LIMITED-DUTY-CYCLE SATELLITE FORMATION CONTROL VIA DIFFERENTIAL DRAG

THESIS

Talon A. Townley, 2nd Lt, USAF
AFIT-ENY

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, the United States Department of Defense or the United States Government. This is an academic work and should not be used to imply or infer actual mission capability or limitations.

AFIT-ENY

LIMITED-DUTY-CYCLE SATELLITE FORMATION CONTROL VIA DIFFERENTIAL DRAG

THESIS

Presented to the Faculty

Department of Aeronautics and Astronautics

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Astronautical Engineering

Talon A. Townley, BS

2nd Lt, USAF

March 2019

AFIT-ENY

LIMITED-DUTY-CYCLE SATELLITE FORMATION CONTROL VIA DIFFERENTIAL DRAG

Talon A. Townley, BS
2nd Lt, USAF

Approved:

Lt Col Kirk W. Johnson	Date
Maj Joshuah A. Hess	Date
Dr. Richard G. Cobb	Date

Abstract

As CubeSat formation flying missions relying on differential drag control become increasingly common, additional missions based on this control must be studied. A mission planning tool is investigated to control the relative spacing of a CubeSat formation where differential drag is the sole control mechanism. System performance is investigated under varying perturbations and a range of system parameters, including limiting the control duty cycle. Optimal solutions based on using a pseudo spectral numerical solver, GPOPS-II, to minimize maneuver time. This study includes the development of a mission planning tool to work with the modeled CubeSat mission to calculate optimal maneuvers for its mission architecture. The effects of mission altitude, solar cycle, various maneuver sizes and formations, limited control, various computational methods, and error checkers were evaluated. The mission planning tool developed can properly execute all desired run parameters and options, though it suffers from computational complexity. Pseudo spectral methods executed in MatLab were determined to be poorly suited to the problem due to memory requirements involved. Limited duty cycle control can be applied with differential drag with varying effectiveness dependent on mission parameters.

Acknowledgements

I would like to thank the Space and Naval Warfare Systems Command for supporting this research.

Talon A. Townley

Table of Contents

Page

Abstract iv

Acknowledgements v

List of Figures viii

List of Tables x

List of Abbreviations xi

I. Introduction 1

1.1 Cube Satellites 1
1.2 Differential Drag 3
1.3 Problem Statement 5

II. Literature Review 7

2.1 Development of Differential Drag Control 7
2.2 Eclipse Limited Control 9
2.3 Summary 10

III. Methodology 11

3.1 Problem Dynamics 11
3.1.1 Model Properties 11
3.1.2 Nonlinear Relative Dynamics 12
3.1.3 Atmospheric Model 18
3.2 Optimal Control 23
3.2.1 Solver Algorithm 23
3.2.2 Algorithm Input 24
3.2.3 Parameter Definition 28
3.2.4 Initial Guess 35
3.2.5 GPOPS-II Bounds and Guess Definition 39
3.2.6 GPOPS-II Initialization and Execution 42
3.2.7 Post Processing 47
3.3 Summary 49

IV. Implementation and Analysis 51

4.1 Resizing Maneuvers 51
4.2 Approximated Dynamics 55
4.3 Error Checker 56
4.4 Initial Separation Control 60

Page

4.5 Formation Keeping 63
4.6 Altitude Effects 64
4.7 Solar Cycle Effects 66
4.8 Implementing Limited Control 67
4.9 Summary 71

V. Conclusions and Recommendations 73

5.1 Review 73
5.2 Tool Evaluation 73
5.3 Potential Future Research 74
5.4 Conclusion 76

A. Data Tables 77

Bibliography 78

List of Figures

Figure		Page
1	Pumpkin Inc Supernova Bus	12
2	Comparison between the full dynamics with and without relative acceleration, and the unperturbed nonlinear dynamics with relative acceleration	17
3	Solver Algorithm	25
4	The location of the equilibrium path compared to the path of minimum in-plane acceleration under J_2 effects.	29
5	Magnitude of the in-plane acceleration vector as a function of position, and a plot if its minimum values based on y location.	30
6	Uncontrolled Spacecraft Motion.	31
7	Component In-Plane Accelerations	33
8	1 m Maneuver	51
9	10 m Maneuver	51
10	100 m Maneuver	52
11	1 km Maneuver	52
12	7 km Maneuver	53
13	Maneuver Time vs Size	54
14	Figure 13 on a log scale	54
15	1 m and 10 m Maneuvers with Assumptions	55
16	100 m and 1 km Maneuvers with Assumptions	55
17	1 km Error Checked Solution and Iterations	57
18	10 km Error Checked Solution	57
19	10 km Error Checked Solution's Iterations	58

Figure Page

20 1 km Maneuver with Interval Error Correction, every 10000 seconds 59

21 10 km Maneuver with Interval Error Correction, every 20000 seconds 59

22 1 cm/s Initial Separation Condition 60

23 10 cm/s Initial Separation Condition 61

24 10 cm/s Initial Separation Condition with an Error Checker Applied 61

25 Initial Separation Velocity vs Maximum Separation During Maneuver 62

26 100 m Formation Keeping for 10000s 63

27 100 km Formation Keeping for 10000s 63

28 Formation Keeping Max Drift vs Location 64

29 10 m Maneuver at a 400km Altitude 64

30 10 m Maneuver at a 800km Altitude 65

31 Maneuver Time vs Altitude 66

32 10 m Maneuver at Solar Minimum 66

33 Maneuver Time vs Start Time 67

34 10 m Maneuver without limits 67

35 Eclipse Limited 10 m Maneuver 68

36 100 m Maneuver without limits 68

37 Eclipse Limited 100 m Maneuver 69

38 Limit Region Size vs 10 m Maneuver Time 69

39 Sample Solution Algorithm Output 77

List of Tables

Table		Page
1	Constituent Data	20
2	l Approximation Data	20
3	Coefficients for Approximating ρ_{125}	20
4	Maneuver Final Position Error	56
5	Maneuver Final Position Error	59
6	Data for Figure 38	70

List of Abbreviations

Abbreviation		Page
LEO	Low Earth Orbit	4
GEO	Geosynchronous Orbit	4
MEO	Medium Earth Orbit	4
STK	Systems Toolkit	7
LQR	Linear Quadratic Regulator	8
GPOPS-II	A General-Purpose MATLAB Software for Solving Multiple-Phase Optimal Control Problems	11
NLP	Nonlinear Program	11
LVLH	Local Vertical Local Horizontal	12
COEs	Classical Orbital Elements	15
ECI	Earth Centered Inertial	16
SNOPT	Sparse Nonlinear Optimizer	23
IPOPT	Interior Point Optimizer	23

LIMITED-DUTY-CYCLE SATELLITE FORMATION CONTROL VIA DIFFERENTIAL DRAG

I. Introduction

1.1 Cube Satellites

Cube satellites or CubeSats are a relatively new and developing technology used for new and more complex missions every year. These nano-satellites are built from standardized 10 cm cube unit blocks of varying numbers and configurations. This flexibility in design and small form factor offer an alternative design option for satellite programs.

CubeSats have substantially cheaper material and development cost compared to other satellite systems due to their reduced size and complexity. Because of this, CubeSat programs are able to design, build, and launch missions faster than any other satellite program, making them ideal for problems that require responsive development. They are also an ideal solution for missions with few payload or mission objectives, such as single component technology development, a dedicated research experiment, or a small scale university project. This enables a slew of small scale projects with limited budgets that would otherwise have to be integrated into another larger mission with lower execution priority.

However, the small form also introduces inherent limitations. By definition, because CubeSats are less complex they are also less capable. The small form factor and low mass budgets preclude more complicated bus architectures and limit payload options and mission capabilities. More common bus support elements typically

on larger satellites, such as redundant systems, Sun-tracking solar panels, multiple communication options, etc., simply are not viable options for CubeSats. While one or two complex systems could be accommodated, most components and payloads are also reduced in scale and capabilities to fit the CubeSat form factor, reducing their potential capabilities. This applies to everything from optical payloads, whose geometry and therefore capabilities are limited, to radar and communications payloads, which are limited by reduced power output and lower gain values. These factors make CubeSats a poor option for missions with high capability requirements and result in inherently lower mission lifetime.

Despite these limitations CubeSat missions continue to become more complex and ambitious. As the complexity and difficulty increase, CubeSat programs shift from single satellite mission architectures to using satellite formations. This transition enables the increased complexity requirement despite the limited nature of an individual CubeSat. Using multiple satellites enables solutions such as synthetic aperture to counter hardware limitations of an individual satellite and enable more advanced optical missions. Formations and constellations of communications satellites could provide coverage where single satellites could not. Proximity operations with CubeSats enables additional payload test options.

However, formations introduce the additional problem of developing formation keeping and maneuver solutions. Formation keeping is necessary to counter the natural orbital perturbations which would otherwise cause the relative positions to drift apart and render the formation useless. While the formation can be established modeling some of these perturbations, drag and solar wind are inherently unpredictable and would introduce errors over time. The ability to control relative position is also necessary to accommodate certain mission requirements. Payloads focused on proximity operations, for example, would require the ability to change the formation's

shape, size. Synthetic aperture missions would also require resizing to change the size of said aperture.

1.2 Differential Drag

One useful formation keeping method is differential drag control. Differential drag works by controlling the drag profile of all satellites in a formation to control the relative drag acceleration. This allows one or more satellites to accelerate relative to the formation in the same manner as firing a very low thrust rocket [1], but achieving this effect only through the use of the natural drag perturbation. This does, however, mean that the control method is dependent on the drag perturbation, which brings with it some serious limitations. Because this control models the difference in drag acceleration as the control thrust, the maximum possible relative acceleration is fundamentally limited to a fraction of the drag perturbation. Since the drag force is small enough to be considered, in most cases, a perturbation rather than a dynamic property, its use as a control produces likewise small accelerations. This requires drag control approaches to consider continuous control much like electric thrusters, rather than impulsive burns. Besides magnitude, the direction of the control is constrained. The drag force, and thus its relative acceleration, acts in the opposite direction of the velocity vector. Drag control can only be applied opposite the velocity vector and therefore can only affect in-plane motion, making it useless for inclination changes. Furthermore, drag is not a consistent perturbation, and its variations affect the functionality of differential drag control. Altitude, solar cycle, and atmospheric properties all affect atmospheric density and thus the drag perturbation. Atmospheric density can vary by several orders of magnitude between solar minimums and maximums for the 11 year solar cycle, and it varies on a daily basis based on temperature and other atmospheric properties. These conditions can affect control authority and make the

effectiveness of drag control methods depend heavily on mission launch time. This all assumes, of course, that the mission launches into a region where drag control is viable. Atmospheric density varies heavily on altitude and is only dense enough in Low Earth Orbit (LEO) to provide meaningful acceleration. Most models do not account for the drag perturbation at Geosynchronous Orbit (GEO) or even at Medium Earth Orbit (MEO) because it is so small; and the effectiveness of drag control drops off much proportionally. This work considers altitudes from 400 km to 800 km to avoid the problem of the drag perturbation becoming so low that other perturbations make its use meaningless [2]. Lastly, this control will affect mission life; intentionally increasing the drag profile will increase the rate of orbital degradation as more energy is removed from the system.

Despite all these limitations, differential drag control remains a valuable control method to CubeSat and small sat formation flying missions because it can preclude the need for complex, massive, and expensive propulsion systems. Many CubeSat missions use ride share launches to get to orbit. Ride share imparts limitations on the secondary payloads to ensure the safety of the primary mission. One common restriction is preventing any secondary payload from having a propulsion system. If not outright prohibited, propulsion can also be limited from chemical thrusters to less effective methods, such as cold gas thrusters. Since it does not require fuel, differential drag remains a viable formation control method when propulsion cannot be implemented. Furthermore, since CubeSats have fundamentally limited capacity, mass and volume are at a premium. This also makes differential drag control valuable as it can usually be implemented without any additional hardware, let alone the complexity of a propulsion system. Since differential drag control only requires modifying the drag profile, cross-sectional area normal to the flight direction, any asymmetrical satellite with sufficient attitude control can successfully implement drag control. This is es-

pecially true for satellites with deployable solar panels. The ability to control panel orientation can provide more drag control options. Because of these benefits, differential drag has been studied extensively–from Leonard et al's work in 1989 using a simplified model to several more modern works studying feasibility, several control methods, and various optimal and robust control designs–all while considering several dynamical models from linearized leader-follower formations to nonlinear non-circular formations [3, 4, 5, 6, 7, 8].

1.3 Problem Statement

While extensive research has been completed in this area, there are still certain aspects that have not been fully developed. One of these areas is the effect of limiting control to specific regions of the orbit. This limited-duty-cycle control requires the system to be discontinuous, having no control authority for the specified region of the orbit. This is useful for missions where required satellite operations interfere with the application of differential drag control. For example, a mission with high power requirements would require a Sun pointing mode on the day side of an orbit. If the same mission had fixed solar panels it was using for differential drag, control then the system would only be controllable during eclipse. If attitude is used to control differential drag, then this limitation extends to any subsystem or payload operation that requires a specific attitude. This study will apply limited differential-drag maneuvers on a model leader-follower CubeSat formation and analyze their effect on optimal control. When in the uncontrolled region it is assumed both satellites are in a minimum drag control profile (minimum cross-sectional area). This formation uses a circular orbit but must contend with maneuver sizes and separation distances up to 2000 km. Thus nonlinear relative dynamics are used to model the formation, applying J_2 and drag perturbations to both satellites. While the desired formation

is circular, the dynamics require no such assumption, nor is out-of-plane motion assumed to be decoupled. Other perturbations are assumed to be small enough to ignore. Metrics used in this study will include time required for the maneuver to complete and control authority of the system. This is both an important research for CubeSats missions to consider, as it applies new dynamics to modeled systems and considers applicability of drag control to implementation feasibility, and an area previous literature has yet to consider.

To develop this problem, a mission planning tool will be developed capable of generating all maneuver and formation keeping results presented in this document. The tool will function as an open-loop optimal controller whose output could be applied to a real system with the same specifications as what is stated in the document. The capabilities of this tool will be evaluated during the development of the stated research questions:

1: How will limited-Duty-Cycle control affect the optimal differential-drag control solution, and how will this affect mission planning?

2: Is a multi-phase pseudospectral solver suitable for solving maneuvers and formations with separation up to 2000 km, and what complications arise?

In summary, CubeSats are a new technology whose complexity increases, driving them towards formations. Due to the limitations inherent to CubeSats, differential-drag is a useful control method. Drag control has been studied extensively, but two less studied areas are limited control and incredibly large formations and maneuvers. These areas are studied in this thesis through the development of a mission planing tool. This document presents a brief overview of explored works, develops the problem dynamics and planning tool solution algorithm, provides sample outputs and analyses these results, and explains the impact of these results.

II. Literature Review

2.1 Development of Differential Drag Control

The study of differential drag as a means of formation control started with Leonard's work in 1989 with a simple feasibility study of drag applied to the formation flight problem [3]. They used a simplified bang-bang controller applied to a limited formation keeping problem that used feedback and a linearized unperturbed model. Their study proved the feasibility for differential drag control to maintain a formation and an orbit's shape. While their results could hardly be effectively applied to actual hardware it was instrumental for kicking off the development of the differential drag control problem.

Since then there has been an extensive amount of work done on the differential drag control problem. This includes works that improve the feasibility analysis originally presented. These works include *Characterizing and Controlling the Effects of Differential Drag on Satellite Formations*; an AFIT thesis by J. Wedekind that provides a higher fidelity feasibility analysis combined with a formation control law. Their work provides similar but more developed results to the original Leonard paper, focusing on formation control but providing a higher fidelity model [9]. Their feasibility analysis has been further developed in works like *Satellite Formation Control using Differential Drag*, a conference paper by S. Omar and Dr. Wersinger. Their work provides a feasibility analysis using numerical simulation in AGI's Systems Toolkit (STK) that studies both separation control and maneuvers. These simulations included J_2 effects, a non-standard atmosphere, non-standard drag coefficients, solar pressure, and third body gravitational forces" and focused on the CubeSat mission type [**?**]. These results, while not optimal, provide a very accurate representation of on orbit behavior.

Differential drag control is developed by other works such as *Satellite Formation Control Using Atmospheric Drag*, AFIT thesis by B. Hajovsky [8]. Their work poses the optimal control problem using a Linear Quadratic Regulator (LQR) to provide feedback control of the minimum energy loss problem. While their work only used the linear relative dynamics, no additional perturbations, and a simple exponential atmospheric model it did provide a starting point for the development of the solution presented in the work herein. For the minimum time solution one must look to other works, such as *Rendezvous Using Differential Drag and Feedback Control*, by M. Harris et al. Their work is another example of the optimal control problem, but uses the full nonlinear dynamics and includes J_2 perturbations [6]. Their work proves that the minimum time solution in bang-bang control for differential drag and, similarly to Hajovsky, develops a feedback control law. While not directly applicable to the method developed here it should be mentioned that robust control has also been extensively developed. An example of this is *Satellite Formation Control Using Continuous Adaptive Sliding Mode Controller*, a paper by H. Cho. Their work uses a very high fidelity solution that solves the differential drag problem with robust control of the nonlinear dynamics with uncertainties in perturbations and spacecraft properties [4]. Furthermore, it develops their robust controller while maintaining reasonable maneuver times and control histories that resemble the optimal profile and provide a stable solution. And finally, optimal control has been developed with a number of numerical techniques such as pseudospectral methods. *Comparison Between Analytical and Optimal Control Techniques in the Differential Drag Based Rendezvous*, by L. Dell'Elce and G. Kerschen, outlines a pseudospectral method for optimal differential drag control implemented on a model system [7]. Their work also uses model predictive control techniques to account for an estimation of the drag control in the effort to develop a method of practical application of differential drag. The research

herein represents a step towards the practical use of planned differential drag control maneuvers.

2.2 Eclipse Limited Control

With the extensive development of the differential drag control problem there remains some areas that have not been fully developed. One of those is applying differential drag to practical mission architectures. This is not, however, to say no work has been done in the area. Rendezvous Maneuvers of Multiple Spacecraft Using *Differential Drag Under J_2 Perturbation*, by R. Bevilacqua, studies the required control to maintain a formation of several satellites tracking a virtual chief [5]. However, their paper does not solve the optimal control problem and uses a linearized set of dynamics assuming a circular orbit. While it successfully controls the formation, additional work could be done to improve the optimization of said control. While this is a potential area of development, this thesis intends to study a different area; limited, or non-continuous, control.

The limited control case is an important area of study for differential drag to be implemented on a practical mission architecture. Maneuvers using differential drag can take days, weeks, or even months to complete depending on the system and environmental conditions and the required maneuver. Many mission architectures, however, would not be able to support a dedicated maneuver of that duration without interruptions for anything from ground contact, payload operation, or sun pointing for charging requirements, as is common for CubeSats. Yet despite this, there are no known studies of limiting differential drag control. However, a comparison can be made to low thrust maneuvers done with an eclipse region. *Optimal Low Thrust Orbit Transfers with Eclipsing*, by J. Betts studies the effects of a low thrust electric thruster performing an orbit resizing maneuver while avoiding thrusting during

eclipse [1]. While this is not directly the problem we are interested in the differential-drag problem can be posed similarly by maneuvering during eclipse and performing formation resizing. Their study solves for an optimal steering history attitude using pseudospectral methods by treating the problem as having multiple phases, of drift and control, that make up the final solution. Moreover, it shows that a similar method should be able to provide results for the differential drag case.

2.3 Summary

The foundation for much of the work in this thesis has already been completed through the numerous studies of differential drag. Precedent has already been set for the dynamics required for this work, though never tested to the ranges this study is interested in. The optimal control method has been shown to solve using pseudospectral methods by several works, and as such was selected as the solver of choice for this problem. This literature review also confirms that the application of limited-duty-cycle differential drag control, and long maneuvers using pseudospectral methods has not been explicitly studied before.

III. Methodology

This optimal control method is solved using GPOPS-II and computed using a set of full nonlinear relative dynamics of a deputy satellite to the chief satellite. A General-Purpose MATLAB Software for Solving Multiple-Phase Optimal Control Problems (GPOPS-II)[1] is a pseudospectral optimal control solver designed around solving Nonlinear Program (NLP) using Gaussian quadrature hp methods [10]. The dynamics include the J2 and Drag perturbations. The full nonlinear dynamics are required due to the interest in maneuvers and formation keeping at linear separation distances up to 2000 km. Drag perturbation must be included due to the nature of the problem and the J2 perturbation is required for solution accuracy due to the duration of the maneuvers and the interest in formation keeping as the most prevalent LEO perturbation. Atmospheric density is calculated using the Jacchia-Roberts atmospheric model.

3.1 Problem Dynamics

3.1.1 Model Properties.

Satellite properties for this study are based around a model leader-follower mission. For the purposes of this study a circular orbit with an inclination of 28.57° is used to match a Cape Canaveral launch. Both the initial argument of latitude (θ) and right ascension of the ascending node (Ω) were set to zero. Simulation start time is set to be 0101.01 UT on 01-02-2014, the year at the last solar maximum. This start time was selected to aid computation, since computer run time varies with maneuver time, and solar maximum provides the system with more control authority, thereby reducing

[1] MATLAB®is a registered trademark of The MathWorks, Inc.

maneuver time. The date 01-01-2014 is not used since the atmospheric model requires at least one day of historic data.

Figure 1. Pumpkin Inc Supernova Bus

This mission models two Supernova buses designed by Pumpkin Inc. These identical satellites consist of two 6U CubeSats (as shown in Fig. 1) with deployable solar panels along the side. When deployed, the solar panels' front face has a surface area of 0.2 m^2 and represents the maximum surface area. The minimum surface area is the 0.03 m^2 of one 3U face. Both satellites are assumed to be the same mass, 9.567 kg. A coefficient of drag C_d of 2.2 was used. The eclipse region where the control is allowed is defined by the umbra region at the given orbital altitude. When not performing a differential-drag control maneuver, the satellites are assumed to be in a minimum-control profile. This is so that each satellite has the same drag profile during limited regions of the orbit.

3.1.2 Nonlinear Relative Dynamics.

This problem uses the full nonlinear set of relative dynamics posed in a Local Vertical Local Horizontal (LVLH) Cartesian frame [11]. For this LVLH frame $\hat{x}$ points in the same direction as the chief's position vector $\bar{r}$, $\hat{z}$ points along the angular momentum vector $\bar{h}$, and $\hat{y}$ is $\hat{z} \times \hat{x}$ to point along the velocity vector. These nonlinear equations are used to ensure solution accuracy even for very large maneuvers or formations. This equation set looks only at J_2 and drag perturbations, as it considers

only a 500 km Low Earth Orbit (LEO), where differential-drag control is viable and other perturbations can often be ignored [12].

Dynamics development begins with the set presented in *Spacecraft Formation Flying*, and confirmed with other sources, as follows [11, 13, 14]:

$$\ddot{x} = 2\dot{y}\omega_z - x(n_j^2 - \omega_z^2) + y\dot{\omega}_z - z\omega_x\omega_z - (\zeta_j - \zeta)\sin i \sin\theta - r(n_j^2 - n^2) + A_x \quad (3.1)$$

$$\ddot{y} = -2\dot{x}\omega_z + 2\dot{z}\omega_x - x\dot{\omega}_z - y(n_j^2 - \omega_z^2 - \omega_x^2) + z\dot{\omega}_x - (\zeta_j - \zeta)\sin i \cos\theta + A_y \quad (3.2)$$

$$\ddot{z} = -2\dot{y}\omega_x - x\omega_x\omega_z - y\dot{\omega}_x - z(n_j^2 - \omega_x^2) - (\zeta_j - \zeta)\cos i + A_z \quad (3.3)$$

$$\zeta = \frac{2K_{J2}\sin i \sin\theta}{r^4} \quad (3.4)$$

$$n^2 = \frac{\mu}{r_j^3} + \frac{K_{J2}}{r_j^5} - \frac{5K_{J2}\sin i^2 \sin\theta^2}{r_j^5} \quad (3.5)$$

$$K_{J2} = \frac{3J_2\mu R_e^2}{2} \quad (3.6)$$

where x, y, and z are the deputy's position relative to the chief; ω_z and ω_x are the rotation of the LVLH frame; A_x, A_y, and A_z are the relative acceleration on the deputy in the appropriate direction; ζ and n are values defined for mathematical convenience from chief gravitational acceleration; and lastly ζ_j and n_j^2 are similar values containing deputy J_2 acceleration terms defined for convenience [11].

However, in this dynamics set the development of n^2 and ζ relies on assumptions that chief orbital elements remain unperturbed by all but J_2 effects. Since drag will be applied to the chief's orbital parameters these values had to be reverted back to the expressions they represented when that approximation was applied. The following equations are found once Equations 3.4 and 3.5 have been substituted in

and simplified:

$$\ddot{x} = 2\dot{y}\omega_z - (x+r)(n_j^2 - \omega_z^2) + y\dot{\omega}_z - z\omega_x\omega_z - \zeta_j \sin i \sin\theta - \ddot{r} + A_{Dx} \tag{3.7}$$

$$\ddot{y} = -2(\dot{x}+\dot{r})\omega_z + 2\dot{z}\omega_x - (x+r)\dot{\omega}_z - y(n_j^2 - \omega_z^2 - \omega_x^2) + z\dot{\omega}_x - \zeta_j \sin i \cos\theta + A_{Dy} \tag{3.8}$$

$$\ddot{z} = -2\dot{y}\omega_x - (x+r)\omega_x\omega_z - y\dot{\omega}_x - z(n_j^2 - \omega_x^2) - \zeta_j \cos i + A_{Dz} \tag{3.9}$$

where A_{Dx}, A_{Dy}, and A_{Dz} are accelerations on the deputy only in the appropriate directions.

These nonlinear dynamics are based on solving the solving the Lagrangian equations of relative motion, where the deputy's kinetic and potential energy are posed in terms of its relative coordinates and the chief satellite's orbital parameters [11, 14, 13]. Because of the effects of the J_2 perturbation, the cross-track motion is no longer decoupled and must be propagated and accounted for in all cases. The deputy's potential energy is incorporated by the ζ_j and n_j terms, defined as follows:

$$\zeta_j = \frac{2K_{J2}r_{jZ}}{r_j^5} \tag{3.10}$$

$$n_j^2 = \frac{\mu}{r_j^3} + \frac{K_{J2}}{r_j^5} - \frac{5K_{J2}r_{jZ}^2}{r_j^7} \tag{3.11}$$

$$K_{J2} = \frac{3J_2\mu R_e^2}{2} \tag{3.12}$$

$$r_j = \sqrt{(r+x)^2 + y^2 + z^2} \tag{3.13}$$

$$r_{jZ} = (r+x)\sin i \sin\theta + y \sin i \cos\theta + z\cos i \tag{3.14}$$

where $R_e = 6378.137$ km is the Earth's radius, $J_2 = 1.08262668355 * 10^-3$, and $\mu = 398600.4418$ km^3/s^2 is Earth's gravitational parameter. The n_j^2, r_{jZ}, and ζ_j terms take into account the effect the J_2 gravitational perturbation has on the deputy

satellite's potential energy based on r_j (the deputy's orbital radius) and express how this affects its motion in the LVLH frame. These equations are based on LVLH and chief orbital element parameters.

The LVLH frame's angular velocity components about its own axes are ω_x, ω_y, and ω_z, respectively. ω_z is fundamentally defined by the angular rotation of the chief satellite, ω_x does not exist in the unperturbed dynamics but must be accounted for due to the J_2 perturbation, and ω_y does not exist as it would require the epicenter of orbital rotation to move [11].

$$\omega_z = \frac{h}{r^2} \tag{3.15}$$

$$\dot{\omega}_z = -\frac{2h\dot{r}}{r^3} + \frac{\dot{h}}{r^2} \tag{3.16}$$

$$\omega_x = \frac{rA_{Cz}m}{h} - \frac{K_{J2}\sin 2i \sin\theta}{hr^3} \tag{3.17}$$

$$\begin{aligned}\dot{\omega}_x = \frac{\dot{r}A_{Cz}m}{h} - \frac{\dot{h}rA_{Cz}m}{h^2} + \frac{r\dot{A}_{Cz}m}{h} + \frac{\dot{h}K_{J2}\sin 2i\sin\theta}{h^2r^3} \\ + \frac{3\dot{r}K_{J2}\sin 2i\sin\theta}{hr^4} - \frac{2\dot{i}K_{J2}\cos 2i\sin\theta}{hr^3} - \frac{\dot{\theta}K_{J2}\sin 2i\cos\theta}{hr^3}\end{aligned} \tag{3.18}$$

Chief orbital parameters consist of magnitude of the chief's radius r, magnitude of angular momentum h, right ascension of the ascending node Ω, inclination i, and argument of latitude θ. These orbital parameters were selected over the standard Classical Orbital Elements (COEs) to reduce computational complexity and avoid singularities when eccentricity is zero. These orbital elements are propagated based on the drag and J_2 perturbations as follows [11, 14, 13, 15].

$$\ddot{r} = -\frac{\mu}{r^2} + \frac{h^2}{r^3} - \frac{K_{J2}(1 - 3\sin^2 i \sin^2\theta)}{r^4} + A_{Cx} \tag{3.19}$$

$$\dot{h} = \frac{-K_{J2} \sin^2 i \sin 2\theta}{r^3} + rA_{Cy} \tag{3.20}$$

$$\dot{\Omega} = \frac{-2K_{J2} \cos i \sin^2 \theta}{hr^3} + \frac{r \sin \theta A_{Cz}}{h \sin i} \tag{3.21}$$

$$\dot{i} = \frac{-K_{J2} \sin 2i \sin 2\theta}{2hr^3} + \frac{r \cos \theta A_{Cz}}{h} \tag{3.22}$$

$$\dot{\theta} = \frac{h}{r^2} + \omega_x \cot i \sin \theta \tag{3.23}$$

$A_{Cx,y,z}$ refers to the combined acceleration due to drag and any lift forces on the chief satellite in any one of the LVLH frame directions [13]. While A_{Cz} would be zero if drag were the only perturbation, J_2 causes slight z direction oscillations in velocity captured in ω_x. This velocity, along with atmospheric rotation velocity, causes the chief to experience drag in the z direction that cannot be ignored.

Because of the Lagrangian formulation of the deputy's dynamics in the LVLH frame, the deputy's motion is captured both by its position and velocity and the chief's position and velocity [14]. Because of this, $A_{Dx,Dy,Dz}$ consists of only the acceleration due to lift and drag on the deputy in the LVLH frame rather than the relative acceleration, $A_{x,y,z}$, of both the chief and the deputy as many other works consider it [14, 11]. Instead, the effects of the relative acceleration due to chief drag are calculated by updating the orbital parameters from the drag perturbation. As these parameters change, the associated relative acceleration we would expect to see in the LVLH frame can be observed as shown in Figure 2.

This allows the chief's parameters to be updated more accurately and is required for formation keeping and large maneuvers, but makes the chief's orbital parameters dynamical states that must be propagated by the optimal control solver as they now vary based on control and directly affect the solution.

Drag acceleration is calculated from the Earth Centered Inertial (ECI) frame position and velocity vectors of the deputy and chief satellite. These vectors are used

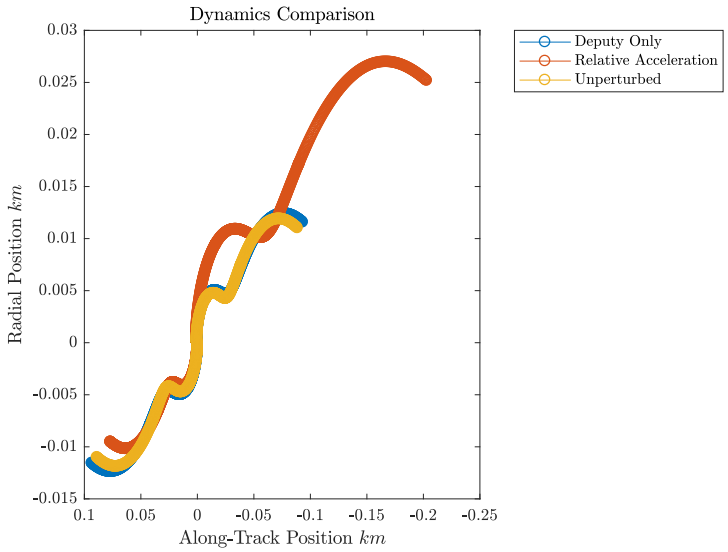

Figure 2. Comparison between the full dynamics with and without relative acceleration, and the unperturbed nonlinear dynamics with relative acceleration

to calculate the relative velocity of both satellites to the atmosphere. This calculation assumes a co-rotating atmosphere.

$$C_{I/lvlh} = \begin{bmatrix} -\mathrm{s}(\Omega)\mathrm{c}(i)\mathrm{s}(\theta) + \mathrm{c}(\theta)\mathrm{c}(\Omega) & -\mathrm{s}(\Omega)\mathrm{c}(i)\mathrm{c}(\theta) - \mathrm{s}(\theta)\mathrm{c}(\Omega) & \mathrm{s}(\Omega)\mathrm{s}(i) \\ \mathrm{c}(\Omega)\mathrm{c}(i)\mathrm{s}(\theta) + \mathrm{c}(\theta)\mathrm{s}(\Omega) & \mathrm{c}(\Omega)\mathrm{c}(i)\mathrm{c}(\theta) - \mathrm{s}(\theta)\mathrm{s}(\Omega) & -\mathrm{c}(\Omega)\mathrm{s}(i) \\ \mathrm{s}(i)\mathrm{s}(\theta) & \mathrm{s}(i)\mathrm{c}(\theta) & \mathrm{c}(i) \end{bmatrix} \tag{3.24}$$

$$\bar{R}_{chief} = C_{I/lvlh} \begin{bmatrix} r \\ 0 \\ 0 \end{bmatrix} \tag{3.25}$$

$$\bar{V}_{chief} = C_{I/lvlh} \begin{bmatrix} \dot{r} \\ \omega_z r \\ 0 \end{bmatrix} \tag{3.26}$$

$$\bar{R}_{deputy} = C_{I/lvlh} \begin{bmatrix} r+x \\ y \\ z \end{bmatrix} \tag{3.27}$$

$$\bar{V}_{deputy} = C_{I/lvlh} \begin{bmatrix} \dot{r} + \dot{x} - \omega_z y \\ \dot{y} + \omega_z(r+x) - \omega_x z \\ \dot{z} + \omega_x y \end{bmatrix} \tag{3.28}$$

$$\bar{V}_{rel} = \bar{V} - \begin{bmatrix} 0 \\ 0 \\ W_e \end{bmatrix} \times \bar{R} \tag{3.29}$$

$$\bar{\alpha}_{drag} = \frac{C_d \rho S}{2m} |\bar{V}_{rel}| \bar{V}_{rel} \tag{3.30}$$

where $C_{I/lvlh}$ is the attitude rotation matrix between the LVLH and ECI frames, s and c are abbreviations of sin and cos, $\bar{R}$ and $\bar{V}$ are the inertial position and velocity vectors of the satellite, W_e is the Earth's rotation, $\bar{\alpha}_{drag}$ is the drag acceleration vector for the respective satellite, m is the mass of the satellite, C_d is the coefficient of drag, ρ is the atmospheric density, and S is the surface are perpendicular to the relative velocity $\bar{V}_{rel}$ [16, 17].

3.1.3 Atmospheric Model.

The atmospheric density is calculated using a supplemented Jacchia-Roberts model [15, 18]. The Jacchia atmospheric model functions on calculating and integrating the diffusion differential of the atmosphere, based primarily on temperature. It calculates the density at any given altitude by integrating from a known altitude. This method requires calculating the exosphere temperature, inflection point temperature, and using a known temperature and density at a base altitude. Finally, once cal-

culated this solution is then modified based on four different atmospheric regions. The primary addition by Roberts is using partial fraction expansion to produce an analytical solution to the diffuse integration. This solution produced identical results at low altitudes and was an approximation within 5% to 1% of the original solution based on the corrective value l [19]. The method used here uses this Jacchia-Roberts model but uses density at 125 km as a known atmospheric value rather than the 90 km standard used by the original model. The atmospheric model used here calculates this value based on the polynomial curve fit of the Goddard Trajectory Determination System. Similarly a more accurate l calculation is used based on a least-squares curve fit developed for the Goddard system. Both methods developed by Draper Laboratory ensure a deviation of 6.7% or less from the Jacchia model [15, 18]. These changes allow the calculation to be simplified from four regions to only the two high altitude regions, improving computational efficiency while maintaining required accuracy.

$$\rho = \sum_{i=1}^{5} \rho_{125_i} \Big(\frac{T_x}{T}\Big)^{1+a_i+\gamma_i} \Big(\frac{T_{corr} - T}{T_{corr} - T_x}\Big)^{\gamma_i} \tag{3.31}$$

$$\gamma_i = \frac{M_i g R_{pole}^2}{R l T_{corr}} \Big(\frac{T_{corr} - T_x}{T_x - T_0}\Big)\Big(\frac{35}{64810766}\Big) \tag{3.32}$$

$$\rho_{125_i} = \frac{M_i}{A_{vg}} 10^{\sum_{j=0}^{6} \delta_{ij} T_{corr}^j} * 1000 \tag{3.33}$$

$$l = \sum_{j=0}^{4} l_j T_{corr}^j \tag{3.34}$$

Here density kg/m^3 is calculated as the summation of the density contributed by the major constituents of the upper atmosphere: N_2, Ar, He, O_2, O, H. Where ρ_{125_i} is the density at 125 km of the particular constituent, $A_{vg} = 6.02257e23$ is Avogadro's number, T_x is the inflection point temperature (K), T is the temperature at the desired altitude (K), a_i is the thermal diffusion coefficient, T_{corr} is the corrected exospheric

temperature (K), M_i is the molecular mass (g/mol), $g = 9.80665$ the gravitational constant at sea level (m/s^2), $R_{pole} = 6356.766$ is the Earth's polar radius (km), $R = 8.31432$ is the ideal gas constant, $T_0 = 183$ is the temperature at 125 km (K). The data needed for the index summations can be found in the Tables 1 and 2 [15, 18].

Table 1. Constituent Data

Index (i)	Name	M_i	a_i
1	N_2	28.0134	0
2	Ar	39.948	0
3	He	4.0026	-0.38
4	O_2	31.9988	0
5	O	15.9994	0
6	H	1.00797	0

Table 2. l Approximation Data

0	1	2	3	4
0.1031445e5	0.2341230e1	0.1579202e-2	-0.1252487e-5	0.2462708e-9

Table 3. Coefficients for Approximating ρ_{125}

Index (j)	0	1	2	3	4	5	6
N_2	0.109316e2	0.118678e-2	-0.167734e-5	0.142023e-8	-0.713979e-12	0.196972e-15	-0.229618e-19
Ar	0.804941e1	0.238282e-2	-0.339137e-5	0.290971e-8	-0.148170e-11	0.412760e-15	-0.483746e-19
He	0.764689e1	-0.438349e-3	0.469432e-6	-0.289489e-9	0.945199e-13	-0.127084e-16	0
O_2	0.992424e1	0.160031e-2	-0.227476e-5	0.193845e-8	-0.978218e-12	0.269845e-15	-0.313181e-19
H	0.109708e2	0.611874e-4	-0.116500e-6	0.923935e-10	-0.349074e-13	0.511630e-17	0

Because helium experiences seasonal variations in density the third index term must be corrected before it can be summed.

$$\rho_{He} = \rho_{He} 10^{\Delta \log_{10} \rho_{He}} \tag{3.35}$$

$$\Delta \log_{10} \rho_{He} = 0.65 \left| \frac{\delta_s}{\epsilon} \right| \left(\sin^3 \left(\frac{\pi}{4} - \frac{\phi_{gd} \delta_s}{2|\delta_s|} \right) - 0.35355 \right) \tag{3.36}$$

where $\epsilon = 23.4(\pi/180)$ is the obliquity of the ecliptic, ϕ_{gd} is the geodetic latitude of the satellite, and δ_s is the declination of the Sun in radians.

The second region occurs at 500 km and above. In this region the density is low enough that hydrogen starts to have an effect and needs to be accounted for. This is done by calculating the density of H and adding it to the density sum.

$$\rho_H = \rho_{H_{500}} \Big(\frac{T_x}{T}\Big)^{1+a_H+\gamma_H} \Big(\frac{T_{corr}-T}{T_{corr}-T_x}\Big)^{\gamma_H} \tag{3.37}$$

To use this equation the density of hydrogen at 500km is approximated using the following equation.

$$\rho_{H_{500}} = \frac{M_H}{A_{vg}} 10^{73.13-(39.4-5.5\log_{10}(T_{500}))(\log_{10}(T500))} \tag{3.38}$$

where T_{500} is the temperature at 500 km.

The exospheric temperature T_c is the nighttime global exospheric temperature calculated from the 10.7 cm solar flux index and is then corrected based on local time and the geomagnetic, k_p, index. This calculation takes into account both the satellite's and the Sun's positions.

$$T_c = 379 + 3.24\bar{F}_{10.7} + 1.3(F_{10.7} - \bar{F}_{10.7}) \tag{3.39}$$

$$T_{unc} = T_c\Big(1 + 0.3\big(\sin^{2.2}(\theta) + \big(\cos^{2.2}(\eta) - \sin^{2.2}(\theta)\big)\cos^3(\tau/2)\big)\Big) \tag{3.40}$$

$$\eta = \frac{|\phi_{gd} - \delta_s|}{2} \tag{3.41}$$

$$\theta = \frac{|\phi_{gd} + \delta_s|}{2} \tag{3.42}$$

$$\tau = LAH_s - 0.64577 + 0.10472\sin(LAH_s + 0.75049) \tag{3.43}$$

$$LAH_s = \frac{r_x r_J - r_y r_I}{|r_x r_J - r_y r_I|}\cos^{-1}\left(\frac{r_x r_I + r_y r_J}{\sqrt{r_x^2 + r_y^2}\sqrt{r_I^2 + r_J^2}}\right) \tag{3.44}$$

$$\phi_{gd} = \tan^{-1}\left(\frac{1}{(1-f)^2}\frac{r_K}{\sqrt{r_I^2 + r_J^2}}\right) \tag{3.45}$$

$$\delta_s = \tan^{-1}\left(\frac{r_z}{\sqrt{r_x^2 + r_y^2}}\right) \tag{3.46}$$

$$T_{corr} = 28k_p + 0.03\exp k_p \tag{3.47}$$

If below 200 km

$$T_{corr} = 14k_p + 0.02\exp k_p \tag{3.48}$$

where $r_{x,y,z}$ are the ECI frame elements of the Sun's unit vector, $r_{I,J,K}$ are the same for the satellite's position vector, $F_{10.7}$ is the solar flux index, and $\bar{F}_{10.7}$ is an 81 day moving average of the solar flux index. For this thesis solar flux and geomagnetic data was pulled from the Space Weather Prediction Center website's data archive by the National Oceanic and Atmospheric Administration.

The Sun's unit vector is calculated as follows [20]:

$$T_{UT1} = \frac{Jd_{UT1} - 2451545}{36525} \tag{3.49}$$

$$T_{TDB} = T_{UT1} \tag{3.50}$$

$$\lambda_{M_S} = 280.4606184 + 36000.77005361T_{UT1} \tag{3.51}$$

$$M_{Sun} = 357.5277233 + 35999.05034T_{TDB} \tag{3.52}$$

$$\lambda_{ecliptic} = \lambda_{M_S} + 1.914666471\sin(M_{Sun}) + 0.918994643\sin(2M_{Sun}) \tag{3.53}$$

$$\epsilon_{Sun} = 23.439291 - 0.0130042T_{TDB} \tag{3.54}$$

$$\hat{r}_{Sun} = \begin{bmatrix} \cos(\lambda_{ecliptic}) \\ \cos(\epsilon_{Sun})\sin(\lambda_{ecliptic}) \\ \sin(\epsilon_{Sun})\sin(\lambda_{ecliptic}) \end{bmatrix} \tag{3.55}$$

where Jd is the Julian date in universal time, λ_{M_S} is the mean longitude of the Sun, M_{Sun} is the mean anomaly of the Sun, $\lambda_{ecliptic}$ is the ecliptic longitude of the Sun, ϵ_{Sun} is the obliquity of the ecliptic; all of these values are in degrees.

The approximation of universal time, T_{UT1}, to Barycentric Dynamics time, T_{TDB}, is made because they differ on the scale of milliseconds, are being used on the scale of years, and conversion is computationally intensive.

3.2 Optimal Control

3.2.1 Solver Algorithm.

Problem optimization is solved using GPOPS-II with Sparse Nonlinear Optimizer (SNOPT) or Interior Point Optimizer (IPOPT). GPOPS-II is a pseudospectral optimal control solver designed around solving Gaussian quadrature hp-adaptive methods [10]. This method calculates approximations of the optimal control problem as constrained by the continuous function, containing the dynamics of the problem, and the endpoint (cost) function. These approximations are calculated by calling SNOPT or IPOPT, two NLP solvers. Theses problems different in how they calculate the problem and what they are optimized for, both of these solvers were used for various problem types to take advantage of their strengths. SNOPT is designed for a sparse problem, where the A matrix for state space dynamics is mostly zeros; this is not the case for this problem. It is, however, still useful as it is computationally more efficient than IPOPT when a good initial guess can be generated. IPOPT is instead more robust and is able to solve the optimal problem with practically no initial guess provided (just initial and final conditions as two time dependent data points). The output from these solvers is then compared against the required GPOPS-II tolerances and the NLP mesh is updated based on the output and solved again. This method,

when executed correctly, ensures solution accuracy on the mesh and, if tolerances are set correctly, accuracy of the problem.

The solver developed in this work is designed to provide accurate solutions for various inputs. As outlined in the flowchart shown in Figure, 3. this algorithm automatically sets up and executes the initial guess for the GPOPS-II problem, defines the solution bounds, sets initial and final conditions, defines mesh tolerances and initial mesh settings, and evaluates the GPOPS-II output. It uses phases to implement limited control and can apply an error checker that updates the initial conditions of the problem, X0. Both of these functions occur iteratively in their respective phase and error checker loops and will be discussed in the following sections. Its execution helps ensure that a solution can be provided by GPOPS-II without requiring additional analysis from the user wile accounting for several desired solution methods. Each block's purpose and function will be described in the following sections.

3.2.2 Algorithm Input.

The algorithm begins with defining user input. The user defines how many runs to perform, defines constant parameters, the initial and final conditions of each run, the desired solver methods, whether a maneuver or formation keeping is desired, and any additional conditions the solution should use. Defining the problem's constants requires defining satellite properties (mass, coefficient of drag, minimum and maximum area), atmospheric data used by the atmospheric model (geomagnetic and solar flux histories or predictions),and other constants for Earth's properties (μ, R_e, J_2, ω_e, and umbra and penumbra angles). Properties are only defined for one satellite as the solution assumes both chief and deputy satellites are identical. Initial conditions of the problem can be given in terms of initial separation distance, initial position and velocity of the deputy in the LVLH frame (both requiring chief orbital element

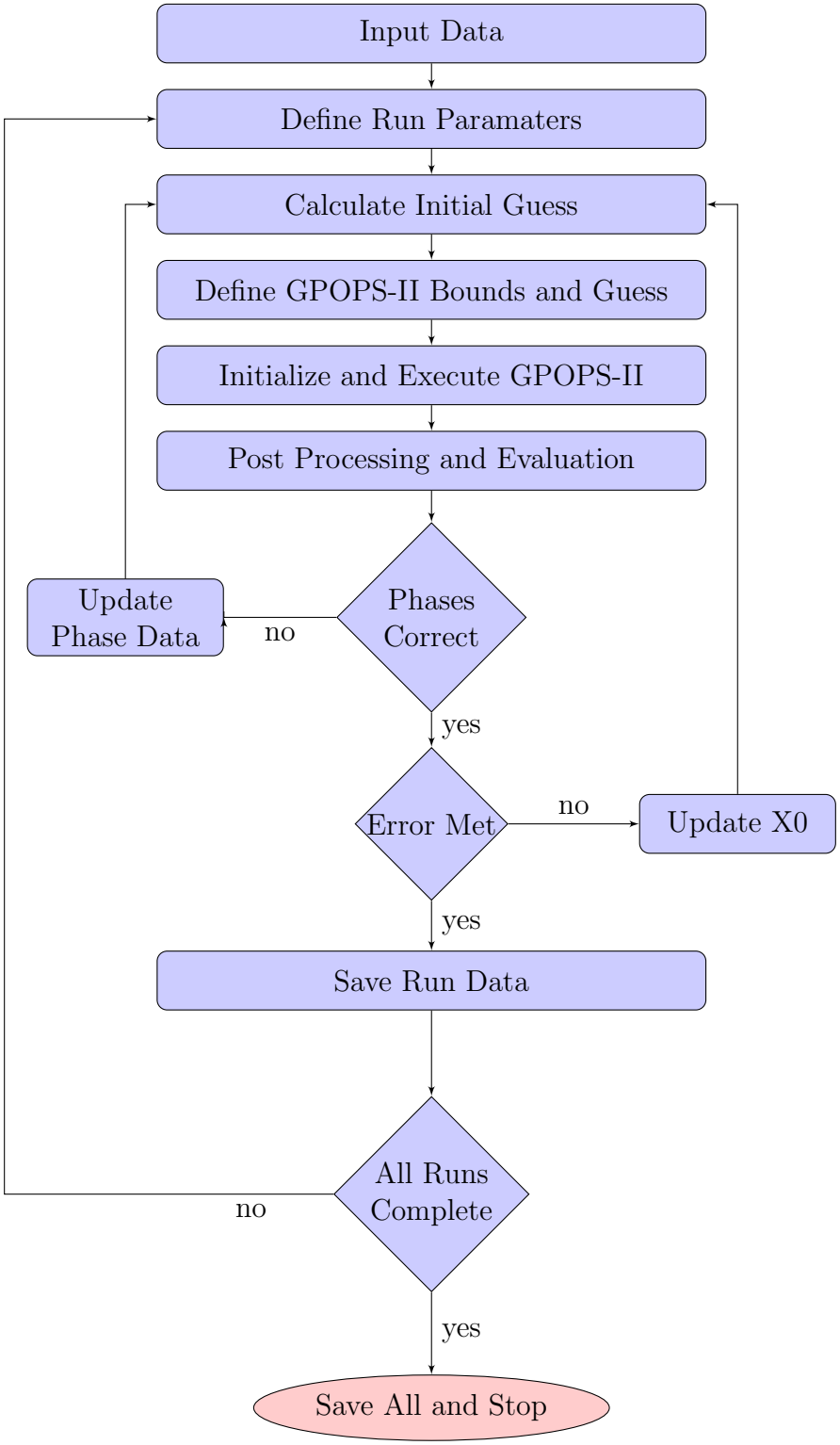

Figure 3. Solver Algorithm

definition), or as ECI position and velocity vectors of the chief and deputy; all three cases define initial time. For all cases final conditions are defined by a desired separation distance, like the first input option, which assumes a circular orbit for the desired formation without any in-plane drift. The sign on final separation distance indicates if the final position will be along (positive) or against (negative) the LVLH y-axis. Defining desired solver methods allow the user to trade solution accuracy for computational efficiency. This not only defines which NLP to use but at what level to calculate the dynamics of the problem. The dynamics can be simplified by assuming there is no J_2 perturbation, no J_2 perturbation and no drag perturbation on chief orbital parameters, or by assuming the chief orbital elements do not change and are circular. This selection can also accompany using a constant approximate atmospheric density rather than calculating the Jacchia-Roberts atmospheric model at every time step. If formation keeping is set a time parameter to define the duration of the formation keeping must be set, it will use the final conditions as the desired formation keeping location, and start at the same initial conditions. Lastly, in the input data block, additional problem conditions are defined. These conditions include the application of limited regions, problem weighting, previous data, and an error checker method. Limited regions are set by turning on either inertial or eclipse only limits on control. Eclipse only maneuvers are set to use either a user defined exclusion angle (angle from the Sun's unit vector where maneuvers cannot occur), the umbra region, or the penumbra region; which are calculated automatically. Inertial limits are defined by a vector and the angle from that vector where control is limited; several regions can be defined for a single run. Problem weighting is assumed to be minimum time unless otherwise specified, to save computation time, and is always minimum time for maneuvers with limited control. If specified the user can define the weighting parameter to value either solution final time or cost. This input func-

tions as a slider from 10 to -10, implementing near minimum time at 10 and heavily weighting cost at -10. Cost in this case is defined as the amount of cross sectional area used by both satellites and is defined to reduce energy loss from the system. Previous data either loads a previous GPOPS-II output or a control history. Previous outputs can be used as an initial guess or a first iteration of the error checker as specified. A limited control history is applied to the dynamics of the problem and resets the initial conditions assuming that control history had been applied to the system for its duration. If use of a control history is specified but no history is defined it assumes a brief acceleration towards the final desired location. The error checker compares the post processed final satellite position to the desired final position and attempts to re-run the solution to correct this error. Doing so provides a more accurate final position but prevents the solution from being truly optimal. The user can define the position error that is deemed acceptable, the maximum number of iterations to use (to include not using it at all), and if they want to use the last iteration as the guess for the next run. The last setting is not generally recommended as it can save time but does not always work. This usually occurs when the error checker switches to more accurate dynamics as error is iteratively reduced. This is useful for solution methods that use approximated dynamics as they can vary from the true propagation. All of this data is saved in arrays and structures called by the function later when the desired run is executed. By using this method several sets of data can be setup to run iteratively without additional user input. As several of the potential solutions can take hours to run this helps to avoid wasted time and effort.

3.2.3 Parameter Definition.

Parameter definition involves converting the appropriate run's input to initial and final conditions required by GPOPS-II, calculating additional condition based values, and setting the error and phase loops as appropriate.

For initial conditions this section calculates both the LVLH initial parameters of the deputy and the initial chief orbital parameters. For the first and second initial condition input types chief orbital parameters are calculated assuming an initial circular orbit of the specified inclination and RAAN placing the chief at the appropriate argument of latitude. The initial angular momentum of the chief satellite is calculated from the given altitude assuming a circular orbit where $\dot{r} = 0$, unless otherwise specified. For ECI input, chief orbital elements are directly converted. Deputy LVLH position and velocity are calculated from the initial separation distance or ECI frame position and velocity or set directly if using the second initial condition type. Final conditions are set so that the deputy satellite ends on the chief's circular orbit at the final separation distance required. This is done so that the deputy satellite ends in what would otherwise be a marginally stable equilibrium solution for the unperturbed nonlinear relative dynamics to minimize any potential drift after solution completion. Since the algorithm is an open-loop control law intended as a ground station maneuver planner implementing the equilibrium final condition is important to accurately control the spacecraft's position. The final position is calculated from the initial orbital radius of the chief, r, the maneuver size, and accounts for the maneuver direction, be it along or against the direction of travel.

$$h_0 = \sqrt{\mu r} \tag{3.56}$$

$$x_f = \frac{M_{size}^2}{-2r} \tag{3.57}$$

$$y_f = \sqrt{r^2 - (r + x_f)^2} sign(M_{size}) \tag{3.58}$$

where h_0 is initial angular velocity and M_{size} is the desired change in separation distance compared to the separation distance. By the definition of the LVLH frame $x_f \leq 0$ for the deputy to end in the same orbit as the chief with a phasing difference. For small maneuvers this distinction is almost negligible, being close to the linear case, but can result in several meters or kilometers of difference for larger maneuvers, especially those approaching 1000 km sizes. For these conditions $\dot{x}_f$ and $\dot{y}_f$ are both set as zero to eliminate any in-plane relative velocity that would perturb the equilibrium. All other final conditions are unconstrained for the optimal control problem. For runs where initial separation distance is defined the same method is applied to find the deputy's relative initial conditions, assuming it starts from the equilibrium solution; otherwise the relative initial conditions are applied directly.

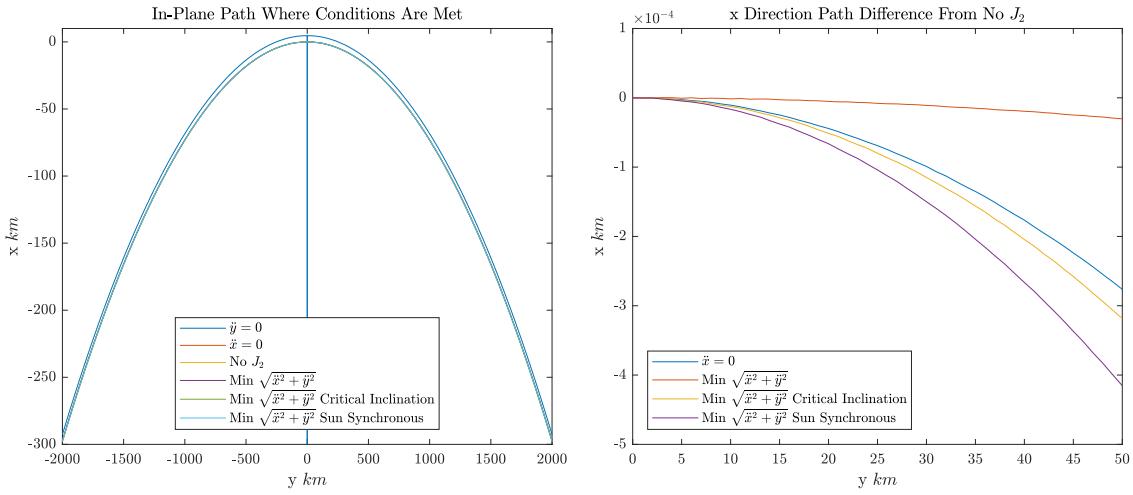

Figure 4. The location of the equilibrium path compared to the path of minimum in-plane acceleration under J_2 effects.

Unfortunately, this equilibrium solution is not a perfect solution. This location is only in equilibrium when the formation is unperturbed. When perturbed by J_2 effects there will always be an in-plane relative acceleration on the deputy. This occurs because the path where $\ddot{x} = 0$ and $\ddot{y} = 0$ no longer equal one another; only providing

an equilibrium solution at (0,0) because $\ddot{y}$ is zero at all x values when y is zero. While this acceleration is small, it will prevent any maneuver presented in this work from maintaining its final position without formation keeping. However, this acceleration can be minimized and its minimum point is approximated by the solution already presented. The minimum path of acceleration follows the unperturbed equilibrium solution very closely at low inclinations; differing only by 12 cm at a 100 km y separation at a 28.47 degree inclination. As inclination increases and the J_2 perturbation becomes more pronounced the path of minimum acceleration differs more from the unperturbed equilibrium and the magnitude of the minimum acceleration increases, as shown in Figures 4 and 5. Because of this, higher inclination solutions will require more control to apply formation keeping.

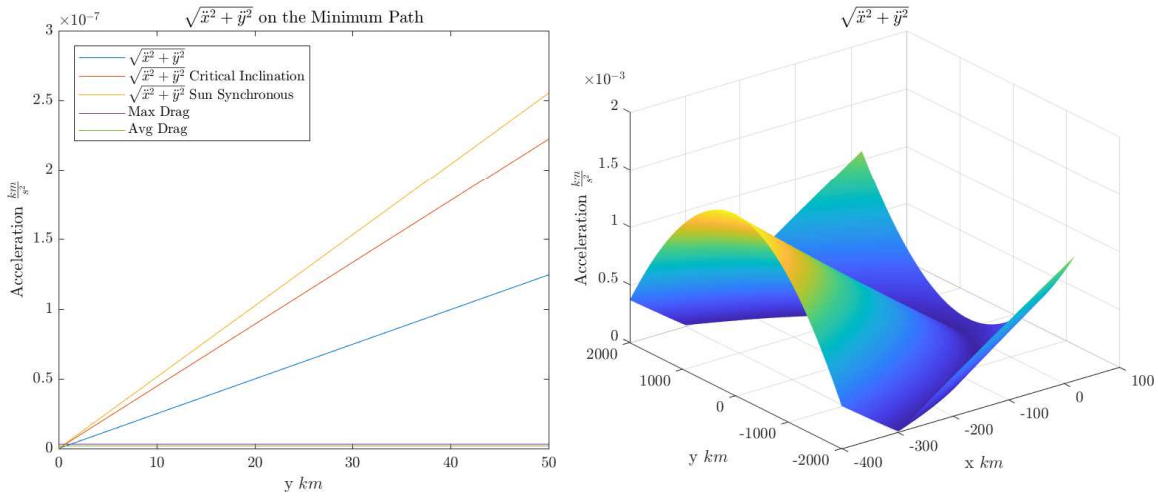

Figure 5. Magnitude of the in-plane acceleration vector as a function of position, and a plot if its minimum values based on y location.

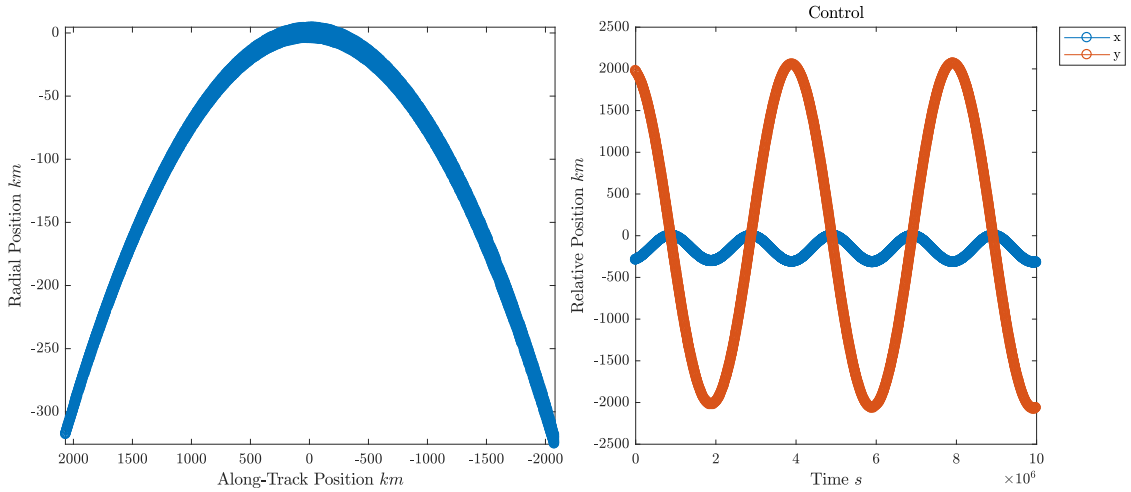

Figure 6. Uncontrolled Spacecraft Motion.

These accelerations are a result of short and long period oscillations caused by J_2. Equations 3.59-3.61 define the conditions necessary to enforce an invariant relative orbit in the presence of the J_2 perturbation [21].

$$\delta\dot{\lambda} = -\Big(\frac{3}{2a_0^{5/2}}\Big)\delta a - \epsilon\Big(\frac{21}{8a_0^{9/2}\eta_0^4}\Big)\big(\eta_0(1-3\cos^2 i_0)+(1-5\cos^2 i_0)\big)\delta a + \epsilon\Big(\frac{3}{4a_0^{7/2}\eta_0^5}\Big)\Big(\big((9\eta_0+20)\cos^2 i_0-(3\eta_0+4)\big)\delta\eta+\eta_0(3\eta_0+5)\delta i\sin 2i_0\Big) \quad (3.59)$$

$$\delta\dot{g} = -\epsilon\Big(\frac{21}{8a_0^{9/2}\eta_0^4}\Big)(1-5\cos^2 i_0)\delta a + \epsilon\Big(\frac{3}{4a_0^{7/2}\eta_0^4}\Big)\big((5\eta_0\sin 2i_0)\delta i-4(1-5\cos^2 i_0)\delta\eta\big) \quad (3.60)$$

$$\delta\dot{\varphi} = -\epsilon\Big(\frac{21}{4a_0^{9/2}\eta_0^4}\Big)\cos i_0\delta a - \epsilon\Big(\frac{3}{2a_0^{7/2}\eta_0^5}\Big)(4\delta\eta\cos i_0+\eta_0\delta i\sin i_0) \quad (3.61)$$

$$\eta = \sqrt{1-e^2} \quad (3.62)$$

where $\delta\dot{\lambda}$, $\delta\dot{g}$, and $\delta\dot{\varphi}$ are argument of latitude, radial, and cross-track drift rates; e_0, a_0, and i_0 are initial eccentricity, semi-major axis, and inclination of the chief;

and δe, δa, and δi are the deputy orbit's differences from the chief parameters. Since the solution starts in the same orbit as the chief δi, δa, and $\delta \eta$ are all initially zero, meeting the requirements of these equations to have an invariant orbit. Acceleration is calculated despite this and so must be attributed to long and short osculating motion caused by J_2 [15]. This is reinforced by observing the uncontrolled system behavior. The acceleration creates long period oscillations that last for approximately 46 days, for the cases presented here, and trace the equilibrium solution to the inverse matching separation and back, as shown in Figure 6. This oscillation time does not vary regardless of the initial separation distance, behaving much like a pendulum who's acceleration depends on distance from the chief. Furthermore, the acceleration and oscillatory motion disappears completely when the J_2 variable is zeroed. This means that for formations that only need to be at separation distance occasionally the system could be allowed to drift; and that small formations will have smaller relative velocity build up. However, cases studied here assumed formations needed to maintain the separation indefinitely. This equilibrium solution will also degrade with time. For the final conditions there will be some small δi, but it is assumed to be small enough to ignore and continue using the stated equilibrium solution. Inclination control requires cross-track control which is not possible with drag [22]; therefore as δi increases additional accelerations will plague the system in the stated equilibrium solution. This issue is ignored as it cannot be controlled, runs presented here all started at $\delta i = 0$ but built up some small variation during the execution.

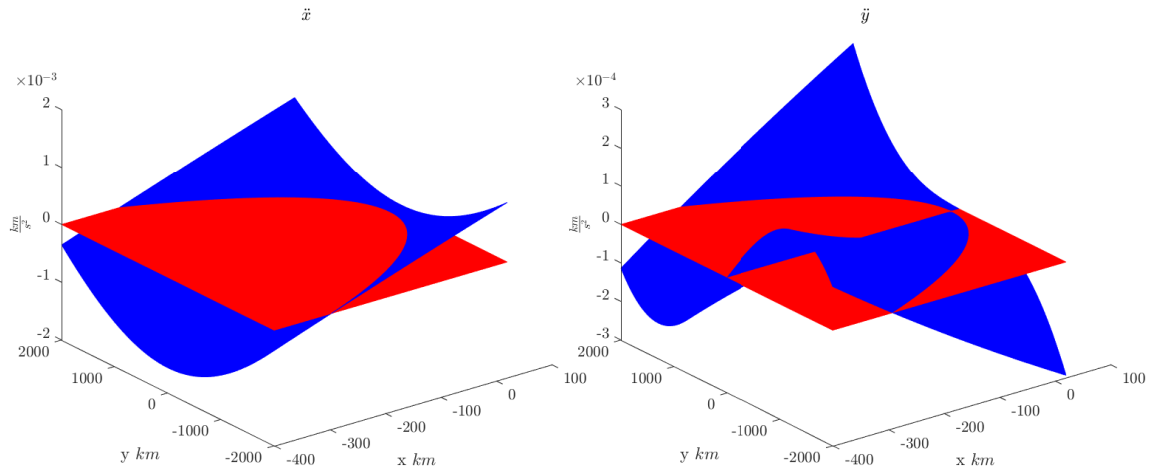

Figure 7. Component In-Plane Accelerations

Additional values calculated by this section are the umbra and penumbra Sun angles and the weights on time and control when used. The exclusion angle for eclipse limited maneuvers is calculated based on the chief's initial orbital radius as follows; depending on if umbra or penumbra angle was desired.

$$SUN_{umbra} = \pi - \frac{R_e}{r_0}\left(\frac{\pi}{2} - \alpha_u\right) + \alpha_u \tag{3.63}$$

$$SUN_{penumbra} = \pi - \frac{R_e}{r_0}\left(\frac{\pi}{2} + \alpha_u\right) - \alpha_u \tag{3.64}$$

where SUN is the respective control exclusion angle, $\alpha_u = 0.26411888°$is the angle of the Earth's umbra, r_0 in the initial chief's orbital radius, and R_e is the Earth's radius [15]. Weights for time and control are set from the single slider variable defined in user input. This scales the input (-10 to 10) to a weight for both parameters from 1 to 1,000,000 linearly where positive values of the slider are associated with weight on final time. At input 0 both final time and control weights are 1, and whatever weight is not being valued stays at one. This scale was selected to allow set weights to produce nearly identical results as a minimum time solution, and also use minimum control only solutions. Due to the nature of the problem (that solution drift allows

for lower control if more time is allowed to the solution) it is not recommended to use a full minimum control solution.

Last in this block, the phase and error loops are initialized. The error loop and phase loops clear the GPOPS-II setup structure here before the rest of the calculation on each run to prevent indexing errors in the structure. Both loops are while loops which will continue until their conditions are met. If either parameter was not set to run the loops will automatically end after one iteration. The error loop only iterates after post processing if the error checker was set for that run, setting the error met condition to true if it was turned off, and contains the phase loop process. It only updates the initial conditions of the problem assuming a section of the control history from the previous GPOPS-II run was applied to the problem dynamics. If a limited control region is set, the solution algorithm has to follow a modified procedure to account for the use of phases. GPOPS-II phases allow several sets of dynamics to be used and solved for simultaneously; and are used for those solutions which require regions without any control authority. This allows the controller to use distinct phases for eclipse and Sun-lit regions for the eclipse only control case, and any other limited regions as required by other cases. Within illuminated and inertial limited regions each satellite's projected area is modeled as the minimum value, which for all cases in this thesis is 0.03 m^2.

Using phases is important as directly removing control from the dynamics during the illuminated phases causes computational difficulties in GPOPS-II. This is because changing the control in these regions has no effect on the cost function and an accurate control history cannot be solved for. Unfortunately, GPOPS-II cannot handle an unknown number of phases. Each phase functions as its own smaller NLP and so for each the bounds, dynamics, initial guess, and even the mesh tolerances used by GPOPS-II must be set. GPOPS-II requires two distinct data points to provide

a solution or its gradient calculator does not work. Since each phase is a smaller GPOPS-II solution set this means each phase must occur for a finite amount of time or the solution cannot solve. Therefore the solver cannot determine the correct number of phases and they must be determined by the solution algorithm. Because neither the optimal number of phases nor maneuver duration is known a priori this is done by using a phase loop. Initially low number of phases are set from the initial guess then as the algorithm determines it needs more phases the following sections are run iteratively using the new set number of phases calculated in post processing. Defining parameters from input initializes the number of phases to zero so that they can be set from the initial guess on the first iteration and be updated later.

3.2.4 Initial Guess.

The initial guess generation is an important component of the algorithm as it generates data used to set the problem bounds, initial guess, mesh setup, and phases. The initial guess for this problem uses a propagation of the dynamics in *ode45*, a fourth-order Runge-Kutta numerical integrator for MatLab, using a rough bang-bang feedback control law. This propagation uses the same start times and initial conditions as the optimal solution using a full set of the dynamics and supporting parameters, such as the Jacchia-Roberts atmospheric model. The feedback control law assumes a bang-bang control law expected for minimum time control problems.

This law's switching is based on the deputy's distance from the initial relative position, the size and direction of the maneuver, and the relative energy state of the two satellites. First this law evaluates the relative initial energy of the satellites, if it is not equal it sets the control of that satellite with a higher state to max. This difference in energy occurs when one satellite would drift relative to another and the law allows the excess energy to bleed off the offending satellite. This is done to mimic

initial separation control where the satellites have an initial drift, and is tailored to y-direction drift. Once the satellites have equivalent energy, evaluated at each time step, the law switches to maneuver control after reseting initial separation to the current location. From there it sets control to move the satellite in the direction of the final conditions; for a positive maneuver (along the velocity vector) the deputy is set to max. This switches to deceleration once either Equations 3.65 or 3.66 becomes false,

$$(M_{size}/2)C_g > y - y_0 \tag{3.65}$$

$$(M_{size}/2)C_g < y - y_0 \tag{3.66}$$

where Equation 3.64 applies to positive maneuvers and Equation 3.65 negative ones. This switch slows the relative velocity as the final conditions are approached. In these equations y_0 and M_{size} are both reset from the end of the energy balancing phase to maintain the same final conditions when y_0 switches to the separation that exists at the end of that phase. This ensures the spacecraft move the right direction once drift is eliminated, regardless of the drift direction. The term C_g is a constant scaling term used to tweak the switching to better fit the required results. This is set so that the switching always causes the solution to overshoot the intended target rather than slow down and come up short. While this does not matter for the GPOPS-II initial guess much it has an impact on bound definition and defining final time appropriately. This value is set to 1.5 heuristically so that the overshoot is achieved for large maneuvers; small maneuvers work with either 1 or 1.5. This control law is scaled by the weighting on control versus final time. This is set so that the maximum control is scaled linearly towards the minimum control as the weighting shifts from the minimum time to the minimum control weighting. This means that while the switching still occurs under the same conditions the solution will take longer as it has less control authority.

Two problems that occur with this method of evaluation are the undefined final time and if the energy difference drift is useful. To account for this the propagation is run iteratively to search for the appropriate final time and evaluate when the satellites' energy balances. This search is a key function of the initial guess as the time bounds need to vary greatly depending on the time of year, altitude, and maneuver size of the run and final time is not known a priori. Once each *ode45* propagation is completed control and energy state histories are calculated and evaluated along with final conditions to detect if the final separation distance has been passed while in the correct energy state. During this iterative search the point where satellite energy becomes equivalent is evaluated based on distance to the final point to determine if it is necessary for the maneuver. It is necessary in those cases where the energy difference causes the solution to overshoot the final separation. If energy cancellation was not required to accurately perform the maneuver that feature is turned off and the drift accumulated during the cancellation is recorded. This distance is then used to update the maneuver size the switching control law considers; so that if the maneuver starts with momentum towards the final point the control law treats it as if its already completed some acceleration portion of a longer maneuver. If the final position and energy conditions are not met propagation time is increased and the initial guess is calculated again. Propagation time is set initially at 1000 seconds and increases by 1000 until 10000 where it increases by 10000; this is repeated at each order of magnitude. This method allows an appropriate propagation time to be determined in a timely manner without requesting too much propagation time, which can generate accuracy issues in *ode45* and take a long time to compute.

For large maneuvers (1 km and larger) the final propagation condition is only that the energy state is satisfied and the final position has been passed. Small maneuvers are propagated further to allow at least one encirclement of the final point. This

encirclement helps alleviate instabilities and naturally occurs when the switching law is propagated further through time. This behavior better mimics optimal control for small maneuvers as the deputy may be too close to the final location to get there without circling the in-plane position several times due to low control authority. This is set to only generate one encirclement and works for data presented here. Highly constrained cases could be encountered which would require more than one encirclement in the initial guess but are not considered here.

Once the loop is termination conditions are met an index for when these conditions are met is recorded. This ensures a solution that required 10456 seconds terminates at 10456 seconds rater than the 20000 seconds it was set to run for, as the data would not be appropriate. Finally, this guess generation also computes a propagation for the same time duration but assuming inverse initial conditions and maneuver size. This is done to accurately generate bound data regardless of which satellite spends more time accelerating. This is particularly important for the chief satellite as its orbital elements only degrade if it accelerates. For formation keeping the control law is set to zero and the system is allowed to drift. This process assumes the system starts in an equilibrium solution and should stay relatively stationary.

To implement limited control the initial guess accounts for where control is not allowed by defining the cross sectional area in those regions. In doing so it keeps track of when the solution is limited during guess generation. This data can then be used to determine the initial phase condition, limited or control, and the number of phases. For the first iteration of phases the initial guess generation remains otherwise unchanged since its used to set the initial phase number. In determining the initial number of phases it is important to underestimate the number of phases. If too many phases are set, the solution will not solve optimally; rather, it will either provide a non-optimal solution to provide enough time for all the phase switches to occur or

it may not solve at all. Since the initial guess is designed to take less time than the optimal solution, by overshooting the final destination instead of fully slowing down, these conditions are already met and can proceed without alteration. However, following iterations must propagate long enough to encounter the set number of phases. This is important since accurate phase switching times improve computation time of the optimal solution. To do this the final time search conditions are changed to terminate the initial guess only if both the previous conditions are met and the appropriate number of phases have occurred. Both the guess propagation check and index determination are changed. The index is defined to include all of the data for the last phase even if it passes the final conditions. This prevents the last phase's initial guess from being sparse.

If an initial guess is defined by the user from either loaded GPOPS-II output or the error iterations (using the last run as a guess) the initial guess propagation process is ignored on the first phase iteration of the first and subsequent error iterations respectively. Additional phases require the propagation to be run as usual due to ensure the phase switching requirements are represented. This section also generates the propagation for the loaded control history, if set. If no control history is provided the assumed generated control is either a brief acceleration in the direction of travel (less than half the time of the generated guess) or is the applied control to cancel energy (if it was determined to be necessary). This propagation is recorded, initial conditions reset, and the initial guess calculated again from the new conditions.

3.2.5 GPOPS-II Bounds and Guess Definition.

For GPOPS-II to run upper and lower bounds on time, states, control, and any integrals must be set. Bounds on states must be set for initial, final, and continuous conditions for every phase. Therefore to set the problem bounds the number of

phases must be determined first. To ensure the solution is feasible the final phase must be set to a control phase. This allows GPOPS-II to ignore the remaining eclipse regions that should have otherwise occurred to provide a solution. To ensure this condition is met the initial phase determination uses the phase count from the initial guess but subtracts a phase if it would have ended in a limited control phase. Initial and final states are easy to set by defining the upper and lower bound variables as the same value, that of the required condition for that state variable; they remain simple for phases as initial and final conditions only apply to the first and last phases respectively; initial and final conditions on other phases match the general state bounds. Control bounds are defined as the maximum and minimum control, the maximum and minimum relative surface area of the satellite. For phases in limited regions control bounds are set to minimum control only. For the general state bounds and final time the bounds must be set to not interfere with the solution. This requires setting the bound large enough that the solution is not inhibited but no too large as it can increase mesh size and solution computation time. To do this the bounds are set based on how much each state variable changed during the initial guess. This calculation determines the maximum absolute difference between the initial state variable and its maximum and minimum propagated values for both the standard and inverse propagation. This delta value it then multiplied by a scaling constant and differenced from the initial conditions to define the state variable's upper and lower bounds. The chief radius variable's minimum is also constrained by the radius of the earth, and is set to whichever is larger to avoid computational errors. Final time simply uses the initial guesses final time scaled by a multiplier. When weighting is required on control, or formation keeping is requested, an integral must be calculated in the continuous function to add up the used control. Integral bounds must be set in this case; for the weight integral the upper bound is set to maximum control over

the duration of the time bound, and the minimum is zero. The formation keeping integral's minimum is zero, the maximum bound is set from the maximum possible separation defined by the state bounds. The scale value of the variables must be equal to or larger than that for time to account for any additional movement during the unpropagated extra time interval. The time scale should be set to at least two to ensure the optimal control has enough time to slow an overshoot initial guess. Heuristically a time and state multipliers of 3 and 3 were determined to fit best. Small maneuvers have a circular optimal path in the relative frame and could hit the boundaries when the state multiplier was set lower. Formation keeping only uses a state multiplier of 100; this was required since the problem can either remain relatively stationary or incur significant drift depending on the desired location. Since phases are used this section must also set the GPOPS-II event functions. This problem governs its phases by imposing three event functions. First, the phases must be continuous–that is, the final conditions of one phase must equal the initial conditions of the subsequent phase. Second, a phase must last for a finite period of time. Third, phase boundaries must occur as the satellite formation is passing into or out of the limited regions as defined by the region vector(s) and the exclusion angle(s), to include eclipse.

Similarly to setting the bounds an initial guess must be set for each individual phase. To do this the generated guess is segmented based on whether or not the propagation was in an a limited region. Foe example, this assigns all generated data until the first limit is reached, or exited, to the first phase. This repeats so that every time a new region is reached that segment of the propagation is assigned to the next phase's initial guess. This continues until the set number of phases are reached. Once this happens the remaining data is set for the final phase regardless of propagated regions. In setting the guess the maximum number of oscillations during that guess

is calculated for every state. This records every time the numerical derivative of each state changes sign. This information is used by the initialization block to calculate the initial mesh and improves solution run time by defining enough mesh intervals to account for periodic variables. This works because hp methods only converge exponentially when increasing collocation points where there are already enough mesh intervals to make each interval relatively flat [10]. This oscillation counting is done for each separate phases' initial guess and applies mesh settings separately for each phase as well.

3.2.6 GPOPS-II Initialization and Execution.

GPOPS-II initialization and execution defines the mesh settings, passes in auxiliary data, build the GPOPS-II setup structure, and excretes the problem. Auxiliary data allows the atmospheric data, constants, phase data, weightings, and limited region data to be passed into GPOPS-II. Mesh settings define a tolerance of at least 1e-4 for most cases but uses 1e-6 for small maneuvers where full dynamics accuracy is requested. Tighter tolerances than this do not usually provide higher levels of accuracy but do require much more time to solve, and anything lower compromises solution validity. This solver uses the hp-PattersonRao mesh update method, the default setting, with a 60 iteration max, 4 initial co-location points, and 10 maximum co-location points [10]. Initial mesh intervals are set as eight times the number of recorded oscillations or a base minimum, whichever is higher. The minimum is 20 or 80 depending on if the low or high mesh tolerance is set. It takes a long time for GPOPS-II standard mesh update methods to generate enough mesh intervals to handle the accuracy required for the several periodic state variables. The lower limit ensures accuracy for small maneuvers where position tolerance replaces periodic states as the driving requirement interval requirement. This mesh setup also

defines finer meshes at the beginning and end of the solution where it approaches final and initial conditions making it more susceptible to errors. These two ends contain as many intervals as the standard mesh at a 1/64 size. The number scales with the standard mesh interval number to ensure enough of the solution endpoints are covered by the finer mesh. $1/64^{th}$ size was determined heuristically as GPOPS-II mesh update doubled the end mesh intervals up to six times before required accuracy was met for standard condition cases. This initial mesh setting is usually enough to provide a viable solution on the first, or first few, mesh(s) cutting down run time significantly. Other settings of note are "sparceCD" for the derivative supplier with a "sparceNaN" dependency, "automatic-guessUpdate" as the scaling method, and "RPM-Differentiation" for the mesh method. "sparceNaN" requires the continuous function to accept NaN input variables to calculate correctly. The scaling selected, "automatic-guessUpdate", sets the problem scaling from the initial guess and updates the scaling based on the solution solved on each mesh. This allows the problem to handle large state variables for the chief orbital elements and small relative separation variables at the same time. It also scales the problem when maneuvers become large.

This section also defines the continuous and endpoint functions, setting the dynamics and cost function. The endpoint function defines the cost function and phase bounds through event functions. Selecting different event functions (as appropriate for the run conditions) allows different cost functions to be used.

$$J = t_f \tag{3.67}$$

$$J = w_t t_f + w_c \int_0^{t_f} (S_c^2 + S_d^2) dt \tag{3.68}$$

$$J = w_c \int_0^{t_f} (S_c^2 + S_d^2) dt + w_t \int_0^{t_f} \left((x - x_f)^2 + (y - y_f)^2 + w_v (\dot{x}^2 + \dot{y}^2) \right) dt \tag{3.69}$$

Equation 3.66 is the standard cost function and Equation 3.67 the only used when weighting is requested. Here S_c and S_c are the surface area of the chief and deputy respectively, w_c and w_t are the minimum time and control weights. Equation 3.68 is used for formation keeping. Here the time weight is used instead for final position (as it is set as a counter to the control weight and time is not a control factor). This cost function sets cost as the distance from the final position and the in-plane relative velocity. It is designed to try and provide a relatively stationary solution. w_v is a weighting on velocity set to account for the differences in magnitude between the variables; and account for how much velocity or position matter to the user. Implementing phases requires a change in the endpoint function to include the event function evaluations and an alteration to the standard cost function. Due to tolerances in solution approximation, SNOPT may encounter numerical difficulties; and IPOPT encounters computational inefficiencies compared to methods of the same duration without any phases. These problems can be mitigated by introducing the dynamics error on the phase boundaries into the cost function heavily weighted.

$$J = t_f + 1000 \sum_{j=1}^{P_N - 1} \delta x(j) + \delta \dot{x}(j) + \delta y_f(j) + \delta \dot{y}_f(j) \tag{3.70}$$

$$\delta v(j) = (v_f(j) - v_0(j+1))^2 \tag{3.71}$$

where P_N is the number of phases and $v_f(j)$ and $v_0(j)$ are the final and initial values of variable v (dummy variable for x, $\dot{x}$, etc.) for the jth phase. The weight was heuristically determined to be 1000; this value works well for the final times encountered in this paper and was not changed based on the number of phases. As more phases are implemented this term will be effectively weighted higher as the summation increases, as it approaches zero this has little impact on the solution final cost. This ensures that any solution that would violate the first event function is not optimal

and reduces numerical difficulties without changing the final solution. Unfortunately, this does not eliminate all numerical difficulties for SNOPT when using more than a few phases; frequently requiring IPOPT despite its longer computation time. This cost function also ignores the minimum control term. This change was done to reduce computation time and avoid potential numerical difficulties. Since control is weighted very little in comparison to the event function component it could be approximated as having zero effect, causing the same problems phases were designed to avoid. For similar reasons the formation keeping cases also had to be calculated using IPOPT; in that SNOPT could not solve the solution accurately. Though in these cases it is due to an initial guess which is too different from the optimal solution rather than phase errors.

Continuous function definition allows different sets of dynamics to be used as defined by user input. These follow all combinations of dynamic simplification and drag approximation outlined in user input. Separate continuous functions are also defined for phases, though this is only in program syntax which is unnecessary if using only one phase (but increases computation time). The first approximation, that J_2 does not influence the system, is simple to implement as it sets the K_{J2} terms to zero. This also simplifies the n_j^2 terms and eliminates the ζ_j terms from Equations 3.6-3.8.

$$\ddot{x} = 2\dot{y}\omega_z - (x+r)(n_j^2 - \omega_z^2) + y\dot{\omega}_z - z\omega_x\omega_z - \ddot{r} + A_{Dx} \tag{3.72}$$

$$\ddot{y} = -2(\dot{x}+\dot{r})\omega_z + 2\dot{z}\omega_x - (x+r)\dot{\omega}_z - y(n_j^2 - \omega_z^2 - \omega_x^2) + z\dot{\omega}_x + A_{Dy} \tag{3.73}$$

$$\ddot{z} = -2\dot{y}\omega_x - (x+r)\omega_x\omega_z - y\dot{\omega}_x - z(n_j^2 - \omega_x^2) + A_{Dz} \tag{3.74}$$

$$n_j^2 = \frac{\mu}{r_j^3} \tag{3.75}$$

$$\omega_x = \frac{rA_{Cz}m}{h} \tag{3.76}$$

$$\dot{\omega}_x = \frac{\dot{r}A_{Cz}m}{h} - \frac{\dot{h}rA_{Cz}m}{h^2} + \frac{r\dot{A}_{Cz}m}{h} \tag{3.77}$$

$$\ddot{r} = -\frac{\mu}{r^2} + \frac{h^2}{r^3} + A_{Cx} \tag{3.78}$$

$$\dot{h} = rA_{Cy} \tag{3.79}$$

$$\dot{\Omega} = \frac{r\sin\theta A_{Cz}}{h\sin i} \tag{3.80}$$

$$\dot{i} = \frac{r\cos\theta A_{Cz}}{h} \tag{3.81}$$

Other than these effects on the relative and chief parameter dynamics the problem remains the same. Removing drag from the chief parameter dynamics removes all A_C terms from the chief orbital parameters, making them all constant with the exception of $\ddot{r}$ and $\dot{\theta}$; ω_x also becomes constant as zero.

$$\ddot{x} = 2\dot{y}\omega_z - (x+r)(n_j^2 - \omega_z^2) + y\dot{\omega}_z - \ddot{r} + A_x \tag{3.82}$$

$$\ddot{y} = -2(\dot{x}+\dot{r})\omega_z - (x+r)\dot{\omega}_z - y(n_j^2 - \omega_z^2) + A_y \tag{3.83}$$

$$\ddot{z} = -zn_j^2 + A_z \tag{3.84}$$

$$\ddot{r} = -\frac{\mu}{r^2} + \frac{h^2}{r^3} \tag{3.85}$$

$$\dot{\theta} = \frac{h}{r^2} \tag{3.86}$$

Here the relative drag acceleration is used again since drag on the chief is no longer propagated in its parameter dynamics. The last dynamics approximation is that the chief has a circular orbit and is not affected by perturbations. Using this approximation r and $\dot{\theta}$ are now constant and h is no longer used for the dynamics.

$$\ddot{x} = \frac{-\mu(r+x)}{((r+x)^2 + y^2 + z^2)^{3/2}} + \frac{\mu}{r^2} + 2\omega_z\dot{y} + x\omega_z^2 + A_x \tag{3.87}$$

$$\ddot{y} = \frac{-\mu y}{((r+x)^2 + y^2 + z^2)^{3/2}} - 2\omega_z \dot{x} + y\omega_z^2 + A_y \tag{3.88}$$

$$\ddot{z} = \frac{-\mu z}{((r+x)^2 + y^2 + z^2)^{3/2}} - 2\omega_z \dot{x} + y\omega_z^2 + A_y \tag{3.89}$$

Using this approximation the relative dynamics regress to the nonlinear dynamics of the LVLH frame which are used to define the HCW equations [21]. Once again the cross-track motion is decoupled from the in-plane dynamics. The last approximation is using an approximation for atmospheric density. To do this an average density is calculated from the density calculated in the initial guess. Using the density history generated from the position and time history of the initial guess allows a more accurate approximation of density than a time static model, such as the exponential density model. This method lets the continuous function avoid evaluating the Jacchia-Roberts model at every time step but still use a density that is close to the values model evaluation would generate. As is longer maneuvers mean the average density will be more inaccurate at any given time.

3.2.7 Post Processing.

In post processing identical and true dynamics are propagated, the phases are evaluated, and the error is evaluated; once all of the phase and error loops are evaluated the data is saved for each run iteration and as a set once every run is completed. Once GPOPS-II output is generated the control history is propagated to ensure solution accuracy. This is done both for an identical set of dynamics as the chosen continuous function and for the full set of dynamics. This is done to ensure GPOPS-II is generating a solution accurately and that the system is behaving as desired. This propagation is completed using *ode45* and runs for the duration of the maneuver. To allow high accuracy solutions even with large data sets the propagation is broken up into manageable chunks. Starting from the GPOPS-II initial conditions 500 interval

blocks of time and control history are used to propagate the dynamics. The final conditions of the last propagation become the initial conditions of the next. This continues until the entire time and control history is used and the data is resembled into a single set. In validation it is important to ensure GPOPS-II generated enough data and that *ode45* has high enough propagation tolerances. If the control history is accurate but sparse it can't be propagated accurately and the problem dynamics are sensitive and a false propagation is easy to produce.

In post processing the number of phases the solution should have encountered is calculated from the *ode45* control history propagation. This value is compared against the number of used phases. If the true number of phases the exceeded number of phases the phase number is updated. This is done to ensure the final phase is still control, but rather than subtracting a phase an additional phase is added. Since we know GPOPS-II solved a minimum time solution there was no way to complete the maneuver faster, therefore if it would have terminated in a limited phase it will require the next control phase. It also prevents the solution from being stuck at almost solving in the current number of phases but trailing slightly into a limited region (subtraction in this case leads to an infinite loop). The updated phase number is recorded and the solution is run again. This process happens iteratively until the number of phases GPOPS-II uses matches the propagated output's phase count.

The Error Checker can operate in two modes as determined by the user. In method one it evaluates the rectangular position error of the final in plane relative position and velocity. It compares the true output to the desired final conditions. If the position accuracy is worse than the desired maximum accuracy another error iteration will be used. This process involves saving a section of the true propagated output and control history and resetting the initial conditions to the final conditions of this section. This treats the saved section as if it is a control history applied to the

system. This section is set as the time interval at the last time the relative position was five times the evaluated error. This allows the system enough time to correct the errors since the larger it is the more time it requires to correct. If the error is larger than 1 km the first half of the propagation is used. If the error is less than 1 km the next run are set to the full dynamics if approximated dynamics or density were used before. Once the error loop is satisfied, by either meeting the requested accuracy or reaching the maximum number of iterations, the total propagation history is resembled from the saved iterations. This method is designed to be used with the approximated versions of the dynamics by correcting for error incurred during those methods.

The second method uses set time intervals to determine when to cut the propagated data and run again. This method is more computationally intensive and requires more user management to work well; but can provide more efficient maneuvers by correcting error sooner and more often. These methods allow high accuracy solutions to be generated from long maneuvers. The longer the maneuver is the more error build up there is in the solution. This alleviates that by iteratively solving smaller maneuvers with initial conditions closer to the final position. By definition, however, this output is not an optimal solution since it is correcting errors. This mimics applying a section of the optimal control history to the system, monitoring real world position, and recalculating the maneuver from the new conditions; thus proving the tool is capable of this calculation.

3.3 Summary

This problem is set up modeling a real system. Nonlinear LVLH dynamics are used accounting for the J_2 and drag perturbations on both satellites. This allows drag to be modeled on each satellite individually rather than using relative drag.

The Jacchia-Roberts atmospheric model is used for accuracy and computational ease compared to other high-fidelity models; and this density is applied assuming a co-rotating atmosphere. The solver algorithm is complex and designed to handle multiple inputs. These are defined in the input block, parameter definition converts input into the proper form and initializes loops, initial guess generates data used by the rest of the problem using *ode45* propagation, bound definition sets GPOPS-II bounds and phase data while sorting the initial guess, initialization selects the dynamics and cost function and runs GPOPS-II, post processing evaluates the output for accuracy, loop conditions are checked, and finally data is saved. Sample data generated by this solver algorithm is presented in the following section.

IV. Implementation and Analysis

4.1 Resizing Maneuvers

All results presented here represent minimum time maneuvers for a formation at 500 km altitude conducted at solar max using no assumptions in the dynamics; unless otherwise stated.

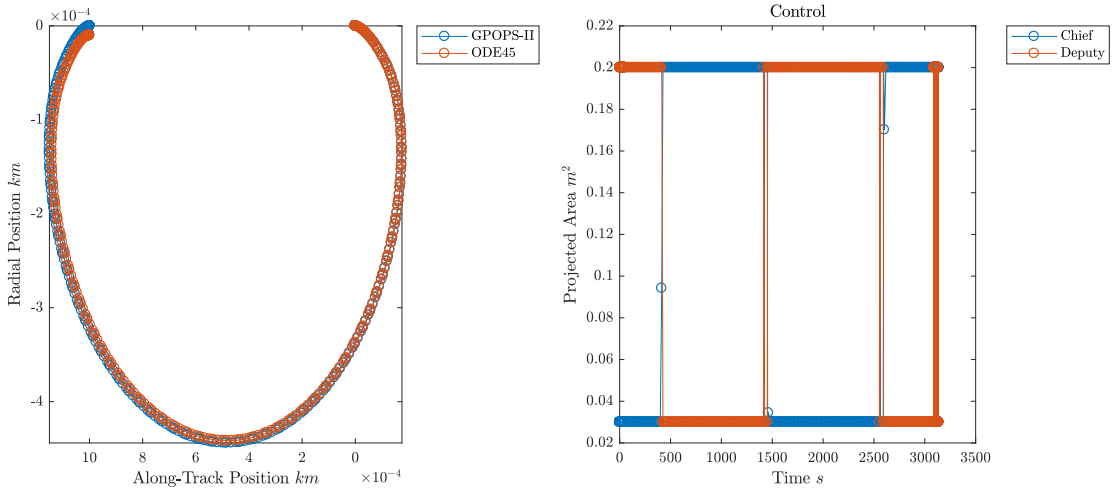

Figure 8. 1 m Maneuver

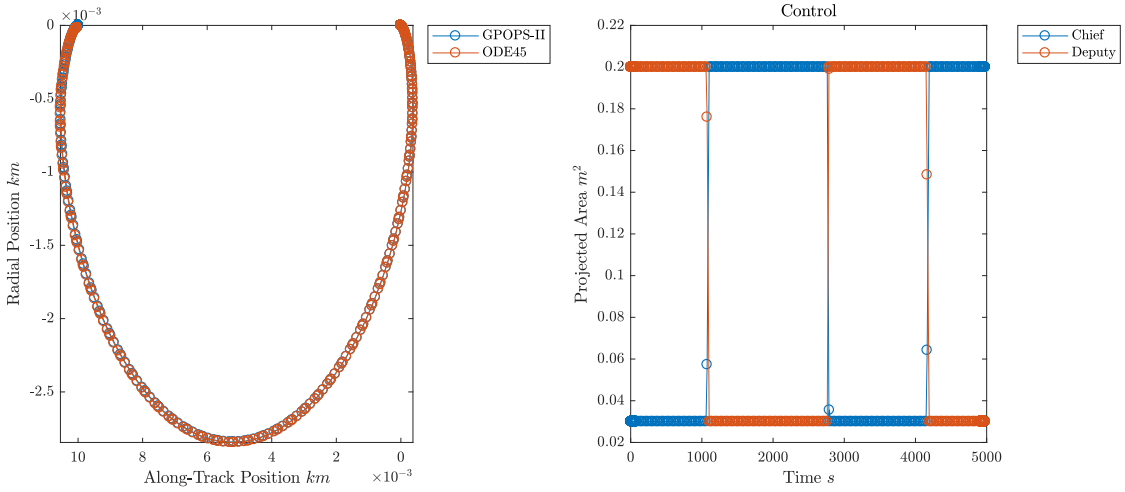

Figure 9. 10 m Maneuver

All minimum time results produced bang-bang control histories despite being able to choose the control. This indicates that there are no points during the maneuver where the dynamics are so insensitive to control that it cannot improve the cost function. This bang bang control shows variation dependent on when the maneuver starts, such as the deviation shown in Figure 10, associated with the atmospheric model and modeled changes in density.

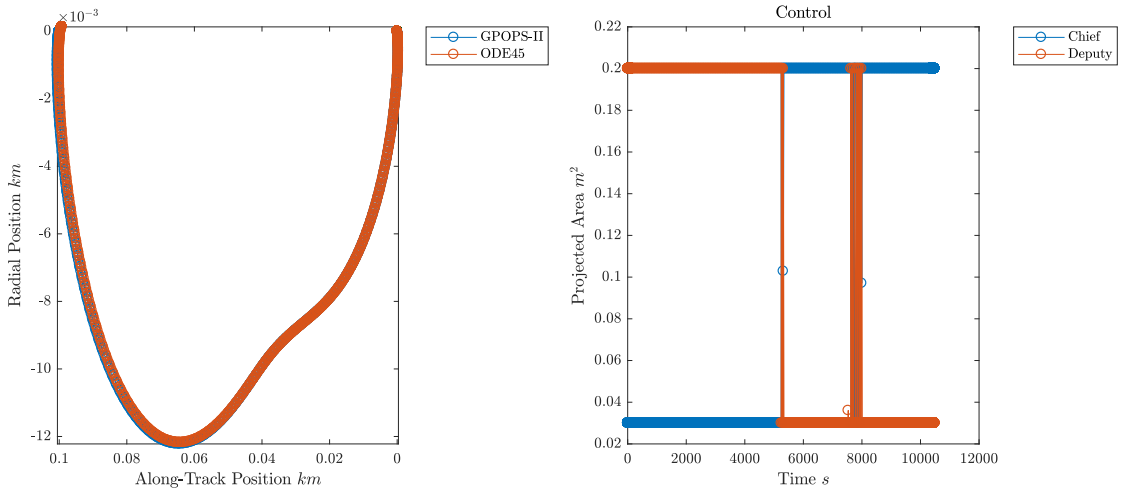

Figure 10. 100 m Maneuver

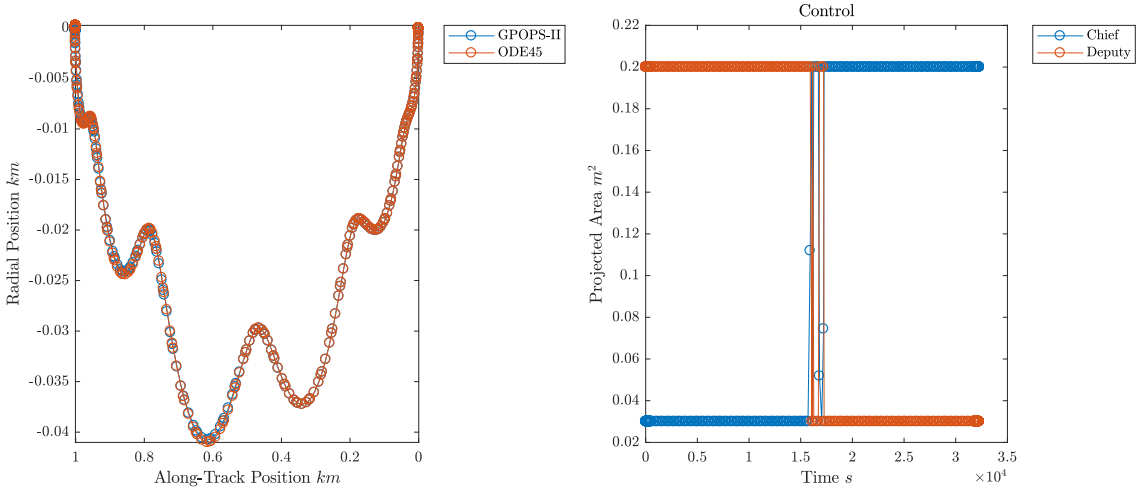

Figure 11. 1 km Maneuver

Figures 8 through 12 show the calculated optimal control history, the output generated by GPOPS-II, and the control history propagated by *ode45* to match the full nonlinear dynamics used to solve the optimal solution. All generated solutions retain some error due to the nature of the numerical approximation used by GPOPS-II. This error affects the larger solutions more when measuring the physical error in final in-plane location, but has a higher percentage effect on the smaller maneuvers. This can be seen in Figure 8 and results from the small maneuvers dependence on small changes in the dynamics.

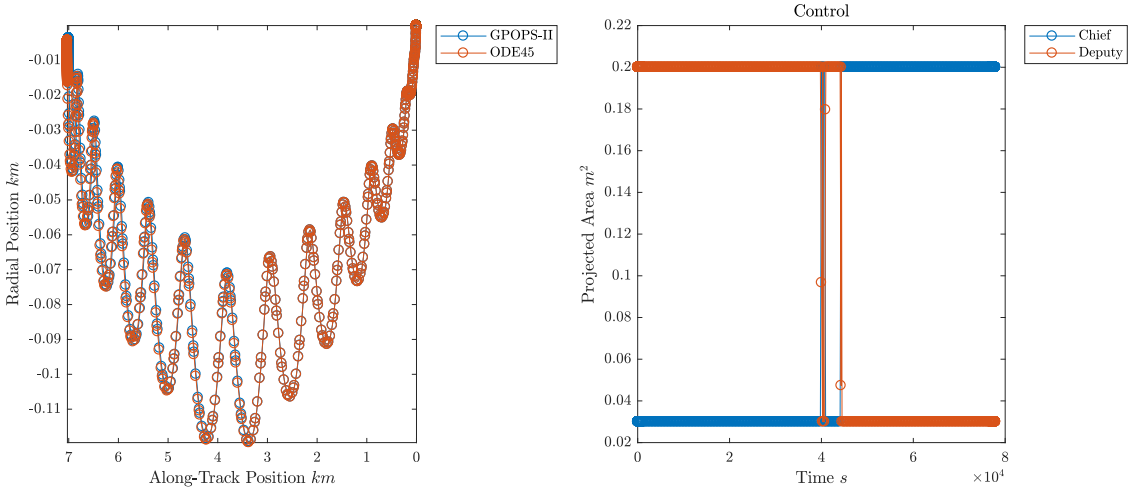

Figure 12. 7 km Maneuver

As the size of the maneuver increases the in-plane trajectory changes. For small maneuvers this trajectory bows out in the beginning and end as the deputy satellite initially decreases its energy to drop into a lower orbit and accelerate relative to the chief; this process is inverted at the end of the maneuver. This affect is still present for the larger maneuvers but has a less visible impact on trajectory shape as the sheer size of the maneuver increases. More importantly, as the maneuver duration increases the relative dynamics become affected by orbital period. Since orbital period at this altitude is 5677 seconds the 1 km maneuver should proceed over 5.7 orbits, and the 7 km solution should take 13.7. As seen from Figures 11 and 12 this coincides with the

number of oscillations in the relative trajectory. Lastly, as maneuver size increases the control history changes from a rapid switching profile to approximating a single switching law, like what is used for the initial guess.

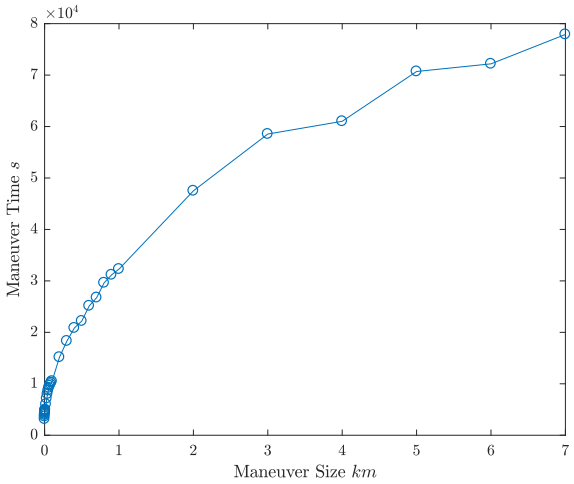

Figure 13. Maneuver Time vs Size

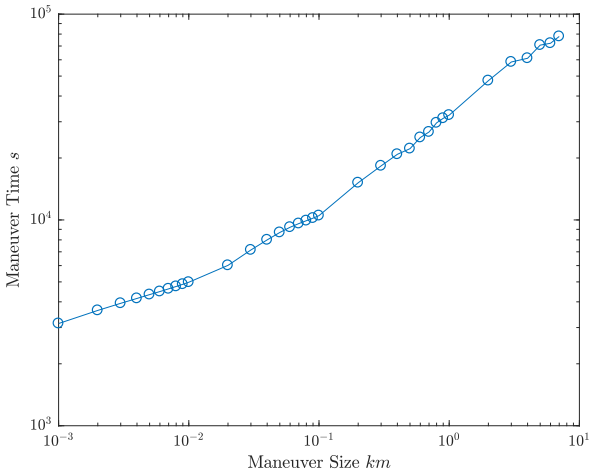

Figure 14. Figure 13 on a log scale

From Figures 13 and 14 we can infer that maneuver size and time are related by a square root power. This relationship can be explained by the acceleration in the problem. Though a maneuver doubles the time to complete it does not because

the spacecraft can build up more speed and complete the maneuver faster. This is possible because of the continuous relative acceleration on the problem.

4.2 Approximated Dynamics

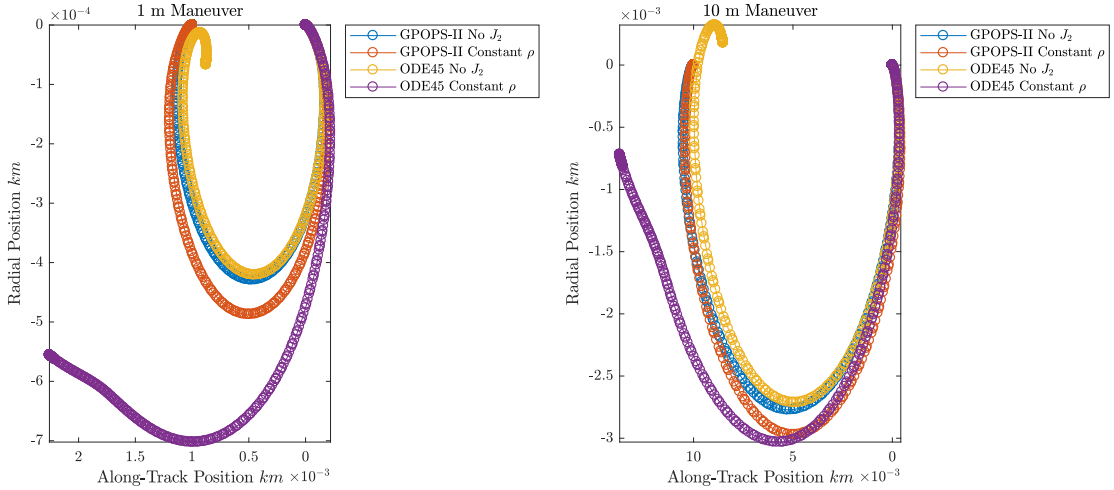

Figure 15. 1 m and 10 m Maneuvers with Assumptions

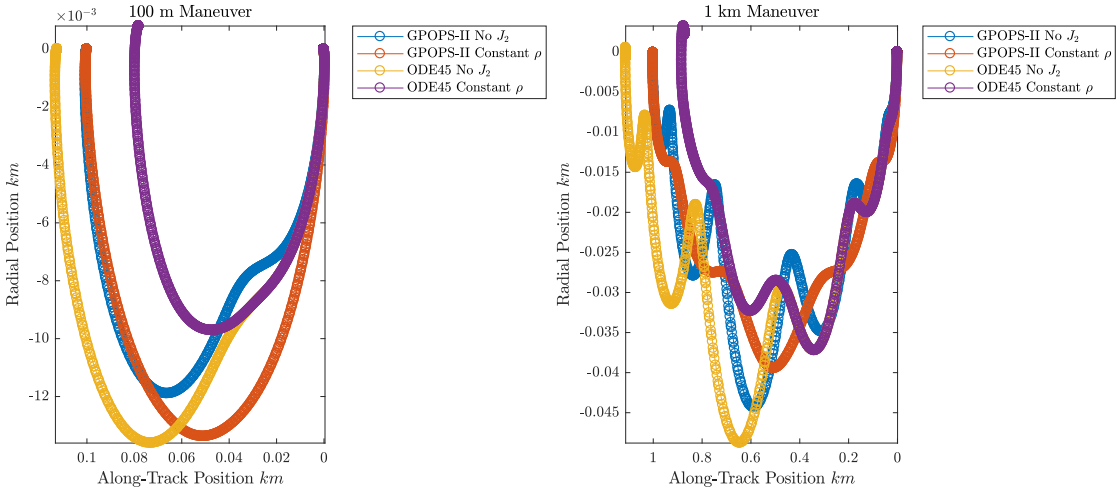

Figure 16. 100 m and 1 km Maneuvers with Assumptions

Figures 15 and 16 show the difference in trajectories and accuracy of the final state when comparing the raw GPOPS-II output and post processing propagation of dynamics assuming no J_2 perturbations and using a constant density. Constant

densities used here were calculated as an average of the Jacchia-Roberts density over the duration of the initial guess. The density used in the 1 m maneuver was $1.2535 * 10^{-12} kg/m^3$ during which the density varies from $1.6102 * 10^{-12} kg/m^3$ to $7.2020 * 10^{-13} kg/m^3$. Despite the obvious error this approximation introduces into the system the density is still much closer to the true value than a lower fidelity model; such as the exponential model which would set $\rho = 6.9670 * 10^{-13} kg/m^3$.

Table 4. Maneuver Final Position Error

Maneuver Size	Full Dynamics	Percent	No J_2	Percent	Constant ρ	Percent
1 m	1.0333e-5	1.03	1.3932e-4	13.9	1.3750e-3	137
10 m	3.8673e-5	0.39	1.4977e-3	15.0	3.7855e-3	37.9
100 m	7.7016e-4	0.77	1.2531e-2	12.5	2.1871e-2	21.9
1 km	3.2543e-4	0.03	1.1195e-1	11.2	1.2955e-1	13.0

As the maneuver's duration increases the constant density assumption has less of a detrimental effect on the problem. This is likely because the duration of the problem allows it to even out where small maneuvers are affected easily by the differences. The removal of J_2 however seems to introduce a relatively constant error between 10% and 20% of the final position. These results validate the need for including the J_2 perturbation in the solution dynamics even at relatively small separation distances.

4.3 Error Checker

As stated by previous sections there are inherent errors in the solution due to GPOPS-II mesh tolerances, the number of collocation points, and the accuracy of the dynamics themselves. The errors inherent to GPOPS-II are usually small, relative to the problem, but the error in simplified versions of the dynamics can be quite large. Despite this there is an interest in using these dynamics because they are much easier to compute and can solve longer solutions when GPOPS-II begins to run into

computational problems with the more accurate dynamics sets. To account for this error the discussed error checker is applied to the problem.

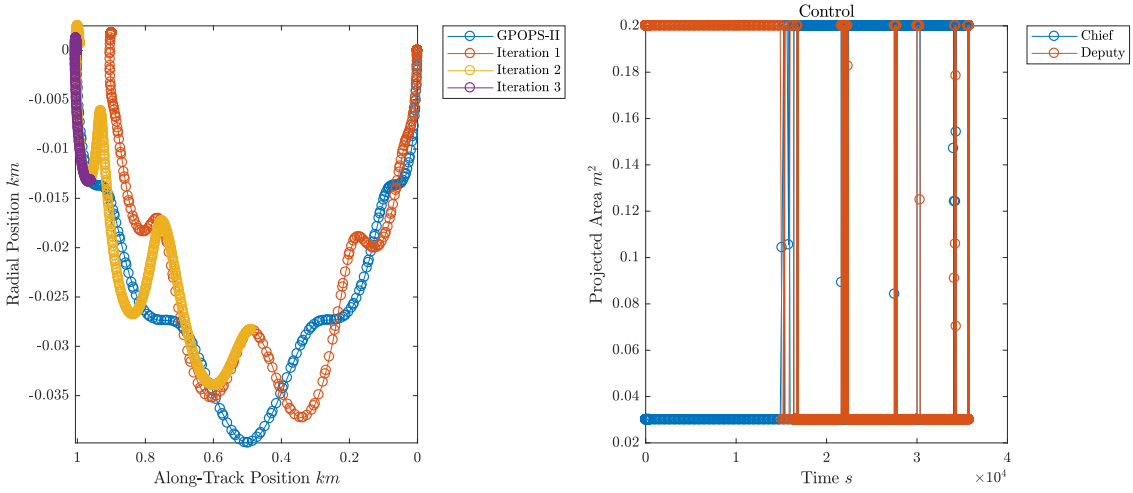

Figure 17. 1 km Error Checked Solution and Iterations

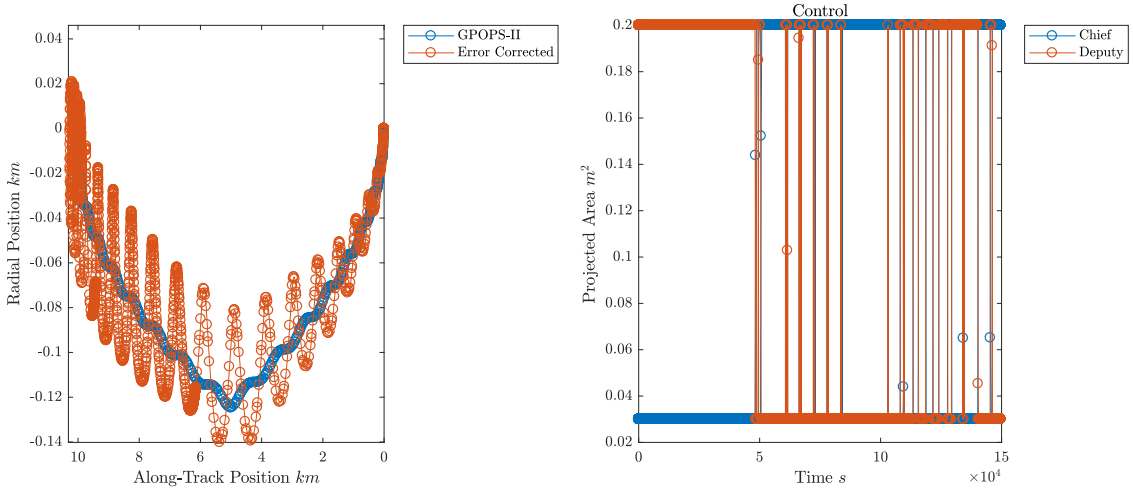

Figure 18. 10 km Error Checked Solution

Figures 17, 18, and 19 show the results from the error checker being applied to solutions using circular orbit, no J_2, no drag on the chief, and constant density approximations. These results show the first error checker method which makes corrections based on final position error. These plots show how each iteration of the solution provides a more accurate final position, but trades off maneuver time

as the solution now takes additional time to correct the error. The optimal 1 km maneuver using the full dynamics requires 32290 seconds to complete, compared to the error checked version's 35761 seconds. This is a 10.7% increase in maneuver time which can be seen as a loss of optimality.

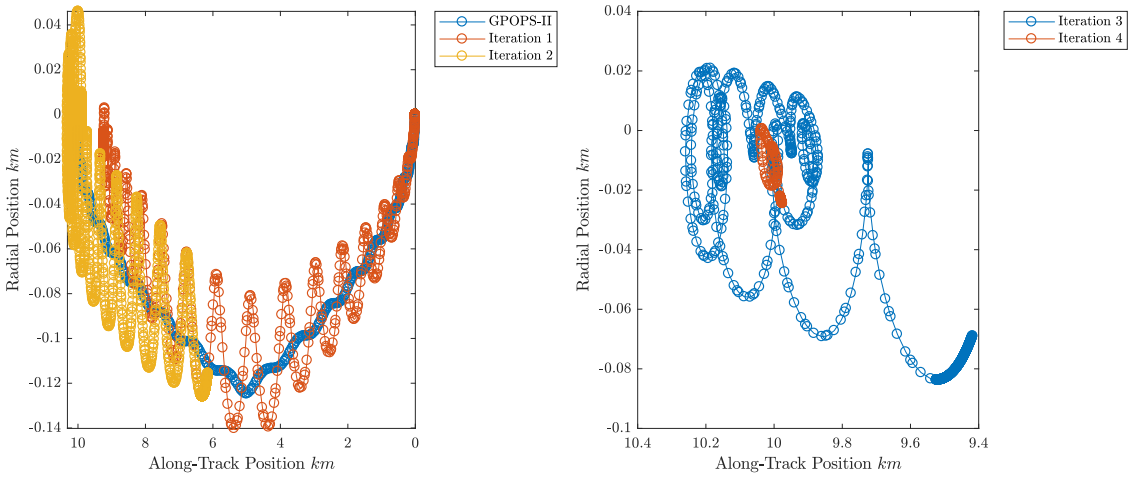

Figure 19. 10 km Error Checked Solution's Iterations

The alternative to this method, the interval error correction, is shown in Figures 20 and 21. This method assumes there will be error and attempts to correct it by recalculating the solution from new initial conditions at set intervals. In doing so it can provide a more accurate solution than if the deviation was ignored. Each iteration of this method uses the same assumptions as the first rather than switching to higher fidelity dynamics like the error correcting method. This method solves faster for individual solutions but can take longer to calculate if a small interval is set. In general this method will provide a solution with a better maneuver time than the error checker but at either an increased final position error or additional computations depending on the size of the time interval.

Table 5. Maneuver Final Position Error

Correction	1 km Error (km)	Maneuver Time (s)	10 km Error (km)	Maneuver Time (s)
Original	0.1015	31710	0.7704	98850
Interval	0.0170	57230	0.1720	94980
2nd	0.0077	35660	0.1161	126000
3rd	1.217e-04	35761	0.0048	141400
4th	NA	NA	9.568e-04	149700

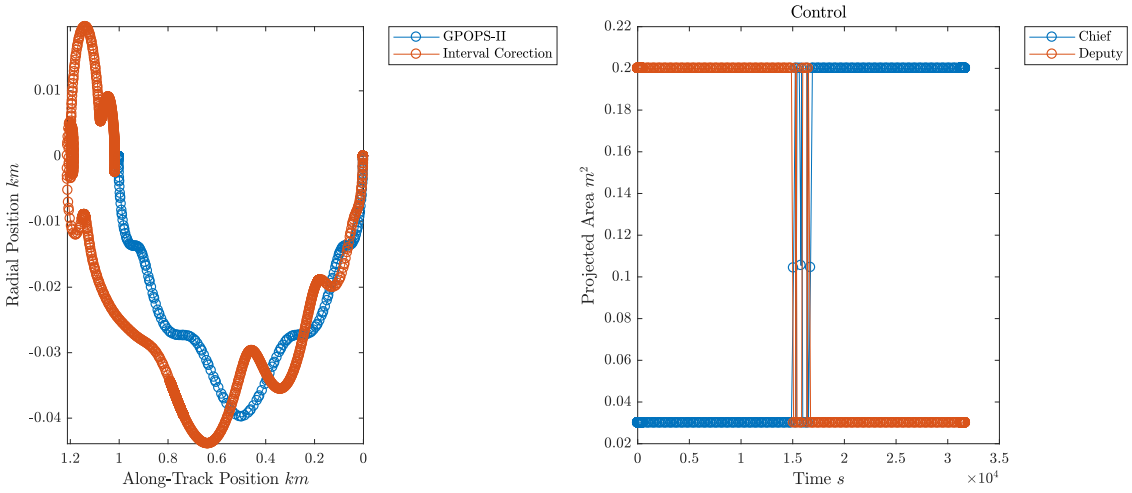

Figure 20. 1 km Maneuver with Interval Error Correction, every 10000 seconds

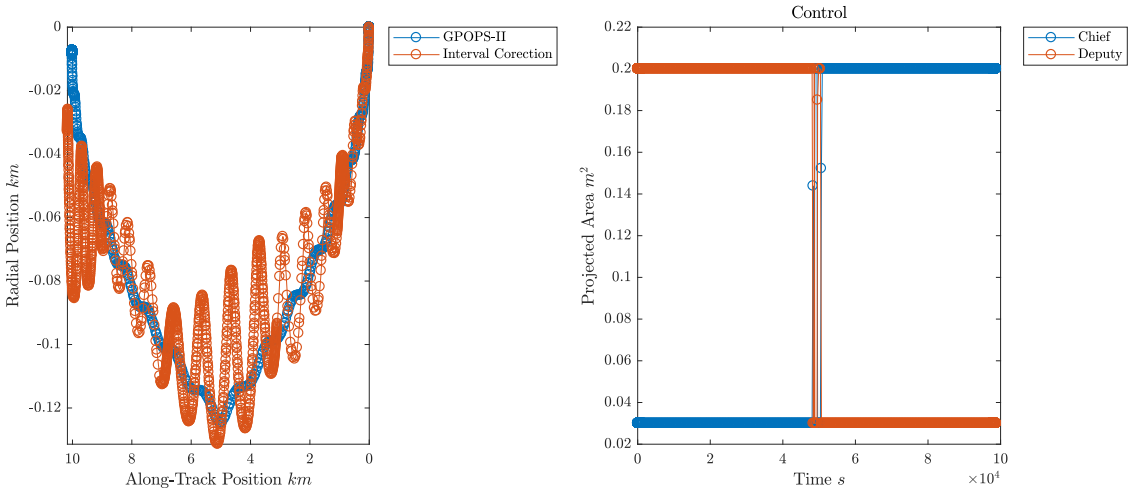

Figure 21. 10 km Maneuver with Interval Error Correction, every 20000 seconds

4.4 Initial Separation Control

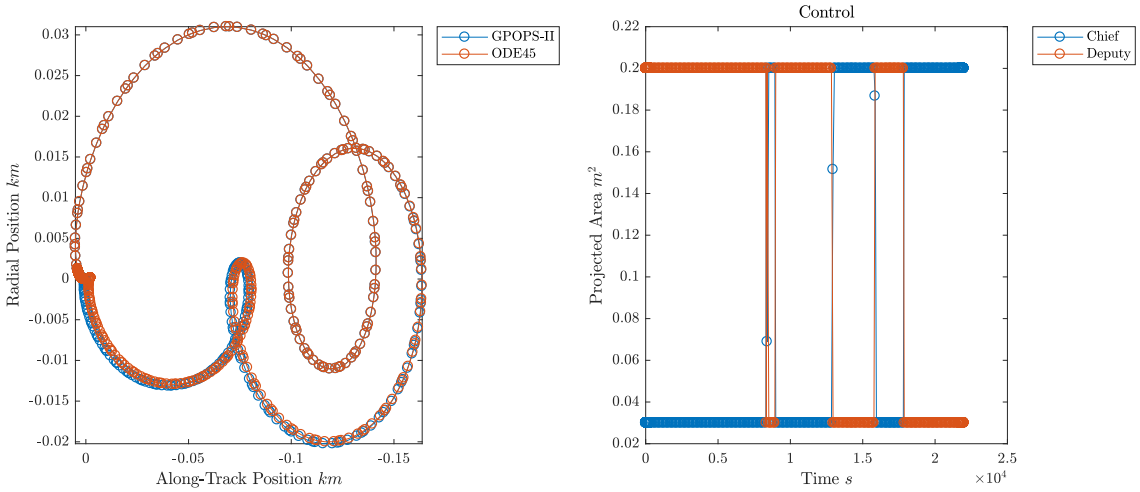

Figure 22. 1 cm/s Initial Separation Condition

Initial separation conditions were studied to evaluate the effects of deployer error on this model system and show the capability to control cases with non-zero velocity. These cases were run as maneuvers with an initial y direction separation of 10 cm and initial velocity in the positive y direction. Cases studying initial separation control produced results that were very similar to the formation resizing maneuvers. The only major difference in these cases is a long initial control phase to slow the relative motion of the two satellites. In slowing the motion it was found that the relative drift was only canceled out once the energy state of both satellites had been balanced. Once the drift was brought under control the motion and control history behaves as if it was a maneuver from the location where the drift was canceled. This shows that the initial separation control is difficult to control only in the same way large maneuvers are difficult to control.

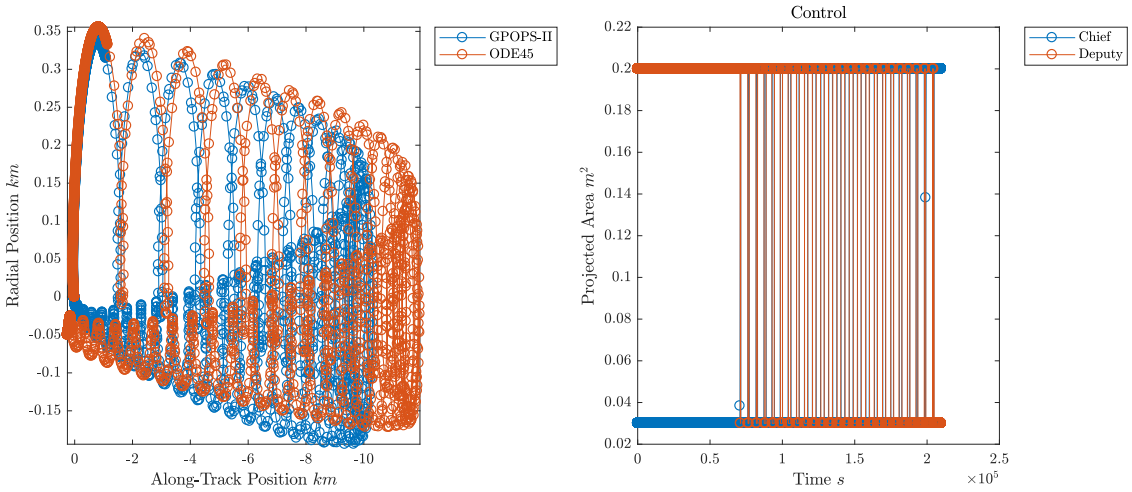

Figure 23. 10 cm/s Initial Separation Condition

The higher the initial separation velocity the more drift will be incurred and the longer the maneuver will take to complete. Similarly to the maneuver cases larger drift cases produce higher errors and become more computationally intensive to solve. To counter this approximated versions of the dynamics can be used to reduce computational demands and compute larger separation velocities, such as the case presented in Figure 23 which uses constant density and assumes a circular orbit.

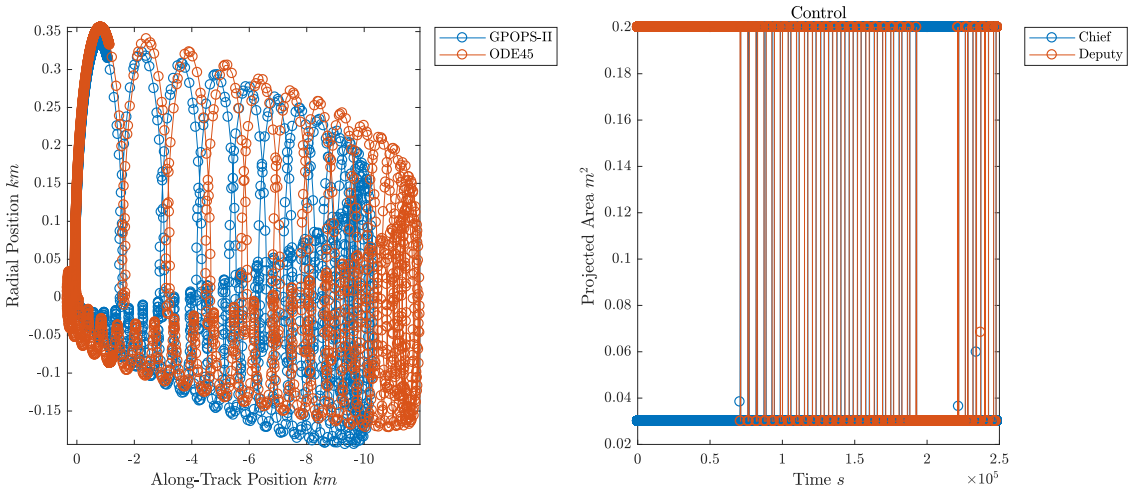

Figure 24. 10 cm/s Initial Separation Condition with an Error Checker Applied

Unfortunately, these assumptions introduce more error into the system as GPOPS-II is no longer solving the problem will all information, and therefore solving for an optimal solution to a similar but wrong problem. To counter this the error checkers can be applied to eliminate the error at the cost of optimality in maneuver final time. The error checker displayed in Figure 24 shows the final position corrector method.

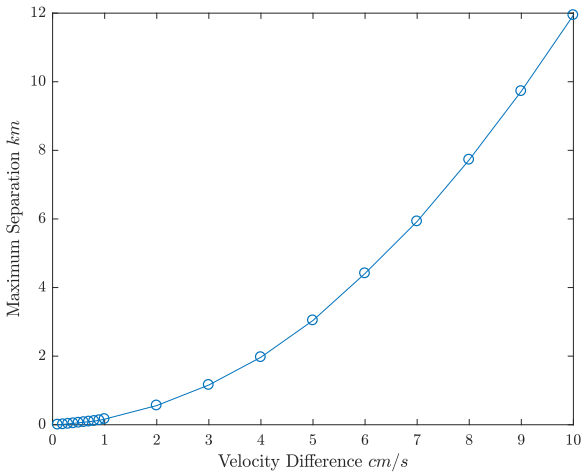

Figure 25. Initial Separation Velocity vs Maximum Separation During Maneuver

The initial separation problem is highly susceptible to differences in relative velocity. This is shown in Figure 25 which recored the maximum distance the deputy drifted from the chief when actively trying to counter the initial velocity difference. For these cases differences of 50 cm/s, approximately the worst case velocity difference due to the deployer expected for this mission, proved difficult to solve and drifted more than 100 km. Though it could not be run, at a 1 m/s initial separation a 910 km drift is expected under the same conditions as the presented data. This was calculated by propagating a deputy maximum drag profile to mimic the initial portion of control and balance energy.

4.5 Formation Keeping

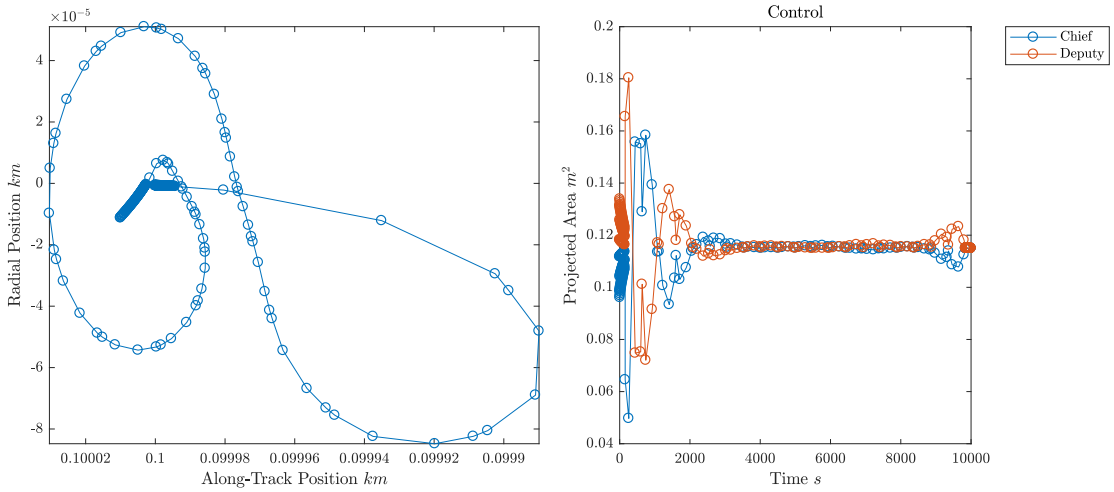

Figure 26. 100 m Formation Keeping for 10000s

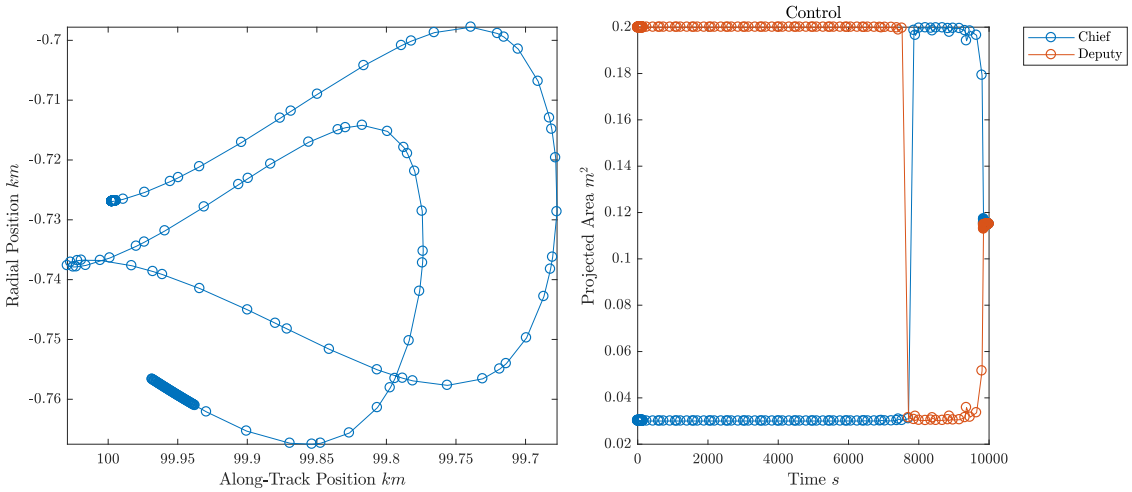

Figure 27. 100 km Formation Keeping for 10000s

Formation keeping was conducted to investigate how close the satellite system can stay to the equilibrium solution given J_2 and drag perturbations. Formations are all maintained on the near equilibrium path described in section 3.2.3 at the desired separation. As the desired formation separation increases the forces imparted by the J_2 perturbation increase. As a result of increased perturbation acceleration compared

to the maximum drag acceleration the formation looses its formation keeping effectiveness. This can be seen in Figures 26 and 27 where the larger formation location must use more control to maintain its position. As a result the formation's precision is reduced the larger the separation becomes. As shown in Figure 28 this distance grows to be 2.75 km at a 2000 km separation.

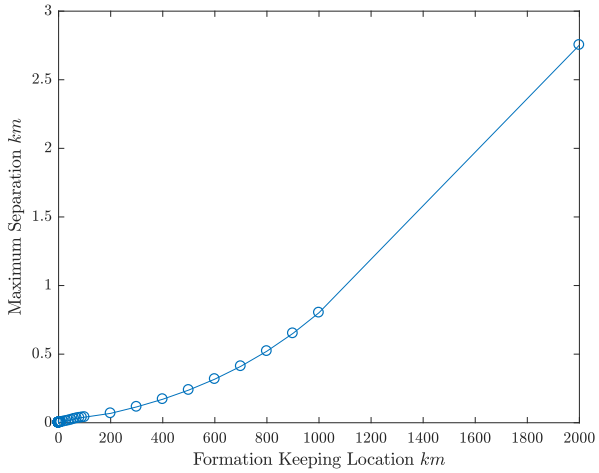

Figure 28. Formation Keeping Max Drift vs Location

4.6 Altitude Effects

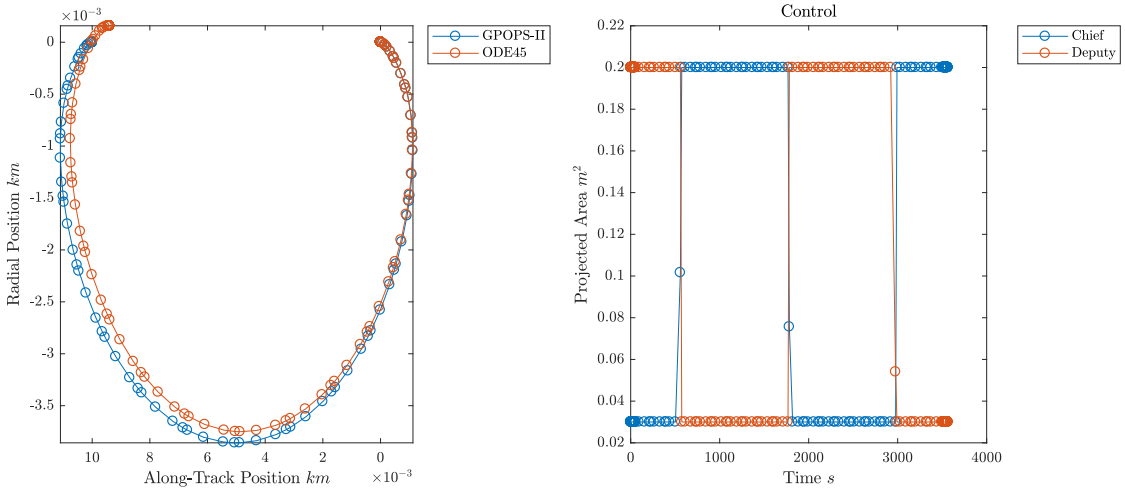

Figure 29. 10 m Maneuver at a 400km Altitude

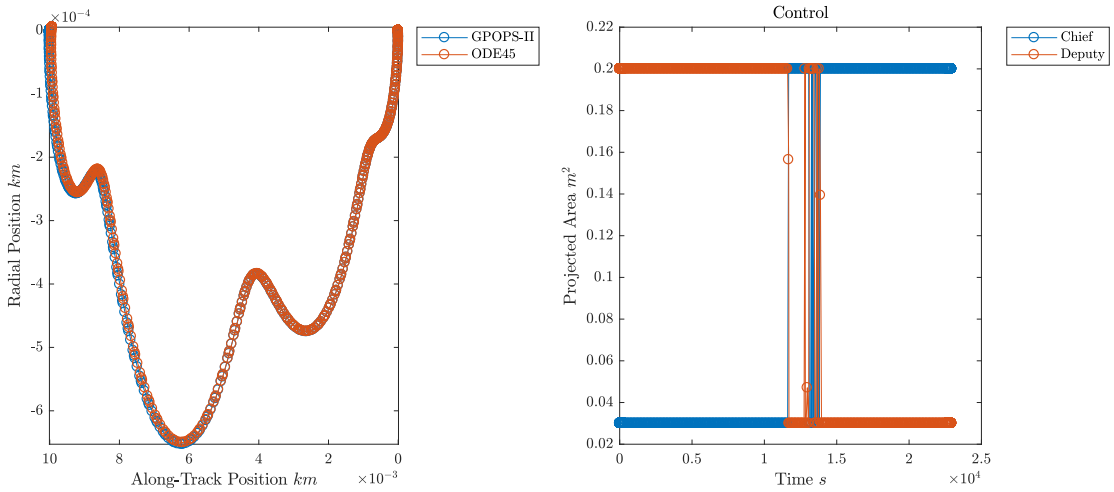

Figure 30. 10 m Maneuver at a 800km Altitude

As altitude increases the atmospheric density drops off exponentially. Because the differential drag control problem relies on this density for acceleration the problem's control authority also drops off exponentially. This is demonstrated in Figure 31; as control authority drops maneuver time increases due to the lower relative acceleration. This affect is also carried over to the relative trajectory. As control authority increases or decreases the relative trajectory of a maneuver will resemble the trajectory of its smaller or larger counterpart respectively. This means maneuver trajectory is not a function of maneuver size but rather the system's control authority, how much acceleration it can generate, in comparison to the maneuver.

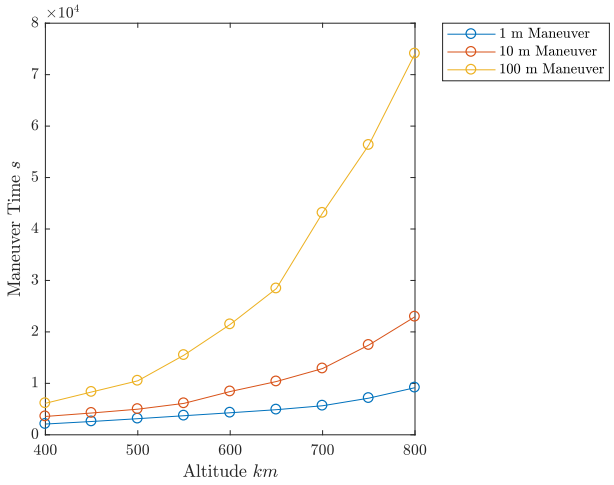

Figure 31. Maneuver Time vs Altitude

4.7 Solar Cycle Effects

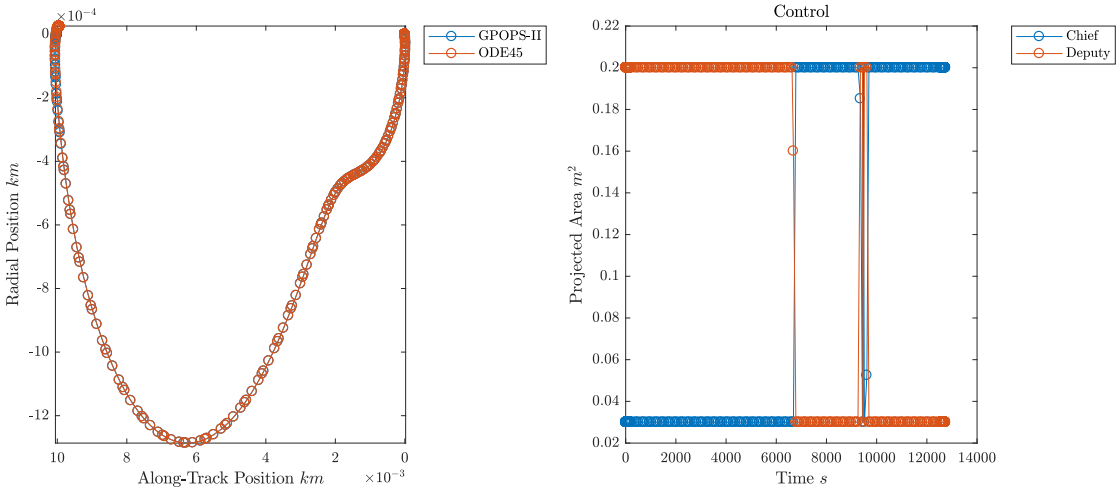

Figure 32. 10 m Maneuver at Solar Minimum

Besides altitude solar activity can also impact atmospheric density in LEO by several orders of magnitude. Changes in density due to solar activity produce the same general effects on the maneuver that changes in altitude have. However, since the magnitude of solar activity in not guaranteed, or predictable over the 11 year solar cycle, it can be difficult to properly take into account.

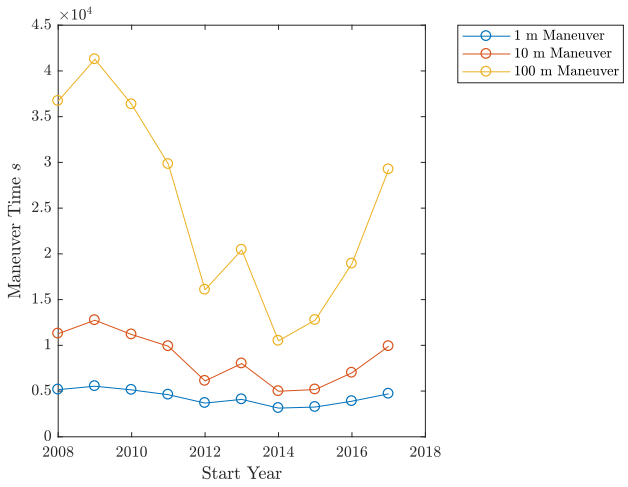

Figure 33. Maneuver Time vs Start Time

To accurately take this data into account an accurate atmospheric model, like the Jacchia-Roberts model, needs to be used in conjunction with an accurate prediction of solar weather. Failure to do so could easily lead to inaccuracies like those shown in Figures 15 and 16.

4.8 Implementing Limited Control

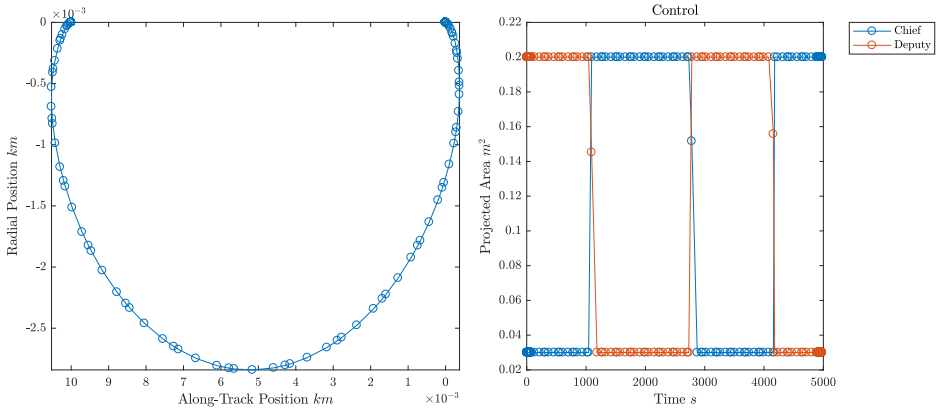

Figure 34. 10 m Maneuver without limits

Figures 34 and 35 depict a simulated maneuver in which the deputy achieved a separation distance of 10 meters using only differential drag control. The minimum-

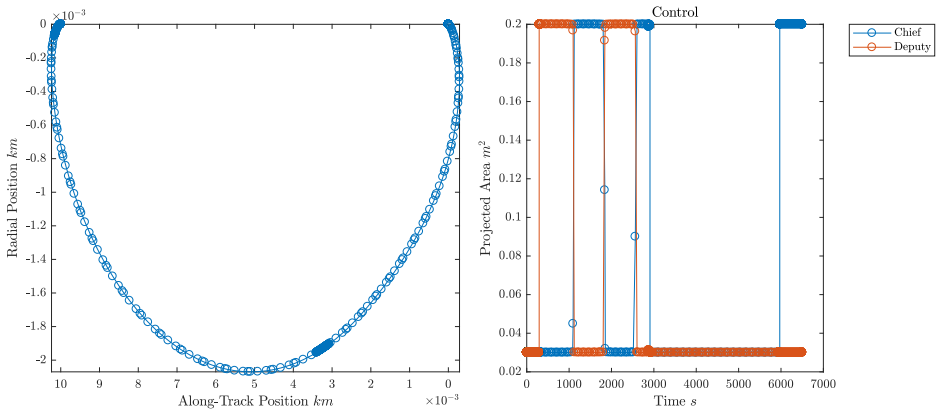

Figure 35. Eclipse Limited 10 m Maneuver

time solution allowing control throughout the orbit (Figure 34) required 4980 s, or 1.38 hr. However, restricting control maneuvers to eclipse periods only (Figure 35) required 6502 s, or 1.81 hr, a 31% increase. Figures 4 and 5 depict the same simulation executed for a 100 m maneuver. This time the continuous control case (Figure 36) required 10490 s, or 2.91 hr; and the eclipse limited case (Figure 37) required 17845 s, or 4.96 hr, a 70% increase. This indicates that as the maneuver size increases, the effects of limiting the control are more pronounced.

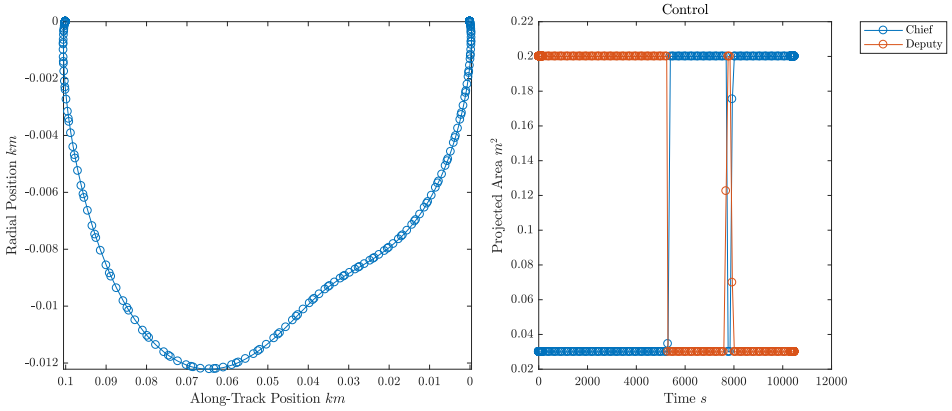

Figure 36. 100 m Maneuver without limits

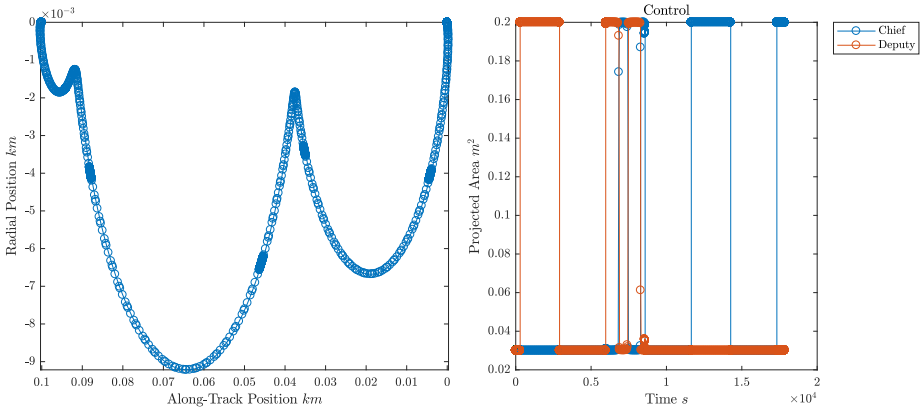

Figure 37. Eclipse Limited 100 m Maneuver

Furthermore all control histories resemble bang-bang control for all phases. This was expected from the continuous cases and proves the minimum number of phases was used. All control histories for both data sets were validated by propagating the control history using *ode45*. As expected, the optimal solution for the eclipse limited maneuvers is to allow drifting to occur during the day side of the maneuver allowing the satellite to build up speed and thereby reduce the maneuver time. Interestingly, the process of slowing the relative velocity takes longer than building it up. This likely results from the solver attempting not to overshoot the final destination, and having to wait for a final control phase to slow the solution down.

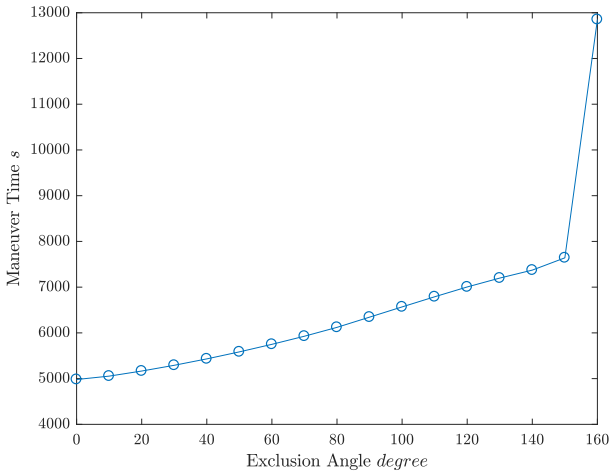

Figure 38. Limit Region Size vs 10 m Maneuver Time

Table 6. Data for Figure 38

Limit Angle	Time s	Control Phases
0°	4980	1
10°	5054	2
20°	5165	2
30°	5291	2
40°	5431	2
50°	5584	2
60°	5749	2
70°	5927	2
80°	6120	2
90°	6344	2
100°	6568	2
110°	6789	2
120°	7004	2
130°	7198	2
140°	7372	2
150°	7640	2
160°	12850	3

A data set was calculated that varied the control region's size, rather than use the eclipse region, to study the effects of decreasing the region where control was possible. This data is recorded in Figure 6 and Table 1 and used the same initial conditions as the first few runs. This data set mimicked previous runs in that all control histories are bang-bang control, the only difference being where the switching occurs. Here linear increases in the limit angle (the angle from the sun that defines where control cannot occur), decreasing the control region, resulted in a nearly linear increase in maneuver time for the generated data. This trend continues until the region becomes very limited where maneuver time increases much faster.

The nearly linear region is caused by drifting, where the satellite can build up relative acceleration and continue to move towards its destination due to that drift. In these cases reducing the control region only had the effect of slightly reducing

the maximum possible acceleration in each control phase. This slowed the maneuver down but affected the maneuver time but at a percentage less than the equivalent reduction in control region. A 50% reduction in the control region between 0° and 90° only increased maneuver time by 27%. The large jump in maneuver time at high limit angles occurs because the system no longer has enough control authority to maneuver the satellite in only two control phases, like in the 150° case, and had to include a third control phase. Between the 150° and 160° cases, a control region size reduction of 33%, the maneuver time increases by 68%. These jumps in maneuver time from requiring another control phase may also explain why the 100 m maneuver had a larger increase in maneuver time; it took 4 control regions rather than the two, an in the 10 m case. It is expected that further limiting the control region would reach a point where the maneuver could no longer be achieved. This would occur only when perturbations on the system cause a greater change in velocity during the limited regions than the system can produce in the control regions. Perturbations in this case include drag, since it isn't being controlled, and other un-modeled forces to include higher orders of Earth's oblateness, third body effects, and solar wind. At this point the drift would be negated or unpredictable and the system would become uncontrollable. This point could not be reached in a hypothetical system without any perturbations in the limited regions, though as the control region size approached zero maneuver time would approach infinity.

4.9 Summary

Results comparing maneuver size, dynamics being used, error checker methods, initial separation velocity, atmospheric conditions, and limited region size were generated. These results help characterize the responsiveness and capabilities of this system, provide insight into the capabilities of the mission planing tool, and provide

data pertaining to the research questions. From this data we can begin to draw conclusions from the research.

V. Conclusions and Recommendations

5.1 Review

This thesis developed a mission planning tool capable of providing accurate maneuvers for various mission parameters. Though designed with a single mission in mind it is capable of operating at several altitudes, mission orbits, start times, and control weights, as well as performing maneuvers of various sizes. In developing this tool it studied the effects of limited-duty-cycle differential drag control, the effects of operating at different altitudes, operating in solar minimum vs solar maximum, and the effects on accuracy of using the Jacchia-Roberts atmospheric model and accounting for the J_2 perturbation. Despite this, the tool does have limitations.

5.2 Tool Evaluation

GPOPS-II is a poor fit for this problem. There is a fundamental limitation in the amount of data GPOPS can handle as a MatLab based function and a limit on how fast it can complete mesh iterations. This is driven by IPOPT and SNOPT, where when large sets of data are required by the problem they slow down significantly as they begin to run into problems with the maximum alloted memory. This may stem from a MatLab design limitation where the program has difficulty handling very large sets of data or some other source. It is possible these settings could be fixed by setting parameters for SNOPT and IPOPT; but these cannot be set through the GPOPS-II interface. As this problem requires meter level accuracy in propagation for maneuvers on the scale of 1000 km, it fundamentally requires a lot of data. Future research on this problem may be better suited to a different solver tool or program environment, as the current one limits the operational envelope where the solver can provide accurate timely results. Maneuver sizes presented in this work are limited to

7 km because maneuvers larger than this size could not be calculated in a reasonable amount of time. A 7 km maneuver only takes 14.8 minutes to solve on a computer with a 4.01 GHz processor and 32 GB of RAM, but the computation time for manuvers 8 km and larger was prohibitive. These solutions could only be calculated using simplified dynamics, such as the results presented in the error checker methods. But as discussed, these are either inaccurate or a less than optimal solution. This means that the developed tool in its current state has a very small operating window. This window is defined by the control authority compared to maneuver size; as control authority decreases and maneuver size increases, the borders of the operating envelope are reached. Large separations, high altitudes, and large relative initial velocities all can cause computational problems. This is compounded if the satellite has high mass, low surface area, or a low C_d; is operating in solar minimum; or must meet limited control maneuvers.

5.3 Potential Future Research

A potential solution alternative is for the tool to be rewritten to operate using a different pseudospectral solver in a different coding language. This would help alleviate the run time problem encountered with the current setup; and pseudospectral methods have been shown to work with much more efficiency in different coding languages. There is no reason the current method should not be able to calculate up to 2000 km maneuvers other than the computational difficulties involved with requesting several thousand data points from the NLP solver. While this would fix many problems and allow much faster development once completed, it would require all of the code to be rebuilt. In doing so, problem dynamics, cost functions, phase boundaries (keep-out region logic), and initial guess calculation could be maintained. But differences in variable input would have to be accounted for and problem scaling

may have to be developed, as the current method relies on the GPOPS-II automatic scaling of the problem.

The problem could be further developed by applying differential lift dynamics to the tool. Besides providing more control authority, it could enable it to solve the cross-track problem. This would entail developing a model for spacecraft material properties, modifying control to use two angles per satellite (rather than cross-sectional area), and redefining initial conditions, final conditions, and the initial guess. The only similarity would be that the dynamics are already built. This process would take a while and then the effectiveness of differential lift would have to be analyzed as it may prove to be negligible to the overall system. For this problem a brief analysis shows lift forces that are an order of magnitude less than the drag forces. While small, this force still has the potential to improve system control authority. This development represents an additional research question to be developed at some later date.

This tool could also be improved to use more than two satellites, that may not be identical, in a formation where a circular orbit may not be desired. These are three assumptions present in the current work that could easily change for future missions. No changes to the fundamental dynamics would be required but keeping track of multiple satellites would require propagating multiple relative satellites; therefore adding six additional states per satellite. The initial and final conditions would also need to change as they are currently hard coded to assume a circular orbit at the desired final location. Enabling these kinds of features in the tool would allow its continued use beyond the current mission.

5.4 Conclusion

It is important for program mission planning to understand the effects various mission conditions have on the functionality of differential-drag control. This study offers a glimpse into the effects of mission limitations on maneuver time and on control authority. First, as maneuver sizes and mission limitations vary, the duration of mission maneuvers can vary drastically. Maneuver time increases could be relatively low if only a few number of control phases are required for limited maneuvers. Or maneuver time could jump dramatically if the control authority (maximum possible change to velocity) from an individual control phase becomes insufficient, requiring more phases to be used and waiting for those phases to occur. Second, due to limited region drift, system control authority is not lost until the control regions become very small. It is important to note that phase control authority is relative to the maneuver size and atmospheric conditions. This means limited regions have different effects on maneuvers of different sizes, done at different altitudes, and with different start times (solar cycle effects). As any one of these metrics reduces control authority of the system, finding an optimal solution becomes much more computationally intensive and the optimal maneuver takes much more time to complete. Finally, once the desired separation becomes too large a stable formation can no longer be maintained due to long period J_2 drift. While a novel concept, the effects of limiting the differential drag control to only limited times throughout the orbit is an important problem for CubeSats to consider and overcome.

Appendix A. Data Tables

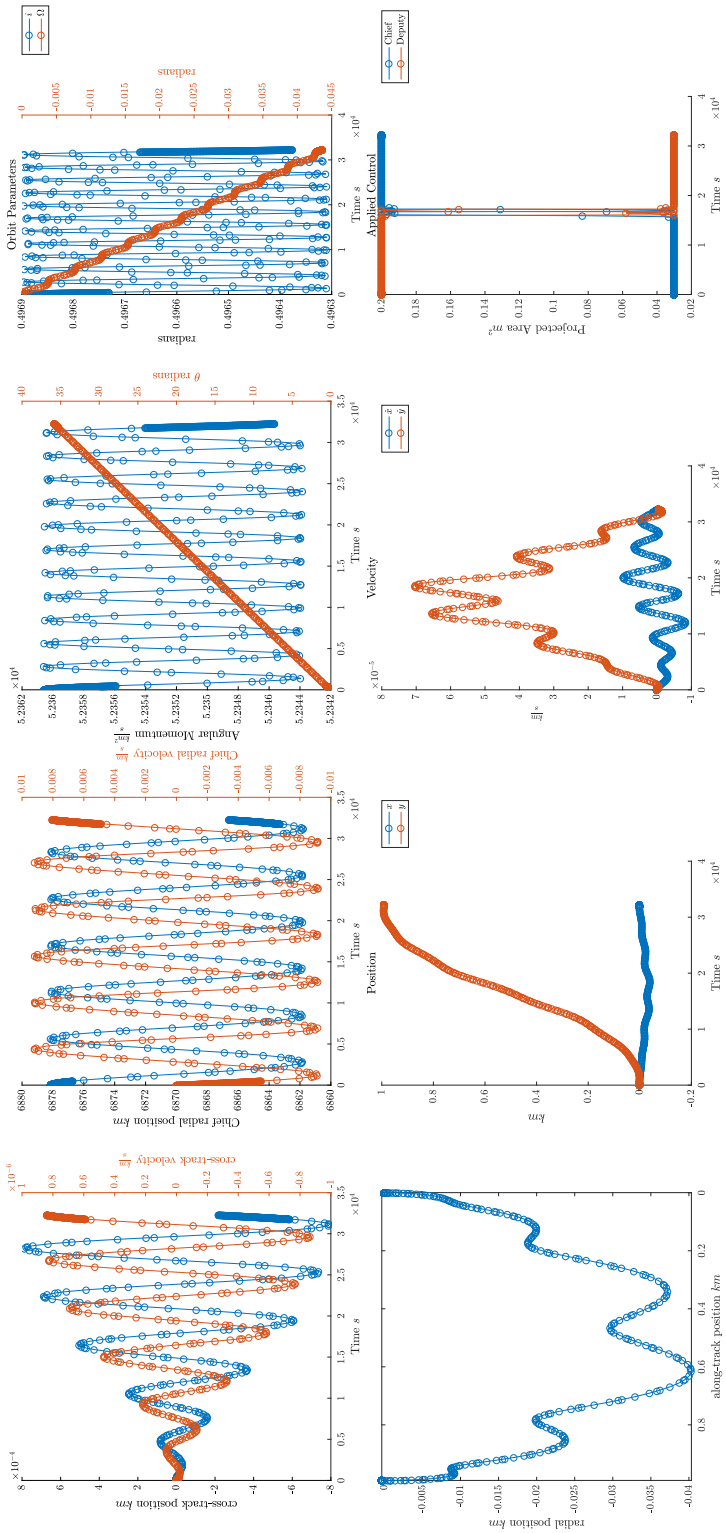

Figure 39. Sample Solution Algorithm Output

Bibliography

[1] J. T. Betts, "Optimal low thrust orbit transfers with eclipsing," *Optimal Control Applications and Methods*, vol. 36, no. 2, pp. 218–240, 2015.

[2] B. S. Kumar, K. Yoshihara, and A. D. Ruiter, "Differential drag as a means of spacecraft formation control," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 2, pp. 1125–1135, 2011.

[3] C. L. Leonard, W. M. Hollister, and E. V. Bergmann, "Orbital formationkeeping with differential drag.," *Journal of Guidance, Control, and Dynamics*, vol. 12, no. 1, pp. 108–113, 1989.

[4] H. Cho and G. Kerschen, "Satellite formation control using continuous adaptive sliding mode controller," AIAA SPACE Forum, (Long Beach, California), AIAA/AAS Astrodynamics Specialist Conference, 2016.

[5] R. Bevilacqua and M. Romano, "Rendezvous maneuvers of multiple spacecraft using differential drag under j2 perturbation," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 6, pp. 1595–1607, 2008.

[6] M. W. Harris, A. W. B. Jr., and B. Açikmeşez, "Rendezvous using differential drag and feedback control," Guidance, Navigation, and Control and Co-located Conferences, (Boston, MA), AIAA Guidance, Navigation, and Control (GNC) Conference, 2013.

[7] L. Dell'Elce and G. Kerschen, "Comparison between analytical and optimal control techniques in the differential drag based rendez-vous," Master's thesis, Chemin des Chevreuils, Liége, Belgium, 2013.

[8] 2nd Lt Blake B. Hajovsky, "Satellite formation control using atmospheric drag," Master's thesis, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, 2007.

[9] J. T. Wedekind, "Characterizing and controlling the effects of differential drag on satellite formations," Master's thesis, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, 2006.

[10] M. A. Patterson and A. 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 Transactions on Mathematical Software*, vol. 41, no. 1, pp. 1–37, 2014.

[11] D. Wang, B. Wu, and E. K. Poh, *Spacecraft Formation Flying Relative Dynamics, Formation Design, Fuel Optimization Maneuvers and Formation Maintenance.* Intelligent Systems, Control and Automation: Science and Engineering 87, Singapore, China: Springer Science+Business Media Singapore, 2017.

[12] S. R. Omar and D. J. Wersinger, "Satellite formation control using differential drag (aiaa 2015-0002)," 53rd AIAA Aerospace Sciences Meeting, (Kissimmee, Florida), AIAA SciTech Forum, 2015.

[13] S. Casotto, "The equations of relative motion in the orbital reference frame," Springer Science+Business Media, 2015.

[14] M. Bakhtiari, K. Daneshjou, and M. Fakoor, "Relative hovering analysis about an elliptical perturbed orbit with consideration of dynamic air drag and oblate earth," *Aerospace Science and Technology*, pp. 286–299, 2017.

[15] D. A. Vallado, *Fundamentals of Astrodynamics and Applications*. Space Technology Library, Hawthorne, CA: Springer, 2007.

[16] F. L. Markley and J. L. Crassidis, *Fundamentals of Spacecraft Attitude Determination and Control*. Space Technology Library, Greenbelt, MD: Springer, 2014.

[17] D. L. Kunz, *Intermidiate Dynamics for Aeronautics & Astronautics*. Centerville, OH: Headmaster Press, 2015.

[18] A. C. Long, J. O. Cappellari, C. E. Velez, and A. J. Fuchs, "Goddard trajectory determination system (gtds) mathematical theory revision 1," National Aeronautics and Space Administration/ Goddard Space Flight Center, 1989.

[19] C. E. Roberts, "An analytic model for upper atmosphere densities based upon jacchia's 1970 models," (Patrick AFB, FL), Pan American World Airways, Aerospace Services Division, 1971.

[20] C. R. Hall, *Spacecraft Attitude Dynamics and Control*. Virginia Tech, 2003.

[21] K. T. Alfriend, S. R. Vadali, P. Gurfil, J. P. How, and L. S. Breger, *Spacecraft Formation Flying Dynamics, Control and Navigation*. Elsevier Astrodynamics Series, Burlington, MA: Elsevier, 2010.

[22] M. Horsley, S. Nikolaev, and A. Pertica, "Small satellite rendezvous using differential lift and drag," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 2, pp. 445–453, 2013.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
08–03–2019	Master's Thesis	Aug 2017 – Mar 2019

4. TITLE AND SUBTITLE

LIMITED-DUTY-CYCLE SATELLITE FORMATION CONTROL VIA DIFFERENTIAL DRAG

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

2nd Lt Talon A. Townley, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/ENY)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENY-MS-19-M-249

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Dan T. Frey
SPAWAR SSC Pacific SIS Division
53560 Hull St.
San Diego,CA,92152
312-553-1445, Daniel.Frey@navy.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

13. SUPPLEMENTARY NOTES

14. ABSTRACT As CubeSat formation flying missions relying on differential drag become common additional mission requirements must be considered. Mission architectures may require maintaining large formations and performing maneuvers where linear dynamics can no longer be applied accurately. It may also be desirable to limit differential-drag maneuvers to specific regions of the orbit due to mission architecture limitations. This thesis studies the effects of these considerations on maneuver time and system performance in the presence of J_2 and drag perturbations. This model leader-follower CubeSat formation is subject to non-linear relative dynamics for various maneuver sizes and altitudes. Optimal control is applied using a pseudo-spectral numerical solver, GPOPS-II, to minimize maneuver time. This develops a mission planning tool to work with the modeled mission to calculate optimal maneuvers. The effects of mission altitude, solar cycle, various maneuver sizes and formations, limited control, various computational methods, and error checkers were evaluated. Thought the mission planning tool developed can accommodate the stated parameters it suffers from computational complexity and is poorly suited to the tools used and the MatLab environment.

15. SUBJECT TERMS

Satellite formations, differential drag, formation control, eclipse only maneuvers, mission planning tool

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Lt Col Kirk W. Johnson
U	U	U	UU	89	19b. TELEPHONE NUMBER *(include area code)* (937) 255–3636 ext 4285; kirk.johnson@afit.edu