tr>
<tr>
<td>5.12</td>
<td>68</td>
</tr>
<tr>
<td>5.13</td>
<td>69</td>
</tr>
<tr>
<td>5.14</td>
<td>70</td>
</tr>
<tr>
<td>5.15</td>
<td>71</td>
</tr>
<tr>
<td>7.1</td>
<td>91</td>
</tr>
</tbody>
</table>
7.2 Three hour approach from 10 km to 200 m along v-bar direction with ±1 km altitude variations and ±1 km cross-track variations ........................ 94

7.3 $\Delta v$/control magnitude history for three hour transfer ........................ 95

7.4 $\Delta v$/control magnitude history for three hour transfer with maximum acceleration level of 5 mm/s$^2$ .................................................. 95

7.5 Six hour approach from 10 km to 200 m along v-bar direction with ±1 km altitude variations and ±1 km cross-track variations ......................... 96

7.6 $\Delta v$/control magnitude history for six hour transfer ................................ 96

7.7 Way-point following/inspection with 30 minute transfers between each way-point .............................................................. 97

7.8 Way-point following/inspection with 30 minute transfers between each way-point and a 80 m spherical keep-out zone ................................. 98

7.9 $\Delta v$/control magnitude history for way-point following/inspection with keep-out zone ................................................................. 99

7.10 10 minute v-bar approach and cone approach trajectories with a 10 degree approach cone constraint .................................................. 100

7.11 $\Delta v$/control magnitude history for v-bar approach and cone approach .. 101

7.12 10 minute v-bar approach trajectory with a 10 degree approach cone con- straint, ±10 m altitude variations, ±10 m cross-track variations ................ 102

7.13 $\Delta v$/control magnitude history for v-bar final approach ......................... 102

7.14 Zoomed in $\Delta v$/control magnitude history for v-bar final approach, while on cone boundary ................................................................. 103

7.15 10 minute r-bar approach trajectory with a 10 degree approach cone con- straint, ±10 m altitude variations, ±10 m cross-track variations ................ 103

7.16 $\Delta v$/control magnitude history for r-bar final approach ......................... 104

7.17 Zoomed in $\Delta v$/control magnitude history for r-bar final approach, while on cone boundary ................................................................. 104

7.18 Zoomed in $\Delta v$/control magnitude history for v-bar final approach, while on cone boundary, using drift without control constraint ..................... 105
8.1 Case 1, three successive transfers around target from v-bar to body x-axis to body y-axis to body z-axis. Body rotation rate is 1 deg/sec about the inertial z-axis. LVLH frame trajectory and total delta-v.

8.2 Case 1, three successive transfers around target from v-bar to body x-axis to body y-axis to body z-axis. Body rotation rate is 1 deg/sec about the inertial z-axis. Left: Body-fixed frame. Right: Inertial frame.

8.3 Case 2, three successive transfers around target from v-bar to body x-axis to body y-axis to body z-axis. Body rotation rate is 0.2 deg/sec about the inertial y-axis. LVLH frame trajectory and total delta-v.

8.4 Case 2, three successive transfers around target from v-bar to body x-axis to body y-axis to body z-axis. Body rotation rate is 0.2 deg/sec about the inertial y-axis. Left: Body-fixed frame. Right: Inertial frame.

8.5 Case 3, three successive transfers around target from v-bar to body x-axis to body y-axis to body z-axis. Body rotation rate is 0.1 deg/sec about the inertial x-axis. LVLH frame trajectory and total delta-v.

8.6 Case 3, three successive transfers around target from v-bar to body x-axis to body y-axis to body z-axis. Body rotation rate is 0.1 deg/sec about the inertial x-axis. Left: Body-fixed frame. Right: Inertial frame.

8.7 Target-relative waypoint following/inspection with 10 minute transfers between each waypoint.

8.8 Target-relative waypoint following/inspection with 10 minute transfers between each waypoint as seen from an LVLH frame.

8.9 Control acceleration magnitude history target-relative waypoint following/inspection with 9 meter keep-out zone with maximum acceleration level of 1 cm/s².

8.10 Target-relative waypoint following/inspection with 10 minute transfers between each waypoint and a 9 meter keep-out zone.

8.11 Target-relative waypoint following/inspection with 10 minutes transfer between each waypoint and a 9 meter keep-out zone as seen from an LVLH frame.

8.12 Control acceleration magnitude history target-relative waypoint following/inspection with 9 meter keep-out zone.

8.13 Zoomed in control acceleration magnitude history target-relative waypoint following/inspection while on 9 meter keep-out zone boundary.

8.14 100 m v-bar transfer to target, where body rotation rate is LVLH. Left: Body-fixed frame. Right: Inertial frame.
xxi
8.15

100 m v-bar transfer to target with approach constraint, body rotation rate
is 0.1 deg/sec. Left: Body xed-frame. Right: Inertial frame

8.16

. . . . . . . . 123

100 m v-bar transfer to target with approach constraint, body rotation rate
is -0.2 deg/sec. Left: Body-xed frame. Right: Inertial frame . . . . . . . . 123

8.17

Target-relative x-axis and y-axis nal approach trajectories with a 10 degree
approach cone constraint and a 12 min transfer . . . . . . . . . . . . . . . . 124

8.18

Target-relative 12 minute x-axis and y-axis nal approach trajectories with
a 10 degree approach cone constraint as seen from the LVLH frame . . . . . 125

8.19

Control acceleration magnitude history for x-axis and y-axis nal approach 125

8.20

Zoomed in control acceleration magnitude history for x-axis and y-axis nal
approach, while at minimum distance

8.21

Control acceleration magnitude history for x-axis and y-axis nal approach
using `drift without control' constraints

8.22

. . . . . . . . . . . . . . . . . . . . . 126

. . . . . . . . . . . . . . . . . . . . 126

Zoomed in control acceleration magnitude history for x-axis and y-axis nal
approach, while at minimum distance using `drift without control' constraints127

8.23

Inertial targeting algorithm results, varying the xed nal time for transfer,
with a chief rotation rate of 0.25 deg/sec about the inertial z-axis. Left:
Final time of 5 minutes for each transfer. Right: Final time of 15 minutes
for each transfer. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 128

8.24

Inertial targeting algorithm results, varying the rotation rate of chief-xed
frame, with a nal transfer time of 5 minutes. Left: Chief rotation rate of
0.01 deg/sec, where miss distance was 0.735 meters. Right: Chief rotation
rate of 0.5 deg/sec, where miss distance was 0.476 meters. . . . . . . . . . . 129

9.1
9.2

Out-of-plane relative ellipse orientations as a function of phase dierence (γ )138
Out-of-plane, in-phase relative ellipses for varying semi-major axes and
cross-track amplitudes, with zero radial oset (xr

= 0)

. . . . . . . . . . . . 139

9.3

Out-of-plane relative ellipse case examples . . . . . . . . . . . . . . . . . . . 139

9.4

Out-of-plane yby ellipse example

9.5

Out-of-plane circumnavigating ellipse example

9.6

LROE state-space plots of
for

ar = 2Az

. . . . . . . . . . . . . . . . . . . . . . . 140
. . . . . . . . . . . . . . . . 141

(ar cos(Er ), ar sin(Er )), and (Az cos(ψz ), Az sin(ψz )),

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 142


9.7 LVLH trajectory plot for flyby to safety ellipse transfer ............. 157
9.8 Control history plot for flyby to safety ellipse transfer in normalized-time . 158
9.9 LROE state propagation for flyby to safety ellipse transfer ............. 158
9.10 LVLH trajectory plot for traveling safety ellipse transfer ............. 159
9.11 Control history plot for traveling safety ellipse transfer in normalized-time . 160
9.12 LROE state propagation for traveling safety ellipse transfer ............. 160
9.13 LVLH trajectory plot for large to small safety ellipse transfer ........... 161
9.14 Control history plot for large to small safety ellipse transfer in normalized-time . 162
9.15 LROE state propagation for large to small safety ellipse transfer ........... 162
9.16 Optimal flyby to safety ellipse in front of target, in-plane and out-of-plane motion . 164
9.17 Optimal flyby to safety ellipse in front of target, optimal control history . 164
9.18 Optimal flyby to safety ellipse behind target, in-plane and out-of-plane motion 165
9.19 Optimal flyby to safety ellipse behind target, optimal control history . 165
9.20 Optimal multi-rev traveling safety ellipse, in-plane and out-of-plane motion 167
9.21 Optimal multi-rev traveling safety ellipse, optimal control history . 167
9.22 Optimal safety ellipse to safety ellipse transfer via flyby, in-plane and out-of-plane motion . 168
9.23 Optimal safety ellipse to safety ellipse transfer via flyby, optimal control history . 168
9.24 Optimal transfer from large to small safety ellipse, in-plane and out-of-plane motion . 170
9.25 Optimal transfer from large to small safety ellipse, optimal control history . 170
9.26 Optimal transfer from small to large safety ellipse, in-plane and out-of-plane motion . 171
9.27 Optimal transfer from small to large safety ellipse, optimal control history . 171
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>11.1</td>
<td>Massif memory usage vs. number of instructions for initial approach scenario using LROE U-only for $N = 800$ points (Scenario 1 Problem 12)</td>
<td>217</td>
</tr>
<tr>
<td>11.2</td>
<td>Massif memory usage vs. number of instructions for final approach scenario with approach corridor using LROEs for $N = 800$ points (Scenario 2 Problem 8)</td>
<td>218</td>
</tr>
<tr>
<td>13.1</td>
<td>Scenario 1, Case 1 trajectories (Cartesian model, without perturbations)</td>
<td>240</td>
</tr>
<tr>
<td>13.2</td>
<td>Scenario 1, Case 1 control history (Cartesian model, circular orbit, without perturbations)</td>
<td>240</td>
</tr>
<tr>
<td>13.3</td>
<td>Scenario 1, Case 1 tracking error (Cartesian model, circular orbit, without perturbations)</td>
<td>241</td>
</tr>
<tr>
<td>13.4</td>
<td>Scenario 1, Case 1 control history (spherical model, circular orbit, without perturbations)</td>
<td>241</td>
</tr>
<tr>
<td>13.5</td>
<td>Scenario 1, Case 1 tracking error (spherical model, circular orbit, without perturbations)</td>
<td>242</td>
</tr>
<tr>
<td>13.6</td>
<td>Scenario 1, Case 1 control history (spherical model, with perturbations)</td>
<td>243</td>
</tr>
<tr>
<td>13.7</td>
<td>Scenario 1, Case 1 tracking error (spherical model, with perturbations)</td>
<td>243</td>
</tr>
<tr>
<td>13.8</td>
<td>Scenario 1, Case 1 control history (spherical model, with perturbations and chief eccentricity of $e_c = 0.01$)</td>
<td>244</td>
</tr>
<tr>
<td>13.9</td>
<td>Scenario 1, Case 1 tracking error (spherical model, with perturbations and chief eccentricity of $e_c = 0.01$)</td>
<td>245</td>
</tr>
<tr>
<td>13.10</td>
<td>Scenario 1, Case 2, Examples 1-3 trajectories (Cartesian model, with perturbations)</td>
<td>247</td>
</tr>
<tr>
<td>13.11</td>
<td>Scenario 1, Case 2, Examples 1-3 control history (Cartesian model, circular orbit, with perturbations)</td>
<td>247</td>
</tr>
<tr>
<td>13.12</td>
<td>Scenario 1, Case 2, Examples 1-3 tracking error (Cartesian model, circular orbit, with perturbations)</td>
<td>248</td>
</tr>
<tr>
<td>13.13</td>
<td>Scenario 1, Case 2, Examples 1-3 trajectories (Cartesian model, with perturbations)</td>
<td>248</td>
</tr>
<tr>
<td>13.14</td>
<td>Scenario 1, Case 2, Examples 1-3 control history (Cartesian model, circular orbit, with perturbations)</td>
<td>249</td>
</tr>
<tr>
<td>13.15</td>
<td>Scenario 1, Case 2, Examples 1-3 tracking error (Cartesian model, circular orbit, with perturbations)</td>
<td>249</td>
</tr>
<tr>
<td>Section</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>13.16</td>
<td>Scenario 1, Case 2 control history (Cartesian model, with perturbations and chief eccentricity of $e_c = 0.01$)</td>
<td>250</td>
</tr>
<tr>
<td>13.17</td>
<td>Scenario 1, Case 2 tracking error (Cartesian model, with perturbations and chief eccentricity of $e_c = 0.01$)</td>
<td>251</td>
</tr>
<tr>
<td>13.18</td>
<td>Scenario 1, Case 3, Examples 1-4 trajectories (Cartesian model, with perturbations)</td>
<td>253</td>
</tr>
<tr>
<td>13.19</td>
<td>Scenario 1, Case 3, Examples 1-4 control history (Cartesian model, circular orbit, with perturbations)</td>
<td>253</td>
</tr>
<tr>
<td>13.20</td>
<td>Scenario 1, Case 3, Examples 1-4 tracking error (Cartesian model, circular orbit, with perturbations)</td>
<td>254</td>
</tr>
<tr>
<td>13.21</td>
<td>Scenario 1, Case 3, Examples 1-4 control history (Cartesian model, with perturbations and chief eccentricity of $e_c = 0.01$)</td>
<td>255</td>
</tr>
<tr>
<td>13.22</td>
<td>Scenario 1, Case 3, Examples 1-4 tracking error (Cartesian model, with perturbations and chief eccentricity of $e_c = 0.01$)</td>
<td>255</td>
</tr>
<tr>
<td>14.1</td>
<td>Scenario 2, Case 1 trajectories (Cartesian model, without perturbations)</td>
<td>261</td>
</tr>
<tr>
<td>14.2</td>
<td>Scenario 2, Case 1 control history (Cartesian model, circular chief, without perturbations)</td>
<td>261</td>
</tr>
<tr>
<td>14.3</td>
<td>Scenario 2, Case 1 tracking error (Cartesian model, circular chief, without perturbations)</td>
<td>262</td>
</tr>
<tr>
<td>14.4</td>
<td>Scenario 2, Case 1 control history (spherical model, circular chief, without perturbations)</td>
<td>262</td>
</tr>
<tr>
<td>14.5</td>
<td>Scenario 2, Case 1 tracking error (spherical model, circular chief, without perturbations)</td>
<td>263</td>
</tr>
<tr>
<td>14.6</td>
<td>Scenario 2, Case 1 control history (spherical model, circular chief, with perturbations)</td>
<td>264</td>
</tr>
<tr>
<td>14.7</td>
<td>Scenario 2, Case 1 tracking error (spherical model, circular chief, with perturbations)</td>
<td>264</td>
</tr>
<tr>
<td>14.8</td>
<td>Scenario 2, Case 1 control history (spherical model, with perturbations and chief eccentricity of $e_c = 0.01$)</td>
<td>265</td>
</tr>
<tr>
<td>14.9</td>
<td>Scenario 2, Case 1 tracking error (spherical model, with perturbations and chief eccentricity of $e_c = 0.01$)</td>
<td>266</td>
</tr>
</tbody>
</table>
14.10 Scenario 2, Case 2 trajectories (spherical model, with perturbations) . . . . 268
14.11 Scenario 2, Case 2 control history (spherical model, with perturbations) . . 268
14.12 Scenario 2, Case 2 tracking error (spherical model, with perturbations) . . 269
14.13 Scenario 2, Case 2 tracking error (Cartesian model, with perturbations) . . 269
14.14 Scenario 2, Case 2 control history (spherical model, with perturbations and
chief eccentricity of $e_c = 0.01$) ....................................................... 270
14.15 Scenario 2, Case 2 tracking error (spherical model, with perturbations and
chief eccentricity of $e_c = 0.01$) ....................................................... 271
14.16 Scenario 2, Case 3, Examples 1-4 trajectories (Cartesian model, with per-
turbations) ................................................................. 273
14.17 Scenario 2, Case 3, Examples 1-4 control history (Cartesian model, with
perturbations) ................................................................. 273
14.18 Scenario 2, Case 3, Examples 1-4 tracking error (Cartesian model, with
perturbations) ................................................................. 274
14.19 Scenario 2, Case 3, Examples 1-4 control history (Cartesian model, with
perturbations and chief eccentricity of $e_c = 0.01$) .................. 275
14.20 Scenario 2, Case 3, Examples 1-4 tracking error (Cartesian model, with
perturbations and chief eccentricity of $e_c = 0.01$) .................. 275
15.1 Scenario 1, Case 1, Example 1 trajectories with $p_r = 5\%$ dispersions (spher-
ical model, with perturbations) .............................................. 280
15.2 Scenario 1, Case 1, Example 1 control history with $p_r = 5\%$ dispersions
(spherical model, circular chief, with perturbations) ................. 281
15.3 Scenario 1, Case 1, Example 1 trajectories with $p_r = 1\%$ dispersions (spher-
ical model, with perturbations) .............................................. 282
15.4 Scenario 1, Case 1, Example 1 control history with $p_r = 1\%$ dispersions
(spherical model, circular chief, with perturbations) ................. 282
15.5 Scenario 1, Case 1, Example 4 trajectories with $p_r = 5\%$ dispersions (spher-
ical model, with perturbations) .............................................. 284
15.6 Scenario 1, Case 1, Example 4 control history with $p_r = 5\%$ dispersions
(spherical model, circular chief, with perturbations) ................. 284
| 15.7 | Scenario 1, Case 1, Example 4 trajectories with $p_r = 1\%$ dispersions (spherical model, with perturbations) | 285 |
| 15.8 | Scenario 1, Case 1, Example 4 control history with $p_r = 1\%$ dispersions (spherical model, circular chief, with perturbations) | 286 |
| 15.9 | Scenario 1, Case 1, Example 4 trajectories with $p_r = 0.5\%$ dispersions (spherical model, with perturbations) | 287 |
| 15.10 | Scenario 1, Case 1, Example 4 control history with $p_r = 0.5\%$ dispersions (spherical model, circular chief, with perturbations) | 287 |
| 15.11 | Scenario 1, Case 2, Example 4 trajectories with $p_r = 10\%$ dispersions (Cartesian model, with perturbations) | 289 |
| 15.12 | Scenario 1, Case 2, Example 4 control history with $p_r = 10\%$ dispersions (Cartesian model, circular chief, with perturbations) | 289 |
| 15.13 | Scenario 1, Case 2, Example 4 trajectories with $p_r = 5\%$ dispersions (Cartesian model, with perturbations) | 290 |
| 15.14 | Scenario 1, Case 2, Example 4 control history with $p_r = 5\%$ dispersions (Cartesian model, circular chief, with perturbations) | 291 |
| 15.15 | Scenario 1, Case 3, Example 3 trajectories with $p_r = 10\%$ dispersions (Cartesian model, with perturbations) | 292 |
| 15.16 | Scenario 1, Case 3, Example 3 control history with $p_r = 10\%$ dispersions (Cartesian model, circular chief, with perturbations) | 293 |
| 15.17 | Scenario 1, Case 3, Example 3 trajectories with $p_r = 5\%$ dispersions (Cartesian model, with perturbations) | 294 |
| 15.18 | Scenario 1, Case 3, Example 3 control history with $p_r = 5\%$ dispersions (Cartesian model, circular chief, with perturbations) | 294 |
| 15.19 | Scenario 2, Case 1, Example 1 trajectories with $p_{SE} = 10\%$ dispersions (spherical model, with perturbations) | 297 |
| 15.20 | Scenario 2, Case 1, Example 1 control history with $p_{SE} = 10\%$ dispersions (spherical model, circular chief, with perturbations) | 298 |
| 15.21 | Scenario 2, Case 1, Example 1 trajectories with $p_{SE} = 5\%$ dispersions (spherical model, with perturbations) | 299 |
| 15.22 | Scenario 2, Case 1, Example 1 control history with $p_{SE} = 5\%$ dispersions (spherical model, circular chief, with perturbations) | 299 |
15.23 Scenario 2, Case 2, Example 2 trajectories with $p_{SE} = 10\%$ dispersions (spherical model, with perturbations) .................................................. 301

15.24 Scenario 2, Case 2, Example 2 control history with $p_{SE} = 10\%$ dispersions (spherical model, circular chief, with perturbations) ................................. 301

15.25 Scenario 2, Case 2, Example 2 trajectories with $p_{SE} = 5\%$ dispersions (spherical model, with perturbations) .................................................. 302

15.26 Scenario 2, Case 2, Example 2 control history with $p_{SE} = 5\%$ dispersions (spherical model, circular chief, with perturbations) ................................. 303

15.27 Scenario 2, Case 2, Example 4 trajectories with $p_{SE} = 10\%$ dispersions (spherical model, with perturbations) .................................................. 304

15.28 Scenario 2, Case 2, Example 4 control history with $p_{SE} = 10\%$ dispersions (spherical model, circular chief, with perturbations) ................................. 304

15.29 Scenario 2, Case 2, Example 3 trajectories with $p_{SE} = 5\%$ dispersions (spherical model, with perturbations) .................................................. 305

15.30 Scenario 2, Case 2, Example 3 control history with $p_{SE} = 5\%$ dispersions (spherical model, circular chief, with perturbations) ................................. 306

15.31 Scenario 2, Case 3, Example 4 trajectories with $p_{r} = 10\%$ dispersions (Cartesian model, with perturbations) .................................................. 307

15.32 Scenario 2, Case 3, Example 4 control history with $p_{r} = 10\%$ dispersions (Cartesian model, circular chief, with perturbations) ................................. 308

15.33 Scenario 2, Case 3, Example 4 trajectories with $p_{r} = 5\%$ dispersions (Cartesian model, with perturbations) .................................................. 309

15.34 Scenario 2, Case 3, Example 4 control history with $p_{r} = 5\%$ dispersions (Cartesian model, circular chief, with perturbations) ................................. 309
# LIST OF ALGORITHMS

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.1 Custom SOCP using Barrier Method for Autonomous RPO</td>
<td>195</td>
</tr>
<tr>
<td>10.2 Barrier Method SOC Gradients and Hessians Function</td>
<td>196</td>
</tr>
<tr>
<td>10.3 Barrier Method Line Search</td>
<td>196</td>
</tr>
<tr>
<td>10.4 Custom SOCP using PDIP Method for Autonomous RPO</td>
<td>202</td>
</tr>
<tr>
<td>10.5 PDIP Method SOC Gradients and Hessians Function</td>
<td>203</td>
</tr>
<tr>
<td>10.6 PDIP Method Line Search</td>
<td>203</td>
</tr>
<tr>
<td>Acronym</td>
<td>Definition</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>CPU</td>
<td>central processing unit</td>
</tr>
<tr>
<td>ESD</td>
<td>extended self-dual</td>
</tr>
<tr>
<td>FLOPS</td>
<td>floating point operations per second</td>
</tr>
<tr>
<td>FPGA</td>
<td>field-programmable gate array</td>
</tr>
<tr>
<td>GN&amp;C</td>
<td>guidance, navigation, and control</td>
</tr>
<tr>
<td>HCW</td>
<td>Hill, Clohessy-Wiltshire equations</td>
</tr>
<tr>
<td>KOZ</td>
<td>keep-out zone</td>
</tr>
<tr>
<td>KKT</td>
<td>Karush-Kuhn-Tucker conditions</td>
</tr>
<tr>
<td>LCP</td>
<td>linear complementary problem</td>
</tr>
<tr>
<td>LEO</td>
<td>low-Earth orbit</td>
</tr>
<tr>
<td>LP</td>
<td>linear program</td>
</tr>
<tr>
<td>LROE</td>
<td>linear relative orbital elements</td>
</tr>
<tr>
<td>LTI</td>
<td>linear time-invariant</td>
</tr>
<tr>
<td>LTV</td>
<td>linear time-varying</td>
</tr>
<tr>
<td>LVLH</td>
<td>local-vertical, local-horizontal</td>
</tr>
<tr>
<td>MARE</td>
<td>matrix algebraic Riccati equations</td>
</tr>
<tr>
<td>MILP</td>
<td>mixed integer linear programming</td>
</tr>
<tr>
<td>MPC</td>
<td>model predictive control</td>
</tr>
<tr>
<td>MSA</td>
<td>method of successive approximations</td>
</tr>
<tr>
<td>PASE</td>
<td>passive abort safety ellipse</td>
</tr>
<tr>
<td>PQP</td>
<td>parallel quadratic programming</td>
</tr>
<tr>
<td>QP</td>
<td>quadratic program</td>
</tr>
<tr>
<td>RIC</td>
<td>radial, in-track, cross-track LVLH frame</td>
</tr>
<tr>
<td>ROE</td>
<td>relative orbital elements</td>
</tr>
<tr>
<td>RPO</td>
<td>rendezvous and proximity operations</td>
</tr>
<tr>
<td>SDP</td>
<td>semidefinite program</td>
</tr>
<tr>
<td>SOCP</td>
<td>second-order cone program</td>
</tr>
<tr>
<td>SOF</td>
<td>safety of flight</td>
</tr>
<tr>
<td>SQP</td>
<td>sequential quadratic programming</td>
</tr>
<tr>
<td>TH</td>
<td>Tschauner-Hempel equations</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

Optimal trajectory planning for orbital rendezvous and proximity operations (RPO) is a long-standing area of research. This involves trajectory design for fuel-optimal orbital maneuvers, and is used extensively in guidance, navigation, and control (GN&C) applications while in the vicinity of a space object. The fundamental RPO problem is described by the relative motion of two satellites: a chief and deputy. The chief’s orbit is defined as the ‘target’ or ‘reference,’ and the relative motion of the deputy is typically described with respect to the chief’s reference frame. A mission with RPO aspects is commonly separated into two mission phases. This consists of the far-field and the near-field phases. The area of proximity operations falls into the category of near-field range, involving operations where the deputy satellite is generally within five kilometers separation from the chief [1].

Over the course of investigating RPO GN&C strategies and concept of operations, several relative orbital motion models have been developed. These may include dynamics models with respect to a circular or noncircular reference orbit, and with or without various perturbations [2–4]. Current formulations are applicable to a variety of RPO scenarios, including initial approach, safety of flight, relative orbit maintenance, and terminal approach [5,6]. The differences between dynamics models usually involve the benefits associated with obtaining greater accuracy, and the costs of complexity and/or numerical efficiency.

Most RPO trajectories are designed with multiple constraints on the state, control effort, and/or time, depending on mission objectives [1,7]. This makes the issue of determining feasible trajectories a substantial component of the comprehensive RPO optimal trajectory planning problem. The need for efficient on-board trajectory planning algorithms that can handle these constraints is significant, since ultimately, these algorithms need to provide autonomous, near real-time trajectory generation.
Research and development in this area is sustained by the desire for greater capabilities with regards to satellite formation flying, inspection, servicing, and docking [8, 9]. Furthermore, the requirement for autonomous trajectory generation is imperative, since the number of orbital debris and resident space objects is greatly increasing. Motivation for self-governing satellite technology is strong, as reliance on ground control is not always practical in a collision avoidance scenario. Therefore, it is critical for developments in optimal RPO trajectory planning to include safety of flight guarantees. Typically, this is comprised of key features in trajectory design such as passive abort (safety) ellipses, approach corridors, keep-out zones, and plume impingement constraints [1, 10, 11]. These elements serve to reduce the level of effort required by ground systems for trajectory monitoring, and safety of flight assurance.

It is evident that the optimization theory applied to this problem must have the capacity to provide optimal trajectory planning, and to readily include various safety of flight constraints. A variety of techniques exist, for example: nonlinear optimization, model predictive control (MPC), mixed integer linear programming (MILP), feedback control methods, and convex programming. Algorithms considered for on-board applications are often compared using factors such as accuracy, convergence rate, convergence guarantees, computational efficiency, and complexity [8]. Acknowledging these criterion, applications of convex optimization are recognized as superior since convex optimization algorithms have deterministic and guaranteed convergence properties to the global minimum [12]. This fact has resulted in other real-time applications of convex optimization such as Mars powered descent guidance [13, 14]. The objective of this research is to develop a prototype autonomous on-board RPO trajectory generation capability based on convex optimization.

1.1 Methodology

This research aims to develop optimal autonomous trajectory planning techniques, which are formulated specifically for orbital RPO. Several previously developed solutions to the autonomous RPO trajectory planning problem have been proposed. It is the goal to first discuss these current methods, to show the benefits and drawbacks for each, and to
fortify the practicality of convex optimization techniques in comparison. These are covered in detail in the initial literature survey, and are summarized here.

Nonlinear control methods clearly benefit from the direct implementation of nonlinear objective functions and constraints, however, these methods are computationally intensive and may provide solutions at local minima. For these reasons, nonlinear optimization may not be a realistic candidate for autonomous guidance [15]. Model predictive control methods implement a quadratic cost function, subject to linear equality and inequality constraints. Therefore, they cannot directly enforce a nonlinear spherical keep-out zone for collision avoidance, but instead use a separating hyper-plane that lies on the sphere boundary and enables the solution set to remain convex [15]. This method, however, may be considered over-constrained and computationally intensive due to the graph-search method that is ultimately employed, to determine a path around the debris [8]. In mixed integer programming formulations, constraints can be turned on or off as needed, and these constraints can be non-convex. This approach can be shown through analytical proof to have deterministic convergence properties making it a possible algorithm for on-board applications [16], but ultimately the algorithms are nonconvex and require branch-and-bound search methods. Feedback control with artificial potential functions are certainly advantageous since they give an analytical control solution and are inherently globally stable. However, due to the repulsive/attractive potential fields that they use to model path constraints, the solution may converge to a local minimum rather than a global minimum, making it a potentially undesirable approach to the autonomous trajectory planning problem [8].

In this work, convex optimization techniques are adopted and applied to optimal RPO trajectory planning scenarios. Convex problems are known to provide global optimality, have polynomial-time convergence properties, and are solved with rapid, numerically stable, and efficient solvers [17,18]. The problem of optimal trajectory planning for orbital RPO is cast as a specific convex optimization problem, a second-order cone program (SOCP) [19], with the objective to minimize propellant. The SOCP is self-starting and has deterministic convergence properties, making it an ideal candidate for solving this problem and providing
autonomous real-time results [20]. In this analysis, all of the convex trajectory optimization problems are cast as fixed-final time problems. While this may be limited, it is the opinion of the author that operational constraints such as lighting, communications, and the overall mission timeline will drive the value of the transfer time or its maximum value. The trajectory planner developed herein can accommodate any of these cases.

Several traditional fixed-time RPO scenarios are considered, including initial approach, satellite inspection or way-point following, and final approach trajectories. In addition, several new scenarios involving trajectory planning to ensure safety of flight (SOF) are also presented. This formulation is based upon the development of constraints which ensure the deputy satellite is in a passively safe relative ellipse at all times. A variety of constraints may be imposed on these trajectories, in addition to the fixed final-time. These include initial/final boundary conditions, maximum thrust availability, approach corridors, keep-out zones, and safety of flight requirements.

In the case of traditional RPO scenarios, all initial and final conditions are interpreted as the initial and final relative position and velocity of the deputy, in the desired frame of reference. The other constraints of interest for these scenarios are maximum thrust acceleration, approach corridors, and a spherical keep-out zone (KOZ). These are optionally enforced for both nadir-pointing and controlled or uncontrolled spinning spacecraft frames of reference. The standard Hill, Clohessy-Wiltshire (HCW) equations in both Cartesian and spherical coordinates are utilized when a trajectory relative to a local-vertical, local-horizontal (LVLH) frame is desired. In the case of optimal trajectory planning relative to a controlled or uncontrolled spinning spacecraft, a newly developed relative orbital dynamics model with respect to a spacecraft-fixed frame is utilized.

In the new SOF scenarios, the initial and final conditions are interpreted as initial and final safety ellipses. These correspond to a set of relative orbital elements, which can be used to calculate relative position and velocity. However, as will be seen, some terminal relative orbital element constraints may be relaxed, in order to provide a more optimal transfer from one relative ellipse to another. The constraints in these scenarios include maximum
thrust acceleration, and safety of flight constraints on the relative ellipse size, position, and orientation.

The solution to the SOCP can be stated in terms of the primal objective function and the dual objective function (from Lagrange’s constrained optimization theory), where the dual function is guaranteed to be convex [20]. Provided there is strong duality, which convex problems are known to have, then there are guaranteed optimal solutions to both the primal and dual problems, and the solutions are identical. Solving the primal-dual problem greatly increases the capabilities of the SOCP and its simplicity makes it advantageous when compared to more complex methods such as the model predictive control algorithms [11]. Additionally, the number of iterations required for convergence to the global minimum can be determined a priori, making convex optimization algorithms more desirable in comparison to algorithms with non-deterministic convergence [20, 21]. Several optimal guidance and trajectory planning formulations that implement the SOCP have been developed. These include the Mars landing powered descent guidance work for the Mars Science Laboratory [13, 14], and autonomous RPO trajectory planning using successive approximations to a linearized gravity model [22].

In addition to adopting convex optimization techniques and the SOCP, a philosophy of trajectory-following is assumed. That is, once the RPO planner generates an optimal relative position and velocity trajectory, it is the responsibility of the on-board position and velocity control systems to follow the prescribed optimal trajectory. Plume-impingement constraints can be cast as convex SOCP [22], however, it was decided early in this research to exclude this constraint, as it requires a specific vehicle and/or thruster configuration. These assumptions eliminate the need to model specific thruster locations and orientations, the rotational dynamics of the chaser spacecraft, and the on-board control system within the optimization algorithm. The result is a simplified optimization problem without a significant loss in overall performance.

In cases where trajectory constraints cannot be convexified, or cast directly in an SOCP, the method of sequential convex programming (SCP) is employed [11]. This involves formu-
lating an approximate convex optimization problem, subject to approximations for each non-
convex constraint, and iteratively solving an SOCP until a convergence criteria is met [21].
The nonconvex constraints consist of the spherical KOZ, and the safety of flight constraints.
The SCP method has been applied to a variety of aerospace RPO-type scenarios, which
include nonlinear and nonconvex objective functions, as well as nonlinear dynamics mod-
els [22, 23]. While the SCP formulation has not yet been shown to have guaranteed con-
vergence in all cases, preliminary results from the specific RPO problems in this work show
fast convergence, requiring only a few sequential iterations.

There are clearly many different types of RPO trajectory problems, especially when
considering variable transfer times, boundary conditions, and desired trajectory constraints.
As a result, five different prototype RPO problem formulations are developed, and will be
used to solve for the different elements of the overarching RPO problem. Each formulation
includes different constraints, depending on the specific trajectory to be planned. These are
summarized as follows:

1. **Problem Formulation I**: Minimum propellant, fixed final-time transfer trajectories
   (e.g. from point A to point B), with maximum acceleration constraints, relative to a
   chief-centered reference frame (LVLH or other rotating frame). The optimal solutions
generally require two or more impulsive maneuvers, or continuous maneuvers when
the maximum acceleration constraint is enforced. Convergence is guaranteed with
deterministic CPU requirements.

2. **Problem Formulation II**: Minimum propellant, fixed final-time transfer trajectories
   (e.g. from point A to point B) with maximum acceleration, and approach corridor con-
   straints, relative to a chief-centered reference frame (LVLH or other rotating frame).
The optimal solutions generally require two or more impulsive maneuvers, or contin-
uous maneuvers when the maximum acceleration constraint is enforced or while near
the boundary of the approach corridor constraint. Convergence is guaranteed with
deterministic CPU requirements.
3. **Problem Formulation III**: Minimum propellant, fixed final-time transfer trajectories (e.g. from point A to point B) with maximum acceleration, approach corridor, and spherical KOZ constraints, relative to a chief-centered reference coordinate frame (LVLH or other rotating frame). Optimal solutions generally require two or more impulsive maneuvers, or continuous maneuvers when the maximum acceleration constraint is enforced or while near the boundary of the approach corridor or spherical KOZ constraints. The method of sequential convex optimization is currently required in the KOZ cases, though all cases examined in this work show convergence in just a few iterations.

4. **Problem Formulation IV**: Minimum propellant, fixed final-time transfer trajectories (e.g. from relative ellipse A to relative ellipse B), with maximum acceleration constraints, relative to a chief-centered reference frame. The optimal solutions generally require two or more impulsive maneuvers, or continuous maneuvers when the maximum acceleration constraint is enforced. Convergence is guaranteed with deterministic CPU requirements.

5. **Problem Formulation V**: Minimum propellant, fixed final-time transfer trajectories (e.g. from relative ellipse A to relative ellipse B), with maximum acceleration constraints, and SOF constraints, relative to a chief-centered reference frame. The optimal solutions generally require two or more impulsive maneuvers, or continuous maneuvers when the maximum acceleration constraint is enforced. The method of sequential convex optimization is currently required in the SOF scenarios, though all cases examined in this work show convergence in just a few iterations.

For each problem formulation shown above, specific cases are examined, depending on the different phases of RPO. The trajectory phases considered in this research are formulated according to the typical time line of an RPO mission. In the traditional RPO scenario, they are separated into an Initial Approach Phase, a Way Point Following/Inspection Phase, and a Final Approach Phase. In the new SOF RPO scenario, the cases are very similar,
but ensure passive safety. These are separated into the Safe Initial Approach Phase, Safe Traveling Ellipse Phase, and Safe Final Approach Phase. A summary is provided here.

**Scenario 1** - Traditional Rendezvous, Inspection, and Final Approach:
- **Case 1** - Initial Approach Phase
- **Case 2** - Way-point Following/Inspection Phase
- **Case 3** - Final Approach Phase

**Scenario 2** - Safe Rendezvous, Traveling Ellipse, and Final Approach:
- **Case 1** - Safe Initial Approach Phase
- **Case 2** - Safe Traveling Ellipse Phase
- **Case 3** - Safe Final Approach Phase

The analysis to be carried out for the scenarios and mission phases described above is divided into two groups. The first is to analyze the results from the optimal trajectory planner, while the second is to evaluate how well the planned trajectories perform in a nonlinear simulation, using the trajectory following method. Trajectory results from the convex RPO planner are included in earlier chapters, directly after the formulation and development of each. In the Performance Analysis chapters, nonlinear simulation results are included, which compare the optimal planner trajectory and propellant use to the simulation trajectory and propellant.

The remainder of this dissertation is organized as follows. Previous research and development relevant to this area of study is presented in the Chapter 2 literature survey. This is comprised of other optimal guidance techniques, relative orbital motion models, and current safety of flight ideas. A detailed review of convex optimization theory, and example applications is provided in Chapter 3. The thesis statement and research overview follows in Chapter 4, which includes a detailed description of the analysis to be performed. Chapters 5 and 6 develop the relative orbital motion models to be used in the convex RPO planner. These developments are followed by Chapters 7, 8, and 9, which formulate the convex problems, and show trajectory planner results. Chapter 10 shows a specific application in
formulating a custom convex RPO solver. Chapter 11 presents results on the required CPU and memory to solve the RPO problem. The nonlinear simulation models are developed in Chapter 12, followed by the nonlinear simulation results in Chapters 13, 14, and 15. Final conclusions are contained in Chapter 16, along with future work.
CHAPTER 2
LITERATURE SURVEY

There exists an extensive amount of literature related to the optimal RPO trajectory planning problem and related problems. The relevant literature also varies greatly from relative orbital motion models, to safety of flight concepts, optimization techniques, and control theory. The purpose of this chapter is to present a broad overview of autonomous aerospace applications, relative satellite motion models, safety of flight assurance techniques, and optimal satellite guidance methods that are currently used. Each of these sections covers a specific topic related to general autonomous RPO trajectory planning. At the end of each section the scope of the problem is narrowed down, and the elements of each RPO category that are directly related to this research are accessed. This allows the main takeaway of each portion of this literature survey to focus on the primary concepts that contribute the largest advances to this field of research.

2.1 Autonomy, Optimal Control, and Convex Optimization: Aerospace Applications

There has been much prior effort to the application of autonomy in aeronautical and aerospace engineering in general. The vast majority of these applications originate in the field of robotics, and more specifically, to the autonomy of drones and UAVs [24]. For example, the autonomous lateral-directional control using Eigenstructure assignment [25], and optimal path planning for UAVs [26]. Several methods for optimal trajectory planning have been developed which efficiently handle nonconvex constraints using a method known as sequential convex programming (SCP). One such application involves generating collision-free trajectories for quadcopters [27], and another applies to multi-robot navigation and formation control with obstacle constraints [28, 29]. A more general approach to robotic path planning using SCP methods is developed Chen, Cutler, and How [30]. These next
few paragraphs focus specifically on aerospace applications of optimal control and trajectory design using convex optimization.

Optimal control and trajectory design in the field of aerospace has a very rich and refined history. First progress began with the optimization techniques developed by Lev Pontryagin, in his formulation of the Hamiltonian and Maximum Principle in the 1960’s [31]. Following his work, other revolutionary contributions include the introduction of primal vector theory, co-state equations, and switching functions, from Theodore Edelbaum and Derek Lawden’s work, which made significant advancements to the area of optimal controls [32, 33]. Many important problems resulted from their work, including Edelbaum’s "number of impulses" problem, which led to an affluent evolution of modern optimization and control theories.

In recent years, convex optimization theory has dominated the field of optimal trajectory planning and led to a rich development of different formulations. These methods apply to a great variety of scenarios, from planetary precision landing, to satellite control, and to the area of RPO. The solution to optimal control problems involves implementing calculus of variations to determine optimal trajectories and controls, and in some cases analytic (or approximate analytic) solutions may be developed for either RPO scenarios or interplanetary trajectories [34-36]. The first developments to emerge involve the coordination and control of multiple spacecraft by Tillerson [37, 37]. This is closely related to other formulation-flying models for relative motion and trajectory optimization by Wu et al. [38], and Goel et al. [23].

Significant developments have also been made for planetary landing problems, in the work of Acikmese and Blackmore [13,14,39–42]. Other enhancements to this problem include interpolation-enhanced methods by Scharf [43,44], asteroid power descent by Pinson and Lu [45], and the results from Harris and Acikmese [46,47] for maximum-divert landings (applications of large-divert problems for rocket guidance are also shown by Scharf et al. in [48]). Custom algorithm development for landing is documented extensively in a paper by Dueri [49]. Two SCP approaches (one developed by Szmuk [50], and another Wang [51]) are the most recent additions to these landing problems. Other trajectory-generation related works cover convex trajectory planning problems for geostationary station-keeping
by Bruijn et al. [52, 53] and continuously-constrained trajectory formulations by Deaconu et al. [54]. Other applications apply to attitude control as in Kim’s work [55], and the work by Kjellberg [56], where nonconvex attitude constraints were implemented.

Full nonlinear solution methods to the two-body trajectory planning problem have been recently developed, which use convex programming algorithms in what is termed a method of successive approximations [22, 57]. The MSA is essentially an outer loop on the convex optimization problem, whereby the optimal solution to the full convex optimization problem is calculated repeatedly, until a convergence criterion for the outer loop is met. The paper by Lu and Liu formulates a relaxed problem (proven to be a lossless relaxation technique) for the control variable, and implements the full nonlinear equations of motion (including J2 and drag) [22]. The relaxed problem is formulated with the addition of slack variables, which allows convenient implementation of the plume impingement constraint. The two-body equations of motion are expressed using a linear gravity model, and then iterated to determine the optimal solution to the full nonlinear problem. There are no assumptions about the reference orbit of the chief vehicle.

Lu’s and Liu’s approach is solved using a second-order cone program, which implements the primal-dual interior point algorithm, allowing for faster convergence rates and guaranteed convergence [58]. He formulates the problem as a second-order cone problem (SOCP), which includes a linear cost function, subject to linear equality constraints for dynamics and second-order conic inequality constraints on the slack variables. The constraints considered are terminal state constraints, approach corridor constraints, terminal thrust plume impingement constraints, thrust maximum magnitude constraints, and a KOZ constraint [22]. A first-order constraint for the KOZ is used, to convexify the domain, and then updated on each iteration of the method of successive approximations.

Convex optimization techniques for optimal guidance have also been tested in real-time environments. The precedent application is Acikmese and Ploen’s Mars powered descent guidance algorithm. It provides a convex optimal control formulation subject to nonconvex control constraints [13,14]. Similar results and documentation are provided by Liu, Shen, and
Lu in [59], and apply specifically to an SOCP problem. There are also methods formulated by Lu et al. and Acikmeze which discuss how to solve inherently nonconvex problems through convex optimization, and are used in the Mars Science Laboratory descent guidance [13, 21]. This allows the control maximum and minimum acceleration levels to be convexified and enforced in the approach. Due to guaranteed, deterministic convergence to a globally optimal solution, convex optimization algorithms are an exceptional option for real-time, on-board implementation.

2.2 Relative Satellite Motion Models

Models for relative motion describe the dynamics of one satellite relative to another, and may be modeled with a variety of different reference orbits. The reference orbit defines the orbit in which one vehicle’s reference frame exists, where all motion of the other vehicle is described as observed from the reference frame. A brief introduction to some relative motion models is provided here. For a more comprehensive list, see the survey by Sullivan, Grimberg, and D’Amico [60].

Reference orbit models range from low eccentricity orbits to highly eccentric orbits, and many models include perturbations such as J2, drag, solar radiation pressure, and third-body terms [4, 61, 62]. These models can be applied to many different scenarios in the terminal rendezvous and proximity operations phase of a satellite’s mission. Most models can be generally classified according to the type of nominal reference orbit considered, primarily between circular or elliptic and perturbed or unperturbed where the reference vehicle is cooperative [61], however, in some instances a non-cooperative vehicle is also considered [63–65]. Many first-order models begin with the linearization of the relative motion dynamics in a gravitational field defined by an inverse square law, and may include first-order terms from perturbations.

Two of the most common formulations for terminal RPO are the Hill-Clohessy, Wiltshire (HCW) and Tschauser-Hempel (TH) equations [66, 67]. The HCW formulation provides a linear time invariant (LTI) solution for a chief satellite in a circular reference orbit, and a deputy in a nearby low-eccentricity orbit. The unperturbed relative motion of the deputy is
well defined using the HCW model, however, for perturbed and/or elliptic reference orbits, the HCW equations may be considered limited. The TH equations offer a closed-form linear time varying (LTV) solution for relative motion about an elliptic chief reference orbit, and may be used in elliptic orbital RPO. Both the HCW and TH linearized equations are typically represented in a local-vertical local-horizontal (LVLH) frame, and may be coordinatized in either rectangular or curvilinear coordinates [68, 69]. For two orbits of arbitrary eccentricity, alternative time-varying models similar to the TH equations have been developed such as the time-explicit representation by Melton [70], the time-varying transformations proposed by Sherrill et al. [71], or the model with time as the independent variable, from Broucke [72]. The results from many recent relative orbital motion models is to make improvements on the state transition matrix formulation, and test these results in a nonlinear simulation to verify model accuracy and validate assumptions [2, 73].

Other types of models are derived from constants of motion, orbital elements, or other orbital parameters, as first suggested by Hill [74]. These methods allow for derivations of higher-order extensions to the linearized HCW and TH equations, which may be greatly simplified and written in closed-form under certain approximations [75–77]. Previous work has also been done by Karlgaard and Lutze using the method of multiple scales to formulate second-order equations of relative motion [78]. Another second-order formulation is derived by Sengupta, Vadali, and Alfriend which includes perturbations [79]. Relative orbital elements have also been used in conjunction with Hill’s equations to distinguish higher-accuracy mappings of relative motion dynamics [80]. For each of these relative orbital motion models, the reference frame varies depending on the problem formulation. Condurache and Martinus’s work provides a more general approach to Keplerian dynamics in a rotating reference frame, where the results presented are frame independent [81].

The HCW model is the fundamental model for this research since it provides a simple, closed-form, and convenient way to describe the relative motion of a deputy satellite in close proximity of a chief satellite’s circular reference orbit. These equations can be utilized in many scenarios, where depending on the degree of proximity, may sufficiently model
the dynamics even for low-eccentricities and perturbed orbits [71, 82]. The HCW system can be written as six first-order LTI differential equations, defined in terms of the relative position and velocity in the chief’s LVLH frame. The linearized equations of motion for the deputy satellite with respect to the rotating chief-centered LVLH frame, and including control accelerations, are [68]

\[
\begin{align*}
\ddot{x} - 2\omega \dot{y} - 3\omega^2 x - a_{Tx} &= 0 \\
\dot{y} + 2\omega \dot{x} - a_{Ty} &= 0 \\
\ddot{z} + \omega^2 z - a_{Tz} &= 0
\end{align*}
\] (2.1)

where the deputy satellite’s relative position and velocity are defined as

\[
\begin{align*}
\mathbf{r}_{\text{rel}} &= \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \\
\mathbf{v}_{\text{rel}} &= \dot{\mathbf{r}}_{\text{rel}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}
\end{align*}
\] (2.2)

and the thrust acceleration control is

\[
\mathbf{u} = \mathbf{a}_T = \begin{bmatrix} a_{Tx} \\ a_{Ty} \\ a_{Tz} \end{bmatrix}
\] (2.3)

The relative position and velocity of the deputy satellite completely define the deputy’s relative orbit with respect to the Cartesian LVLH frame. For purposes of convention, the radial, in-track, cross-track (RIC) version of the LVLH frame is preferred, as presented in Schaub and Junkins [83].
2.3 Safety of Flight

Safety of flight methods generally stem from the ability to guarantee zero probability of collision, given the trajectory errors and uncertainties of two nearby vehicles, or two vehicles with intersecting orbits. These uncertainties are typically a result of modeling, navigation, and control errors. There are two main elements of collision avoidance, active safety of flight and passive safety of flight. These may also be divided into short-term and long-term collision avoidance guarantees, where active collision avoidance is generally associated with short-term, and passive with long-term [84]. Work in short-term collision probability determination has proven to give fast and reliable solutions which may be used to calculate collision avoidance maneuvers [85, 86]. Passively safe trajectories guarantee safety of flight over much longer times and over wider flight regimes, including in the event of on-board power loss or control failure [6]. Methods have been developed and applied to guarantee safe RPO for spacecraft swarm design with operational constraints by Koenig [87]. Many problems for spacecraft swarms must also adopt safe trajectory design, which is inherent for spacecraft cooperation and coordination, as shown by Tillerson, Inalhan, and How [37, 88, 89].

The safety of flight (SOF) concepts that relate directly to the RPO problem formulation can be summarized with a few key terms. These include the passive abort safety ellipse (PASE), the approach corridor, and a keep-out zone (KOZ). Each represents a constraint on the current or future relative position of the deputy satellite. A PASE is a convenient way to ensure that, in the case of an internal failure on the deputy satellite, there is zero probability that it will drift into the chief satellite, within the range of motion described by the linear assumptions [10]. Approach corridors and keep-out zone constraints can also be enforced, but generally are not passive transfer methods, and may not ensure terminal SOF. However, the implementation of these constraints is still valid as an active RPO transfer solution, and can be enforced over the entire duration of the trajectory. The approach corridor may be desired for a variety of reasons, for instance, optics/sensor constraints, structural avoidance constraints, and/or other operational considerations [16]. Other constraint varieties for SOF include constraints on relative velocity, and control effort. One example of a constraint on
control that has been implemented is the plume impingement constraint [22].

In the case of determining optimal PASE formulations, several approaches have been developed. For example, control formulations using ROEs from Yin and Han’s work decouples the in-plane and out-of-plane motion for a separated control analysis [77]. Hybrid combinations of Cartesian and orbital element feedback laws, and feedback control using mean orbital elements have also been developed by Schaub and Alfriend, with applications in the area of formation flying [90, 91]. Examples and constraints for SOF using infinite horizon passive (or receding horizon) collision avoidance are established in the paper by Breger and How, where cost/benefit tradeoffs are examined [6]. With the focus of this research on SOF formulation using the HCW equations, a new set of relative orbital elements characterized by these equations are the core of the SOF formulation.

The HCW ROE states, developed by Lovell and Spencer, describe the size and position of a relative ellipse, in terms of the original HCW LVLH states [92]. This set of ROEs is considerably effective for interpreting the in-plane and out-of-plane dynamics of relative ellipses in the Cartesian LVLH frame. Due to this property, the ROEs lend themselves very nicely to the implementation of PASE constraints. These relative orbital elements have also been used to develop relative teardrop trajectory design and guidance algorithms by Prince and Cobb [93], and optimal range-observability maneuvers for satellite relative navigation by Franquiz and Udrea [94], which applies well to the range-observability problem documented by Woffinden [95-97]. Similar to the original HCW state vector, there are four in-plane states and two out-of-plane states. The in-plane HCW relative orbital elements are the $x_r$, $y_r$ location of the center of the relative ellipse, the semi-major axis of the 2×1 ellipse, $a_r$, and the in-plane phase angle $E_r$. The two out-of-plane states are the cross-track amplitude, $A_z$, and the out-of-plane phase angle $\psi_z$. These states are shown separately for the in-plane and out-of-plane motion, in Figure 2.1.
Two different classes of relative ellipses, depending on the ROEs, are distinguished. The relative ellipse is either a stationary ellipse, or a traveling ellipse, and is either out-of-phase, or in-phase. Stationary ellipses have a zero radial center of motion \((x_r = 0)\) and therefore ideally do not change position in the LVLH Cartesian frame. Two different types of stationary ellipses are also described, as either a circumnavigating or an offset safety ellipse. Traveling ellipses have a nonzero radial center of motion \((x_r \neq 0)\), and therefore ‘drift’ in either the positive or negative in-track \((y_{LVLH})\) direction. Whether or not a relative ellipse is in-phase depends on the in-plane and out-of-plane phase angles. When these phase angles are equal \((E_r = \psi_z)\), the relative ellipse is termed an in-phase ellipse, and when they are not \((E_r \neq \psi_z)\), then the ellipse is out-of-phase. The difference between the two angles is called the phase difference, \(\gamma = E_r - \psi_z\) \([92]\). Selecting size and location via specification of \(a_r\), \(A_z\), \(x_r\), and \(y_r\) ROEs, as well as ensuring range of possible values for the phase difference of the relative ellipse, is enough information to ensure a PASE for the deputy satellite. An example of a PASE that is both stationary and in-phase is shown in Figure 2.2.
2.4 Current Optimal Guidance Techniques

A brief overview of several current optimization techniques is provided in this section. Each subsection summarizes a different approach to trajectory optimization, and the inclusion of constraints using Lagrange classic optimization theory. Advantages and disadvantages to each method are briefly discussed, and may be compared by criterion outlined in Introducing Computational Guidance and Control by Ping Lu [58]. Convex optimization methods are introduced here, and discussed in greater detail in Chapter 3. For more information and a comprehensive list, see survey papers by Betts [98] and Zagaris [8].

2.4.1 Model Predictive Control

Model predictive control (MPC) is the first method for discussion due to its very rich history, with literature that is widely available and applicable in many areas of controls engineering. The objective in MPC formulations is to minimize a quadratic cost function, subject to linear equality and inequality constraints, making it solvable by a parallel quadratic program (PQP) method as shown by Weiss et. al. [99]. In the area of vehicle robotics and control, much work has been done developing MPC algorithms that have
guaranteed feasibility and completion times [15]. MCP methods have also efficiently solved quadcopter interception problems which involve similar constraints [100], however, many aerospace problems require inequality constraints with higher-dimensional vectors.

Thrust constraints, line-of-sight constraints and KOZ constraints may all be considered in an MPC problem, but must be linearized. Therefore, thrust magnitude constraints cannot be enforced explicitly. This is shown in the development of MPC techniques for small-body proximity operations by Carson [101]. The KOZ constraint is enforced by placing a plane on the specified side of the debris KOZ sphere. As the spacecraft moves, the plane is rotated, similar to a MPC/convex optimization hybrid method from Morgan and Chung, which implements sequential convex programming [102]. The MPC may also include J2 effects which are implemented by linearizing Gauss’ variational equations for ROEs. However, these KOZ problem formulations may be considered over-constrained, due to the plane constraint, and computationally intensive, due to the graph-search method that is employed to determine an optimal path around the KOZ.

2.4.2 Nonlinear Optimization

In general, many nonlinear optimization techniques exist, however in nearly all cases difficulties arise regarding initial guesses for the determination of a global optimal solution [103]. Often, dealing with local minima and saddle points can be problematic, as well as algorithm stability, therefore a practical solution must be realized [104–106]. A sequential quadratic programming (SQP) (gradient based) solver approach from Luo et. al. allows for the nonlinear, nonconvex KOZ constraint and is a viable option for on-board applications [107]. But, this still encompasses a nonlinear optimization method. Therefore, most of these methods are computationally intensive, and require a user-input initial guess [8]. In recent work by Lu, a fully nonlinear RPO problem was cast as a convex optimization problem, by approximating the nonlinear gravity model with a linear gravity model [22]. The solution requires an iterative approach to the convex problem using the method of successive approximations (MSA), and is shown to converge in only a few iterations.
2.4.3 Mixed Integer Linear Programming

Mixed integer linear programs (MILP) implement constraints by using binary variables which can be switched on or off as needed. These include nonconvex KOZ avoidance constraints. Applications of MILP in the area of real-time trajectory planning for UAVs has led to increased confidence and fidelity of these problems. These results are given by Kamal [108] and Culligan [109,110]. Proof of convergence criteria are included in a formulation by Richards and How, guaranteeing that the algorithm converges within a fixed amount of time [16]. These algorithms are considered a good candidate for on-board RPO guidance, provided that the convergence guarantee is met, which greatly depends on the solver. Currently, solvers which implement MILP in conjunction with convex optimization techniques exist, although the MILP problem is inherently nonconvex and requires search methods such as branch-and-bound [111].

2.4.4 Feedback Control

Feedback control methods also have a very rich history in trajectory optimization methods [98]. These provide an analytic control solution, and are globally stable [8]. They model constraints using artificial potential fields, which adaptively change the weights on the potential functions to improve performance. One example is the repulsive potential field used to model path constraints for keep-out zones on the trajectory, implemented by Munoz et. al. [112]. However, due to the nature of these constraints, the solution may converge to a local minimum rather than a global minimum. Dual quaternions are also used in one formulation by Filipe and Tsiotras, in which translation and attitude control is combined, but, control or state constraints are not implemented [113].

2.4.5 Convex Optimization

Convex optimization techniques involve minimizing convex functions over convex sets. These types of problems have a convex objective function, linear equality constraints, and convex inequality constraints. The solutions to optimal control problems using convex optimization results in the use of calculus of variations to determine an optimal solution to
a set of nonlinear optimality conditions [20, 103]. In many cases, as in this research, the convex objective function can be formulated as a linear function via a lossless convexification technique [21, 114]. The benefits of convex optimization include the polynomial-time convergence rates, guaranteed convergence to the global minimum, and the infeasible-start techniques [20, 115]. The focus of the research in this report is formulating convex problems as SOCPs. A more in-depth discussion of convex optimization and the SOCP are provided in Chapter 3.
CHAPTER 3

CONVEX OPTIMIZATION AND THE SOCP

3.1 Convex Optimization

Convex optimization is defined as the minimization of a convex function over a convex set. It encompasses a subset of general optimization problems, including linear programming (LP), quadratic programming (QP), second-order-cone programming (SOCP), semidefinite programming (SDP), and geometric programming [20]. A diagram of the programming hierarchy is shown in Figure 3.1, where it is shown that the semidefinite program subsumes all other convex programming types, excepting geometric programming [116]. The domain shown here as $\mathcal{PC}$ contains all convex programs.

![Venn Diagram of Convex Programming Hierarchy](image)

Fig. 3.1: Venn diagram of convex programming hierarchy

Convex optimization programs have deterministic convergence properties, however convergence error depends on values for maximum allowable residual error, to terminate the program [20, 103]. Convex optimization methods also guarantee convergence to the global minimum, and can be initiated via an infeasible start method [116]. While this does not guarantee the problem to be feasible, it does eliminate the need for an initial guess, thus making convex optimization a practical solution to the autonomous RPO problem. Conditions of optimality for a convex optimization problem are derived from the Karush-Kuhn-Tucker
(KKT) conditions. The KKT conditions provide both necessary and sufficient conditions for a convex optimization problem, and take a form similar to the Euler-Lagrange equations for optimal control problems [117]. The solution method for inequality constrained convex optimization problems consists of interior point methods whereby efficient interior-point methods are used [118, 119], or the inequality constraint boundaries are enforced using a log-barrier function [120]. Equality constraints for the primal and dual problems (described below) can be solved by a variety of gradient descent methods, such as steepest descent or Newton methods [20].

3.1.1 Duality and Convergence

The solution to a convex optimization problem can be stated in terms of the primal objective function and the dual objective function (from Lagrange’s constrained optimization theory), where the dual function is guaranteed to be convex [20]. Provided there is strong duality, which convex problems are known to have via Slater’s condition, there are guaranteed optimal solutions to both the primal and dual problems and the solutions are identical. Solving the primal and/or dual problem greatly increases the capabilities of a convex optimization algorithm, and its simplicity makes it advantageous when compared to more complex methods such as the model predictive control algorithms [11]. Due to the fact that some problems are solved more efficiently via either the primal or the dual, the most general convex optimization algorithms must include the ability to choose between the two, depending on the problem complexity and constraints [111]. Many convex optimization solvers developed within the last decade have made computational improvements by solving both the primal and dual problems at the same time [121]. These are commonly named self-dual embedding methods. As a result of convex duality, there is a great amount of flexibility when it comes to solving either primal-only, dual-only, primal-and-dual, or even formulating a means to intelligently switch between the two.

The solution to a convex optimization problem requires determining the optimal solution to a system of nonlinear equations using Newton-step methods, however, since the objective and all constraint functions are convex, the convergence properties of the nonlinear
equations are conveniently described. The overall centering steps for a given algorithm accuracy, for example, is written as a function of the desired accuracy, number of constraints, and a few selected initialization parameters. The total number of Newton-steps is written as a function of the number of centering steps, constraints, backtracking line search parameters, and a few set tolerances [20, 103]. Thus, the number of iterations required in a convex programming algorithm for convergence to the global minimum can be determined a priori, making convex optimization algorithms more desirable in comparison to algorithms with non-deterministic convergence [19]. Convex algorithms are also known to have polynomial-time convergence, where the overall solution time is a polynomial function of the problem size [18].

3.1.2 General Form of a Convex Optimization Problem

The general form of a convex optimization problem is as follows [19]. The objective is to determine optimal values \( x^* \in \mathcal{X} \), where \( \mathcal{X} \subset \mathbb{R}^n \) is a feasible set. The problem is written as

\[
\begin{align*}
\text{minimize} & \quad f_0(x) \\
\text{subject to} & \quad f_i(x) \leq 0, \quad i = 1, \ldots, m \\
& \quad g_j(x) = 0, \quad j = 1, \ldots, p
\end{align*}
\]

where the objective function \( f(x) : \mathbb{R}^n \to \mathbb{R} \), is convex, the inequality constraint functions \( f_i(x) \) for \( i = 1, \ldots, m \), are convex functions, and all equality functions, \( g_j(x) \) for \( j = 1, \ldots, p \), are affine. Variants of convex optimization programs each fit into the problem formulation shown in Equation 3.1, and include LP, QP, SOCP, SDP, and geometric programs, although geometric programs are not convex but can be converted to a convex form.

3.2 Common Forms for the Second-order Cone Program

The characteristics of a second-order cone program include optimization over an affine objective function, subject to second-order conic inequality constraints, and affine equality
The objective function is written as an inner product between vector $f \in \mathbb{R}^n$ and all primal variables $x \in \mathbb{R}^n$, defined here as
\[
\langle f, x \rangle = f^T x = \sum_{k=1}^{n} f_k x_k \tag{3.2}
\]

A typical SOCP consists of $m$ second-order conic constraints, where the dimension of the $i$th cone is $n_i$, for $i = 1, \ldots, m$. All second-order conic constraints take the form
\[
||A_i x + b_i||^2 \leq c_i^T x + d_i, \quad i = 1, \ldots, m \tag{3.3}
\]

where $A_i \in \mathbb{R}^{(n_i-1) \times n}$. This is termed a second-order cone constraint since it requires the affine functions $(c_i^T x + d_i, A_i x + b_i)$ to each lie in the second-order cones in $\mathbb{R}^{n_i}$, for $i = 1, \ldots, m$ [20]. The norm in this constraint is the Euclidean norm, or $\ell_2$ norm, which for a vector $z \in \mathbb{R}^n$ is defined as
\[
||z||_2 = (z^T z)^{1/2} = (z_1^2 + \ldots + z_n^2)^{1/2} \tag{3.4}
\]

The problem may also include $p$ affine equality constraints, defined by vectors $g_j \in \mathbb{R}^n$ and scalar $h_j \in \mathbb{R}$, for $j = 1, \ldots, p$. Each equality constraint is the inner product
\[
\langle g_j, x \rangle = g_j^T x = \sum_{i=1}^{n} g_{ji} x_i = h_j, \quad \text{for } j = 1, \ldots, p \tag{3.5}
\]

Thus, one standard form for an SOCP with variables $x \in \mathbb{R}^n$ is [20]

minimize \quad $f^T x$ \tag{3.6}

subject to \quad $||A_i x + b_i||_2 \leq c_i^T x + d_i, \quad i = 1, \ldots, m$

\[
g^T_j x = h_j, \quad j = 1, \ldots, p
\]

where the linear objective function is defined by vector $f \in \mathbb{R}^n$. The problem data are $A_i \in \mathbb{R}^{(n_i-1) \times n}, b_i \in \mathbb{R}^{n_i-1}, c_i \in \mathbb{R}^n$ (where $n_i$ is the dimension of the $i$th cone), $d_i \in \mathbb{R}$, for
\[ i = 1, \ldots, n, \text{ and } g_j \in \mathbb{R}^n, \quad h_j \in \mathbb{R}, \quad \text{for } j = 1, \ldots, p. \]

Note that all linear equality constraints can be written as
\[ g_j^T x = h_j \iff G x = h \quad (3.7) \]

where the \( j \)th row of \( G \) correspond to the vector \( g_j^T \), in the SOCP formulation, Eq. 3.6, and the vector \( h \) is composed of the elements \( h_j \). For \( G \in \mathbb{R}^{p \times n} \) and \( h \in \mathbb{R}^p \), it is assumed that \( G \) is full row-rank, or \( \text{rank}(G) = p \), so that \( h \in \mathcal{R}(G) \).

The set of points that satisfy the second-order cone constraint is the inverse image of the unit second-order cone under the affine mapping \[122]\n
\[ ||A_i x + b_i||_2 \leq c_i^T x + d_i \iff \begin{bmatrix} c_i^T \\ A_i \\ b_i \end{bmatrix} x \in \mathcal{Q}_n \quad (3.8) \]

The unit second-order convex cone of dimension \( n_i \) is defined as \[123]\n
\[ \mathcal{Q}_{n_i} = \left\{ \begin{bmatrix} y_0_i \\ y_i \end{bmatrix} \mid y_0_i \in \mathbb{R}, \quad y_i \in \mathbb{R}^{n_i-1}, \quad y_0_i \geq ||y_i||_2 \right\} \quad (3.9) \]

From this, define \( y_0_i \) and \( y_i \) as
\[ y_0_i = c_i^T x + d_i, \quad y_i = A_i x + b_i, \quad i = 1, \ldots, m \quad (3.10) \]

Then, the second-order cone constraints shown in Eq. 3.8 take the form
\[ ||y_i||_2 \leq y_0_i \iff \begin{bmatrix} y_0_i \\ y_i \end{bmatrix} \in \mathcal{Q}_{n_i} \quad (3.11) \]

Next, define the \( i \)th second-order cone vector \( \tilde{y}_i \in \mathbb{R}^{n_i} \) as
\[ \tilde{y}_i = \begin{bmatrix} y_0_i \\ y_i \end{bmatrix} = \begin{bmatrix} c_i^T x + d_i \\ A_i x + b_i \end{bmatrix}, \quad i = 1, \ldots, m \quad (3.12) \]
The problem represented in Eq. 3.6 can now be rewritten with variables \( x \in \mathbb{R}^n \) and \( \bar{y}_i \in \mathbb{R}^{n_i} \), for \( i = 1, \ldots, m \).

\[
\begin{align*}
\text{minimize} \quad & f^T x \\
\text{subject to} \quad & \bar{y}_i = A_i x + e_i, \quad i = 1, \ldots, m \\
& Gx = h \\
& \bar{y}_i \succeq_{C_i} 0, \quad i = 1, \ldots, m
\end{align*}
\]

where

\[
A_i = \begin{bmatrix} c_i^T \\ A_i \end{bmatrix}, \quad \text{and} \quad e_i = \begin{bmatrix} d_i \\ b_i \end{bmatrix}
\]

and the generalized inequalities \( y_i \succeq_{C_i} 0 \) are with respect to a self-dual convex cone \( C \), where \( C \) is the Cartesian product \([124]\)

\[
C = \mathcal{C}_1 \times \mathcal{C}_2 \times \ldots \times \mathcal{C}_m
\]

Each cone, \( \mathcal{C}_k \) for \( k = 1, \ldots, m \), may be a nonnegative orthant \((\mathcal{C}_i = \mathbb{R}^{n_i}_+)\), second-order cone \((\mathcal{C}_i = \mathbb{Q}_{n_i})\), or positive semidefinite cone \((\mathcal{C}_i = \mathbb{S}_{n_i})\). For the purposes of deriving the second-order cone program, the inequalities are restricted to represent a second-order cone. Thus, the second-order cone is

\[
\bar{y}_i \succeq_{\mathcal{Q}_{n_i}} 0 \iff \|y_i\|_2 \leq y_{0_i}, \quad i = 1, \ldots, m
\]

All of the equality constraints between \( \bar{y}_i \) for \( i = 1, \ldots, m \) and \( x \) can be written as

\[
\bar{y} = \bar{A} x + \bar{e}
\]
where $\bar{y}^T = [\bar{y}_1^T, \ldots, \bar{y}_m^T]$ and

$$
\bar{A} = \begin{bmatrix}
c_1^T \\
A_1 \\
\vdots \\
c_m^T \\
A_m
\end{bmatrix}, \quad \bar{e} = \begin{bmatrix}
d_1 \\
b_1 \\
\vdots \\
d_m \\
b_m
\end{bmatrix}
$$

The SOCP primal problem takes the final form with variables $x \in \mathbb{R}^n$.

$$
\begin{align*}
\text{minimize} \quad & f^T x \\
\text{subject to} \quad & \bar{y} = \bar{A}x + \bar{e} \\
& Gx = h \\
& \bar{y} \succeq C 0
\end{align*}
$$

where $\bar{y} \in \mathbb{R}^M$ are a set of slack variables, for $M = \sum_{i=1}^m n_i$. The problem data are $f \in \mathbb{R}^n$, $\bar{A} \in \mathbb{R}^{M \times n}$, $\bar{e} \in \mathbb{R}^M$, $G \in \mathbb{R}^{p \times n}$, $h \in \mathbb{R}^p$. The set for all conic slack variables, $\bar{y}$, is defined by the space $C$.

Taking into account the equality constraints $\bar{y} = \bar{A}x + \bar{e}$ (which is an affine function of $x$) and the requirement that $\bar{y}$ must satisfy $\bar{y} \succeq C 0$, these two constraints can be combined and written simply as $\bar{y} \in C$ (where $\bar{y}$, a subset of all primal variables, are all slack variables associated with the conic constraints) signifying that $\bar{y}$ must satisfy all second-order conic constraints, defined by the set $C$.

The standard form for the SOCP primal problem can therefore be written as

$$
\begin{align*}
\text{minimize} \quad & f_p^T x_p \\
\text{subject to} \quad & \bar{y} \in C \\
& Fx_p = k
\end{align*}
$$
where $f_p^T = [f^T, 0_{M \times 1}^T]$, and all primal variables are

$$x_p = \begin{bmatrix} x \\ y \end{bmatrix}$$

and

$$F = \begin{bmatrix} G & 0 \\ \bar{A} & -I \end{bmatrix}, \quad k = \begin{bmatrix} h \\ -\bar{e} \end{bmatrix}$$

### 3.2.1 Smoothness and Convexity of SOCP Constraint Functions

For the second-order conic constraints represented in Eq. 3.6, the convex inequality functions, $f_i(x) \leq 0$, can be written as

$$f_i(x) = ||A_i x + b_i||_2 - c_i^T x - d_i$$

This function has the same smoothness and convexity characteristics as $\epsilon_i(y_0, y_i) = ||y_i||_2 - y_0$.

In terms of $y_0$ and $y_i$, the gradient and Hessian for this function are

$$\nabla \epsilon_i = \begin{bmatrix} -1 \\ \frac{y_i}{||y_i||} \end{bmatrix}, \quad \nabla^2 \epsilon_i = \frac{1}{||y_i||^3} \begin{bmatrix} 0 & 0 \\ 0 & y_i^T y_i I - y_i y_i^T \end{bmatrix}$$

First, notice that the Hessian is always positive semidefinite, and therefore the functions $\epsilon_i$ are convex. However, the place of nonsmoothness for function $\epsilon_i$ exists when $y_i = 0$. At this point, the gradient is infinite [125]. This problem must be remedied before the gradient and Hessian can be used in the interior point method.

**Smoothing by Squaring**

Due to the nonsmooth nature of the function $\epsilon_i$ presented, this function must be replaced
by equivalent inequalities [122]. Moving $y_{0_i}$ to the other side of the $\epsilon_i$ function, then squaring both sides, the corresponding squared function for the second-order cone, $f_i$, is redefined as

$$f_i(y_{0_i}, y_i) = ||y_i||^2_2 - y_{0_i}^2 = y_i^T y_i - y_{0_i}^2$$

(3.26)

which is now smooth, as can be seen in the following gradient

$$\nabla f_i = 2 \begin{bmatrix} -y_{0_i} \\ y_i \end{bmatrix}, \quad \nabla^2 f_i = 2 \begin{bmatrix} -1 & 0 \\ 0 & I \end{bmatrix}$$

(3.27)

The final form for the inequality constraint functions can now be written as

$$f_i(\bar{y}_i) = \bar{y}_i^T J \bar{y}_i$$

(3.28)

where $J$ is (note $J = J^T$)

$$J = \begin{bmatrix} -1 & 0 \\ 0 & I_{n_i-1} \end{bmatrix}$$

(3.29)

and its gradient and Hessian are

$$\nabla f_i(\bar{y}_i) = 2J\bar{y}_i, \quad \nabla^2 f_i(\bar{y}_i) = 2J$$

(3.30)

Note that this function is not convex, as can be seen by the Hessian, but can be used to define a convex set, provided that the log-barrier function is well-defined, signifying that $\bar{y}_i$ initially satisfies

$$||y_i|| \leq y_{0_i}, \quad \text{or} \quad \bar{y}_i^T J \bar{y}_i \leq 0$$

(3.31)

This may require an infeasible start method, so that the initial point in the feasible start is such that the log-barrier function is well-defined. The last task is to use function $f_i$ in the log-barrier function.
Convexification by Logarithm

A logarithm is applied to the function $-f_i$ to convexify the constraint functions and form a logarithmic barrier function. In this formulation, recall $\tilde{y}_i = [y_0, y_i^T]^T$ and $\tilde{y} = [\tilde{y}_1^T, \ldots, \tilde{y}_m^T]^T$. Applying the logarithm to $-f_i$, given in Eq. 3.26 or 3.28, and multiplying by negative one-half (to simplify the gradient) leads to the log barrier function [115, 126]

$$\phi_i(\tilde{y}_i) = -\frac{1}{2} \log(-f_i) = -\frac{1}{2} \log(-\tilde{y}_i^T J \tilde{y}_i)$$  (3.32)

The gradient of the log barrier function is then written as $\nabla \phi_i(\tilde{y}_i)$

$$\nabla \phi_i(\tilde{y}_i) = - (\tilde{y}_i^T J \tilde{y}_i)^{-1} J \tilde{y}_i$$  (3.33)

The Hessian for the log barrier function is $\nabla^2 \phi_i(\tilde{y}_i)$

$$\nabla^2 \phi_i(\tilde{y}_i) = \frac{1}{(\tilde{y}_i^T J \tilde{y}_i)^2} [2J \tilde{y}_i \tilde{y}_i^T J - (\tilde{y}_i^T J \tilde{y}_i) J]$$  (3.34)

It can be shown that the Hessian is positive definite, therefore making functions $\phi_i$ and $\phi$ convex, where $\phi$ is simply the sum of $\phi_i$ for $i = 1, \ldots, m$, as

$$\phi(\tilde{y}) = \sum_{i=1}^{m} \phi_i(\tilde{y}_i)$$  (3.35)

3.3 Conditions of Optimality for the SOCP

The optimal conditions for an SOCP are defined next. From Lagrange’s constrained optimization theory, a linear Lagrangian is formulated, whereby the problem constraints are appended to the objective function using Lagrange multipliers. The necessary and sufficient conditions of optimality are described by applying methods of calculus of variations to the Lagrangian. These conditions are termed the KKT conditions, and are presented here.
3.3.1 Lagrangian

The Lagrangian for the SOCP in Eq. 3.6 is written as

\[
\mathcal{L}(x, \lambda, \nu) = f^T x + \sum_{i=1}^m \lambda_i \left( ||A_i x + b_i|| - c_i^T x - d_i \right) + \sum_{j=1}^p \nu_j \left( g_j^T x - h_j \right) \tag{3.36}
\]

where \(m\) is the number of second-order conic constraints, and \(p\) is the number of equality constraints. The vectors of dual variables are defined as \(\lambda = [\lambda_1 \ldots \lambda_m]^T\), which contains all Lagrange multipliers for the second-order cones, and the vector \(\nu = [\nu_1 \ldots \nu_p]^T\) contains all Lagrange multipliers for the equality constraints.

3.3.2 KKT Conditions

The necessary and sufficient KKT conditions of optimality for the SOCP in Eq. 3.6 are derived from the Lagrangian in Eq. 3.36.

\[
Gx^* = h \tag{3.37}
\]

\[
||A_i x^* + b_i|| - c_i^T x^* - d_i \leq 0, \quad i = 1, \ldots, m \tag{3.38}
\]

\[
\lambda_i^* \geq 0, \quad i = 1, \ldots, m \tag{3.39}
\]

\[
f + ([D\bar{f}(x)]^*)^T \lambda^* + G^T \nu^* = 0 \tag{3.40}
\]

\[
\lambda_i^* \left( ||A_i x^* + b_i|| - c_i^T x^* - d_i \right) = 0, \quad i = 1, \ldots, m \tag{3.41}
\]

where \([D\bar{f}(x)]^*\) is a matrix of gradients of the second-order conic functions, given as

\[
[D\bar{f}(x)] = \begin{bmatrix}
[\nabla \left( ||A_1 x^* + b_1|| - c_1^T x^* - d_1 \right)]^T \\
\vdots \\
[\nabla \left( ||A_m x^* + b_m|| - c_m^T x^* - d_m \right)]^T
\end{bmatrix} \tag{3.42}
\]

evaluated at \(x^*\). Eq. 3.37 are the linear equality constraints, Eq. 3.38 are the second-order cones, Eq. 3.39 are the constraints for all second-order cone Lagrange multipliers to be nonnegative, Eq. 3.40 is the requirement that the gradient must vanish at \(x^*\), and Eq. 3.41 is the complementary slackness condition.
3.4 Dual Problem for the Second-order Cone Program

The dual problem for the SOCP primal problem in Eq. 3.6, may be conveniently written in terms of the primal data \( f, A_i, b_i, c_i, d_i, g_j, h_j \), and dual variables \( w_i \in \mathbb{R}^{n_i-1} \), for \( i = 1, \ldots, m \), \( \lambda \in \mathbb{R}^m_{\geq 0} \), and \( \nu \in \mathbb{R}^p \). One form of the dual problem shown below is taken from *Convex Optimization* [20] by Boyd.

\[
\begin{align*}
\text{maximize} & \quad - \sum_{i=1}^{m} (b_i^T w_i + \lambda_i d_i) - \sum_{j=1}^{p} \nu_j h_j \\
\text{subject to} & \quad ||w_i||_2 \leq \lambda_i, \quad i = 1, \ldots, m \\
& \quad \sum_{i=1}^{m} (A_i^T w_i + \lambda_i c_i) + \sum_{j=1}^{p} \nu_j g_j = f
\end{align*}
\]  

(3.43)

where the vector \( w^T = [w_1^T, \ldots, w_m^T] \) are the associated Lagrange multipliers for the affine functions \( y_i = A_i x + b_i \) for \( i = 1, \ldots, m \), and vector \( \lambda^T = [\lambda_1, \ldots, \lambda_m] \) are the Lagrange multipliers for the second-order cone constraints \( ||y_i|| \leq y_0 \), and \( \nu = [\nu_1, \ldots, \nu_p]^T \) are the Lagrange multipliers for the equality constraints \( Gx = h \), in the associated Lagrangian shown in Eq. 3.44.

\[
L(x, y, \lambda, w, \mu, \nu) = f^T x + \sum_{i=1}^{m} \lambda_i (||y_i|| - y_0) + \sum_{i=1}^{m} w_i^T (y_i - A_i x - b_i) \\
+ \sum_{i=1}^{m} \mu_i(y_0 - c_i^T x - d_i) + \sum_{j=1}^{p} \nu_j (g_j^T x - h_j)
\]

(3.44)

The dual problem in Eq. 3.43 is formulated by determining the conditions which the dual variables must satisfy, at the infimum of the Lagrangian in Eq. 3.44 over all primal variables \( x \). The vector \( \mu^T = [\mu_1, \ldots, \mu_m]^T = \lambda^T \), when the dual problem is bounded below, so that

\[-\sum_{i=1}^{m} (b_i^T w_i + \lambda_i d_i) \neq -\infty.\]

**Standard Form for the SOCP Dual Problem**

As can be seen, the dual of the SOCP is itself an SOCP, therefore, a similar method for generalizing the dual problem can be carried out, as was done for the primal. The SOCP dual problem in standard form [126] is written in terms of all dual variables \( x_d \) as
maximize \(-k^T x_d\) \hspace{1cm} (3.45)

subject to \(z_d \in C\)
\[Hx_d = f\]

where the dual variables are defined by \(\nu\) and \(z_d\) as

\(x_d = \begin{bmatrix} \nu \\ z_d \end{bmatrix}\) \hspace{1cm} (3.46)

\(z_d = \begin{bmatrix} z_1 \\ \vdots \\ z_m \end{bmatrix}, \text{ where } z_i = \begin{bmatrix} \lambda_i \\ w_i \end{bmatrix}, \text{ for } i = 1, \ldots, m\) \hspace{1cm} (3.47)

and where
\[H = \begin{bmatrix} G^T \\ \bar{A}^T \end{bmatrix}\] \hspace{1cm} (3.48)

3.5 Solving the General Convex Problem

The solution method for convex optimization problems is focused on satisfying all of the nonlinear KKT conditions at optimality. However, the means for introducing the inequality constraints into the problem are different, depending on different methods. The three main forms for solving convex optimization problems are the Barrier Method, Primal-dual Interior Points (PDIP) Method, and Extended Self-dual (ESD) embedding [20, 121, 126, 127]. In these methods, the inequality constraints \((f_i \text{ in Eq. 3.1})\) are handled by formulating an equality constrained method, where the inequalities are appended to the objective function using an indicator function [119]. This allows Newton methods to be applied in solving the KKT equations, and driving the KKT residuals to zero. The KKT conditions of optimality for the problem in Eq. 3.1 are written as
\[ Gx^* = h \]  

(3.49)

\[ f_i(x^*) \leq 0, \quad i = 1, \ldots, m \]  

(3.50)

\[ \lambda_i^* \geq 0, \quad i = 1, \ldots, m \]  

(3.51)

\[ f + ([D\bar{f}(x)]^*)^T\lambda^* + G^T\nu^* = 0 \]  

(3.52)

\[ \lambda_i^* f_i(x^*) = 0, \quad i = 1, \ldots, m \]  

(3.53)

In the equality constrained formulation, the optimization problem in Eq. 3.1 becomes

\[
\begin{align*}
\text{minimize} & \quad f_0(x) + \sum_{i=1}^{m} I_\_f_i(x) \\
\text{subject to} & \quad Gx = h 
\end{align*}
\]

(3.54)

where the indicator function is

\[
I_\_u = \begin{cases} 
0, & u \leq 0 \\
\infty, & u > 0 
\end{cases}
\]

(3.55)

and is not continuously differentiable. Therefore, an approximation of the indicator function is made, depending on the solution method. These are described in more detail in the following sections.

3.5.1 Barrier Method

The solution to the convex optimization problem in Eq. 3.1 is approached by implementing the barrier method on all inequality constraints. These constraints are appended to the objective function, using an approximation of the indicator function for each constraint in the form of the log-barrier function [20]. The log-barrier function is shown in Eqs. 3.32 and 3.35. This results in implementing the problem in Eq. 3.54 as
minimize \( f_0(x) + \frac{2}{a} \sum_{i=1}^{m} \phi_i(x) \) \hspace{1cm} (3.56)

subject to \( Gx = h \)

Thus, the indicator function in Eq. 3.55 is approximated as

\[
\hat{I}_{f_i}(x) = -\frac{1}{a} \log(-f_i(x)) = \frac{2}{a} \phi_i(x)
\] \hspace{1cm} (3.57)

where \( a > 0 \) is a parameter that sets the approximation of the indicator function, and as \( a \to \infty \), the approximation of the indicator function becomes more accurate. Implementing the log-barrier in this manner leads to a central path problem, where for the current value of \( a \), a set of central points \( x^*(a) \) are defined for the current central path. As \( a \) is increased with each ‘centering step,’ the inequality constraints are better approximated, and the central path is improved. The objective function for Eq. 3.56 is equivalently written as

\[
J = a f_0(x) + 2 \sum_{i=1}^{m} \phi_i(x)
\] \hspace{1cm} (3.58)

Now, the Lagrangian for this problem, with \( \phi_i \) as defined previously, is

\[
\mathcal{L}(x, \nu) = a f_0(x) + 2 \sum_{i=1}^{m} \phi_i(x) + \sum_{i,j} \nu_j (g_j^T x - h_j)
\] \hspace{1cm} (3.59)

The modified KKT that result from this Lagrangian are

\[
a \nabla f_0(x^*) + 2 \sum_{i=1}^{m} \nabla \phi_i(x^*) + G^T \nu^* = 0
\] \hspace{1cm} (3.60)

\[
Gx^* = h
\] \hspace{1cm} (3.61)

Therefore, in order to determine an optimal solution to the problem, we must satisfy the \( n + p \) nonlinear KKT equations shown above, where the second-order cone constraints
and associated dual variable, $\lambda$, have been eliminated [20]. The objective function consists of the original objective function, and the contributions of the log-barrier functions for each inequality constraint. Therefore, the only constraints directly implemented in the log-barrier method include the linear equality constraints. The barrier method involves solving the linearly constrained minimization problem, using the last point found as the starting point for the next linearly constrained minimization problem. On each centering step, $a$ is increased until eventually the primal solution satisfies all second-order cone constraints, and the contributions of the log-barrier functions to the objective function are very small.

### 3.5.2 Primal-dual Interior-point Method

The solution to convex optimization problems using PDIP methods is approached by implementing the modified KKT equations that result from the barrier method [119,128]. Starting from the modified objective function in Eq. 3.56, the Lagrangian is

$$
\mathcal{L}(x, \nu) = f_0(x) + \frac{2}{a} \sum_{i=1}^{m} \phi_i(x) + \sum_{i=1}^{p} \nu_i (g_i^T x - h_i)
$$

(3.62)

The modified KKT that result from this Lagrangian are

$$
\nabla f_0(x^*) + \frac{2}{a} \sum_{i=1}^{m} \nabla \phi_i(x^*) + G^T \nu^* = 0
$$

(3.63)

$$
Gx^* = h
$$

(3.64)

By definition of the log-barrier functions, $\phi_i(x)$, the partial derivatives with respect to $x$ can be written as

$$
\nabla \phi_i(x) = -\nabla \left[ \frac{1}{2} \log(-f_i(x)) \right] = -\frac{1}{f_i(x)} \left( \frac{1}{2} \nabla f_i(x) \right)
$$

(3.65)

Therefore, the first modified KKT equation, Eq. 3.63, can be written in terms of the inequality functions, $f_i(x)$, as

$$
\nabla f_0(x^*) - \frac{1}{a f_i(x^*)} \sum_{i=1}^{m} \nabla f_i(x^*) + G^T \nu^* = 0
$$

(3.66)
Notice that the coefficient on the second term is equivalent to the dual variable $\lambda_i$ in the original KKT conditions. Setting this equal to $\lambda_i$ leads to the final modified KKT equation for the PDIP method.

$$-\lambda_i^* f_i(x^*) = \frac{1}{a}, \quad \text{for} \quad i = 1, \ldots, m$$

(3.67)

Where now, as opposed to the barrier method, the complementary slackness condition in the second modified KKT equation remains in the problem variable space, and updates to all primal and dual variables are made using the Newton method. The $n + m + p$ nonlinear modified KKT equations define the necessary and sufficient conditions of optimality for the PDIP method. For the general convex problem, these are written as

$$\nabla f_0(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla f_i(x^*) + G^T \nu^* = 0$$

(3.68)

$$-\lambda_i^* f_i(x^*) = \frac{1}{a}, \quad \text{for} \quad i = 1, \ldots, m$$

(3.69)

$$Gx^* = h$$

(3.70)

### 3.5.3 Extended Self-dual Embedding Method

The self-dual embedding technique solves both the primal and dual general SOCPs, shown in Eq. 3.20 and 3.45, simultaneously in a self-dual cone LP [126, 128]. Combining these two problems allows for the dual problem to be embedded within the primal problem in the Newton system, and improvements to both may be made simultaneously. Additionally, an initial feasible point to the extended self-dual problem can always be determined. Starting from the CVXOPT algorithm documentation by Vandenberghe [115, 126], the extended self-dual problem results from the homogeneous self-dual embedding reformulation used to solve the primal and dual problems in one dual cone LP. The extended self-dual problem is written in terms of previously defined matrices in Eq. 3.20 and 3.45 as the linear complementary problem (LCP)
minimize \((m + 1)\theta\) \hspace{1cm} (3.71)

subject to \(w = Hv + n\)

\[
(\bar{y}, k, z_d, \tau) \succeq C \theta
\]

where

\[
w = \begin{bmatrix} 0 \\ \bar{y} \\ k \\ 0 \end{bmatrix}, \quad v = \begin{bmatrix} x \\ \nu \\ \tau \\ \theta \end{bmatrix}, \quad n = \begin{bmatrix} 0 \\ 0 \\ 0 \\ m + 1 \end{bmatrix}
\] \hspace{1cm} (3.72)

and

\[
H = \begin{bmatrix} 0 & G^T & \bar{A}^T & f & q_x \\ -G & 0 & 0 & h & q_\nu \\ -\bar{A}^T & 0 & 0 & \bar{e} & q_{z_d} \\ -f^T & -h^T & -\bar{e}^T & 0 & q_\tau \\ -q_x^T & -q_\nu^T & -q_{z_d}^T & -q_\tau & 0 \end{bmatrix}
\] \hspace{1cm} (3.73)

The variable \(\theta\) is used to minimize the nonhomogeneity of the problem in Eq. 3.71, and is defined as

\[
\theta = \frac{-z_d^T \bar{y} + k\tau}{m + 1}
\] \hspace{1cm} (3.74)

The \(q\) vectors are

\[
\begin{align*}
q_x &= m + 1 \\
q_\nu &= \frac{m + 1}{\bar{y}_0 z_d + 1} \\
q_{z_d} &= \frac{m + 1}{\bar{y}_0 z_d + 1} \\
q_\tau &= \frac{m + 1}{\bar{y}_0 z_d + 1}
\end{align*}
\] \hspace{1cm} (3.75)

where \(x_0, \bar{y}, \nu_0, z_d\) may be chosen arbitrarily, provided that \((\bar{y}_0, z_{d_0}) \succeq 0\).
Formulating this LCP allows for strict initial feasibility, and also, since it is self-dual, the optimality conditions are exactly the problem constraints, with the addition that

\[-z_d^T \bar{y}^* + k^* \tau^* = 0\]  \hspace{1cm} (3.76)

so that \(\theta^* = 0\). Solving the Newton system associated with this ESD embedded problem allows for improvements to be made for both the primal and dual problems simultaneously.

### 3.6 Newton’s Method

Newton’s method, when applied to the conditions of optimality for the convex problems described previously, drives the solution towards optimality in an iterative manner. Newton’s method is to approximate the nonlinear optimality function (defined here as \(f(x_i)\)) by a local quadratic model, \(q(x_i, \Delta x_i)\), i.e. a second-order Taylor series expansion, as

\[f(x_i + \Delta x_i) \approx f(x_i) + \nabla f(x_i)^T \Delta x_i + \frac{1}{2} \Delta x_i^T \nabla^2 f(x_i) \Delta x_i \triangleq q(x_i, \Delta x_i) \]  \hspace{1cm} (3.77)

where the search direction, \(\Delta x_i\), is chosen to minimize this quadratic model. Therefore, under optimality \(\nabla_{\Delta x_i} q(x_i, \Delta x_i) = 0\), the Newton step direction is solved for as

\[\Delta x_i = -[\nabla^2 f(x_i)]^{-1} \nabla f(x_i) \]  \hspace{1cm} (3.78)

Due to strong convexity, the Hessian is positive definite, and the inverse of the Hessian \([\nabla^2 f(x_i)]^{-1}\) exists.

In addition to determining a search direction, the step size associated with the search direction must also be determined. In solving convex optimization problems, it is ensured foremost that the step remains within the feasible set. There are many ways to implement such a procedure, and the most common to convex optimization is the backtracking line search. Secondly, it is preferred that step minimize the functions of interest, and in the case of convex problems these functions are the optimality conditions. More details on these specific techniques used are included in the custom algorithm in Chapter 10.
3.7 Sequential Convex Optimization

The method of sequential convex optimization, or sequential convex programming (SCP) is one approach to apply convex optimization techniques to solve inherently nonconvex (or nonlinear) problems [20,129]. This method effectively convexifies each nonconvex problem constraint, via affine or convex approximations, and iteratively solves an approximate convex optimization problem until a convergence criterion is met [130]. Upon termination of the SCP, the goal is to satisfy all nonconvex constraints, therefore adequately solving the original nonconvex problem. Depending on the complexity of the nonconvex problem, SCP methods can prove to efficiently solve nonconvex problems, and in many cases only a few iterations are required.

Let each iteration of SCP be denoted by $k$, such that $x^{(k)}$ is the optimal solution to the convex optimization problem on iteration $k$. Consider the nonconvex problem with parameters $x^{(k)} \in \mathbb{R}^n$

$$\begin{align*}
\text{minimize} & \quad f_0(x^{(k)}) \\
\text{subject to} & \quad f_i(x^{(k)}) \leq 0, \quad i = 1, \ldots, m \\
& \quad g_j(x^{(k)}) = 0, \quad j = 1, \ldots, p
\end{align*}$$

(3.79)

where here $f_0(x^{(k)})$, $f_i(x^{(k)})$, and $g_j(x^{(k)})$ may be nonconvex functions.

The basic approach is to formulate convex approximations $\hat{f}_0$ and $\hat{f}_i$, of $f_0$ and $f_i$, and an affine approximation $\hat{g}_j$ of $g_j$, over a trust region $\mathcal{T}^{(k)}$ on each sequential iteration, $k$ [20]. The trust region, $\mathcal{T}^{(k)}$, is strictly defined by these convex approximations, and is updated on each iteration, $k$. The problem is re-solved over the new trust region so that $x^{k+1}$ is the optimal solution to the new problem. This process is repeated until the convergence criteria used for exiting the sequential convex optimization problem is met (for example, once the trust region $\mathcal{T}^{(k)}$ only changes by some small degree, compared to the previous iteration). The approximate convex optimization problem is written as
minimize $\hat{f}_0(x^{(k+1)})$ \hfill (3.80)
subject to $\hat{f}_i(x^{(k+1)}) \leq 0, \quad i = 1, \ldots, m$
$\hat{g}_j(x^{(k+1)}) = 0, \quad j = 1, \ldots, p$
$x^{(k+1)} \in \mathcal{T}^{(k)}$

The convex approximation of the objective function is a first-order Taylor series expansion of the nonconvex function $f_0$ as

$$\hat{f}_0(x^{(k+1)}) = f_0(x^{(k)}) + \nabla f_0(x^{(k)})^T(x^{(k+1)} - x^{(k)})$$ \hfill (3.81)

The convex or affine approximations of the nonconvex constraints take the form of a first-order Taylor series expansion of the nonconvex constraint functions $f_i$ and $g_j$ as

$$\hat{f}_i(x^{(k+1)}) = f_i(x^{(k)}) + \nabla f_i(x^{(k)})^T(x^{(k+1)} - x^{(k)}) \leq 0, \quad i = 1, \ldots, m$$ \hfill (3.82)
$$\hat{g}_j(x^{(k+1)}) = g_j(x^{(k)}) + \nabla g_j(x^{(k)})^T(x^{(k+1)} - x^{(k)}) = 0, \quad j = 1, \ldots, p$$ \hfill (3.83)

In some cases, a second-order expansion of these functions can be used to approximate the nonconvex constraint, provided that the nonconvex functions are locally convex (Hessian is positive definite at $x_i$).

The implementation of SCP methods to nonlinear and/or nonconvex problems can be potentially hazardous. The SCP may lead to local minima, or may have very poor convergence properties [21, 130]. Therefore it is important to fully understand the problem to which it is applied, including the degree of nonlinearity and nonconvexity in the problem that is approximated.
CHAPTER 4
THESIS STATEMENT AND RESEARCH OVERVIEW

4.1 Thesis Statement

Formulating the RPO problem using convex optimization techniques, due to the flexibility of convex constraints, and their naturally deterministic and guaranteed convergence properties, has the capability to plan optimal RPO trajectories and efficiently implement a variety of safety of flight constraints, in real-time environments.

4.2 Research Overview

The feasibility and interest for this research manifests itself from several aforementioned developments. Namely, the HCW model of relative orbital motion, safety of flight concepts, and the technology, that is, convex optimization. The linear time-invariant HCW model takes different forms, depending on the coordinate system. Both the Cartesian and spherical formulations are tested in this research.

In addition, a new rotating body-fixed chief coordinate system is considered. This leads to the derivation of a new set of linearized relative motion differential equations coordinatized in the rotating chief’s body-fixed reference frame, which are linear time-varying [131, 132]. Ultimately, the purpose for each of these frames is to describe the orbital relative motion as viewed from an IVLH frame in Cartesian or spherical coordinates, and from a frame fixed to a controlled or uncontrolled spinning spacecraft. In one rotating chief body-fixed frame formulation, it is to be assumed that the chief spacecraft is rotating at constant rate, uncontrolled, about one of its principal axes. This may be a result of damping from gravity gradient torques, drag, or other perturbations. The other rotating chief body-fixed frame formulation allows the angular rotation vector to vary with time, possibly as a result of a tumbling spacecraft. The equations for each formulation are applied to the RPO problem
using convex optimization to generate optimal delta-v trajectories.

Safety of flight considerations to be implemented fall into two categories: active and passive safe trajectories. Active safe trajectories typically have safety constraints along the duration of the trajectory, but do not guarantee unpowered passive safety of flight. Should active control be lost mid-course, there is typically no guarantee of a passive abort scenario. Active safety constraints include the approach corridor, spherical keep-out zone (KOZ), and plane or half-space constraints. Approach corridors and half-spaces inherently define a convex solution space on the relative position of the deputy, and therefore conform to the second-order conic constraint. The spherical KOZ enforces the deputy’s relative position to lie outside a desired sphere at all times, ensuring a minimum safe distance, and is inherently nonconvex. Three main flight phases are examined for this formulation, and they include the Initial Approach, Way Point Following/Inspection, and Final Approach.

These active safety ideas are countered by the formulation of passive safe trajectories, generally formulated as a PASE. The goal is to determine optimal passive abort safety ellipse (PASE) trajectories using the ROEs to formulate relative ellipses [92]. The constraints for safety of flight are initially formulated using relative orbital elements (ROEs), but the differential equations for the ROEs as a function of control input are highly nonlinear. Therefore, a new set of Linear Relative Orbital Elements (LROEs) are developed, so that the dynamics of relative ellipses can be conveniently described.

The ROEs are first written in terms of the original HCW state, while noting linear relationships. This leads to a linear transformation, that describes the ROEs in terms of the HCW states. The new states are termed LROE states, to signify that they are linear functions of HCW states, and also functions of the ROEs. Several illustrations of common relative motion trajectories are used to gain a better understanding of these ROEs with regards to typical RPO scenarios. Constraints can be added to this formulation so that relative ellipses are guaranteed to be a PASE with zero probability of collision, however, these constraints are in general nonconvex and also require SCP methods. The LROEs are traditionally formulated in an LVLH frame in both Cartesian and spherical coordinates.
Several transfer scenarios are examined, including a Safe Initial Approach (co-elliptic flyby to PASE), Safe Traveling Ellipse (multi-rev. PASE transfers in the in-track direction), and Safe Final Approach (changes in PASE size, including semi-major axis and/or cross-track amplitude). In these cases, optimal delta-v trajectories for transfers between PASEs are calculated using the sequential convex optimization algorithm.

Important trajectory constraints are also implemented. These include constraints on maximum allowable control acceleration, approach corridors, spherical KOZ, and plane (half-space) constraints. The approach cone constraint is convex and is directly implemented in the convex optimization problem. The spherical KOZ, however, requires an iterative approach using SCP. This is approached by implementing a half-space constraint, defined as a plane. The plane constraint, being a cone with a full angle, is a convex constraint that is used to approximate the sphere. Using SCP in conjunction with plane placement results in an optimal solution to the KOZ problem [22].

For the SOF formulation, the specification of all terminal ROEs for the transfer between relative ellipses is a fully convex problem. However, further relaxing the problem is shown to provide a more optimal solution. This shows the significance of relaxing the terminal phase angle for the final desired PASE. Relaxing this constraint makes the problem nonconvex, however, this can be approached in a manner similar to the spherical KOZ, by using SCP methods. Further considerations involve constraints to ensure mid-course passive SOF. This ensures that the deputy has guaranteed collision avoidance along the entire duration of the trajectory. Ensuring the semi-major axis, and cross-track amplitude are safe, and that the phase difference for the relative ellipse is within a desired range, is sufficient to provide passive safety. These constraints are inherently nonconvex, but are used to ensure passive safety of flight via and iterative approach, also requiring SCP.

The above convex optimization problems are initially solved using a general MatLab based CVX solver [111]. However, in order to determine the applicability of convex optimization to real-time environments, a custom algorithm tailored to a specific RPO problem is developed. The focus is on the development of an algorithm tailored specifically to RPO
scenarios in the LVLH frame. Second-order conic SOF constraints such as the maximum allowable acceleration and approach corridor are included. In the initial problem formulations, all states and control variables are included in the problem. However, for the custom solver, dynamics constraints in terms of only the control variables are developed. This greatly reduces the size of the linear system of equations, at the expense of creating more complex second-order conic constraint equations.

Several other solvers are tested for CPU and memory requirements, namely the MOSEK and Gurobi solvers [133, 134]. Achieving this allows for testing of the efficiency and practicality of the convex optimization techniques in a real-time environment. These metrics are measured by determining the CPU and memory requirements for each solver, and on a variety of platforms, as well as the approximate number of floating point operations per second (FLOPS). Convergence properties and residuals on the optimal value are also recorded for a variety of scenarios.

The solutions from the convex optimization solver are tested in a full nonlinear simulation. An LQR control law is used for the trajectory following method, and trajectory errors and delta-v differences between the RPO planner and nonlinear simulation are presented. Simulation results include an analysis of the effects of eccentricity, J2, and drag in the LEO environment. The nonlinear simulation analysis is separated into three chapters. The first two chapters present the nonlinear simulation results from nominal trajectories, and the final chapter includes an analysis of the effects of trajectory dispersions on the planner and nonlinear simulation.

4.2.1 Dissertation Outline

The relative orbital motion model development begins with the well-defined HCW formulation, presented in Chapter 5. This includes the Cartesian and spherical forms for these equations. The ROEs and LROEs, which are based upon the HCW equations, are also presented, and important characteristics of each are noted.

The second relative orbital motion model, derived in Chapter 6, is with respect to a rotating chief body-fixed frame, and is termed the generalized HCW model. This model is
derived from the nonlinear two-body equations of relative motion with respect to a rotating frame, as described in Greenwood [131]. These equations are linearized and coordinatized in the chief’s body-fixed frame to produce the desired LTV differential equations of relative orbital motion with respect to an arbitrary rotating frame.

In order for these models to be implemented in a convex program, the objective function and dynamics must be discretized. Dynamics are discretized so that linear relationships between state variables can be established. All constraint equations are similarly discretized, and the overall problem formulations are presented in Chapters 7 and 8, including several planner results.

The third and final relative orbital motion model considered is primarily used for SOF development, presented in Chapter 9. The LROE states, along with the associated new sate space are examined and implemented in RPO trajectory planning to ensure the deputy flies in a PASE. From this, several safety of flight constraints are created, which place strict constraints on the allowable size and location of the relative ellipse.

Once the convex trajectory planner problems have been developed, and several planner results are shown, the customized solver techniques are introduced in Chapter 10. This involves showing how the RPO convex optimization problems reduce from the general forms, and different methods to enforce the problem dynamics.

Chapter 11 covers the CPU and memory analysis. Tables are included to show the number of FLOPS, due to the effects of problem formulation, problem size, and number of equality and second-order cone constraints. Three CPUs are used to compare and contrast the required CPU for specific problems. Memory requirements are measured for a variety of problems as well, and are included in tables.

The nonlinear simulation model is presented in Chapter 12. This includes the differential equations for both the Inertial/Inertial and Inertial/Relative simulation. The reference trajectories and optimal LQR control laws for the trajectory following method are shown, followed by the implementation of the spherical HCW model in the nonlinear simulation.

All nonlinear simulation results are included in Chapters 13, 14, and 15. This material
is presented by comparing the trajectories and delta-v results for the convex planner versus simulation, in a variety of scenarios. The goal for Chapter 13 and 14 results is to present the effects of eccentricity, J2 and drag, and the nonlinearities in the dynamics on the optimal planned delta-v. In Chapter 15, trajectory dispersions are included in the nonlinear simulation analysis. This dispersion analysis is conducted primarily to test the RPO planner, and to visualize the effects of dispersions on the optimal trajectory and delta-v. Final remarks and conclusions are in Chapter 16.
CHAPTER 5
RELATIVE ORBITAL MOTION, THE HCW EQUATIONS

The dynamics in this document are primarily governed by the linear time-invariant Hill, Clohessy-Wiltshire equations which describe the relative motion of a deputy satellite in close proximity of a chief's circular reference orbit. This relative motion model is commonly used in the initial RPO maneuver planning phase of a mission. These dynamics apply very accurately to many areas of vehicle GN&C, provided the reference orbit is circular or near-circular (with eccentricity < 0.01). Different coordinate frames are considered, including the Cartesian LVLH frame and spherical coordinates. Spherical frame coordinates are implemented to gain more accuracy in the along-track and cross-track directions, due to the natural curvature of the true reference orbit. Illustrations of each frame will be provided in Section 5.1.

The safety of flight concepts in chapter 9 involve the analysis of a set of ROEs [92]. Some important properties of these are summarized. From these, a new form for the HCW equations is provided, defined as a set of linear relative orbital element (LROE) states. The LROE formulation provides states that more conveniently describe the instantaneous position, size, and time-evolution of relative ellipses. The LROEs are used to develop passive safety of flight techniques in Chapter 9, by placing constraints on the allowable size and position of a relative ellipse so that a passive abort safety ellipse is guaranteed at all times. Additionally, a normalized-time form of the LROE dynamics is defined, where the time rate of change in the dynamics are with respect to the natural angular rate of motion of the chief’s orbit. These new states with normalized-time dynamics are more efficient when used in the custom solver presented in Chapter 10, since all problem variables are naturally scaled to have a similar order of magnitude.
5.1 The Hill, Clohessy-Wiltshire Equations

The differential equations of motion given by Eq. 2.1 govern the linearized motion of a deputy satellite in close proximity to a chief satellite in a near-circular orbit, and are coordinatized in the chief’s local-vertical local-horizontal (LVLH) frame. These equations are provided here in Eq. 5.1, where \( x, y, \) and \( z \) (and their derivatives) correspond to the Cartesian radial, along-track, and cross-track components, respectively.

\[
\begin{align*}
\ddot{x} - 2\omega \dot{y} - 3\omega^2 x - a_{rx} &= 0 \\
\dot{y} + 2\omega \dot{x} - a_{ry} &= 0 \\
\ddot{z} + \omega^2 \dot{z} - a_{rz} &= 0
\end{align*}
\]  

(5.1)

It can be seen that these equations take the form of two coupled in-plane second-order differential equations and one coupled out-of-plane simple harmonic oscillator. The state-space form in Cartesian LVLH coordinates \([135]\) is

\[
\dot{x}(t) = Ax(t) + Bu(t)
\]  

(5.2)

where the relative position and velocity states are given by

\[
x = \begin{bmatrix} r_{rel}(t) \\ x_{rel}(t) \end{bmatrix}, \quad \text{where} \quad r_{rel}(t) = \begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix}, \quad v_{rel}(t) = \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{z}(t) \end{bmatrix}
\]  

(5.3)

and where \( u(t) = a_T(t) \) is the thrust acceleration. The \( A \) and \( B \) matrices are

\[
A = \begin{bmatrix}
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
3\omega^2 & 0 & 0 & 0 & 2\omega & 0 \\
0 & 0 & 0 & -2\omega & 0 & 0 \\
0 & 0 & -\omega^2 & 0 & 0 & 0
\end{bmatrix}, \quad B = \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 1 & 0 \\
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]  

(5.4)

The value for the constant, \( \omega \), is the mean orbital rate of the chief’s circular reference orbit.
\[ \omega = \sqrt{\frac{\mu}{R_{\text{ref}}^3}} \]  

(5.5)

where \(R_{\text{ref}}\) is the reference radius of the chief’s circular orbit. When the HCW equations are applied to a near-circular orbit, then the approximate value for \(\omega\) may be determined via

\[ \omega \approx \sqrt{\frac{\mu}{a_c^3}} \]  

(5.6)

where \(a_c\) is the semi-major axis of the chief’s near-circular reference orbit.

### 5.1.1 The Cartesian LVLH Frame

A Cartesian spacecraft-centered LVLH frame may be defined for either the deputy or chief spacecraft, as illustrated in Fig. 5.1. This frame’s origin is the center of mass of the vehicle, and is composed of three vectors: radial, out-of-plane, and the third simply completes the triad. Each triad consists of three mutually orthogonal vectors (\(\hat{i}_x\), \(\hat{i}_y\), and \(\hat{i}_z\)), which come from the inertial position of the vehicle, and the angular momentum vector of the vehicle’s orbit. These vectors are defined for the deputy (using subscript \(d\)) as

\[ \hat{i}_{xd} = \hat{i}_{R_d} = \frac{R_d}{||R_d||} \]  

(5.7)

\[ \hat{i}_{yd} = \hat{i}_{h_d} \times \hat{i}_{R_d} = \hat{i}_{zd} \times \hat{i}_{xd} \]  

(5.8)

\[ \hat{i}_{zd} = \hat{i}_{h_d} = \frac{R_d \times V_d}{||R_d \times V_d||} \]  

(5.9)

and for the chief (using subscript \(c\)),

\[ \hat{i}_{xc} = \hat{i}_{R_c} = \frac{R_c}{||R_c||} \]  

(5.10)

\[ \hat{i}_{yc} = \hat{i}_{h_c} \times \hat{i}_{R_c} = \hat{i}_{zc} \times \hat{i}_{xc} \]  

(5.11)

\[ \hat{i}_{zc} = \hat{i}_{h_c} = \frac{R_c \times V_c}{||R_c \times V_c||} \]  

(5.12)

where \(R\) is the inertial position and \(V\) is the inertial velocity of each vehicle.
The frames shown in Fig. 5.2 represent each of these two LVLH frames developed here. Note that one primary assumption of the HCW equations is that one of the LVLH frames is rotating at a constant angular rate about the angular momentum vector $\hat{\mathbf{i}}_h$, thus leading to the time-invariant differential equations.

Fig. 5.1: The Cartesian LVLH frames, defined from the inertial position and velocity of the chief and deputy spacecraft

Fig. 5.2: The Cartesian LVLH frames, shown for both the chief and deputy spacecraft
Though both the chief and deputy LVLH frames are initially defined, it is more common to use the chief-centered LVLH frame to describe the relative motion dynamics of the deputy in RPO scenarios. The chief’s LVLH frame may be broken into two separate planes, in-plane and out-of-plane. In-plane refers to "in the plane of the orbit," i.e. the plane defined by the span of the instantaneous inertial position and velocity vectors. Out-of-plane refers to "out of the plane of the orbit," i.e. the radial, cross-track plane, where cross-track is in the direction of the angular momentum of the chief’s orbit. These two planes and their separate components of position and velocity (from Eq. 5.3) are illustrated in Fig. 5.3.

\[
\begin{align*}
\mathbf{r}_{\text{rel}} &= \mathbf{R}_d - \mathbf{R}_c \\
\mathbf{v}_{\text{rel}} &= \mathbf{V}_d - \mathbf{V}_c - \frac{\mathbf{R}_c \times \mathbf{V}_c}{||\mathbf{R}_c||^2} \times (\mathbf{R}_d - \mathbf{R}_c)
\end{align*}
\]

Which, when linearized and coordinatized in the chief’s Cartesian LVLH frame (with constant rotation rate \(\omega\)), leads to the relative motion equations given by Eq. 5.2-5.4.
5.1.2 The HCW Equations in Spherical Coordinates

Although the Cartesian HCW coordinate system approximates the relative motion of the deputy spacecraft very well in many scenarios where the deputy is in close proximity of the chief, a more accurate and considerate model for these dynamics involves coordinatizing the relative state vector using 'curvilinear,' or spherical coordinates. These coordinates are used to recapture modeling errors, which are inherent to the Cartesian frame, due to the natural spherical nature of the reference orbit in the along-track and cross-track directions.

First, spherical coordinates for both the chief and deputy are illustrated from the viewpoint of an inertial observer in Fig. 5.4. Note that in this frame, the three vector components are \( \hat{i}_\rho, \hat{i}_\theta, \) and \( \hat{i}_\phi \), defined from \( \rho, \theta, \) and \( \phi \), for both the chief and deputy.

Next, consider a quasi-inertial frame that is fixed to the chief’s reference orbital plane, as shown in Fig. 5.5. Under two-body assumptions, this frame is non-rotating and can be used to define the spherical coordinates in a more convenient way (hence, quasi-inertial).
The chief and deputy’s spherical coordinates are now shown in this frame, where \( \hat{i}_{\phi_c} \) is directly out of the page, \( \phi_c = 0 \), and \( \phi_d = \phi_d - \phi_c = \delta \phi \).

![Quasi-inertial Frame](image)

**Quasi-inertial Frame**  
(aligned with chief orbital plane)

When the relative position, \( \mathbf{R}_d - \mathbf{R}_c \), is very small, relative spherical states (\( \mathbf{x}_{Sph} \)) with respect to the chief’s spherical coordinates, may be defined. Under these certain conditions, the two approximate radial position and velocity states are \( x \approx \delta \rho \) and \( \dot{x} \approx \dot{\delta} \rho \), while the other four states are interpreted as arc-lengths and angular rotation rates [69]. These four states are divided by a time-varying reference radius, \( \rho_{ref}(t) \).

\[
\mathbf{x}_{Sph} = \begin{bmatrix}
\delta \rho \\
\delta \theta \\
\delta \phi \\
\delta \rho \\
\delta \theta \\
\delta \phi
\end{bmatrix} \approx \begin{bmatrix}
x \\
y/\rho_{ref}(t) \\
z/\rho_{ref}(t) \\
\dot{x} \\
\dot{y}/\rho_{ref}(t) \\
\dot{z}/\rho_{ref}(t)
\end{bmatrix} = \begin{bmatrix}
\rho_d - \rho_c \\
\theta_d - \theta_c \\
\phi_d - \phi_c \\
\dot{\rho}_d - \dot{\rho}_c \\
\dot{\theta}_d - \dot{\theta}_c \\
\dot{\phi}_d - \dot{\phi}_c
\end{bmatrix} \tag{5.15}
\]
The spherical coordinates for the deputy with respect to the chief are shown in Fig. 5.6, where each unit vector of the triad is shown, along with the interpreted arc-lengths, $\delta \theta$ and $\delta \phi$.

Fig. 5.6: The chief’s spherical LVLH frame, from chief-centered frame

The relative position, $r_{rel}$, of the deputy in the quasi-inertial frame in Fig. 5.5 is written as a function of the spherical states of the chief and the relative spherical coordinates using Eq. 5.13 where

$$R_d = (\rho_c + \delta \rho) \hat{i}_{\rho_d}$$

$$R_c = \rho_c \hat{i}_{\rho_c}$$ (5.17)

and the deputy’s relative velocity, $v_{rel}$, can be calculated via Eq. 5.14, where the velocities of the deputy and chief in the quasi-inertial frame, and in terms of spherical coordinates, are given by [131]

$$V_d = \left( \dot{\rho}_c + \delta \dot{\rho} \right) \hat{i}_{\rho_d} + (\rho_c + \delta \rho) \left( \dot{\theta}_c + \delta \dot{\theta} \right) \cos(\delta \phi) \hat{i}_{\theta_d} + (\rho_c + \delta \rho) \delta \phi \hat{i}_{\phi_d}$$ (5.18)

$$V_c = \dot{\rho}_c \hat{i}_{\rho_c} + \rho_c \dot{\theta}_c \hat{i}_{\theta_c}$$ (5.19)
5.2 Relative Orbital Elements

Another set of states that can be used to define the chaser’s relative orbital motion are known as the HCW relative orbital elements [92]. Similar to the original HCW state vector, there are four in-plane states and two out-of-plane states. The in-plane HCW relative orbital elements are the \( x_r, y_r \) location of the center of the relative ellipse, the semi-major axis of the \( 2 \times 1 \) ellipse, \( a_r \), and the in-plane phase angle \( E_r \). The two out-of-plane states are the cross-track amplitude, \( A_z \), and the out-of-plane phase angle \( \psi_z \). Each of these ROEs is defined as a function of the original relative position and velocity states of the chaser given in Eq. 5.3 [92].

\[
x_r = 4x + \frac{2\dot{y}}{\omega} \quad (5.20)
\]
\[
y_r = y - \frac{2\dot{x}}{\omega} \quad (5.21)
\]
\[
a_r = \sqrt{(6\omega x + 4\dot{y})^2 + \left(\frac{2\dot{x}}{\omega}\right)^2} \quad (5.22)
\]
\[
E_r = \arctan\left(\frac{2\dot{x}}{6\omega x + 4\dot{y}}\right) \quad (5.23)
\]
\[
A_z = \sqrt{z^2 + \left(\frac{\dot{z}}{\omega}\right)^2} \quad (5.24)
\]
\[
\psi_z = \arctan\left(\frac{\omega z}{\dot{z}}\right) \quad (5.25)
\]

Relative orbital elements are illustrated in Figure 5.7 for the in-plane and out-of-plane components. Notice that the position and size of the instantaneous in-plane relative ellipse can be defined by specifying \( x_r, y_r, \) and \( a_r \), while the in-plane phase angle, \( E_r \), is used to define where in the relative ellipse the chaser satellite is currently at (with respect to relative periapsis of the chaser orbit and measured as a positive rotation about the \(-z_{LVLH}\) axis). Similarly, for the out-of-plane ellipse the amplitude of the cross-track sinusoidal motion, \( A_z \), can be specified, while \( \psi_z \) describes where the chaser is currently located in the out-of-plane ellipse (again, with respect to relative periapsis of the chaser orbit and measured as a positive rotation about the \(+y_{LVLH}\) axis).
To determine the orientation of the out-of-plane ellipse in the $x$-$z$ plane, first define the phase difference between the in-plane and out-of-plane ellipses as

$$\gamma_{\text{diff}} = E_r - \psi_z$$  \hspace{1cm} (5.26)

Then, the radial-intercept or $x$-intercept locations for the out-of-plane ellipse can be written as

$$x_{\text{int}} = x_r \pm \frac{a_r}{2} \cos(\gamma_{\text{diff}})$$  \hspace{1cm} (5.27)

From this result, if the phase difference of the relative ellipse is $\gamma_{\text{diff}} = \pm \pi/2$, the out-of-plane ellipse will cross the plane $z = 0$ and the radial axis of LVLH at the $x_r$ center of motion. This is an example of a perfectly out-of-phase relative ellipse. If the phase difference is $\gamma_{\text{diff}} = 0$ or $\pi$, then the relative ellipse will cross the plane $z = 0$ and the radial axis of LVLH at $x_{\text{int}} = x_r \pm \frac{a_r}{T}$. This second example is termed an in-phase safety ellipse, and signifies that the in-plane and out-of-plane phase angles are equal, or $E_r = \psi_z$.

Note that the first two ROEs for the center of motion of the relative ellipse are linear in terms of the original position and velocity, while the remaining ROEs are nonlinear functions of these states. However, in Section 5.3 a new set of relative orbital element states that are
functions of the HCW relative orbital elements, and linear functions of the original HCW position and velocity states are presented.

5.2.1 ROE Differential Equations

Taking derivatives of Eqs. 5.20-5.25, the differential equations for the relative orbital elements are derived. These nonlinear differential equations are written in terms of the control acceleration vector, $a_T = [a_x \ a_y \ a_z]^T$, and the constant of motion $\omega$. Analyzing these equations allows for important characteristics of optimal maneuvers to be defined in terms of the ROEs.

$$\dot{x}_r = \frac{2}{\omega}a_y$$  \hspace{1cm} (5.28)

$$\dot{y}_r = -\frac{3}{2} \omega x_r - \frac{2}{\omega} a_x$$  \hspace{1cm} (5.29)

$$\dot{a}_r = \frac{2}{\omega} \sin(E_r) a_x + \frac{4}{\omega} \cos(E_r) a_y$$  \hspace{1cm} (5.30)

$$\dot{E}_r = w + \frac{2}{\omega a_r} \cos(E_r) a_x - \frac{4}{\omega a_r} \sin(E_r) a_y$$  \hspace{1cm} (5.31)

$$\dot{A}_z = \frac{1}{\omega} \cos(\psi_z) a_z$$  \hspace{1cm} (5.32)

$$\dot{\psi}_z = w - \frac{1}{\omega A_z} \sin(\psi_z) a_z$$  \hspace{1cm} (5.33)

From these equations, it can be shown that for zero-input, $\nu(t) = 0$, the dynamics for the ROEs can be simply derived. These are written as

$$\dot{x}_r = \dot{a}_r = \dot{A}_z = 0$$  \hspace{1cm} (5.34)

$$\dot{y}_r = \frac{3}{2} \omega x_r$$  \hspace{1cm} (5.35)

$$\dot{E}_r = \dot{\psi}_z = \omega$$  \hspace{1cm} (5.36)

These zero-input equations prove to be very useful in the formulation of passive abort safety ellipse constraints.
5.2.2 Properties of the ROE Differential Equations

A few characteristics of the differential equations shown in Eqs. 5.28-5.33 are now described. First, it is noted that the $x_r$, $y_r$ center of motion is entirely linear in terms of the ROEs and control input; this will also be true in the LROE transformation that follows. The secular drift of the $y_r$ along-track center of motion is seen to be a linear function of the $x_r$ radial center of motion. For unforced motion it is conveniently shown that $y_r(t_f) = y_r(t_0) - \frac{3}{2} \omega x_r(t_f - t_0)$.

Secondly, the coefficients of the terms with control accelerations show where locations for optimal control occur, for example, changes in $a_r$ are most efficiently executed when $\sin(E_r) = 0$, and $\cos(E_r) = -1, 1$ (or equivalently, when $E_r = 0, \pi$). This corresponds to apogee and perigee of the relative orbit. Another example is that efficient changes in $A_z$ occur at $\cos(\psi_z) = -1, 1$ (or when $\psi_z = 0, \pi$). This corresponds to the locations of maximum and minimum cross-track amplitude.

Finally, the differential equations for the phase angles show that with zero input, the phase angles change linearly with $\omega t$. Additionally, any forced changes of these two ROEs can be much more costly when compared to the others. This is evident since the coefficients on the control terms for $E_r$ contain the inverse of $a_r$, and the coefficient on the $\dot{\psi}_z$ term contains the inverse of $A_z$. Therefore, for large $a_r$ and/or $A_z$ relative ellipses, it is generally better to avoid changing $E_r$ and/or $\psi_z$ phase angles, if possible. From these equations we also find that for orbital maneuvers with large final times, phase angle changes will be gradual, with slight changes occurring over multiple burns. The converse is also true, where for orbital maneuvers with short final times ($< 1$ orbital period), the phase angles may require significant changes in fewer burns.

5.3 Linear Relative Orbital Elements

The dynamics of the RPO problem can be transformed into a new set of ROE states that are linear functions of the original HCW states, and their differential equations remain LTI. They are termed the linear relative orbital element (LROE) states [136]. The new system of equations decouples the relative motion into three components: the in-plane
secular drift terms, an in-plane harmonic oscillator, and an out-of-plane harmonic oscillator (which is identical to the HCW out-of-plane oscillator). The new LROE states will be used to formulate relative motion trajectories in terms of traveling ellipses which, if properly constrained, can ensure a passively safe trajectory.

The new LROE state vector \( \bar{x} \) can be defined in terms of either the HCW Cartesian states or the traditional ROE states as

\[
\bar{x} = \begin{bmatrix}
\bar{x}_1 \\
\bar{x}_2 \\
\bar{x}_3 \\
\bar{x}_4 \\
\bar{x}_5 \\
\bar{x}_6 \\
\end{bmatrix} = \begin{bmatrix}
4x + \frac{2y}{\omega} \\
y - \frac{2x}{\omega} \\
\frac{2z}{\omega} \\
6x + \frac{4y}{\omega} \\
z \\
\frac{z}{\omega} \\
\end{bmatrix} = \begin{bmatrix}
x_r \\
y_r \\
a_r \sin(E_r) \\
a_r \cos(E_r) \\
A_z \sin(\psi_z) \\
A_z \cos(\psi_z) \\
\end{bmatrix}
\]

(5.37)

Furthermore, the ROEs can be also be written in terms of the new LROEs as

\[
x_r = \bar{x}_1, \quad y_r = \bar{x}_2
\]

(5.38)

\[
a_r = ||[\bar{x}_3 \quad \bar{x}_4]||, \quad A_z = ||[\bar{x}_5 \quad \bar{x}_6]||
\]

(5.39)

\[
E_r = \arctan\left(\frac{\bar{x}_3}{\bar{x}_4}\right), \quad \psi_z = \arctan\left(\frac{\bar{x}_5}{\bar{x}_6}\right)
\]

(5.40)

Using Eq. 5.37, the linear transformation matrix from the original HCW state to the LROE state is

\[
T = \frac{\partial \bar{x}}{\partial x} = \begin{bmatrix}
4 & 0 & 0 & 0 & 2/\omega & 0 \\
0 & 1 & 0 & -2/\omega & 0 & 0 \\
0 & 0 & 0 & 2/\omega & 0 & 0 \\
6 & 0 & 0 & 0 & 4/\omega & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1/\omega \\
\end{bmatrix}
\]

(5.41)
The new state dynamics follow the relationship

$$\dot{\bar{x}} = \bar{A}\bar{x} + \bar{B}\nu$$  \hspace{1cm} (5.42)

where, a new control, $\nu$, is defined as thrust acceleration divided by $\omega$, which now has units of velocity.

$$\nu = \frac{a_r}{\omega} = \begin{bmatrix} \nu_x \\ \nu_y \\ \nu_z \end{bmatrix}$$

The transformed $\bar{A}$ and $\bar{B}$ matrices are written as

$$\bar{A} = TAT^{-1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -3\omega/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \omega & 0 & 0 & 0 \\ 0 & 0 & -\omega & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \omega \\ 0 & 0 & 0 & 0 & -\omega & 0 \end{bmatrix}, \quad \bar{B} = TB = \begin{bmatrix} 0 & 2 & 0 \\ -2 & 0 & 0 \\ 2 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \hspace{1cm} (5.43)$$

The $\bar{A}$ matrix now takes a form similar to a block-diagonal Jordan form. There are two rigid body modes associated with the center motion of the relative ellipse, and an in-plane and out-of-plane harmonic oscillator. The two center of motion states are decoupled from the remaining two in-plane states. For the open loop system, the first two rigid body mode eigenvalues correspond to the $(x_r, y_r)$ center of motion of the relative ellipse, while the remaining eigenvalues include two simple harmonic oscillator modes (two pairs of $\pm i$). The LROEs are illustrated in Fig. 5.8.
Fig. 5.8: The Linear Relative Orbital Elements

To further simplify these linearized equations, a normalized-time form for the LROE dynamics is developed. Define a new variable for time \( \tilde{t} \), as \( \tilde{t} = \omega t \). This will produce relative motion dynamics in terms of the derivative of the new states with respect to the change in angle of the target in the circular reference orbit. The derivative of the new state with respect to \( \tilde{t} \) is written as

\[
\tilde{x}' = \frac{d\tilde{x}}{d\tilde{t}} = \frac{\dot{x}}{\omega} \tag{5.44}
\]

With this new time variable, the linearized state equations can then be written independent of the constant \( \omega \). Define the normalized-time control, \( \mu \), as

\[
\mu = \frac{\alpha x}{\omega^2} = \begin{bmatrix} \mu_x \\ \mu_y \\ \mu_z \end{bmatrix} \tag{5.45}
\]

Which, as a result of the previous transformations, now has the units of length. Thus, the LROE normalized-time states all have the same units. The simplified linearized state dynamics with respect to the change in angle of the target in its circular orbit, and with the new control, are
\[
\dot{x'} = \bar{A}_n \bar{x} + \bar{B} \mu
\]

where
\[
\bar{A}_n = \bar{A}/\omega = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
-3/2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 \\
\end{bmatrix}
\]

5.4 Pseudo-chief Reference Frame

Often, in the initial phases of far-to-mid-field RPO trajectory design, the chief satellite has not yet been acquired by the deputy. In this case, the LVLH frame for the chief may not be well-defined. Therefore, it may be advantageous to describe the relative motion of the deputy satellite with respect to and as observed from a ‘pseudo-chief’ reference frame, for the purpose of planning and executing optimal orbital maneuvers to correct relative ellipse size, position, or phasing [137]. This frame is not centered on any actual object, but is instead defined to be located in a perfectly circular reference orbit. One such reference orbit may be defined as a circular orbit with the same inertial semi-major axis as the deputy’s orbit. An illustration for the implementation of a pseudo-chief in this case is provided in Fig. 5.9.
As seen in Fig. 5.9, the relative motion of the deputy can be described entirely with respect to the pseudo-chief. The problem of determining optimal maneuvers from an initial state to a final state in this frame (with a known final time), can also be considered. This simplifies the problem, since the pseudo-chief’s reference LVLH frame is typically known as accurately as the knowledge of the deputy’s inertial state.

If we introduce the chief’s true orbit into the pseudo-chief’s reference frame, and consider chief and deputy eccentricities that satisfy \( e_c < e_d < 0.01 \), both trajectories relative to the pseudo-chief would appear as shown in the center of Fig. 5.10. However, the deputy’s relative ellipse with respect to the chief’s reference frame would still look as pictured on the right in Fig. 5.10.

\[
d_c = \frac{a_{c \pi}}{2} \approx a_c e_c \\
d_d = \frac{a_{d \pi}}{2} \approx a_d e_d
\]

Fig. 5.10: Pseudo-chief reference frame with deputy and chief trajectories

The purpose of introducing the pseudo-chief is to show the versatility and degrees of freedom when it comes to selecting the reference orbit of choice. As previously mentioned, these ideas may prove to be valuable for solving optimal far-to-mid-field trajectory planning problems to adjust relative ellipse size, position, and phasing. However, though these ideas may apply in some trajectory design and approach scenarios, this theory is not specifically addressed in this report, and is left for future work.

5.5 Common Relative Motion Trajectories

Several common in-plane relative motion trajectories are shown in this section, for
reference in later chapters, beginning with the simplest example and increasing in complexity [95]. In general, the in-plane trajectory described by the HCW equations of motion is similar to that of a cycloid [138]. A cycloid is the path traced out by an off-center point in circle that is rotating without slipping. In the HCW equations, however, the circle is replaced by a 2×1 ellipse, and instead of rotating, the ellipse is ‘drifting’ (i.e. slipping). This motion can be easily described by the LROE differential equations, where the in-plane center of motion dictates how rapidly the ellipse is drifting, and the other two in-plane states describe the simple harmonic oscillator for the ‘rotating’ motion.

The first example is termed a ‘station-keeping’ orbit. This trajectory involves a separation distance in the along-track direction (either positive or negative), with zero radial component. Additionally, the relative velocity is zero. An example of the in-plane conditions for this trajectory are written in terms of the relative position and velocity, and LROEs (\(\bar{x}_{1,2,3,4}\)) in Eq. 5.48. A v-bar station-keeping trajectory in front of the chief is shown in the inertial and LVLH frames in Fig. 5.11.

\[
\mathbf{r}_{\text{rel}}(t_0) = \begin{bmatrix} 0 \\ y_{\text{des}} \\ 0 \end{bmatrix}, \quad \mathbf{v}_{\text{rel}}(t_0) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad \text{or,} \quad \bar{x}(t_0) = \begin{bmatrix} 0 \\ y_{r_{\text{des}}} \\ 0 \\ 0 \\ 0 \end{bmatrix}
\]  

(5.48)

Fig. 5.11: Station-keeping trajectory example
The next example is called a ‘flyby’ orbit (often termed ‘co-elliptic flyby’, though this is generalized to the condition that \( e_d a_d = e_c a_c \)). In this scenario, the deputy is in a co-circular orbit either above of below the chief, and therefore has a larger/smaller semi-major axis (and longer/shorter period). An example of the in-plane conditions for this trajectory are written in terms of the relative position and velocity, and LROEs in Eq. 5.49. A flyby-above trajectory is shown in the inertial and LVLH frames in Fig. 5.12.

\[
\begin{align*}
\mathbf{r}_{\text{rel}}(t_0) &= \begin{bmatrix} x_{\text{des}} \\ y_{\text{des}} \\ 0 \end{bmatrix}, \\
\mathbf{v}_{\text{rel}}(t_0) &= \begin{bmatrix} 0 \\ -\frac{3}{2} \omega x_{\text{des}} \\ 0 \end{bmatrix}, \text{ or, } \mathbf{x}(t_0) = \begin{bmatrix} x_{r\text{des}} \\ y_{r\text{des}} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}
\end{align*}
\] (5.49)

Fig. 5.12: Flyby trajectory example

The following trajectory is known as a ‘v-bar hopping’ orbit (or ‘v-bar hop’). In this scenario, the deputy is in an elliptic orbit with a larger/smaller semi-major axis (and longer/shorter period) than the chief, and is ‘hopping’ either towards or away from the chief, and either above or below the v-bar. An example of the in-plane conditions for this
trajectory are written in terms of the relative position and velocity, and LROEs in Eq. 5.50, where the in-plane velocity is written as a function of the desired radian center of motion for the hopping ellipse. A v-bar hop trajectory approaching the chief from in front is shown in the inertial and LVLH frames in Fig. 5.13.

\[
\mathbf{r}_{rel}(t_0) = \begin{bmatrix} 0 \\ y_{des} \\ 0 \end{bmatrix}, \quad \mathbf{v}_{rel}(t_0) = \begin{bmatrix} 0 \\ \frac{1}{2} \omega x_{r_{des}} \\ 0 \end{bmatrix}, \quad \text{or,} \quad \mathbf{x}(t_0) = \begin{bmatrix} x_{r_{des}} \\ y_{r_{des}} \\ 0 \\ 2x_{r_{des}} \\ 0 \\ 0 \end{bmatrix} \tag{5.50}
\]

The following trajectory is a very common relative orbital motion trajectory known as a ‘football’ orbit. For this trajectory, the deputy is in an elliptic orbit with the same semi-major axis (and same period) as the chief. Due to the natural motion of the HCW dynamics, this trajectory traces out a 2×1 ellipse, which looks similar to a football. An example of the in-plane conditions for this trajectory are written in terms of the relative position and velocity, and LROEs in Eq. 5.51, where the in-plane velocity is written as a

Fig. 5.13: V-bar hopping trajectory example
function of the semi-major axis of the football ellipse. A football orbit that circumnavigates the chief is shown in the inertial and LVLH frames in Fig. 5.14.

\[
\begin{align*}
\mathbf{r}_{rel}(t_0) &= \begin{bmatrix} x_{des} \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{v}_{rel}(t_0) = \begin{bmatrix} 0 \\ \omega a_{r_{des}} \\ 0 \end{bmatrix}, \quad \text{or}, \quad \mathbf{x}(t_0) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ ar_{des} \\ 0 \end{bmatrix}
\end{align*}
\] (5.51)

The final trajectory example shown here may be used to inject into a ‘teardrop’ orbit. This trajectory is similar to the v-bar hop in that the deputy is in an elliptic orbit with either a larger/smaller semi-major axis (and larger/smaller orbital period) as the chief, however, the radial center of motion \((x_r)\) is larger than it was for the v-bar (with the same semi-major axis \(a_r\)). The cycloid-like motion due to the drift and harmonic oscillator in the HCW equations can clearly be seen in this case. An example of the in-plane conditions for this trajectory are written in terms of the in-plane relative position and velocity, and in-plane LROEs in Eq. 5.52, where the position/velocity is written as a function of the
radial center of motion and semi-major axis of the drifting ellipse. A teardrop-inject orbit that approaches the chief from above is shown in the inertial and LVLH frames in Fig. 5.15.

\[
\begin{align*}
\mathbf{r}_{rel}(t_0) &= \begin{bmatrix} x_{r,des} - \frac{1}{2} a_{r,des} \\ y_{des} \\ 0 \end{bmatrix}, \\
\mathbf{v}_{rel}(t_0) &= \begin{bmatrix} 0 \\ \omega (a_{r,des} - \frac{3}{2} x_{r,des}) \\ 0 \end{bmatrix}, \quad \text{or,} \quad \mathbf{x}(t_0) = \begin{bmatrix} x_{r,des} \\ y_{r,des} \\ 0 \\ a_{r,des} \\ 0 \end{bmatrix}
\end{align*}
\] (5.52)

Fig. 5.15: Teardrop-inject orbit trajectory example
CHAPTER 6
THE GENERALIZED HCW EQUATIONS

While the HCW model provides a convenient linearized form of the relative motion equations that can be utilized in many scenarios for close proximity of a circular reference orbit, it may be restrictive when applied to the area of RPO about an uncontrolled, spinning spacecraft. The analysis performed in this chapter considers a different reference frame, a body-fixed frame rotating with the chief, and the associated relative orbital motion equations are derived. This formulation allows for the possibility of constraints on trajectory design that can be coordinated directly in the chief spacecraft’s body-fixed frame. These constraints include approach corridors, spherical KOZ, and a safe stand-off distance.

The objective is to reformulate the relative motion of a deputy satellite in the chief satellite’s body-fixed frame. Similar to the HCW formulation, the nonlinear equations of relative motion are linearized about a circular orbit. However, this analysis considers two different formulations for the rotational dynamics of the chief’s body-fixed frame. One formulation assumes that the reference frame is rotating with an arbitrary inertial constant rotation vector. Another application allows for the possibility of a time-varying rotation vector of the chief’s body-fixed frame. In both cases, the relative motion is directly coordinated in a rotating chief body-fixed frame. The resulting equations can be used to generate relative trajectories in the chief’s rotating frame.

Formulating the problem in this way leads to a generalized form of the Hill, Clohessy-Wiltshire equations. However, by allowing a constant, or time-varying rotation vector of the relative frame, the linearized dynamics become time-varying. The new equations can be applied in trajectory planning scenarios in which a deputy satellite desires to maneuver around a spinning chief where the problem is best formulated in the chief’s body-fixed frame. An example of such a trajectory design would be for inspection, approach, and docking with a controlled or uncontrolled spinning chief spacecraft. The new formulation in
the spacecraft’s body-fixed frame would allow for, e.g., an approach corridor constraint to be directly coordinatized in the spacecraft’s body-fixed frame. Additional considerations for structural avoidance, positioning for optics/sensing, and plume impingement can be included as constraints in this dynamics model.

For the case in which the chief is rotating at the LVLH rate, the equations reduce to the standard HCW equations. If the chief is rotating at a rate much greater than that of the LVLH frame, the new formulation produces a linear time invariant form which is called the free-space scenario, since the affect of gravity is negligible when compared to the centrifugal acceleration, over relatively short periods of time. In this case, a simple targeting algorithm is developed, termed Inertial Targeting. This targeting method is tested for a variety of chief rotation rates and final transfer times.

The remainder of this chapter is organized as follows. First the two-body nonlinear equations of relative motion are presented in a chief-centered rotating reference frame. These equations are then linearized and coordinatized in the chief’s body-fixed frame and produce the desired linear time-varying differential equations of relative orbital motion with respect to an arbitrary rotating frame. Two possibilities are considered: a constant rotation vector, and a time-varying rotation vector of the chief’s body-fixed frame. In the first instance, it is assumed that the chief spacecraft is rotating, uncontrolled, about one of its principal axes. This may be a result of damping from gravity gradient torques, drag, or other perturbations. In the second case, the dynamics allow for a time-varying rotation vector, assumed to be known from on-board estimation algorithms, which includes the possibility of a tumbling chief spacecraft. These equations are shown to reduce to the HCW equations when the chief’s body frame is rotating at LVLH rate. When the chief’s body frame is rotating at a rate much greater than LVLH, it is shown that the the linear time-varying differential equations reduce to a set of linear time-invariant differential equations. An analytical solution to these LTI differential equations is presented and a simple targeting algorithm for this scenario based on the state transition matrix is developed.
6.1 The Generalized HCW Equations

The inertial accelerations of the two spacecraft (chief and deputy), exhibiting Keplerian motion, are given by

\[ a_c = -\frac{\mu}{||R_c||^3}R_c, \quad a_d = -\frac{\mu}{||R_d||^3}R_d \]  \hspace{1cm} (6.1)

where \( R_c \) and \( R_d \) are the inertial positions of the chief and deputy, respectively.

If a reference frame attached to the chief is rotating with angular velocity vector \( \omega \) with respect to an inertial frame, the acceleration of the deputy relative to the chief as observed in the chief’s rotating body-fixed frame is given by [131, 132]

\[ a_{rel} = a_d - a_c - 2\omega \times v - \omega \times (\omega \times r) - \dot{\omega} \times r \]  \hspace{1cm} (6.2)

where \( r \) is the position of the deputy relative to the chief, and \( v = \dot{r} \) is the relative velocity of the deputy with respect to the chief as observed in the rotating chief’s body-fixed frame. Thus the orbital relative motion as observed in the rotating body-fixed frame is described by

\[ \dot{r} = v \]  \hspace{1cm} (6.3)

\[ \dot{v} = -\frac{\mu}{||R_d||^3}R_d + \frac{\mu}{||R_c||^3}R_c - 2\omega \times v - \omega \times (\omega \times r) - \dot{\omega} \times r \]  \hspace{1cm} (6.4)

where

\[ R_d = R_c + r \]  \hspace{1cm} (6.5)

6.1.1 Linearized Equations of Motion for Constant Angular Rotation Vector

If the relative position is much smaller than the chief orbital radius \( R_c \), \( ||r|| \ll ||R_c|| \) then a Taylor series expansion of \( a_d \) to first-order results in

\[ a_d = -\frac{\mu}{||R_c + r||^3} (R_c + r) \approx -\frac{\mu}{||R_c||^3}R_c - \frac{\mu}{||R_c||^3} \left( I_{3\times3} - 3\hat{R}_c\hat{R}_c^T \right) r \]  \hspace{1cm} (6.6)
where \( \hat{i}_{R_c} = R_c/\|R_c\| \). Substituting this into Eq. 6.4, and assuming a circular chief orbit such that the magnitude \( R_c = \|R_c\| \) is constant, produces

\[
\dot{r} = v \quad (6.7)
\]

\[
\dot{v} = -\Omega^2 \left( I_{3 \times 3} - 3\hat{i}_{R_c}(t)\hat{i}_{R_c}^T(t) \right) r - 2\omega \times v - \omega \times (\omega \times r) - \dot{\omega} \times r \quad (6.8)
\]

The value for \( \Omega = \|\Omega\| \) in this chapter is defined to be the same value for the mean orbital rate of the chief’s reference orbit, shown by Eq. 5.5, and is distinguished from the rotation rate of the chief’s body-fixed frame, \( \omega = \|\omega\| \).

If the angular acceleration of the chief is \( \ddot{\omega} = 0 \) (this constraint can be relaxed later), and if all vectors are coordinatized in the chief’s rotating frame, Equations 6.7-6.8 reduce to a set of linear-time-varying differential equations

\[
\dot{r} = v \quad (6.9)
\]

\[
\dot{v} = -\Omega^2 \left[ I_{3 \times 3} - 3\hat{i}_{R_c}(t)\hat{i}_{R_c}^T(t) \right] r - 2\omega \times v - \omega \times (\omega \times r) \quad (6.10)
\]

which can be described in state-space form as

\[
\dot{x} = A(t)x
\]

where \( x = [r^T, v^T]^T \), and

\[
A(t) = \begin{bmatrix}
0_{3 \times 3} & I_{3 \times 3} \\
-\Omega^2 \left[ I_{3 \times 3} - 3\hat{i}_{R_c}(t)\hat{i}_{R_c}^T(t) \right] & -[\omega \times]^2 -2[\omega \times]
\end{bmatrix} \quad (6.11)
\]

and where the skew symmetric matrix \( [\omega \times] \) is given by

\[
[\omega \times] = \begin{bmatrix}
0 & -\omega_3 & \omega_2 \\
\omega_3 & 0 & -\omega_1 \\
-\omega_2 & \omega_1 & 0
\end{bmatrix}
\]
Notice that it is only the tensor product \( \hat{i}_{R_c}(t)\hat{i}_{R_c}^T(t) \) that is time-varying. This term is a projection matrix onto the span of \( R_c \), the position vector of the chief, coordinatized in the body frame. This matrix at time \( t \) can be written as

\[
P_{R_c}(t) = \frac{R_c^B(t)R_c^B(t)^T}{\|R_c\|^2}
\]

where the position of the chief in the body frame is

\[
R_c^B(t) = T_I^B(\omega, t)R^I_c(t)
\]

The matrix \( T_I^B \) is a transformation matrix from the inertial frame to the body fixed frame. This transformation can be separated into two transformations

\[
T_I^B(\omega, t) = T_{B_0}^B(\omega, t)T_I^{B_0}
\]

where \( T_{B_0}^B \) is a known constant and defines the rotation from the inertial frame to the body frame at time zero, and \( T_{B_0}^B(\omega, t) \) defines the rotation from the body frame at time zero to the body frame at time \( t \). \( T_{B_0}^B(\omega, t) \) is function of only the angular velocity vector, \( \omega \), of the spacecraft’s body-fixed frame, and time. It is assumed without loss of generality that the rotation of the spacecraft \( \omega \) is aligned with the body \(-z\)-axis. Thus, this rotation matrix is given by

\[
T_{B_0}^B(\omega I, t) = \cos \theta_\omega I + \sin \theta_\omega \begin{bmatrix} i^B_z \times \end{bmatrix} + (1 - \cos \theta_\omega) \hat{i}^B_z \otimes \hat{i}^B_z
\]

where \( \otimes \) is the tensor product, and where the angle of rotation about the \(-z\)-axis at time \( t \) is

\[
\theta_\omega(t) = \omega t
\]

The inertial position of the chief spacecraft at time \( t \) is obtained by rotating the initial position in the inertial frame by the constant angular rate of the orbit, \( \Omega \).
\[ R_c^I(t) = R(\Omega^I, t)R_c^I(t_0) \] (6.14)

where

\[ R(\Omega^I, t) = \cos \theta \Omega I_{3 \times 3} + \sin \theta \Omega \left[ \hat{i}_\Omega \times \right] + (1 - \cos \theta \Omega) \hat{i}_\Omega \otimes \hat{i}_\Omega \] (6.15)

and \( \hat{i}_\Omega \) is the unit orbital angular velocity vector, and the orbit rotation angle is \( \theta \Omega(t) = \Omega t \).

The projection matrix becomes

\[
P_c(\Omega, \omega, t) = \frac{\left[ T_B^B(\omega, t)T_I^{R_0} R(\Omega, t)R_c^I(t_0) \right] \left[ T_B^B(\omega, t)T_I^{R_0} R(\Omega, t)R_c^I(t_0) \right]^T}{||R_c||^2}
\]

Thus, given the constant values of \( \omega, \Omega, T_I^{B_0}, \) and \( R_c^I(t_0) \), the LTV differential equations of relative motion are completely determined.

\[
\dot{x} = A(t)x \quad (6.16)
\]

where \( x = [r^T, v^T]^T \), and

\[
A(t) = \begin{bmatrix}
0_{3 \times 3} \\
-\Omega^2 \left( I_{3 \times 3} - 3 \left[ T_B^B(\omega, t)T_I^{R_0} R(\Omega, t)\hat{i}_{R_c}(t_0) \right] \left[ T_B^B(\omega, t)T_I^{R_0} R(\Omega, t)\hat{i}_{R_c}(t_0) \right]^T \right) - [\omega \times]^2 & -2[\omega \times]
\end{bmatrix}
\]

where the chief’s inertial position vector at the initial time \( t_0 \) is \( \hat{i}_{R_c}(t_0) = R_c^I(t_0)/||R_c^I(t_0)||. \)

### 6.1.2 Linearized Equations of Motion for Time-Varying Angular Rotation Vector

A non-constant rotation vector, \( \omega(t) \), can be accommodated without much difficulty. The dynamics remain time-varying but now include the possibility of a tumbling chief body-fixed frame. The resulting dynamics are

\[
\dot{x} = A(t)x \quad (6.18)
\]
where now, the $A(t)$ matrix is written as

$$A(t) = \begin{bmatrix} 0_{3 \times 3} & I_{3 \times 3} \\ \frac{\partial \mathbf{v}(t)}{\partial t} & -2[\mathbf{\omega}(t) \times] \end{bmatrix}$$

(6.19)

where

$$\frac{\partial \mathbf{v}(t)}{\partial t} = -\Omega^2 \left( I_{3 \times 3} - 3 \left[ T_{B_0}^B(\mathbf{\omega}(t), t) T_{I}^{B_0} R(\Omega, t) \hat{I}_{R_c}(t_0) \right] \left[ T_{B_0}^B(\mathbf{\omega}(t), t) T_{I}^{B_0} R(\Omega, t) \hat{I}_{R_c}(t_0) \right]^T \right) - \mathbf{\omega}(t) \times \mathbf{\omega}(t)$$

(6.20)

Notice the addition of the time-varying cross product matrix for the chief’s angular acceleration vector is

$$[\mathbf{\dot{\omega}}(t) \times] = \begin{bmatrix} 0 & -\dot{\omega}_3(t) & \dot{\omega}_2(t) \\ \dot{\omega}_3(t) & 0 & -\dot{\omega}_1(t) \\ -\dot{\omega}_2(t) & \dot{\omega}_1(t) & 0 \end{bmatrix}$$

Further, the time-varying rotation matrix from the chief’s body frame at time zero to the chief’s body frame at time $t$ follows the differential equation

$$\dot{T}_{B_0}^B(\mathbf{\omega}(t), t) = [\mathbf{\omega}(t) \times] T_{B_0}^B(t_0)$$

(6.21)

This differential equation for the transformation from the body frame at time zero to the body frame at time $t$ can be numerically integrated using measurements of $\mathbf{\omega}(t)$, to calculate $T_{B_0}^B(\mathbf{\omega}(t), t)$ in Eq. 6.20. These LTV differential equations describe the orbital relative motion as viewed from a frame fixed to a controlled or uncontrolled spinning spacecraft.

### 6.2 Reduction to HCW Equations

To validate the body-frame formulation, it is shown that the linear-time-varying differential equations reduce to the linear time invariant Hill, Clohessey-Wiltshire equations
when the vehicle’s body frame is nadir pointing, and rotating at the LVLH rate. Let

$$R(\Omega^I, t) = \begin{bmatrix} \cos(\Omega t) & -\sin(\Omega t) & 0 \\ \sin(\Omega t) & \cos(\Omega t) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$ (6.22)

Next, let the initial normalized position vector of the chief in the inertial frame be

$$\hat{i}^I_{Rc}(t_0) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}^T$$ (6.23)

Then,

$$\hat{i}^B_{Rc}(t) = T^B_{B0}(\omega, t) T^I_{I0} \begin{bmatrix} \cos(\Omega t) \\ \sin(\Omega t) \\ 0 \end{bmatrix}$$ (6.24)

Furthermore, if the body frame is arbitrarily aligned with the LVLH frame at $t_0$, it is also aligned with the inertial frame at $t_0$, so $T^I_{I0} = I$. Then, if the angular rotation rate of the body is equal to the angular rotation rate of LVLH, i.e., $\omega^B = \Omega \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}^T$, the resulting unit position vector as a function of time in the body frame is

$$\hat{i}^B_{Rc}(t) = \begin{bmatrix} \cos(\Omega t) & \sin(\Omega t) & 0 \\ -\sin(\Omega t) & \cos(\Omega t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\Omega t) \\ \sin(\Omega t) \\ 0 \end{bmatrix}^I = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}^B$$ (6.25)

which is no longer time-varying. Substituting $\hat{i}^B_{Rc} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}^T$ this into Eqs. 6.16-6.17 produces

$$\dot{x} = Ax$$ (6.26)

where

$$A = \begin{bmatrix} 0_{3\times3} & I_{3\times3} \\ -\Omega^2 \left( I_{3\times3} - 3 \begin{bmatrix} \hat{i}^B_{Rc} \\ \hat{i}^B_{Rc} \end{bmatrix}^T \right) - [\omega \times]^2 & -2[\omega \times] \end{bmatrix}$$ (6.27)
When \( \omega = [0 \ 0 \ \omega]^T \) and \( \hat{\mathbf{i}}_{\mathbf{B}}^{\mathbf{R}_c} = [1 \ 0 \ 0]^T \), then \( A \) simplifies to

\[
A = 
\begin{bmatrix}
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
3\omega^2 & 0 & 0 & 0 & 2\omega & 0 \\
0 & 0 & 0 & -2\omega & 0 & 0 \\
0 & 0 & -\omega^2 & 0 & 0 & 0
\end{bmatrix}
\] (6.28)

These equations are identical to the HCW equations shown in Eq. 5.2 and 5.4 [66].

### 6.3 Approximate Analytic Solution When \( \Omega^2 \ll \omega^2 \)

When the chief’s body-fixed frame is rotating at constant rate, and the rotation is much larger than the rate of LVLH, i.e., \( \Omega^2 \ll \omega^2 \), the differential gravitational acceleration term associated with \( \Omega^2 \) becomes much smaller than the centripetal acceleration term \( [\omega \times]^2 \). In this instance, the differential gravitational acceleration term may be neglected, and the LTV differential equations of motion in Eqs. 6.9-6.10 reduce to a set of LTI differential equations.

\[
\dot{\mathbf{r}} = \mathbf{v} 
\] (6.29)

\[
\dot{\mathbf{v}} = -2\omega \times \mathbf{v} - \omega \times ( \omega \times \mathbf{r} )
\] (6.30)

These equations can be written in state space form as

\[
\dot{\mathbf{x}} = A\mathbf{x}
\]

where \( \mathbf{x} = [\mathbf{r}^T, \mathbf{v}^T]^T \), and

\[
A = \frac{\partial \mathbf{x}}{\partial \mathbf{x}} = 
\begin{bmatrix}
0_{3\times3} & I_{3\times3} \\
-\omega \times] & -2[\omega \times]
\end{bmatrix}
\]
This is an important result since the angular velocity of a spacecraft $\omega$ may often be much greater than the orbital rates $\Omega_{LEO} \approx 0.06$ degrees per second or $\Omega_{GEO} \approx 0.004$ degrees per second. Although the gravitational acceleration is much smaller than the centrifugal acceleration, its effect over long periods of time will accumulate, and thus, these LTI equations are only applicable over relatively short periods of time. This effect is shown in the inertial targeting and rotating chief body-fixed frame results in Chapter 8.

The state-transition matrix of the LTI system can be written as

$$
\Phi(t-t_0) = \begin{bmatrix}
C_z (I + (t-t_0) [\omega \times]_z) & (t-t_0) C_z \\
(t-t_0) \omega^2 \left( C_z - \hat{i}_z \hat{i}_z^T \right) C_z (I - (t-t_0) [\omega \times]_z)
\end{bmatrix} =
\begin{bmatrix}
\Phi_{rr}(t-t_0) & \Phi_{rv}(t-t_0) \\
\Phi_{vr}(t-t_0) & \Phi_{vv}(t-t_0)
\end{bmatrix}
$$

(6.31)

where $C_z$, and $[\omega \times]_z$ are

$$
C_z = \begin{bmatrix}
\cos(\omega t) & -\sin(\omega t) & 0 \\
\sin(\omega t) & \cos(\omega t) & 0 \\
0 & 0 & 1
\end{bmatrix}
$$

and

$$
[\omega \times]_z = \omega \begin{bmatrix}
0 & -1 & 0 \\
1 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix} = \omega J
$$

where

$$
J = [\hat{i}_x \hat{i}_y^T - \hat{i}_y \hat{i}_x^T]
$$

and

$$
[\hat{i}_x \hat{i}_y^T] = \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 1
\end{bmatrix}
$$

Therefore, the solution to the LTI dynamics at time $t$ as a function of the initial state at time $t_0$ becomes
\[
x(t) = \Phi(t, t_0)x(t_0) = \begin{bmatrix} \Phi_{rr} & \Phi_{rv} \\ \Phi_{vr} & \Phi_{vv} \end{bmatrix} x(t_0)
\]

(6.32)

\[
\begin{bmatrix} r(t) \\ v(t) \end{bmatrix} = \begin{bmatrix} C_z \left(I + (t - t_0) [\omega \times]_z\right) & (t - t_0) C_z \\ (t - t_0) \omega^2 \left(C_z - i_z i_z^T \right) & C_z \left(I - (t - t_0) [\omega \times]_z\right) \end{bmatrix} \begin{bmatrix} r(t_0) \\ v(t_0) \end{bmatrix}
\]

(6.33)

### 6.3.1 Inertial Targeting Algorithm

Under the assumption that \( \Omega^2 \ll \omega^2 \), the state transition matrix can be used to determine the required initial relative velocity at time \( t_0 \) that puts the spacecraft on a path to reach a desired relative position, \( r_{\text{des}} \), in the fixed amount of time, \( t \). Solving for the desired relative velocity at \( t_0 \), gives the expression

\[
v(t_0) = [\Phi_{rv}(t, t_0)]^{-1} \left\{ r_{\text{des}}(t) - \Phi_{rr}(t, t_0)r(t_0) \right\}
\]

(6.34)

or substituting in previous definitions,

\[
v(t_0) = \frac{1}{t - t_0} \left\{ C_z^{-1} r_{\text{des}}(t) - \left[I + \theta \omega J\right] r(t_0) \right\}
\]

(6.35)

When transformed back to the inertial frame, this targeting equation reduces to the following expected form,

\[
v^I(t_0) = \frac{1}{t - t_0} \left( r^I_{\text{des}}(t) - r^I(t_0) \right)
\]

(6.36)

i.e., the required inertial velocity is simply the difference between the initial and final inertial positions divided by the fixed time. Thus, an approximate solution to maneuvering from one position relative to the rotating frame to another desired position relative to the rotating frame is simply linear motion in inertial space when the rotation rate of the vehicle is much greater than the orbital rotation rate. In this case, the simple targeting algorithm may be run closed-loop in a chief body-fixed frame trajectory following scenario, at minimal cost to delta-v.
Summary

This chapter introduces the relative motion equations for a deputy satellite, with respect to a rotating chief’s body-fixed frame. The resulting equations are LTV, however, when the rotation rate of the chief’s body-fixed frame is LVLH (nadir pointing), then the equations reduce to the standard HCW model. Furthermore, if the spin rate of the chief satellite is much greater than the orbital rate, the resulting dynamics may be reduced to a LTI model. In this case, an Intertial Targeting algorithm is developed. The rotating chief body-fixed model is used in Chapter 8 to formulate a convex RPO planner algorithm for maneuvers which are best described in an uncontrolled chief’s body-fixed frame. These trajectories include Way-point Following/Inspection and Final Approach cases, in which constraints such as approach corridors, spherical keep-out zones, and minimum safe stand-off distance are included.
CHAPTER 7
RPO PLANNING RELATIVE TO AN LVLH REFERENCE FRAME

Many RPO trajectory problems are best formulated in an LVLH frame centered on a target vehicle with the objective to minimize $\Delta v$, and subject to constraints such as approach corridors and keep-out zones. The Hill, Clohessy-Wiltshire equations in Cartesian coordinates are often utilized in this flight regime, provided the chief is in a near-circular orbit and the deputy vehicle is in close proximity of the chief.

In this chapter, the RPO trajectory planning problem is formulated as a convex optimization problem using the traditional Cartesian LVLH coordinates. This includes defining the problem dynamics, objective function, and constraints using the SOCP format presented in Chapter 3. The constraints in these problems include maximum thrust acceleration, approach corridors, planes, and spherical keep-out zone. While the first three constraints allow for fully convex problems, the spherical keep-out zone is inherently nonconvex, and therefore requires an iterative sequential convex programming (SCP) approach.

Once the problem constraints and objective is defined, the full problem formulation is stated. This is followed by the method for determining an optimal solution to the spherical keep-out zone using SCP. The remaining sections present the optimal trajectories that result from each of the different problem formulations.

7.1 Problem Formulation

Before the RPO trajectory planning problem can be expressed as a second-order cone program, several important steps must be taken. The first involves discretizing the problem dynamics. This is done using a standard state transition matrix for the relative position and velocity states in the LVLH frame. A brief derivation for the objective function which minimizes fuel over the duration of the trajectory is presented. Next, the objective function must be written in a linear form. Since the objective function is inherently nonlinear for the
RPO trajectory planning problem, a technique is used whereby slack variables are added so that the objective function is amenable to an SOCP. The final step in this section is to present the trajectory constraints which are considered in the problem.

7.1.1 Relative Motion Dynamics with Respect to an LVLH Frame

The continuous HCW dynamics are discretized into \( N \) equal time-steps, \( \Delta t \), where \( \Delta t = t_{i+1} - t_i \), for a total of \( x_i \) states for \( i = 1, \ldots, N + 1 \). These represent states at times \( t_i \), for \( i = 1, \ldots, N + 1 \), where initial time \( t(0) = t_1 \) and final time \( t_f = t_{N+1} \). It is assumed that the thrust acceleration is constant over each time-step, so that the control \( u_i = a_T(t_{i+1} - t_i) = a_T \). The solution to Eq. 5.2 can now be written in the state-transition matrix form.

\[
x_{i+1} = \Phi(t_{i+1}, t_i)x_i + B_d u_i
\]  

where the state transition matrix \( \Phi(t_{i+1}, t_i) \) and \( B_d \) are

\[
\Phi(t_{i+1}, t_i) = e^{A dt}, \quad B_d = \int_{t_i}^{t_{i+1}} \Phi(t_{i+1}, \tau) B d\tau
\]  

The \( A \) and \( B \) matrices are given in Eq. 5.4. Eq. 7.1 represents linear equality constraints that can be directly implemented in a SOCP. The initial state, \( x_1 \), and final state, \( x_{N+1} \), are the boundary constraints on these discretized dynamics equations, and are determined by the RPO trajectory specifications.

7.1.2 Objective Function

The objective for the optimal guidance problem is to minimize the amount of propellant used by a chemical combustion engine, over the duration of the transfer. The rate of propellant usage for a chemical engine is given by

\[
\dot{m}(t) = -\frac{||T(t)||_2}{V_{ex}} = -\frac{m(t)||a_T(t)||_2}{V_{ex}}
\]  

where \( T \) is the thrust vector, \( a_T(t) \) is the thrust acceleration vector, \( m(t) \) is the vehicle
mass, and \( V_{ex} \) is the exit velocity of the engine’s exhaust gas. Using separation of variables, this differential equation can be integrated directly to obtain

\[
\ln (m_f) - \ln (m_0) = \int_{m_0}^{m_f} \frac{1}{m(t)} \, dm = - \int_{t_0}^{t_f} \frac{||a_T(t)||}{V_{ex}} \, dt \quad (7.4)
\]

or

\[
\frac{m_f}{m_0} = 1 - \frac{m_p}{m_0} = \exp \left[ - \int_{t_0}^{t_f} \frac{||a_T(t)||}{V_{ex}} \, dt \right] \quad (7.5)
\]

Therefore, to maximize the final mass or minimize the propellant consumed over the duration of the trajectory, the object is to minimize

\[
J = \int_{t_0}^{t_f} ||a_T(t)|| \, dt \quad (7.6)
\]

This is the objective function for the formulations presented in this chapter, as well as Chapters 8 and 9. Note that this performance index is nonlinear and does not conform to the standard SOCP problem. This is remedied by the introduction of slack variables [11].

In this analysis it is assumed that the mass of the vehicle is constant, so that \( \dot{m}(t) = 0 \).

The objective function shown in Eq. 7.6 is to be discretized in order to be implemented in a convex optimization problem. It is assumed that the thrust acceleration over each time step is constant. If the trajectory is discretized into \( N \) equal segments, with points at times \( t_i \) for \( i = 1, 2, \ldots, N + 1 \), then the objective function is written as

\[
J = \sum_{i=1}^{N} ||a_{T_i}|| = \sum_{i=1}^{N} ||u_i|| \quad (7.7)
\]

where constant \( \Delta t = t_{i+1} - t_i \) is omitted, as it does not change the minimization of the sum of thrust acceleration magnitudes.

The nonlinear discretized objective function in Eq. 7.7 can be ‘convexified’, and written in a linear form as

\[
J = \Delta t \sum_{i=1}^{N} \eta_i \quad (7.8)
\]
by adding the following slack variable inequality constraints.

\[ ||u_i|| \leq \eta_i, \text{ for } i = 1, \ldots, N \]  (7.9)

\[ 0 \leq \eta_i \leq u_{\text{max}}, \text{ for } i = 1, \ldots, N \]  (7.10)

where a relaxed convex optimization problem is created, using a lossless transformation technique, whose solution is identical to the original problem [22].

The optimal RPO trajectory planning problem requires an SOCP formulation due to these inherent second-order conic constraints, which will be described in the following sections. However, the standard form of the SOCP does not immediately accommodate all the elements of the RPO trajectory planning problem, in particular, the nonlinear objective function. The nonlinear objective function will be transformed to a linear SOCP function by introducing slack variables, and new constraints, via lossless relaxation technique.

### 7.1.3 Constraints

The constraints for the RPO trajectory planner consist of linear equalities, linear inequalities, and second-order cones. All of the dynamics in the problem are written as linear equality constraints, while other important constraints such as maximum thrust acceleration and approach cones are written in a second-order conic form. The keep-out sphere is an inherently nonconvex inequality constraint, however, this is implemented using sequential convex programming as shown in the solution method.

### Boundary Conditions

For the RPO trajectory planner, it is assumed that the final time is a parameter that is known beforehand from mission requirements and planning. Therefore, the states are constrained at the initial time and final time by strict boundary conditions. It is assumed that the initial relative position and velocity is given, and the final relative position and velocity is specified a priori. Therefore, the boundary conditions on the problem are written
as
\[ x_1 = \begin{bmatrix} r_1 \\ v_1 \end{bmatrix} = \text{given}, \quad x_{N+1} = \begin{bmatrix} r_{N+1} \\ v_{N+1} \end{bmatrix} = \text{specified} \] (7.11)

**Maximum Thrust Acceleration**

The constraint on maximum magnitude of thrust acceleration over each time step is

\[ ||u_i|| \leq u_{\text{max}} \text{ for } i = 1, \ldots, N \] (7.12)

where \( u_{\text{max}} \) is the maximum thrust acceleration magnitude.

**Conic Approach Corridor**

A cone constraint is optionally used to define an approach corridor for the trajectory. This is formulated as an inner product of the relative position vector and a unit vector \( \hat{i}_c(t) \) in the rotating frame coordinates (LVLH or other rotating frame) defining the center-line of the cone [22]. The cone constraint is given by

\[ ||r(t)|| \cos(\alpha) \leq \hat{i}_c^T(t) r(t) \] (7.13)

where \( \alpha \) corresponds to the cone half-angle. If we wish to implement a half-space constraint for the relative position, this includes defining an inequality constraint using a plane. This constrains the relative position such that it must lie a minimum distance, \( d_{\text{min}} \), away in a given direction, \( \hat{i}_p(t) \). This can be written as

\[ d_{\text{min}} \leq \hat{i}_p^T(t) r(t) \] (7.14)

Since these are convex inequality constraints, they can be implemented directly in a SOCP.
Spherical Keep-out Zone

An optional keep-out zone constraint is defined by the nonconvex function

\[ ||r(t)|| \geq r_{\text{min}} \]  \hspace{1cm} (7.15)

where \( r_{\text{min}} \) is the spherical keep out zone radius. The keep-out zone, however, represents a concave solution set and is not directly amenable to a SOCP. This will be dealt with in Section 3.2.

Drift Without Control Constraints

The cone, plane, and keep-out zone constraints are strictly enforced, and results have shown that under the assumption of constant acceleration over each time step, the control required in many cases resembles a step-like function on the boundary of the constraint. These solutions require alternating high/low thrust to stay strictly on the cone or sphere boundary. If we desire a more continuous control solution, we may relax these constraints in a formulation that requires the position at \( i \), without the previous control at \( i - 1 \), to remain within the cone or outside the sphere. These are termed the drift without control constraints, and for the cone, plane, and sphere they are written as

\[ ||M_r \Phi(t_{i+1}, t_i)x_i|| \cos(\alpha) \leq \hat{t}_c^T (M_r \Phi(t_{i+1}, t_i)x_i) \]  \hspace{1cm} (7.16)

\[ d_{\text{min}} \leq \hat{t}_p^T (t_i)M_p \Phi(t_{i+1}, t_i)x_i \]  \hspace{1cm} (7.17)

\[ ||M_r \Phi(t_{i+1}, t_i)x_i|| \geq r_{\text{min}} \]  \hspace{1cm} (7.18)

respectively, for \( i = 1, \ldots, N \), where \( M_r \) is a selection matrix that selects the position component of the state vector. Examples of implementing these constraints are included in the results section of this chapter.
7.2 Solution Method

7.2.1 Second-order Cone Problem Statement

The objective function and all the constraints except the keep-out zone can now be written as a standard SOCP. Here the objective function includes the constant coefficient, \(\Delta t^{-1}\omega^{-2}\), where \(\omega\) is the mean-motion of the chief’s orbit given by Eq. 5.5. This serves to numerically conditions the problem for the convex optimization solver, and improves convergence properties.

\[
\text{Minimize } J = \frac{1}{\omega^2} \sum_{i=1}^{N} \eta_i
\]

Subject to

\[
\|u_i\| \leq \eta_i
\]

\[
\eta_i \leq u_{\text{max}}
\]

\[
\|M_r x_i\| \cos(\alpha) \leq \hat{r}_i^T M_r x_i
\]

\[
x_1 = \begin{bmatrix} r_1 \\ v_1 \end{bmatrix} \text{ is given}
\]

\[
x_{N+1} = \begin{bmatrix} r_{N+1} \\ v_{N+1} \end{bmatrix} \text{ is specified}
\]

\[
x_{i+1} = \Phi(t_{i+1}, t_i) x_i + B_d u_i
\]

for \(i = 1, \ldots, N\), and \(M_r = \begin{bmatrix} I_{3\times3} & 0_{3\times3} \end{bmatrix}\). Eq. 7.21 is a constraint on the magnitude of the thrust acceleration, Eq. 7.22 requires the relative trajectory to fall within a specified cone, Eqs. 7.23-7.24 are the constraints on the initial and final relative positions and velocity, and Eq. 9.44 enforces the dynamics constraints.

This defines an SOCP with \(n = 10N-6\) optimization variables,

\[
\eta_i, \quad i = 1, \ldots, N
\]
\[ x_i, \quad i = 2, \ldots, N \tag{7.27} \]
\[ u_i, \quad i = 1, \ldots, N \tag{7.28} \]

and \( m = 3N \) inequality constraints, \( p = 6N \) equality constraints, for a total of \( l = m + p = 9N \) constraints.

### 7.2.2 Solution to the Spherical Keep-out Zone

The keep-out zone constraint in Eq. 7.15 is non-convex and does not fit the SOCP model. One method used to convexify the sphere is to find a solution using statically-attached planes which are placed at boundary points around the sphere. A plane, being a convex cone with a full angle, is a convex constraint that can be implemented to maintain a convex solution set for position. The planar constraint is setup to insure that the optimal trajectory stays behind the plane (outside the sphere) at the time and point of interest.

A solution for the placement of the boundary planes is to first solve the unconstrained problem, i.e., without the keep-out zone. Fig. 7.1 shows an optimal unconstrained trajectory from a position on the positive v-bar \((+y\text{-axis}, x = 0)\) ahead of the target to a position on the negative v-bar \((-y\text{-axis}, x = 0)\) behind the target.

![Fig. 7.1: Optimal transfer without keep-out zone constraint](image-url)
The unconstrained optimal trajectory violates the spherical keep-out zone at the points shown in red. These points are used as a means for boundary plane placement. Three such planes are shown for three of the fifteen points that intersect the sphere. At each point, \( \mathbf{r}_i \), a plane is attached to the surface of the sphere. The additional planar constraint can be written in convex form, as

\[
\mathbf{r}_{\min} \leq \mathbf{r}_i^T \hat{\mathbf{i}}_i^* \tag{7.29}
\]

where \( \hat{\mathbf{i}}_i^* = \mathbf{r}_i^*/\|\mathbf{r}_i^*\| \) are the unit vectors of the positions that violate the sphere constraint (and \( \mathbf{r}_i \) are the optimization parameters). This is now a convex constraint taking the form of Eq. 7.14 and 7.17, and can be implemented in the SOCP to approximate the sphere.

Using the method of sequential convex programming [20] and the boundary plane setup, an optimal path around the keep-out zone can be determined in an iterative manner. This involves solving the SOCP recursively until the approximate optimal path from the previous iteration converges to the path of the most recent iteration. The steps for this process are as follows:

1. Generate the optimal solution to the unconstrained problem, i.e., without the keep-out zone.
2. Set up the boundary plane constraints in Eq. 7.29 using the unconstrained solution.
3. Solve the SOCP with the plane constraints in Eq. 7.29 to determine an approximate optimal path around the keep-out zone.
4. Set up new boundary plane constraints using the solution from the SOCP.
5. Return to step 3 and resolve the problem until the path converges to within the desired tolerance.

Convergence of the SCP method is determined by the difference between the trajectories generated by the previous iteration and the current iteration [22]. When the difference is less than the desired tolerance, the algorithm exits and returns the solution from the last iteration.
7.3 Planner Results

Several RPO scenarios were designed to test the trajectory planning algorithm’s capability to generate optimal trajectories relative to an LVLH frame, including an initial approach, a way-point following and inspection mission with and without a spherical keep-out zone, a final v-bar (y-axis) approach, and a final r-bar (x-axis) approach. Small relative velocity perturbations were included at the start points of each trajectory. The velocity perturbations were normally distributed with zero-mean and a standard deviation of 10 cm/s, $v_p \sim \mathcal{N}(0 \text{m/s}, 0.1^2 \text{m}^2/\text{s}^2)$. All scenarios shown below are for low Earth orbit (LEO) and converged in a few seconds on an ordinary desktop computer. All of the planner results shown in this section are based on the simple linear HCW dynamics model. Later, in Chapters 12-15, the planner will be implemented in a full nonlinear simulation.

7.3.1 Initial Approach with Maximum Thrust Constraint

The initial position of the chaser relative to the target is approximately 10 km on the v-bar and are shown in Table 7.1. In each case, the final desired relative position and velocity are 200 m on the v-bar and 0.0 cm/s/axis, respectively. Two approach scenarios were selected, one with a three hour transfer and one with a 6 hour transfer. For each of these scenarios, the number of discretization points was $N = 108$. The optimal trajectories based on CW dynamics for the 5 cases are shown in Fig. 7.2. The $\Delta v$ for each case is shown in Fig. 7.3. In general, the optimal solutions require three or more maneuvers. These five cases were also solved with the additional constraint on maximum thrust acceleration over each time step. For a maximum acceleration of 5 mm/s$^2$, the optimal control solution is shown in Fig. 7.4.

Similarly, the six hour transfer trajectories are shown in Fig. 7.5, and the corresponding optimal $\Delta v$/control is shown in Fig. 7.6. Once again, the optimal solutions require three or more maneuvers.
Table 7.1: Initial approach conditions

<table>
<thead>
<tr>
<th>Case</th>
<th>( r(t_0) [x,y,z] ) (km)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1</td>
<td>[0, 10, 0]</td>
</tr>
<tr>
<td>Case 2</td>
<td>[1, 10, 0]</td>
</tr>
<tr>
<td>Case 3</td>
<td>[-1, 10, 0]</td>
</tr>
<tr>
<td>Case 4</td>
<td>[0, 10, 1]</td>
</tr>
<tr>
<td>Case 5</td>
<td>[0, 10, -1]</td>
</tr>
</tbody>
</table>

Fig. 7.2: Three hour approach from 10 km to 200 m along v-bar direction with ±1 km altitude variations and ±1 km cross-track variations.
Fig. 7.3: $\Delta v$/control magnitude history for three hour transfer

Fig. 7.4: $\Delta v$/control magnitude history for three hour transfer with maximum acceleration level of 5 mm/s$^2$
Fig. 7.5: Six hour approach from 10 km to 200 m along v-bar direction with ± 1 km altitude variations and ± 1 km cross-track variations

Fig. 7.6: $\Delta v$/control magnitude history for six hour transfer

### 7.3.2 Way Point Following/Inspection

In this scenario, the chaser vehicle is commanded to perform an inspection of the target vehicle by maneuvering to a set of way-points (A, B, C, D, and E) with a 30 minute transfer between each way-point. The positions of the way-points are shown in Table 7.2. For each of these scenarios, the number of discretization points was $N = 100$. The chaser satellite
maneuvers from point A to B (Case 1), B to C (Case 2), C to D (Case 3), then D to E (Case 4), in succession, as shown in Fig. 7.7. The $\Delta v$ control history for each of these four cases consists of two constant accelerations, one at the beginning of each transfer, and one at the end of each transfer. Clearly these maneuvers could have been more easily computed using ordinary CW-targeting, however, three-burn maneuvers are possible in some way-point-following cases and must be considered.

Table 7.2: Inspection Way-points

<table>
<thead>
<tr>
<th>LVLH Way-point</th>
<th>$r(t_0)$ [x, y, z] (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>[0, 200, 0]</td>
</tr>
<tr>
<td>B</td>
<td>[0, 0, 100]</td>
</tr>
<tr>
<td>C</td>
<td>[100, 0, 0]</td>
</tr>
<tr>
<td>D</td>
<td>[0, -100, 0]</td>
</tr>
<tr>
<td>E</td>
<td>[0, 100, 0]</td>
</tr>
</tbody>
</table>

Fig. 7.7: Way-point following/inspection with 30 minute transfers between each way-point.
7.3.3 Way Point Following/Inspection with Keep-out Zone

In this scenario, the chaser vehicle is commanded to follow the same way-points as above with the addition of a keep-out zone, a sphere with radius 80 m. For each of these scenarios, the number of discretization points was \( N = 100 \). Since the keep-out zone is non-convex, the method of successive approximations is required.

The optimal trajectories are shown in Fig. 7.8, and the optimal \( \Delta v / \)control for these cases is shown in Fig. 7.9. The only trajectory affected by the keep-out zone is Case 4, the transfer from point D to E. In this case, a segment of the trajectory rides along the keep-out zone requiring continuous maneuvering as can be seen from time \( t = 600 \) to \( t = 1000 \) sec in Fig. 7.9. The total \( \Delta v \) for this case is also slightly greater than the scenario without the keep-out zone. Notice that this control solution is step-like, required alternating high/low constant thrust acceleration over these time periods in order to ride the boundary of the constraint.

Fig. 7.8: Way-point following/inspection with 30 minute transfers between each way-point and a 80 m spherical keep-out zone
7.3.4 Final Approach with Approach Corridor

The first two cases presented here show the resulting trajectories and control solutions for a 10 minute approach along the v-bar from 100 meters away. The approach is investigated as a traditional approach strictly along the v-bar, and as an approach which implements a 10 degree approach corridor. The number of discretization points in these cases was $N = 100$. The trajectories are shown in Fig. 7.10, and the corresponding control accelerations and total $\Delta v$ are shown in Fig. 7.11. The resulting $\Delta v$ savings calculated as a percent difference by implementing the cone, when compared to the traditional v-bar approach, is approximately 27%.

Next, two final approach scenarios are investigated, a v-bar approach, and an r-bar approach. Both approaches include a 10 deg approach cone/corridor constraint. The initial relative positions are approximately 100 meters on the v-bar and r-bar, respectively, and are shown in Table 7.3. The final desired relative position is 5 meters on the v-bar and r-bar, respectively. The final desired relative velocity for all cases is 0.0 meters/s/axis. The transfer time in all cases is 10 minutes, and the number of discretization points was $N = 60$.

The optimal v-bar final approach trajectories are shown in Fig. 7.12, and the associated
optimal \( \Delta v/\text{control} \) is shown in Fig. 7.13. It can be seen that the optimal control requires a large maneuver at the beginning and end of the approach while the chaser is well inside the approach cone constraint. However, a closer look at the \( \Delta v/\text{control} \) in Fig. 7.14 shows that the optimal solution also requires small continuous maneuvers between 200 sec and 600 sec. It is during this time period that the optimal solution rides along the approach cone constraint. An example of an implementation of the drift without control constraint is shown in Fig. 7.18, while on the boundary of the cone. With this new constraint, the cone boundary is slightly relaxed, allowing for a more continuous control solution.

The optimal \( r \)-bar final approach trajectories are shown in Fig. 7.15, and the associated optimal \( \Delta v/\text{control} \) is shown Fig. 7.16. Once again, the optimal control requires a large maneuver at the beginning and end of the approach while the chaser is well inside the approach cone constraint. A closer look at the \( \Delta v/\text{control} \) in Fig. 7.17 shows once again that the optimal solution rides along the approach cone constraint and requires small continuous maneuvers between times 200 sec and 600 sec. Here the constant control over each time period is again step-like, requiring high/low acceleration to ride on the cone boundary.

Fig. 7.10: 10 minute \( v \)-bar approach and cone approach trajectories with a 10 degree approach cone constraint
Fig. 7.11: \( \Delta v/\text{control} \) magnitude history for v-bar approach and cone approach

Table 7.3: Final approach initial conditions

<table>
<thead>
<tr>
<th>Approach Type</th>
<th>( r(t_0) ) [x, y, z] (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>V-bar Approach</td>
<td></td>
</tr>
<tr>
<td>Case 1</td>
<td>[0, 100, 0]</td>
</tr>
<tr>
<td>Case 2</td>
<td>[-10, 100, 0]</td>
</tr>
<tr>
<td>Case 3</td>
<td>[-10, 100, 0]</td>
</tr>
<tr>
<td>Case 4</td>
<td>[0, 100, 10]</td>
</tr>
<tr>
<td>Case 5</td>
<td>[0, 100, -10]</td>
</tr>
<tr>
<td>R-bar Approach</td>
<td></td>
</tr>
<tr>
<td>Case 1</td>
<td>[100, 0, 0]</td>
</tr>
<tr>
<td>Case 2</td>
<td>[100, 10, 0]</td>
</tr>
<tr>
<td>Case 3</td>
<td>[100, -10, 0]</td>
</tr>
<tr>
<td>Case 4</td>
<td>[100, 0, 10]</td>
</tr>
<tr>
<td>Case 5</td>
<td>[100, 0, -10]</td>
</tr>
</tbody>
</table>
Fig. 7.12: 10 minute v-bar approach trajectory with a 10 degree approach cone constraint, ±10 m altitude variations, ±10 m cross-track variations

Fig. 7.13: Δv/control magnitude history for v-bar final approach
Fig. 7.14: Zoomed in $\Delta v$/control magnitude history for $v$-bar final approach, while on cone boundary

Fig. 7.15: 10 minute $r$-bar approach trajectory with a 10 degree approach cone constraint, $\pm 10$ m altitude variations, $\pm 10$ m cross-track variations
Fig. 7.16: $\Delta v$/control magnitude history for r-bar final approach

Fig. 7.17: Zoomed in $\Delta v$/control magnitude history for r-bar final approach, while on cone boundary
7.4 Summary

In this chapter, the RPO trajectory planning problem is first formulated under the HCW dynamics model. This includes the linear equality constraints for the dynamics, and the inequality constraints for the slack variables, maximum control acceleration, and approach corridor. Spherical keep-out zone constraints are also considered, which are inherently nonconvex. A linearized approximation technique and SCP methods are required to determine the optimal path around the sphere.

Several planned trajectory results are also shown. These include the trajectory and optimal control history plots for the following scenarios: initial approach, way-point following (with or without a spherical keep-out zone), and final approach with an approach corridor. The ‘drift without control’ constraints are shown to smooth the optimal control solution while on the boundary of the path constraints.
CHAPTER 8

RPO PLANNING RELATIVE TO ROTATING SPACECRAFT FRAME

In some RPO scenarios, the optimal trajectory planning problem is best formulated in a rotating body-fixed spacecraft frame, centered on the chief spacecraft. This may be especially true for proximity operations near a controlled or uncontrolled spinning spacecraft. The objective in this case is still to minimize propellant subject to approach corridor constraints and keep-out zones, but the CW equations are not ideal for flight regime. Instead, a new problem formulation based on the linearized dynamics equations with respect to a rotating chief’s body-fixed frame is introduced.

This chapter presents the problem of RPO trajectory planning with respect to a rotating body-fixed spacecraft frame. The dynamics which govern the relative motion in this frame are the Generalized HCW Equations, presented in Chapter 6. In these problems, trajectory constraints include maximum thrust acceleration, approach corridors, spherical keep-out zones and stand-off planes. All constraints are convex except for the spherical keep-out zone, which is handled using the method of sequential convex programming as described in Chapter 7. The stand-off plane is a new constraint to ensure a minimum distance of the deputy from the chief, and is implemented in the rotating body-fixed frame.

The problem formulation for this RPO planning scenario is stated. Much of the resulting analysis from Chapter 7 remains, with the primary exception of the different dynamics model for the rotating body-fixed frame. Next, the objective function, trajectory constraints and solution method are outlined. Several planner results and example trajectories in a rotating body-fixed frame are presented. The dynamics model and planner results are summarized in the final section.
8.1 Problem Formulation

8.1.1 Relative Motion Dynamics with Respect to Rotating Spacecraft Frame

The relative orbital motion as observed from a spinning body-fixed frame is governed by a set of linear-time-varying (LTV) differential equations \[139\]

\[
\dot{x} = A(t)x + Bu \tag{8.1}
\]

where \(x = [r^Tv^T]^T\) is the relative position and velocity with respect to a body-fixed frame, \(u = a_T\) is the thrust acceleration, and \(A\) and \(B\) matrices are given by

\[
A(t) = \begin{bmatrix}
0_{3\times3} & I_{3\times3} \\
-\Omega^2 \left( I_{3\times3} - \left[ T^B_{B_0}(\omega, t) T^B_{I_0}(t) \right] \left[ T^B_{B_0}(\omega, t) T^B_{I_0}(t) \right]^T \right) & -[\omega \times]^2 - 2[\omega \times]
\end{bmatrix}
\tag{8.2}
\]

\[
B = \begin{bmatrix}
0_{3\times3} \\
I_{3\times3}
\end{bmatrix}
\]

The superscripts \(B\), \(B_0\), and \(I\) refer to the body frame, body frame at time zero, and the inertial frame, respectively. To simplify this analysis, we assume that the angular velocity of the target, \(\omega\), is aligned with the target body \(z\)-axis, \(i^B_z\). Its magnitude is given by \(\omega\). The initial unit target position vector is given by \(i^T_I(t_0)\). The orbital angular rate is given by \(\Omega\), and without loss of generality it is assumed that the initial position in the target’s orbit is aligned with the \(x - y\) equatorial plane of the Earth, or \(i^I_\Omega = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T\). The rotation matrix, \(T^B_{I_0}\), is a constant transformation matrix from the inertial frame to the initial target body frame, and the rotation matrices \(T^B_{B_0}(\omega, t)\) and \(R(\Omega, t)\) are defined by

\[
T^B_{B_0}(\omega^I, t) = \cos \theta_\omega I_{3\times3} + \sin \theta_\omega \left[i^B_z \times \right] + (1 - \cos \theta_\omega) i^B_z \otimes i^B_z
\tag{8.3}
\]

\[
R(\Omega^I, t) = \cos \theta_\Omega I_{3\times3} + \sin \theta_\Omega \left[i^I_\Omega \times \right] + (1 - \cos \theta_\Omega) i^I_\Omega \otimes i^I_\Omega
\tag{8.4}
\]
where, $\mathbf{v}_z^B = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$ is the unit rotation vector of the body-fixed frame in the z-direction, and the rotation angles for each matrix are $\theta_\omega = \omega t$, and $\theta_\Omega = \Omega t$. Eqs. 8.1-8.4 describe the orbital relative motion as viewed from a frame fixed to a controlled or uncontrolled spinning spacecraft.

If the dynamics are discretized into $N$ equal time-steps $dt$, and the thrust acceleration is assumed to be constant over each time-step, an approximate first-order (Euler) solution to Eq. 8.1 is given by

$$x_{i+1} = x_i + \dot{x}_i dt = x_i + [A(t_i)x_i + Bu_i] dt \quad (8.5)$$

or

$$x_{i+1} = [I_{6\times6} + A(t_i)dt] x_i + Bu_i dt \quad (8.6)$$

When a more accurate second-order solution is desired, a trapezoidal approach can be taken

$$x_{i+1} = x_i + \left[\frac{\dot{x}_{i+1} + \dot{x}_i}{2}\right] dt = x_i + \left[\frac{A(t_{i+1})x_{i+1} + A(t_i)x_i}{2} + u_i\right] dt \quad (8.7)$$

or

$$\left[I - \frac{A(t_{i+1})dt}{2}\right] x_{i+1} = \left[I + \frac{A(t_i)dt}{2}\right] x_i + Bu_i dt \quad (8.8)$$

For this analysis, we use the average state transition matrix approach to solve the dynamics in the body-fixed frame. This method is formulated by interpolating the discretized time-varying $A(t)$ matrix by taking the midpoint value between points $i$ and $i + 1$. Then, the average state transition matrix is calculated and used in the dynamics constraints. This can be written as

$$\overline{A}_i = \frac{A(t_i) + A(t_{i+1})}{2} \quad (8.9)$$

$$\overline{\Phi}_i(t_{i+1}, t_i) = e^{\overline{A}_i dt} \quad (8.10)$$
The average discretized $B_d$ matrix is then calculated as

$$\overline{B}_{d_i} = \int_{t_i}^{t_{i+1}} \Phi_i(t_{i+1}, \tau) B \, d\tau$$  \quad (8.11)

To simplify numerical computation required for these integrations, expand the integral out to four terms, and integrate analytically. This leads to

$$\overline{B}_{d_i} \approx \int_{t_i}^{t_{i+1}} \left[ I + \overline{A}_i(t_{i+1} - \tau) + \frac{\overline{A}_i^2(t_{i+1} - \tau)^2}{2} + \frac{\overline{A}_i^3(t_{i+1} - \tau)^3}{6} \right] d\tau B$$  \quad (8.12)

$$\overline{B}_{d_i} \approx \int_{t_i}^{t_{i+1}} \left[ I(t_{i+1} - t_i) + \frac{\overline{A}_i(t_{i+1} - t_i)^2}{2} + \frac{\overline{A}_i^2(t_{i+1} - t_i)^3}{6} + \frac{\overline{A}_i^3(t_{i+1} - t_i)^4}{24} \right] B$$  \quad (8.13)

The dynamics using the average state transition matrix now take the form

$$\mathbf{x}_{i+1} = \Phi_i(t_{i+1}, t_i) \mathbf{x}_i + \overline{B}_{d_i} \mathbf{u}_i$$  \quad (8.14)

Equations 8.6, 8.8, and 8.14 represent linear equality constraints of increasing fidelity that can be directly implemented in an SOCP. In the work that follows in this chapter, the linear equality constraints in Eq. 8.14 will be utilized to plan optimal trajectories in the chief spacecraft’s rotating body-fixed frame.

### 8.1.2 Objective Function

The objective function for the rotating body-fixed frame trajectory planning problem is the same as was derived for the HCW trajectory planning problem presented in Chapter 7. This involves minimizing the total amount of thrust acceleration over the duration of the trajectory, where the thrust acceleration is constant over each time step. This is shown in Eq. 7.7, and is rewritten here as

$$J = \sum_{i=1}^{N} ||\mathbf{a}_T|| = \sum_{i=1}^{N} ||\mathbf{u}_i||$$  \quad (8.15)
Once again, in order to accommodate the SOCP problem, slack variables are introduced, as in Eq. 7.8-7.10, so that the objective is written as

\[ J = \Delta t \sum_{i=1}^{N} \eta_i \] \hspace{1cm} (8.16)

### 8.1.3 Constraints

All constraints for the trajectory planning problem in a rotating body-fixed frame are similar to the problem formulated under the HCW model dynamics (as shown in Chapter 7). The constraints included here are the fixed final time, boundary constraints, maximum thrust acceleration, conic approach corridor, spherical keep-out zone, and a planar minimum distance constraint (a.k.a. "stand-off plane").

The linear equality constraints for the boundary conditions are similar to the formulation shown previously in Eq. 7.11. The main difference is that now these constraints are coordinatized in the chief’s rotating body-fixed frame. At the initial and final times \((t_1 \text{ and } t_{N+1})\), respectively, these are written as

\[
\begin{bmatrix}
  r_1 \\
  v_1
\end{bmatrix} = \text{given}, \quad \begin{bmatrix}
  r_{N+1} \\
  v_{N+1}
\end{bmatrix} = \text{specified},
\] \hspace{1cm} (8.17)

Second-order conic constraints for the maximum acceleration and the approach corridor are shown in Eq. 7.12 and 7.13, respectively. These equations are repeated here for convenience. For max acceleration,

\[ ||u_i|| \leq u_{max} \text{ for } i = 1, \ldots, N \] \hspace{1cm} (8.18)

which, once relaxed into SOCP form, results in

\[ ||u_i|| \leq \eta_i, \quad 0 \leq \eta_i \leq u_{max}, \text{ for } i = 1, \ldots, N \] \hspace{1cm} (8.19)
and for the conic approach corridor

$$||r_i|| \cos(\alpha) \leq \hat{i}_c^T r_i, \text{ for } i = 2, \ldots, N$$  \hspace{1cm} (8.20)

The final constraint considered here may be used to replace the spherical keep-out zone in scenarios where the approach corridor is also implemented. This is termed the minimum distance constraint. The inequality for this constraint was shown previously in Eq. 7.14, and is rewritten here in discretized form, where $\hat{i}_c$ is constant, as

$$d_{\min} \leq \hat{i}_p^T r_i, \text{ for } i = 2, \ldots N$$  \hspace{1cm} (8.21)

The idea is that by enforcing the cone approach corridor and the minimum distance constraints at all times along the trajectory, the spherical keep-out zone is inherently satisfied. Note that for the cone approach constraint and the minimum distance constraint, it is assumed that the initial and final points ($r_1$ and $r_{N+1}$) satisfy these constraints, and are thus not included.

The ‘drift without control’ constraints are also included in the rotating body-fixed frame analysis. These are exactly as defined in Chapter 7, and are shown by Eqs. 7.16-7.18.

8.2 Solution Method

8.2.1 Second-order Cone Problem Statement

The objective function and all the constraints for the trajectory planning problem in a rotating body-fixed frame can now be written as a standard SOCP. Here the objective function is multiplied by the constant coefficient, $\Delta t^{-1} \omega^{-2}$, where $\omega$ is the mean-motion of the chief’s orbit given by Eq. 5.5. This serves to numerically condition the problem for the convex optimization solver, and improves convergence properties.

$$\text{Minimize } J = \frac{1}{\omega^2} \sum_{i=1}^{N} \eta_i$$  \hspace{1cm} (8.22)
Subject to

\[ ||u_i|| \leq \eta_i \] \hspace{1cm} (8.23)

\[ \eta_i \leq u_{\text{max}} \] \hspace{1cm} (8.24)

\[ ||M_r x_i|| \cos(\alpha) \leq \hat{I}_c^T M_r x_i \] \hspace{1cm} (8.25)

\[ d_{\text{min}} \leq \hat{I}_p^T M_r x_i \] \hspace{1cm} (8.26)

\[ x_1 = \begin{bmatrix} r_1 \\ v_1 \end{bmatrix} \text{ = given} \] \hspace{1cm} (8.27)

\[ x_{N+1} = \begin{bmatrix} r_{N+1} \\ v_{N+1} \end{bmatrix} \text{ = specified} \] \hspace{1cm} (8.28)

\[ x_{i+1} = \Phi(t_{i+1}, t_i) x_i + \bar{B}_d u_i \] \hspace{1cm} (8.29)

for \( i = 1, \ldots, N \), and \( M_r = \begin{bmatrix} I_{3 \times 3} & 0_{3 \times 3} \end{bmatrix} \). Eq. 8.24 is a constraint on the magnitude of the thrust acceleration, Eq. 8.25 requires the trajectory to fall within a specified cone, Eq. 8.26 ensures the minimum distance constraint, Eqs. 8.27-8.28 are the constraints on the initial and final relative positions and velocity, and Eq. 8.29 enforces the dynamics constraints for the rotating body-fixed frame.

This defines an SOCP with \( n = 10N-6 \) optimization variables,

\[ \eta_i, \ i = 1, \ldots, N \] \hspace{1cm} (8.30)

\[ x_i, \ i = 2, \ldots, N \] \hspace{1cm} (8.31)

\[ u_i, \ i = 1, \ldots, N \] \hspace{1cm} (8.32)

and \( m = 3N \) inequality constraints, \( p = 6N \) equality constraints, for a total of \( l = m + p = 9N \) constraints.

As in Chapter 7, the cases that include the keep-out zone are handled using SCP
methods. The equation for how the spherical keep-out zone is implemented on each SCP iteration is shown in Eq. 7.29, and the steps for determining the optimal path around the sphere directly follow the outline in Chapter 7.

8.3 Planner Results

In order to test the body-fixed time varying equations of relative motion, a few scenarios were set up to show some example trajectories. In this Chapter, three main scenarios are examined. These include waypoint following without a spherical KOZ, waypoint following with the spherical KOZ, and final approach with an approach corridor. All trajectory solutions for these scenarios were generated based on the rotating body-fixed dynamics model. Several RPO cases were designed to test the trajectory planning algorithm’s capability for each scenario. Each result produces an optimal trajectory (for the approximated dynamics model) relative to a rotating chief body-fixed frame with a known constant spin rate. Small relative velocity perturbations were included at the starting point of each trajectory. The velocity perturbations were normally distributed with zero-mean and a standard deviation of 10 centimeters per second, $v_p \sim N(0 \text{m/s}, 0.1^2 \text{m}^2/\text{s}^2)$.

The scenarios and cases are outlined in the following sections. All results shown below are for low Earth orbit (LEO), and converged in a few seconds on an ordinary desktop computer. The constraints – final time, initial and final conditions, spherical keep-out zones, approach cones/corridors, and minimum distance – are explicitly modeled in the rotating body-fixed frame. For more examples of optimal trajectory planning for orbital rendezvous and proximity operations in a body-fixed frame, please see the results of the associated paper by Geller, et. al. [140].

The final results section in this chapter presents brief results from the inertial targeting algorithm developed in Chapter 6. Several important traits of the rotating body-fixed dynamics model are distinguished and compared to a frame that is rotating without the effects of gravity. A brief summary concludes the chapter.
8.3.1 Inspection/Waypoint Following

The following cases shown in Fig. 8.1-8.6 will show an example of several body-frame maneuvers done in succession. The deputy is instructed to go from the initial point at 10 meters on the v-bar to 10 meters on the body x-axis, then to 10 meters on the body y-axis, and finally, to 10 meters on the body z-axis. At each final point, the deputy has zero relative velocity in the rotating chief’s body-fixed frame. Each of the three maneuvers has a specified final time of 10 minutes, for a total of 30 minutes. In the first case, the chief is rotating at 1 degree per second about the inertial z-axis. The results are shown in Fig. 8.1 and 8.2.

![Diagram](image)

- V-bar to 10m x-axis DV = 0.20534 m/s
- 10m x-axis to 10m y-axis DV = 0.36806 m/s
- 10m y-axis to 10m z-axis DV = 0.20144 m/s

Fig. 8.1: Case 1, three successive transfers around target from v-bar to body x-axis to body y-axis to body z-axis. Body rotation rate is 1 deg/sec about the inertial z-axis. LVLH frame trajectory and total delta-v.
In the second case, the deputy performs the same maneuvers, but now the chief is rotating at 0.2 degrees per second about the inertial y-axis. The results are shown in Fig. 8.3 and 8.4.

<table>
<thead>
<tr>
<th></th>
<th>V-bar to 10m x-axis DV</th>
<th>10m x-axis to 10m y-axis DV</th>
<th>10m y-axis to 10m z-axis DV</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.062815 m/s</td>
<td>0.12316 m/s</td>
<td>0.066254 m/s</td>
</tr>
</tbody>
</table>

Fig. 8.2: Case 1, three successive transfers around target from v-bar to body x-axis to body y-axis to body z-axis. Body rotation rate is 1 deg/sec about the inertial z-axis. Left: Body-fixed frame. Right: Inertial frame.

Fig. 8.3: Case 2, three successive transfers around target from v-bar to body x-axis to body y-axis to body z-axis. Body rotation rate is 0.2 deg/sec about the inertial y-axis. LVLH frame trajectory and total delta-v.
Fig. 8.4: Case 2, three successive transfers around target from v-bar to body x-axis to body y-axis to body z-axis. Body rotation rate is 0.2 deg/sec about the inertial y-axis. Left: Body-fixed frame. Right: Inertial frame.

For the final case, again, the deputy performs the same maneuvers, and now the chief is rotating at 0.1 degrees per second about the inertial x-axis. The results are shown in Fig. 8.5 and 8.6.

<table>
<thead>
<tr>
<th>Transfer</th>
<th>Delta-V (m/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>V-bar to 10m x-axis</td>
<td>0.039915</td>
</tr>
<tr>
<td>10m x-axis to 10m y-axis</td>
<td>0.065739</td>
</tr>
<tr>
<td>10m y-axis to 10m z-axis</td>
<td>0.053364</td>
</tr>
</tbody>
</table>

Fig. 8.5: Case 3, three successive transfers around target from v-bar to body x-axis to body y-axis to body z-axis. Body rotation rate is 0.1 deg/sec about the inertial x-axis. LVLH frame trajectory and total delta-v.
In this scenario, the chaser vehicle is commanded to perform an inspection of the target vehicle by maneuvering to a set of target relative waypoints (A, B, C, and D) with a 10 minute transfer between each waypoint. The target-relative positions of the waypoints are shown in Table 8.1. For each of these scenarios, the number of discretization points was $N = 60$. The chaser satellite maneuvers from point A to B (Case 1), B to C (Case 2), and C to D (Case 3), in succession, in the target frame, as shown in Fig.8.7. Fig. 8.8 shows the same optimal trajectories in the LVLH frame. The optimal $\Delta v$ control history for the inspection scenarios consists of two constant accelerations, one at the beginning of each transfer and one at the end of each transfer. Again, CW targeting could have been used here to compute the maneuvers, however, there is the possibility of three-burn maneuvers in some cases of spacecraft-relative transfers. These three waypoint-following trajectories were also solved with the constraint on maximum acceleration level. For a max acceleration of 1 cm/s$^2$, the optimal control solution for these cases is shown in Fig. 8.9.
Table 8.1: Inspection Waypoints

<table>
<thead>
<tr>
<th>Target-Relative Waypoint</th>
<th>$r(t_0) [x, y, z]$ (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>[0, 10, 0]</td>
</tr>
<tr>
<td>B</td>
<td>[10, 0, 0]</td>
</tr>
<tr>
<td>C</td>
<td>[0, 10, 0]</td>
</tr>
<tr>
<td>D</td>
<td>[0, 0, 10]</td>
</tr>
</tbody>
</table>

Fig. 8.7: Target-relative waypoint following/inspection with 10 minute transfers between each waypoint
8.3.2 Inspection/Waypoint Following with Keep-out Zone

In this scenario, the chaser vehicle is commanded to follow the same waypoints as above with the addition of a 9 m spherical keep-out zone. For each of these scenarios, the number of discretization points was $N = 100$. The method of successive approximations is employed.
The optimal trajectories in the body-fixed frame are shown in Fig. 8.10. Fig. 8.11 shows the same optimal trajectories in the LVLH frame, and Fig. 8.12 shows the optimal control acceleration history for this inspection scenario. The optimal solutions for all three cases required a maneuver at the beginning and end of the transfer, however, the keep-out zone changed the $\Delta v$/optimal control solutions for Case 1 (A to B) and Case 3 (C to D). Fig. 8.13 shows that small continuous maneuvers are required in these cases between 150 sec and 600 sec, a time period during which the optimal trajectory solution rides along the keep-out zone.

Fig. 8.10: Target-relative waypoint following/inspection with 10 minute transfers between each waypoint and a 9 meter keep-out zone
Fig. 8.11: Target-relative waypoint following/inspection with 10 minutes transfer between each waypoint and a 9 meter keep-out zone as seen from an LVLH frame.

Fig. 8.12: Control acceleration magnitude history target-relative waypoint following/inspection with 9 meter keep-out zone.
8.3.3 Final Approach with Approach Corridor

Each of the following three cases shown in Fig. 8.14-8.16 was done for a half-hour (1800 sec.) transfer. The orbital radius was set to be at 7100 km with an LVLH rotation rate of 0.06046 deg/sec, while the rotation of the body frame was varied. The number of discretization points is $N = 200$.

The first case was run for a body frame rotating at LVLH rate, in the direction of LVLH rotation, and the deputy approached the chief from 100 m on the positive v-bar. The results are shown in Fig. 8.14.
The next case tested was for a target rotating at 0.1 deg/sec about the positive z-axis of the LVLH frame, aligned with the cross-track direction. We have added a convex approach cone constraint into the problem in order to show the application of a path constraint within the body-fixed frame. The total rotation was 180°. All other variables were kept the same. The results are shown in Fig. 8.15.

Fig. 8.15: 100 m v-bar transfer to target with approach constraint, body rotation rate is 0.1 deg/sec. Left: Body fixed-frame. Right: Inertial frame

The next case that was tested was for a body rotation of 0.2 deg/sec about the negative z-axis of LVLH, aligned with the cross-track direction. The total rotation was 360°. The results are shown in Fig. 8.16.

Fig. 8.16: 100 m v-bar transfer to target with approach constraint, body rotation rate is -0.2 deg/sec. Left: Body-fixed frame. Right: Inertial frame

In the next cases, two final approaches are investigated, one along a body-fixed x-axis (case 1), and one along a body-fixed y-axis (case 2). For each of these cases, the number of discretization points was \( N = 72 \). Both approaches include a a 10 deg approach
cone/corridor constraint. The initial position is 10 m on the body-fixed x- and y-axis, respectively, and the final desired relative position is 1 m on the body-fixed x- and y-axes, respectively. A 0.5 m minimum distance requirement is also enforced via a plane constraint at \( x = 0.5 \) m and \( y = 0.5 \) m, respectively. The final desired relative velocity for all cases is 0.0 m/s/axis, and the final time is 12 minutes.

The optimal trajectory solutions are shown in the target frame in Fig. 8.17. It is interesting to note that the optimal trajectory takes the vehicle from the initial 10 m position to the 0.5 m position constraint before continuing on to the final desired position at 1 m. In this case, the minimum propellant solution is to first move up to the 0.5 m position, wait there to conserve propellant, and then continue on to the final desired position. The same optimal trajectories in the LVLH frame are shown in Fig. 8.18. The optimal control acceleration for each case is plotted separately in Fig. 8.19 and Fig. 8.20. Another control solution using the drift without control constraints was implemented here for the body final approach cases. These results show the more continuous controls in Fig. 8.21 and Fig. 8.22.

![Body Frame](image)

**Fig. 8.17:** Target-relative x-axis and y-axis final approach trajectories with a 10 degree approach cone constraint and a 12 min transfer
Fig. 8.18: Target-relative 12 minute x-axis and y-axis final approach trajectories with a 10 degree approach cone constraint as seen from the LVLH frame.

Fig. 8.19: Control acceleration magnitude history for x-axis and y-axis final approach.
Fig. 8.20: Zoomed in control acceleration magnitude history for x-axis and y-axis final approach, while at minimum distance

Fig. 8.21: Control acceleration magnitude history for x-axis and y-axis final approach using ‘drift without control’ constraints
8.4 Inertial Targeting Results

The inertial targeting algorithm was tested by varying the fixed final transfer time as well as the rotation rate of the chief-fixed frame. The inertial targeting algorithm was implemented in an open-loop targeting method, where the chaser targets only once for the next desired relative position. In these cases, the chaser was instructed to go from 10 meters on the x-axis to 10 meters on the y-axis, then to 10 meters on the z-axis, and back to 10 meters on the x-axis, where each relative position is in the chief-fixed frame. The rotation rate of the chief’s frame was set to 0.25 degrees per second about the inertial z-axis, and the fixed final times tested for each of these maneuvers were 5 minutes and 15 minutes. These results are presented in Fig. 8.23 where the solid lines show the optimal trajectories, and the dashed lines are the resulting trajectories from the targeting algorithm. In these cases, the shorter fixed final times for these trajectories produce more accurate targeting results.
Inertial targeting algorithm results, varying the fixed final time for transfer, with a chief rotation rate of 0.25 deg/sec about the inertial z-axis. Left: Final time of 5 minutes for each transfer. Right: Final time of 15 minutes for each transfer.

In the next cases, the final time for the transfer was set at 5 minutes and the rotation rate of the chief-fixed frame was varied. These examples show one simple trajectory from 10 meters on the x-axis to 10 meters on the y-axis of the chief-fixed frame. The results are presented in Fig. 8.24, where the rotation rates that were tested for the chief-fixed frame were 0.01 deg/sec and 0.5 deg/sec about the inertial z-axis. The solid lines show the optimal trajectories, and the dashed lines are the resulting trajectories from the inertial targeting algorithm. These cases show that for higher rotation rates of the chief-fixed frame, the targeting algorithm gives more accurate results for a fixed final time.
Fig. 8.24: Inertial targeting algorithm results, varying the rotation rate of chief-fixed frame, with a final transfer time of 5 minutes. Left: Chief rotation rate of 0.01 deg/sec, where miss distance was 0.735 meters. Right: Chief rotation rate of 0.5 deg/sec, where miss distance was 0.476 meters.

8.5 Summary

In this chapter, the RPO trajectory planning problem is formulated under a rotating body-fixed frame dynamics model. This involves developing the SOCP problem with the rotating body-fixed frame dynamics derived in Chapter 6. This problem has linear equality constraints for the dynamics, and the inequality constraints for the slack variables, maximum control acceleration, approach corridor, and minimum distance. Spherical keep-out zone constraints are also considered, which are inherently nonconvex. A linearized approximation technique and SCP methods are required to determine the optimal path around the sphere. In the final approach scenario, an approach corridor and minimum distance constraint results in scenarios where the spherical keep-out zone is inherently satisfied.

Several planned trajectory results are also shown. These include the trajectory and optimal control history plots for the following scenarios: initial approach, way-point following (with or without a spherical keep-out zone), and final approach with an approach corridor and minimum distance constraint. The ‘drift without control’ constraints are also analyzed here, and show much smoother optimal control solution while on the boundary of the path constraints.
The inertial targeting algorithm examples show the effects of the rotation rate of the chief's body-fixed frame under the inertial assumptions. When the rotation rate is much greater than the mean motion of the chief's orbit and the final time is relatively small, then the inertial targeting algorithm works very well. However, when the rotation rate of the chief is low and/or the final time is large, then targeting errors begin to grow. This inertial targeting algorithm may be implemented in closed-loop fashion to accurately execute short-duration transfers in a chief's body-fixed frame rotating at a relatively high rate (compared to the chief's mean orbital motion).
CHAPTER 9
RPO PLANNING FOR SAFETY OF FLIGHT

Recent developments in safety of flight concepts produce relative trajectories for RPO that can reduce the probability of accidental collisions with resident space objects (RSOs). Safety of flight concepts typically incorporate constraints such as approach corridors, spherical keep-out zones, and passive abort safety ellipses [6, 22, 87]. These constraints produce trajectories that fall into two different categories: actively safe trajectories, and passively safe trajectories. Actively safe trajectories generally require maneuvers to maintain safety and do not ensure a passive abort capability in the event of passive failures such as computer reboot, power loss, or a suspension of normal activities due to mission/vehicle anomalies [141]. Passively safe trajectories ensure a passive abort capability at all times over the duration of the trajectory [1, 141]. The focus of this analysis is on the formulation of passively safe trajectories in terms of relative orbital elements (ROEs) for a chaser vehicle in close proximity to a target vehicle’s circular reference orbit.

The method of sequential convex programming (SCP) is a traditional approach that uses convex optimization to solve inherently nonconvex problems, and is used extensively in this chapter [20, 21, 129]. This method formulates the nonconvex constraints as linearized approximations within a ‘trust region’ of the nonconvex set. Furthermore, sequential convex optimization has been successfully implemented in RPO trajectory planning algorithms that include nonlinear, nonconvex dynamics, and nonconvex spherical keep-out zones (KOZ) [22, 23, 142]. Optimal trajectory solutions with nonconvex KOZs produce actively safe trajectories that prevent accidental collisions only when the chaser vehicle is active, functioning properly, and capable of executing maneuvers [141]. A KOZ does not prevent accidental collisions when a passive failure occurs and maneuvers are undesirable or not possible.

Several approaches to the relative orbital ellipse transfer problem have been developed,
including control formulations using the traditional relative orbital elements [76, 77], combinations of Cartesian and Orbital element feedback laws [91], and feedback control using mean orbital elements for spacecraft formations [90]. It has also been shown that optimal transfers between relative ellipses results in a sequence of optimal impulsive burn-coast-burn arcs [143]. The approach taken in this chapter is to ensure that each phase of the optimal trajectory between relative ellipses is passively safe, by enforcing constraints on the ROEs. This involves formulating optimal trajectories that consist of relative safety ellipses which are guaranteed to have a passive abort capability at all times.

It is very important to note that the term relative ellipse is intended be a generic term that applies to all relative orbital motion trajectories: flyby trajectories, stationary trajectories (v-bar station-keeping), stationary trajectories that that circumnavigate the target, stationary trajectories offset from the target, and traveling ellipses that move toward or away from the target (including traditional "hopping" relative motion trajectories). All of these trajectories can be modeled as relative ellipses with different sizes and centers, however, only a subset can be considered relative safety ellipses.

The safety ellipses have a guaranteed passive abort capability and are robust to uncertainties in relative position and velocity, as well as the effects of perturbations such as of J2 and drag [79, 143]. In this manner, an end-to-end optimal trajectory is designed such that the chaser is always on a collision-free trajectory if a system problem occurs. Furthermore, the use of ROEs provides an intuitive and guaranteed approach to ensure a safe trajectory, rather than having to check at each instant of time if a collision is possible.

As will be shown, passive safety of flight constraints may not be needed in all cases. Often, a problem may be designed with sufficient initial and final conditions and transfer times such that the solution maintains inherent passive safety of flight. In these cases, once the optimal trajectory is known, it can be evaluated to determine whether or not it is inherently safe. If it is, then it provides sufficient passive safety, however if it is not, then safety of flight constraints should be imposed on the next iteration of the trajectory planning problem. The approach taken in this chapter does not explicitly address the problem of
passive safety in the event of an under-burn or an uncompleted maneuver. However, it
will be seen that many of the cases investigated do maintain safety after an uncompleted
burn, and the cases that do not maintain safety can be remedied by properly redesigning
the mission.

The work presented here is also based on a fixed maximum final time. It is assumed
that mission operational constraints such as lighting, power, communications, tracking, etc.,
will dictate a maximum final transfer time. The algorithms presented here are based only on
the HCW equations, however, the key element of the dynamics model (the state-transition
matrix), can be updated to include target orbit eccentricity and environment perturbations
such $J_2$, drag, and solar radiation pressure $[4, 77, 79, 143]$.

9.1 Basic Problem Formulation

The core of this problem formulation is to determine an optimal, minimum $\Delta v$ tra-
jectory to transfer from an initial relative position and velocity to a desired final position
and velocity while maintaining safety of flight over the entire duration of the transfer $[144]$.
Each point in the trajectory will be defined by an instantaneous relative ellipse constrained
to be passively safe. This ensures passive safety of flight when future maneuvers cannot be
executed. In many cases, provided the mission is well-designed, it can be shown that passive
safety of flight is also maintained when failures occur mid-burn.

9.1.1 Linear Relative Orbital Element Dynamics

The dynamics for the LROE states follow the relationship from Eqs. 5.42-5.43.

$$
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
-3\omega/2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \omega & 0 & 0 \\
0 & 0 & -\omega & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \omega & 0 \\
0 & 0 & 0 & 0 & -\omega & 0 \\
\end{bmatrix}
\begin{bmatrix}
0 \\
2 \\
0 \\
0 \\
2 \\
0 \\
\end{bmatrix} +
\begin{bmatrix}
0 & 0 \\
0 & 0 \\
2 & 0 \\
0 & 4 \\
0 & 0 \\
0 & 0 \\
\end{bmatrix} \nu = \begin{bmatrix}
0 \\
2 \\
0 \\
0 \\
4 \\
0 \\
\end{bmatrix}
$$

(9.1)
The normalized-time LROE dynamics may also be used, from Eqs. 5.46 and 5.47, which take the form

\[
\dot{x} = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
-3/2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & -1 & 0 \\
\end{bmatrix} x + \begin{bmatrix}
0 & 2 & 0 \\
-2 & 0 & 0 \\
2 & 0 & 0 \\
0 & 4 & 0 \\
0 & 0 & 0 \\
0 & 0 & 1 \\
\end{bmatrix} \mu
\]

Normalized-time examples for optimal trajectory design using the LROEs are included in the results section of this chapter.

### 9.1.2 Objective Function

The objective function for the RPO problem is first defined. The objective is to minimize the \( \Delta v \) required for a chemical combustion engine, to transfer from the initial conditions to the desired final conditions in a fixed amount of time. This objective function is written as

\[
J = \int_{t_0}^{t_f} ||a_T(t)|| \, dt = \omega \int_{t_0}^{t_f} ||\nu(t)|| \, dt \quad (9.3)
\]

Where \( t_0 \) is the initial time, and \( t_f \) is the final time.

### 9.1.3 Constraints

#### Control Effort Constraint

Due to thruster limitations, a constraint on maximum allowable thrust acceleration is imposed. A maximum upper bound on the thrust acceleration level is a convex constraint defined as

\[
||\nu|| \leq \frac{a_{\text{max}}}{\omega} \quad (9.4)
\]

where \( a_{\text{max}} \) is the maximum thrust acceleration.
Boundary Constraints

The boundary constraints include the initial states and desired final states. It is assumed that the full initial state is given in terms of relative position and velocity in LVLH, so that the full LROE state can be calculated via Eq. 5.37. It is also assumed that the initial relative ellipse of the chaser is a passively safe ellipse, therefore the initial point is always feasible and satisfies any initial passive safety of flight constraints.

\[ \bar{x}(t_0) = T \bar{x}(t_0) = \text{given} \quad (9.5) \]

The desired final states can be specified in terms of the final center of motion \( x_r, y_r \), and the size of the final relative ellipse, defined by specifying \( a_r \) and \( A_z \). This leaves two degrees of freedom for the final values of the in-plane and out-of-plane phase angles. A fully-convex final boundary condition can also be implemented, which requires the final phase angles \( E_r(t_f) \) and \( \psi_z(t_f) \) to be specified. In this case, the final condition is fully defined as

\[ \bar{x}(t_f) = \text{specified} \quad (9.6) \]

However, relaxing the final phase angles, although it results in nonconvex constraints, leads to a more optimal trajectory. The terminal boundary constraints with relaxed final phase angles are written as

\[ \bar{x}_1(t_f) = x_{r_{des}}(t_f), \quad \bar{x}_2(t_f) = y_{r_{des}}(t_f) \quad (9.7) \]
\[ a_r(t_f) = a_{r_{des}}(t_f), \quad A_z(t_f) = A_{z_{des}}(t_f) \quad (9.8) \]

The specification of the final desired center of motion is a linear equality constraint on the terminal LROE state. The specification of the final size of the final relative ellipse in terms of the LROE states are quadratic equality constraints and are written as

\[ a_r(t_f) = ||M_{a_r}\bar{x}(t_f)|| = a_{r_{des}}(t_f), \quad A_z(t_f) = ||M_{A_z}\bar{x}(t_f)|| = A_{z_{des}}(t_f) \quad (9.9) \]
where $M_{ar}$ and $M_{Az}$ are selection matrices defined by

$$M_{ar} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad \text{and} \quad M_{Az} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$ (9.10)

Note that the convex constraints for $a_r(t_f)$ and $A_z(t_f)$ below are always satisfied when the
equality constraints are satisfied.

$$||M_{ar}\bar{x}(t_f)|| \leq a_{r_{des}}(t_f), \quad ||M_{Az}\bar{x}(t_f)|| \leq A_{z_{des}}(t_f)$$ (9.11)

This signifies that in some cases, the convex constraints in 9.11 may be deemed sufficient, and
fully convex terminal constraints are defined.

In this analysis the final conditions must also be feasible, so that the terminal ellipse
is a passively safe ellipse that satisfies the safety of flight constraints. It should also be
noted that if all of the terminal states are specified (including the in-plane and out-of-
plane phase angles), the final constraints on the LROEs are all linear as shown in previous
formulations [136].

The solution to the above optimization problem results in a sequence of optimal burn-
coast-burn arcs. [143] The thrusting phases can be relatively short when the limiting thrust
magnitude is large (i.e., near impulsive maneuvers), or relatively long when the limiting
thrust magnitudes are small.

### 9.1.4 Passive Safety of Flight Constraints

While the above problem formulation defines a minimum $\Delta v$ problem with thrust mag-
nitude and boundary constraints, it does not guarantee a passively safe trajectory solution.
A naive approach to addressing passive safety is to set up a cylindrical KOZ aligned with
the along-track $y$-axis to forbid trajectories that might drift into the target after, e.g., a
power failure. Such a KOZ constraint can be simply defined as

$$\sqrt{x^2(t) + z^2(t)} = \sqrt{(\bar{x}_1 - \frac{1}{2}\bar{x}_4)^2 + \bar{x}_5^2} \geq d_{min}$$ (9.12)
where \( d_{\text{min}} \) is the radius of the KOZ and the minimum acceptable target miss distance.

An optimal trajectory solution with this additional constraint will produce a trajectory that lies outside the cylindrical KOZ (assuming the initial and final condition are also outside the KOZ), but it is not guaranteed to be safe since the failure to execute any of the required optimal maneuvers may result in a free-drift trajectory that violates the KOZ and result in a possible collision with the target.

To be more clear, a passively safe trajectory is one in which the loss of control effort (e.g. due to power failure or computer shutdown/reboot) or a decision to temporarily suspend future maneuvers (e.g. due to non-active spacecraft anomalies) will not result in a collision with the target spacecraft, i.e., the subsequent free-drift of the vehicle in the \( y \)-axis direction must not result in an accidental collision.

This problem is best visualized in the \( x - z \) (radial/cross-track) plane. The \( x - z \) plane solutions to the CW equations are simple ellipses that are conveniently described in terms of ROEs.

\[
\frac{4(x - x_r)^2}{a_r^2} + \frac{z^2}{A_z^2} - \frac{4\sin(\gamma)}{a_rA_z}(x - x_r)z = \cos(\gamma)^2
\]

(9.13)

where \( \gamma \) is the phase difference between the in-plane and out-of-plane ellipses

\[
\gamma = E_r - \psi_z
\]

(9.14)

Figure 9.1 shows several examples of these ellipses with varying relative phase difference. When \( x_r \) is greater than zero, the ellipses are drifting in the negative \( y \)-axis direction, and when \( x_r \) is less than zero the ellipse are drifting in the positive \( y \)-axis direction.
Fig. 9.1: Out-of-plane relative ellipse orientations as a function of phase difference ($\gamma$)

The radial-intercept or x-intercept locations for the out-of-plane ellipse can be written as

$$x_{\text{int}} = x_r \pm a_r \frac{\cos(\gamma_{\text{diff}})}{2}$$  \hspace{1cm} \text{(9.15)}

From this result, if the phase difference of the relative ellipse is $\gamma_{\text{diff}} = \pm \pi/2$, the out-of-plane ellipse will cross the plane $z = 0$ and the radial axis of LVLH at the $x_r$ center of motion. This is an example of a perfectly out-of-phase relative ellipse. If the phase difference is $\gamma_{\text{diff}} = 0$ or $\pi$, then the relative ellipse will cross the plane $z = 0$ and the radial axis of LVLH at $x_{\text{int}} = x_r \pm \frac{a_r}{2}$. This second example is termed an in-phase safety ellipse, and signifies that the in-plane and out-of-plane phase angles are equal, or $E_r = \psi_z$. One method to ensure the safety of flight of the chaser satellite after executing a maneuver is to require the terminal relative ellipse to be in phase, with the radial center of motion at $x_r = 0$. Examples of possible out-of-plane ellipse orientations for in-phase safety ellipse trajectories are shown in Figure 9.2 for varying semi-major axes and cross-track amplitudes. One can see that in the case of following an in-phase relative ellipse, the specification of a final desired semi-major axis and cross-track amplitude is sufficient to ensure safety of flight.
If a minimum target miss distance $d_{\text{min}}$ is imposed, three families of passively safe relative motions ellipses exist: a flyby above the target at a distance larger than $d_{\text{min}}$, a flyby below the target at a distance larger than $d_{\text{min}}$, and a relative motion trajectory that circumnavigates the along-track $y$-axis at a distance greater than $d_{\text{min}}$. These three case are shown in Figure 9.3.
Notice that the magnitude of the relative phase difference, $\gamma$, and cross-track amplitude, $A_z$, are unimportant for the family of ellipses that are entirely above or below the minimum miss distance. A view of these flyby trajectories in the $x - y$ plane (radial/along-track) is shown in Figure 9.4. Since $x_r$ can be positive or negative, and since $a_r$ is always positive, the only constraint required to ensure a passively safe trajectory for this family of flyby trajectories (above or below the target) can be expressed in terms of the ROEs.

![Diagram of out-of-plane flyby ellipse example](image)

**Fig. 9.4: Out-of-plane flyby ellipse example**

\[ |x_r| - \frac{a_r}{2} \geq d_{\text{min}} \]  \hfill (9.16)

In terms of the LROEs, this constraint is written as the concave constraint

\[ |\bar{x}_1| - \frac{1}{2} ||M_a, \bar{x}|| \geq d_{\text{min}} \]  \hfill (9.17)

This constraint ensures that all flyby trajectories (above or below) are passively safe.

Next, the family of trajectories that circumnavigate the along-track $y$-axis are considered. In this case, the in-plane motion, the out-of-plane motion, and the phasing of the in-plane/out-of-plane motion is critical to ensure a passively safe trajectory. The $x - y$ in-plane is considered first.
In Figure 9.5, it can be seen that when \( x_r \) is greater than zero, the relative periapse must be less than \(-d_{\text{min}}\)

\[
x_r - \frac{a_r}{2} \leq -d_{\text{min}}
\]  

(9.18)

and when \( x_r \) is less than zero, the relative apoapse must be greater than \(d_{\text{min}}\).

\[
x_r + \frac{a_r}{2} \geq d_{\text{min}}
\]  

(9.19)

Since \( a_r \) is always greater than zero, these two constraints can be combined into a single constraint

\[
\frac{a_r}{2} - |x_r| \geq d_{\text{min}}
\]  

(9.20)

and in terms of the LROEs, this can be expressed as the concave constraint

\[
\frac{1}{2}||M_{a_r}\bar{x}|| - |\bar{x}_1| \geq d_{\text{min}}
\]  

(9.21)

This constraint alone, however, will not guarantee passive safety at all times, for this family of trajectories.

As can be seen in Figure 9.1, constraints on \( A_z \) and the relative phase difference \( \gamma \) must also be imposed. While there are many combinations of \( A_z \) and \( \gamma \) that will result in a passively safe trajectory, a general expression for this constraint was not obvious or
forthcoming. Thus, the approach adopted here is to ensure that the in-plane and out-of-plane motion is nearly in phase ($\gamma \approx 0, \pi$) and then impose the constraint

$$A_z \geq d_{\text{min}}$$ (9.22)

In terms of the LROEs, this can be expressed as the concave constraint

$$||M_{A_z\bar{x}}|| \geq d_{\text{min}}$$ (9.23)

The relative phase angle constraint takes the form

$$-\gamma_{\text{max}} \leq \gamma \leq \gamma_{\text{max}}$$ (9.24)

or

$$\pi - \gamma_{\text{max}} \leq \gamma \leq \pi + \gamma_{\text{max}}$$ (9.25)

To develop these phase difference constraints in terms of the LROEs, consider the following. In the new LROE state-space, the last four LROEs $a_r \sin(E_r)$, $a_r \cos(E_r)$ and $A_z \sin(\psi_z)$, $A_z \cos(\psi_z)$ can be plotted in the same figure. An example of this is shown in Figure 9.6 for $a_r = 2A_z$.

![Diagram showing LROE state-space plots](image)

Fig. 9.6: LROE state-space plots of $(a_r \cos(E_r), a_r \sin(E_r))$, and $(A_z \cos(\psi_z), A_z \sin(\psi_z))$, for $a_r = 2A_z$
In this figure, the unit vectors for the points \((\bar{x}_4, \bar{x}_3)\), and \((\bar{x}_6, \bar{x}_5)\) can be written directly in terms of the LROE states. They are denoted as \(\hat{i}_{E_r}\) for the in-plane phase angle, and \(\hat{i}_{\psi_z}\) for the out-of-plane phase angle.

\[
\hat{i}_{E_r} = \frac{M_{a_r}\bar{x}}{||M_{a_r}\bar{x}||} \quad \text{and} \quad \hat{i}_{\psi_z} = \frac{M_{A_z}\bar{x}}{||M_{A_z}\bar{x}||} \tag{9.26}
\]

Now the constraint \(-\gamma_{\text{max}} \leq \gamma \leq \gamma_{\text{max}}\) can be written in terms of the scalar product of these two unit vectors, as

\[
\hat{i}_{E_r} \cdot \hat{i}_{\psi_z} \geq \cos(\gamma_{\text{max}}) \tag{9.27}
\]

Similarly, the constraint \(\pi - \gamma_{\text{max}} \leq \gamma \leq \pi + \gamma_{\text{max}}\) results in

\[
\hat{i}_{E_r} \cdot \hat{i}_{\psi_z} \leq \cos(\gamma_{\text{max}}) \tag{9.28}
\]

The initial conditions of the problem determine which of the two above constraints are enforced. If \(-\pi/2 \leq \gamma_0 \leq \pi/2\), then the constraint in Eq. 9.27 is enforced, whereas if \(3\pi/2 \geq \gamma_0 > \pi/2\), then the constraint in Eq. 9.28 is enforced. This serves to minimize the control effort required for changes in phase difference.

### 9.2 Safety of Flight Complete Problem Formulation

The complete RPO trajectory planning problem is now formulated as a nonconvex parameter optimization problem. Safety of flight constraints are imposed along the entire trajectory to ensure passive safety of flight. Several of these constraints are inherently nonconvex and will require an SCP technique [21].

#### 9.2.1 Objective Function and Control Effort Constraint

First, the objective function is discretized into \(N\) equal steps, \(\Delta t = t_{i+1} - t_i\). It is assumed that there is constant acceleration over each time step, where \(a_{T_i} = \omega \nu_i\), so that
the integral shown in Eq. (9.3) can be written as

\[ J = \sum_{i=1}^{N} ||aT_i|| = \omega \sum_{i=1}^{N} ||\nu_i|| \tag{9.29} \]

for \( i = 1, \ldots, N \) where constant \( \Delta t \) is omitted. This objective function is convex. The maximum thrust acceleration constraint over each time step is also convex,

\[ ||\nu_i|| \leq \frac{a_{\max}}{\omega} \tag{9.30} \]

### 9.2.2 Dynamics and Boundary Constraints

The dynamics for the system are similarly discretized into points \( \bar{x}(t_i) = \bar{x}_i \) for \( i = 1, \ldots, N + 1 \) with equal steps \( \Delta t = t_{i+1} - t_i \) between each point, and where \( t_1 = t(0) \). Using Eq. 5.42, the dynamics are expressed in terms of linear convex constraints

\[ \bar{x}_{i+1} = \bar{\Phi}_{\Delta t} \bar{x}_i + \bar{B}_d \nu_i \tag{9.31} \]

where

\[ \bar{\Phi}_{\Delta t} = e^{\bar{A} \Delta t}, \quad \bar{B}_d = \int_{t_i}^{t_{i+1}} e^{\bar{A}(t_{i+1} - \tau)} \bar{B} d\tau \tag{9.32} \]

for \( i = 1, \ldots, N \). The initial conditions are linear convex equality constraints

\[ \bar{x}_1 = \text{given} \tag{9.33} \]

The final conditions in terms of the LROEs are given by Eqs. 9.7 and 9.9. The constraints for the final location of the center of the ellipse \( x_r(t_f) \) and \( y_r(t_f) \) are convex linear equality constraints

\[ \bar{x}_{1N+1} = x_{r\text{des}}, \quad \bar{x}_{2N+1} = y_{r\text{des}} \tag{9.34} \]
and the final conditions for the size of the ellipse \( a_r(t_f) \) and \( A_z(t_f) \) are nonconvex quadratic equality constraints

\[
\|M_{ar}\bar{x}_{N+1}\| = a_{r_{des}}, \quad \|M_{Az}\bar{x}_{N+1}\| = A_{z_{des}}
\]  \hspace{1cm} (9.35)

### 9.2.3 Passive Safety of Flight Constraints

The safety of flight constraints are implemented at specific times \( t_i \) along the trajectory. The set \( i_f \in \mathbb{Z} \) is the set of time indices \( t_i \) where the flyby safety of flight constraint is imposed, and the set \( i_c \in \mathbb{Z} \) is the set of time indices \( t_i \) where the circumnavigating safety of flight constraints are imposed. These sets are generally not known a priori, but they can be selected in an efficient manner using the SCP method, as will be seen in the next section.

Thus, the constraint for flyby trajectories is given by Eq. 9.17

\[
|x_{1i}| - \frac{1}{2}\|M_{ar}\bar{x}_i\| \geq d_{min}, \quad i \in i_f
\]  \hspace{1cm} (9.36)

The constraints for circumnavigating trajectories are given by Eqs. 9.21, 9.23, 9.27, and 9.28

\[
\frac{1}{2}\|M_{ar}\bar{x}_i\| - |\bar{x}_{1i}| \geq d_{min}, \quad i \in i_c
\]  \hspace{1cm} (9.37)

\[
\|M_{Az}\bar{x}_i\| \geq d_{min}, \quad i \in i_c
\]  \hspace{1cm} (9.38)

\[
\hat{i}_{E_{ri}} \cdot \hat{i}_{\psi_{zi}} \geq \cos(\gamma_{max}), -\pi/2 \leq \gamma_0 \leq \pi/2, \quad i \in i_c
\]  \hspace{1cm} (9.39)

\[
\hat{i}_{E_{ri}} \cdot \hat{i}_{\psi_{zi}} \leq \cos(\gamma_{max}), 3\pi/2 \geq \gamma_0 \geq \pi/2, \quad i \in i_c
\]  \hspace{1cm} (9.40)

### 9.2.4 Summary

The full nonconvex optimization problem can now be stated
minimize $J = \sum_{i=1}^{N} ||\nu_i||$ \hfill (9.41)

subject to $||\nu_i|| \leq \frac{a_{\text{max}}}{\omega}, \quad i = 1, \ldots, N$ \hfill (9.42)

$\bar{x}_1 = \text{given}$ \hfill (9.43)

$\bar{x}_{i+1} = \Phi_{\Delta t} \bar{x}_i + \bar{B}_d \nu_i, \quad i = 1, \ldots, N$ \hfill (9.44)

$\bar{x}_{1N+1} = x_{r\text{des}}, \quad \bar{x}_{2N+1} = y_{r\text{des}}$ \hfill (9.45)

$||M_{a_r} \bar{x}_{N+1}|| = a_{r\text{des}}, \quad ||M_{A_z} \bar{x}_{N+1}|| = A_{z\text{des}}$ \hfill (9.46)

$|\bar{x}_{1i}| - \frac{1}{2} ||M_{a_r} \bar{x}_i|| \geq d_{\text{min}}, \quad i \in i_f$ \hfill (9.47)

$\frac{1}{2} ||M_{a_r} \bar{x}_i|| - |\bar{x}_{1i}| \geq d_{\text{min}}, \quad i \in i_c$ \hfill (9.48)

$||M_{A_z} \bar{x}_i|| \geq d_{\text{min}}, \quad i \in i_c$ \hfill (9.49)

$\hat{i}_{E_{\nu_i}} \cdot \hat{i}_{\psi_{\mathbf{z}_i}} \leq \cos(\gamma_{\text{max}}), \quad i \in i_c$ \hfill (9.50)

The convex objective function is shown in Eq. 9.41. The constraints in Eqs. 9.42-9.45 are all convex and consist of the maximum thrust acceleration constraint, the initial conditions, the dynamics constraints, and the terminal center of motion constraint. The nonconvex constraints are listed in Eqs. 9.46-9.50. The nonconvex constraints consist of the terminal ellipse size (in-plane and out-of-plane), and the passive safety of flight constraints for circumnavigating and flyby trajectories.

### 9.2.5 Fully Convex Problem

In many cases a fully convex problem may solved, which under certain circumstances may provide inherent safety of flight. Fully convex problems are also significant for solving the nonconvex problem above, since a fully convex problem is used to initialize the method of sequential convex programming. In this section, a fully convex problem (for transferring from the initial conditions to the final conditions in a fixed amount of time) is defined in
terms of the LROEs. This takes the form

\[
\text{minimize } J = \sum_{i=1}^{N} ||\nu_i|| \quad (9.51)
\]

subject to \( ||\nu_i|| \leq \frac{a_{\text{max}}}{\omega}, \quad i = 1, \ldots, N \) \quad (9.52)

\( \bar{x}_1 \) = given \quad (9.53)

\( \bar{x}_{i+1} = \bar{\Phi}_\Delta \bar{x}_i + \bar{B}_d \nu_i, \quad i = 1, \ldots, N \) \quad (9.54)

\( \bar{x}_{N+1} = \text{specified or,} \quad \begin{cases} 
\bar{x}_{1N+1} = x_{r\text{des}}, \quad \bar{x}_{2N+1} = y_{r\text{des}} \\
||M_a \bar{x}_{N+1}|| \leq a_{r\text{des}}, \quad ||M_{A_{\text{r}}} \bar{x}_{N+1}|| \leq A_{z\text{des}}
\end{cases} \) \quad (9.55)

The objective, maximum acceleration, initial conditions, and dynamics (Eqs. 9.51, 9.52, 9.53, 9.54, respectively) all remain the same as in the previous nonconvex problem. However, now the final conditions are fully convex. These can either be fully specified, as shown on the left in Eq. 9.55, or they can be relaxed into a convex form, as shown on the right.

9.3 Solution Using Sequential Convex Programming

The solution to the trajectory optimization problem using sequential convex optimization is to first solve the fully convex problem on the first iteration, excluding all nonconvex constraints. Then, the nonconvex constraints are imposed in each successive iteration via linearized convex approximations, which require the optimal solution from the previous iteration.

Let each iteration of SCP be denoted by \( k \), such that \( y^{(k)} \) is the optimal solution to the convex optimization problem on iteration \( k \). For the safety of flight RPO problem, the variable \( y^{(k)} \) consists of all states \( \bar{x}_i^{(k)} \), and controls \( \nu_i^{(k)} \), on iteration \( k \). Consider the nonconvex problem with parameters \( y^{(k)} \in \mathbb{R}^n \).
minimize $f_0(y^{(k)})$ 

subject to $f_i(y^{(k)}) \geq 0, \ i = 1, \ldots, m$

$h_j(y^{(k)}) = 0, \ j = 1, \ldots, p$

For the trajectory planning problem, the objective function, $f_0$ is convex, while the inequality and equality constraints $f_i$ and $h_j$ are generally nonconvex. The basic approach is to formulate a convex approximation $\hat{f}_i$ of $f_i$, and an affine approximation $\hat{h}_j$ of $h_j$, over a trust region $T^{(k)}$ on each sequential iteration, $k$ [20]. The trust region is updated on each iteration, $k$, and the problem is re-solved over the new trust region so that $y^{k+1}$ is the optimal solution to the new problem. This process is repeated until the convergence criteria used for exiting the sequential convex optimization problem is met.

minimize $f_0(y^{(k+1)})$ 

subject to $\hat{f}_i(y^{(k+1)}) \geq 0, \ i = 1, \ldots, m$

$\hat{h}_j(y^{(k+1)}) = 0, \ j = 1, \ldots, p$

$y^{(k+1)} \in T^{(k)}$

The convex or affine approximations of the nonconvex constraints take the form of a first-order Taylor series expansion of the nonconvex constraint functions $f_i$ and $h_j$, as

$$\hat{f}_i(y^{(k+1)}) = f_i(y^{(k)}) + \nabla f_i(y^{(k)})^T (y^{(k+1)} - y^{(k)}) \geq 0$$ (9.58)

$$\hat{h}_j(y^{(k+1)}) = h_j(y^{(k)}) + \nabla h_j(y^{(k)})^T (y^{(k+1)} - y^{(k)}) = 0$$ (9.59)

9.3.1 Convex Approximations of Nonconvex Constraints

The nonconvex equality constraints, $h_j$, for the trajectory planning problem are given in Eq. 9.46, and the nonconvex inequality constraints, $f_i$, for the trajectory planning problem
include the safety of flight constraints in Eqs. 9.47-9.50. Each of these nonconvex constraints must be approximated within a trust region for each successive iteration.

Beginning with the terminal $a_r$ and $A_z$ constraints in Eq. 9.46, the form for these equations are quadratic equalities

$$h_{a_r}(\bar{x}_{N+1}^{(k)}) = ||M_{a_r}\bar{x}_{N+1}^{(k)}|| - a_{r\text{des}}$$ (9.60)

$$h_{A_z}(\bar{x}_{N+1}^{(k)}) = ||M_{A_z}\bar{x}_{N+1}^{(k)}|| - A_{z\text{des}}$$ (9.61)

When linearized, the approximate functions are written as the lines

$$\hat{h}_{a_r}(\bar{x}_{N+1}^{(k+1)}) = \bar{E}_{r}^{T} |_{\bar{x}_{N+1}^{(k)}} M_{a_r}\bar{x}_{N+1}^{(k)} - a_{r\text{des}}$$ (9.62)

$$\hat{h}_{A_z}(\bar{x}_{N+1}^{(k+1)}) = \bar{E}_{r}^{T} |_{\bar{x}_{N+1}^{(k)}} M_{A_z}\bar{x}_{N+1}^{(k)} - A_{z\text{des}}$$ (9.63)

Next, the nonconvex safety of flight constraints in Eqs. 9.47 and 9.48 are considered. These functions are written as

$$f_{f}(\bar{x}_{i}^{(k)}) = |\bar{x}_{i}^{(k)}| - \frac{1}{2}||M_{a_r}\bar{x}_{i}^{(k)}|| - d_{\text{min}}, \quad i \in i_f$$ (9.64)

$$f_{c}(\bar{x}_{i}^{(k)}) = \frac{1}{2}||M_{a_r}\bar{x}_{i}^{(k)}|| - |\bar{x}_{i}^{(k)}| - d_{\text{min}}, \quad i \in i_c$$ (9.65)

When each is linearized and simplified, the approximate functions are the halfspaces

$$\hat{f}_{f}(\bar{x}_{i}^{(k)}) = \text{sgn}(\bar{x}_{i}^{(k)}) \bar{x}_{i}^{(k+1)} - \frac{1}{2}\bar{E}_{r}^{T} |_{\bar{x}_{i}^{(k)}} M_{a_r}\bar{x}_{i}^{(k+1)} - d_{\text{min}}, \quad i \in i_f$$ (9.66)

$$\hat{f}_{c}(\bar{x}_{i}^{(k)}) = \frac{1}{2}\bar{E}_{r}^{T} |_{\bar{x}_{i}^{(k)}} M_{a_r}\bar{x}_{i}^{(k+1)} - \text{sgn}(\bar{x}_{i}^{(k)}) \bar{x}_{i}^{(k+1)} - d_{\text{min}}, \quad i \in i_c$$ (9.67)
The safety of flight constraint in Eq. 9.49 is similar to the terminal constraint in Eq. 9.46, but is an inequality given by

\[ f_{A_i}(\bar{x}_i^{(k)}) = ||M_{A_i}x_i^{(k)}|| - d_{min}, \quad i \in i_c \]  

(9.68)

and its linearized form is a halfspace

\[ \hat{f}_{A_i}(\bar{x}_i^{(k+1)}) = \hat{i}^{T}_{\psi_z} M_{A_i}x_i^{(k+1)} - d_{min}, \quad i \in i_c \]  

(9.69)

The final nonconvex constraint is the phase difference constraint in Eq. 9.50. This is another inequality constraint taking the form of

\[ f_{pd}(\bar{x}_i^{(k)}) = (\hat{i}_{E_r} \cdot \hat{i}_{\psi_z}) \bigg|_{x_i^{(k)}} - \cos(\gamma_{max}), \quad i \in i_c \]  

(9.70)

and the linearized form is

\[ \hat{f}_{pd}(\bar{x}_i^{(k+1)}) = \left( \hat{i}_{E_r} \cdot \hat{i}_{\psi_z} \right) \bigg|_{x_i^{(k)}} + \nabla f_{pd} \bigg|_{x_i^{(k)}} (\bar{x}_i^{(k+1)} - \bar{x}_i^{(k)}) - \cos(\gamma_{max}), \quad i \in i_c \]

where

\[ \nabla f_{pd} \bigg|_{x_i^{(k)}} = \left[ \begin{array}{cccc} \bar{x}_k ||\hat{i}_{\psi_z} \times \hat{i}_{E_r}|| & \bar{x}_k ||\hat{i}_{E_r} \times \hat{i}_{\psi_z}|| & \bar{x}_k ||\hat{i}_{\psi_z} \times \hat{i}_{E_r}|| & \bar{x}_k ||\hat{i}_{E_r} \times \hat{i}_{\psi_z}|| \\ 0 & 0 & 0 & 0 \end{array} \right] \]  

(9.71)

In this equation, the phase difference is contained within the cross products, as \( \gamma = \arcsin \left( ||\hat{i}_{\psi_z} \times \hat{i}_{E_r}|| \right) \).

9.4 Approximate Convex Optimization Problem

The approximate convex optimization problem can now be defined. Assuming a solution \( y^{(k)} \) is provided from a previous iteration, the approximate convex problem is
\[
\text{minimize } J^{(k+1)} = \sum_{i=1}^{N} ||\nu_i^{(k+1)}|| \\
\text{subject to } ||\nu_i^{(k+1)}|| \leq \frac{a_{\text{max}}}{\omega}, \quad i = 1, \ldots, N \quad (9.72)
\]

\[
\bar{x}_1^{(k+1)} = \text{given} \quad (9.74)
\]

\[
\bar{x}_i^{(k+1)} = \tilde{\Phi}_t \bar{x}_i^{(k+1)} + \tilde{B}_d \nu_i^{(k+1)}, \quad i = 1, \ldots, N \quad (9.75)
\]

\[
\bar{x}_1^{(k+1)} = x_{r_{des}}, \quad \bar{z}_{2N+1}^{(k+1)} = y_{r_{des}} \quad (9.76)
\]

\[
\begin{aligned}
\hat{i}_{E_i}^T \bigg|_{\bar{x}_i^{(k)}} M_{a_x} \bar{x}_{N+1}^{(k+1)} &= a_{r_{des}}, \\
\hat{i}_{\psi_z}^T \bigg|_{\bar{x}_i^{(k)}} M_{A_x} \bar{x}_{N+1}^{(k+1)} &= A_{z_{des}} \\
\end{aligned} \quad (9.77)
\]

\[
\begin{aligned}
\text{sgn}(\bar{x}_{1_1}^{(k)}) \bar{x}_{1_1}^{(k+1)} &- \frac{1}{2} \hat{i}_{E_i}^T \bigg|_{\bar{x}_i^{(k)}} M_{a_x} \bar{x}_i^{(k+1)} \geq d_{\text{min}}, \quad i \in i_f \\
\frac{1}{2} \hat{i}_{E_i}^T \bigg|_{\bar{x}_i^{(k)}} M_{a_x} \bar{x}_i^{(k+1)} - \text{sgn}(\bar{x}_{1_1}^{(k)}) \bar{x}_{1_1}^{(k+1)} &\geq d_{\text{min}}, \quad i \in i_c \\
\hat{i}_{\psi_z}^T \bigg|_{\bar{x}_i^{(k)}} M_{A_x} \bar{x}_i^{(k+1)} &\geq d_{\text{min}}, \quad i \in i_c \\
\end{aligned} \quad (9.78, 9.79, 9.80)
\]

This approximate convex optimization problem is solved iteratively until the difference between the current and previous optimal values of the objective function is within a desired tolerance, \(\epsilon_J\), so that \(\epsilon_J \geq J^{(k+1)} - J^{(k)}\). The nonconvex constraints are evaluated after each iteration to ensure they are satisfied.

### 9.5 Initializing the Sequential Convex Program

The first iteration of the SCP calculates an initial minimum \(\Delta v\) trajectory based on a fully convex problem, as in Eqs. 9.51-9.55, that does not include the passive safety of flight constraints. This first iteration contains the convex constraints shown in Eqs. 9.73 - 9.76 of the nonconvex problem. The remaining constraints include the constraints at the final time, shown in Eq. 9.77 for terminal ellipse size, and Eq. 9.81 for the terminal ellipse phase.
difference. These are handled differently depending on different methods used to initialize the SCP.

Each different method has pros and cons for each problem solved in terms of the number of total iterations. The idea here is to generate a trajectory on the first iteration that is the closest to the true optimal trajectory. In many cases this highly depends on the initial and final conditions, as well as the overall transfer time. Each method is discussed in the following sections.

**Terminal Phase Angle Targeting**

In this section, the safety ellipse size and phase difference constraints are implemented in a convex form on the first iteration by targeting a final in-plane and out-of-plane phase angle \( \theta_f \), which is calculated based on the zero-input ROE differential equation in Eq. 5.36. Under the zero-input assumption, the terminal phase angles are

\[
E_{r_f} = \omega(t_f - t_0) + E_{r_0}
\]

\[
\psi_{z_f} = \omega(t_f - t_0) + \psi_{z_0}
\]

Using these two phase angles, the mean phase angle \( \theta_f \) located between the two zero-input terminal phase angles is

\[
\theta_f = \frac{E_{r_f} + \psi_{z_f}}{2}
\]

so that

\[
E^{(1)}_{r_f} = \psi^{(1)}_{z_f} = \theta_f
\]

Now, since \( \theta_f \) is specified, an in-phase terminal ellipse of the desired size can be targeted to initialize the SCP. The constraints in Eq. 9.77 become convex linear equality constraints, and the constraint in Eq. 9.81 is enforced at the final time, since the final ellipse is in phase. These are now implemented by specifying the entire final state as

\[
\bar{x}^{(1)}_f = \text{specified}
\]
The zero-input assumption for targeting $\theta_f$ works very well when the initial trajectory is passively safe, i.e., when the initial trajectory is a flyby or a circumnavigating trajectory and is nearly in phase ($\gamma_0 \leq 0.1\pi$ rad). Extra consideration is required when the initial trajectory is not passively safe, and will be addressed in future work.

**Terminal In-phase Targeting**

The final in-phase constraint is one method to implement Eq. 9.77, however the terminal ellipse is constrained to be in-phase so that $E_{rf} = \psi_{zf}$ on the first iteration. Given the desired $a_{rdes}$ and $A_{zdes}$, the terminal constraint for final phase difference of zero is implemented as

$$
\bar{x}_{3f}^{(1)} = \frac{a_{rdes}}{A_{zdes}} \bar{x}_{5f}^{(1)}, \quad \bar{x}_{4f}^{(1)} = \frac{a_{rdes}}{A_{zdes}} \bar{x}_{6f}^{(1)}
$$

(9.87)

From Eq. 9.87, this results in

$$
\frac{a_{rf}}{a_{rdes}} \sin(E_{rf}) = \frac{A_{zf}}{A_{zdes}} \sin(\psi_{zf}), \quad \frac{a_{rf}}{a_{rdes}} \cos(E_{rf}) = \frac{A_{zf}}{A_{zdes}} \cos(\psi_{zf})
$$

(9.88)

$$
\frac{\sin(\psi_{zf})}{\sin(E_{rf})} = \frac{a_{rf}}{a_{rdes}} \frac{A_{zdes}}{A_{zf}} = \frac{\cos(\psi_{zf})}{\cos(E_{rf})}
$$

(9.89)

$$
\tan(E_{rf}) = \tan(\psi_{zf}) \rightarrow E_{rf} = \psi_{zf}
$$

(9.90)

so that the final in-plane phase angle, $E_{rf}$, and out-of-plane phase angle, $\psi_{zf}$, are equal, for the case where $\gamma_{max}$ is specified about $\gamma = 0$. For the case where $\gamma_{max}$ is specified about $\gamma = \pi$, the constraints in Eq. 9.87 are

$$
\bar{x}_{3f}^{(1)} = -\frac{a_{rdes}}{A_{zdes}} \bar{x}_{5f}^{(1)}, \quad \bar{x}_{4f}^{(1)} = -\frac{a_{rdes}}{A_{zdes}} \bar{x}_{6f}^{(1)}
$$

(9.91)

This allows the initial trajectory to target an in-phase ellipse with the desired location on the first iteration. After the first iteration, the in-phase terminal constraint is relaxed via the phase difference constraint in Eq. 9.81, and sequential convex programming calculates the optimal trajectory that lies within the desired phase difference, $\gamma_{max}$. The in-phase targeting approach is more accurate as a starting point for an initial ellipse that is highly
out-of-phase, however does not allow the specification of the final ellipse size, and so leads to a minimum control effort solution on the first iteration of $a_{rf} \approx a_{r0}$ and $A_{zf} \approx A_{z0}$. Therefore, a combination of the zero-input and terminal in-phase constraint is implemented, where the terminal phase angle $E_{rf} = \psi_{zf}$ is specified.

**Terminal Phase Angle and In-phase Targeting**

The result of the previous formulations for the initialization of the sequential convex approximation method leads to a combination of the zero-input assumption and the in-phase targeting. This may be applied in a general scenario, for a much more accurate initial guess, leading to faster trajectory convergence. For an initial estimate of the terminal in-phase angle $\theta_f^{(1)}$, where $\theta_f^{(1)} = E_{rf} = \psi_{zf}$, first take a fraction ($\alpha$) of the difference between the two zero-input results in Eq. 9.82 and 9.83.

\[
\alpha \gamma_f^{(1)} = \alpha \left( E_{r}^{(1)} - \psi_{z}^{(1)} \right) \tag{9.92}
\]

then

\[
\theta_f^{(1)} = E_{rf}^{(1)} - \alpha \gamma_f^{(1)} \tag{9.93}
\]

Instead of strict equality constraints, Eq. 9.84 and Eq. 9.85 are relaxed, and the intersection of two convex sets is introduced, (two convex functions of the form $f(x) = ||Ax|| \leq d$ and $g(x) = b^T x \geq d$, representing a norm-ball and a half-space, respectively.)

\[
||M_a \tilde{x}_{N+1}^{(1)}|| \leq a_{r_{des}} \quad \text{and} \quad \tilde{i}_{\theta_f^{(1)}}^T M_a \tilde{x}_{N+1}^{(1)} \geq a_{r_{des}} - d_{ar} \tag{9.94}
\]

\[
||M_{Az} \tilde{x}_{N+1}^{(1)}|| \leq A_{z_{des}} \quad \text{and} \quad \tilde{i}_{\theta_f^{(1)}}^T M_{Az} \tilde{x}_{N+1}^{(1)} \geq A_{z_{des}} - A_{A_z} \tag{9.95}
\]

where $\tilde{i}_{\theta_f^{(1)}} = [\cos(\theta_f^{(1)}) \sin(\theta_f^{(1)})]$. Dividing Eq. 9.94 and 9.95 by $a_{r_{des}}$ and $A_{z_{des}}$, respectively, $d_{ar}/a_{r_{des}}$ and $d_{A_z}/A_{z_{des}}$ are percent difference boundaries on the terminal semi-major axis and cross-track amplitude, respectively. The terminal in-phase constraint is also implemented via Eq. 9.87 or 9.91. Good values for $\alpha$ typically range from 0.5 to 1.1, depend-
ing on the problem initial conditions. In this analysis, for nearly in-phase initial ellipses, 
\[ \frac{d_a}{a_{r,des}} = \frac{d_{A_z}}{A_{z,des}} = 0.25, \text{ and } \alpha = 0.75 \text{ are used.} \]

9.6 Selecting Passive Safety Constraints

The next problem is to determine the sets of time indices \( i_f \) and \( i_c \) where passive safety constraints must be implemented for flyby and circumnavigation trajectories, respectively, for all subsequent iterations. These time indices are initially calculated based on the solution from the first iteration of the sequential convex programming problem. In between each non-zero control from the first iteration, the chaser is in a coasting flight scenario, and the zero-input assumptions to the differential equations are satisfied. Therefore, the flyby or circumnavigating constraints need only be implemented at one point between each non-zero control input. This may be taken as the midpoint between the nonzero controls.

Before the second iteration, these points are the time indices where passive safety constraints must be enforced, and are calculated by evaluating the control solution from the first iteration. This process continues on each iteration, where the passive safety of flight constraints are only implemented between maneuvers, from the solution to the previous iteration. This allows the controls on the following iteration to shift forward or backward in time, while the passive safety of flight constraints are fixed over each particular coasting period of the overall transfer.

First, each state of the trajectory solution from the first iteration is examined to see if it belongs to a flyby component of the trajectory. A state that belongs to a flyby component of a trajectory satisfies \( |x_r| - a_r/2 \geq 0 \). If \( \bar{x}_j^{(k)} \) is determined to be a state belonging to a flyby component of the trajectory, the time index \( j \) is added to the set of indices \( i_f \).

Next, each state of the trajectory solution from iteration \( k \) is examined to see if it belongs to a circumnavigating component of the trajectory. A state that belongs to a circumnavigating component of a trajectory satisfies \( a_r/2 - |x_r| \geq 0 \). If \( \bar{x}_j^{(k)} \) is determined to be a state belonging to a circumnavigating component of the trajectory, the time index \( j \) is added to the set of indices \( i_c \).
9.7 SOF Planner Results

A great variety of scenarios are possible when safety of flight constraints, various initial and final conditions, and various times-of-flight are considered. In this analysis, three common and important scenarios are of primary focus: 1) transferring from an initial flyby trajectory to an offset circumnavigating safety ellipse, 2) transferring from a safety ellipse with large along-track offset to a zero-offset safety ellipse (a.k.a. ‘traveling’ safety ellipse), and 3) changing the size of a safety ellipse.

9.7.1 Fully Convex Results

Several different safety ellipse transfers were solved using the fully-convex optimization method. The constraints for each problem, initial and final conditions, and final transfer time are given for each example. In all of these examples, the mean motion is \( \omega = 0.001 \) radians per second, corresponding to a low-Earth orbit. In the LVLH trajectory plots, the initial relative ellipse is first propagated over the duration of the transfer, as though there were no control on the chaser satellite, and is shown in red. The desired final ellipse is also propagated and shown in green. The optimal transfer trajectory is shown in blue.

The first results show an example of a flyby to safety ellipse transfer. For this scenario, the initial flyby orbit is defined as an initial ellipse with a small semi-major axis and cross-track amplitude. For an approaching flyby, the initial ellipse requires a positive radial, and positive along-track center of motion, or a negative radial, and negative along-track center of motion. The initial ellipse is also defined to be out-of-phase, and the desired final ellipse is in-phase. The initial ROEs are \( x_{r0} = 50 \) meters, \( y_{r0} = 1000 \) meters, \( a_{r0} = 10 \) meters, \( A_{z0} = 5 \) meters, \( E_{r0} = 0 \) radians, and \( \psi_{z0} = 0.1\pi \) radians (18 degrees). The final desired ROEs are \( x_{rf} = 0 \) meters, \( y_{rf} = 0 \) meters, \( a_{rf} = 500 \) meters, \( A_{zf} = 250 \) meters, and final in-phase constraint, \( \theta_f = E_{rf} = \psi_{zf} = E_{r0} + \bar{t}_f \) radians. All initial and final conditions are shown in Tab. 9.1. The final time for this problem is specified to be one orbital period of the target, \( \bar{t}_f = \omega T_p \) radians, where \( T_p = \frac{2\pi}{\omega} \). The problem is discretized into \( N = 100 \) points where \( \Delta \bar{t} = \frac{\omega T_p}{N} \) radians. The LVLH trajectory is shown in Figure 9.7, and the associated controls are shown in Figure 9.8. The plots of the LROE states are shown in Figure 9.9.
Table 9.1: Initial and final ROEs for fully-convex Scenario 1

<table>
<thead>
<tr>
<th></th>
<th>(x_r) (m)</th>
<th>(y_r) (m)</th>
<th>(a_r) (m)</th>
<th>(A_z) (m)</th>
<th>(E_r) (rad)</th>
<th>(\psi_z) (rad)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial</td>
<td>50</td>
<td>1000</td>
<td>10</td>
<td>5</td>
<td>0</td>
<td>0.1 (\pi)</td>
</tr>
<tr>
<td>Final</td>
<td>0</td>
<td>0</td>
<td>500</td>
<td>250</td>
<td>(\theta_f = E_{r0} + \hat{t}_f)</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 9.7: LVLH trajectory plot for flyby to safety ellipse transfer
Fig. 9.8: Control history plot for flyby to safety ellipse transfer in normalized-time

Fig. 9.9: LROE state propagation for flyby to safety ellipse transfer

The next scenario shows an example of a traveling PASE. The initial relative ellipse has a large along-track center of motion, and the final relative ellipse has the center of motion specified to be at the origin of LVLH. Again, the initial ellipse is out-of-phase, and the desired final ellipse in-phase. The initial ROEs are $x_{r_0} = 10$ meters, $y_{r_0} = 2000$ meters, $a_{r_0} = 480$ meters, $A_{z_0} = 260$ meters, $E_{r_0} = 0.3\pi$ radians (54 degrees), and $\psi_{z_0} = 0.15\pi$ radians (27 degrees). The final desired ROEs are $x_{r_f} = 0$ meters, $y_{r_f} = 0$ meters, $a_{r_f} = 500$ meters, $A_{z_f} = 250$ meters, and final in-phase constraint, $\theta_f = E_{r_f} = \psi_{z_f} = E_{r_0} + \dot{\theta}_f$ radians. All initial and final conditions are shown in Tab. 9.2. The final time for this problem is specified
to be three orbital period of the target, $\bar{t}_f = 3\omega T_p$ radians. The problem is discretized into $N = 300$ points where $\Delta\bar{t} = 3\omega T_p N$ radians. The LVLH trajectory is shown in Figure 9.10, and the associated controls are shown in Figure 9.11. The plots of the LROE states are shown in Figure 9.12.

The results from this fully-convex optimization problem show to produce a trajectory that may be considered inherently passively safe (for a minimum safe distance of $\approx 160$ m). This can be seen by the LROE results in Fig. 9.12. In this case, the safety of flight constraints may be evaluated to ensure they are satisfied. Thus, there is no need to add safety of flight constraints into the problem and implement SCP methods.

Table 9.2: Initial and final ROEs for fully-convex Scenario 2

<table>
<thead>
<tr>
<th></th>
<th>$x_r$ (m)</th>
<th>$y_r$ (m)</th>
<th>$a_r$ (m)</th>
<th>$A_z$ (m)</th>
<th>$E_r$ (rad)</th>
<th>$\psi_z$ (rad)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial</td>
<td>10</td>
<td>2000</td>
<td>480</td>
<td>260</td>
<td>0.3$\pi$</td>
<td>0.15$\pi$</td>
</tr>
<tr>
<td>Final</td>
<td>0</td>
<td>0</td>
<td>500</td>
<td>250</td>
<td>$\theta_f = E_{r0} + \bar{t}_f$</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 9.10: LVLH trajectory plot for traveling safety ellipse transfer
The final scenario shows a transfer from a large to small PASE. The initial relative ellipse is nearly centered about the target at the origin of LVLH. The final safety ellipse has a final semi-major axis and cross-track amplitude that are both smaller than the original safety ellipse. Again, the initial ellipse is out-of-phase, and the desired final ellipse is in-phase. The initial ROEs are $x_{r0} = -10$ meters, $y_{r0} = 20$ meters, $a_{r0} = 1010$ meters, $A_{z0} = 535$ meters, $E_{r0} = 1.6\pi$ radians (288 degrees), and $\psi_{z0} = 1.45\pi$ radians (261 degrees). The final desired ROEs are $x_{rf} = 0$ meters, $y_{rf} = 0$ meters, $a_{rf} = 400$ meters, $A_{zf} = 200$ meters, and final in-phase constraint, $\theta_f = E_{rf} = \psi_{zf} = E_{r0} + \bar{t}_f$ radians. All initial and
final conditions are shown in Tab. 9.3. The final time for this problem is specified to be one orbital period of the target, $\bar{t}_f = \omega T_p$ radians. The problem is discretized into $N = 100$ points where $\Delta \bar{t} = \frac{\omega T_p}{N}$ radians. The LVLH trajectory is shown in Figure 9.13, and the associated controls are shown in Figure 9.14. The plots of the LROE states are shown in Figure 9.15.

Similar to the last trajectory, this one also is shown to be inherently safe, via the LROE results in Fig. 9.15. The safety of flight constraints are satisfied on the fully-convex iteration, and the SCP method is not needed.

Table 9.3: Initial and final ROEs for fully-convex Scenario 3

<table>
<thead>
<tr>
<th>$x_r$ (m)</th>
<th>$y_r$ (m)</th>
<th>$a_r$ (m)</th>
<th>$A_z$ (m)</th>
<th>$E_r$ (rad)</th>
<th>$\psi_z$ (rad)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial</td>
<td>-10</td>
<td>20</td>
<td>1010</td>
<td>535</td>
<td>1.6$\pi$</td>
</tr>
<tr>
<td>Final</td>
<td>0</td>
<td>0</td>
<td>400</td>
<td>200</td>
<td>$\theta_f = E_{r0} + \bar{t}_f$</td>
</tr>
</tbody>
</table>

Fig. 9.13: LVLH trajectory plot for large to small safety ellipse transfer
9.7.2 Guaranteed Passive Safety (SCP) Results

In these results, all intermediate relative ellipses in each trajectory are passively safe, guaranteeing a passive abort scenario, if no future maneuvers are performed. Each SCP problem is solved using the CVX package MOSEK solver, with MatLab interface.\cite{111,133}

For consistency, each trajectory is discretized with \( N = 100 \) points per target revolution, and the final time is specified in terms of the number of target revolutions, \( M \), or \( t_f = M(2\pi/\omega) \).

In these results, the value \( \omega = 0.001 \text{ rad/s} \) is used, corresponding to low-Earth orbital RPO.
The maximum control for these scenarios is $a_{\text{max}} = 0.5 \text{ mm/second}^2$. The tolerance for the convergence of the SCP is set to $\epsilon_J = 0.1 \text{ mm/second}$.

For Scenario 1, examples of flyby to circumnavigating ellipses are presented. The flyby consists of a nearly co-elliptic flyby orbit at a sufficiently higher or lower altitude than the target satellite with large along-track separation. The ellipse associated with the flyby consists of a small semi-major axis and cross-track amplitude. The final desired safety ellipse circumnavigates the along-track y-axis and is offset by a specified value. In Scenario 1, trajectories generally consist of multi-rev transfer final times.

The first case in Scenario 1 is an example of a transfer from a lower altitude flyby to a safety ellipse in front of the target. For this case, the trajectory is safe at all times, including during maneuvers. Therefore, if any maneuver cannot be completed, the trajectory remains passively safe. The trajectory has a final transfer time of 2.2 target revs, with $d_{\text{min}} = 80 \text{ m}$, and $\gamma_{\text{max}} = 0.1\pi$. The initial and final ROEs are shown in Table 9.4.

<table>
<thead>
<tr>
<th>Case</th>
<th>Initial</th>
<th>Final</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>$x_r (\text{m})$</td>
<td>$y_r (\text{m})$</td>
</tr>
<tr>
<td>Initial</td>
<td>-500</td>
<td>-4000</td>
</tr>
<tr>
<td>Final</td>
<td>0</td>
<td>1000</td>
</tr>
<tr>
<td>B</td>
<td>Initial</td>
<td>Final</td>
</tr>
<tr>
<td>Initial</td>
<td>-500</td>
<td>-4000</td>
</tr>
<tr>
<td>Final</td>
<td>0</td>
<td>-1000</td>
</tr>
</tbody>
</table>

The number of SCP iterations is $k = 3$, with a total $\Delta v$ of 0.267 m/s. The resulting in-plane and out-of-plane trajectory is shown in Fig. 9.16 and the control history is shown in Fig. 9.17.
Fig. 9.16: Optimal flyby to safety ellipse in front of target, in-plane and out-of-plane motion

Fig. 9.17: Optimal flyby to safety ellipse in front of target, optimal control history

The second case for Scenario 1 is an example of a transfer from a lower altitude flyby to a safety ellipse behind the target. For this case, the trajectory is guaranteed safe if every maneuver is fully executed. However, if the final maneuver is not completed, the resulting trajectory may possibly drift into the target. The trajectory has a final transfer time of 2.2 target revs, with \( d_{\text{min}} = 80 \text{ m} \), and \( \gamma_{\text{max}} = 0.1\pi \). All of the initial and final ROEs are the

same as the first case, except \( y_{rf} = -1000 \) m. The number of SCP iterations is \( k = 3 \), with a total \( \Delta v \) of 0.267 m/s. The resulting in-plane and out-of-plane trajectory is shown in Fig. 9.18 and the control history is shown in Fig. 9.19. It is significant to note that for a transfer from a greater altitude flyby to a circumnavigating safety ellipse, it is safer to transfer to a desired ellipse behind the target.

Fig. 9.18: Optimal flyby to safety ellipse behind target, in-plane and out-of-plane motion

Fig. 9.19: Optimal flyby to safety ellipse behind target, optimal control history
The trajectory design for Scenario 2 involves transferring from one safety ellipse to another safety ellipse where the initial safety ellipse has a large along-track offset, and the final safety ellipse is centered about the target. The change in along-track center of motion \( (\Delta y_r = y_{rf} - y_{ro}) \) is assumed to be relatively large for Scenario 2.

The first case for Scenario 2 is an example with a multi-rev final transfer time. The optimal solution is a traveling safety ellipse that circumnavigate the along-track axis over the entire duration of the trajectory. The relatively large multi-rev transfer time results in a small \( y \)-axis secular drift rate term associated with the radial center of motion of the traveling ellipse. In this scenario, the transfer is safe at all times, including \textit{during} maneuvers, providing passive safety even if a maneuver is not completed. The trajectory has a final transfer time of 2.2 target revs, with \( d_{min} = 80 \) m, and \( \gamma_{max} = 0.1\pi \). The initial and final ROEs are shown in Table 9.5.

Table 9.5: Initial and final ROEs for Scenario 2

<table>
<thead>
<tr>
<th></th>
<th>( x_r ) (m)</th>
<th>( y_r ) (m)</th>
<th>( a_r ) (m)</th>
<th>( A_z ) (m)</th>
<th>( E_r ) (rad)</th>
<th>( \psi_z ) (rad)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case A: Initial</td>
<td>-10</td>
<td>-1000</td>
<td>210</td>
<td>98</td>
<td>0</td>
<td>1.9\pi</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>200</td>
<td>100</td>
<td>( \gamma \leq 0.1\pi )</td>
<td></td>
</tr>
<tr>
<td>Case B: Initial</td>
<td>-10</td>
<td>1000</td>
<td>210</td>
<td>98</td>
<td>( \pi )</td>
<td>1.1\pi</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>200</td>
<td>100</td>
<td>( \gamma \leq 0.1\pi )</td>
<td></td>
</tr>
</tbody>
</table>

The number of SCP iterations is \( k = 3 \), with a total \( \Delta v \) of 0.059 m/s. The resulting in-plane and out-of-plane trajectory is shown in Fig. 9.20 and the control history is shown in Fig. 9.21.
The second case for Scenario 2 is an example with a relatively short transfer time of 0.9 revolutions. With less time, the resulting optimal trajectory is a flyby trajectory from the initial ellipse to the final ellipse. Flyby transfers like this one typically occur when a large change in along-track separation is required in a relatively small time. For this case, \( d_{\min} = 80 \) m, \( \gamma_{\max} = 0.1\pi \), and the initial and final ROEs are given in Table 9.5.
The number of SCP iterations is \( k = 5 \), with a total \( \Delta v \) of 0.123 m/s. The in-plane and out-of-plane trajectory is shown in Fig. 9.22, and the control history is shown in Fig. 9.23. For this case, a possibility of collision exists if the first or last maneuvers are not fully executed.

Fig. 9.22: Optimal safety ellipse to safety ellipse transfer via flyby, in-plane and out-of-plane motion

Fig. 9.23: Optimal safety ellipse to safety ellipse transfer via flyby, optimal control history

The trajectory design for Scenario 3 requires transfers that change the safety ellipse size.
This includes either an increase or decrease in a circumnavigating safety ellipse semi-major axis \( (\Delta a_r = a_{rf} - a_{ro}) \) and/or an increase or decrease in cross-track amplitude \( (\Delta A_z = A_{zf} - A_{zo}) \), with a possible small change in along-track center of motion, and with a transfer final time of approximately one rev. The two cases examined include a large and a small transfer final time, similar to the traveling ellipse scenario. In both cases, the transfer is safe at all times, even if maneuvers are not completed, so that the trajectory is passively safe in the event of control failures.

The first case for Scenario 3 is a transfer from a large to small safety ellipse. The trajectory has a final transfer time of 1.6 target revs, with \( d_{\text{min}} = 30 \) m, and \( \gamma_{\text{max}} = 0.1\pi \). The initial and final ROEs are shown in Table 9.6.

<table>
<thead>
<tr>
<th>Case A:</th>
<th>( x_r ) (m)</th>
<th>( y_r ) (m)</th>
<th>( a_r ) (m)</th>
<th>( A_z ) (m)</th>
<th>( E_r ) (rad)</th>
<th>( \psi_z ) (rad)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial</td>
<td>-10</td>
<td>20</td>
<td>195</td>
<td>110</td>
<td>0.5\pi</td>
<td>0.4\pi</td>
</tr>
<tr>
<td>Final</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>50</td>
<td>( \gamma \leq 0.1\pi )</td>
<td></td>
</tr>
<tr>
<td>Case B:</td>
<td>Initial</td>
<td>10</td>
<td>-20</td>
<td>105</td>
<td>48</td>
<td>0.5\pi</td>
</tr>
<tr>
<td>Final</td>
<td>0</td>
<td>0</td>
<td>200</td>
<td>100</td>
<td>( \gamma \leq 0.1\pi )</td>
<td></td>
</tr>
</tbody>
</table>

The number of SCP iterations is \( k = 4 \), with a total \( \Delta v \) of 0.064 m/s. The resulting in-plane and out-of-plane trajectory is shown in Fig. 9.24 and the control history is shown in Fig. 9.25.
Fig. 9.24: Optimal transfer from large to small safety ellipse, in-plane and out-of-plane motion

The second case for Scenario 3 is an example of a transfer from a small to large safety ellipse. The final transfer time is 0.4 revolutions, with $d_{\text{min}} = 30$ m, and $\gamma_{\text{max}} = 0.1\pi$. The initial and final ROEs for this case are shown in Table 9.6.

The number of SCP iterations is $k = 4$, with a total $\Delta v$ of 0.080 m/s. The resulting in-plane and out-of-plane trajectory is shown in Fig. 9.26 and the control history is shown
in Fig. 9.27.

Fig. 9.26: Optimal transfer from small to large safety ellipse, in-plane and out-of-plane motion

Fig. 9.27: Optimal transfer from small to large safety ellipse, optimal control history

9.8 Summary

This chapter presents an approach to design optimal RPO trajectories that ensures zero probability of collision in the event of a passive failure on a chaser satellite during close proximity operations near a target satellite in a circular reference orbit. This required developing
passive safety of flight constraints using relative orbital elements, which include constraints
on relative ellipse size, position, and orientation. Many of these constraints are inherently
nonconvex, therefore, a method of sequential convex programming is adopted. The problem
is formulated as a minimum $\Delta v$ approximate convex optimization problem, in which the
optimal transfer trajectories are guaranteed to not pass within a prescribed distance of the
target satellite in the event of passive failures. Each nonconvex constraint is convexified via
linearized approximations and implemented in a sequential convex programming problem.

It has been shown that a passively safe trajectory consists of three fundamental build-
ing blocks: 1) flyby trajectories from above, 2) flyby trajectories from below, and 3) cir-
cumnavigating trajectories with the proper in-plane/out-of-plane phasing. Furthermore, an
end-to-end passively safe rendezvous and proximity operations trajectory can be designed
in terms of three trajectory elements: 1) initial approach from a flyby ellipse to an offset
circumnavigating safety ellipse, 2) a transfer from an offset circumnavigating safety ellipse
to another offset circumnavigating safety ellipse of the same size, and 3) increasing or de-
creasing the size of a circumnavigating safety ellipse. By considering these three trajectory
elements, the constraints for passive safety of flight, in terms of relative orbital elements,
are greatly simplified.

Planner results show that for the case where the initial ellipse is a flyby and the final
ellipse is circumnavigating, safety is maintained by switching from a flyby to a circumnavi-
gating trajectory, and this occurs near a point of relative periapse or apoapse of the transfer
trajectory. Careful design of the initial approach trajectory can also ensure passive safety
in the event of a control failure during maneuver execution.

Planner results also demonstrate cases where the initial and final constraints are offset
circumnavigating safety ellipses of the same size, the transfer is either a traveling circum-
navigating ellipse the entire time, or the transfer switches from circumnavigating ellipse to
flyby, then back to circumnavigating ellipse (at points of relative periapse or apoapse). The
optimal solution depends on two important factors: the overall change in along-track center
of motion, and the final transfer time. When a large change in along-track center of motion
is required in a relatively small amount of time (e.g. less than one rev), the secular drift term associated with the radial center of motion may dominate the transfer trajectory, and a flyby is used for the optimal safe approach. Conversely, if more time is allowed for the transfer to take place (e.g. multi-rev), the transfer trajectory will require a smaller secular drift rate and the transfer can be accomplished with a circumnavigating safety ellipse. In both cases, passive safety can be ensured at all times. However, only in the second case can passive safety be ensured in the event of control failures during maneuver execution.

Finally, for cases where a change in the size of a circumnavigating safety ellipse is desired, planner results show that the transfer trajectory is always a circumnavigating ellipse since desired change in the along track motion is always small. It is also shown that these transfers can be performed optimally in approximately one orbital revolution, resulting in an optimal bi-elliptic transfer. Simulation results for these cases have proven to be passively safe, even when control failures occur during a maneuver.
CHAPTER 10
CUSTOM SOCP ALGORITHM FOR ORBITAL RPO

This chapter describes the specific implementation of the SOCP for the orbital RPO trajectory planner. The dynamics for the problem are entirely linear, therefore taking the form of linear equality constraints, and the second-order conic constraints for thrust acceleration magnitude and the approach corridor are included. It is determined from the initial results and analysis that the custom solver is much more efficient when written in terms of the LROE normalized-time dynamics, as opposed to the HCW dynamics. This is due to the fact that in the LROE normalized-time dynamics, all of the problem variables are of similar magnitude, therefore reducing the required amount of scaling and iterations in the custom RPO solver. The LROE dynamics and the associated linear equality constraints are presented in Section 10.1.

The algorithm is also written in terms of the general SOCP formulation described in Chapter 3. Formulating the solver in terms of the most general matrices is required for the commercial solvers. The idea is to distinguish the primary variables in the general SOCP problem from the second-order cone variables in more specific conic solvers. As defined previously, this relationship is given in Eq. 3.17, and is shown here.

\[ \bar{y} = \bar{A}x + \bar{e} \]  

(10.1)

The vector \( \bar{y} = [\bar{y}_1^T, \ldots, \bar{y}_m^T]^T \) contains all second-order conic variables, defined by the affine functions shown in Eq. 3.10. One major advantage of the custom solver presented in this chapter is the fact that, for the RPO trajectory planning problem, many of the second-order cone variables \( \bar{y}_i \) for \( i = 1, \ldots, m \) shown in Eq. 3.12, are simple subsets of the primal variables, \( x \). Therefore, each affine function is explicitly defined using a series of selection matrices. This greatly reduces the dimensions of the gradients and Hessians of
each second-order cone function with respect to $\bar{y}_i$, which are required for Newton’s method. This in turn reduces the dimensions of matrices and number of operations when computing the gradients and Hessians of the KKT conditions, and for programming purposes this serves to reduce the overall memory and time required.

The remainder of this chapter is organized as follows. All second-order cone constraints are presented in Section 10.2, and the total problem statement (i.e. the objective and constraints) is presented in Section 10.3. Problem initialization for satisfying the dynamics equality constraints is shown in Section 10.4, and the conditions of optimality are provided in Section 10.5. This is followed by the solutions for these optimality functions, based on the barrier and PDIP methods, which are presented in Sections 10.6 and 10.7, respectively.

10.1 LROE Dynamics Equality Constraints

The differential equations for the LROE normalized-time dynamics given in Eq. 5.46 are LTI, and take the form

$$\ddot{x}(t) = \bar{A}_n \bar{x}(t) + \bar{B} \bar{\mu}(t)$$

(10.2)

When discretized into $N$ equal time-steps, these dynamics follow the STM relationship, as shown previously

$$\bar{x}(t_{i+1}) = \Phi(t_{i+1}, t_i) \bar{x}(t_i) + \int_{t_i}^{t_{i+1}} \Phi(t_{i+1}, \tau) B \bar{\mu}(t_i) d\tau$$

(10.3)

for $i = 1, \ldots, N$. Where $\Phi(t_{i+1}, t_i) = e^{\bar{A}_n (t_{i+1} - t_i)}$ and $\bar{\mu}(t)$ is assumed to be constant over the interval $t_i \leq t < t_{i+1}$. Note that this is also written as

$$\bar{x}_{i+1} = \Phi_{\Delta t} \bar{x}_i + \bar{B}_d \bar{\mu}_i$$

(10.4)

where $\bar{B}_d$ is constant over each equal time step

$$\bar{B}_d = \int_{t_i}^{t_{i+1}} \Phi(t_{i+1}, \tau) B d\tau$$

(10.5)
and where $\Phi(\Delta t) = \Phi(t_{i+1}, t_i) = e^{A_n \Delta t}$, for $\Delta t = t_{i+1} - t_i$. Writing out each of the dynamics equations in Eq. 10.4, for $i = 1, \ldots, N$, leads to

\[
\begin{align*}
\bar{x}_2 &= \Phi_{\Delta t} \bar{x}_1 + \bar{B}_d \mu_1 \\
\bar{x}_3 &= \Phi_{\Delta t} \bar{x}_2 + \bar{B}_d \mu_2 \\
&\vdots \\
\bar{x}_{N+1} &= \Phi_{\Delta t} \bar{x}_N + \bar{B}_d \mu_N
\end{align*}
\]  

(10.6)

(10.7)

(10.8)

These dynamics equations fully define the linear equality constraints between all control variables, $\mu_i$, and state variables, $\bar{x}_i$, in the SOCP. These linear equality constraints can be realized in two ways. The first method involves formulating the problem in terms of all controls and states as variables, and the second formulates the problem only in terms of the controls as variables, as shown in the following sections.

10.1.1 Control and State Variables Equality Constraints

Using the results from Eq. 10.6-10.8, the full linear dynamics equations can be written, using all state variables, $\bar{x}_i$ for $i = 2, \ldots, N$, and controls, $\mu_i$ for $i = 1, \ldots, N$. Note that the full initial and final states are assumed to be known, or provided, from the problem specifications, therefore, these are not considered problem variables in this formulation.

This results in $3N$ control variables and $6(N - 1)$ state variables, for all intermediate states between $\bar{x}_1$ and $\bar{x}_{N+1}$. Let all of these variables be written in vector $x_{ux} \in \mathbb{R}^{9N-6}$ as

\[
x_{ux} = \begin{bmatrix} u \\ x \end{bmatrix}, \text{ where } u = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_N \end{bmatrix}, \text{ and } x = \begin{bmatrix} \bar{x}_2 \\ \vdots \\ \bar{x}_N \end{bmatrix}
\]

(10.9)

Then, all of the dynamics can be written in a block-partitioned matrix form. First, let the $6N \times 3N$ matrix associated with the control vectors be defined as $B$, which is filled with
block-diagonal $B_d$ matrices as shown below.

$$
B = \begin{bmatrix}
\bar{B}_d & 0 & \ldots & 0 \\
0 & \bar{B}_d & \ldots & 0 \\
\vdots & \ddots & \ddots & \vdots \\
0 & \ldots & 0 & \bar{B}_d
\end{bmatrix}_{6N \times 3N}
$$

(10.10)

Next, define the $6N \times 6(N - 1)$ matrix associated with the states as $A$, which is filled with identity, and state transition matrices, $\tilde{\Phi}_{\Delta t}$, as shown.

$$
A = \begin{bmatrix}
-I & 0 & 0 & \ldots & 0 \\
\tilde{\Phi}_{\Delta t} & -I & 0 & \ldots & 0 \\
0 & \tilde{\Phi}_{\Delta t} & -I & \ldots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots \\
0 & \ldots & 0 & \tilde{\Phi}_{\Delta t} & -I \\
0 & \ldots & 0 & 0 & \tilde{\Phi}_{\Delta t}
\end{bmatrix}_{6N \times 6(N-1)}
$$

(10.11)

The known parameters $\bar{x}_1$ and $\bar{x}_{N+1}$ are moved to the right-hand-side of these equations, since they are not variables, resulting in the $6N \times 1$ vector $h_{ux}$.

$$
h_{ux} = \begin{bmatrix}
-\tilde{\Phi}_{\Delta t} \bar{x}_1 \\
0 \\
\vdots \\
0 \\
\bar{x}_{N+1}
\end{bmatrix}
$$

(10.12)

The full dynamics for the control and state variables formulation can now be written as

$$
Bu + Ax = h_{ux}
$$

(10.13)
or, using the previously defined vector of all variables, $x_{ux}$

$$G_{ux}x_{ux} = h_{ux} \quad (10.14)$$

where

$$G_{ux} = [B \ A] \quad (10.15)$$

This results in a linear system of $6N$ equations (therefore, $6N$ equality constraints), where $G_{ux} \in \mathbb{R}^{6N \times 9N-6}$ is full row-rank, or $\text{rank}(G_{ux}) = 6N$.

### 10.1.2 Control-only Variables Equality Constraints

To remove all states $\bar{x}_i$ from the problem, first notice that each equality constraint in Eqs. 10.6-10.8 can be substituted into the next one, in order to combine intermediate states $\bar{x}_j$ for $j = 2, \ldots, N$. For example, substituting Eq. 10.6 into Eq. 10.7 leads to

$$\bar{x}_3 = \Phi_{2t}\bar{x}_1 + \bar{x}_1 = \Phi_{2t}\bar{x}_1 + \bar{x}_1$$

Doing this for all of the equality constraints leads to a linear equation that is only a function of the control vectors, $\mu_i$ for $i = 1, \ldots, N$, and boundary states, $\bar{x}_1$ and $\bar{x}_{N+1}$. This is written as

$$\bar{x}_{N+1} - \Phi_{Nt}\bar{x}_1 = \sum_{i=1}^{N} \Phi_{it}^{N-i} B_d \mu_i \quad (10.17)$$

or, more generally for any $\bar{x}_i$ such that $t_1 \leq t_i \leq t_{N+1}$ as

$$\bar{x}_i - \Phi_{it}\bar{x}_1 = \sum_{j=1}^{i-1} \Phi_{it-j}^{i-j-1} B_d \mu_j \quad (10.18)$$
Eq. 10.17 can be written as

\[
\begin{bmatrix}
\Phi^{N-1} B_d & \Phi^{N-2} B_d & \ldots & \Phi^1 B_d & B_d
\end{bmatrix}
\begin{bmatrix}
\mu_1 \\
\mu_2 \\
\vdots \\
\mu_N
\end{bmatrix} = x_{N+1} - \Phi^1 \bar{x}_1
\] (10.19)

Define all control vectors as the variables \( x_u = [\mu_1^T, \mu_2^T, \ldots, \mu_N^T]^T \) and

\[
G_u = \begin{bmatrix}
\Phi^{N-1} B_d & \Phi^{N-2} B_d & \ldots & \Phi^1 B_d & B_d
\end{bmatrix}
\] (10.20)

\[
h_u = x_{N+1} - \Phi^1 \bar{x}_1
\] (10.21)

so that the equality constraints can be written as

\[
G_u x_u = h_u
\] (10.22)

This results in six linear equations (6 equality constraints), where \( G_u \in \mathbb{R}^{6 \times 3N} \) is full row-rank, or \( \text{rank}(G_u) = 6 \), and therefore all vectors \( d_u \in \mathcal{R}(B) \) for \( d \in \mathbb{R}^6 \). From this result, the position and velocity states \( (x) \) in LVLH have been removed and are no longer variables in the problem.

### 10.2 Second-order Conic Constraints

The second-order cone constraints that are desired for this custom solver involve the specification of maximum control acceleration level, slack variables, and approach corridor constraints. In this section, all constraints are written in terms of the original second-order cone constraints in Eq. 3.6. There are a total of \( m = 3N \) second-order cone constraints in this formulation. Note that these constraints can be easily converted to represent the approximated spherical KOZ constraints or safety of flight constraints, since each takes the form of a hyperplane constraint, which is a subset of a second-order cone constraint.
Maximum Control and Slack Variable Constraints

Specifying a maximum acceleration level, \( u_{\text{max}} \), over each time step. This constraint is written as

\[
\|\mu_i\| \leq \frac{u_{\text{max}}}{\omega^2}, \quad \text{for } i = 1, \ldots, N \tag{10.23}
\]

Using the slack variable form for the objection function, as shown in Eq. 7.8, the second-order cone constraints for both the maximum acceleration level and the slack variables are

\[
\|\mu_i\| \leq \eta_i, \quad \text{for } i = 1, \ldots, N \tag{10.24}
\]

\[
\eta_i \leq \frac{u_{\text{max}}}{\omega^2}, \quad \text{for } i = 1, \ldots, N \tag{10.25}
\]

for a total of \( 2N \) inequality constraints, where \( \mu_i \) are the problem variables, and \( \eta_i \) are slack variables. It is significant to note that depending on the initial and final conditions, the overall time for the transfer, and the number of points \( N \), the problem may be infeasible for a small values of \( u_{\text{max}} \). This possibility can be visualized by considering a scenario in which the initial and final position are very distant, while the final time and maximum control effort are very small. This scenario may be infeasible due to limited control effort over the duration of the problem.

Approach Corridor Constraints

The convex approach corridor constraint for the LROE formulation is given by

\[
\|M_rT^{-1}\hat{x}_i\| \leq \frac{1}{\cos(\alpha)} \hat{i}_c^T M_rT^{-1}\hat{x}_i, \quad i = 1, \ldots, N \tag{10.26}
\]

where \( M_r = [I_{3\times3} \ 0_{3\times3}] \) selects the position component, and \( T^{-1} \) is the transformation from LOREs to the HCW states, given in Eq. 5.41. The cone is defined by the vector \( \hat{i}_c \) and half-angle, \( \alpha \).

In the control-only formulation of this problem, these cone constraints become a bit more complex. Eq. 10.18 defines the LROEs at \( t_i \) as a function of controls and the initial
condition. This can be substituted into Eq. 10.26 to get the control-only formulation of the cone constraint as

\[
\left\| M_r T^{-1} \left[ \Phi_{\Delta t}^{i-1} \dot{x}_1 + \sum_{j=1}^{i-1} \Phi_{\Delta t}^{i-j-1} \dot{B}_d \mu_j \right] \right\| \leq \frac{1}{\cos(\alpha)} \hat{T}_c M_r T^{-1} \left[ \Phi_{\Delta t}^{i-1} \dot{x}_1 + \sum_{j=1}^{i-1} \Phi_{\Delta t}^{i-j-1} \dot{B}_d \mu_j \right],
\]

for \( i = 1, \ldots, N \)

(10.27)

This formulation of the approach corridor constraint for the control-only formulation will be utilized in the general SOCP problem provided in the following section.

10.3 SOCP for RPO Problem Statement

The SOCP algorithm for developing a custom solver involves the implementation of the RPO second-order conic constraints for maximum control acceleration and approach corridor, subject to the linear dynamics constraints.

Controls and States Formulation

The primal problem for the control and state formulation is written as

\[
\begin{align*}
\text{minimize} & \quad f_\eta^T x_{\etaux} \\
\text{subject to} & \quad \|A_i x_{\etaux}\| - c_i^T x_{\etaux} - d_i \leq 0, \text{ for } i = 1, \ldots, 3N \\
& \quad G_{\etaaux} x_{\etaaux} = h_{aux}
\end{align*}
\]

where for the control and state formulation, the variables include controls, slack variables, and the position/velocity states, \( \mu_i, \eta_i \) for \( i = 1, \ldots, N \), and \( x_i \) for \( i = 2, \ldots, N \). Let all of these primal variables be defined by the vector \( x_{\etaaux} \in \mathbb{R}^{10N-6} \), where

\[
\begin{bmatrix}
\eta_1 \\
\vdots \\
\eta_N \\
\end{bmatrix}, \quad 
\begin{bmatrix}
\mu_1 \\
\vdots \\
\mu_N \\
\end{bmatrix}, \quad 
\begin{bmatrix}
x_2 \\
\vdots \\
x_N \\
\end{bmatrix}
\]

(10.29)
\[
x_{\eta u x}^T = \left[ \eta^T \quad u^T \quad x^T \right]
\]  

(10.30)

To define the user-supplied parameters, \(f_\eta, A_i, c_i, d_i, G_{\eta u x}, \) and \(h_{u x},\) the following selection matrices are defined. These matrices extract portions of the full state vector \(x_{\eta u x}\) and are used to simplify the notation.

\[
\eta = M_{\eta u x}^\eta x_{\eta u x}, \quad \text{where} \quad M_{\eta u x}^\eta = \begin{bmatrix} I_{\eta} & 0 & 0 \\ N & 3N & 6N-6 \end{bmatrix} \]  

(10.31)

\[
u = M_{\eta u x}^u x_{\eta u x}, \quad \text{where} \quad M_{\eta u x}^u = \begin{bmatrix} 0 & I_u & 0 \\ N & 3N & 6N-6 \end{bmatrix} \]  

(10.32)

\[
x = M_{\eta u x}^x x_{\eta u x}, \quad \text{where} \quad M_{\eta u x}^x = \begin{bmatrix} 0 & 0 & I_x \\ N & 3N & 6N-6 \end{bmatrix} \]  

(10.33)

\[
x_{u x} = M_{\eta u x}^{u x} x_{\eta u x}, \quad \text{where} \quad M_{\eta u x}^{u x} = \begin{bmatrix} 0 & I_u & 0 \\ 0 & 0 & I_x \\ N & 3N & 6N-6 \end{bmatrix} \]  

(10.34)

Additional selection matrices are defined to select individual elements of the \(\eta, u, \) and \(x\) variables.

\[
\eta_i = M_{\eta}^{\eta i} \eta, \quad \text{where} \quad M_{\eta}^{\eta i} = \begin{bmatrix} 0, \ldots, 1_{\eta_i}, \ldots, 0 \end{bmatrix}
\]  

(10.35)

\[
\mu_i = M_{u}^{\mu i} u, \quad \text{where} \quad M_{u}^{\mu i} = \begin{bmatrix} 0, \ldots, 1_{\mu_i}, \ldots, 0 \end{bmatrix}
\]  

(10.36)

\[
x_i = M_{x}^{x i} x, \quad \text{where} \quad M_{x}^{x i} = \begin{bmatrix} 0, \ldots, 1_{x_i}, \ldots, 0 \end{bmatrix}
\]  

(10.37)

Now these selection matrices are used to conveniently define the problem parameters. Since the objective function in Eq. 7.8 is given by

\[
J = \sum_{i=1}^{N} \eta_i
\]  

(10.38)
the vector $f^T_\eta$ in Eq. 10.28 is written as

$$f^T_\eta = 1^T M^\eta_{ux}$$

(10.39)

The $\eta_i$'s are not included in the linear equality constraints, therefore, the matrix $G_{ux}$ is

$$G_{ux} = G_{ux} M^ux_{\etaux}$$

(10.40)

where $G_{ux}$ is given by Eq. 10.15. The vector $h_{ux}$ is given by Eq. 10.12.

The $A_i$, $c_i^T$, and $d_i$ data are based on the second-order cone constraints in Eq. 10.24, 10.25, and 10.26. All results for the $A_i$, $c_i^T$, and $d_i$ data are summarized below.

$$A_i = \begin{cases} M^{ux}_{\etaux}, & \text{for } 1 \leq i \leq N \\ 0, & \text{for } N + 1 \leq i \leq 2N \\ M_rT^{-1}M^x_{\etaux}M^x_{\etaux}, & \text{for } 2N + 1 \leq i \leq 3N, j = i - 2N \end{cases}$$

(10.41)

$$c_i^T = \begin{cases} M^\eta_{\etaux}M^\eta_{\etaux}, & \text{for } 1 \leq i \leq N \\ -M^\eta_{\etaux}M^\eta_{\etaux}, & \text{for } N + 1 \leq i \leq 2N, j = i - N \\ \frac{1}{\cos(\alpha)}c^T_i M_rT^{-1}M^x_{\etaux}M^x_{\etaux}, & \text{for } 2N + 1 \leq i \leq 3N, j = i - 2N \end{cases}$$

(10.42)

$$d_i = \begin{cases} 0, & \text{for } 1 \leq i \leq N \\ u_{max}/\omega^2, & \text{for } N + 1 \leq i \leq 2N \\ 0, & \text{for } 2N + 1 \leq i \leq 3N \end{cases}$$

(10.43)

All matrix data for the RPO trajectory planning problem is now defined and the SOCP formulation in Eq. 3.6.

**Control-only Formulation**

For the control-only formulation, the SOCP takes the form
minimize \quad f^T_x \eta_{\nu} \quad (10.44)

subject to \quad ||A_i \eta_{\nu} + b_i|| - c_i^T \eta_{\nu} - d_i \leq 0, \quad for \ i = 1, \ldots, 3N

\quad G_{\eta \nu} \eta_{\nu} = h_u

where the variables for the control-only problem include all controls \( \mu_i \), and slack variables \( \eta_i \), in vector \( x_{\eta \nu} \), defined as

\[ \eta = \begin{bmatrix} \eta_1 \\ \vdots \\ \eta_N \end{bmatrix}, \quad u = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_N \end{bmatrix} \quad (10.45) \]

\[ x_{\eta \nu}^T = \begin{bmatrix} \eta^T & u^T \end{bmatrix} \quad (10.46) \]

where \( x_{\eta \nu} \in \mathbb{R}^{4N} \) contains all 4N problem variables.

To define the user-supplied parameters \( f^T_\eta, A_i, b_i, c_i, d_i, G_{\eta x}, \) and \( h_u \), the following selection matrices are defined for the control-only formulation.

\[ \eta = M_{\eta \nu}^\eta x_{\eta \nu}, \quad where \quad M_{\eta \nu}^\eta = \begin{bmatrix} I_\eta & 0 \\ N & 3N \end{bmatrix}^N \quad (10.47) \]

\[ u = M_{\eta \nu}^u x_{\eta \nu}, \quad where \quad M_{\eta \nu}^u = \begin{bmatrix} 0 & I_u \\ N & 3N \end{bmatrix}^{3N} \quad (10.48) \]

and the selection matrices for the individual elements of \( \eta \) and \( u \) at times \( t_i \), are

\[ \eta_i = M_{\eta \nu}^\eta \eta, \quad where \quad M_{\eta \nu}^\eta = \begin{bmatrix} 0, \ldots, 1_{\eta_i}, \ldots, 0 \end{bmatrix} \quad (10.49) \]

\[ \mu_i = M_{u \nu}^\mu u, \quad where \quad M_{u \nu}^\mu = \begin{bmatrix} 0, \ldots, I_{\mu_i}, \ldots, 0 \end{bmatrix} \quad (10.50) \]
The objective function from Eq. 7.8 is given by

\[ J = \sum_{i=1}^{N} \eta_i \]  \hspace{1cm} (10.51)

Thus, the vector \( f^T_\eta \) in Eq. 10.28 is

\[ f^T_\eta = 1^T M^\eta_{\eta u} \]  \hspace{1cm} (10.52)

The \( \eta_i \)'s are not included in the linear equality constraints, therefore, the matrix \( G_{\eta u} \) is

\[ G_{\eta u} = G_u M^u_{\eta u} \]  \hspace{1cm} (10.53)

where \( G_u \) is given by Eq. 10.20. The vector \( h_u \) is given by Eq. 10.21.

The \( A_i, b_i, c^T_i, \) and \( d_i \) data are based on the second-order cone constraints in Eq. 10.24, 10.25, and 10.27. All results for the \( A_i, c^T_i, \) and \( d_i \) data are summarized below.

\[
A_i = \begin{cases} 
M^u_{\eta u}, & \text{for } 1 \leq i \leq N \\
0, & \text{for } N + 1 \leq i \leq 2N \\
M_r T^{-1} \left[ \sum_{k=1}^{j-1} \Phi_{\Delta t}^{j-k-1} B_d M^u_{u_k} \right] M^u_{\eta u}, & \text{for } 2N + 1 \leq i \leq 3N, j = i - 2N
\end{cases} \]  \hspace{1cm} (10.54)

\[
b_i = \begin{cases} 
0, & \text{for } 1 \leq i \leq N \\
0, & \text{for } N + 1 \leq i \leq 2N \\
M_r T^{-1} \Phi_{\Delta t}^{j-i-1} \hat{x}_1, & \text{for } 2N + 1 \leq i \leq 3N, j = i - 2N
\end{cases} \]  \hspace{1cm} (10.55)

\[
c^T_i = \begin{cases} 
M^\eta_{\eta u} + M^\eta_{\eta u}, & \text{for } 1 \leq i \leq N \\
-M^\eta_{\eta u} + M^\eta_{\eta u}, & \text{for } N + 1 \leq i \leq 2N, j = i - N \\
\frac{1}{\cos(\alpha)} i^T M_r T^{-1} \left[ \sum_{k=1}^{j-1} \Phi_{\Delta t}^{j-k-1} B_d M^u_{u_k} \right] M^u_{\eta u}, & \text{for } 2N + 1 \leq i \leq 3N, j = i - 2N
\end{cases} \]  \hspace{1cm} (10.56)
\[ d_i = \begin{cases} 0, & \text{for } 1 \leq i \leq N \\ u_{\text{max}}/\omega^2, & \text{for } N + 1 \leq i \leq 2N \\ \frac{1}{\cos(\alpha)} \hat{i}_c^T M_r T^{-1} \Phi_{\Delta t}^{-1} \hat{x}_1, & \text{for } 2N + 1 \leq i \leq 3N, j = i - 2N \end{cases} \] (10.57)

All problem matrices are now defined for the SOCP problem formulation shown in Eq. 3.6.

### 10.4 Initialization

The initialization for the SOCP involves satisfying all of the linear equality constraints (dynamics equations) and does not require initially satisfying all second-order cone constraints, since an infeasible start method is used to satisfy them before the feasible start method begins. The initialization for the infeasible start is done using a least-squares solution. For the controls and state variables formulation, all controls \( u_i \), and states \( \bar{x}_i \), can be initialized by

\[
x_{u_\bar{x}}^0 = G_{u_\bar{x}}^T (G_{u_\bar{x}} G_{u_\bar{x}}^T)^{-1} h_{u_\bar{x}}
\] (10.58)

where \( G_{u_\bar{x}} \) is taken from Eq. 10.15. For the control-only formulation, all control vectors \( u_i \) are initialized by

\[
x_{u}^0 = G_{u}^T (G_{u} G_{u}^T)^{-1} h_{u}
\] (10.59)

where \( G_u \) is taken from Eq. 10.20. In both formulations, the \( \eta_i \)'s are initialized to be feasible via

\[
\eta_i = ||u_i|| + \epsilon_\eta
\] (10.60)

where \( \epsilon_\eta > 0 \) is a relatively small value.

A fully feasible solution to the RPO SOCP problem can be determined a priori, provided that \( u_{\text{max}} \) and all other cone/corridor/halfspace constraints are omitted from the initial problem formulation. In this case, a feasible \( x_{u_\bar{x}} \) or \( x_u \) is fully determined from the results of the above two equations. The addition of \( u_{\text{max}} \) constraints, and other cone/halfspace constraints requires an infeasible start method to first determine a fully feasible initial point, provided one exists.
The objective of the infeasible start method is to determine an initial point \(x_{\text{feas}}\) that satisfies all constraints \(Gx_{\text{feas}} = h\) and \(f_i(x_{\text{feas}}) \leq 0\) for \(i = 1, \ldots, m\). From the initialization technique shown in Eqs. 10.58-10.60, all affine equality constraints and slack variables for control magnitude can be satisfied, however, the inequality constraints for the cone approach and maximum control shown in Eq. 10.24, 10.25, and 10.26 may not be. In this case, we would have

\[
f_i(x_{\text{feas}}) = ||A_i x_{\text{feas}}|| - c_i^T x_{\text{feas}} - d_i > 0
\]  

(10.61)

for some \(i\). Therefore, an infeasible start is required. The variables for this problem, denoted by subscript ‘\(\text{feas}\)’, are the infeasible start variables, \(x_{\text{feas}}\). From the problem shown in Eq. 10.28, the infeasible start is formulated by adding a slack variable \(e\) (scalar), to all second-order cone constraints, so that

\[
f_i(x_{\text{feas}}) = ||A_i x_{\text{feas}}|| - c_i^T x_{\text{feas}} - d_i \leq e, \text{ for } i = 1, \ldots, m
\]  

(10.62)

The objective is then to minimize the slack variable, \(e\), to drive the solution towards the feasible set. The infeasible start problem is reformulated by adding the slack variable \(e\) to the infeasible start variables \(x_{\text{feas}}\), and is stated as

\[
\begin{align*}
\text{minimize} & \quad f_e^T x_{\text{feas}} \\
\text{subject to} & \quad ||A_i x_{\text{feas}}|| - c_i^T x_{\text{feas}} - d_i \leq e, \text{ for } i = 1, \ldots, m \\
& \quad Gx_{\text{feas}} = h
\end{align*}
\]  

(10.63)

where

\[
f_e = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad x_{\text{feas}} = \begin{bmatrix} x_{\text{qux}} \\ e \end{bmatrix}, \quad c_i = \begin{bmatrix} c_i \\ 1 \end{bmatrix}
\]  

(10.64)

and the matrices \(G\) and \(A_i\) from Eq. 10.15 and 10.41 are adjusted accordingly, by adding a column of zeros.
It is sufficient to terminate the infeasible start problem when the solution $x_{feas}$ is feasible, to within some desired tolerance $\epsilon_{feas}$. This is given by

$$e \leq -|\epsilon_{feas}|$$

(10.65)

When this condition is met, then the infeasible start terminates, and the original feasible problem is solved.

### 10.5 Conditions of Optimality for RPO SOCP Algorithm

This section presents the conditions of optimality specifically for the convex RPO planner problem. For brevity, the Lagrangian, KKT conditions, and solution method for only the controls and states formulation is presented here, though the same theory can also be applied for the control-only formulation.

#### 10.5.1 Lagrangian

The Lagrangian for the specific controls and states SOCP in Eq. 10.28 is now written. In this form, all data matrices for the constraints are inserted into this equation, thus the Lagrangian is written in terms of the original control and state variables. All convex inequality constraint functions in Eq. 10.24-10.26 take the squared form as shown in Eq. 3.28.

$$L(\eta, u, x, \lambda, \nu) = \sum_{i=1}^{N} \eta_i + \sum_{i=1}^{N} \lambda_i (\mu_i^T \mu_i - \eta_i^2) + \sum_{i=1}^{N} \lambda_{i+N} \left(\eta_i - \frac{u_{\max}}{\omega^2}\right)^2 + \sum_{i=1}^{N} \lambda_{i+2N} \left(\tilde{x}_i^T M_i T^{-1} \tilde{x}_i - \frac{1}{\cos^2 \alpha} (\hat{i}_c^T M_i T^{-1} \hat{x}_i)^2\right) + \nu^T (G_{\eta u x} x_{\eta u x} - h_{\eta u x})$$

(10.66)

where there are $m = 3N$ inequality constraints and, $p = 6N$ equality constraints. The matrix $G_{\eta u x}$ is from Eq. 10.40 and $h_i$ is from Eq. 10.12. The vector $\lambda = [\lambda_1 \ldots \lambda_m]^T$ contains all Lagrange multipliers for the second-order cones, and the vector $\nu = [\nu_1 \ldots \nu_p]^T$ contains
all Lagrange multipliers for the equality constraints.

10.5.2 KKT Conditions

The KKT conditions for this problem are [20]

\[ G_{\etaux}x_{\etaux}^* = h_{ux} \quad (10.67) \]
\[ \mu_i^* - \eta_i^2 \leq 0, \quad i = 1, \ldots, N \quad (10.68) \]
\[ \eta_i^2 - \frac{u_{max}}{\omega^4} \leq 0, \quad i = 1, \ldots, N \quad (10.69) \]
\[ x_i^T T^{-T} M_i T M_i T^{-1} x_i^* \]
\[ - \frac{1}{\cos^2 \alpha} \left( i_i^T M_i T^{-1} x_i^* \right)^2 \leq 0, \quad i = 1, \ldots, N \quad (10.70) \]
\[ \lambda_i^* \geq 0, \quad i = 1, \ldots, 3N \quad (10.71) \]
\[ f_\eta + ([D\bar{f}(x_{\etaux}^*)])^T \lambda^* + G_{\etaux}^T \nu^* = 0 \quad (10.72) \]
\[ \lambda_i^* \left( f_i(x_{\etaux}^*) \right) = 0, \quad i = 1, \ldots, 3N \quad (10.73) \]

where \([D\bar{f}(x_{\etaux}^*)]^*\) is a matrix of the derivatives of \(f_i(x_{\etaux})\) at \(x_{\etaux}^*\) for the SOCP, as shown in Eq. 3.42, where \(\bar{f}(x_{\etaux}) = [f_1(x_{\etaux}), \ldots, f_{3N}(x_{\etaux})]^T\) (10.74)

The KKT equations are described as follows. Eq. 10.67 are the linear equality constraints for the dynamics, Eq. 10.68-10.70 are all of the second-order cones, Eq. 10.71 are the constraints for all second-order cone Lagrange multipliers to be nonnegative, Eq. 10.72 is the requirement that the gradient must vanish at \(x_{\etaux}^*\), and Eq. 10.73 is the complementary slackness condition.

10.6 RPO Planning Algorithm Using Barrier Method

In this section, the barrier method is applied to the custom RPO problem. This requires solving the optimality conditions for the modified objective function with the log-barrier functions, as presented in Chapter 3. The solver method is written in terms of \(x_{\etaux}\), where \(x_{\etaux}\) is a vector of all primal slack variables, controls, and states, for the states and controls
problem formulation. Similarly, the matrix $G_{\etaux}$ and vector $h_{ux}$ represent the dynamics matrices for the states and controls formulation. The algorithm outline is documented in Alg. 10.1.

### 10.6.1 Barrier Method Formulation

The solution to the RPO SOCP is approached by implementing the barrier method for all second-order cone constraints. These second-order cone constraints are appended to the objective function, using an approximation of the indicator function for each constraint in the form of the log-barrier function [20]. The log-barrier function is shown in Eq. 3.35. This results in implementing the barrier method problem as

$$
\text{minimize} \quad f_{\eta}^T x_{\etaux} + 2 \sum_{i=1}^{3N} \phi_i(x_{\etaux})
$$

subject to $G_{\etaux} x_{\etaux} = h_{ux}$

where

$$
\phi_i(x_{\etaux}) = -\frac{1}{2} \log \left[ (c_i^T x_{\etaux} + d_i)^2 - x_{\etaux}^T A_i^T A_i x_{\etaux} \right]
$$

The Lagrangian for this problem, after multiplying the objective function by $a$, and with $\phi_i$ as defined above, is

$$
\mathcal{L}(x_{\etaux}, \nu) = af_{\eta}^T x_{\etaux} + 2 \sum_{i=1}^{3N} \phi_i(x_{\etaux}) + \nu^T (G_{\etaux} x_{\etaux} - h_{ux})
$$

The modified KKT that result from this Lagrangian are

$$
a f_{\eta} + 2 \sum_{i=1}^{3N} \nabla_{x_{\etaux}} \phi_i(x_{\etaux})^* + G_{\etaux}^T \nu^* = 0
$$

$$
G_{\etaux} x_{\etaux}^* = h_{ux}
$$

The minimization process is performed using the barrier method, and by solving the KKT conditions in Eq. 10.78 and 10.79. The objective function consists of the original
objective function, plus the contributions of the log-barrier functions for each second-order cone constraint, as shown in Eq. 10.75.

**Newton Step**

The Newton step equations, shown in Eq. 3.78, are used to solve the modified KKT conditions of optimality for the barrier method. From Eqs. 10.78 and 10.79, these are written as

\[
\begin{bmatrix}
2\nabla^2_{\eta_{ux}} \phi(x_{\eta_{ux}}) & G^T_{\eta_{ux}} \\
G_{\eta_{ux}} & 0
\end{bmatrix}
\begin{bmatrix}
\Delta x_{\eta_{ux}} \\
\nu
\end{bmatrix}
= -\begin{bmatrix}
af_{\eta} + 2\nabla_{\eta_{ux}} \phi(x_{\eta_{ux}}) \\
0
\end{bmatrix}
\]

where

\[
\phi(x_{\eta_{ux}}) = \sum_{i=1}^{3N} \phi_i(x_{\eta_{ux}})
\]

(10.80)

The gradient and Hessian of the log-barrier functions (\(\nabla_{x_{\eta_{ux}}} \phi(x_{\eta_{ux}})\), and \(\nabla^2_{x_{\eta_{ux}}} \phi(x_{\eta_{ux}})\)) are now defined. These are written specifically for the RPO problem using Eq. 3.33 and 3.34.

The conic variables, \(\bar{y}_i\) for \(i = 1, \ldots, 3N\) from Eq. 10.1 in the barrier method are now defined for clarity. This is done to simplify the notation for the gradients and Hessians, and also to save memory and reduce the number of operations within the code. The conic variables are written as a function of all variables in the controls-and-states formulation, \(x_{\eta_{ux}}\), as

\[
\bar{y}_i(x_{\eta_{ux}}) = \begin{bmatrix}
c_i^T x_{\eta_{ux}} + d_i \\
A_i x_{\eta_{ux}} + b_i
\end{bmatrix}
= \begin{cases}
\eta_i \\
\mu_i
\end{cases}
- \eta_i + \frac{\omega_{max}}{\omega^2}
- \bar{\bar{i}}_i^T M_r T^{-1}/ \cos(\alpha)
\begin{cases}
\hat{x}_j \\
M_r T^{-1}
\end{cases}
\]

for \(1 \leq i \leq N\)

for \(N + 1 \leq i \leq 2N, j = i - N\)

for \(2N + 1 \leq i \leq 3N, j = i - 2N\)

(10.81)

and the gradient of \(\bar{y}_i\) over all variables \(x_{\eta_{ux}}\) is
\[ \nabla_{x_{\eta u}} \bar{y}_i(x_{\eta u}) = \begin{bmatrix} c_i^T \\ A_i \end{bmatrix} = \begin{cases} \begin{bmatrix} M_{\eta}^R M_{\eta u}^R \\ M_{\eta u}^{i1} \\ M_{\eta u}^{i2} \\ M_{\eta u}^{i3} \\ M_{\eta u}^{i4} \end{bmatrix} & \text{for } 1 \leq i \leq N \\ -M_{\eta}^R M_{\eta u}^R & \text{for } N + 1 \leq i \leq 2N, \; j = i - N \\ \frac{1}{\cos(\alpha)} c_i^T M_r T^{-1} M_x^T M_{\eta u}^x & \text{for } 2N + 1 \leq i \leq 3N, \; j = i - 2N \end{cases} \]

Note that \( A_i \) for \( N + 1 \leq i \leq 2N \) does not exist, therefore only \( c_i \) is defined. Also note that the Hessian is zero, i.e., \( \nabla^2_{x_{\eta u}} \bar{y}_i = 0 \).

Making use of Eqs. 3.33 and 3.34, the gradients and Hessians of the log-barrier functions over \( \bar{y}_i \) are now written as a function of the primal variables \( x_{\eta u} \) for the controls-and-states formulation as

\[ \nabla_{\bar{y}_i} \phi_i(\bar{y}_i) = \begin{cases} \begin{bmatrix} \frac{1}{f_{\eta 1}} \eta_i \\ -\mu_i \end{bmatrix} & \text{for } 1 \leq i \leq N \\ \frac{-\eta_i + u_{\max}/\omega^2}{f_{\eta 2}} & \text{for } N + 1 \leq i \leq 2N, \; j = i - N \\ \frac{1}{f_{\eta 3}} J_c \bar{x}_j & \text{for } 2N + 1 \leq i \leq 3N, \; j = i - 2N \end{cases} \]  

(10.83)

\[ \nabla^2_{\bar{y}_i} \phi_i(\bar{y}_i) = \begin{cases} \begin{bmatrix} \frac{1}{f_{\eta 1}} \eta_i^2 + f_{\eta 1} & -2\eta_i \mu_i^T \\ -2\eta_i \mu_i & 2\mu_i \mu_i^T \end{bmatrix} & \text{for } 1 \leq i \leq N \\ \frac{1}{f_{\eta 2}} & \text{for } N + 1 \leq i \leq 2N, \; j = i - N \\ \frac{1}{f_{\eta 3}} \left( 2J_c \bar{x}_j \bar{x}_j^T J_c^T J - f_{\eta 3} J_c^T J_c \right) & \text{for } 2N + 1 \leq i \leq 3N, \; j = i - 2N \end{cases} \]  

(10.84)

where

\[ J_c = \begin{bmatrix} \frac{i^T M_r T^{-1}}{\cos(\alpha)} \\ M_r T^{-1} \end{bmatrix} \]  

(10.85)
and where the inequality constraints associated with the max. acceleration and slack variable are given by

\[ f_{i_{a1}} = \mu_i^T \mu_i - \eta_i^2 \] (10.86)

\[ f_{j_{a2}} = (\eta_j - \frac{u_{\text{max}}}{\omega^2})^2 \] (10.87)

and the inequality constraint associated with the approach corridor is given by

\[ f_{j_c} = x_j^T T^{-T} M_r^T M_r T^{-1} x_j - \frac{1}{\cos^2 \alpha} (i_c^T M_r T^{-1} x_j)^2 \] (10.88)

The second-order constraint log-barrier function gradients and Hessians over all variables \( x_{\eta,ux} \) are now conveniently written as

\[ \nabla_{x_{\eta,ux}} \phi(x_{\eta,ux}) = \sum_{i=1}^{3N} \nabla_{x_{\eta,ux}} \phi_i(x_{\eta,ux}) = \sum_{i=1}^{3N} [\nabla_{x_{\eta,ux}} \tilde{y}_i(x_{\eta,ux})] [\nabla_{\tilde{y}_i} \phi_i(\tilde{y}_i)] \] (10.89)

and for the Hessian, note that \( \nabla_{x_{\eta,ux}}^2 \tilde{y}_i(x_{\eta,ux}) = 0 \), so then

\[ \nabla_{x_{\eta,ux}}^2 \phi(x_{\eta,ux}) = \sum_{i=1}^{3N} \nabla_{x_{\eta,ux}}^2 \phi_i(x_{\eta,ux}) = \sum_{i=1}^{3N} [\nabla_{x_{\eta,ux}} \tilde{y}_i(x_{\eta,ux})] \nabla_{x_{\eta,ux}} [\nabla_{\tilde{y}_i} \phi_i(\tilde{y}_i)] \]

where \( \nabla_{x_{\eta,ux}} [\nabla_{\tilde{y}_i} \phi_i(\tilde{y}_i)] = [\nabla_{\tilde{y}_i}^2 \phi_i(\tilde{y}_i)] [\nabla_{x_{\eta,ux}} \tilde{y}_i(x_{\eta,ux})]^T \), leading to

\[ \nabla_{x_{\eta,ux}}^2 \phi(x_{\eta,ux}) = \sum_{i=1}^{3N} [\nabla_{x_{\eta,ux}} \tilde{y}_i(x_{\eta,ux})] [\nabla_{\tilde{y}_i}^2 \phi_i(\tilde{y}_i)] [\nabla_{x_{\eta,ux}} \tilde{y}_i(x_{\eta,ux})]^T \] (10.90)

where \( [\nabla_{x_{\eta,ux}} \tilde{y}_i(x_{\eta,ux})] \) is defined in Eq. 10.82 and \(( [\nabla_{\tilde{y}_i} \phi_i(\tilde{y}_i)], [\nabla_{\tilde{y}_i}^2 \phi_i(\tilde{y}_i)] \) are defined in Eq. 10.83 and 10.84.

### Backtracking Line Search

The backtracking line search for the barrier method consists of two main components. The first is to ensure that the step is within the feasible domain, and the second ensures
that the Newton step reduces the current modified objective function. The current iterates are all primal variables \( x_{\eta u} \), and the next iteration is denoted as \( x_{\eta u}^+ \). The Newton step is

\[
x_{\eta u}^+ = x_{\eta u} + s\Delta x_{\eta u}
\] (10.91)

where \( s \) is the step size. Backtracking starts with initializing the Newton step at \( s = 1 \), then shrinking the step by multiplying \( s \) by \( \beta \in (0, 1) \). This is first performed to ensure feasibility, where the step is decreased until \( f_i < 0 \) for \( i = 1, \ldots, 3N \).

Once a feasible step is determined, the objective is to continue to decrease the step size until the modified objective function is minimized. This is written as

\[
a^T f(x_{\eta u} + s\Delta x_{\eta u}) + 2 \sum_{i=1}^{3N} \phi_i(x_{\eta u} + s\Delta x_{\eta u}) \leq a^T f(x_{\eta u}) + 2 \sum_{i=1}^{3N} \phi_i(x_{\eta u}) + \alpha \zeta_d
\] (10.92)

where \( \alpha \) is a percent reduction of the residuals, and \( \zeta_d \) is the inner product of the optimality function gradient and the current primal variables Newton step. This term ensures that the residuals are minimized at least by a certain percentage, \( \alpha \).

\[
\zeta_d = s[a^T f + 2\nabla x_{\eta u} \phi(x_{\eta u})]^T \Delta x_{\eta u}
\] (10.93)

The value \( s \) is multiplied by \( \beta \) until Eq. 10.92 is satisfied. At this point, the Newton step is feasible, and the modified objective function is minimized in the search direction \( \Delta x_{\eta u} \). The new iterate of \( x_{\eta u} \) is \( x_{\eta u}^+ = x_{\eta u} + s\Delta x_{\eta u} \).
10.6.2 Barrier Method Pseudocode

Algorithm 10.1 Custom SOCP using Barrier Method for Autonomous RPO

Input:
Problem data, $G_{ux}$, $h_{ux}$, and $A_i$, $b_i$, $c_i$, $d_i$ for $i = 1, \ldots, m$ (Eqs. 10.12, 10.40 and 10.41-10.43)
Initial $x_{ux}$ satisfying $G_{ux}x_{ux} = h_{ux}$ (Eq. 10.58)
Initial centering step parameter†, $a = a^{(0)} > 0$
Centering scale factor†, $\mu > 1$
Desired tolerance, $\epsilon > 0$
Initial feasibility, $\epsilon_{feas} < 0$

Output:
Optimal primal variables, $x_{ux}^*$
Optimal dual variables, $\nu^*$
Total iterations $k$ and primal tolerance $\epsilon_p$

Infeasible start
1.) Select large enough $e$ to ensure $f_i(x_{ux}) < e$ for $i = 1, \ldots, m$
2.) Modify objective to $J = e$
3.) Solve feasible start under condition While $f_i(x_f) > \epsilon_{feas}$ for $i = 1, \ldots, m$

Feasible start
While $m/a > \epsilon$
1.) Centering step. Compute $x_{ux}^{(k+1)}(a)$ by
   I. Compute the Newton-step search direction $\Delta x_{ux}$. (Eq. 10.6.1)
      a.) Compute gradient and Hessian of KKT optimality functions.
      (See Alg. 10.2, using Eqs. 10.83, 10.84, 10.89, and 10.90)
      b.) Perform matrix inversion to determine search direction $\Delta x_{ux}$.
   II. Minimize $a_f^T x_{ux} + \phi$, subject to $G_{ux}x_{ux}^{(k)} = h_{ux}$, and starting at $x_{ux}^{(k)}$
      a.) Backtracking line search until step is feasible, $f_i(x_{ux}) \leq 0$ for $i = 1, \ldots, m$
      b.) Continue backtracking search until the mod. objective function is minimized.
      (See Alg. 10.3, using Eqs. 10.74 and 10.92)
   III. Take the Newton-step. Update, $x_{ux}^{(k+1)}(a) = x_{ux}^{(k)}(a) + s\Delta x_{ux}(a)$.
2.) Calculate primal tolerance, $\epsilon_p = ||a_f + \sum_{i=1}^m \nabla \phi_i(x_{ux}) + G_{ux}^T \nu||_2$
3.) Increase $a$, via $a = \mu a$
4.) Iteration update, $k = k + 1$
5.) Save data, $x_{ux}$, $s$, $a$, $\epsilon_p$

End

† More details on initialization parameters can be found in Convex Optimization [20].

In this analysis, a value of 10 for parameters $a^{(0)}$ and $\mu$ was used.
Algorithm 10.2 Barrier Method SOC Gradients and Hessians Function

**Input:**
- Function data, $A_i$, $b_i$, $c_i$, $d_i$ for $i = 1, \ldots, m$ (Eqs. 10.41-10.43)
- Current $x_{kux}^{(k)}$ iterate

**Output:**
- Current gradient and Hessian of barrier function, $[\nabla_{x_{kux}} \phi(x_{kux}^{(k)}), \nabla^2_{x_{kux}} \phi(x_{kux}^{(k)})]$

**Calculate gradient and Hessian**

1.) Initialize $\nabla_{x_{kux}} \phi(x_{kux}^{(k)})$ and $\nabla^2_{x_{kux}} \phi(x_{kux}^{(k)})$

2.) Loop through SOC functions:

   For $i = 1, \ldots, m$

   I. Calculate $\bar{y}_i$ from Eq. 10.81

   II. Compute the $i$th components of the gradient
       a.) Calculate $\nabla_{\bar{y}_i} \phi_i(\bar{y}_i)$ from Eq. 10.83
       b.) Calculate $\nabla_{x_{kux}} \phi_i(x_{kux})$ in terms of selection matrix indices Eq. 10.89

   III. Compute the $i$th components of the Hessian
       a.) Calculate $\nabla^2_{\bar{y}_i} \phi_i(\bar{y}_i)$ from Eq. 10.84
       b.) Calculate $\nabla^2_{x_{kux}} \phi_i(x_{kux})$ in terms of selection matrix indices Eq. 10.90

   IV. Compute gradient and Hessian sums with selection matrix indices from II. and III.
       a.) Update grad. $\nabla_{x_{kux}} \phi(x_{kux}) = \nabla_{x_{kux}} \phi(x_{kux}) + \nabla_{x_{kux}} \phi_i(x_{kux})$ in Eq. 10.89
       b.) Update Hess. $\nabla^2_{x_{kux}} \phi(x_{kux}) = \nabla^2_{x_{kux}} \phi(x_{kux}) + \nabla^2_{x_{kux}} \phi_i(x_{kux})$ in Eq. 10.90

End

Algorithm 10.3 Barrier Method Line Search

**Input:**
- Function data, $A_i$, $b_i$, $c_i$, $d_i$ for $i = 1, \ldots, m$ (Eqs. 10.41-10.43)
- Current $x_{kux}^{(k)}$ iterate

**Output:**
- Step size, $s$

**Calculate the step size via backtracking line search**

1.) Set $s = 1$

2.) Ensure the step is within the feasible domain:

While $f(x_{kux}) \geq 0$

   I. Compute all SOC functions, $\bar{f}^T(x_{kux}) = [f_1(x_{kux}), \ldots, f_m(x_{kux})]$ from Eq. 10.74.

   II. Reduce $s$, by $s = \beta s$.

End

3.) Ensure the step minimizes the mod. objective function (Eq. 10.92):

While $at^T_\eta(x_{kux} + s\Delta x_{kux}) + 2\sum_{i=1}^m \phi_i(x_{kux} + s\Delta x_{kux}) \leq at^T_\eta x_{kux} + 2\sum_{i=1}^m \phi_i(x_{kux}) + \alpha \zeta_d$

   I. Reduce $s$, by $s = \beta s$.

End
10.7 RPO Planning Algorithm Using Primal-dual Interior-point Method

In this section, the custom RPO trajectory planner algorithm is formulated using primal-dual interior point methods [17, 20, 126]. The algorithm outline is documented in Alg. 10.4. This requires solving the KKT conditions of optimality, as presented in Chapter 3. The solver method is written in terms of $x_{qusz}$, which represents a vector of all primal variables, (slack variables, states and controls) for the states and controls problem formulation. The matrix $G_{qusz}$ and vector $h_{ux}$ are given in Eq. 10.40 and 10.12, respectively. The primal-dual variables in this problem are denoted as vector $z_{pd}$, which includes $x_{qusz}$, $\lambda$, and $\nu$ where $x_{qusz} \in \mathbb{R}^{10N-6}$, $\lambda \in \mathbb{R}^{3N}$ ($m = 3N$ inequality constraints), and $\nu \in \mathbb{R}^{6N}$ ($p = 6N$ equality constraints).

10.7.1 Primal-dual Interior-point Method Formulation

The solution to the SOCP using primal-dual interior-point methods is approached by implementing the modified KKT equations that resulted from the implementation of the barrier method. The conditions of optimality for the primal-dual interior-point method are derived in Eqs. 3.68-3.70, and the resulting problem is formulated as

$$ f_{q} + \sum_{i=1}^{m} \lambda_{i}^{*} \nabla x_{qusz} f_{i}(x_{qusz}^{*}) + G_{qusz}^{T} \nu^{*} = 0 \quad (10.94) $$

$$ -\lambda_{i} f_{i}(x_{qusz}^{*}) = \frac{1}{a}, \text{ for } i = 1, \ldots, m \quad (10.95) $$

$$ G_{qusz} x_{qusz}^{*} = h_{ux} \quad (10.96) $$

The PDIP method, as opposed to the barrier method, now contains the conditions of optimality for complementary slackness. Therefore, the modified KKT condition in Eq. 10.95 remains in the problem variable space, and updates to all primal and dual variables are now made.

Newton Step

The Newton step used for the PDIP method is first formulated from the modified KKT
equations that resulted in Eq. 10.94-10.96. The modified KKT equations are written as the residual equations $r_t(x_{\eta u}, \lambda, \nu) = 0$, where

$$
\begin{bmatrix}
 r_{\text{dual}} \\
 r_{\text{cent}} \\
 r_{\text{pri}}
\end{bmatrix}
= \begin{bmatrix}
 f_t + [D\tilde{f}(x_{\eta u})]^T \lambda + G_{\eta u}^T \nu \\
 -\text{diag}(\lambda)\tilde{f}(x_{\eta u}) - \frac{1}{a} 1 \\
 G_{\eta u} x_{\eta u} - h_{\eta u}
\end{bmatrix}
$$

(10.97)

Functions $f(x_{\eta u})$ and their derivatives are defined as

$$
\tilde{f}(x_{\eta u}) = \begin{bmatrix}
 f_1(x_{\eta u}) \\
 \vdots \\
 f_{3N}(x_{\eta u})
\end{bmatrix}, \quad [D\tilde{f}(x_{\eta u})] = \begin{bmatrix}
 \nabla f_1(x_{\eta u})^T \\
 \vdots \\
 \nabla f_N(x_{\eta u})^T
\end{bmatrix}
$$

(10.98)

The values of $x_{\eta u}$, $\lambda$, and $\nu$ that satisfy $r_t(x_{\eta u}, \lambda, \nu) = 0$, results in the optimal primal variables, $x^*_{\eta u}$, and dual variables, $\lambda^*$, $\nu^*$, that satisfy the $n+m+p = (10N-6)+3N+6N = 19N-6$ nonlinear modified KKT equations. The primal-dual search direction is implemented using a Newton step to solve this system of nonlinear equations. Let the current point and the Newton step be defined as

$$
z = (x_{\eta u}, \lambda, \nu), \quad \Delta z = (\Delta x_{\eta u}, \Delta \lambda, \Delta \nu)
$$

(10.99)

respectively. Then, the Newton step is characterized by the first-order expansion of the nonlinear equations, resulting in

$$
r_t(z + \Delta z) \approx r_t(z) + [Dr_t(z)]\Delta z = 0
$$

(10.100)

Then, in terms of $x_{\eta u}$, $\lambda$, and $\nu$, we have

$$
\begin{bmatrix}
 \sum_{i=1}^{m} \lambda_i \nabla^2 f_i(x_{\eta u}) & [D\tilde{f}(x_{\eta u})]^T & G_{\eta u}^T \\
 -\text{diag}(\lambda)[D\tilde{f}(x_{\eta u})] & -\text{diag}(\tilde{f}(x_{\eta u})) & 0 \\
 G_{\eta u} & 0 & 0
\end{bmatrix}
\begin{bmatrix}
 \Delta x_{\eta u} \\
 \Delta \lambda \\
 \Delta \nu
\end{bmatrix}
= -\begin{bmatrix}
 r_{\text{dual}} \\
 r_{\text{cent}} \\
 r_{\text{pri}}
\end{bmatrix}
$$

(10.101)
The Newton step primal-dual search direction is solved for by

$$\Delta z = -[Dr_t(z)]^{-1}r_t(z) \quad (10.102)$$

Using Eq. 10.101, the variable $\Delta \lambda$ can be eliminated, from the second block of equations,

$$\Delta \lambda = -\text{diag}(f(x_{\etaux}))^{-1}\text{diag}[\lambda][D\bar{f}(x_{\etaux})]\Delta x_{\etaux} + \text{diag}[f(x_{\etaux})]^{-1}r_{\text{cent}} \quad (10.103)$$

and using this in the first block of equations leads to

$$\begin{bmatrix} H & G^T \\ G & 0 \end{bmatrix} \begin{bmatrix} \Delta x_{\etaux} \\ \Delta \nu \end{bmatrix} = -\begin{bmatrix} r_{\text{dual}} + [D\bar{f}(x_{\etaux})]^T \text{diag}(f(x_{\etaux}))^{-1}r_{\text{cent}} \\ r_{\text{pri}} \end{bmatrix} \quad (10.104)$$

where

$$H = \sum_{i=1}^{3N} \lambda_i \nabla^2 f_i(x_{\etaux}) + \sum_{i=1}^{3N} -\frac{\lambda_i}{f_i(x_{\etaux})} [\nabla x_{\etaux} f_i(x_{\etaux})][\nabla x_{\etaux} f_i(x_{\etaux})]^T \quad (10.105)$$

The gradient and Hessian for the second-order cone functions in the convex RPO planner problem are written using the result from Chapter 3, shown in Eq. 3.30. These are defined over the conic variables $\bar{y}_i$, from in Eq. 10.81, and written in terms of the states-and-controls formulation variables, $x_{\etaux}$.

$$\nabla_{\bar{y}_i} f_i(\bar{y}_i) = \begin{cases} 2 \begin{bmatrix} -\eta_i \\ \mu_i \end{bmatrix} & \text{for } 1 \leq i \leq N \\ -2(-\eta_j + u_{\text{max}}/\omega^2) & \text{for } N + 1 \leq i \leq 2N, j = i - N \\ 2 \begin{bmatrix} -\hat{t}_c & r \cos(\alpha) \\ M_r T^{-1} \end{bmatrix} & \text{for } 2N + 1 \leq i \leq 3N, j = i - 2N \end{cases} \quad (10.106)$$
\[ \nabla_{x_i} f_i(x_{\eta u}) = \begin{cases} \begin{bmatrix} -1 & 0 \\ 2 \end{bmatrix} & \text{for } 1 \leq i \leq N \\ 2 & \text{for } N + 1 \leq i \leq 2N \\ -2 & \text{for } 2N + 1 \leq i \leq 3N \end{cases} \] (10.107)

\[ \nabla_{x_i} f_i(x_{\eta u}) = \begin{bmatrix} -1 & 0 \\ 2 \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & I_{3 \times 3} \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 2 \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & I_{6 \times 6} \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 2 \end{bmatrix} T, \text{ for } i = 1, \ldots, 3N \] (10.108)

The gradients and Hessians of the second-order cone constraints over all variables \( x_{\eta u} \) are now conveniently written as

The backtracking line search is implemented as that of a backtracking line search method, based on the residual, and modified to ensure that \( \lambda > 0 \) and \( \bar{f}(x_{\eta u}) < 0 \). The current iterates are \( x_{\eta u}, \lambda, \) and \( \nu \), and the next iterate is denoted as \( x^+_{\eta u}, \lambda^+, \) and \( \nu^+ \). The residual at \( z^+ \) is denoted by \( r^+_i \).

First, the largest step size for \( \lambda^+ > 0 \), not exceeding 1 is [20]

\[ s_{\text{max}} = \min \{1, \min(-\lambda_i/\Delta \lambda_i, |\Delta \lambda_i| < 0)\} \] (10.111)

Then, backtracking starts with \( s = 0.99s_{\text{max}} \), and multiplies \( s \) by \( \beta \in (0, 1) \) until \( \bar{f}(x^+_{\eta u}) < 0 \) [20].
The next step, once a feasible solution is guaranteed, is to minimize the residuals 
\( r_t(x^{+ \tau}, \lambda^+, \nu^+) \). This is done by continuing to multiply \( s \) by \( \beta \) until

\[
||r_t(x^{+ \tau}, \lambda^+, \nu^+)||_2 \leq (1 - \alpha s)||r_t(x^{\tau}, \lambda, \nu)||_2
\]

(10.112)
given a desired percent reduction \( \alpha \). Typical choices for \( \alpha \) are between 0.01 and 0.1, while choices for \( \beta \) are typically between 0.3 and 0.8.

**Surrogate Duality Gap**

The surrogate duality gap between iterates of the interior-point algorithm is defined as

\[
\hat{\eta}(x^{\tau}, \lambda) = -\tilde{f}(x^{\tau})^T \lambda
\]

(10.113)

It must be ensured that the surrogate duality gap converges, as the algorithm progresses, so that \( r_{\text{pri}} = 0 \) and \( r_{\text{dual}} = 0 \).

**Infeasible Start**

The infeasible start for the primal-dual interior point method is the exact same approach as in the barrier method, which requires solving the initial feasibility problem shown in Eq. 10.63. The additional variables in this problem are \( \lambda \), which are feasible provided that the lambdas are positive, i.e. \( \lambda \succ 0 \).
### 10.7.2 Primal-dual Interior Point Method Pseudocode

**Algorithm 10.4** Custom SOCP using PDIP Method for Autonomous RPO

**Input:**
- Problem data, \( G_{\etaux}, h_{\etaux} \), and \( A_i, b_i, c_i, d_i \) for \( i = 1, \ldots, m \) (Eqs. 10.12, 10.40 and 10.41-10.43)
- Initial \( x_{\etaux} \) satisfying \( G_{\etaux} x_{\etaux} = h_{\etaux} \) (Eq. 10.58)
- Initial feasible dual variables, \( \lambda > 0 \)
- Centering scale factor \( \mu > 1 \)
- Desired primal-dual and surrogate duality tolerance, \( \epsilon_{\text{tol}} > 0, \epsilon > 0 \)
- Initial feasibility, \( \epsilon_{\text{feas}} < 0 \)

**Output:**
- Optimal primal variables, \( x_{\etaux}^* \)
- Optimal dual variables, \( (\lambda^*, \nu^*) \)
- Residuals and duality gap, \( ||r_{\text{pri}}||, ||r_{\text{dual}}||, \hat{\eta} \)

**Infeasible start**
1.) Select large enough \( e \) to ensure \( f_i(x_{\etaux}) < e \) for \( i = 1, \ldots, m \)
2.) Modify objective to \( J = e \)
3.) Solve feasible start under condition While \( f_i(x_{\etaux}) > \epsilon_{\text{feas}} \) for \( i = 1, \ldots, m \)

**Feasible start**

While \( ||r_{\text{pri}}||_2 > \epsilon_{\text{tol}}, ||r_{\text{dual}}||_2 > \epsilon_{\text{tol}}, \text{ and } \hat{\eta} > \epsilon \)
1.) Determine \( a \). Set \( a = 3\mu N/\hat{\eta} \).
2.) Solve the primal-dual equations at \( z_{pd} \), and minimize residuals using Newton’s method

**I.** Compute the primal-dual Newton-step search direction \( \Delta z_{pd} \). (Eq. 10.101)
   a.) Compute the gradient and Hessian of KKT optimality functions.
   (See Alg. 10.5, using Eqs. 10.106, 10.107, 10.108, and 10.109)
   b.) Perform matrix inversion to find \( (\Delta x_{\etaux}, \Delta \nu) \), and compute the \( \lambda \) step, \( \Delta \lambda \).

**II.** Determine step size via backtracking line search.
   a.) Calculate the minimum feasible step such that \( \lambda^+ > 0 \).
   b.) Continue backtracking to minimize the residuals, \( r_t \).
   (See Alg. 10.6, using Eqs. 10.97, 10.98, 10.111, and 10.112)

**III.** Take the Newton-step. Update \( z_{pd}^{(k+1)} = z_{pd}^{(k)} + s\Delta z_{pd} \).
3.) Calculate the residuals, \( r_{\text{dual}}, r_{\text{cent}}, \text{ and } r_{\text{pri}} \). (Eq. 10.97)
4.) Calculate the surrogate duality gap, \( \hat{\eta} \). (Eq. 10.113)
5.) Iteration update, \( k = k + 1 \)
6.) Save data, \( z_{pd}, s, ||r_{\text{dual}}||, ||r_{\text{cent}}||, ||r_{\text{pri}}||, \hat{\eta} \)

End

† More details on initialization parameters can be found in *Convex Optimization* [20].

In this analysis, a value of 10 for the parameter \( \mu \) was used.
Algorithm 10.5 PDIP Method SOC Gradients and Hessians Function

Input:
Function data, $A_i$, $b_i$, $c_i$, $d_i$ for $i = 1, \ldots, m$ (Eqs. 10.41-10.43)
Current $x^{(k)}_{\etaux}$ iterate

Output:
Current PDIP gradient and Hessian, $\{\bar{Df}(x_{\etaux}), \bar{D^2f}(x_{\etaux})\}$

Calculate gradient and Hessian
1.) Initialize $\{\bar{Df}(x_{\etaux})\}$ and $\{\bar{D^2f}(x_{\etaux})\}$
2.) Loop through SOC functions:
For $i = 1, \ldots, m$
   I. Calculate $\bar{y}_i$ from Eq. 10.81
   II. Compute the $i$th components of the gradient
      a.) Calculate $\nabla_{\bar{y}_i} f_i(\bar{y}_i)$ from Eq. 10.106
      b.) Calculate $\nabla_{x_{\etaux}} f_i(x_{\etaux})$ in terms of selection matrix indices Eq. 10.108
   III. Compute the $i$th components of the Hessian
      a.) Calculate $\nabla^2_{\bar{y}_i} f_i(\bar{y}_i)$ from Eq. 10.107
      b.) Calculate $\nabla^2_{x_{\etaux}} f_i(x_{\etaux})$ in terms of selection matrix indices Eq. 10.109
   IV. Compute gradient and Hessian sums with selection matrix indices from II. and III.
      a.) Update grad. $\{\bar{Df}(x_{\etaux})\} = \{\bar{Df}(x_{\etaux})\} + \nabla_{x_{\etaux}} f_i(x_{\etaux})$ in Eq. 10.108
      b.) Update Hess. $\{\bar{D^2f}(x_{\etaux})\} = \{\bar{D^2f}(x_{\etaux})\} + \nabla^2_{x_{\etaux}} f_i(x_{\etaux})$ in Eq. 10.109
End

Algorithm 10.6 PDIP Method Line Search

Input:
Function data, $A_i$, $b_i$, $c_i$, $d_i$ for $i = 1, \ldots, m$ (Eqs. 10.41-10.43)
Current $x^{(k)}_{\etaux}$ iterate

Output:
Step size, $s$

Calculate the step size via backtracking line search
1.) Select maximum $s$, via $s_{\max} = \min \{1, \min(-\lambda_i/\Delta \lambda_i \mid \Delta \lambda_i < 0)\}$, Eq. 10.111
2.) Ensure the step is within the feasible domain:
While $f(x_{\etaux}) \geq 0$
   I. Compute all SOC functions, $\bar{f}^T(x_{\etaux}) = [f_1(x_{\etaux}), \ldots, f_m(x_{\etaux})]$ from Eq. 10.98.
   II. Reduce $s$, by $s = \beta s$.
End
3.) Minimize the residual, $r_t(x_{\etaux}^+, \lambda^+, \nu^+)$ (Eqs. 10.97 and 10.112):
While $\|r_t(x_{\etaux}^+, \lambda^+, \nu^+\|_2 \leq (1 - \alpha s)\|r_t(x_{\etaux}, \lambda, \nu)\|_2$
   I. Reduce $s$, by $s = \beta s$.
CHAPTER 11
RPO PLANNER CPU AND MEMORY REQUIREMENTS

Several significant elements are used to test the RPO trajectory planning algorithm, and each is described in this chapter. This begins by first characterizing the varieties of currently available hardware, with an emphasis on hardware that has real-time embedding capabilities. The components specifically tested in these results are outlined. This is followed by a discussion which covers the great variety of software that is applicable to solving problems formulated as an SOCP. All software used in the CPU and memory requirement analysis is described in detail.

The most important metrics are initially defined, where the primary consideration is solving the optimal RPO trajectory planning algorithm on-board a satellite and in real-time. The results sections are presented using tables of collected data, which include CPU timing requirements for a variety of CPUs, and memory requirements for several different problems. A number of RPO planning problems are analyzed, which consist of formulations presented in previous chapters. These problems vary in size (number of variables), formulation method, number of constraints, and complexity. The essence of each problem is captured by presenting the costs and benefits of the timing and memory required.

11.1 Hardware

Given the simplicity and efficiency of convex optimization algorithms, the hardware capabilities required to run these programs is not extraordinary. Convex optimization algorithms can be run on simple system-on-a-chip platforms, such as a Raspberry Pi. The main components required are the CPU, storage for program data, and memory storage. Complexities arise, however, once a space system is considered. Typically spacecraft data processing systems require low power, volume, and mass, while maintaining high reliability and tolerance to the space environment [145]. Space systems also frequently implement em-
bedded systems with build-in processors. These provide additional real-time data processing capabilities, with operations occurring at or very near the processor clock speed [145].

The most common types of processing architectures used are CPUs (consisting of a single, or master processor) or distributed systems with multiple processors. A CPU may be tasked with multiple processes at once, which are either completed consecutively or according to resource management laws. Distributed systems have the option to dedicate one processor to a specific task, while the master processor is coordinating the information it sends and receives. Distributed processing conducts this appropriation of resources according to the current mission phase, hardware availability, and subsystem failures, thereby making these systems highly fault-tolerant [146]. For many applications, the advanced field-programmable gate array (FPGA) may be preferred, which allows for more flexibility in digital computation and extends the capabilities of microprocessors. FPGA architecture is composed of a system of logic blocks and hard blocks, which may be custom-tailored for a specific application. Though FPGAs may be favored by customers in many instances, the implementation of convex optimization algorithms on FPGAs is left for future work. The focus of this analysis is to test these algorithms on a few different CPUs which are well-known and widely available.

Memory availability may be considered one of the most valued resources for a spacecraft computer system [145]. The memory required on a spacecraft is separated into read-only memory (ROM), random-access memory (RAM), and special purpose (cache). ROM is long-term storage, generally does not require power to store data, and stores programs and other software. RAM is more volatile, requiring power at all times, and is accessed considerably faster than ROM. In many cases RAM is stringently allocated to different mission processes, so as to not exceed the necessary allotment. Processes on-board a satellite are regulated in this manner, so that all other systems may operate nominally, and without interference. Because of this, it is important to determine the amount of RAM required for the optimal RPO trajectory planning algorithms.

The hardware considered in this analysis includes three different computers, each with different CPU and RAM accessories. These encompass a wide scope of current technology.
The first selection is a J-board, with system-on-a-chip architecture, used to resemble the capabilities of a small satellite. The J-board has a 1.3 GHz Intel Atom, with 1 GB DDR3 and 16 GB flash memory. It also allows easy implementation of an embedded system, and has low-energy requirements, making it an ideal comparison to current small satellite capabilities. The mid- and high-tier computers tested are a MacBook laptop and a custom desktop, respectively. The Macbook Pro has a 2.6 GHz processor with 2.7 GB of RAM, and the desktop has a 4.0 GHz processor with 16 GB of RAM. A table of these statistics, along with operating system (OS), is provided in Table 11.1.

![Table 11.1: Computers tested, showing processor and memory](image)

<table>
<thead>
<tr>
<th>Computer</th>
<th>OS</th>
<th>CPU</th>
<th>Speed (GHz)</th>
<th>RAM (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>J-board</td>
<td>Ubuntu 15 64-bit</td>
<td>Intel Atom Z3735G (4 cores)</td>
<td>1.3</td>
<td>1.0</td>
</tr>
<tr>
<td>Macbook</td>
<td>Windows 7 32-bit</td>
<td>Intel Core 2 Duo (2 cores)</td>
<td>2.6</td>
<td>2.7</td>
</tr>
<tr>
<td>Desktop</td>
<td>Windows 10 64-bit</td>
<td>AMD FX-8350 (8 cores)</td>
<td>4.0</td>
<td>16</td>
</tr>
</tbody>
</table>

### 11.2 Software

The software considered in this study ranges from the most general convex program solvers, to conic solvers, and to less general second-order cone solvers. The majority of convex optimization software that implement primal-dual interior points methods accept the most general SDP format, and simplify the problem as needed [17, 147]. This includes implementing a conic programming format, specific to cone programs which are efficiently solved by a suite of conic optimization software [148]. Several industrial-grade commercial solvers exist for evaluation or purchase and have been tested in the case of solving the RPO trajectory planner problem. Most of these solvers accept any general convex program, but their solution methods narrow down the problem into the most efficient form to be handled by the solver using a disciplined mathematical program format. Therefore, these currently available programs prove to efficiently solve SOCPs as well, as shown by results from solver comparisons [121]. The two primary commercial solvers tested here are MOSEK [133, 149] and Gurobi [134]. From the results of previous solver comparisons, it is shown that MOSEK
outperforms Gurobi when timed for the portfolio optimization problem examined in the ECOS paper [121].

Several other commercial and non-commercial conic optimization solvers were also tested. These include CVXOPT [126], CPLEX [114], LOQO [122], SDPT3 [150], SeDuMi [151], and ECOS [121]. Though the majority of these solvers are implemented via the MatLab CVX interface, the non-commercial solvers also have source code available online, and may be compiled and tested independently [152]. These include CVXOPT (Python), SDPT3 (MatLab/C), and ECOS (C). Though a few of these were examined, the focus of this analysis is to collect algorithm CPU and memory requirements. For this reason, the commercial solvers are preferred due to their superior efficiency, documentation, and customer support.

Of the solvers listed, five main solvers are examined and applied to the RPO trajectory planning problem. These include SDPT3, SeDuMi, ECOS, MOSEK, and Gurobi. The SDPT3 solver (CVX default solver) is a MatLab-based software package for semidefinite-quadratic-linear programming. It solves conic programming problems using an infeasible primal-dual predictor-corrector path-following method [153]. The SeDuMi software package is similar to SDPT3, in that it is MatLab-based and solves convex optimization problems with linear equations and inequalities, second-order cone constraints, and semidefinite constraints. Both SDPT3 and SeDuMi provide convenient MatLab scripts that formulate convex problems, but ultimately call a number of key subroutines using Mex files. The ECOS solver takes advantage of an extended self-dual embedding technique, and is applied specifically to second-order cone problems. It is numerically robust, has a tiny footprint, and is capable of being implemented on embedded platforms [121]. The remaining solvers are commercial, including MOSEK and Gurobi. These both solve a variety of problems, including non-convex mixed-integer programs. More information on each of these solvers can be found in their documentation.

The problem formats, factorization methods, implementation, convergence measures, and terminating procedures vary greatly between each different solver considered [111].
Therefore, it is difficult to suggest a means for lateral comparison between each algorithm when it comes to benchmarking. Even the similar aspects of each solver can not be directly compared without a formal analysis of the documentation and code. Therefore, it is sufficient in this research to compare, first and foremost, the algorithm’s computation time. This is determined by each algorithm individually, and is printed in the output. Many algorithms include initial setup time, solution time, and final writing time as well. The timing considered for algorithm testing is that of the actual solution time, (i.e., once the problem is set up, the time required to return the optimal solution). Carrying out this examination on the hardware of choice requires a testing method that provides the most fair environment to each algorithm. Therefore, CVX MatLab interface is used on the Macbook, with no user programs (excepting MatLab) and minimal other processes running in the background.

In addition to this, the number of total iterations is included. Though this may equate to a vastly different number of total computations for each algorithm, in general, the total number of iterations represents how many Newton-steps (matrix inversions) and backtracking line searches were performed to reach the optimal solution. Correlation between solution time and number of iterations is evident in the results, except in the case of the Gurobi solver, which has a minimum number of iterations (50) that are required for conic problems. Upon inspection of the Gurobi solver output, it can be seen that the number of iterations where actual progress is made is similar to that of MOSEK, therefore explaining the difference in number of iterations between the two solvers when compared to the solution times.

### 11.3 Solver Comparison

From the solver options provided, an initial analysis is performed using the MacBook, to determine which solver to select for the CPU and memory data collection process. This is done by considering the five main solvers (each implemented using CVX) which are used to solve different RPO trajectory planning problems, and collect timing and total iteration data. The five solvers tested include SDPT3, SeDuMi, ECOS, MOSEK, and Gurobi. These solvers are tested for both the LROE controls-and-states (LROE U-X) and the LROE control-only (LROE U-only) formulations of the RPO trajectory planning problem, including the option...
to implement an approach corridor \((w/\text{corridor} \text{ or } w/o \text{corridor})\). Two different problem sizes are examined, for \(N = 100\) and \(N = 200\) discretization points. Therefore, a total of eight problems are developed for each solver. These each have an associated number of scalar variables, equality constraints (Eq. Const.), and inequality constraints (Ineq. Const.). All eight problems considered in this analysis are outlined in Table 11.2.

Table 11.2: Problem formulations tested for solver comparison

<table>
<thead>
<tr>
<th>Problem</th>
<th>Formulation</th>
<th>(N)</th>
<th>Variables</th>
<th>Eq. Const.</th>
<th>Ineq. Const.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LR OEG U-X</td>
<td>100</td>
<td>1000</td>
<td>600</td>
<td>200</td>
</tr>
<tr>
<td>2</td>
<td>w/o corridor</td>
<td>200</td>
<td>2000</td>
<td>1200</td>
<td>400</td>
</tr>
<tr>
<td>3</td>
<td>LR OEG U-X</td>
<td>100</td>
<td>1000</td>
<td>600</td>
<td>300</td>
</tr>
<tr>
<td>4</td>
<td>w/corridor</td>
<td>200</td>
<td>2000</td>
<td>1200</td>
<td>600</td>
</tr>
<tr>
<td>5</td>
<td>LR OEG U-only</td>
<td>100</td>
<td>400</td>
<td>6</td>
<td>200</td>
</tr>
<tr>
<td>6</td>
<td>w/o corridor</td>
<td>200</td>
<td>800</td>
<td>6</td>
<td>400</td>
</tr>
<tr>
<td>7</td>
<td>LR OEG U-only</td>
<td>100</td>
<td>400</td>
<td>6</td>
<td>300</td>
</tr>
<tr>
<td>8</td>
<td>w/corridor</td>
<td>200</td>
<td>800</td>
<td>6</td>
<td>600</td>
</tr>
</tbody>
</table>

Each problem was run a total of ten times, and an average solution time was collected (though it is noted that the solution time only varied to within 1-5%). For the Gurobi solver, the minimum number of iterations (50) is denoted by the asterisk (*), and for ECOS the maximum number of iterations is 100. The instances where the solver failed to find a solution are shown by an ‘F’. All results collected are shown in Tables 11.3-11.6.

Table 11.3: Solvers tested for problems 1 & 2, controls-and-states formulation of RPO problem without approach corridor, on Macbook

<table>
<thead>
<tr>
<th>Solver</th>
<th>Sol. Time (s)</th>
<th>Iterations</th>
<th>Sol. Time (s)</th>
<th>Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDPT3</td>
<td>1.12</td>
<td>43</td>
<td>1.42</td>
<td>48</td>
</tr>
<tr>
<td>SeDuMi</td>
<td>0.32</td>
<td>13</td>
<td>0.41</td>
<td>14</td>
</tr>
<tr>
<td>ECOS</td>
<td>0.34</td>
<td>21</td>
<td>1.58</td>
<td>100 (max)</td>
</tr>
<tr>
<td>MOSEK</td>
<td>0.23</td>
<td>13</td>
<td>0.25</td>
<td>14</td>
</tr>
<tr>
<td>Gurobi</td>
<td>0.31</td>
<td>51*</td>
<td>0.47</td>
<td>51*</td>
</tr>
</tbody>
</table>
Table 11.4: Solvers tested for problems 3 & 4, controls-and-states formulation of RPO problem with approach corridor, on Macbook

<table>
<thead>
<tr>
<th>Solver</th>
<th>$N = 100$</th>
<th>$N = 200$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sol. Time (s)</td>
<td>Iterations</td>
</tr>
<tr>
<td>SDPT3</td>
<td>1.36</td>
<td>51</td>
</tr>
<tr>
<td>SeDuMi</td>
<td>0.63</td>
<td>20</td>
</tr>
<tr>
<td>ECOS</td>
<td>1.44</td>
<td>91</td>
</tr>
<tr>
<td>MOSEK</td>
<td>0.31</td>
<td>22</td>
</tr>
<tr>
<td>Gurobi</td>
<td>0.39</td>
<td>68</td>
</tr>
</tbody>
</table>

Table 11.5: Solvers tested for problems 5 & 6, control-only formulation of RPO problem without approach corridor, on Macbook

<table>
<thead>
<tr>
<th>Solver</th>
<th>$N = 100$</th>
<th>$N = 200$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sol. Time (s)</td>
<td>Iterations</td>
</tr>
<tr>
<td>SDPT3</td>
<td>0.67</td>
<td>19</td>
</tr>
<tr>
<td>SeDuMi</td>
<td>0.27</td>
<td>11</td>
</tr>
<tr>
<td>ECOS</td>
<td>0.43</td>
<td>27</td>
</tr>
<tr>
<td>MOSEK</td>
<td>0.23</td>
<td>13</td>
</tr>
<tr>
<td>Gurobi</td>
<td>0.16</td>
<td>51*</td>
</tr>
</tbody>
</table>

Table 11.6: Solvers tested for problems 7 & 8, control-only formulation of RPO problem with approach corridor, on Macbook

<table>
<thead>
<tr>
<th>Solver</th>
<th>$N = 100$</th>
<th>$N = 200$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sol. Time (s)</td>
<td>Iterations</td>
</tr>
<tr>
<td>SDPT3</td>
<td>4.30</td>
<td>44</td>
</tr>
<tr>
<td>SeDuMi</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>ECOS</td>
<td>0.68</td>
<td>21</td>
</tr>
<tr>
<td>MOSEK</td>
<td>0.50</td>
<td>25</td>
</tr>
<tr>
<td>Gurobi</td>
<td>7.82</td>
<td>51*</td>
</tr>
</tbody>
</table>

From this analysis, it is clear that there are significant differences between each solver, in both solution time and number of iterations. Between the two problem formulations, the
MOSEK solver performed the best; MOSEK provided a solution in all cases, while requiring the least amount of time, and fewest iterations for most problems. The case where Gurobi outperforms MOSEK is for the control-only formulation without the approach corridor, although this difference is not significant.

It is also evident that there are advantages and disadvantages to both the controls-and-states and control-only formulations. In the cases where the approach corridor is excluded from the problem, the control-only formulation outperforms the controls-and-states formulation. This is likely due to the fact that there are significantly fewer equality constraints, compared to the states-and-controls formulation. However, the majority of the solvers tested have difficulties implementing the approach corridor constraint in the control-only formulation, due to its complexity. This is shown by the failures in Table 11.6. It is clear, nonetheless, that both MOSEK and ECOS still prevail here.

From these results, the MOSEK solver is chosen to perform a more in-depth analysis for RPO trajectory planning. In the following results, MOSEK will be used to solve a great number of different problems (each with different formulations), on all of the hardware varieties shown in Table 11.1.

11.4 Important Metrics

The remainder of this chapter focuses on collecting data for the important algorithm metrics. This is done entirely using the MOSEK solver’s interface and output, as well as one other program which determines the total memory consumption [149]. The most important metrics include the CPU requirements and memory consumption of the RPO planning problem. In all cases to follow, the MOSEK solver is operated using the command line application program interface in either Windows or Linux. A real-time flag is incorporated before each call so that maximum available CPU resources are allocated to run the program. This is followed by ‘mosek’ and the problem data file. Each problem is formulated using a .mps (mathematical programming system) file format, which interprets all problem data in a disciplined format accepted by MOSEK. An example command to solve a problem using MOSEK is ‘/k /realtime mosek prob_1.mps’. The data required for analyzing CPU and
memory requirements is output upon termination.

A variety of data is collected to determine the CPU speed requirements for solving an RPO trajectory planning problem. One of the most important values is the total time required by the MOSEK solver to formulate and solve the problem, then output the problem data. This is provided by MOSEK upon termination of the program. As mentioned previously, the total iterations is significant, as it provides a measure of the required Newton-steps for optimality. This is closely associated with the number of floating-point operations per iteration, for factorization of the Newton system matrix, which is also provided by MOSEK. The total number of floating-point operations due to all Newton-step factorizations can therefore be determined, and is an approximation of the cost of all iterations. Using this in conjunction with the total solution time gives an approximation of the floating-point operations per second (FLOPS). The CPU total time, iterations, and FLOPS are all used to determine the requirements on the CPU.

The second metric is the memory (RAM) requirement. To determine the total (heap) memory used by MOSEK, a side program called valgrind is implemented. Valgrind is a multipurpose tool which can be used as a profiler to accurately regulate the amount of memory used by a program. Valgrind also has a tool called massif that may be used to illustrate the dynamics of the memory allocation by the program, over the run time. The valgrind profiler is operated alongside MOSEK using a command line call: first to valgrind and then to MOSEK. Valgrind generates results which show the memory allocated by the main process, as well as all other processes/subroutines, as a function of the number of instructions [154]. The end result is an accurate measure of how much memory is required to solve a specific RPO trajectory planning problem, in addition to the total number of instructions required. The number of discretization points (problem size), number of equality/inequality constraints, and problem formulation (controls-and-state or control-only), is shown to greatly affect the total required heap memory used by MOSEK.

The following sections present the collected data for CPU and memory requirements. This data is collected directly from MOSEK, which is run on each of the three computers
shown in Table 11.1. A number of problems are outlined; each have different problem formulations, problem sizes, and constraints. Each of these problems is solved using MOSEK from the command line prompt in either Linux or Windows. During testing, MOSEK is called with the real-time flag, and is the only user program running, with minimal background processes.

**RPO Scenarios Tested**

Two main scenarios are considered here. The first, Scenario 1, is an approach scenario from 5 km away to 100 m away from the chief without an approach corridor. This includes linear dynamics equality constraints and second-order cone constraints for the control acceleration magnitude. The control acceleration maximum value is set to 1 cm/s² and is reached at several points along the trajectory. The second, Scenario 2, is an approach scenario like the first, from 5 km away to 100 m away from the chief, but includes an approach corridor with a 15-degree half-angle. The control acceleration maximum value is the same as in Scenario 1. This problem has linear dynamics equality constraints and second-order cone constraints for both the approach corridor and the control acceleration magnitudes.

Three different formulations for these scenarios are tested. These include the traditional Cartesian controls-and-states (Cart. U-X), the LROE controls-and-states (LROE X-U), and the LROE control-only (LROE U-only) formulations. For each problem, the number of discretization steps is varied as well. The number of steps tested here include 100, 200, 400, and 800 points. Each of the two scenarios are shown in the following tables, Table 11.7 for Scenario 1 and Table 11.8 for Scenario 2. Each table includes the three different formulations and four different number of points in the discretization, for a total of 12 different problems for Scenario 1 and Scenario 2.
Table 11.7: Scenario 1, without approach corridor, problems tested for CPU and memory requirements

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Cart.</td>
<td>100</td>
<td>1000</td>
<td>600</td>
<td>200</td>
</tr>
<tr>
<td>2</td>
<td>Cart.</td>
<td>200</td>
<td>2000</td>
<td>1200</td>
<td>400</td>
</tr>
<tr>
<td>3</td>
<td>U-X</td>
<td>400</td>
<td>4000</td>
<td>2400</td>
<td>800</td>
</tr>
<tr>
<td>4</td>
<td>LR OE</td>
<td>800</td>
<td>8000</td>
<td>4800</td>
<td>1600</td>
</tr>
<tr>
<td>5</td>
<td>LR OE</td>
<td>100</td>
<td>1000</td>
<td>600</td>
<td>200</td>
</tr>
<tr>
<td>6</td>
<td>LR OE</td>
<td>200</td>
<td>2000</td>
<td>1200</td>
<td>400</td>
</tr>
<tr>
<td>7</td>
<td>U-X</td>
<td>400</td>
<td>4000</td>
<td>2400</td>
<td>800</td>
</tr>
<tr>
<td>8</td>
<td>LR OE</td>
<td>800</td>
<td>8000</td>
<td>4800</td>
<td>1600</td>
</tr>
<tr>
<td>9</td>
<td>U-only</td>
<td>100</td>
<td>400</td>
<td>6</td>
<td>200</td>
</tr>
<tr>
<td>10</td>
<td>LR OE</td>
<td>200</td>
<td>800</td>
<td>6</td>
<td>400</td>
</tr>
<tr>
<td>11</td>
<td>U-only</td>
<td>400</td>
<td>1600</td>
<td>6</td>
<td>800</td>
</tr>
<tr>
<td>12</td>
<td>U-only</td>
<td>800</td>
<td>3200</td>
<td>6</td>
<td>1600</td>
</tr>
</tbody>
</table>

Table 11.8: Scenario 2, with approach corridor, problems tested for CPU and memory requirements

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Cart.</td>
<td>100</td>
<td>1000</td>
<td>600</td>
<td>300</td>
</tr>
<tr>
<td>2</td>
<td>Cart.</td>
<td>200</td>
<td>2000</td>
<td>1200</td>
<td>600</td>
</tr>
<tr>
<td>3</td>
<td>U-X</td>
<td>400</td>
<td>4000</td>
<td>2400</td>
<td>1200</td>
</tr>
<tr>
<td>4</td>
<td>U-X</td>
<td>800</td>
<td>8000</td>
<td>4800</td>
<td>2400</td>
</tr>
<tr>
<td>5</td>
<td>LR OE</td>
<td>100</td>
<td>1000</td>
<td>600</td>
<td>300</td>
</tr>
<tr>
<td>6</td>
<td>LR OE</td>
<td>200</td>
<td>2000</td>
<td>1200</td>
<td>600</td>
</tr>
<tr>
<td>7</td>
<td>U-X</td>
<td>400</td>
<td>4000</td>
<td>2400</td>
<td>1200</td>
</tr>
<tr>
<td>8</td>
<td>U-X</td>
<td>800</td>
<td>8000</td>
<td>4800</td>
<td>2400</td>
</tr>
<tr>
<td>9</td>
<td>LR OE</td>
<td>100</td>
<td>400</td>
<td>6</td>
<td>300</td>
</tr>
<tr>
<td>10</td>
<td>LR OE</td>
<td>200</td>
<td>800</td>
<td>6</td>
<td>600</td>
</tr>
<tr>
<td>11</td>
<td>U-only</td>
<td>400</td>
<td>1600</td>
<td>6</td>
<td>1200</td>
</tr>
<tr>
<td>12</td>
<td>U-only</td>
<td>800</td>
<td>3200</td>
<td>6</td>
<td>2400</td>
</tr>
</tbody>
</table>
11.5 CPU Requirements Tables

The resulting CPU requirements for the scenarios and problems tested are defined in this section. First, CPU timing tables are included, and from these results conclusions are drawn to define the CPU capabilities required to run the RPO trajectory planning problems. Also shown are the differences between each problem formulation, and the effects of the number of discretization points.

The results for Scenario 1 are shown in Table 11.9, and the results for Scenario 2 with the approach corridor are shown in Table 11.10. This data is collected for each of the hardware options in Table 11.1, which include the J-board (J-B), the Macbook Pro (MBP), and the Desktop (DTP). Each of the problems was run a total of 10 times on each device, and the CPU timing presented here is the average of the 10 runs. These tables include the solution time, the problem solved (primal or dual, denoted by P/D), number of iterations, and estimated number of FLOPS per iteration provided by MOSEK. As a reminder, the four problems for each formulation represent the four discretization levels ($N = 100, 200, 400$, and 800).

Table 11.9: Scenario 1, without approach corridor, problem results for CPU requirements

<table>
<thead>
<tr>
<th>Prob.</th>
<th>Formulation</th>
<th>Sol. Time (s)</th>
<th>P/D Iter.</th>
<th>FLOPS Iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>J-B</td>
<td>MBP</td>
<td>DTP</td>
</tr>
<tr>
<td>1</td>
<td>Cart.</td>
<td>0.13</td>
<td>0.14</td>
<td>0.09</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>0.13</td>
<td>0.17</td>
<td>0.12</td>
</tr>
<tr>
<td>3</td>
<td>U-X</td>
<td>0.29</td>
<td>0.19</td>
<td>0.12</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>0.72</td>
<td>0.30</td>
<td>0.21</td>
</tr>
<tr>
<td>5</td>
<td>LROE</td>
<td>0.06</td>
<td>0.13</td>
<td>0.05</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>0.15</td>
<td>0.14</td>
<td>0.09</td>
</tr>
<tr>
<td>7</td>
<td>U-X</td>
<td>0.22</td>
<td>0.16</td>
<td>0.10</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>0.42</td>
<td>0.21</td>
<td>0.14</td>
</tr>
<tr>
<td>9</td>
<td>LROE</td>
<td>0.05</td>
<td>0.11</td>
<td>0.05</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>0.07</td>
<td>0.11</td>
<td>0.06</td>
</tr>
<tr>
<td>11</td>
<td>U-only</td>
<td>0.16</td>
<td>0.11</td>
<td>0.07</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>0.27</td>
<td>0.12</td>
<td>0.07</td>
</tr>
</tbody>
</table>
Table 11.10: Scenario 2, with approach corridor, problem results for CPU requirements

<table>
<thead>
<tr>
<th>Prob.</th>
<th>Formulation</th>
<th>Sol. Time (s)</th>
<th>P/D Iter.</th>
<th>FLOPS Iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>J-B</td>
<td>MBP</td>
<td>DTP</td>
</tr>
<tr>
<td>1</td>
<td>Cart.</td>
<td>0.35</td>
<td>0.20</td>
<td>0.14</td>
</tr>
<tr>
<td>2</td>
<td>U-X</td>
<td>0.56</td>
<td>0.26</td>
<td>0.16</td>
</tr>
<tr>
<td>3</td>
<td>U-X</td>
<td>1.11</td>
<td>0.39</td>
<td>0.25</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>2.28</td>
<td>1.15</td>
<td>0.78</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>0.20</td>
<td>0.18</td>
<td>0.12</td>
</tr>
<tr>
<td>6</td>
<td>LROE</td>
<td>0.39</td>
<td>0.20</td>
<td>0.13</td>
</tr>
<tr>
<td>7</td>
<td>U-X</td>
<td>0.78</td>
<td>0.27</td>
<td>0.15</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>1.67</td>
<td>0.58</td>
<td>0.39</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>1.38</td>
<td>0.44</td>
<td>0.29</td>
</tr>
<tr>
<td>10</td>
<td>LROE</td>
<td>3.63</td>
<td>1.37</td>
<td>0.92</td>
</tr>
<tr>
<td>11</td>
<td>U-only</td>
<td>12.40</td>
<td>6.27</td>
<td>4.24</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>62.59</td>
<td>37.16</td>
<td>24.71</td>
</tr>
</tbody>
</table>

11.6 Memory Requirements Tables

All memory requirements for the problems in this analysis are shown in the following tables. These values were collected using valgrind, and the values shown include the total heap memory and the number of instructions. Scenario 1 is shown in Table 11.11 and Scenario 2 is shown in Table 11.12.

Table 11.11: Scenario 1, without approach corridor, problem results for memory requirements

<table>
<thead>
<tr>
<th>Prob.</th>
<th>Formulation</th>
<th>RAM (Mb)</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>2.8</td>
<td>1.7e8</td>
</tr>
<tr>
<td>2</td>
<td>Cart.</td>
<td>7.2</td>
<td>2.8e8</td>
</tr>
<tr>
<td>3</td>
<td>U-X</td>
<td>7.3</td>
<td>6.0e8</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>17.1</td>
<td>1.3e9</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>2.8</td>
<td>1.4e8</td>
</tr>
<tr>
<td>6</td>
<td>LROE</td>
<td>4.5</td>
<td>2.9e8</td>
</tr>
<tr>
<td>7</td>
<td>U-X</td>
<td>10.3</td>
<td>4.7e8</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>16.0</td>
<td>8.7e8</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>2.4</td>
<td>8.3e7</td>
</tr>
<tr>
<td>10</td>
<td>LROE</td>
<td>2.8</td>
<td>1.3e8</td>
</tr>
<tr>
<td>11</td>
<td>U-only</td>
<td>4.4</td>
<td>2.3e8</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>7.6</td>
<td>4.2e8</td>
</tr>
</tbody>
</table>
Table 11.12: Scenario 2, with approach corridor, problem results for memory requirements

<table>
<thead>
<tr>
<th>Prob.</th>
<th>Formulation</th>
<th>RAM (Mb)</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Cart.</td>
<td>6.5</td>
<td>3.8e8</td>
</tr>
<tr>
<td>2</td>
<td>Cart.</td>
<td>8.4</td>
<td>5.0e8</td>
</tr>
<tr>
<td>3</td>
<td>U-X</td>
<td>12.4</td>
<td>1.0e9</td>
</tr>
<tr>
<td>4</td>
<td>U-X</td>
<td>21.0</td>
<td>2.0e9</td>
</tr>
<tr>
<td>5</td>
<td>LROE</td>
<td>6.3</td>
<td>3.8e8</td>
</tr>
<tr>
<td>6</td>
<td>LROE</td>
<td>8.5</td>
<td>5.0e8</td>
</tr>
<tr>
<td>7</td>
<td>U-X</td>
<td>12.8</td>
<td>1.0e9</td>
</tr>
<tr>
<td>8</td>
<td>U-X</td>
<td>21.2</td>
<td>2.0e9</td>
</tr>
<tr>
<td>9</td>
<td>LROE</td>
<td>14.0</td>
<td>1.6e9</td>
</tr>
<tr>
<td>10</td>
<td>LROE</td>
<td>41.3</td>
<td>5.1e9</td>
</tr>
<tr>
<td>11</td>
<td>U-only</td>
<td>154.4</td>
<td>2.8e10</td>
</tr>
<tr>
<td>12</td>
<td>&gt;500</td>
<td>&gt;1.0e11</td>
<td></td>
</tr>
</tbody>
</table>

Example plots for the memory allocation as a function of instructions from the massif tool are included for two of the above cases. The plot in Fig. 11.1 shows the results for scenario 1, LROE U-only formulation, with $N = 800$ points. The plot in Fig. 11.2 shows the results for scenario 2, LROE U-X formulation, with $N = 800$ points.

![Massif memory usage vs. number of instructions](image-url)
As can be seen in these plots, the initial phase for the MOSEK solver is the memory allocation. The memory usage here increases when new data structures are introduced and decreased during MOSEK’s simplification and number-of-nonzero reduction processes. A few minor plateaus can be seen here where loops occur, for example, in the row-rank reduction processes. The plateau in the middle of the graph shows the area where MOSEK is solving the problem. During this phase all data and variables have been previously defined and are being operated upon, therefore, the memory remains constant. The final plateau shows the approximate instructions required to write the solution to an output file.

### 11.7 Custom Solver Comparison

The results from the custom solver, based on the results from Chapter 10, are now compared to the MOSEK solver. The results are presented in tables showing solution time, objective function value, and number of iterations. Due to the long solution times required by the MatLab custom solver for large problems, a set of smaller problems ($N = 20$, and 40) is analyzed, and the required time is compared to MOSEK. In these results, the same two scenarios are examined, with all of the same parameters except the number of discretization points. The problems analyzed are shown in Tables 11.13 and 11.14.
Table 11.13: Scenario 1, without approach corridor, cases tested for custom solver comparison

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LROE</td>
<td>20</td>
<td>200</td>
<td>120</td>
<td>40</td>
</tr>
<tr>
<td>2</td>
<td>U-X</td>
<td>40</td>
<td>400</td>
<td>240</td>
<td>80</td>
</tr>
</tbody>
</table>

Table 11.14: Scenario 2, with approach corridor, cases tested for custom solver comparison

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LROE</td>
<td>20</td>
<td>200</td>
<td>120</td>
<td>60</td>
</tr>
<tr>
<td>2</td>
<td>U-X</td>
<td>40</td>
<td>400</td>
<td>240</td>
<td>120</td>
</tr>
</tbody>
</table>

The results for the custom barrier and PDIP methods are shown in Tables 11.15 and 11.16 for Scenarios 1 and 2, respectively. All timing results are shown in seconds, and were tested on the MacBook Pro laptop. The objective function ($J$) shows the optimal $\Delta v$ value in meters per second, calculated via $\Delta v = \Delta t \sum_{i=1}^{N} ||u_i||$.

Table 11.15: Scenario 1, without approach corridor, problem results for custom solver comparison

<table>
<thead>
<tr>
<th>Prob.</th>
<th>MOSEK</th>
<th>Custom Barrier</th>
<th>Custom PDIP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.20</td>
<td>9</td>
<td>4.09</td>
</tr>
<tr>
<td>2</td>
<td>0.25</td>
<td>10</td>
<td>4.05</td>
</tr>
</tbody>
</table>

Table 11.16: Scenario 2, with approach corridor, problem results for custom solver comparison

<table>
<thead>
<tr>
<th>Prob.</th>
<th>MOSEK</th>
<th>Custom Barrier</th>
<th>Custom PDIP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.22</td>
<td>15</td>
<td>7.652</td>
</tr>
<tr>
<td>2</td>
<td>0.22</td>
<td>16</td>
<td>7.638</td>
</tr>
</tbody>
</table>
11.8 Conclusions

The results from the CPU timing analysis show the effects of the overall computation time required for the different scenarios, formulations, and number of discretization points. It is evident that for the CPUs with slower clock speed, the total time required to solve the RPO trajectory planner problem increases. This is seen to correlate with the number of instructions required. As expected, for a certain number of required instructions the CPU with greater instructions-per-second capabilities is able to solve the problem faster.

The effects of the formulation show cases where the control-only formulation may be preferred over the traditional states-and-control equations. This can clearly be seen in the initial approach scenarios where there aren’t any approach corridor constraints. Excluding these constraints allows the control-only formulation to be implemented much faster, since there are fewer overall equality constraints. Additionally, the LROE states-and-controls formulation is shown to outperform the traditional Cartesian HCW formulation when the approach corridor constraints are included in the problem. In these cases where an approach corridor is desired, the control-only formulation does not perform nearly as well as others, due to the complexity of those specific second-order cone constraints.

Memory requirement results show that the control-only formulation without the approach corridor may significantly save memory, while the LROE and Cartesian states-and-controls are comparable. However, with the addition of the approach corridor, the memory required for control-only formulation grows rapidly for large problems, thus making it undesirable. The total memory and number of instructions is shown to nearly double, as the size of the problem doubles (from $N = 100, 200, 400,$ and $800$), while for each problem size the memory required between the Cartesian HCW and LROE formulation does not differ greatly.
CHAPTER 12
NONLINEAR SIMULATION

12.1 Introduction

This Chapter describes the nonlinear simulation that is used to test the trajectory planner. This includes the full nonlinear differential equations for both the chief and deputy, which are formulated using two methods: the inertial-relative and the inertial-inertial chief and deputy states. These are implemented in two separate simulations, where for the inertial-relative simulation, the deputy’s differential equations are defined with respect to the chief’s LVLH frame, and in the inertial-inertial simulation both vehicles states are with respect to the inertial frame.

The control laws in the simulation implement an optimal LQR control law for trajectory following, and take two main forms. These include a trajectory following method that tracks the planner position and velocity, and a secondary method that tracks the planner LROEs. The control law also includes the option of implementing a feed-forward optimal control provided by the planner.

The trajectory following method also uses two different reference trajectory models. The first model implements the reference trajectory in the traditional Cartesian LVLH frame. The second model uses spherical coordinates to convert the reference trajectory, and is implemented in a spherical formulation. The purpose of implementing the spherical reference trajectory is to gain more accuracy in approximating the true relative orbit, for trajectories with large along-track or cross-track separation.

12.2 Nonlinear Simulation Model

12.2.1 Inertial Chief States and Relative Deputy States

Twelve nonlinear differential equations are implemented in the simulation for inertial
chief satellite states and relative deputy states. These consist of position and velocity states for each vehicle. The state for the chief $X_c$, is the inertial position and velocity, defined as

$$X_c(t) = \begin{bmatrix} R_c(t)^T & V_c(t)^T \end{bmatrix}^T$$

(12.1)

and the state for the deputy is the relative position and velocity coordinatized in the inertial frame, defined as

$$x_d(t) = \begin{bmatrix} r_{rel}(t)^T & v_{rel}(t)^T \end{bmatrix}^T$$

(12.2)

The differential equations for the chief satellite in the inertial frame are

$$\dot{R}_c = V_c$$

(12.3)

$$\dot{V}_c = -\mu \frac{R_c}{||R_c||^3} + a_{J2}(R_c) + a_{\text{Drag}}(R_c, V_c, C_B) + a_{\text{SRP}}(C_{R_c}, A_c, m_c)$$

(12.4)

where $a_{J2}$, $a_{\text{Drag}}$, and $a_{\text{SRP}}$ represent the inertial accelerations from the effects of both J2, drag, and solar radiation pressure, respectively. The equations for the J2 accelerations are defined as [155]

$$a_{J2}(R) = -\frac{3J_2\mu R^2}{2||R||^5} \left( (I + 2\hat{i}_z\hat{i}_z^T)R - \frac{5}{||R||^2} |R|^T\hat{i}_z\hat{i}_z^TR |R| \right)$$

(12.5)

Where $J2$ is a constant, $R_c$ is the radius of the Earth, and $\hat{i}_z = [0 0 1]^T$. The equation for acceleration due to drag is [155]

$$a_{\text{Drag}}(R, V, C_B) = -\frac{1}{2} \rho(R) \frac{1}{C_B} ||V||V$$

(12.6)

The atmospheric density model is based on the Handbook of Geophysics and the Space Environment, by Adolph S Jursa from the US Airforce Geophysics Laboratory [156]. The value $C_B$ is the ballistic coefficient, defined as the inverse of the product of the drag coefficient
and area to mass ratio \( C_B(C_D, A, m) \), or

\[
C_B(C_D, A, m) = \frac{m}{C_D A}
\]  
(12.7)

The density, \( \rho \), of the atmosphere is a function of the position vector

\[
\rho(R) = \rho_0 e^{-\frac{h - 175 \text{ (km)}}{H_{scale}}}
\]  
(12.8)

where,

\[
h = ||R|| - 6371 \text{ (km)} , \quad H_{scale} = \frac{T}{M} \text{ (km)}
\]

\[
T = 900 + 2.5(\text{F10}_7 - 70) \text{ (K)}
\]

\[
M = 27 - 0.012(h - 200) \text{ (K/km)}
\]

where \( h \) and \( H_{scale} \) are in kilometers. The atmospheric temperature constant used here is \( \text{F10}_7 = 120 \text{ K} \). The equation for \( M \) is the effective change in temperature per kilometer of altitude.

The equation for the acceleration due to solar radiation pressure on the chief is given by \([155]\)

\[
a_{\text{SRP}}(C_R, A, m) = -pC_R A m \hat{u}_S
\]  
(12.9)

where \( p \) is the radiation pressure, \( C_R \) is the coefficient of reflectivity of the satellite, \( A \) is the cross-sectional area, \( m \) is the chief satellite’s mass, and \( \hat{u}_S \) is the sun-satellite direction unit vector. Over short duration simulations (< 1 day), the sun-satellite vector and radiation pressure can be assumed to be constant, where \( p \) is generally between \( 4.38 \times 10^{-6} \text{ N/m}^2 \leq p \leq 4.68 \times 10^{-6} \text{ N/m}^2 \) over the duration of one Earth orbit around the sun \([138]\).
The equations of motion for the relative dynamics of the deputy satellite follow [131]

\[ \dot{r}_{rel} = v_{rel}, \quad \dot{v}_{rel} = a_d - a_c - 2\omega \times v_{rel} - \omega \times (\omega \times r_{rel}) - \dot{\omega} \times r_{rel} \] (12.10)

where \( a_d \) is the inertial acceleration of the deputy satellite, \( a_c \) is the inertial acceleration of the chief, and \( \omega \) is the angular velocity vector of the LVLH frame. These equations for relative velocity and acceleration are with respect to LVLH, and coordinatized in the inertial frame. The \( \omega \) vector can be written as

\[ \omega = \frac{R_c \times V_c}{||R_c||^2} \] (12.11)

and the time derivative is

\[ \dot{\omega} = \frac{R_c \times \dot{V}_c}{||R_c||^2} - 2\frac{R_c \times V_c}{||R_c||^2} R_c^T V_c \] (12.12)

In the relative acceleration equation, the chief’s inertial acceleration has been previously defined, and the deputy satellite’s is written as

\[ a_d = a_{\text{Thrust}} - \mu \frac{R_d}{||R_d||^2} + a_{J2}(R_d) + a_{\text{Drag}}(R_d, V_d, C_{B_d}) + a_{\text{SRP}}(C_{R_d}, A_d, m_d) \] (12.13)

where

\[ R_d = R_c + r_{rel} \] (12.14)

\[ V_d = V_c + \omega \times r_{rel} + v_{rel} \] (12.15)

The expressions for \( a_{J2}, a_{\text{Drag}}, \) and \( a_{\text{SRP}} \) are given in Eq. 12.5, 12.6, and 12.9, respectively.

12.2.2 Inertial Chief States and Inertial Deputy States

In this formulation, twelve nonlinear differential equations are implemented in the fully inertial simulation, which consist of inertial position and velocity states for each vehicle.
The state for the chief $\mathbf{X}_c$, is the inertial position and velocity, defined as

$$ \mathbf{X}_c(t) = \begin{bmatrix} \mathbf{R}_c(t)^T & \mathbf{V}_c(t)^T \end{bmatrix}^T $$

and the state for the deputy is the inertial position and velocity, defined as

$$ \mathbf{X}_d(t) = \begin{bmatrix} \mathbf{R}_d(t)^T & \mathbf{V}_d(t)^T \end{bmatrix}^T $$

The differential equations for the chief satellite in the inertial frame are the same as previously defined. The differential equations for the deputy satellite are now

$$ \dot{\mathbf{R}}_d = \mathbf{V}_d $$

$$ \dot{\mathbf{V}}_d = \mathbf{a}_{\text{Thrust}} - \mu \frac{\mathbf{R}_d}{||\mathbf{R}_d||^3} + \mathbf{a}_{J2}(\mathbf{R}_d) + \mathbf{a}_{\text{Drag}}(\mathbf{R}_d, \mathbf{V}_d, C_{B_d}) + \mathbf{a}_{\text{SRP}}(C_{R_d}, A_d, m_d) $$

All perturbations in this equation have been defined in Eq. 12.5, 12.6, and 12.9. The inertial state can be transformed to the relative state via

$$ \mathbf{r}_{rel} = \mathbf{R}_d - \mathbf{R}_c $$

$$ \mathbf{v}_{rel} = \mathbf{V}_d - \mathbf{V}_c - \mathbf{\omega} \times \mathbf{r}_{rel} $$

where all vectors are coordinatized in the inertial frame.

### 12.2.3 Differential Equations Summary

For the inertial simulation, all states are denoted as $\mathbf{X}_I$, where

$$ \mathbf{X}_I = \begin{bmatrix} \mathbf{X}_c^T & \mathbf{X}_d^T \end{bmatrix}^T = \begin{bmatrix} \mathbf{R}_c^T & \mathbf{V}_c^T & \mathbf{R}_d^T & \mathbf{V}_d^T \end{bmatrix}^T $$

and for the inertial/relative simulation, all states are denoted as $\mathbf{X}_{IR}$, where

$$ \mathbf{X}_{IR} = \begin{bmatrix} \mathbf{X}_c^T & \mathbf{X}_d^T \end{bmatrix}^T = \begin{bmatrix} \mathbf{R}_c^T & \mathbf{V}_c^T & \mathbf{r}_{rel}^T & \mathbf{v}_{rel}^T \end{bmatrix}^T $$
Now all derivatives of the states can be written as a function of the states, i.e. \( \dot{X}_I = f(X_I) \), or \( \dot{X}_{IR} = f(X_{IR}) \).

### 12.3 Reference Trajectory

The reference trajectory generated by the planner is composed of the full state \( x^{LVLH}_{ref_i} \), and control \( u^{LVLH}_{ref_i} \), discretized at each time \( t_i \). The reference trajectory is always with respect to the LVLH frame, and coordinatized in the LVLH frame. The full state at each time is defined as the relative position and velocity in the chief’s Cartesian LVLH frame, defined with x-radial, y-along-track, and z-cross-track. The reference state is written as

\[
x^{LVLH}_{ref_i} = \begin{bmatrix} r^{LVLH}_{rel\ i} \\ v^{LVLH}_{rel\ i} \end{bmatrix}^T, \quad \text{for } i = 1, \ldots, N + 1 \quad (12.24)
\]

This reference state can be transformed to the LROE states, \( \bar{x}_{ref_i} \), using a linear transformation, \( T \) in Eq. 5.41. Therefore, an alternative set of reference states that can be implemented are

\[
\bar{x}_{ref_i} = T x^{LVLH}_{ref_i} = \begin{bmatrix} x_r \\ y_r \\ a_r \sin(E_r) \\ a_r \cos(E_r) \\ A_z \sin(\psi_z) \\ A_z \cos(\psi_z) \end{bmatrix}^T_{ref_i}, \quad \text{for } i = 1, \ldots, N + 1 \quad (12.25)
\]

The reference control, or reference input \( u^{LVLH}_{ref_i} \), is defined as the translational thrust acceleration vector (also coordinatized in the LVLH frame). This is written as

\[
u^{LVLH}_{ref_i} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}^T_{ref_i}, \quad \text{for } i = 1, \ldots, N + 1 \quad (12.26)
\]

If the LROE reference trajectory is used, then the reference input, \( \bar{u}^{LVLH}_{ref_i} \), is defined as

\[
\bar{u}^{LVLH}_{ref_i} = \frac{u^{LVLH}_{ref_i}}{\omega} = \begin{bmatrix} \nu_x \\ \nu_y \\ \nu_z \end{bmatrix}^T_{ref_i}, \quad \text{for } i = 1, \ldots, N + 1 \quad (12.27)
\]

where \( \omega \) is the chief’s mean orbital motion, from the HCW dynamics model.
Although the reference trajectory is discretized, and is defined at times \( t_i \), a value at intermediate times \( t \) may be required for trajectory following. Define \( x_{ref}(t, t_i) \) at all times \( t \) between points \( t_i \leq t < t_{i+1} \) as

\[
x_{ref}^{LVLH}(t, t_i) = \Phi(t, t_i)x_{ref}^{LVLH} + B_d(t, t_i)u_{ref}^{LVLH}
\]

and for the LROEs,

\[
x_{ref}(t, t_i) = \Phi(t, t_i)x_{ref} + B_d(t, t_i)u_{ref}^{LVLH}
\]

where \( \Phi(t, t_i) = e^{A(t-t_i)} \), \( B_d(t, t_i) = \int_{t_i}^{t} e^{A(t-\tau)}B\,d\tau \), \( \bar{\Phi}(t, t_i) = e^{\bar{A}(t-t_i)} \), and \( \bar{B_d}(t, t_i) = \int_{t_i}^{t} e^{\bar{A}(t-\tau)}\bar{B}\,d\tau \). The matrices \( A, B, \bar{A}, \) and \( \bar{B} \) are given in Eqs. 5.4 and 5.43.

### 12.4 Optimal LQR Control Design

The design for the LQR controller begins with the definition of the objective function. The LQR objective function is written in terms of the state, control acceleration, and reference trajectory as

\[
J = \int_{t_0}^{\infty} \left( x_{rel}^{LVLH}(t) - x_{ref}^{LVLH} \right)^T Q \left( x_{rel}^{LVLH}(t) - x_{ref}^{LVLH} \right) + \left( u^{LVLH}(t) + u_{ref}^{LVLH} \right)^T R \left( u^{LVLH}(t) + u_{ref}^{LVLH} \right) \, dt
\]

where \( u^{LVLH}(t) \) is the optimal control that minimizes this objective function. In terms of the LROE states and controls, the objective function is expressed as

\[
J = \int_{t_0}^{\infty} \left( \bar{x}(t) - \bar{x}_{ref} \right)^T \bar{Q} \left( \bar{x}(t) - \bar{x}_{ref} \right) + \left( \bar{u}^{LVLH}(t) - \bar{u}_{ref}^{LVLH} \right)^T \bar{R} \left( \bar{u}^{LVLH}(t) - \bar{u}_{ref}^{LVLH} \right) \, dt
\]

where \( \bar{x} = T x_{ref}^{LVLH} \) and \( x_{ref} = [(r_{rel})^T (v_{rel})]^T \). The optimal control solution to this problem is given by

\[
u^{LVLH} = K(x_{rel}^{LVLH} - x_{ref}^{LVLH}) + u_{ref}^{LVLH}
\]
Here, a feed-forward acceleration has been added to the standard LQR solution. In terms of the LROEs, the control law with the feed-forward term is

\[
\tilde{u}^{LVLH} = \bar{K}(\bar{x} - \bar{x}_{ref}) + \bar{u}^{LVLH}_{ref}
\]  

(12.31)

The optimal static gain matrices \(K\) and \(\bar{K}\) are determined from the solutions to the matrix algebraic Riccati equations (MARE), where the dynamics for the original and transformed systems follow the relationship

\[
\dot{x} = Ax + Bu, \quad \dot{\bar{x}} = \bar{A}\bar{x} + \bar{B}\bar{u}
\]

(12.32)

where \(A, B, \bar{A},\) and \(\bar{B}\) are given in Eqs. 5.4 and 5.43. Then, the MARE equations are [157]

\[
PA + A^TP + Q - PBR^{-1}B^TP = 0
\]

(12.33)

\[
\bar{P}\bar{A} + \bar{A}^T\bar{P} + \bar{Q} - \bar{P}\bar{B}\bar{R}^{-1}\bar{B}^T\bar{P} = 0
\]

(12.34)

so that \(K\) and \(\bar{K}\) are

\[
K = -R^{-1}B^TP, \quad \bar{K} = -\bar{R}^{-1}\bar{B}^T\bar{P}
\]

(12.35)

The values for the \(Q\) and \(\bar{Q}\) matrices can be selected, from Bryson’s rule [158], according to the maximum allowable errors while tracking the trajectory. These errors are written for each of the states, \(x^{LVLH}_{rel}\), as

\[
q = [x_{max} \; y_{max} \; z_{max} \; \dot{x}_{max} \; \dot{y}_{max} \; \dot{z}_{max}]^T
\]

(12.36)

and for each of the LROE states, \(\bar{x}\), as

\[
\bar{q} = [x_{rmax} \; y_{rmax} \; a_r \sin(E_r)_{max} \; a_r \cos(E_r)_{max} \; A_z \sin(\psi_z)_{max} \; A_z \cos(\psi_z)_{max}]^T
\]

(12.37)
Then, the $Q$ matrix can be allocated as

\[ Q_{ii} = \frac{1}{q_i^2}, \quad \bar{Q}_{ii} = \frac{1}{\bar{q}_i^2} \quad (12.38) \]

Note that in this formulation, the maximum allowable state errors can be selected depending on the requirements for the problem. For example, the in-plane $\dot{x}$, and $\dot{y}$ velocities of the deputy can be tracked more closely than the other states. Similarly, for $\bar{x}$, the center of motion can be tracked very closely, while the other states may be considered less important. Additionally, it is possible to track $a_r$ and $A_z$ to within a desired value, by selecting the maximum errors for $a_r \sin(E_r)_{\text{max}}, a_r \cos(E_r)_{\text{max}}$ and $A_z \sin(\psi_z)_{\text{max}}, A_z \cos(\psi_z)_{\text{max}}$ appropriately. This is shown by

\[ a_r \sin(E_r)_{\text{max}} = a_r \cos(E_r)_{\text{max}} = \frac{a_{r_{\text{max}}}}{\sqrt{2}} \quad (12.39) \]
\[ A_z \sin(\psi_z)_{\text{max}} = A_z \cos(\psi_z)_{\text{max}} = \frac{A_{z_{\text{max}}}}{\sqrt{2}} \quad (12.40) \]

To define the $R, \bar{R}$ matrices, again this may be based on the maximum allowable control error while tracking the trajectory. From Bryson’s rule [158], allocate $R$ using

\[ r = [a_{x_{\text{max}}} \ a_{y_{\text{max}}} \ a_{z_{\text{max}}}]^T \quad (12.41) \]

and for the $\bar{R}$ matrix, $\bar{r}$ is

\[ \bar{r} = [\nu_{x_{\text{max}}} \ \nu_{y_{\text{max}}} \ \nu_{z_{\text{max}}}]^T \quad (12.42) \]

Then, the $R$ and $\bar{R}$ matrices are

\[ R_{ii} = \frac{1}{r_i^2}, \quad \bar{R}_{ii} = \frac{1}{\bar{r}_i^2} \quad (12.43) \]
All vectors in these equations are co-ordinatized in the LVLH frame, where the transformation from inertial to LVLH is written as

\[
T^{I}_{LVLH} = \begin{bmatrix}
\hat{i}_{LVLH}^T \\
\hat{j}_{LVLH}^T \\
\hat{k}_{LVLH}^T
\end{bmatrix}
\]  
(12.44)

Each unit vector for the LVLH frame is defined as

\[
\hat{i}_{LVLH} = \frac{R_c}{||R_c||}
\]

\[
\hat{j}_{LVLH} = \hat{i}_{LVLH} \times \hat{i}_{LVLH}
\]

\[
\hat{k}_{LVLH} = \frac{R_c \times V_c}{||R_c \times V_c||}
\]

The transformation from LVLH to inertial is then \(T^{I}_{LVLH} = (T^{I}_{LVLH})^T\). The commanded acceleration Eqs. 12.30 and 12.31 can also be written in the inertial frame.

\[
\mathbf{u}^{I}_{cmd} = T^{I}_{LVLH} \left[ K (\mathbf{x}^{LVLH}_{ref} - \mathbf{x}^{LVLH}_{ref}) + \mathbf{u}^{LVLH}_{ref} \right]
\]  
(12.45)

or,

\[
\tilde{\mathbf{u}}^{I}_{cmd} = T^{I}_{LVLH} \left[ \tilde{K} (\tilde{x} - \tilde{x}_{ref}) + \tilde{u}^{LVLH}_{ref} \right]
\]  
(12.46)

### 12.5 Cartesian to Spherical Conversion of Reference Trajectory

The reference trajectory can be optionally converted from the LVLH Cartesian coordinates to spherical coordinates [69]. This may be desired to improve the along-track and cross-track performance of the trajectory following method in the simulation. First, the
relative spherical coordinates are defined, from Eq. 5.15, as

\[
x_{\text{Sph}} = \begin{bmatrix}
\delta \rho \\
\delta \theta \\
\delta \phi \\
\dot{\delta} \rho \\
\dot{\delta} \theta \\
\dot{\delta} \phi \\
\end{bmatrix} = \begin{bmatrix}
\rho_d - \rho_c \\
\theta_d - \theta_c \\
\phi_d - \phi_c \\
\dot{\rho}_d - \dot{\rho}_c \\
\dot{\theta}_d - \dot{\theta}_c \\
\dot{\phi}_d - \dot{\phi}_c \\
\end{bmatrix}
\] (12.47)

The conversion from Cartesian to spherical coordinates can be represented as a linear transformation, \( T_{\text{Sph}}(t) \), which is a function of the radial distance of the chief satellite, \( \rho_c(t) \).

\[
T_{\text{Sph}}(t) = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{1}{\rho_c(t)} & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{1}{\rho_c(t)} & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{1}{\rho_c(t)} & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{1}{\rho_c(t)} \\
\end{bmatrix}
\] (12.48)

where,

\[
\rho_c(t) = ||R_c(t)||
\] (12.49)

The relative spherical coordinate states, \( x_{\text{Sph}} \), are then

\[
x_{\text{Sph}} = T_{\text{Sph}}x^{\text{LV}_L\text{L}_H}_{\text{rel}}
\] (12.50)
To develop the spherical coordinates, the unit vectors for the chief and deputy are first defined and coordinatized in the LVLH frame. For the chief

\[ \hat{i}_{\rho_c}^{LVLH} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \hat{i}_{\theta_c}^{LVLH} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \hat{i}_{\phi_c}^{LVLH} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \] (12.51)

which are similar to the \( x-y-z \) unit vectors of LVLH Cartesian. Then for the deputy,

\[ \hat{i}_{\rho_d}^{LVLH} = \begin{bmatrix} \cos(\delta \phi) \cos(\delta \theta) \\ \cos(\delta \phi) \sin(\delta \theta) \\ \sin(\delta \phi) \end{bmatrix}, \quad \hat{i}_{\theta_d}^{LVLH} = \begin{bmatrix} -\sin(\delta \theta) \\ \cos(\delta \theta) \\ 0 \end{bmatrix}, \quad \hat{i}_{\phi_d}^{LVLH} = \begin{bmatrix} -\sin(\delta \phi) \cos(\delta \theta) \\ -\sin(\delta \phi) \sin(\delta \theta) \\ \cos(\delta \phi) \end{bmatrix} \] (12.52)

This analysis is done entirely in spherical coordinates, for an arbitrary chief orbit (non-circular), and is ultimately implemented in order to gain more accuracy by representing the curvilinear nature of the reference trajectory in the simulation. The relative position in the LVLH frame is written as a function of the spherical states of the chief and the relative spherical coordinates as

\[ \mathbf{r}_{rel}^{LVLH} = \mathbf{r}_d^{LVLH} - \mathbf{r}_c^{LVLH} \] (12.53)

\[ \mathbf{r}_{rel}^{LVLH} = (\rho_c + \delta \rho) \hat{i}_{\rho_d}^{LVLH} - \rho_c \hat{i}_{\rho_c}^{LVLH} \] (12.54)

and the relative velocity in LVLH is similarly written as a function of the spherical states as

\[ \mathbf{v}_{rel}^{LVLH} = \mathbf{v}_d^{LVLH} - \mathbf{v}_c^{LVLH} - \mathbf{\omega}^{LVLH} \times \mathbf{r}_{rel}^{LVLH} \] (12.55)

where the inertial velocities in terms of spherical coordinates are given by [131]

\[ \mathbf{v}_d^{LVLH} = \left( \dot{\rho}_c + \delta \rho \right) \hat{i}_{\rho_d}^{LVLH} + (\rho_c + \delta \rho) \left( \dot{\theta}_c + \delta \theta \right) \cos(\delta \phi) \hat{i}_{\theta_d}^{LVLH} + (\rho_c + \delta \rho) \delta \phi \hat{i}_{\phi_d}^{LVLH} \] (12.56)

\[ \mathbf{v}_c^{LVLH} = \dot{\rho}_c \hat{i}_{\rho_c}^{LVLH} + \rho_c \dot{\theta}_c \hat{i}_{\theta_c}^{LVLH} \] (12.57)
are with respect to the inertial frame, and coordinatized in LVLH. The values for \( \rho_c \), \( \dot{\rho}_c \), and \( \dot{\theta}_c \) are calculated from

\[
\rho_c = ||R_c||
\]

\[
\dot{\rho}_c = \frac{V_c \cdot R_c}{\rho_c}
\]

\[
\dot{\theta}_c = \frac{||V_c - \dot{\rho}_c \hat{i}_{R_c}||}{\rho_c}, \text{ where } \hat{i}_{R_c} = \frac{R_c}{||R_c||}
\]

Substituting in these definitions leads to

\[
v_{rel}^{LVLH} = \left( \dot{\rho}_c + \delta \rho \right) \hat{i}_{\rho_d}^{LVLH} + (\rho_c + \delta \rho) \left( \dot{\theta}_c + \delta \theta \right) \cos(\delta \phi) \hat{i}_{\theta_d}^{LVLH} + (\rho_c + \delta \rho) \delta \phi \hat{i}_{\phi_d}^{LVLH} - \dot{\rho}_c \hat{i}_{\rho_c}^{LVLH} - \rho_c \dot{\theta}_c \hat{i}_{\theta_c}^{LVLH} - \omega^{LVLH} \times \left( (\rho_c + \delta \rho) \hat{i}_{\rho_d}^{LVLH} - \rho_c \hat{i}_{\rho_c}^{LVLH} \right)
\]

where the rotation vector of the chief’s relative LVLH frame, with respect to the inertial frame, is the instantaneous angular rotation vector. This is coordinatized in LVLH as

\[
\omega^{LVLH} = \dot{\theta}_c \hat{i}_{\phi_c}^{LVLH}
\]

These equations show that the reference trajectory, using the spherical model, is a function of the chief’s true spherical states, and the deputy’s relative spherical states. Therefore, this conversion must take place within the simulation, in order to more accurately describe the reference trajectory with respect to the chief’s true inertial states.

12.5.1 Spherical Feed-forward Control

With the spherical dynamics model, the reference optimal control for feed-forward in the control law requires a similar change from Cartesian to spherical coordinates. All radial, along-track and cross-track control in the chief LVLH frame is interpreted as radial, along-track and cross-track control in the spherical frame. This leads to the result of expressing the reference control in the deputy’s LVLH frame, while ultimately coordinatizing these vectors in the chief’s LVLH frame. Therefore, the control vector at each time is ‘rotated’ from LVLH to spherical, then re-coordinatized in the chief’s Cartesian LVLH frame. This
is performed via the following equation

\[ u_{ref}(t) = u_{xref}(t) \hat{i}_{\rho d} + u_{yref}(t) \hat{i}_{\theta d} + u_{zref}(t) \hat{i}_{\phi d} \]  

(12.63)

where \( u_{xref}, \ u_{yref}, \) and \( u_{zref} \) are the LVLH Cartesian elements of the reference control vector, each unit vector \( \hat{i}_{\rho d}, \ \hat{i}_{\theta d}, \) and \( \hat{i}_{\phi d} \) is similarly coordinatized in the LVLH Cartesian frame, therefore \( u_{ref} \) is ultimately coordinatized in LVLH Cartesian.

12.6 Navigation and Control Assumptions

The navigation portion of the simulation is modeled using an idealistic design. The navigation errors are assumed to be zero, therefore the nominal relative position and velocity are provided to the guidance and control algorithms in the case of trajectory planning and trajectory following. The main reason for assuming perfect navigation is to eliminate navigation errors from this analysis and see how well the planner performs under nominal conditions. This could of course be relaxed in future work, where navigation errors may be included in the simulation design.

It is also assumed that the true control acceleration is identical to the nominal commanded control. Therefore, control bias, scale factor (i.e. pointing errors), and noise is excluded from this analysis. This also serves to test the trajectory planner under nominal conditions. The idea is that under the perfect navigation and control assumptions, the primary sources for error are modeling errors (HCW equations, control law/design), perturbations, nonlinearities, and numerical errors.
CHAPTER 13
PERFORMANCE ANALYSIS I, RENDEZVOUS, INSPECTION, AND FINAL APPROACH - SCENARIO 1

13.1 Scenario 1 Methodology

The results from the nonlinear simulation are presented using two primary scenarios. The first scenario includes trajectory planning for orbital rendezvous, satellite inspection, and final approach. In this scenario, the planner is given either the initial relative orbital elements and final desired relative position and velocity in LVLH, or the initial and final relative position and velocity in LVLH. These take the form of three standard cases, which are named Initial Approach, Way-point Following/Inspection, and Final Approach.

**Scenario 1** - Rendezvous, Inspection, and Final Approach

- **Case 1** - Initial Approach
- **Case 2** - Way-point Following/Inspection
- **Case 3** - Final Approach

The following sections describe the method by which the Scenario 1 simulation results are analyzed, and how the trajectories for each case are initialized. These are separated into the analysis setup and initial conditions sections.

13.2 Scenario 1 Analysis Setup

The analysis for the rendezvous, inspection and final approach scenario (Scenario 1) in this chapter includes several nominal trajectories for each case. Examples of nominal trajectories are plotted, and these are accompanied by the control history plots. In the control history plots, the simulation control history is plotted for a variety of example trajectories. Several tracking error plots are also included, to evaluate the controller performance and
show where the differences in $\Delta v$ come from. A table is included for each case examined, which shows the optimal planner $\Delta v$ and the resulting nonlinear simulation $\Delta v$ in each case.

The first objective of this analysis is to show the potential advantages of generating reference trajectories in spherical coordinates, as compared to reference trajectories generated in Cartesian coordinates. It is expected that the spherical reference trajectory will perform better in the cases that have large along-track and/or cross-track separation. Thus, the goal is to prove this point by comparing the Cartesian model to the spherical model, specifically for the initial approach case (Case 1). These trajectories are modeled in the planner using two-body dynamics and a circular chief reference orbit.

The next objective is to show the effects of eccentricity and perturbations in the nonlinear simulation. The effects of eccentricity are shown by plotting trajectories and control histories, and including $\Delta v$ tables for multiple reference orbits with varying eccentricity. Similarly, the effects of J2, drag, and SRP perturbations are shown by plotting trajectories and control histories, and including $\Delta v$ tables for simulations with and without perturbations. The $\Delta v$ tables are used to compare the optimal planner $\Delta v$ to the simulation $\Delta v$ for a variety of cases. The $\Delta v$ tables include the optimal $\Delta v$, $\Delta v_{\text{opt}}$, the simulation $\Delta v$, $\Delta v_{\text{sim}}$, and the percent difference between the two, $p_D$, where percent difference is calculated as the difference divided by the average.

$$p_D = \frac{|\Delta v_{\text{opt}} - \Delta v_{\text{sim}}|}{\frac{\Delta v_{\text{opt}} + \Delta v_{\text{sim}}}{2}} \times 100\%$$  \hspace{1cm} (13.1)

### 13.2.1 Simulation Parameters

The parameters for the simulation are held constant throughout the results. They include both vehicle parameters and simulation constants. For the vehicles, the drag coefficient, coefficient of reflectivity, area and mass are defined for each. These values are presented in Table 13.1 and are defined as in [155].
Table 13.2: Chief initial orbital elements (* in some cases, chief orbit eccentricity is varied)

<table>
<thead>
<tr>
<th>$a_c$</th>
<th>$e_c$</th>
<th>$i_c$</th>
<th>$\Omega_c$</th>
<th>$\omega_c$</th>
<th>$\nu_c$</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.07499742 km</td>
<td>0*</td>
<td>0.488692190</td>
<td>0.349065850</td>
<td>1.46607657</td>
<td>0.78539816</td>
</tr>
</tbody>
</table>
The value for the mean motion based on the given semi-major axis shown in Table 13.2 is \( \omega_c = 1.06091165 \times 10^{-3} \text{ rad/sec} \). The orbital elements shown here, excepting eccentricity, are used to initialize the chief orbit in all cases and scenarios. The eccentricity of the chief is changed in different simulation scenarios, to investigate the effects of varying eccentricity on the RPO planner.

The initialization for the deputy in each case for each scenario is defined in terms of the mission requirements for the trajectory to be planned. For example, in Scenario 1, Case 1, initial approach, it is expected that the deputy is already at a station keeping point ahead of the chief in the along-track LVLH direction, as a result of previous far-field mission plans. Therefore, in this case the initial conditions are specified by traditional Keplerian orbital elements, shown in Eq. 13.3. These include \( a, e, i, \Omega, \omega, \) and \( \nu \) (semi-major axis, eccentricity, inclination, right-ascension, argument of perigee, and true anomaly, respectively) for both a chief (denoted by subscript \( c \)) and deputy (denoted by subscript \( d \)).

\[
\delta e = \begin{bmatrix}
\delta \alpha \\
\delta e \\
\delta i \\
\delta \Omega \\
\delta \omega \\
\delta \nu \\
\end{bmatrix} = \begin{bmatrix}
(a_d - a_c)/a_c \\
e_d - e_c \\
i_d - i_c \\
\Omega_d - \Omega_c \\
\omega_d - \omega_c \\
\nu_d - \nu_c \\
\end{bmatrix} \tag{13.3}
\]

However, for Case 2, way-point following, the deputy initial conditions are defined as the nominal outcome from the Case 1, initial approach, as an example of the nominal relative position and velocity that may result from Case 1, for example, a v-bar station keeping location (and similarly for Case 3, final approach, the deputy initial conditions are defined by the nominal relative position and velocity from Case 2).
13.3 Case 1 - Initial Approach

The Scenario 1 analysis begins with the Case 1, initial approach. In this case, the deputy is initialized by specifying the set of nominal classical relative orbital elements, $\delta \epsilon$, as shown in Eq. 13.3. Four examples are included in this analysis. These examples consist of two different along-track station keeping positions, 5 km station-keeping and 10 km station keeping.

The final conditions are specified in terms of final relative position and velocity in LVLH Cartesian, and the final transfer time. The final point in these examples is a station-keeping orbit at 200 m on the v-bar of LVLH Cartesian. The final transfer time is specified via the number of chief revolutions, $M$, so that the final time in seconds is

$$t_f = M \frac{2\pi}{\omega_c}$$

(13.4)

All initial and final conditions for each example are shown in Table 13.3. The final conditions consist of two different final times; 1.2 revolutions and 2.4 revolutions. These examples implement a maximum control acceleration constraint of 0.5 mm/s$^2$.

Table 13.3: Deputy initial traditional relative orbital elements and final relative position/velocity for Scenario 1, Case 1

<table>
<thead>
<tr>
<th>Example</th>
<th>$\delta \epsilon(t_0)(km)$</th>
<th>$r_{rel}(t_f)$, (m)</th>
<th>$v_{rel}(t_f)$, (m/s)</th>
<th>$M$ (rev.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$[0 0 0 0 0 5/a_c]^T$</td>
<td>$[0 200 0]^T$</td>
<td>$[0 0 0]^T$</td>
<td>1.2</td>
</tr>
<tr>
<td>2</td>
<td>$[0 0 0 0 0 10/a_c]^T$</td>
<td>$[0 200 0]^T$</td>
<td>$[0 0 0]^T$</td>
<td>1.2</td>
</tr>
<tr>
<td>3</td>
<td>$[0 0 0 0 0 5/a_c]^T$</td>
<td>$[0 200 0]^T$</td>
<td>$[0 0 0]^T$</td>
<td>2.4</td>
</tr>
<tr>
<td>4</td>
<td>$[0 0 0 0 0 10/a_c]^T$</td>
<td>$[0 200 0]^T$</td>
<td>$[0 0 0]^T$</td>
<td>2.4</td>
</tr>
</tbody>
</table>

Results with Circular Chief Orbit and without Perturbations

Examples 1-4 are first examined for a circular chief orbit, without perturbations, and using the Cartesian dynamics and control model. The trajectories in this case are shown in Fig. 13.1, the associated controls are shown in Fig. 13.2, and the simulation position and velocity tracking errors are shown in Fig. 13.3.
Fig. 13.1: Scenario 1, Case 1 trajectories (Cartesian model, without perturbations)

Fig. 13.2: Scenario 1, Case 1 control history (Cartesian model, circular orbit, without perturbations)
These four examples were also run using the spherical dynamics and control model, as presented in Chapter 12. The resulting trajectories in LVLH look very similar, and are omitted. However, the control history and tracking errors are clearly better using spherical coordinates, for these cases which include large along-track separation. The control history for the spherical case without perturbations is shown in Fig. 13.4, and the associated simulation error is shown in Fig. 13.5.
The $\Delta v$ results for Scenario 1, Case 1, and all examples (including Cartesian and spherical models), are presented in Table 13.4 below.

<table>
<thead>
<tr>
<th>Example</th>
<th>Form.</th>
<th>$\Delta v_{opt}$ (m/s)</th>
<th>$\Delta v_{sim}$ (m/s)</th>
<th>Diff. (m/s)</th>
<th>$p_D$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Cart.</td>
<td>0.542</td>
<td>0.557</td>
<td>0.015</td>
<td>2.72</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>1.117</td>
<td>1.162</td>
<td>0.046</td>
<td>3.99</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>0.270</td>
<td>0.301</td>
<td>0.031</td>
<td>10.69</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>0.561</td>
<td>0.675</td>
<td>0.113</td>
<td>18.41</td>
</tr>
<tr>
<td>1</td>
<td>Sph.</td>
<td>0.538</td>
<td>0.539</td>
<td>0.001</td>
<td>0.28</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>1.101</td>
<td>1.107</td>
<td>0.006</td>
<td>0.57</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>0.266</td>
<td>0.267</td>
<td>0.001</td>
<td>0.22</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>0.545</td>
<td>0.547</td>
<td>0.002</td>
<td>0.46</td>
</tr>
</tbody>
</table>

**Results with Circular Chief and with Perturbations**

The Scenario 1, Case 1 examples were also run including perturbations. The resulting control history and tracking error using the Cartesian and spherical model, and including perturbations, is shown in Fig. 13.6 and 13.7, respectively. The $\Delta v$ for the Carte-
sian and spherical examples with perturbations are also shown in Table 13.5. Although the spherical coordinates formulation slightly outperforms the Cartesian formulation, the planner/simulation $\Delta v$ differences are now dominated by the perturbations that were not modeled in the planner.

![Control History](image1)

**Fig. 13.6:** Scenario 1, Case 1 control history (spherical model, with perturbations)

![Position and Velocity Error](image2)

**Fig. 13.7:** Scenario 1, Case 1 tracking error (spherical model, with perturbations)
Table 13.5: Scenario 1, Case 1 with circular chief and perturbations, \( \Delta v \) from planner and simulation for reference trajectory in Cartesian and spherical coordinates

<table>
<thead>
<tr>
<th>Example</th>
<th>Form.</th>
<th>( \Delta v_{\text{opt}} ) (m/s)</th>
<th>( \Delta v_{\text{sim}} ) (m/s)</th>
<th>Diff. (m/s)</th>
<th>( p_D ) (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Cart.</td>
<td>0.542</td>
<td>0.605</td>
<td>0.062</td>
<td>10.83</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>1.117</td>
<td>1.279</td>
<td>0.162</td>
<td>13.53</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>0.270</td>
<td>0.344</td>
<td>0.074</td>
<td>23.94</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>0.561</td>
<td>0.749</td>
<td>0.188</td>
<td>28.68</td>
</tr>
<tr>
<td>1</td>
<td>Sph.</td>
<td>0.538</td>
<td>0.588</td>
<td>0.049</td>
<td>8.74</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>1.101</td>
<td>1.231</td>
<td>0.130</td>
<td>11.16</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>0.266</td>
<td>0.326</td>
<td>0.059</td>
<td>20.08</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>0.545</td>
<td>0.655</td>
<td>0.110</td>
<td>18.32</td>
</tr>
</tbody>
</table>

Results with Elliptic Chief Orbit and with Perturbations

The effects of eccentricity of the chief’s orbit is also examined for Scenario 1, Case 1. The previous four examples were run with chief eccentricities \( (e_c) \) of 0.0001, 0.001 and 0.01, and also including perturbations. Only the spherical model was used for these cases, due to the improved performance over the Cartesian model from previous results. Example control history and tracking error plots for for \( e_c = 0.01 \) are shown in Fig. 13.8 and 13.9. The \( \Delta v \) values for each eccentricity are shown in Table 13.6. Since the planner does not model the eccentricity of the chief orbit, the \( \Delta v \) difference clearly increases as the chief orbit eccentricity \( (e_c) \) increases, due to modeling error.

Fig. 13.8: Scenario 1, Case 1 control history (spherical model, with perturbations and chief eccentricity of \( e_c = 0.01 \))
Fig. 13.9: Scenario 1, Case 1 tracking error (spherical model, with perturbations and chief eccentricity of $e_c = 0.01$)

Table 13.6: Scenario 1, Case 1 with varying chief ellipticity and perturbations, $\Delta v$ from planner and simulation for reference trajectory in Cartesian and spherical coordinates

<table>
<thead>
<tr>
<th>Example</th>
<th>Ecc. ($e_c$)</th>
<th>$\Delta v_{opt}$ (m/s)</th>
<th>$\Delta v_{sim}$ (m/s)</th>
<th>Diff. (m/s)</th>
<th>$P_D$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0001</td>
<td>0.538</td>
<td>0.587</td>
<td>0.048</td>
<td>8.64</td>
</tr>
<tr>
<td>2</td>
<td>0.0001</td>
<td>1.101</td>
<td>1.235</td>
<td>0.134</td>
<td>11.50</td>
</tr>
<tr>
<td>3</td>
<td>0.0001</td>
<td>0.267</td>
<td>0.323</td>
<td>0.056</td>
<td>19.27</td>
</tr>
<tr>
<td>4</td>
<td>0.0001</td>
<td>0.545</td>
<td>0.650</td>
<td>0.105</td>
<td>17.60</td>
</tr>
<tr>
<td>1</td>
<td>0.001</td>
<td>0.537</td>
<td>0.588</td>
<td>0.050</td>
<td>8.93</td>
</tr>
<tr>
<td>2</td>
<td>0.001</td>
<td>1.099</td>
<td>1.234</td>
<td>0.135</td>
<td>11.56</td>
</tr>
<tr>
<td>3</td>
<td>0.001</td>
<td>0.265</td>
<td>0.330</td>
<td>0.064</td>
<td>21.48</td>
</tr>
<tr>
<td>4</td>
<td>0.001</td>
<td>0.543</td>
<td>0.669</td>
<td>0.125</td>
<td>20.69</td>
</tr>
<tr>
<td>1</td>
<td>0.01</td>
<td>0.530</td>
<td>0.866</td>
<td>0.336</td>
<td>48.11</td>
</tr>
<tr>
<td>2</td>
<td>0.01</td>
<td>1.086</td>
<td>1.604</td>
<td>0.518</td>
<td>38.50</td>
</tr>
<tr>
<td>3</td>
<td>0.01</td>
<td>0.260</td>
<td>0.900</td>
<td>0.640</td>
<td>110.43</td>
</tr>
<tr>
<td>4</td>
<td>0.01</td>
<td>0.532</td>
<td>1.793</td>
<td>1.261</td>
<td>108.51</td>
</tr>
</tbody>
</table>

13.4 Case 2 - Way-point Following

The results for Case 2 show way-point following trajectories, control histories, tracking errors, and $\Delta v$ tables. For this Case, all examples presented are initialized by specifying
the initial relative position and velocity in LVLH. Since each of these trajectories is in relatively close proximity of the chief, there is no need to implement the reference trajectory in spherical coordinates, therefore, all results shown are modeled in the Cartesian LVLH frame. This case is examined in a method similar to Case 1, by first presenting results under perturbed orbits, then taking a brief look at the effects of chief eccentricity. A maximum control acceleration constraint of 0.5 mm/s\(^2\) was used in this case.

The way-point following examples explore an inspection scenario of the chief satellite including perturbations, and where initially the chief is in a circular orbit. These trajectories consist of moving from 100 m on the along-track axis (in front of the chief) to 100 m on the cross-track axis, then continuing from there to 100 m on the radial axis, and finally to -100 m on the along track axis (behind the chief). This results in a total of three trajectories. At each initial and final point in these trajectories the desired relative velocity in the LVLH frame is zero.

Two final times are considered, one completes each trajectory in 30% of a chief revolution, and the other completes each trajectory in 60% of a chief revolution. Thus a total of six examples are examined, and each of the initial/final conditions and final times are shown in Table 13.7.

<table>
<thead>
<tr>
<th>Example</th>
<th>(\mathbf{r}_{rel}(t_0), (m))</th>
<th>(\mathbf{v}_{rel}(t_0), (m/s))</th>
<th>(\mathbf{r}_{rel}(t_f), (m))</th>
<th>(\mathbf{v}_{rel}(t_f), (m/s))</th>
<th>M (rev.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>([0 \ 100 \ 0]^T)</td>
<td>([0 \ 0 \ 0]^T)</td>
<td>([0 \ 0 \ 100]^T)</td>
<td>([0 \ 0 \ 0]^T)</td>
<td>0.3</td>
</tr>
<tr>
<td>2</td>
<td>([0 \ 0 \ 100]^T)</td>
<td>([0 \ 0 \ 0]^T)</td>
<td>([100 \ 0 \ 0]^T)</td>
<td>([0 \ 0 \ 0]^T)</td>
<td>0.3</td>
</tr>
<tr>
<td>3</td>
<td>([100 \ 0 \ 0]^T)</td>
<td>([0 \ 0 \ 0]^T)</td>
<td>([-100 \ 0 \ 0]^T)</td>
<td>([0 \ 0 \ 0]^T)</td>
<td>0.3</td>
</tr>
<tr>
<td>4</td>
<td>([0 \ 100 \ 0]^T)</td>
<td>([0 \ 0 \ 0]^T)</td>
<td>([0 \ 0 \ 100]^T)</td>
<td>([0 \ 0 \ 0]^T)</td>
<td>0.6</td>
</tr>
<tr>
<td>5</td>
<td>([0 \ 0 \ 100]^T)</td>
<td>([0 \ 0 \ 0]^T)</td>
<td>([100 \ 0 \ 0]^T)</td>
<td>([0 \ 0 \ 0]^T)</td>
<td>0.6</td>
</tr>
<tr>
<td>6</td>
<td>([100 \ 0 \ 0]^T)</td>
<td>([0 \ 0 \ 0]^T)</td>
<td>([-100 \ 0 \ 0]^T)</td>
<td>([0 \ 0 \ 0]^T)</td>
<td>0.6</td>
</tr>
</tbody>
</table>
Results with Circular Chief Orbit and with Perturbations

The resulting trajectories from simulation for the first three examples, with final time $M = 0.3$ are shown in Fig. 13.10, and the resulting control history and tracking errors are shown in Fig. 13.11 and 13.12, respectively.

![Fig. 13.10: Scenario 1, Case 2, Examples 1-3 trajectories (Cartesian model, with perturbations)](image)

![Fig. 13.11: Scenario 1, Case 2, Examples 1-3 control history (Cartesian model, circular orbit, with perturbations)](image)
Fig. 13.12: Scenario 1, Case 2, Examples 1-3 tracking error (Cartesian model, circular orbit, with perturbations)

The next three figures show the results for the examples with final time $M = 0.6$ revolutions. The trajectories are shown in Fig. 13.13, and the resulting control history and tracking errors are shown in Fig. 13.14 and 13.15, respectively.

Fig. 13.13: Scenario 1, Case 2, Examples 1-3 trajectories (Cartesian model, with perturbations)
Fig. 13.14: Scenario 1, Case 2, Examples 1-3 control history (Cartesian model, circular orbit, with perturbations)

Fig. 13.15: Scenario 1, Case 2, Examples 1-3 tracking error (Cartesian model, circular orbit, with perturbations)

The $\Delta v$ for the planner and simulation results for Scenario 1, Case 1, with a circular chief and perturbations, are given in Table 13.8. In all cases the $\Delta v$ differences are small, though the effects of longer transfer times is evident.
Table 13.8: Scenario 1, Case 2 with circular chief and perturbations, \( \Delta v \) from planner and simulation with Cartesian model

<table>
<thead>
<tr>
<th>Example</th>
<th>Form.</th>
<th>( \Delta v_{\text{opt}} ) (m/s)</th>
<th>( \Delta v_{\text{sim}} ) (m/s)</th>
<th>Diff. (m/s)</th>
<th>( p_D ) (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Cart.</td>
<td>0.195</td>
<td>0.196</td>
<td>0.001</td>
<td>0.33</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>0.379</td>
<td>0.381</td>
<td>0.002</td>
<td>0.43</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>0.285</td>
<td>0.286</td>
<td>0.001</td>
<td>0.29</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>0.140</td>
<td>0.141</td>
<td>0.002</td>
<td>1.11</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>0.377</td>
<td>0.379</td>
<td>0.002</td>
<td>0.61</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>0.259</td>
<td>0.260</td>
<td>0.001</td>
<td>0.40</td>
</tr>
</tbody>
</table>

**Results with Elliptic Chief Orbit and with Perturbations**

The Case 2 way-point following examples are also examined under the effects of varying chief orbit eccentricity, as in Case 1. The eccentricities examined include \( e_c = 0.001 \) and \( e_c = 0.01 \). The effects of \( e_c = 0.0001 \) in these scenarios is insignificant. Example control and tracking error plots are included in Fig. 13.16 and 13.17, respectively, for the longer final time, Cases 4-6. The \( \Delta v \) in Table 13.9 shows the results for varying eccentricity. Once again, the \( \Delta v \) differences are small, though the effects of the chief eccentricity is evident.

![Control History](image)

Fig. 13.16: Scenario 1, Case 2 control history (Cartesian model, with perturbations and chief eccentricity of \( e_c = 0.01 \))
Fig. 13.17: Scenario 1, Case 2 tracking error (Cartesian model, with perturbations and chief eccentricity of $e_c = 0.01$)

Table 13.9: Scenario 1, Case 2 with varying chief ellipticity and perturbations, $\Delta v$ from planner and simulation with Cartesian model

<table>
<thead>
<tr>
<th>Example</th>
<th>Ecc. ($e_c$)</th>
<th>$\Delta v_{opt}$ (m/s)</th>
<th>$\Delta v_{sim}$ (m/s)</th>
<th>Diff. (m/s)</th>
<th>$PD$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.001</td>
<td>0.195</td>
<td>0.196</td>
<td>0.001</td>
<td>0.26</td>
</tr>
<tr>
<td>2</td>
<td>0.001</td>
<td>0.379</td>
<td>0.380</td>
<td>0.001</td>
<td>0.28</td>
</tr>
<tr>
<td>3</td>
<td>0.001</td>
<td>0.285</td>
<td>0.287</td>
<td>0.002</td>
<td>0.61</td>
</tr>
<tr>
<td>4</td>
<td>0.001</td>
<td>0.140</td>
<td>0.141</td>
<td>0.001</td>
<td>0.84</td>
</tr>
<tr>
<td>5</td>
<td>0.001</td>
<td>0.377</td>
<td>0.379</td>
<td>0.002</td>
<td>0.50</td>
</tr>
<tr>
<td>6</td>
<td>0.001</td>
<td>0.258</td>
<td>0.260</td>
<td>0.002</td>
<td>0.66</td>
</tr>
<tr>
<td>1</td>
<td>0.01</td>
<td>0.195</td>
<td>0.197</td>
<td>0.002</td>
<td>1.18</td>
</tr>
<tr>
<td>2</td>
<td>0.01</td>
<td>0.379</td>
<td>0.378</td>
<td>-0.001</td>
<td>0.32</td>
</tr>
<tr>
<td>3</td>
<td>0.01</td>
<td>0.285</td>
<td>0.296</td>
<td>0.011</td>
<td>3.85</td>
</tr>
<tr>
<td>4</td>
<td>0.01</td>
<td>0.140</td>
<td>0.144</td>
<td>0.005</td>
<td>3.22</td>
</tr>
<tr>
<td>5</td>
<td>0.01</td>
<td>0.377</td>
<td>0.382</td>
<td>0.004</td>
<td>1.12</td>
</tr>
<tr>
<td>6</td>
<td>0.01</td>
<td>0.259</td>
<td>0.271</td>
<td>0.012</td>
<td>4.65</td>
</tr>
</tbody>
</table>

13.5 Case 3 - Final Approach

The final case, Case 3, for Scenario 1 is the final approach. This case is analyzed in a manner similar to Case 1 and 2, where simulations are run to determine the overall effects of perturbations and chief eccentricity. This case includes a cone approach corridor of half-
angle $\alpha = 20$ degrees. The Cartesian model is used due to the close proximity of the deputy to the chief in this case.

Two main examples are tested, each with two different final transfer times. All initial and final conditions are specified in terms of the relative position and velocity, where the desired relative velocity at the beginning and end of the trajectory is zero. The first examples are an approach from 100 m on the v-bar to 20 m on the v-bar, with a final time of 30% of an orbit and 60% of an orbit. The second examples are an r-bar approach from 100 m radial to 20 m radial, and also with a final time of 30% of an orbit and 60% of an orbit. All initial and final boundary conditions for the deputy, and the final transfer times, are shown in Table 13.10.

<table>
<thead>
<tr>
<th>Example</th>
<th>$r_{rel}(t_0)$, (m)</th>
<th>$v_{rel}(t_0)$, (m/s)</th>
<th>$r_{rel}(t_f)$, (m)</th>
<th>$v_{rel}(t_f)$, (m/s)</th>
<th>M (rev.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$[0 100 0]^T$</td>
<td>$[0 0 0]^T$</td>
<td>$[0 20 0]^T$</td>
<td>$[0 0 0]^T$</td>
<td>0.3</td>
</tr>
<tr>
<td>2</td>
<td>$[100 0 0]^T$</td>
<td>$[0 0 0]^T$</td>
<td>$[20 0 0]^T$</td>
<td>$[0 0 0]^T$</td>
<td>0.3</td>
</tr>
<tr>
<td>3</td>
<td>$[0 100 0]^T$</td>
<td>$[0 0 0]^T$</td>
<td>$[0 20 0]^T$</td>
<td>$[0 0 0]^T$</td>
<td>0.6</td>
</tr>
<tr>
<td>4</td>
<td>$[100 0 0]^T$</td>
<td>$[0 0 0]^T$</td>
<td>$[20 0 0]^T$</td>
<td>$[0 0 0]^T$</td>
<td>0.6</td>
</tr>
</tbody>
</table>

**Results with Circular Chief and with Perturbations**

The trajectories that resulted from Case 3, circular chief orbit, are shown in Fig. 13.18, and the associated control histories and tracking errors are shown in Fig. 13.19 and 13.20, respectively.
Fig. 13.18: Scenario 1, Case 3, Examples 1-4 trajectories (Cartesian model, with perturbations)

Fig. 13.19: Scenario 1, Case 3, Examples 1-4 control history (Cartesian model, circular orbit, with perturbations)
Fig. 13.20: Scenario 1, Case 3, Examples 1-4 tracking error (Cartesian model, circular orbit, with perturbations)

The results for all $\Delta v$'s in this case are presented in Table 13.11. In all cases, the $\Delta v$ differences are small.

Table 13.11: Scenario 1, Case 3 with circular chief and perturbations, $\Delta v$ from planner and simulation with Cartesian model

<table>
<thead>
<tr>
<th>Example</th>
<th>Form.</th>
<th>$\Delta v_{opt}$ (m/s)</th>
<th>$\Delta v_{sim}$ (m/s)</th>
<th>Diff. (m/s)</th>
<th>$PD$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>0.143</td>
<td>0.143</td>
<td>0.000</td>
<td>0.09</td>
</tr>
<tr>
<td>2</td>
<td>Cart.</td>
<td>0.379</td>
<td>0.380</td>
<td>0.002</td>
<td>0.40</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>0.061</td>
<td>0.062</td>
<td>0.001</td>
<td>0.99</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>0.411</td>
<td>0.413</td>
<td>0.002</td>
<td>0.44</td>
</tr>
</tbody>
</table>

Results with Elliptic Chief Orbit and with Perturbations

Case 3 final approach results are also analyzed for varying chief eccentricity. In these results, a chief eccentricity of $e_c = 0.001$ and $e_c = 0.01$ are included. Example control and tracking error plots are included in Fig. 13.21 and 13.22, respectively, for Cases 1-4. The $\Delta v$ in Table 13.12 shows the results for varying eccentricity. Once again, the $\Delta v$ differences are small, but the effect of chief eccentricity is evident.
Fig. 13.21: Scenario 1, Case 3, Examples 1-4 control history (Cartesian model, with perturbations and chief eccentricity of $e_c = 0.01$)

Fig. 13.22: Scenario 1, Case 3, Examples 1-4 tracking error (Cartesian model, with perturbations and chief eccentricity of $e_c = 0.01$)
Table 13.12: Scenario 1, Case 3 with varying chief ellipticity and perturbations, $\Delta v$ from planner and simulation with Cartesian model

<table>
<thead>
<tr>
<th>Example</th>
<th>Ecc. $(e_c)$</th>
<th>$\Delta v_{opt}$ (m/s)</th>
<th>$\Delta v_{sim}$ (m/s)</th>
<th>Diff. (m/s)</th>
<th>$p_D$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.001</td>
<td>0.143</td>
<td>0.143</td>
<td>0.000</td>
<td>0.09</td>
</tr>
<tr>
<td>2</td>
<td>0.01</td>
<td>0.379</td>
<td>0.381</td>
<td>0.003</td>
<td>0.88</td>
</tr>
<tr>
<td>3</td>
<td>0.062</td>
<td>0.062</td>
<td>0.062</td>
<td>0.000</td>
<td>0.61</td>
</tr>
<tr>
<td>4</td>
<td>0.411</td>
<td>0.411</td>
<td>0.414</td>
<td>0.004</td>
<td>0.85</td>
</tr>
</tbody>
</table>

13.6 Conclusions

The resulting comparison between using the reference trajectory generated in Cartesian versus spherical coordinates, in Table 13.4, shows that there is clearly greater accuracy when using the spherical model for cases with large along-track separation, and absence of perturbations ($J_2$/drag/SRP). By representing the position and velocity in a curvilinear LVLH frame, the modeling errors associated with a Cartesian LVLH frame are recaptured, and the results more closely model the chief’s true reference orbit.

From the results shown in Table 13.6, it is evident that the ellipticity of the chief plays a major role in Case 1, when considering values greater than $e_c = 0.001$. This can be seen by the large increase in required $\Delta v$ for trajectory following, from $e_c = 0.001$ to $e_c = 0.01$, and in the tracking error plots for $e_c = 0.01$ shown in Fig. 13.9. In these cases, and for more elliptic target orbits, the orbital model in the planner must be improved to account for these effects.

The results in Table 13.9 show that in Case 2, the effects of eccentricity are minimal. This can be seen by the percent difference error between the planned and simulated trajectories. The primary variables that drive the effects of percent difference error (from the results of Case 1 and 2) are the initial/final distance from the chief, and the final transfer time. When tracking a trajectory at large distances from the chief (with relatively large final times), the benefits of adding eccentricity/perturbations into the dynamics model may
help to improve the overall performance. However, when chief/deputy separation distances are small (and with relatively small transfer times), the HCW model alone performs very well, with minimal percent difference errors.

Case 3 nonlinear simulation results show a similar trend to those of Case 2. It can be seen by the $\Delta v$ results in Table 13.12 that the effects of eccentricity on these final approach trajectories is minimal. This is again due to the fact that in these examples the deputy is in relatively close proximity of the chief and the final transfer times are small.

As a reminder, this entire analysis (and the concluding remarks) is based on the assumption of perfect navigation and control. Additionally, the controller in this simulation has a relatively slow response time. This was done on purpose, since implementing a faster controller led to much larger $\Delta v$ percent difference errors.
CHAPTER 14
PERFORMANCE ANALYSIS II, SAFE RENDEZVOUS, TRAVELING ELLIPSE, AND SAFE FINAL APPROACH - SCENARIO 2

14.1 Scenario 2 Methodology

The second scenario focuses on the safety of flight cases. This scenario is termed the safe rendezvous, traveling ellipse, and safe final approach scenario. The initial states are interpreted as relative ellipses, and are specified in terms of either the initial relative orbital elements or initial LROEs (one can be easily computed from the other via Eqs. 5.38-5.40), and final states are specified via final ROEs. This scenario also includes three standard cases, named Safe Initial Approach, Safe Traveling Ellipse, and Safe Final Approach. The three cases for each scenario include multiple examples which are described in greater detail in each of their respective following sections. The scenarios and cases to be presented are summarized here.

**Scenario 2** - Safe Rendezvous, Traveling Ellipse, and Final Approach

- **Case 1** - Safe Initial Approach
- **Case 2** - Safe Traveling Ellipse
- **Case 3** - Safe Final Approach

The following sections describe the method by which the Scenario 2 nonlinear simulation results are analyzed, and how the trajectories for each case are initialized. These are separated into the analysis setup and initial conditions sections.

14.2 Scenario 2 Analysis Setup

The analysis for the safe rendezvous, safe traveling ellipse, and safe final approach scenario (Scenario 2) includes nominal trajectories for each case, as well as control histories and tracking errors. In each control history plot, the simulation control history is plotted.
A table is included for each case, that shows the optimal $\Delta v$ from the planner and the nonlinear simulation $\Delta v$ in each case.

This analysis compares the performance of reference trajectories generated in Cartesian coordinates versus trajectories generated in spherical coordinates. As previously noted, it is expected that the spherical reference trajectory will perform much better in the cases that have large along-track and/or cross-track separation. As will be seen, the results show that the spherical model outperforms the Cartesian model in both Case 1 and Case 2 for Scenario 2.

The next objective is to show the effects of eccentricity and perturbations in the Scenario 2 cases. The effects of $J_2$ and drag perturbations are shown by plotting trajectories and control histories, and including $\Delta v$ tables for simulations with and without perturbations. Similarly, the effects of eccentricity are shown by plotting trajectories and control histories, and including $\Delta v$ tables for multiple reference orbits with varying eccentricity. The $\Delta v$ tables are used to compare the optimal planner $\Delta v$ to the simulation $\Delta v$ for a variety of cases. The $\Delta v$ tables include the optimal $\Delta v$, $\Delta v_{opt}$, the simulation $\Delta v$, $\Delta v_{sim}$, and the percent difference between the two, $p_D$, where percent difference is given in Eq. 13.1.

### 14.2.1 Simulation Parameters

The simulation parameters in Scenario 2 are defined exactly as in Scenario 1. All important vehicle parameters are shown previously in Table 13.1, and the solar radiation pressure and sun vector are provided in Section 13.2.

### 14.2.2 Initial Conditions

In Scenario 2, the same method of initializing the trajectory in the nonlinear simulation is used, as was used in Scenario 1. The first case (Safe Initial Approach) requires an initial nominal flyby trajectory. Thus, Case 1 is initialized using nominal traditional relative orbital elements for the planned flyby orbit. The traditional ROEs ($\delta e$) used in this case are given in Eq. 13.3. It follows that Cases 2 and 3 of Scenario 2 are initialized using the nominal HCW ROEs that result from Case 1 and Case 2, respectively. Note that these ROEs can
be easily converted to relative position and velocity in LVLH, however in these cases it is simpler to specify initial conditions in terms of the ROEs. These ROE values are given for Case 2 and Case 3 in each of the specific results sections to follow.

14.3 Case 1 - Safe Initial Approach

The Scenario 2 analysis begins with the Case 1, Safe Initial Approach. In this case, the deputy is initialized by specifying the set of nominal traditional relative orbital elements, $\delta \epsilon$, as shown in Eq. 13.3. Four examples are included in this analysis. These examples consist of two different flyby examples; the first is 300 m below and 2500 m behind the chief, and the second is 600 m below and 5000 m behind the chief.

The final conditions are specified in terms of final relative orbital elements, and the final transfer time. The final passive abort safety ellipses that are specified in these examples have a 100 m semi-major axis and 50 m cross-track amplitude. The final center of motion $[x_r(t_f), y_r(t_f)]$ is either (0, 500) m or (0, 1000) m, depending on the example. The final transfer time is specified via the number of chief revolutions, $M$, and is given by Eq. 13.4. In the Case 1 examples, the trajectories generated were all passively safe on the the first iteration, therefore sequential convex programming was not needed.

All initial and final conditions for each example are shown in Table 14.1. The final conditions consist of two different final times; 1.2 revolutions and 2.4 revolutions. These examples implement a maximum control acceleration constraint of 0.5 mm/s$^2$.

<table>
<thead>
<tr>
<th>Example</th>
<th>$\delta \epsilon(t_0)$ (km)</th>
<th>$[x_r \ y_r \ a_r \ A_z \ E_r \ \psi_z]^T(t_f)$ (m)</th>
<th>M (rev.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$[-0.3/a_c \ 0 \ 0 \ 0 \ -2.5/a_d]^T$</td>
<td>$[0 \ 500 \ 100 \ 50 \ \pi \ \pi]^T$</td>
<td>1.2</td>
</tr>
<tr>
<td>2</td>
<td>$[-0.6/a_c \ 0 \ 0 \ 0 \ -5.0/a_d]^T$</td>
<td>$[0 \ 1000 \ 100 \ 50 \ \pi \ \pi]^T$</td>
<td>1.2</td>
</tr>
<tr>
<td>3</td>
<td>$[-0.3/a_c \ 0 \ 0 \ 0 \ -2.5/a_d]^T$</td>
<td>$[0 \ 500 \ 100 \ 50 \ \pi \ \pi]^T$</td>
<td>2.4</td>
</tr>
<tr>
<td>4</td>
<td>$[-0.6/a_c \ 0 \ 0 \ 0 \ -5.0/a_d]^T$</td>
<td>$[0 \ 1000 \ 100 \ 50 \ \pi \ \pi]^T$</td>
<td>2.4</td>
</tr>
</tbody>
</table>
Results with Circular Chief Orbit and without Perturbations

The first results for the Scenario 2, Case 1, Examples 1-4, show the trajectories and controls without perturbations, using a reference trajectory in Cartesian coordinates. The trajectories in this case are shown in Fig. 14.1, the associated controls are shown in Fig. 14.2, and the simulation position and velocity tracking errors are shown in Fig. 14.3.

Fig. 14.1: Scenario 2, Case 1 trajectories (Cartesian model, without perturbations)

Fig. 14.2: Scenario 2, Case 1 control history (Cartesian model, circular chief, without perturbations)
These four examples were also run using a reference trajectory generated in spherical coordinates. The resulting trajectories in LVLH look very similar, and are omitted. The control history and tracking errors show the difference due to modeling error, which include large along-track separation. The control history for the spherical case without perturbations is shown in Fig. 14.4, and the associated tracking error is shown in Fig. 14.5.
Fig. 14.5: Scenario 2, Case 1 tracking error (spherical model, circular chief, without perturbations)

The $\Delta v$ results for Scenario 2, Case 1, and all examples (including Cartesian and spherical models), are presented in Table 14.2 below.

Table 14.2: Scenario 2, Case 1 with circular chief and without perturbations, $\Delta v$ from planner and simulation for Cartesian and spherical coordinates

<table>
<thead>
<tr>
<th>Example</th>
<th>Form.</th>
<th>$\Delta v_{opt}$ (m/s)</th>
<th>$\Delta v_{sim}$ (m/s)</th>
<th>Diff. (m/s)</th>
<th>$p_D$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Cart.</td>
<td>0.209</td>
<td>0.212</td>
<td>0.003</td>
<td>1.22</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>0.403</td>
<td>0.413</td>
<td>0.010</td>
<td>2.46</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>0.168</td>
<td>0.172</td>
<td>0.003</td>
<td>1.71</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>0.326</td>
<td>0.338</td>
<td>0.011</td>
<td>3.41</td>
</tr>
<tr>
<td>1</td>
<td>Sph.</td>
<td>0.210</td>
<td>0.210</td>
<td>0.000</td>
<td>0.17</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>0.406</td>
<td>0.408</td>
<td>0.001</td>
<td>0.34</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>0.168</td>
<td>0.168</td>
<td>0.000</td>
<td>0.15</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>0.323</td>
<td>0.324</td>
<td>0.001</td>
<td>0.29</td>
</tr>
</tbody>
</table>

While the differences between the planner and simulation $\Delta v$’s are all small, the effect of using spherical coordinates is evident.
Results with Circular Chief Orbit and with Perturbations

The Scenario 2, Case 1 examples were also run including perturbations. The resulting control history and tracking error using the spherical model and including perturbations is shown in Fig. 14.6 and 14.7, respectively. The $\Delta v$ for the Cartesian and spherical examples with perturbations are also shown in Table 14.3.

Fig. 14.6: Scenario 2, Case 1 control history (spherical model, circular chief, with perturbations)

Fig. 14.7: Scenario 2, Case 1 tracking error (spherical model, circular chief, with perturbations)
Table 14.3: Scenario 2, Case 1 with circular chief and perturbations, \( \Delta v \) from planner and simulation for Cartesian and spherical coordinates

<table>
<thead>
<tr>
<th>Example</th>
<th>Form.</th>
<th>( \Delta v_{opt} ) (m/s)</th>
<th>( \Delta v_{sim} ) (m/s)</th>
<th>Diff. (m/s)</th>
<th>( p_D ) (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Cart.</td>
<td>0.209</td>
<td>0.235</td>
<td>0.025</td>
<td>11.34</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>0.403</td>
<td>0.449</td>
<td>0.047</td>
<td>10.99</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>0.169</td>
<td>0.197</td>
<td>0.028</td>
<td>15.43</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>0.326</td>
<td>0.379</td>
<td>0.053</td>
<td>14.95</td>
</tr>
<tr>
<td>1</td>
<td>Sph.</td>
<td>0.210</td>
<td>0.237</td>
<td>0.027</td>
<td>12.05</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>0.406</td>
<td>0.459</td>
<td>0.053</td>
<td>12.19</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>0.168</td>
<td>0.198</td>
<td>0.030</td>
<td>16.47</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>0.323</td>
<td>0.382</td>
<td>0.059</td>
<td>16.87</td>
</tr>
</tbody>
</table>

With the addition of perturbations, the \( \Delta v \) differences range from 10-17\%, and the effects of using spherical coordinates is negligible.

**Results with Elliptic Chief and with Perturbations**

The effects of eccentricity of the chief’s orbit is examined for Scenario 2 Case 1. The previous four examples were run with chief eccentricities (\( e_c \)) of 0.0001, 0.001 and 0.01, and also including perturbations. The spherical model was used for these cases. Example control history and tracking error plots for for \( e_c = 0.01 \) are shown in Fig. 14.8 and 14.9. The \( \Delta v \) values for each eccentricity are shown in Table 14.4.

![Control History](image.png)

**Fig. 14.8:** Scenario 2, Case 1 control history (spherical model, with perturbations and chief eccentricity of \( e_c = 0.01 \))
Fig. 14.9: Scenario 2, Case 1 tracking error (spherical model, with perturbations and chief eccentricity of $e_c = 0.01$)

Table 14.4: Scenario 2, Case 1 with varying chief ellipticity and perturbations, $\Delta v$ from planner and simulation with spherical model

<table>
<thead>
<tr>
<th>Example</th>
<th>Ecc. ($e_c$)</th>
<th>$\Delta v_{opt}$ (m/s)</th>
<th>$\Delta v_{sim}$ (m/s)</th>
<th>Diff. (m/s)</th>
<th>$p_D$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0001</td>
<td>0.210</td>
<td>0.236</td>
<td>0.026</td>
<td>11.76</td>
</tr>
<tr>
<td>2</td>
<td>0.001</td>
<td>0.406</td>
<td>0.457</td>
<td>0.051</td>
<td>11.81</td>
</tr>
<tr>
<td>3</td>
<td>0.0001</td>
<td>0.167</td>
<td>0.196</td>
<td>0.029</td>
<td>15.82</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>0.322</td>
<td>0.380</td>
<td>0.058</td>
<td>16.44</td>
</tr>
<tr>
<td>1</td>
<td>0.01</td>
<td>0.210</td>
<td>0.234</td>
<td>0.023</td>
<td>10.44</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>0.408</td>
<td>0.449</td>
<td>0.041</td>
<td>9.67</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>0.167</td>
<td>0.192</td>
<td>0.025</td>
<td>13.77</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>0.321</td>
<td>0.369</td>
<td>0.048</td>
<td>13.88</td>
</tr>
<tr>
<td>1</td>
<td>0.01</td>
<td>0.219</td>
<td>0.362</td>
<td>0.143</td>
<td>49.18</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>0.423</td>
<td>0.654</td>
<td>0.231</td>
<td>42.84</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>0.160</td>
<td>0.356</td>
<td>0.196</td>
<td>76.03</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>0.306</td>
<td>0.691</td>
<td>0.385</td>
<td>77.30</td>
</tr>
</tbody>
</table>

As can be seen, the $\Delta v$ differences are generally small (<15%), except in the high eccentricity case ($e_c = 0.01$).
14.4 Case 2 - Safe Traveling Ellipse

This section presents results for Scenario 2, Case 2, with safe traveling ellipse trajectories. This includes control histories, tracking errors, and $\Delta v$ tables for several examples. In this case, all examples presented are initialized using the final ROEs from case 1. This case is examined in a method similar to Case 1, by first presenting results under perturbed orbits, then taking a brief look at the effects of chief eccentricity. A maximum control acceleration constraint of 0.5 mm/s$^2$ was used in this case.

The examples for Case 2 are set up for the traveling safety ellipse, where the chief is in a circular orbit and including perturbations. These trajectories consist of moving from a $100 \times 50$ m safety ellipse either at 1 km on the v-bar, or 500 m on the v-bar, to a stationary circumnavigating $100 \times 50$ m safety ellipse. Similar to the planner results in Chapter 9, passive safety of flight constraints are included, and the terminal phase angle is relaxed (constrained such that the final ellipse is in-phase by $\gamma \leq 0.1\pi$). The sequential convex programming method is required, due to the nonconvex final conditions, and since the trajectories generated on the first iteration were not passively safe.

Two final times are considered, one completes each trajectory in 60% of a chief revolution, and the other completes each trajectory in 240% of a chief revolution. Thus, a total of four examples are examined, and each of the initial/final conditions and final times are shown in Table 14.5.

Table 14.5: Deputy initial and final ROEs for Scenario 2, Case 2

<table>
<thead>
<tr>
<th>Example</th>
<th>$x_r$ (m)</th>
<th>$y_r$ (m)</th>
<th>$a_r$ (m)</th>
<th>$A_z$ (m)</th>
<th>$E_r$ (rad)</th>
<th>$\psi_z$ (rad)</th>
<th>M (rev.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Initial</td>
<td>0</td>
<td>1000</td>
<td>100</td>
<td>50</td>
<td>$0.5\pi$</td>
<td>$0.4\pi$</td>
<td>0.6</td>
</tr>
<tr>
<td>1 Final</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>50</td>
<td>$\gamma \leq 0.1\pi$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 Initial</td>
<td>0</td>
<td>500</td>
<td>100</td>
<td>50</td>
<td>$0.5\pi$</td>
<td>$0.4\pi$</td>
<td>0.6</td>
</tr>
<tr>
<td>2 Final</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>50</td>
<td>$\gamma \leq 0.1\pi$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 Initial</td>
<td>0</td>
<td>1000</td>
<td>100</td>
<td>50</td>
<td>$0.5\pi$</td>
<td>$0.4\pi$</td>
<td>2.4</td>
</tr>
<tr>
<td>3 Final</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>50</td>
<td>$\gamma \leq 0.1\pi$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 Initial</td>
<td>0</td>
<td>500</td>
<td>100</td>
<td>50</td>
<td>$0.5\pi$</td>
<td>$0.4\pi$</td>
<td>2.4</td>
</tr>
<tr>
<td>4 Final</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>50</td>
<td>$\gamma \leq 0.1\pi$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Results with Circular Chief Orbit and with Perturbations

The first results for the Scenario 2, Case 2, Examples 1-4, show the trajectories and controls without perturbations, using reference trajectories in spherical and Cartesian coordinates. The trajectories in this case are all passively safe, as shown in Fig. 14.10. The associated controls are shown in Fig. 14.11 and the simulation position and velocity tracking errors for spherical and Cartesian are shown in Fig. 14.12 and Fig. 14.13, respectively.

Fig. 14.10: Scenario 2, Case 2 trajectories (spherical model, with perturbations)

Fig. 14.11: Scenario 2, Case 2 control history (spherical model, with perturbations)
Fig. 14.12: Scenario 2, Case 2 tracking error (spherical model, with perturbations)

Fig. 14.13: Scenario 2, Case 2 tracking error (Cartesian model, with perturbations)

The Δv results for Case 2, Examples 1-4, and for the Cartesian and spherical coordinate models are shown in Table 14.6.
Table 14.6: Scenario 2, Case 2 with circular chief and perturbations, \( \Delta v \) from planner and simulation, and SCP iterations, for Cartesian and spherical examples

<table>
<thead>
<tr>
<th>Example</th>
<th>Form.</th>
<th>SCP Iter.</th>
<th>( \Delta v_{opt} ) (m/s)</th>
<th>( \Delta v_{sim} ) (m/s)</th>
<th>Diff. (m/s)</th>
<th>( p_D ) (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Cart.</td>
<td>6</td>
<td>0.424</td>
<td>0.432</td>
<td>0.008</td>
<td>1.95</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>0.159</td>
<td>0.164</td>
<td>0.005</td>
<td>3.21</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>0.053</td>
<td>0.069</td>
<td>0.016</td>
<td>26.02</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>0.027</td>
<td>0.037</td>
<td>0.011</td>
<td>33.30</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
<th>Sph.</th>
<th>6</th>
<th>0.423</th>
<th>0.432</th>
<th>0.009</th>
<th>2.06</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>6</td>
<td>0.158</td>
<td>0.163</td>
<td>0.005</td>
<td>3.21</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>0.053</td>
<td>0.068</td>
<td>0.016</td>
<td>25.94</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>0.027</td>
<td>0.037</td>
<td>0.011</td>
<td>33.41</td>
<td></td>
</tr>
</tbody>
</table>

While the \( \Delta v \) differences are small for small transfer times (0.6 revs.), the \( \Delta v \) differences increase for the large transfer times (2.4 revs.). The effects of using spherical coordinates is negligible.

**Results with Elliptic Chief Orbit and with Perturbations**

The Case 2 safe traveling ellipse examples are examined under the effects of varying chief orbit eccentricity, as in Case 1. The eccentricities examined include \( e_c = 0.0001 \), \( e_c = 0.001 \), and \( e_c = 0.01 \). Example control and tracking error plots using the spherical model are included in Fig. 14.14 and 14.15, respectively. The \( \Delta v \) table in Table 14.7 shows the results for varying eccentricity.

![Control History](image)

Fig. 14.14: Scenario 2, Case 2 control history (spherical model, with perturbations and chief eccentricity of \( e_c = 0.01 \))
Fig. 14.15: Scenario 2, Case 2 tracking error (spherical model, with perturbations and chief eccentricity of $e_c = 0.01$)

Table 14.7: Scenario 2, Case 2 with varying chief ellipticity and perturbations, $\Delta v$ from planner and simulation with spherical model

<table>
<thead>
<tr>
<th>Example</th>
<th>Ecc. ($e_c$)</th>
<th>$\Delta v_{opt}$ (m/s)</th>
<th>$\Delta v_{sim}$ (m/s)</th>
<th>Diff. (m/s)</th>
<th>$p_D$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0001</td>
<td>0.423</td>
<td>0.432</td>
<td>0.008</td>
<td>1.98</td>
</tr>
<tr>
<td>2</td>
<td>0.158</td>
<td>0.163</td>
<td>0.005</td>
<td>3.06</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.052</td>
<td>0.068</td>
<td>0.015</td>
<td>24.60</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.027</td>
<td>0.037</td>
<td>0.010</td>
<td>32.22</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.001</td>
<td>0.424</td>
<td>0.430</td>
<td>0.006</td>
<td>1.38</td>
</tr>
<tr>
<td>2</td>
<td>0.159</td>
<td>0.162</td>
<td>0.003</td>
<td>1.99</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.054</td>
<td>0.067</td>
<td>0.014</td>
<td>22.98</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.027</td>
<td>0.035</td>
<td>0.008</td>
<td>26.22</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.01</td>
<td>0.429</td>
<td>0.473</td>
<td>0.044</td>
<td>9.83</td>
</tr>
<tr>
<td>2</td>
<td>0.161</td>
<td>0.184</td>
<td>0.023</td>
<td>13.47</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.061</td>
<td>0.198</td>
<td>0.137</td>
<td>105.94</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.031</td>
<td>0.091</td>
<td>0.060</td>
<td>98.02</td>
<td></td>
</tr>
</tbody>
</table>

It can be seen that in these examples, the major $\Delta v$ differences are due to the overall final transfer time, as well as the eccentricity of the chief’s orbit.
14.5 Case 3 - Safe Final Approach

The final case, Case 3, for Scenario 2 is the Safe Final Approach. This case is analyzed in a manner similar to Case 1 and 2, where simulations are run to determine the overall effects of perturbations and chief eccentricity. The Cartesian model is used in this case, due to the close proximity of the deputy to the chief.

Two main examples are tested, each with two different final transfer times. All initial and final conditions are specified in terms of the relative orbital elements, where the final phase angles are specified to be in-phase using the in-plane natural drift, \( \theta_f = E_{r_0} + o_{t_f} \) (this is one method to preserve in-plane lighting conditions). The initial conditions for this case are based on the final conditions from Case 2. This case results in transfers from a large to a smaller safety ellipse, from results in Chapter 9. These case results are fully convex, since all final conditions are specified and the trajectories analyzed are inherently passively safe. The first example is an approach from a 100\( \times \)50 m safety ellipse to a 50\( \times \)25 m safety ellipse, with a final time of 60\% of an orbit and 120\% of an orbit. The second is an approach from a 200\( \times \)100 m safety ellipse to a 100\( \times \)50 m safety ellipse, and also with a final time of 60\% of an orbit and 120\% of an orbit. All initial and final boundary conditions for the deputy, and the final transfer times, are shown in Table 14.8.

| Table 14.8: Deputy initial and final ROEs for Scenario 2, Case 2 |
|---|---|---|---|---|---|---|---|
| Example | Initial | Final |
| 1 | Initial | 0 | 0 | 100 | 50 | 1.9\( \pi \) | 1.8\( \pi \) | 1.2 |
| | Final | 0 | 0 | 50 | 25 | 0.5\( \pi \) | 0.5\( \pi \) | 0.6 |
| 2 | Initial | 0 | 0 | 200 | 100 | 1.9\( \pi \) | 1.8\( \pi \) | 0.6 |
| | Final | 0 | 0 | 100 | 50 | 0.5\( \pi \) | 0.5\( \pi \) | 0.6 |
| 3 | Initial | 0 | 0 | 100 | 50 | 1.9\( \pi \) | 1.8\( \pi \) | 1.2 |
| | Final | 0 | 0 | 50 | 25 | 1.1\( \pi \) | 1.1\( \pi \) | 1.2 |
| 4 | Initial | 0 | 0 | 200 | 100 | 1.9\( \pi \) | 1.8\( \pi \) | 1.2 |
| | Final | 0 | 0 | 100 | 50 | 1.1\( \pi \) | 1.1\( \pi \) | 1.2 |
Results with Circular Chief Orbit and with Perturbations

The trajectories that resulted from Case 3 are shown in Fig. 14.16, and the associated control histories and tracking errors are shown in Fig. 14.17 and 14.18, respectively.

Fig. 14.16: Scenario 2, Case 3, Examples 1-4 trajectories (Cartesian model, with perturbations)

Fig. 14.17: Scenario 2, Case 3, Examples 1-4 control history (Cartesian model, with perturbations)
The results for all \( \Delta v \) in this case are presented in Table 14.9, where again, the effects of larger final transfer times give higher overall \( \Delta v \) differences.

<table>
<thead>
<tr>
<th>Example</th>
<th>Form.</th>
<th>( \Delta v_{opt} ) (m/s)</th>
<th>( \Delta v_{sim} ) (m/s)</th>
<th>Diff. (m/s)</th>
<th>( p_D ) (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Cart.</td>
<td>0.047</td>
<td>0.048</td>
<td>0.001</td>
<td>1.55</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>0.095</td>
<td>0.097</td>
<td>0.001</td>
<td>1.47</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>0.034</td>
<td>0.036</td>
<td>0.002</td>
<td>5.02</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>0.069</td>
<td>0.072</td>
<td>0.003</td>
<td>4.66</td>
</tr>
</tbody>
</table>

Results with Elliptic Chief Orbit and with Perturbations

Case 3 safe final approach results are also analyzed for varying chief eccentricity. In these results, a chief eccentricity of \( e_c = 0.001 \) and \( e_c = 0.01 \) are included (The effects of \( e_c = 0.0001 \) are minimal). Example control and tracking error plots are included in Fig. 14.19 and 14.20, respectively. The \( \Delta v \) table in Table 14.10 shows the results for varying eccentricity.
Fig. 14.19: Scenario 2, Case 3, Examples 1-4 control history (Cartesian model, with perturbations and chief eccentricity of $e_c = 0.01$)

Fig. 14.20: Scenario 2, Case 3, Examples 1-4 tracking error (Cartesian model, with perturbations and chief eccentricity of $e_c = 0.01$)
Table 14.10: Scenario 2, Case 3 with varying chief ellipticity and perturbations, $\Delta v$ from planner and simulation with Cartesian model

<table>
<thead>
<tr>
<th>Example</th>
<th>Ecc. ($e_c$)</th>
<th>$\Delta v_{opt}$ (m/s)</th>
<th>$\Delta v_{sim}$ (m/s)</th>
<th>Diff. (m/s)</th>
<th>$p_D$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.001</td>
<td>0.047</td>
<td>0.048</td>
<td>0.001</td>
<td>1.32</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>0.095</td>
<td>0.096</td>
<td>0.001</td>
<td>1.33</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>0.034</td>
<td>0.036</td>
<td>0.002</td>
<td>4.91</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>0.069</td>
<td>0.072</td>
<td>0.003</td>
<td>4.72</td>
</tr>
<tr>
<td>1</td>
<td>0.01</td>
<td>0.047</td>
<td>0.051</td>
<td>0.004</td>
<td>7.55</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>0.095</td>
<td>0.102</td>
<td>0.007</td>
<td>7.46</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>0.034</td>
<td>0.042</td>
<td>0.007</td>
<td>18.93</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>0.069</td>
<td>0.083</td>
<td>0.014</td>
<td>18.66</td>
</tr>
</tbody>
</table>

As shown by these results, the effects of eccentricity and final transfer times are the primary factors that contribute to large $\Delta v$ differences. The $\Delta v$ difference is still relatively low due to the short duration transfer trajectories, and their close proximity to the chief.

14.6 Conclusions

For Scenario 2, similar conclusions may be drawn for the comparison between implementing the reference trajectory in Cartesian versus spherical coordinates. The results shown in Table 14.2 show that there is clearly greater accuracy when using the spherical model for cases with large along-track separation, and absence of (J2/drag/SRP) perturbations. The controller tracked the optimal trajectory more closely via the spherical representations of the LROEs. Modeling errors associated with the Cartesian LVLH frame are recaptured in many cases, although once the perturbations are included in the simulation, these effects are not significant, as seen in Table 14.3.

From the $\Delta v$ results in Table 14.4, it is evident that the ellipticity of the chief plays a major role in Case 1 (as in Scenario 1), when considering values greater than $e_c = 0.001$. The large increase in required $\Delta v$ for trajectory following, from $e_c = 0.001$ to $e_c = 0.01$, signifies this conclusion. In these cases, and for more elliptic target orbits, the orbital dynamics model for the LROEs in the planner must be improved to account for these effects.

The results in Table 14.7 show that in Case 2, the effects of eccentricity still have a
significant effect. This can be seen by the percent difference error between the planned and simulated trajectories. Similar to Scenario 1, the primary variables that drive the effects of \( \Delta v \) percent difference error (from the results of Scenario 2, Case 1 and 2) are the chief/deputy separation, and the final transfer time. While tracking a trajectory far away from the chief (with relatively large final times), the benefits of adding eccentricity/perturbations into the dynamics model may help to improve the overall performance. The errors due to the effects of eccentricity/perturbations may be countered by incorporating improved control laws, or dwarfed by navigation and maneuver execution errors in the simulation. It may turn out that with all of these error sources included in the simulation, the effects of \( J_2/drag/SRP \) and low-level eccentricity may be a minimal source of error in the overall planner model.

Scenario 1 Case 3 (safe final approach) nonlinear simulation results show a similar trend to those in Cases 2 and 3 of Scenario 1. It can be seen by the \( \Delta v \) results in Table 14.10 that the effects of eccentricity on these final approach trajectories is minimal. This is again due to the fact that in these examples the deputy is in relatively close proximity of the chief, and the final transfer times are small in comparison to safe initial approach and safe traveling ellipse results. Also interesting to note in this case is the fact that the \( \Delta v \) from the planner for the 200×100 m safety ellipse to the 100×50 m safety ellipse is double that from the 100×50 m safety ellipse to 50×25 m. This is one effect of doubling the semi-major axis and cross-track amplitude, while the other parameters (initial/final center of motion, final time) are held constant. The same situation occurred in the Case 1 results.
CHAPTER 15

PERFORMANCE ANALYSIS III, MONTE CARLO DISPERSION ANALYSIS -
SCENARIOS 1 & 2

The objective of this chapter is to present and analyze the planner and simulation results in a Monte Carlo dispersion analysis. The dispersions that are used in each case are defined, and trajectories, control histories, and Δv tables are included. This dispersion analysis will show the effects of adding dispersions to the initial relative position and velocity, and how the resulting optimal trajectories and optimal control histories change. Examples of Scenarios 1 and 2 (defined in Sections 15.2 and 15.4, respectively) are analyzed.

15.1 Dispersions Analysis

A dispersion analysis is performed in this chapter, whereby the initial position and velocity are perturbed from the nominal. Resulting trajectories and optimal control histories are plotted for comparison, for multiple orbital transfers. The dispersion for the initial conditions from the nominal trajectory are added into the trajectory planner and the simulation, at the initial time \( t_0 \). This includes dispersions in the initial relative position and velocity. These values are normally distributed, and are specified using a 1-σ standard deviation for both position and velocity.

In Scenario 1, the nominal initial relative position is specified, and the nominal initial relative velocity for each trajectory is zero. Therefore, dispersions are defined by 1-σ position dispersions and 1-σ velocity dispersions, where the 1-σ radial, along-track, and cross-track component of the dispersions are the same (thus, the dispersion ellipses for both position and velocity are simple spheres). The initial 1-σ dispersion for position is simply a fixed percentage of the initial nominal relative position magnitude, and the initial velocity dispersion is the same percentage of the initial position dispersion, times the mean orbital rate, \( \omega \). This is a good approximation of the expected velocity dispersions for near-circular orbits.
In Scenario 2, the radial, along-track, and cross-track dispersions are defined separately. For example, in the safe initial approach case (Scenario 2, Case 1), the initial trajectory is a co-elliptic flyby orbit. For this type of orbit, the true relative ellipse is very sensitive to the radial component of the position, for a given flyby velocity. Therefore, in this case the radial dispersions must be much smaller, to generate trajectories with similar control histories. If the dispersions are not specified in this manner, then the trajectories can generally be very different, with a great variety of control histories. This leads to defining the dispersions in Scenario 2 in terms of LROEs. For example, the initial radial position dispersion for the flyby orbit is now a fixed percentage of the nominal initial radial center of motion, $x_r$. Similarly, the along-track dispersion is defined as a fixed percentage of the initial along-track center of motion, $y_r$. All dispersions on velocity are calculated in a similar manner, where the standard deviations in velocity are dependent on the initial nominal velocity in each case.

15.2 Scenario 1 Dispersions

The cases in Scenario 1 are now analyzed including the effects of dispersions on the initial state. The equations for the true relative position and velocity at $t_0$ are

$$\mathbf{x}(t_0) = \begin{bmatrix} \mathbf{r}_{\text{ref}}(t_0) \\ \mathbf{v}_{\text{rel}}(t_0) \end{bmatrix} = \begin{bmatrix} \mathbf{r}_{\text{ref}}(t_0) \\ \mathbf{v}_{\text{ref}}(t_0) \end{bmatrix} + \begin{bmatrix} \mathbf{e}_r \\ \mathbf{e}_v \end{bmatrix}$$  \hspace{1cm} (15.1)

The vectors $\mathbf{e}_r$ and $\mathbf{e}_v$ are zero-mean, normally distributed, and are specified by a 1-$\sigma$ factor for the standard deviation, $\sigma_r$.

$$\mathbf{e}_r \sim \mathcal{N}(0, \sigma_r^2 I)$$

$$\mathbf{e}_v \sim \mathcal{N}(0, \omega^2 \sigma_r^2 I)$$

where it is seen that the standard deviation for velocity is simply the standard deviation of
position, multiplied by the mean orbital rate, $\omega$. The standard deviation for the position, $\sigma_r$, in these cases is defined by taking a percentage, $p_r$, of the initial relative position magnitude, as

$$\sigma_r = \frac{p_r}{100\%} ||r_{rel}(t_0)||$$

(15.2)

The velocity dispersions are then defined using $\sigma_v = \omega \sigma_r$.

### 15.3 Scenario 1 Planner and Simulation Results with Dispersions

#### 15.3.1 Case 1 - Initial Approach with Circular Chief Orbit and Perturbations

The examples analyzed in Scenario 1, Case 1, Initial Approach, include Example 1 and Example 4, as defined in Table 13.3. Example 1 is an initial approach from 5 km on the $v$-bar to a 200 m station-keeping point with final time of 1.2 chief revolutions. Dispersions with relative position percentage errors of $p_r = 5\%$ and $1\%$ are examined. These correspond to standard deviation of $\sigma_r = 250$ m, and $50$ m on relative position and $\sigma_v = 0.25$ m/s and $0.05$ m/s on relative velocity, respectively. The trajectories with $p_r = 5\%$ dispersions are shown in Fig. 15.1, the corresponding control histories are in Fig. 15.2, and $\Delta v$ in Table 15.1.

![Fig. 15.1: Scenario 1, Case 1, Example 1 trajectories with $p_r = 5\%$ dispersions (spherical model, with perturbations)
Fig. 15.2: Scenario 1, Case 1, Example 1 control history with \( p_r = 5\% \) dispersions (spherical model, circular chief, with perturbations)

Table 15.1: Monte Carlo \( \Delta v \) for Scenario 1, Case 1, Example 4 with \( p_r = 5\% \) dispersions

<table>
<thead>
<tr>
<th>MC Run</th>
<th>( \Delta v_{\text{opt}} ) (m/s)</th>
<th>( \Delta v_{\text{sim}} ) (m/s)</th>
<th>Diff. (m/s)</th>
<th>( p_D ) (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.193</td>
<td>1.239</td>
<td>0.046</td>
<td>3.75</td>
</tr>
<tr>
<td>2</td>
<td>0.950</td>
<td>0.998</td>
<td>0.049</td>
<td>4.99</td>
</tr>
<tr>
<td>3</td>
<td>0.768</td>
<td>0.820</td>
<td>0.052</td>
<td>6.50</td>
</tr>
<tr>
<td>4</td>
<td>0.751</td>
<td>0.802</td>
<td>0.050</td>
<td>6.50</td>
</tr>
<tr>
<td>5</td>
<td>1.032</td>
<td>1.084</td>
<td>0.052</td>
<td>4.89</td>
</tr>
<tr>
<td>6</td>
<td>0.451</td>
<td>0.503</td>
<td>0.052</td>
<td>10.98</td>
</tr>
<tr>
<td>7</td>
<td>0.587</td>
<td>0.630</td>
<td>0.043</td>
<td>7.10</td>
</tr>
<tr>
<td>8</td>
<td>1.449</td>
<td>1.500</td>
<td>0.051</td>
<td>3.47</td>
</tr>
<tr>
<td>9</td>
<td>0.680</td>
<td>0.727</td>
<td>0.047</td>
<td>6.65</td>
</tr>
<tr>
<td>10</td>
<td>0.778</td>
<td>0.833</td>
<td>0.055</td>
<td>6.78</td>
</tr>
</tbody>
</table>

The trajectories for Example 1 with \( p_r = 1\% \) dispersions are shown in Fig. 15.3, the corresponding control histories are in Fig. 15.4, and \( \Delta v \) in Table 15.2.
Fig. 15.3: Scenario 1, Case 1, Example 1 trajectories with \( p_r = 1\% \) dispersions (spherical model, with perturbations)

Fig. 15.4: Scenario 1, Case 1, Example 1 control history with \( p_r = 1\% \) dispersions (spherical model, circular chief, with perturbations)
Table 15.2: Monte Carlo $\Delta v$ for Scenario 1, Case 1, Example 1 with $p_r = 1\%$ dispersions

<table>
<thead>
<tr>
<th>MC Run</th>
<th>$\Delta v_{opt}$ (m/s)</th>
<th>$\Delta v_{sim}$ (m/s)</th>
<th>Diff. (m/s)</th>
<th>$p_D$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.301</td>
<td>0.350</td>
<td>0.049</td>
<td>15.01</td>
</tr>
<tr>
<td>2</td>
<td>0.377</td>
<td>0.426</td>
<td>0.048</td>
<td>12.03</td>
</tr>
<tr>
<td>3</td>
<td>0.513</td>
<td>0.557</td>
<td>0.044</td>
<td>8.28</td>
</tr>
<tr>
<td>4</td>
<td>0.473</td>
<td>0.519</td>
<td>0.046</td>
<td>9.22</td>
</tr>
<tr>
<td>5</td>
<td>0.594</td>
<td>0.644</td>
<td>0.049</td>
<td>7.97</td>
</tr>
<tr>
<td>6</td>
<td>0.496</td>
<td>0.543</td>
<td>0.047</td>
<td>9.06</td>
</tr>
<tr>
<td>7</td>
<td>0.536</td>
<td>0.581</td>
<td>0.045</td>
<td>8.11</td>
</tr>
<tr>
<td>8</td>
<td>0.690</td>
<td>0.735</td>
<td>0.045</td>
<td>6.33</td>
</tr>
<tr>
<td>9</td>
<td>0.424</td>
<td>0.470</td>
<td>0.046</td>
<td>10.27</td>
</tr>
<tr>
<td>10</td>
<td>0.507</td>
<td>0.555</td>
<td>0.047</td>
<td>8.94</td>
</tr>
</tbody>
</table>

It can be seen in the Example 1 results for $p_r = 5\%$, in Table 15.1, that the planner $\Delta v$ varies greatly (from 0.4 m/s to 1.4 m/s), due to the initial position and velocity dispersions. A diverse set of trajectories are planned, as seen in Fig. 15.1, and the control histories in Fig. 15.2 show great variation. However, when the dispersions are decreased to $p_r = 1\%$, the trajectories and controls are much more tightly grouped, as seen in Figs. 15.3 and 15.4. This corresponds to a 1-σ of 50 m on relative position and 0.05 m/s on relative velocity. The resulting $\Delta v$ in Table 15.2 only varies from 0.3 m/s to 0.6 m/s.

The next results are for Example 4 defined in Table 13.3. This is an initial approach from 10 km on the v-bar to a 200 m station-keeping point with final time of 2.4 chief revolutions. Dispersions with relative position percentage errors of $p_r = 5\%, 1\%$, and 0.5% are examined. These correspond to standard deviation of $\sigma_r = 500$ m, 100 m, and 50 m on relative position and $\sigma_v = 0.5$ m/s, 0.1 m/s, and 0.05 m/s on relative velocity, respectively. The trajectories with $p_r = 5\%$ dispersions are shown in Fig. 15.5, the corresponding control histories are in Fig. 15.6, and $\Delta v$ in Table 15.3.
Fig. 15.5: Scenario 1, Case 1, Example 4 trajectories with $p_r = 5\%$ dispersions (spherical model, with perturbations)

Fig. 15.6: Scenario 1, Case 1, Example 4 control history with $p_r = 5\%$ dispersions (spherical model, circular chief, with perturbations)
Table 15.3: Monte Carlo $\Delta v$ for Scenario 1, Case 1, Example 4 with $p_r = 5\%$ dispersions

<table>
<thead>
<tr>
<th>MC Run</th>
<th>$\Delta v_{opt}$ (m/s)</th>
<th>$\Delta v_{sim}$ (m/s)</th>
<th>Diff. (m/s)</th>
<th>$p_D$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.340</td>
<td>2.484</td>
<td>0.144</td>
<td>5.95</td>
</tr>
<tr>
<td>2</td>
<td>1.028</td>
<td>1.167</td>
<td>0.139</td>
<td>12.69</td>
</tr>
<tr>
<td>3</td>
<td>0.863</td>
<td>0.981</td>
<td>0.119</td>
<td>12.90</td>
</tr>
<tr>
<td>4</td>
<td>1.675</td>
<td>1.765</td>
<td>0.091</td>
<td>5.27</td>
</tr>
<tr>
<td>5</td>
<td>2.306</td>
<td>2.420</td>
<td>0.113</td>
<td>4.80</td>
</tr>
<tr>
<td>6</td>
<td>1.312</td>
<td>1.431</td>
<td>0.119</td>
<td>8.66</td>
</tr>
<tr>
<td>7</td>
<td>1.318</td>
<td>1.457</td>
<td>0.139</td>
<td>10.01</td>
</tr>
<tr>
<td>8</td>
<td>1.200</td>
<td>1.364</td>
<td>0.164</td>
<td>12.80</td>
</tr>
<tr>
<td>9</td>
<td>2.646</td>
<td>2.740</td>
<td>0.095</td>
<td>3.52</td>
</tr>
<tr>
<td>10</td>
<td>1.264</td>
<td>1.345</td>
<td>0.081</td>
<td>6.24</td>
</tr>
</tbody>
</table>

The Example 4 trajectories with $p_r = 1\%$ dispersions are shown in Fig. 15.7, the corresponding control histories are in Fig. 15.8, and $\Delta v$ in Table 15.4.

Fig. 15.7: Scenario 1, Case 1, Example 4 trajectories with $p_r = 1\%$ dispersions (spherical model, with perturbations)
Fig. 15.8: Scenario 1, Case 1, Example 4 control history with \( p_r = 1\% \) dispersions (spherical model, circular chief, with perturbations)

Table 15.4: Monte Carlo \( \Delta v \) for Scenario 1, Case 1, Example 4 with \( p_r = 1\% \) dispersions

<table>
<thead>
<tr>
<th>MC Run</th>
<th>( \Delta v_{opt} ) (m/s)</th>
<th>( \Delta v_{sim} ) (m/s)</th>
<th>Diff. (m/s)</th>
<th>( p_D ) (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.783</td>
<td>0.906</td>
<td>0.123</td>
<td>14.61</td>
</tr>
<tr>
<td>2</td>
<td>0.791</td>
<td>0.922</td>
<td>0.131</td>
<td>15.27</td>
</tr>
<tr>
<td>3</td>
<td>0.722</td>
<td>0.851</td>
<td>0.129</td>
<td>16.37</td>
</tr>
<tr>
<td>4</td>
<td>0.989</td>
<td>1.115</td>
<td>0.126</td>
<td>11.96</td>
</tr>
<tr>
<td>5</td>
<td>0.576</td>
<td>0.704</td>
<td>0.128</td>
<td>20.03</td>
</tr>
<tr>
<td>6</td>
<td>0.403</td>
<td>0.534</td>
<td>0.131</td>
<td>28.00</td>
</tr>
<tr>
<td>7</td>
<td>0.750</td>
<td>0.879</td>
<td>0.130</td>
<td>15.91</td>
</tr>
<tr>
<td>8</td>
<td>0.457</td>
<td>0.585</td>
<td>0.128</td>
<td>24.61</td>
</tr>
<tr>
<td>9</td>
<td>1.041</td>
<td>1.173</td>
<td>0.131</td>
<td>11.86</td>
</tr>
<tr>
<td>10</td>
<td>0.458</td>
<td>0.572</td>
<td>0.114</td>
<td>22.19</td>
</tr>
</tbody>
</table>

The Example 4 trajectories with \( p_r = 0.5\% \) dispersions are shown in Fig. 15.9, and the corresponding control histories are in Fig. 15.10, and \( \Delta v \) in Table 15.5.
Fig. 15.9: Scenario 1, Case 1, Example 4 trajectories with $p_r = 0.5\%$ dispersions (spherical model, with perturbations)

Fig. 15.10: Scenario 1, Case 1, Example 4 control history with $p_r = 0.5\%$ dispersions (spherical model, circular chief, with perturbations)
Table 15.5: Monte Carlo $\Delta v$ for Scenario 1, Case 1, Example 4 with $p_r = 0.5\%$ dispersions

<table>
<thead>
<tr>
<th>MC Run</th>
<th>$\Delta v_{opt}$ (m/s)</th>
<th>$\Delta v_{sim}$ (m/s)</th>
<th>Diff. (m/s)</th>
<th>$p_D$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.374</td>
<td>0.505</td>
<td>0.131</td>
<td>29.89</td>
</tr>
<tr>
<td>2</td>
<td>0.293</td>
<td>0.423</td>
<td>0.130</td>
<td>36.37</td>
</tr>
<tr>
<td>3</td>
<td>0.594</td>
<td>0.720</td>
<td>0.126</td>
<td>19.23</td>
</tr>
<tr>
<td>4</td>
<td>0.609</td>
<td>0.737</td>
<td>0.128</td>
<td>19.02</td>
</tr>
<tr>
<td>5</td>
<td>0.588</td>
<td>0.717</td>
<td>0.128</td>
<td>19.65</td>
</tr>
<tr>
<td>6</td>
<td>0.638</td>
<td>0.768</td>
<td>0.130</td>
<td>18.46</td>
</tr>
<tr>
<td>7</td>
<td>0.622</td>
<td>0.751</td>
<td>0.130</td>
<td>18.87</td>
</tr>
<tr>
<td>8</td>
<td>0.407</td>
<td>0.537</td>
<td>0.130</td>
<td>27.56</td>
</tr>
<tr>
<td>9</td>
<td>0.634</td>
<td>0.760</td>
<td>0.126</td>
<td>18.11</td>
</tr>
<tr>
<td>10</td>
<td>0.696</td>
<td>0.827</td>
<td>0.131</td>
<td>17.15</td>
</tr>
</tbody>
</table>

From the Example 4 dispersion analysis, it is evident in Figs. 15.9-15.10 and Table 15.5 that at larger ranges of 10 km, trajectory variability is small when $p_r = 0.5\%$ (corresponding to a 1-$\sigma$ of 50 m on relative position and 0.05 m/s on relative velocity). Larger percent errors led to more diverse trajectories and control histories, as shown in Figs. 15.5-15.8. The planner $\Delta v$ is also highly diverse, shown in Table 15.3 and 15.4. The result for Scenario 1, Case 1, is that even for a larger initial range of 10 km and larger final time of 2.4 chief revolutions, the effects of dispersions in Example 4 are similar to the results from Example 1, where smaller initial dispersions produce less trajectory variability.

15.3.2 Case 2 - Way-point Following with Circular Chief Orbit and Perturbations

The Monte Carlo example for Scenario 1, Case 2, Way-point Following, is Example 4, as defined in Table 13.7. In this example, the deputy goes from 100 m on the y-bar to 100 m cross-track with a final time of 0.6 chief revolutions. Dispersions with relative position percentage errors of $p_r = 10\%$, and $5\%$ are examined. These correspond to standard a deviation of $\sigma_r = 10$ m, and 5 m on relative position and $\sigma_v = 0.01$ m/s, and 0.005 m/s on relative velocity, respectively. The trajectories with $p_r = 10\%$ dispersions are shown in Fig.
the corresponding control histories are in Fig. 15.6, and $\Delta v$ in Table 15.6.

Fig. 15.11: Scenario 1, Case 2, Example 4 trajectories with $p_r = 10\%$ dispersions (Cartesian model, with perturbations)

Fig. 15.12: Scenario 1, Case 2, Example 4 control history with $p_r = 10\%$ dispersions (Cartesian model, circular chief, with perturbations)
Table 15.6: Monte Carlo $\Delta v$ for Scenario 1, Case 2, Example 4 with $p_r = 10\%$ dispersions

<table>
<thead>
<tr>
<th>MC Run</th>
<th>$\Delta v_{opt}$ (m/s)</th>
<th>$\Delta v_{sim}$ (m/s)</th>
<th>Diff. (m/s)</th>
<th>$p_D$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.133</td>
<td>0.135</td>
<td>0.002</td>
<td>1.40</td>
</tr>
<tr>
<td>2</td>
<td>0.147</td>
<td>0.149</td>
<td>0.002</td>
<td>1.30</td>
</tr>
<tr>
<td>3</td>
<td>0.144</td>
<td>0.145</td>
<td>0.002</td>
<td>1.28</td>
</tr>
<tr>
<td>4</td>
<td>0.162</td>
<td>0.164</td>
<td>0.002</td>
<td>1.14</td>
</tr>
<tr>
<td>5</td>
<td>0.144</td>
<td>0.146</td>
<td>0.002</td>
<td>1.18</td>
</tr>
<tr>
<td>6</td>
<td>0.133</td>
<td>0.134</td>
<td>0.001</td>
<td>1.12</td>
</tr>
<tr>
<td>7</td>
<td>0.153</td>
<td>0.155</td>
<td>0.002</td>
<td>1.09</td>
</tr>
<tr>
<td>8</td>
<td>0.110</td>
<td>0.112</td>
<td>0.002</td>
<td>1.58</td>
</tr>
<tr>
<td>9</td>
<td>0.138</td>
<td>0.140</td>
<td>0.002</td>
<td>1.23</td>
</tr>
<tr>
<td>10</td>
<td>0.157</td>
<td>0.159</td>
<td>0.002</td>
<td>1.01</td>
</tr>
</tbody>
</table>

The Example 4 trajectories with $p_r = 5\%$ dispersions are shown in Fig. 15.13, and the corresponding control histories are in Fig. 15.14, and $\Delta v$ in Table 15.7.

![Fig. 15.13: Scenario 1, Case 2, Example 4 trajectories with $p_r = 5\%$ dispersions (Cartesian model, with perturbations)](image-url)
Table 15.7: Monte Carlo $\Delta v$ for Scenario 1, Case 2, Example 4 with $p_r = 5\%$ dispersions

<table>
<thead>
<tr>
<th>MC Run</th>
<th>$\Delta v_{opt}$ (m/s)</th>
<th>$\Delta v_{sim}$ (m/s)</th>
<th>Diff. (m/s)</th>
<th>$p_D$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.144</td>
<td>0.146</td>
<td>0.002</td>
<td>1.18</td>
</tr>
<tr>
<td>2</td>
<td>0.128</td>
<td>0.129</td>
<td>0.002</td>
<td>1.31</td>
</tr>
<tr>
<td>3</td>
<td>0.155</td>
<td>0.156</td>
<td>0.002</td>
<td>1.05</td>
</tr>
<tr>
<td>4</td>
<td>0.154</td>
<td>0.156</td>
<td>0.002</td>
<td>1.07</td>
</tr>
<tr>
<td>5</td>
<td>0.128</td>
<td>0.130</td>
<td>0.002</td>
<td>1.31</td>
</tr>
<tr>
<td>6</td>
<td>0.131</td>
<td>0.133</td>
<td>0.002</td>
<td>1.28</td>
</tr>
<tr>
<td>7</td>
<td>0.134</td>
<td>0.136</td>
<td>0.002</td>
<td>1.29</td>
</tr>
<tr>
<td>8</td>
<td>0.135</td>
<td>0.136</td>
<td>0.002</td>
<td>1.25</td>
</tr>
<tr>
<td>9</td>
<td>0.143</td>
<td>0.145</td>
<td>0.002</td>
<td>1.16</td>
</tr>
<tr>
<td>10</td>
<td>0.128</td>
<td>0.130</td>
<td>0.002</td>
<td>1.34</td>
</tr>
</tbody>
</table>

In the Scenario 1, Case 2, dispersion analysis, the example included here shows that though the planned trajectories have relatively large percent errors ($p_r = 5\%, 10\%$), the optimal trajectories and controls have consistent characteristics. This can be seen by the ‘elbow’ in the trajectory shown in Fig. 15.13, and the grouped three-burn optimal control solutions shown in Fig. 15.14. It is evident that at close ranges, the optimal trajectory planner is less affected by larger percent errors.
15.3.3 Case 3 - Final Approach with Circular Chief Orbit and Perturbations

The Monte Carlo example for Scenario 1, Case 3, Final Approach, is Example 3 defined in Table 13.10. In this example, the deputy is approaching from the v-bar at 100 m away to 20 m v-bar station-keeping with a final time of 0.6 chief revolutions. Dispersions with relative position percentage errors of $p_r = 10\%$, and $5\%$ are examined. These correspond to standard deviation of $\sigma_r = 10$ m, and 5 m on relative position and $\sigma_v = 0.01$ m/s, and 0.005 m/s on relative velocity, respectively. The trajectories with $p_r = 10\%$ dispersions are shown in Fig. 15.15, the corresponding control histories are in Fig. 15.16, and $\Delta v$ in Table 15.8.

![Fig. 15.15: Scenario 1, Case 3, Example 3 trajectories with $p_r = 10\%$ dispersions (Cartesian model, with perturbations)](image-url)
Fig. 15.16: Scenario 1, Case 3, Example 3 control history with $p_r = 10\%$ dispersions (Cartesian model, circular chief, with perturbations)

Table 15.8: Monte Carlo $\Delta v$ for Scenario 1, Case 3, Example 3 with $p_r = 10\%$ dispersions

<table>
<thead>
<tr>
<th>MC Run</th>
<th>$\Delta v_{opt}$ (m/s)</th>
<th>$\Delta v_{sim}$ (m/s)</th>
<th>Diff. (m/s)</th>
<th>$p_D$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.073</td>
<td>0.074</td>
<td>0.001</td>
<td>0.84</td>
</tr>
<tr>
<td>2</td>
<td>0.069</td>
<td>0.069</td>
<td>0.001</td>
<td>0.86</td>
</tr>
<tr>
<td>3</td>
<td>0.068</td>
<td>0.068</td>
<td>0.000</td>
<td>0.58</td>
</tr>
<tr>
<td>4</td>
<td>0.046</td>
<td>0.047</td>
<td>0.001</td>
<td>1.60</td>
</tr>
<tr>
<td>5</td>
<td>0.075</td>
<td>0.076</td>
<td>0.001</td>
<td>0.86</td>
</tr>
<tr>
<td>6</td>
<td>0.082</td>
<td>0.083</td>
<td>0.001</td>
<td>0.75</td>
</tr>
<tr>
<td>7</td>
<td>0.057</td>
<td>0.058</td>
<td>0.000</td>
<td>0.75</td>
</tr>
<tr>
<td>8</td>
<td>0.062</td>
<td>0.063</td>
<td>0.000</td>
<td>0.56</td>
</tr>
<tr>
<td>9</td>
<td>0.066</td>
<td>0.066</td>
<td>0.001</td>
<td>0.78</td>
</tr>
<tr>
<td>10</td>
<td>0.059</td>
<td>0.060</td>
<td>0.001</td>
<td>0.91</td>
</tr>
</tbody>
</table>

The Example 3 trajectories with $p_r = 5\%$ dispersions are shown in Fig. 15.17, and the corresponding control histories are in Fig. 15.18, and $\Delta v$ in Table 15.9.
Fig. 15.17: Scenario 1, Case 3, Example 3 trajectories with $p_r = 5\%$ dispersions (Cartesian model, with perturbations)

Fig. 15.18: Scenario 1, Case 3, Example 3 control history with $p_r = 5\%$ dispersions (Cartesian model, circular chief, with perturbations)
Table 15.9: Monte Carlo $\Delta v$ for Scenario 1, Case 3, Example 3 with $p_r = 5\%$ dispersions

<table>
<thead>
<tr>
<th>MC Run</th>
<th>$\Delta v_{opt}$ (m/s)</th>
<th>$\Delta v_{sim}$ (m/s)</th>
<th>Diff. (m/s)</th>
<th>$p_D$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.077</td>
<td>0.077</td>
<td>0.001</td>
<td>0.74</td>
</tr>
<tr>
<td>2</td>
<td>0.068</td>
<td>0.068</td>
<td>0.001</td>
<td>0.86</td>
</tr>
<tr>
<td>3</td>
<td>0.070</td>
<td>0.070</td>
<td>0.001</td>
<td>0.89</td>
</tr>
<tr>
<td>4</td>
<td>0.066</td>
<td>0.067</td>
<td>0.001</td>
<td>0.98</td>
</tr>
<tr>
<td>5</td>
<td>0.058</td>
<td>0.059</td>
<td>0.001</td>
<td>1.08</td>
</tr>
<tr>
<td>6</td>
<td>0.065</td>
<td>0.066</td>
<td>0.001</td>
<td>0.94</td>
</tr>
<tr>
<td>7</td>
<td>0.084</td>
<td>0.084</td>
<td>0.001</td>
<td>0.62</td>
</tr>
<tr>
<td>8</td>
<td>0.058</td>
<td>0.059</td>
<td>0.000</td>
<td>0.85</td>
</tr>
<tr>
<td>9</td>
<td>0.078</td>
<td>0.079</td>
<td>0.001</td>
<td>0.79</td>
</tr>
<tr>
<td>10</td>
<td>0.065</td>
<td>0.066</td>
<td>0.001</td>
<td>1.02</td>
</tr>
</tbody>
</table>

The Scenario 1, Case 3, results are similar to that of Scenario 1, Case 2. Since the range is relatively small, larger percent errors do not greatly affect the planner in terms of trajectories or control histories. These trends are seen in Fig. 15.17-15.18, and Table 15.9.

### 15.4 Scenario 2 Dispersions

The cases in Scenario 2 are now analyzed including the effects of dispersions on the initial state. The equations for the true LROEs at $t_0$ are

$$\mathbf{x}(t_0) = \begin{bmatrix} x_r(t_0) \\ y_r(t_0) \\ a_r \sin(E_r)(t_0) \\ a_r \cos(E_r)(t_0) \\ A_z \sin(\psi_z)(t_0) \\ A_z \cos(\psi_z)(t_0) \end{bmatrix} = \begin{bmatrix} x_{r,ref}(t_0) \\ y_{r,ref}(t_0) \\ [a_r \sin(E_r)]_{ref}(t_0) \\ [a_r \cos(E_r)]_{ref}(t_0) \\ [A_z \sin(\psi_z)]_{ref}(t_0) \\ [A_z \cos(\psi_z)]_{ref}(t_0) \end{bmatrix} + T \begin{bmatrix} e_r \\ e_v \end{bmatrix} \tag{15.3}$$

where $T$ is given in Eq. 5.41.
From preliminary results for the Scenario 2 analysis, the dependence on the dispersions are recognized to be very different for each of the radial, along-track, and cross-track components. To fully analyze the effects of dispersions in this scenario, they are separated into each component. Therefore, the values for vectors $e_r$ and $e_v$ are zero-mean, normally distributed, and are specified using $1$-$\sigma$ factors for radial, along-track, and cross-track, as $\sigma_x$, $\sigma_y$, and $\sigma_z$, respectively.

$$e_r \sim \mathcal{N}(0, \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{bmatrix})$$

$$e_v \sim \mathcal{N}(0, \begin{bmatrix} \sigma_{v_x}^2 & 0 & 0 \\ 0 & \sigma_{v_y}^2 & 0 \\ 0 & 0 & \sigma_{v_z}^2 \end{bmatrix})$$

where each component of the standard deviation for velocity is defined as a function of the nominal velocity. The standard deviations for both the position and velocity are defined in terms of the nominal position and velocity, and the ROEs, by taking a percentage of these values for each safety ellipse, $p_{SE}$, as

$$\sigma_x = \frac{p_{SE}}{100\%} \left| x_r(t_0) - \frac{1}{2}a_r(t_0) \right|$$

$$\sigma_y = \frac{p_{SE}}{100\%} \left| y_r(t_0) + a_r(t_0) \right|$$

$$\sigma_z = \frac{p_{SE}}{100\%} A_z(t_0)$$

$$\sigma_{v_x} = \frac{p_{SE}}{100\%} \frac{\omega}{2} a_r(t_0)$$

$$\sigma_{v_y} = \frac{p_{SE}}{100\%} \omega \left| a_r(t_0) - \frac{3}{2}x_r(t_0) \right|$$

$$\sigma_{v_z} = \frac{p_{SE}}{100\%} \omega A_z(t_0)$$

These standard deviations are defined by the linear transformation from LROEs to
relative position and velocity, \( x = T^{-1} \tilde{x} \), which results in functions of \( \sin(E_r), \cos(E_r) \) and \( \sin(\psi_z), \cos(\psi_z) \). The standard deviations on the safety ellipse size and position in Eqs. 15.4-15.9 are defined by setting all \( \sin/\cos \) functions to 1, such that \( \sin(E_r) = \cos(E_r) = \sin(\psi_z) = \cos(\psi_z) = 1 \). Therefore, the size of the initial relative ellipse (semi-major axis, \( a_r \), and cross-track amplitude, \( A_z \)) ultimately define the deviations in relative position and velocity for any given initial phase angles.

15.5 Scenario 2 Planner and Simulation Results with Dispersions

15.5.1 Case 1 - Safe Initial Approach with Circular Chief Orbit and Perturbations

The examples analyzed in Scenario 2, Case 1, Safe Initial Approach, include Example 1 and Example 4, as defined in Table 14.1. Example 1 is a safe initial approach from -2.5 km v-bar -300 m radial to a 100×50 m safety ellipse at 500 m on the v-bar, with final time of 1.2 chief revolutions. Dispersions with safety ellipse percentage errors of \( p_{SE} = 10\% \) and 5\% are examined. The trajectories with \( p_{SE} = 10\% \) dispersions are shown in Fig. 15.19, the corresponding control histories are in Fig. 15.20, and \( \Delta v \) in Table 15.10.

![Fig. 15.19: Scenario 2, Case 1, Example 1 trajectories with \( p_{SE} = 10\% \) dispersions (spherical model, with perturbations)](image-url)
Fig. 15.20: Scenario 2, Case 1, Example 1 control history with $p_{SE} = 10\%$ dispersions (spherical model, circular chief, with perturbations)

Table 15.10: Monte Carlo $\Delta v$ for Scenario 2, Case 1, Example 1 with $p_{SE} = 10\%$ dispersions

<table>
<thead>
<tr>
<th>MC Run</th>
<th>$\Delta v_{opt}$ (m/s)</th>
<th>$\Delta v_{sim}$ (m/s)</th>
<th>Diff. (m/s)</th>
<th>$p_D$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.348</td>
<td>0.376</td>
<td>0.028</td>
<td>7.74</td>
</tr>
<tr>
<td>2</td>
<td>0.245</td>
<td>0.269</td>
<td>0.023</td>
<td>9.05</td>
</tr>
<tr>
<td>3</td>
<td>0.256</td>
<td>0.285</td>
<td>0.029</td>
<td>10.77</td>
</tr>
<tr>
<td>4</td>
<td>0.270</td>
<td>0.298</td>
<td>0.028</td>
<td>10.01</td>
</tr>
<tr>
<td>5</td>
<td>0.190</td>
<td>0.215</td>
<td>0.026</td>
<td>12.61</td>
</tr>
<tr>
<td>6</td>
<td>0.213</td>
<td>0.237</td>
<td>0.024</td>
<td>10.81</td>
</tr>
<tr>
<td>7</td>
<td>0.235</td>
<td>0.264</td>
<td>0.029</td>
<td>11.53</td>
</tr>
<tr>
<td>8</td>
<td>0.214</td>
<td>0.237</td>
<td>0.024</td>
<td>10.52</td>
</tr>
<tr>
<td>9</td>
<td>0.337</td>
<td>0.366</td>
<td>0.029</td>
<td>8.28</td>
</tr>
<tr>
<td>10</td>
<td>0.204</td>
<td>0.228</td>
<td>0.025</td>
<td>11.36</td>
</tr>
</tbody>
</table>

The Example 1 trajectories with $p_{SE} = 5\%$ dispersions are shown in Fig. 15.21, and the corresponding control histories are in Fig. 15.22, and $\Delta v$ in Table 15.11.
Fig. 15.21: Scenario 2, Case 1, Example 1 trajectories with $p_{SE} = 5\%$ dispersions (spherical model, with perturbations)

Fig. 15.22: Scenario 2, Case 1, Example 1 control history with $p_{SE} = 5\%$ dispersions (spherical model, circular chief, with perturbations)
Table 15.11: Monte Carlo $\Delta v$ for Scenario 2, Case 1, Example 1 with $p_{SE} = 5\%$ dispersions

<table>
<thead>
<tr>
<th>MC Run</th>
<th>$\Delta v_{opt}$ (m/s)</th>
<th>$\Delta v_{sim}$ (m/s)</th>
<th>Diff. (m/s)</th>
<th>$p_D$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.232</td>
<td>0.260</td>
<td>0.028</td>
<td>11.49</td>
</tr>
<tr>
<td>2</td>
<td>0.214</td>
<td>0.243</td>
<td>0.029</td>
<td>12.62</td>
</tr>
<tr>
<td>3</td>
<td>0.207</td>
<td>0.233</td>
<td>0.026</td>
<td>11.84</td>
</tr>
<tr>
<td>4</td>
<td>0.183</td>
<td>0.209</td>
<td>0.025</td>
<td>13.00</td>
</tr>
<tr>
<td>5</td>
<td>0.195</td>
<td>0.222</td>
<td>0.028</td>
<td>13.22</td>
</tr>
<tr>
<td>6</td>
<td>0.192</td>
<td>0.218</td>
<td>0.026</td>
<td>12.92</td>
</tr>
<tr>
<td>7</td>
<td>0.202</td>
<td>0.229</td>
<td>0.027</td>
<td>12.39</td>
</tr>
<tr>
<td>8</td>
<td>0.236</td>
<td>0.263</td>
<td>0.027</td>
<td>10.97</td>
</tr>
<tr>
<td>9</td>
<td>0.202</td>
<td>0.230</td>
<td>0.028</td>
<td>12.77</td>
</tr>
<tr>
<td>10</td>
<td>0.197</td>
<td>0.222</td>
<td>0.025</td>
<td>12.06</td>
</tr>
</tbody>
</table>

In the Scenario 2, Case 1 example, the effects of specifying dispersions in terms of relative ellipse size and position are illustrated. These trajectories allow for percent differences up to $p_r = 10\%$, and maintain relatively similar trajectories and clustering of controls. For $p_r = 5\%$, there are very clearly four locations where the controls occur in each Monte Carlo run, shown in Fig. 15.22. The trajectories in Fig. 15.21 and $\Delta v$ results in Table 15.11 also show that although the initial along-track position varies by hundreds of meters, the planner $\Delta v$ only varies by 0.03 m/s. This is due to the fact that the chaser is initialized in nearly the same Initial Approach flyby orbit for each run.

15.5.2 Case 2 - Safe Traveling Ellipse with Circular Chief Orbit and Perturbations

The examples for Scenario 2, Case 2, Safe Traveling Ellipse, are Example 2 and Example 3, as defined in Table 14.5. Example 2 is a 100×50 m safety ellipse at 500 m on the v-bar to a 100×50 m safety ellipse that circumnavigates the chief, with final time of 0.6 chief revolutions. Dispersions with safety ellipse percentage errors of $p_{SE} = 10\%$ and 5% are examined. The trajectories with $p_{SE} = 10\%$ dispersions are shown in Fig. 15.23, the corresponding control histories are in Fig. 15.24, and $\Delta v$ in Table 15.12.
Fig. 15.23: Scenario 2, Case 2, Example 2 trajectories with $p_{SE} = 10\%$ dispersions (spherical model, with perturbations)

Fig. 15.24: Scenario 2, Case 2, Example 2 control history with $p_{SE} = 10\%$ dispersions (spherical model, circular chief, with perturbations)
Table 15.12: Monte Carlo $\Delta v$ for Scenario 2, Case 2, Example 2 with $p_{SE} = 10\%$ dispersions

<table>
<thead>
<tr>
<th>MC Run</th>
<th>$\Delta v_{opt}$ (m/s)</th>
<th>$\Delta v_{sim}$ (m/s)</th>
<th>Diff. (m/s)</th>
<th>$p_D$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.185</td>
<td>0.191</td>
<td>0.006</td>
<td>3.21</td>
</tr>
<tr>
<td>2</td>
<td>0.128</td>
<td>0.133</td>
<td>0.005</td>
<td>3.74</td>
</tr>
<tr>
<td>3</td>
<td>0.174</td>
<td>0.180</td>
<td>0.006</td>
<td>3.38</td>
</tr>
<tr>
<td>4</td>
<td>0.202</td>
<td>0.208</td>
<td>0.006</td>
<td>3.10</td>
</tr>
<tr>
<td>5</td>
<td>0.195</td>
<td>0.201</td>
<td>0.006</td>
<td>3.27</td>
</tr>
<tr>
<td>6</td>
<td>0.175</td>
<td>0.180</td>
<td>0.006</td>
<td>3.24</td>
</tr>
<tr>
<td>7</td>
<td>0.182</td>
<td>0.188</td>
<td>0.006</td>
<td>3.23</td>
</tr>
<tr>
<td>8</td>
<td>0.141</td>
<td>0.146</td>
<td>0.005</td>
<td>3.44</td>
</tr>
<tr>
<td>9</td>
<td>0.213</td>
<td>0.219</td>
<td>0.006</td>
<td>2.84</td>
</tr>
<tr>
<td>10</td>
<td>0.169</td>
<td>0.175</td>
<td>0.006</td>
<td>3.20</td>
</tr>
</tbody>
</table>

The Example 2 trajectories with $p_{SE} = 5\%$ dispersions are shown in Fig. 15.25, and the corresponding control histories are in Fig. 15.26, and $\Delta v$ in Table 15.13.

![Fig. 15.25: Scenario 2, Case 2, Example 2 trajectories with $p_{SE} = 5\%$ dispersions (spherical model, with perturbations)](image-url)
Fig. 15.26: Scenario 2, Case 2, Example 2 control history with $p_{SE} = 5\%$ dispersions (spherical model, circular chief, with perturbations)

Table 15.13: Monte Carlo $\Delta v$ for Scenario 2, Case 2, Example 2 with $p_{SE} = 5\%$ dispersions

<table>
<thead>
<tr>
<th>MC Run</th>
<th>$\Delta v_{opt}$ (m/s)</th>
<th>$\Delta v_{sim}$ (m/s)</th>
<th>Diff. (m/s)</th>
<th>$p_D$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.151</td>
<td>0.156</td>
<td>0.005</td>
<td>3.39</td>
</tr>
<tr>
<td>2</td>
<td>0.168</td>
<td>0.174</td>
<td>0.006</td>
<td>3.23</td>
</tr>
<tr>
<td>3</td>
<td>0.156</td>
<td>0.162</td>
<td>0.005</td>
<td>3.43</td>
</tr>
<tr>
<td>4</td>
<td>0.173</td>
<td>0.179</td>
<td>0.006</td>
<td>3.23</td>
</tr>
<tr>
<td>5</td>
<td>0.177</td>
<td>0.183</td>
<td>0.006</td>
<td>3.17</td>
</tr>
<tr>
<td>6</td>
<td>0.163</td>
<td>0.168</td>
<td>0.005</td>
<td>3.25</td>
</tr>
<tr>
<td>7</td>
<td>0.188</td>
<td>0.194</td>
<td>0.006</td>
<td>3.06</td>
</tr>
<tr>
<td>8</td>
<td>0.177</td>
<td>0.183</td>
<td>0.006</td>
<td>3.15</td>
</tr>
<tr>
<td>9</td>
<td>0.185</td>
<td>0.190</td>
<td>0.006</td>
<td>3.13</td>
</tr>
<tr>
<td>10</td>
<td>0.148</td>
<td>0.153</td>
<td>0.005</td>
<td>3.45</td>
</tr>
</tbody>
</table>

Scenario 2, Case 2, Example 2 results show that for the Safe Traveling Ellipse, when a small final time is specified (0.6 chief revolutions), the trajectories primarily result in two-burn solutions. These cases are relatively unaffected by the dispersions on initial ellipse size and position (even up to $p_{SE} = 10\%$), as can be seen in the $\Delta v$ results in Table 15.12 and 15.13.
Example 3 is a 100×50 m safety ellipse at 1 km on the v-bar to a 100×50 m safety ellipse that circumnavigates the chief, with final time of 2.4 chief revolutions. Dispersions with safety ellipse percentage errors of $p_{SE} = 10\%$ and 5$\%$ are examined. The trajectories with $p_{SE} = 10\%$ dispersions are shown in Fig. 15.27, the corresponding control histories are in Fig. 15.28, and $\Delta v$ in Table 15.14.

Fig. 15.27: Scenario 2, Case 2, Example 4 trajectories with $p_{SE} = 10\%$ dispersions (spherical model, with perturbations)

Fig. 15.28: Scenario 2, Case 2, Example 4 control history with $p_{SE} = 10\%$ dispersions (spherical model, circular chief, with perturbations)
Table 15.14: Monte Carlo $\Delta v$ for Scenario 2, Case 2, Example 4 with $p_{SE} = 10\%$ dispersions

<table>
<thead>
<tr>
<th>MC Run</th>
<th>$\Delta v_{opt}$ (m/s)</th>
<th>$\Delta v_{sim}$ (m/s)</th>
<th>Diff. (m/s)</th>
<th>$p_D$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.059</td>
<td>0.072</td>
<td>0.013</td>
<td>19.51</td>
</tr>
<tr>
<td>2</td>
<td>0.081</td>
<td>0.095</td>
<td>0.015</td>
<td>16.59</td>
</tr>
<tr>
<td>3</td>
<td>0.079</td>
<td>0.097</td>
<td>0.018</td>
<td>20.44</td>
</tr>
<tr>
<td>4</td>
<td>0.060</td>
<td>0.074</td>
<td>0.014</td>
<td>21.58</td>
</tr>
<tr>
<td>5</td>
<td>0.079</td>
<td>0.096</td>
<td>0.017</td>
<td>18.97</td>
</tr>
<tr>
<td>6</td>
<td>0.080</td>
<td>0.095</td>
<td>0.015</td>
<td>16.81</td>
</tr>
<tr>
<td>7</td>
<td>0.065</td>
<td>0.080</td>
<td>0.015</td>
<td>20.25</td>
</tr>
<tr>
<td>8</td>
<td>0.079</td>
<td>0.095</td>
<td>0.016</td>
<td>18.03</td>
</tr>
<tr>
<td>9</td>
<td>0.107</td>
<td>0.123</td>
<td>0.017</td>
<td>14.61</td>
</tr>
<tr>
<td>10</td>
<td>0.062</td>
<td>0.077</td>
<td>0.015</td>
<td>21.79</td>
</tr>
</tbody>
</table>

The Example 3 trajectories with $p_{SE} = 5\%$ dispersions are shown in Fig. 15.29, and the corresponding control histories are in Fig. 15.30, and $\Delta v$ in Table 15.15.

Fig. 15.29: Scenario 2, Case 2, Example 3 trajectories with $p_{SE} = 5\%$ dispersions (spherical model, with perturbations)
Fig. 15.30: Scenario 2, Case 2, Example 3 control history with $p_{SE} = 5\%$ dispersions (spherical model, circular chief, with perturbations)

![Control History Diagram](image)

Table 15.15: Monte Carlo $\Delta v$ for Scenario 2, Case 2, Example 3 with $p_{SE} = 5\%$ dispersions

<table>
<thead>
<tr>
<th>MC Run</th>
<th>$\Delta v_{opt}$ (m/s)</th>
<th>$\Delta v_{sim}$ (m/s)</th>
<th>Diff. (m/s)</th>
<th>$p_D$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.072</td>
<td>0.087</td>
<td>0.015</td>
<td>19.15</td>
</tr>
<tr>
<td>2</td>
<td>0.076</td>
<td>0.093</td>
<td>0.017</td>
<td>19.77</td>
</tr>
<tr>
<td>3</td>
<td>0.080</td>
<td>0.095</td>
<td>0.015</td>
<td>17.30</td>
</tr>
<tr>
<td>4</td>
<td>0.070</td>
<td>0.087</td>
<td>0.017</td>
<td>22.14</td>
</tr>
<tr>
<td>5</td>
<td>0.076</td>
<td>0.091</td>
<td>0.016</td>
<td>18.90</td>
</tr>
<tr>
<td>6</td>
<td>0.075</td>
<td>0.090</td>
<td>0.016</td>
<td>18.87</td>
</tr>
<tr>
<td>7</td>
<td>0.069</td>
<td>0.084</td>
<td>0.016</td>
<td>20.30</td>
</tr>
<tr>
<td>8</td>
<td>0.076</td>
<td>0.091</td>
<td>0.015</td>
<td>18.17</td>
</tr>
<tr>
<td>9</td>
<td>0.074</td>
<td>0.090</td>
<td>0.016</td>
<td>19.59</td>
</tr>
<tr>
<td>10</td>
<td>0.071</td>
<td>0.087</td>
<td>0.016</td>
<td>19.96</td>
</tr>
</tbody>
</table>

In Case 2, Example 3, the effects of dispersions are more significant due to the final transfer time of 2.4 chief revolutions. For these runs, the difference in trajectories and control histories between just $p_{SE} = 10\%$ and $p_{SE} = 5\%$ is large, as seen when comparing Figs. 15.27-15.28 to Figs. 15.29-15.30. Very little variability is observed when $p_{SE} = 5\%$, compared to the case where $p_{SE} = 10\%$. 
15.5.3 Case 3 - Safe Final Approach with Circular Chief Orbit and Perturbations

The Monte Carlo example for Scenario 2, Case 3, Safe Final Approach, is Example 4 defined in Table 14.8. In this example, the deputy is approaching from a $200 \times 100$ m circumnavigating safety ellipse to a $100 \times 50$ m circumnavigating safety ellipse, and with a final time of 1.2 chief revolutions. Dispersions with safety ellipse percentage errors of $p_r = 10\%$, and $5\%$ are examined. The trajectories with $p_r = 10\%$ dispersions are shown in Fig. 15.31, the corresponding control histories are in Fig. 15.32, and $\Delta v$ in Table 15.16.

Fig. 15.31: Scenario 2, Case 3, Example 4 trajectories with $p_r = 10\%$ dispersions (Cartesian model, with perturbations)
Fig. 15.32: Scenario 2, Case 3, Example 4 control history with \(p_r = 10\%\) dispersions (Cartesian model, circular chief, with perturbations)

Table 15.16: Monte Carlo \(\Delta v\) for Scenario 2, Case 3, Example 4 with \(p_r = 10\%\) dispersions

<table>
<thead>
<tr>
<th>MC Run</th>
<th>(\Delta v_{opt}) (m/s)</th>
<th>(\Delta v_{sim}) (m/s)</th>
<th>Diff. (m/s)</th>
<th>(p_D) (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.104</td>
<td>0.104</td>
<td>-0.000</td>
<td>0.17</td>
</tr>
<tr>
<td>2</td>
<td>0.105</td>
<td>0.105</td>
<td>-0.000</td>
<td>0.10</td>
</tr>
<tr>
<td>3</td>
<td>0.073</td>
<td>0.077</td>
<td>0.003</td>
<td>4.65</td>
</tr>
<tr>
<td>4</td>
<td>0.066</td>
<td>0.069</td>
<td>0.003</td>
<td>4.89</td>
</tr>
<tr>
<td>5</td>
<td>0.083</td>
<td>0.086</td>
<td>0.003</td>
<td>3.95</td>
</tr>
<tr>
<td>6</td>
<td>0.056</td>
<td>0.059</td>
<td>0.004</td>
<td>6.17</td>
</tr>
<tr>
<td>7</td>
<td>0.087</td>
<td>0.091</td>
<td>0.003</td>
<td>3.69</td>
</tr>
<tr>
<td>8</td>
<td>0.193</td>
<td>0.196</td>
<td>0.003</td>
<td>1.71</td>
</tr>
<tr>
<td>9</td>
<td>0.067</td>
<td>0.069</td>
<td>0.003</td>
<td>4.13</td>
</tr>
<tr>
<td>10</td>
<td>0.122</td>
<td>0.122</td>
<td>0.000</td>
<td>0.06</td>
</tr>
</tbody>
</table>

The Example 4 trajectories with \(p_r = 5\%\) dispersions are shown in Fig. 15.33, and the corresponding control histories are in Fig. 15.34, and \(\Delta v\) in Table 15.17.
Fig. 15.33: Scenario 2, Case 3, Example 4 trajectories with $p_r = 5\%$ dispersions (Cartesian model, with perturbations)

Fig. 15.34: Scenario 2, Case 3, Example 4 control history with $p_r = 5\%$ dispersions (Cartesian model, circular chief, with perturbations)
Table 15.17: Monte Carlo $\Delta v$ for Scenario 2, Case 3, Example 4 with $p_r = 5\%$ dispersions

<table>
<thead>
<tr>
<th>MC Run</th>
<th>$\Delta v_{opt}$ (m/s)</th>
<th>$\Delta v_{sim}$ (m/s)</th>
<th>Diff. (m/s)</th>
<th>$p_D$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.075</td>
<td>0.076</td>
<td>0.000</td>
<td>0.57</td>
</tr>
<tr>
<td>2</td>
<td>0.073</td>
<td>0.077</td>
<td>0.003</td>
<td>4.60</td>
</tr>
<tr>
<td>3</td>
<td>0.093</td>
<td>0.093</td>
<td>-0.000</td>
<td>0.06</td>
</tr>
<tr>
<td>4</td>
<td>0.072</td>
<td>0.076</td>
<td>0.003</td>
<td>4.65</td>
</tr>
<tr>
<td>5</td>
<td>0.070</td>
<td>0.070</td>
<td>0.000</td>
<td>0.59</td>
</tr>
<tr>
<td>6</td>
<td>0.068</td>
<td>0.071</td>
<td>0.004</td>
<td>5.08</td>
</tr>
<tr>
<td>7</td>
<td>0.075</td>
<td>0.078</td>
<td>0.003</td>
<td>4.26</td>
</tr>
<tr>
<td>8</td>
<td>0.063</td>
<td>0.066</td>
<td>0.003</td>
<td>4.98</td>
</tr>
<tr>
<td>9</td>
<td>0.067</td>
<td>0.071</td>
<td>0.003</td>
<td>5.00</td>
</tr>
<tr>
<td>10</td>
<td>0.076</td>
<td>0.079</td>
<td>0.003</td>
<td>4.52</td>
</tr>
</tbody>
</table>

The final results for Case 3 show a similar trend to that of Case 2. For larger safety ellipse dispersions with $p_{SE} = 10\%$, the control histories vary greatly as seen in Fig. 15.32. However, for $p_{SE} = 5\%$, there are clearly three distinct locations for the controls in the case of transferring from a large to a small safety ellipse, shown by Fig. 15.34.

15.6 Conclusions

In this chapter, the optimal planner and simulation were tested under the effects of dispersions on the trajectory’s initial conditions. The results of the dispersion analysis show conclusively that in many cases relatively small initial dispersions can result in large variations of the planned trajectories. These dispersions affected both the trajectories and the optimal control histories. However, as expected, smaller dispersions on initial conditions resulted in less variation of planned trajectories and controls.

The effects of final transfer time and proximity of the deputy to the chief also play a major role in the dispersion analysis. For far initial ranges, the initial relative position and velocity dispersions are generally much larger. This leads to a great variety of possible optimal trajectories. Examples are shown in Figs. 15.1, 15.5, 15.19, and 15.27. However, if the final time is decreased, range decreased, or (certainly) the initial dispersions are
decreased, then the trajectories from the Monte Carlo analyses exhibit less variation. These trends can be seen in Figs. 15.3, 15.11, 15.24, and 15.29.

The percent difference between the simulation and planner $\Delta v$ results were as expected from previous chapters and nothing unusual occurred due to the effects of adding dispersions. All percent differences were relatively small and the primary contributions were due to the effects of tracking perturbations over long-duration simulations (See Table 15.13 compared to Table 15.15). The main variations from the planner optimal $\Delta v$'s are primarily due to the initial velocity dispersions. This is first seen in the Scenario 1, Case 1 results in Table 15.1, where for 0.25 m/s 1-$\sigma$ velocity dispersions, the 1-$\sigma$ standard deviation of the planner optimal $\Delta v$ is 0.29 m/s. Additionally, the initial out-of-plane velocity dispersions greatly affected the overall LVLH trajectory from each Monte Carlo run, since the RSS of the in-plane and out-of-plane components are minimized at each discretization point along the trajectory.
CHAPTER 16
CONCLUSIONS AND FUTURE WORK

16.1 Conclusions

New solutions to the optimal autonomous trajectory planning problem for rendezvous and proximity operations (RPO) of two satellites were developed using convex optimization theory. Traditional spacecraft rendezvous, inspection, and final approach trajectories with respect to a circular chief reference orbit were considered, as well as new approaches which ensure the passive safety of flight of two satellites. A variety of linear dynamics models were investigated, including: Hill Clohessy-Wiltshire (HCW) dynamics to describe the relative motion in a local-horizontal local-vertical frame, relative orbital motion dynamics relative to a spinning or uncontrolled spacecraft, and a new formulation that implemented linear relative orbital elements (LROEs). In this work, convex optimization technology provided the foundation for the formulation of an optimal RPO trajectory planner. Convex optimization theory was successfully applied to optimal RPO trajectory planning scenarios using each of the dynamics models considered in this research. These convex problems provided global optimality, polynomial-time convergence, and were solved with rapid, numerically stable, and efficient solvers.

The problem of optimal trajectory planning for orbital RPO was cast as a specific convex optimization problem; a second-order cone program (SOCP), with the objective to minimize propellant. The SOCP is self-starting and has deterministic convergence properties, making it an ideal candidate for solving this problem and providing autonomous real-time results. In this analysis, all of the convex trajectory optimization problems were cast as fixed-final time problems, by virtue of satisfying lighting conditions, mission communications constraints, and mission time line requirements. A traditional fixed-time RPO scenario was considered including rendezvous, inspection, and final approach trajectories.
In addition, a new approach involving trajectory planning to ensure passive safety of flight was also presented. This formulation was based upon the development of constraints which ensured zero probability of collision in the event of a passive failure on the deputy satellite, such as power loss, computer shutdown/reboot, or a suspension of normal activities due to mission/vehicle anomalies. This scenario consisted of safe initial approach, safe traveling ellipse, and safe final approach trajectories.

A variety of constraints were imposed on these trajectories, in addition to the fixed final-time constraint. These included initial/final boundary conditions, maximum thrust availability, approach corridors, spherical keep-out zones, stand-off planes, and passive safety of flight constraints. These constraints were implemented as a means to ensure either active or passive safety of flight. Active safe trajectories typically had safety constraints along the duration of the trajectory, however, if active control was lost mid-course, there was typically no guarantee of a passive abort scenario. Active safety constraints included the approach corridor, spherical keep-out zone, and stand-off plane (or half-space) constraints. Passive safety of flight constraints for trajectory design ensured the deputy satellite is in a passively safe relative ellipse at all times, to avoid collision with the chief satellite in the event of a passive failure.

Maximum thrust constraints, approach corridors, and half-spaces defined a convex solution space for the SOCP problem variables, and therefore conformed to the second-order conic constraint. The spherical keep-out zone enforced the deputy’s relative position to lie outside a desired sphere at all times (ensuring a minimum safe distance) and was inherently nonconvex. The passive safety of flight constraints were also nonconvex. In the event that nonconvex constraints were required, a method of sequential convex programming (SCP) was adopted, whereby all nonconvex constraints were convexified via linearized approximations, and a convex program was iteratively solved.

Many different types of RPO trajectory problems were examined. These problem formulations are summarized as follows:
1. **Problem Formulation I**: Minimum propellant, fixed final-time transfer trajectories from point A to point B, with maximum acceleration constraints, relative to a chief-centered reference frame (LVLH or other rotating frame). The optimal solutions generally require two or more impulsive maneuvers, or continuous maneuvers when the maximum acceleration constraint is enforced. Convergence is guaranteed with deterministic CPU requirements.

2. **Problem Formulation II**: Minimum propellant, fixed final-time transfer trajectories from point A to point B, with maximum acceleration and approach corridor constraints, relative to a chief-centered reference frame (LVLH or other rotating frame). The optimal solutions generally require two or more impulsive maneuvers, or continuous maneuvers when the maximum acceleration constraint is enforced or while near the boundary of the approach corridor constraint. Convergence is guaranteed with deterministic CPU requirements.

3. **Problem Formulation III**: Minimum propellant, fixed final-time transfer trajectories from point A to point B, with maximum acceleration, approach corridor, and spherical keep-out zone constraints, relative to a chief-centered reference coordinate frame (LVLH or other rotating frame). Optimal solutions generally require two or more impulsive maneuvers, or continuous maneuvers when the maximum acceleration constraint is enforced or while near the boundary of the approach corridor or spherical keep-out zone constraints. The method of sequential convex optimization is currently required in the keep-out zone cases, though all cases examined in this work show convergence in just a few iterations.

4. **Problem Formulation IV**: Minimum propellant, fixed final-time transfer trajectories from relative ellipse A to relative ellipse B, with maximum acceleration constraints, relative to a chief-centered reference frame. The optimal solutions generally require two or more impulsive maneuvers, or continuous maneuvers when the maximum acceleration constraint is enforced. Convergence is guaranteed with deterministic CPU
requirements.

5. **Problem Formulation V**: Minimum propellant, fixed final-time transfer trajectories from relative ellipse A to relative ellipse B, with maximum acceleration constraints and safety of flight constraints, relative to a chief-centered reference frame. The optimal solutions generally require two or more impulsive maneuvers, or continuous maneuvers when the maximum acceleration constraint is enforced. The method of sequential convex optimization is currently required in the safety of flight scenarios, though all cases examined in this work show convergence in just a few iterations.

For each problem formulation shown above, specific cases were examined, depending on the different RPO phases. The trajectory phases considered in this research were formulated according to the typical time line of an RPO mission. In the traditional RPO scenario, they were separated into Initial Approach, Way Point Following/Inspection, Final Approach. In the new safety of flight RPO scenario, the cases are very similar, but ensure passive safety. These are separated into Safe Initial Approach, Safe Traveling Ellipse, and Safe Final Approach. A summary of the scenarios and cases that were examined is provided here.

**Scenario 1** - Traditional Rendezvous, Inspection, and Final Approach:
- **Case 1** - Initial Approach
- **Case 2** - Way-point Following/Inspection
- **Case 3** - Final Approach

**Scenario 2** - Safe Rendezvous, Traveling Ellipse, and Final Approach:
- **Case 1** - Safe Initial Approach
- **Case 2** - Safe Traveling Ellipse
- **Case 3** - Safe Final Approach

The HCW relative orbital motion model, rotating chief body-fixed frame dynamics, and the LROE dynamics were the primary orbital dynamics models used in the optimal RPO
trajectory planner. These dynamics models were presented in Chapters 5 and 6. The HCW model and the new LROE dynamics model were described in detail in Chapter 5. Important characteristics of these dynamics were also illustrated, and several illustrations of common relative motion trajectories were provided, to gain a better understanding of these ROEs with regards to typical RPO scenarios. Chapter 6 presented the relative motion equations for a deputy satellite, with respect to a rotating chief’s body-fixed frame. The resulting equations are LTV, however, it was shown that when the rotation rate of the chief’s body-fixed frame is LVLH (nadir pointing), then the equations reduced to the standard HCW model. Furthermore, if the spin rate of the chief satellite is much greater than the orbital rate, the resulting dynamics reduced to a LTI model.

The first RPO trajectory planning algorithm using the HCW dynamics model was developed and formulated as an SOCP in Chapter 7. This included linear equality constraints for the problem dynamics, and inequality constraints for the slack variables, maximum control acceleration, and approach corridor. Spherical keep-out zone constraints were also considered, and were implemented using the linearized approximation technique for the sphere constraints. This successfully led to optimal planned paths around the nonconvex spherical keep-out zone. Several planned trajectory results for these cases were shown. This resulted in optimal $N$-impulse trajectories and control histories while not on the boundaries of path constraints, and continuous control while on the boundaries of the constraints. This analysis was performed for the following scenarios: Initial Approach, Way-point following (with or without a spherical keep-out zone), and Final Approach with an approach corridor. Furthermore, the utility of the ‘drift without control’ constraints resulted in much smoother optimal control solutions, while on the boundary of the path constraints.

The rotating chief body-fixed model was used in Chapter 8 to formulate a convex RPO planner algorithm for maneuvers which were best described in an uncontrolled chief’s body-fixed frame. These trajectories included Way-point Following/Inspection and Final Approach cases, where constraints such as maximum thrust acceleration, approach corridors, spherical keep-out zones, and minimum safe distance stand-off planes were included.
The optimal RPO trajectory planner in this case involved developing an SOCP problem with the rotating body-fixed frame dynamics, as derived in Chapter 6. This problem had linear equality constraints for the dynamics, and inequality constraints for the slack variables, maximum control acceleration, approach corridor, and minimum stand-off plane. Spherical keep-out zone constraints were also considered here, which were nonconvex, therefore requiring SCP methods. In the final approach scenario, implementing an approach corridor and the minimum distance constraint resulted in scenarios where the spherical keep-out zone was inherently satisfied. Many optimal trajectories were planned using this technique, resulting in trajectory and optimal control history plots for the following scenarios: Initial Approach, Way-point Following (with or without a spherical keep-out zone), and Final Approach with an approach corridor and minimum distance constraint. The ‘drift without control’ constraints were also used here, and provided a much smoother optimal control solution while on the boundary of the path constraints.

Chapter 9 presented an approach to design optimal RPO trajectories that ensure zero probability of collision in the event of a passive failure on the deputy satellite. This required developing passive safety of flight constraints using relative orbital elements, which included constraints on relative ellipse size, position, and orientation. Several of these constraints were inherently nonconvex, so the SCP method was adopted. The problem was formulated as a minimum $\Delta v$ approximate convex optimization problem, in which the optimal transfer trajectories are guaranteed to not pass within a prescribed distance of the target satellite, in the event of passive failures. Each nonconvex constraint was convexified via a linearized approximation and implemented in the SCP problem.

It was shown that a passively safe trajectory consists of three fundamental building blocks: 1) flyby trajectories from above, 2) flyby trajectories from below, and 3) circumnavigating trajectories with the proper in-plane/out-of-plane phasing. Furthermore, end-to-end passively safe rendezvous and proximity operations trajectories were designed in terms of three trajectory elements: 1) initial approach from a flyby ellipse to an offset circumnavigating safety ellipse, 2) a transfer from an offset circumnavigating safety ellipse to another
offset circumnavigating safety ellipse of the same size, and 3) increasing or decreasing the size of a circumnavigating safety ellipse. By considering these three trajectory elements, the constraints for passive safety of flight in terms of relative orbital elements were greatly simplified. The planner results in Chapter 9 showed that the implementation of the passive safety constraints led to globally optimal N-impulse trajectories that satisfied all passive safety requirements.

A custom SOCP algorithm used to solve the optimal trajectory planning problem was developed in Chapter 10. This involved formulating the problem dynamics in two different ways, the states-and-controls formulation and the control-only formulation. The custom SOCP also defined each second-order conic constraint specifically for the slack variables, maximum acceleration, and approach corridor constraints. Implementing these constraints and using indexing greatly reduced the total computation time and memory required for calculating the gradient and Hessian of the second-order conic functions on each Newton-step iteration. However, ultimately the code ran slowly compared to the commercial solvers (specifically Mosek), due to the numerical conditioning, factorization, row-rank reduction, and coding techniques that commercial solvers take advantage of.

Data on algorithm CPU and memory requirements for a variety of scenarios was collected and reported in Chapter 11. The results from the CPU timing analysis showed the effects of the overall computation time required for different scenarios, formulations, and number of discretization points. It was evident that for CPUs with slower clock speeds, the total time required to solver the RPO trajectory planner problem increased. This was seen to closely correlate with the number of instructions required. As expected, for a certain number of required instructions, the CPU with greater instructions-per-second capabilities was able to solve the convex optimization problem faster. Memory requirement results showed that some formulations of the RPO trajectory planning problem significantly saved memory, while others required much more memory for larger problems, making these formulations undesirable. As expected, the total memory and number of instructions is shown to nearly double, as the size of the problem doubles. However, for each problem size, the memory
required between the different problem formulations in many cases did not differ greatly.

The final results sections implemented planned optimal trajectories in a nonlinear simulation. These results include Scenario 1 (traditional approach, inspection, and final approach) in Chapter 13, and Scenario 2 (safe rendezvous, traveling ellipse, and final approach) in Chapter 14. These two scenarios were followed by a Monte Carlo dispersion analysis, which illustrated the effects of adding random dispersions on the initial conditions, to see the resulting optimal trajectories and control histories. The nonlinear simulation analysis was based on the assumption of perfect navigation and control actuation. The controller was purposely designed to have a relatively slow response time, since implementing a faster controller led to much larger $\Delta v$ percent difference errors.

The conclusions reached regarding the generation of reference trajectories in Cartesian versus spherical coordinates were the same in all cases. There was clearly greater accuracy when using the spherical model for cases with large along-track separation without perturbations. By representing the position and velocity in a curvilinear LVLH frame, the modeling errors associated with a Cartesian LVLH frame were modeled more accurately. However, once the perturbations were added into the simulation, the benefits of modeling using spherical coordinates were not significant. In these cases, the model must be improved (by adding the linearized J2/drag/SRP terms to the HCW dynamics), to account for these effects.

It was evident that the ellipticity of the chief’s orbit also plays a major role in the cases with large along-track/cross-track separation, or long-duration planned trajectories with large final transfer times. The results show that in the Initial Approach cases, the effects of chief eccentricity had a significant effect on the overall $\Delta v$ differences due to the planner modeling errors and controller trajectory following errors. In these cases, and for more elliptic target orbits, the orbit dynamics model in the planner must be improved to account for these effects. However, it was seen in the $\Delta v$ results that the effects of eccentricity on the Final Approach trajectories is minimal. This was due to the fact that in these examples the deputy was in relatively close proximity of the chief, and the final transfer times were
small in comparison to Initial Approach results.

In Chapter 15, the optimal planner and simulation were tested under the effects of dispersions on the trajectory's initial conditions. The results of the dispersion analysis conclusively showed that in many cases relatively small initial dispersions can result in large variations of the planned trajectories and optimal controls. However, as expected, smaller dispersions on initial conditions resulted in less variation of planned trajectories and controls.

The effects of final transfer time and proximity of the deputy to the chief also played a major role in the dispersion analysis. For far initial ranges, the initial relative position and velocity dispersions are generally much larger. This lead to a great variety of possible optimal trajectories. However, if the final time is decreased, range decreased, or the initial dispersions are decreased, then the trajectories from the Monte Carlo analyses exhibit less variation.

The percent difference between the simulation and planner $\Delta v$ results were as expected from previous chapters and nothing unusual occurred due to the effects of adding dispersions. All percent differences were relatively small and the primary contributions were due to the effects of tracking perturbations over long-duration simulations. The main variations from the planner optimal $\Delta v$'s are primarily due to the initial velocity dispersions. Additionally, the initial out-of-plane velocity dispersions greatly affected the overall LVLH trajectory from each Monte Carlo run.

### 16.2 Future Work

Though convex optimization techniques may be practical for autonomous RPO trajectory planning, significant work remains in this area of study for further assurance. One primary area for future exploration, as a result of this research, is the development of higher-fidelity dynamics models in the convex optimization problems. The HCW equations have proven to work very well in the case of an unperturbed circular chief reference orbit, however, the addition of perturbations and low-eccentricity terms would serve to reduce the trajectory following $\Delta v$ errors, shown in the nonlinear simulation results sections.

One method to improve the dynamics model would be to implement the approximate
state transition matrices from the literature, which include eccentric reference orbits, J2, and
drag. Another method would be to numerically integrate the time-varying state transition
matrix which includes the linearized two-body, J2, and drag terms. One final method
would be to implement the full nonlinear relative motion dynamics, using the method of
successive approximations, whereby a nonlinear function of the relative states are updated in
each iteration of sequential convex programming. This would be similar to the formulation
by Lu [22], in using the linear gravity model to develop an autonomous RPO problem.
Upgrading the state transition matrix in this way may prove to model the relative motion
dynamics much more accurately.

Another major aspect of the investigation into autonomous trajectory planning involves
the addition of navigation and controller actuation errors into the nonlinear simulation.
Adding angles-only or LiDAR measurements for autonomous relative navigation, and the
associated effects on the relative position and velocity dispersions, would show with greater
certainty how efficient (or inefficient) autonomous trajectory planning truly is. Though it is
reasonable to propose this analysis, there are numerous variables in this problem, therefore
the study may quickly become very vehicle- and/or scenario-dependent. For guidance to run
autonomously alongside navigation, the development of trajectory planning methods that
include navigation and control actuation errors would be highly beneficial, as considered in
the work by Louembet [159].

Further development of a custom autonomous RPO trajectory planning algorithm, and
testing it on hardware in a great variety of scenarios, is eventually required to examine
robustness and fault-tolerance. Due to the great number of variables (transfer time, ini-
tial/final conditions, maximum acceleration, measurement error, control law and error) and
other constraints not considered in this analysis (lighting conditions, under-burns, plume
impingement, attitude, thruster selection, communications) there is always the possibility of
setting up a problem for which there is no feasible solution. For example, commanding a ma-
neuver that requires much higher thrust levels than are currently available might be deemed
infeasible by an autonomous algorithm. Autonomous systems must have the capability to
detect these infeasibilities, and handle them accordingly.
Bibliography


CURRICULUM VITAE

Nicholas G. Ortolano

Published Conference Papers


• Nicholas Ortolano, David Geller, and Aaron Avery. "Relative Orbital Motion Dynamics with Respect to a Rotating Spacecraft-fixed Frame." Proceedings of AAS GN&C Conference, Rocky Mountain Section 17-016, 2017.