Multi-phase Spacecraft Proximity Operations Using Model Predictive Control

THESIS

Julia C. Bell, Capt, USSF

AFIT-ENY-MS-21-J-094

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A. APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION IS UNLIMITED.

The views expressed in this document 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 material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT-ENY-MS-21-J-094

MULTI-PHASE SPACECRAFT PROXIMITY OPERATIONS USING MODEL PREDICTIVE CONTROL

THESIS

Presented to the Faculty

Department of Aeronautics & 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 Astronautical Engineering

Julia C. Bell, BS

Capt, USSF

June 17, 2021

DISTRIBUTION STATEMENT A. APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION IS UNLIMITED.

AFIT-ENY-MS-21-J-094

MULTI-PHASE SPACECRAFT PROXIMITY OPERATIONS USING MODEL PREDICTIVE CONTROL

THESIS

Julia C. Bell, BS
Capt, USSF

Committee Membership:

Maj Costantinos Zagaris, PhD
Chair

Lt Col Bryan D. Little, PhD
Member

Maj Joshuah A, Hess, PhD
Member

Adedeji B. Badiru, PhD
Dean, Graduate School of Engineering and Management

AFIT-ENY-MS-21-J-094

Abstract

The evolving space environment has created a demand for autonomous spacecraft that can maneuver in complex and sometimes contentious environments. Constraint enforcement, such as an avoidance zone to prevent collision with a target, is a key component of autonomous control to ensure safety and performance requirements are met. Finite-horizon Model Predictive Control (MPC) is a popular control method due to its improved computation time while optimizing performance. Two areas of MPC in need of expansion are time-varying constraints and phase transitions in multi-phase applications. In this work, MPC is employed to track the reference trajectory of a multi-phase satellite inspection mission. During certain phases, conical and spherical constraints are activated to represent a sun vector keep-in-zone (KIZ) and a collision avoidance keep-out-zone (KOZ), respectively. The equations of motion follow the linearized Hill-Clohessey-Wiltshire (HCW) equations. The problem is solved to minimize control effort and tracking error using MPCTools, a control and estimation tool for dynamic models. Results are compared to the validated results from literature and found to be comparable in costs while lending flexibility to phase design.

Acknowledgements

First, I would like to thank my committee: Lt Col Little, Maj Zagaris, and Maj Hess. The exceptional time, effort, and passion you put in to your research and teaching have positively impacted me as an officer and an engineer. This sentiment extends to my academic advisor and one of my first mentors at AFIT, Maj Bettinger. Thank you all for your encouragement and always making time to help me.

Next, I would like to thank my husband, who worked so hard to take care of us so that I could focus on school. You are an amazing dad and husband. Thank you also to my son for (mostly) patiently enduring this crazy time; it has been so fun watching you grow from a baby to a toddler. Your smiles, babbles, and hugs are a constant source of joy and motivation.

To my family, friends, and classmates: thank you for your advice, reassurance, feedback, and more. I'm especially grateful to everyone who joined in on our late-night study groups to offer help or solidarity. Also, a special thank you to my mother-in-law, who traveled often to help us with the baby, house, and meals.

Finally, I thank and praise God for daily guiding and sustaining me. "For I am convinced that neither death nor life, neither angels nor demons, neither the present nor the future, nor any powers, neither height nor depth, nor anything else in all creation, will be able to separate us from the love of God that is in Christ Jesus our Lord." Romans 8:38-39

Julia C. Bell

Table of Contents

Page

Abstract iv

Acknowledgements v

List of Figures viii

List of Tables x

List of Abbreviations xi

List of Symbols xiii

I. Introduction 1

1.1 Motivation and Background 1
1.2 Problem Statement 3
1.3 Research Questions 3
1.4 Organization of the Thesis 4

II. Literature Review 6

2.1 Rendezvous and Proximity Operations 6
2.1.1 Reference Frame 7
2.1.2 Hill-Clohessey Wiltshire Equations 8
2.1.3 RPO Formations 11
2.2 Model Predictive Control 16
2.2.1 Background 16
2.2.2 Formulation 17
2.2.3 Solution Methods 20
2.3 Multi-Phase Operations 20
2.4 Summary 21

III. Methodology 23

3.1 Dynamics and Relative Motion 23
3.2 Constraints and Bounds 25
3.2.1 Path Constraints 25
3.2.2 Bounds 27
3.3 Optimal Control Problem 28
3.4 MPC Solution 28
3.5 Phase Transition 29
3.6 Summary 31

Page

IV. Results and Analysis 32

4.1 Test Instances 32
4.1.1 Chief and Deputy Parameters 32
4.1.2 Constraints 35
4.1.3 Controller 35
4.1.4 Demonstration of Slack Variables 37
4.2 Results 39
4.2.1 Trajectories 39
4.2.2 Control Effort 50
4.2.3 Computation Costs 56
4.3 Comparison with Literature 57
4.3.1 Original Problem 57
4.3.2 Phase Transition Implementation 59
4.3.3 Comparison Results 61
4.4 Summary 64

V. Conclusions and Recommendations 65

5.1 Conclusions 65
5.2 Recommendations 66
Bibliography 68

List of Figures

Figure		Page
1	LVLH Reference Frame	7
2	Relative Orbit Transfer Using Natural Motion Circumnavigation	14
3	Lovell's Teardrop Formulation	15
4	MPC Algorithm Flow	19
5	Illustration of Constraint Definitions	26
6	Single-Phase Maneuver with Two Slack Variables	37
7	Single-Phase Maneuver with One Slack Variable	37
8	Trajectory (LEO,Phase 1)	40
9	Constraint Enforcement (LEO,Phase 1)	40
10	Trajectory (LEO,Phase 2)	41
11	Constraint Enforcement (LEO,Phase 2)	41
12	Planar View (LEO, Phase 2)	42
13	Trajectory (LEO,Phase 3)	43
14	Constraint Enforcement (LEO,Phase 3)	43
15	Trajectory (LEO,Overall)	44
16	Trajectory (GEO,Overall)	45
17	Trajectory (GEO, Phase 1)	46
18	Constraint Enforcement (GEO,Phase 1)	46
19	Trajectory (GEO,Phase 2)	47
20	Constraint Enforcement (GEO,Phase 2)	47
21	Planar View (GEO, Phase 2)	48
22	Trajectory (GEO,Phase 3)	49

Figure Page

23 Constraint Enforcement (GEO,Phase 3) 49

24 LEO Control Effort 50

25 GEO Control Effort 51

26 LEO Orbit with Slack Weighted 10,000x More than Costs 54

27 LEO Orbit with Slack Weighted 10,000x More than Costs 55

28 GEO Case Slack Weighting Comparison 56

29 Becker Constraint Definitions 58

30 Becker Phase Definitions 58

31 Trajectory Comparison 61

List of Tables

Table		Page
3	Inspecton Phases	25
4	Chief Orbital Elements	33
5	Initial and Final Conditions for Each Phase, LEO	34
6	Initial and Final Conditions for Each Phase, GEO	34
7	Constraints	35
8	MPC Controller Settings	36
9	Constraint Definition for Each Phase	38
10	Control Effort Comparisons	52
11	Control Effort, Computation Time Comparisons for Slack Weights	53
12	Computation Times	57
13	Soft Constraint Assignments by Phase	60
14	Control Effort for Hard vs Soft Constraints	62
15	Computation Times for Hard vs Soft Constraints	63

List of Abbreviations

Abbreviation		Page
RPO	Rendezvous and Proximity Operations	1
MPC	Model Predictive Control	1
ESA	European Space Agency	1
USSF	United States Space Force	2
DARPA	Defense Advanced Research Projects Agency	2
DART	Demonstrations for Autonomous Rendezvous Technology	2
OCP	Optimal Control Problem	3
FF	Formation Flying	6
GNC	Guidance, Navigation, and Control	6
LVLH	Local-Vertical, Local-Horizontal	7
DCM	Direction Cosine Matrix	8
NERMs	Nonlinear Equations of Relative Motion	9
ODEs	Ordinary Differential Equations	9
HCW	Hill-Clohessey-Wiltshire	10
CW	Clohessey-Wiltshire	10
6-DoF	Six Degree of Freedom	10
DMPC	Decoupled MPC	11
NMC	Natural Motion Circumnavigation	11
ROE	Relative Orbital Elements	11
STM	State Transition Matrix	11
KOZ	Keep-Out-Zone	16
SOCP	Second-Order Cone Programming	17

Abbreviation		Page
NMPC	Nonlinear MPC	20
NLP	Non-Linear Programming	20
COE	Classical Orbital Elements	23
LTI	Linear Time-Invariant	24
KIZ	Keep-in-Zone	25
LOS	Line-of-Sight	27
IPOPT	Interior Point Optimizer	28
GPOPS-II	General Purpose OPtimal Control	28
CasADi	Computer Algebra System + Algorithmic Differentiation	29
LEO	Low Earth Orbit	32
GEO	Geostationary Orbit	32
ISS	International Space Station	32
FOV	Field-of-View	58
LOS	Line-of-Sight	58
OCP	Optimal Control Problem	65

List of Symbols

Variables

A	Continuous Plant Matrix
A_k	Discrete Plant Matrix
a	Semimajor Axis
B	Continuous Input Matrix
B_k	Discrete Input Matrix
d	Distance from Chief
e	Eccentricity
f	True Anomaly
$\mathcal{I}$	Inertial Frame
i	Inclination
J	Total Optimization Cost
K	Feedback Control Law
k	Discrete Time Step
k_Q	Terminal Cost Weighting
k_{Q+R}	Stage Cost Weighting
k_s	Slack Variable Weighting
$\mathcal{L}$	Hill Frame
$\mathscr{L}$	Laplace Transform
R	Direction Cosine Matrix
l	Stage Cost
N	Prediction Horizon
N_{sim}	Simulation Number
n	Mean Motion
Q	State Gain Matrix

Q_f	Terminal State Gain Matrix
R	Control Gain Matrix
r	Keep-Out-Zone Radius
$\mathbf{r}$	Relative Position State Vector
r_c	Chief Distance from Earth
$\vec{r}_c$	Chief Inertial Position Vector
$\dot{\vec{r}}_c$	Chief Velocity Vector
$\vec{r}_d$	Deputy Inertial Position Vector
$\vec{r}_0$	Initial Position Vector
$\vec{r}_f$	Final Position Vector
s	Slack Variable
T	Orbital Period
t	Current Time
t_0	Initial Time
U_k	Discretized Optimal Control Solution
$\mathbf{u}$	State Space Control Matrix
$\mathbf{u_k}$	Discretized State Space Control Matrix
$\vec{u}$	Control Vector
u_x	Control in x
u_y	Control in y
u_z	Control in z
V_f	Terminal Cost
$\mathbf{v}$	Relative Velocity State Vector
$\vec{v}_0$	Initial Velocity Vector
$\hat{v}_a$	Position Unit Vector in Constraint Zone
$\hat{v}_s$	Sun-Pointing Unit Vector

$\mathbf{x}$	State Vector
x	Relative Position in x
y	Relative Position in y
z	Relative Position in z
$\dot{x}$	Relative Velocity in x
$\dot{y}$	Relative Velocity in y
$\dot{z}$	Relative Velocity in z
$\ddot{x}$	Relative Acceleration in x
$\ddot{y}$	Relative Acceleration in y
$\ddot{z}$	Relative Acceleration in z
$\dot{\mathbf{x}}$	Derivative of State Vector
$\mathbf{x}_k$	Discretized State Vector at Step k
$\mathbf{x}_k$	Desired State Vector
$\hat{x}$	Radial Unit Vector
$\hat{y}$	In-Track Unit Vector
$\hat{z}$	Cross-Track Unit Vector

Symbols and Greek Letters

Δv	Delta-v
θ	Cone Half-Angle
μ	Earth's Gravitational Parameter
$\vec{\rho}$	Relative Position Vector
$\dot{\vec{\rho}}$	Relative Velocity Vector
$\ddot{\vec{\rho}}$	Relative Acceleration Vector
$\mathbf{\Phi}$	State Transition Matrix
Ω	Right Ascension of the Ascending Node

$\vec{\omega}$ Angular Velocity Vector

ω Argument of Periapsis

MULTI-PHASE SPACECRAFT PROXIMITY OPERATIONS USING MODEL PREDICTIVE CONTROL

I. Introduction

Control methods for spacecraft operations have been heavily studied and refined throughout the years. As reliance on space continues to grow for government and commercial applications, so does the need for more versatile, efficient, and cost-effective satellites. Innovations in control have enabled sophisticated space missions such as servicing, inspecting, docking, collecting debris, and more. Autonomous control methods are especially desirable in Rendezvous and Proximity Operations (RPO) where satellites operate at close relative distances. In time-sensitive situations, such as collision avoidance, instantaneous control feedback is often advantageous over external, operator-driven inputs [1]. Of the many autonomous control algorithms that have been developed, Model Predictive Control (MPC) shows great promise for its robustness to constraint enforcement and optimization. Moreover, certain MPC methods are easily adaptable to on-board systems due to their computational efficiency.

1.1 Motivation and Background

The space environment is becoming increasingly crowded and contested due to worldwide competing scientific, commercial, and military interests. The European Space Agency (ESA) Space Debris Office [2] predicts that orbiting Earth are more than 34,000 objects greater than 10 cm, and 900,000 objects from 1 mm to 1 cm. The office also estimates that over 560 significant fragmentation events are estimated to have occurred since 1957. These include break-ups, explosions, collisions, and more.

The United States has long emphasized the importance of maintaining dominance and security in space [3, 4]. On December 20, 2019, the National Defense Authorization Act for Fiscal Year 2020 was signed into law, creating the United States Space Force (USSF) [5]. The new military branch has re-energized the space race and accelerated the need to innovate solutions that are cheaper and more versatile than before. USSF doctrine cites Space Mobility as one of five core competencies of the new service [6]. This is just one of many areas that benefit from innovations in satellite design.

As RPO missions become more sophisticated, satellites require control systems that are capable of obstacle avoidance and complex navigation in real time. Several missions such as the Defense Advanced Research Projects Agency (DARPA) Phoenix Spacecraft servicing program have utilized autonomy to varying degrees [7]. Some missions have failed, such as NASA's infamous Demonstrations for Autonomous Rendezvous Technology (DART). DART collided with its target on approach, underscoring the need for built-in safety. Application and study of nonlinear control techniques could prevent such incidents in the future, as well as build confidence - especially in autonomous control - throughout the industry[8].

Autonomy requires increased on-board computing power, which has limitations based upon the satellite's size, cost, redundancy level, and more. Model Predictive Control (MPC) is a constrained optimization, closed-loop control approach with the advantage of increased robustness and decreased computational complexity [9]. MPC is desirable in aerospace applications for its ability to enforce constraints while optimizing performance [10]. While infinite-horizon MPC methods exist, the focus of this work will be on finite-horizon MPC, as limiting the prediction horizon is considered to be a great contributor to improved computational efficiency [11].

1.2 Problem Statement

This thesis examines a multi-phase RPO mission requiring an efficient control algorithm that is robust to time-varying path constraints while maintaining stability and minimizing control effort. Previous MPC literature has examined constraint enforcement and multi-phase scenarios [1, 12, 13, 14]. However, there have been few studies on time-varying constraints and different phase transition methods. Prior multi-phase missions have relied primarily on hard entry and exit criteria between phases, leading to sudden changes in control and costly state transitions. MPC's use of a finite prediction horizon saves on computation costs. However, that finite horizon can also limit the ability to predict boundary and path constraints. This work aims to fill this gap by presenting a phase transition strategy within the MPC framework.

1.3 Research Questions

This thesis is driven by three primary research questions:

1. How can an autonomous guidance scheme be applied to a time-varying conical constraint?

2. How can MPC be used to minimize control effort when constraints are activated in multi-phase maneuvers?

3. How does MPC compare to the infinite horizon Optimal Control Problem (OCP)?

Question one will be investigated through the process of defining constraints and implementing them into the MPC framework. The time-varying characteristic of the constraints adds computational complexity, especially when combined with multi-phase missions. Additionally, linearity and convexity of the constraints are important considerations for on-board implementation.

Question two will be answered by designing and refining a phase transition that can address constraint activation or deactivation. To visualize this concept, consider a car approaching a traffic light. The changing light represents the concept of constraint activation, as it creates a change to the car's options for its future trajectory. If the light is green, the car can continue. For a yellow light, the car must slow down and prepare to stop. If the light is red, the car must stop in order to avoid colliding with traffic or pedestrians. But what if there were no yellow light? A driver might have to slam on the brakes if the light changed suddenly from green to red. Alternately, the driver could speed up when approaching every green light in the hopes it can get through before the light turns. Sudden stopping or accelerating costs a driver fuel and risks a collision. Applying this analogy to satellite guidance, is there an equivalent of a yellow light that enables safer, more efficient transition between phases? The particular challenge here is minimizing the control effort.

Question three will be explored by comparing the simulated experimental MPC model with a validated optimal control model from literature. The two will be compared in terms of computational cost, control effort, accuracy, and constraint enforcement. Additionally, the MPC model will be evaluated for stability. It is expected that MPC will be more computationally efficient while preserving optimality, making it an excellent candidate for use as an on-board controller.

1.4 Organization of the Thesis

Chapter two of this thesis reviews prior work done in RPO, MPC, and phase transitions. Chapter three discusses the methodology used to design a controller with time varying-constraints that implements a proposed phase transition method. Chapter four will present test cases, results, and analysis. Additionally, the MPC solution will be compared to a validated optimal control solution. Finally, chapter

five will summarize the thesis and give recommendations for future work.

II. Literature Review

This chapter reviews the literature and provides the governing principles for the primary topics in this thesis: rendezvous and proximity operations to include relative motion dynamics; model predictive control with formulation and solution methods; and, phase transition methods for multi-phase controllers. Governing equations and relevant derivations are provided as needed.

2.1 Rendezvous and Proximity Operations

RPO encompasses the maneuvers required of spacecraft to perform missions within the vicinity of other spacecraft. These operations cover a broad range of spacecraft missions including target tracking, rendezvous and docking, debris removal, imagery, communications, and more [15]. While satellites of many sizes can be involved in RPO, the concept of smaller, lighter satellites working in concert is becoming a cost-efficient alternative to large satellite operations [16].

RPO is also sometimes referred to as Formation Flying (FF) in literature. The name can imply a cooperative target, although a survey of Guidance, Navigation, and Control (GNC) applications by Di Mauro et al. [16] discusses autonomous missions involving both cooperative and non-cooperative targets. Cooperative simply means that all controllers share a common objective, whereas non-cooperative controllers each have their own objectives [1]. The focus of this thesis will be a proximity operation utilizing different satellite maneuvers, referred to as formations. The target's cooperation is outside the scope of this work. More information about the topic can be found in studies on differential gaming such as Prince et al.'s [17] 2019 paper.

2.1.1 Reference Frame

It is helpful in RPO to refer to the motion of satellites with respect to one another. The satellite performing the maneuver is designated the chaser, or "deputy." The motion of the deputy can be described relative to its target, or "chief." Rather than defining the motion of both objects with respect to an inertial reference frame, the motion of objects can be formulated in a more convenient local frame, called the Local-Vertical, Local-Horizontal (LVLH) frame, illustrated in Figure 1. This is a rotating reference frame centered at the chief. The fundamental plane is located on the orbital plane with the $\hat{x}$ direction pointing radially outward, aligned with the vector from the Earth to the chief. The $\hat{z}$ direction is normal to the orbital plane, and the in-track $\hat{y}$ direction is orthogonal to $\hat{x}$ and $\hat{z}$ [18, 19].

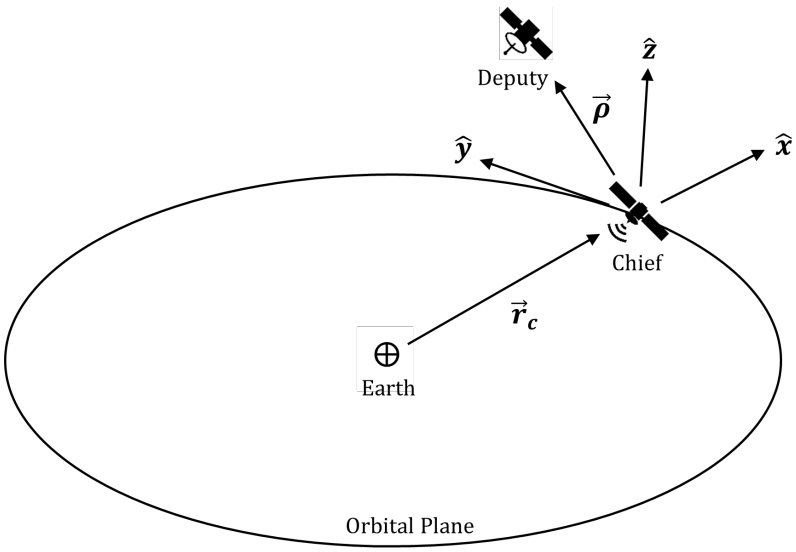


Figure 1. LVLH Reference Frame

Using the LVLH frame, the deputy's position can then be described in terms of its relative position vector to the chief, $\vec{\rho}$. The equations of motion for the deputy will be derived in the next section.

2.1.2 Hill-Clohessey Wiltshire Equations

Dynamic models for relative spacecraft motion are derived in texts such as Alfriend's *Spacecraft Formation Flying*[18] and Schaub's and Junkins' *Analytical Mechanics of Space Systems*[20]. Alfriend's derivation is summarized here, starting with the relative position vector between a deputy and chief spacecraft in the inertial frame, $\mathcal{I}$,

$$^{\mathcal{I}}\vec{\rho} = (\vec{r}_d - \vec{r}_c), \tag{1}$$

where $\vec{r}_d$ and $\vec{r}_c$ are the position vectors of the deputy and chief in the inertial frame, respectively. This is then converted to the LVLH or Hill frame, $\mathcal{L}$, using a Direction Cosine Matrix (DCM), $R^{\mathcal{LI}}$ such that,

$$^{\mathcal{L}}\vec{\rho} = R^{\mathcal{LI}}(\vec{r}_d - \vec{r}_c), \tag{2}$$

where $R^{\mathcal{LI}}$ is given by,

$$R^{\mathcal{LI}} = [\hat{o}_r^T; \hat{o}_\theta^T; \hat{o}_h^T]. \tag{3}$$

The vectors $\hat{o}_r^T, \hat{o}_\theta^T, \hat{o}_h^T$ are the basis vectors of the Hill frame. They depend upon $\vec{r}_c$ and the chief's orbital angular momentum vector. The relative velocity vector $\dot{\vec{\rho}}$ can then be found by taking the time derivative of $\vec{\rho}$ and applying the transport theorem. The result is,

$$^{\mathcal{L}}\dot{\vec{\rho}} = {}^{\mathcal{I}}\dot{\vec{\rho}} - {}^{\mathcal{LI}}\vec{\omega} \times {}^{\mathcal{I}}\vec{\rho}, \tag{4}$$

where $\vec{\omega}$ is the angular velocity vector given by,

$$^{\mathcal{LI}}\vec{\omega} = [0, 0, \dot{\theta}_c]^T. \tag{5}$$

$\dot{\theta}_c$ is the chief's time rate of change of the argument of true latitude. It is the sum

of the rate of argument of periapsis - which is zero for unperturbed motion - and rate of true anomaly. The subscript is used here for clarity, as a different meaning of θ will be used later.

Taking the derivative of the velocity vector, $\dot{\vec{\rho}}$, then yields the relative acceleration between the two bodies,

$$\ddot{\vec{\rho}} = \frac{d^{2\mathcal{L}}\vec{\rho}}{dt^2} + 2^{\mathcal{LI}}\vec{\omega} \times \frac{d^{\mathcal{L}}\vec{\rho}}{dt} + \frac{d^{\mathcal{LI}}\vec{\omega}}{dt} \times \vec{\rho} + {}^{\mathcal{LI}}\vec{\omega} \times ({}^{\mathcal{LI}}\vec{\omega} \times \vec{\rho}). \tag{6}$$

Breaking the relative velocity into its vector components and simplifying algebraically gives the Nonlinear Equations of Relative Motion (NERMs), [18],

$$\ddot{x} - 2\dot{\theta}_c\dot{y} - \ddot{\theta}_c y - \dot{\theta}_c^2 x = -\frac{\mu(r_c + x)}{\left[(r_c + x)^2 + y^2 + z^2\right]^{3/2}} + \frac{\mu}{r_c^2} \tag{7}$$

$$\ddot{y} + 2\dot{\theta}_c\dot{x} + \ddot{\theta}_c x - \dot{\theta}_c^2 y = -\frac{\mu y}{\left[(r_c + x)^2 + y^2 + z^2\right]^{3/2}} \tag{8}$$

$$\ddot{z} = -\frac{\mu z}{\left[(r_c + x)^2 + y^2 + z^2\right]^{3/2}} \tag{9}$$

This is a ten-dimensional system of nonlinear Ordinary Differential Equations (ODEs) whose states are: x,y,z,$\dot{x}$,$\dot{y}$, $\dot{z}$,r_c,$\dot{r}_c$,$\dot{\theta}_c$,$\ddot{\theta}_c$. x, y, z, are the components of relative position, $\vec{\rho}$; $\dot{x}, \dot{y}, \dot{z}$ are the components of relative velocity, $\dot{\vec{\rho}}$; and, $\ddot{x}$, $\ddot{y}$, and $\ddot{z}$ are the components of the relative acceleration, $\ddot{\vec{\rho}}$, in the LVLH frame.

If the chief is assumed to be in a circular orbit, then $\dot{\theta}_c = n_c = constant$, where n_c is the mean motion of the chief. Additionally, $\ddot{\theta}_c = 0$. The chief's relative position can be written in the LVLH frame as $[r_c, 0, 0]^T$, where r_c is scalar and equal to the chief's constant semimajor axis, a. The equations above then become,

$$\ddot{x} - 2n\dot{y} - n^2 x = -\frac{\mu(a + x)}{\left[(a + x)^2 + y^2 + z^2\right]^{3/2}} + \frac{\mu}{a^2} \tag{10}$$

$$\ddot{y} + 2n\dot{x} - n^2 y = -\frac{\mu y}{\left[(a+x)^2 + y^2 + z^2\right]^{3/2}} \tag{11}$$

$$\ddot{z} = -\frac{\mu z}{\left[(a+x)^2 + y^2 + z^2\right]^{3/2}}. \tag{12}$$

The subscript is omitted from the mean motion, n, after this point for clarity. Linearizing the system above and neglecting higher order terms yields the linearized Hill-Clohessey-Wiltshire (HCW) or Clohessey-Wiltshire (CW) equations [18]:

$$\ddot{x} - 2n\dot{y} - 3n^2 x = u_x \tag{13}$$

$$\ddot{y} + 2n\dot{x} = u_y \tag{14}$$

$$\ddot{z} + n^2 z = u_z. \tag{15}$$

u_x,u_y, u_z are the control accelerations for each axis. The control can also be represented in vector form by $\vec{u}$, and will be utilized in the controller cost function to minimize control effort. This is now a 6-dimensional, linear system of ODE's whose out-of-plane motion is decoupled from the radial and in-track motion.

The use of HCW dynamics, though useful in simplifying problems in order to demonstrate control concepts, has some limitations. For one, HCW is only valid for circular or near circular orbits. Elliptical orbits require other models of relative motion, many of which are outlined in Sullivan et al.'s [21] survey on the subject. Also, the concept of stability for HCW is local to the chief within small relative distances. As such, this dynamic model is only valid for close proximity applications [22].

It is important to note that this thesis chiefly considers translational motion of the spacecraft. However, an expanding area of research has utilized the full Six Degree of Freedom (6-DoF) model of spacecraft motion [9]. In the 6-DoF model, components of rotation and translation become state variables. The vectors can be decoupled if

the translational motion does not depend on the satellite's torque input, and if the rotational motion is independent of thrust. Becker's [9] work utilized Decoupled MPC (DMPC), in which he treated rotation and translation as two separate controllers. Some of his results are compared to this work in Chapter IV.

Whether the dynamics can feasibly be decoupled depends upon the spacecraft's design. For example, a spacecraft whose thrusters are used for both translational and rotational control could not use DMPC [13]. Zagaris [23], derived models for the fully coupled model as well as decoupled translation and rotation. His work analyzed reachability for rendezvous with a tumbling object using a full 6-DoF MPC controller. Other work in 6-DoF control has been done by Zhou et al. [24] for MPC, Li et al. [25] for sliding mode control, and Dong [26] and Buckner [27] for tube-based MPC.

2.1.3 RPO Formations

Two types of formations have been well-studied for RPO scenarios involving a non-maneuvering target: the Natural Motion Circumnavigation (NMC) orbit, and the pogo or teardrop maneuver. Lovell and Tollefson [28] detailed a method for an impulsive hover using the HCW closed-form solution and Relative Orbital Elements (ROE). Prince [29] revisited this work when implementing both an NMC orbit and teardrop maneuver in his work on optimal finite thrust guidance for constraint inspection maneuvers.

Both maneuvers rely on the natural motion of the deputy satellite, in this case the HCW dynamics. Propagating the spacecraft's motion requires the closed-form solution to the HCW equations. This solution is derived in texts such as Alfriend's [18]. One solution method is via Laplace transforms of the first-and second-order terms for each differential equation. Another method uses the exponential of the plant matrix to find the State Transition Matrix (STM) $\mathbf{\Phi}(t, t_0)$, as a function of

current time, t, and initial time, t_0. This is used to calculate the state according to the equation,

$$\mathbf{x}(t) = \mathbf{\Phi}(t, t_0)\mathbf{x}(t_0). \tag{16}$$

To find the STM, the unforced HCW equations are first written in state space form,

$$\dot{\mathbf{x}} = A\mathbf{x}, \tag{17}$$

where $\mathbf{x}$ is the state vector and $\dot{\mathbf{x}}$ is its derivative. The n-by-n matrix A is the plant matrix of the HCW equations given by,

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 3n^2 & 0 & 0 & 0 & 2n & 0 \\ 0 & 0 & 0 & -2n & 0 & 0 \\ 0 & 0 & -n^2 & 0 & 0 & 0 \end{bmatrix}. \tag{18}$$

$\mathbf{\Phi}(t, t_0)$ can be found through the solution to Equation 16 and by using the exponential of the matrix. Since the problem is time-invariant, the exponential can be found using the Laplace transform $\mathscr{L}$,

$$\mathbf{\Phi}(t, t_0) = e^{A(t-t_0)} = \mathscr{L}^{-1}\{(sI - A)^{-1}\}. \tag{19}$$

The STM is found to be,

$$\mathbf{\Phi} = \begin{bmatrix} 4-3\cos nt & 0 & 0 & \frac{\sin nt}{n} & \frac{-(2(\cos nt-1))}{n} & 0 \\ 6\sin nt-6nt & 1 & 0 & \frac{(2(\cos nt-1))}{n} & \frac{(4\sin nt-3nt)}{n} & 0 \\ 0 & 0 & \cos nt & 0 & 0 & \frac{\sin nt}{n} \\ 3n\sin nt & 0 & 0 & \cos nt & 2\sin nt & 0 \\ 6n(\cos nt-1) & 0 & 0 & -2\sin nt & 4\cos nt-3 & 0 \\ 0 & 0 & -n\sin nt & 0 & 0 & \cos nt \end{bmatrix}. \tag{20}$$

The closed-form solution of the HCW equations can be used to propagate the natural motion of the trajectory given the desired beginning and end states. It is important to note that the in-track equation has a secular term, indicating drift in that direction. A stable subspace can be found for the state space with the initial condition,

$$\dot{y}(0) = -2nx(0). \tag{21}$$

Equation 21 is referred to as the energy-matching condition. By satisfying this initial condition, the deputy matches the energy of the chief's orbit and is able to remain in a closed relative orbit [18]. The projection of the deputy's relative orbit onto the chief's orbital plane is a two-by-one ellipse. The solution above was used by Lovell [28] to formulate Relative Orbital Elements. For this work it is sufficient to use Cartesian coordinates when defining the deputy's position relative to the chief. A simple maneuver which transfers the deputy from one NMC orbit to another is illustrated in Figure 2.

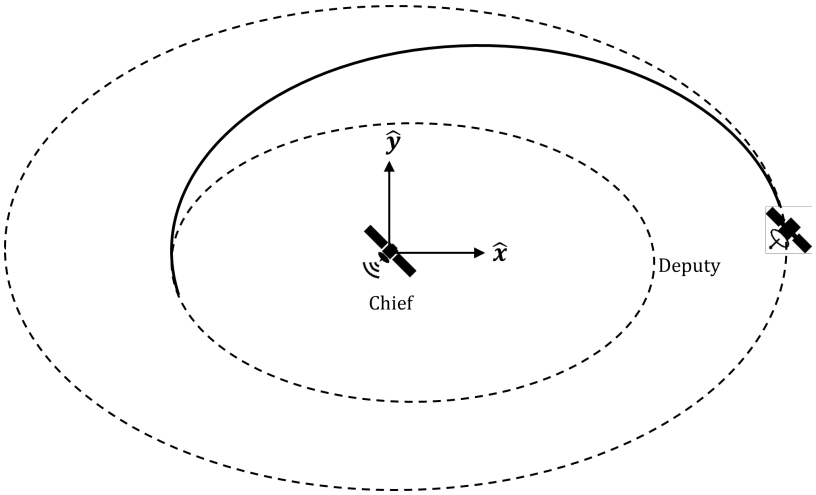


Figure 2. Relative Orbit Transfer Using Natural Motion Circumnavigation

By utilizing the closed-form HCW solution, the transfer in Figure 2 is a matter of specifying the desired initial and final states of the deputy's transfer. If the positions are defined using Cartesian coordinates, the relative velocities can be calculated to satisfy the energy matching conditions for a stable orbit. Such a maneuver can be accomplished using either impulsive or continuous thrust. Further considerations can be made for thruster configuration and direction of thrust [16], but are outside the scope of this work.

A pogo or teardrop formation is a type of "quasi-hovering" orbit that can be achieved for non-zero radial displacements. In this case, the satellite will drift in the negative in-track direction, tracing the outline of a teardrop. This maneuver is useful for inspection scenarios, as the maneuver can be repeated as many times as necessary necessary by applying the required Δv. The trajectory was formulated in the aforementioned work by Lovell [28], as seen in Figure 3.

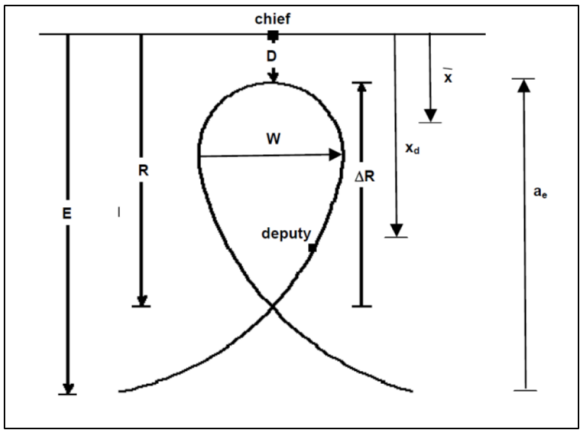


Figure 3. Lovell's Teardrop Formulation

The trajectory in Figure 3 can be calculated using the CW Targeting problem. As its name suggests, a desired state is targeted, and then the value of Δv can be calculated using the STM. First, the state vector can be re-written in terms of the relative position, $\mathbf{r}$ and velocity, $\mathbf{v}$,

$$\boldsymbol{x} = [\boldsymbol{r}, \boldsymbol{v}]^T. \tag{22}$$

Then, the STM is partitioned as follows,

$$\mathbf{\Phi} = \begin{bmatrix} \mathbf{\Phi}_{rr} & \mathbf{\Phi}_{rv} \\ \mathbf{\Phi}_{vr} & \mathbf{\Phi}_{vv} \end{bmatrix}. \tag{23}$$

From this, the required impulsive Δv to reach the final position $\mathbf{r_f}$ in a given time is found by,

$$\Delta v = \mathbf{\Phi}_{rv}^{-1}(\mathbf{r_f} - \mathbf{\Phi}_{rr}\mathbf{r_0}) - \mathbf{v_0}. \tag{24}$$

Here, $\mathbf{r_0}$ is the initial position, and $\mathbf{v_0}$ is the initial velocity. This initial velocity is applied to the initial state vector, and the motion is propagated using the HCW STM. By setting the initial state equal to the final state and $t - t_0 \geq 0$, a reference trajectory can be created in the shape of the desired teardrop. The natural teardrop trajectory can then be tracked by a controller, while optionally enforcing safety constraints. Zagaris et al. [12]

created a teardrop tracking problem which brought a satellite within close proximity of a Keep-Out-Zone (KOZ) constraint. The proposed MPC controller was designed to avoid the KOZ while tracking the teardrop trajectory as closely as possible. The CW targeting problem can occasionally become problematic, as $\mathbf{\Phi}_{rv}$ is sometimes singular due to the periodic terms in the STM [18].

2.2 Model Predictive Control

MPC has steadily gained popularity in the field of autonomous control as a robust, computationally efficient control framework. While classical optimal control methods are useful for mission planning and validation, they tend to be too computationally costly for on-board implementation [30]. MPC, on the other hand, offers reduced computational complexity due to its finite receding horizon framework.

2.2.1 Background

MPC was first developed in the 1960's, but not applied to aerospace until 1984. Since then, the field has expanded to applications in many areas of aerospace. Eren et al.'s [10] survey paper is an excellent summary of developments through 2017. They expound on the importance of autonomy in spaceflight, saying, "...it is standard that current flight control systems have to provide a priori guarantees of constraint satisfaction, safety, optimized performance, robustness, and/or adaptability with respect to the changes in employed models and operating conditions [10]."

In its earlier development, MPC was relegated to slow industrial processes with near-steady-state conditions. Its progress was initially hindered by hardware capabilities and a lack of appropriate numerical solvers. As advances in hardware and optimal control solution methods grew, these problems began to fade and MPC's potential became realizable. Since 2004 alone, it has been utilized in chemical process control, automotive engine control, and multi-agent control in transportation networks [31]. Improved solution methods and advancements in stability conditions enabled the application of MPC to linear or nonlinear

systems with hard state and control constraints [1]. Since then, extensive research into RPO using MPC has been developed. Di Cairano et al. [32], for example, demonstrated rendezvous with a tumbling object using MPC.

One advantage of MPC over the OCP is that it can be used to force both linear and nonlinear constraints. While there are OCP methods that can handle some constraints, algorithms can quickly become intractable [33]. Weiss et al. [34, 35] have provided substantial overviews of strategies for constraint handling using MPC. Malyuta describes a process of lossless convexification which relies on Second-Order Cone Programming (SOCP) [36]. Other constraint considerations such as collision avoidance, approach direction, and slow impact are examined by Tevfik [37]. While methods have been developed for MPC to handle nonlinear constraints, and even nonconvex constraints in some cases, there has also been research into constraint linearization specifically for MPC. Zagaris et al. [38] compared three methods for obtaining linear approximations of ellipsoidal KOZ constraints: rotating-hyperplane, dual-hyperplane, and direct-linearization. Keyanpour's paper also discusses constraint linearization in order to apply quadratic programming [39].

Although outside the scope of this work, noise and perturbations are significant factors in real-world RPO missions. Several studies have been done in this area with respect to MPC that are worth mentioning for further reading. Weiss [34] explored a multi-phase maneuver using a controller augmented with an Extended Kalman Filter (EKF). Sanchez et al. [40] developed robust MPC for the restricted three-body problem (R3BP) for applications to cislunar space. Larsen's [41] 2019 study examines a fuel-efficient rendezvous maneuver with an uncontrolled object . Jin's papers examine the robustness of MPC for elliptical, perturbed orbits [42, 43].

2.2.2 Formulation

MPC uses a discretized dynamics model to predict the optimal trajectory out to a finite prediction horizon, N. The future state is a function of the state and control input at the current time. The optimal control problem is recursively solved for each horizon. The

final state of the segment is the initial condition for the next prediction horizon [11]. A natural benefit of this is that MPC is not as sensitive to an initial guess as the OCP. A larger advantage is the decreased computation time as the infinite optimal control problem is effectively broken into smaller segments. The disadvantage is that the model's foresight is limited to the length of the prediction horizon [9].

The book by Rawlings and Mayne [11], *Model Predictive Control: Theory, Computation, and Design*, is a comprehensive overview of MPC theory and development. The problem is briefly summarized here. In general, the MPC optimization problem minimizes the cost, J, according to

$$J = V_f + \frac{1}{2}\sum_{k=1}^{N-1} l. \tag{25}$$

The stage cost, l, is defined as a standard quadratic regulator,

$$l = [(x_k - x_d)^T Q(x_k - x_d) + u_k^T R u_k], \tag{26}$$

where $(x_k - x_d)$ is the state tracking error to the desired state, x_d. x_k is the discretized state vector at each time step, k, for the prediction horizon of length N. Q and R are the weighting matrices for the state and control, respectively.

The terminal cost, V_f is given by,

$$V_f = (x_N - x_d)^T Q_f (x_N - x_d), \tag{27}$$

where Q_f is the weighting for the terminal cost. It is found from the solution to the discrete algebraic Ricatti equation for the unconstrained infinite horizon LQR [44]. In his work, Zagaris [23] explains that including the terminal cost provides a guarantee in stability.

Eren et al. presented a general flow for the MPC solution algorithm, which can be seen in Figure 4 [10].

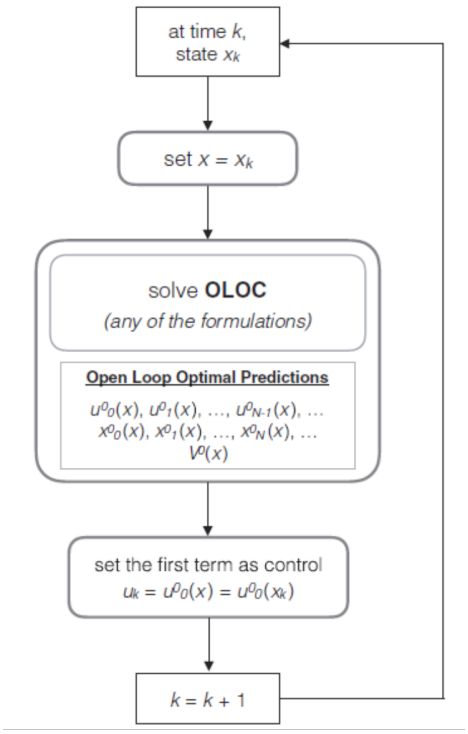


Figure 4. MPC Algorithm Flow

For each prediction horizon, the OCP returns the open-loop optimal control solution in the form,

$$U_k = [u_{1|k}, u_{2|k}, u_{3|k}, ..., u_{N|k}]. \tag{28}$$

$u_{N|k}$ is the control effort for each time step, k, of each prediction horizon, N. So, the overall solution U_k is a vector, each entry of which is an individual vector for the prediction horizon. Each $u_{N_{h|k}}$ can then be applied to return the feedback control law for each horizon,

$$K = [u_{1|k}, u_{1|k+1}, u_{1|k+2}, ..., u_{1|k+N_{sim}}]. \tag{29}$$

The feedback loop applies the first control input from Equation 28 to the system, which propagates the plant forward in time. The initial values of the control and states at the beginning of each prediction horizon serve as predicted end states for the preceding horizon.

N_{sim} is the number of desired simulation steps. In many cases it is helpful to define

this value as the total scenario time divided by a desired time step. A default time step of one second can be helpful in some cases, but may be too computationally expensive in others, such as a long, uncomplicated maneuver at high orbit altitudes. Emara-Shabaik demonstrated that the solution to the MPC usually performs similarly to the infinite horizon OCP [9].

2.2.3 Solution Methods

Both linear and nonlinear solution methods have been developed throughout the years. One solution to a MPC problem is found through standard quadratic programming (LQ-MPC). In his 2012 paper, Weiss analyzed spacecraft relative motion control using application of LQ-MPC and dynamically reconfigurable constraints [34]. The computational advantages of quadratic programming were demonstrated by another study done by Jung et al. [45]. In their 2018 paper, Zagaris et al. discussed how the use of LQ-MPC is advantageous for on-board implementation [38]. Additional studies in quadratic programming were found by Laiu [46], Keyanpour [39], and Di Cairano [32]. More generally, Di Cairano's paper examines MPC suitability for embedment in aerospace platforms.

Nonlinear MPC (NMPC) is another possibility using Non-Linear Programming (NLP) solvers such as MPCTools, which is used in this work [11]. The advantage of NMPC is that performance can be optimized without conservative approximations to nonlinear constraints. For example, approximating a conical constraint as a series of planes necessitates consideration for performance versus computation cost trade-off. The disadvantage is that NLP is not as computationally efficient, making it less useful for on-board implementation. A list of available MPC solution tools is given in Eren et al.'s survey [10].

2.3 Multi-Phase Operations

This work builds upon prior work by exploring how phase transitions occur between mission phases that utilize MPC. This research explores both hard and soft constraints in the MPC framework for both nonlinear constraints. Overall, there is not much literature

surrounding the topic of phase transitions for MPC, especially for 6-DoF problems [9]. Possible methods to implement phase transitions include hard entry and exit criteria, mixed-integer programming, logistic functions, and soft constraints.

Most of the MPC literature appears to rely on hard constraints for phase transition. Weiss et al. [35] applied MPC to a rendezvous and docking scenario with different constraints for each phase. In his thesis, Becker [9] designed an MPC controller for a 6-phase satellite inspection mission. His phase transitions consisted of hard exit criteria which ensured constraint satisfaction allowing entry into the next phase. The drawback of his trajectory was that there were often sharp turns near his phase transitions. Similarly, Jewison et al [47] designed a multi-phase mission. Their phase entry and exit criteria were relatively simple, relying on relative distance to their target in order to trigger the next phase, leading to abrupt transitions. An application of mixed integer programming to MPC was not found in the research. This method shows potential, as it offers a way to solve finite horizon optimal control problems using mixed integer and linear programming [48].

Saunders [49] took his work with phase transitions a step further by deriving a logistic function. He applied MPC to a cooperative formation flying inspection using an artificial potential function. His phase transition utilized a statistical curve to provide a variable measure of closeness to satisfying constraints. This method is again based on range to the target, but provided for smooth phase transitions by utilizing statistical analysis [49]. The last phase-transition method proposed, softening the constraints, was not found in literature for use as a phase transition method. It is investigated in this work for its potential in multi-phase RPO.

2.4 Summary

This chapter provided an overview of the guiding principles and reviewed the literature in RPO, MPC, and multi-phase missions. As noted, the largest gaps found in the literature regarded time-varying constraints and phase transitions. These concepts are linked to one another by the fact that each phase requires constraint definition. Chapters III and IV will

discuss how constraint definition can be used to streamline phase transitions and optimize a satellite maneuver overall.

III. Methodology

This chapter defines a multi-phase satellite inspection scenario and explores the methods used to solve for the optimal trajectory. The controller components such as the plant, constraints, cost functions, and bounds are formulated specifically for use with MPC. Then, slack variables are incorporated into the constraint and cost functions as the mechanism for phase transitions. Chosen software and solution methods are described.

3.1 Dynamics and Relative Motion

The satellite inspection mission specifies a chief assumed to be in near-circular orbit about the earth. The chief's initial position and velocity in the inertial frame are calculated from its Classical Orbital Elements (COE) [19], which are specified for each test instance in the next chapter. This information is used later to calculate the sun vector for one of the path constraints. The chief's semimajor axis, a, is used to calculate the chief's mean motion, n, and orbital period, T,

$$n = \sqrt{\frac{\mu}{a^3}} \tag{30}$$

$$T = \frac{2\pi}{n} \tag{31}$$

Since the deputy is orbiting the chief in relatively close proximity, it shares the value of the chief's mean motion. Thus, the only information needed to propagate the deputy's orbit using the HCW equations is the chief's semimajor axis and its relative position in the LVLH frame. This is done using Cartesian coordinates with the chief at the center of the frame. The relative velocity is initially defined using the energy matching condition from Equation 21 to place it on an NMC about the chief. The deputy is assumed to have constant mass and be capable of continuous thrust. As such, the deputy's initial and final

conditions for the overall maneuver are given by,

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ \frac{ny}{2} \\ -2nx \\ 0 \end{bmatrix} \tag{32}$$

.

The HCW equations of relative motion are selected for the inspection scenario because because literature has shown them to be sufficiently accurate for near-circular orbits over short periods of time [21]. The ODEs will form the plant of the controller. The Linear Time-Invariant (LTI) system can be written in state space as,

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u} \tag{33}$$

where A is the plant or system matrix in Equation 18, and B is the input matrix corresponding to the control, u,

$$\boldsymbol{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{34}$$

The inspection mission consists of the three phases summarized in Table 3.

Table 3. Inspecton Phases

Phase	Maneuver	Entry Criteria	Exit Criteria
1) Appproach	Transfer from NMC	N/A	KIZ Entry
2) Observe	Teardrop, CW Targeting	Sun Vector Alignment	Desired Time
3) Depart	Transfer to NMC	Natural Motion	N/A

In the approach phase, the deputy will transfer from its current NMC to an insertion point closer to the chief. Using the NMC transfer concept discussed in Chapter II is still applicable here, except that the deputy will navigate to a stationary point at a certain relative distance from the chief. Once the deputy is within a conical Keep-in-Zone (KIZ) representing illumination of the sun on the chief, it will begin its quasi-hovering teardrop maneuver. After a specified amount of time, the deputy will exit the teardrop and return to natural motion about the chief.

It should be noted that this problem assumes perfect knowledge of the deputy's state vector. Noise is not taken into account, and as such no filtering is discussed in this work. Previous literature has explored filtering using MPC, see Weiss [34], Wen and Gurfil [50], Mammarella [51], Larsen [41], and Sanchez [40].

3.2 Constraints and Bounds

The scenario is subject to two path constraints, a KOZ and KIZ, as well as bounds on the states and controls. The path constraints are defined mathematically. The bounds are discussed here in terms of selection and considerations, but final values will be given with each test case in Chapter IV.

3.2.1 Path Constraints

The KOZ is a spheroid of radius r with the chief at its center. The constraint is formed with respect to the norm of the deputy's relative position vector, $\vec{v}_a$, at any given time. The

formula is,

$$r - \|\vec{v}_a\| \leq 0. \tag{35}$$

During the observation and hover phases, it is desired that the deputy maintain the sun vector constraint such that the chief is constantly illuminated. This is defined by a cone whose point originates at the chief. The cone is formed by a unit vector, $\hat{v}_s$, which points from the chief to the sun, and a half-angle, θ. Within the KIZ, the angle between the deputy relative to the chief and the sun vector must be less than or equal to the half-angle of the cone. This gives the constraint,

$$\cos\theta - \hat{v}_a^T \hat{v}_s \leq 0. \tag{36}$$

An illustration of both the spherical KOZ and the conical KIZ are given in Figure 5 below.

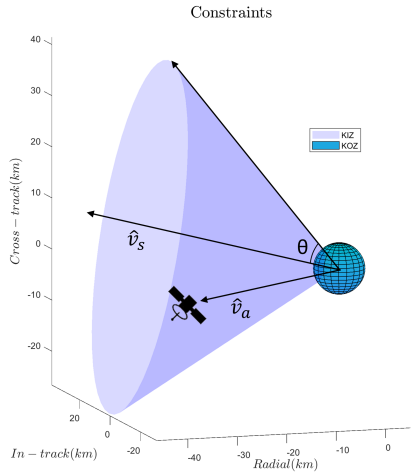


Figure 5. Illustration of Constraint Definitions

The unit sun vector, $\hat{v}_s$, changes with every time step, although the distance from the Earth to the Sun makes it a relatively slow change, especially at higher orbital regimes. Nevertheless, it is possible to re-compute the vector using a for loop based upon the Julian Day. For this process, the code developed by David Vallado [19] was used.

Vallado's code calculates the vector from the Sun to the Earth, not to the chief, so additional computation is required. The chief's COEs are converted to its initial position and velocity relative to the Earth in the inertial frame. Then, its motion is propagated by

integrating the two-body problem in MATLAB. The resulting position vector is added to the vector from Sun to Earth at each time step. The resulting vector must be converted to the LVLH frame and normalized, yielding the final $\hat{v}_s$ required to enforce the constraint.

Conical path constraints have been implemented in literature [9, 49] either as KIZs or KOZs. The KOZ can be formulated by simply flipping the inequality. There are many instances in which such a constraint might be applied. In this work, it is used to enable illumination of the chief for observation purposes. It could be used as a KOZ to keep delicate sensors from pointing directly into the sun. It could also be implemented as a Line-of-Sight (LOS) attitude requirement for observing the chief.

Although the KIZ is inherently non-convex, convexification is outside the scope of this work. In general, non-convex constraints render an optimization NP-hard [52]. However, nonlinear optimizers such as MPCTools can be used at the cost of computational efficiency. There are two other potential complications with using non-convex constraints. One is that, the problem loses its guaranteed convergence. The other is that the optimality of the solution is not guaranteed [9, 12].

3.2.2 Bounds

State and control bounds allow the solver to narrow the search for the solution's starting point. As mentioned earlier, the recursive MPC framework makes it less sensitive to an initial guess than an open-loop OCP solver such as GPOPS-II. As such, the bounds can be relatively generous as long as they encompass the initial state.

Mathematically, the maximum control output is bounded using the infinity norm of the control, $\vec{u}$,

$$\|\vec{u}\|_\infty \leq u_{max}. \tag{37}$$

In this work, the state was assigned a minimum value for position in kilometers and velocity in kilometers per second. The maximums are the negatives of those bounds.

3.3 Optimal Control Problem

The optimal control problem minimizes the continuous running cost given by the function,

$$J = \int_{t_0}^{t_f} \frac{1}{2} u^T(t) R u(t)\, dt. \tag{38}$$

subject to the constraints described above. The R in Equation 38 is a weighting matrix for the controls. In Chapter IV, the methods in this work are compared to methods from literature which have been validated. The previous work solved the OCP problem using the Interior Point Optimizer (IPOPT) [53] in the General Purpose OPtimal Control (GPOPS-II) software [54]. In order to validate the solution, the control output from GPOPS-II was applied to the HCW equations, which were then integrated using MATLAB's *ode45* function. The solution was propagated and the final state of the validation was compared to the final state of the GPOPS-II output. The validated model can be used as a comparison for the MPC model.

3.4 MPC Solution

As discussed in Chapter II, MPC works by applying optimal control at discrete time steps. The HCW ODEs can be represented in discretized form in state space by,

$$x_{k+1} = A_d x_k + B_d u_k. \tag{39}$$

where $k = 1, 2, ..., N-1$ is the prediction horizon and N is the number of steps being predicted forward. Note that A_d and B_d are discrete versions of the continuous state-space matrices seen in Equations 18 and 34. From Equations 25, 26, and 27, the cost function is written as,

$$J = (x_N - x_d)^T Q_f (x_N - x_d) + \frac{1}{2} \sum_{k=1}^{N-1} [(x_k - x_d)^T Q (x_k - x_d) + u_k^T R u_k]. \tag{40}$$

Two programs were used to implement the MPC solution. The first is the MPCTools,

developed by James Rawlings [44]. The package is an open-source control and estimation tool for both linear and nonlinear dynamic modeling. The program was designed for Octave but also supports MATLAB. It provides an interface for creating the MPC variables and structure required for the solution. The user assigns the bounds, state variables, initial conditions, linear or nonlinear dynamics, slack variables, and prediction horizons using the pre-defined structures in MPCTools. Additionally, the path constraint, stage cost, and terminal costs are all defined. MPCTools discretizes the state matrices and can linearize nonlinar dynamics. It then "packages" the information into symbolic variables and functions which are passed into the Computer Algebra System + Algorithmic Differentiation (CasADi) program.

CasADi is a symbolic algebraic and differentiation solver. CasADi takes the variables and functions from MPCTools and constructs algebraic expressions. It can even do some simplification. Once it formulates the constrained optimization problem, it can pass the model into IPOPT to generate a solution. So, CasADi serves as an interface between MPCTools and IPOPT or other solvers. While CasADi is a powerful tool, it is very difficult to formulate an MPC problem in it directly, which is why Rawlings Group created MPCTools [44]. MPCTools and CasADi have been demonstrated in literature to be powerful nonlinear optimization tools. Malladi's [14] 2019 paper used them to simulate a 6-DoF manuever using MPC.

3.5 Phase Transition

Phase transition methods from literature were discussed in Section 2.3. The purpose of this work is to show that phase transitions can be accomplished by modifying the constraints of the optimal control problem. This allows the deputy to move between constraint zones while still adhering to performance requirements.

Constraints in an optimization problem often take the form of an inequality, requiring the constraint equation to be less than or equal to some value. When that value is zero, the constraint is considered to be "hard". However, it is possible that the equality can be

within some margin, causing the constraint to be "soft." The margin, or slack, is given by the variable s. This value can be some pre-defined number, or it can be optimized using the cost function. MPCTools automatically optimizes the value of s when slack variables are implemented. This softens the constraints and allow them to become violable.

The slackened KOZ constraint can be written as,

$$r - \|v_a\| \leq s. \tag{41}$$

Collision of the deputy with a chief would mean unacceptable mission failure. Therefore, it is rare that this constraint could be relaxed. In fact, making the KOZ a hard constraint adds a passive safety quality to the controller that makes it desirable for on-board implementation. MPC's constraint enforcement is one of the qualities that make it so popular in aerospace. Of course, hardware failures or an insufficient prediction horizon on the MPC controller could still cause constraint violations. Nevertheless, it is important to note that even a soft constraint is still a constraint, and sufficient radius around the chief is another safety factor.

With the slack variable in the KIZ, the zero is replaced with s to give,

$$\cos\theta - \hat{v}_a^T \hat{v}_s \leq s \tag{42}$$

Unlike with the spherical KOZ, there are many instances in which it could be useful to soften this constraint. If the deputy needs to navigate in or outside of a an area bounded by a constraint, softening the constraint is one way to allow this. Another instance is one in which a satellite may wish to avoid a certain area, but control costs or time are a larger concern. This could occur when a satellite does not wish to obstruct imaging equipment of another satellite, but skirting the edge of the viewing area would not adversely impact the mission.

When the slack variables are used, they are implemented into the cost functions. So the

stage cost becomes,

$$l = k_{Q+R}[x_e^T Q x_e + u^T R u] + k_s[s^T s], \tag{43}$$

and the terminal cost is,

$$V_f = k_Q[x_e^T Q_f x_e] + k_s[s^T s]. \tag{44}$$

The variables k_{Q+R}, k_s, and k_Q are weighting parameters introduced into the cost functions. This allows the optimal control costs to be weighed against the slack variables. Violating the slack variables comes at a cost to the state and control. Therefore, it must be determined whether the driving factor of each phase is constraint enforcement or performance. As with most engineering problems, the answer is not one or the other but a matter of cost-versus-benefit. In Chapter IV, the slack will be weighted for a range of values to observe the impact to the trajectory, costs, and control effort.

3.6 Summary

This chapter formulated the dynamics, constraints, and MPC problem necessary to analyze time-varying constraints and the use of slack variables as a means for phase transition. Having a ready-made framework in which to implement MPC significantly aids in applying the concepts above to an RPO scenario. While MPCTools and CasADi are unique in this regard and perhaps not easily adapted to mission planning tools, they are excellent programs for testing various trajectory optimization methods. The next chapter will discuss specific test instances and their results.

IV. Results and Analysis

This chapter consists of three main sections. First, the mutli-phase inspection mission described in Chapter III is defined for Low Earth Orbit (LEO) and Geostationary Orbit (GEO) scenarios. Then, simulation results are compared for the trajectories, constraint enforcement, controls, and computation times. Lastly, the slack variable methodology is implemented for a 6-DoF multi-phase, MPC problem from literature and compared with the original result.

All coding and simulations for this work were completed using a Lenovo Yoga C930 laptop. The processor is an Intel(R) Core(TM) i7-8550U (8th gen) CPU at 1.80 GHz. The operating system is 64-bit with 16 GB of RAM. No additional hardware was required. All software is either free-use or provided via AFIT's student license. The most recent versions of MPCTools and CasaDi are open-source, available online [44, 55]. MATLAB 2019b and GPOPS-II are licensed through the school. GPOPS-II contains IPOPT.

4.1 Test Instances

The inspection scenario in Chapter III was created in MATLAB for both LEO and GEO examples. Additionally, a single-phase "toy" instance was simulated for the purposes of validating the constraint enforcement using slack variables. This instance uses the GEO parameters.

4.1.1 Chief and Deputy Parameters

Each test instance is defined below in terms of the chief and deputy orbits, initial and final conditions of each phase, constraint settings, and controller settings. The chief's orbital elements are given in Table 4. The elements are notional values but were motivated by the orbits of the International Space Station (ISS) for LEO and EchoStar 23 for GEO [56], [57]. Although the orbits are assumed to be circular in theory, a very small eccentricity was

added to each orbit in order to avoid singularities when calculating the chief's position and velocity vectors from the COEs using Vallado's code [19].

Table 4. Chief Orbital Elements

Variable	**Element (units)**	**LEO**	**GEO**
a	Semimajor Axis (km)	6,800	42,000
T	Period (hours)	1.5501	23.7948
e	Eccentricity	0.001	0.001
i	Inclination (deg)	45	0.01
Ω	Ascending Node (deg)	145	300
ω	Argument of Periapsis (deg)	3.8	112
f	True anomaly at t_0 (deg)	90.1	7

As discussed in Section 3.1, the deputy's position relative to the chief is defined using the Cartesian coordinate system in the LVLH reference frame. Then, the velocity components of the state vectors follow according to the closed-form solution to the unforced HCW equations, making the state vector,

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ \frac{ny}{2} \\ -2nx \\ 0 \end{bmatrix}. \tag{45}$$

The selection of the states required some tuning for each individual orbital regime. The states and times are defined in Table 5 for the LEO case and Table 6 for the GEO case.

The overall initial and final states were selected to ensure the maneuver started outside the KIZ. Since the KIZ is already well-defined by its time-varying unit sun vector, it was decided that the final state of the first phase would be chosen to align with $\hat{v}_s$ at a specified distance in kilometers from the chief. By constraining the final position to the vector, the goal was for the deputy to rotate with the vector and stay inside the KIZ. This position becomes the starting point for the teardrop maneuver in phase two. The teardrop forma-

tion using CW-targeting specifies that the initial and final positions be the same. It was hypothesized that the deputy would try to adhere to this constraint, but with the added path constraint of the KIZ it would adjust course in order to avoid constraint violation.

For the third and final phase, it was desired that the deputy exit the KIZ and return to natural motion about the chief to await its next instructions. This was accomplished by defining the NMC and using the tracking problem from phase two.

Table 5. Initial and Final Conditions for Each Phase, LEO

Phase	**Initial** (x, y, z)	**Final** (x, y, z)	**Time**
1) Approach	$\mathbf{x} = [-1, -2, 0.25]^T$	$0.4 * \hat{v}_s$	$0.5 * T$
2) Observe	Phase 1 X_d	Phase 2 X_0	$0.6 * T$
3) Depart	Phase 2 X_d	[1, 2, 0]	$0.3 * T$

The observation phase for the LEO case was more difficult to define using MPC, as the chief's shorter period meant a more dynamic KIZ. Additionally, the teardrop maneuver was very sensitive to the time and states, and required significant manipulation to provide a trajectory sufficient for observation. Providing a small inclination to the deputy's initial orbit helped it to better align with the sun vector.

Table 6. Initial and Final Conditions for Each Phase, GEO

Phase	**Initial** (x, y, z)	**Final** (x, y, z)	**Time**
1) Approach	$\mathbf{x} = [10, -20, 0]^T$	$3 * \hat{v}_s$	$0.5 * T$
2) Observe	Phase 1 X_d	Phase 2 X_0	$0.3 * T$
3) Depart	Phase 2 X_d	[-6, 6, 5]	$0.4 * T$

Some care had to be taken in assigning times to each phase. The phases in GEO were much longer than those in LEO due to the longer period. While this proved to be beneficial for remaining inside of the constraint cone, it was also challenging to select the appropriate amount of time for each maneuver. After too much time the shifting constraint zone will increase computation cost and control effort. Additionally, the time and weights of Q and R had to be balanced against one another to avoid either over-shooting the final state or simply never reaching it.

4.1.2 Constraints

The KIZ and KOZ constraints are defined in the same way for each mission phase. Their activation is phase-dependent. The KOZ is defined by a sphere of radius r centered on the chief. The KIZ is defined by the sun vector, calculated using the methods in Chapter III, and a cone half-angle, θ. The constraints are summarized for each orbital regime in Table 7.

Table 7. Constraints

Constraint Name	**LEO**	**GEO**
KOZ Radius, r (km)	0.25	1
KIZ Half-angle, θ (deg)	40	40

Constraint definition using MPCTools was relatively straightforward. The initial path constraint is assigned as a variable in CasaDi, and then the solver setting updates the constraint at each time step. Therefore, a time-varying constraint can be treated as a constant at each time step. No special modifications were required for the KIZ depending upon the orbit. The code is generic enough that it will propagate the chief's trajectory and then, given a starting Julian Day, calculate the sun vector at each point of the simulation. The KOZ was chosen to have a 0.25 km radius for LEO and a one-kilometer radius for GEO.

No assumptions are made about the reflectance of the chief. It is possible that aligning the deputy directly with the sun vector could result in glare that limits visibility of the chief. Defining another vector at a slight offset is one way to address this. Defining a KOZ within the KIZ is another possibility, albeit a more complicated one. However, the focus here is enforcing the constraint. The concept can be expanded later once it is more proven.

4.1.3 Controller

The controller settings were selected according to the MPCTools framework, beginning with the prediction horizon and time step. The Q and R matrices were selected in order

to ensure a balance of constraint enforcement and, in the cases of phases two and three, tracking regulation. The Q matrix was chosen to be a function of the maximum position and velocity values, as well as a scaling value for each phase, Q_i, such that,

$$Q = Q_i * \begin{bmatrix} \frac{.4^2}{{x_{max}}^2} I_{3x3} & 0_{3x3} \\ 0_{3x3} & \frac{.4^2}{{v_{max}}^2} \end{bmatrix}. \tag{46}$$

Similarly, R was chosen to be a function of the maximum control and a scaling value, R_i,

$$R = \frac{R_i}{{u_{max}}^2} * I_{3x3}. \tag{47}$$

The terminal cost weighting matrix, Q_f, was calculated using MATLAB's *dlqr* function, which is the solution to the discrete algebraic Riccati equation. The solution depends upon the discretized plant matrices as well as the Q and R matrices.

The final cost scalars for each instance and phase are in Table 8. Q and R selection was a significant factor in ensuring the solution's convergence and had to be balanced against the simulation time.

Table 8. MPC Controller Settings

Variable	**Name**	**LEO**	**GEO**
N_t	Prediction Horizon	10	25
$Delta$	Sampling Time	20	60
$Q1$	State Cost Scalar, 1	1e3	1e2
$R1$	Control Cost Scalar, 1	1e1	1.5e0
$Q2$	State Cost Scalar, 1	1e1	1e2
$R2$	Control Cost Scalar, 1	1.5e1	1e2
$Q3$	State Cost Scalar, 1	1e3	1e3
$R3$	Control Cost Scalar, 1	1e2	1e4

Tuning each parameter by hand for each phase was time-intensive. Automatic tuning methods do exist in literature for other applications, and could be of use in this area [49].

4.1.4 Demonstration of Slack Variables

In order to test the use of slack variables in MPCTools, a single-phase scenario using the GEO configuration was simulated. Notional initial and final states were chosen for testing purposes only. In the first instance, both path constraints were made soft. The result is in Figure 6 below.

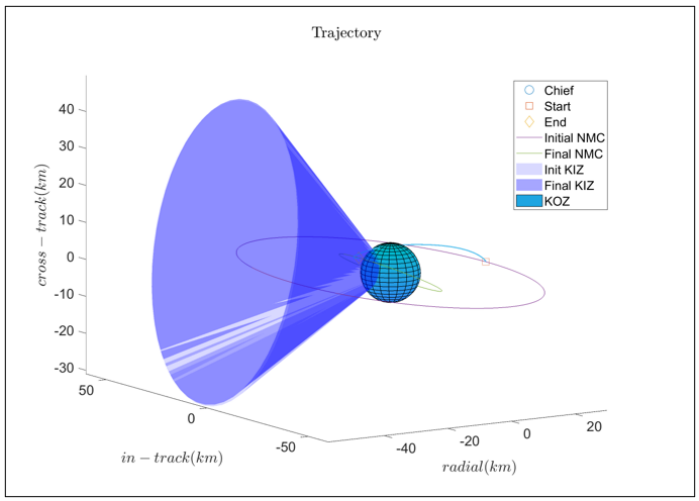


Figure 6. Single-Phase Maneuver with Two Slack Variables

The deputy clearly violates both constraint zones in order to arrive at its destination inside the KIZ. In the second instance, only the KIZ constraint was slackened. The spherical KOZ about the chief was hardened. This was accomplished in the code by changing the number of slack variables from two to one.

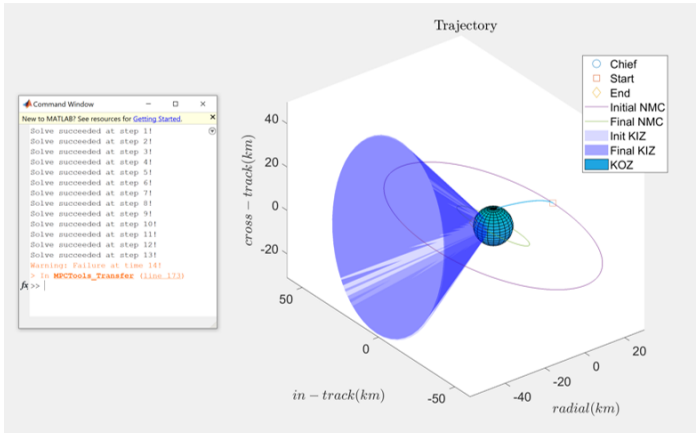


Figure 7. Single-Phase Maneuver with One Slack Variable

It can be seen in Figure 7 that the deputy was unable to complete the maneuver because it violated the KOZ constraint. The operation was terminated after 14 time steps. Thus, recursive feasibility was lost.

For both the LEO and GEO test instances, the constraints are softened in individual phases by assigning them a slack variable, s. Table 9 reiterates the phases with their respective slack variables.

Table 9. Constraint Definition for Each Phase

Phase	KIZ	KOZ	Number of Slacks
1	Soft	Hard	1
2	Hard	Hard	0
3	Soft	Hard	1

The value of s is optimized by MPCTools. Most importantly, it allows for violation of the constraint zone, allowing the deputy to pass in and out of it without requiring additional phase definitions. This will be examined more in Section 4.3. In these cases, the KOZ was kept as a hard constraint. As the results will show, doing so creates a de facto built-in safety feature of the controller.

MPC should generally be robust to constraint enforcement while still ensuring optimality. However, several factors can still render the controller unable to solve, as seen in the previous instance. Available control effort, Q and R tuning, horizon length, and more can all impact the feasibility of the problem. For example, a high R matrix will prioritize reducing control effort at the cost of meeting the state or trajectory requirements. Alternately, a large Q will increase the control effort in order to meet the final state. However, the available control effort is often limited physically, meaning that some cost values will require unobtainable controls. Another danger to this is placing the spacecraft on a trajectory that cannot recover if it encounters constraints, especially if these constraints are outside of the current prediction horizon. Time step is also important, as the open-loop control solution is found for each step. If a step is too large, important information could be overlooked.

In general, it is best to have a time step much smaller than the prediction horizon, with enough steps to meet constraint requirements without becoming prohibitively costly.

4.2 Results

This section provides the resulting trajectories, control effort, and computation times for the LEO and GEO test instances. Additionally, an analysis is performed of various weightings of the slack variables.

4.2.1 Trajectories

The resulting trajectories were plotted along with the constraint zones, any reference trajectories in the case of tracking, and the deputy's starting and ending points for clarity. Additionally, each phase was analyzed to ensure that the constraints were enforced.

For the constraint enforcement plots in Figures 9, 11, 14, 18, 20, and 23, the value of the KIZ and KOZ constraints were calculated and determined to be either less than zero in the case of hard constraints, or less than the computer-optimized value of s in the case of soft constraints. The value of zero or s is shown as a solid line, while the calculated constraint value is shown by the dashed line, where,

$$KIZ = \cos\theta - \hat{v}_a^T \hat{v}_s, \tag{48}$$

and,

$$KOZ = r - \|\vec{v}_a\|. \tag{49}$$

Recall that the KIZ constraint requires the angle between the relative distance vector of the chief and the sun vector to be less than the half-angle of the cone. The KOZ constraint states that the magnitude of the distance of the deputy from the chief must be outside of the designated KOZ radius. The following pages give the plots of the trajectory in the LEO instance. Each phase is presented with the plot of its constraint enforcement. In the case of phase two, a planar plot of the teardrop is also provided for clarity.

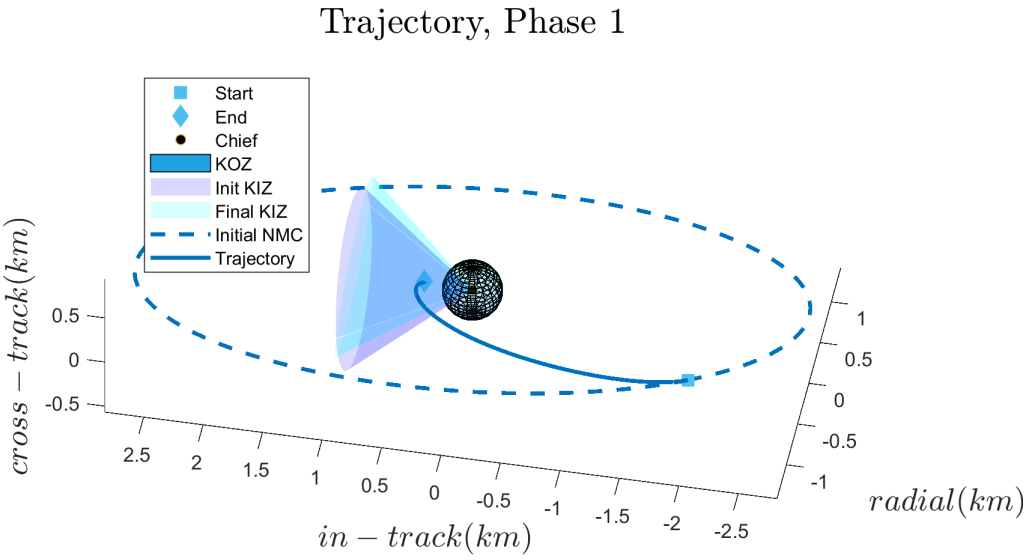


Figure 8. Trajectory (LEO,Phase 1)

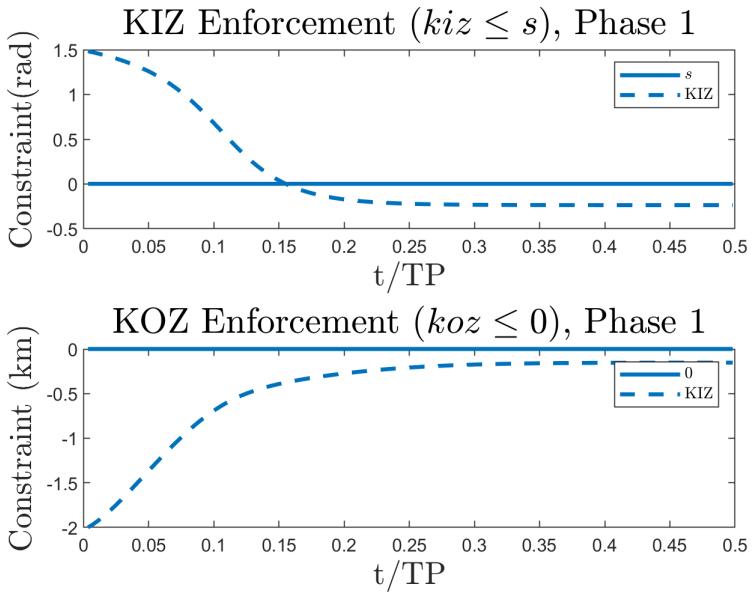


Figure 9. Constraint Enforcement (LEO,Phase 1)

Trajectory, Phase 2

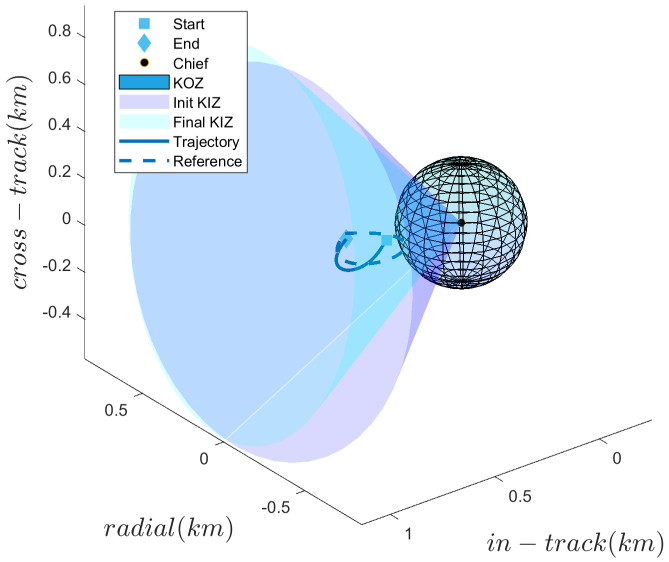


Figure 10. Trajectory (LEO,Phase 2)

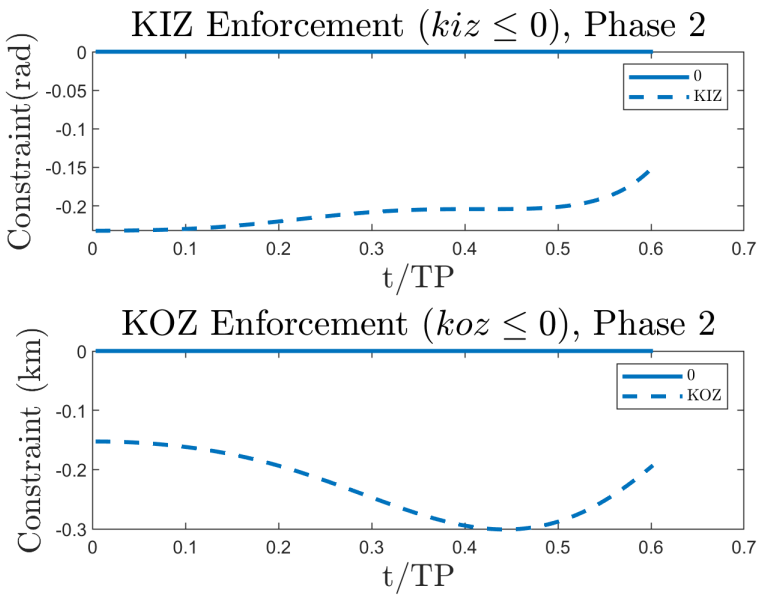


Figure 11. Constraint Enforcement (LEO,Phase 2)

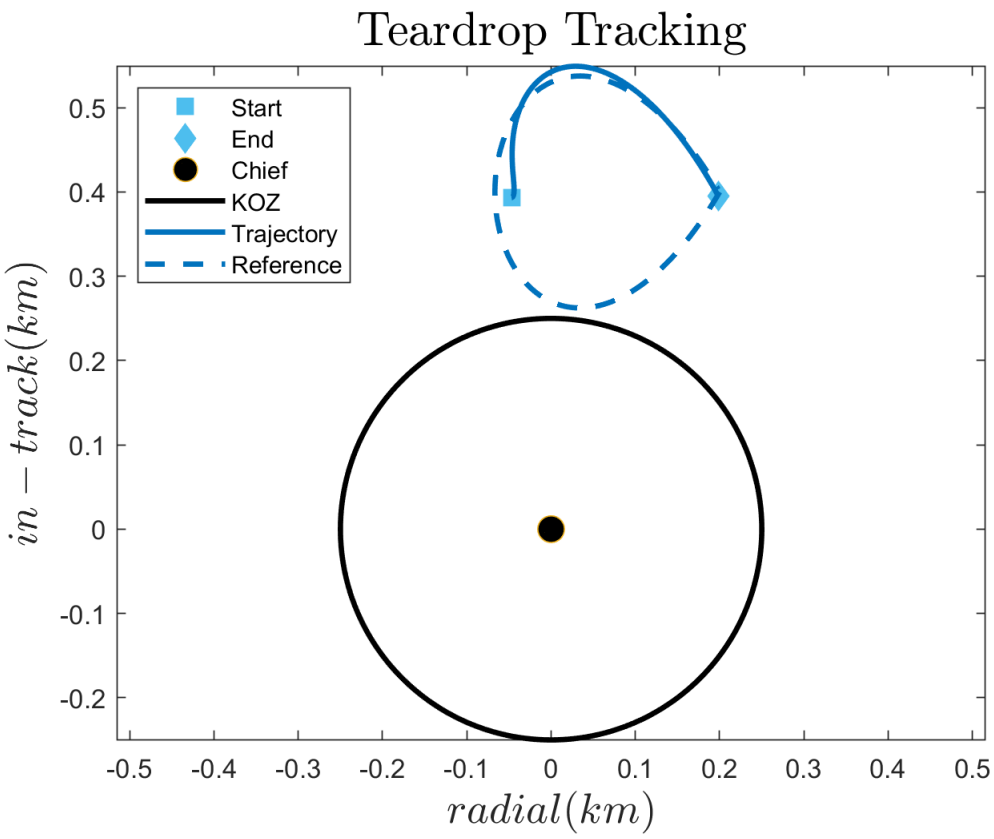


Figure 12. Planar View (LEO, Phase 2)

Phase two of this maneuver appears to be less than optimal. Significant tuning and manipulation was required to produce a teardrop orbit that did not violate the constraints. It still needed to allow some meaningful time for observation. Clearly this phase was the most challenging on constraint enforcement, due to trying to keep the deputy stationary above the chief, maintain illumination, and manage control costs. It does not appear that a teardrop maneuver was ideal in this case, and another formation could be considered in future work, such as leader-follower [18].

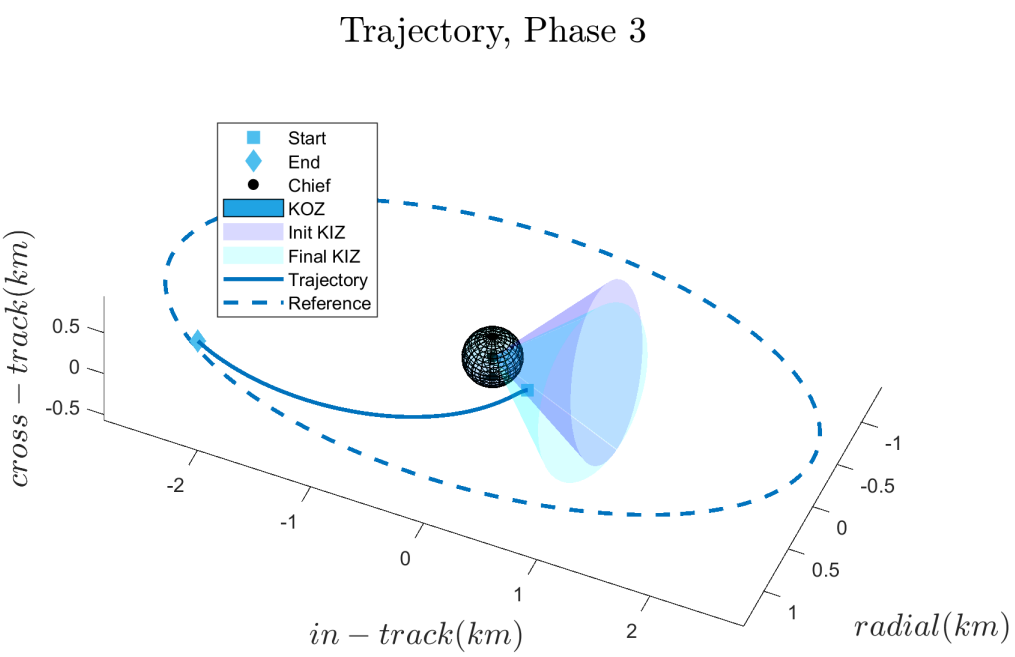


Figure 13. Trajectory (LEO,Phase 3)

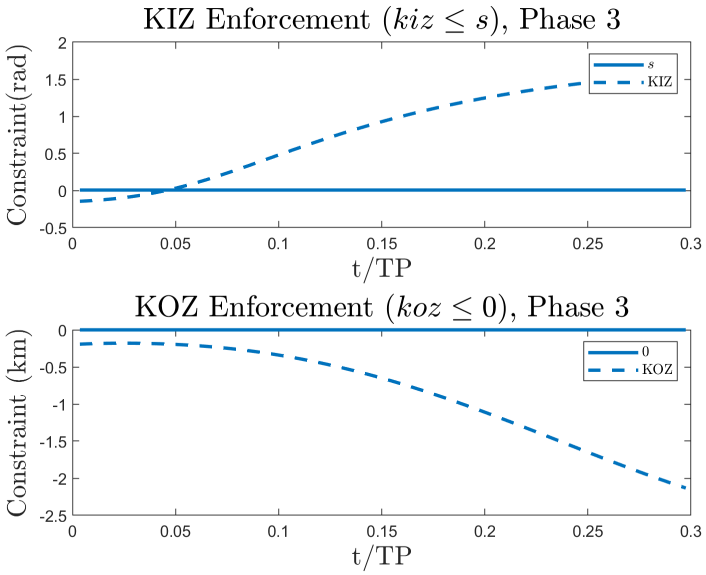


Figure 14. Constraint Enforcement (LEO,Phase 3)

Figures 8 through 13 show that the deputy behaved as expected with respect to the constraints. In all phases, the deputy enforced the KOZ constraint with no violations. The violations of the KIZ in phases one and three were intentional as a result of the slack variables. Phase one begins in violation of the KIZ constraint and ends inside the KIZ. Once inside the KIZ, the constraint is hardened for phase two. The violation in phase one is seen in Figure 9 as the constraint value crosses the line marked by s, which although close to zero was a small positive number. In phase three, the deputy begins in compliance with the KIZ constraint and ends in violation of it.

The overall trajectory for the LEO case is plotted in Figure 15.

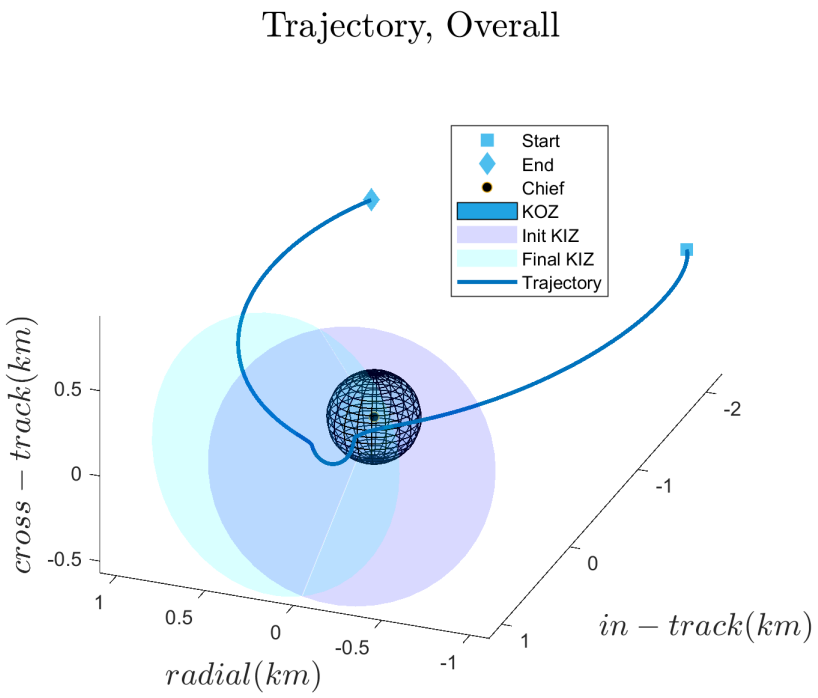


Figure 15. Trajectory (LEO,Overall)

The movement of the sun vector cone is quite prominent in the LEO case. This was expected, as the chief's period is less than two hours compared to the GEO scenario's period of almost a day. This makes constraint enforcement more challenging, but here MPC demonstrates some inherent robustness in being able to adjust to the changing environment while still meeting its desired final state. None of the constraints were violated.

The overall trajectory in the GEO case, seen in Figure 16, is similar to the LEO case. However, the relative distances are larger and the cone does not move as significantly throughout the maneuver. Additionally, the teardrop formation appears to be better defined.

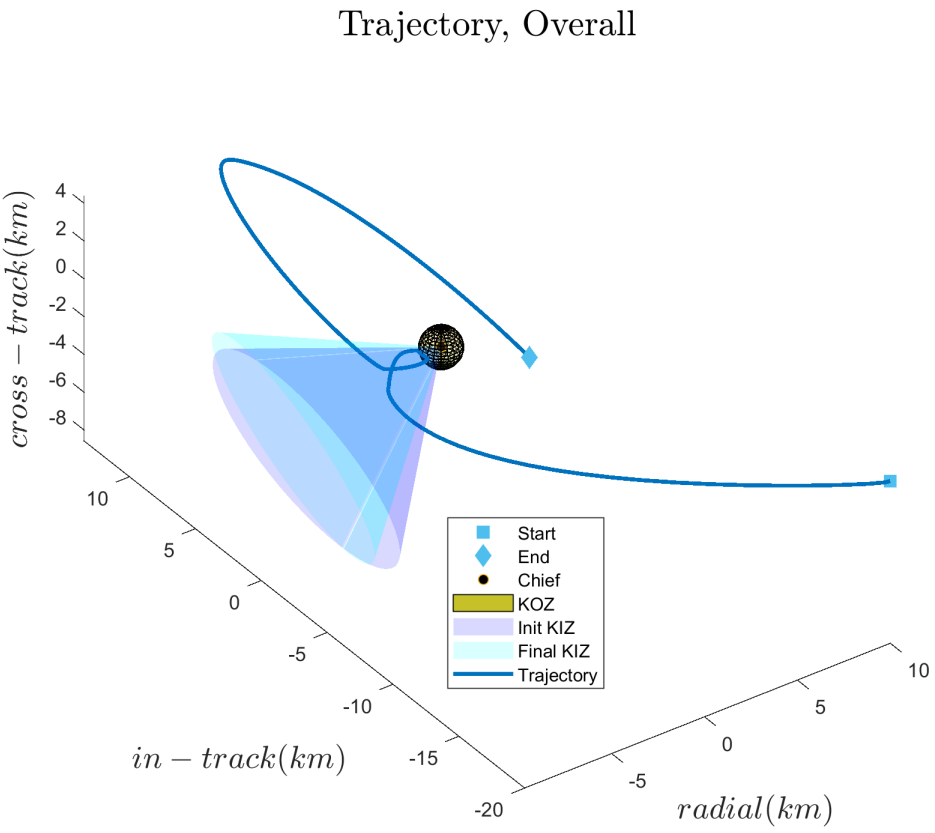


Figure 16. Trajectory (GEO,Overall)

The trajectories and constraint enforcement for each phase of the GEO case are shown below. The teardrop in phase two is again plotted in-plane for clarity.

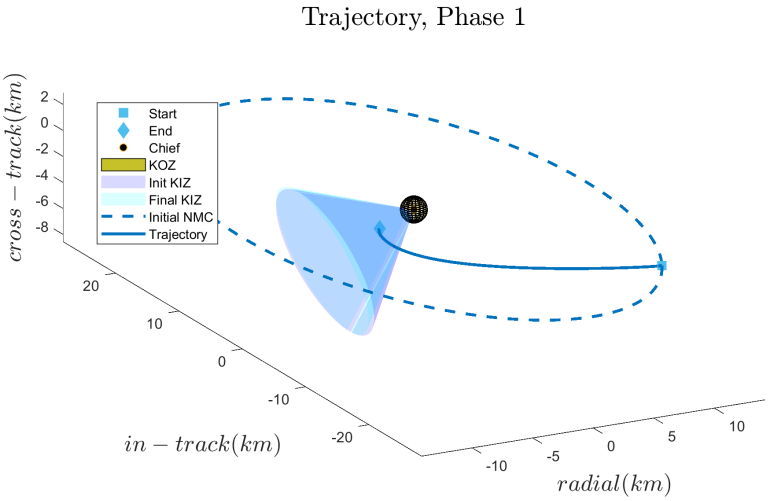


Figure 17. Trajectory (GEO, Phase 1)

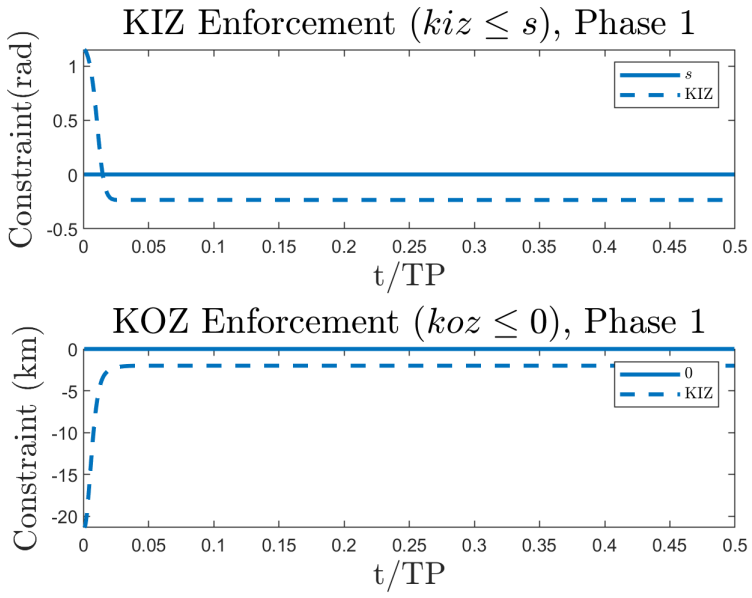


Figure 18. Constraint Enforcement (GEO,Phase 1)

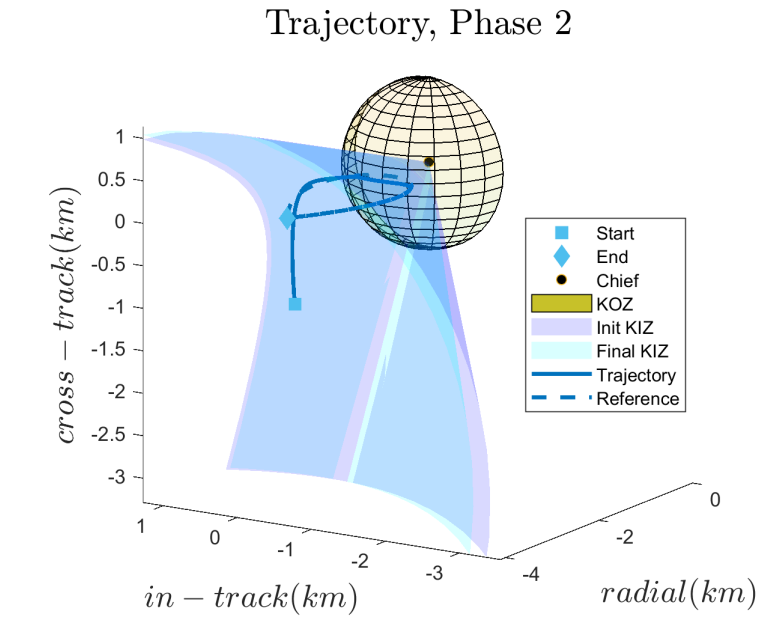


Figure 19. Trajectory (GEO,Phase 2)

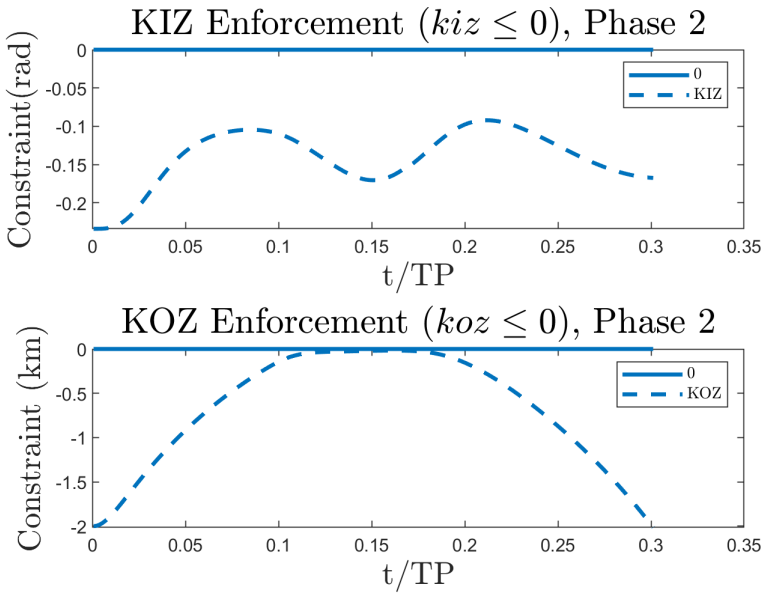


Figure 20. Constraint Enforcement (GEO,Phase 2)

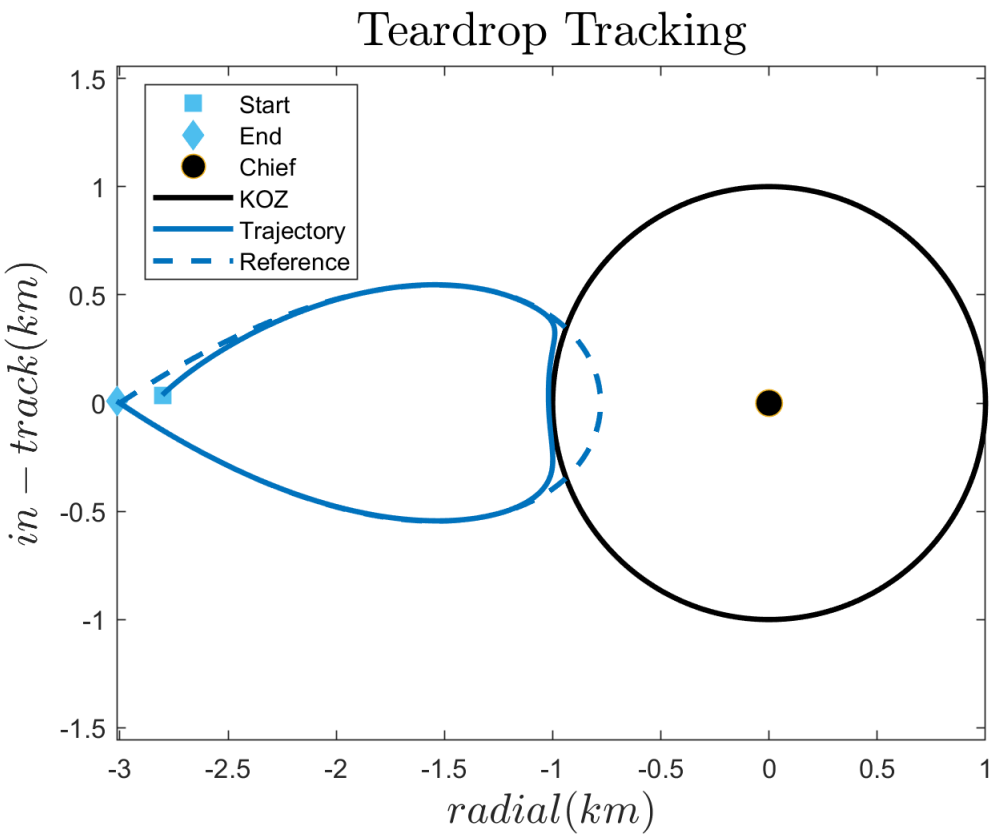


Figure 21. Planar View (GEO, Phase 2)

Unlike with the LEO example, the slower movement of the KIZ cone at GEO enables ample room for the teardrop maneuver. It appears to have good alignment with the sun vector, ensuring that the chief will be illuminated. In Figure 19, it appears as though the trajectory makes contact with the boundary of the KOZ. To better see whether a constraint violation has occurred, it is easier to look at the planar plot in Figure 21. Clearly, even though the reference trajectory violates the KOZ, the controller does not, and adjusts the trajectory accordingly. This is also corroborated in 20, where the difference between the KOZ radius and the deputy's relative distance approaches zero, but does not become positive.

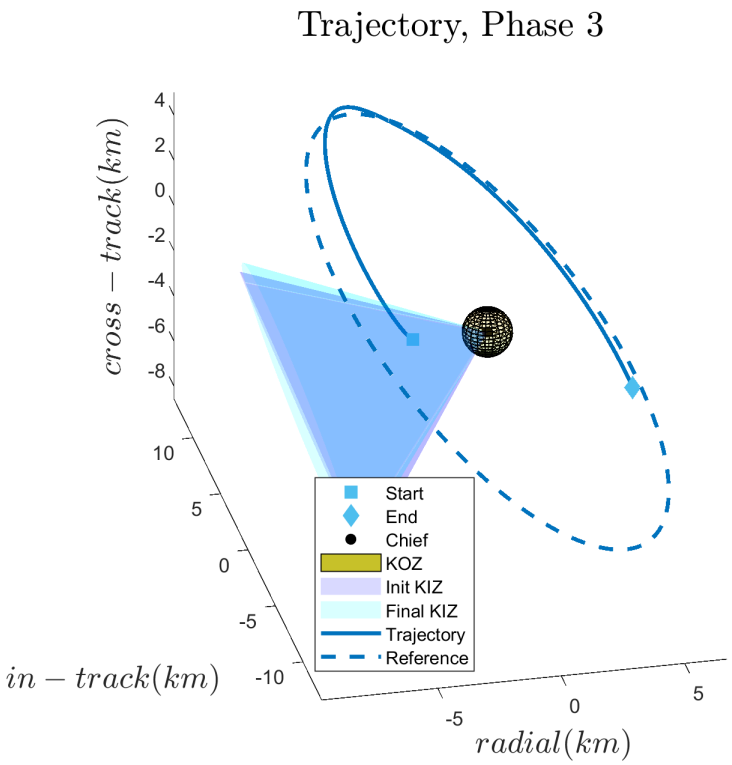


Figure 22. Trajectory (GEO,Phase 3)

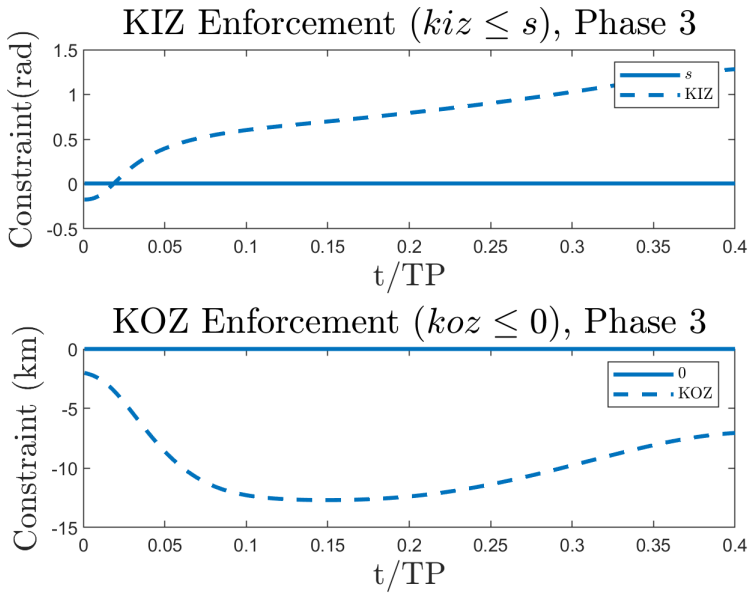
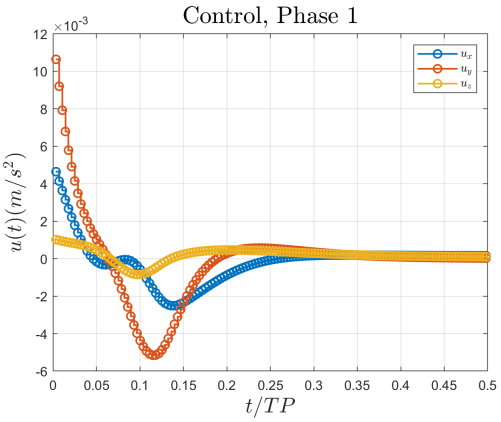


Figure 23. Constraint Enforcement (GEO,Phase 3)

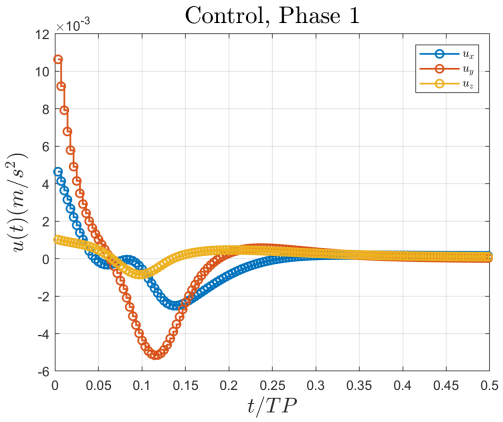
MPC again shows in the GEO instance that, while MPC is not adaptive, it does have some built-in robustness to constraint enforcement. It successfully avoided the KOZ, demonstrating passive safety qualities. The trajectory for GEO was easier to define and tune.

4.2.2 Control Effort

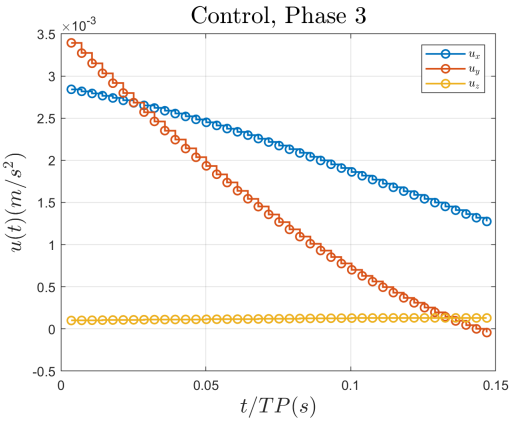
Overall control effort was found by calculating the magnitude of the control vector for each phase of each test instance. The control input in this problem is given in terms of acceleration, with units of m/s^2. Newtons are not used as the mass of the deputy is not considered in this work. The control effort for each phase of the LEO instance is in Figure 24 below.



(a) Phase 1



(b) Phase 2

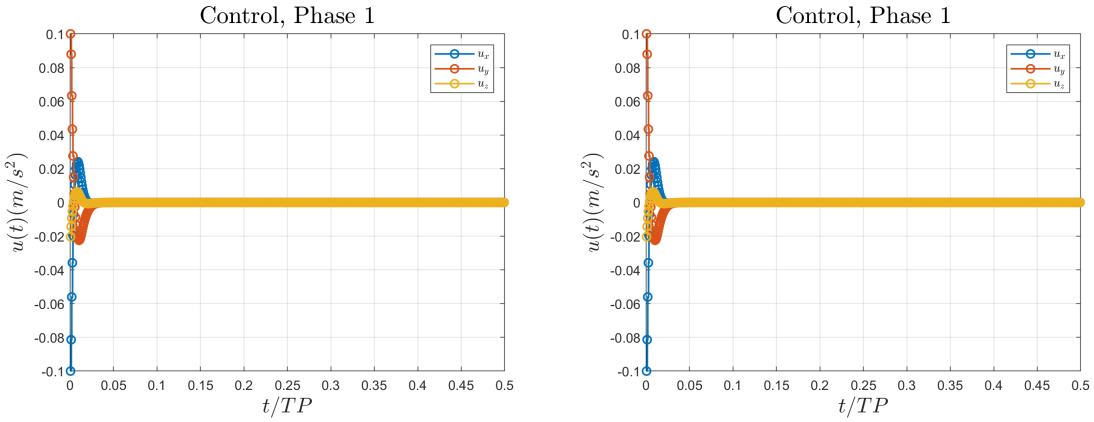


(c) Phase 3

Figure 24. LEO Control Effort

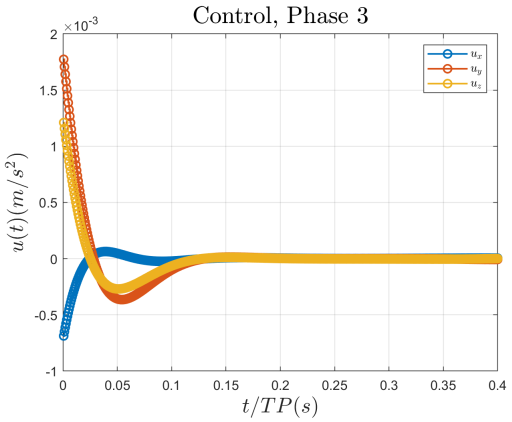
The magnitude of the control effort is very small, which speaks to the benefits of using unforced natural motion where possible. Unfortunately, phase 3 does not meet its final goal of a return to natural motion. The control efforts do not converge to zero, implying some instability in the final maneuver. This could be impacted by tuning, final state, and prediction time. However, the control bound remains unsaturated, which means corrected station-keeping would be an option in a real-life scenario.

Likewise, the control effort for each phase of the GEO instance is in Figures 25 below.



(a) Phase 1

(b) Phase 2



(c) Phase 3

Figure 25. GEO Control Effort

The magnitudes of the control efforts at GEO are much larger in phases one and two. This makes sense, as the plant matrix is dependent upon time and the chief's mean motion. The longer times and larger distances will natural require more control. There is some

saturation of the control in the first two phases, but eventually the effort in all three phases converges to zero. This means that the deputy successfully returned to natural motion in each case.

Table 10. Control Effort Comparisons

Phase	**LEO** (m/s^2)	**GEO** (m/s^2)
1	0.01824	2.8298
2	2.03713e-05	0.0028
3	0.0022	0.0049

Table 10 compares the magnitude of the control effort for each test instance. This confirms what the control plots indicated, which is that the overall control effort tends to be larger at higher orbital regimes. However, the magnitude of the effort is still quite small.

Another metric considered was the impact of the weighting of the slack variable on the control effort and final state. To do this, the simulations for LEO and GEO were performed for various values of s, and the magnitude of the control effort and errors in the final state were calculated. Results are in Table 11. It was expected that the control effort should increase as the weighting of s increases. As the controller creates more distance between the constraint boundary and the deputy, it will also fight harder to meet the final state requirements. Eventually, it is expected that the weighting of s will be too high for the deputy to reach its final state.

Table 11. Control Effort, Computation Time Comparisons for Slack Weights

LEO	**Control Magnitude (m/s^2)**		**Computation Time (s/step)**	
Weight of s	**Phase 1**	**Phase 3**	**Phase 1**	**Phase 3**
0.1	0.01263	0.0022	0.03137	0.02898
1	0.01824	0.0026	0.02809	0.06014
10	0.3882	0.0118	0.025889	0.02824
100	2.4794	0.0090	0.02811	0.02829
1000	5.1509	0.0071	0.02703	0.02803
10000	7.5980	0.0067	0.02520	0.04254
GEO	**Control Magnitude (m/s^2)**		**Computation Time (s/step)**	
s	**Phase 1**	**Phase 3**	**Phase 1**	**Phase 3**
0.1	2.1057	0.0049	0.0450	0.04785
1	2.8298	0.0049	0.0536	0.05767
10	6.2810	0.0046	0.1231	0.1353
100	15.5438	0.0047	0.1015	0.1311
1000	23.3842	0.0046	0.1162	0.1296
10000	29.0037	0.004533	0.04796	0.05693

Computation times did not appear to be significantly impacted by changing the weighting of the slack variable. These times are very fast, and much smaller than the time between steps. This is more evidence that MPC is a good candidate for on-board applications.

In phase one of each instance, the expected increase in the control effort occurred. It is apparent that, as the weighting of the slack variable increased, the deputy had to exert more effort to enter the KIZ and meet its final constraint. An analogy of this concept is a pillow on a hard surface. The thicker the pillow, the farther the user is from a firm floor. If they were to try to push through the pillow's stuffing to the floor beneath, the force required would depend in part upon the pillow's thickness.

However, in the case of phase three, the control effort actually began to decrease. To see why the effort increased in phase one and decreased in phase 3, the plots of the trajectories are compared below for the LEO instance. First is phase one in Figure 26.

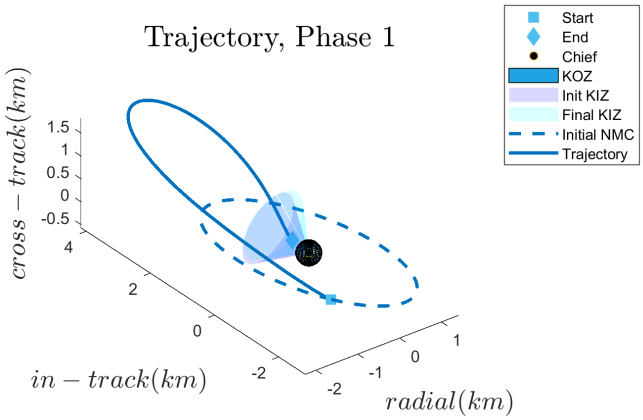


Figure 26. LEO Orbit with Slack Weighted 10,000x More than Costs

It is apparent that the buffer between the trajectory and the constraints becomes larger and larger as the control effort is put into maintaining the constraints. Eventually, however, the controller still forces the deputy to its final state, but with a significant increase in the control magnitude as shown above.

The opposite effect occurs when the deputy tries to escape the KIZ in phase three with a very high slack weighting. Eventually, it becomes too difficult and the deputy remains trapped inside the cone, close to the surface. The controller is driving it to the final state, but the high slack value prevents it from getting there. A figure of the LEO Phase 3 trajectory for $10,000 * s$ is in Figure 27 below.

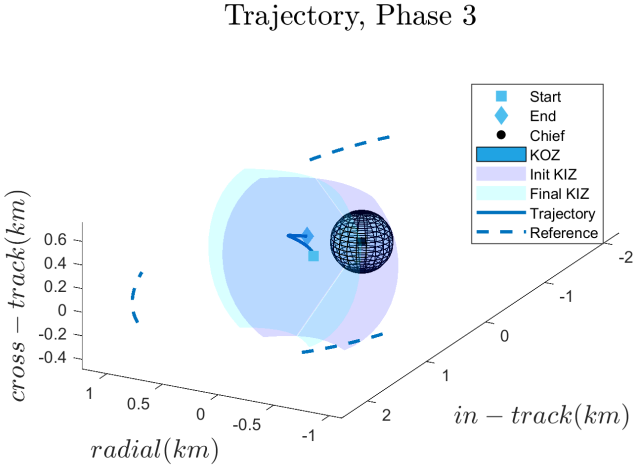
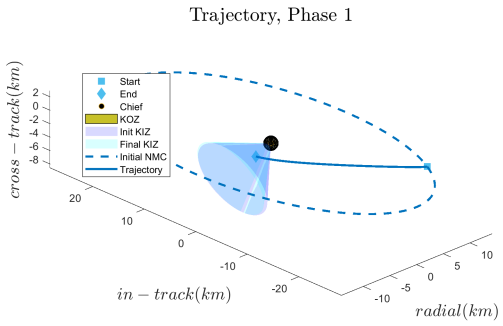


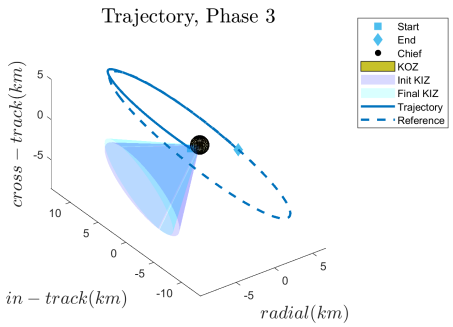
Figure 27. LEO Orbit with Slack Weighted 10,000x More than Costs

It is clear that such a high weighting of the slack reduced the available control and state costs until neither tracking nor terminal constraints were achievable. In a similar vein, the figures below show the results of low and high slack variable weighting of the GEO example for phases one and three.

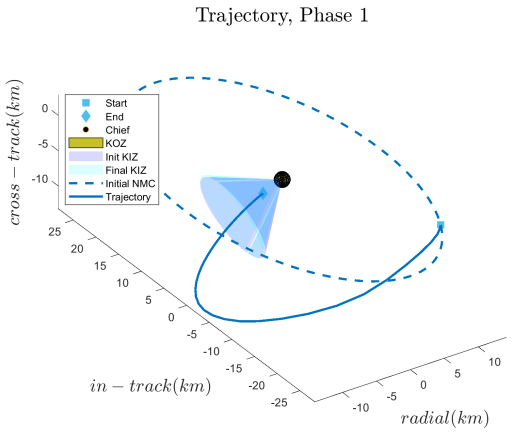
The cases are shown for a weight, $k_s = 0.1$ and for $k_s = 10,000$. As with the LEO example, the high weighting made a dramatic impact on the trajectories. The impact was dependent upon whether the deputy was trying to enter or exit the constraint zone. With a low slack weighting, the trajectory performed very similarly to how it normally would, simply putting less emphasis on constraint enforcement.



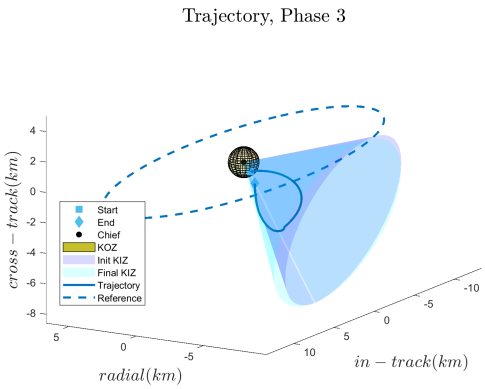
(a) GEO, Phase 1, Slack 0.1x



(b) GEO, Phase 3, Slack 0.1x



(c) GEO, Phase 1, Slack 10000x



(d) GEO, Phase 3, Slack 10000x

Figure 28. GEO Case Slack Weighting Comparison

A more detailed study of weighting and tuning for slack variables would be helpful. The figures for low slack weightings raise the question of whether a very low slack cost could still enable phase transitions with a lower cost, but maintain constraint enforcement. This will be a recommendation for further study.

4.2.3 Computation Costs

Even though this is a standard NLP with non-convex constraints, the computation time of MPC is very fast. This is again attributed to the finite prediction horizon. The

computation costs for each test are given in Table 12 below.

Table 12. Computation Times

	Phase	MPCTools (s)	Std Deviation
LEO	**1**	0.02809	0.0030
	2	0.0180	0.0014
	3	0.02771	0.0087
GEO	**1**	0.0536	0.0058
	2	0.0547	0.0089
	3	0.05767	0.0037

The low standard deviation is assurance that the solution performance is consistent throughout each phase, in spite of the time varying constraint. As an added precaution, the controller did have logic built in to skip the current computation step if it was taking too long. However, because the size of the control vector was one larger than the state vector, with no empty cells, it is clear that this logic was never used.

4.3 Comparison with Literature

This section compares the methodology of this thesis to the work of Becker [9], who performed a 6-phase, 6-DoF maneuver using MPCTools. For a one-on-one comparison, slack variables were implemented into his code and the results were compared. Unlike the test instances in Section 2.3, the focus here is more on the phase transition than the time-varying constraint.

4.3.1 Original Problem

Becker's scenario was a satellite inspection mission with twelve constraints. Ten of these were path constraints and two were boundary constraints. The path constraints were further categorized into rotational and translational requirements. The rotational constraints

exist to satisfy Field-of-View (FOV) and Line-of-Sight (LOS) operating requirements. His constraints are shown in Figure 29.

Number	Description	Constraint Equation	Type
1	30m Range	$0.03 - \|\vec{\rho}\| \leq 0$	KOZ
2	1km Range	$1 - \|\vec{\rho}\| \leq 0$	KOZ
3	1.1km Range	$\|\vec{\rho}\| - 1.1 \leq 0$	KIZ
4	80m Range	$0.08 - \|\vec{\rho}\| \leq 0$	KOZ
5	120m Range	$\|\vec{\rho}\| - 0.12 \leq 0$	KIZ
6	Camera FOV out of Sun	$\hat{v}_b \cdot \hat{v}_s - \cos(10°) \leq 0$	KOZ
7	Solar Panels in Sun	$\cos(45°) - \hat{v}_{sp} \cdot \hat{v}_s \leq 0$	KIZ
8	Deputy out of Chief Mission FOV	$\frac{\vec{p}}{\|\vec{\rho}\|} \cdot \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} - \cos(8.7°) \leq 0$	KOZ
9	FOV Requirement	$\cos(10°) - \hat{v}_b \cdot \frac{\vec{p}}{\|\vec{\rho}\|} \leq 0$	KIZ
10	Line of Sight (LOS) Requirement	$\cos(30°) - \frac{\vec{p}}{\|\vec{\rho}\|} \cdot \hat{v}_i \leq 0$	KIZ
11	Inspection Velocity Limit	$\|\vec{v}\| - .002 \leq 0$	Boundary
12	Inspection Angular Velocity Limit	$\|\vec{\omega}\| - 0.02 \leq 0$	Boundary

Figure 29. Becker Constraint Definitions

Becker implemented DMPC, which solved side-by-side in MPCTools. His mission took place in six phases: approach, positioning, inspection of one side, re-position, inspection of another side, and depart. The constraints for each phase are summarized in Figure 30

Phase	Desired State	Constraints	Exit Criteria
1	Equation 3.40	2, 6, 7	$\vec{p} \leq 1.05km$
2	Equation 3.40	1, 3, 6, 8	Constraints 4, 5, 9, 10, 11, 12 satisfied
3	Equation 3.40	4, 5, 6, 8, 9, 10, 11, 12	Phase 3 duration is 600s.
4	Equation 3.42	1, 3, 6, 8	Constraints 4, 5, 9, 10, 11, 12 satisfied
5	Equation 3.42	4, 5, 6, 8, 9, 10, 11, 12	Phase 5 duration is 600s.
6	Equation 3.44	1, 3, 6, 8	$\vec{p} \geq 1.02km$

Figure 30. Becker Phase Definitions

The exit criteria for the phases are crucial to ensuring that the next phase can begin

within the constrained area. There is no feasible solution for a problem that begins in violation of a constraint. With hard constraints, this is straightforward. However, as previously discussed, keeping the constraints hard necessitates additional phase-building. This is discussed more in the next section.

4.3.2 Phase Transition Implementation

Slack variables were implemented into this scenario for comparison of the original results, which were also validated using GPOPS-II. Thus, this method can be compared to the open-loop OCP. In order to implement slack variables, each constraint had to be evaluated for its function. The boundary constraints, 11 and 12, were not modified, as they are upper bounds for the velocity and angular velocity. These are necessary for safety and constraint enforcement.

There are two main priorities during the inspection phase: keep the deputy's camera from pointing into the sun, and avoiding interruption of the chief's mission. These are manifested in two pointing constraints, labeled in Table 13 as the FOV KOZ and the Chief KOZ, respectively. They are kept as hard constraints for the entire mission.

In his constraint definitions, Becker states that the overlap between constraints in phases two and three allow for the phases to transition. The next phase cannot begin if the current phase is already in violation of that constraint. Softening the constraint allows the next phase to begin even if the constraints are violated. Section 4.3.3 will analyze whether this incurs additional computational or control cost.

Additionally, softening some of the constraints actually renders the overlaps redundant. Using slack variables, these phases could be modified further to eliminate extraneous constraints, and even consolidate phases. For example, had the scenario in Section 4.2 been designed using hard constraints, it would have required at least one additional phase for entry into the KIZ. Using the slack variables, the hard constraints were activated while the deputy was within a certain distance of its final desired state, allowing a smoother transition.

Table 13 gives the final assignments of the constraints for each phase, including their hardness or softness.

Table 13. Soft Constraint Assignments by Phase

	Range KOZ (1,2,4)	**Range KIZ (3,5)**	**FOV KOZ (6)**	**Solar KIZ (7)**	**Chief KOZ (8)**	**FOV KIZ (9)**	**LOS KIZ (10)**
1	H (2)	–	H	H	–	–	–
2	**H (4)**	**S (5)**	H	–	H	**S**	**S**
3	H (4)	H (5)	H	–	H	H	H
4	**H (4)**	**S (5)**	H	–	H	**S**	**S**
5	H (4)	H (5)	H	–	H	H	H
6	H (1)	H (3)	H	–	H	–	–

The primary change was softening KIZ constraint five and adding it to the second and fourth phases. This drives the controller in the direction of meeting this constraint without being held to it absolutely. Ideally, this process should set the controller up in a way that it is within the constrained area when the constraints become hard in phases three and five. The same concept was applied to the FOV KIZ constraint and the LOS KIZ, which are necessary in phases three and five for maintaining a line of sight on the chief and adhering to the NADIR-pointing field of view.

The largest problem with this process, however, is a potential feasibility issue when hardening the constraint in the subsequent phase. The process worked without issue when softening the constraints in phase two, allowing the model to transition into phase three. However, from phases four to five, the controller initially would not solve. It was found that the final state in phase four met the constraints. However, this did not ensure that the deputy was inside of the constraint zones for phase five. In order to overcome this, a hard terminal constraint had to be enforced in phase four, that the deputy had to meet the final

state before moving to the next phase.

This proved to be a limitation of transitioning into hard constraints using slack variables. With slack in the prior phase, there is no guarantee of being in a feasible space for the following phase. Adjusting the weights could also address this, but still doesn't provide the guarantee.

4.3.3 Comparison Results

The results of the simulation with the softened constraints are compared here to the original results, as well as results obtained from the validated GPOPS-II solution. Since the DMPC controller is effectively two controllers, albeit with some coupling between the constraint and exit criteria functions, the control efforts and computation costs are given for each controller, referred to as the Attitude and Trajectory controller, respectively.

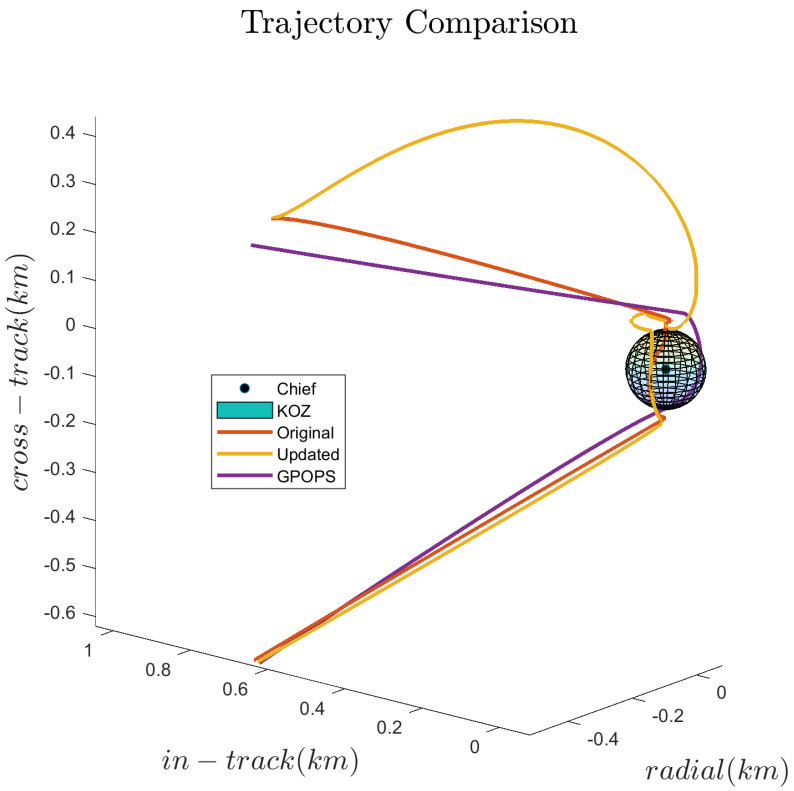


Figure 31. Trajectory Comparison

The resulting trajectories in Figure 31 show the original results with hard constraints only, updated results with softened constraints, and the GPOPS-II results. Phase one is omitted for scaling purposes, and because it is not as relevant to constraint enforcement. The KOZ pictured is the 80-meter KOZ enforced in phases three and five during the inspection of the +z and -z sides of the chief, respectively. All three methods demonstrate that the constraints were enforced during inspection. The updated trajectory, shown in yellow, does appear to depart from the other two results on approach and during the transition between inspections. This is most likely an issue with tuning. However, it is also likely that the slack variables from phase two incurred additional cost and produced less emphasis on state and control optimization, leading to a less streamlined trajectory. The rest of the results seem to align very well, however.

Table 14. Control Effort for Hard vs Soft Constraints

Phase	Controller	Original	Updated	GPOPS-II
1	**Torque (Nm)**	8.8553e+03	4.2006e+03	6.5198e-05
	Thrust (kN)	0.0321	0.0321	8.7073e-05
2	**Torque (Nm)**	1.0266e-09	172.2132	4.9895e-05
	Thrust (kN)	0.0106	0.0262	4.7992e-05
3	**Torque (Nm)**	1.2260e-09	153.0475	7.1214e-05
	Thrust (kN)	0.0012	4.4809e-05	7.7284e-05
4	**Torque (Nm)**	1.4031e+04	2.5708e+04	3.9153e-05
	Thrust (kN)	6.5232e-04	0.0021	5.2555e-05
5	**Torque (Nm)**	3.0372e-05	2.6713e+04	8.8189e-05
	Thrust (kN)	6.8572e-05	0.0202	9.1120e-05
6	**Torque (Nm)**	3.0267e+03	3.0256e+03	4.7272e-05
	Thrust (kN)	0.0189	0.0187	5.4850e-05

Table 14 shows the computed magnitudes of the control effort for the old and new

methods. The magnitudes were found by integrating the squared components of each control vector using MATLAB's *trapz* function. Unfortunately, the updated method with slack variables shows a significant increase in the control effort for phases two, three, and five. Certainly, this work has already shown that the introduction of slack variables can cause an increase to the control effort. Tuning is likely a large factor in the difference between results. Phases three and five required a large scaling of the Q matrix in order to obtain results for the terminal cost, Q_f, via the algebraic Riccati equation. This is likely due to nonlinearities in the attitude dynamics. By comparison, the magnitudes of control from GPOPS-II are very small. This is understandable, as the infinite-horizon solver can see the solution in its entirety and apply the absolute minimal control. An increase to the control effort for soft constraints is not unexpected, but the magnitude of the increase indicates that slack variables may have limitations as a means of phase transition.

Table 15. Computation Times for Hard vs Soft Constraints

Phase	Controller	Original (s/step)	Updated (s/step)	GPOPS-II (S)
1	**Attitude**	0.1953	0.1328	2.5964e+03
	Trajectory	0.1757	0.1771	”
2	**Attitude**	0.1706	0.2746	5.5931e+03
	Trajectory	0.2381	0.6222	”
3	**Attitude**	0.3846	0.2808	6.3408e+03
	Trajectory	0.2891	0.4774	”
4	**Attitude**	0.2187	0.3065	8.3717e+03
	Trajectory	0.3238	0.4175	”
5	**Attitude**	0.1927	0.2300	1.0373e+04
	Trajectory	0.2701	0.3598	”
6	**Attitude**	0.1950	0.1330	1.1645e+04
	Trajectory	0.1810	0.1428	”

The comparison of computations times in Table 15 gives better news. There is a slight increase in the average computation time per step, mostly in phases two and four. Again, this is to be expected as the cost function must now calculate the slack variable at each horizon and apply it to the path constraint function. However, the time difference is not prohibitively large. This indicates that the use of slack variables in MPC would not be limited by computation cost. The comparison to the cost of GPOPS-II shows the greatest benefit of the MPC controller. GPOPS-II takes significantly longer to solve because it is not broken into finite horizons and discrete steps. This shows that the OCP may produce more optimal results, but finite-horizon methods are much more feasible for on-board computation.

4.4 Summary

In reviewing the costs and potential feasibility issues of this comparison, it seems that there are ways in which slack variables should and should not be used. Manipulating single constraints does not seem to be the best use of this method. Rather, it should be seen as a way to avoid the necessity of transitioning altogether. For example, Becker's scenario could be reformulated to maneuver in and out of constraint zones, only activating the necessary constraints during active inspection phases. If done correctly, phases two and four could perhaps be eliminated altogether, saving significantly on computation costs. If entire paths or phases can be avoided, this also helps save on control which correlates to fuel savings.

The results of this chapter demonstrate the robustness of MPC to time-varying constraints and phase transitions. A scenario involving both concepts was designed and implemented in MATLAB, using MPCTools and CasaDi for the MPC solution. Then, the method was applied to a previous multi-phase study and was shown to improve control effort during phase transitions. Some limitations of the method exist, especially feasibility of the next phase when constraints are hardened. It appears that this method, while showing potential for many applications, is probably best not to be used very close to the boundaries of constraints.

V. Conclusions and Recommendations

The first section of this chapter summarizes the research, results, and significant insights. The section also revisits the research questions. The second part of this chapter will discuss recommendations for future work.

5.1 Conclusions

This work was motivated by the increasing demand for autonomy in space. In the currently congested environment, satellite controls require optimal performance while remaining robust to changing environments. This thesis address the gap in literature that considers time-varying constraint enforcement during multi-phase missions using MPC.

The three primary research questions from Chapter I were:

1. How can an autonomous guidance scheme be applied to a time-varying conical constraint?

2. How can MPC be used to minimize control effort when constraints are activated in multi-phase maneuvers?

3. How does MPC compare to the infinite horizon Optimal Control Problem (OCP)?

Question one was explored by implementing the time-varying KIZ for the RPO mission. Propagating the chief's orbit to calculate the sun vector did incur some additional computation cost, however this is data that would already be needed on-board. It was easier and more efficient in this case to compute ahead of time and store for later use. This did not significantly impact the computation cost. The results showed that MPC was robust to constraint enforcement, obeying the path constraints even when the reference trajectory violated it. Overall, it appears that MPC multi-phase scenarios are well-suited to time-varying constraints, even if they are non-convex.

Question two was the primary focus of the phase transition methodology. Modifying the constraint definitions with slack variables enabled smooth phase transitions between

zones. The phase transitions did still initiate an increase in control effort. The nature of the method means that "predicting" the path using the prediction horizon is still only limited to the current phase. There is no way for the controller to predict the next phase until the current phase is complete. However, the activation or deactivation of constraints without use of additional phases is certainly an advantage of this method.

Question three was answered by comparing the method with validated data from literature. The OCP provided a baseline against which hard phase transitions and slackened phase transitions could be measured. Although still dealing with convex constraints, so the comparison is imperfect, the OCP still provides a picture of the entire trajectory versus the finite-horizon of MPC. The most significant observation from this aspect was that using slack variables solely as a method of phase transition may not always be optimal. Instead, phase planning and constraints should be considered in concert to mutually benefit one another. It is very likely that re-defining the original mission phases in the same way as the controller from Section 4.2 could have provided an even more streamlined, computationally efficient trajectory.

5.2 Recommendations

This work creates a framework for phase transitions using MPC. However, there are many avenues for further study in this area with regard to the dynamics, safety, solution methods, and more.

First, nonlinear dynamic models for more robust modeling and more complicated orbits could be done. Limitations of HCW were discussed in Chapter II, and were made apparent while building the controller. Only nearly-circular orbits could be considered. Singularities appear at periodic intervals in the closed-form solution. Several relative motion models exist which would provide insightful comparison to this problem. A further addition to the dynamic aspect of this study would be perturbations. It would be useful to see how MPC responds to nonlinearities in the controller plant. Lastly, noise is a crucial element to any control system. While some work has been done on filtering with MPC, it would be helpful

to see how it could be combined with phase transitions.

As discussed in Chapter II, there is not much literature on multi-phase MPC for the full 6-DoF scenario. An investigation of phase transitions on attitude constraints is recommended in future work. This leads naturally into considerations for thruster configurations. Additionally, studies on tuning, especially potential auto-tuning methods, would greatly benefit all aspects of this area.

One of the more important aspects of autonomous spaceflight is safety. It has been shown that MPC provides a measure of passive safety. However, there is more than could be done here. How does MPC's built-in constraint enforcement guarantee safety, and by what metrics would it be measured? Is passive safety adequate in most applications, and what redundancies might be required?

Lastly, it is recommended that this work be applied to different solution methods for comparison. In particular, adapting this work to LQ-MPC would open doors for on-board implementation. Doing so would require constraint linearization as discussed previously. A comparison of the linearized constraints to the nonlinear ones in this work would also be helpful in determining the impact of linearization on cost and performance. Further work into solution methods could also have ties to machine learning and more adaptive methods for tuning and trajectory planning.

Overall, the work done in this thesis demonstrates the immense computational cost savings afforded by MPC. This makes it a highly feasible option for on-board implementation. Additionally, its robustness to constraint enforcement gives it a passive safety feature that is highly desirable in the costly, complex, and often unpredictable space environment. Lastly, as methods for phase transitions continue to improve, it is likely that MPC controller design and solution methods will evolve to enable increasingly sophisticated missions, achieving optimality while minimizing costs. Continued work in this field will benefit spaceflight in the future.

Bibliography

1. Mayne, D. Q., "Model predictive control: Recent developments and future promise," *Automatica*, Vol. 50, 2014.

2. ESA, *Space Debris by the Numbers*, European Space Agency Debris Office, 2021.

3. Trump, D., "National Security Strategy 2017," *Foundations of Homeland Security: Law and Policy: Second Edition*, 2017.

4. Magnuson, S., "NEWS FROM SPACE SYMPOSIUM: Tracking Objects in Space Both Easier, More Complicated," *National Defense Magazine*, 4 2019.

5. *National Defense Authorization Act for Fiscal Year 2020*, United States House of Representatives, 2019.

6. Raymond, J. W., "Space Power: Doctrine for Space Forces," *Ussf*, Vol. 51, 2020.

7. Henshaw, C. G., "The DARPA Phoenix Spacecraft Servicing Program : Overview and Plans for Risk Reduction," *Proceedings of 'i-SAIRAS 2014 - The 12th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 2014.

8. Li, Q., Yuan, J., and Wan, H., "Sliding mode control for autonomous spacecraft rendezvous with collision avoidance," *Acta Astronautica*, Vol. 151, 2018, pp. 743–751.

9. Becker, J. A., *Autonomous Constrained Spacecraft Inspection via Model Predictive Control*, Ph.D. thesis, Air Force Institute of Technology, 2020.

10. Eren, U., Prach, A., Koçer, B. B., Rakovic, S. V., Kayacan, E., and Açikmese, B., "Model predictive control in aerospace systems: Current state and opportunities," *Journal of Guidance, Control, and Dynamics*, Vol. 40, 2017.

11. Rawlings, J. B., Mayne, D. Q., and Diehl, M. M., *Model predictive control theory, computation, and design*, Nob Hill Publishing, 2017.

12. Zagaris, C., *Autonomous Spacecraft Rendezvous with a Tumbling Object: Applied Reachability Analysis and Guidance and Control Strategies*, Ph.D. thesis, Naval Postgraduate School, 2018.

13. Walsh, A., Cairano, S. D., and Weiss, A., "MPC for coupled station keeping, attitude control, and momentum management of low-thrust geostationary satellites," *Proceedings of the American Control Conference*, Vol. 2016-July, 2016.

14. Malladi, B. P., Cairano, S. D., and Weiss, A., "Nonlinear model predictive control of coupled rotational-translational spacecraft relative motion," *Proceedings of the American Control Conference*, Vol. 2019-July, 2019.

15. Sun, L. and Huo, W., "6-DOF integrated adaptive backstepping control for spacecraft proximity operations," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 51, No. 3, 2015, pp. 2433–2443.

16. Mauro, G. D., Lawn, M., and Bevilacqua, R., "Survey on guidance navigation and control requirements for spacecraft formation-flying missions," *Journal of Guidance, Control, and Dynamics*, Vol. 41, 2018.

17. Prince, E. R., Hess, J. A., Cobb, R. G., and Carr, R. W., "Elliptical orbit proximity operations differential games," *Journal of Guidance, Control, and Dynamics*, Vol. 42, 2019, pp. 1458–1472.

18. Alfriend, K. T., *Spacecraft formation flying: dynamics, control and navigation*, Elsevier, 2010.

19. Vallado, D. A. and McClain, W. D., *Fundamentals of astrodynamics and applications*, Microcosm Press, 2013.

20. Schaub, H. and Junkins, J. L., *Analytical mechanics of space systems*, American Institute of Aeronautics and Astronautics, Inc., 2018.

21. Sullivan, J., Grimberg, S., and D'Amico, S., "Comprehensive survey and assessment of spacecraft relative motion dynamics models," *Journal of Guidance, Control, and Dynamics*, Vol. 40, 2017.

22. Alfriend, K. T., *Spacecraft formation flying: dynamics, control and navigation*, Elsevier, 2010.

23. Zagaris, C., *Autonomous Spacecraft Rendezvous with a Tumbling Object: Applied Reachability Analysis and Guidance and Control Strategies*, Ph.D. thesis, Naval Postgraduate School, 2018.

24. Zhou, D., Zhang, Y., and Li, S., "Receding horizon guidance and control using sequential convex programming for spacecraft 6-DOF close proximity," *Aerospace Science and Technology*, Vol. 87, 2019.

25. Li, X., Zhu, Z., and Song, S., "Non-cooperative autonomous rendezvous and docking using artificial potentials and sliding mode control," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 233, 2019.

26. Dong, K., Luo, J., Dang, Z., and Wei, L., "Tube-based robust output feedback model predictive control for autonomous rendezvous and docking with a tumbling target," *Advances in Space Research*, Vol. 65, 2020.

27. Buckner, C. and Lampariello, R., "Tube-Based Model Predictive Control for the Approach Maneuver of a Spacecraft to a Free-Tumbling Target Satellite," *Proceedings of the American Control Conference*, Vol. 2018-June, 8 2018, pp. 5690–5697.

28. Lovell, T. A. and Tollefson, M. V., "Calculation of impulsive hovering trajectories via relative orbit elements," *Advances in the Astronautical Sciences*, Vol. 123 III, 2006.

29. Prince, E. R., *Optimal Finite Thrust Guidance Methods for Constrained Satellite Proximity Operations Inspection Maneuvers*, Ph.D. thesis, Air Force Institute of Technology, 2018.

30. Boyarko, G. A., Romano, M., and Yakimenko, O. A., "Time-Optimal Reorientation of a Spacecraft Using an Inverse Dynamics Optimization Method," *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 4, 2011, pp. 1197–1208.

31. Brand, M., Shilpiekandula, V., Yao, C., Bortoff, S. A., Nishiyama, T., Yoshikawa, S., and Iwasaki, T., "A Parallel Quadratic Programming algorithm for Model Predictive Control," *IFAC Proceedings Volumes (IFAC-PapersOnline)*, Vol. 44, 2011, pp. 1031–1039.

32. Cairano, S. D. and Kolmanovsky, I. V., "Real-time optimization and model predictive control for aerospace and automotive applications," *Proceedings of the American Control Conference*, Vol. 2018-June, 2018.

33. Lobo, M. S., Vandenberghe, L., Boyd, S., and Lebret, H., "Applications of second-order cone programming," *Linear Algebra and its Applications*, 1998, pp. 193–228.

34. Weiss, A., Kolmanovsky, I., Baldwin, M., and Erwin, R. S., "Model Predictive Control of three dimensional spacecraft relative motion," *Proceedings of the American Control Conference*, 2012.

35. Weiss, A., Baldwin, M., Erwin, R. S., and Kolmanovsky, I., "Model predictive control for spacecraft rendezvous and docking: Strategies for handling constraints and case studies," *IEEE Transactions on Control Systems Technology*, Vol. 23, 2015.

36. Malyuta, Danylo, and Behcet, "Lossless Convexification of Optimal Control Problems with Semi-continuous Inputs," Nov 2019.

37. Büyükkoçak, A. T. and Tekinalp, O., "Safe spacecraft rendezvous using dual quaternions on time-dependent trajectories generated by model predictive control," *AIAA Scitech 2019 Forum*, 2019.

38. Zagaris, C., Park, H., Virgili-Llop, J., Zappulla, R., Romano, M., and Kolmanovsky, I., "Model predictive control of spacecraft relative motion with convexified keep-out-zone constraints," *Journal of Guidance, Control, and Dynamics*, Vol. 41, 2018, pp. 2051–2059.

39. Keyanpour, M. and Osmanpour, N., "On solving quadratically constrained quadratic programming problem with one non-convex constraint," *OPSEARCH*, Vol. 55, 2018.

40. Sanchez, J. C., Gavilan, F., and Vazquez, R., "Chance-constrained Model Predictive Control for Near Rectilinear Halo Orbit spacecraft rendezvous," *Aerospace Science and Technology*, Vol. 100, 5 2020.

41. Larsén, A. K., Chen, Y., Bruschetta, M., Carli, R., Cenedese, A., Varagnolo, D., and Felicetti, L., "A computationally efficient model predictive control scheme for space debris rendezvous," *IFAC-PapersOnLine*, Vol. 52, 2019.

42. Jin, K., Luo, J., and Geller, D. K., "Effect of navigation and maneuver execution errors on optimal RPO trajectory design," *Advances in the Astronautical Sciences*, Vol. 169, 2019.

43. Jin, K., Geller, D. K., and Luo, J., "Robust trajectory design for rendezvous and proximity operations with uncertainties," *Journal of Guidance, Control, and Dynamics*, Vol. 43, 2020.

44. Rawlings, J. B. and Risbeck, M. J., "MPCTools: Nonlinear model predictive control tools for CasADi," 2016.

45. Jung, J. H., O'Leary, D. P., and Tits, A. L., "Adaptive constraint reduction for convex quadratic programming," *Computational Optimization and Applications*, Vol. 51, 2012.

46. Laiu, M. P., Tits, A. L., and Paul, M., "A Constraint-Reduced MPC Algorithm for Convex Quadratic Programming, with a Modified Active Set Identification Scheme," *Computational Optimization and Applications*, 2018.

47. Jewison, C. and Erwin, R. S., "A spacecraft benchmark problem for hybrid control and estimation," *2016 IEEE 55th Conference on Decision and Control, CDC 2016*, 12 2016, pp. 3300–3305.

48. Schouwenaars, T., De Moor, B., Feron, E., and How, J., "Mixed integer programming for multi-vehicle path planning," 2001.

49. Saunders, D., *A Framework for Autonomous Cooperative Optimal Assignment and Control of Multi-Agent Systems*, Ph.D. thesis, Air Force Institute of Technology, 2021.

50. Wen, C. and Gurfil, P., "Guidance, navigation and control for autonomous R-bar proximity operations for geostationary satellites," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 231, 2017.

51. Mammarella, M., Lee, D. Y., Park, H., Capello, E., Dentis, M., and Guglieri, G., "Attitude control of a small spacecraft via tube-based model predictive control," *Journal of Spacecraft and Rockets*, Vol. 56, 2019, pp. 1662–1679.

52. Boyd, S. P. and Vandenberghe, L., *Convex optimization*, Cambridge University Press, 2018.

53. Wächter, A. and Biegler, L. T., "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, Vol. 106, 2006.

54. Patterson, M. A. and Rao, A. V., "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, 8 2014, pp. 1:1 – 1:37.

55. Andersson, J. A. E., Gillis, J., Horn, G., Rawlings, J. B., and Diehl, M., "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, Vol. 11, No. 1, 2019, pp. 1–36.

56. Wie, B., *Space vehicle dynamics and control*, American Inst. of Aeronautics and Astronautics, 2008.

57. Peat, C., "Satellite Database," 4 2021.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704–0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202–4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
17–06–2021	Master's Thesis	January 2020 — June 2021

4. TITLE AND SUBTITLE

Multi-phase Spacecraft Proximity Operations Using Model Predictive Control

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Bell, Julia C., Capt, USSF

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/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENY-MS-J-094

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

Intentionally Left Blank

10. SPONSOR/MONITOR'S ACRONYM(S)

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

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approval for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The evolving space environment has created a demand for autonomous spacecraft that can maneuver in complex and sometimes contentious environments. Constraint enforcement, such as an avoidance zone to prevent collision with a target, is a key component of autonomous control to ensure safety and performance requirements are met. Finite-horizon Model Predictive Control (MPC) is a popular control method due to its improved computation time while optimizing performance. Two areas of MPC in need of expansion are time-varying constraints and phase transitions in multi-phase applications. In this work, MPC is employed to track the reference trajectory of a multi-phase satellite inspection mission. During certain phases, conical and spherical constraints are activated to represent a sun vector keep-in-zone (KIZ) and a collision avoidance keep-out-zone (KOZ), respectively. The equations of motion follow the linearized Hill-Clohessey-Wiltshire (HCW) equations. The problem is solved to minimize control effort and tracking error using MPCTools, a control and estimation tool for dynamic models. Results are compared to the validated results from literature and found to be comparable in costs while lending flexibility to phase design.

15. SUBJECT TERMS

Model predictive control, optimal control, rendezvous and proximity operations

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Maj. Costantinos Zagaris, AFIT/ENY
U	U	U	UU	88	19b. TELEPHONE NUMBER *(include area code)* (937) 255-3636, x4774; costantinos.zagaris@afit.edu