A Framework for Autonomous Cooperative Optimal Assignment and Control of Multi-Agent Systems

THESIS

Devin E. Saunders, Second Lieutenant, USSF

AFIT-ENY-MS-21-M-317

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

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

AFIT-ENY-MS-21-M-317

A Framework for Autonomous Cooperative Optimal Assignment and Control of Multi-Agent Systems

THESIS

Presented to the Faculty

Department of Aeronautics and Astronautics

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Astronautical Engineering

Devin E. Saunders, BS

Second Lieutenant, USSF

25 March 2021

AFIT-ENY-MS-21-M-317

A Framework for Autonomous Cooperative Optimal Assignment and Control of Multi-Agent Systems

Devin E. Saunders, BS
Second Lieutenant, USSF

Approved:

Maj Constantinos Zagaris, PhD (Chairman)	Date
Dr. Richard Cobb, PhD (Member)	Date
Maj Joshuah Hess, PhD (Member)	Date

Abstract

Space has always been a domain requiring a high degree of autonomy. The challenges presented by the required autonomy have made it difficult to accomplish complex tasks and operations with short response windows. With the growing use of multi-agent systems to enhance old and demonstrate new capabilities in the aerial domain, the need for development in multi-agent operations on orbit and in proximity operations has never been greater. A decentralized, cooperative multi-agent optimal control framework is presented to offer a solution to the assignment and control problems associated with performing multi-agent tasks in a proximity operations environment. However, the framework developed may be applied to a variety domains such as air, space, and sea. The solution presented takes advantage of a second price auction assignment algorithm to optimally task each satellite, while model predictive control is implemented to control the agents optimally while adhering to safety and mission constraints. The solution is compared to a direct orthogonal collocation method, and a study on tuning parameters is included. Results demonstrate the proposed technique allows the user to optimize control beyond phase horizons with Model Predictive Control and achieve a formation rendezvous with three tuning parameters. This better approximates phase transition in collocation techniques compared to traditional multi-phase MPC.

Acknowledgements

I'd like to thank my committee Maj Constantinos Zagaris, Maj Joshuah Hess, and Dr. Richard Cobb for all they've taught me over the course of my program. I've without a doubt grown as an engineer with your help. Additionally, I'd like to thank my parents Sharyn and Mike for their support throughout my program.

Devin E. Saunders

Contents

Page

List of Figures

Figure Page

List of Tables

List of Symbols

A	continuous form of HCW plant matrix
A_{con}	linear constraint matrix
A_d	discrete form of HCW plant matrix
B	continuous form of HCW input matrix
B_d	discrete form of HCW input matrix
b_{con}	linear constraint vector
f	linear part of Parallel Quadratic Form
H	solution to the Algebraic Riccati
$\mathcal{H}$	quadratic part of the Parallel Quadratic Form
$\mathcal{J}$	cost
N	matrix of normal vectors
n	mean motion or number of planes depending on context
$\hat{\mathbf{n}}$	normal vector
P	Riccati variable
P_{ij}	the price of agent i to perform task j
Q	state weighting matrix
Q_f	state weighting matrix with respect to patrol formation
Q_r	state weighting matrix with respect to rendezvous formation
R	control weighting matrix
r_c	chief's orbital radius
r_d	deputy's orbital radius
S	scale centering parameter
s	scale
T	thrust

$\bar{U}$	control input vector
U_{lb}	input lower bounds
U_{ub}	input upper bounds
V	keep out zone ellipsoid semi-axis
$\hat{\mathbf{v}}_{\odot}$	Sun vector
X	state vector
X_B	state vector of centroid of patrol formation
X_R	state vector of centroid of target formation
X_d	state vector of partrol vertecies
x	radial relative distance
y	in-track relative distance
z	cross-track relative distance
Γ	reference trajectory of an agent
γ	half angle of keep in zone cone
ε	complementary slackness
ϑ	angle of chief from inertial hati
μ	gravitational constant of Earth
ρ	relative vector of deputy with respect to chief
Φ	plant prediction matrix
Ω	input prediction matrix

List of Acronyms

APF	Artificial Potential Function
DoD	Department of Defense
GEO	Geostationary Orbit
GNC	Guidance, Navigation, and Controls
GPOPS	General Purpose Optimal Control Software
HCW	Hill-Clohessy-Wiltshire
HEO	Highly Elliptical Orbit
KIZ	Keep In Zone
KOZ	Keep Out Zone
LEO	Low Earth Orbit
LQG	Linear Quadratic Gaussian
LQR	Linear Quadratic Regulator
LVLH	Local Vertical Local Horizontal
MPC	Model Predictive Control
NERM	Nonlinear Equations of Relative Motion
NASA	National Aeronautics and Space Agency
PQP	Parallel Quadratic Program
QP	Quadratic Program
USAF	United States Air Force
USSF	United States Space Force
DoF	Degree of Freedom

A Framework for Autonomous Cooperative Optimal Assignment and Control of Multi-Agent Systems

I. Introduction

1.1 Motivation

Today multi-agent systems are growing in popularity. Traditionally, assets act alone or in small groups operated by many human actors. With the advances in autonomy and control, autonomous systems are beginning to supplement and replace these human actors leading to more robust systems that accomplish the same task with greater than or equal performance. Simultaneously, autonomous systems minimize the human exposure and risk of operator error in performing these tasks. This thesis offers an implementation of an optimal assignment algorithm to optimize the objectives of a mission while model predictive control minimizes the control effort to preform satellite rendezvous and proximity operations in a non-cooperative multi-agent environment. The resultant of these combined methods is a control framework useful for accomplishing cooperative formation rendezvous to minimize the fuel and control effort costs to the system while adhering to relevant mission and safety constraints.

Optimal formation rendezvous is of great importance to the Department of Defense due to the growing congested and contested nature of the space environment [1]. As space develops into a more congested and contested environment risk accumulates, posing a threat to the diverse network of satellite constellations operated by the DoD

[2]. In order to mitigate this risk, cutting edge autonomy must be introduced to new systems. The goal of the research herein is to develop an autonomous control framework viable for implementation on multi-agent systems in space. However, other domains such as air, sea, and land would also be able to take advantage of the resulting optimal assignment and control framework.

Space, as a domain, requires high degrees of autonomy due to the difficulty involved in operations. The National Aeronautics and Space Agency has outlined many of the environmental challenges autonomous technology in space must solve. This study attempts to investigate methods for solving the difficulties associated with communication delays, relative guidance, rendezvous, and docking. NASA's focus in these areas comes down to three measures: real-time implementability, optimality, and verfiability [3]. Autonomous satellite navigation is especially important due to the restrictions ground-in-the-loop operations place onto missions. The time delay associated with the vast distances in space greatly impacts the ability to preform precise tasks in orbits higher than low Earth orbit. Autonomy is a necessity to continue to improve mission frequency, robustness, and reliability.

The need for autonomy is clear, however, multi-agent systems in space would require a greater focus due to the significant loss to mission capability and high cost associated with needs to communicate with multiple satellites. Without autonomy, multi-agent satellite systems could be at risk of colliding, possibly resulting in catastrophic and nearly irreversible damage to the mission. The financial risk associated with a lack of autonomy is also significant due to the high cost of launching and operating even simple satellites. A complex system, such as the one described herein, would be costly to maintain, and need a large support network on the ground for de-

conflicting and executing the mission if a high degree of autonomy were not achieved. For these reasons, the degree of autonomy used in traditional satellites is not viable. The system must assign tasks, plan burns, and execute collections free of constant interaction and detailed planning from a ground station to allow for a small multi-agent system to have the flexibility necessary to accomplish the mission. Users on the ground would be responsible for assigning an area of interest and analyzing the collections. This will reduce the need for a large support network and allow the DoD to have a capability necessary to stay competitive in this contested environment. The framework poses to fulfill the DoD's position on autonomous space systems expressed in a report by the Defense Science Board [4]. The DoD's mission to maintain space domain awareness and dominance will be dependent on being able to leverage new technology in autonomy to alleviate the demand on operators while increasing capability to its users.

1.2 Approach

Multi-agent systems are of interest, rather than a single satellite, because they can be smaller and therefore cheaper and more agile. Additionally, this decrease in size provides ease of replacement should a satellite run out of fuel or become damaged while performing its mission. The quick development cycle of smaller satellites also allows the users to update the hardware and adapt the software as new models are produced. This will result in better performance as computation time is a limiting factor when using model predictive control. To summarize, multi-agent systems offer a resiliency and robustness not achievable by a singular spacecraft. This thesis takes advantage of developments in auction theory to handle the assignment problem. In

proximity operations involving a formation of satellites, it may be necessary to perform maneuvers such that the formation's states match a predetermined condition to achieve a desirable configuration. As in all space operations it is important to consider the cost of a maneuver in terms of fuel use, so an optimal control that minimizes the required control is desirable. Additionally, maintaining the formation throughout the maneuver ensures the formation is able to perform its mission while undergoing the reconfiguration maneuver. Model predictive control (MPC), also known as receding horizon optimal control, is used to balance staying in formation and performing the rendezvous mission while maintaining favorable lighting conditions, control limits, and implementing collision avoidance. The proposed method provides a near optimal control framework capable of reacting to a dynamic target structure in real time on hardware.

Traditionally, Hungarian Assignment is used to formulate a one-to-one assignment given a weighted cost matrix [5]. This thesis uses a second-price auction to optimally assign tasks quickly. In doing this, it becomes necessary to discretize the position and velocities of the targets. However, this is also done when using MPC. Once the agents are assigned, this data is passed to the agents' respective controllers. Each agents' control system then solves an optimal control problem that accounts for potential collisions in the various trajectories as well as the direction of approach. For example, a satellite may want to approach from the direction of the sun vector in order to achieve favorable illumination conditions. Model Predictive Control does this by using a receding finite horizon. Traditional optimal control typically uses an infinite time horizon to accomplish its task. However, with a moving horizon, computation time can be greatly reduced and changes in the task can be enforced. This allows updates after a fixed duration increasing the resiliency of the multi-agent system.

In this work, MPC solves two distinct problems. The first problem considers the control of the centroid of the agents' formation towards the targets' formation. The outer-loop MPC solves this problem while adhering to a keep-in-zone (KIZ) constraint. A second-order cone is approximated by a number of planes to form the KIZ. The second inner-loop MPC performs a reconfiguration of the formation to the target structure while each agent is considered a keep-out-zone (KOZ). An ellipsoid is used to approximate each of the other members of the formation to accomplish collision avoidance. The pre-maneuver formation, or "patrol formation", is user-defined while the targets define the terminal structure. The transition between these two formations is balanced using a logistic function. As a result, the user gains several parameters used for tuning the maneuver to achieve desired performance. The optimality of the framework is assessed by comparing the resulting trajectory to the solution found using commercially available optimal control software (GPOPS-II) [6]. Figure 1 overviews the formation maneuver problem.

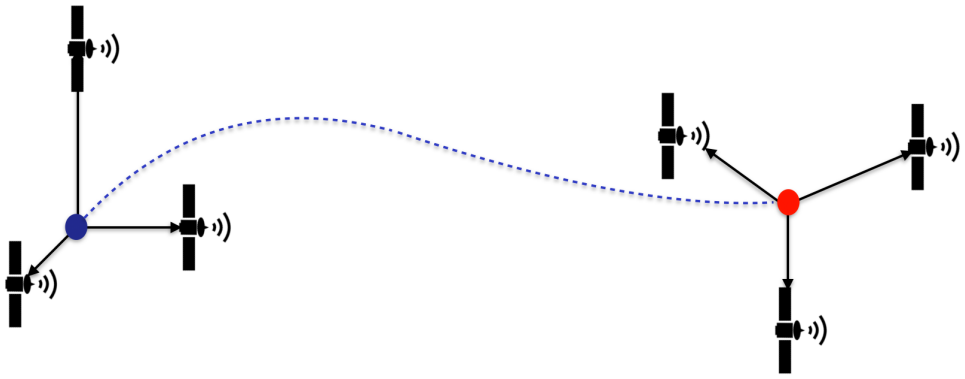

Figure 1. Multi-Agent Control Scenario; formation center translates from blue dot to red dot while agents achieve desired relative formation.

1.3 Assumptions and Limitations

The reservation towards adoption of highly autonomous systems in the DoD is attributed to three factors according to the Defense Advanced Research Projects Agency: reliability, complexity, and unpredictability [7]. This approach attempts to compile commonly used modern methods into an autonomous system to alleviate these reservations. The auction assignment algorithm and model predictive control are well established and understood. The research described in this thesis hopes to demonstrate a system, being comprised of several well understood reliable and relatively simple parts, can successfully execute complex missions [8].

This study makes the assumptions that position information of the target structure is known and perfect. Additionally, full state information is being passed between the agent structure and is also without error. The target structure may vary to simulate dynamic changes to the target structure. In other words, the target structure is passively dynamic and does not make decisions about its structural changes. Each agent in the simulation is modeled as a point mass and the translational dynamics are predicted with MPC using linearized Hill-Clohessy-Wiltshire equations of motion [9]. The control will be implemented with 3 degrees of freedom and will be bounded to ensure realistic inputs are used to track the desired reference signal. Models for individual subsystems of the agents (i.e. sensors, actuators, etc.) are ignored, but saturation values for applied forces are limited to realistic values for these satellite subsystems. This problem shares a similar form to Linear Quadratic Regulator (LQR) control modified to handle the formation mechanics.

The research accomplished in this thesis is wholly reliant on simulations. No experimental investigation is conducted. While this provides a sterile environment to

test the methods outlined, this benefit is also a limitation. The difficulty involved in replicating a space environment on a large enough scale to test control algorithms poses a risk to new and experimental methods. For this reason, it is important to analyze the system using high-fidelity models that represent the true motion of the agents. The Nonlinear Equations of Relative Motion are chosen to do this [9]. However, this model poses challenges that will be further discussed in later sections of this study. The goal of this research is to provide a proof of concept that will allow for higher fidelity models that consider multi-body gravity, oblate Earth, as well as other perturbations and complexities.

1.4 Thesis Overview

The topics and research described in this thesis can serve as a foundation for the DoD on the use of multi-agent systems in space. Taking advantage of autonomous multi-agents space systems has the potential to decrease mission costs and operator workload while increasing mission effectiveness. Multi-Agents systems pose an answer to the problems of a congested and contested modern space environment. New and innovative methods must be used to increase the capabilities of an aging fleet of DoD space assets. Autonomy needs to be a significant player in the future systems of the DoD to assure space superiority. Chapter 2 takes a deeper dive into the individual fields that this research investigates to provide the reader with a more detailed understanding of the aspects of the problem. After the background, the methodology of the solution is described in detail in Chapter 3 to allow the reader to replicate the optimal assignment algorithm and control formulation. The results of this study follow in Chapter 4 before the conclusion in Chapter 5, which expresses the significance of the findings and the next steps to be taken.

II. Literature Review

In this chapter the major aspects of the formation rendezvous and reconfiguration problem are discussed. This complex problem is not limited to the basis reviewed in this chapter, however, the focus on dynamics, multi-agent systems, and optimization are meant to give the reader the necessary knowledge to be able to replicate the work described in this thesis.

2.1 Rendezvous and Proximity Operations

Rendezvous and Proximity Operations has been of great interest in the space community for almost as long as satellites have been launched. The first ever rendezvous occurred only nine short years after the launch of the first satellite, Sputnik, in 1957. The Gemini program accomplished this feat on 16 March 1966 in order to develop techniques that would be necessary to put a man on the Moon [10].

2.1.1 Nonlinear Equations of Relative Motion

The Nonlinear Equations of Relative Motion (NERMs) describe how a satellite, generally referred to as the deputy, behaves with respect to some reference orbit, or chief. As a result, a relative frame must be developed to describe this motion. The common choice is the Local Vertical Local Horizontal (LVLH) frame. In the LVLH frame the chief is placed at the origin and the frame rotates around the Earth as the chief would in orbit (see Figure 2).

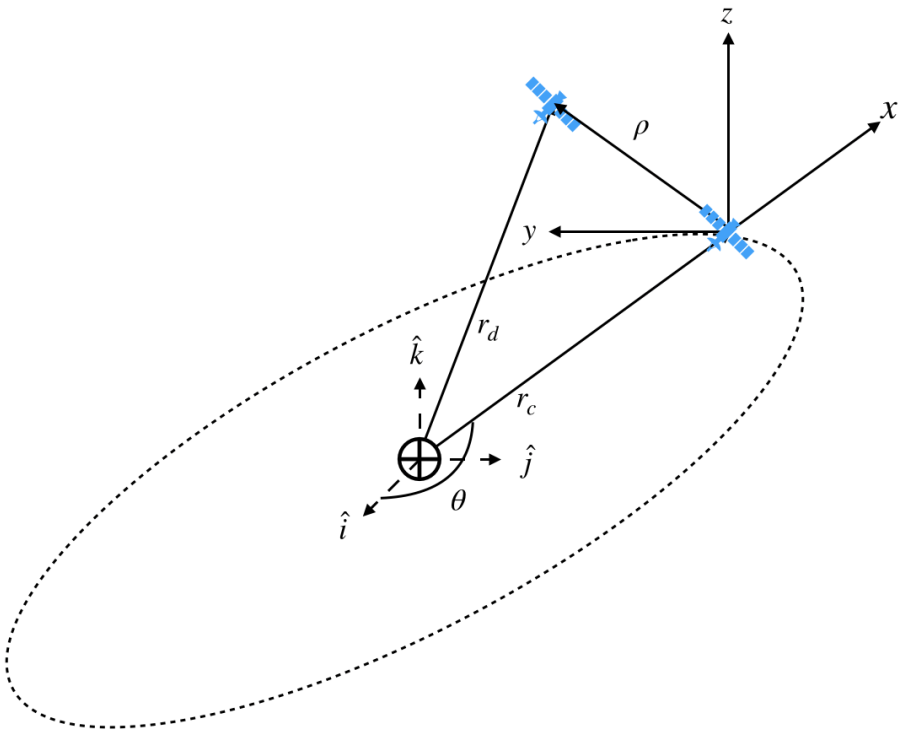

Figure 2. Chief and Deputy in Local Vertical Local Horizontal(LVLH) Frame.

In the unperturbed case, the equations of motion are developed from two body Keplerian motion. Earth is treated as a point mass with a uniform gravitational potential field; other perturbations such as solar radiation pressure and atmospheric drag are ignored. However, in the NERMs differential gravity is considered. This accounts for the differences in attractive potential at different orbital altitudes. The unperturbed NERMs can be derived from the difference of two satellites exhibiting two body Keplerian motion to develop the following second-order differential equa-

tions shown in Eq. 2.2-2.5 [9]:

$$\ddot{x} = \ddot{\vartheta}y + 2\dot{\vartheta}\dot{y} + \dot{\vartheta}^2 x - \frac{\mu}{r_d^3}(r_c + x) + \frac{\mu}{r_c^2} \tag{2.1}$$

$$\ddot{y} = -\ddot{\vartheta}x - 2\dot{\vartheta}\dot{x} + \dot{\vartheta}^2 y - \frac{\mu}{r_d^3}y \tag{2.2}$$

$$\ddot{z} = -\frac{\mu}{r_d^3}z \tag{2.3}$$

$$\ddot{\vartheta} = -\frac{2\dot{r_c}\dot{\vartheta}}{r_c} \tag{2.4}$$

$$\ddot{r_c} = r_c\dot{\vartheta}^2 - \frac{\mu}{r_c^2}. \tag{2.5}$$

The variables x, y, z, and their derivatives $\dot{x}$, $\dot{y}$, $\dot{z}$ represent relative displacements and velocities in the radial, in-track, and cross track directions respectively. The remaining four states in this ten state model apply to the chief's orbit. The argument of latitude and its derivative ϑ and $\dot{\vartheta}$ make up the seventh and eighth terms, while the chief's radius and its derivative r_c and $\dot{r}_c$ make up the ninth and tenth states. With the state vector constructed as in Eq. 2.6 these equations can be expanded into ten first-order differential equations to describe the motion of each state.

$$\bar{X} = \begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} & \vartheta & \dot{\vartheta} & r_c & \dot{r}_c \end{bmatrix}^T \tag{2.6}$$

The nonlinearities are clear in each equation which have a multitude of state products, states raised to a power, or both. The only equilibrium point that exists in the NERMs is,

$$\rho^* = \begin{bmatrix} x & y & z \end{bmatrix}^T = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T. \tag{2.7}$$

The equilibrium point ρ^* implies the only location where the relative motion between two orbiting bodies vanishes occurs when the bodies are co-located. The stability of this equilibrium point can be calculated by first calculating the Jacobian of the

system, then supplying the equilibrium point while truncating the higher-order terms, and finally by determining the eigenvalues of the resulting matrix (Eq. 2.9-2.11).

$$\dot{\bar{X}} = f(\bar{X}) \tag{2.8}$$

$$= \left(\frac{\partial f(\bar{X})}{\bar{X}}\right)_{\bar{X}=\varrho^*} + \cancelto{0}{h.o.t} \tag{2.9}$$

$$= A\bar{X} \tag{2.10}$$

$$eig(A) \geqslant 0 \tag{2.11}$$

When attempting this, an eighth order polynomial is found that represents the characteristic equation of the ten state system. The order of this polynomial implies that there are repeated eigenvalues at the origin, and as a result the system is unstable. Additionally, the coefficients of the polynomial change signs and contain states in them. According to Routh stability criteria [11] this is further evidence of the instability of the system. The instability found through the process of Lyapunov's Indirect method implies that Lyapunov's Direct method cannot be used to find a stable Lyapunov Candidate function for the system [12], and as a result controls must be developed to achieve stability within the system.

2.1.2 Hill-Clohessy-Wiltshire Equations of Motion

The Hill-Clohessy-Wiltshire equations of motion are a linearized model of the dynamics of relative motion [9]. The linerization is accomplished by making the assumption that the chief inhabits a circular orbit. The unforced motion can be modeled as in Eqs. 2.12-2.14, and an applied force (T) can be added to give control to each satellite with Eq. 2.15 -2.17. After converting to first order form, develop the plant matrix, A, from Eqs. 2.12-2.14, and the input matrix, B, from Eqs. 2.15 -2.17 to arrive at a simple model for a satellite with 3 DoF control. In this model,

n represents the mean motion of a circular orbit for a desired altitude, T represents the thrust of the satellite, and m the mass. A linear differential equation can then be constructed as in Eq. 2.18 to model the satellite.

$$\ddot{x} = 3n^2x + 2n\dot{y} \tag{2.12}$$

$$\ddot{y} = -2n\dot{x} \tag{2.13}$$

$$\ddot{z} = -n^2z \tag{2.14}$$

$$\ddot{x} = \frac{T}{m}u_1 \tag{2.15}$$

$$\ddot{y} = \frac{T}{m}u_2 \tag{2.16}$$

$$\ddot{z} = \frac{T}{m}u_3 \tag{2.17}$$

$$\dot{X}_i = AX_i + BU_i \tag{2.18}$$

The state vector of in Eq. 2.18 is a decatenated form of the state vector in Eq. 2.6 where only the first six states are used.

2.2 Multi-Agent Control Strategies

Multi-agent control can take many forms. Virtual structures, artificial potential functions, and behavior based strategies are some of the most common [13]. These different strategies deal with how an agent defines the error between its current state and its desired state. However, there are other aspects of navigation unique to multi-agent systems. One of these is communication topologies. This concerns how, and how much, information is being shared within members of the formation [13]. For this study, communication topologies will not be investigated, and all information is assumed to be shared within the formation. This is done to allow focus on the control development of a working framework for accomplishing the reconfiguration mission. Behavior-based strategies are also not considered as they tend to rely heavily on

machine learning which currently does not promote the simplicity or predictability outlined by DARPA as a necessity for a highly autonomous system [7].

2.2.1 Virtual Structure Based Control

There has been some development in the field of multi-agent control of satellites performing proximity operations. The most common technique involves virtual structures. Virtual structures fix an agent's location relative to a pre-defined center. The agents involved in the formation define their error states with respect to this relative position as done by Ren [14]. Proportional-Derivative control is then used to keep the members of the formation at their respective coordinates relative to the center. Another method in virtual structure control is to define a constraint function that represents the formation as demonstrated by Egerstedt [15]. The constraint function is then parameterized along a path and the agents of the formation follow this path. However, to the author's knowledge none of these works expanded on the nature of multi-agent optimal control. Work done by Palacios et al. [16] introduces optimal control into the problem with a solution that results in a multi-agent rendezvous through the use of LQR control. Error states are calculated by differencing the current and desired states of the agents within the formation, but formation keeping is not considered in the formulation of the LQR. When performing formation proximity operations it may be important to consider the formation shape to ensure mission requirements are met as Egerstedt [15] and Ren [14] enforce throughout their works. At the same time, the operator may find it necessary to change the profile of the formation, and do so in a way that minimizes fuel costs. These constrained optimization problems cannot be solved with a traditional LQR formulation. Therefore, an optimal control problem that allows for formation reconfiguration fills a perceived research gap.

2.2.2 Artificial Potential Function Based Control

Artificial potential functions also accomplish the multi-agent control and path following problems, but instead of relative positions for the agents to track, an individual agent seeks to minimize its potential function [17]. APF-based control relies heavily on Lyapunov theory. This leads to a gradient-based control approach which has been proven to be effective because it also allows for obstacle avoidance through the use of repulsive potential functions [13]. However, because constraints are generally parameters in this manner they are typically considered "soft" constraints, as there is no guarantee that it will not be violated. "Soft" constraints are not viable for this problem so another method must be employed.

2.3 Optimal Assignment

Assignment problems are combinatorial optimization problems involving several tasks and a number of parties. The parties have some value associated with each task involved. The goal is to determine, in an optimal manner, which party should perform which task [18]. Figure 3 gives the reader a visual aid to the nature of the assignment problem.

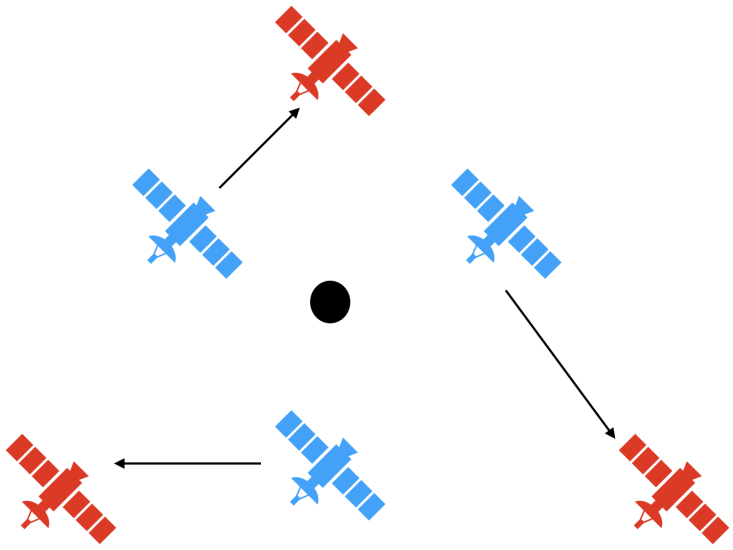

Figure 3. The Assignment Problem from the perspective of Rendezvous and Proximity Operations. Each blue satellite is assigned a task to rendezvous with a red satellite. The center of each formation is collocated.

2.3.1 Hungarian Method of Assignment

Assignment problems stem from logistics and began being analyzed by mathematicians in the mid-1950s [5]. Kuhn's Hungarian algorithm for linear assignment was the first major breakthrough in this area of research. The Hungarian algorithm prescribed a method for assigning these tasks in an optimal way [5]. The computational complexity of this algorithm is $O(n^4)$ and Kuhn initially limited the algorithm to having an equal number of parties and tasks. Kuhn later modified his algorithm to handle the general assignment problem which allows for unbalanced assignments by including dummy variables. Mathmetician Lester Ford looked at the problem through the lens of flows through a network. As a result, the Ford-Fulkerson algorithm was developed [19].In the early 1970s additional variants were able to optimize the Hungarian algorithm to have a computational complexity of $O(n^3)$ [20].

2.3.2 Auction Based Assignment

The same assignment problem can be analyzed through this lens in the form of auction algorithms [21]. There are several ways to hold auctions, but the basis in analyzing these algorithms holds in that the final cost to the system is measured by the sum of each parties assigned tasks. The task is to analyze techniques and conditions that promote the highest utility. Utility can be taken from the perspective of the buyer, seller, or both. In first price auctions, the most traditional auction style, the highest bidder pays the price he or she bids for an object. In second price auctions, the highest bidder pays the price of the second highest price instead. This promotes truthful bidding and as a result a Nash Equilibrium [22] forms where the bidder maximizes their utility by bidding what they consider the true value of the object or task to be.

While both the Hungarian and the Second Price Auction algorithms provide the optimal solution to the assignment problem, in practice the auction algorithm is more common. This is because of the speed, $O(n^2 \log n)$, and recursive nature of auction algorithms. This allows the program to return the best solution after a user defined number of iterations which is desirable because the computational time of the program can be adjusted to suit the user's needs. The returned assignment's utility monotonically increases or decreases, depending on the nature of the problem, with each iteration. As with the Hungarian algorithm, the problem is deconstructed into a bipartite graph which the auction algorithm is designed to solve with the added flexibility that partitions of the graph need not have the same order. This allows for arbitrary number of agents on either team. A cost matrix can be formed by having rows represent the parties interested in completing tasks and the columns representing the tasks to be performed.

With the cost matrix formed, an auction can be held to assign tasks to the agents responsible for maneuvering. The algorithm for accomplishing this follows [23] [21]:

1. Start with a set U of all bidders. U denotes the set of all unassigned bidders. Also maintain a set of prices which are all initialized to 0, and any structure that stores the current tentative (partial) assignment.

2. Pick any bidder i from U. Search for the item j that gives the bidder the highest net payoff C_{ij} - p_j, and also an item k that gives them the second highest net payoff.

3. The price p_j of item j is updated to be $p_j = p_j + (C_{ij} - p_j) - (C_{ik} - p_k) + \varepsilon$. This update simply says that p_j is raised to the level at which bidder i is different(in terms of net payoff) between item j and item k.

4. Now assign item j to bidder i. If item j was previously assigned to another bidder S, then remove that assignment and add S to U .

5. If U becomes empty, the algorithm is over; otherwise, go back to Step (2).

An example of the implementation of this algorithm follows with Table 3 for reference. Begin with a cost matrix, W; due to the algorithm being optimal for maximization problems we multiply the matrix by negative one to turn it into a minimization problem.

$$W = \begin{bmatrix} -14 & -5 & -8 & -7 \\ -2 & -12 & -6 & -5 \\ -7 & -8 & -3 & -9 \\ -2 & -4 & -6 & -10 \end{bmatrix} \tag{2.19}$$

Table 3. Auction Algorithm Example

Iteration	Prices	Assignments	Bidder	Task	Bid
1	(0,0,0,0)		1	2	2.2
2	(0,2.2,0,0)	(1,2)	2	1	3.2
3	(3.2,2.2,0,0)	(1,2),(2,1)	3	3	6.2
4	(3.2,2.2,6.2,0)	(1,2),(2,1),(3,3)	4	1	4.4
5	(4.4,2.2,6.2,0)	(1,2),(3,3),(4,1)	2	4	1.6
6	(4.4,2.2,6.2,1.6)	(1,2),(3,3),(4,1),(2,4)			

$$W^* = W_{12} + W_{33} + W_{41} + W_{24} = \text{-}15 \tag{2.20}$$

Calculating the payoff for bidder 1 can be done by subtracting the vector of prices from the first row of the cost matrix. The two tasks that maximize payoff become task 2 and task 4 at -5 and -7 respectively. The bid for task 2 is calculated with a complementary slackness of $\varepsilon = 0.2$. Finally, task 2 is assigned to bidder 1. This process repeats until iteration 3. In this iteration we can see maximizing payoff results in picking task 3 and 4. This is the first instance where the tasks chosen are not the largest values of in the bidder's row of the cost matrix. This occurs because the price of objects 2 and 1 have gone up as a result of the bids from bidders 1 and 2. Bidder 3 makes a large bid on object 3 because there is a large difference between the payoff of this object and the next best task. Iteration 4 shows that despite the increased price of objects one through three the payoff is still maximized by bidder 4 choosing tasks 1 and 2. As a result, bidder 4 outbids bidder 2 for task 1. Bidder 2 then returns to the set of unassigned bidders. In iteration 6 the payoffs of each task have changed from the last instance bidder two was involved, and this time it is in bidder 2's best interest to select task 4. This concludes the auction, and an optimal assignment has

formed. This means there does not exist an assignment that will have a cost lower than the sum of the elements chosen.

To demonstrate that bidding honestly is a dominant strategy in this auction, assume that the other agents will bid arbitrarily, and consider the two possible outcomes for bidder i. A visual representation of this can be seen in Figure 4.

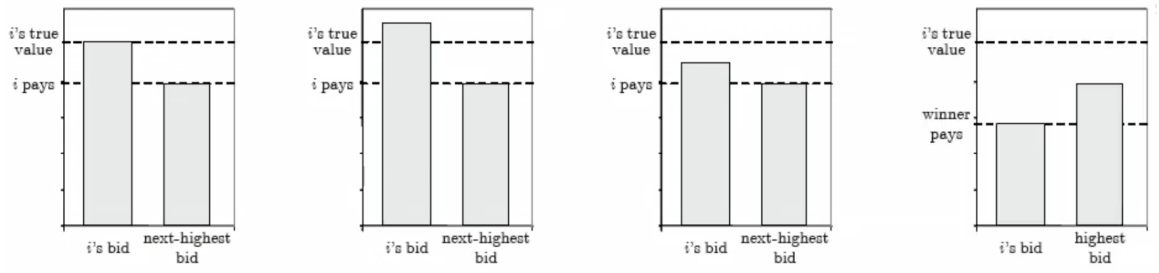

Figure 4. Payoff of different bids for bidder i. A Nash Equilibrium forms when the bidder bids at the true value maximizing utility.

Consider first outcomes where bidder i wins. For any object that may be bid upon imagine a price π_k that represents the true value of that object k. If bidder i bids at this price, and wins than they pay the next highest price. If bidder i bids higher than π_k than they still win and again pay the next highest price. If bidder i bids slightly lower than π_k and still wins they will again pay the next highest price. In all these scenarios, assuming the next highest price is constant, the bidder i has no increase in utility for raising or lowering its price any higher than the true value of the object. Now consider the situation where bidder i bids lower than the others price, as a result bidder i gains no utility because they lose the auction. As a result, the best strategy for bidder i is to bid their perceived value of object k because any lower or higher results in no benefit to the bidder. Therefore, a Nash Equilibrium is reached among the bidders in the auction if they follow this dominant strategy and the algorithm

above takes advantage of this dominant strategy by assigning objects to the bidders accordingly to maximize the utility to all agents.

2.4 Optimal Control

Classical control techniques aim to stabilize a system to perform either regulation or tracking to a target state. However, these techniques do not inherently optimize performance. To address this problem, optimal control techniques were developed to determine the best control according to some performance measure. For linear control systems, the most prevalent of these techniques are Linear Quadratic Regulator (LQR) and Linear Quadratic Guassian (LQG) control. LQR and LQG control allow the user to tune the performance of the system according to some weighting matrices while guaranteeing closed-loop stability. However, constraints on the trajectory or control are not generally enforced. As a result, more general techniques, such as pseudospectral methods and model predictive control, were developed to find solutions to increasingly difficult control problems [24].

2.4.1 Linear Quadratic Regulator Control

LQR control is a common technique in linear systems to provide an optimal control law [25]. The standard form of the cost functional when using LQR control is shown in Eq. 2.21.

$$\mathcal{J} = \mathrm{X_f^T H X_f} + \int_{\mathrm{t_0}}^{\mathrm{t_f}} \mathrm{X^T Q X} + \mathrm{U^T R U dt} \tag{2.21}$$

Using the algebraic Riccati equation a constant optimal gain matrix can be solved for resulting in a desired response for the system. The use of this steady-state LQR assumes that the control system will operate for long time periods compared to the transient of the solution to the Riccati equation [26]. Defining a stable LQR requires

the state weighting matrix Q to be positive semi-definite and the control weighting matrix R is required to be positive definite. LQR offers the user two tunable parameters that will alter the weighting of the state error and control use. However, as the name describes, this technique requires a linear system. If using a nonlinear system a linear model must be developed in order to use this technique. In doing this, limits on the states are likely to develop to describe a viable region where the linearization holds. However, these and any necessary control boundaries are not able to be enforced using this technique. As a result, solving the problem with a different approach would be more appropriate.

2.4.2 Numerical Methods for Optimal Control

Dynamic programming is one of the most widely used approaches for constrained optimal control. For the majority of the history of mathematics, problems of this class were considered unsolvable. However, the relatively recent developments in computation power unlocked new capabilities allowing accurate solutions to some of the most complex problems to be determined [27]. The basis for determining optimal control through these methods stems from the principle of optimality developed by Bellman: "An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision [28]." The problem is deconstructed into a two point boundary value problem and is solved at a number of decision points recursively until a solution is found [24]. In doing this, the initial trajectory may not satisfy all constraints, however, as the program continues the weight of unsatisfied constraints increases to attempt to find a solution that satisfies all constraints while minimizing the performance measure. Pseudospectral methods are often used to determine which decision points a program selects to determine the solution.

Numerical methods of optimization are extremely powerful, and can result in solutions to a predetermined accuracy. Once solved, the entire control profile is available to the user. However, this means open-loop control must be implemented which does not guarantee stability outside the bounds of the problem initially solved. Additionally, if disturbances and uncertainty are present in the system, the control may not result in the desired performance or trajectory demanded by the system. One idea may be to continuously use these numerical methods to update the problem as disturbances occur effectively closing the loop on the system. However, due to the iterative nature of the program and the complexity of the problems these programs can take extremely long to solve, and are not guaranteed to converge to a solution. Therefore, in highly dynamic systems this technique will likely not be viable and another solution must be found.

2.4.3 Model Predictive Control

Due to the highly dynamic nature of the environments, the aerospace industry has been reluctant to use iterative computational methods to determine control in real time for its systems. However, with the significant increase in computational power over the past decades, a move towards computational guidance algorithms should be the focus of future endeavors [29]. Model Predictive Control (MPC) is the prescribed solution to this problem. MPC has been growing in popularity in recent years. The goal of MPC is to utilize the developments in computational speed to make predictions on how a system will behave in the future, and use this information to make forward thinking control inputs to increase performance. MPC takes aspects from both LQR and numerical methods in that closed loop feedback is used to generate robust control strategies while optimization routines are used to determine the best

control action. Additionally, constraints may be levied on the system to ensure safe and viable trajectories are taken to accomplish the mission.

By limiting the horizon of the optimization problem a solution can be found for a segment of the trajectory in the immediate future. Controls can be implemented within this horizon, and the horizon can then be shifted forward with the updated information gained about the present conditions. This process is repeated, and knowledge from previous solutions can be levied to increase the speed of the optimization routines. This has successfully been implemented for a wide range of problems including formation reconfiguration of spacecraft [30]. Wahl suggests next steps for formation reconfiguration missions on orbit include a focus in ensuring on-board implementation, alternative structures in the formulation of the cost function, and distributing computation across members of the formation. This thesis aims to develop a solution to these problems.

2.5 Summary

The NERMs are used to model "truthful" dynamics to agents acting in the region of space around a chief. An alternative linear model for these dynamics is the HCW model. In this linear model the user assumes the chief's orbit is circular. The assumption of a circular chief orbit is valid because with the exception of HEO orbits like Molnyia, the majority of orbits have very small eccentricities.

There are a multitude of factors when considering multi-agent systems. A virtual structure approach is chosen over an APF or behavior based approach due to simplicity. Factors related to inter-formation communication including topology, false information, and rate of transmission are not considered. This thesis focuses on the

dynamics and relative positioning of the members of a formation so collision avoidance is a factor.

Assignment algorithms are used to distribute tasks to a group in an optimal manner. The Hungarian algorithm was the first method used to perform this, however, with time auction based algorithms became more widely used due to their computational complexity. The second price auction algorithm encourages the bidders to bid according to their perceived value of a task, and as a result, a Nash Equilibrium forms.

Optimal control techniques differ from classical techniques in that a performance measure is considered. LQR control is common for linear systems, but constraints are generally not enforced when using this technique. Dynamic programming allows for more complex problems to be solved. This includes, but is not limited to, nonlinear and constrained problems. However, numerical methods are not guaranteed to converge, and as a result the computation time can be long. This does not allow for the use of these techniques in highly dynamic environments or scenarios where quick reaction time is key. MPC instead solves a portion of the problem, and this horizon is shifted as the mission develops. As a result, MPC allows for quick near optimal solutions to complex problems, and is guaranteed to be closed-loop stable. This makes MPC a desirable technique for use in scenarios with complex constraints or dynamics where a quick reaction time is necessary.

III. Methodology

The aim of this work is to advance research in the development of on-board computational guidance for formations of spacecraft on orbit performing rendezvous and proximity operations missions. This is accomplished by compiling thoroughly researched solutions to the problems of task dispersal and constrained optimal control of satellites in close proximity into a tunable framework capable of handling formation rendezvous and reconfiguration missions. The NERMs are used to enforce "truthful" dynamics to members of two separate formations. An auction algorithm is used to distribute tasks to the members of one formation that minimizes the cost of the maneuver to the formation as a whole. With an individualized task the agents in the formation use the HCW model with MPC to predict the future states of itself and its target. Control actions are determined through numerical optimization routines limited to accomplish "real-time" solutions to minimize a performance measure. Constraints are included to ensure collision avoidance, appropriate approaches, and realistic controls are maintained throughout the completion of the mission.

3.1 Optimal Assignment

As previously described, the goal of this work is to develop a framework capable of performing formation rendezvous and reconfiguration missions potentially in real time on flight hardware. The first aspect of this framework is the optimal assignment program. To develop the cost matrix associated with the problem a scalar representation of the state vectors of both the agent and target being inspected must be determined. To do this, begin by taking the state difference of the i-th agent and the j-th target. Secondly, formulate the quadratic form of the state difference weighted with the terminal cost matrix. (Eq. 3.1) If the optimal control problem is formulated

in a similar manner to an LQR the terminal cost matrix can be found by finding a solution to the algebraic Riccati equation. (Eq. 3.2)

$$W_{ij} = (X_i - X_j)^T P (X_i - X_j) \tag{3.1}$$

$$0 = A^T P + PA - PBR^{-1}B^T P + Q \tag{3.2}$$

It is important to note that because the MPC controller will be working in the discrete domain, the matrices A and B in Eq 3.2 must also be in the discrete domain. Additionally, ($Q \geqslant 0$) and ($R > 0$) should represent the weighting associated to matching the j-th target's state while using the i-th agent's control respectively. Finally, the terminal cost matrix is constant across the maneuver, so this matrix can be calculated once and stored in memory to increase computational efficiency.

With the cost matrix formulated, an optimal assignment can be determined to assign each agent in the formation to a target using MATLAB's `assignauction` function. A cost of non-assignment of 10^5 is selected. The cost of non-assignment allows the user to exclude targets from assignment if the cost is sufficiently large, so making this value large ensures that all targets are assigned. The built-in MATLAB function does not contain a iteration limiter, however, this function could be implemented on flight hardware to ensure the program is run within a specified interval of time. The decision to use MATLAB's function over a custom built program was made after determining that the program was sufficiently fast for the problem involving the rendezvous of four agents and four targets without the need for an iteration limiter.

The optimal assignment program can be placed in the framework as described in Figure 5. This framework represents the control loop of an individual in the agent formation. A target's control loop is assumed to contain only the plant. The plant, as previously discussed, uses the NERMs to determine the states of a given agent or

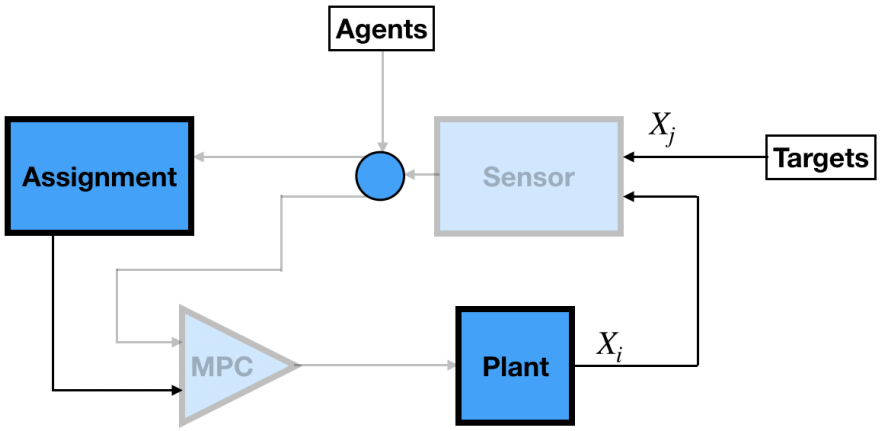

Figure 5. Optimal Assignment Program and Plant in Control Framework.

target. The box described as "Agents" represents the information of each agent in the formation. The box described as "Targets" represents the states of all targets in the opposing formation. The circle in Figure 5 represents the communication between the agents of the formation. The communication is without errors or delay, and relays the information of each agent's state and predicted control profile to all other members of the formation. The input of the assignment program requires knowledge of both the agent and target states, and results in an optimal assignment. The output of the assignment algorithm is then passed to the MPC controller.

3.2 Optimal Control Problem

3.2.1 Cost Function

Before adapting the problem for MPC, the optimization problem is formulated in a manner similar to a LQR. The traditional formulation considers only states and control, however, the proposed formulation considers the patrol formation and the

desired configuration as well as control. The patrol formation represents state error with respect to the virtual center of the formation. This is used to enforce a desired structure on the agents as they maneuver towards the target formation. A desired formation may be desirable to the user to ensure proper spacing, enforce lighting constraints, or take advantage of stereo-vision in order to develop state estimates of targets. In order to accomplish the transition from the patrol formation to the desired rendezvous configuration a logistic function is used to change the weighting of each of the error terms throughout the mission. Begin with a cost function in the form of Eq. 3.3,

$$\mathcal{J} = \int_{t_0}^{t_f} J_1 + J_2 + J_3 + J_4 \, dt, \tag{3.3}$$

$$\text{where :} \tag{3.4}$$

$$J_1 = (\bar{X}_B - \bar{X}_R)^T Q (\bar{X}_B - \bar{X}_R), \tag{3.5}$$

$$J_2 = (\bar{X}_i - \bar{X}_d)^T Q_f (\bar{X}_i - \bar{X}_d), \tag{3.6}$$

$$J_3 = (\bar{X}_i - \bar{X}_j)^T Q_r (\bar{X}_i - \bar{X}_j), \tag{3.7}$$

$$J_4 = \bar{U}^T R \bar{U}. \tag{3.8}$$

$$\tag{3.9}$$

In the above equations $\bar{X}$ represents the six state vector described in Eq. 2.18. The subscripts B and R represent the centroid of the agent and target formation respectively. The subscripts i and j represent any given agent and its assigned target respectively. $\bar{U}$ represents the vector of controls for the agent i, while Q, Q_f, Q_r, and R represent weighting matrices the user defines.

With this cost function an agent will seek to co-locate the the centroids of each formation (Eq. 3.5), while maintaining its position relative to the centroid (Eq. 3.6)

and simultaneously seeking to rendezvous with its target (Eq. 3.7) all the while considering the control necessary to complete the maneuver (Eq. 3.8). It is clear that the term J_2 will inhibit the agent from performing the task in term J_3 and vice versa, so a scheme must be developed in order to properly weight these terms throughout the mission.

To properly weight the terms responsible for the patrol and desired formation configurations, Q_f and Q_r, respectively a logistic function is used to scale the weighting terms between zero and one according to the magnitude of the difference between the two centroid vectors, (Eq. 3.10)

$$s = \frac{1}{1 + e^{-k((\bar{X}_B - \bar{X}_R)^T(\bar{X}_B - \bar{X}_R) - S^2)}}, \tag{3.10}$$

$$Q = sQ_f + (1 - s)Q_r. \tag{3.11}$$

The parameter, k, allows the user to determine the aggressiveness of the switch between profiles, and S allows the user to develop a spherical switching surface around the target formation that once passed, signals the formation to transfer from the patrol formation to the desired configuration. If the state weighting matrices are assumed to be identity, Eq. 3.11 can be used to determine the weighting matrices if the scaling term, s, is calculated. Choosing the scaling this way results in three tunable parameters for the optimization problem in Eq. 3.3: scale rate (k), scale centering (S), and control weighting (R). Figure 6 demonstrates how the scale of these matrices changes throughout the mission. If the patrol formation starts at a range from the target formation the weighting of error with respect to the patrol formation will dominate, and as a result be minimized. However, as the formation maneuvers towards its target the weights balance at the centering value S before the term Q_r begins to dominate the state weighting.

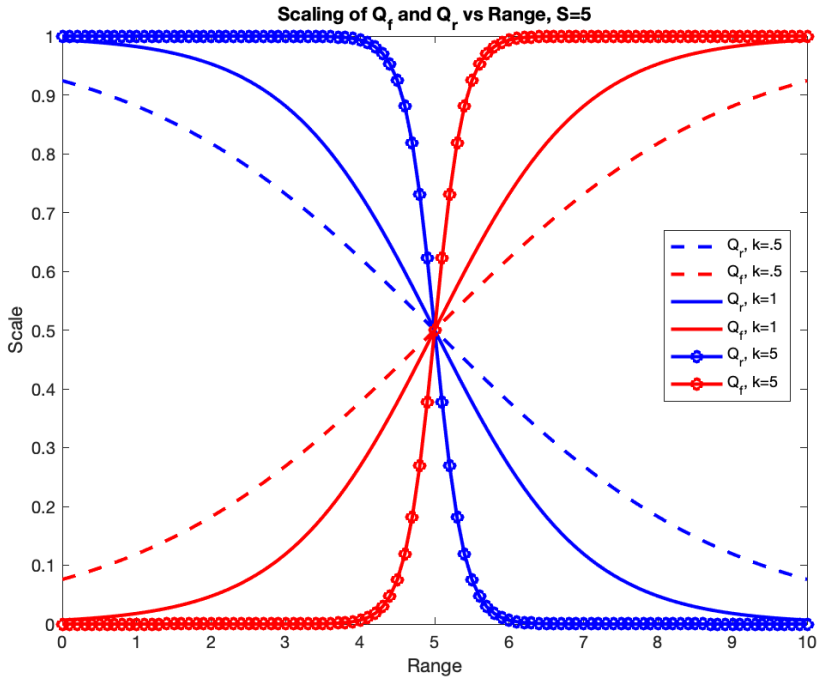

Figure 6. Scaling of Q_f and Q_r throughout the mission.

3.2.2 Constraints

In addition to the cost function there are multiple constraints used in developing the optimization problem. These constraints include: collision avoidance, favorable lighting conditions, control bounds, and state bounds. Levying these constraints onto the system will ensure the agents keep safe separation and remain in areas that will promote a successful mission while ensuring that realistic inputs are used to accomplish the tasks. Collision avoidance constraints are developed using only the relative positions to form a keep out zone (KOZ) for the agent (Eq. 3.12). The other members of the formation determine where the KOZs are located and in order to meet the constraint the agent must not enter the various KOZs at any point in the trajectory.

$$1 \leqslant (X_i - X_{koz})^T V (X_i - X_{koz}) \tag{3.12}$$

If a KOZ is violated a collision is assumed and the mission is considered a catastrophic failure. The KOZs are modeled as ellipsoids with the semi-axes' represented by the diagonal positive definite matrix V. However, the nature of this formulation creates a nonlinear non-convex constraint. While optimal control software, such as GPOPS, can handle constraints in this form, linear solvers such as the quadratic program (QP) solver used in MPC cannot. Therefore, a direct linearization is used to develop planes tangent to the KOZ surface. [31] This keeps the constraint convex allowing for use in convex optimization solvers. The constraint in Eq. 3.12, again using only position state information, is replaced with the form in Eq. 3.14

$$\mathrm{f} = (\mathrm{X_i} - \mathrm{X_{koz}})^{\mathrm{T}} \mathrm{V} (\mathrm{X_i} - \mathrm{X_{koz}}) \tag{3.13}$$

$$1 \leqslant \mathrm{f}(\mathrm{X_0}) + \frac{\partial \mathrm{f}}{\partial \mathrm{X}}^{\mathrm{T}} (\mathrm{X_i} - \mathrm{X_0}) \tag{3.14}$$

In this form $\mathrm{X_0}$ represents the points in the trajectory about which the constraints are linearized. When the problem is adapted to MPC these linearization points represent each timestep within the prediction horizon. This is demonstrated visually in Figure 7.

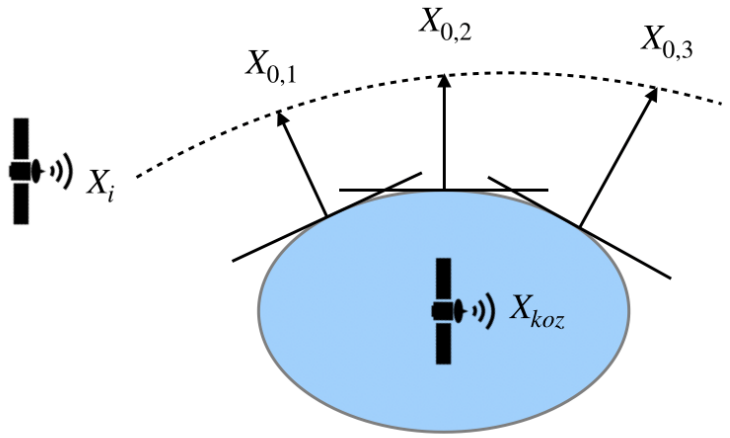

Figure 7. Direct Linearization of the Keep Out Zone about an obstacle.

The lighting condition constraint is a keep in zone (KIZ) in the shape of a second-order cone. This represents a region where the Sun will be at the back of the formation. This ensures that the target formation is always lit and that the agents will be masked by the Sun on approach. The Sun vector will be aligned with the axial vector of the cone, so the constraint can be represented in the form of Eq. 3.15 where X represents the positions of agent and target formation, and γ represents the half angle of the constraint cone.

$$\hat{\mathbf{v}}_{\odot}^{\mathrm{T}}(\mathrm{X_B} - \mathrm{X_R}) \leqslant ||\mathrm{X_B} - \mathrm{X_R}||\cos\gamma \tag{3.15}$$

However, due to the Euclidian norm in this form, the constraint is again nonlinear. To linearize this constraint the cone can be represented as n planes. An algorithm to develop the n planes follows:

1. Begin with the first normal vector $\hat{\mathbf{n}} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^{\mathrm{T}}$

2. Rotate the normal vector $\hat{\mathbf{n}}$ about the two axis by the half angle γ

3. For the n-1 additional planes rotate the previously found normal vector $\hat{\mathbf{n}}$ about the one axis by $\frac{2\pi}{n}$

4. Determine the rotation that will map the assumed axial vector of the cone, $\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^{T}$, to the Sun vector $\hat{\mathbf{v}}_{\odot}$ and rotate the previously found n planes by this rotation.

This algorithm determines n planes to represent the surface of the cone. The constraint in Eq. 3.15 can then take the form in Eq. 3.16

$$-\hat{\mathbf{n}}^{T}(X_i - X_j) \leqslant 0 \tag{3.16}$$

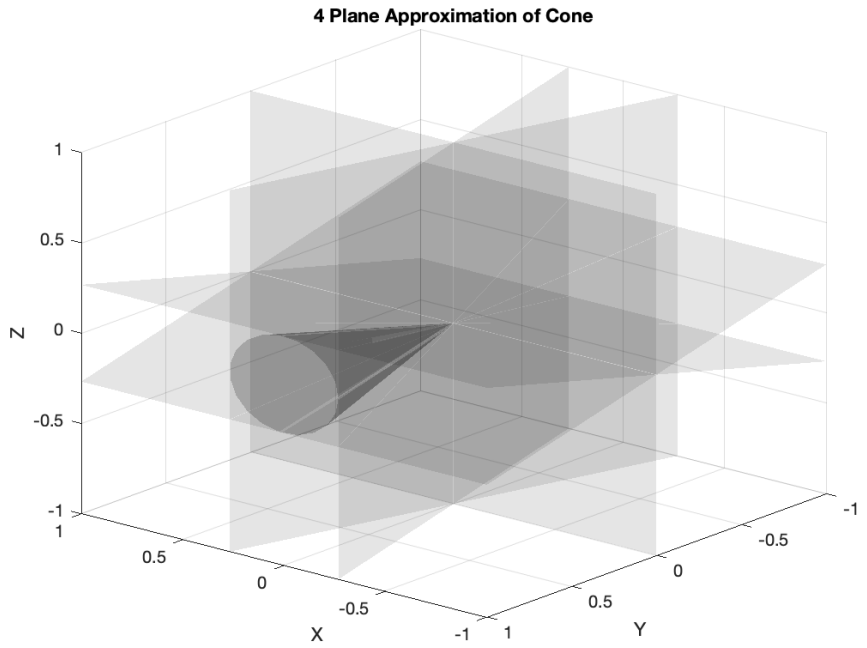

Figure 8. A 4 plane approximation of a second-order cone.

The final constraints are used to bound the relative position and velocities of the target orbit (Eq. 3.17). This ensures the linear HCW model used in the MPC

controller is accurate. Additionally constraints are placed onto the control to limit the magnitude of thrust in any principal direction an agent can supply at a given time (Eq. 3.18).

$$X_{lb} \leqslant X_i \leqslant X_{ub} \tag{3.17}$$

$$U_{lb} \leqslant U_i \leqslant U_{ub} \tag{3.18}$$

3.3 Adapting the Problem to MPC

The problem above can be solved using any optimal control solver. GPOPS II is used by the author [6]. However, to allow for real-time solutions the problem can be adapted for use in a convex QP solver to take advantage of MPC. MATLABs quadprog is used to solve the problem once Eq. 3.3 is adapted in the form of a quadratic program.

The first challenge the cost function (Eq. 3.3) poses is the desired position relative to the centroid, X_d, is dependent on the location of the centroid (X_B). To alleviate this dependency, break the problem into an outer and an inner loop. The outer loop will determine a trajectory that the centroid should follow that will minimize control and the inner loop will add the prescribed offset to the centroid. This results in two optimization problems, a regulation problem for the centroids of the formations and a tracking problem for individual agents. The bounding constraints are levied on each problem, but the KIZ is supplied solely to the outer loop while the KOZ is supplied solely to the inner loop. The result is the following outer (Eq. 3.19-Eq. 3.25) and

inner loop problems (Eq. 3.26-Eq. 3.32).

$$\mathcal{J}_{\text{outer}} = \int_{t_0}^{t_f} (\bar{X}_B - \bar{X}_R)^T Q (\bar{X}_B - \bar{X}_R) + \bar{U}^T R \bar{U} dt \tag{3.19}$$

$$\text{s.t. :} \tag{3.20}$$

$$\dot{X}_B = AX_B + BU_B \tag{3.21}$$

$$\dot{X}_R = AX_R \tag{3.22}$$

$$X_{lb} \leqslant X_B \leqslant X_{ub} \tag{3.23}$$

$$U_{lb} \leqslant U_B \leqslant U_{ub} \tag{3.24}$$

$$\hat{\mathbf{v}}_{\odot}^T (X_B - X_R) \leqslant ||X_B - X_R|| \cos\gamma \tag{3.25}$$

$$\mathcal{J}_{\text{inner}} = \int_{t_0}^{t_f} (\bar{X}_i - \bar{X}_d)^T Q_f (\bar{X}_i - \bar{X}_d) + (\bar{X}_i - \bar{X}_j)^T Q_r (\bar{X}_i - \bar{X}_j) + \bar{U}^T R \bar{U} dt \tag{3.26}$$

$$\text{s.t. :} \tag{3.27}$$

$$\dot{X}_i = AX_i + BU_i \tag{3.28}$$

$$\dot{X}_j = AX_j \tag{3.29}$$

$$X_{lb} \leqslant X_i \leqslant X_{ub} \tag{3.30}$$

$$U_{lb} \leqslant U_i \leqslant U_{ub} \tag{3.31}$$

$$1 \leqslant f(X_0) + \frac{\partial f}{\partial X}^T (X_i - X_{koz}) \tag{3.32}$$

With the problem broken up into two loops the user is ready to convert it into a discrete quadratic form. Begin by performing a discrete transform on the continuous forced dynamics of the system. Using the HCW dynamics previously discussed (Eq. 2.18) a future state can be predicted using the current state, a prescribed input, and

a difference in time (Eq. 3.35),

$$
\begin{aligned}
A_d &= e^{A(t\text{-}t_0)} && (3.33)\\
B_d &= \int_{t_0}^{t} e^{A\tau} B \, d\tau && (3.34)\\
x_{k+1} &= A_d(\Delta t)x_k + B_d(\Delta t)u_k. && (3.35)
\end{aligned}
$$

To predict forward after successive unique inputs from a current position arrange the discrete dynamics and input matrices in the following form (Eq. 3.36). It is important to note the difference in time ($t\text{-}t_0 \equiv \Delta t$)shown in Eq. 3.35 is implied and kept constant in the following equations.

$$
\begin{aligned}
\bar{X}_{k+1,k+p} &= \Phi x_k + \Omega \bar{U} && (3.36)\\
\Phi = \begin{bmatrix} A_d \\ A_d^2 \\ \vdots \\ A_d^p \end{bmatrix} \qquad \Omega &= \begin{bmatrix} B_d & \cdots & 0 \\ A_d B_d & \ddots & \vdots \\ \vdots & & \vdots \\ A_d^p B_d & A_d B_d & B_d \end{bmatrix} && (3.37)
\end{aligned}
$$

Assuming that the target formation is not maneuvering the error between the two formations can be predicted by finding the difference between the current states of each and adding the input of the agents' formation (Eq. 3.44). Following the process outlined by Brand [32] the quadratic form of the outer loop optimization problem can be derived.

$$\mathcal{J}_{\text{outer}} = \frac{1}{2}\bar{\mathrm{U}}^{\mathrm{T}}\mathcal{H}\bar{\mathrm{U}} + \mathrm{f}^{\mathrm{T}}\bar{\mathrm{U}} \tag{3.38}$$

s.t.: (3.39)

$$\mathrm{A_{con}}\bar{\mathrm{U}} \leqslant \mathrm{b_{con}} \tag{3.40}$$

where: (3.41)

$$\mathcal{H} = 2(\Omega^{\mathrm{T}}\mathrm{L}_1\Omega + \mathrm{L}_2) \tag{3.42}$$

$$\mathrm{f}^{\mathrm{T}} = 2(\bar{\mathrm{E}}_{\mathrm{k}}^{\mathrm{T}}\Phi^{\mathrm{T}}\mathrm{L}_1\Omega) \tag{3.43}$$

$$\bar{\mathrm{E}}_{\mathrm{k+1,k+p}} = \Phi(\bar{\mathrm{X}}_{\mathrm{B_k}} - \bar{\mathrm{X}}_{\mathrm{R_k}}) + \Omega\bar{\mathrm{U}} \tag{3.44}$$

$$\mathrm{L}_1 = \begin{bmatrix} \mathrm{Q} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathrm{P} \end{bmatrix} \mathrm{L}_2 = \begin{bmatrix} \mathrm{R} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathrm{R} \end{bmatrix} \tag{3.45}$$

$$\mathrm{A_{con}} = \begin{bmatrix} \Omega \\ -\Omega \\ \mathrm{I} \\ -\mathrm{I} \\ -\mathrm{N}\Omega \end{bmatrix} \mathrm{b_{con}} = \begin{bmatrix} \mathrm{X_{ub}} - \Phi\mathrm{X_{B_k}} \\ -\mathrm{X_{lb}} + \Phi\mathrm{X_{B_k}} \\ \mathrm{U_{ub}} \\ -\mathrm{U_{lb}} \\ \mathrm{N}\Phi\bar{\mathrm{E}}_{\mathrm{k}} \end{bmatrix} \tag{3.46}$$

To find the terminal weighting matrix, P, in Eq. 3.45, use the positive definite solution to the algebraic Riccati equation (Eq. 3.2). This ensures that the solution to the optimization problem in Eq. 3.38 is asymptotically stable even if the rendezvous occurs outside the prediction horizon. The matrix, N, holds the normal vectors of the n planes used to construct the KIZ. This matrix can be developed by predicting where the Sun vector will be in the future at each discrete timestep, using the previously described algorithm to find the normal vectors, and constructing a block diagonal

matrix of the transposed vectors (Eq. 3.48). The future position of the Sun vector can be predicted by using a rotation about the third axis by the product of the mean motion, n, of the chief's orbit and the difference between the future and present time (Eq. 3.47).

$$\hat{\mathbf{v}}_{\odot_{k+1}} = \begin{bmatrix} \cos(n\Delta t) & \sin(n\Delta t) & 0 \\ -\sin(n\Delta t) & \cos(n\Delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \hat{\mathbf{v}}_{\odot_k} \tag{3.47}$$

$$N = \begin{bmatrix} -\hat{\mathbf{n}}_{k+1}^T & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & -\hat{\mathbf{n}}_{k+p}^T \end{bmatrix} \tag{3.48}$$

This completes the conversion of the outer loop into the discrete quadratic form used by MATLAB's interior point solver `quadprog`. Supplying the centroid of the agent and target formations will result in a control that will guide the members of the formation towards the target formation while minimizing the total control of the agents according to Eq. 3.52. Additionally, agents adhere to state bounds, control bounds, and remain within the KIZ to ensures favorable lighting conditions. This control can be used to generate reference trajectories for each of the agents in the formation (Eq. 3.49). The inner loop will minimize each agent's control while balancing the tracking of this reference trajectory with the rendezvous of the target supplied to the agent by the assignment algorithm.

$$\bar{\Gamma} = \Phi \bar{X}_{B_k} + \Omega \bar{U}^* + \bar{X}_d \tag{3.49}$$

The reference trajectory, $\bar{\Gamma}$ can be thought to contain two parts. The homogeneous part built from the formation's centroid and the optimal control, and the unique part

dependent on an agent's desired offset from the virtual center. It is important the user characterizes the patrol formation's virtual structure in a manner that ensures the centroid is the origin. For example, in the studies completed in this work the desired offsets of the agents are:

$$\bar{\mathrm{X}}_{\mathrm{d}} = \left\{ \begin{bmatrix} 10\text{ m} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 10\text{ m} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \text{-}10\text{ m} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \text{-}10\text{ m} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \right\} \tag{3.50}$$

Before adapting the inner loop of the optimal control problem to the discrete quadratic form the user should use the homogeneous portion of the reference trajectory to determine the scaling at each prediction step in the horizon using Eq. 3.10. In doing this, Kronecker products (Eq. 3.51) may be used to create the time variant weighting matrices used to formulate the discrete quadratic form (Eq. 3.60 - 3.62). Again, by following the process outlined in [32] the user can develop the discrete quadratic form of the inner loop MPC controller (Eq. 3.52).

$$\mathrm{A} \bigotimes \mathrm{b} = \begin{bmatrix} \mathrm{A}_{11}\mathrm{b} & \cdots & \mathrm{A}_{1\mathrm{n}}\mathrm{b} \\ \vdots & \ddots & \vdots \\ \mathrm{A}_{\mathrm{m}1}\mathrm{b} & \cdots & \mathrm{A}_{\mathrm{mn}}\mathrm{b} \end{bmatrix} \tag{3.51}$$

$$\mathcal{J}_{\text{inner}} = \frac{1}{2}\bar{\mathrm{U}}^{\mathrm{T}}\mathcal{H}\bar{\mathrm{U}} + \mathrm{f}^{\mathrm{T}}\bar{\mathrm{U}} \tag{3.52}$$

s.t.: (3.53)

$$\mathrm{A_{con}}\bar{\mathrm{U}} \leqslant \mathrm{b_{con}} \tag{3.54}$$

where: (3.55)

$$\mathcal{H} = 2(\Omega^{\mathrm{T}}(\mathrm{L}_2 + \mathrm{L}_3)\Omega + \mathrm{L}_1) \tag{3.56}$$

$$\mathrm{f}^{\mathrm{T}} = 2(\bar{\mathrm{X}}_{\mathrm{i_k}}^{\mathrm{T}}\Phi^{\mathrm{T}}(\mathrm{L}_2 + \mathrm{L}_3)\Omega - \Gamma^{\mathrm{T}}\mathrm{L}_2\Omega - \bar{\mathrm{X}}_{\mathrm{j_k}}^{\mathrm{T}}\Phi^{\mathrm{T}}\mathrm{L}_3\Omega) \tag{3.57}$$

$$\bar{\mathrm{X}}_{\mathrm{i_{k+1,k+p}}} = \Phi\bar{\mathrm{X}}_{\mathrm{i_k}} + \Omega\bar{\mathrm{U}} \tag{3.58}$$

$$\bar{\mathrm{X}}_{\mathrm{j_{k+1,k+p}}} = \Phi\bar{\mathrm{X}}_{\mathrm{j_k}} \tag{3.59}$$

$$\mathrm{L}_1 = \mathrm{I} \bigotimes \mathrm{R} \tag{3.60}$$

$$\mathrm{L}_2 = \begin{bmatrix} \bar{\mathrm{s}}_{\mathrm{k+1,k+p\text{-}1}} \bigotimes \mathrm{C}^{\mathrm{T}}\mathrm{QC} & 0 \\ 0 & \mathrm{s_{k+p}}\mathrm{C}^{\mathrm{T}}\mathrm{PC} \end{bmatrix} \tag{3.61}$$

$$\mathrm{L}_3 = \begin{bmatrix} (1 - \bar{\mathrm{s}}_{\mathrm{k+1,k+p\text{-}1}}) \bigotimes \mathrm{Q} & 0 \\ 0 & (1 - \mathrm{s_{k+p}})\mathrm{P} \end{bmatrix} \tag{3.62}$$

$$\mathrm{A_{con}} = \begin{bmatrix} \Omega \\ -\Omega \\ \mathrm{I} \\ -\mathrm{I} \\ \mathrm{N}\Omega \end{bmatrix} \quad \mathrm{b_{con}} = \begin{bmatrix} \mathrm{X_{ub}} - \Phi\mathrm{X_{B_k}} \\ -\mathrm{X_{lb}} + \Phi\mathrm{X_{B_k}} \\ \mathrm{U_{ub}} \\ -\mathrm{U_{lb}} \\ -\mathrm{d} - \mathrm{N}\Phi\bar{\mathrm{X}}_{\mathrm{i_k}} \end{bmatrix} \tag{3.63}$$

The matrix, C, is used to exclude the velocity states of the reference trajectory. This is necessary because the agent is meant to rendezvous with the target, so all states must be considered. However, the agent is only attempting to track the trajectory. The difference in the constraint matrix and vector, $\mathrm{A_{con}}$ and $\mathrm{b_{con}}$ respectively, is a result of the KOZ constructed around the other members of the formation. Beginning with the previously described constraint (Eq. 3.12), the user does a Taylor series ex-

pansion of the right side to arrive at the convex version of the constraint (Eq. 3.14). Using knowledge of the previously found control solutions predict forward the positions of the agent and the desired obstacle and develop the relative vectors (Eq. 3.64).

Determine the linearization points by finding where the KOZ's ellipsoid this relative vector intersects (Eq. 3.65). Since $f(X_0) = 1$ by definition in Eq. 3.14 the constraint can be rewritten (Eq. 3.66). The normal vector of the constraint plane can then be solved with Eq. 3.67. After distributing the normal vector and isolating the input, the constraint takes the form in Eq. 3.54.

$$r_{0_{k+1,k+p}} = (\Phi\bar{X}_{i_k} + \Omega\bar{U}_{i_k}) - (\Phi\bar{X}_{j_k} + \Omega\bar{U}_{j_k}) \tag{3.64}$$

$$X_{0_k} = X_{j_k} + \left(\frac{1}{\hat{\mathbf{r}}_{0_k}^T V \hat{\mathbf{r}}_{0_k}}\right)^{1/2} \hat{\mathbf{r}}_{0_k} \tag{3.65}$$

$$-\hat{\mathbf{n}}^T(\bar{X}_i - \bar{X}_0) \leqslant 0 \tag{3.66}$$

$$\hat{\mathbf{n}}_k = \frac{2V(X_{0_k} - X_{j_k})}{||2V(X_{0_k} - X_{j_k})||} \tag{3.67}$$

$$d_k = -\hat{\mathbf{n}}_k^T X_{0_k} \tag{3.68}$$

$$N = \begin{bmatrix} -\hat{\mathbf{n}}_{k+1}^T & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & -\hat{\mathbf{n}}_{k+p}^T \end{bmatrix} \tag{3.69}$$

3.4 Summary

With each aspect of the optimal control problem adapted to MPC the user is able to perform formation rendezvous and reconfiguration missions. The presented framework yields the user several tunable parameters. For simplicity assume that state weighting matrix, Q, is identity. This implies that the control weighting matrix, R, will be the parameter used for shaping the response of the formations. The scale

rate, k, allows the user to adjust how aggressive the formation switches from one profile to the next. The scale centering, S, will shift the range at which the formation makes the transition from the patrol formation to the reconfigured rendezvous formation. The agents communicate their present states, and most updated control solution. With this information and the states of the targets the auction assignment algorithm allocates each agent a target. The MPC controller uses the centroid of each formation to first determine the optimal trajectory for the formation to take to rendezvous with the target formation while maintaining the Sun at its back. Each agent then tracks an offset to this virtual trajectory while avoiding the other members of the formation. As the range between the formations closes, the agents in the formation begin to prioritize their rendezvous scenario over the formation keeping task. The problem has been formulated to ensure real-time solutions can be found using quadratic program optimization software. Figure 9 shows the framework, now including the MPC controller.

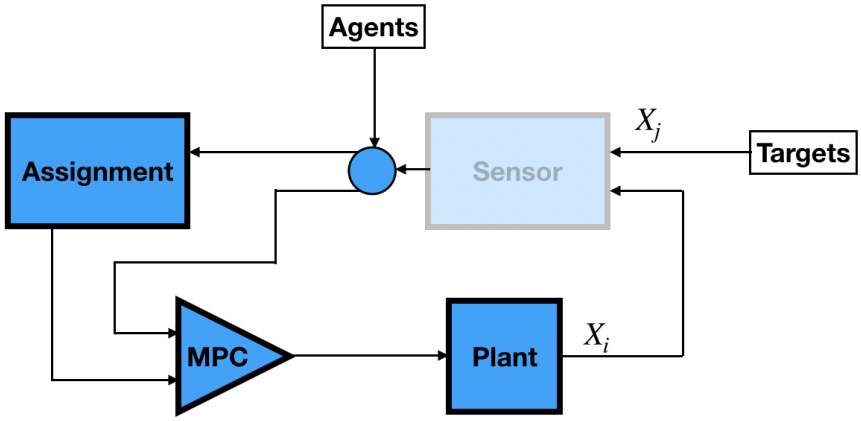

Figure 9. MPC in Control Framework.

IV. Results and Analysis

The following chapter describes the findings of the aforementioned method used to accomplish a formation rendezvous and reconfiguration. A random day in the year 2020 is selected and converted to the Julian date; this date is used to determine the Sun vector ($\hat{\mathbf{v}}_{\odot}$). The initial position of the formation is chosen from the truncated conical region described by the parameters $\psi = 15^{\circ}$, $\mathrm{l} = [.95\ , 1.05]$km. A visual representation can be seen in Figure 10. The result is a toroidal region around the origin from which initial conditions are chosen at random. One hundred randomly initialized simulations are ran with scale rate ($\mathrm{k} = 100\frac{1}{\mathrm{km}^2}$), scale centering ($\mathrm{S} = \sqrt{.1}\mathrm{km}$), and control weighting ($\mathrm{R} = 10^6\mathrm{I}$). The approach corridor constraint is described by the half angle ($\gamma = 20^{\circ}$), and the collision avoidance constraint is parameterized by an ellipsoid with semi-axes 1 meter in each direction. Control bounds are imposed at $\pm 10\frac{\mathrm{cm}}{\mathrm{s}^2}$ in each principal direction. The relative position and velocity bounds are ± 100km and $\pm 10\frac{\mathrm{m}}{\mathrm{s}}$ respectively.

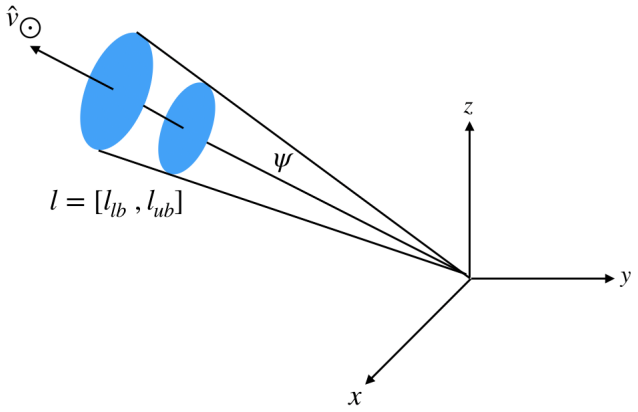

Figure 10. Truncated conical region to choose initial conditions at random

4.1 Solution Time

In order for the framework to be viable for use on flight hardware both the assignment algorithm and the MPC optimization problem must be completed before the opportunity to administer a controlling action has passed. The duration of both the auction algorithm and the QP solver is recorded during the simulations. These solution times are displayed in Figure 11. The controller is executing both algorithms at a frequency of 0.1 Hz, therefore the controller has 10 seconds to determine the assignment and control for an agent. Figure 11 is made up of box and whisker plots from all agents across the one hundred simulations, so each box and whisker plot represents 400 data points for a given time step in the simulation. The bull's-eye represents the median solution time, while the solid blue box represents the lower and upper bounds of the 25th and 50th percentile. The whiskers illustrate the range of solution times across the simulations. It is clear that the auction algorithm is consistent throughout the mission profile, while the MPC solver decreases as the formation approaches its target. There is a noticeable bump approximately 100 seconds into the mission. This is likely the result of differences in the approach profile of the formation in order to take advantage of the natural dynamics as much as possible. Additionally, the problem becomes more difficult to solve as the formation balances between the two profiles rather than solely focusing on one.

The simulations are ran on a MacBook Pro with a 2.6 GHz 6-Core Intel Core i7 processor and 16 GB of RAM, however, it is unlikely the satellite will have similar computational specifications. Assuming the satellite does have the previously described computational power the user could confidently run the existing program which predicts 75 10-second time-steps into the future at a higher frequency near 1 Hz according to the whiskers in Figure 11. However, the user may also elect to increase

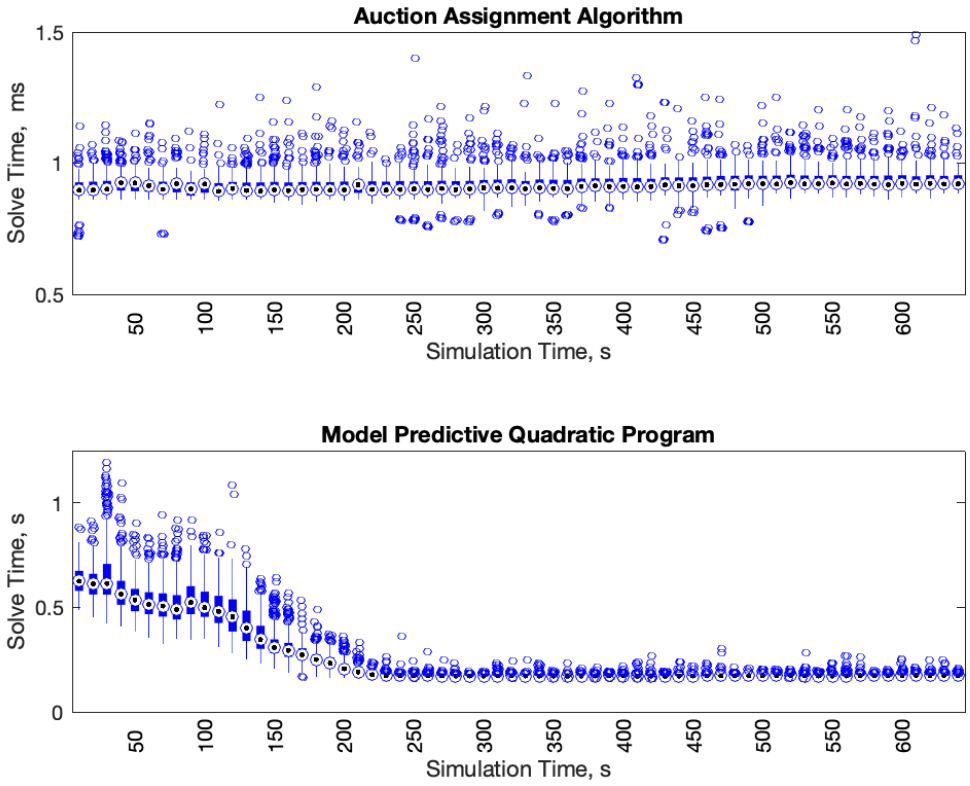

Figure 11. Solution time of Auction Algorithm and MPC QP solver.

the prediction horizon, or decrease the discretization interval in an effort to develop a more considerate prediction. Additionally, if an outlier occurs and the program has not completed in the allotted solution time, the controller has knowledge of the previous prediction. This can be used to ensure a control is delivered in the event that the controller could not solve the problem in the allotted time. The process of determining the hyper-parameters of the MPC controller to account for real time solutions is extremely specialized based on the user's hardware, software, and mission needs, so no additional detail pertaining to this matter is described in this thesis.

The auction algorithm successfully achieves optimal task assignment. In a trial of ten thousand randomized cost matrices the auction was compared to the Hungarian

algorithm. The auction algorithm found the correct assignment 99.81% of the time. Additionally, the duration of the auction compared to the duration of the Hungarian assignment was significantly less. This can be seen in Figure 12 and Table 4. Regardless of the size of the cost matrix, the time of the auction algorithm is outside of six standard deviations of the Hungarian algorithm's duration. This is a result of the computational complexity of the two algorithms. The Hungarian algorithm was originally $O(n^4)$, however, over the years it has been optimized to be $O(n^3)$[5]. This is still significantly slower than the auction algorithm's $O(n^2 \log n)$

Table 4. Computation Time Mean and Variance of 100 Random Cost Matrices of Size nxn

	Comparison of Algorithms			
	Auction		Hungarian	
size	mean, μs	std, σ	mean, μs	std, σ
4	142.8	0.1993	327.14	0.0054
10	262.5	0.4061	1,004.5	0.0331
20	577.0	0.7035	2,217.2	0.1944
50	1,823.4	2.4600	6,001.8	1.2850
100	6,589	13.3900	14,969	6.7725

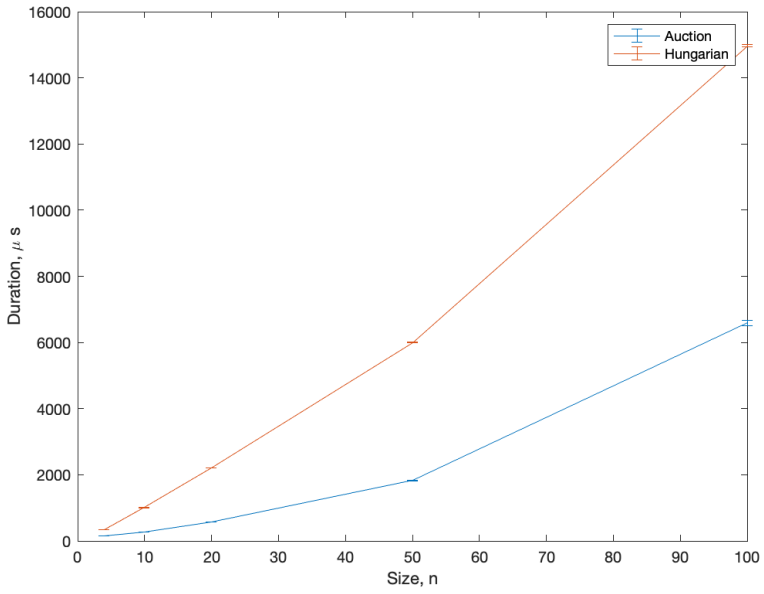

Figure 12. A Comparison of Auction and Hungarian Assignment Algorithms with 6σ Errorbars

4.2 Transition Between Profiles

When representing a complex problem in the form of several tuning parameters it is important that the tuning parameters represent independent variables. This suggests that regardless of initial conditions, the tuning parameters should represent similar resulting trajectories. Figure 13 shows how the scale, s, changes throughout the simulations. Again each time-step is represented by a box and whisker plot. It is clear that there is very little variation in the trajectories. The transition between the "patrol" and "target" profile can be determined regardless of initial condition to be within a period of ten seconds. This is shown in Figure 13 by the box and whisker plot at the 130 second interval. Considering the definition of the transition function (Eq. 3.10) and that the initial conditions place formations at a similar range from the target formation, Figure 13 implies that the transition occurs consistently at the surface of a sphere around the target formation defined by the scale centering

parameter (S).

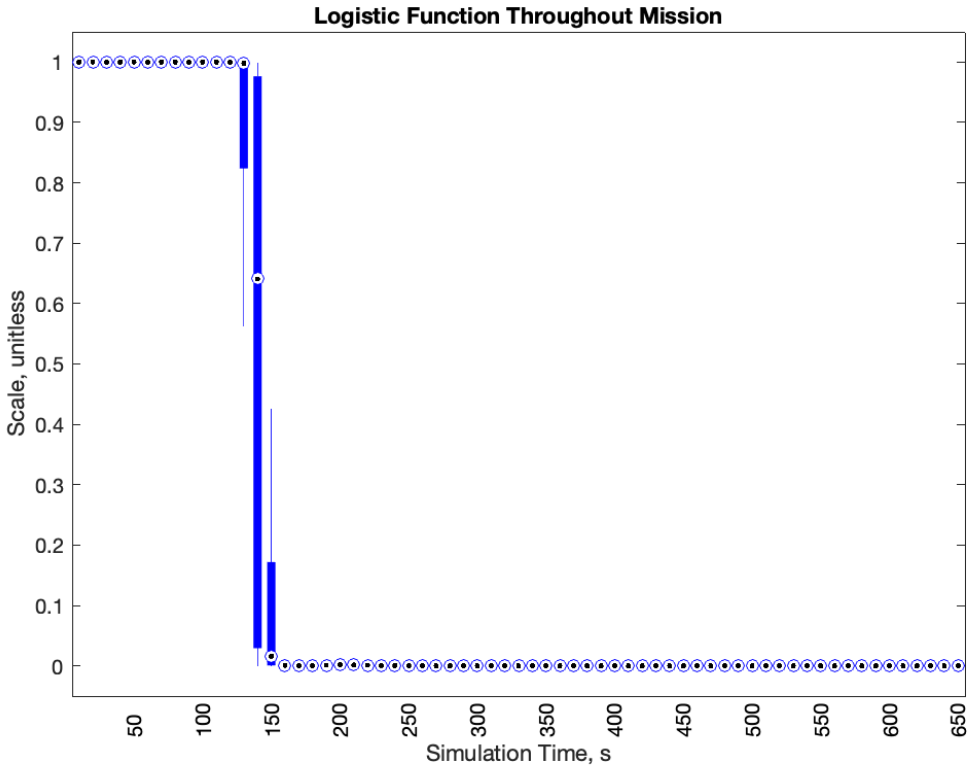

Figure 13. Scales of Agents vs Simulation Time.

With knowledge of approximately where the transition occurs, the performance of the agents with respect to the patrol formation can be assessed by determining the Euclidean distance between an agent and their prescribed location in the patrol formation for times between the beginning of the simulation and the transition point illustrated in Figure 13. The error with respect to the "patrol" formation is only dependent on position and is calculated according to Eq. 4.1.

$$e_p = ||\bar{X}_i - \bar{X}_d||_2 \tag{4.1}$$

Figure 14 shows the distribution of error an agent experiences throughout the first phase of the mission approaching the transition. Taken out of context any controls

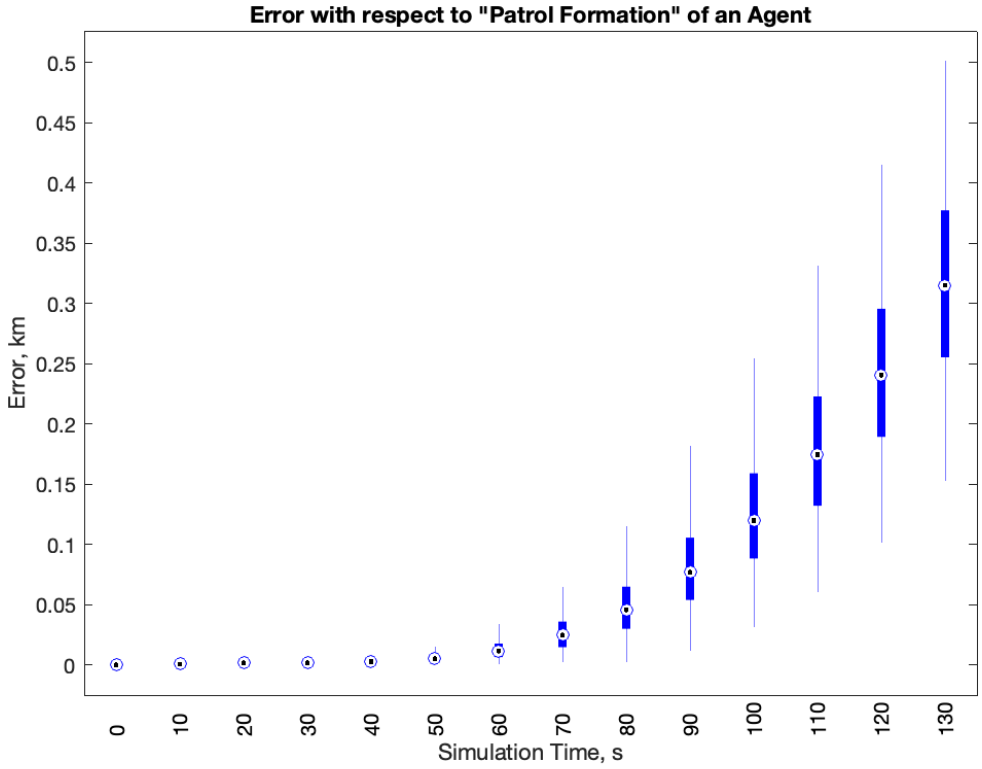

Figure 14. Euclidean Distance from X_d of an agent prior to phase transition.

engineer would find the trend in Figure 14 unsettling. However, there are two important subtleties depicted in this plot. As the time approaches zero the error approaches zero, and until approximately 60 seconds there is no variance in the error with respect to the "patrol" formation. This implies that if the formation is at sufficiently large ranges from its target, all members can inhabit the patrol formation without error. Finally, and of the most importance, the formation is able to predict forward and optimize past the transition horizon. The error increase with respect to the "patrol" formation is the result of the MPC controller understanding the transition must begin prior to the switch in order to account for the transition to rendezvous with the

assigned target. This becomes clear when the error is analyzed after the transition occurs. Figure 15 demonstrates the agents accomplish the rendezvous after the phase transition. The error with respect to the target formation includes both position and velocity. The majority of the error is eliminated by approximately 210 seconds into the simulation before a slight overshoot by the agents. The entirety of the maneuver is complete by approximately 330 seconds. By concatenating the two figures a visual representation is achieved, and the three distinct parts of the two-phase maneuver can be distinguished. In Figure 16, the user can distinguish the patrol formation (the time period from the start to 70s), the transition (80- 210s), and the target formation(220s to the end of the simulation). Table 5 parameters the distribution of the error in the terminal state of the agents with respect to their targets. The Euclidian norm of the difference between the agent and target positions and velocity is used.

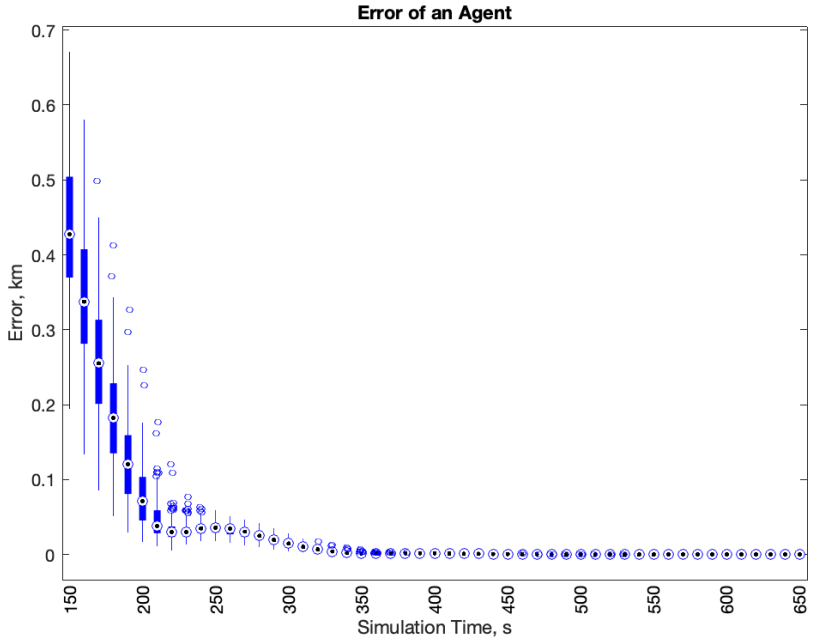

Figure 15. Euclidean distance from the assigned target of an agent after phase transition.

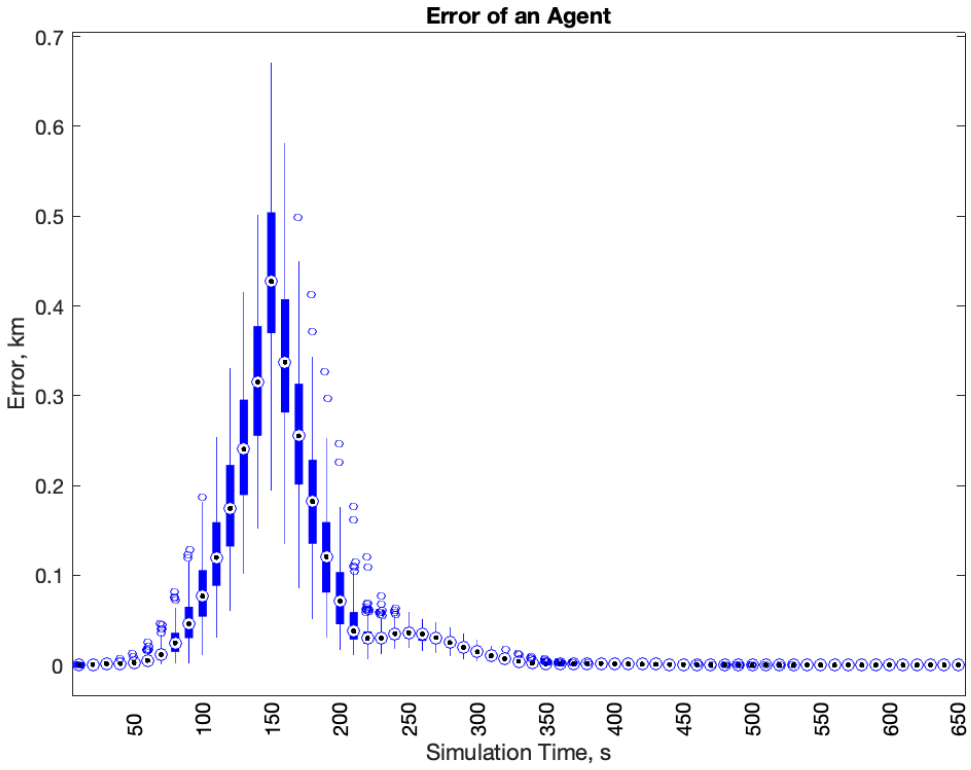

Figure 16. Euclidean distance from the dominating state of an agent throughout mission.

Table 5. Tracking Error and Percent Error of each agent's MPC trajectory compared to GPOPS trajectory.

	Terminal Error of Agents	
Agent	Tracking Error, cm	Standard Deviation, σ
1	10.78	2.93
2	10.59	2.93
3	8.56	2.98
4	8.65	2.99

4.3 Satisfaction of Constraints

To visualize the collision avoidance constraint the relative range of an agent is divided from the magnitude of the relative range vectors intersection with the constraint

surface (Eq. 4.2).

$$\rho_N = \frac{\rho_{con}}{||\bar{X}_i - \bar{X}_j||_2} \tag{4.2}$$

As previously described the constraint surface in the simulations is a sphere with a radius of 1 m, so if an agent is closer than this they would be considered inside the KOZ and the constraint is considered violated. This would present itself visually in the form of a data point greater than 1 on Figure 17 which has neglected data points after the transition as the relative ranges become large with the rendezvous of the formations. However, if a different target formation was used the data points may become relevant. There are no such points in Figure 17, and in fact, the user can determine that the majority of the agents inhabit a normalized relative range of 0.0707 m/m. When considering the structure of the "patrol" formation (Eq. 3.6) this is the range of two adjacent members of the formation. Occasionally, the agents do come in closer proximity with one another; the closest approach in all simulations is 10.7 m.

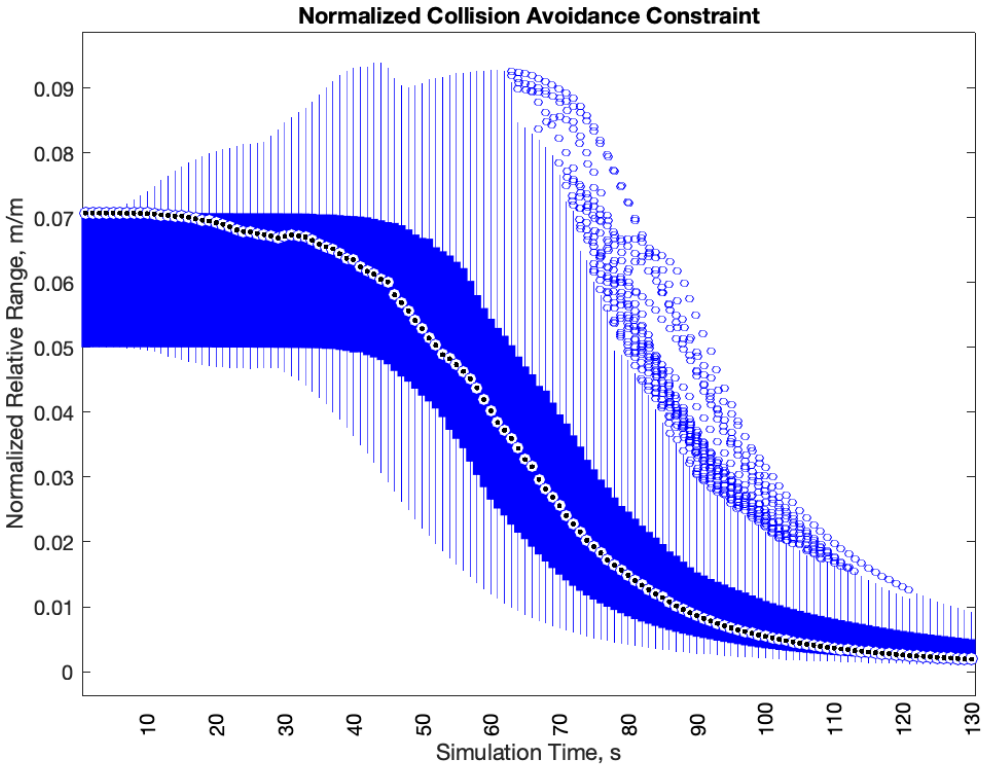

Figure 17. Constraint normalized relative range of an agent to other members of the formation.

The approach corridor constraint uses Vallado's script [33] to determine the Sun vector at each time in the simulation. This is then used with the agent formation's centroid to determine the angle between the Sun and the formation's relative position. Figure 18 shows how this angle evolves throughout the simulations. The black line represents the half angle which bounds the constraint, and it is clear the formation does not satisfy this constraint for the entirety of the trajectory in all simulations. In 13% of simulations a trajectory exceeds the half angle of the cone, and therefore does not satisfy the constraint. The constraint is generally not satisfied in the last two to three time-steps prior to the transition.

This is the result of the way the outer loop of the controller is setup, the centroid of the formation is a virtual state, and as a result, an individual agent has no control over it. The individual's reference trajectories are built to propagate the virtual centroid in a cooperative manner, however, the virtual state is extremely sensitive to an agent's error relative to the "patrol" formation reference trajectories. As a result, the centroid may not satisfy the constraint as the weighting shifts between phases of the mission. A solution to this problem may be to include a factor of safety when determining the half-angle for the constraint. Figure 18 shows that the constraint is violated, but by adding a factor of safety to the constraint a fewer percentage of trajectories may violate the constraint. Enforcing the constraint on both the inner and outer control loop increased computation time by a large factor, making the problem infeasible for use in real-time systems. Further research is needed into developing a more sophisticated way to ensure the constraint is met at all times.

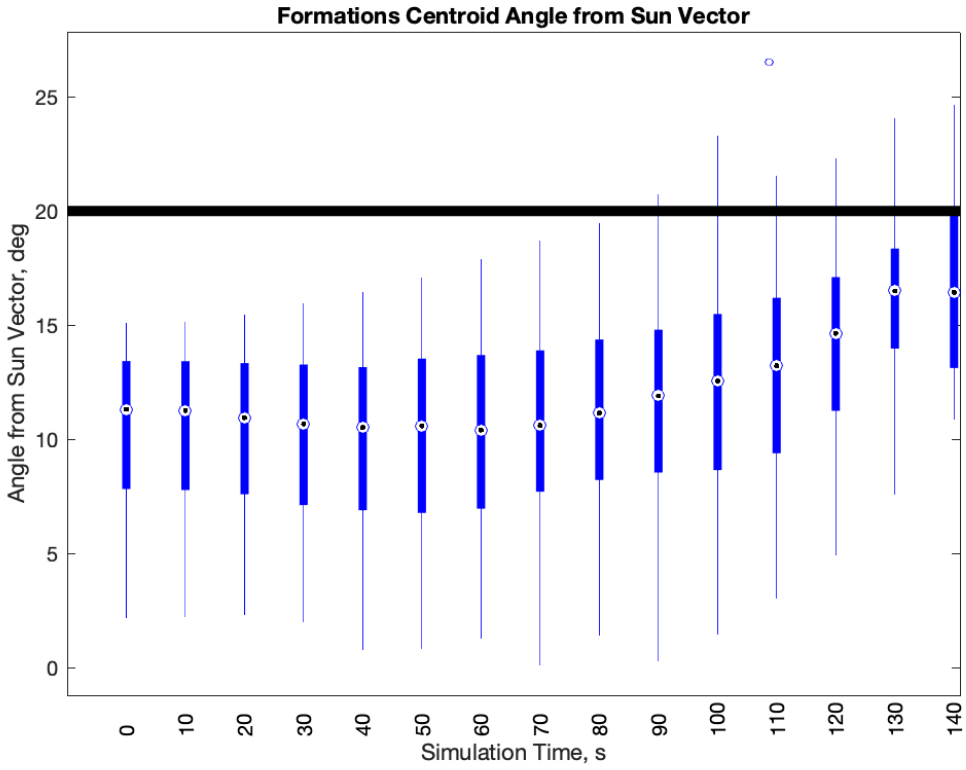

Figure 18. Formation's angle from the Sun vector on approach.

4.4 Comparison to Collocation Technique

To gauge the optimality of the MPC's solution to the optimal control problem the solution is compared to commercial optimal control software; the author's choice is GPOPS which uses a Gaussian quadrature collocation technique [6]. To properly compare the proposed framework to the solution found using GPOPS, the scenario is modified slightly. First, optimal assignment of agents to their respective targets is programmed into the problem. Second the target states are static offsets from the origin, and third the path constraints such as collision avoidance and an approach corridor are neglected. As previously discussed, numerical methods for optimal control do not implement feedback, so the target states are fixed, and if they were to

change the problem would need to be resolved. Formulating the problem this way provides the simplest case of an optimal formation reconfiguration, and represents a problem that MPC may be expected to solve every time-step. To emulate the process outlined in this work, the optimal control software first solves for the regulation of the centroid of the agent formation to the target's centroid. This solution is then used as an initial guess for the full multi-agent system, and agents must track trajectories offset from this centroid-to-centroid maneuver until the switching conditions are met. A cubic spline interpolation of the centroid trajectory is used to provide the proper scaling to the full agent system at any given collocation point.

The solution found using GPOPS is compared to the solution found using MPC using three metrics: control effort, tracking error, and solution time. The control of each solution is integrated over the entire maneuver; trapezoidal numerical integration is done for the GPOPS solution, and due to the zero-order hold of the MPC solution simple rectangular integration is used. The results are shown in Table 6 and Figure 19.

Table 6. Comparision between control effort of GPOPS and MPC framework with percent error of MPC framework when compared to the optimal solution.

	Comparison of Control Effort		
	MPC	GPOPS	Percent Increase
Agent	Control Effort m/s	Control Effort m/s	Percent Error %
1	12.2987	12.0411	2.1391
2	12.2494	12.0944	1.2815
3	12.2472	12.1404	0.8790
4	12.3114	12.0878	1.8500

It is clear from the table that the MPC solution is sub-optimal, however, this is expected. By definition, MPC is sub-optimal, and in the scenario used to compare the two solutions the shape of the control signals, shown in Figure 19 are similar. The difference in control effort is on the order of a percent or two, so MPC represents a good approximation of the optimal control. The trajectories exhibit a similar shape as shown in 20, however, there are slight differences in the control in Figure 19 and these differences result in different trajectories. This can be seen in the tracking error Figure 21.

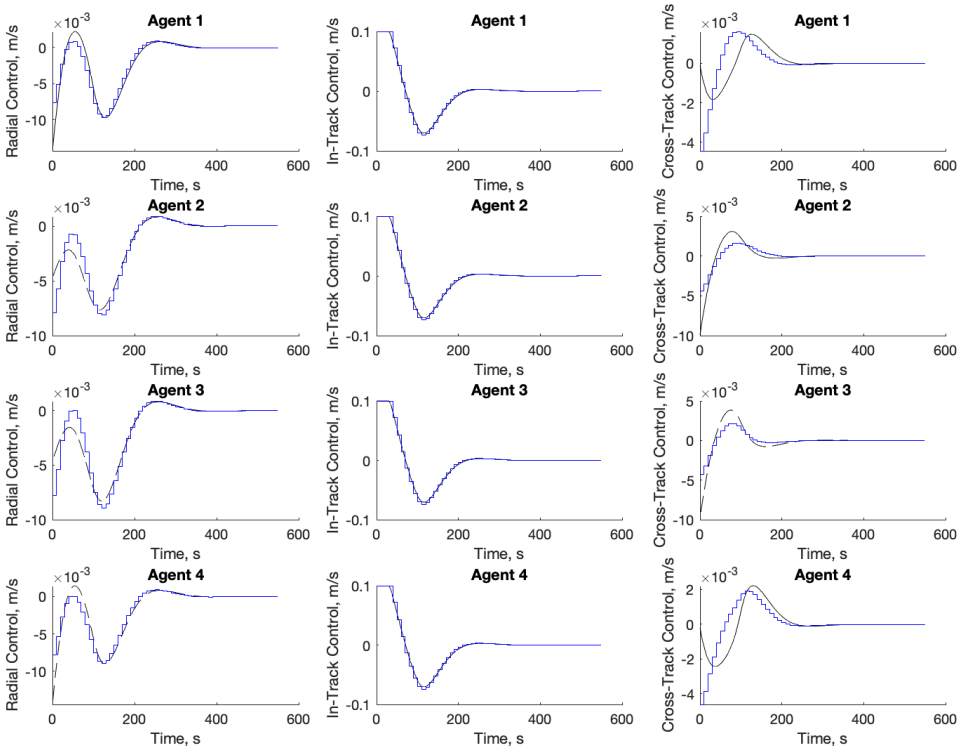

Figure 19. Comparison of control signals between GPOPS an MPC.

There is a pattern in the tracking error, the error is the same across the in-track direction for all agents. Given that the bulk of the maneuver occurs in this direction, as shown in both Figure 19 and 20 this is likely the result of error in the centroid-

to-centroid aspect of the problem. There are also small differences in the radial and cross-track directions. This is a result of the optimization weighting; because control is heavily penalized, GPOPS results in some error with respect to the patrol formation to use slightly less control compared to MPC.

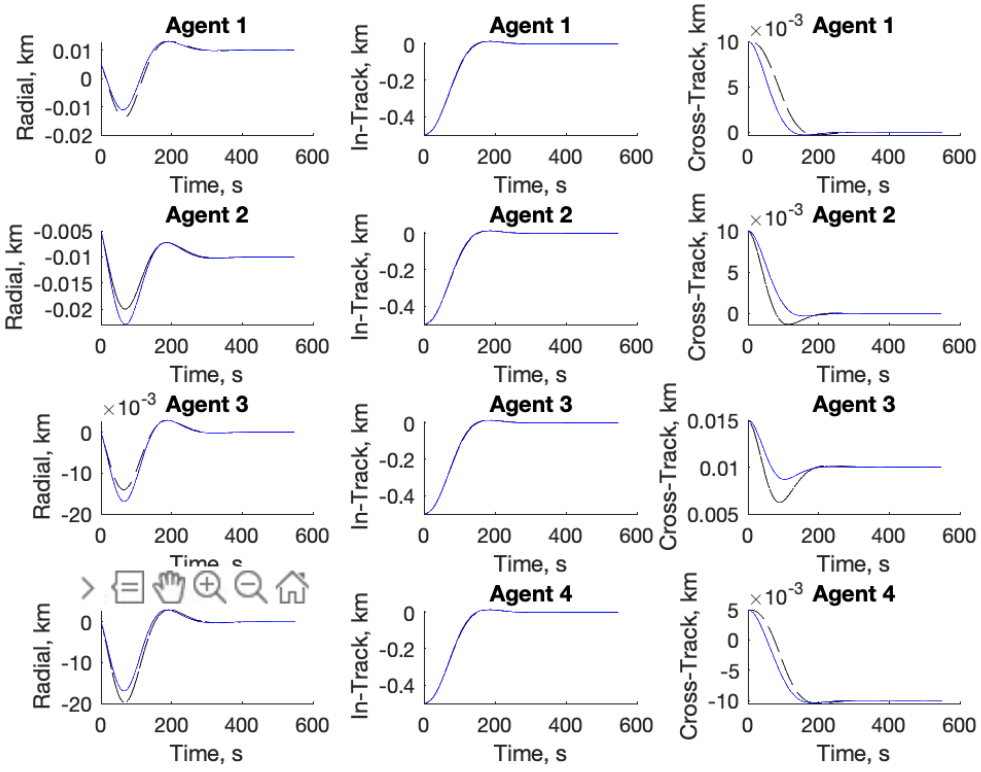

Figure 20. Comparison of trajectories between GPOPS an MPC.

The solution time is the greatest difference between the two methods. As previously discussed the solution time of the MPC framework is approximately one second, and this is conducted at a frequency of 0.1 Hz so feedback control can be implemented. The solution time for the GPOPS program is approximately 140 seconds. Additionally, it is important to note that this simplified version of the problem contains no path constraints and is solving for the entire maneuver which is conducted over the

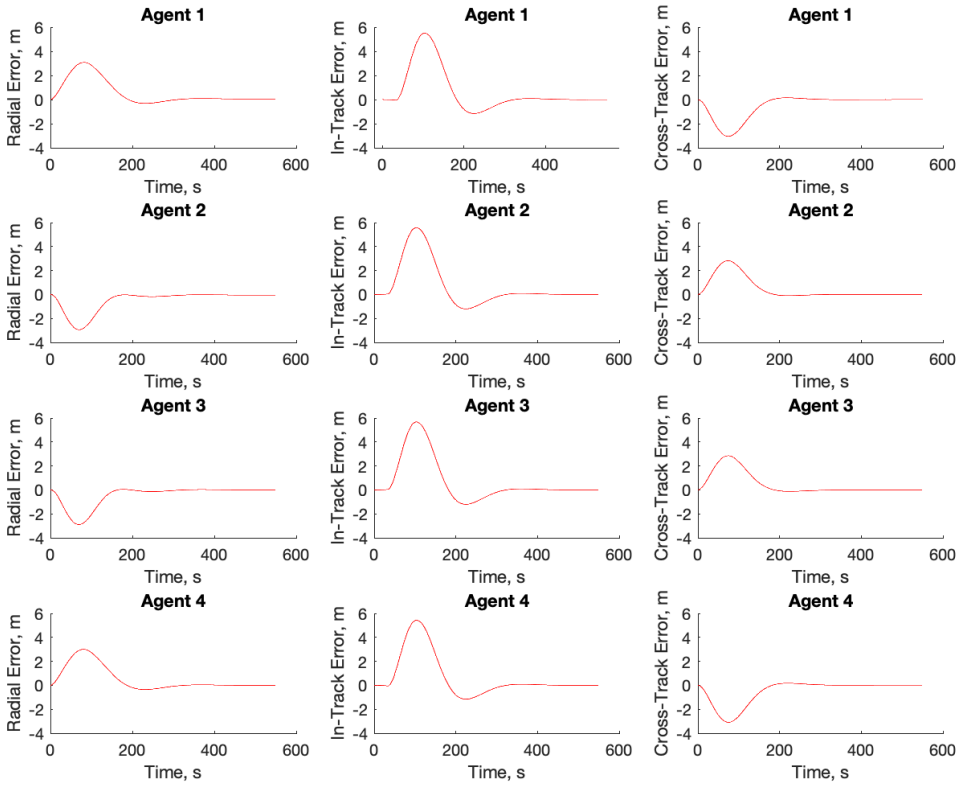

Figure 21. Trajectory errors between GPOPS and MPC.

course of approximately 550 seconds. The prediction horizon of the MPC controller is 750 seconds, so it is effectively determining the course of action for the entire maneuver as well, with additional constraints to maintain safety and promote a successful reconfiguration in less than 1% of the time. This is accomplished with the added benefit of maintaining approximately 98% of the optimal solution.

The control signal of the GPOPS and MPC solutions are interpolated using a cubic spline, and this information is used to verify the system using MATLAB's `lsim`. The error of the solutions, shown in Table 7, are negligible. The error is determined by using the interpolated control with a numerical integration of the dynamics. When the simulation is complete the difference in the terminal states of the GPOPS and

MPCs solutions compared to the desired terminal condition is taken, a Euclidian norm is used to represent this vector as a scalar.

Table 7. Terminal Error of each agent's MPC and GPOPS trajectory.

	Terminal Error of Agents	
Agent	MPC Error, cm	GPOPs Error, cm
1	24.79	16.35
2	24.80	17.59
3	8.26	18.11
4	8.26	15.84

4.5 Summary

One hundred simulations are performed. In these simulations it is determined that the optimal assignment and MPC controller are able to run the program in a time period fast enough to meet the 0.1 Hz update requirement. In fact, the update frequency could be increased to closer to 1 Hz according to simulations. The transition between profiles is successful, as shown in Figure 22, and the MPC controller is able to optimize beyond the phase horizon successfully to conduct the maneuver in a manner that recovers approximately 98% of the optimal solution in less than 1% of the time. This is accomplished with the added benefit of imposing collision avoidance constraints and an approach corridor constraint. However, the approach corridor constraint is not satisfied in 13% of simulations due to the manner in which it is formulated. More research needs to be conducted to determine a more sophisticated way to satisfy this constraint, but a simple solution proposed in this research allows for some factor of safety when selecting the defining parameters of the constraint.

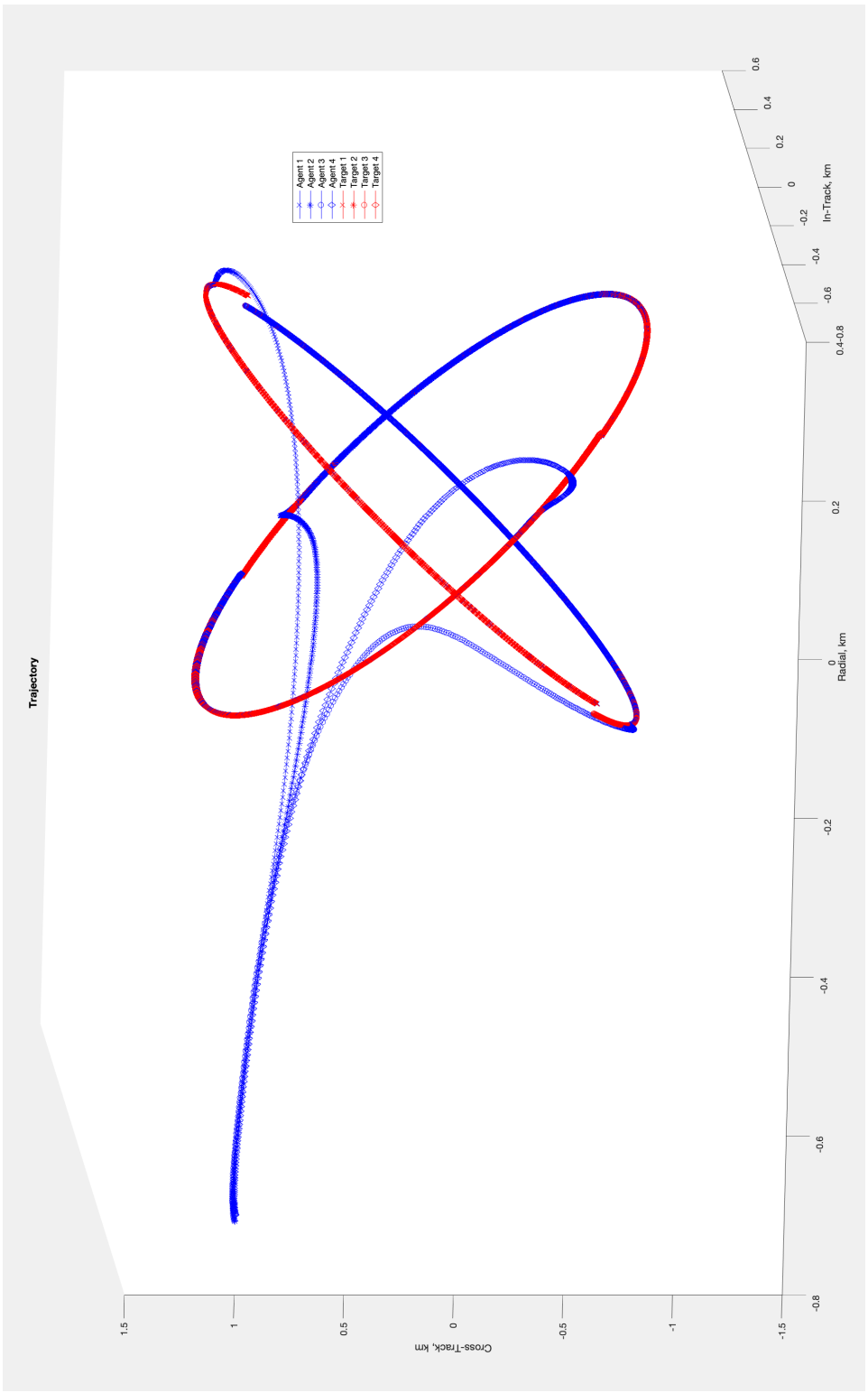

Figure 22. Formation Rendezvous Trajectory

V. Conclusions and Recommendations

This thesis presented a control framework for a formation of satellites. The proposed algorithm provided a solution to optimally task the members of the formation, and provide near optimal controls for each agent which maintains a formation structure before transitioning to a desired configuration. To be successful, the maneuver must be capable of determining the actions of each member at a given update frequency that promotes reactivity to dynamic changes in the target formation in real time. Additionally, safety measures must be enforced to prevent collisions and approach from proper directions. This formation reconfiguration mission is accomplished using a logistic function to change the penalties within the optimization's cost function as the formation approaches the target formation. One hundred simulations of the framework were conducted with randomized initial conditions to determine the performance of the framework in a range of similar scenarios. Chapter 4 described the results of such simulations, and compared a single case to the solution determined through the commercial optimal control software GPOPS.

5.1 Conclusion

The viability of the framework is demonstrated, and results suggest that there is a strong possibility that the framework could be used on hardware to supply control actions to a small formation of satellites. The second-price auction algorithm provides speed advantages over the Hungarian Assignment algorithm, while supplying near optimal assignments, and the MPC controller can determine near optimal control in significantly less time than collocation techniques. The duration of the calculations for a given moment in time suggests that an update frequency of 0.1 Hz could be accomplished with time to spare should the hardware on orbit need additional time

to perform other tasks, and should the user find it necessary, the update frequency could likely be increased to closer to 1 Hz if measures are put in place to ensure a previous solution could be used in the event a calculation takes more time than permitted to determine the next action.

The user can use this framework with the confidence that the resulting control recovers approximately 98% of the optimal solution if the control were compared to other options for determining the optimal control. Additionally, in using this framework the user gains several tuning parameters that allow them to change the profile of the maneuver in a way that suits their specific mission. The scale rate, k, allows the user to change how quickly the weighting switches from the patrol formation to the desired configuration. This will influence the transition period between the two formation profiles. The scale centering parameter, S, gives the user control on where the transition occurs with respect to the target formation. The control weighting, R, will promote a more aggressive maneuver when small, and a maneuver that is more conscious of control effort as the parameter becomes large. The framework is capable of achieving these controls while considering constraints that may effect the mission in reality such as collision avoidance and proper approaches. However, the approach corridor constraint may need some additional effort on the part of the reader to ensure that conditions are met in their desired scenario.

The significance of these findings is that through the use of this formulation the user gains the benefit, in multi-phase optimization scenarios, of being able to optimize past the phase horizon. This alleviates the harsh transitions that occur in traditional multi-phase scenarios, and results in controls that better approximate the solutions found using collocation techniques without the added computation time. As for spe-

cific relevance to the United States Space Force, and other DoD assets in need of GNC strategies, Appendix A provides the reader with factors that may influence the decisions in what these parameters might be, and how to pick them.

Future work includes a more sophisticated study on the approach corridor constraint, and how it might be parameterized differently to promote a higher success rate in the framework. Other areas of interest would be the introduction of uncertainty to the states of both the agents and targets. Due to the formation aspects of this problem, an estimation scheme that takes advantage of the properties of stereo-vision and angles only navigation would be suggested. In doing this, an optimization problem could be developed to determine the shape characteristics of the patrol formation that may promote the maximum observability of the target formation. Accomplishing this would likely require the inclusion of attitude dynamics to the framework, and this would pose new challenges as well as constraints that must be considered. Additionally, the strengths of machine-learning could be levied to provide the user with the tunable parameters through logistic regression. This would enhance the autonomy of the system by allowing for the tuning parameters to change in reaction to differing behaviors in the target formation. For example, if the target formation is static and tightly clustered the best scale rate, and scale centering may be different than a dynamic or widely distributed formation. In conclusion, the framework outlined in this thesis advances the efforts of others to open the space domain to the recent developments of multi-agent operations, and in doing so poses new challenges that must be solved as the world takes steps to develop autonomous solutions to space operations. This framework suggests an option that would likely be suited for use on-board hardware, and considers the mission constraints that may be present.

Bibliography

[1] N. Air and S. I. Center, *Competing in Space*. https://media.defense.gov/2019/Jan/16/2002080386/-1/-1/1/190115-F-NV711-0002.PDF.

[2] T. W. House, *National Security Strategy*. https://www.whitehouse.gov/wp-content/uploads/2017/12/NSS-Final-12-18-2017-0905.pdf.

[3] J. A. S. et al, "Spacecraft autonomy challenges for next-generation space missions,"

[4] D. S. Board, *The Role of Autonomy in DoD Systems*. https://fas.org/irp/agency/dod/dsb/autonomy.pdf.

[5] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, no. 2, pp. 83–97, 1955.

[6] M. A. Patterson and A. V. Rao, "Gpops-ii: A matlab software for solving multiple-phase optimal control problems using hp-adaptive gaussian quadrature collocation methods and sparse nonlinear programming," *ACM Trans. Math. Softw.*, vol. 41, Oct. 2014.

[7] S. Neema, *Assured Autonomy*. https://www.darpa.mil/program/assured-autonomy.

[8] V. Braitenberg, *Vehicles: Experiments in Synthetic Psychology*. Cambridge, MA: MIT Press, 1984.

[9] K. T. A. et al, *Spacecraft Formation Flying*. Dover Publishing Inc, 2010.

[10] J. J. Sellers, *Understanding Space, An Introduction to Astronautics, Third Edition*. McGraw Hill, 2005.

[11] K. Ogata, *Modern Control Engineering*. Pearson, 2010.

[12] J.-J. Slotine, *Applied Nonlinear Control*. Prentice Hall, 1991.

[13] A. Soni and H. Hu, "Formation control for a fleet of autonomous ground vehicles: A survey," *MDPI Robotics*, vol. 7, no. 67, pp. 1–25, 2018.

[14] W. Ren and R. Beard, "Decentralized scheme for spacecraft formation flying via the virtual structure approach," *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 1, pp. 73–82, 2004.

[15] M. Egerstedt and X. Hu, "Formation constrained multi-agent control," *IEEE Transaction on Robotics and Automation*, vol. 17, no. 6, pp. 947–951, 2001.

[16] L. Palacios, M. Ceriotti, and G. Radice, "Autonomous distributed lqr/apf control algorithms for cubesat swarms manoeuvring in eccentric orbits," 2013. ISSN 1995-6258.

[17] P. Ogren *et al.*, "Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment," *IEEE Transactions on Automatic Control*, vol. 49, no. 8, pp. 1292–1301, 2004.

[18] D. Morgan, G. P. Subramanian, S.-J. Chung, and F. Y. Hadaegh, "Swarm assignment and trajectory optimization using variable-swarm, distributed auction assignment and sequential convex programming," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1261–1285, 2016.

[19] L. R. Ford and D. R. Fulkerson, "Maximal flow through a network," *Canadian Journal of Mathematics*, vol. 8, p. 399–404, 1956.

[20] J. Edmonds and R. M. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems," *J. ACM*, vol. 19, p. 248–264, Apr. 1972.

[21] D. Bertsekas, "The auction algorithm for the transportation problem," *Annals of Operations Research*, no. 20, pp. 67–96, 1989.

[22] J. F. Nash, "Equilibrium points in n-person games," *Proceedings of the National Academy of Sciences*, vol. 36, no. 1, pp. 48–49, 1950.

[23] X. Jin, *Parallel Auction Algorithm for Linear Assignment Problem.* `https://stanford.edu/~rezab/classes/cme323/S16/projects_reports/jin.pdf`.

[24] D. E. Kirk, *Optimal Control Theory.* Dover Publishing Inc, 2004.

[25] B. Friedland, *Control System Design: An Introduction to State Space Methods.* Dover Publications, Inc, 1986.

[26] J. B. Burl, *Linear Optimal Control,* H_2andH$_\infty$Methods. Addison - WesleyPublishing, 1999.

[27] P. Tsiotras and M. Mesbahi, "Toward an algorithmic control theory," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 2, pp. 194–196, 2017.

[28] R. Bellman and S. E. Dreyfus, *Applied Dynamic Programming.* Princeton University Press, 1962.

[29] P. Lu, "Introducing computational guidance and control," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 2, pp. 193–193, 2017.

[30] T. Wahl, "Autonomous guidance strategy for spacecraft formations," vol. 40, no. 2, pp. 193–193, 2017.

[31] J. V.-L. R. Z. I. Costantinos Zagaris, Hyeongjun Park and M. Romano, "Model predictive control of spacecraft relative motion with convexified keep-out-zone constraints," vol. 41, no. 9, pp. 2051–2059, 2018.

[32] M. Brand, V. Shilpiekandula, C. Yao, S. A. Bortoff, T. Nishiyama, S. Yoshikawa, and T. Iwasaki, "A parallel quadratic programming algorithm for model predictive control," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 1031 – 1039, 2011. 18th IFAC World Congress.

[33] D. A. Vallado, *Fundamentals of Astrodynamics and Applications, Fourth Ed.* Space Technology Library, 2013.

Vita

2Lt Devin Saunders grew up in the city of Alexandria, VA outside of Washington D.C. He was born into a military family; both of his parents, and his grand-fathers served in the U.S. military. After graduating high school he attended the United States Air Force Academy and studied Astronautical Engineering. While at the Air Force Academy Devin was also an instructor pilot, and participated in several airmanship programs. After conducting research at the Massachusetts Institute of Technology's Lincoln Laboratory over the summer between his junior and senior years, Devin made his decision to pursue a career as an astronautical engineer. Devin received the Advanced Academic Degree scholarship allowing him to begin his Master's immediately after graduation. After graduating from the Air Force Academy in 2019, Devin began work at the Air Force Institute of Technology in the Center for Space Research and Assurance. Upon completing his program he will begin work at the Space and Missile Systems Center in Los Angeles, CA.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
13. SUPPLEMENTARY NOTES
14. ABSTRACT
15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER *(Include area code)*