

Real-Time Quasi-Analytical Trajectory Generation for Docking with Tumbling Objects

by

Alejandro D. Cabrales Hernandez

S.B., Massachusetts Institute of Technology (2018)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2020

© Massachusetts Institute of Technology 2020. All rights reserved.

Author
Department of Aeronautics and Astronautics
May 19, 2020

Certified by.....
David W. Miller
Professor, Aeronautics and Astronautics
Thesis Supervisor

Accepted by
Sertac Karaman
Chairman, Department Committee on Graduate Theses

Real-Time Quasi-Analytical Trajectory Generation for Docking with Tumbling Objects

by

Alejandro D. Cabrales Hernandez

Submitted to the Department of Aeronautics and Astronautics
on May 19, 2020, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

The capability to rendezvous and dock with tumbling objects has become prominent with increased interest in active debris removal, satellite servicing, and in-space assembly. Guidance and control algorithms have been developed in the literature to allow for a spacecraft to capture an uncooperative and tumbling object under several constraints such as collision avoidance, speed bounds, and thruster saturation. However, current algorithms for this capability do not address plume impingement due to thrusters, which can lead to damage to the target object, and can require the use of nonlinear solvers that neither guarantee convergence of a solution nor be deployed in real-time using current computational capabilities of spacecraft. This thesis presents a quasi-analytical guidance algorithm that allows for a spacecraft to soft-dock with a target, avoids plume impingement, and allows for real-time generation of trajectories with low computational expense. Several test cases compare the solution from this algorithm against a solution using pseudospectral methods and show similar performance at less than 0.1% computational cost, and an example scenario for docking with the the European Space Agency's ENVISAT is presented. Additionally, a discrete transport trajectory optimizer is presented for use as a first cut solution to transporting several components to the same halo orbit for in-space assembly.

Thesis Supervisor: David W. Miller

Title: Professor, Aeronautics and Astronautics

Acknowledgments

Funding and support for this research was provided by the Massachusetts Institute of Technology Department of Aeronautics and Astronautics through the Jack and Vickie Kerrebrock Fellowship as well as a research assistantship and teaching assistantship. The author is very grateful to all the sponsors for their support throughout my studies.

Thank you Professor David Miller for your advice and mentorship throughout my master's. Your guidance and support not only in my academic development but also in my career and personal development are extremely appreciated. I would also like to thank all the wonderful mentors that I have been lucky to have—and convinced me to pursue my master's and the Ph.D.—including Prof. Paulo Lozano, Dr. Alice Liu, Dr. Tupper Hyde, and Dr. Paul Mason.

I would also like to thank everyone in the Rumpus Room—Keenan, Michael, Oliver, and Tonio. The frequent trips to the Banana Lounge, grabbing coffee, lunch, etc. made this year great. Thanks to the rest of the SSL that makes the lab such a great place: Will, Becky, Hailee, Alex, Daniel, and everyone else. Additionally, I would like to thank all my other friends in the department, MIT, internships, and everywhere else including: David, Anthony, Annika, Katie, Matt, Cynthia. Your friendship has been a great treasure and I am incredibly lucky and thankful to have met everyone of you. Special thanks to Michael, Keenan and Cynthia for reading and correcting this thesis.

Finally, and most importantly, I would like to thank my mom for all her love, support, and patience throughout my life. Your countless sacrifices, hard work, and love have always inspired me to be a better person, and have made accomplishing this thesis possible.

Contents

1	Introduction	17
1.1	Motivation and Background	17
1.2	Overview of Methodology for Relative Operations for Autonomous Ma- neuvers	19
1.3	Research Objective	20
1.4	Thesis Organization	20
2	Literature Review	23
2.1	History of Rendezvous and Docking	23
2.2	State-of-the-Art Algorithms for Docking with Tumbling Objects . . .	25
3	Trajectory Optimization for Docking with Tumbling Objects	33
3.1	Description of Trajectory Rendezvous and Docking Phase	34
3.1.1	Necessary Information for Initiating Phase 3	36
3.1.2	Problem Formulation for Docking with Tumbling Objects . . .	37
3.2	Characterization of Docking Problem	39
3.3	Analytical Solution for the Flat Spin Docking Case	41
3.3.1	Necessary Conditions for Optimality	42
3.3.2	Possible Solutions to Flat Spin Problem	46
3.3.3	Summary of Flat Spin Solution	52
3.4	Quasi-Analytical Solution to the General Spin Docking Problem . . .	54
3.4.1	Necessary Conditions for Optimality	55
3.4.2	toc	57

3.4.3	Analytical Solution to Bang-Off Controller	60
3.4.4	Numerical Solution to Bang-Off-Bang Controller	63
3.4.5	Summary of Solution for the General Spin, Spherical Inertia Docking Problem	66
3.5	Analytical Trajectory Optimizer for the General Spin, General Inertia Docking Problem	69
3.5.1	Formulation of Limited Horizon Problem Under Constant An- gular Velocity Assumption	71
3.5.2	Analytical Model Predictive Controller Algorithm (AGA) . . .	74
4	Results and Analysis for the Analytical Guidance Algorithm	77
4.1	Trajectory Verification through GPOPS-II	77
4.1.1	Flat Spin Solution	78
4.1.2	for the co	84
4.1.3	uno	85
4.2	Simulation Environment for Analytical Guidance Algorithm Testing .	89
4.2.1	Assumptions for Simulation Environment	89
4.2.2	High Level Simulation Overview	90
4.2.3	Dynamics of Analytical Guidance Simulink Model	92
4.2.4	Guidance of AGA Simulink Model	93
4.2.5	Control of Analytical Guidance Simulink Model	93
4.3	Results for Analytical Guidance Simulation	95
4.3.1	Effect of Inertia Ratios in AGA Solutions	108
4.3.2	Summary of Analysis of the Analytical Guidance Algorithm .	111
4.4	ENVISAT Example Mission	112
5	Conclusion and Future Work	119
5.1	Thesis Contributions	120
5.2	Recommendations for Future Work	121
5.3	Code Source and Repositories	121

A	Trajectory Generation for In-Space Assembled Telescope	123
A.1	Background and Motivation for an In-Space Assembled Telescope . . .	123
A.2	Stable Manifold Trajectories in EML1 and SEL2 System	125
A.2.1	Generation of Halo Orbits	127
A.2.2	Generation of Stable Manifold Trajectories	130
A.3	Discrete Transport Trajectory Optimizer	133
A.3.1	Frame Conversions	134
A.3.2	Trajectory Optimizer Algorithm	136
A.4	Test Cases and Result	139
A.4.1	Earth-Moon L1	140
A.4.2	Sun-Earth L2	142
A.5	Discussion and Future Work for the Trajectory Optimizer	143
B	Analytical and Quasi-Analytical Solutions for Impulsive Docking Problems	145
B.1	Impulsive Solution to Flat Spin	145
B.2	Impulsive Solution to General Spin with Spherical Inertia	147
B.2.1	Impulsive Bang-Off Solution	147
B.2.2	Impulsive Bang-Off-Bang Solution	148

List of Figures

1-1	Artist departure of Restore-L spacecraft approaching for grabbing Landsat-7 as part of a mission servicing operation. Source: NASA [6]	18
1-2	Depiction of the four stages of Docking. Source: Fig. 1, Terán Espinoza et al [8]	19
2-1	High level overview of a typical cooperative docking procedure [9]	24
2-2	Notional example of Pontryagin’s Minimum Principal for a minimum fuel problem. The function $\tilde{\mathcal{H}}$, which corresponds to a simplified Hamiltonian for a double integrator, has different behavior for different values of the costate p	28
2-3	Basic structure for a model predictive controller (MPC). Source [32]	30
3-1	Representative example for Phase 3: trajectory rendezvous and docking. Note, x, y represents the body-fixed Target frame	35
3-2	Docking problem setup. Three frames are shown: an inertial frame, a Target body fixed frame, and a Chaser body fixed frame.	37
3-3	Possible forms for p_2 with $R_0 = 15$, $R_f = 5$, $\omega = 30^\circ/s$, $t_f = 4 s$, $u_{sat} = 10 m/s^2$	48
3-4	γ condition for all possible ω direction. Blue area represents the analytical Bang-Off controller. Red area represents the numerical Bang-Off-Bang controller.	58
3-5	Sensitivity Analysis for Analytical Switch and Final Times	62
3-6	Qualitative description for solution to the General Spin General Inertia docking problem	71

3-7	Chaser plume impingement on Target due to Bang-Off-Bang controller. t_f^- indicates the instantaneous time before the end of the maneuver when the Chaser is slowing down before achieving soft docking condition.	72
3-8	Proposed approach for limiting plume impingement in Target. R_p indicates minimum relative distance at which Bang-Off-Bang controller is allowed.	73
4-1	Comparison between GPOPS and analytical solution to Flat Spin nominal case	80
4-2	Resultant Hamiltonian for nominal Flat Spin Case. Lower GPOPS Error leads to Hamiltonian approach zero for all time.	82
4-3	Comparison of analytical and GPOPS solution for different flat spin scenarios	83
4-4	Comparison between GPOPS and analytical solution to General Spin with Spherical Inertia $\gamma \leq 1$	86
4-5	Comparison between GPOPS and analytical solution to General Spin with Spherical Inertia $\gamma > 1$	88
4-6	High level overview of Simulink model for testing AGA controller.	91
4-7	Visualization of AGA Tests's angular velocity history vs. Bang-Off/Bang-Off-Bang regions. Blue region corresponds to $\gamma \leq 1$ or the impulsive Bang-Off controller, red region corresponds to impulsive Bang-Off-Bang region.	97
4-8	Test 1: Results for analytical guidance simulation for spherical inertia with angular velocity such that $\gamma \leq 1$. Vertical grey bar represents end of guidance solution by AGA.	98
4-9	Visualization of AGA Tests's angular velocity history vs. Bang-Off/Bang-Off-Bang regions. Blue region corresponds to $\gamma \leq 1$ or the impulsive Bang-Off controller, red region corresponds to impulsive Bang-Off-Bang region.	100

4-10	Test 2: Results for analytical guidance simulation for spherical inertia with angular velocity such that $\gamma > 1$. Vertical grey bar represents end of guidance solution by AGA.	101
4-11	Test 3: Results for analytical guidance simulation for general inertia with angular velocity such that $\gamma \leq 1$ for all time. Vertical grey bar represents end of guidance solution by AGA.	104
4-12	Test 3: Relative position difference between AGA trajectory and GPOPS solution	105
4-13	Test 4: Results for analytical guidance simulation for general inertia with angular velocity such that $\gamma > 1$ for all time. No plume impingement protection. Vertical grey bar represents end of guidance solution by AGA.	106
4-14	Test 5: Results for analytical guidance simulation for general inertia with angular velocity such that $\gamma > 1$ for all time. With plume impingement protection, $R_p = 3 [m]$. Vertical grey bar represents end of guidance solution by AGA.	107
4-15	Effect if inertia ratio on fuel cost for AGA vs. GPOPS. Solution were obtain with $R_0 = 10 [m]$, $R_f = 1 [m]$, $u_{sat} = 5 [m/s^2]$, $\omega = [1, -5, 5] [^\circ/s]$ as nominal values	110
4-16	Comparison between the AGA and GPOPS solution for Run 1 and Run 10.	111
4-17	ENVISAT model, x, y, z represent the body axis. Source [44]	113
4-18	ENVISAT mission test results	115
4-19	AGA simulated docking trajectory for ENVISAT. Blue line corresponds to the docking point, $[4, 0, 0]$, and black line corresponds to the Chaser trajectory. Figures shown in inertial frame.	116
A-1	Overview of optimization framework. Source [7]	124
A-2	Diagram of the CR3BP. Inertial and rotating frame for the CR3BP are shown by $\{0, \hat{X}, \hat{Y}, \hat{Z}\}$ and $\{0, \hat{x}, \hat{y}, \hat{z}\}$	126

A-3	Qualitatively diagram of location of Libration points in the CR3BP .	127
A-4	Example of Sun-Earth L2 halo orbit with amplitude of 110,000 [km].	129
A-5	Example of un-Earth L2 halo orbit with amplitude of 110,000 [km]. Earth and Moon Orbit are shown for scale.	130
A-7	Selection of 60 stable manifold trajectories for nominal halo orbit of 110,000 km amplitude. Note some of the manifold trajectories do not go towards Earth.	132
A-8	EML1 transport trajectory example. Figure shows the proposed tra- jectory from launch site to point of joining the selected halo orbit. As- terisk represents point at which the lambert transfer trajectory joins the stable manifold trajectory.	141
B-1	Behavior of $\dot{x}(0)$, $\dot{x}(t_f)$ as a function of t_f for General Spin Spherical Inertia Impulsive Solution	150

List of Tables

3.1	Characterization of Different Docking Situations	40
3.2	Forms of Solution to Different Docking Situations	41
3.3	Possible Behavior for p_2	47
3.4	Summary of Solution to the Flat Spin Docking Problem	53
3.5	Nominal Values for Sensitivity Analysis of General Spin, Spherical Inertia Analytical Solution	61
3.6	$\gamma \leq 1$: Summary of Solution to the General Spin, General Inertia Problem	67
3.7	$\gamma > 1$: Summary of Solution to the General Spin, General Inertia Problem	68
4.1	GPOPS-2 Set up	78
4.2	Flat Spin Nominal Case	78
4.3	Flat Spin Nominal Case Result	79
4.4	Flat Spin Test Matrix	81
4.5	Flat Spin Test Result Matrix	82
4.6	General Spin Spherical Inertia, $\gamma \leq 1$ Nominal Case	84
4.7	General Spin Spherical Inertia, $\gamma \leq 1$ Nominal Case Result	84
4.8	General Spin Spherical Inertia, $\gamma > 1$ Nominal Case	87
4.9	General Spin Spherical Inertia, $\gamma > 1$ Nominal Case Result	87
4.10	AGA Simulation Testing Matrix	96
4.11	Test 1: Difference between AGA simulation and solution to general spin spherical inertia problem	99

4.12	Test 2: Difference between AGA simulation and solution to general spin spherical inertia problem	102
4.13	Test 3: Final time and final cost from AGA simulation vs. GPOPS	103
4.14	Test 4 and 5: Final time and final cost from AGA with and without plume protection	108
4.15	Effects of inertia ratio on solution of AGA vs. GPOPS. Note for all cases $R_0 = 10 [m]$, $R_f = 1 [m]$, $u_{sat} = 5 [m/s^2]$, $\boldsymbol{\omega} = [1, -5, 5] [^\circ/s]$	109
4.16	ENVISAT Example Mission Parameters	114
A.1	Required inputs for the transport trajectory optimizer	133
A.2	Required inputs for the transport trajectory optimizer	134
A.3	Test set up for transport trajectory optimizer	140
A.4	Test results for EML1 trajectory optimization	140
A.5	Iterations results for EML1 trajectory optimization	142
A.6	Test results for SEL2 trajectory optimization	143
A.7	Iterations results for SEL2 trajectory optimization	144
B.1	Nominal Values for Analysis of $\dot{x}(t_0)$, $\dot{x}(t_f)$ vs. t_f	150

Chapter 1

Introduction

1.1 Motivation and Background

The increase in orbital debris from low-Earth orbit (LEO) to geostationary orbit (GEO) presents one of the biggest dangers to future space operations. Currently, the European Space Agency estimates that over 900,000 objects¹ greater than 1 cm are orbiting the Earth [2], each of which could pose a threat to existing satellites by colliding with them. As the population of objects in space increases, the chances of collision increases, and subsequently the generation of more debris in what is known as the Kessler syndrome [3]. Additionally, many of such debris are known to have nonzero angular velocity [4] which presents challenges to current techniques used to rendezvous and capture objects. Currently, there is great interest in mitigating the amount of orbital debris by performing what is known as active debris removal (ADR). Accomplishing successful ADR not only requires technological advancements in robotics for capturing, but also advancements in algorithms for docking with a target that might be spinning.

In addition to ADR, the need to dock with tumbling objects presents itself in situations such as satellite servicing as well as in-space robotic assembly. Satellite servicing presents a promising future of reusing the resources already on orbit and will allow for

¹This number is considerably higher than that the 300,000 reported by Tommei just 12 years ago [1] and demonstrates the urgency of this issue

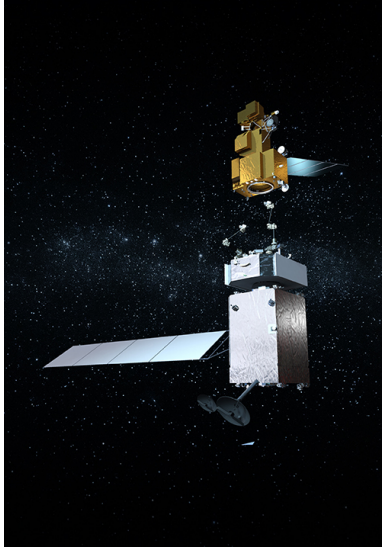


Figure 1-1: Artist departure of Restore-L spacecraft approaching for grabbing Landsat-7 as part of a mission servicing operation. Source: NASA [6]

extension of mission lifespan, reduction in overall mission cost, and a more environmentally friendly space activities by reducing the number of debris. Satellite servicing missions such Northrop Grumman's MEV-1 [5] as well as the planned Restore-L [6] (see Figure 1-1) demonstrate the compelling benefit of this new capability. Yet, both missions required the target spacecraft to have some form of attitude control, or near zero angular velocity which might restrict the type of satellites that could be serviced. Additionally, due to launch volume constraints, large space structures such as space telescopes will rely on autonomous robotic assembly [7]. However, due to cost and complexity reduction not all components being assembled may have a full attitude and position control system, and a situation might arise in which there is a need to capture a tumbling component.

It is clear that the need for safe and autonomous docking or capturing of uncooperative targets has never been more important than in today's space environment. This work seeks to contribute to research that addresses docking with tumbling objects.

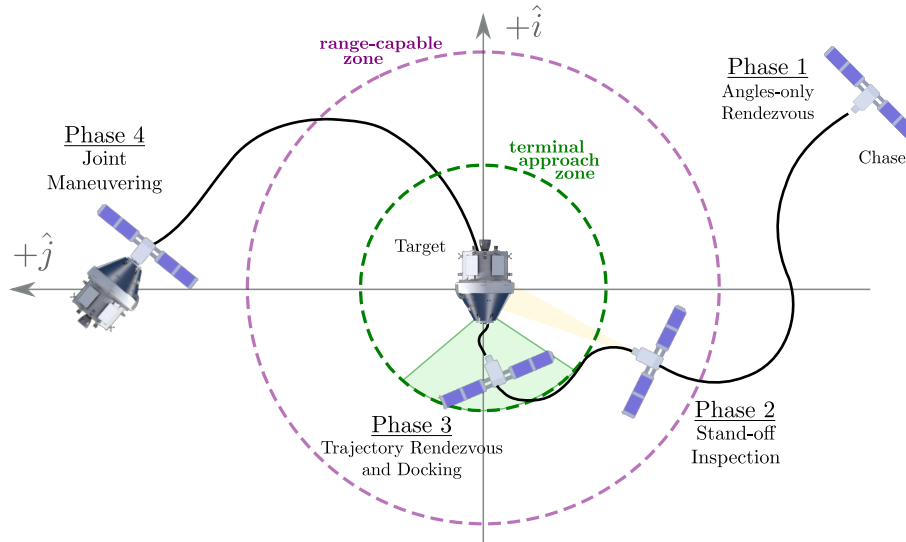


Figure 1-2: Depiction of the four stages of Docking. Source: Fig. 1, Terán Espinoza et al [8]

1.2 Overview of Methodology for Relative Operations for Autonomous Maneuvers

The main stages for a general docking procedure are shown in Figure (1-2) which was obtained from [8]. Phase 1 corresponds to angles-only rendezvous and is characterized by the servicing or capturing spacecraft (henceforth known as Chaser) being too far away from the target object (henceforth known as Target) for inspection. Phase 2 occurs once the Chaser reaches a range-capable zone in which either a LIDAR or camera system can be used to inspect the Target in what is known as stand-off inspection. After the Target is well characterized the next phase—trajectory rendezvous and docking—begins with the objective of capturing the Target. Phase 4 begins when the Chaser either captures or docks with the Target; both detumbling the Target and joint maneuvering occurs in this phase.

It should be noted that extensive research has been done in all the four stages of the general docking procedure [9]. Techniques for angles-only navigation have been researched and used in both manned and unmanned rendezvous and can include in-space as well as on ground assets [10]. Phase 2 has also been a subject of previous and ongoing research, particularly on the area of visual estimation of the Target's

relative dynamics [11, 12, 13]. Phase 3, which is of particular interest in this thesis, has also been previously studied; the current methodology is the subject of Chapter 2. In regard to Phase 4, work has been done in how to optimally detumble the Target once captured [14], as well as how to learn the mass properties of the Target [15]. For full treatment on the current technologies available as well as research gaps for all the phases, the reader is referred to Flores-Abad’s extensive review [16]. Additionally, Terán Espinoza et al gives a high level overview of of the inputs and outputs required for each phase for an end-to-end framework for proximity operations [8].

1.3 Research Objective

Due to the richness and complexity of the problem, only a small part of the problem of docking with tumbling objects can be addressed in this thesis. Thus, a research objective was developed to scope the problem and is summarized as follows:

- **To** develop guidance algorithms that allow for a Chaser spacecraft to capture or soft dock with a tumbling Target
- **By** utilizing techniques from optimal control, as well as the knowledge gained from exploring the natural dynamics of the system
- **While** minimizing the total fuel expenditure of the Chaser and ensuring convergence of the algorithm in a manner that would allow for real-time computation

The general principle that guided this research is that either more fuel or computationally efficient solutions can be obtained by gaining insight from the dynamics of the problem through analytical methods.

1.4 Thesis Organization

The thesis primarily focuses on the ability to generate guidance solutions for a Chaser soft docking with a Target with both low computational resources and fast solution speeds. Chapter 2 provides a background on current algorithms developed for tackling

Phase 3 and identifies the gap in research that the algorithm presented in this work seeks to address. Chapter 3 is the main methodology chapter that gives analytical solutions for specific docking scenarios as well as the overall docking algorithm to tackle Phase 3 of the general docking procedure. Chapter 4 provides the validation of the solution found in Chapter 3 as well as analysis of the results of the general algorithm for docking in a simulation environment. Chapter 5 then summarizes the contributions of the thesis as well as areas for future work. Additionally, Appendix B provides an impulsive version of the solutions presented in Chapter 3. Appendix A presents an algorithm for selecting launch trajectories that were developed to aid a NASA study on the feasibility of in-space assembly of a space telescope.

Chapter 2

Literature Review

This chapter focuses on literature review for docking with tumbling objects. Section 2.1 gives an overview of the history and classical methodologies for rendezvous and docking, namely the current in-flight methodology for tackling Phase 3 of the general docking procedure. Section 2.2 focuses on state-of-the-art algorithms for docking with tumbling objects.

2.1 History of Rendezvous and Docking

Rendezvous and docking (RVD) has been a necessary capability for space activities throughout history. The volume and mass constraint from launch vehicles translated into the need for a capability that allowed for multiple spacecraft to either assemble or combine in space, transfer fuel and equipment, or perform servicing functions. This first occurred in 1966 when Neil Armstrong and Dave Scott, aboard a Gemini capsule, manually rendezvoused and docked with an unmanned vehicle [17]. The U.S.S.R. subsequently achieved the first automatic rendezvous and docking in October of 1967. Since then, a myriad of rendezvous and docking maneuvers have been successfully accomplished, such as in the Apollo missions, the assembly of the International Space Station (ISS), the Hubble servicing missions, etc.

In general, safe trajectories used in rendezvous and docking procedures must satisfy the following constraints [9]:

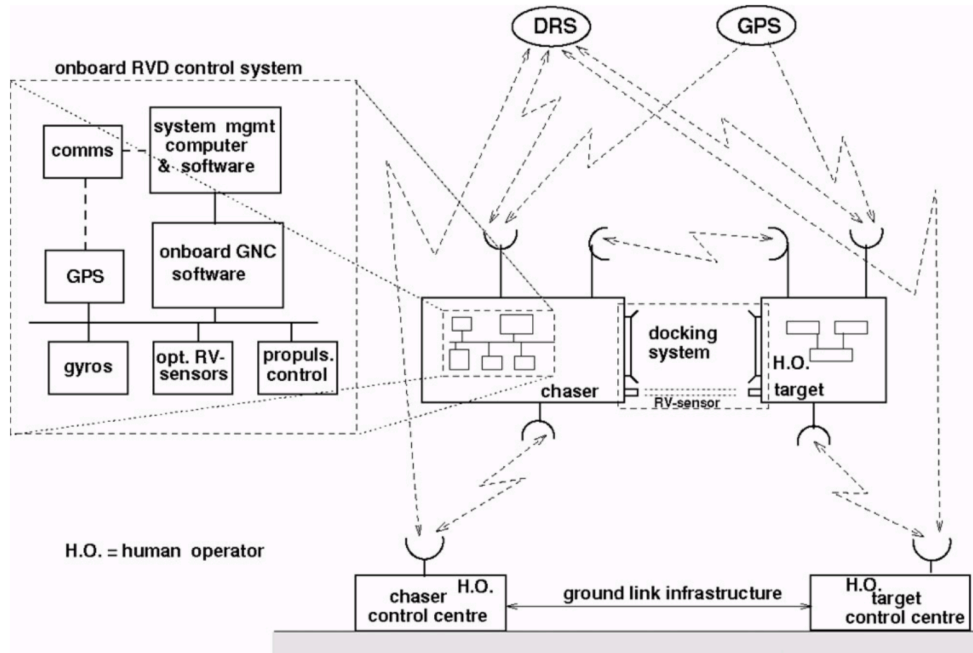


Figure 2-1: High level overview of a typical cooperative docking procedure [9]

- Be collision free throughout entire trajectory
- Have the capability to monitor the relative state vector at all times
- Be able to detect any faults through either on board systems or by a human operator
- Be capable of executing a collision avoidance trajectory at all times

It is therefore unsurprising that most of the RVD scenarios rely on both spacecraft cooperating with either a manned or unmanned component providing the support. For example, NASA's Space Shuttle relied on both internal control algorithms such as the Orbital Maneuvering System, and a human operator to perform the last steps before berthing [9]. In contrast, the Soyuz and Progress spacecraft rely mostly on closed loop systems between the Chaser and the Target to provide information of the relative states. A diagram showing the functionalities as well as processes needed for a typical rendezvous and docking procedure is shown in Figure (2-1): the Target is controlled by a human operator who has full attitude control, which enables the Chaser to perform the docking procedure without worry of the Target tumbling.

Typically, RDV algorithms rely on the so called r-bar and v-bar approaches. R-bar approaches are characterized by the Chaser approaching the Target through the radial orbital direction (or the axis from the center of mass of the Earth to the center of mass of the Target). In contrast, during a v-bar approach, the Chaser approaches the Target through the orbital velocity axis. The design of these trajectories are mostly system specific. For example, Yamanaka et al. [18] explains how the choice between using r-bar or v-bar approaches relies on either the need for a fast approach or one that allows for the Chaser to hold position and extend the duration of the trajectory.

Other systems, however, require more intricate trajectories compared to typical line r-bar or v-bar approaches. For example, the upcoming Restore-L spacecraft uses a complex RVD trajectory design to satisfy all the missions constraints [19]. Since it is a servicing mission, part of the trajectory requires the Restore-L (the Chaser) to perform close range inspections—typical for Phase 2 of the general docking procedure—before approaching Landsat-7 for capturing. This is required because Landsat-7 is not considered a cooperative rendezvous—Landsat-7 will be commanded to orient at a Sun-pointing attitude which is not conducive to an r-bar or v-bar approach. Despite this, the trajectory designed for Restore-L accounts for Landsat-7 attitude control system maintaining its own orientation which is not a valid assumption for orbital debris removal or servicing missions in which the Target is tumbling.

2.2 State-of-the-Art Algorithms for Docking with Tumbling Objects

There has been an abundance of research in various methodologies for achieving ADR of tumbling objects. The growing interest in this subject has been without a doubt impacted by the rise of planned satellite mega-constellations by commercial entities for the near future [20]. Various types of either direct or indirect methods have been sought for solving the control or guidance component of the problem. Direct methods usually rely on shooting methods in which the solver directly selects and

changes the control variable until a minimum is found using a nonlinear program. Indirect methods, on the other hand, operate on satisfying the optimality conditions by either using calculus of variations or the maximum principle, which transforms the optimal control problem into a boundary value problem [21].

This interest has not been limited to control and guidance techniques. For example, Zagaris conducted a reachability analysis to demonstrate the ability for a spacecraft to dock with a tumbling target within a certain period of time [22]. This analysis considered the set of feasible initial conditions that would allow for a Chaser to successfully complete a RVD maneuver. This analysis could prove as a starting point for mission designers as well as aid in the selection of potential debris targets.

Pseudospectral Optimal Control Methods

The use of pseudospectral methods for optimizing a guidance trajectory has been extensively studied. Pseudospectral optimal control methods rely on the use of basis functions to approximate the dynamics, solutions to differential equations, and constraints to solve an optimal control problem [23]. In essence, it converts an infinite variable problem to a finite variable problem by enforcing the dynamics and constraints at selected collocation points. Currently there exists various algorithms for solving pseudospectral methods, including commercially available tools such the gauss pseudospectral method [24]. This technique of solving optimization problems has been extensively tested including in-flight testing such as TRACE's reorientation trajectory and ISS's zero-propellant maneuver [23].

One of the main tools used in pseudospectral methods is the use of Pontryagin's Minimum Principle (PMP). The Pontryagin's Minimum Principle (PMP) is a powerful addition to the set of necessary conditions that an optimal trajectory must satisfy [25] and is related to the Hamilton-Jacobi-Bellman equation. In essence, the PMP converts the optimal control problem to a simple ODE equation with boundary conditions (or a two point boundary value problem).

For example, given a general optimization problem,

$$\begin{aligned}
 J^* = \min & \quad \left(h(t, \mathbf{x}) + \int_{t_0}^{t_f} g(t, \mathbf{u}, \mathbf{x}) dt \right) \\
 \text{Subject to} & \quad \dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{u}, \mathbf{x})
 \end{aligned} \tag{2.1}$$

where \mathbf{x} is the state of the system, \mathbf{u} is the control vector, $h(t, \mathbf{x})$ and $g(t, \mathbf{u}, \mathbf{x})$ are user defined scalar functions, and $\mathbf{f}(t, \mathbf{x}, \mathbf{u})$ represents the dynamics of the system, the typical necessary conditions for an optimal trajectory can be stated as follows,

$$\begin{aligned}
 \mathcal{H}(t_f, \mathbf{x}^*(t_f), \mathbf{u}^*(t_f)) + \frac{d}{dt}h(t, \mathbf{x}) &= 0 \\
 \dot{\mathbf{x}} &= \frac{\partial \mathcal{H}(t, \mathbf{x}, \mathbf{u}, \mathbf{p})}{\partial \mathbf{p}} \\
 \dot{\mathbf{p}} &= -\frac{\partial \mathcal{H}(t, \mathbf{x}, \mathbf{u}, \mathbf{p})}{\partial \mathbf{x}} \\
 \mathcal{H}(t, \mathbf{x}, \mathbf{u}, \mathbf{p}) &= \text{const.}
 \end{aligned} \tag{2.2}$$

where $\mathcal{H} = h(t, \mathbf{x}) + g(t, \mathbf{u}, \mathbf{x}) + \mathbf{p}^T \mathbf{f}(t, \mathbf{u}, \mathbf{x})$ is the Hamiltonian of the system, t_f is the final time of the trajectory, and \mathbf{p} is the costate of the dynamics function with dimension same as \mathbf{x} . Pointryagin's Minimum Principal then says that the choice of the best control policy \mathbf{u}^* is one that minimizes the Hamiltonian for all time,

$$\mathbf{u}^* = \arg \min_u (\mathcal{H}(t, \mathbf{x}, \mathbf{u}, \mathbf{p})) \tag{2.3}$$

This powerful statement is extremely useful in situations for which the actuation is saturated. For example, when there exists a saturation, u_{sat} , the choice of value for \mathbf{u} depends on the value of the costate for simple double integrators [25] as shown in Figure (2-2). When the corresponding costate has a magnitude less than one, the best control policy is to turn off the control. This leads to controllers of the form bang-off-bang for minimum fuel problems.

The use of pseudospectral methods as well as the exploitation of Pontryagin's Minimum Principal has been used to generate guidance solutions for the docking

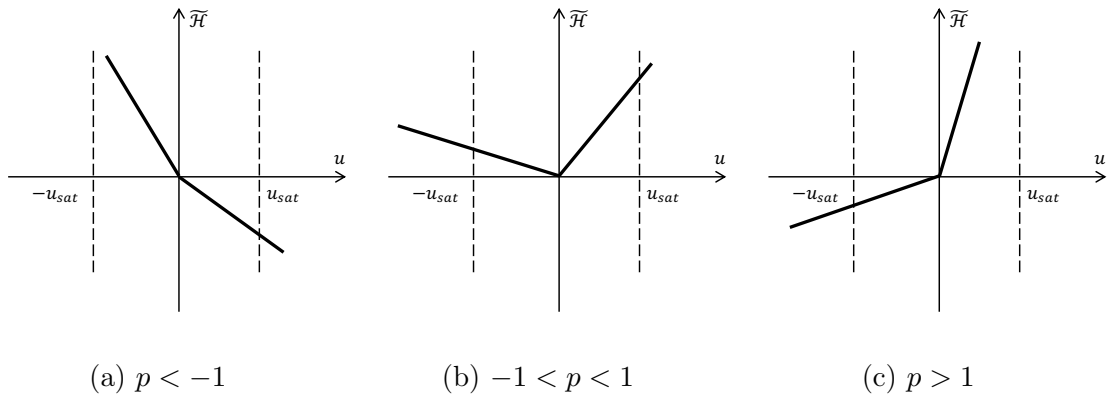


Figure 2-2: Notional example of Pontryagin’s Minimum Principal for a minimum fuel problem. The function $\tilde{\mathcal{H}}$, which corresponds to a simplified Hamiltonian for a double integrator, has different behavior for different values of the costate p .

with tumbling objects problem. Boyarko [26] implemented a 6 degree of freedom (DOF) optimization problem for docking with tumbling objects. The optimization problem not only included the relative position vector, but also considered the attitude of the Chaser as part of the optimization program. Additionally, PMP was utilized along with the classical necessary conditions for optimality given in Equation (2.2) to verify the optimality of solutions used by other algorithms [27]. Nevertheless, both approaches had to be performed offline due to their large computational cost, which is not ideal for an autonomous spacecraft with the current processing capabilities. Furthermore, the lack of an analytical form to the solution is also not desired because it means that an interpolation must be made onboard which could lead to a suboptimal result. Finally, Aghili proposed a methodology for capturing tumbling objects that included the Target’s state estimation as well as the guidance generation; Aghili’s method relied on the use of necessary conditions of calculus variation and showed the ability to capture a Target with a robotic arm [28].

Inverse Dynamics in Virtual Domain

Another powerful method for solving trajectory optimization problems is the inverse dynamics in virtual domain (IDVD) method. IDVD is a direct method in which the time component is decoupled from the state of the trajectory and optimized using a

nonlinear program (NLP) [29]. The general steps for this method, taken from [30], are as follows

1. Generate a reference function in virtual domain that is independent of time derivative constraints
2. Convert the reference trajectory back into time domain using speed factor
3. Employ inverse dynamics to calculate states and controls
4. Operate optimization routine considering boundary condition, constraints and performance index.

The reference functions can be selected to be polynomials which have good properties for NLP solvers.

Boyarko [26] utilizes this method to generate a 6DOF trajectory to capture a tumbling object, demonstrating an increase in computational speed compared to pseudospectral methods. Furthermore, Ventura et al. demonstrated the use of IDVD for real-time trajectory generation in a ground testbed [31]. This method is very promising for future ADR attempts as it shows the ability to generate a successful trajectory under parameter uncertainties such as the Target's inertia and angular velocity. The IDVD method is also extremely flexible for additions of new constraints while keeping the overall dimension of the variables small. The main drawbacks of IDVD are that the solution for the IDVD can be suboptimal and highly dependent on the degree of the solution used, and the need for a NLP solver which does not have a guaranteed solve time.

Model Predictive Control

Another method for both control and trajectory optimization is model predictive control (MPC). Model predictive control is a well-studied control method that got its start in chemical process industries. The typical architecture for a MPC is shown in Figure (2-3). Typically, the design of a MPC requires a plant model of the system, a cost function, constraints, and an optimizer. The output of the optimizer is then the

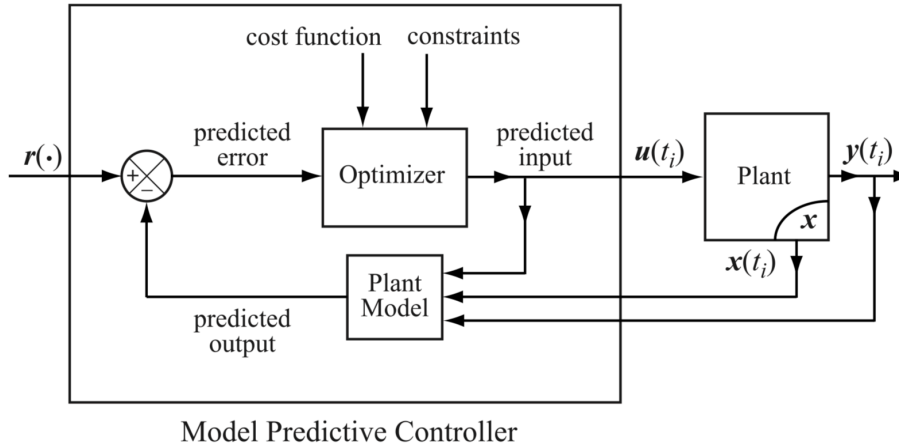


Figure 2-3: Basic structure for a model predictive controller (MPC). Source [32]

predicted input, which can be used as a limited time horizon controller. Since the main bottleneck for the controller is the optimization operation, typical approaches formulate the problem into a quadratic form; however, nonlinear optimization is also available.

Li et al. presented a MPC which was robust to the full nonlinear dynamics as well as relative position and velocity error [33]. The controller implemented velocity constraints, thruster saturations, and a form of collision avoidance all while formulating the problem as a quadratic program (QP) which ensured rapid solving time. Furthermore, Buckner and Lampariello proposed a tube-based MPC that provided robustness guarantees in the receding horizon framework. The controller provided collision avoidance under uncertainty of the Target's inertia and angular velocity. Furthermore, due to the tube-based approach, the MPC also provided robustness to unmodeled uncertainties with the level of robustness still left as a mission designer choice or engineering best practices. Finally, Park et al. developed a real-time nonlinear model predictive controller (NMPC) which had a multi-phase RVD procedure [34]. The NMPC was implemented in the Naval Postgraduate School POSEIDON testbed and showed real-time capability with their hardware.

Outside of these methods, other approaches have been considered for approaching the docking with tumbling objects problem. Sternberg developed an optimization frame-

work to develop relative position trajectory [35]. Sternberg, then proceeded to find methods for fitting the trajectories using a minimum set of parameters, and concluded that two exponential terms represented an excellent fit to the solution. The resultant fit of exponential function showed very small amount of error to the optimized trajectory and could allow for a look up table (LUT) approach. Additionally, Hettrick developed a method of forwards and backwards propagation of the Chaser's relative position that could be used for docking a tumbling Target [14]. The algorithm was implemented on the MIT Space Systems Lab's SPHERES simulation. Akhloumadi and Ivanov proposed a different method of tackling the problem by avoiding trajectory design; their approach involved solving a state-dependent Ricatti equation (SDRE) [36]. This approach benefited from the inherent robustness of SDRE, and, by providing an analytical solution to the Ricatti equation, ensured that their solution was real-time capable.

Chapter 3

Trajectory Optimization for Docking with Tumbling Objects

This chapter focuses on the proposed methodology and algorithm for tackling Phase 3 (Trajectory Rendezvous and Docking) of the general docking procedure shown in Figure (1-2). This Phase encompasses the trajectory optimization that the Chaser satellite must perform in order to ensure a soft docking with the Target. The proposed algorithm is derived in this chapter with an emphasis on the methodology used to derive the solution. First, a solution in which the target is spinning about a single axis normal to the docking axis, or the direction axis connecting the Target's center of mass with the desired docking location, is derived fully analytically. Then, the solution is extended to the case where the spin occurs about any axis but the Target is constrained to have a spherical inertia tensor. Finally, a general algorithm is presented for any combination of angular velocity and inertia tensor for the Target satellite.

3.1 Description of Trajectory Rendezvous and Docking Phase

Phase 3 begins once the Chaser has finalized the stand-off inspection (Phase 2) of the Target spacecraft. The goal for this phase is to initiate the final stages of the proximity operations to achieve a soft docking with the Target spacecraft. Soft docking again is defined as having zero relative velocity between the Target and Chaser when the two spacecraft are separated at the desired docking distance (e.g. when the two docking ports are touching). To fully formulate this problem, it is assumed that there exists a cost function and constraints that guide the generation of this trajectory. In this case, the cost function is assumed to be minimum fuel spent by the Chaser where the fuel is defined by the integral of the 1-norm of the thrust vector, $\int \|\mathbf{u}\|_1 dt$. Note that this trajectory only encompasses the translation aspect of the trajectory. The attitude “guidance” is not considered; however, an approach for obtaining such guidance solution will be presented.

Additionally, one of the main constraints that will be placed on the trajectory will be that during the trajectory, the Target and Chaser must be aligned along the “docking port axis,” which is the axis that is defined from the center of mass of the Target to the centroid of the docking location, henceforth known as the x-axis. The rationale behind this constraint is fourfold. First, by ensuring that Chaser is always pointing and positioned along the x-axis, the need to have collision checking from the rotating Target’s body is eliminated. For example, the Chaser does not have to take into account an antenna or solar array that is rotating *relative* to the Target body-fixed frame. Second, the equation of motion is simplified by forcing the relative position vector to always be of the form $\mathbf{r}(t) = [x(t), 0, 0]^T$ which reduces the number of nonzero terms for the problem formulation. Third, the implementation of the attitude guidance is reduced in complexity as the current desired quaternion is simply the inverse quaternion that defines the direction of the x-axis. Forth, the synchronicity always has the Target’s docking port in view of the Chaser for safety. It is this assumption that enables the possibility of obtaining a fully analytical solution

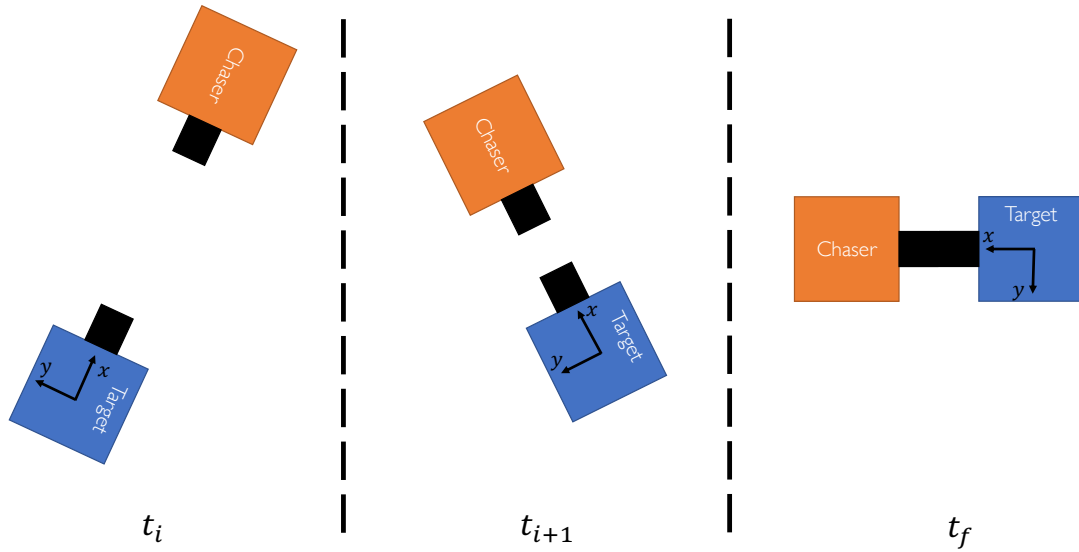


Figure 3-1: Representative example for Phase 3: trajectory rendezvous and docking. Note, x, y represents the body-fixed Target frame

for several cases.

Figure (3-1) shows an example of Phase 3 at different points in time. The Chaser begins at a radial offset from the Chaser and remains aligned along the docking port (shown as the black segment in the figure) for the remainder of the trajectory. As the trajectory continues, the Target and Chaser continue to move in the Target's x -axis direction reducing the distance between the two spacecraft. The phase ends when the Target and Chaser achieve the soft docking condition. Note that achieving soft docking condition indicates only that the relative velocity between the Target and the Chaser is zero at a desired radial offset; it does not imply that the Chaser and Target might have a docking attachment.

Once the final condition shown in right side of Figure (3-1) occurs two situations may happen. The Chaser and Target might dock using a traditional docking port which would then initiate Phase 4: joint maneuvering. Alternatively, the Chaser might use a robotic arm to grapple the Target and begin Phase 4. This is an important distinction as this allows for the trajectory optimization presented on this chapter to be used when different docking mechanisms are used. Furthermore, the problem

of grappling a tumbling object using a robotic arm is greatly simplified if there is no relative velocity between the two spacecraft—a situation created by the guidance solution—as it reduces the number of degrees of freedom of the system.

3.1.1 Necessary Information for Initiating Phase 3

Several pieces of information must be gathered during Phase 2 in order for Phase 3 to be allowed to begin. Namely,

- Final docking destination such as a docking port, servicing ring, or handle
- Estimated relative position between the Chaser and Target
- Estimated relative velocity between the Chaser and Target
- Estimated angular velocity of the Target
- Estimated principal inertia ratios of the Target

It is assumed that the location for docking is along one of the Target's principal axes. The benefit of this constraint is that the dynamics of the problem statement will be simplified substantially, namely the nonzero elements of the Newton-Euler equation. This constraint yields a diagonal inertia tensor, therefore, the expression for the rate of change of angular velocity is simplified considerably. Furthermore, this allows for a possibility of easily aligning the angular momentum vector with the angular velocity vector which will result in torque free motion once the Chaser and the Target begin joint maneuvers.

The relative position and velocity between the Chaser and Target might be obtained through visual sensors as part of the closed-loop controller for the Chaser. This is consistent with the idea that the algorithm will develop an open loop guidance trajectory that a closed loop controller will regulate. The estimate of the angular velocity and the principal inertia ratios of the Target will be used to recompute the guidance trajectory once a better estimate is obtained.

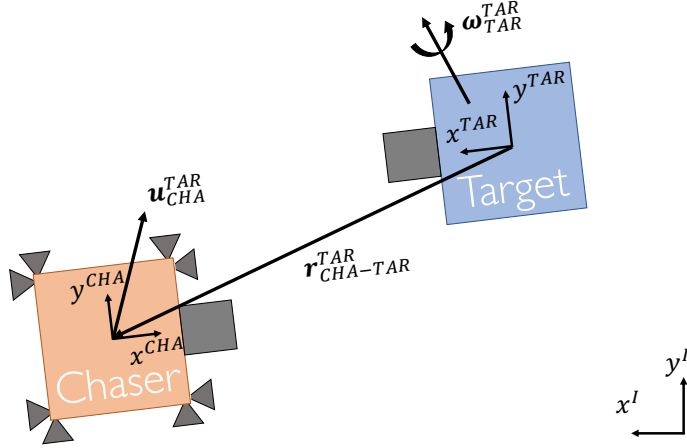


Figure 3-2: Docking problem setup. Three frames are shown: an inertial frame, a Target body fixed frame, and a Chaser body fixed frame.

3.1.2 Problem Formulation for Docking with Tumbling Objects

Figure (3-2) shows the overall system that will be discussed in this chapter. Three frames are shown in the figure: an inertial frame $\{x^I, y^I, z^I\}$, a Target body-fixed frame $\{x^{TAR}, y^{TAR}, z^{TAR}\}$, and a Chaser body-fixed frame $\{x^{CHA}, y^{CHA}, z^{CHA}\}$. The distance between the Target and Chaser is labeled as $\mathbf{r}_{CHA-TAR}$, and the angular velocity of the Target is labeled as $\boldsymbol{\omega}_{TAR}$, finally the control vector from the Chaser is labeled as \mathbf{u}_{TAR} . Note that the superscripts for all three vectors represent the coordinate system in which they are defined.

Given the constraints and assumptions stated in the previous section, the problem statement—written as an optimal control problem—can then be formulated as follows,¹

¹For clarity in the equation, the superscripts and subscripts for the vectors will be removed. For example, the equation will reference the force acted upon the Chaser as \mathbf{u} instead of the more explicit \mathbf{u}^{TAR}

$$\begin{aligned}
\min_{\mathbf{u}} \quad & \int_{t_0}^{t_f} \|\mathbf{u}\|_1 dt && \text{Cost Function} \\
\text{s.t.} \quad & \mathbf{u} = \ddot{\mathbf{r}} + 2\boldsymbol{\omega} \times \dot{\mathbf{r}} + \dot{\boldsymbol{\omega}} \times \mathbf{r} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}) && \text{Rel. Translational Dynamics} \\
& \dot{\boldsymbol{\omega}} = I^{-1}(-\boldsymbol{\omega} \times (I\boldsymbol{\omega})) && \text{Newton-Euler Dynamics} \\
& x(t_0) = R_0, \dot{x}(t_0) = 0 && \text{Begin radially aligned} \\
& x(t_f) = R_f, \dot{x}(t_f) = 0 && \text{Soft dock conditions} \\
& y = z = 0, \dot{y} = \dot{z} = 0 \quad \forall t \in [t_0, t_f] && \text{Pointing along x-axis} \\
& 0 \leq u_x^+ \leq u_{max}, 0 \leq u_x^- \leq u_{max} && \text{Thruster saturation x} \\
& 0 \leq u_y^+ \leq u_{max}, 0 \leq u_y^- \leq u_{max} && \text{Thruster saturation in y} \\
& 0 \leq u_z^+ \leq u_{max}, 0 \leq u_z^- \leq u_{max} && \text{Thruster saturation in z}
\end{aligned} \tag{3.1}$$

where \mathbf{u} represents the thruster force acted on the Chaser in the Target body-fixed frame. Also, $\mathbf{r} = [x, y, z]^T$, $\dot{\mathbf{r}}$, and $\ddot{\mathbf{r}}$ represent the position, velocity, and acceleration of the Chaser relative to the Target in the Target body-fixed frame, respectively. The angular velocity of the Target relative to the inertial frame and expressed on the Target body-fixed frame is given by $\boldsymbol{\omega}$. Finally, I is the inertia tensor of the Target expressed in the Target body-fixed frame.

The problem statement assumes that the Chaser begins radially aligned at time $t_0 = 0$. This states that at the end of Phase 2 the Chaser must perform a maneuver to align itself along the docking location. The rationale behind not including this segment in the cost function or as part of the minimization problem is that during Phase 2 the Chaser will perform an active inspection maneuver. During this maneuver, the Chaser will select a feasible docking location at which point the Chaser can “align” itself so that Phase 3 may begin. Alternatively, the cost for aligning the Chaser from an inertial frame to a body-fixed Target frame is simply $\Delta v = R_0(\omega_y^2 + \omega_z^2)$. Finally, it should be noted that thruster saturation is included as part of the problem statement.

3.2 Characterization of Docking Problem

Solving Eq. (3.1) where all the terms in the constraint equations are non-zero requires numerical methods as there does not exist a closed form solution for the general Newton-Euler Equation [37]. Additionally, the relative Translational Dynamics—which depend on the solution to the Newton-Euler Equation—is extremely nonlinear and thus represents a system that cannot be integrated analytically. Therefore, to better understand the dynamics as well as to obtain results for simple cases, the overall solution to Eq. (3.1) was divided into different cases.

Although there are many ways to parameterize the different types of docking situations, the approach taken here is the same as that provided by David Sternberg in his thesis [35]: spin axis and inertia ratios of the Target. The reasoning behind this is as follows, The Newton-Euler equations of motion are given by:

$$\frac{d}{dt} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} \frac{I_y - I_z}{I_x} \omega_y \omega_z + \frac{\tau_x}{I_x} \\ \frac{I_z - I_x}{I_y} \omega_z \omega_x + \frac{\tau_y}{I_y} \\ \frac{I_x - I_y}{I_z} \omega_x \omega_y + \frac{\tau_z}{I_z} \end{bmatrix} \quad (3.2)$$

where $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^T$ represents the angular velocity vector relative to the inertial frame in the principal inertia frame. $I = \text{diag}([I_1, I_2, I_3])$ represents the inertia tensor. Finally, $\boldsymbol{\tau} = [\tau_x, \tau_y, \tau_z]^T = \mathbf{0}$ are the external torques exerted on the Target also in the principle inertia frame. There are many ways to simplify these equations based on the geometry of the Target, however, the two distinctions that this work will employ are those of a **spherical** and **general** inertia. If the Target has a spherical inertia (such as perfect cube, sphere, or object with symmetry about any axes) then the Equation shown in (3.2) simply states that the angular velocity is not changing, $\dot{\boldsymbol{\omega}} = \mathbf{0}$.

The second parameter is the angular velocity vector. Parametrizing the direction of this vector can simplify the relative translational dynamics. For example, the most general expression for the right hand side of the relative translational dynamics in a rotating frame equation is

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} + 2 \begin{bmatrix} \omega_y \dot{z} - \omega_z \dot{y} \\ \omega_z \dot{x} - \omega_x \dot{z} \\ \omega_x \dot{y} - \omega_y \dot{x} \end{bmatrix} + \begin{bmatrix} \dot{\omega}_y z - \dot{\omega}_z y \\ \dot{\omega}_z x - \dot{\omega}_x z \\ \dot{\omega}_x y - \dot{\omega}_y x \end{bmatrix} + \begin{bmatrix} -(\omega_y^2 + \omega_z^2)x + \omega_x \omega_y y + \omega_x \omega_y z \\ \omega_x \omega_y x - (\omega_x^2 + \omega_z^2)y + \omega_y \omega_z z \\ \omega_x \omega_z x + \omega_z \omega_y y - (\omega_x^2 + \omega_y^2)z \end{bmatrix} \quad (3.3)$$


The assumption that the Chaser is always approaching the Target along the x-axis simplifies the above equation to

$$\begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \begin{bmatrix} \ddot{x} \\ 0 \\ 0 \end{bmatrix} + 2 \begin{bmatrix} 0 \\ \omega_z \dot{x} \\ -\omega_y \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ \dot{\omega}_z x \\ -\dot{\omega}_y x \end{bmatrix} + \begin{bmatrix} -(\omega_y^2 + \omega_z^2)x \\ \omega_x \omega_y x \\ \omega_x \omega_z x \end{bmatrix} \quad (3.4)$$

Therefore, a way to distinguish between docking cases is whether the spin is a **flat spin** (the angular velocity is a single axis either in y or z) or a **general spin** (which corresponds to a spin about any direction).

Table 3.1: Characterization of Different Docking Situations

	Spherical Inertia	General Inertia
Flat Spin		
General Spin		



With these simplifications it is possible to characterize any type of spin by any combination of the two parameters as shown in Table 3.1. The upper left case represents the simplest case in which the Target is a sphere and rotating about an axis normal to the docking. The lower right most case represents a docking scenario in which the Target can have any inertia ratios that satisfies the triangle inequality as well as any angular velocity vector, and it is the most complex case. The methodology for solving the general problem (lower right case) was to follow the complexity arrow shown in Table 3.1 by solving the simplest problem first and then utilize insight gained from it to tackle the more general problem.

Furthermore, the solution form for each of the problems in Table 3.1 can be

summarized in Table 3.2. The computational complexity of a fully analytical solution is negligible. A Quasi-Analytical solution represents a situation in which whether the solution will be either analytical or numerical depends on whether a certain condition is met. Finally, an Analytical Guidance Algorithm (AGA) represents a full trajectory that has to be computed on a receding horizon basis (similar to a Model Predictive Controller); however, the solution to the AGA is analytical and the computational complexity is significantly reduced as opposed to traditional problem formulations.

Table 3.2: Forms of Solution to Different Docking Situations

	Spherical Inertia	General Inertia
Flat Spin	Fully Analytical	Fully Analytical
General Spin	Quasi-Analytical	AGA

3.3 Analytical Solution for the Flat Spin Docking Case

The first case that was solved for the general docking problem is that of a Target with a flat spin and a spherical inertia.² To begin, the simplified problem statement is given as follows,

$$\begin{aligned}
\min \quad & \int_{t_0}^{t_f} (|u_x| + |2\omega\dot{x}|) \, dt \quad \text{Cost Function} \\
\text{Subject to} \quad & \ddot{x} = \omega^2 x + u_x \\
& x(t_0) = R_0, \dot{x}(t_0) = 0 \quad \text{Begin radially aligned} \\
& x(t_f) = R_f, \dot{x}(t_f) = 0 \quad \text{Soft dock} \\
& -u_{sat} \leq u_x \leq u_{sat} \quad \text{Thruster saturation in x direction} \\
& t_f \text{ free}
\end{aligned} \tag{3.5}$$

²The spherical inertia specification is not necessary as the solution for the flat spin with general inertia is the same as the solution with a spherical inertia tensor.

where the cost function integrand $|u_x| + |2\omega\dot{x}|$ was modified so that the Coriolis acceleration due to radial alignment at all times is included in the cost function instead of defining a new thrust variable u_y [m/s^2]. This aids in simplifying the dynamics by not including an equation for \ddot{y} and having a path constraint where $y, \dot{y} = 0 \forall t$.

3.3.1 Necessary Conditions for Optimality

Noting that there are two states and, thus, the need for two costate variables the Hamiltonian of the problem is written as follows:

$$\mathcal{H} = |u_x| + |2\omega\dot{x}| + p_1\dot{x} + p_2(\omega^2x + u_x) \quad (3.6)$$

Following the optimality conditions for a minimization problem shown in Kirk [25], the equations of motion are written in state space form:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \omega^2 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_x \quad (3.7)$$

The first optimal control necessary condition tells us the dynamics or equation (3.7). The next condition related to the dynamics of the costate is

$$\begin{bmatrix} \dot{p}_1 \\ \dot{p}_2 \end{bmatrix} = \begin{bmatrix} 0 & -\omega^2 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} \pm \begin{bmatrix} 0 \\ 2\omega \end{bmatrix} \quad (3.8)$$

where the $\pm 2\omega$ depends on whether $2\omega\dot{x} < 0$ or $2\omega\dot{x} > 0$, respectively. Since the objective for docking is to go from a higher radius to a lower radius, and the dynamics represent an unstable system that will push the Chaser naturally outwards, there is an intuitive sense that the least expensive trajectory decreases monotonically. Thus, the sign of $\pm 2\omega\dot{x}$ simply depends on $\text{sign}(\omega)$. Assuming that $\dot{x} < 0$ for all time the solution to the homogeneous solution to Eq. (3.8) is given as

$$\begin{aligned}
p_1(t) &= -c_1\omega e^{\omega t} + c_2\omega e^{-\omega t} \\
p_2(t) &= c_1e^{\omega t} + c_2e^{-\omega t}
\end{aligned} \tag{3.9}$$

To solve the non-homogeneous solution, the method of variation of parameters is utilized by assuming that the coefficients in the homogeneous solution are a function of time (i.e. $c_1(t)$ and $c_2(t)$). Thus, plugging into the solution in Eq. (3.9) with the assumption that c_1 and c_2 are functions of time into the costate equations in (3.8) leads to

$$\begin{aligned}
-\dot{c}_1\omega e^{\omega t} + \dot{c}_2\omega e^{-\omega t} - c_1\omega^2 e^{\omega t} - c_2\omega^2 e^{-\omega t} &= -c_1\omega^2 e^{\omega t} - c_2\omega^2 e^{-\omega t} \\
\dot{c}_1 e^{\omega t} + \dot{c}_2 e^{-\omega t} + c_1\omega e^{\omega t} - c_2\omega e^{-\omega t} &= c_1\omega e^{\omega t} - c_2\omega e^{-\omega t} + 2\omega
\end{aligned} \tag{3.10}$$

The above equations can be manipulated to obtain equations for \dot{c}_1 and \dot{c}_2 which leads to

$$\begin{aligned}
\dot{c}_1 = \omega e^{\omega t} &\implies c_1(t) = -e^{-\omega t} + \tilde{c}_1 \\
\dot{c}_2 = \omega e^{-\omega t} &\implies c_2(t) = e^{\omega t} + \tilde{c}_2
\end{aligned} \tag{3.11}$$

where the $\tilde{c}_\#$ represents a constant value. Thus, the solution to the non-homogeneous equation for the costate equations can be obtained by plugging Eq. (3.11) into (3.8), obtaining the full costate equations for the case in which $\dot{x} < 0$ as follows:

$$\begin{aligned}
p_1(t) &= \text{sign}(\omega)2\omega - \tilde{c}_1\omega e^{\omega t} + \tilde{c}_2\omega e^{-\omega t} \\
p_2(t) &= \tilde{c}_1 e^{\omega t} + \tilde{c}_2 e^{-\omega t}
\end{aligned} \tag{3.12}$$

Once the equations for the costate for the soft-docking problem are known, it is possible to continue with the rest of the necessary conditions for an optimal trajectory. In fact, knowledge of the structure of Equation (3.12) will allow us to eliminate several candidate optimal solutions and reduce the search space for the optimal solution.

Continuing on, the next necessary condition for optimality according to Pontryagin's Minimum Principle (PMP) states that an optimal control policy is obtained by minimizing the Hamiltonian in (3.6). To do this the partial of the Hamiltonian with respect to u is computed, which yields

$$\frac{\partial \mathcal{H}}{\partial u} = 0 = \pm 1 + p_2 \quad (3.13)$$

When the control variable is not present on the control policy condition as shown above, the optimal control approach per PMP indicates that the policy must be as follows:

$$u_x^* = \begin{cases} -u_{sat} & p_2 \geq 1 \\ 0 & -1 < p_2 < 1 \\ u_{sat} & p_2 \leq -1 \end{cases} \quad (3.14)$$

Obtaining a general form for the costate equation as well as the control policy is extremely useful. From here different approaches can be taken for obtaining a set of trajectories that satisfy all the conditions. For example, Boyarko utilizes the costate equation form to set up a two point boundary value problem (BVP) to obtain the optimal trajectory [26]. However, it is possible to go beyond this and fully solve the problem by using the transversality condition to obtain the terminal value for the costate,

$$|u_x(t_f)| + |2\omega\dot{x}(t_f)| + p_1(t_f)\dot{x}(t_f) + p_2(t_f)(\omega^2 x(t_f) + u_x(t_f)) = 0 \quad (3.15)$$

Given that the problem statement has a final radial velocity equal to zero, $\dot{x}(t_f) = 0$ the final value for the costate will be $p_2(t_f) = \frac{-|u_x(t_f)|}{\omega^2 R_f + u_x(t_f)}$.

Per Equation (3.14) only three possible scenarios for values of u_x with the constraints on the range of p_2 satisfy the necessary condition for PMP. Therefore, it is possible to test each of these control values in order to determine what will be the final value for the controller when docking is achieved.

Case: $\mathbf{u}_x^*(\mathbf{t}_f) = -\mathbf{u}_{sat}$

The condition of $u_x(t_f) = -u_{sat}$ has the second costate at the final time equal to

$$p_2(t_f) = \frac{u_{sat}}{u_{sat} - \omega^2 R_f} > 1 \quad (3.16)$$

This value is admissible as it is in accordance with the conditions for PMP optimality in (3.14); this means that having a final control value $u = -u_{sat}$ is a possible value for the second costate.

Case: $\mathbf{u}_x^*(t_f) = \mathbf{0}$

Selecting the final control as zero would lead to a final costate of also zero.

$$p_2(t_f) = 0 \tag{3.17}$$

The case in which the costate, p_2 , is equal to zero is also admissible as it is within the range of $-1 < p_2 < 1$. In fact, it will turn out that this will be the true condition for the actual optimal condition that assures the soft dock and will be expanded upon later in this document.

Case: $\mathbf{u}_x^*(t_f) = \mathbf{u}_{sat}$

Finally, the last condition of selecting the final control value as the maximum allowable control, u_{sat} yields the costate as

$$p_2(t_f) = \frac{-u_{sat}}{u_{sat} + \omega^2 R_f} \not\leq -1 \tag{3.18}$$

This costate violates Equation (3.14) and is therefore not a possible final costate. Therefore, the only allowable final costates are $u_x(t_f) = 0$ or $-u_{sat}$.

It is possible to obtain an additional boundary point on the values for the costate by noting that the Hamiltonian has to be constant for all time per the PMP conditions *and* that at the final time the Hamiltonian has to be zero per the transversality condition. This means that at the initial time, the Hamiltonian has to be equal to zero,

$$|u_x(t_0)| + |2\omega\dot{x}(t_0)| + p_1(t_0)\dot{x}(t_0) + p_2(t_0) (\omega^2 x(t_0) + u_x(t_0)) = 0 \tag{3.19}$$

The same approach as before can be applied to conclude that the only possible

values for the initial control is $u_x(t_0) = 0$ or $-u_{sat}$ with the costate values of $p_2(t_0) = 0$ or $u_{sat}/(u_{sat} - \omega^2 R_0)$, respectively. In summary, this analysis has shown the various possible values for the costate vector at the initial and final boundary conditions. This is the power of the PMP condition: it modifies the problem from a general (infinite dimension) *optimal control problem* to a finite dimension *two point boundary problem*.

3.3.2 Possible Solutions to Flat Spin Problem

The analysis on the previous section yielded the following information regarding the optimal solution to the Flat Spin Problem, which again refers to the case in which the angular velocity of the Target (either with spherical or general inertia) is perpendicular to the docking axis:

- The general equations of motion:
$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \omega^2 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_x$$
- The general solution to the costate:
$$\begin{bmatrix} p_1(t) \\ p_2(t) \end{bmatrix} = \begin{bmatrix} 2\omega - \tilde{c}_1\omega e^{\omega t} + \tilde{c}_2\omega e^{-\omega t} \\ \tilde{c}_1 e^{\omega t} + \tilde{c}_2 e^{-\omega t} \end{bmatrix}$$
- The functional form of the optimal control policy:
$$u_x^* = \begin{cases} -u_{sat} & p > 1 \\ 0 & -1 < p < 1 \\ u_{sat} & p < -1 \end{cases}$$
- The value of the Hamiltonian for all time: $\mathcal{H} = |u_x| + |2\omega\dot{x}| + p_1\dot{x} + p_2(\omega^2x + u_x) = 0 \forall t \in [t_0, t_f]$
- The initial possible values for the costate and control: $u_x(t_0) = -u_{sat}$, $p_2(t_0) = \frac{u_{sat}}{u_{sat} - \omega R_0^2}$ or $u_x(t_0) = 0$, $p_2(t_0) = 0$
- The final possible values for the costate and control: $u_x(t_f) = -u_{sat}$, $p_2(t_f) = \frac{u_{sat}}{u_{sat} - \omega R_f^2}$ or $u_x(t_f) = 0$, $p_2(t_f) = 0$

where from now on, without loss of generality, $t_0 = 0$ for simplicity.

Table 3.3: Possible Behavior for p_2

Case ID	$\mathbf{p}_2(\mathbf{t}_0)$	$\mathbf{p}_2(\mathbf{t}_f)$
Case 1	0	0
Case 2	0	$u_{sat}/(u_{sat} - \omega^2 R_f)$
Case 3	$u_{sat}/(u_{sat} - \omega^2 R_0)$	0
Case 4	$u_{sat}/(u_{sat} - \omega^2 R_0)$	$u_{sat}/(u_{sat} - \omega^2 R_f)$

Based on the information from the last two bullet points, four possible cases exist for the costate and are summarized in Table 3.3.

The possible boundary values for the costate as well as their closed form solution in Equation (3.12) will force the trajectory for the costates to have the form shown in Figure (3-3).

In Figure (3-3) the shaded region below $p_2 = 1$ indicates the switching condition for optimal control. It should be noted that despite choosing arbitrary values for R_0, ω, R_f and t_f the overall shape of the costates will remain the same. In fact there are only three possible types of controllers based on Fig. (3-3) either a controller with no switches (Case 1), one switch (Case 2 and 3), or two switches (Case 4). Due to this, an analysis must be conducted into the nature of each of the four cases to determine which one corresponds to the optimal solution.

Case 1: $p_2(t_0) = 0, p_2(t_f) = 0$, No Control Switch

The first case corresponds to a time history in which control is inactive for all time, $u_x(t) = 0 \forall t \in [t_0, t_f]$. This assumes that by not applying control the natural dynamics will move the system from its initial point $x(t_0) = R_0$ and $\dot{x}(t_0) = 0$ to the final state $x(t_f) = R_f$ and $\dot{x}(t_f) = 0$. This clearly cannot happen, as the resultant dynamics would lead to

$$\ddot{x} = \omega^2 x$$

Given that $\omega^2 \geq 0$, the equation corresponds to an unstable system in which the time history of x will exponentially increase. Furthermore, at time zero, $\ddot{x} > 0$ which would indicate that $\dot{x}(t_0 + \epsilon) > 0$. Since the optimal trajectory must be monotonically decreasing—otherwise the amount of control effort to bring the radial position from a

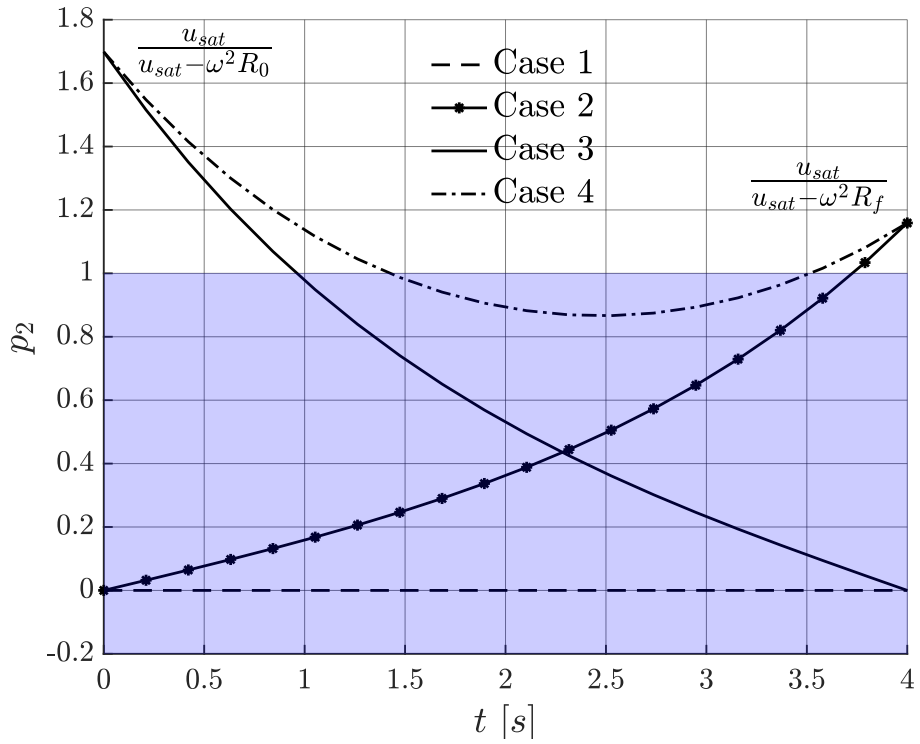


Figure 3-3: Possible forms for p_2 with $R_0 = 15$, $R_f = 5$, $\omega = 30^\circ/s$, $t_f = 4$ s, $u_{sat} = 10$ m/s²

higher radius will be significantly higher than from a lower radius—so Case 1 cannot correspond to the optimal trajectory.

Case 2: $p_2(t_0) = 0$, $p_2(t_f) = u_{sat}/(u_{sat} - \omega^2 R_f)$, **Single Control Switch**

Case 2 corresponds to the case in which the thruster starts turned off and at some later time it turns on until the end of the trajectory: $u(t) = 0, \forall t \in [t_0, t_1], u(t) = -u_{sat} \forall t \in [t_1, t_f]$. This corresponds to an off-bang controller which is a common form for minimum fuel problems. The main issue with this controller with a costate time history shown in Figure (3-3) is that at $t = t_0 + \epsilon$ the rate of derivative of \dot{x} is greater than zero. This case would see a trajectory go from an initial lower radial distance to a higher radial distance which is not the case for the general docking problem.

Case 3: $p_2(t_0) = u_{sat}/(u_{sat} - \omega^2 R_0)$, $p_2(t_f) = 0$, **Single Control Switch**

Case 3 corresponds to a bang-off controller where the Chaser starts firing at full control saturation $u_x = -u_{sat}$ for a specified time period until the costate p_2 attains a value of 1. At this point, the thruster will turn off and will coast until the final position and velocity at which time soft dock is achieved. A quick check of the dynamics at time $t = t_0 = 0$ yields

$$\ddot{x} = \omega^2 R_0 - u_{sat} < 0$$

where it is assumed that the Chaser has enough control authority such that $\omega^2 R_0$ is smaller than u_{sat} . This means that at $t = t_0 + \epsilon$ the relative velocity between the Chaser and Target will be less than zero which is consistent with our problem statement. Exploring this case further, it is assumed that from time t_0 to t_1 the controller operates at saturation level $-u_{sat}$. Then, from t_1 to t_f the thruster will be turned off. Finally, t_1 corresponds to the critical time at which the costate p_2 equals to one.

Continuing with this assumption, the equations of motion between these two seg-

ments are given by

$$\begin{aligned} \ddot{x} = \omega^2 x - u_{sat} &\implies x_1(t) = c_{11}e^{\omega t} + c_{12}e^{-\omega t} + \frac{u_{sat}}{\omega^2} & t \in [0, t_1] \\ \ddot{x} = \omega^2 x &\implies x_2(t) = c_{21}e^{\omega t} + c_{22}e^{-\omega t} & t \in [t_1, t_f] \end{aligned} \quad (3.20)$$

where c_{11} , c_{12} , c_{21} and c_{22} represent constant coefficients for the first and second segment of the solution, respectively. To obtain the values for the coefficients, the boundary conditions for the trajectory, namely that the Chaser begins radially aligned and ends in the soft dock condition, were applied.

Applying the boundary conditions yields the two segments of the trajectory as follows:

$$\begin{aligned} x_1(t) &= \frac{R_0 - \frac{u_{sat}}{\omega^2}}{2} (e^{\omega t} + e^{-\omega t}) + \frac{u_{sat}}{\omega^2} & t \in [0, t_1] \\ x_2(t) &= \frac{R_f}{2} (e^{\omega(t-t_f)} + e^{-\omega(t-t_f)}) & t \in [t_1, t_f] \end{aligned} \quad (3.21)$$

where t_1 and t_f are the only unknowns. The next step is to check if this solution is valid to set up the equations such that at the switch time, t_1 , both equations are continuous and differentiable. The reason is that the discontinuity caused by switching the controller from saturated to off happens on the acceleration term, so the velocity and position term must be continuous. This yields the following relationship

$$\begin{aligned} \frac{R_0 - \frac{u_{sat}}{\omega^2}}{2} (e^{\omega t_1} + e^{-\omega t_1}) + \frac{u_{sat}}{\omega^2} &= \frac{R_f}{2} (e^{\omega(t_1-t_f)} + e^{-\omega(t_1-t_f)}) \\ \frac{R_0 - \frac{u_{sat}}{\omega^2}}{2} (\omega e^{\omega t_1} - \omega e^{-\omega t_1}) &= \frac{R_f}{2} (\omega e^{\omega(t_1-t_f)} - \omega e^{-\omega(t_1-t_f)}) \end{aligned} \quad (3.22)$$

To obtain a solution for t_1 and t_f , let $a = \frac{R_0 - \frac{u_{sat}}{\omega^2}}{2}$, $b = \frac{u_{sat}}{\omega^2}$, $c = \frac{R_f}{2}$. Also, have $Y = e^{\omega t_1}$, $X = e^{\omega(t_1-t_f)}$. Thus, Eq. 3.22 has the form:

$$\begin{aligned} a \left(Y + \frac{1}{Y} \right) + b &= c (X + 1/X) \\ a \left(Y - \frac{1}{Y} \right) &= c (X - 1/X) \end{aligned}$$

The above expression admits two solutions:

$$\begin{aligned}
X_{1,2} &= \frac{-4a^2 + b^2 + 4c^2}{4bc} \pm \sqrt{\left(\frac{-4a^2 + b^2 + 4c^2}{4bc}\right)^2 - 1} \\
Y_{1,2} &= \frac{-4a^2 - b^2 + 4c^2}{4ab} \pm \sqrt{\left(\frac{-4a^2 + b^2 + 4c^2}{4ab}\right)^2 - \frac{c^2}{a^2}}
\end{aligned} \tag{3.23}$$

where in the \pm symbol, the positive value corresponds to the first, and the negative to the second solution. To differentiate between the two note that $X = e^{\omega(t_1 - t_f)} < 1$ if $\omega > 0$. Additionally, the form of the solution for X is $X = A \pm \sqrt{A^2 - 1}$. This represents a right triangle with sides A , 1, and $\sqrt{A^2 - 1}$. Thus, $X = A - \sqrt{A^2 - 1} < 1$ by the triangle inequality and X_2, Y_2 represents our solution. **Note** that the choice of the solution depends on the sign of ω . However, this correction can be done by just simply multiplying t_1 by the sign of ω .

Therefore, the fully analytical expression for t_1 and t_f is:

$$\begin{aligned}
t_1 &= \frac{1}{\omega} \log(Y) \\
t_f &= t_1 - \frac{1}{\omega} \log(X)
\end{aligned} \tag{3.24}$$

where X, Y are the second solution pair in Eq. 3.23, and \log represents the natural logarithm.

Case 4: $p_2(t_0) = u_{sat}/(u_{sat} - \omega^2 R_0)$, $p_2(t_f) = u_{sat}/(u_{sat} - \omega^2 R_f)$, **Two Control Switches**

Case 4 corresponds to a bang-off-bang controller where the Chaser starts firing at full control saturation $u_x = -u_{sat}$ for a specified time period until the costate p_2 attains a value of 1. At this point, the thruster will turn off and will coast until the costate p_2 reaches the value of -1 and a corresponding control saturation $u_x = u_{sat}$ until final position and velocity at soft dock is achieved. A quick check of the dynamics at time $t = t_0 = 0$ yields

$$\ddot{x}(0) = \omega^2 R_0 - u_{sat} < 0$$

which show that the assumption of monotonically decreasing holds, and is thus a possible solution. The dynamics for this case are given by

$$\begin{aligned}
\ddot{x} &= \omega^2 x - u_{sat} &\implies x_1(t) &= c_{11}e^{\omega t} + c_{12}e^{-\omega t} + \frac{u_{sat}}{\omega^2} &t \in [t_0, t_1] \\
\ddot{x} &= \omega^2 x &\implies x_2(t) &= c_{21}e^{\omega t} + c_{22}e^{-\omega t} &t \in [t_1, t_2] \\
\ddot{x} &= \omega^2 x - u_{sat} &\implies x_3(t) &= c_{31}e^{\omega t} + c_{32}e^{-\omega t} + \frac{u_{sat}}{\omega^2} &t \in [t_2, t_f]
\end{aligned} \tag{3.25}$$

where the same procedure can be applied as in Case 3 by imposing the boundary conditions given by the docking problem. Imposing these conditions result in

$$\begin{aligned}
x_1(t) &= \left(\frac{R_0 - \frac{u_{sat}}{\omega^2}}{2} \right) (e^{\omega t} + e^{-\omega t}) + \frac{u_{sat}}{\omega^2} &t \in [t_0, t_1] \\
x_2(t) &= c_{21}e^{\omega t} + c_{22}e^{-\omega t} &t \in [t_1, t_2] \\
x_3(t) &= \left(\frac{R_f - \frac{u_{sat}}{\omega^2}}{2} \right) (e^{\omega(t-t_f)} + e^{-\omega(t-t_f)}) + \frac{u_{sat}}{\omega^2} &t \in [t_2, t_f]
\end{aligned} \tag{3.26}$$

Equation (3.26) contains five unknowns: t_1 , t_2 , t_f , c_{21} , and c_{22} . The continuity conditions at t_1 and t_2 only accounts for four equations. It is possible to bring an extra equation to this unconstrained system by forcing either $p_2(t_1) = 1$ or $p_2(t_2) = -1$. This set of five equations has no analytical solution and a numerical method must be used. However, the only admissible solution that accepts either $p_2(t_1) = 1$ or $p_2(t_2) = -1$ in addition to the continuity condition is $t_2 = t_f$. This results in the same form as Case 3. The only situation in which Case 4 represents a viable solution is the degenerate case in which $\omega = 0$ at which point the solution is a simple bang-off-bang controller with a linear trajectory.

3.3.3 Summary of Flat Spin Solution

This section presented a fully-analytical solution to the flat spin case. This solution is also admissible to the case in which the docking problem has a general inertia. Below, a summary of the trajectory, control history, costate, and cost is presented, which corresponds to the solution of Case 3 from the previous section.

One of the benefits of having this analytical solution is to observe the trends as

Table 3.4: Summary of Solution to the Flat Spin Docking Problem

Trajectory	$x_1(t) = \frac{R_0 - \frac{u_{sat}}{\omega^2}}{2} (e^{\omega t} + e^{-\omega t}) + \frac{u_{sat}}{\omega^2} \quad t \in [0, t_1]$ $x_2(t) = \frac{R_f}{2} (e^{\omega(t-t_f)} + e^{-\omega(t-t_f)}) \quad t \in [t_1, t_f]$
Switch Time	$t_1 = \frac{1}{\omega} \log(Y)$ with Y from Eq. (3.23)
Final Time	$t_f = t_1 - \frac{1}{\omega} \log(X)$ with X from Eq. (3.23)
Control	$u_x = \begin{cases} -u_{sat} \\ 0 \end{cases} \quad \text{and} \quad u_y = \begin{cases} \omega^2 \text{sign}(\omega) (R_0 - \frac{u_{sat}}{\omega^2}) (e^{\omega t} - e^{-\omega t}) & t \in [t_0, t_1] \\ \omega^2 \text{sign}(\omega) R_f (e^{\omega(t-t_f)} - e^{-\omega(t-t_f)}) & t \in [t_1, t_f] \end{cases}$
Costate	$\begin{bmatrix} p_1(t) \\ p_2(t) \end{bmatrix} = \begin{bmatrix} -2\omega \text{sign}(2\omega\dot{x}) - \tilde{c}_1\omega e^{\omega t} + \tilde{c}_2\omega e^{\omega t} \\ \tilde{c}_1 e^{\omega t} + \tilde{c}_2 e^{-\omega t} \end{bmatrix}$
Cost	$ u_{sat}t_1 + \left \omega \left(R_0 - \frac{u_{sat}}{\omega^2} \right) (e^{\omega t_1} + e^{-\omega t_1} - 2) \right $ $+ \left \omega R_f (2 - e^{\omega(t_1-t_f)} - e^{-\omega(t_1-t_f)}) \right $

either the angular velocity increases or the saturation limit increases. For example, as the magnitude of the saturation increases, u_{sat} , the value for Y in Equation (3.23) decreases, thus making the value for t_1 smaller. Furthermore, as the angular velocity decreases, the final time t_f increases as expected. In the limit of $\omega \rightarrow 0$, t_f goes to infinity. This trend makes intuitive sense because having no angular velocity would make the rotating reference frame become inertial. This will in turn cause the dynamics to simplify to a simple double integrator, which has the optimal fuel policy of a bang-off-bang controller that approaches the target at infinite time with almost no fuel (i.e. $t_1 \rightarrow 0$) and becomes the solution to Case 4. Finally, The fuel cost as well as final time increases when $R_f \rightarrow 0$ or $R_0 \rightarrow \infty$. Additionally, the special case where $R_0 = R_f$ yields that the critical time, final time, and cost go to zero as expected.

It should be noted that the solution given in Table 3.4 is fully analytical and can be implemented in flight software with almost zero computational complexity as compared to any other method. One benefit for having a flat spin solution is that almost all objects in space will eventually rotate about the maximum moment of

inertia axis due to the natural stability of Newton-Euler dynamics in the presence of energy dissipating forces such as fuel slosh [37] or structural viscoelasticity. Therefore, many of the space debris such as spent rocket bodies would be spinning about a single principle axis for which the solution presented is applicable if the docking port is aligned with the maximum moment of inertia.

Furthermore, it should be noted that a singularity occurs when the docking problem has flat spin about the docking axis. This solution presents a problem to this formulation as the only feasible trajectory is a bang-off-bang controller (corresponding to Case 4 in Figure (3-3)). For this case, the recommendation would be to instead use the solution to the general spin problem given in Section 3.4. Nevertheless, there exist situations in which the Target has a complex tumble such as satellite servicing or space assembly (i.e. situations in which the uncontrollable tumble has not reached stability due to energy dissipation). In terms of plume impingement, the solution for the flat spin, with the exception for the degenerate case $\omega = 0$, is a Bang-Off controller which naturally eliminates the risk of plume impingement due to the lack of firing towards the docking port.

3.4 Quasi-Analytical Solution to the General Spin Docking Problem

The next case that was solved was that of a Target having a general spin about any axis and a spherical inertia. This is a more complex scenario as now the angular velocity vector (or angular momentum vector, since they are pointing along the same direction) is not perpendicular to the docking axis. In the case of the flat spin, the docking port axis, as seen in the inertial frame, traces a disk, whereas in the case of general spin the docking axis traces a cone. The inner angle of that cone can be computed as $\cos(\theta) = (\hat{x} \cdot \boldsymbol{\omega})/|\boldsymbol{\omega}|$ where \hat{x} is the axis of the docking port.

The approach to solve this case is similar to that of the flat spin case in that the conditions for optimality along with Pontryagin's Minimum Principle were used. To

begin, the optimal control problem for the general spin and spherical inertia can be stated as follows,

$$\begin{aligned}
& \min_{u_x} \int_{t_0}^{t_f} \left(|u_x| + |2\omega_z \dot{x} + \omega_y \omega_x x| + |-2\omega_y \dot{x} + \omega_z \omega_x x| \right) dt \\
& \text{Subject to } \ddot{x} = (\omega_y^2 + \omega_z^2)x + u_x \\
& \quad x(t_0) = R_0, \dot{x}(t_0) = 0 \\
& \quad x(t_f) = R_f, \dot{x}(t_f) = 0 \\
& \quad -u_{sat} \leq u_x \leq u_{sat} \\
& \quad t_f \text{ free}
\end{aligned} \tag{3.27}$$

Equation (3.27) has a cost function that also includes the effects of Coriolis and Centripetal acceleration that the controllers u_y and u_z have to apply in order to always remain aligned along \hat{x} . Furthermore, the dynamics of the problem are updated to include the effect of the angular velocity in both the y and z axes. The constraint that the Chaser begins radially aligned and ends in soft dock remains the same.

3.4.1 Necessary Conditions for Optimality

As before, the Hamiltonian for this problem is given by,

$$\mathcal{H} = |u_x| + |2\omega_z \dot{x} + \omega_y \omega_x x| + |-2\omega_y \dot{x} + \omega_z \omega_x x| + p_1(\dot{x}) + p_2((\omega_y^2 + \omega_z^2)x + u_x) \tag{3.28}$$

The equation of the costates can be obtained by the second optimality condition and yields

$$\begin{bmatrix} \dot{p}_1 \\ \dot{p}_2 \end{bmatrix} = \begin{bmatrix} 0 & -(\omega_y^2 + \omega_z^2) \\ -1 & 0 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} + \begin{bmatrix} -\omega_y \omega_x & -\omega_z \omega_x \\ -2\omega_z & 2\omega_y \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \end{bmatrix} \tag{3.29}$$

where $\sigma_1 = \text{sign}(2\omega_z \dot{x} + \omega_y \omega_x x)$ and $\sigma_2 = \text{sign}(-2\omega_y \dot{x} + \omega_z \omega_x x)$ and arises due to the optimality condition $\dot{\mathbf{p}} = -\partial \mathcal{H} / \partial \mathbf{x}$. Since the distance between the Chaser and Target, x , will be a positive monotonically decreasing function and \dot{x} will be a negative

function that begins and ends at zero, then σ_1 and σ_2 will at most switch sign twice.

Due to this uncertainty of when the signs switch, one approach would be to numerically integrate Equation (3.29). This will undoubtedly introduce numerical errors, so to avoid that, a valid approach would be to analytically integrate Equation (3.29) at time periods for which σ_1 and σ_2 are constant.

Assuming that there is no sign switch that would cause the value of σ_1 and σ_2 to not be constant, then the solution of the differential equations shown in Equation (3.29) is

$$\begin{aligned} p_1(t) &= -c_1 \sqrt{\omega_y^2 + \omega_z^2} e^{\sqrt{\omega_y^2 + \omega_z^2} t} + c_2 \sqrt{\omega_y^2 + \omega_z^2} e^{-\sqrt{\omega_y^2 + \omega_z^2} t} - 2\sigma_1 \omega_z + 2\sigma_2 \omega_y \\ p_2(t) &= c_1 e^{\sqrt{\omega_y^2 + \omega_z^2} t} + c_2 e^{-\sqrt{\omega_y^2 + \omega_z^2} t} - \frac{\sigma_1 \omega_y \omega_x + \sigma_2 \omega_z \omega_x}{\omega_y^2 + \omega_z^2} \end{aligned} \quad (3.30)$$

At each step of the integration, c_1 and c_2 from Equation (3.30) must be chosen such that there is continuity between p_1 and p_2 . Integration would start at a point in which both values $p_1(t)$ and $p_2(t)$ are known, that is at the first switch point in which $p_2(t_1) = 1$ and $p_1(t_1)$ can be obtained by solving the equation $\mathcal{H}(t_1) = 0$ for p_1 .

Additionally, the control policy for the problem is given by $\frac{\partial \mathcal{H}}{\partial u} = 0 = \pm 1 + p_2$ as before,

$$u_x^* = \begin{cases} -u_{sat} & p_2 > 1 \\ 0 & -1 < p_2 < 1 \\ u_{sat} & p_2 < -1 \end{cases} \quad (3.31)$$

The same procedure as the flat spin can be applied by using the transversality condition which yields,

$$\begin{aligned} p_2(t_f) &= \frac{-\left(|u_x(t_f)| + |R_f \omega_y \omega_x| + |R_f \omega_z \omega_x|\right)}{u_x(t_f) + R_f (\omega_y^2 + \omega_z^2)} \\ p_2(t_0) &= \frac{-\left(|u_x(t_0)| + |R_0 \omega_y \omega_x| + |R_0 \omega_z \omega_x|\right)}{u_x(t_0) + R_0 (\omega_y^2 + \omega_z^2)} \end{aligned} \quad (3.32)$$

Given Equations (3.31) and (3.32), the same approach can be taken as the flat spin case in which a guess is made on as to the final control value. Then, if the value

of the costate is consistent with the constraints due to the control policy, that value represents a possible optimal trajectory. The result from this analysis indicates that the condition for a Bang-Off controller or Bang-Off-Bang controller can be obtained by defining

$$\gamma = \frac{|\omega_y \omega_x| + |\omega_z \omega_x|}{\omega_y^2 + \omega_z^2} \quad (3.33)$$

If $\gamma \leq 1$, then a Bang-Off controller $u_x = -u_{sat} \forall t \in [t_0, t_1]$, $u_x = 0 \forall t \in [t_1, t_f]$ can be used as the value for the costate p_2 , which is consistent with the control policy by Equation (3.31). Furthermore, if $\gamma > 1$, then a Bang-Off-Bang controller is needed: $u_x = -u_{sat} \forall t \in [t_0, t_1]$, $u_x = 0 \forall t \in [t_1, t_2]$, and $u_x = u_{sat} \forall t \in [t_2, t_f]$.³

It will turn out that there exists a fully analytical solution for the Bang-Off controller; however, a numerical approach will have to be taken to solve for a Bang-Off-Bang Controller. An analysis must then be made to determine whether the analytical solution is needed instead of the numerical solution, as this will impact the computational complexity of the General Spin Spherical Inertia problem.

3.4.2 Differentiation Between Controllers Using γ Condition

Since γ is only dependent on the values of $\boldsymbol{\omega}$, a method of exploring the nature of γ is to plot its value for each possible direction vector of $\boldsymbol{\omega}$. A mesh grid was used to sample a unit sphere representing every possible angular velocity direction of the

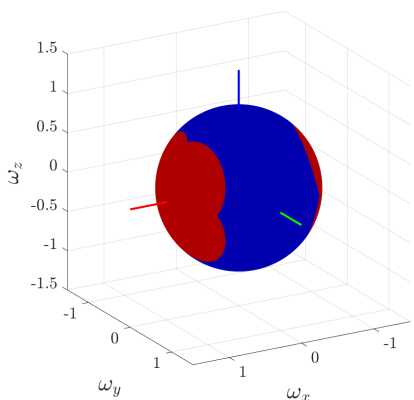
³**Proof:** Due to PMP conditions, the control time history must be either 0, u_{sat} , or $-u_{sat}$ at the final time. If $u_x(t_f) = 0$ then by the Transversality condition, $p_2(t_f) = \frac{-(|R_f \omega_y \omega_x| + |R_f \omega_z \omega_x|)}{R_f(\omega_y^2 + \omega_z^2)}$. Since $\|p_2\| \leq 1$ has to be true for the optimal control policy to hold per Eq. (3.31), then $(|\omega_y \omega_x| + |\omega_z \omega_x|) < (\omega_y^2 + \omega_z^2)$ must hold. Otherwise, the Transversality condition will not hold and $\mathcal{H}(t_f) > 0$. Note that $\|p_2\| \leq 1$ instead of $\|p_2\| < 1$, this is valid as the problem does not contain a singular arc due to the dynamics having poles on the RHS, so although $p_2(t) = 1$ at some point, the dynamics will push the costate such that $p_2(t + \epsilon) \neq 1$.

Alternatively, if $u_x(t_f) = u_{sat}$ then $p_2(t_f) = -(u_{sat} + |R_f \omega_y \omega_x| + |R_f \omega_z \omega_x|) / (u_{sat} + R_f(\omega_y^2 + \omega_z^2))$. Again, $p_2(t_f) < -1$ in accordance with Eq. (3.31). This means that $-(u_{sat} + |R_f \omega_y \omega_x| + |R_f \omega_z \omega_x|) / (u_{sat} + R_f(\omega_y^2 + \omega_z^2)) < -1$ or $|\omega_y \omega_x| + |\omega_z \omega_x| > \omega_y^2 + \omega_z^2$. Therefore, γ given in Eq. 3.33 indicates whether at the final time the controller will be at 0, or u_{sat} .

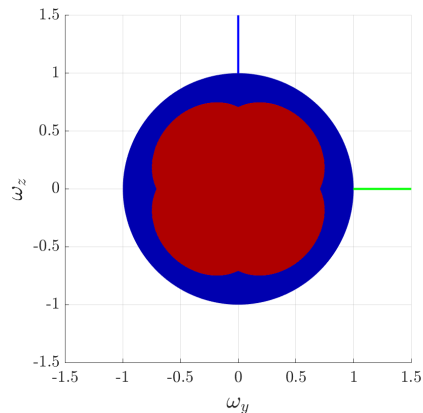
Furthermore, at the beginning the Chaser must fire inward, $u_x(t_0) = -u_{sat}$. Since for both cases $p_2(t_0) > p_2(t_f)$ and the general form of $p_2(t)$ does not oscillate, then there can only be one switch in the case where $u_x(t_f) = 0$ or two switches when $u_x(t_f) = u_{sat}$.

For completeness, the case where $u_x(t_f) = -u_{sat}$ was explored; however, the trajectory for $x(t)$ violated the PMP condition that p_2 must equal 1 when a switch occur. Therefore, $u_x(t_f) = -u_{sat}$ is not a valid value.

Target. As before, the docking port is assumed to be aligned with the body x-axis that is also one of the principal inertia axes. To differentiate between the controllers two colors were used, blue for when $\gamma \leq 1$ and red when $\gamma > 1$. Figure (3-4) shows the resultant surface.



(a) 3D projection of γ condition



(b) YZ projection of γ condition

Figure 3-4: γ condition for all possible ω direction. Blue area represents the analytical Bang-Off controller. Red area represents the numerical Bang-Off-Bang controller.

Both Figure (3-4a) and (3-4b) show the unit vector axis representing the x , y and z axis in the usual RGB colors. The blue region represents the Bang-Off controller solution which can be computed analytically. The red region represents the Bang-Off-Bang controller in which a numerical approach is needed to obtain the solution. Although the numerical solution will be significantly more computationally expensive than an analytical solution, the solution obtained in the red region will be faster than using a Gauss pseudospectral method or a model predictive controller. Furthermore, the blue region represents approximately 76% of all possible spin directions and includes the Flat Spin problem (which represents the outer edge of the sphere seen in Figure (3-4b)).

The shape of the Bang-Off-Bang region has to be symmetric about both the YX-plane and ZX-plane due to the absolute value in the function gamma,

$$|\omega_y\omega_x| + |\omega_z\omega_x| = |-\omega_y\omega_x| + |\omega_z\omega_x| = |\omega_y\omega_x| + |-\omega_z\omega_x|$$

It is possible to obtain the minimum angle between the docking port (red axis or center of Figure (3-4b)) and the angular velocity of the Target at which the analytical solution is valid. This is equivalent to the location of the kinks in the sphere along the axis $\omega_y = 0$ and $\omega_z = 0$ in Figure (3-4b). By assuming that the y-component of the angular velocity is zero, the condition for the Bang-Off-Bang reduces to

$$\frac{|\omega_z \omega_x|}{\omega_z^2} \geq 1$$

Additionally, $\boldsymbol{\omega}$ represents a direction of the angular vector, so $\omega_x^2 + \omega_y^2 + \omega_z^2 = 1$. Given that for this case $\omega_y = 0$, then based on the aforementioned condition the values for the angular velocity represent $\boldsymbol{\omega}_{crit,low} = [\pm \frac{1}{\sqrt{2}}, 0, \pm \frac{1}{\sqrt{2}}]^T$. The angle corresponds to 45° .

The angle for the angular velocity direction that represents the maximum point at which the analytical solution begins to be valid (or the peak of the petals in the red area) can also be analytically computed. First, due to the absolute value in the γ condition, it makes intuitive sense that there will not be destructive interference and the maximum will occur when $\omega_y = \omega_z$. Letting $\boldsymbol{\omega} = [\sqrt{1 - 2a^2}, a, a]^T$, the γ condition for which the analytical solution is valid simplifies to

$$\frac{|a(1 - \sqrt{1 - 2a^2})| + |a(1 - \sqrt{1 - 2a^2})|}{2a^2} \leq 1$$

Solving for the value of a yields the angular velocity vector as $\boldsymbol{\omega}_{crit,upp} = [\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}]^T$. The corresponding angle from the critical angular velocity vector with the docking port is 54.73° .

It should be noted that the shape of Figure (3-4) is a mathematical construct, but does not reflect the physical nature of having a spherical inertial Target. Because any axis in a spherical object is, by definition, a principal axis, it is possible to define a rotation such that the petal shape is not observed. For example, if two different principal axes y' and z' are selected to be a x axis rotation of the original y and z , then the minimum angle location will also rotate. Thus, if the angle of the docking port and the angular velocity vector is less than 54.73° and greater than 45° , it is always

possible to find a rotation matrix such that the solution is a Bang-Off controller. In practicality, this would correspond to applying a rotation matrix transformation to the current estimated angular velocity vector to a set of axis that allows the Bang-Off solution, $\omega = {}^C\omega$, and perform the control firing in the x , y' , and z' axes.

Overall, by inspecting the expression for γ , it has been determined that in order to use the analytical method the angle between the docking port axis ($+x$ -axis) and the angular velocity vector must be at least 45° , or at most 54.73° . This constraint can be used in the choice of the docking location by the Chaser algorithms in Phase 2 as it will determine whether the analytical function can be used.

3.4.3 Analytical Solution to Bang-Off Controller

Assuming that the Target angular spin is such that $\gamma \leq 1$ per Equation (3.33), the trajectory will correspond to a Bang-Off controller,

$$\begin{aligned}\ddot{x}_1 &= \omega_{eff}^2 x_1 - u_{sat} \implies \frac{R_0 - \frac{u_{sat}}{\omega_{eff}^2}}{2} (e^{\omega_{eff} t} + e^{-\omega_{eff} t}) + \frac{u_{sat}}{\omega_{eff}^2} & t \in [0, t_1] \\ \ddot{x}_2 &= \omega_{eff}^2 x_2 \implies \frac{R_f}{2} (e^{\omega_{eff}(t-t_f)} + e^{-\omega_{eff}(t-t_f)}) & t \in [t_1, t_f]\end{aligned}\quad (3.34)$$

where $\omega_{eff} = \sqrt{\omega_y^2 + \omega_z^2}$ was used for simplicity. Two constraints were applied: the first section of the trajectory must begin radially aligned with zero speed i.e. $x(0) = R_0$, $\dot{x}(0) = 0$, and the second section must end at soft docking conditions, i.e. $x(t_f) = R_f$, $\dot{x}(t_f) = 0$. Equation (3.34) contains two unknowns: the switch time and final time, t_1 and t_f respectively. The switch time and final time can be solved analytically using the same procedure as in the Flat spin case, namely assuming that $Y = e^{\omega_{eff} t_1}$ and $X = e^{\omega_{eff}(t_1 - t_f)}$. The solution can then be solved as

$$\begin{aligned}X &= \left(\frac{-4a^2 + b^2 + 4c^2}{4bc} \right) - \sqrt{\left(\frac{-4a^2 + b^2 + 4c^2}{4bc} \right)^2 - 1} \\ Y &= \frac{c}{a} \left(\left(\frac{-4a^2 - b^2 + 4c^2}{4bc} \right) - \sqrt{\left(\frac{-4a^2 + b^2 + 4c^2}{4bc} \right)^2 - 1} \right)\end{aligned}\quad (3.35)$$

where $a = (R_0 - \frac{u_{sat}}{\omega_{eff}^2})/2$, $b = u_{sat}/\omega_{eff}^2$, $c = R_f/2$. Note that $Y \neq \frac{c}{a}X$ as there is a negative sign in the first term of Y compared to X . With the analytical solution to X and Y , the resultant value for the switch time and end time can be readily available by evaluating

$$\begin{aligned} t_1 &= \frac{1}{\omega_{eff}} \log(Y) \\ t_f &= t_1 - \frac{1}{\omega_{eff}} \log(X) \end{aligned} \tag{3.36}$$

Sensitivity Analysis of Switch and Final Time

Due to the nature of Equation (3.35) and (3.36) it is difficult to obtain trends because how the switch time and final time differ with a change in one of the parameters R_0 , R_f , u_{sat} , or ω . However, one of the main benefits of obtaining an analytical solution is the possibility to obtain a fast analysis by simply modifying a value and computing the resultant switch and final time. To carry out this analysis, nominal values for R_0 , R_f , u_{sat} , or ω and respective times were arbitrarily chosen. These values are summarized in Table 3.5. The choice of the value for the parameters does not represent an actual system or case scenario in particular, yet the overall trend is similar with other choices for the parameters. The only consideration was that u_{sat} was selected such that there is enough control authority to enable the maneuver at all. Furthermore, out of ω_x and ω_y , only ω_y was modified; however, the trend would be similar if ω_z changed instead.

Table 3.5: Nominal Values for Sensitivity Analysis of General Spin, Spherical Inertia Analytical Solution

$\mathbf{R}_0^{\text{nom}}$ (m)	$\mathbf{R}_f^{\text{nom}}$ (m)	$\mathbf{u}_{\text{sat}}^{\text{nom}}$ (m/s ²)	ω_y^{nom} (°/s)	$\mathbf{t}_1^{\text{nom}}$ (s)	$\mathbf{t}_f^{\text{nom}}$ (s)
10	5	10	20	0.49	2.93

The nominal parameters were modified by $\pm 50\%$ and a plot was produced by keeping all but one parameter nominal. The respective switch time and final time corresponding to the candidate optimal trajectory were recorded. The resultant plots are shown in Figure (3-5).

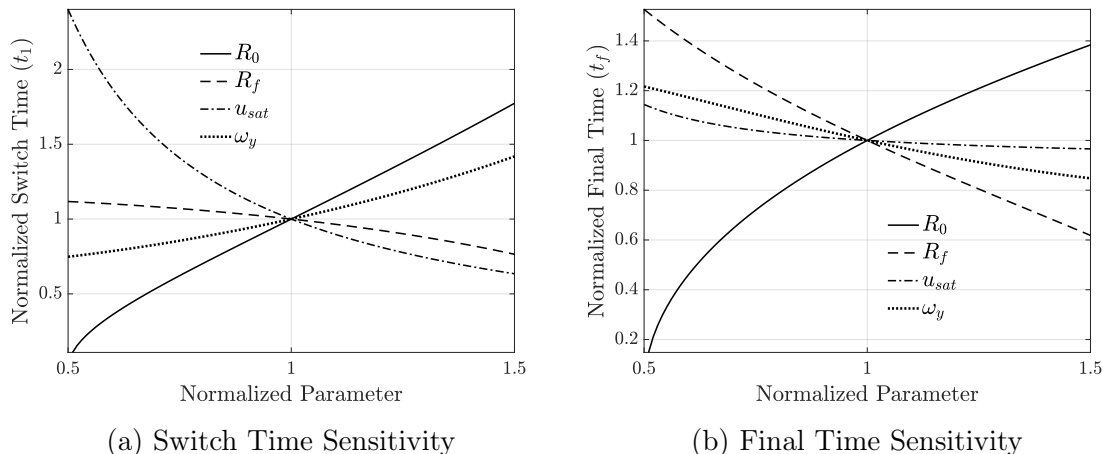


Figure 3-5: Sensitivity Analysis for Analytical Switch and Final Times

Figures (3-5a) and (3-5b) represent the normalized change in the switch time and final time respectively, as a function of the change in a normalized parameter. As expected, both t_1 and t_f increase as the initial radial distance from the Target increases. This makes intuitive sense as the trajectory time as well as the time it takes for the Chaser to achieve the correct radial speed should increase with an increased initial distance. The opposite behavior is observed with the increase in the final offset distance, R_f . As expected, t_1 and t_f will approach 0 when the final docking distance is the same as the initial radial offset distance.

Both t_1 and t_f will decrease with an increase in the magnitude of the saturation and vice versa. An interesting trend to note is that although the switch time *will* approach zero with an increase in the value of u_{sat} , t_f *will not*. The reasoning behind this trend is as follows: as $u_{sat} \rightarrow \infty$, $t_1 \rightarrow 0$ and the controller will become an impulsive controller which will set the initial velocity $\dot{x}(0)$ to a value such that, in the radial direction, the Chaser will simply coast. For each radial distance offset, R_0 there exists a minimum fuel radial velocity $\dot{x}(0)$ such that after t_f seconds the trajectory will reach R_f with $\dot{x} = 0$. Thus, the u_{sat} curve on Figure (3-5b) will asymptote to the final time of the impulsive general spin, general inertia docking solution. The impulsive version can be seen in Appendix B.

Finally, from Figure (3-5) the switch time, t_1 , increases while the final time, t_f ,

decreases as the angular velocity parameter in y , ω_y , increases and vice versa. This is somewhat unintuitive, but it is expected that the radial velocity needed to reach the radial coasting speed (i.e. to begin the trajectory with control set to zero) will increase as the angular velocity component normal to the docking port increases. However, as the angular velocity increases, staying aligned along x-axis by continuously applying the Coriolis and Centripetal acceleration becomes more expensive. Therefore, as the Target spins faster, the Chaser should dock faster to spend less time being radially aligned.

Overall, all the trends defined by the parameters needed to solve the switch and final time are consistent with the expected behavior based on the system dynamics. This is one of the most powerful results of having a fully analytical expression as it allows the observation of trends where a full optimization routine will not allow (or will be too computationally expensive to obtain).

3.4.4 Numerical Solution to Bang-Off-Bang Controller

The other case in which $\gamma > 1$ per Equation (3.33) corresponds to a Bang-Off-Bang Controller,

$$\begin{aligned}
\ddot{x}_1 = \omega_{eff}^2 x_1 - u_{sat} &\implies \frac{R_0 - \frac{u_{sat}}{\omega_{eff}^2}}{2} (e^{\omega_{eff} t} + e^{-\omega_{eff} t}) + \frac{u_{sat}}{\omega_{eff}^2} & t \in [0, t_1] \\
\ddot{x}_2 = \omega_{eff}^2 x_2 &\implies c_{21} e^{\omega_{eff} t} + c_{22} e^{-\omega_{eff} t} & t \in [t_1, t_2] \\
\ddot{x}_3 = \omega_{eff}^2 x_3 + u_{sat} &\implies \frac{R_f + \frac{u_{sat}}{\omega_{eff}^2}}{2} (e^{\omega_{eff}(t-t_f)} + e^{-\omega_{eff}(t-t_f)}) - \frac{u_{sat}}{\omega_{eff}^2} & t \in [t_2, t_f]
\end{aligned} \tag{3.37}$$

where c_{21} and c_{22} represent unknown coefficients such that the three segments remain continuous up to the first derivative. Additionally, in Equation (3.37) the first and third segment of the trajectory already satisfy the boundary conditions set up by the problem statement.

Because all three segments of the trajectory must be continuous up to the first derivative there are four equations ($x_1(t_1) = x_2(t_1)$, $\dot{x}_1(t_1) = \dot{x}_2(t_1)$, $x_2(t_2) = x_3(t_2)$,

and $\dot{x}_2(t_2) = \dot{x}_3(t_2)$), but five unknowns (c_{21} , c_{22} , t_1 , t_2 , and t_f). Unfortunately, due to the extra degree of freedom and complexity of the expressions no analytical solution exists for the five unknowns. Thus, there is a need for a numerical procedure to be used.

Several approaches were used for solving the five unknowns. The first approach was to utilize Newton's Method using Matlab's *fminunc.m*. The issue with this approach as is common with Newton's Method, is that there is no guarantee that the solution to this problem represents the global minimum. In other words, there could be several sets of c_{21} , c_{22} , t_1 , t_2 , and t_f values that satisfy the continuity condition but only represent a local minimum. Even adding another constraint such as $p_2(t_1) = 1$ using Equation (3.30) leads to other problem as *fminunc* is highly sensitive to the initial condition. Due to this, the Newton's Method approach is unreliable to finding the true solution to the five unknowns in Equation (3.37). Two different approaches were attempted: the use of a nonlinear program (NLP) and line search algorithm. The approach and results from each are presented here.

Nonlinear Program For Solution to the Bang-Off-Bang Controller

The original problem formulation in Equation (3.27) can now be rewritten as follows,

$$\begin{aligned}
 & \min_{c_{21}, c_{22}, t_1, t_2, t_f} \int_{t_0}^{t_f} \left(|u_x| + |2\omega_z \dot{x} + \omega_y \omega_x x| + |-2\omega_y \dot{x} + \omega_z \omega_x x| \right) dt \\
 & \text{Subject to } x_1(t_1) = x_2(t_1) \\
 & \dot{x}_1(t_1) = \dot{x}_2(t_1) \\
 & x_2(t_2) = x_3(t_2) \\
 & \dot{x}_2(t_2) = \dot{x}_3(t_2) \\
 & p_2(t_1) = 1 \text{ or } p_2(t_2) = -1 \\
 & 0 \leq t_1 \leq t_2 \leq t_f
 \end{aligned} \tag{3.38}$$

where the known structure of the trajectory given in Equation (3.37) was substituted instead of having the original dynamics. The benefit of the problem formulation in Equation (3.38) is that it is now a NLP with only 5 variables in contrast with the infinite dimension problem given by Equation (3.27). The issue remains that an initial guess must be made for the five variables; the choice of the guess can lead to situation in which the NLP takes significant amount of iterations to converge which is not ideal for situations in which a solution must be recomputed online.

Line Search Approach for Solution to the Bang-Off-Bang Controller

Since the result of using a Newton's Method or NLP is very sensitive to the initial guess, it can lead to situations in which the convergence to the solution takes too many iterations or does not converge at all. Therefore, a different approach was used. Instead of guessing a value for five variables, the algorithm only had to search through one variable. Assuming a guess for the first switch time t_1 the trajectory x_2 can be found by imposing the first two conditions of the constraints in the NLP program and finding c_{21} and c_{22} .

$$\begin{aligned} c_{21} &= \frac{\omega_{eff}^2 R_0 + u_{sat} (e^{-\omega_{eff} t_1} - 1)}{2\omega_{eff}^2} \\ c_{22} &= \frac{\omega_{eff}^2 R_0 + u_{sat} (e^{\omega_{eff} t_1} - 1)}{2\omega_{eff}^2} \end{aligned} \quad (3.39)$$

With these variables, only two unknowns remain, namely t_2 and t_f , along with two equations. The same approach of finding the root of a solution with exponentials as in the Flat Spin case can be taken by imposing the continuity condition on $x_2(t_2)$, $\dot{x}_2(t_2)$ and $x_3(t_2)$, $\dot{x}_3(t_2)$. Letting that $\zeta = e^{\omega_{eff} t_2}$ and $\xi = e^{\omega(t_2 - t_f)}$,

$$\begin{aligned} \zeta &= \frac{1}{4c_{21}\beta} \left(-4c_{21}c_{22} + 4\alpha^2 - \beta^2 - \sqrt{-16c_{21}c_{22}\beta^2 + (4c_{21}c_{22} - 4\alpha^2 + \beta^2)^2} \right) \\ \xi &= \frac{1}{4\alpha\beta} \left(-4c_{21}c_{22} + 4\alpha^2 + \beta^2 - \sqrt{-16c_{21}c_{22}\beta^2 + (4c_{21}c_{22} - 4\alpha^2 + \beta^2)^2} \right) \end{aligned} \quad (3.40)$$

where $\alpha = (R_f + \frac{u_{sat}}{\omega_{eff}^2})/2$ and $\beta = u_{sat}/\omega_{eff}^2$. Finally, the last two variables can be

found as,

$$\begin{aligned} t_2 &= \frac{1}{\omega_{eff}} \log(\zeta) \\ t_f &= t_2 - \frac{1}{\omega_{eff}} \log(\xi) \end{aligned} \quad (3.41)$$

Having an analytical solution to all the unknown variables, it is possible to check the validity of the solution for a given t_1 by performing the logic check $(t_f > t_2) \& (t_2 > t_1) \& (\text{isreal}(t_2))$ where if any of the statements is false then the selected t_1 is also invalid. By reducing the problem to one dimension we can then perform a line search approach

$$t_1^* = \min(f(t_1)) \quad (3.42)$$

where

$$f(t_1) = \int_{t_0}^{t_f} \left(|u_x| + |2\omega_z \dot{x} + \omega_y \omega_x x| + |-2\omega_y \dot{x} + \omega_z \omega_x x| \right) dt \quad (3.43)$$

At each iteration, a new t_1 is selected and the values for c_{21} , c_{22} , t_2 , and t_f are computed by Equation (3.39) and (3.41), respectively. The expression for x and \dot{x} can be obtained by Equation (3.37), and a numerical computation of the function $f(t_1)$ can be performed using Matlab's trapz.m function. The search ends when the minimum value for $f(t_1)$ is found. Matlab's fminsearch can be used to perform the line search. Finally, a very good approximation is the t_1 value obtained from the Bang-Off solution given by Equation (3.36). In fact, this value will represent a lower bound on the solution and tests show that the initial guess is around 1 – 4% off from the optimal t_1^* .

3.4.5 Summary of Solution for the General Spin, Spherical Inertia Docking Problem

This section provides a summary of the procedure to obtain a solution to the general spin, spherical inertia problem.

Determine which type of controller is needed by computing γ from Equation (3.33):

$$\gamma = \frac{|\omega_y \omega_x| + |\omega_z \omega_x|}{\omega_y^2 + \omega_z^2}$$

if $\gamma \leq 1$, use Table 3.6 for the analytical solution (excluding cost), else use Table 3.7 for the numerical solution.

Table 3.6: $\gamma \leq 1$: Summary of Solution to the General Spin, General Inertia Problem

$x(t)$	$x_1(t) = \frac{R_0 - \frac{u_{sat}}{\omega_{eff}^2}}{2} (e^{\omega_{eff}t} + e^{-\omega_{eff}t}) + \frac{u_{sat}}{\omega_{eff}^2}$	$t \in [0, t_1]$
	$x_2(t) = \frac{R_f}{2} (e^{\omega_{eff}(t-t_f)} + e^{-\omega_{eff}(t-t_f)})$	$t \in [t_1, t_f]$
	where $\omega_{eff} = \sqrt{\omega_y^2 + \omega_z^2}$	
$\dot{x}(t)$	$\dot{x}_1(t) = \omega_{eff} \left(\frac{R_0 - \frac{u_{sat}}{\omega_{eff}^2}}{2} \right) (e^{\omega_{eff}t} - e^{-\omega_{eff}t})$	$t \in [0, t_1]$
	$\dot{x}_2(t) = \omega_{eff} \left(\frac{R_f}{2} \right) (e^{\omega_{eff}(t-t_1)} - e^{-\omega_{eff}(t-t_1)})$	$t \in [t_1, t_f]$
Switch Time	$t_1 = \frac{1}{\omega_{eff}} \log(Y)$ with Y from Eq. (3.35)	
Final Time	$t_f = t_1 - \frac{1}{\omega_{eff}} \log(X)$ with X from Eq. (3.35)	
Control x	$u_x = \begin{cases} -u_{sat} & t \in [0, t_1] \\ 0 & t \in [t_1, t_f] \end{cases}$	
	$u_y = \begin{cases} 2\omega_z \dot{x}_1 + x_1 \omega_y \omega_x \\ 2\omega_z \dot{x}_2 + x_2 \omega_y \omega_x \end{cases}$ and $u_z = \begin{cases} -2\omega_y \dot{x}_1 + x_1 \omega_z \omega_x & t \in [0, t_1] \\ -2\omega_y \dot{x}_2 + x_2 \omega_z \omega_x & t \in [t_1, t_f] \end{cases}$	
Costate	Numerically integrating Equation (3.29)	
Cost	$\int_{t_0}^{t_f} (u_x + 2\omega_z \dot{x} + \omega_y) dt$ computed numerically	

It should be noted that the Bang-Off-Bang solution, although fuel optimal, can cause plume impingement due to the application of the control towards the Target during the period $[t_2, t_f]$. This is in contrast to the Bang-Off solution in which the

Table 3.7: $\gamma > 1$: Summary of Solution to the General Spin, General Inertia Problem

	$x_1(t) = \frac{R_0 - \frac{u_{sat}}{\omega_{eff}^2}}{2} (e^{\omega_{eff}t} + e^{-\omega_{eff}t}) + \frac{u_{sat}}{\omega_{eff}^2}$	$t \in [0, t_1]$
$x(t)$	$x_2(t) = c_{21}e^{\omega_{eff}t} + c_{22}e^{-\omega_{eff}t}$	$t \in [t_1, t_2]$
	$x_3(t) = \frac{R_f + \frac{u_{sat}}{\omega_{eff}^2}}{2} (e^{\omega_{eff}(t-t_f)} + e^{-\omega_{eff}(t-t_f)}) - \frac{u_{sat}}{\omega_{eff}^2}$	$t \in [t_2, t_f]$
	where $\omega_{eff} = \sqrt{\omega_y^2 + \omega_z^2}$	
	$\dot{x}_1(t) = \omega_{eff} \left(\frac{R_0 - \frac{u_{sat}}{\omega_{eff}^2}}{2} \right) (e^{\omega_{eff}t} - e^{-\omega_{eff}t})$	$t \in [0, t_1]$
$\dot{x}(t)$	$\dot{x}_2(t) = \omega_{eff} (c_{21}e^{\omega_{eff}t} - c_{22}e^{-\omega_{eff}t})$	$t \in [t_1, t_2]$
	$\dot{x}_3(t) = \omega_{eff} \left(\frac{R_f + \frac{u_{sat}}{\omega_{eff}^2}}{2} \right) (e^{\omega_{eff}(t-t_f)} - e^{-\omega_{eff}(t-t_f)})$	$t \in [t_2, t_f]$
t_1	Free variable subject to solution Eq. (3.42)	
c_{21} and c_{22}	$c_{21} = \frac{\omega_{eff}^2 R_0 + u_{sat} (e^{-\omega_{eff}t_1^*} - 1)}{2\omega_{eff}^2}$	
	$c_{22} = \frac{\omega_{eff}^2 R_0 + u_{sat} (e^{\omega_{eff}t_1^*} - 1)}{2\omega_{eff}^2}$	
t_2	$t_2 = \frac{1}{\omega_{eff}} \log(\zeta)$ with ζ from Eq. (??)	
t_f	$t_f = t_2 - \frac{1}{\omega_{eff}} \log(\xi)$ with ξ from Eq. (3.35)	
Control x	$u_x = \begin{cases} -u_{sat} & t \in [0, t_1] \\ 0 & t \in [t_1, t_f] \end{cases}$	
	$u_y = \begin{cases} 2\omega_z \dot{x}_1 + x_1 \omega_y \omega_x & t \in [0, t_1] \\ 2\omega_z \dot{x}_2 + x_2 \omega_y \omega_x & t \in [t_1, t_f] \end{cases}$	and $u_z = \begin{cases} -2\omega_y \dot{x}_1 + x_1 \omega_z \omega_x & t \in [0, t_1] \\ -2\omega_y \dot{x}_2 + x_2 \omega_z \omega_x & t \in [t_1, t_f] \end{cases}$
Costate	Eq. (3.30) with $p_2(0) = \frac{u_{sat} + \omega_y \omega_x R_0 + \omega_z \omega_x R_0 }{u_{sat} - (\omega_y^2 + \omega_z^2) R_0}$, $p_2(t_f) = -\frac{ \omega_y \omega_x + \omega_z \omega_x }{(\omega_y^2 + \omega_z^2)}$	
Cost	$\int_{t_0}^{t_f} \left(u_x + 2\omega_z \dot{x} + \omega_y \omega_x x + -2\omega_y \dot{x} + \omega_z \omega_x x \right) dt$ computed numerically	

thrusters are always fired away from the target during $[0, t_1]$. Due to this, it is possible that the form of the Bang-Off solution is more adequate to situations that deal with docking with objects. The next question is then to determine the level of suboptimality that the Bang-Off solution yields compared to the Bang-Off-Bang case when inside the region defined by $\gamma > 1$. This analysis is conducted in Appendix B.

3.5 Analytical Trajectory Optimizer for the General Spin, General Inertia Docking Problem

Since the assumption of a spherical inertia is invalid for most satellites, a procedure is required to solve the docking problem when the inertia of the system is more general, or in other words, when the inertia can be written as

$$I_{TAR} = \begin{bmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{bmatrix}$$

where I_1 , I_2 and I_3 corresponds to the principal moments of inertia in the x , y and z axes respectively.

This situation, as described in section 3.2, causes the angular velocity to change with respect to time due to Newton-Euler's equation.

Thus, the problem formulation for the general spin, general inertia docking problem is given by

$$\begin{aligned}
& \min_{u_x} \int_{t_0}^{t_f} \left(|u_x| + |2\omega_z \dot{x} + \dot{\omega}_z x + \omega_y \omega_x x| + |-2\omega_y \dot{x} - \dot{\omega}_y x + \omega_z \omega_x x| \right) dt \\
\text{Subject to } & \ddot{x} = (\omega_y^2 + \omega_z^2)x + u_x \\
& \dot{\omega}_x = \frac{I_2 - I_3}{I_1} \omega_y \omega_z \\
& \dot{\omega}_y = \frac{I_3 - I_1}{I_2} \omega_z \omega_x \\
& \dot{\omega}_z = \frac{I_1 - I_2}{I_3} \omega_x \omega_y \\
& x(t_0) = R_0, \quad \dot{x}(t_0) = 0 \\
& x(t_f) = R_f, \quad \dot{x}(t_f) = 0 \\
& -u_{sat} \leq u_x \leq u_{sat} \\
& t_f \text{ free}
\end{aligned} \tag{3.44}$$

Since no known analytical solutions exist to the Newton-Euler equations, the problem cannot be solved in a similar method as the flat spin or general spin. Instead, the approach that was taken was to assume a constant angular velocity for the remainder of the trajectory and recompute the solution every short period of time compared to the period of the Target's spin. This will, of course, lead to some sub-optimality in the solution compared to solving the full optimal controller problem where $\dot{\omega} \neq 0$ as is the case with a non-spherical inertia. The reason behind this sub-optimality is due to the optimization problem which yields the solutions in Appendix B is different from the general docking problem in Equation (3.44); namely, the cost function in Equation (3.44) includes the values of $\dot{\omega}$ where the cost functions in Appendix B does not. Therefore, a better approximation would be to assume that the angular acceleration is constant for a short period of time, and recompute the problem often. Assuming a constant angular acceleration would improve the solution (akin to an increase in the order of approximation of a function using higher order terms in a Taylor series); however, this analysis was not included here, and is it not known if an analytical solution exists for this case.

3.5.1 Formulation of Limited Horizon Problem Under Constant Angular Velocity Assumption

The availability of an impulsive solution for the general spin with spherical inertia presented in Appendix B was utilized as a method of generating a trajectory for the Chaser to follow for a small amount of time. The impulsive solutions give an optimal trajectory depending on the value γ assuming a constant angular velocity for all remaining time. After a certain time interval, Δt_{guid} , the solution is recomputed with the new estimated angular velocity.

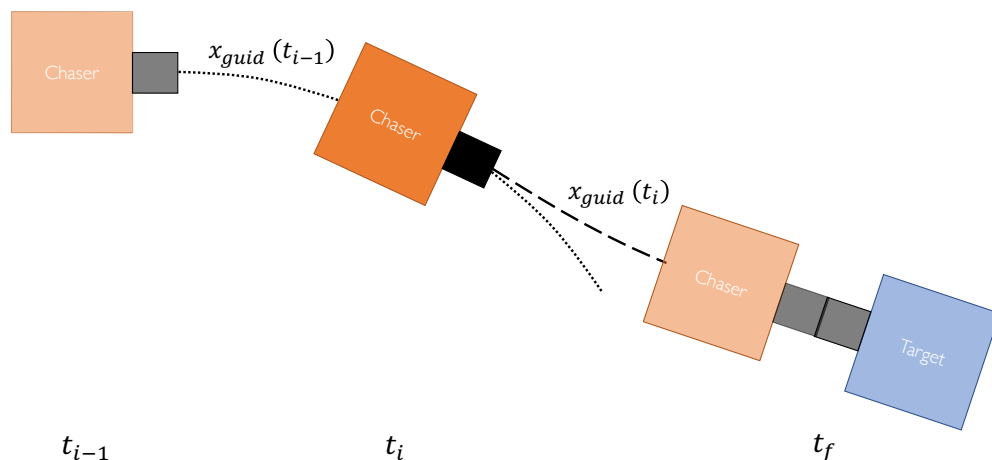


Figure 3-6: Qualitative description for solution to the General Spin General Inertia docking problem

The high level overview of the algorithm is shown in Figure (3-6). At each time step the trajectory recomputes the overall guidance trajectory $\mathbf{x}_{guid}(t)$ using the impulsive solution for the Bang-Off controller in Section B.2.1 or B.2.2 depending on their γ value. However, because of the solution at time t_i assumed a constant velocity for all time, a new optimal trajectory must be regenerated at each controller time step, noted by $\mathbf{x}_{guid}(t_i)$. To follow the guidance solution $\mathbf{x}_{guid}(t_i)$ a closed loop controller with a set point of $\mathbf{x}_{des} = \mathbf{x}_{guid} = [x_{guid}, 0, 0, \dot{x}_{guid}, 0, 0]^T$ is used.

Since the impulsive solutions in B.2.1 and B.2.2 assume that the trajectory starts

at zero relative velocity, $\dot{x}(t_i) = 0$, the impulsive solution will return a desired $\dot{x}_{guid}(t_i)$ that the Chaser must achieve in order to “hop” to the current optimal trajectory through a $\Delta v_x(t_i) = \dot{x}_{guid}(t_i) - \dot{x}(t_i)$. Thus, although the complete time history for the guidance will be continuous on x , the derivatives will have a discontinuity.

Because the Bang-Off-Bang solution is obtained through a minimization problem a measure of the $\Delta v_x(t_i)$ must be included by assuming that the trajectory does not begin at $\dot{x}(t_i) = 0$. This is done by modifying the cost function $f(t_f)$ in Equation (B.13) to include the current velocity,

$$f(t_f) = \int_{t_0}^{t_f} \left(|2\omega_z \dot{x}_{guid}(t) + \omega_y \omega_x| + | - 2\omega_y \dot{x}_{guid}(t) + \omega_z \omega_x| \right) dt + |\dot{x}_{guid}(t_i) - \dot{x}(t_i)| + |\dot{x}_{guid}(t_f)| \quad (3.45)$$

where in Equation (3.45) the cost function now accounts for the effective $\Delta v_x(t_i)$. Note that since the Bang-Off solution is solved fully analytically without an extra degree of freedom, there is no need to modify the solution presented in Equation (B.8).

Plume Impingement Avoidance

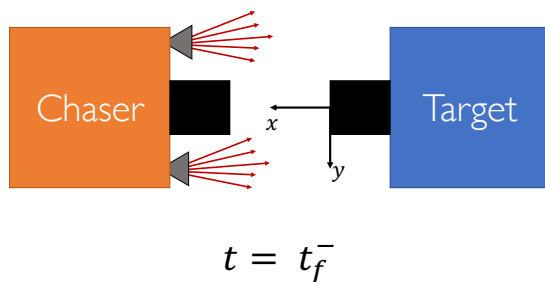


Figure 3-7: Chaser plume impingement on Target due to Bang-Off-Bang controller. t_f^- indicates the instantaneous time before the end of the maneuver when the Chaser is slowing down before achieving soft docking condition.

The nature of the Bang-Off-Bang solution dictates that the thruster pointing along the $-\hat{x}$ direction (which produce a positive thrust on the Chaser) must fire until final

time. This means that the thrusters will fire until the Chaser arrives at the final radius. For the impulsive solution, the result is the same: the Chaser must have an impulsive change in velocity at $t = t_f$. This behavior is shown in Figure (3-7) where t_f^- is the instantaneous time before reaching the final time which decelerates the Chaser to the soft docking condition.

To avoid the dangers of plume impingement, a strategy can be developed in which the Chaser switches from using the Bang-Off-Bang solution to the Bang-Off solution regardless of the value of γ . This essentially creates a trade-off in which the fuel optimality of the solution is weighted against the risk of plume impingement that could lead to the structural damage or even disintegration of the Target.

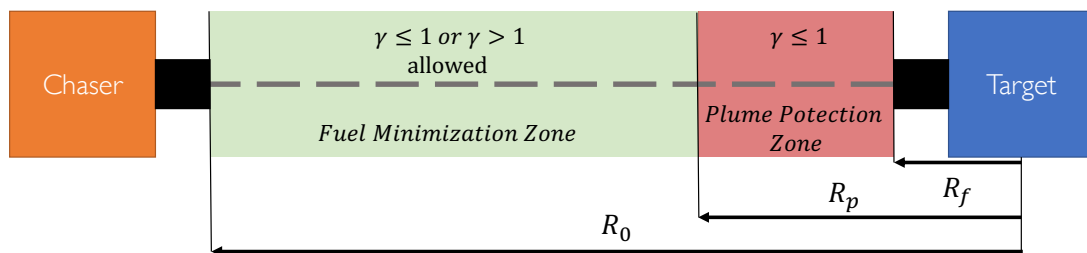


Figure 3-8: Proposed approach for limiting plume impingement in Target. R_p indicates minimum relative distance at which Bang-Off-Bang controller is allowed.

Figure (3-8) shows a proposed solution for limiting the plume impingement in the Target⁴. For all distances between the initial offset R_0 , and a user defined distance R_p , the Chaser will select between the Bang-Off and Bang-Off-Bang controller depending on the value γ . However, once it reaches R_p the Chaser will use $\gamma \leq 1$ and thus the Bang-Off solution for completing the maneuver. This means that even if the current value of ω_{TAR} suggest to use the Bang-Off-Bang solution for minimum fuel, the Chaser will use the safer Bang-Off solution. The choice of R_p is user- or system-defined as various satellites will have different thruster configurations which might increase or decrease the R_p distance.

⁴It should be noted that the definition of all the distances in Figure (3-8) are based on the distance between the center of mass of the Target and the edge of the docking port of the Chaser. This is not a requirement and is mostly done for simplicity of the diagram. It is preferred for the distances to be measured between the center of masses of the two satellites.

Choice of Guidance Refresh Time, Δt_{guid}

The choice of Δt_{guid} or the refresh time for the algorithm is a design variable that has not been considered in detail. Typically having a fast refresh or update rate for a model predictive controller or a global trajectory optimizer is desired. As the dynamics change or errors grow, being able to recompute an optimal solution is beneficial to continue on the fuel optimal path, but is not always feasible in terms of computation time.

Fortunately the solution for the impulsive docking problems are extremely fast to obtain due to their analytical or quasi-analytical forms—the computational time comparison is shown in Chapter 4. Therefore, it is possible to have a high refresh rate for the guidance solution—fast defined as a small percentage (e.g. 5-10%) of the period of the fastest angular velocity, or $t_{fast} \leq \min(2\pi./\omega)$. Since there is no use for obtaining a new guidance solution faster than the rate at which the Chaser can actuate, a recommendation would be to have $\Delta t_{guid} = n\Delta t_{ctrl}$, where n is a number greater or equal than 1. Thus, if the computational resources on-board allow for the solving of a bisection method within each controller sample time, then setting $\Delta t_{guid} = \Delta t_{ctrl}$ would be desired.

3.5.2 Analytical Model Predictive Controller Algorithm (AGA)

The analytical guidance algorithm (AGA) for docking with a tumbling object with a general spin and inertia is shown in Algorithm 1. The function *SolveAG* takes as inputs the current time, t_{cur} , the current position and velocity of the Chaser relative to Target, $x(t_{cur}), \dot{x}(t_{cur})$, the current angular velocity and acceleration of the Target, $\omega(t_{cur}), \dot{\omega}(t_{cur})$, the plume protection radius, R_p , the final desired offset, R_f , and the function tolerance.

For outputs, the function returns the change in relative velocity needed to join the new trajectory, Δv_x , the expected feedforward terms for the controller in the normal directions, u_y^{fwd} and u_z^{fwd} , and the predicted final time t_f . All of these values are relative to the Target's body-fixed frame. The change in velocity Δv_x is then fed into

a closed loop controller, and the values u_y^{fwd} and u_z^{fwd} can be used a feed forward terms to maintain the Chaser aligned along the docking axis.

Algorithm 1 Analytical Guidance Algorithm (AGA)

```

1:  $[\Delta v_x, u_y^{fwd}, u_z^{fwd}, t_f] = \text{SolveAG}(t_{cur}, x(t_{cur}), \dot{x}(t_{cur}), \boldsymbol{\omega}(t_{cur}), \dot{\boldsymbol{\omega}}(t_{cur}), R_p, R_f, tol)$ 
Require:  $|x(t_{cur}) - R_f| \geq tol$ 
2:  $\gamma = (|\omega_y \omega_x| + |\omega_z \omega_x|) / (\omega_y^2 + \omega_z^2)$ 
3: if  $x(t_{cur}) - R_p \geq tol$  then
4:    $[\Delta v_x, t_f] = \text{SolveImpulsiveProb}(t_{cur}, x(t_{cur}), \dot{x}(t_{cur}), \boldsymbol{\omega}(t_{cur}), R_f, \gamma)$ 
5: else
6:    $\gamma = 0$ 
7:    $[\Delta v_x, t_f] = \text{SolveImpulsiveProb}(t_{cur}, x(t_{cur}), \dot{x}(t_{cur}), \boldsymbol{\omega}(t_{cur}), R_f, \gamma)$ 
8: end if
9:  $u_y^{fwd} = 2\omega_z (\Delta v_x + \dot{x}(t_{cur})) + \dot{\omega}_z x(t_{cur}) + \omega_y \omega_x x(t_{cur})$ 
10:  $u_z^{fwd} = -2\omega_y (\Delta v_x + \dot{x}(t_{cur})) - \dot{\omega}_y x(t_{cur}) + \omega_z \omega_x x(t_{cur})$ 

```

The value of γ is always computed to determine which type of controller to use. If the current relative distance between the Chaser and the Target is larger than the plume protection distance, then either the Bang-Off or Bang-Off-Bang controller solutions are found using *SolveImpulsiveProb*. Otherwise, the Bang-Off solution is used by setting $\gamma = 0$ (alternative, any value for γ such that $\gamma < 1$ works too).

Finally, on line 9 and 10 of Algorithm 1, the feed-forward control forces required to maintain the Chaser radially aligned are computed. This ensures that if $|x(t_{cur}) - R_f| \leq tol$ the thrusters will still provide the correct force to be aligned radially in the body frame. The value for the tolerance could be chosen to be equal to the controller deadband.

Algorithm 2 Impulsive Docking Solution Algorithm

```

1:  $[\Delta v_x, t_f] = \text{SolveImpulsiveProb}(t_{cur}, x(t_{cur}), \dot{x}(t_{cur}), \boldsymbol{\omega}(t_{cur}), R_f, \gamma)$ 
2:  $\gamma = (|\omega_y \omega_x| + |\omega_z \omega_x|) / (\omega_y^2 + \omega_z^2)$ 
3: if  $\gamma \leq 1$  then
4:    $t_f$  from Equation (B.7)
5:    $\dot{x}_{guid}(t_{cur})$  from Equation (B.8)
6: else
7:    $t_f$  from solving min problem (B.12)
8:    $\dot{x}_{guid}(t_{cur})$  from Equation (B.14)
9: end if
10:  $\Delta v_x = \dot{x}_{guid}(t_{cur}) - \dot{x}(t_{cur})$ 

```

The function *SolveImpulsive Prob* is shown in Algorithm 2. This function is essentially a wrapper function for the impulsive solution to the Bang-Off or Bang-Off-Bang problem presented in Section B.2.

In conclusion, this section presented the formulation of the algorithm for the general spin, general inertia docking problem. This problem corresponds to the general docking problem presented in Section 3.1.2 and given by Equation (3.1). The solution represents a fully analytical or quasi-analytical solution to an optimization problem that can be formulated as a classical Model Predictive Controller scheme. Furthermore, an additional constraint was placed to eliminate the issue of plume impingement by limiting the form of the solution to one that minimizes firing the Chaser's thruster towards the Target. The values provided by the AGA must be used in conjunction to a closed loop controller; for example, if using an LQR controller the state error is given by $\mathbf{x}_{err} = [0, -y_{cur}, -z_{cur}, \Delta v_x, -\dot{y}_{cur}, -\dot{z}_{cur}]^T$ where $y_{cur}, z_{cur}, \dot{y}_{cur}$ and \dot{z}_{cur} are the current position and velocity of the Chaser relative to the Target, and represent the current misalignment from the docking axis.

Chapter 4

Results and Analysis for the Analytical Guidance Algorithm

Chapter 4 focuses on analyzing the results and behavior from the trajectory optimization through the AGA given in Chapter 3. Section 1 will focus on using GPOPS as a method of verifying the analytical and quasi-analytical solutions. Section 2 will focus on the simulation that was used to validate the results of the AGA controller. Section 3 will then present and analyze the results of the ability to dock with tumbling objects using the AGA algorithm.

4.1 Trajectory Verification through GPOPS-II

For each of the problem statements presented in Chapter 3 an equivalent problem statement was implemented in the commercial optimization software GPOPS-2. GPOPS-2 is an optimization software that uses Legendre-Gauss-Radau quadrature collocation method to solve the user's problem [24]. The solution of the problem (if it exists) satisfies the optimality conditions given in [25]; however, it does not guarantee global optimality. Despite the lack of guarantee of global optimality, there would be an increase in confidence in the solutions proposed in Chapter 3 if GPOPS arrived to

the same solution.¹

For all the cases, unless otherwise stated, the options for GPOPS setup are shown in Table 4.1. The verification steps for each solution are all similar. For the flat spin and general spin with spherical inertia, a nominal case will be shown overlaid with the GPOPs solution. Furthermore, a series of of different examples with a variables changed from that of the nominal case will be presented.

Table 4.1: GPOPS-2 Set up

Description	Value
Mesh Method	HP Petterson Rao
Mesh Tolerance	1e-5
Mesh Max Iterations	50
Max Collocation Points	200
NLP Solver	IPOPT
NLP Tolerance	1e-7
Derivatives	SparseCD

4.1.1 Flat Spin Solution

For the flat spin, a nominal example was generated to compare the analytical solution to that of GPOPS. The parameters for that example are given in Table 4.2. These values do not represent a real system and were only used for testing.

Table 4.2: Flat Spin Nominal Case

Parameter	Value
R_0 [m]	10
R_f [m]	1
ω_z [$^\circ/s$]	10
u_{sat} [m/s ²]	2

The results for the switch time, final time, and overall cost are shown in Table 4.3. The critical time t_1 , is computed analytically through the flat spin solution

¹As with any optimization software GPOPS requires an initial feasible solution as a guess. For all the tests, the guess was a straight line on x that connected the initial radius to the final desired radius. This solution was chosen to not bias GPOPS to pick a solution similar to solutions in Chapter 3.

Table 4.3: Flat Spin Nominal Case Result

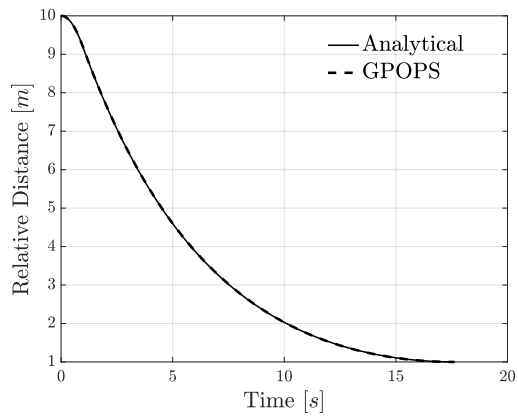
Parameter	Analytical	GPOPS	% Diff.
t_1 [s]	0.9420	0.9482	-0.67
t_f [s]	17.6403	17.6000	+0.23
Cost [m/s]	5.0256	5.0214	+0.08

given in Equation (3.24); however, for GPOPS this value is not given. Instead, the value that was taken as the critical time is the point at which the controller in \mathbf{x} , u_x , reaches $u_{sat}/2$. Thus, it is expected that the value with the largest percent difference corresponds to t_1 .

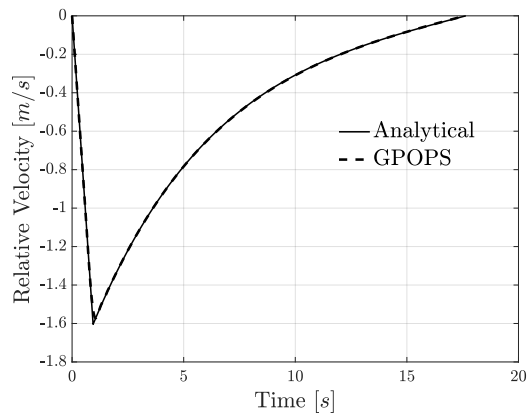
The figure results for this nominal case are presented in Figure (4-1). The relative position and velocity between the Chaser and Target are shown in Figures (4-1a) and (4-1b), respectively. Additionally, the control, costate, and Hamiltonian time history were plotted in Figures (4-1c) (4-1d) and (4-1e). For all these figures, the dashed line represent the results from GPOPS optimization while the solid line represents the solution obtained through the analytical method.

From Figure (4-1a) it is easy to observe that the overall behavior for the trajectory follows that of an exponential curve as expected. The concavity of the solution changes. The two sections of the solution can be seen more clearly in the relative velocity figure. Interestingly, at time $t_1 \approx 0.94$ [s], the solution hits the peak of around $\dot{x} \approx -1.6$ [m/s]; this is the equivalent velocity that would be obtained if instead an impulsive solution (given in Appendix B) was used for the current relative position value at t_1 .

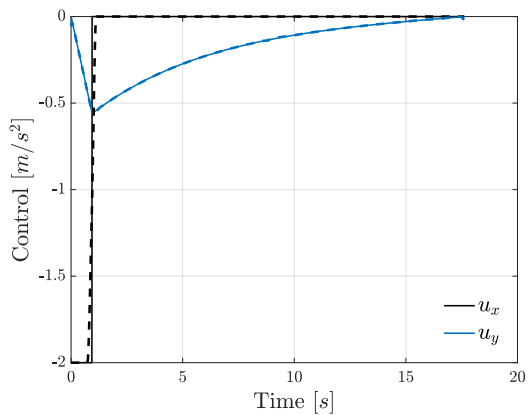
The control and costate vectors also follow the trend of having two different sections between $[0, t_1]$ and $(t_1, t_f]$. As with many numerical methods, GPOPS struggles with discontinuities as seen in Figure (4-1c). At switch time t_1 , the controller according to PMP must switch from $-u_{sat}$ to 0. Since using a collocation must mean that the solution is always continuous and differentiable, as the basis functions are continuous and differentiable, it is very difficult to truly obtain a bang-bang behavior. Nevertheless, GPOPS approximates this by having more collocations around t_1 . Thus having more collocation points as well as higher precision in the NLP solver



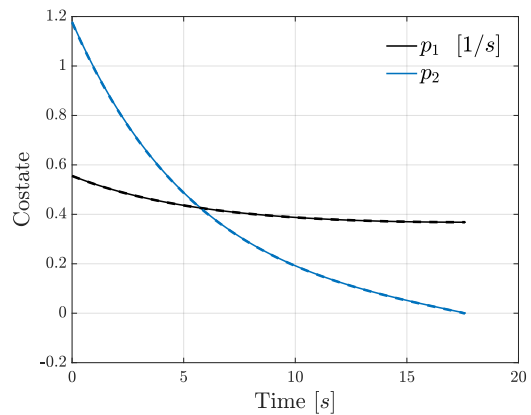
(a) Relative Chaser-Target position



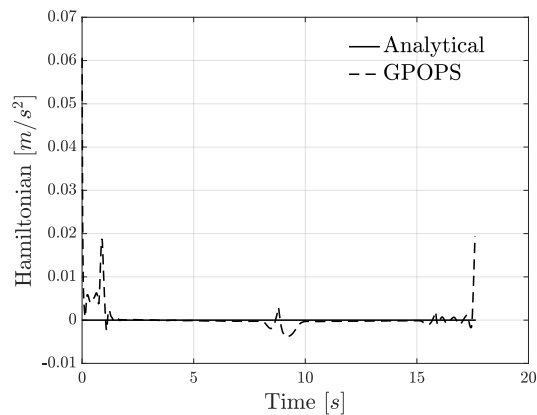
(b) Relative Chaser-Target velocity



(c) Control time history



(d) Costate time history



(e) Hamiltonian time history

Figure 4-1: Comparison between GPOPS and analytical solution to Flat Spin nominal case

will yield a better approximation. Finally, from Figure (4-1d), the analytical costate p_2 matches that of GPOPS in that the controller switches when $p_2 = 1$ and the final value for p_2 is zero.

Lastly, the Hamiltonian was plotted in Figure (4-1e). Per optimality conditions, the Hamiltonian for all time must be equal to zero (since the transversality condition requires the Hamiltonian be zero at the final time). GPOPS confirms this condition as it attempts to maintain the condition that the Hamiltonian - given in Equation (3.6)—is equal to zero. An analysis must be carried out to determine whether the instances in which the Hamiltonian given by GPOPS (the dashed line) do not equal zero are due to numerical error. Nevertheless, since the analytical solution shows the Hamiltonian is equal to zero for all time, there is a greater feeling of certainty that it indeed represents the optimal solution.

Figure (4-2) shows this analysis in which the degree of accuracy for GPOPS was increased to determine if the Hamiltonian approached zero for all time. GPOPS Error is an error reported by the optimization software that gives a measure of the error in the mesh grids. A higher error in the mesh grid will indicate a higher degree of error in the solution given by GPOPS due to constraint violations. Nevertheless, as the error decreases, the Hamiltonian approaches zero for all time. This again not only confirms the optimality conditions are satisfied, but also the analytical solution is accurate.

Table 4.4: Flat Spin Test Matrix

ID	R_0 [m]	R_f [m]	ω_z [$^\circ$/s]	u_{sat} [m/s²]
Case 1	15	1	10	2
Case 2	10	5	10	2
Case 3	10	1	5	2
Case 4	10	1	10	0.5

To confirm that the analytical solution is valid for any choice of R_0 , R_f , ω_z , and u_{sat} , four cases were conducted in which each value was changed from the nominal example. The test matrix is shown in Table 4.4. Again, the values for the test example were chosen randomly and do not represent any physical system. Although

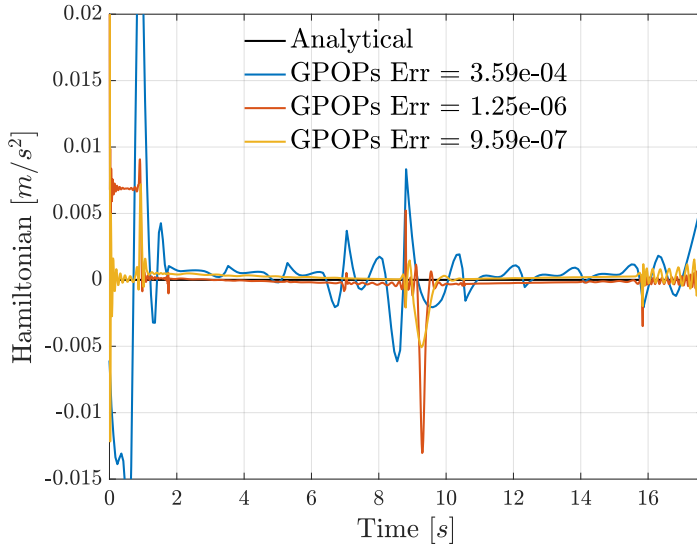


Figure 4-2: Resultant Hamiltonian for nominal Flat Spin Case. Lower GPOPS Error leads to Hamiltonian approach zero for all time.

Table 4.5: Flat Spin Test Result Matrix

ID	t_1 [m]	t_f [m]	Total Cost [m/s]		Computation Time [s]	
			AFS	GPOPS	AFS	GPOPS
Case 1	1.483	20.270	7.853	7.852	2.44e-04	4.921
Case 2	0.820	7.973	3.386	3.385	2.44e-04	0.705
Case 3	0.443	34.523	2.456	2.456	2.81e-04	5.072
Case 4	5.359	20.437	5.821	5.820	2.61e-04	3.964

a Monte Carlo approach could have been sought to validate the analytical solution, there will be a great deal of confidence in the generalizability of the analytical solution if GPOPS agrees with the tests in Table 4.4.

Figure (4-3) shows the different resultant trajectories for the cases given in Table 4.4. Figure (4-3a) shows the relative distance between the Chaser and Target obtained by the analytical solution and GPOPS. For all cases, the trajectory produced by the analytical solution and the GPOPS solution match closely. Figure (4-3b) shows the differences between the two solutions. The jump in all the plots are at the critical time in which the Chaser's thruster in the radial direction shuts off.

The detailed results for the tests are shown in Table 4.5, noting that the results for the nominal case are in Table 4.3. Also, note that AFS was a chosen abbreviation

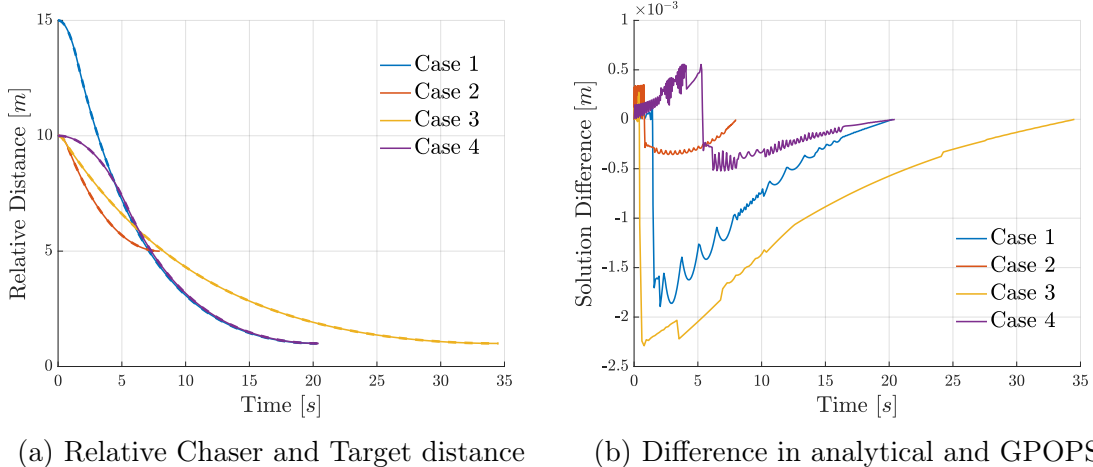


Figure 4-3: Comparison of analytical and GPOPS solution for different flat spin scenarios

of the Analytical Flat Spin (AFS) solution. As expected, Case 1 lasts approximately a third of the time longer than the nominal case due to the increase in initial distance between the Chaser and Target. This case also follows an increase in cost, as expected. Conversely, Case 2 lasts approximately half the time than that of the nominal case as the final desired distance is increased from one meter to five. Case 3 has a decrease in the critical time and increase in final time, as expected, as the Target moves slower; this leads to a decrease in cost as well as the amount of effort required to maintain the Chaser aligned with the Target as the rotating frame decreases. Finally, by decreasing the saturation value for the Chaser's thrusters, the fuel cost as well as the time increases. This is expected as the fuel cost for an impulsive thruster (or one that approaches the impulsive case) should decrease compared to one that has limited saturation.

Additionally, on Table 4.5 the computation time is included as a measure of computational complexity. Provided that both the analytical solution and the GPOPS optimization are done on the same computer, the computation time reported can be thought as a measure of number of operations. Nevertheless, no algorithmic analyses were done for the flat spin solution. A trend that is clear is that although the analytical solution appears to solve each case in about the same amount of time, GPOPS solution time is heavily dependent on the case. For example, cases in which there

is a low angular velocity (relative to other cases) will spend an increased amount of time computing the GPOPS solution. This shows a major benefit of the analytical solution: the computational complexity is *independent* of the parameters as the same operations are applied.

4.1.2 General Spin Spherical Inertia Solution, Bang-Off ($\gamma \leq 1$)

The same procedure was done to test the general spin solution. Due to the two forms of the solution depending on the value γ , the general spin spherical inertia case was split. Nevertheless, the same problem formulation in Equation (3.27) was used for both cases. The nominal example used for the Bang-Off controller case is given in Table 4.6. The value for γ in this example is given by

$$\gamma = \frac{|\omega_y \omega_x| + |\omega_x \omega_z|}{\omega_y^2 + \omega_z^2} = 0.2 \leq 1$$

which is consistent with a Bang-Off controller.

Table 4.6: General Spin Spherical Inertia, $\gamma \leq 1$ Nominal Case

Parameter	Value
R_0 [m]	10
R_f [m]	1
$\boldsymbol{\omega}$ [$^\circ/s$]	$[2, 10, 10]^T$
u_{sat} [m/s ²]	2
I_{TAR} [kg m ²]	$diag([1, 1, 1])$

Table 4.7: General Spin Spherical Inertia, $\gamma \leq 1$ Nominal Case Result

Parameter	Analytical	GPOPS	% Diff.
t_1 [s]	1.4646	1.4644	+0.0106
t_f [s]	12.9252	12.8224	+0.8024
Cost [m/s]	9.2202	9.1889	+0.3406
Comp. Time [s]	0.007	471.988	-99.999

Table 4.7 shows the results for the nominal case. Overall, the critical time t_1 , final time t_f , and cost match very closely. Their differences are most likely due to numerical errors from GPOPS. However, the computation time for the analytical case

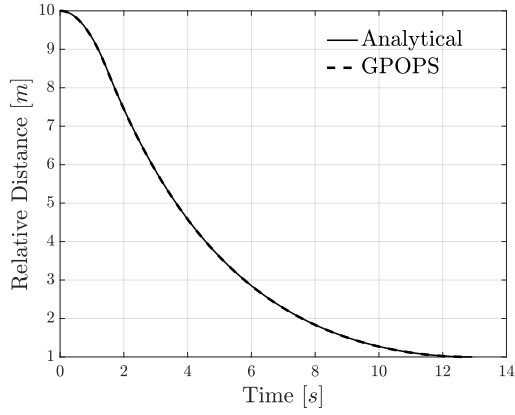
is nearly 5 orders of magnitude less than GPOPS. This is consistent with the idea that having an analytical solution not only gives more insight into the dynamics and solution of the problem but is also more computationally efficient.

Figure (4-4) shows the result of comparing GPOPS solution for the general spin spherical inertia to that of the analytical solution found in Section 3.4. As in the case for the flat spin, the relative position, relative velocity, control, costate, and Hamiltonian time history were plotted in Figures (4-4a-e), respectively. In a similar manner with the flat spin case, the trajectory for the relative distance between the Chaser and the Target changes concavity at the critical time t_1 . In the same way the maximum relative velocity seen in Figure (4-4b) is equivalent to the impulsive solution at the respective relative distance at t_1 . The costate vector behaves as expected in that it turns off the thruster value when p_2 hits 1. It can be seen in Figures (4-4c) and (4-4e) that GPOPS has numerical issues at time equal zero as well as the critical time. This can be improved by increasing the accuracy of GPOPS as seen in Figure (4-2); however, that will come at the cost of higher computational time.

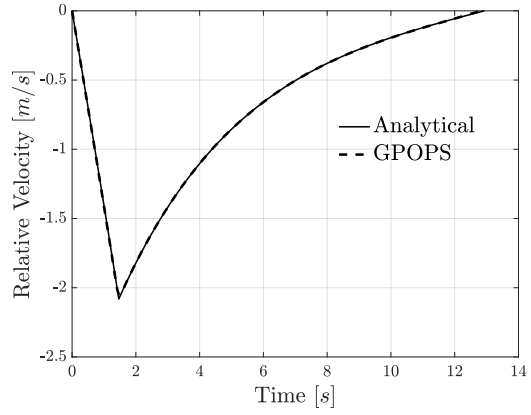
For this nominal problem, the periodicity of the motion of the docking port of the Target is given by $2\pi/|\boldsymbol{\omega}| = 25.205$ [s]. In other words, on an inertial frame, the docking port will face the same direction every 25.205 seconds. The final time $t_f = 12.9252$ [s] then corresponds to about half of this period. This makes intuitive sense as the Chaser should be able to finish a trajectory within one period; otherwise the Chaser will waste too much fuel following the Target on the rotating frame. This result is consistent with the same claim made by Hettrick [14] and Sternberg [35] in their theses.

4.1.3 General Spin Spherical Inertia Solution, Bang-Off-Bang ($\gamma > 1$)

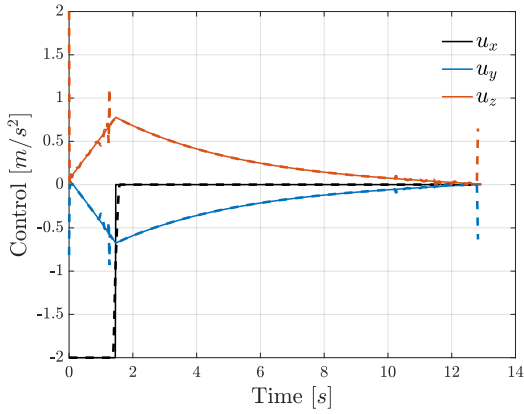
Lastly, the solution to the general spin, spherical inertia docking problem occurs when $\gamma > 1$. To check this case, the tests described in Table 4.8 were used to compare the solution with that of GPOPS. The angular velocity in the nominal case has a



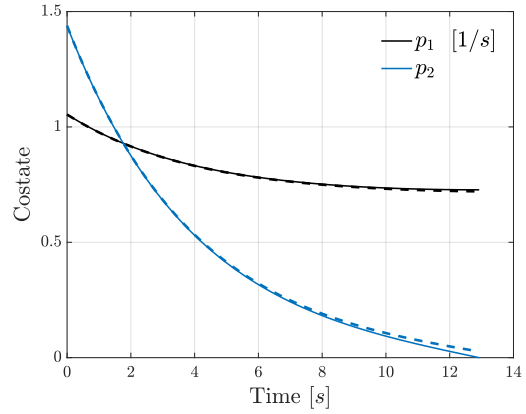
(a) Relative Chaser-Target position



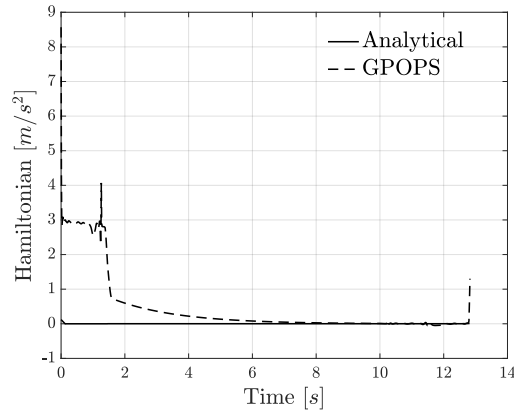
(b) Relative Chaser-Target velocity



(c) Control time history



(d) Costate time history



(e) Hamiltonian time history

Figure 4-4: Comparison between GPOPS and analytical solution to General Spin with Spherical Inertia $\gamma \leq 1$

corresponding value of $\gamma = 2$. The result for this case is shown in Table 4.9. Whereas the flat spin and the case in which $\gamma \leq 1$ have a single switch time, the case with $\gamma > 1$ requires two switch times. Note that the difference in switch time t_2 , and final time t_f , is very small and can lead to numerical error due to their sensitivity. Furthermore, despite the magnitude of the angular velocity being higher than the case for $\gamma \leq 1$, the overall cost is very similar to the other case. In fact if the solution for the Bang-Off case were used in this problem (despite the value for γ indicating a need to use the Bang-Off-Bang solution), the cost would be 12.3332 m/s which is a 38% increase penalty for using the less optimal solution for this case.

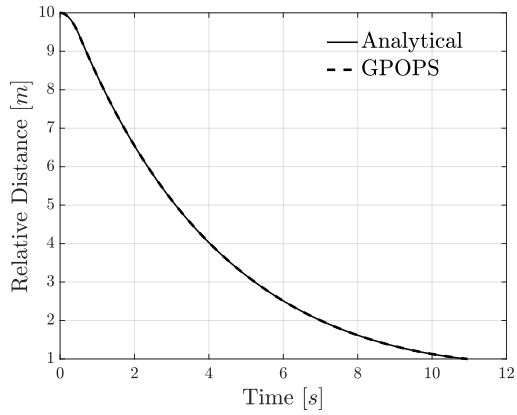
Table 4.8: General Spin Spherical Inertia, $\gamma > 1$ Nominal Case

Parameter	Value
R_0 [m]	10
R_f [m]	1
ω [$^\circ/s$]	$[20, 10, 10]^T$
u_{sat} [m/s^2]	5
I_{TAR} [$kg\ m^2$]	$diag([1, 1, 1])$

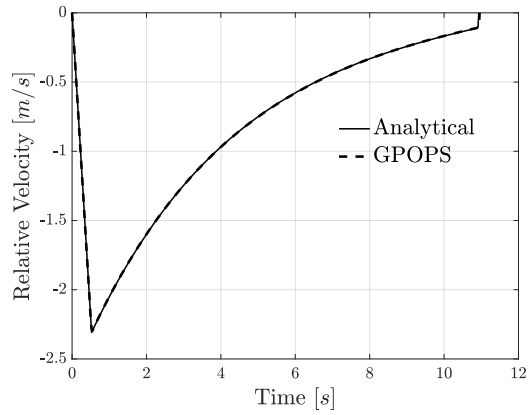
Table 4.9: General Spin Spherical Inertia, $\gamma > 1$ Nominal Case Result

Parameter	Analytical	GPOPS	% Diff.
t_1 [s]	0.5243	0.5267	-0.4568
t_2 [s]	10.9109	10.9397	-0.2637
t_f [s]	10.9325	10.9554	-0.2091
Cost [m/s]	9.2887	9.2700	+0.2017
Comp. Time [s]	0.008	41.7162	-99.9808

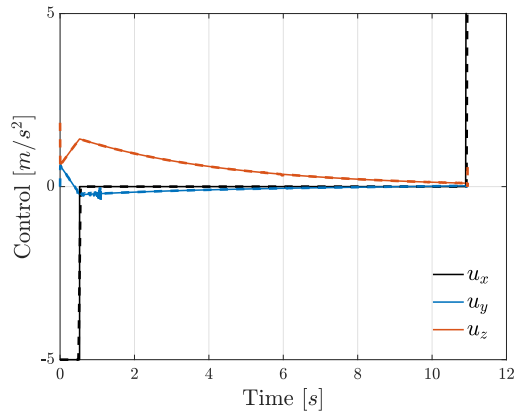
Figure (4-5) shows the same plots as in the case for $\gamma \leq 1$. There are very subtle differences between the plots in Figures (4-5) and (4-4). For example Figure (4-5b) which shows that, relative, to the Target, the Chaser has a large jump in the velocity near the final time. This is compared to Figure (4-4b) in which the velocity plot gently approaches zero. This new behavior is consistent with the Bang-Off-Bang response. Furthermore, the controller behavior shown in Figure (4-5c) is as expected, with thrusting at saturation in the $+x$ direction during the last few seconds. Note that this would lead to thruster impingement onto the Target. The



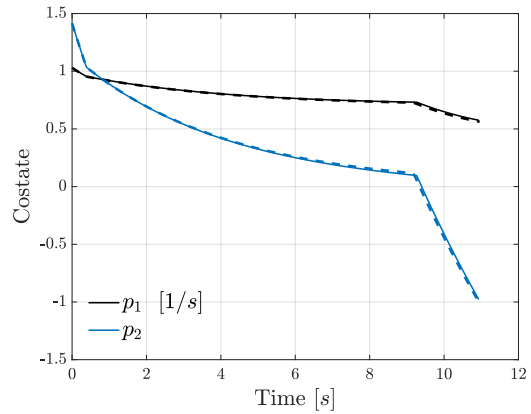
(a) Relative Chaser-Target position



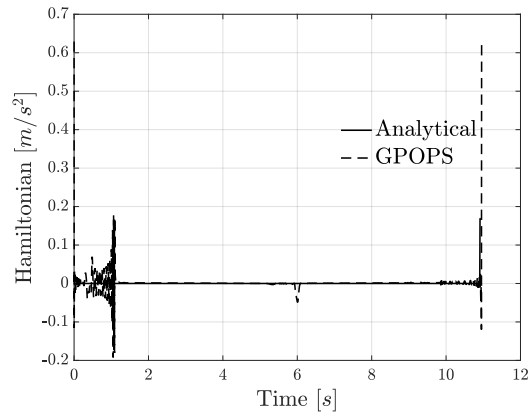
(b) Relative Chaser-Target velocity



(c) Control time history



(d) Costate time history



(e) Hamiltonian time history

Figure 4-5: Comparison between GPOPS and analytical solution to General Spin with Spherical Inertia $\gamma > 1$

piecewise behavior of the costate shown in Figure (4-5d) arises because the σ_1 and σ_2 value given by Equation (3.30) have discontinuities. Finally, as with the other cases, the Hamiltonian is always zero (within numerical precision) in the analytical solution but not in the solution given by GPOPS.

Thus, this shows that the analytical solution for the flat spin as well as for the general spin with spherical inertia are consistent with the results from GPOPS. This gives a greater confidence in the validity of the solution presented in Chapter 3.

4.2 Simulation Environment for Analytical Guidance Algorithm Testing

This section goes over the simulation environment that was used to test the trajectory optimizer developed in Chapter 3. The overall assumptions, dynamics, controller, and guidance algorithm are presented here, and the limitations of the simulation are also addressed.

4.2.1 Assumptions for Simulation Environment

Since overall goal of the simulation is to test the viability of the analytical guidance algorithm given in Algorithm 1, the simulation will have the following assumptions:

1. Only the translational dynamics will be considered
2. No measurement or process noise
3. Thrusters will be assumed to be ideal
4. No Keplerian dynamics

The first assumption essentially states that no attitude dynamics will be considered. Because the solution given in Chapter 3 only focuses on the relative vector between the Chaser and Target, only those dynamics will be considered. Otherwise, the simulation would be more complex, as the Chaser must compute the estimated

quaternion and do a frame translation from its orientation to that of the Target. Furthermore, a thruster mapping matrix would also need to be constructed, which adds another level of complexity in the simulation. Since all of these steps from modeling a 6 DOF spacecraft [38, 39, 26], quaternion error and attitude control [40, 41], as well as optimal thruster mapping have been extensively studied [42], this simulation will not focus on that, but it can be included in future work to demonstrate a higher fidelity model of how a Chaser satellite would behave.

The lack of measurement of process noise as well as having ideal thrusters were assumed. Because the simulation is only testing that the analytical guidance will give a convergent solution that minimizes fuel, it is believed that adding other complexities such as noise would detract from the performance of the algorithm. Nevertheless, these are important parts of any simulation and should be included in future work.

Finally, since the time scales for the guidance solution are very small compared to a period of a GEO orbit, the Keplerian dynamics are ignored. This assumption is valid because the relative distance between the two spacecraft are in the order of tens of meters, at which it is a reasonable assumption that the spacecraft are behaving as simple double integrators on inertial space. In a real scenario the small relative effects due to Keplerian dynamics would be removed by the controllers and treated as disturbance forces.

4.2.2 High Level Simulation Overview

Figure (4-6) shows the highest level of the Simulink model for the analytical guidance testing. Simulink was chosen as the simulation environment as it allows for a fast modification of subsystems and a high level of flexibility in integration and signal routing. The simulation consists of five main subsystems (three model related, and two data processing related): **Guidance**, **Control**, **Dynamics**, Saving, and Plotting.

Although not explicitly shown, the signal with the state information coming out of the integrator in the dynamics block gets sent back to the guidance and control block using a Go To Simulink block. Thus, the model shown in Figure (4-6) is a closed loop system.

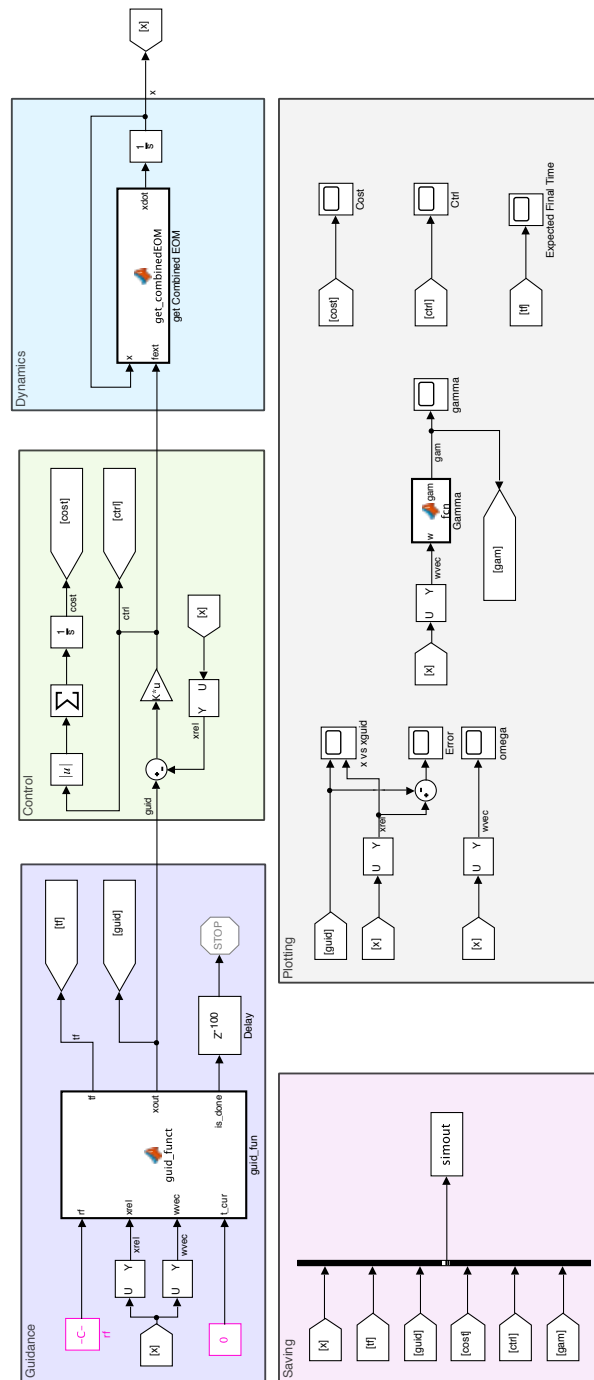


Figure 4-6: High level overview of Simulink model for testing AGA controller.

4.2.3 Dynamics of Analytical Guidance Simulink Model

The dynamics of the simulation can be divided into two parts: dynamics of the Target's angular velocity and dynamics of position of the Chaser relative to the Target. The angular velocity of the Target is obtained by the solution to Equation (3.2) with the exclusion of external torques

$$\frac{d}{dt} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} \frac{I_y - I_z}{I_x} \omega_y \omega_z \\ \frac{I_z - I_x}{I_y} \omega_z \omega_x \\ \frac{I_x - I_y}{I_z} \omega_x \omega_y \end{bmatrix} \quad (4.1)$$

Equation (4.1) are simplified Newton-Euler equations for an object acting under no disturbances nor external torques. The equations assume that the coordinate system is aligned to the Target body-fixed reference frame, and it is assumed without loss of generality that the Target body-fixed frame is also aligned with the Target's principal axes.

The position of the Chaser relative to the Target in the Target's body frame is given by

$$\ddot{\mathbf{r}} = \mathbf{u} - 2\boldsymbol{\omega} \times \dot{\mathbf{r}} - \dot{\boldsymbol{\omega}} \times \mathbf{r} - \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}) \quad (4.2)$$

where \mathbf{r} is the relative position in the Target's body frame, $\boldsymbol{\omega}$ is the Target's angular velocity, and \mathbf{u} is the force per unit mass applied on the Chaser.

The full state vector integrated in the dynamics block is then $\mathbf{x} = [\mathbf{r}, \boldsymbol{\omega}]^T$ with the equation of motion,

$$\dot{\mathbf{x}} = f(\mathbf{x}) \quad (4.3)$$

where $f(\mathbf{x}) = [\dot{\mathbf{r}}, \dot{\boldsymbol{\omega}}]^T$ from Equation (4.1) and (4.2).

4.2.4 Guidance of AGA Simulink Model

The guidance subsystem of the Simulink model shown in Figure (4-6) is the algorithm shown in Algorithm 1. The only difference is that once the relative position between the Chaser and Target is below the tolerance level, the Chaser will be commanded to stay in that position, $\mathbf{x}_{guid} = [R_f, 0, 0, 0, 0, 0]^T$. The guidance algorithm will send a flag that the maneuver is achieved through a delay block (shown by the Simulink Delay block) for 500 control cycles equivalent to five seconds. After this, the Simulink model will terminate.

4.2.5 Control of Analytical Guidance Simulink Model

The controller subsystem of the Simulink model was chosen as a simple Linear Quadratic Regulator (LQR) controller. This creates a couple of challenges as generating the K matrix that solves the Riccati equation needs a model of the dynamics. Thus, this gives a freedom of choice of whether the LQR should be a time varying LQR or not. Although the derivation for both controllers will be presented in this section, the time invariant version will be used on the model.

Linear Time Invariant (LTI) LQR Controller

The LTI LQR controller is one of the simplest controllers that can be set up for a Simulink Model. The objective of designing of the controller is to minimize a cost function for all time, which is given by

$$J = \int_0^{t_f} \mathbf{x}^T Q \mathbf{x} + \mathbf{u}^T R \mathbf{u} dt \quad (4.4)$$

where Q, R are positive definite matrices that weight the state and control, respectively. Additionally, the dynamics for the state are given by

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u} \quad (4.5)$$

where A, B are matrices that define the dynamics of the problem. Since the dynamics for the simulation given by Equation (4.2) are time invariant due to the possible time change in ω , the best representation is that of a double integrator,

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix} \mathbf{x} + \begin{bmatrix} \mathbf{0}_{3 \times 3} \\ \mathbf{I}_{3 \times 3} \end{bmatrix} \mathbf{u} \quad (4.6)$$

where $\mathbf{I}_{3 \times 3}, \mathbf{0}_{3 \times 3}$ represent a 3 by 3 identity or zero matrix, respectively. This is the most flexible state as having a wrong measure of the current angular velocity could lead to a controller with wrong gains.

The minimization problem can be transformed into a solution of the Riccati equation given by

$$0 = A^T P + P A + Q - P B R^{-1} B^T P \quad (4.7)$$

where P is a symmetric and positive semi-definite matrix. Solving by matrix P can then be used to obtain the K_{lqr} gain matrix as

$$K_{lqr} = R^{-1} B^T P \quad (4.8)$$

For example, when $Q = \mathbf{I}_{6 \times 6}$, $R = \mathbf{I}_{3 \times 3}$, the resultant K_{lqr} matrix yields

$$K_{lqr} = \begin{bmatrix} 1 & 0 & 0 & 1.7321 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1.7321 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1.7321 \end{bmatrix} \quad (4.9)$$

where K_{lqr} was obtained using Matlab's *lqr.m* function. This will be the matrix that will be used for the AGA Simulink Model.

Linear Time Variant (LTV) LQR Controller

The linear time varying LQR controller accounts for the fact that the dynamics in Equation (4.2) are changing due to the change in ω . At each time step, the matrices $A(t)$ and $B(t)$ as well as the corresponding gain matrix $K_{lqr}(t)$ must be recomputed. This can be an extra computational burden by having to solve a Riccati equation

each time the controller must act, or by having a tabel lookup of various K_{lqr} gain matrix for each value of $\boldsymbol{\omega}, \dot{\boldsymbol{\omega}}$.

At each iteration the matrix A is calculated by

$$A = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \\ -2[\boldsymbol{\omega}]_{\#} & -[\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \boldsymbol{\omega}]_{\#} \end{bmatrix} \quad (4.10)$$

where $\boldsymbol{\omega}, \dot{\boldsymbol{\omega}}$ are the current best estimates for the Target's angular velocity and acceleration, respectively. Furthermore, $[a]_{\#}$ is a 3 by 3 skew symmetric matrix representing a cross product and is given by

$$[a]_{\#} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \quad (4.11)$$

The matrix B remains the same for all time. Thus, after computing A , the next step is to solve the Riccati to obtain P and then K_{lqr} . Due to the larger overburden of computing the A matrix and the K_{lqr} gain matrix, the Simulink model used the simple double integrator LQR controller.

4.3 Results for Analytical Guidance Simulation

This section focuses on testing the analytical guidance with a controller in the Simulink model. The objectives for this section are as follows

- The AGA solution leads to a convergent solution in which the final position of the Chaser relative to the Target is $R_f \pm tol$
- Under a Target with spherical inertia, the AGA controller reduces to the general spin spherical inertia solution
- The inclusion of the plume protection zone, R_p , leads to a significant reduction thrust being applied on the $+x$ direction

- The cost of the solution for the AGA controller is comparable to that given by GPOPS

Five simulations were performed to check the first three objectives and the values used for the simulation are shown in Table 4.10. Test 1 and Test 2 ensure that depending on the value of γ the analytical guidance should return the same solution as the general spin with spherical inertia. Test 3 will check that the algorithm converges to a feasible solution when the inertia is not spherical. Finally, Tests 4 and 5 will study how the inclusion of the plume protection zone affects the thrust in the $+x$ direction.

Table 4.10: AGA Simulation Testing Matrix

Value	Test 1	Test 2	Test 3	Test 4	Test 5
R_0 [m]	12	10	12	10	10
R_f [m]	0.5	1	0.5	1	1
R_p [m]	0	0	0	0	3
$\boldsymbol{\omega}(t = 0)$ [$^\circ/s$]	$[-2, 5, 7]^T$	$[20, 10, 10]^T$	$[-2, 5, 7]^T$	$[9, 5, 3]^T$	$[9, 5, 3]^T$
$[I_x, I_y, I_z]_{TAR}$ [$kg\ m^2$]	$[1, 1, 1]$	$[1, 1, 1]$	$[1, 2, 3]$	$[1, 2, 3]$	$[1, 2, 3]$
tol [mm]	1	1	1	1	1
u_{sat} [m/s^2]	2	5	2	2	2

Figure (4-7) shows the five tests on the surface that determines which type of controller to use. This confirms that Tests 1 and 3 will test the impulsive Bang-Off solution of the trajectory optimizer, while Test 2, 4 and 5 will test the impulsive Bang-Off-Bang. Test 5 also tests the plume protection option for Test 5. Due to the spherical inertias for Tests 1 and 2, the time history of the angular velocity is only a single point.

Test 1: General Spin Spherical Inertia Solution from Analytical Guidance, $\gamma \leq 1$

The results for Test 1 are shown in Figure (4-8). The position and velocity of the Chaser relative to the Target are shown in Figure (4-8a,b), respectively. The corresponding error of the simulation output versus the guidance are shown in Figure

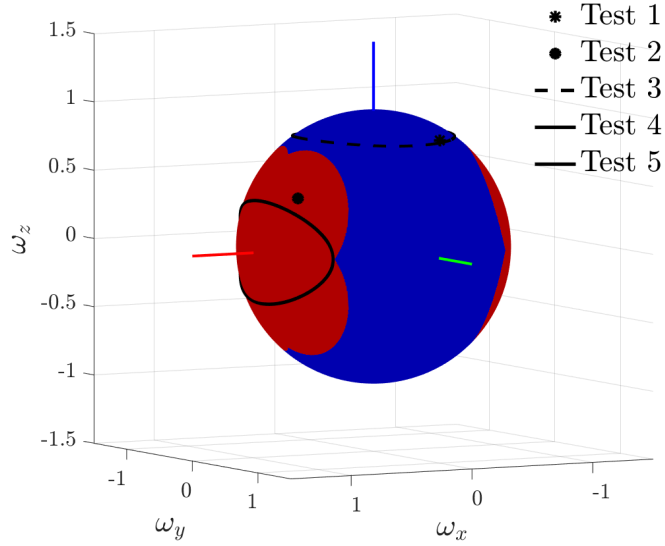
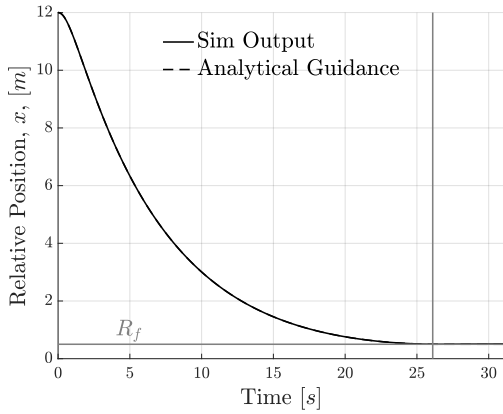


Figure 4-7: Visualization of AGA Tests’s angular velocity history vs. Bang-Off/Bang-Off-Bang regions. Blue region corresponds to $\gamma \leq 1$ or the impulsive Bang-Off controller, red region corresponds to impulsive Bang-Off-Bang region.

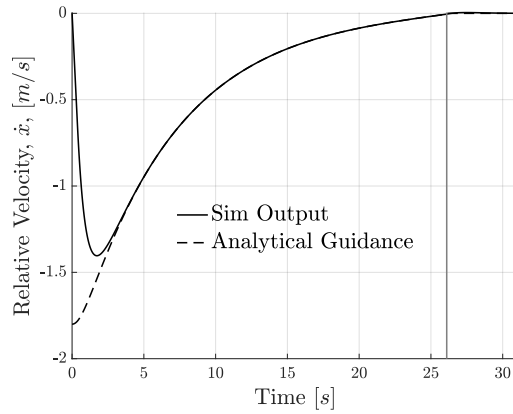
(4-8c,d). Finally the control time history and the cumulative cost, which is the integral of the sum of the norms of control effort, is shown in Figure (4-8e,f).

For all the parts of Figure (4-8) the vertical grey line indicates when the guidance solution by the AGA finished (or the current relative position of the Chaser is $R_f + tol$) at which point the guidance solution switches to just maintain that relative position for all time. In other words, the guidance command is $[\mathbf{r}, \dot{\mathbf{r}}]^T = [R_f, 0, 0, 0, 0, 0]^T$. The guidance solution of main importance is given in figure (4-8a,b) as they show the solution given by the analytical guidance algorithm.

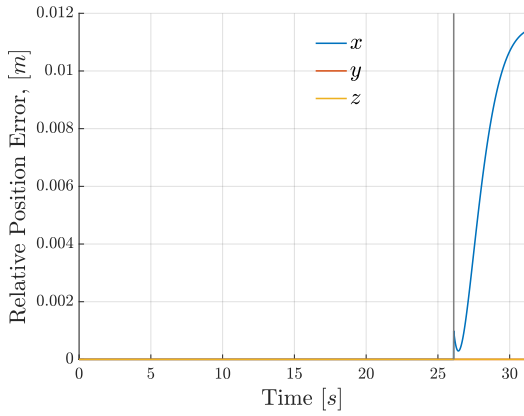
For the position and velocity error figures, the dominant term is on the x axis as expected, as the Chaser begins radially aligned with $y = z = 0, \dot{y} = \dot{z} = 0$. Throughout the trajectory given by the analytical guidance, the position error (Figure (4-8c)) on x is zero by definition since at each control time step the guidance trajectory is recomputed *given* the current position. As soon as the trajectory ends, the relative position error on x increases, which is due to the guidance given to the controller switch from a smooth trajectory to a steady command at zero speed. This behavior can be improved by reducing the tolerance of the error as when $x(t) = R_f + tol \dot{x} \neq 0$. The relative velocity commanded at the time when the guidance switches to the hold



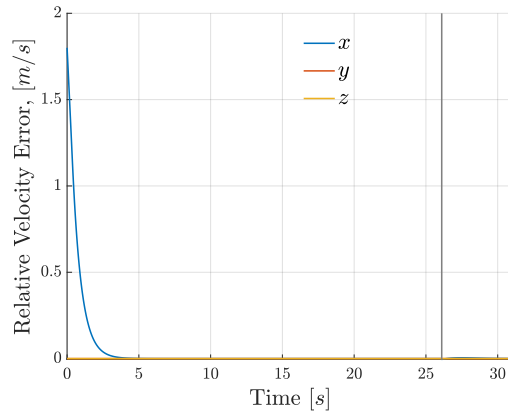
(a) Relative Chaser-Target position, x



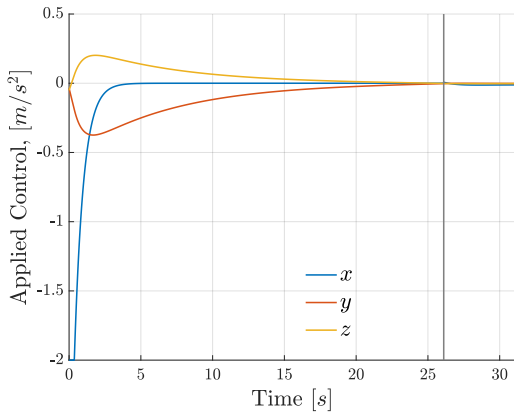
(b) Relative Chaser-Target velocity, \dot{x}



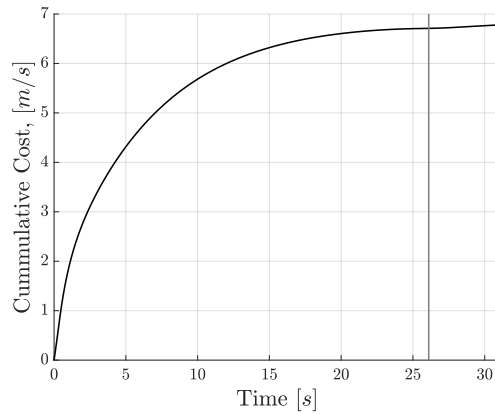
(c) Guidance position error



(d) Guidance velocity error



(e) Control time history



(f) Cummulative cost time history

Figure 4-8: Test 1: Results for analytical guidance simulation for spherical inertia with angular velocity such that $\gamma \leq 1$. Vertical grey bar represents end of guidance solution by AGA.

value is around 4 [mm/s].

The relative velocity error begins with a large error on \dot{x} due to the Chaser beginning at zero relative speed and the analytical guidance. The error approaches zero and around time $t = 4$ [s], the difference between the guidance and sim output is [mm/s]. During the position hold, the error in the relative speed increases to 6 [mm/s] but appears to converge as time goes on.

Figure (4-8e) shows the control time history. As expected due to the impulsive relative velocity from the guidance, the controller saturates at 2 [m/s²] at time $t = 0$. Once the relative velocity matches that of the guidance around four seconds the main contributors become the force in y and z . Again, the control forces are mainly those that are being fed forward by the AGA, and the LQR is used to reduced the error between the guidance in x and maintaining y and z near zero. During the holding section of the trajectory, the main controller present is on the x axis and is negative which means that the plume is fired behind the Target. This behavior is shown in Figure (4-9): as soon as the analytical guidance switches to hold there is a small firing towards the Target to remove the 4 [mm/s] velocity, and then it proceeds to hold the current position by firing in the negative direction. This is due to the natural dynamics pushing the spacecraft away from the Target when in a rotating frame.

Finally, the cumulative cost shown in Figure (4-8f) appears to converge to a steady state until the holding section, at which it begins to increase at a slower rate than before. This is consistent with the rest of the figure, as holding at R_f is cheaper than when $x(t) > R_f$.

Table 4.11: Test 1: Difference between AGA simulation and solution to general spin spherical inertia problem

Value	AGA	Solution from Section 3.4	% Diff
t_f [m]	26.100	26.2823	-0.6936
Cost [m/s]	6.7095	6.6577	+0.7780

In order to check if the solution given by this simulation including the cost and final time, matches that of the general spin spherical inertia solution from Section 3.4.3,

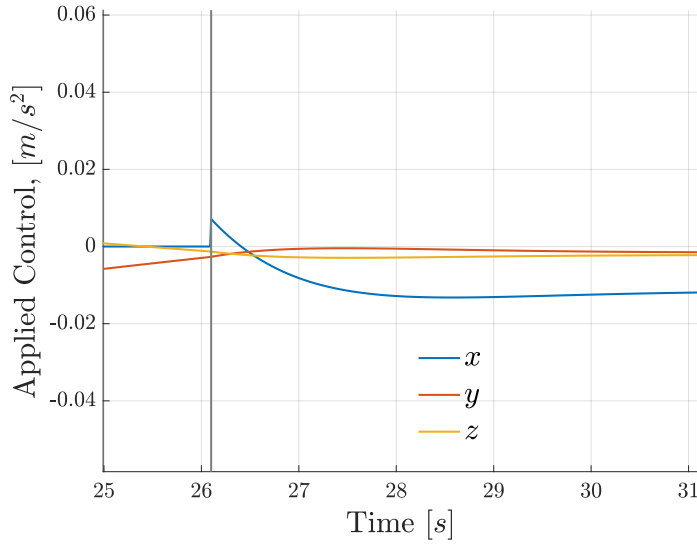
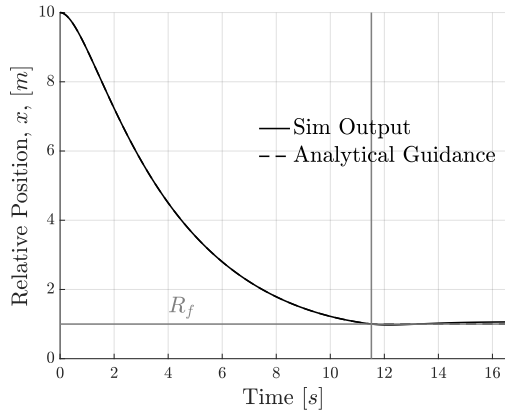


Figure 4-9: Visualization of AGA Tests’s angular velocity history vs. Bang-Off/Bang-Off-Bang regions. Blue region corresponds to $\gamma \leq 1$ or the impulsive Bang-Off controller, red region corresponds to impulsive Bang-Off-Bang region.

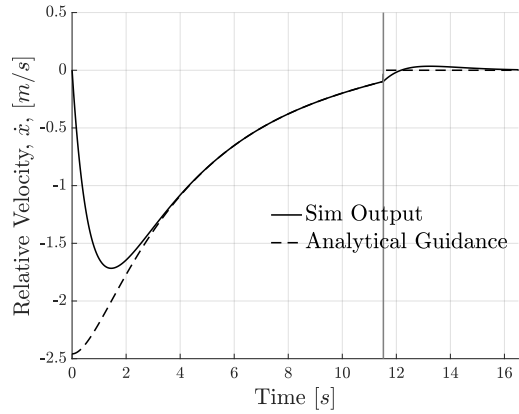
the same problem given in Test 1 was implemented on the general spin solution with spherical inertia. The results are shown in Table 4.11. Given that the AGA simulation ended when the difference in the relative distance was 1 *mm* from the desired R_f it seems reasonable that the final time reported by AGA is smaller. Nevertheless, the cost difference between the two solutions is negligible. Thus the solution of the general spin with spherical inertia when $\gamma \leq 1$ is recovered from the analytical guidance algorithm as expected, confirming the first and part of the second objective of the this section.

Test 2: General Spin Spherical Inertia Solution from Analytical Guidance Algorithm, $\gamma > 1$

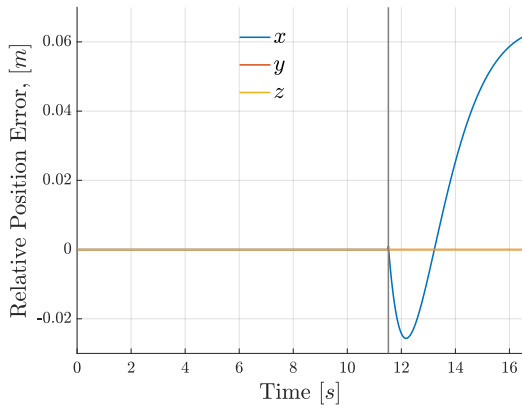
The second test for the Simulink model tested if the AGA solution return a solution similar to the one observed in Section 3.4.4. The same plots as in Test 1 are shown in Figure (4-10). The behavior of the Bang-Off-Bang controller can clearly be seen by Figure (4-10e): before the start of the holding period a control effort of about 160 [*mm/s*²] is exerted, which might cause plume impingement (this compares to



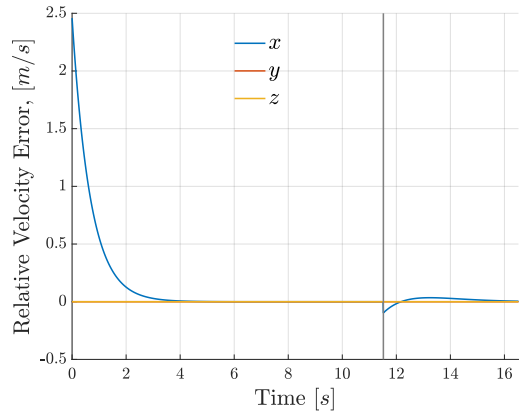
(a) Relative Chaser-Target position, x



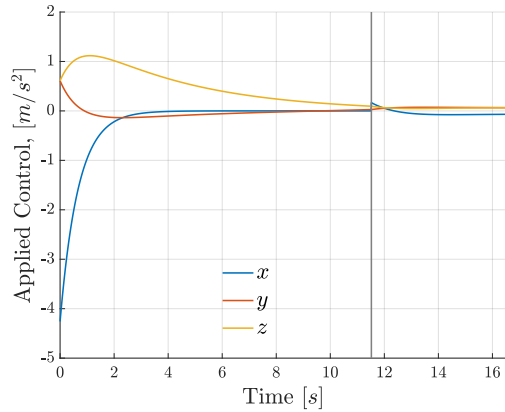
(b) Relative Chaser-Target velocity, \dot{x}



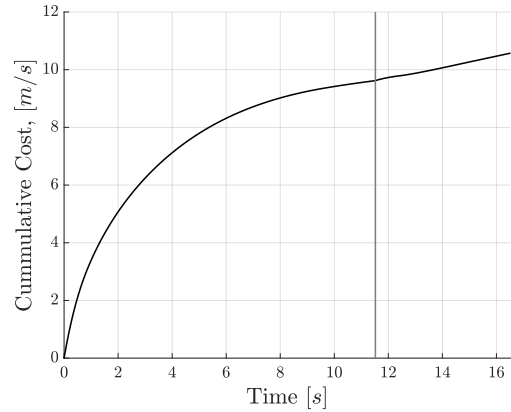
(c) Guidance position error



(d) Guidance velocity error



(e) Control time history



(f) Cummulative cost time history

Figure 4-10: Test 2: Results for analytical guidance simulation for spherical inertia with angular velocity such that $\gamma > 1$. Vertical grey bar represents end of guidance solution by AGA.

about $4[mm/s^2]$ in the Bang-Off case).

The other significant difference is that, unlike the Bang-Off case, at the instant at which the relative distance between the two satellites are within tolerance the speed is not zero, shown in Figure (4-10b). Nevertheless, after the settling the controller appears to be firing in the negative direction to hold attitude, as seen in the last few couple of seconds in Figure (4-10e). Note that the controller this time does not reach saturation limit.

Table 4.12: Test 2: Difference between AGA simulation and solution to general spin spherical inertia problem

Value	AGA	Solution from Section 3.4	% Diff
t_f [m]	11.5200	10.9325	+5.3739
Cost [m/s]	9.6212	9.2202	+4.3493

As with Test 1 the same problem was implemented using the Bang-Off-Bang solution and the results for comparison are shown in Table 4.12. Although not exactly the same, the solutions from the AGA closely resemble those in Figure (4-5). The differences in the cost and final time can be due to a couple of issues but mainly the LQR controller. The choices of Q and R directly affect the behavior of the controller. By weighing the matrix Q over matrix R the controller would behave more aggressively and reach the saturation value, which will make the final time and cost decrease. For example, changing the matrix Q to $Q' = 10Q$ causes the final time to become $t' = 11.06$ [s] and reduces the cost to 9.2335 [m/s²]. This is a very powerful result as normally increasing the weight of Q leads to a controller that is more aggressive and spends more fuel. However, since a more aggressive controller would follow the optimal fuel trajectory better, increasing the value of the matrix Q actually reduces the overall cost. Finally, it should be noted that this behavior is partly due to the lack of noise on the system, and due to this reason, it is recommended to have a balance between the Q and R matrix.

Test 1 and Test 2 showed that the analytical guidance returns the correct solution to the general spin and spherical inertia. The next step is to evaluate the AGA when

the inertia is that of a general object. This corresponds to Tests 3-5.

Test 3: General Spin General Inertia Solution from Analytical Guidance Algorithm, $\gamma \leq 1$

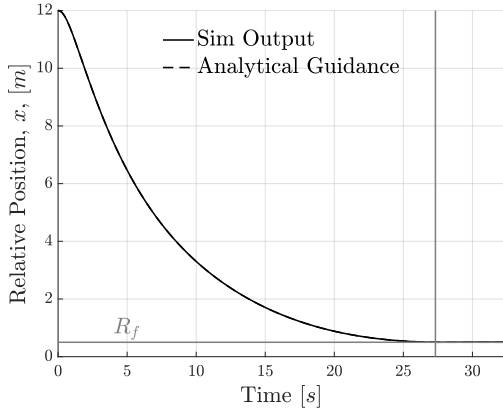
Test 3 tests the behavior of the AGA when the Target’s inertia is not spherical. Because of the nonspherical inertia, the dynamics of the simulation do not match exactly that of the optimization problem solved by the AGA. This causes the feed forward control terms produced by the AGA to not exactly cancel the errors that would keep the relative position error close to zero. This can be seen in Figure (4-11c) as compared to Test 1 and Test 2 the relative position error on y and z are larger.

Figure (4-11a) and (4-11b) show the relative position and velocity of the trajectory output by the Sim as well as the relative velocity, respectively. Overall, the difference between Test 3 and Test 1 in terms of the trajectory generated is not easy to observe: the overall shape and time are similar with the solution for Test 3. The difference between the two is observed in Figure (4-11e), which shows the control time history. Unlike in Test 1, in Test 3 the control effort on z changes sign. This is due to the change of the sign of the angular velocity as the trajectory progresses.

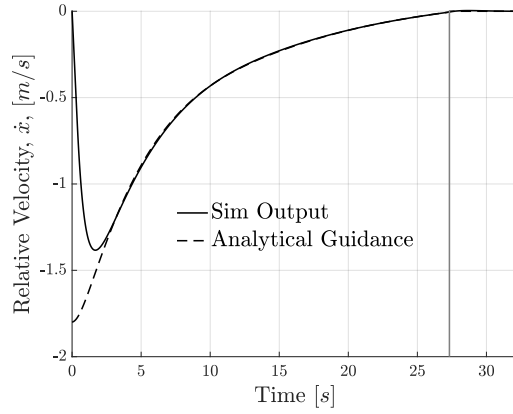
Table 4.13: Test 3: Final time and final cost from AGA simulation vs. GPOPS

Value	AGA	GPOPS	% Diff
t_f [m]	27.3200	26.9662	+1.3120
Cost [m/s]	6.3698	6.1101	+4.2503
Computation Time* [s]	1.2	2.0363e+03	-99.9945

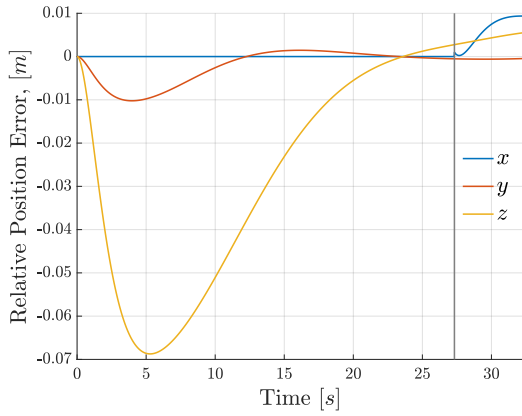
The problem was implemented on GPOPs with the same parameters as defined by Test 3 in Table 4.10. The reported final time and cost for both AGA and GPOPS are included. The computation times are defined as the total time incurred by GPOPS for solving the problem and the total time it took for Simulink to finish the simulation. They are reported here just as a measure of complexity. The difference between the final time and cost being less than 5% can be attributed to the error in optimizing a



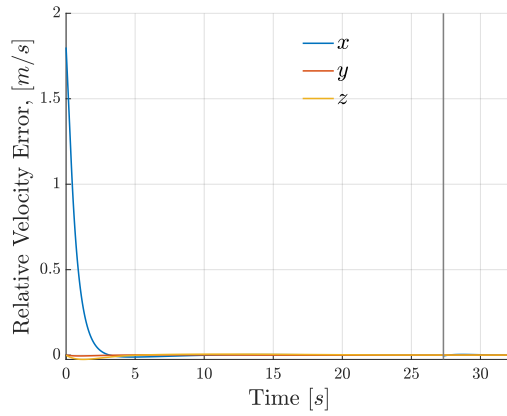
(a) Relative Chaser-Target position, x



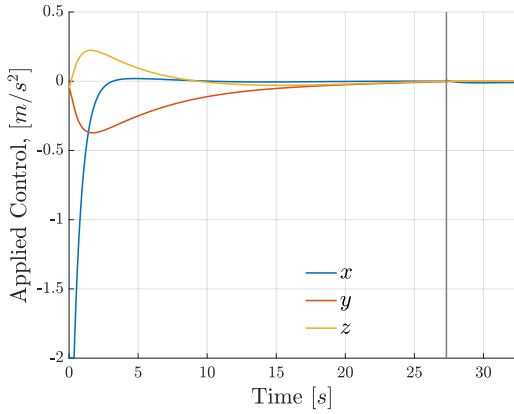
(b) Relative Chaser-Target velocity, \dot{x}



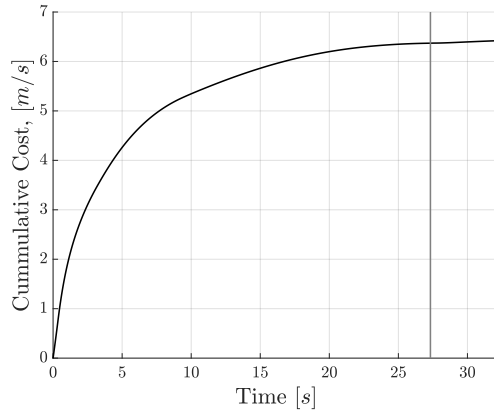
(c) Guidance position error



(d) Guidance velocity error



(e) Control time history



(f) Cummulative cost time history

Figure 4-11: Test 3: Results for analytical guidance simulation for general inertia with angular velocity such that $\gamma \leq 1$ for all time. Vertical grey bar represents end of guidance solution by AGA.

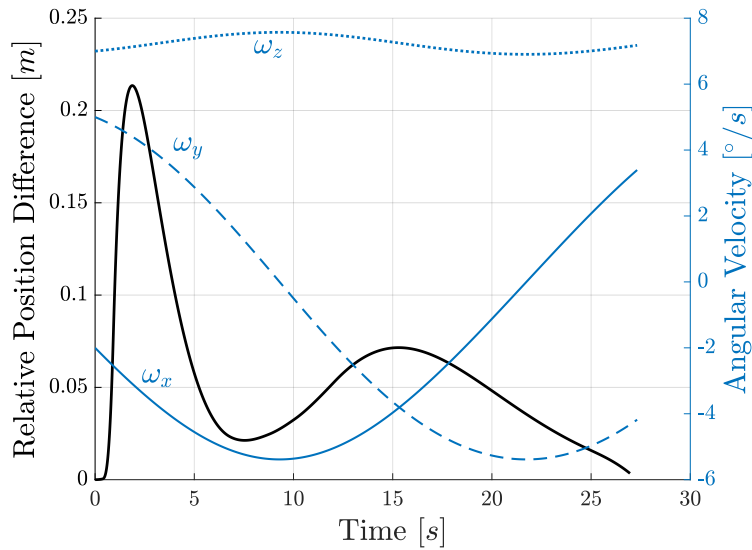
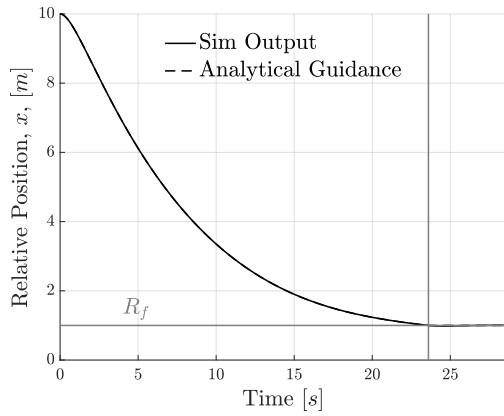


Figure 4-12: Test 3: Relative position difference between AGA trajectory and GPOPS solution

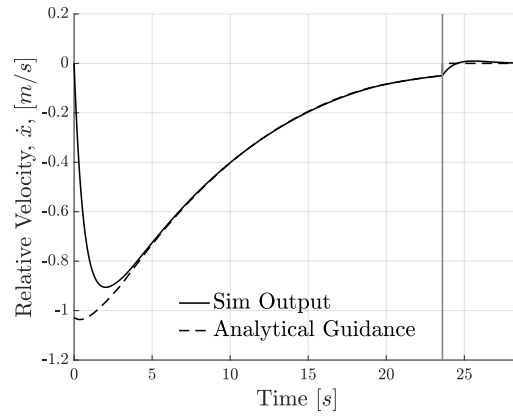
problem in which the Target’s angular velocity is assumed to be constant. Nevertheless, the time it takes for solving the GPOPS problem is orders of magnitude higher than the Simulink problem which speaks to the low computational resources needed to solve the analytical guidance algorithm. The difference between the trajectory produced by the AGA and GPOPS can be seen in Figure (4-12). On the right hand side of this figure, the time history of the angular velocity is plotted. It can easily be seen that the assumption of having a constant angular velocity is invalid. Finally, note that in Figure (4-12) the trajectory is completed within a period of the angular velocity.

Test 4 and 5: General Spin General Inertia Solution from Analytical Guidance Algorithm, $\gamma > 1$. Comparison of Response With and Without Plume Protection

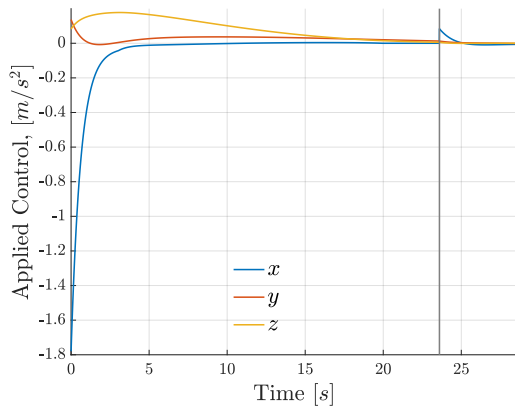
Tests 4 and 5 compared the inclusion and omission of the plume protection to determine if a reduction in the thrust along the x direction was possible. Figures (4-13) and (4-14) show the resultant trajectory for both cases. Figures (4-13a) and (4-14a), show the relative position between the Chaser and the Target, with Figure (4-14a)



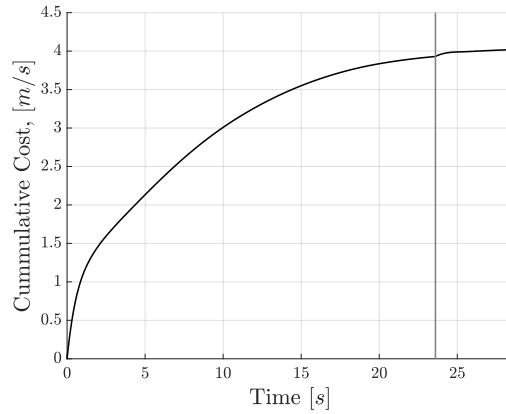
(a) Relative Chaser-Target position, x



(b) Relative Chaser-Target velocity. \dot{x}

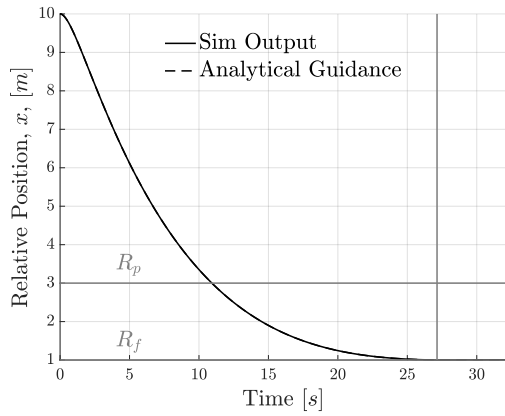


(c) Control time history

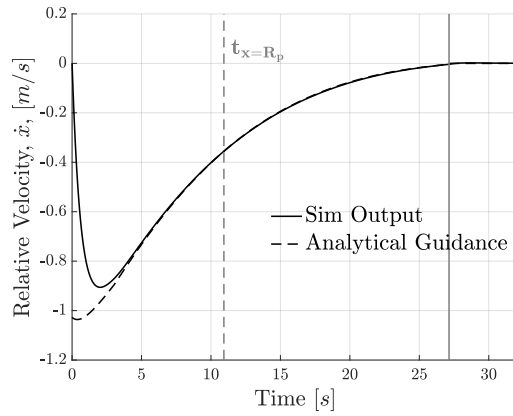


(d) Cumulative cost time history

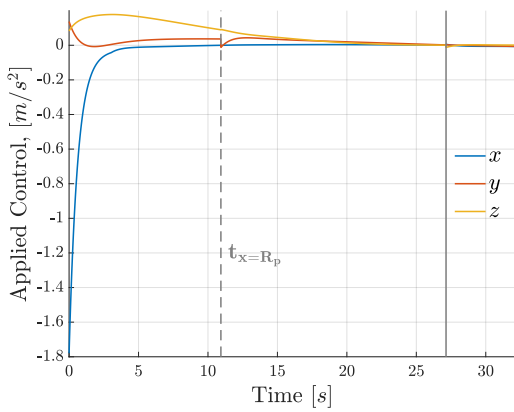
Figure 4-13: Test 4: Results for analytical guidance simulation for general inertia with angular velocity such that $\gamma > 1$ for all time. No plume impingement protection. Vertical grey bar represents end of guidance solution by AGA.



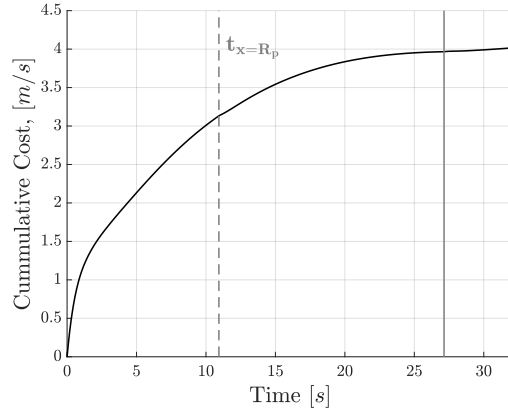
(a) Relative Chaser-Target position, x



(b) Relative Chaser-Target velocity, \dot{x}



(c) Control time history



(d) Cumulative cost time history

Figure 4-14: Test 5: Results for analytical guidance simulation for general inertia with angular velocity such that $\gamma > 1$ for all time. With plume impingement protection, $R_p = 3$ [m]. Vertical grey bar represents end of guidance solution by AGA.

having a grey line representing the radius at which the plume protection began. At that radius, the only allowable solution out of the AGA was the impulsive Bang-Off controller. Subsequently, on all the other subfigures for Test 5 a dashed vertical line indicates the time at which the trajectory reaches R_p .

The control figures for Tests 4 and 5 confirm that indeed having plume protection leads to a decrease in thruster firing on the $+x$ direction. Figure (4-13c) shows a large spike at the end of the trajectory around time $t = 24$ [s]. This spike corresponds to 83 [mm/s^2] acted on the positive x direction towards the Target. In contrast, Figure (4-14c), which has the plume protection set to 3 [m], only shows a firing of less than 5 [mm/s^2] in the positive x direction which corresponds to a 94.3% decrease in the thruster magnitude. Instead, an impulsive firing on the $-y$ axis is shown at the time when the offset is 3 [m]; this corresponds to the change in the trajectory from a Bang-Off-Bang controller to a Bang-Off controller. In terms of cost, the trajectory without plume impingement protection is more fuel efficient than the trajectory with plume impingement, as expected. The differences between the two is less than 1 % as shown in Table 4.14.

Table 4.14: Test 4 and 5: Final time and final cost from AGA with and without plume protection

Value	AGA $R_p = 0$	AGA $R_p = 3$	% Diff
t_f [m]	23.6000	27.1400	-13.0435
Cost [m/s]	3.9307	3.9664	-0.9001
$u_x(t_f)$ [m/s^2]	83.8830	5.5062	+1,4234

4.3.1 Effect of Inertia Ratios in AGA Solutions

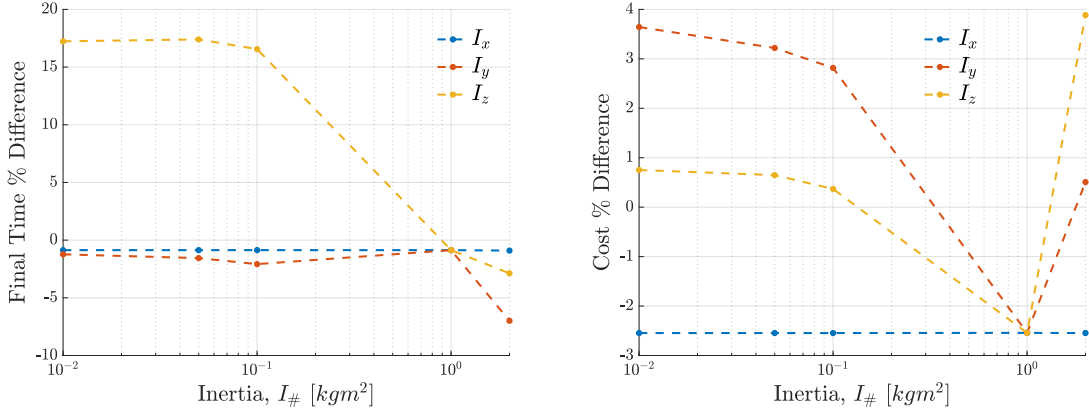
An analysis was conducted to understand the effects of having non spherical inertia on the quality of the solution by the AGA controller. For example, if the inertia tensor of the Target in principle axes was $I_{TAR} = \text{diag}([a, 1, 1])$ with a having the possible values $a \in (0, 2)$ to satisfy the triangle inequality, the period of the angular velocity would increase as $a \rightarrow 0$ or $a \rightarrow 2$. To understand this behavior's effect on the fuel efficiency of the solution, a test case was made in which the value of a changed and

the solution given by GPOPS followed by the LQR controller is compared to that of AGA followed by the same LQR controller. Note that to have a fair comparison of the algorithm, the plume impingement variable, R_p , was set to zero. Furthermore, the GPOPS solution was computed beforehand and an interpolation function was used to obtain the relative position needed to followed by the LQR controller at a given time. Thus, GPOPS corresponds to following a trajectory generated *offline* whereas AGA is to generate a trajectory *online*.

Table 4.15: Effects of inertia ratio on solution of AGA vs. GPOPS. Note for all cases $R_0 = 10$ [m], $R_f = 1$ [m], $u_{sat} = 5$ [m/s²], $\omega = [1, -5, 5]$ [°/s]

Run	I _{TAR} [kgm ²] [I_x, I_y, I_z]	Final Time [s]			Fuel Cost [m/s]		
		AGA	GPOPS	% Diff.	AGA	GPOPS	% Diff.
1	[1,1,1]	24.02	24.2300	-0.8667	4.3620	4.5058	-2.5421
2	[0.01,1,1]	24.02	24.2300	-0.8667	4.3591	4.4730	-2.5456
3	[0.05,1,1]	24.02	24.2300	-0.8667	4.3620	4.4743	-2.5456
4	[0.1,1,1]	24.02	24.2300	-0.8667	4.3913	4.4760	-2.5456
5	[2,1,1]	24.02	24.2400	-0.9076	4.4223	4.5377	-2.5452
6	[1,0.01,1]	28.8900	29.2500	-1.2308	4.7409	4.5743	+3.6428
7	[1,0.05,1]	29.0500	29.5100	-1.5588	4.7390	4.5911	+3.2199
8	[1,0.1,1]	29.2200	29.8400	-2.0777	4.7303	4.6008	+2.8164
9	[1,2,1]	27.5500	29.6200	-6.9885	4.4106	4.3883	+2.8164
10	[1,1,0.01]	28.3700	24.2000	+17.2314	4.6942	4.6591	0.7519
11	[1,1,0.05]	28.3400	24.1400	+17.3985	4.6811	4.6509	0.6484
12	[1,1,0.1]	28.2300	24.2200	16.5566	4.6593	4.6422	0.3680
13	[1,1,2]	28.3800	29.2200	-2.8747	4.2986	4.1378	3.8859

Table 4.15 shows the runs that were done to compare the solutions obtained from GPOPS and AGA. For each test, a single moment of inertia was changed by a value ranging from [0.01, 2] and the subsequent final time and total cost from the LQR were recorded. The first set of rows up to Run 5 corresponds to a change in I_x , the next four to I_y , and the final four to I_z . Overall, the cost difference between AGA and GPOPS is merely 5%. This shows that for the selected angular velocity vector of $\omega = [1, -5, 5]$ [°/s] the difference of the solution is very low and could be used in a spacecraft since normally the fuel allowable for docking is enough for several attempts. There is a difference, however, in the proposed final time between the GPOPS solution and the AGA solution. Whereas the cost difference ranges



(a) Final time difference for runs shown in Table 4.15 (b) Final fuel cost difference for runs shown in Table 4.15

Figure 4-15: Effect if inertia ratio on fuel cost for AGA vs. GPOPS. Solution were obtain with $R_0 = 10$ [m], $R_f = 1$ [m], $u_{sat} = 5$ [m/s²], $\omega = [1, -5, 5]$ [°/s] as nominal values

from (-3%, 4%), the difference between the time ranges from (-3%, 18%). This demonstrates an interesting aspect of the solution: the difference in time does not always correlate with difference in cost.

It should be noted that the cases in which the AGA solution leads to a lower cost than GPOPS; this does not imply that the solution that GPOPS gives is wrong or suboptimal. One of the issues with implementing an offline trajectory is the ability to track it perfectly. Any type of controller would introduce a time lag of the actual output versus the trajectory. This time lag will cause the trajectory to not be followed perfectly and introduce extra fuel cost. For example, for Run 5 the cost reported by GPOPS is **4.4172** [m/s] compared to the 4.5377 [m/s] that was obtained from the simulation. This reported cost is less than the AGA, which indicates that the solution *given perfect tracking* is more optimal. Lastly, the average computation time for obtaining the offline GPOPS solution was 13.76 seconds, compared to an average solve time of 22 milliseconds for the AGA solution. Depending on the computational power of the spacecraft, the time for the solution to be obtained by GPOPS was more than half of the time for performing the trajectory.

Figure (4-15) shows the plotted results of Table 4.15. The two plots clearly show

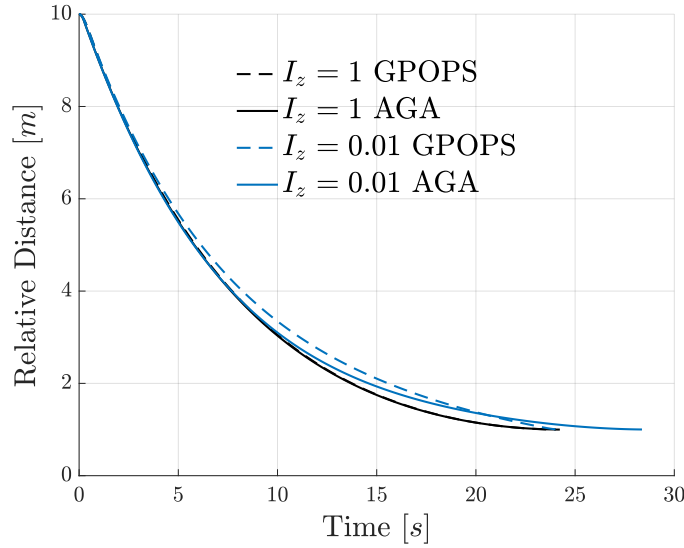


Figure 4-16: Comparison between the AGA and GPOPS solution for Run 1 and Run 10.

that the difference in final time for this particular of angular velocity is affected more by a change in I_z versus the other moments of inertia while the cost is affected more by a change in I_y . Furthermore, the cost as shown in Figure (4-15b) confirms the suspicion that the difference between GPOPS and AGA increases as the inertia becomes less spherical. Thus, we should expect that the minimum of the absolute difference is when the inertia is equal to one. For completeness, Run 1 and Run 3 were plotted on Figure (4-16) where the dashed line represents GPOPS and the solid line represents the AGA controller. It is clear that there is almost no difference between the solution of GPOPS and AGA for Run 1 as it corresponds to the situation in which the Target has a spherical inertia. However, Run 10 shows that the solutions for AGA and GPOPS deviate, which is the cause for their difference not only in time but also in cost as seen in Table 4.15.

4.3.2 Summary of Analysis of the Analytical Guidance Algorithm

This section analyzed the results fo the behavior of the AGA on simulation. The AGA was able to converge within a small percentage of error to the case when the Target's

inertia is that of the sphere. Furthermore, the addition of the plume impingement protection was able to significantly reduce the thrust firing towards the Target by an order of magnitude. The case in which the Target’s moment of inertia was not of a sphere was also studied, and the results showed that the difference between the cost associated to following GPOPS versus the AGA was within 5 % for the selected angular velocity. Further tests must be done for different angular velocity directions to bound the differences of the fuel cost between both solutions. Nevertheless, the analytical guidance algorithm is a viable solution for a situation in which there must be a re-computation of the guidance solution while ensuring **convergence** and *real-time* capability.

4.4 ENVISAT Example Mission

In order to provide a realistic test case scenario for the AGA, the Environmental Satellite 1 (ENVISAT) was used as a proposed orbital removal mission. The ENVISAT is a European Space Agency satellite that lost contact with ground on April 8th, 2012 after presumably a collision with an uncategorized orbital debris [43]. Due to the orbital plane location, the ENVISAT has been a source of concern among space situational awareness analysts due to the danger it poses if it were to collide with active satellites or orbital debris. Thus, there has been interests in continuous observation of the satellite as well as proposals for an ADR attempt [44].

ENVISAT is a large satellite with dimensions 26 m by 10 m by 5 m. The satellite features a large canted solar array as shown in Figure (4-17). The satellites’ body axes are labeled in the diagram. Additionally the satellites moment of inertia is given by

$$I_{ENVISAT} = \begin{bmatrix} 17023.3 & 397.1 & -2171.4 \\ 397.1 & 124825.7 & 344.2 \\ -2171.4 & 344.2 & 129112.2 \end{bmatrix} [kgm^2] \quad (4.12)$$

The center of mass for the satellite is given by $x_{CM} = [-3.905 \text{ m}, -0.009 \text{ m}, 0.003 \text{ m}]$

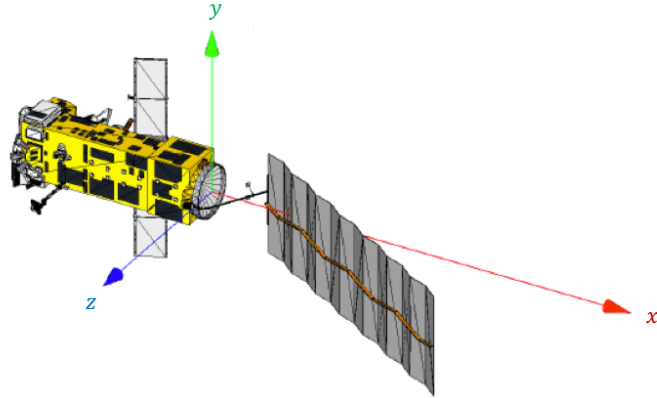


Figure 4-17: ENVISAT model, x , y , z represent the body axis. Source [44]

from the origin of the frame shown in Figure (4-17).

Various studies have been conducted on determining the attitude of ENVISAT. Due to gravity gradient, solar radiation pressure, and drag torques there is a belief that ENVISAT has an angular spin of at most $3.5^\circ/s$ about an axis as well as some residual spin on the other axes. Therefore, guidance trajectories such as the one employed for Restore-L or MEV-1 would not be permissible due to the tumbling behavior of ENVISAT.

For this example mission the parameters shown in Table 4.16 were used. The Initial offset was a conservative value chosen so that the Chaser would be safe from being hit by the solar panel during Phase 2 of the trajectory inspection. The final offset is an assumption that the servicing ring was 4 meters away from the center of mass on the $-x$ direction. The servicing ring was chosen as the docking target to match the Restore-L mission description. The initial angular velocity was arbitrarily chosen such that the y -axis had the maximum angular velocity predicted by ENVISAT.

For the Chaser, an arbitrary maximum acceleration saturation of $30 \text{ cm}/s^2$ was chosen. Finally, the initial position was selected to not be exactly aligned with the x axis. This adds more realism because, while it is expected that during Phase 2 the satellite will aligned with the chosen docking axis, it might not be perfect. Therefore, this example tests the robustness of AGA in two ways: the ENVISAT's inertia is

Table 4.16: ENVISAT Example Mission Parameters

Parameter	Value
Initial Offset, R_0 [m]	30
Final Offset, R_f [m]	4
Initial Ang. Velocity, $\boldsymbol{\omega}(t_0)$ [$^\circ/s$]	$[-0.5, 3.5, 0.5]^T$
Target Inertia, [kgm ²]	$I_{ENVISAT}$
Max Saturation, u_{sat} [m/s ²]	0.3
Initial Position, [m]	$[-30, 0.5, 1]^T$

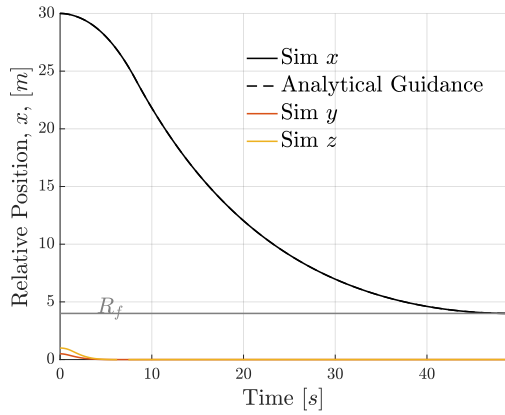
not in principal axes and thus has off-diagonal terms, and the initial position of the Chaser is not perfectly aligned with the Target’s docking axis.

Figure (4-18) shows the result of the ENVISAT mission. The trajectory is able to converge to the desired docking location as shown in Figure (4-18a). The y, z errors in the relative position go towards zero and are negligible at time $t = 12$ [s]. Additionally, Figure (4-18b) shows that at time $t = 8.5$ [s] the sim trajectory converges to the analytical guidance at which point the control effort on x becomes close zero. This is encouraging as it shows that AGA solution resulted in low plume impingement close to docking, ensuring ENVISAT’s safety.

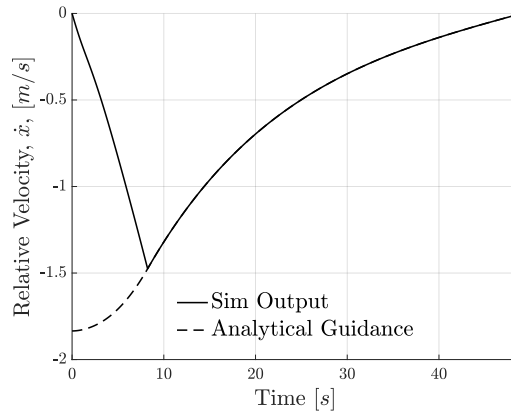
The control time history appears different from the other cases, especially from time $t \in [0, 8.5)$ due to the initial offset in the normal axis. Finally, this behavior is also present on the cumulative cost as shown in Figure (4-18d), as once the sim output converges to the analytical guidance the resultant cost increases at a slower rate.

Figure (4-19) shows the trajectory of the example mission in inertial space. The XY and YZ projection are shown in Figures (4-19b) and (4-19b), respectively. Additionally, the angular velocity time history is shown in figure (4-19d). From these figures we can see that the attitude of ENVISAT changed dramatically and a typical r-bar or v-bar approach would not have been able to dock. Furthermore, the trajectory is completed within a full rotation period of the Target, which is consistent with the idea that waiting too long would correspond to fuel wasted.

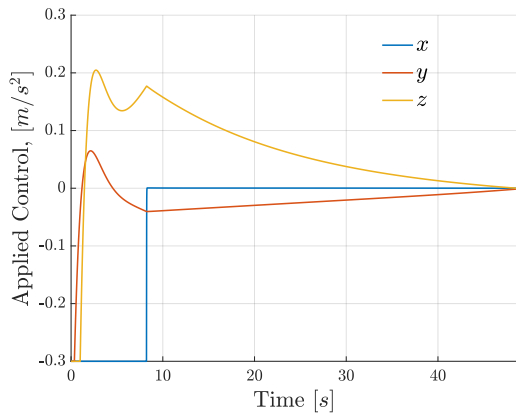
The results shown in this section demonstrate the ability for the AGA to yield an efficient and convergent algorithm to be used in an actual mission. The trajectory



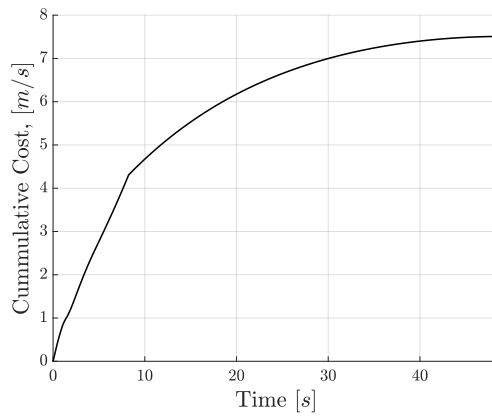
(a) Relative Chaser-Target position



(b) Relative Chaser-Target velocity

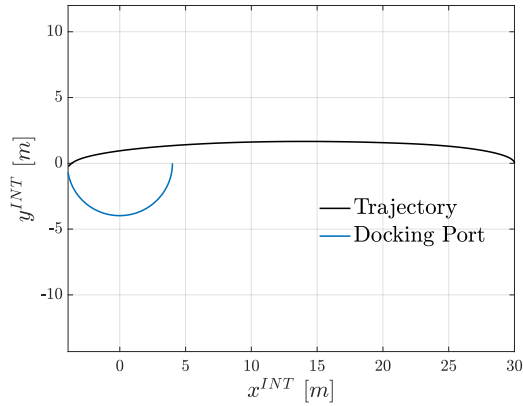


(c) Control time history

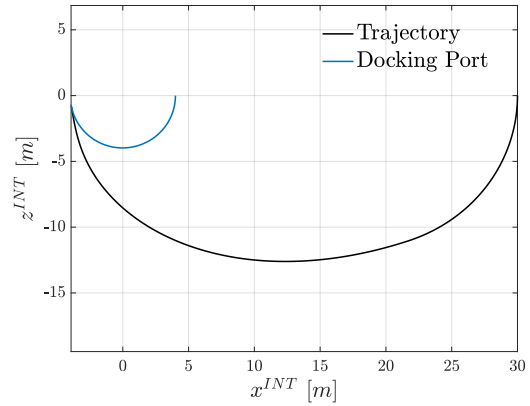


(d) Cumulative cost time history

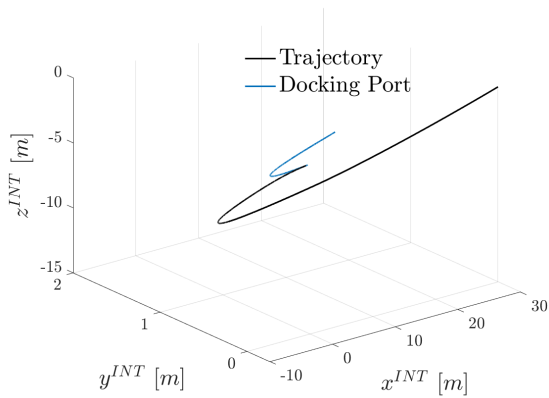
Figure 4-18: ENVISAT mission test results



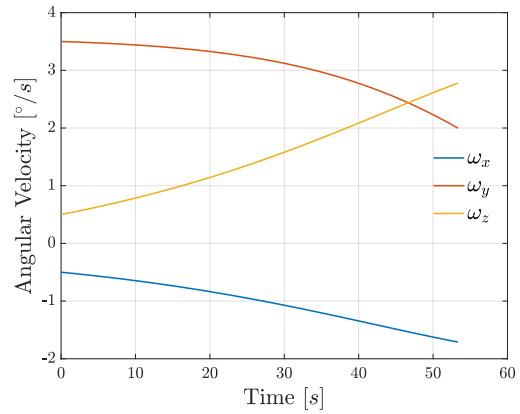
(a) Trajectory as seen in XY inertial plane



(b) Trajectory as seen in XZ inertial plane



(c) Trajectory as seen in 3D view



(d) Target's angular velocity time history

Figure 4-19: AGA simulated docking trajectory for ENVISAT. Blue line corresponds to the docking point, $[4, 0, 0]$, and black line corresponds to the Chaser trajectory. Figures shown in inertial frame.

shown in the ENVISAT example obeys the constraint on saturation and is able to demonstrate low impingement risk once the Chaser approaches the satellite. Further analysis must be done for this case by adding the frame conversion that would be needed to follow the trajectory on the Chaser's body frame. Furthermore, as stated before, addition of noise and perturbations are needed to gain more confidence on the viability of using this trajectory. Nevertheless, this example shows promising results that are worth further studying.

Chapter 5

Conclusion and Future Work

A guidance and control algorithm to allow for a spacecraft to capture a tumbling object was presented in this thesis. The need for such algorithms was motivated in Chapter 1 and includes situations in which the target object has not attitude control or is uncooperative, such as active debris removal and satellite servicing or in-space assembly. Furthermore, current state-of-the-art algorithms for accomplishing soft docking of tumbling objects were described in Chapter 2. However, many of these algorithms require nonlinear programs, which have neither a guaranteed feasible solution nor a guaranteed solving time, that might not be suitable for current on-orbit computational capabilities. Furthermore, the current algorithms do not directly address the ability to perform a soft docking without thruster plume impingement from the Chaser satellite.

The formulation of a quasi-analytical guidance algorithm for use in real-time trajectory generation was presented in Chapter 3. The formulation entailed dividing the general docking problem into subproblem depending on the Target angular velocity and inertia ratio. In that chapter, a fully analytical expression for guidance was obtained for a Target with a flat spin. Furthermore, it was shown that depending on the angular velocity direction, the docking problem with a spherical inertia can either be solved fully analytically or through a simple bisection method procedure. Finally, an algorithm was developed for the general docking problem based on the impulsive solution to the simpler spherical inertia problem. The analytical guidance algorithm

was shown to be orders of magnitude faster than a pseudospectral method, and also includes the ability to avoid plume impingement.

Chapter 4 presented the results for several cases for both the analytical guidance algorithm as well as the solution to the subproblems when the Target’s inertia was spherical. This chapter showed that the analytical and numerical version for the situation in which the Target’s inertia was spherical exactly matched the solution provided by the commercially available optimal control solver, GPOPS. Also, the analytical guidance algorithm provided a solution with an increase in fuel cost within 5% from the solution given by GPOPS; however, the solution was obtained five orders of magnitude faster. The chapter concluded with an example scenario for ENVISAT—a European Space Agency weather satellite that is currently tumbling on orbit—and showed that despite ENVISAT’s inertia in body frame *not* being in the assumed principal axes and the initial position *not* being aligned with the body’s x -axis, the algorithm was still able to converge and dock with effectively no plume impingement.

Additionally, Appendix A presents a discrete trajectory optimizer for transporting components to halo orbit by exploiting dynamical system theory through stable manifolds. The optimizer was implemented as part of a high level study on a cost-benefit analysis for in-space assembly of telescopes. The results from the optimizer could be used as a first-cut solution for launch ΔV needed to transport a component to a halo orbit during a selected time window. Finally, Appendix B presents the impulsive solution versions of those obtained in Chapter 3, and the equations are used as part of the analytical guidance algorithm.

5.1 Thesis Contributions

- Developed a quasi-analytical guidance algorithm for docking with a tumbling object with very low computational complexity for use in real-time spacecraft proximity operations
- Demonstrated the ability to avoid plume impingement on Target by leveraging the possible solution forms provided by Pontryagin’s Minimum Principle which

lead to an easy-to-implement addition to the analytical guidance algorithm

- Showed the ability for the algorithm to converge and lead to a soft dock even when the Target's inertia was not in principal axes or with an initial misalignment
- Developed a discrete transport trajectory optimizer for use in analyzing the ΔV cost for in-space assembly

5.2 Recommendations for Future Work

- Analyze the limits of robustness of analytical guidance algorithm with initial x -axis offset
- Develop procedure for obtaining reference attitude command for the Chaser spacecraft during proximity operation in conjunction with the analytical guidance algorithm
- Implementation of solution in a higher fidelity simulation which includes sensor and process noise
- Hardware demonstration using NASA's Astrobe platform for validation of algorithm in on-orbit hardware
- Addition of nominal orbit destinations for discrete transport trajectory optimizer
- Addition of phasing component to the discrete transport trajectory optimizer to allow for rendezvous of multiple components for in-space assembly

5.3 Code Source and Repositories

The code for the algorithms developed in this thesis can be accessed at <https://github.mit.edu/cabrales/ROAM> or by contacting the author at cabrales@mit.edu.

Appendix A

Trajectory Generation for In-Space Assembled Telescope

This chapter focuses in the trajectory design aspect of the in-Space assembled telescope study. The study was conducted to analyze the design trade-space of the benefits for assembling a telescope in-space versus traditional assembly on ground. Section 1 covers an introduction to the overall study as well as the motivation for fuel-efficient trajectories to the assembly destination. Section 2 covers the methodology of generating a stable manifold trajectory to the Earth-Moon system and the Sun-Earth system. Section 3 gives an overview of the trajectory optimizer that was designed for this study. Finally, Section 4 presents results for a notional test case.

A.1 Background and Motivation for an In-Space Assembled Telescope

Larger apertures for telescopes directly correlate to higher angular resolution as well as a potential increase in signal-to-noise ratio [45]. Due to this, there is a natural interest in maximizing the aperture diameter for a given telescope while satisfying the constraints of the launch vehicle volume as well as cost associated with producing the full mirror. For example, the James Webb Space Telescope has a 6.5 m diameter

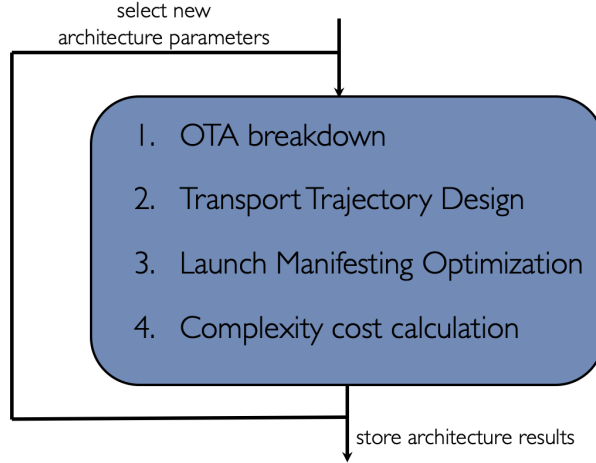


Figure A-1: Overview of optimization framework. Source [7]

and the planned Large UV Optical Infrared Surveyor (LUVOIR) is planned to have a 15 meter diameter, which would be over 40 times more powerful than the Hubble Telescope [46]. Unfortunately, current launch vehicles only fit a diameter of around 15 m [47]. Thus, as larger telescopes are required for advancements in the field of astrophysics, the ability for in-space assembly will be required. The question remains, however, of determining at which point assembling in-space is more cost efficient and less complex than designing a folding mirror or launching a monolithic mirror. The preliminary answer to this question is addressed in the ISAT study presented in [7].

Sanchez et al. presented a preliminary optimization framework for in-space assembled telescopes. The analysis included the complexity associated with segmenting the primary mirror, such as the overall mirror diameter, size of each segment, and the choice of raft geometry (or the groups of mirror segments transported). Additionally, Albee and Bart presented a new methodology of performing launch manifesting that included maximizing the volume and mass usage per launch as well as prioritizing the launch components [48]. The framework for the Pareto optimization is shown in Figure (A-1). For each architecture, the Optical Telescope Assembly (OTA) was deconstructed in terms of components that could be individually launched. The transport trajectory was designed based on the launch manifesting optimization. Finally, the “complexity cost” was calculated and used in the Pareto analysis.

Designing fuel efficient transport trajectories is necessary in order to determine

a viable solution. Since a large percentage of the total budget for a mission is due to launch costs, a strategy to minimize the fuel of the trajectory—and thus maximize the amount of payload per launch—is required. For this study, it was assumed that the entire spacecraft would be assembled at a single location. Two locations were studied: Sun-Earth Lagrange point 2 (SEL2) and Earth- Moon Lagrange point 1 (EML1), namely because the former will be the location of future space telescopes [49] [50] and the latter is the proposed destination for the NASA Lunar Gateway [51]. Assembling in LEO was not considered in this study as transporting a fully assembled large telescope would require a specialized tug to provide a large enough Δv to enter a transfer trajectory to either SEL2 or EML1. Instead, it has been shown that transfer Δv between the EML1 environment and SEL2 orbits can be on the order of cm/s [52].

A.2 Stable Manifold Trajectories in EML1 and SEL2 System

The circular restricted three body problem (CR3BP) is a powerful simplification that allows insight into natural systems. The typical (CR3BP) considers the effect of two main bodies, for example Earth and Moon, on the motion of a smaller body (e.g. a satellite). It is assumed that the mass of the smaller body is insignificant to the other two bodies, $M_3 \ll M_1, M_2$. Furthermore, the two main bodies are assumed to be rotating in a circular orbit about the the center of mass of the combined M_1, M_2 system called the barycenter. This assumption is valid because even though the Moon and Earth have elliptical orbits about their respective CM systems, it is useful in simplifying the equations of motion. Finally, an assumption that comes out of the circular trajectory assumption is that both M_1 and M_2 rotate with constant angular velocity.

A diagram with reference frames for the three body problem is shown in Figure (A-2). The primary body is indicated by the red circle, the secondary by the blue

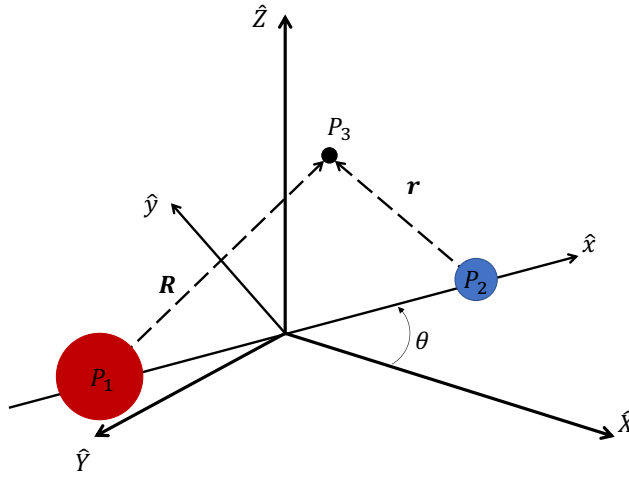


Figure A-2: Diagram of the CR3BP. Inertial and rotating frame for the CR3BP are shown by $\{0, \hat{X}, \hat{Y}, \hat{Z}\}$ and $\{0, \hat{x}, \hat{y}, \hat{z}\}$

circle, and the tertiary by the black dot. The frame $\{0, \hat{X}, \hat{Y}, \hat{Z}\}$ is an inertial frame where it is assumed that the two bodies lie on the XY -plane. The rotating frame $\{0, \hat{x}, \hat{y}, \hat{z}\}$ is defined with the primary and secondary body aligned along the x axis. The y axis is normal to the x axis but in the XY plane. Finally the z axis is aligned with the inertial frame z axis. Since the body fixed frame is rotating about the z axis, it is possible to define an angle θ and a corresponding angular velocity ω to determine the orientation at all times. Finally, for the sake of completeness, the vectors \mathbf{R} and \mathbf{r} correspond to the position of the third body relative to the first and second body, respectively.

It is well known that by defining the equations of motion of the CR3BP in a rotating frame five equilibrium points arise in which the centripetal acceleration match the Coriolis acceleration [53, 54]. These points are unique to the three body problem and would not be present on the classical Kepler's two body problem. These points called Lagrange or libration are located as shown qualitatively in Figure (A-3). Nonlinear analysis informs that L_1, L_2, L_3 are inherently unstable, while L_4 and L_5 are stable. Nevertheless, it has been known that periodic or quasi-periodic orbits exists in L_1

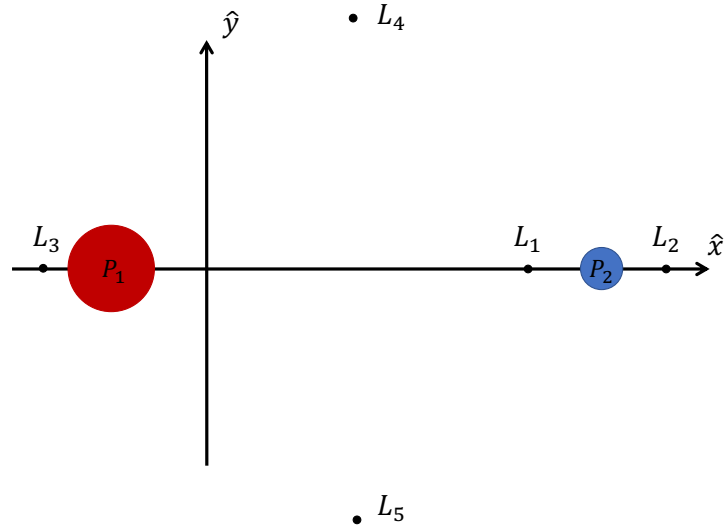


Figure A-3: Qualitatively diagram of location of Libration points in the CR3BP and L_2 points [55]¹.

A.2.1 Generation of Halo Orbits

Generating the periodic and quasi-periodic orbits has been known in literature for a very long time. First order solutions to planar Lyapunov orbits (or periodic orbits constrained in the xy plane) were known from the time of Lagrange [53], however, analytical expressions for more complicated orbits such as Lyssajous or halo orbits were developed much later. Farquhar on his Ph.D. thesis proposed the use of halo orbits for satellite relays and space stations due to their benefits of always having direct view of Earth [56]. Furthermore, Farquhar and Kamel developed a third degree order solution to both Lyssajous and halo orbits [55]. Richardson expanded this work by developing an analytical expression for halo orbits and also providing a methodology for producing numerical solutions [57]. The initial conditions from Richardson’s seminal paper were used in this analysis for creating the halo orbit in both the Sun-Earth L_2 and Earth Moon L_1 points.

Unfortunately, Richardson’s initial condition for a halo orbit is not enough to

¹ L_3 also admits periodic and quasi-periodic orbits and for the Sun-Earth system is the allegedly location of Planet X

produce a full periodic orbit using numerical propagation. Due to this a differential corrector (DC) is needed. The one used in this analysis was the one proposed by Howell reproduced here [58].

Given the state vector in the rotating frame as $[x, y, z, \dot{x}, \dot{y}, \dot{z}]^T$, the equations of motion of the third body are given by

$$\begin{aligned}\ddot{x} - 2\dot{y} &= \frac{\partial U}{\partial x} \\ \ddot{y} + 2\dot{x} &= \frac{\partial U}{\partial y} \\ \ddot{z} &= \frac{\partial U}{\partial z}\end{aligned}\tag{A.1}$$

where the angular velocity of the rotating frame as well as the distance between P_1 and P_2 were normalized to one, and

$$U = \frac{1}{2}(x^2 + y^2) + \frac{1 - \mu}{((x + \mu)^2 + y^2 + z^2)^{1/2}} + \frac{\mu}{((x - 1 + \mu)^2 + y^2 + z^2)^{1/2}}\tag{A.2}$$

where U represents the potential energy of the system and $\mu = \frac{m_2}{m_1 + m_2}$. Furthermore, a state and time dependent state transition matrix can be defined as

$$\frac{d}{dt}\Phi(t, 0) = F(t)\Phi(t, 0)\tag{A.3}$$

where $F(t) = \begin{bmatrix} 0_{3 \times 3} & I_{3 \times 3} \\ U_{xx} & 2\Omega \end{bmatrix}$ and $\Omega = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ and are shown in Equation (3) of [58].

The equations of motion in Equation (A.1) as well as Equation (A.3) are integrated starting at the initial condition given in Richardson's paper (and with the form $\mathbf{x}(t_0) = [x_0, 0, z_0, 0, \dot{y}_0, 0]^T$) until the trajectory crosses the xz -plane. Due to nonlinearities and numerical error from integration, the solution will not be of the same form as the initial condition and a DC will be needed to correct the value. Given a state-vector at crossing be of the form $\mathbf{x}(T/2) = [x(T/2), 0, z(T/2), \delta\dot{x}, \dot{y}(T/2), \delta\dot{z}]^T$ and the current

integrated state transition matrix $\Phi(T/2, 0)$, it is possible to develop a DC as

$$\begin{aligned} \dot{x}'_0 &= \dot{x}_0 + \Delta\dot{x} \\ \dot{z}'_0 &= \dot{z}_0 + \Delta\dot{z} \end{aligned} \tag{A.4}$$

where

$$\begin{bmatrix} \Delta\dot{x} \\ \Delta\dot{z} \end{bmatrix} = \begin{bmatrix} \Phi_{41} - \delta\dot{x}/\ddot{y}\Phi_{21} & \Phi_{45} - \delta\dot{x}/\ddot{y}\Phi_{25} \\ \Phi_{61} - \delta\dot{z}/\ddot{y}\Phi_{21} & \Phi_{65} - \delta\dot{z}/\ddot{y}\Phi_{25} \end{bmatrix}^{-1} \begin{bmatrix} -\delta\dot{x} \\ -\delta\dot{z} \end{bmatrix} \tag{A.5}$$

where \ddot{y} represents the acceleration due to the equations of motion in Equation (A.1) at time $t = T/2$. The process of modifying the initial x and z speed is repeated until $\delta\dot{x}$ and $\delta\dot{z}$ are below a desired tolerance value or reach machine precision. Usually, less than 5 iterations are needed in order to reach tolerances of $1e - 14$.

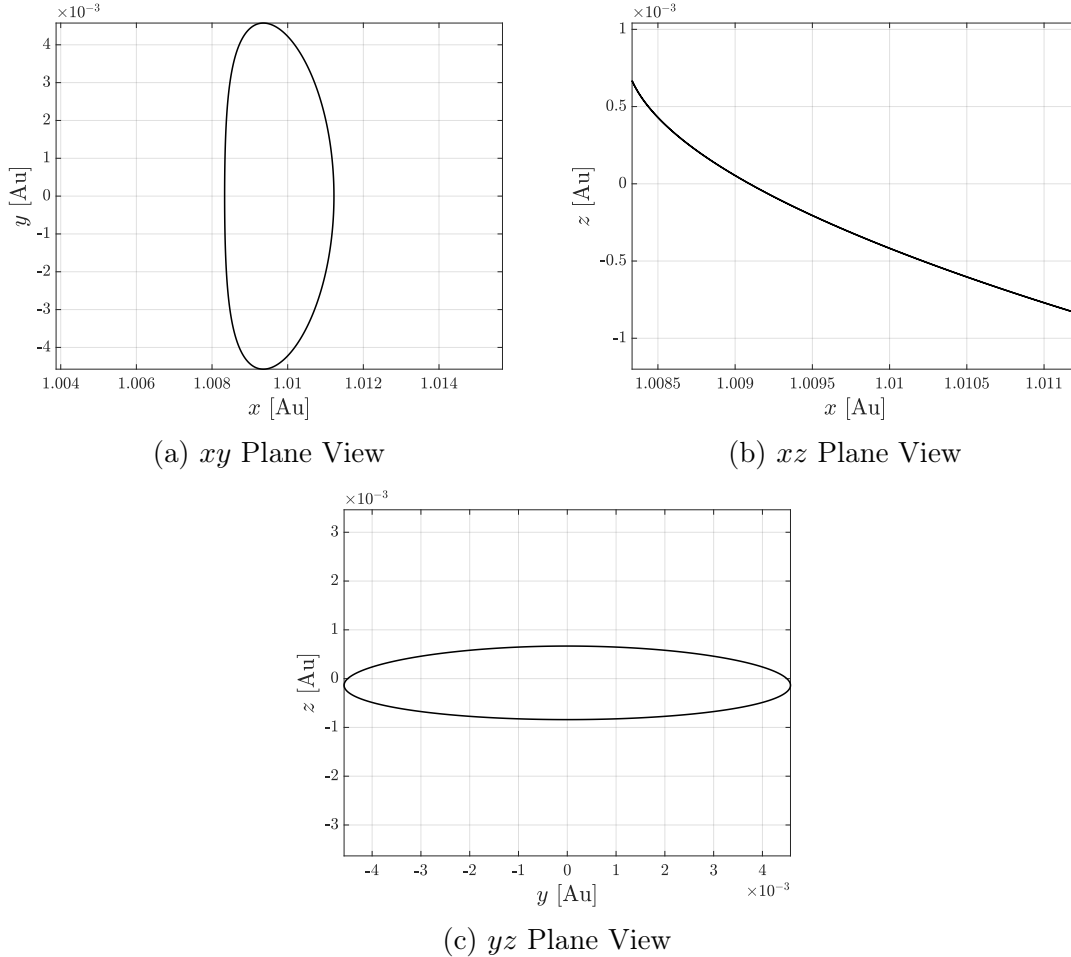


Figure A-4: Example of Sun-Earth L2 halo orbit with amplitude of 110,000 [km].

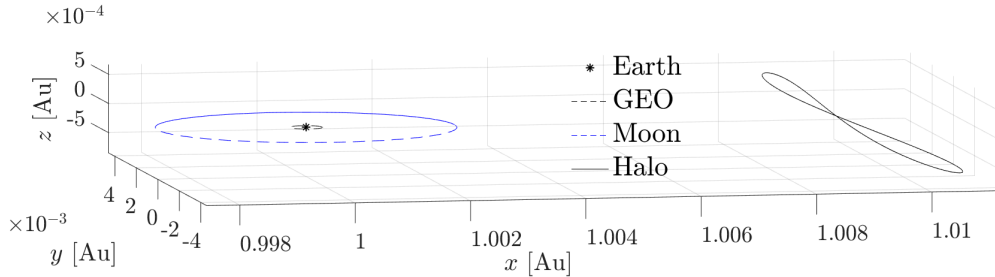


Figure A-5: Example of un-Earth L2 halo orbit with amplitude of 110,000 [km]. Earth and Moon Orbit are shown for scale.

Once the DC converges, the initial condition can be used to propagate the full equation of motion to a full orbit. To give a nominal example, a halo orbit with an amplitude of 110,000 kilometers located at Sun-Earth L2 was propagated for a full period (roughly 180 days). The results are shown in Figure (A-5). Additionally, the full orbit with the location of the Earth as well as the orbit of the Moon are shown in Figure (A-5).

A.2.2 Generation of Stable Manifold Trajectories

Since the third body problem is essence a chaotic system, dynamical system theory can be used to investigate the state space. Manifolds represent a surface inside the system state space that represents a set of all solutions in that local region [59]. For the three body problem, manifolds can be thought in reference to a surface of trajectories in a halo orbit that are either stable or unstable, where stable means that all trajectories approach the nominal halo orbit as $t \rightarrow \infty$. For example, by selecting an orbit and integrating the equations of motion as well as the state transition matrix, we can obtain trajectories in either the stable or unstable manifolds. At any point in

time the eigenvalues of the state transition matrix, $\lambda(\Phi(t, t_0))$, will have one of these forms:

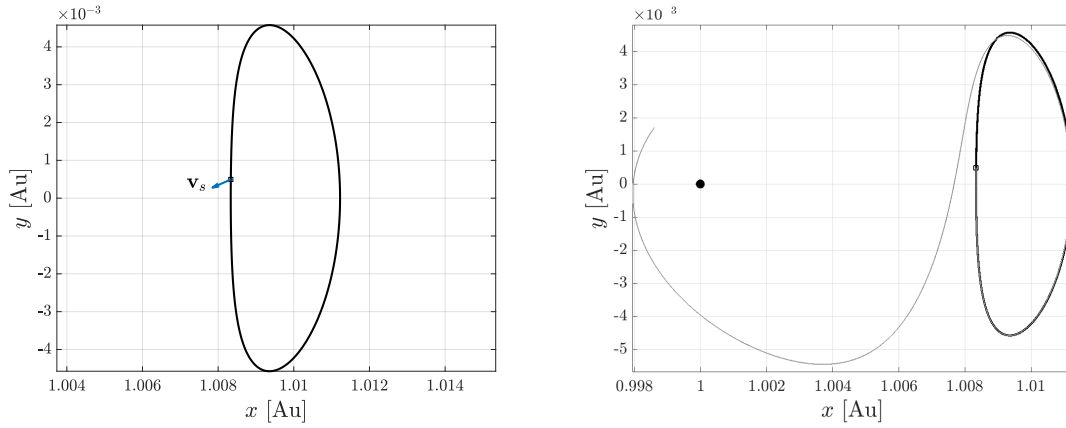
- $|\lambda_i| < 1$, which corresponds to the stable manifold (approach the orbit)
- $|\lambda_i| = 1$, which corresponds to the center manifold (stay on the orbit)
- $|\lambda_i| > 1$, which corresponds to the unstable manifold (leave the orbit)

Thus, by displacing the nominal halo orbit by a small distance along the corresponding stable eigenvector, it is possible to enter the stable invariant manifold. This technique was used to generate “free” trajectories from around the primary to the desired halo orbit². The steps for obtaining these trajectories are as follows:

1. Propagate the halo orbit for at least one period, while integrating the state transition matrix, $\Phi(t, t_0)$
2. Obtain the eigenvector of $\Phi(t, t_0)$, \mathbf{v}_s associated with the stable manifold at several points along the orbit (see Figure A-6a)
3. Perturb the current state by a small distance along the eigenvector, $\mathbf{x}'(t) = \mathbf{x}(t) + d\mathbf{v}_s$
4. Propagate the system backwards in time $t = -\tau$ (see Figure A-6b)

These processes can be repeated at various points throughout the nominal halo orbit. These will generate a subsample of the full stable manifold. Figure (A-7) shows 60 stable manifold trajectories that were obtained by sampling points around a nominal halo orbit of amplitude of 110,000 [km] about SE-L2. Note that some orbits do not approach the Earth, so an extra step must be taken to remove those orbits and only look at orbits close to Earth.

²Free trajectory just means that once the state of the spacecraft reaches the manifold trajectory and matches its state, the spacecraft’s trajectory will naturally evolve such that it reaches the halo orbit. In reality, due to external perturbations such as solar radiation pressure and outer bodies gravity, the spacecraft will have to perform mid course corrective maneuver to join the manifold trajectory again



(a) Nominal halo orbit with stable eigenvalue direction shown (b) Propagation of stable manifold trajectory. Dark dot represents Earth.

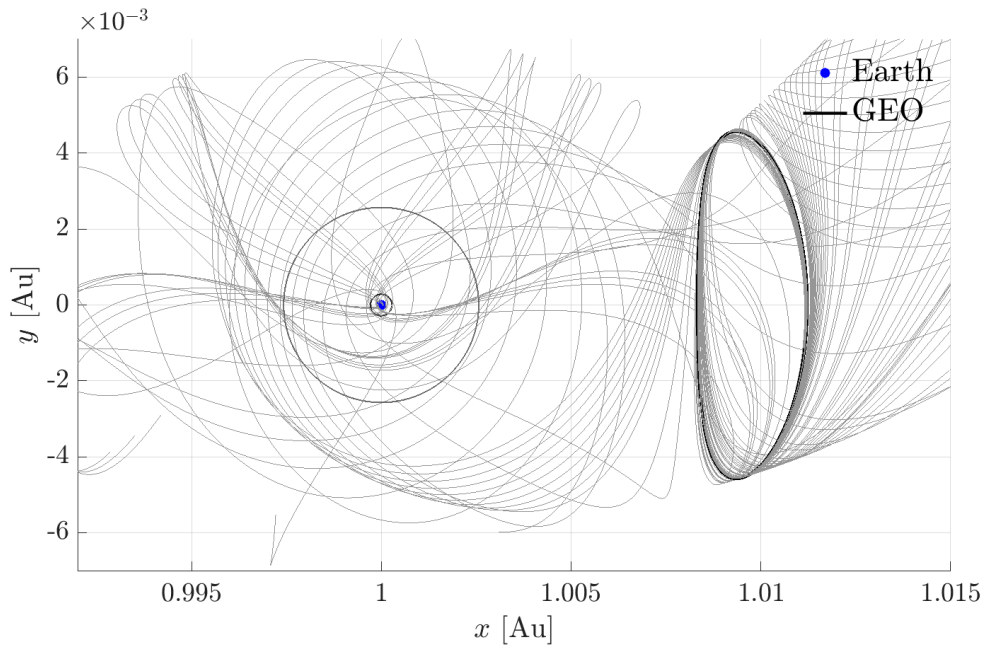


Figure A-7: Selection of 60 stable manifold trajectories for nominal halo orbit of 110,000 km amplitude. Note some of the manifold trajectories do not go towards Earth.

A.3 Discrete Transport Trajectory Optimizer

The transport trajectory design was one of the main functions for the optimization framework developed in the study. The goal of the optimizer was to find a low-fuel trajectory for each of the components that were provided by the launch manifest function. The corresponding inputs of the trajectory, set by the launch manifest function, are shown in Table A.1.

Table A.1: Required inputs for the transport trajectory optimizer

Input	Description
Target location	Location of halo orbit (SE-L2 or EM-L1)
Launch location	Location of departure from Earth (e.g. Cape Canaveral, French Guiana, etc.)
Time of Launch	Initial time of when launch can happen
Launch Window	Range of days over which launch can occur

Although only halo orbits in SE-L2 or EM-L1 were considered, the optimization routine is structured in a way that additional locations can be selected such as a Near Rectilinear Halo Orbit (NRHO), the planned operating orbit for the NASA Lunar Gateway [51]. Furthermore, with flexibility in mind, the launch location was parametrized as a function the latitude and longitude.

The following assumptions were made in the trajectory generation formulation. First, dynamics of the system remain that of the CR3BP and other celestial bodies were ignored; this assumption is also made in [49, 50] in order to generate a first cut solution. The effects of Solar radiation pressure, Earth’s oblateness, and other n-body effects were ignored. These effects are known to change the location of the Lagrange point and the period of the orbit [50, 60], as well as necessitate the performance of mid-course correction maneuvers. However, for a first cut solution, these effects on the total Δv needed to reach the destination are not significant. Additionally, the objective of the trajectory design process was only to provide transport of the assembly components to a defined Halo orbit. The problem of phasing—i.e., ensuring all satellite components arrive in close proximity to each other—was not addressed. Finally, a drag penalty of 1.7 km/s was added to the results to account for the drag

needed to combat the atmosphere when launching from Earth.

A.3.1 Frame Conversions

One of the challenges of designing a transport trajectory optimizer using dynamical system theory is the need for frame conversions. The stable manifold trajectories shown in Section A.2.2 represents solutions in the rotating frame. To ease the burden of keeping track of the positions and velocities, several frames were generated to simplify the conversion between the different frames and are shown in Table A.2.

Table A.2: Required inputs for the transport trajectory optimizer

Frame	Name	Description
LLA	Longitude, Latitude, Altitude	Geodetic polar frame (body-fixed) corresponding to the latitude and longitude of a point at a given altitude from Earth.
ECEF	Earth Centered Earth Fixed	Geographic cartesian coordinate frame (body-fixed about Earth's rotation). Origin defined as centered of mass of Earth.
ECI	Earth-Centered Inertial	Inertial Earth centered frame. Origin is at Earth's center of mass. X axis is aligned with the vernal equinox. Z axis points 90° perpendicular to the Equator towards the celestial North pole. Refer to J2000.
BEI	Barycenter Ecliptic Inertial	Inertial frame centered around the CR3BP barycenter (either in Sun-Earth system or Earth-Moon system). X axis points towards the first point of Aires, Z points towards the angular momentum of the system. Y completes the frame.
BER	Barycenter Ecliptic Rotating	Body-fixed frame centered around the CR3BP barycenter (either in Sun-Earth system or Earth-Moon system). X axis points towards the line that produced by connecting the two main bodies. Z axis points towards the angular momentum of the system. Y completes the right hand frame.

Several of the frame conversions particularly those that go from a body-fixed frame to inertial frame require time as one of the inputs. Due to this, the Julian date was selected as the reference time, and the ECI's J2000 was used ³. The conversion

³J2000 Frame Description

between LLA to ECEF, and ECEF to ECI, and vice versa are well known in literature. Thus, the only conversions that will be explained are those from BEI to BER and BER to BEI as they are system dependent.

To generate the rotation matrix between a BEI and BER frame, the Matlab's *planetEphemeris*(*JD, Primary, Secondary*) function was used. The *planetEphemeris.m* function returns the velocity and position of the secondary planet *relative* to the primary planer (or Sun) and are expressed on a frame that is aligned with the ECI frame, \mathbf{r}_{rel} , \mathbf{v}_{rel} . To obtain the rotating matrix ${}^{BEI}C^{BER}$ we defined the unit vectors as

$$\begin{aligned}\hat{\mathbf{i}} &= \frac{\mathbf{r}_{rel}}{\|\mathbf{r}_{rel}\|} \\ \hat{\mathbf{k}} &= \frac{\mathbf{r}_{rel} \times \mathbf{v}_{rel}}{\mathbf{r}_{rel} \times \mathbf{v}_{rel}} \\ \hat{\mathbf{j}} &= \hat{\mathbf{k}} \times \hat{\mathbf{i}}\end{aligned}\tag{A.6}$$

The rotation matrix between the BER to BEI frame is then simply

$${}^{BEI}C^{BER}(t) = \begin{bmatrix} \hat{\mathbf{i}} & \hat{\mathbf{j}} & \hat{\mathbf{k}} \end{bmatrix}\tag{A.7}$$

The rotating matrix described in Equation (A.7) gives the coordinate frame transformation from the barycenter ecliptic rotating frame to the barycenter ecliptic inertial coordinate system.

Conversion between BER to BEI frame

Given the matrix ${}^{BEI}C^{BER}(t)$, the current position and velocity described in the rotating frame, \mathbf{r}_{rel}^{BER} and \mathbf{v}_{rel}^{BER} , and the angular velocity of the rotating frame, $\boldsymbol{\omega}^{BER}$, it is possible to change the frames as follows

$$\begin{aligned}\mathbf{r}_{rel}^{BEI} &= {}^{BEI}C^{BER} \mathbf{r}_{rel}^{BER} \\ \mathbf{v}_{rel}^{BEI} &= {}^{BEI}C^{BER} (\mathbf{v}_{rel}^{BER} + \boldsymbol{\omega}^{BER} \times \mathbf{r}_{rel}^{BER})\end{aligned}\tag{A.8}$$

where the Transport Theorem was used to account for the rotating frame.

Conversion between BEI to BER frame

Alternatively, it is possible to switch from the inertial BEI frame to the rotational BER frame given ${}^{BER}C^{BEI}(t)$, \mathbf{r}_{rel}^{BEI} , \mathbf{v}_{rel}^{BEI} and the angular velocity $\boldsymbol{\omega}^{BEI}$. Note that since ${}^{BER}C^{BEI}(t)$ is a coordinate transform matrix the identity ${}^{BER}C^{BEI}(t) = {}^{BEI}C^{BER T}(t)$ is valid.

$$\begin{aligned}\mathbf{r}_{rel}^{BER} &= {}^{BER}C^{BEI}\mathbf{r}_{rel}^{BEI} \\ \mathbf{v}_{rel}^{BER} &= {}^{BER}C^{BEI}\left(\mathbf{v}_{rel}^{BEI} - \boldsymbol{\omega}^{BEI} \times \mathbf{r}_{rel}^{BEI}\right)\end{aligned}\tag{A.9}$$

A.3.2 Trajectory Optimizer Algorithm

The algorithms wrapper used for the trajectory optimizer is shown in Algorithm 3. The algorithm takes the following as inputs,

- t_{start} : the initial start time for when to begin looking for the minimum trajectory. Value is a time vector in the form [Year, Month, Day].
- Δt : the time discretization for the algorithm; units are in hours.
- t_{end} : the final search time for when the launch of the particular component is allowed to happen. Value is a time vector in the form [Year, Month, Day].
- Launch Loc: launch location in terms of latitude and longitude of Earth.
- Target Loc: desired target location. Currently only a halo orbit in SEL2 or EML1 are allowed

The algorithm, depending on the desire launch location, loads the manifold trajectories (as seen in line 8 or 12) from a MATLAB “.mat” data structure. These manifolds would correspond to those found by means shown in Subsection A.2.2 and seen in Figure (A-7). Each structure has the entire time history of each of the candidate stable manifold trajectories. Furthermore, despite only allowing SEL2 or EML1 as the possible launch locations, the algorithm is written to allow for new orbit destinations. The user only has to compute a new manifold “.mat” structure and add a new case to the if statement.

Algorithm 3 Discrete Trajectory Optimizer Algorithm

```
1:  $[\Delta v, t_{launch}, t_{dur}] = traj\_lookup(t_{start}, \Delta t, t_{end}, Target\ Loc, Launch\ Loc)$ 
2:  $\Delta v = NaN$ 
3:  $t_{launch} = NaN$ 
4:  $t_{dur} = NaN$ 
5: if Launch Loc = SEL2 then
6:    $primary = Sun$ 
7:    $secondary = Earth$ 
8:    $manifolds = SEL2\ Manifolds$ 
9: else if Launch Loc = EML! then
10:   $primary = Earth$ 
11:   $secondary = Moon$ 
12:   $manifolds = EML1\ Manifolds$ 
13: end if
14: for  $t_{cur} = t_{start} : \Delta t : t_{end}$  do
15:   $\mathbf{r}_{launch}^{ECI}, \mathbf{v}_{launch}^{ECI} = ECEFtoECI(t_{cur}, Launch\ Loc)$ 
16:   $\mathbf{r}_{rel}^{BER}, \mathbf{v}_{rel}^{BER} = planetEphemeris(t_{cur}, primary, secondary)$ 
17:   $BEIC^{BER} = gen\_beiCber(\mathbf{r}_{launch}^{ECI}, \mathbf{v}_{launch}^{ECI})$ 
18:   $\Delta v_{cur}, t_{dur,cur} = manifold\_transfer\_lookup(\mathbf{r}_{launch}^{ECI}, \mathbf{v}_{launch}^{ECI}, BEIC^{BER}, manifolds)$ 
19:  if  $\Delta v_{cur} < \Delta v$  then
20:     $\Delta v = \Delta v_{cur}$ 
21:     $t_{launch} = t_{cur}$ 
22:     $t_{dur} = t_{dur,cur}$ 
23:  end if
24: end for
```

The trajectory algorithm performs a discrete optimization by looping through each time value and finding the least expensive manifold trajectory during that time. This is done during the for loop routine (line 14) of the algorithm. At each instant of time, the launch pad velocity, relative to the ECI inertial frame, is computed using MATLAB's *ECEFtoECI.m* function. Additionally, the relative velocities of the primary and secondary velocity are computed using MATLAB's *planetEphemeris.m* function. The relative velocity \mathbf{r}_{rel}^{BER} , \mathbf{v}_{rel}^{BER} are then used to generate the coordinate transformation using the procedure shown in the previous section.

The function *manifold_transfer_lookup.m* performs the main step algorithm. This function takes launch position and velocity, the coordinate transformation matrix, and the manifolds to find the least expensive trajectory to reach the desired halo orbit. The function returns the fuel cost in Δv_{cur} and the time it takes for reaching the halo orbit, $t_{dur,cur}$. Finally, the algorithm keeps track of the current minimum Δv , launch time, and travel duration as shown in line 19.

Algorithm 4 Manifold Transfer Look Up Algorithm

```

1:  $\Delta v, t_{dur} = manifold\_transfer\_lookup(\mathbf{r}_{launch}^{ECI}, \mathbf{v}_{launch}^{ECI}, C^{BER}, manifolds)$ 
2:  $\Delta v = NaN$ 
3:  $t_{dur} = NaN$ 
4: for  $i = 1$  to  $N$  do
5:    $\mathbf{r}_{tar}^{BER}, \mathbf{v}_{tar}^{BER}, t_{dur,i} = manifolds\{i\}$ 
6:    $\mathbf{r}_{tar}^{BEI}, \mathbf{v}_{tar}^{BEI} = BER\_to\_BEI(\mathbf{r}_{tar}^{BER}, \mathbf{v}_{tar}^{BER})$ 
7:    $\mathbf{r}_{launch}^{BEI}, \mathbf{v}_{launch}^{BEI} = ECI\_to\_BEI(\mathbf{r}_{launch}^{ECI}, \mathbf{v}_{launch}^{ECI}, C^{BER})$ 
8:    $v_{lamb,1}, v_{lamb,2}, t_{trans} = lambert\_min(\mathbf{r}_{tar}^{BEI}, \mathbf{v}_{tar}^{BEI}, \mathbf{r}_{launch}^{BEI}, \mathbf{v}_{launch}^{BEI})$ 
9:    $\Delta v_{cur} = \|\mathbf{v}_{tar}^{BEI} - v_{lamb,2}\| + \|v_{lamb,1} - \mathbf{v}_{launch}^{BEI}\|$ 
10:  if  $\Delta v_{cur} < \Delta v$  then
11:     $\Delta v = \Delta v_{cur}$ 
12:     $t_{dur} = t_{dur,i} + t_{trans}$ 
13:  end if
14: end for

```

The algorithm for the *manifold_transfer_lookup.m* function is shown in Algorithm 4. The objective of this algorithm is to find the minimum Δv as well as the duration of travel for a given position and velocity launch location. The algorithm takes the manifolds “.mat” structure in which is assume has N possible transfer trajectories.

For each possible trajectory the launch position and velocity, $\mathbf{r}_{launch}^{ECI}$, $\mathbf{v}_{launch}^{ECI}$, are converted from the ECI to the BEI frame.

For each manifold trajectory a target position, \mathbf{r}_{tar}^{BER} , and velocity, \mathbf{v}_{tar}^{BER} , are obtained. These are again user defined. However, for this function the position is defined as the closest point of approach for a manifold trajectory to Earth. The choice of where along the manifold is it globally cheaper to enter is not addressed in this study and is left for future work.

Finally, the transfer trajectory that takes the component from the launch pad to the target location along the stable manifold trajectory is obtained by the solution to Lambert’s problem as shown in line 8. Lambert’s problem is a classic astrodynamics problem which seeks to determine a feasible orbit between two points [53]. The typical problem takes two state vectors and finds the possible orbits that could exist that transverse the two points. For this analysis the minimum energy version of the Lambert’s problem was implemented. The implementation is exactly the one given by Prussing and Conway in [53].

Algorithm 4 and 3 were implemented in MATLAB and used for the iSAT study. The code for the analysis can be found at <https://github.mit.edu/cabrales/SSL-cabrales/> or <https://github.com/acabralesh/isat-TrajDesign/>

A.4 Test Cases and Result

Two test cases were conducted to verify that the algorithm reported the least expensive, in terms of Δv , trajectory to either EML1 or SEL2. For both cases, the parameters utilized are shown in Table A.3. The time window for searching was limited to a single day in order to reduce the number of outputs launch. The discretization remained at 1 hour; however, further discretization is possible to determine the best launch time. The launch location corresponds to Cape Canaveral. Finally, an atmosphere drag cost of 1.5 km/s was added to the cost that accounts for the fuel expenditure to overcome Earth’s atmosphere at launch.

Table A.3: Test set up for transport trajectory optimizer

Input	Units	Description
Initial time, t_0	N/A	06/18/2021
Time discretization, Δt	Hours	1
Final time, t_f	N/A	06/19/2021
Launch latitude	$^\circ$	28.39
Launch longitude	$^\circ$	80.61
Atmosphere Drag Cost	km/s	1.5

A.4.1 Earth-Moon L1

The first case corresponded to finding a trajectory that reaches a halo orbit around EML1 with an amplitude of 32,000 km or around five Earth radii.

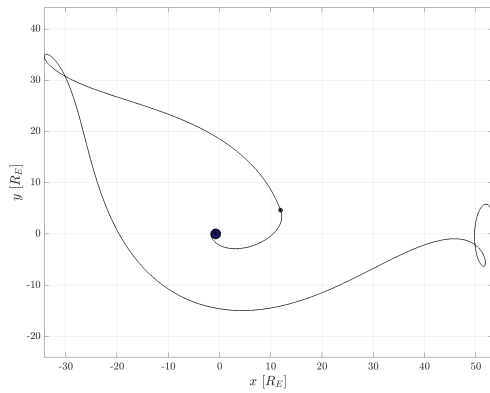
Table A.4: Test results for EML1 trajectory optimization

Parameter	Value
Launch Window	7:00:00 18-Jul-2021
Total Transport ΔV [km/s]	13.71
Total Transfer Duration [$Days$]	28.02
Halo Arrival Time	7:30:43 18-Jul-2021

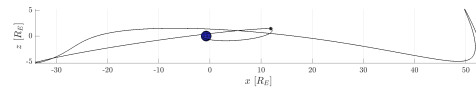
Table A.4 shows the overall results of the optimization for this test case. The trajectory departure happens at 7 am on July 18th, 2021. The duration is approximately one moon period. Finally, the total transport ΔV is 13.71 km/s which is on expected for these types of trajectories. Additionally, the iterations results produced by the optimization algorithm are shown in Table A.5. Note that the results in the iteration table does not include the atmospheric drag equivalent to 1.5 km/s .

The Instances in which the iteration produced not a number or infinite delta t (iteration 18) corresponds to a situation in which the transfer trajectory produced by the solution to the Lambert’s problem led to collision on earth. Finally, it can be seen that the ΔV cost is not constant and the lower Δv values corresponds to times from 3 am to 1 pm. This is also expected as there will be an orientation of the Earth relative to the side facing the Moon at which the least expensive trajectory is least expensive.

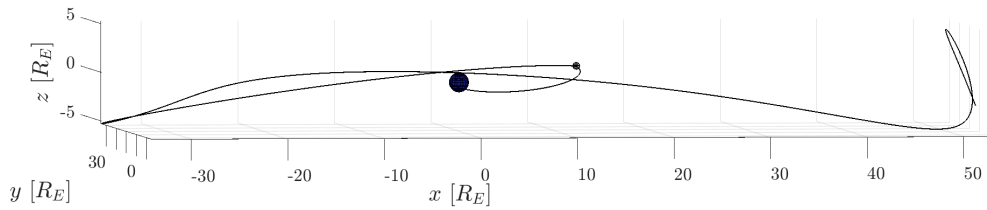
Figure (A-8) shows the proposed minimum fuel trajectory to reach EML1 halo



(a) XY-plane view



(b) Xz-plane view



(c) 3D view

Figure A-8: EML1 transport trajectory example. Figure shows the proposed trajectory from launch site to point of joining the selected halo orbit. Asterisk represents point at which the lambert transfer trajectory joins the stable manifold trajectory.

Table A.5: Iterations results for EML1 trajectory optimization

Iter.	Date	ΔV [km/s]	Duration [Days]	Arrival Date
1	18-Jun-2021 00:00:00	12.66	25.22	13-Jul-2021 05:23:14
2	18-Jun-2021 00:59:59	12.73	25.22	13-Jul-2021 06:12:31
3	18-Jun-2021 02:00:00	12.81	25.21	13-Jul-2021 07:02:55
4	18-Jun-2021 03:00:00	12.76	28.42	16-Jul-2021 12:57:43
5	18-Jun-2021 03:59:59	12.41	29.31	17-Jul-2021 11:30:17
6	18-Jun-2021 05:00:00	12.26	29.31	17-Jul-2021 12:26:07
7	18-Jun-2021 06:00:00	12.24	28.03	16-Jul-2021 06:37:01
8	18-Jun-2021 06:59:59	12.21	28.02	16-Jul-2021 07:30:43
9	18-Jun-2021 08:00:00	12.24	28.02	16-Jul-2021 08:22:24
10	18-Jun-2021 09:00:00	12.29	28.01	16-Jul-2021 09:12:33
11	18-Jun-2021 09:59:59	12.36	28.00	16-Jul-2021 10:01:44
12	18-Jun-2021 11:00:00	12.45	27.99	16-Jul-2021 10:50:38
13	18-Jun-2021 12:00:00	12.55	27.99	16-Jul-2021 11:39:59
14	18-Jun-2021 12:59:59	12.66	27.98	16-Jul-2021 12:30:31
15	18-Jun-2021 14:00:00	12.78	27.97	16-Jul-2021 13:22:53
16	18-Jun-2021 15:00:00	12.90	27.97	16-Jul-2021 14:17:38
17	18-Jun-2021 15:59:59	13.03	27.97	16-Jul-2021 15:15:11
18	18-Jun-2021 17:00:00	Inf	NaN	NaN
19	18-Jun-2021 18:00:00	13.38	25.26	14-Jul-2021 00:12:33
20	18-Jun-2021 18:59:59	13.09	25.26	14-Jul-2021 01:10:57
21	18-Jun-2021 20:00:00	12.85	25.25	14-Jul-2021 02:06:52
22	18-Jun-2021 21:00:00	12.70	25.25	14-Jul-2021 03:00:31
23	18-Jun-2021 21:59:59	12.62	25.24	14-Jul-2021 03:52:15
24	18-Jun-2021 23:00:00	12.59	25.24	14-Jul-2021 04:42:31
25	19-Jun-2021 00:00:00	12.60	25.23	14-Jul-2021 05:31:52

orbit. The trajectory begins at Earth, at which a transfer trajectory obtained by a solution to the Lambert’s problem connects the launch site with the stable manifold—the beginning of which is shown by the asterisk at $(11.87, 4.69, 1.44) [R_E]$. The stable manifold trajectory then joins the halo orbit at $(53.11, 4.23, -2.43) [R_E]$.

A.4.2 Sun-Earth L2

The same initial time, time window, and final time were chosen for the SEL2 example. The results are shown in Table A.6. The trajectory takes approximately 248 days to complete the journey to join the halo orbit. Interestingly, the Δv associated with

joining a halo orbit in SEL2 is lower than EML1. This result of lower cost as well as magnitude cost is consistent with results reported by [61].

Table A.6: Test results for SEL2 trajectory optimization

Parameter	Value
Launch Window	23:00:00 18-Jul-2021
Total Transport ΔV [km/s]	13.35
Total Transfer Duration [$Days$]	248.44
Halo Arrival Time	09:31:02 22-FEB-2022

Table A.7 shows the individual iteration results for the SEL2 optimization. As with EML1, the iterations at which the $\Delta V = Inf$ represents instances at which the solution to Lambert’s problem is unable to trajectories that does not collide with Earth or the Moon. However, this does not mean that there does not exist a stable manifold trajectory capable of transporting the component at that time. Instead it is just an artifact of the discrete optimization not using enough possible manifold trajectories.

A.5 Discussion and Future Work for the Trajectory Optimizer

This Appendix presented a discrete optimizer algorithm to find fuel efficient trajectories to a halo orbit in Earth-Moon Libration point 1 or Sun-Earth Libration point 2. The algorithm presents a possible “first cut” to the problem of designing trajectories of multiple components to the same orbit for in-space assembly. The algorithm has flexibility in allowing for different types of orbits for inclusion of Near Rectilinear Halo Orbits, or other types of orbits that are of interests for in-space assembly of space telescopes. Finally, the magnitude and duration of the results are consistent with those present in literature.

Future work for this algorithm includes adding different nominal orbit options for a more complete analysis. Additionally, the issue of phasing, or the process in which all the components around the nominal orbit converge to once another, is not

Table A.7: Iterations results for SEL2 trajectory optimization

Iter.	Date	ΔV [km/s]	Duration [Days]	Arrival Date
1	18-Jun-2021 00:00:00	11.87	248.42	21-Feb-2022 10:10:47
2	18-Jun-2021 00:59:59	11.90	248.41	21-Feb-2022 10:49:36
3	18-Jun-2021 02:00:00	11.94	248.40	21-Feb-2022 11:28:54
4	18-Jun-2021 03:00:00	12.00	248.38	21-Feb-2022 12:10:07
5	18-Jun-2021 03:59:59	12.06	248.37	21-Feb-2022 12:54:33
6	18-Jun-2021 05:00:00	12.12	248.36	21-Feb-2022 13:43:20
7	18-Jun-2021 06:00:00	12.19	248.36	21-Feb-2022 14:37:16
8	18-Jun-2021 06:59:59	12.85	232.43	05-Feb-2022 17:14:59
9	18-Jun-2021 08:00:00	12.96	232.42	05-Feb-2022 18:05:58
10	18-Jun-2021 09:00:00	13.07	232.42	05-Feb-2022 18:59:29
11	18-Jun-2021 09:59:59	13.19	232.41	05-Feb-2022 19:56:02
12	18-Jun-2021 11:00:00	Inf	NaN	NaT
13	18-Jun-2021 12:00:00	Inf	NaN	NaT
14	18-Jun-2021 12:59:59	Inf	NaN	NaT
15	18-Jun-2021 14:00:00	Inf	NaN	NaT
16	18-Jun-2021 15:00:00	Inf	NaN	NaT
17	18-Jun-2021 15:59:59	Inf	NaN	NaT
18	18-Jun-2021 17:00:00	Inf	NaN	NaT
19	18-Jun-2021 18:00:00	12.42	247.24	20-Feb-2022 23:43:55
20	18-Jun-2021 18:59:59	12.17	248.47	22-Feb-2022 06:18:04
21	18-Jun-2021 20:00:00	12.02	248.47	22-Feb-2022 07:13:11
22	18-Jun-2021 21:00:00	11.92	248.46	22-Feb-2022 08:03:15
23	18-Jun-2021 21:59:59	11.87	248.45	22-Feb-2022 08:48:53
24	18-Jun-2021 23:00:00	11.85	248.44	22-Feb-2022 09:31:00
25	19-Jun-2021 00:00:00	11.87	248.42	22-Feb-2022 10:10:48

addressed in this problem formulation. This must be addressed as part of a stage needed for rendezvous and docking of all the components. Finally, an analysis should be conducted in finding the point along the stable manifold trajectory for which minimum fuel transfer occurs as currently the closest approach to Earth is being used as the entry point of the manifold.

Appendix B

Analytical and Quasi-Analytical Solutions for Impulsive Docking Problems

This appendix presets the impulsive versions for the solution to the flat spin and general spin with spherical inertia. Having this situation is equivalent to solving the docking problem with no control saturation on the Chaser translational control. Finally, the results from this appendix provide the main functions used in the AMPC algorithm for generating the docking trajectory.

B.1 Impulsive Solution to Flat Spin

The procedure for obtaining the impulsive solution for the Flat Spin is by letting $u_{sat} \rightarrow \infty$ which effectively makes $t_1 \rightarrow 0$ from Equation (3.23). Thus the trajectory time history between the distance from the Chaser to Target given by Equation 3.21 simplifies to

$$x(t) = c_1 e^{\omega(t-t_f)} + c_2 e^{-\omega(t-t_f)} \quad t \in [t_0, t_f] \quad (\text{B.1})$$

where c_1, c_2 , and t_f are unknown. Furthermore, it is assumed that t_0 is known and

defined as the desired time for beginning of the trajectory. It is possible to enforce the boundary conditions that $x(t_0) = R_0$ and $x(t_f) = R_f$, which gives c_1 and c_2

$$\begin{aligned} c_1 &= \frac{1}{e^{-\omega(t_0-t_f)} - e^{\omega(t_0-t_f)}} (R_f e^{-\omega(t_0-t_f)} - R_0) \\ c_2 &= \frac{1}{e^{-\omega(t_0-t_f)} - e^{\omega(t_0-t_f)}} (R_0 - R_f e^{\omega(t_0-t_f)}) \end{aligned} \quad (\text{B.2})$$

which reduces the unknowns of Eq. (B.1) to,

$$x(t) = \frac{(R_f e^{-\omega(t_0-t_f)} - R_0) e^{\omega(t-t_f)} + (R_0 - R_f e^{\omega(t_0-t_f)}) e^{-\omega(t-t_f)}}{e^{-\omega(t_0-t_f)} - e^{\omega(t_0-t_f)}} \quad (\text{B.3})$$

Finding the value of t_f can be obtained by enforcing the boundary condition of soft dock at the final time

$$t_f = t_0 - \frac{\text{sign}(\omega)}{\omega} \log \left(\frac{R_0}{R_f} - \sqrt{\left(\frac{R_0}{R_f} \right)^2 - 1} \right) \quad (\text{B.4})$$

Finally, given t_f it is possible to compute the initial velocity of the trajectory in order to achieve soft docking:

$$\dot{x}(t_0) = -R_f \text{sign}(\omega) \omega \sqrt{\left(\frac{R_0}{R_f} \right)^2 - 1} \quad (\text{B.5})$$

It is easy to see the trends for $\dot{x}(t_0)$ from Equation (B.5). As expected, $\dot{x}(t_0)$ increases with an increase in R_f , which is consistent with the fact that if the Chaser must transverse a longer distance it should spend more fuel to achieve a larger $\dot{x}(t_0)$. The same can be said for an increase in $|\omega|$, as the dynamics will push the Chaser away from the target, so a larger value of initial speed is needed to reach the Target at t_f .

If Equation (B.4) is substituted into Equation (B.3) the trajectory is simplified to

$$x(t) = \frac{R_f}{2} (e^{\omega(t-t_f)} + e^{-\omega(t-t_f)}) \quad (\text{B.6})$$

The control time history on the normal direction can be computed by $u_{y/z} = 2\omega\dot{x}(t)$. Finally, the total cost can be computed by $\int |2\omega\dot{x}(t)|dt + |\dot{x}(t_0)|$.

B.2 Impulsive Solution to General Spin with Spherical Inertia

The same approach for obtaining the impulsive solution was taken for the thruster saturated solutions presented in Section 3.4. As stated in Section 3.4, the difference between the Bang-Off controller and Bang-Off-Bang controller corresponds to the value of γ given in Equation (3.33):

$$\gamma = \frac{|\omega_y \omega_x| + |\omega_z \omega_x|}{\omega_y^2 + \omega_z^2}$$

B.2.1 Impulsive Bang-Off Solution

When $\gamma \leq 1$ the solution per PMP is that of a Bang-Off controller (see Section 3.4.2). Since the trajectory for the General Spin Spherical Inertia problem when the controller is Bang-Off is very similar to that of the flat spin, the solution given by Equations (B.4) and (B.5) can be adapted for the more complicated problem as follows:

$$t_f = t_0 - \frac{1}{\omega_{eff}} \log \left(\frac{R_0}{R_f} - \sqrt{\left(\frac{R_0}{R_f}\right)^2 - 1} \right) \quad (\text{B.7})$$

$$\dot{x}(t_0) = -R_f \omega_{eff} \sqrt{\left(\frac{R_0}{R_f}\right)^2 - 1} \quad (\text{B.8})$$

where $\omega_{eff} = \sqrt{\omega_y^2 + \omega_z^2}$. Also note that due to the nature of having ω_{eff} in the solution for the trajectory in Equation (3.34), there is no need for the $\text{sign}(\omega)$ function that appeared in the Flat Spin case. For completeness, the optimal trajectory for the impulsive spin will be given by

$$x(t) = \frac{R_f}{2} (e^{\omega_{eff}(t-t_f)} + e^{-\omega_{eff}(t-t_f)}) \quad (\text{B.9})$$

The control time history can be computed as $u_y = 2\omega_z \dot{x} + x\omega_y \omega_x$ and $u_z = -2\omega_y \dot{x} + x\omega_z \omega_x$. The total cost is again $\int_{t_0}^{t_f} (|2\omega_z \dot{x} + \omega_y| dt + |\dot{x}(t_0)|)$ and can be solved

either numerically or analytically.

B.2.2 Impulsive Bang-Off-Bang Solution

Whereas the case for the Bang-Off controller the solution for the impulsive case can be solved analytically, the Bang-Off-Bang controller corresponding to the case where $\gamma > 1$ cannot be solved with a closed form solution. Nevertheless, the same approach can be taken for obtaining a single trajectory when $u_{sat} \rightarrow \infty$. Thus, Equation (3.37) will simplify to a single equation for the form,

$$x(t) = c_1 e^{\omega_{eff}(t-t_f)} + c_2 e^{-\omega_{eff}(t-t_f)} \quad t \in [t_0, t_f] \quad (\text{B.10})$$

where in Equation (B.10) there are three unknowns, namely c_1 , c_2 and t_f , but there are only two equations available, namely $x(t_0) = R_0$ and $x(t_f) = R_f$. The extra degree of freedom comes from having the final velocity of the Chaser relative to the Target, $\dot{x}(t_f)$, free. In essence, this removes the soft docking condition in the trajectory equation. Applying the boundary conditions removes the terms for c_1 and c_2 which yields,

$$x(t) = \frac{(R_f e^{-\omega_{eff}(t_0-t_f)} - R_0) e^{\omega_{eff}(t-t_f)} + (R_0 - R_f e^{\omega_{eff}(t_0-t_f)}) e^{-\omega_{eff}(t-t_f)}}{e^{-\omega_{eff}(t_0-t_f)} - e^{\omega_{eff}(t_0-t_f)}} \quad (\text{B.11})$$

which is of the same form as Eq. (B.3). Having a form $x(t)$ and its subsequent derivative to obtain $\dot{x}(t)$ is needed to formulate a minimization problem similar to that in Section 3.4.4 to obtain the optimal value for t_f .

Optimization Method for t_f

The optimization problem for finding t_f can be formulated as follows,

$$\begin{aligned} \min_{t_f} \quad & f(t_f) \\ \text{Subject to} \quad & t_f \geq 0 \end{aligned} \quad (\text{B.12})$$

where,

$$f(t_f) = \int_{t_0}^{t_f} \left(|2\omega_z \dot{x}(t) + \omega_y \omega_x| + |-2\omega_y \dot{x}(t) + \omega_z \omega_x| \right) dt + |\dot{x}(t_0)| + |\dot{x}(t_f)| \quad (\text{B.13})$$

where $x(t)$ is given by Equation (B.11) and $|\dot{x}(t_0)|$, $|\dot{x}(t_f)|$ represents the impulsive thrust that the Chaser must produce to join the trajectory and end the trajectory at zero relative speed, respectively. Equation (B.12) can be solved by a simple minimization method such as Matlab `fminbnd.m`¹. Additionally, a lower bound or initial guess for the minimization problem is given by Equation (B.7). In fact, a simple bisection method can be formulated by setting the lower and upper bound of t_f to $[t_{f,BO}^*, 2t_{f,BO}^*]$ where $t_{f,BO}^*$ is the final time for the Bang-Off controller.

Finally, for completeness, finding the final time that solves the optimization problem in (B.12), the initial velocity from the guidance is given by,

$$\dot{x}(0) = \frac{\omega_{eff}}{e^{-\omega_{eff}t_f} - e^{\omega_{eff}t_f}} [R_0 (-\omega_{eff}t_f + e^{\omega_{eff}t_f}) - 2R_f] \quad (\text{B.14})$$

Analysis of $\dot{x}(t_0)$, $\dot{x}(t_f)$ vs t_f

Since an analytical expression cannot be easily obtained for the optimization problem in Equation (B.12), an exploration of the trends for $\dot{x}(t_0)$ and $\dot{x}(t_f)$ can be performed. Without loss of generalization it is possible to choose $t_0 = 0$ which yields a value for $\dot{x}(0)$ and $\dot{x}(t_f)$ to be:

$$\begin{aligned} \dot{x}(0) &= \frac{\omega_{eff}}{e^{-\omega_{eff}t_f} - e^{\omega_{eff}t_f}} [R_0 (-\omega_{eff}t_f + e^{\omega_{eff}t_f}) - 2R_f] \\ \dot{x}(t_f) &= \frac{\omega_{eff}}{e^{-\omega_{eff}t_f} - e^{\omega_{eff}t_f}} [2R_0 - R_f (-\omega_{eff}t_f + e^{\omega_{eff}t_f})] \end{aligned} \quad (\text{B.15})$$

To analyze the trends of Equation (B.15), a plot was made in which the value of the final time was changed. The parameters used for the study are in Table B.1. Note that despite the choice of values, the overall shape of the plot will remain the

¹It should be noted that although an optimization method was used for solving the Bang-Off-Bang controller, an analytical equation can be sought after by carefully resolving the integral in Equation (B.13)

Table B.1: Nominal Values for Analysis of $\dot{x}(t_0)$, $\dot{x}(t_f)$ vs. t_f

Parameter	Value
Initial Time t_f (s)	0
Initial Radius R_0 (m)	10
Final Radius R_f (m)	5
Angular Velocity ω ($^\circ$ /s)	[50, 20, 20]

same.

Figure (B-1) shows a plot of the initial and final velocity of the Chaser relative to the Target as a function of final time. As expected, in order for there to reach a lower relative distance, the initial Chaser velocity must be less than zero. On the other hand, depending on the desired value of t_f , the final velocity will either be negative, zero or positive.

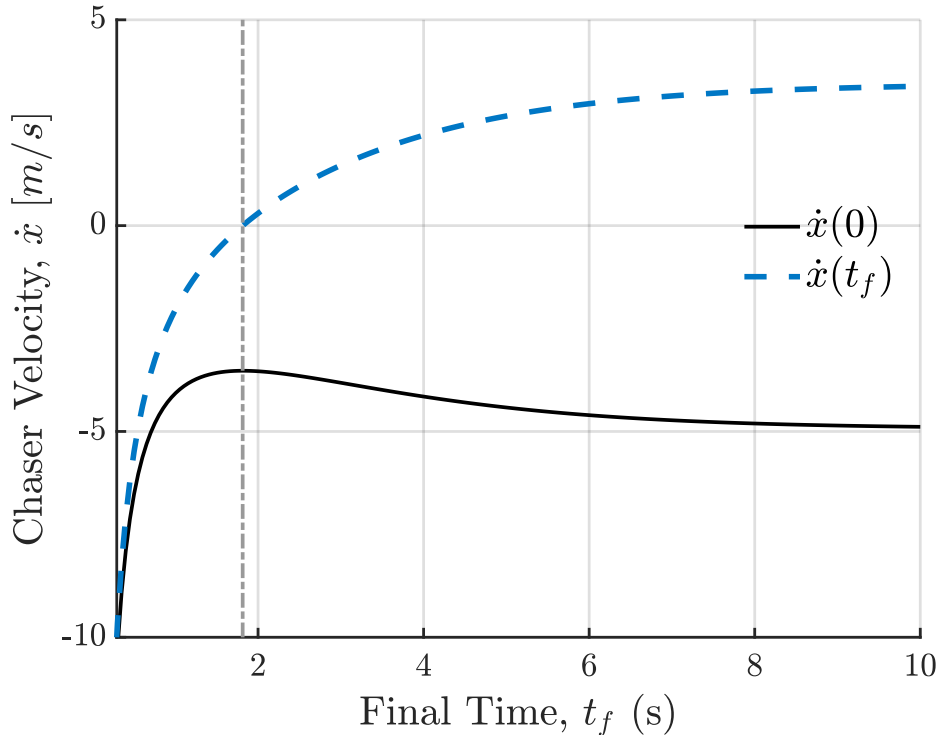


Figure B-1: Behavior of $\dot{x}(0)$, $\dot{x}(t_f)$ as a function of t_f for General Spin Spherical Inertia Impulsive Solution

The trends in Figure (B-1) are as expected. For example, selecting a final time close to zero causes both the initial and final velocities to approach infinity—the Chaser

would require an infinite amount of velocity to move a finite distance amount. Two points of interest are the values in which the sum of the initial and final velocity are the least (noted by the grey vertical dash line), and the limits for both the initial and final velocity.

Using L'Hôpital's rule, the initial value becomes $\lim_{t_f \rightarrow \infty} \dot{x}(0) = -\omega_{eff} R_0$ and $\lim_{t_f \rightarrow \infty} \dot{x}(t_f) = \omega_{eff} R_f$. The values for the two are equivalent to the norm of the velocity, in inertial coordinates, of a point located at either $x(0)$ and $x(t_f)$. In other words, $\lim_{t_f \rightarrow \infty} \dot{x}(\#) = || [x(\#), 0, 0]^T \times \boldsymbol{\omega} ||$.

The location of the minimum sum of $\dot{x}(0)$, $\dot{x}(t_f)$ is equivalent as finding the maximum of $\dot{x}(0)$. This occurs at $t_f^* = \frac{1}{\omega_{eff}} \log \left(R_0/R_f + \sqrt{(R_0/R_f)^2 - 1} \right)$ with $\dot{x}(t_0) = -R_f \omega_{eff} \sqrt{(R_0/R_f)^2 - 1}$, $\dot{x}(t_f) = 0$. This is equivalent to the Impulsive Bang-Off solution, as expected.

Bibliography

- [1] G. Tommei, A. Milani, and A. Rossi. Orbit determination of space debris: Admissible regions. *Celestial Mechanics and Dynamical Astronomy*, 97(4):289–304, 2007.
- [2] ESA Space Debris Office. Space Debris by the Numbers, 2019.
- [3] Donald J. Kessler and Burton G. Cour-Palais. Collision Frequency of Artificial Satellites: the Creation of a Debris Belt. *J Geophys Res*, 83(A6):2637–2646, 1978.
- [4] Gregory W Ojakangas, H Cowardin, and N Hill. Toward realistic dynamics of rotating orbital debris, and implications for lightcurve interpretation. 2011.
- [5] Northrop Grumman Corporation. Mission extension vehicle, 2019.
- [6] NASA Goddard Space Flight Center. Restore-1 robotic servicing mission, 2019.
- [7] William D Sanchez, Keenan Albee, Rosemary Davidson, Ryan De Freitas Bart, Alejandro Cabrales Hernandez, and Jeffrey Hoffman. A Preliminary Architecture for In-Space Assembled Telescopes. In *70th International Astronautical Congress*, Washington D.C., 2019. International Astronautical Federation.
- [8] Antonio Terán Espinoza, Hailee Hettrick, Keenan Albee, Alejandro Cabrales Hernandez, and Richard Linares. End-to-end framework for close proximity in-space robotic missions.
- [9] Wigbert Fehse. *Automated rendezvous and docking of spacecraft*, volume 16. Cambridge university press, 2003.
- [10] Max William Yates et al. *Angles-only navigation technique for maneuver-free spacecraft proximity operations*. PhD thesis, Massachusetts Institute of Technology, 2017.
- [11] D Geller. Analysis of the relative attitude estimation and control problem for satellite inspection and orbital rendezvous. *The Journal of the Astronautical Sciences*, 55(2):195–214, 2007.

- [12] A Teran and C Jewison. Performance analysis of distinct control and estimation frameworks for spacecraft proximity operations involving severe uncertainties. In *Proceedings of 10th ESA International Guidance Navigation and Control Conference*. ESA, 2017.
- [13] Andrea Valmorbida, Mattia Mazzucato, and Marco Pertile. Calibration procedures of a vision-based system for relative motion estimation between satellites flying in proximity. *Measurement*, 151:107161, 2020.
- [14] Hailee Elida Hettrick. *Autonomous rendezvous and docking with tumbling, uncooperative, and fragile targets under uncertain knowledge*. 2019.
- [15] Keenan Albee, Monica Ekal, Rodrigo Ventura, and Richard Linares. Combining parameter identification and trajectory optimization: Real-time planning for information gain. *arXiv preprint arXiv:1906.02758*, 2019.
- [16] Angel Flores-Abad, Ou Ma, Khanh Pham, and Steve Ulrich. A review of space robotics technologies for on-orbit servicing. *Progress in Aerospace Sciences*, 68:1–26, 2014.
- [17] Michael Neufeld. The world’s first space rendezvous, 2015.
- [18] Koji Yamanaka, Kiyomi Yokota, Katsuhiko Yamada, Hiroshi Koyama, Katsumi Tsukahara, Shoji Yoshikawa, and Taichi Nakamura. Guidance and navigation system design of r-bar approach for rendezvous and docking. In *17th AIAA International communications satellite systems conference and exhibit*, page 1299, 1998.
- [19] Matthew A Vavrina, C Eugene Skelton, Keith D DeWeese, Bo J Naasz, David E Gaylor, and Christopher D’Souza. Safe rendezvous trajectory design for the restore-l mission. In *Proceedings of the 29th AAS/AIAA Space Flight Mechanics Meeting*, 2019.
- [20] Jonas Radtke, Christopher Kebschull, and Enrico Stoll. Interactions of the space debris environment with mega constellations: Using the example of the OneWeb constellation. *Acta Astronautica*, 131(May 2016):55–68, 2017.
- [21] Oskar Von Stryk and Roland Bulirsch. Direct and indirect methods for trajectory optimization. *Annals of operations research*, 37(1):357–373, 1992.
- [22] Costantinos Zagaris and Marcello Romano. Applied reachability analysis for spacecraft rendezvous and docking with a tumbling object. In *2018 space flight mechanics meeting*, page 2220, 2018.
- [23] I Michael Ross and Mark Karpenko. A review of pseudospectral optimal control: From theory to flight. *Annual Reviews in Control*, 36(2):182–197, 2012.

- [24] Michael A. Patterson and Anil V. Rao. Gpops-ii: A matlab software for solving multiple-phase optimal control problems using hp-adaptive gaussian quadrature collocation methods and sparse nonlinear programming. *ACM Trans. Math. Softw.*, 41(1):1:1–1:37, October 2014.
- [25] Donald E. Kirk. *Optimal control theory. [electronic resource] : an introduction*. Mineola, N.Y. : Dover Publications, 2004., 2004.
- [26] George A Boyarko. Spacecraft guidance strategies for proximity maneuvering and close approach with a tumbling object. Technical report, Naval Postgraduate School, 2010.
- [27] George Boyarko, Oleg Yakimenko, and Marcello Romano. Optimal rendezvous trajectories of a controlled spacecraft and a tumbling object. *Journal of Guidance, Control, and dynamics*, 34(4):1239–1252, 2011.
- [28] Farhad Aghili. A prediction and motion-planning scheme for visually guided robotic capturing of free-floating tumbling objects with uncertain dynamics. *IEEE Transactions on Robotics*, 28(3):634–649, 2012.
- [29] Oleg A Yakimenko. Direct method for rapid prototyping of near-optimal aircraft trajectories. *Journal of Guidance, Control, and Dynamics*, 23(5):865–875, 2000.
- [30] Amir Mehdi Yazdani, Karl Sammut, Andrew Lammas, and Youhong Tang. Real-time quasi-optimal trajectory planning for autonomous underwater docking. In *2015 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS)*, pages 15–20. IEEE, 2015.
- [31] Jacopo Ventura, Marco Ciarcia, Marcello Romano, and Ulrich Walter. Fast and near-optimal guidance for docking to uncontrolled spacecraft. *Journal of Guidance, Control, and Dynamics*, 40(12):3138–3154, 2017.
- [32] Stanislaw Zak. An introduction to model-base predictive control, 2017.
- [33] Qi Li, Jianping Yuan, Bo Zhang, and Chen Gao. Model predictive control for autonomous rendezvous and docking with a tumbling target. *Aerospace Science and Technology*, 69:700–711, 2017.
- [34] Hyeonjun Park, Richard Zappulla, Costantinos Zagaris, Josep Virgili-Llop, and Marcello Romano. Nonlinear model predictive control for spacecraft rendezvous and docking with a rotating target. *Advances in the Astronautical Sciences*, 160:1135–1148, 2017.
- [35] David C. Sternberg. *Optimal docking to tumbling objects with uncertain properties*. PhD thesis, 2017.
- [36] Satellite relative motion SDRE-based control for capturing a noncooperative tumbling object. *Proceedings of 9th International Conference on Recent Advances in Space Technologies, RAST 2019*, pages 253–260, 2019.

- [37] F. P. J. Rimrott. *Introductory Attitude Dynamics*. Mechanical engineering series. New York : Springer-Verlag, c1989., 1989.
- [38] Ling Jiang, Yue Wang, and Shijie Xu. Integrated 6-dof orbit-attitude dynamical modeling and control using geometric mechanics. *International Journal of Aerospace Engineering*, pages 1 – 13, 2017.
- [39] Gang Zhao, Lin-li Zhou, and Xi-yue Huang. Modeling and simulation of formation flight for spacecrafts with 6 dof based on sliding mode controls. *Journal of Sichuan University*, 43(1):161 – 167, 2011.
- [40] Pablo Ghiglini, Jason L. Forshaw, and Vaios J. Lappas. Optal: Optimal quaternion tracking using attitude error linearization. *IEEE Transactions on Aerospace and Electronic Systems, Aerospace and Electronic Systems, IEEE Transactions on, IEEE Trans. Aerosp. Electron. Syst*, 51(4):2715 – 2731, 2015.
- [41] Gui Haichao and A.H.J. de Ruiter. Quaternion invariant extended kalman filtering for spacecraft attitude estimation. *Journal of Guidance, Control, and Dynamics*, 41(4):863 – 878, 2018.
- [42] Trent Yang. *Optimal thruster selection with robust estimation for formation flying applications*. $\hat{\text{A}}\text{I}2003.$, 2003.
- [43] Daniel Kucharski, Georg Kirchner, Franz Koidl, Cunbo Fan, Randall Carman, Christopher Moore, Andriy Dmytrotsa, Martin Ploner, Giuseppe Bianco, Mikhailo Medvedskij, et al. Attitude and spin period of space debris envisat measured by satellite laser ranging. *IEEE Transactions on Geoscience and Remote Sensing*, 52(12):7651–7657, 2014.
- [44] B Bastida Virgili, S Lemmens, and H Krag. Investigation on envisat attitude motion. In *e. Deorbit workshop*, 2014.
- [45] Rudranarayan Mukherjee, N Siegler, H Thronson, K Aaron, J Arenberg, P Backes, A Barto, K Belvin, L Bowman, D Calero, et al. When is it worth assembling observatories in space? In *Bulletin of the American Astronomical Society*, volume 51, 2019.
- [46] Daniel Clery. Nasa is planning four of the largest space telescopes ever. but which one will fly?, 2018.
- [47] Marc Postman, Vic Argabright, Bill Arnold, David Aronstein, Paul Atcheson, Morley Blouke, Tom Brown, Daniela Calzetti, Webster Cash, Mark Clampin, et al. Advanced technology large-aperture space telescope (atlast): A technology roadmap for the next decade. *arXiv preprint arXiv:0904.0941*, 2009.
- [48] Keenan Eugene Sumner Albee. *Toward optimal motion planning for dynamic robots: applications on-orbit*. PhD thesis, Massachusetts Institute of Technology, 2019.

- [49] Wayne H Yu and Karen Richon. Launch window trade analysis for the james webb space telescope. 2014.
- [50] Natasha Bosanac, Cassandra M Webster, Kathleen C Howell, and David C Folta. Trajectory design and station-keeping analysis for the wide field infrared survey telescope mission. In *AAS/AIAA Astrodynamics Specialist Conference*, 2017.
- [51] Ryan J Whitley, Diane C Davis, Laura M Burke, Brian P McCarthy, Rolfe J Power, Melissa L McGuire, and Kathleen C Howell. Earth-moon near rectilinear halo and butterfly orbits for lunar surface exploration. In *AAS/AIAA Astrodynamics Specialists Conference*, 2018.
- [52] David C Folta and Cassandra Webster. Transfer trajectory options for servicing sun-earth-moon libration point missions. 2019.
- [53] J Prussing and B Conway. Orbital mechanics.
- [54] Roger R Bate, Donald D Mueller, Jerry E White, and William W Saylor. *Fundamentals of astrodynamics*. Courier Dover Publications, 2020.
- [55] Robert W Farquhar and Ahmed A Kamel. Quasi-periodic orbits about the translunar libration point. *Celestial mechanics*, 7(4):458–473, 1973.
- [56] Robert Willard Farquhar. *The utilization of halo orbits in advanced lunar operations*, volume 6365. National Aeronautics and Space Administration, 1971.
- [57] David L Richardson. Analytic construction of periodic orbits about the collinear points. *Celestial mechanics*, 22(3):241–253, 1980.
- [58] Kathleen Connor Howell. Three-dimensional, periodic, halo orbits. *Celestial mechanics*, 32(1):53–71, 1984.
- [59] Brian T Barden. Using stable manifolds to generate transfers in the circular restricted problem of three bodies. *Master degree thesis. West Lafayette: School of Aeronautics and Astronautics, Purdue University*, 1994.
- [60] Vineet K Srivastava, Jai Kumar, and Badam Singh Kushvah. The effects of oblateness and solar radiation pressure on halo orbits in the photogravitational sun-earth system. *Acta Astronautica*, 129:389–399, 2016.
- [61] Wendell W Mendell and Steven Hoffman. Strategic considerations for cislunar space infrastructure. In *IAF, International Astronautical Congress, 44 th, Graz, Austria*, 1993.