APPLICATION OF MODEL PREDICTIVE CONTROL FOR THE AUTONOMOUS RENDEZVOUS AND DOCKING OF SMALL SATELLITES

A Dissertation
Presented to
The Academic Faculty

By

Andrew Fear

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Engineering
Department of Aerospace

Georgia Institute of Technology

May 2023

APPLICATION OF MODEL PREDICTIVE CONTROL FOR THE AUTONOMOUS RENDEZVOUS AND DOCKING OF SMALL SATELLITES

Thesis committee:

Dr. E. Glenn Lightsey
Department of Aerospace Engineering
Georgia Institute of Technology

Dr. Brian Gunter
Department of Aerospace Engineering
Georgia Institute of Technology

Dr. Koki Ho
Department of Aerospace Engineering
Georgia Institute of Technology

Dr. John Christian
Department of Aerospace Engineering
Georgia Institute of Technology

Dr. Srinivas Bettadpur
Department of Aerospace Engineering
The University of Texas at Austin

Date approved: April 13, 2023

For my family and friends who have always provided much needed support.

ACKNOWLEDGMENTS

I would like to acknowledge my thesis advisor Dr. E. Glenn Lightsey for all of the support he has given me over the last decade of my education and his never-waning faith in my ability. I would also like to thank my committee for their time and patience with scheduling and my work.

I would also like to give thanks to all of my family and friends without whose support I would not have been able to continue at times. I thank my parents for their patience and wishes for me to succeed. A big thanks to Byron Davis, Kip Hart, Sterling Peet, and Shivani Patel for keeping me company throughout the many long years. I would also like to acknowledge appreciation to Terry Stevenson and his guidance and sacrificed time to help me through my struggles.

I want to recognize my mental support group Teresa Silva, Dr. Danae Evans, and Dr. Allison Nitsche for their counseling and therapy that never let me lose sight of my goal.

Finally, I would like to acknowledge myself and overcoming my struggles with depression. I am not alone, and this work is proof of my accomplishments.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

6DOF Six Degree-of-Freedom

AR&D Autonomous Rendezvous and Docking

CoM Center of Mass

CW Clohessy-Wiltshire

ECI Earth-Centered Inertial

GM Gauss-Markov

LEO Low-Earth Orbit

LoS Line-of-Sight

LP Linear Program

LQR Linear Quadratic Regulator

LTI Linear Time-Invariant

LTV Linear Time-Varying

LVLH Local-Vertical Local-Horizontal

MIB Minimum impulse bit

MILP Mixed-Integer Linear Program

MPC Model Predictive Control

NLP Nonlinear Program

QP Quadratic Program

SDE Stochastic Differential Equation

YA Yamanaka-Ankersen

SUMMARY

Autonomous rendezvous and docking (AR&D) maneuvers are a key enabling technology for many types of space missions. For example, in the realm of small satellites it would facilitate on-orbit construction of larger assemblies. The volume and mass limit constraints are a crucial challenge imposed by the form factor. First, the presented work details a three-phase model predictive control (MPC) algorithm. MPC provides robustness to uncertainties in the dynamics and utilizes optimal control techniques that can handle state and control constraints directly. The three phases highlight changing constraint conditions within the underlying optimal control problem. Second, the work provides a detailed analysis of the implemented algorithm to changing parameters and tests the overall robustness to actuation uncertainties. A simulation specific to the AR&D of small satellites was created to assess the MPC algorithm. Rendezvous to a non-maneuvering target has been considered for both non-rotating and constant tumbling cases. Finally, hardware emulation is used to verify that the proposed guidance algorithm is capable of running onboard a flight computer analog. The computational performance is benchmarked for a couple of parameters to investigate the effect on performance.

CHAPTER 1
INTRODUCTION

Autonomous Rendezvous and Docking (AR&D) has been identified by NASA as a key enabling technology as far back as the 1960's [1, 2, 3, 4]. Rendezvous maneuvers are crucial for missions involving spacecraft repair, re-supply and crew exchanges, retrieval of objects, and on-orbit assembly of larger structures [3]. The need for autonomy in these maneuvers is a result of increasing mission frequencies, and a desire to improve robustness [5]. An increase in mission frequencies puts strain on ground communication systems as more spacecraft need ground access, increasing the likelihood of scheduling conflicts for spacecraft that need ground-in-the-loop guidance, as well as labor costs [5]. Spacecraft that can perform the necessary maneuvers without needing ground contact can avoid these issues. System robustness is crucial in guidance algorithms due to model uncertainty, perturbations, measurement noise, and actuator faults [6].

The work presented here considers the use of AR&D for small satellites. Small satellite AR&D enables missions like the on-orbit assembly of larger constructs, such as a large telescope [7]. On-orbit assembly lowers costs associated with launch and repair. For instance, small satellites can be launched as rideshares, which is cheaper than being the primary payload. Repairing a large spacecraft is costly, if at all possible. If a piece of a small satellite assembled structure has malfunctioned, a replacement for the single satellite housing the malfunctioning part can be launched. The malfunctioning part would undocked from the assembly and the replacement part performs the AR&D maneuvers to reform the assembly. However, the benefit of the size of small satellites does not come without its challenges. Limited commercial thruster options for guidance are available. Additionally, small satellites are a typically volume-limited form factor as is the case with the well-known CubeSat class satellites. This restricts the total available fuel volume to perform the necessary

AR&D maneuvers.

The types of challenges due to form factor are not an unknown aspect of spacecraft design. An interesting parallel can be drawn to the history of rendezvous with NASA's space shuttle program [8]. New rendezvous techniques and profiles had to be made to accommodate for the differences in thrust capability and limited processing power. However, processing power has changed dramatically since the early days of the shuttle program and a much different approach is taken in this paper. Whereas historically rendezvous and proximity operations have always had an element of human-in-the-loop interaction, this research removes that element in favor of autonomy as described above.

The AR&D process consists of a controlled trajectory that brings an active spacecraft (the chaser) into the vicinity of (and eventually contact with) a non-maneuvering object (the target) while handling the model uncertainties, environmental and actuation disturbances, and measurement noise. During the approach trajectory, the chaser must adhere to constraints on position, velocity, angular rates, and attitude for safe and successful docking. The challenge of an AR&D guidance algorithm is to then meet these needs while remaining solvable in real-time on a small flight computer typical of small satellites.

Guidance algorithms and techniques being studied tend to fall within two schools: stability based and optimal control based [6]. Stability based controllers handle uncertainties very well, but may have additional parameters that require expert tuning. An example of stability control is sliding-mode control [9, 10]. Optimal control techniques being studied include artificial potential functions (APF) [11], differential dynamic programming (DDP) [12], and model predictive control (MPC) [13, 14, 15, 16]. Of particular interest is guidance using MPC algorithms, because they have a strong capability to handle constraints [6, 13].

In an MPC approach, an optimal control problem is constructed and an optimized control sequence is found. A portion of the optimal control sequence is then applied until the optimal control problem is solved again with new state estimate information. The underlying optimal control problems may be or linear [14, 15] or nonlinear [17]. With nonlinear programming (NLP) comes increased complexity and specialized solvers [13]. To reduce computational burden, linear system (time-varying or time invariant) are often employed [15]. However, these approximations are valid locally and may not be accurate if large nonlinearities are present. The MPC formulation may also be fixed or variable horizon. In fixed-horizon MPC, the prediction model looks ahead a specified number of timesteps at every sampling time, whereas variable-horizon MPC the final time is encoded into the objective to enforce finite-time completion [15]. The approaches in [18] and [19] use variable horizon MPC and leads to solving a Mixed-Integer Linear Program (MILP) at every time instance which are hard to solve. Briefly mentioned in [15] is employing variable horizon by solving a sequence of fixed-horizon problems and taking the lowest cost option. A variable horizon guidance law that is easier to solve than an MILP formulation has been proposed in [20].

In [15], a tutorial of using MPC with Linear Time-Invariant (LTI) and Linear Time-Varying (LTV) models is presented. Different techniques to the MPC objective cost functions such as quadratic and linear cost functions are also shown. When a quadratic cost is used with a linear model, the optimal control problem becomes a Quadratic Program (QP) problem. Similarly, if a linear cost is used the problem becomes a Linear Program (LP). The tutorial also shows how to design a variable horizon formulations through terminal constraints and the objective cost. Constraints in the optimal control problem such as maximum control and approach direction through a Line-of-Sight (LoS) corridor are discussed. The presented method is general AR&D and is applicable to a cooperating non-rotating spacecraft. A similar fixed-horizon MPC solution that incorporates a ”keep-out” zone for avoiding collisions is presented in [14]. This method focuses on the QP cost for its relation to Linear Quadratic Regulator (LQR) control and the classical Clohessy-Wiltshire (CW) equations are used for an LTI prediction model. The ”keep-out” zone is shown in the context of avoiding an obstacle during the AR&D maneuver. Weiss utilizes a two-phase guidance that switches MPC formulations in a prescribed box-distance around the target spacecraft. Additionally, the farther range formulation is a reference-governor approach where a forced equilibrium set point is included in the optimized variables. This set point is supposed to improve feasibility of shorter control horizons. However, the values chosen for the examples are on a scale much more applicable to large spacecraft propulsion systems and do not show viability in context of low-thrust spacecraft. The QP MPC problem is expanded on in [16] to allow for docking with a tumbling spacecraft. An LTI model for the dynamics (through the CW equations) is used for real-time computation. Within the prediction model, the constraints for the LoS corridor are updated to approximate the tumbling target motion. There is a focus on the closing section of the maneuver where the controlled spacecraft is already within approximately 20 m distance to the target and therefore only has one phase. Instead of using the CW model, an LTV model is used in [21] for an LTV formulation. This is important for highly elliptical orbits where the CW equations start to break down in accuracy.

The research presented is valuable because it takes the general rendezvous MPC approaches and assesses its use for small satellite use. Ideas from the previous works, namely a phased maneuver with QP focus with a tumbling target, are combined with modifications to fit the investigated small satellite scenario. The first contribution consists of combining these works into a working guidance algorithm as well as setting up the testing simulation. The following contributions deal with verifying and validating the developed algorithm for small satellite use by checking the robustness and performance. To that end, the guidance is tested through simulation to check how the performance changes with the MPC parameters in each phase. A working algorithm is not enough to ensure that an algorithm is ready for flight on a small satellite. The final contribution aims at answering the question of flight readiness through hardware emulation.

The constraints in the MPC formulations to be shown may be equivalent to previous works in form and function, but are unique to small satellites in regard to permissible values and implementation. The values for maximum impulsive Δv in application of the algorithm are on the order of $<$1 N as is typical of small satellite propulsion systems [22]. The thrusters are also not able to be throttled, but produce a fixed amount of thrust when the valves are opened, which is not true for all propulsion systems. In terms of computational efficiency, small satellite flight computers are typically single or dual-core ARM processors with $<$1 GHz clock speed. The developed algorithm is designed with this limitation in mind.

To make it small satellite viable, a three-phase guidance algorithm that focuses on computationally efficient control solutions is developed. The three-phase guidance approaches the problem from a wider perspective than just the closing phase where the chaser and target spacecraft are tens of meters apart. Three phases are considered instead of two to relax the problem when the chaser spacecraft is approaching the target from the intended direction but the constraints for docking may be too strict. The first phase guides the chaser to an appropriate approach direction. In the second and third phases, the chaser spacecraft closes in on the target while remaining inside an allowable area. As docking interfaces may have strict angle-of-approach and velocity requirements it is necessary to relax these constraints as the chaser closes in during the added second phase, as mentioned previously. Instead of a forced equilibrium set point in the rendezvous phase, a fixed set point is used to guide the chaser towards the approach corridor instead of directly to the origin which conflicts with the safety ”keep out zone” constraint. To remain computationally simple for small satellite flight computers, the three phases use fixed-horizon QP for their optimal control problems and the predictive model remains LTI. The developed algorithm is tested for three different cases regarding the approach direction of the target: in the transverse (so-called V-bar) direction, the radial (dubbed R-bar) and for a constantly rotating target.

For the viability contribution, the domain is investigated by changing the initial conditions of the chaser spacecraft. The purpose of this is to check if there are areas of relative positions that tend to produce infeasible guidance solutions and should be avoided by mission planners. Random initial relative conditions were selected to measure the resulting Δv and time to rendezvous characteristics of the guidance law in relation to their starting angle and distance from the target.

Next, disturbances from the environment and noises are added to the simulation to verify the robustness. The noises added are what can be considerable levels in the measurements and propulsion system actuation. The noisy actuation measurements are modelled as a time-varying system such that it also captures unknown modeling errors. The simulations are run for the randomly chosen initial chaser starting conditions for numerous simulations to check the average effect of the noises and disturbances on the algorithm characteristics, namely Δv and time to rendezvous.

The developed MPC laws are constrained by hardware limitations in propulsion systems and flight computers that are common to the type of small satellites being investigated. The presented work evaluates the performance of the phases for use on typical hardware. This is done through the use of hardware emulation to remain mission design agnostic, although the type of processor chosen resembles those found on commercial small satellite flight computers. The simulation is split into a server-client setup, where the dynamics are controlled by the server portion running on a desktop computer. The flight software with the MPC resides is located on the emulated hardware and profiling is used to check the computation times. This analysis intends to decide if the proposed algorithm is computationally simple enough to run quickly on this type of hardware.

The first contribution to create an MPC guidance algorithm that meets the constraints set by an example small satellite scenario is described in Chapter 3. The domain check and MPC parameters sensitivity contribution are the subject of chapter 4. To validate the robustness of the algorithms, Chapter 5 presents the discussion on disturbances and noise. Finally, Chapter 6 focuses on benchmarking the performance of an implementation of the algorithm to assess its ability to run in a flight-like environment.

CHAPTER 2

METHODOLOGY AND TOOLS

This chapter introduces the models and tools used for performing AR&D operations. First, the linearized equations of relative motion, perturbations, and thruster actuation models are introduced. Second, an introduction to model predictive control is presented. Finally, the simulation and optimization tools are briefly discussed.

2.1 Spacecraft Models

This section presents the assumed models used to create the guidance algorithm. First, the equations of relative motion needed for the MPC prediction step are shown along with perturbation models. Then, the thruster models used in the guidance testing simulations are described.

2.1.1 Equations of Relative Motion

The reference frame used for this work is based on a Local-Vertical Local-Horizontal (LVLH) frame centered at the target spacecraft's Center of Mass (CoM). The $\hat{x}$-axis is directed radially outwards from Earth, with the $\hat{y}$-axis in the velocity direction. The $\hat{z}$-axis completes the right hand rule being in the direction of the target orbit angular velocity vector.

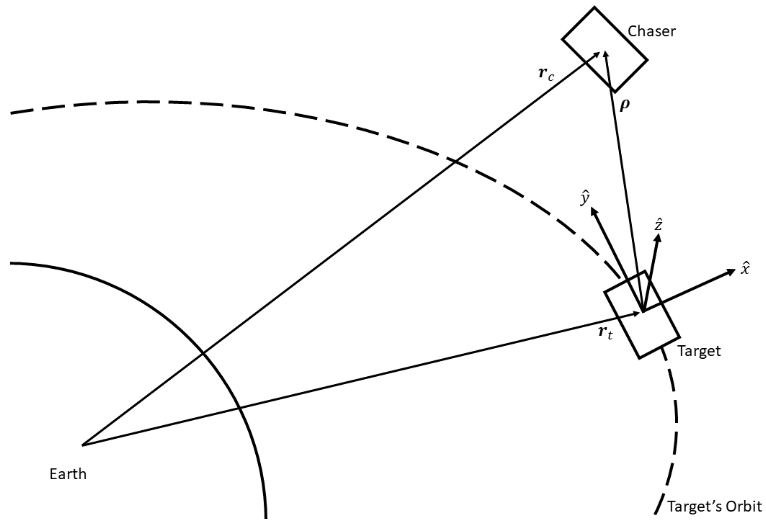


Figure 2.1: LVLH Reference frame of the target

The classic Clohessy-Wiltshire (CW) equations [23] describe the linearized relative motion of two spacecraft in orbit about the Earth in the given reference frame. As stated previously, the choice of these equations is to make the prediction model an LTI system. An LTI system will allow the MPC optimal control problems to be set up as QP or LP. This is important for small satellite viability as NLP solvers are not yet as developed in terms of computational speed [13]. However, linearization has the drawback that the prediction model is accurate only locally, and in the case of the CW equations this means near circular orbits ($e = 0$). Therefore, it is assumed that the target is in a circular orbit about the Earth, which is not unreasonable for the Low-Earth Orbit (LEO) scenario considered in this paper. This assumption would not work if the target spacecraft was in a geostationary transfer orbit

(GTO) or a near-halo rectilinear orbit (NRHO). The CW equations are given by

$$\dot{\boldsymbol{x}}\left(t\right) = \tilde{\boldsymbol{A}}\boldsymbol{x}\left(t\right) + \tilde{\boldsymbol{B}}\boldsymbol{u}\left(t\right) \tag{2.1}$$

$$\tilde{\boldsymbol{A}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 3\omega^2 & 0 & 0 & 0 & 2\omega & 0 \\ 0 & 0 & 0 & -2\omega & 0 & 0 \\ 0 & 0 & -\omega^2 & 0 & 0 & 0 \end{bmatrix}, \quad \tilde{\boldsymbol{B}} = \begin{bmatrix} \boldsymbol{0}_{3\times3} \\ \boldsymbol{I}_{3\times3} \end{bmatrix} \tag{2.2}$$

where ω is the mean motion of the target's orbit in the Earth-Centered Inertial (ECI) frame. The state vector $\boldsymbol{x}(t) \in \mathbb{R}^6$ is defined as $\boldsymbol{x}(k) = [\boldsymbol{\rho}(t) \quad \dot{\boldsymbol{\rho}}(t)]$, where $\boldsymbol{\rho}(k) \in \mathbb{R}^3$ and $\dot{\boldsymbol{\rho}}(t) \in \mathbb{R}^3$ are defined as the components of the relative position and velocity vectors in the LVLH frame, respectively, at time $t \in \mathbb{R}$. The vector $\boldsymbol{u}(t) \in \mathbb{R}^3$ defines the applied control to the chaser spacecraft.

The continuous-time model of the CW equations is discretized to give the following Linear Time-Invariant (LTI) system

$$\boldsymbol{x}\left(k+1\right) = \boldsymbol{A}\boldsymbol{x}\left(k\right) + \boldsymbol{B}\boldsymbol{u}\left(k\right) \tag{2.3}$$

where time index $k \in \mathbb{N}_0$ defines the time $t_k \in \mathbb{R}^+$. From this point on, a subscript k is used as shorthand to describe a vector at time instant k. It is imperative that the impulsive invariant discretization method is used so that the control is modeled as impulsive velocity changes, $\boldsymbol{u}_k = \Delta\boldsymbol{v}_k$. A typical zero-order hold (ZOH) discretization method may work well for small sample periods, the impulsive assumption is more appropriate for the longer maneuvers [15]. The matrix, $\boldsymbol{A}$, is the state transition matrix over the discrete sample time $\Delta T_s = t_{k+1} - t_k$, and the control mapping $\boldsymbol{B} = \boldsymbol{A}\bar{\boldsymbol{B}}$, so that the state is propagated through an impulsive velocity change over the sample time ΔT_s.

While the CW equations are only applicable to circular target orbits, the so-called Tschauner-Hempel equations and various solutions [24, 25, 26] are used for the linearized relative motion in elliptical orbits. The Yamanaka-Ankersen (YA) state-transition matrix [25] can be used as a LTV system of equations for elliptical orbits in place of the CW equations. It should also be noted that the YA system reduced exactly to the CW equations for $e = 0$.

2.1.2 Thruster Force Models

Small satellite propulsion systems may use cold gas, monopropellants, or electric propulsion [22]. The discretization method used for the MPC model will be shown to be impulsive so while electric propulsion has high I_sp values for the propellant, the thrust provided is on the order of μN and is typically too small for an impulsive approximation. Typical values provided by a cold gas system range from 1-25 mN [22] a monopropellant system provides between 0.25-1 N [22, 27]. While not perfect, these ranges are a much better fit for an impulsive assumption. The type of small satellite propulsion considered in this study work on the basis of ”on/off” control rather than throttleable. The amount of force applied is either all or nothing, and cannot be throttled between zero and the maximum value. When commanded on, a thruster will apply a nominal force, F_{thr} for the duration of the on-time (or pulse-width), t_{pulse}. The amount of impulse to a system is changed by adjusting the amount of time that the thruster is on.

$$J_{cmd} = F_{thr} * t_{pulse} \tag{2.4}$$

As stated previously, the discretization method uses impulsive Δv assumptions and is not realistic for this type of actuation due to the finite pulse-width. Additionally, the thruster valves do not open and close instantaneously so the nominal thrust instead has some ramp-up and down time. As a result, the realized impulse may not be exactly the same as the idealized impulse. These timings are able to be well-defined by experimentation and can be calibrated out as a constant low bias and have therefore been ignored. Also related is a limit to how short the pulse may be, leading to the Minimum impulse bit (MIB) that the thruster can provide. A constraint included in the MPC formulation creates an upper bound on the thruster on-time to achieve as close to an impulsive-like behavior as desired.

For simplicity, a constant mass for the spacecraft is assumed. This assumption provides a conservative estimate needed for overall Δv as realistically the mass of the spacecraft will decrease with expelled propellant. As an example, assuming a 14 kg CubeSat using the AF-M315E green monopropellant (recently renamed ASCENT) with an I_{sp} of 220 s, the needed propellant mass to provide 80 m/s is around 0.5 kg, which is approximately 3.5% of the spacecraft mass.

With the constant mass assumption, thrust applied is $\boldsymbol{F}_k \Delta t = m\Delta \boldsymbol{v}_k$. The amount of force that the ith thruster with axis $\boldsymbol{n}_{thr,i}$ needs to provide is simply the amount of force in the thruster direction, $F_{k,i} = \boldsymbol{F}_k \cdot \boldsymbol{n}_{thr,i}$. Then, the on-time needed for the ith thruster is found with

$$t_{pulse,i} = \frac{F_{k,i}}{F_{max,i}} \tag{2.5}$$

Then, the control applied is related to the provided impulse and chaser spacecraft mass, m_c through

$$u = \begin{cases} 0, & t_{pulse} < t_{MIB} \\ J_{cmd}/m_c, & t_{pulse} \geq t_{MIB} \end{cases} \tag{2.6}$$

where t_{MIB} is the pulse width corresponding to the MIB. If the needed pulse width is smaller than the MIB the control is simply discarded as too small to provide. Another option is to calculate the pulse width on an opposing thruster such that the resultant thrust provided is the desired amount, but this would cost more propellant so was not implemented.

2.2 Model Predictive Control

Optimal control theory is the basis for MPC. A dynamics model predicts future states based on current state information, and a control input at the current time optimizes the predicted states [28, 29]. The general optimization problem to be solved is given by:

$$
\begin{aligned}
\min_{\boldsymbol{u}(t)} J(\boldsymbol{x}(t), \boldsymbol{u}(t)) &= \int_0^\infty \ell(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau))\, d\tau \\
\text{s.t.} \quad \dot{\boldsymbol{x}}(t) &= f(\boldsymbol{x}(t), \boldsymbol{u}(t)) \\
\boldsymbol{x}(0) &= \boldsymbol{x}_0 \\
(\boldsymbol{x}(t), \boldsymbol{u}(t)) &\in \mathbb{Z} \qquad \forall t \in \mathbb{R}_{\geq 0}
\end{aligned}
\tag{2.7}
$$

where $\mathbb{Z} \subseteq \mathbb{X} \times \mathbb{U}$, $\mathbb{X} \subseteq \mathbb{R}^6$, and $\mathbb{U} \subseteq \mathbb{R}^3$. However, solving this minimization problem is still intractable for most problems, as it is an infinite horizon problem. To make the minimization more computationally feasible, the cost has been changed to a finite-horizon with an additional cost for the subsequent infinite interval.

$$
\begin{aligned}
\min_{\boldsymbol{u}_k} J(\boldsymbol{x}_k, \boldsymbol{u}_k) &= \sum_{i=0}^{N} \ell(\boldsymbol{x}_{i|k}, \boldsymbol{u}_{i|k}) + J_N(\boldsymbol{x}_{N|k}, \boldsymbol{u}_{N|k}) \\
\text{s.t.} \qquad \boldsymbol{x}_{i+1|k} &= f(\boldsymbol{x}_{i|k}, \boldsymbol{u}_{i|k}) \\
\boldsymbol{x}_{0|k} &= \boldsymbol{x}(t_k) \\
\left(\boldsymbol{x}_{i|k}, \boldsymbol{u}_{i|k}\right) &\in \mathbb{Z} \qquad\qquad i = 0...N-1 \\
\boldsymbol{x}_{N|k} &\in \mathbb{X}_f
\end{aligned}
\tag{2.8}
$$

It should be noted that the state, control, and constraints are discretized. In general, the solution to Equation 2.8 is an open-loop control sequence, $\boldsymbol{u}_k = \left\{\boldsymbol{u}_{0|k}, \boldsymbol{u}_{1|k}, ..., \boldsymbol{u}_{N|k}\right\}$, and not a feedback control law. The resulting predicted state and open-loop optimal control sequence are for a finite number of time steps, t_k to t_{k+N}, or the prediction horizon as seen demonstrated in Figure 2.2. The MPC feedback law is constructed by solving Equation 2.8

online every time step and applying the first element of the control sequence, $\boldsymbol{u}_{0|k}$. At the next sample time, the prediction horizon will now be over a new time interval. This is what makes this type of finite MPC as a moving horizon problem.

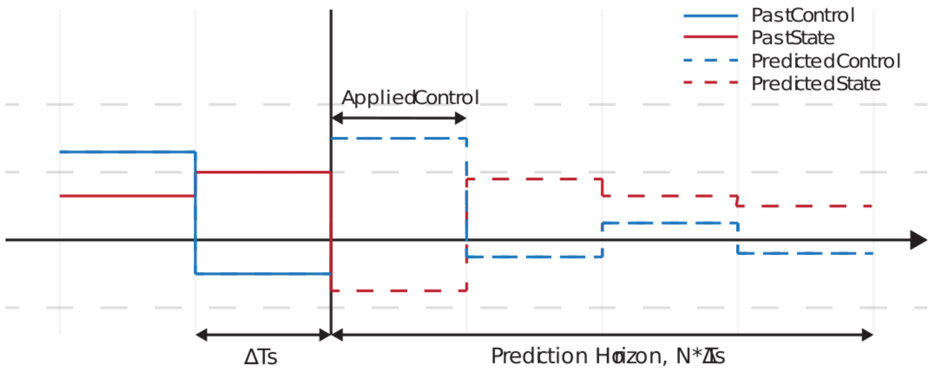


Figure 2.2: MPC state and control example horizon

2.2.1 Cost Objectives

There exists a variety of possible running cost functions for optimal trajectories depending on the desired result. The classic quadratic cost for the stage and terminal cost are given by

$$\ell(\boldsymbol{x}_i, \boldsymbol{u}_i) = \boldsymbol{x}_i^T Q \boldsymbol{x}_i + \boldsymbol{u}_i^T R \boldsymbol{u}_i \tag{2.9a}$$

$$J_N(\boldsymbol{x}_N, \boldsymbol{u}_N) = \boldsymbol{x}_N^T Q_f \boldsymbol{x}_N \tag{2.9b}$$

From this point, the shorthand $\|\boldsymbol{x}\|_A^2 := \boldsymbol{x}^T A \boldsymbol{x}$ is used to save space. With appropriate weighting, the quadratic cost function can produce smooth trajectories with intrinsic robustness [15, 13]. Another option is the 1-norm cost function:

$$\ell(\boldsymbol{x}_i, \boldsymbol{u}_i) = \|Q\boldsymbol{x}_i\|_1 + \|R\boldsymbol{u}_i\|_1 \tag{2.10a}$$

$$J_N(\boldsymbol{x}_N, \boldsymbol{u}_N) = \|Q_f \boldsymbol{x}_N\|_1 \tag{2.10b}$$

The 1-norm cost function tends toward sparser ”bang-off/bang-on” style control, which can reduce fuel consumption compared to quadratic cost. However, it may also be less robust to uncertainties and disturbances [15].

For variable horizon MPC the stage cost $\ell(\boldsymbol{x}_i, \boldsymbol{u}_i)$ must contain a constant term which will be summed over to represent a penalty on the number of time steps taken [15]. The error in the state is also removed from the stage cost and there must be a terminal state constraint.

$$\ell(\boldsymbol{x}_i, \boldsymbol{u}_i) = 1 + \|R\boldsymbol{u}_i\|_1 \qquad \text{(1-norm)} \qquad (2.11a)$$

$$\ell(\boldsymbol{x}_i, \boldsymbol{u}_i) = 1 + \|R\boldsymbol{u}_i\|_1 \qquad \text{(quadratic)} \qquad (2.11b)$$

An alternative way to implement a variable horizon is to solve a sequence of fixed-horizon problems up to a horizon of N_{max} and take the action with the minimum weighted objective. Since the presented research is interested in minimizing computational effort, the variable horizon methods were not investigated.

2.2.2 Quadratic Program MPC

The stability and controllability of MPC is discussed in [28] and [29]. Appropriate weighting for the terminal cost term, J_N, is globally known for special cases to provide stability. In particular, it is known for the quadratic cost case which, as stated previously, is similar to the well-known LQR feedback control law [15, 28] If the state prediction is assumed to be linear $f\left(\boldsymbol{x}_{i|k}, \boldsymbol{u}_{i|k}\right) = A\boldsymbol{x}_{i|k} + B\boldsymbol{u}_{i|k}$, $Q \geq 0$, and $R > 0$, then Q_f can be shown to be the solution to the algebraic Riccati equation [28, 29]. The design choices of Q and R are usually tricky, a trait from the underlying LQR that is inherited by the MPC problem. Taking the cost to be quadratic also has the benefit that Equation 2.8 is computationally feasible to be solved online every sample instance. Using the quadratic cost, the full cost for our control minimization problem is

$$J(\boldsymbol{x}_k, \boldsymbol{u}_k) = \sum_{i=0}^{N} \left(\left\| \boldsymbol{x}_{i|k} \right\|_Q^2 + \left\| \boldsymbol{u}_{i|k} \right\|_R^2 \right) + \left\| \boldsymbol{x}_{N|k} \right\|_{Q_f}^2 \tag{2.12}$$

At each time step, Equation 2.12 is minimized to find the optimal control sequence, $\boldsymbol{u}_k = \left\{ \boldsymbol{u}_{0|k}, \boldsymbol{u}_{1|k}, ...\boldsymbol{u}_{N|k} \right\}$, subject to constraints on the state and control, which are now discussed. Since the weighting matrices Q and R are positive-definite, it follows that the cost is convex. Therefore, if the constraints are also chosen to be convex, the minimization of Equation 2.12 will be a convex QP which is relatively easy to solve with modern tools. These constraints may be given as sets of equality or inequality constraints on the state and control variables.

$$\boldsymbol{g}\left(\boldsymbol{x}_i, \boldsymbol{u}_i\right) = 0 \tag{2.13a}$$

$$\boldsymbol{h}\left(\boldsymbol{x}_i, \boldsymbol{u}_i\right) \leq 0 \tag{2.13b}$$

With the assumption of the quadratic cost, linear state update, and convex constraints, the full convex QP minimization problem for the optimal control sequence at time t_k is

$$\begin{aligned} \min_{\boldsymbol{u}_k} J(\boldsymbol{x}_k, \boldsymbol{u}_k) = & \sum_{i=0}^{N} \left(\left\| \boldsymbol{x}_{i|k} \right\|_Q^2 + \left\| \boldsymbol{u}_{i|k} \right\|_R^2 \right) + \left\| \boldsymbol{x}_{N|k} \right\|_{Q_f}^2 \\ \text{s.t.} \quad & \boldsymbol{x}_{i+1|k} = A\boldsymbol{x}_{i|k} + B\boldsymbol{u}_{i|k} \\ & \boldsymbol{x}_{0|k} = \boldsymbol{x}(t_k) \\ & \boldsymbol{g}(\boldsymbol{x}_{i|k}, \boldsymbol{u}_{i|k}) = \boldsymbol{0} \qquad i = 0, ..., N \\ & \boldsymbol{h}(\boldsymbol{x}_{i|k}, \boldsymbol{u}_{i|k}) \leq \boldsymbol{0} \end{aligned} \tag{2.14}$$

For the case of satellite rendezvous, A and B are taken from Equation 2.1, the CW equations of relative motion. The optimization in Equation 2.14 forms the basis for each control optimization setup for the phases in the AR&D problem.

As stated in 2.1.2, thrusters considered are a type of on/off actuation, which is not well described by the convex constraints in Equation 2.14. It is possible to set up the MPC problem to include discrete actuators, but this increases solving complexity as it becomes a mixed-integer QP. Increased complexity generally results in longer computation times. Considering the minimization has to complete before every sample time, the mixed-integer problem is currently not considered to conserve computation effort.

2.3 Spacecraft Dynamics and Environment Simulation

Testing scenarios for the algorithms presented in this work are run using the 42 Spacecraft Simulation tool independently developed at NASA Goddard Space Flight Center [30]. 42 is an open-source simulation framework for developing formation flying guidance and attitude control laws. Written in C, it provides spacecraft dynamics propagation, sensor models, and actuation models for a user to develop a flight software-like routine to test a control algorithm. The framework allows the user to set up the environment as desired in input files. This includes turning on and off perturbation effects such as atmospheric drag, solar radiation pressure, and Earth's oblateness. For the work presented, 42 is used to model the chaser and target spacecraft with atmospheric drag, and Earth's oblateness perturbations. Six thrusters are attached to the chaser spacecraft to give Six Degree-of-Freedom (6DOF) motion. It should be noted that 42 is just a simulation tool and not expected to run on a flight system.

42 includes a graphics output for viewing the spacecraft created for the simulation, Earth, and other celestial bodies. This viewer is useful for watching attitude control and guidance algorithms in action. Doing so does require a 3D object model that the framework can use. These models are also necessary for more advanced environmental disturbances such as atmospheric drag.

The propagation methods within 42 are based off of relative motion. Each spacecraft is given a reference orbit and a model for propagating the system relative to that orbit is used for the spacecraft motion. Relative motion models include Euler-Hill (equivalent to the CW equations), the Encke method, and the Cowell method.

2.4 Optimization Problem Solver

The minimization QP problems are solved using the CVXGEN [31] solver. CVXGEN generates custom code in either MATLAB or C for a high-speed QP solver from a convex QP problem description. Specifically, this paper will be making use of CVXGEN to solve problems of the type shown by Equation 2.14. The code generation works best for smaller problems and may fail for large systems due to the size of variables.

2.5 Hardware Emulation

QEMU is an open-source emulator and virtualizer for a variety of processor boards [32]. This tool is used to emulate an ARM processor system used to test the developed algorithm in a flight-like environment. QEMU is used as a full-system emulator running Linux on the virtual machine. While the full-system emulation is not exactly cycle-accurate, it is a baseline for testing on the intended system.

CHAPTER 3

AUTONOMOUS RENDEZVOUS USING MPC

General approaches for using MPC for rendezvous and docking operations [14, 13, 15] with a non-rotating target have been previously developed. In [16], the case of a tumbling target is considered, and only focuses on the final portion of the rendezvous where the chaser is already in the vicinity of target spacecraft. The work presented here examines the whole maneuver starting from a large distance between the chaser and target spacecraft. This research expands on the efforts in [14, 15, 16] to the whole maneuver with a focus on adapting the algorithms to one usable onboard a small satellites. To accomplish this, the developed algorithm centers on QP optimal control and using fixed-horizon methods.

In order to reduce overall computational load, the AR&D scenario was split into three phases, shown in Figure 3.1, based on the relative distance between the chaser and target spacecrafts: rendezvous, approach, and closing/docking. Each phase has a different sample time between control inputs. The sampling time is smaller for closer range phases to provide finer and faster response as the chaser spacecraft approaches the target. When far from the target, the chaser does not need to adhere to the same constraints that are needed during the docking event. For example, the chaser does not need to meet attitude or keep-in zone constraints when the distance to the target is large. Splitting the maneuver allows the control sequence optimization constraints to be adjusted or tightened to reduce unnecessary actuation early in the maneuver.

Since finite horizon MPC uses discrete time steps, the CW equations must be discretized. In [16], the discretization method used for the control mapping assumes that the applied control is constant between sampling times. For distances with long sample times, this may not be feasible. Instead, the proposed guidance algorithm uses the follow-

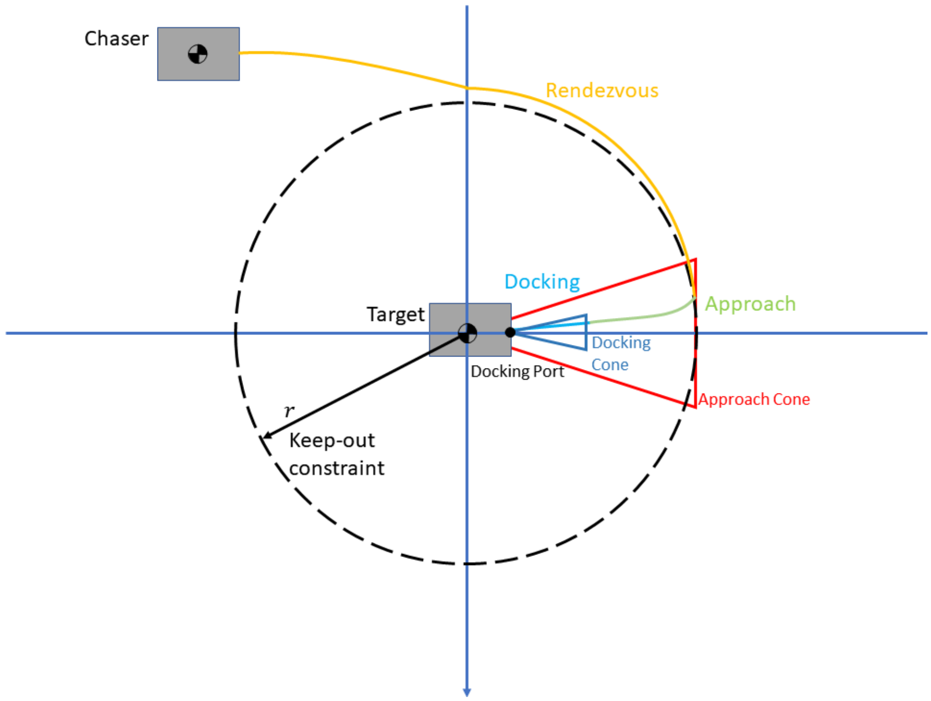


Figure 3.1: The 3 Phases of the Guidance Algorithm with the Rendezvous phase keep-out zone and keep-in cones for Approach and Docking.

ing impulsive invariant discretization

$$x_{k+1} = \bar{A}x_k + \bar{B}u_k \tag{3.1}$$

where A is the usual discretized form $\bar{A} = e^{A\Delta T}$, and the control mapping is $\bar{B} = AB$. This control mapping is such that the state is propagated through an impulsive velocity change over the sample time ΔT.

3.1 Rendezvous Phase

At relative distances greater than the keep-out zone (up to a few km) and outside of the approach cone, the algorithm is considered to be in the rendezvous phase. There is low risk of collision between the two spacecraft and the time between control updates may be upwards of a couple minutes. Lower sampling times can still be used to shorten the overall rendezvous time, but come at the cost of expending more fuel. This section describes the formulation of the constraints needed in Equation 2.14, as well as modifications to the objective cost, for the rendezvous phase MPC QP.

3.1.1 Reference Point

As the purpose of the rendezvous phase is to guide the chaser into the approach cone, a reference point $x^{(s)}$ is added to the quadratic cost terms in Equation 2.12 to drive the chaser state to this set point. A set point should be within the approach cone, preferably on or near the boundary of the keep-out zone.

$$J(\boldsymbol{x}_k, \boldsymbol{u}_k) = \sum_{i=0}^{N} \left(\left\| \boldsymbol{x}_{i|k} - \boldsymbol{x}_k^{(s)} \right\|_Q^2 + \left\| \boldsymbol{u}_{i|k} \right\|_R^2 \right) + \left\| \boldsymbol{x}_{N|k} - \boldsymbol{x}_k^{(s)} \right\|_{Q_f}^2 \tag{3.2}$$

Without the reference point, the minimization of the cost will attempt to drive the state of the chaser to the origin (ie. the CoM of the target). This was the case in previous iterations of this work [33, 34]. While this may be close the end goal, it runs counter to the purpose of the rendezvous phase and will fight against the keep-out zone constraint which is explored later in this section.

3.1.2 Terminal Penalty

Additionally, an external penalty on the final state of the chaser spacecraft may be applied to the cost as

$$J_{penalty}(\boldsymbol{x}_k, \boldsymbol{u}_k) = \lambda \mathbf{1}^{\mathbf{T}} \left(\boldsymbol{H} \boldsymbol{x}_{N|k} \right)_+ \tag{3.3}$$

where $\mathbf{1}^{\mathbf{T}}$ is defined as a vector of ones and the operator $(\cdot)_+$ denotes taking only the positive vector elements. $\lambda \in \mathbb{R}_+$ is a weighting on the penalty function. The approach cone is defined by an n-order polyhedron $\mathcal{P} = \{\boldsymbol{x} \in \mathbb{R}^3 : \boldsymbol{H}\boldsymbol{x} \leq 0\}$ where $\boldsymbol{H} \in \mathbb{R}^{n \times 3}$. For the purposes of this work, the polyhedron was taken to be of order 4 for simplicity.

Since the penalty was added onto the objective cost and not the state constraints of the QP, it is not a strict requirement. This allows for more feasible trajectories, where the chaser may not be within the approach cone within the first prediction horizon. With the addition of the reference cost, this external penalty on the terminal state is not strictly needed, but may help to drive the final state towards a feasible set point. One may choose to implement one or the other or both of these cost modifications to desired effect.

3.1.3 Convexified Keep-out Constraint

To guide the chaser safely to the rendezvous phase goal, a keep-out zone around the origin is considered. Without this constraint, the resulting trajectory may attempt to fly-by (or even through) the target spacecraft towards the rendezvous goal of the approach cone. This collision avoidance can be modeled as a sphere of keep-out radius, r_1

$$\left\|\boldsymbol{x}_{i|k}\right\|_2 \geq r_1 \tag{3.4}$$

However, the definition in Equation 3.4 is not convex and does not fit within the desired convex QP framework. To convexify this constraint, the keep-out zone is approximated as a set of inequalities that describe a rotating plane on the surface of the sphere [35], as shown in Figure 3.2.

$$\hat{\boldsymbol{n}}_{i|k}^T \boldsymbol{x}_{i|k} \geq r_1 \quad i = 0, ..., N \tag{3.5}$$

The vector $\hat{\boldsymbol{n}}_{i|k}$ defines a normal vector of the plane tangent to the sphere at point $r\hat{\boldsymbol{n}}_{i|k}$. Throughout the prediction horizon, the normal vector rotates according to a prescribed rate so that at time t_N the end vector is aligned with the approach cone center axis. This splits the rotation into N equal angles rotations given by

$$\theta_k = \arccos \frac{\boldsymbol{x}_{0|k}^T \boldsymbol{n}_{N|k}}{\left\|\boldsymbol{x}_{0|k}\right\| \left\|\boldsymbol{n}_{N|k}\right\|} / N \tag{3.6}$$

However, if the rate of the plane's rotation around the keep-out zone is too high, this

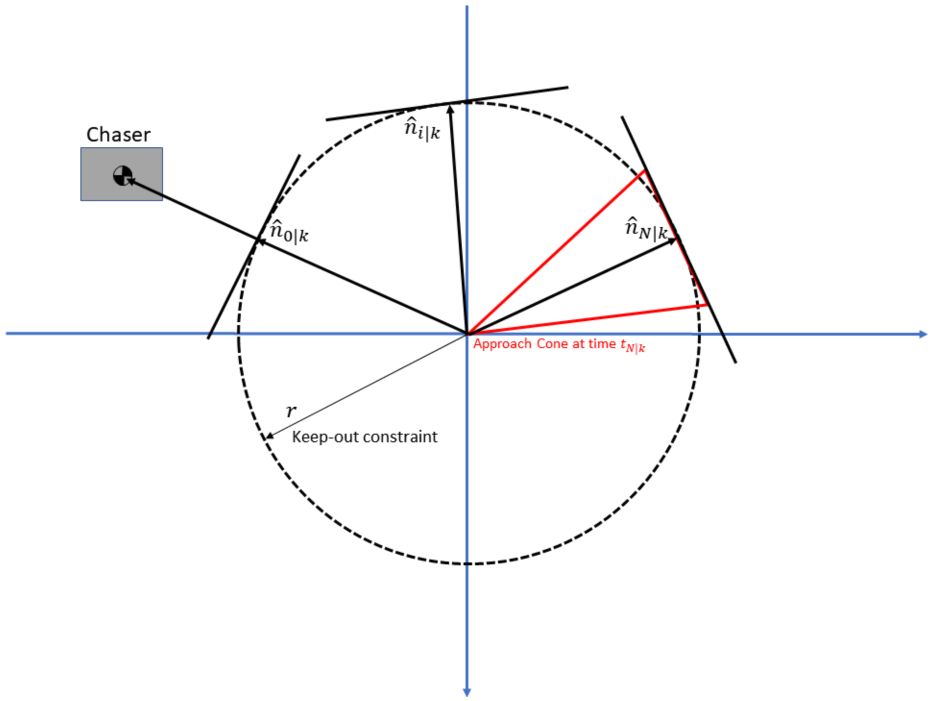


Figure 3.2: Rotating plane constraints to approximate the non-convex keep-out zone.

may lead to infeasible solutions for the chaser's trajectory [35]. To combat this effect, an upper limit is set on the rotation rate, which may result in the plane at the end of the prediction horizon not being aligned with the polyhedron axis. During subsequent prediction horizons, the keep-out planes will continue to rotate towards the desired approach cone until a horizon where the center axis is reached and can be split into equal angle rotations. It was found that the target orbit's mean motion, ω_t, was an acceptable upper limit for the purposes of this work. Slightly different than in [35] is the approach of plane rotation directions. If the approach cone center axis is within N steps during the prediction, the planes may rotate either direction. However, if outside this prediction horizon, there is a preference for the plane rotation to occur in the direction most closely aligned with the natural relative orbit motion.

3.1.4 Rendezvous QP

Using the modified cost function (Equation 3.2), terminal penalty (Equation 3.3), and the series of inequalities to approximate the keep-out zone (Equation 3.5) in (Equation 2.14) will yield the QP for the rendezvous phase MPC.

$$
\begin{aligned}
\min_{\boldsymbol{u}_k} J(\boldsymbol{x}_k, \boldsymbol{u}_k) = \sum_{i=0}^{N-1} \left(\left\| \boldsymbol{x}_{i|k} - \boldsymbol{x}_k^{(s)} \right\|_Q^2 + \left\| \boldsymbol{u}_{i|k} \right\|_R^2 \right) + \left\| \boldsymbol{x}_{N|k} - \boldsymbol{x}_k^{(s)} \right\|_{Q_f}^2 + \lambda \mathbf{1}^{\mathbf{T}} \left(\boldsymbol{H} \boldsymbol{x}_{N|k} \right)_+ \\
\text{s.t.} \quad \boldsymbol{x}_{i+1|k} = \boldsymbol{A}\boldsymbol{x}_{i|k} + \boldsymbol{B}\boldsymbol{u}_{i|k} \\
\boldsymbol{x}_{0|k} = \boldsymbol{x}(t_k) \\
\hat{\boldsymbol{n}}_{i|k}^T \boldsymbol{x}_{i|k} \geq r_1 \qquad i = 0, ..., N \\
|\boldsymbol{u}_k|_\infty \leq u_{max}
\end{aligned}
\tag{3.7}
$$

3.2 Approach Phase

The policy switches to the mid-range approach phase when the chaser spacecraft enters the designed approach cone emanating from the target vehicle's CoM and is on the boundary of the keep-out zone radius r_1. In the approach phase, the sample time, T_s, is decreased for finer control.

3.2.1 Objective Cost

The running cost of the approach phase is similar to the rendezvous phase with some key differences. The penalty of the cone constraint is now taken over the entire prediction states to remain within the approach cone. It is still an external penalty because it was decided that it is acceptable to violate this cone slightly to ensure feasible solutions. The reference point was also carried over into this phase due to how well it worked in the rendezvous phase. The reference point is set to a point within the approach cone just behind the switching distance to the docking phase.

$$J(\boldsymbol{x}_k, \boldsymbol{u}_k) = \sum_{i=0}^{N-1} \left(\left\| \boldsymbol{x}_{i|k} - \boldsymbol{x}_k^{(s)} \right\|_Q^2 + \left\| \boldsymbol{u}_{i|k} \right\|_R^2 \right) + \left\| \boldsymbol{x}_{N|k} - \boldsymbol{x}_k^{(s)} \right\|_{Q_f}^2 + \lambda \sum_{i=1}^{N} \mathbf{1}^T \left(\boldsymbol{H_i} \boldsymbol{x}_{i|k} \right)_+ \tag{3.8}$$

3.2.2 Constraints

Instead of following a keep-out zone, there is an overshoot constraint that follows the vertex of the approach cone and does not allow the chaser to enter past a distance of r_2 from the target. This acts as a safety barrier and acts as the switching point for the final phase.

$$\boldsymbol{n}_{cone,i|k}^T \boldsymbol{x}_{i|k} \geq r_2 \qquad\qquad i = 0, ..., N \tag{3.9}$$

As an example, if the target was non-rotating and the desired docking direction was along the y-axis, then this constraint would reduce to $y_{i|k} \geq r_2$, which is a hard constraint on the y-direction of the chaser's relative position.

3.2.3 Approach QP

The full minimization QP for the approach phase is given by

$$\begin{aligned}
\min_{\boldsymbol{u}_k} J(\boldsymbol{x}_k, \boldsymbol{u}_k) = \sum_{i=0}^{N-1} \left(\left\| \boldsymbol{x}_{i|k} - \boldsymbol{x}_k^{(s)} \right\|_Q^2 + \left\| \boldsymbol{u}_{i|k} \right\|_R^2 \right) &+ \left\| \boldsymbol{x}_{N|k} - \boldsymbol{x}_k^{(s)} \right\|_{Q_f}^2 + \lambda \sum_{i=1}^{N} \mathbf{1}^T \left(\boldsymbol{H_i} \boldsymbol{x}_{i|k} \right)_+ \\
\text{s.t.} \quad \boldsymbol{x}_{i+1|k} &= \boldsymbol{A} \boldsymbol{x}_{i|k} + \boldsymbol{B} \boldsymbol{u}_{i|k} \\
\boldsymbol{x}_{0|k} &= \boldsymbol{x}(t_k) \\
\hat{\boldsymbol{n}}_{cone,i|k}^T \boldsymbol{x}_{i|k} &\geq r_2 \qquad\qquad i = 0, ..., N \\
|\boldsymbol{u}_k|_\infty &\leq u_{max}
\end{aligned} \tag{3.10}$$

3.3 Closing or Final Phase

The final phase starts when the relative distance between the chaser and target spacecrafts hits the r_2 distance within the approach cone. The LoS cone constraint is tightened to the angle required by the target's docking mechanism.

3.3.1 Objective Cost

In the previous phases, the state objective was being guided relative to some chosen reference point within the approach cone. If this reference point is left out then the state objective is driven to the target's CoM. To account for a docking port at a known location on the target spacecraft that may not correspond to the CoM, the state error from the docking port position is used as the quadratic objective in the optimization. Additionally, the LoS cone penalty in the objective function also must be centered at the docking port position, $x^{(d)}$, in the LVLH frame. Originally, the cone constraint was an external penalty function in the running cost (as in previous iterations of this work [33, 34]), but has been removed and made a constraint in the QP problem.

$$\min_{\boldsymbol{u}_k} J(\boldsymbol{x}_k, \boldsymbol{u}_k) = \sum_{i=0}^{N-1} \left(\left\| \boldsymbol{x}_{i|k} - \boldsymbol{x}_{i|k}^{(d)} \right\|_Q^2 + \left\| \boldsymbol{u}_{i|k} \right\|_R^2 \right) + \left\| \boldsymbol{x}_{N|k} - \boldsymbol{x}_{N|k}^{(d)} \right\|_{Q_f}^2 \tag{3.11}$$

3.3.2 Constraints

The cone constraint is added as a hard constraint that must be satisfied by the solution. In this manner it serves as the overshoot constraint as well saying that the chaser spacecraft must be within whatever positional requirements are set by the docking mechanism. The polyhedron for the docking cone is in general more restrictive than the approach polyhedron penalty.

$$\boldsymbol{H}_i \left(\boldsymbol{x}_{i|k} - \boldsymbol{x}_{N|k}^{(d)} \right) \leq 0 \qquad i = 0, ..., N \tag{3.12}$$

Another new constraint that has been added to the set of docking constraints is on the velocity of the chaser spacecraft. This is the so-called "soft-docking" constraint and is to limit how fast the chaser can approach the docking port. Using $\boldsymbol{v} = \dot{\boldsymbol{\rho}}$ and some allowable max docking velocity v_{max}:

$$\left|\boldsymbol{v}_{i|k}\right|_{\infty} \leq v_{max} \qquad\qquad i = 0, ..., N \tag{3.13}$$

In [14] there is an exponential velocity decay constraint used in place of an upper velocity limit. However, this was only done on the velocity in the approach direction, when it should be ensured that the chaser is approaching slowly in any direction.

3.3.3 Docking QP

Using the running cost from Equation 3.11 and the constraints Equation 3.12 and Equation 3.13 the full docking QP problem can be formed as

$$\begin{aligned}
\min_{\boldsymbol{u}_k} J(\boldsymbol{x}_k, \boldsymbol{u}_k) = \sum_{i=0}^{N-1} \left(\left\|\boldsymbol{x}_{i|k} - \boldsymbol{x}_{i|k}^{(d)}\right\|_Q^2 + \left\|\boldsymbol{u}_{i|k}\right\|_R^2 \right) + \left\|\boldsymbol{x}_{N|k} - \boldsymbol{x}_{N|k}^{(d)}\right\|_{Q_f}^2 \\
\text{s.t.} \quad \boldsymbol{x}_{i+1|k} = \boldsymbol{A}\boldsymbol{x}_{i|k} + \boldsymbol{B}\boldsymbol{u}_{i|k} \\
\boldsymbol{x}_{0|k} = \boldsymbol{x}(t_k) \\
\boldsymbol{H}_i \left(\boldsymbol{x}_{i|k} - \boldsymbol{x}_{N|k}^{(d)}\right) \leq 0 \qquad\qquad i = 0, ..., N \\
\left|\boldsymbol{u}_{i|k}\right|_{\infty} \leq u_{max} \\
|\boldsymbol{v}| \leq v_{max}
\end{aligned} \tag{3.14}$$

3.4 Constant Rotating Target

In a scenario with a tumbling target, the docking port will not be fixed in the LVLH frame. Consequently, the LoS cone must also rotate in the LVLH frame in the minimization QP's.

For example, assuming a constant rotation about the x-axis, $\boldsymbol{\omega}_B = \omega_B \hat{i}$, the LoS polyhedron evolves as a simple rotation about the LVLH x-axis.

$$H_{i+1|k} = H_{i|k} \boldsymbol{R}_1^T \left(\omega_b \Delta T_s \right) \tag{3.15}$$

In Equation 3.7, the approach cone polyhedron H should be taken at time t_N in the prediction horizon. In Equation 3.10 and Equation 3.14, the approach and docking cone axes $\hat{\boldsymbol{n}_{cone}}$ also must rotate in accordance with the target body rotation.

3.5 Guidance Algorithm

This section describes the full guidance algorithm strategy as presented in algorithm 1. The algorithm begins in the rendezvous phase at time $t_k = t_0$, solving Equation 3.7 for the optimal control sequence, $\boldsymbol{u}_k = \left\{ \boldsymbol{u}_{0|k}, \boldsymbol{u}_{1|k}, ..., \boldsymbol{u}_{N|k} \right\}$ with prediction horizon $N = N_1$ and sample time $t_s = t_{s,1}$. Only the first element in the control sequence, $\boldsymbol{u}_{0|k}$, is applied to the chaser. At the next sample instance, if the chaser spacecraft is within the approach cone the algorithm proceeds to the approach phase. Otherwise, the rendezvous optimal control sequence calculation is repeated.

The approach phase is similar to the rendezvous phase, but uses Equation 3.10 to solve for the optimal control sequence. Again, only the first element in the control sequence is applied. The prediction horizon and sample time are shorter than those of the rendezvous phase. The shorter times between control allows for quicker response as the chaser closes in on the target's position. When the chaser reaches a prescribed distance from the target, r_2, the docking phase begins.

In the final phase, docking, Equation 3.14 is solved for the optimal control sequence. As with the other phases, the first element of the sequence is the control applied to the system. The prediction horizon is shortest in this phase. The time between control is also at its shortest due to the close-proximity to the target where quick control is needed.

Algorithm 1: AR&D Guidance

```
Initialize t_k = t(0), x_k = x(0)
begin phase 1 Rendezvous
    Set rendezvous prediction horizon parameters
    N ← N_1
    t_s ← t_{s,1}
    repeat
        Solve (Equation 3.7) for optimal control sequence u_k
        Apply first control element, u_k^0
        k ← k + 1
        Update state estimate, x_k
    until x_k <= H x_k and ||x_k|| ≤ r_1      // End Rendezvous phase if
     inside approach cone
end
begin phase 2 Approach
    Set approach prediction horizon parameters
    N ← N_2
    t_s ← t_{s,2}
    repeat
        Solve (Equation 3.10) for optimal control sequence u_k
        Apply first control element, u_k^0
        k ← k + 1
        Update state estimate, x_k
    until x_k ≤ r_2                            // End Approach phase
end
begin phase 3 Docking
    Set docking prediction horizon parameters
    N ← N_3
    t_s ← t_{s,3}
    repeat
        Solve (Equation 3.14) for optimal control sequence u_k
        Apply first control element, u_k^0
        k ← k + 1
        Update state estimate, x_k
    until x_k ≤ r_3                            // End Maneuver
end
```

3.6 Simulation and Results

The 42 simulation created to utilize the three-phase MPC algorithm considers two CubeSat spacecraft in Low-Earth Orbit (LEO). Both are assumed to be 14 kg and 6U in size. The chaser vehicle is fitted with 500 mN thrusters that are configured to provide thrust in all axes. Three cases of this scenario are run. The first case utilizes a V-bar (or transverse direction) approach of the chaser to a non-rotating target. In Case 2, the target is again non-rotating, but the chaser approaches from the R-bar (or radial) direction. Case 3 considers a target that is rotating at a constant rate about its Body z-axis, which is initially aligned with the LVLH z-axis.

The scenarios begin with an initial chaser position approximately 1.7 km from the target. For the far-range rendezvous phase, the control horizon is $N_1 = 30$ steps with a time between control of 30 seconds. In the approach phase the time steps are reduced to $t_{s,2} = 20$ seconds and a prediction horizon of $N_2 = 15$. During the docking phase the LoS cone is tightened for a smaller keep-in zone and the sample time is again reduced to $t_{s,3} = 5$ second intervals with a prediction horizon of $N_3 = 12$. The parameters considered for the MPC problem of each phase are shown in Table 3.1 and Table 3.2.

3.6.1 V-bar approach (Case 1)

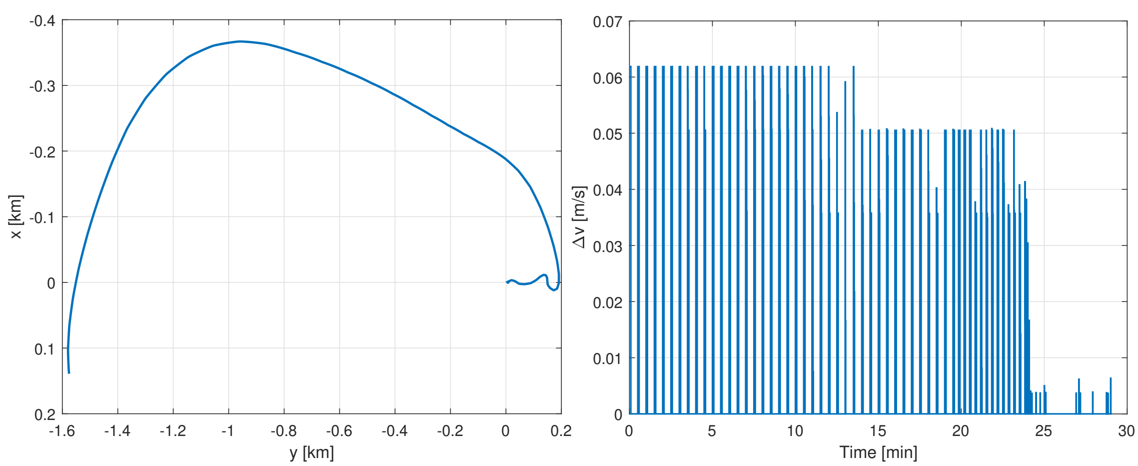
The first scenario considered is a chaser performing a V-bar approach to a non-rotating target. The axis of the LoS cones and overshoot constraint in the QP's is taken to be along the velocity (y-axis) direction. The resulting trajectory of the chaser in the CW-frame is shown in Figure 3.3a. From Table 3.3 it can be seen that the rendezvous is completed in around 29 minutes with a total Δv of 13.28 m/s.

Table 3.1: Simulation MPC parameters

Parameter	**Far-Range**	**Mid-Range**	**Close-Range**
T_s [s]	30	20	5
N	30	15	12
Q	$\text{diag}\{1, 1, 100, 1, 1, 10\}$	$\mathbf{I}_{6\times6}$	$20\mathbf{I}_{6\times6}$
R	$10^3\mathbf{I}_{3\times3}$	$10^3\mathbf{I}_{3\times3}$	$10^6\mathbf{I}_{3\times3}$
λ	10	10	10
LoS Angle	15	15 deg	8 deg
u_{max} [m/s^2]	0.1786	0.1786	0.0357

Table 3.2: Spacecraft Initial Conditions

Parameter	**Value**
Chaser Position in CW Frame (x, y, z)[m]	$(140, -1580, 60)$
Chaser Velocity in CW Frame (v_x, v_y, v_z) [m/s]	$(-0.90, -0.32, 0)$
Docking Port in Target Body Frame (x_d, y_d, z_d) [m]	$(0.5, 0, 0)$



(a) (b)

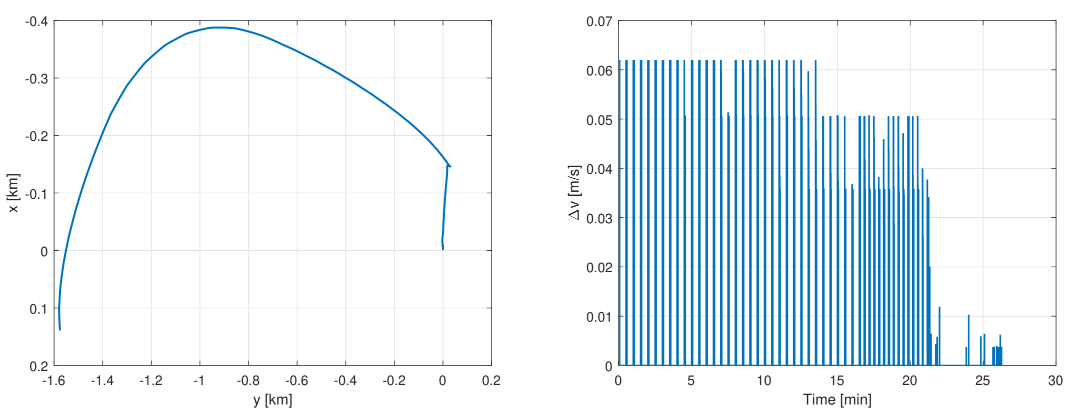
Figure 3.3: V-bar docking (Case 1) results. (a) In-plane motion of the chaser in the CW frame. (b) Magnitude of resultant thrust forces during the maneuver.

Table 3.3: Case 1. V-bar Maneuver Results

Phase	**Total Δv [m/s]**	**Time [s]**
Far Range	10.24	1169
Mid-Range	2.84	260
Final Approach	0.20	315
Total	**13.28**	**1744**

3.6.2 R-bar approach (Case 2)

For Case 2, the chaser performs an R-bar approach to the target. The LoS cones and overshoot constraint in the control minimization QP are adjusted accordingly. The resulting trajectory and control thrust magnitudes are shown in Figure 3.4. From Table 3.4, it can be seen that for this initial state, the R-bar approach takes less time as well as less fuel than in Case 1. This is expected because the chaser does not have to continue as far around the target to reach the approach cone.



(a) (b)

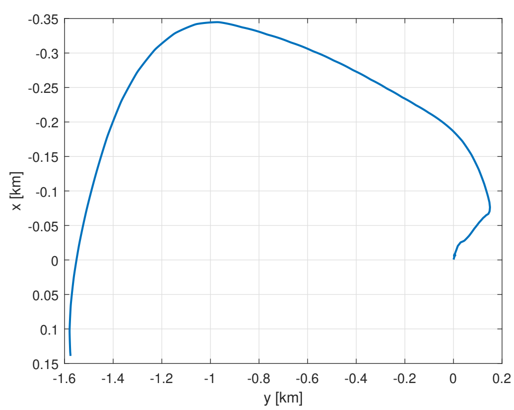
Figure 3.4: Case 2 simulation (R-bar approach) results. (a) In-plane chaser motion in the CW frame. (b) Magnitude of resultant thrust forces during the maneuver.

Table 3.4: Case 2. R-bar Maneuver Results.

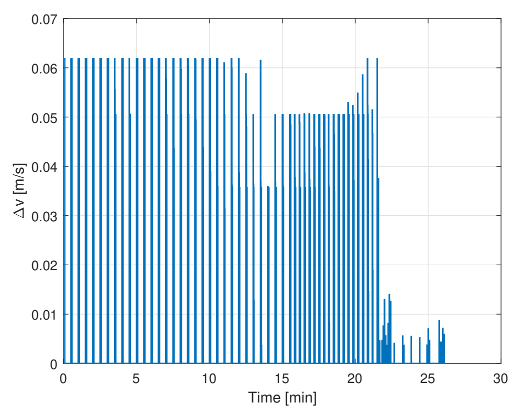
Phase	Total Δv [m/s]	Time [s]
Far Range	9.19	989
Mid-Range	2.76	280
Final Approach	0.18	310
Total	**12.13**	**1579**

3.6.3 Rotating Target (Case 3)

In the last simulation, the target is rotating with a constant rate about its body z-axis. The location of the docking port is initially aligned in the V-bar (y-axis) direction. The total amount of Δv needed is comparable to Cases 1 and 2, as seen in Table 3.5. The biggest difference lies in the closer approach and docking phases. This is expected as the maneuver needs to match the docking port's rotation. The overall implication is that matching a rotating object does not appear to add a significant amount of Δv.



(a)



(b)

Figure 3.5: Case 3 Target with a constant rotation. (a) In-plane rendezvous maneuver in the CWH frame with resultant control vectors. (b) Magnitude of resultant thrust forces during the maneuver.

Table 3.5: Case 3. Rotating Target Maneuver Results

Phase	**Total Δv [m/s]**	**Time [s]**
Far Range	8.38	929
Mid-Range	4.04	360
Final Approach	0.25	280
Total	**12.67**	**1569**

3.6.4 Rendezvous Cost Comparisons

This section showcases the differences between having the reference point as a part of the rendezvous phase objective cost (Equation 3.2). An additional comparison is using a 1-norm running cost objective as in Equation 2.10 and making the rendezvous phase problem an LP instead of a QP. The full maneuver was simulated three times using the same starting conditions but with the different rendezvous optimal control problems. The trajectories for all three maneuvers are shown in Figure 3.6 and the resulting Δv and t_f are displayed in Table 3.6.

First, shown in red is the QP setup that uses a reference point inside of the approach cone. This is the result from the V-bar case in the above sections. Without the set point (in blue), it can be seen that the trajectory is driven to the keep-out zone and is guided around the zone to the approach cone. Due to the fighting of the optimization to drive the chaser state to the origin and to adhere to the keep-out zone, the chaser moves slowly around resulting in a much longer maneuver time of 71.23 min compared to the previous case of around 29 minutes. The LP formulation in yellow interestingly takes a different direction towards the approach cone for this particular set of initial conditions. While the maneuver also takes longer (40.57 min) it does use less Δv which is the expected result.

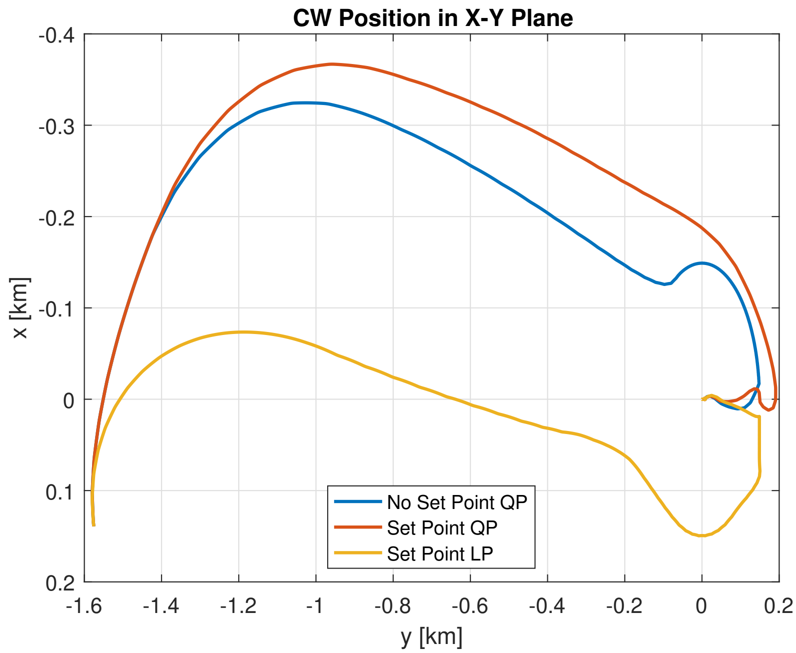


Figure 3.6: Comparison of Rendezvous QP and LP Optimal Control.

Table 3.6: Comparison of Rendezvous Cost Objectives

Cost Strategy	**Total Δv [m/s]**	**Time [min]**
QP without Set Point	12.65	71.23
QP with Set Point	13.28	29.07
LP	12.74	40.57

Moving forward, this paper assumes a rendezvous setup using the QP optimal control problem with a set point. However, if desired the LP problem setup may be used for sparser control to reduce fuel if desired by a mission. The following analyses in later chapters can be seen for maneuvers with LP problem rendezvous in the appendix.

3.7 Future Considerations

While the well-known CW equations are a good starting point and commonly used, they assume circular orbits and do not capture effects from disturbances like J2. Including J2 in the linearized prediction model would benefit the algorithm accuracy. One approach to consider more elliptical orbits is shown in [36] by using the previously mentioned Tschauner-Hempel model for relative motion. Another option to consider is the use of the Schweighart-Sedwick model [37], which is similar to the CW model, but includes effects of J2. Additionally, there are two schools of thought for improving the handling of uncertainties with MPC: stochastic MPC [38] and tube-based robust MPC [36, 39] that could be explored for implementation. These types of controllers would benefit the algorithm to more stability and knowledge about response to disturbances.

CHAPTER 4
SENSITIVITY TO MPC PARAMETERS

The resulting Δv and time-to-dock t_f are heavily dependent on where the chaser spacecraft begins the maneuver relative to the target. Additionally, the developed algorithm response in each phase is dependent on the length of the prediction horizon and the sampling time for the optimal control problems. The effect of these two parameters on the performance of the algorithm was investigated for any trends that may help with parameter selection. The two parameters were chosen for assessment instead of the QP weightings since they have more of a computational effect of the algorithm, namely the time between solutions and the computational complexity. For each phase of the algorithm, the sample time of the phase was varied while holding the other phase sample times and prediction horizons constant. This sample time varying was repeated for differing prediction horizons.

The tests consisted of 500 chaser initial states that were chosen by random distribution. The generated points are shown in Figure 4.1 with a note that the x-axis negatives are towards the top and in the Earth direction. The relative distance is uniformly distributed between $0.15 - 2.5$ km. The relative velocity is perturbed off of the required stable orbit velocity with a normal distribution of zero mean and 0.5 m/s standard deviation in each axis.

4.1 Initial Conditions

To test the ending values based on the initial relative starting position the algorithm was run for the 500 generated points to perform a V-bar maneuver. This simulation was done with the baseline MPC parameters from Table 3.1. Two parameters were checked for their importance on the ending Δv and t_f: starting quadrant and radial distance from the target.

The starting quadrant in the LVLH frame is of interest because it affects how far around

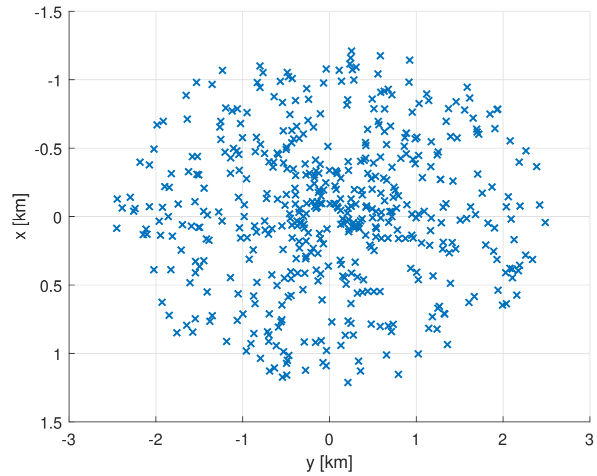
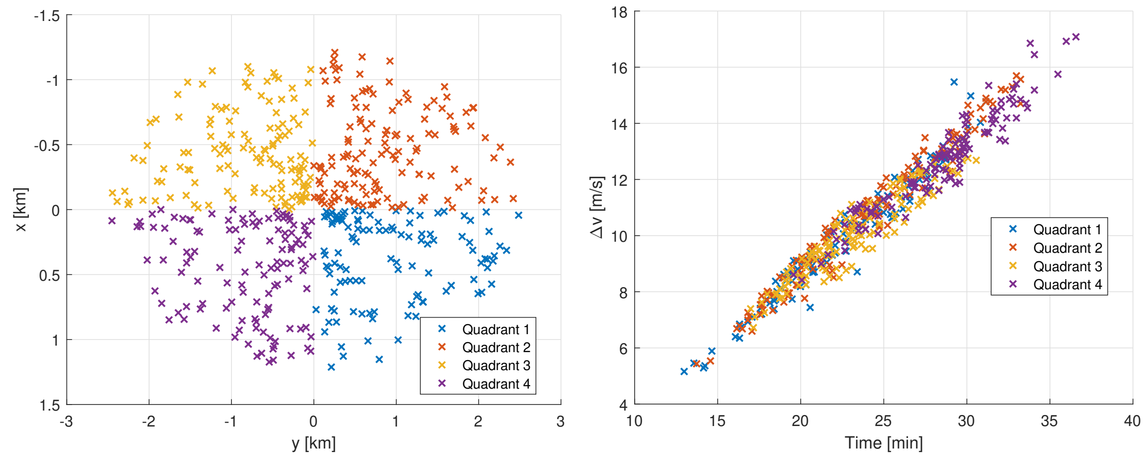


Figure 4.1: 500 Randomly Generated Initial Chaser Positions in the LVLH Frame.

the target the chaser must fly in the rendezvous phase to reach the approach cone. This extra flying time obviously would increase the time-to-dock, but the question remains by just how much compared to starting conditions in the same half-plane as the target. Figure 4.2a colors the initial chaser positions by quadrant while Figure 4.2b shows the ending Δv and final maneuver time. The quadrant 3 and 4 positions are, in general, farther to the right on the plot indicating the expected increased time it takes for the maneuver to finish. The quadrant 3 and 4 Δv's also have high lower bounds indicating that starting the maneuver on the other half-plane to the docking mechanism is more likely to result in more needed Δv. Interestingly, however, the maximum needed Δv does not seem as affected by quadrant as the minimum, as it is not uncommon for the quadrant 1 and 2 maximums to also reach high values. In conclusion, it is not possible to determine a range of expected Δv and t_f on quadrant alone.

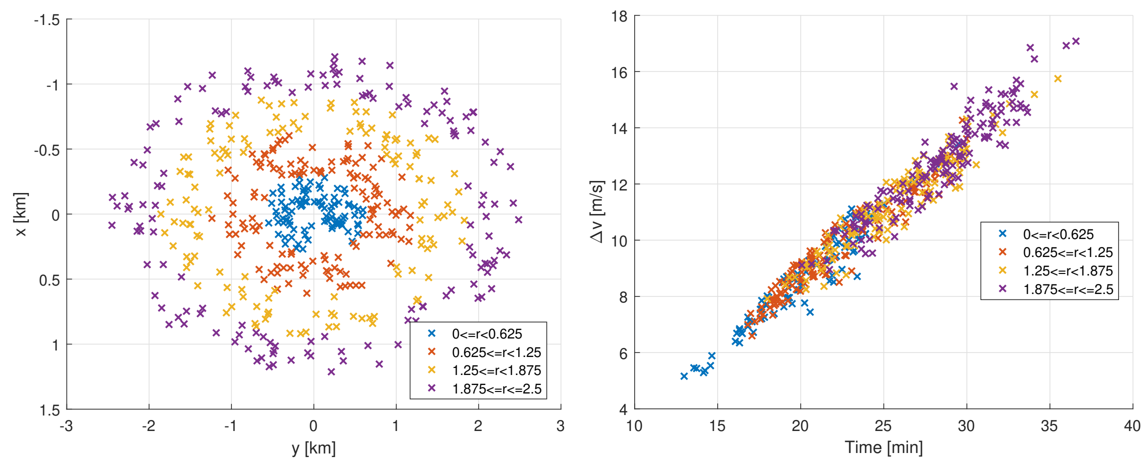
The radial starting distance from the target CoM is an obvious area of interest because the chaser must travel farther which would take more time and actuation leading to higher needed Δv. From the radially segmented results in Figure 4.3, a couple observations are made. Like the quadrant results, there is a noticeable effect on the minimum Δv and t_f. Farther starting distances correlate to higher values, as expected. The consequence of



(a) (b)

Figure 4.2: Quadrant colored (a) initial chaser positions and (b) resulting Δv vs t_f.

starting distance is also realizable in the maximum values is also noticeable than changing the quadrant.



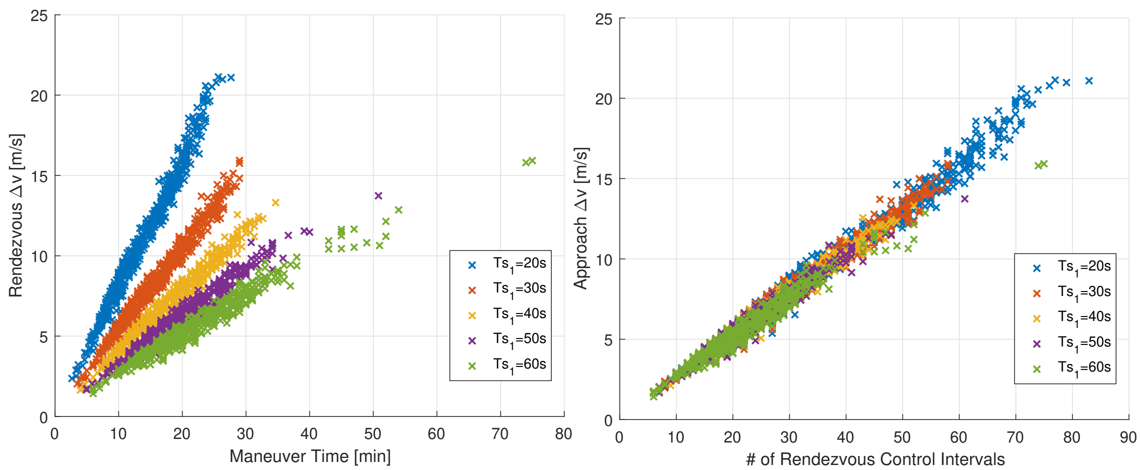
(a) (b)

Figure 4.3: Radially colored (a) initial chaser positions and (b) resulting Δv vs t_f.

From these results, it can be seen how important the selection of starting chaser state relative to the target is and shows how implementations of the algorithm can test for worse case scenarios.

4.2 Rendezvous Phase

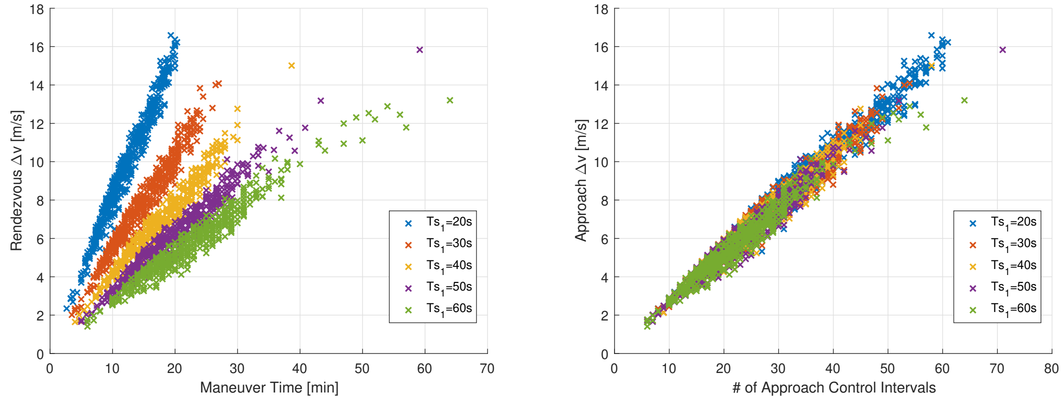
The tested sample times for the rendezvous phase are $T_s = 20, 30, 40, 50, 60$ s. For the prediction horizons, the chosen values are $N_1 = 15, 30, 45$. The prediction horizon of $N_1 = 45$ is pushing the ability of CVXGEN's code generation to the limit and acts as an upper limit capability. It should be noted that it is expected that longer times in between control actuations would result in longer times for the rendezvous phase to finish. To normalize this effect, in addition to Δv vs t_f, the Δv vs the number of control actuations needed to complete the rendezvous phase is explored. The results from $N_1 = 15, 30, 45$ are shown in Figure 4.4, Figure 4.5, and Figure 4.6.



(a) (b)

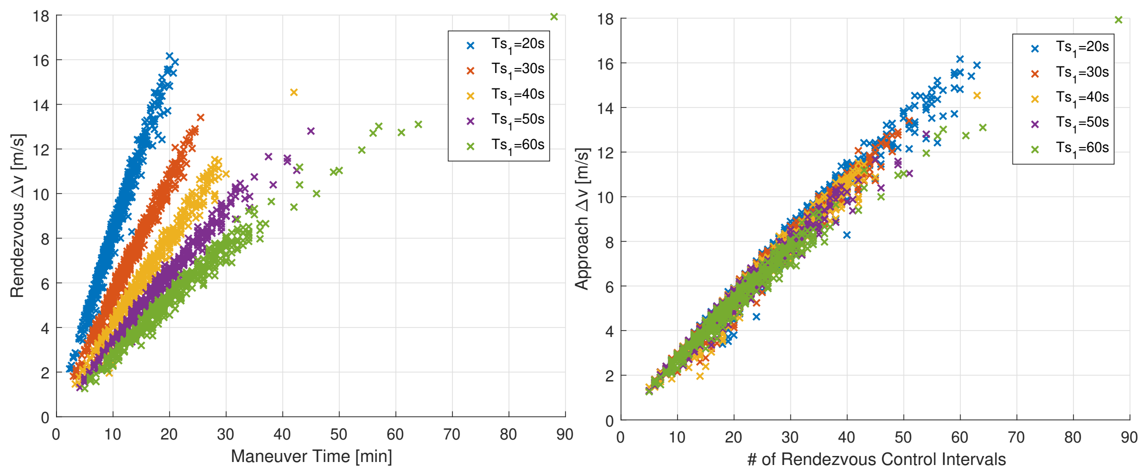
Figure 4.4: Rendezvous Phase with $N_1 = 15$.

The first parameter to look at is the differences between the sample times. Longer time between actuations decreases the needed Δv in the rendezvous phase across all three tested prediction horizons. The decrease in Δv does appear to have diminishing returns, as the difference between $T_{s,1} = 50$ s and $T_{s,2} = 60$ s is not as large as the other intervals. Additionally, the longer sampling times also decrease the number of needed actuations. However, it must be noted that this results in longer rendezvous times overall. Another observation can be made in the spread of the resulting Δv's. The faster sampling times have a wider amplitude range of Δv when compared to the longer sampling times. However,



(a) (b)

Figure 4.5: Rendezvous Phase with $N_1 = 30$.



(a) (b)

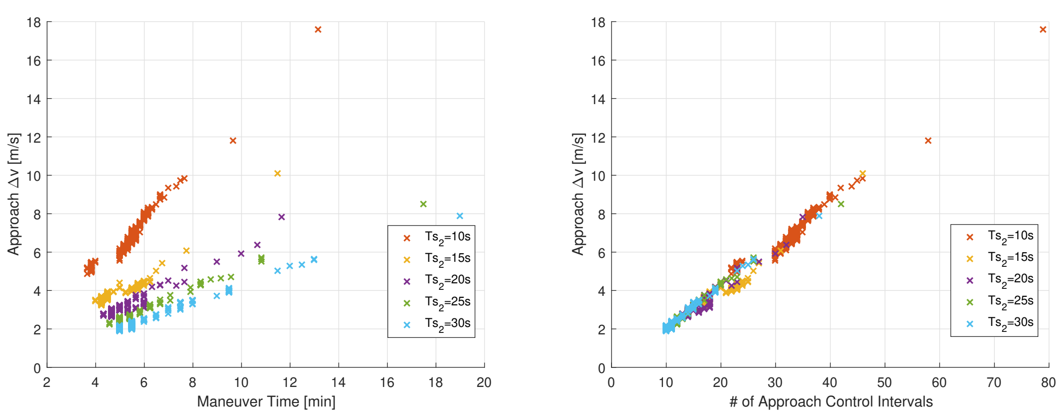
Figure 4.6: Rendezvous Phase with $N_1 = 45$.

the longer sampling times are not without issue as there started to be non-convergence in the algorithm. When running the simulations, maneuvers were given 200 minutes to complete as this is more than enough time on average. Failure to complete within this time is strongly correlated with the rendezvous solution converging to a point on the keep-out zone boundary, something that is rarely seen in the faster sampling times as well as lower prediction horizons. For $N_1 = 30$ and $T_{s,1} = 60$ s, 2 out of the 500 test cases failed to rendezvous within the 200 minute limit. For the $N_1 = 45$ case, the $T_{s,1} = 50$ s test had 1 case fail and the $T_{s,1} = 60$ s test had 6 failures. It is recommended then that sampling times between the 30-40s range are best considered for this phase.

Increasing the prediction horizon also has an appreciable effect on the rendezvous results. From $N_1 = 15$ to $N_1 = 30$ the needed Δv and t_f decreases for all sampling times in general. The decrease appears to have diminishing returns, as the results between $N_1 = 30$ and $N_1 = 45$ have similar characteristics. Since there does not seem to be an appreciable decrease in Δv from the longer horizon, the $N_1 = 30$ should be preferred as it is expected to have less computational effort, which will be shown in a later chapter.

4.3 Approach Phase

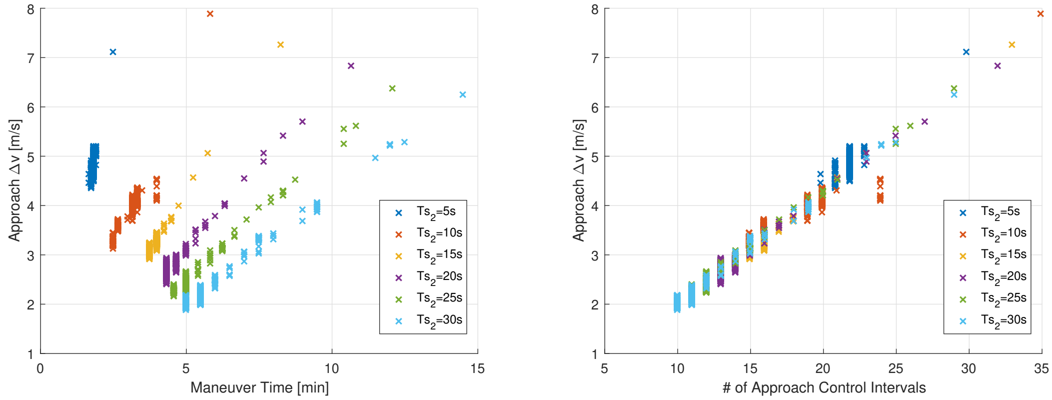
For the approach phase, the sample times tested are $T_s = 5, 10, 15, 20, 25, 30$ s with prediction horizons of $N_2 = 5, 15, 30$. To test the approach phase, the ending points of the rendezvous phase with $N_1 = 30$ and $T_{s,1} = 30$ s were selected as the initial conditions. Similar to the rendezvous results, the time from approach phase start to end and number of control actuations within the approach phase are studied. The results for $N_2 = 5$, $N_2 = 15$ and $N_2 = 30$ are shown in Figure 4.8, Figure 4.7, and Figure 4.9, respectively.



(a) (b)

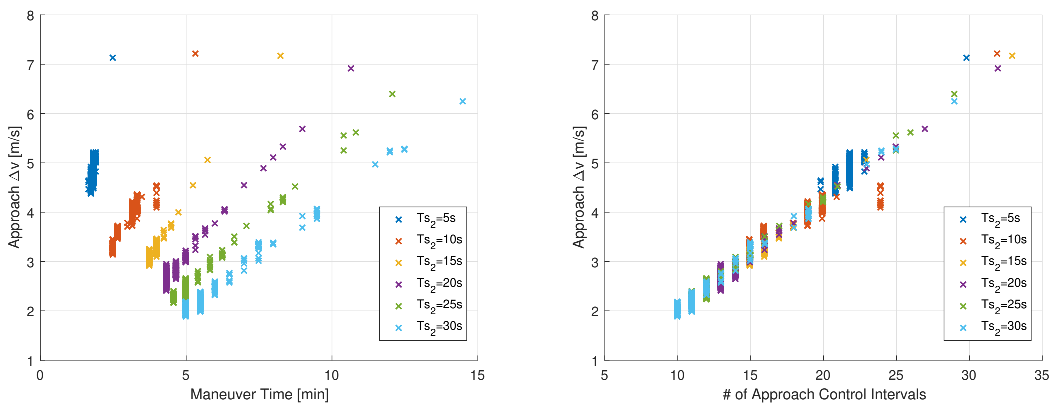
Figure 4.7: Approach Phase with $N_2 = 5$.

As with the rendezvous phase, the effect of sampling time on the approach phase is easily observable. Increasing the time between control actuations lowers the minimum needed



(a) (b)

Figure 4.8: Approach Phase with $N_2 = 15$.



(a) (b)

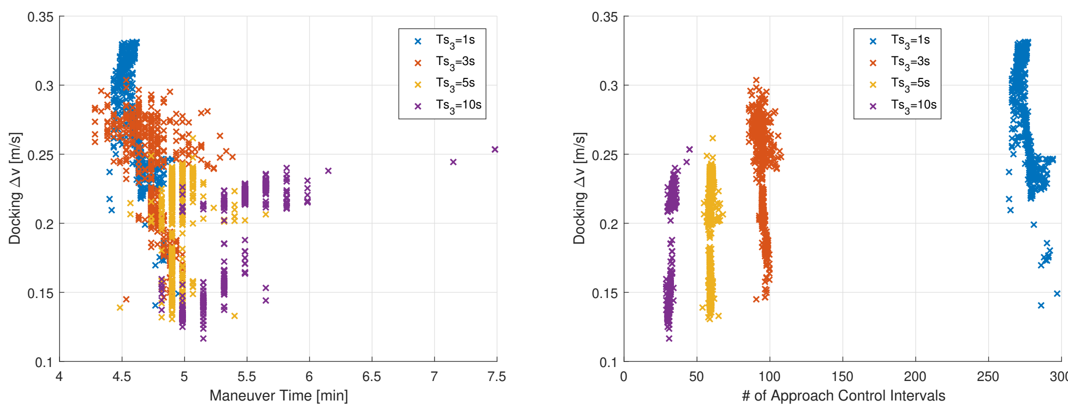
Figure 4.9: Approach Phase with $N_2 = 30$.

Δv for these test cases. However, the spread of needed Δv becomes larger with the increasing sample time. This increase in spread with increased sampling time is also observed in the time of the approach phase. The $T_{s,2} = 10$ s and $T_{s,2} = 15$ s are recommended for the lowest Δv needed to range of possible values. It should be noted that for the $N_2 = 5$ horizon that sample times of $T_{s,2} = 5s$ did not have feasible solutions.

The Δv dropped dramatically between $N_2 = 5$ and $N_2 = 15$. Interestingly, for prediction horizon values of $N_2 = 15$ and $N_2 = 30$, there is not an appreciable difference observed. The resulting Δv and time characteristics are mostly the same. It is recommended to select the smaller of these two prediction horizons for the lesser computational effort needed to solve for the optimal control sequence.

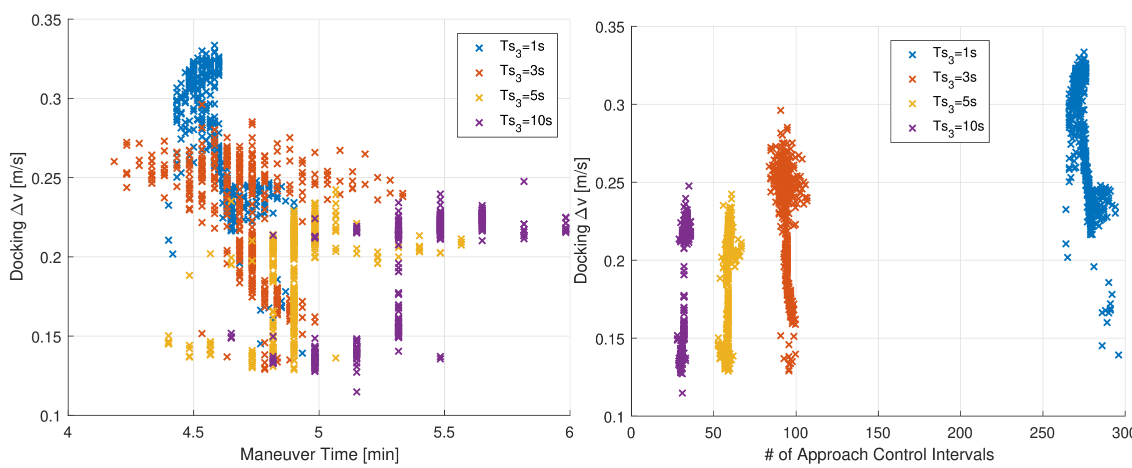
4.4 Docking Phase

For the docking phase, the sample times are $T_s = 1, 3, 5, 10$ s with prediction horizons of $N_3 = 5, 12$. The initial conditions for this phase were taken from the ending of the approach phase using scenarios with $N_1 = 30$, $T_{s,1} = 30$ s, $N_2 = 15$, and $T_{s,2} = 20$ s.



(a) (b)

Figure 4.10: Docking with $N_3 = 5$.



(a) (b)

Figure 4.11: Docking with $N_3 = 12$.

The most obvious observation for the docking phase is the wide gap between needed control actuations for the varying sampling times, with 1 second intervals needing more

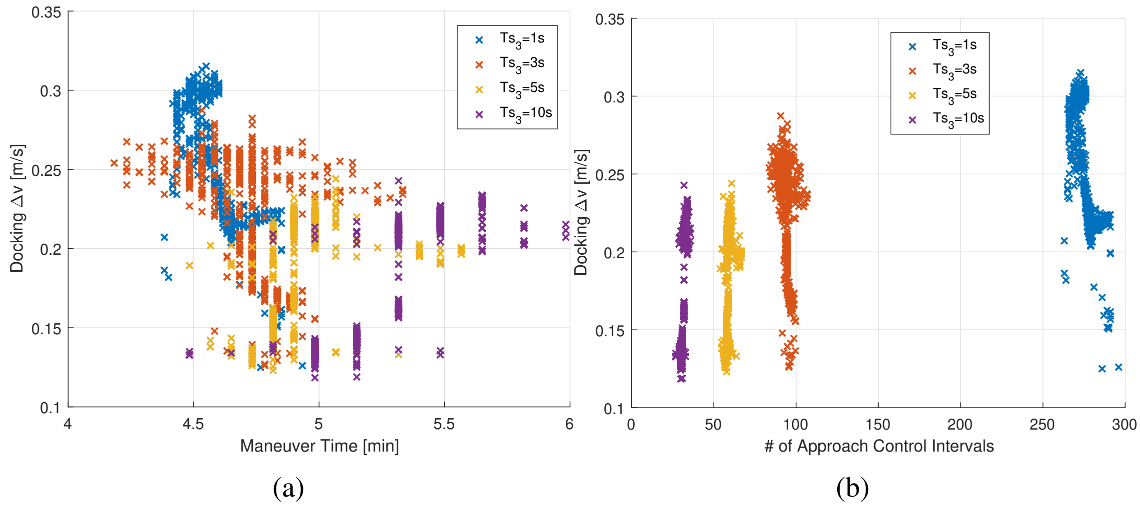


Figure 4.12: Docking with $N_3 = 30$.

than double the other intervals. The spread of the docking phase time may look compact within sampling times when looking at the number of control intervals, but this is due to the scaling. The spread is more easily seen in the time to dock, showing that longer sampling times need more time. This is most likely attributed to taking longer to correct the spacecraft to remain within the tight docking corridor. The maximum and minimum needed Δv does decrease with longer sampling times as with the other phases. However, this difference is relatively tiny compared to the ranges of the rendezvous and approach phases. There is not much difference between the time to dock for sampling times when compared across prediction horizons.

Based on these observations, shorter prediction horizon in this phase is recommended for faster docking times. This is similar to the approach phase conclusion where the relatively small change with longer prediction horizon is not very important compared to the possible computation time increase. Smaller sampling times around $T_{s,3} = 3 - 5$ s are also recommended for minimizing the needed Δv.

4.5 Future Considerations

The presented analysis shows the relative differences between the chosen MPC parameters, but may not give the optimal Δv and t_f values for every situation. Other parameters that can be changed are the weights in the cost objectives of each phase. Studying these weights would improve the understanding of the overall characteristics and confirm that these relative findings remain for different weights.

CHAPTER 5

ROBUSTNESS OF MPC GUIDANCE

A desired property of the developed MPC law is robustness to disturbances. The robustness is a result of the closed-loop nature of the algorithm. Disturbances may be present in the dynamics of the system model, sensor measurements, or in the actuators. This chapter focuses on testing the resilience of the MPC guidance under the effects of unknown state errors and the provided actuation.

5.1 Actuation Noise

As stated previously (2.1.2), the impulse provided by a thruster is related to the commanded on-time to that thruster. Variations in the ramp-up and ramp-down timings of the valves are one source of error in the realized impulse. However, these variations are typically very small compared to the minimum pulse and can be characterized through testing. Another source of error is in the nominal thrust applied by the thruster. The nominal thrust may change with factors such as temperature and pressure of the system (T_{sys} and P_{sys}) as well as time in between pulses.

$$J_{cmd}(t) = F_{applied}(T_{sys}, P_{sys}, \Delta t) * t_{pulse} \tag{5.1}$$

Instead of trying to model variations in the applied thrust force due to these sources, the error in the actual thrust versus commanded thrust was modeled as a time-correlated noise in the amplitude of the commanded thrust.

$$F_{applied} = F_{nom} + w(t)F_{nom} = (1 + w(t))F_{nom} \tag{5.2}$$

where $w \in (-1, 1)$ is a random variable for the percent change in the nominal thrust force.

The amplitude error, $w(t)$, is described by the following stable error model developed by Carpenter and Lee in [40].

$$\begin{bmatrix} \dot{w} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} -1/\tau & 1 \\ -\omega_n^2 & -2\xi\omega_n \end{bmatrix} \begin{bmatrix} w \\ v \end{bmatrix} + \begin{bmatrix} \eta_w \\ \eta_v \end{bmatrix} \tag{5.3}$$

This Stochastic Differential Equation (SDE) behaves similarly to a random walk over a designed length of time. The benefit of using this Gauss-Markov (GM) model is that the steady-state covariance is asymptotically bounded to avoid cases where the error increases indefinitely. The GM parameters τ, ω_n, and ξ are chosen to match the physics and variability of the error phenomenons (such as pressure variations). Additionally, η_ω and η_v, are tuned for appropriate covariance amplitudes. While this SDE may not be the most realistic model for a propulsion system, it allows for testing the algorithm's abilities to handle errors in the actuation without attempting to model a whole propulsion system. This encompasses a variety of error sources, including unknown modeling errors, within one model. A sampling of propulsion amplitude errors using this model are shown in Figure 5.1.

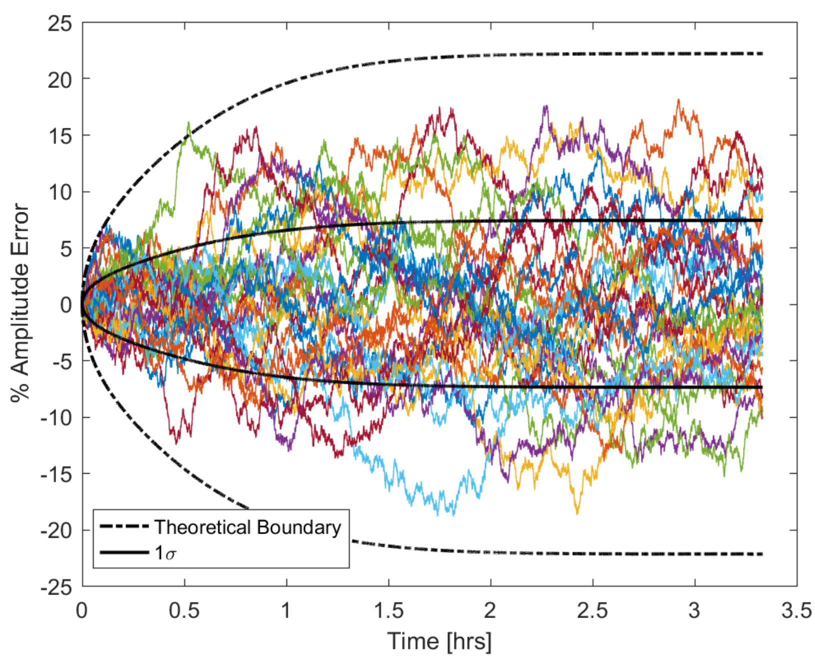


Figure 5.1: Sampling of thruster amplitude errors.

The black dotted line marks the 3-sigma bounds for the steady-state covariance over time. As can be seem from the figure, the value tends to stay within the $\pm 7\%$ thrust amplitude error range (within 1σ) which is not unreasonably high.

5.2 Disturbance Models

The 42 spacecraft simulator is capable of simulating perturbations from solar radiation pressure and third body effects, but these were determined to be much smaller in magnitude than the effect of Earth's gravity model and atmospheric drag so they are not covered here.

5.2.1 Earth's Oblateness

The 42 spacecraft simulator uses the EGM96 reference geoid to provide the spherical harmonics. For the purposes of the presented work the spherical harmonics up to degree $n = 8$ and order $m = 8$ were used to provide gravitational perturbations. The largest effect of spherical harmonics is the so-called J_2 term. The higher degrees and orders are not strictly needed since they are comparatively smaller compared to J_2, but since these magnitudes can be provided easily by 42 they were implemented.

5.2.2 Atmospheric Drag

Disturbance from atmospheric effects is also considered within the simulation framework. Spacecraft are modeled as a series of N flat plate objects with area A. Given the velocity of the atmosphere relative to the spacecraft motion, $\boldsymbol{v}_{vrel}$, the angle between the jth flat plate with normal vector $\boldsymbol{n}_j$ and the relative atmosphere velocity in the spacecraft body frame B is given by

$$\theta_j = \arccos\left(\frac{\boldsymbol{n}_{j_B} * \boldsymbol{v}_{rel_B}}{\|\boldsymbol{v}_{rel_B}\|^2}\right) \tag{5.4}$$

Summing the atmospheric drag on each flat plate gives the total atmospheric drag on the vehicle:

$$F_{aero_B} = -\frac{1}{2}\rho_{atm} \left\| \boldsymbol{v}_{rel_B} \right\|^2 C_D \sum_{j=0}^{N} A_j \max\left(\cos\theta_j, 0\right) \tag{5.5}$$

where ρ_{atm} is the atmospheric density and C_D is the dimensionless drag coefficient. The flat plate model is based on the 3D model object that is used by the 42 spacecraft simulator.

5.3 Measurement Noise

For simplicity, an Extended Kalman Filter (EKF) was not created for state estimation for either the target or chaser spacecraft. Although sensor errors due to bias or misalignment can be provided in 42, this was not used to provide measurement errors to remain spacecraft design agnostic in addition to the lack of a true EKF. Instead, zero-mean Guassian noise with standard deviation σ was added to the relative state to simulate a relative state estimate prior to use in the MPC algorithms.

$$\hat{\boldsymbol{\rho}}_k = \boldsymbol{\rho}_k + \boldsymbol{\eta}_{\rho|k} \qquad \boldsymbol{\eta}_{\rho|k} \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\sigma}_{\rho}\right) \tag{5.6}$$

$$\hat{\dot{\boldsymbol{\rho}}}_k = \dot{\boldsymbol{\rho}}_k + \boldsymbol{\eta}_{\dot{\rho}|k} \qquad \boldsymbol{\eta}_{\dot{\rho}|k} \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\sigma}_{\dot{\rho}}\right) \tag{5.7}$$

5.4 Monte Carlo Simulation

The robustness of the MPC guidance was tested with the discussed disturbances. Disturbances were provided by 42 for Earth's spherical harmonics and atmospheric drag. The spherical harmonics used went to degree and order 8 each. The drag coefficients for the target and chaser spacecraft were $C_{D_t} = 2.0$ and $C_{D_c} = 1.4$, respectively. The actuation noise parameters are given in Table 5.1 and also corresponds to the samplings in Figure 5.1. The initial thruster amplitude $\%$ error is initialized with a random normal distribution with

Table 5.1: Actuation Noise Parameters

Parameter	Value
τ	3600
ω_n	$2.6\text{e-}4$
ξ	1
η_w	$3/\sqrt{600}$
η_v	$1.1267\text{e-}8$

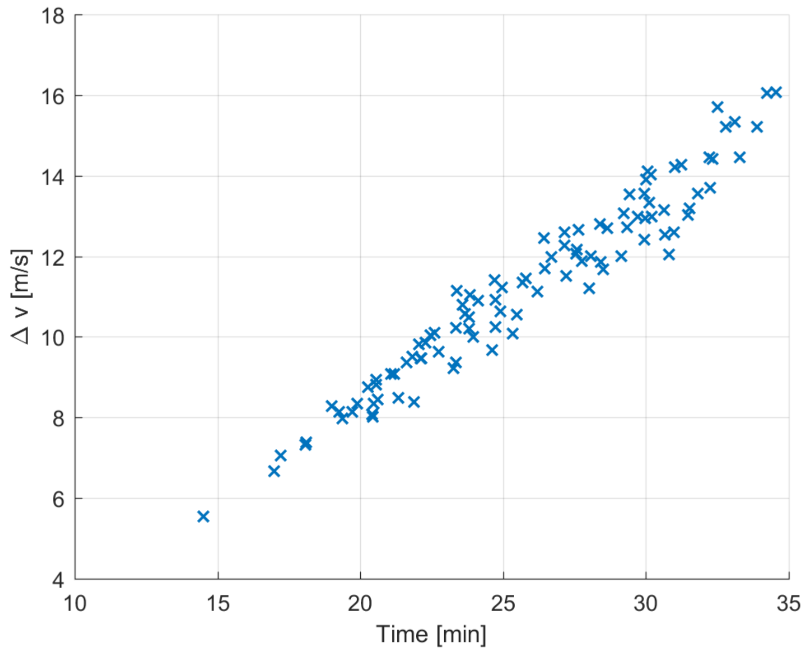


Figure 5.2: Average Δv vs t_f for measurement errors $\sigma_\rho = 5$ cm and $\sigma_{\dot{\rho}} = 1$ mm/s.

zero-mean and 1σ standard deviation.

The chaser initial relative state is selected by random distributions. The relative distance is uniformly distributed between $0.15 - 2.5$ km. The relative velocity is perturbed off of the required stable orbit velocity with a normal distribution of zero mean and 0.5 m/s standard deviation in each axis. For the MPC parameters, the values from Table 3.1 were used.

A set of 100 initial states were selected and each used 60 times to asses the response of the algorithm. For a first case, measurement errors of $\sigma_\rho = 5$ cm and $\sigma_{\dot{\rho}} = 1$ mm/s were used. The average Δv vs final time for the 100 states are shown in Figure 5.2.

The average standard deviation for the Δv and t_f are 4.25 m/s and 19.536 min, respectively. Most of the increase in Δv and t_f come from the docking phase where the chaser has a tendency to "bounce" around as it approaches the docking port. The effect of the error on the system in this phase is expected to improve with better assumptions on the measurement and sensor errors during this phase.

5.5 Future Considerations

As stated previously, implementation of stochastic MPC [38] and tube-based robust MPC [36, 39] could be explored to benefit the algorithm robustness. Additionally, the inclusion of a true EKF would create a more flight-like scenario. The expected measurement error in the approach and docking phases most likely would not be the same as for the rendezvous phase since there would be access to measurements from radio communications and visual navigation tools. Implementation of this would help reducing the bouncing effect near the end state of the simulation reducing the simulation Δv and t_f considerably.

Another source of error that could be considered is the control delay. The current assumption with the MPC model is that the optimal control $\boldsymbol{u}_{k|0}$ is solved for and instantaneously applied at time t_k. However, as the next chapter will show, solving the QP's is not instantaneous and neither would be the realized actuation. This delay in the control could have an effect on the robustness of the system. To combat this, control delay could be added in the QP's by adjusting the control horizon to start later than the state prediction horizon.

CHAPTER 6

DEMONSTRATION OF IMPLEMENTATION FOR FLIGHT SCENARIO

This chapter demonstrates the guidance algorithm as it would run on a small satellite flight-like hardware and is an expansion on the work done in [41]. To remain hardware agnostic, an emulation of an ARM processor is used to run the guidance algorithm. As most small satellite available flight computers contain ARM processors, this type was chosen as the generic processor to use for emulation. For example, the ClydeSpace Q7S [42], ATI [43], and ISIS [44] onboard computers contain ARM Cortex-A9 processors [45]. Another popular solution particularly for universities is to fly a BeagleBone, which in the current iteration of BeagleBone Black [46] has a processor with an ARM Cortex-A8 CPU [47].

The open-source machine virtualizer, QEMU [32], is used to create the target platform. If needed, this abstraction allows for easy swapping of hardware specifications while benchmarking performance. The emulation is set up to run in-the-loop providing the actuation commands to a dedicated computer running a 42 server simulation that handles the dynamics and sensor measurements. This contribution verifies that the developed algorithm can be run on its intended systems and is a valid methodology for use. As shown in Figure 6.1, a 42 simulation server (the dynamics) runs on the host machine, while the 42 client (the flight software) runs on the virtualized ARM machine. A socket connection passes necessary state and actuation effort between the server-client programs.

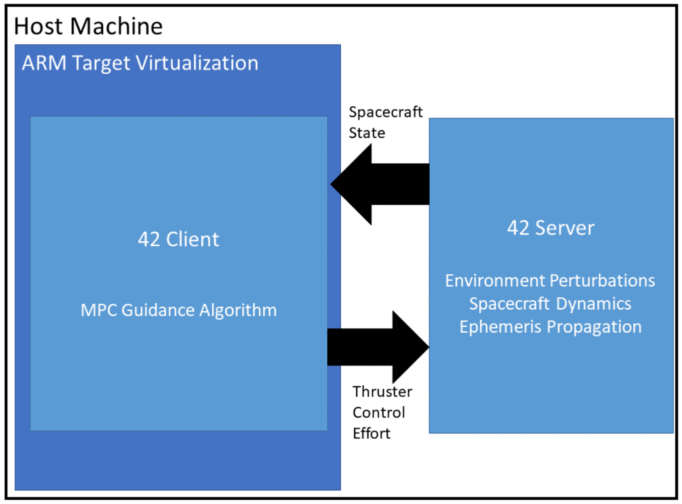


Figure 6.1: Diagram showing the interprocess communication between the environment simulation server and the flight software client running on a virtual machine.

The limitations of QEMU are that it may not be entirely cycle-accurate. However, it provides a preliminary evaluation of running the developed algorithm on an appropriate system.

6.1 Algorithm Benchmarking

The computational resources needed to solve an MPC problem is related to the problem size [13]. The problem size is affected by problem parameters such as the length of the prediction horizon N and the number of constraints. Since the purpose of the presented guidance algorithm is to run onboard a small spacecraft in real-time, the computation time must be much less than the MPC sample time. Otherwise, there is the chance that a control solution will not be prepared within enough time for predicted trajectory accuracy. In other words, a faster solution time creates less delay in the applied actuation effort. The purpose of this section is to then benchmark the optimal control problems in each of the three phases to gain insight into how the complexity affects the computation time.

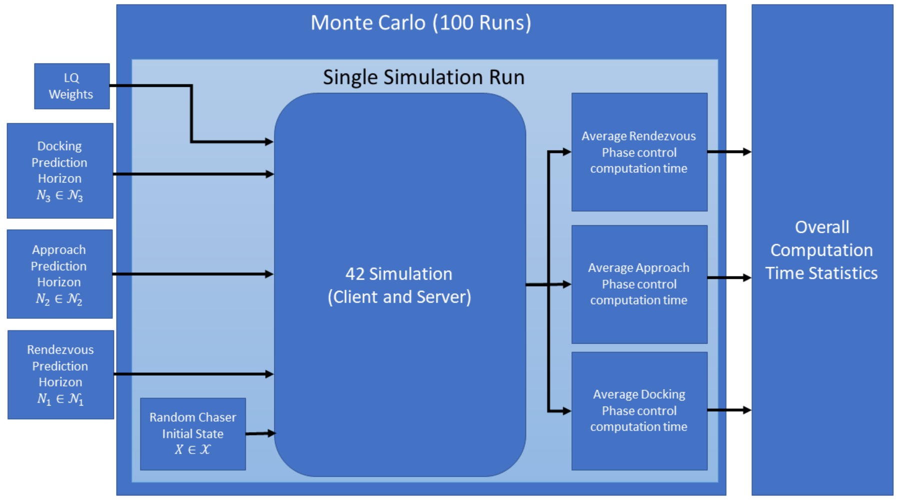


Figure 6.2: Algorithm computation time test plan.

Using the server-client simulation setup, a Monte Carlo analysis was performed with randomized chaser initial conditions. During a maneuver, the computation time of the CVXGEN optimizer calls at each time step was logged. After completion of the maneuver, the average solution computation time for each phase was calculated. The Overall computation time statistics for the phases were compiled from 100 simulation runs. The test plan, shown in Figure 6.2, involves sets of 100 simulations for varying values of prediction horizon lengths to investigate how this parameter affects the computation time. While it is not expected to increase complexity, the MPC control interval times were also a parameter of interest as a matter of sanity checking. From the resulting data, an inference into the efficacy of running the guidance algorithm on currently available hardware and the amount of control delay to expect is made.

6.1.1 Rendezvous Phase

For the rendezvous phase, the prediction horizon varies between $N_1 = 15, 30, 45$, while the approach and docking phase prediction horizons were kept constant at $N_2 = 15$ and $N_3 = 12$, respectively.

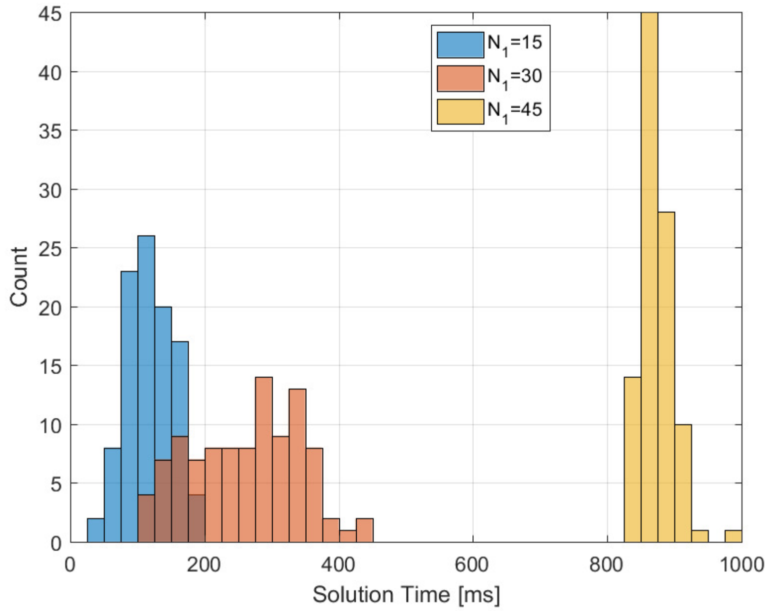


Figure 6.3: Solution time counts for Rendezvous phase with varying prediction lengths.

In Figure 6.3, the bin width of the rendezvous optimal control problem is 25 ms. As expected, the prediction horizon length is correlated to longer computation times. However, there is more spread and unpredictability with smaller prediction horizons, particularly the $N_1 = 30$ case. The average computation time for the rendezvous phase prediction horizons of 15, 30, and 45 was 119.18 ms, 270.40 ms, and 863.91 ms, respectively. It should be noted that for all prediction horizon cases the computation times were all less than a second. This gives ample time between control updates during the rendezvous phases for all investigated sample times. Another implication is that there would not be much actuation delay.

The rendezvous phase control intervals were tested to determine if there was an unexpected contribution to the problem complexity. A set of 100 scenarios were run for rendezvous control times of $T_s = 30, 45, 60$ s with a prediction length of $N_1 = 30$. Figure 6.4 confirms that the spread of computation times for each sample time is similar.

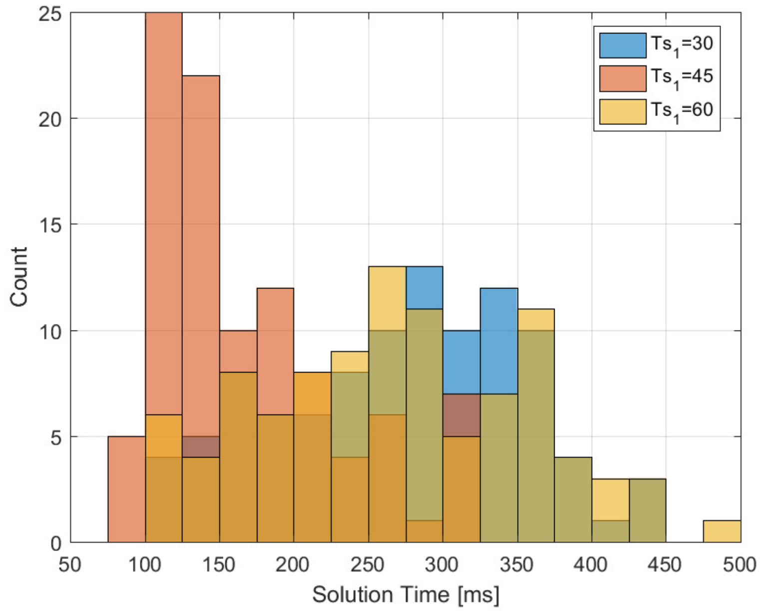


Figure 6.4: Solution time counts for Rendezvous phase with varying control intervals.

6.1.2 Approach Phase

The prediction horizon values tested for the approach phase were $N_2 = 5, 10, 15$. For this analysis, the initial conditions were taken as the end points for a rendezvous phase with prediction length of $N_1 = 30$ and $T_{s,1} = 30$ s. As with the rendezvous phase, the distribution of computation times over 100 samplings for each prediction horizon length are shown in Figure 6.5.

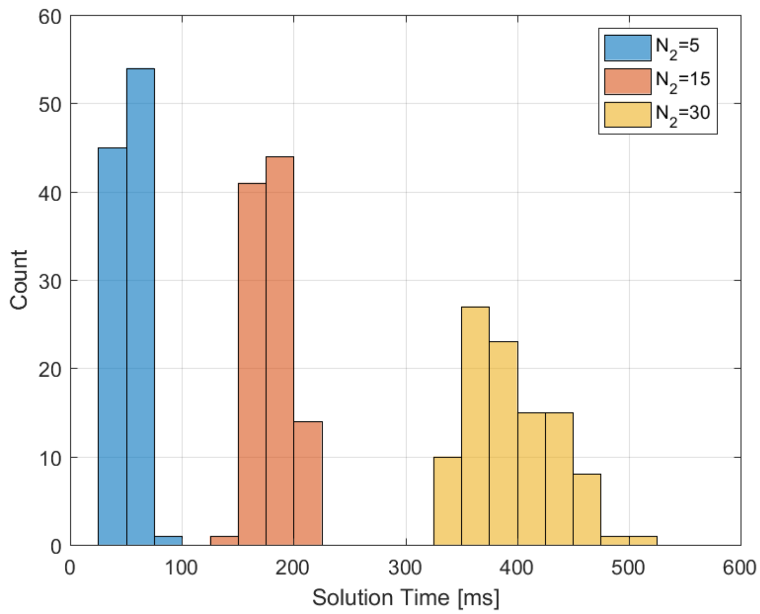


Figure 6.5: Solution time counts for Approach phase with varying prediction lengths.

The results of the approach analysis have less range and spread than for the rendezvous phase. The general trend of increased computation time with increasing prediction horizon is still prominent. For $N_2 = 5$, the average computation time was 53.40 ms, increasing to 181.55 ms for $N_2 = 15$ and finally to 394.89 ms for the $N_2 = 30$. Recalling back to the MPC parameter study in Chapter 4, there was not much solution change between $N_2 = 15$ and $N_2 = 30$ and yet the computation time is almost double. This reinforces the idea that such a high prediction length is inefficient or this phase.

As with the rendezvous phase, the test of sampling times on the complexity was tested for interval times of $T_{s,2} = 10, 20, 30$ s. As expected, this phase also shows the same spread and similar distribution for all control interval times. It can be concluded that this sanity check has confirmed that sample time has no meaningful affect on the problem complexity and therefore computation time.

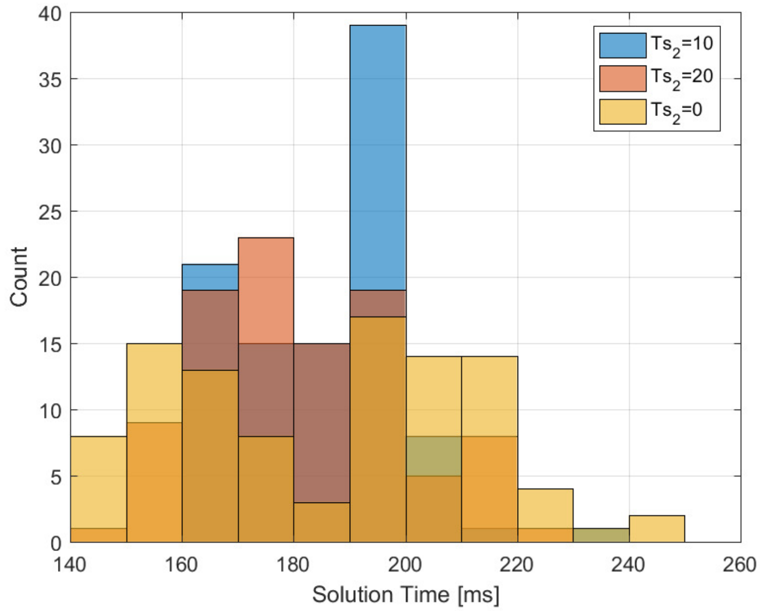


Figure 6.6: Solution time counts for Approach phase with varying control intervals.

6.2 Docking Phase

Lastly, the same analysis was performed for the docking phase. The prediction lengths used here were $N_3 = 5, 12, 30$. A prediction horizon of 12 was selected instead of 15, because of the default choice of a control time of 5 s. With 5 s the docking prediction would be looking then an exact minute ahead which was an arbitrary choice. The results are expected to be very close to a prediction horizon of 15. The resulting distributions to prediction horizon length are seen in Figure 6.7. The average computation times were calculated to be 9.1 ms, 30.1 ms, and 98.8 ms for increasing prediction horizon length. These values are all relatively fast compared to the previous two phases. Recalling that the sensitivity results were also not affected much by the prediction horizon length, it is suggested that a value is chosen to meet mission needs and constraints.

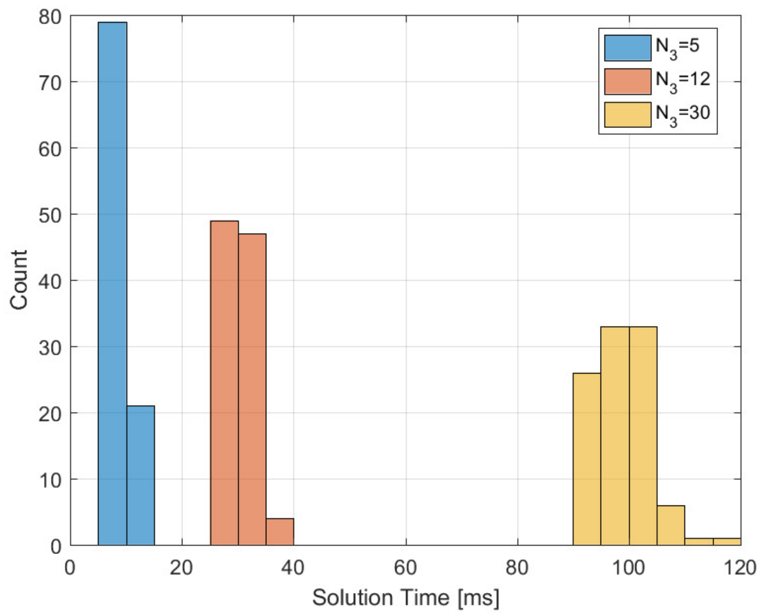


Figure 6.7: Solution time counts for Docking phase with varying prediction lengths.

6.3 Future Considerations

Future considerations consist of finding improved emulation techniques that are cycle-accurate to confirm the findings of this report [48, 49]. Additionally, a select number of real hardware could be obtained to confirm the computation time accuracy.

CHAPTER 7
CONCLUSION

In summary, simulation scenario running an MPC guidance algorithm for the AR&D of small satellites was created. The simulation was designed with the 42 spacecraft simulator that includes disturbances from J2 and atmospheric drag. The problem scenario investigated involved two 6U CubeSats with 6DOF propulsion capabilities.

First, a guidance algorithm that splits the maneuver into three MPC phases based on distance to the target was developed. The optimal control solution within each phase was designed to be solved quickly by utilizing convex QP formulations. The guidance algorithm was shown to successfully perform an AR&D maneuver by approaching a target with docking ports in the V-bar and R-bar directions as well as for a target with a constant rotation.

The sensitivity of the algorithm to initial chaser position relative to the target was investigated. It was found that radial distance has an appreciable effect on the Δv and t_f values as expected. Additionally, the effect of the in-plane quadrant of the chaser position was shown to have an effect on the expected minimum Δv and t_f.

Then, an analysis of the robustness of the algorithm with disturbances attributed to actuation and measurement errors was discussed. For actuation errors, a SDE model was used to provide time-varying noise characteristics in the propulsion system. The measurement errors followed a normal zero-mean distribution added onto the relative position of the chaser to the target. The average deviation in Δv and final time characteristics for 100 randomly generated chaser relative initial states were calculated over 60 runs. Using measurement errors with position standard deviation $\sigma_\rho = 5$ cm and velocity standard deviation $\sigma_{\dot{\rho}} = 1$ mm/s, the average deviation in Δv was 4.25 m/s and 19.54 min in t_f.

The last contribution consisted of using hardware emulation with QEMU to analyze the computation time needed to solve the optimal control problems in the rendezvous and approach phases. For the rendezvous phase, increasing the prediction horizon length from 30 to 45 had a very noticeable effect on the computation time, more than doubling the effort. The increased computation time for longer prediction horizons within the approach and docking phases were also observed but with less spread overall compared to the rendezvous tests. In both phases, the computation time remained under a second, which is much less than the tested sampling times, validating that this algorithm can be run onboard a real flight system.

Therefore, it has been shown that the use of MPC for the AR&D of small satellites is a viable strategy and should be considered for use on flight. Limitations of this work exist in that the shown examples are accurate locally for circular orbits in LEO. The work can applied to elliptical cases by changing the LTI models to LTV. Improvements can also be made for robustness through the use of techniques such as tube-based MPC.

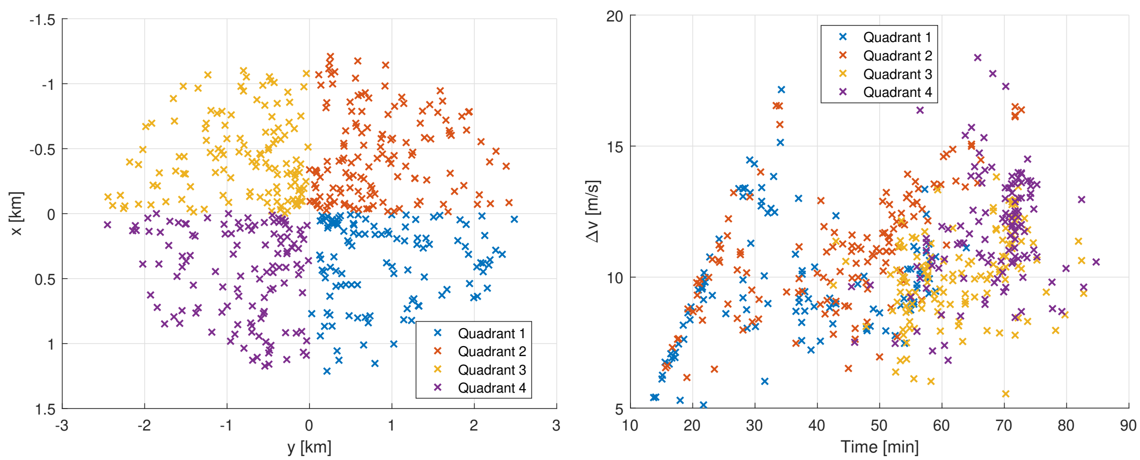
Appendices

APPENDIX A

EXTRA SENSITIVITY PARAMETER RESULTS

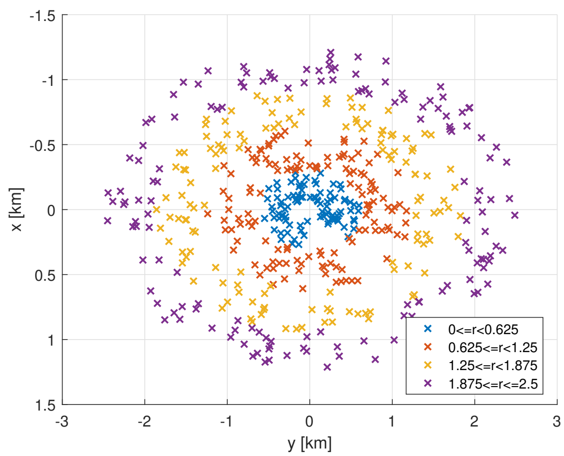
The sensitivity to MPC parameters was shown in Chapter 4 for the rendezvous phase QP formulation. Here similar results for a rendezvous QP without a reference point (in effect the origin) and a LP formulation are shown.

A.1 QP with no Reference Point

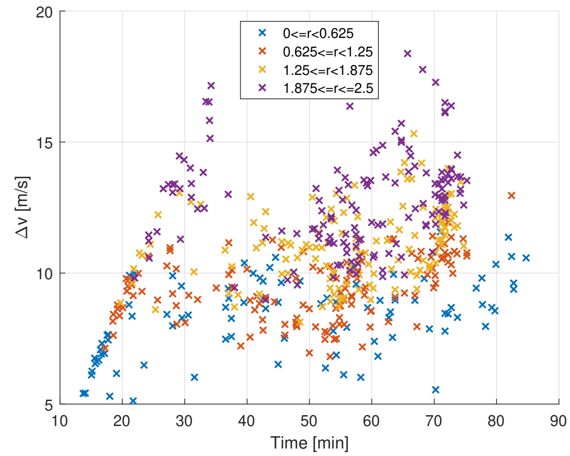


(a) (b)

Figure A.1: Quadrant colored (a) initial chaser positions and (b) resulting Δv vs t_f.



(a)



(b)

Figure A.2: Radially colored (a) initial chaser positions and (b) resulting Δv vs t_f.

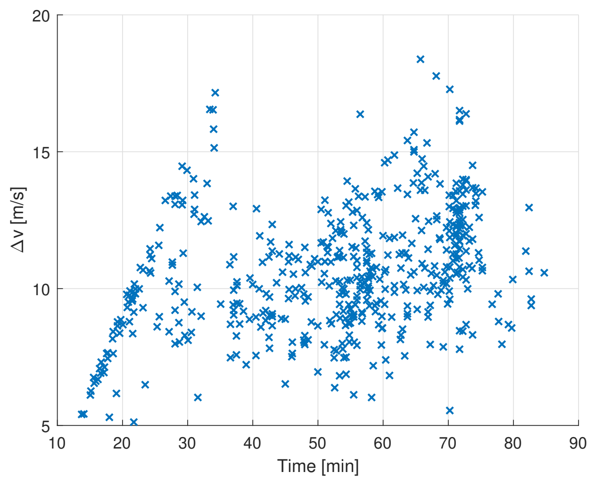
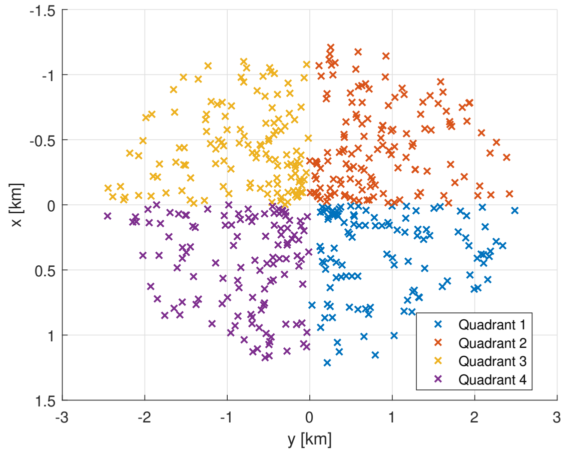
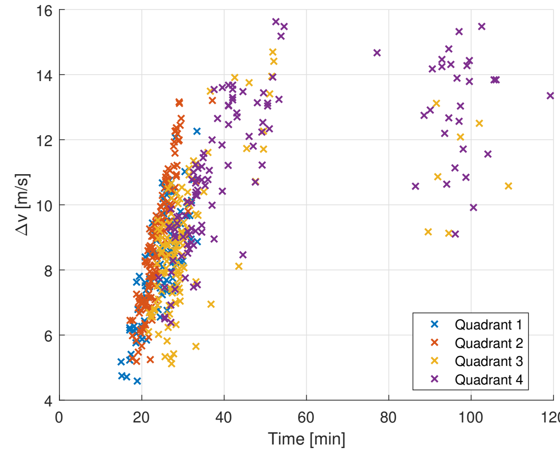


Figure A.3: Δv vs t_f for Maneuver with LP Rendezvous Problem.

A.2 LP Rendezvous

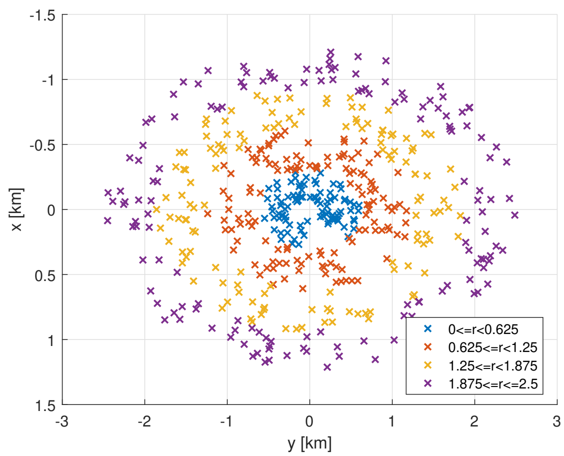


(a)

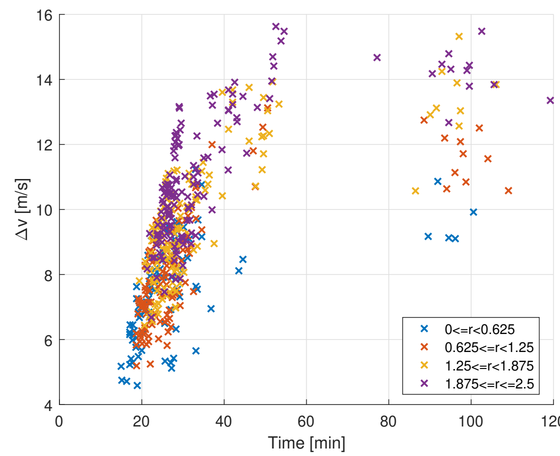


(b)

Figure A.4: Quadrant colored (a) initial chaser positions and (b) resulting Δv vs t_f.



(a)



(b)

Figure A.5: Radially colored (a) initial chaser positions and (b) resulting Δv vs t_f.

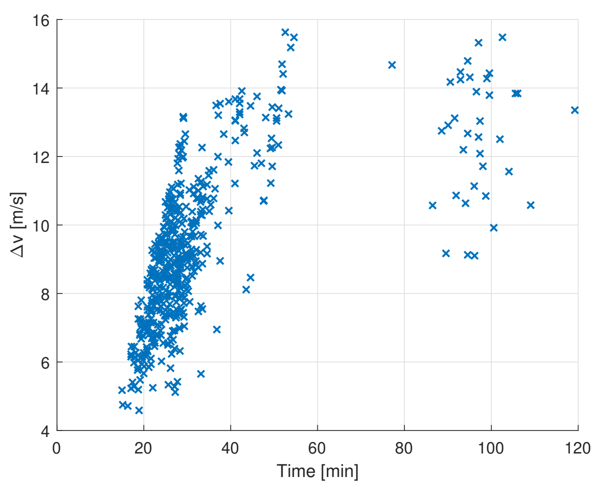


Figure A.6: Δv vs t_f for Maneuver with LP Rendezvous Problem.

REFERENCES

[1] M. E. Polites, “Technology of Automated Rendezvous and Capture in Space,” *Journal of Spacecraft and Rockets*, vol. 36, no. 2, pp. 280–291, Mar. 1999.

[2] D. C. Woffinden and D. K. Geller, “Navigating the Road to Autonomous Orbital Rendezvous,” *Journal of Spacecraft and Rockets*, vol. 44, no. 4, pp. 898–909, Jul. 2007.

[3] W. Fehse, *Automated rendezvous and docking of spacecraft*. Cambridge; New York: Cambridge University Press, 2003.

[4] “2020 NASA Technology Taxonomy.” (May 2021), [Online]. Available: https://www.nasa.gov/offices/oct/taxonomy/index.html.

[5] J. A. Starek, B. Açıkmeşe, I. A. Nesnas, and M. Pavone, “Spacecraft Autonomy Challenges for Next-Generation Space Missions,” in *Advances in Control System Technology for Aerospace Applications*, E. Feron, Ed., vol. 460, Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 1–48.

[6] R. Chai, A. Tsourdos, A. Savvaris, S. Chai, Y. Xia, and C. L. Philip Chen, “Review of advanced guidance and control algorithms for space/aerospace vehicles,” *Progress in Aerospace Sciences*, vol. 122, Apr. 2021.

[7] C. Pirat, P. Ribes-Pleguezuelo, F. Keller, A. Zuccaro Marchi, and R. Walker, “Toward the Autonomous Assembly of Large Telescopes Using CubeSat Rendezvous and Docking,” *Journal of Spacecraft and Rockets*, vol. 59, no. 2, pp. 375–388, Mar. 2022. [Online]. Available: https://arc.aiaa.org/doi/10.2514/1.A34945.

[8] J. L. Goodman, “History of Space Shuttle Rendezvous and Proximity Operations,” *Journal of Spacecraft and Rockets*, vol. 43, no. 5, pp. 944–959, Sep. 2006.

[9] H. Liu, J. Li, and B. Hexi, “Sliding mode control for low-thrust Earth-orbiting spacecraft formation maneuvering,” *Aerospace Science and Technology*, vol. 10, no. 7, pp. 636–643, Oct. 2006.

[10] E. Capello, E. Punta, F. Dabbene, G. Guglieri, and R. Tempo, “Sliding-Mode Control Strategies for Rendezvous and Docking Maneuvers,” *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 6, pp. 1481–1487, Jun. 2017.

[11] R. Zappulla, H. Park, J. Virgili-Llop, and M. Romano, “Real-Time Autonomous Spacecraft Proximity Maneuvers and Docking Using an Adaptive Artificial Potential Field Approach,” *IEEE Transactions on Control Systems Technology*, vol. 27, no. 6, pp. 2598–2605, Nov. 2019.

[12] W. Sun, Y. Pan, J. Lim, E. A. Theodorou, and P. Tsiotras, "Min-Max Differential Dynamic Programming: Continuous and Discrete Time Formulations," *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 12, pp. 2568–2580, 2018.

[13] U. Eren, A. Prach, B. B. Koçer, S. V. Raković, E. Kayacan, and B. Açıkmeşe, "Model Predictive Control in Aerospace Systems: Current State and Opportunities," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 7, pp. 1541–1566, Jul. 2017.

[14] A. Weiss, M. Baldwin, R. S. Erwin, and I. Kolmanovsky, "Model Predictive Control for Spacecraft Rendezvous and Docking: Strategies for Handling Constraints and Case Studies," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 4, pp. 1638–1647, Jul. 2015.

[15] E. N. Hartley, "A tutorial on model predictive control for spacecraft rendezvous," in *2015 European Control Conference (ECC)*, Jul. 2015, pp. 1355–1361.

[16] Q. Li, J. Yuan, B. Zhang, and C. Gao, "Model predictive control for autonomous rendezvous and docking with a tumbling target," *Aerospace Science and Technology*, vol. 69, pp. 700–711, Oct. 2017.

[17] B. P. Malladi, S. Di Cairano, and A. Weiss, "Nonlinear Model Predictive Control of Coupled Rotational-Translational Spacecraft Relative Motion," in *2019 American Control Conference (ACC)*, ISSN: 2378-5861, Jul. 2019, pp. 3581–3586.

[18] A. Richards and J. How, "Performance Evaluation of Rendezvous Using Model Predictive Control," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Austin, Texas: American Institute of Aeronautics and Astronautics, Aug. 2003, ISBN: 978-1-62410-090-1.

[19] A. Richards and J. P. How, "Robust variable horizon model predictive control for vehicle maneuvering," *International Journal of Robust and Nonlinear Control*, vol. 16, no. 7, pp. 333–351, 2006.

[20] M. Leomanni, R. Quartullo, G. Bianchini, A. Garulli, and A. Giannitrapani, "Variable-Horizon Guidance for Autonomous Rendezvous and Docking to a Tumbling Target," *Journal of Guidance, Control, and Dynamics*, vol. 45, no. 5, pp. 846–858, May 2022.

[21] P. Li and Z. H. Zhu, "Model predictive control for spacecraft rendezvous in elliptical orbit," *Acta Astronautica*, vol. 146, pp. 339–348, May 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0094576517309426.

[22] S. Miller, M. L. R. Walker, J. Agolli, and J. Dankanich, "Survey and Performance Evaluation of Small-Satellite Propulsion Technologies," *Journal of Spacecraft and Rockets*, vol. 58, no. 1, pp. 222–231, Jan. 2021.

[23] W. H. Clohessy and R. S. Wiltshire, "Terminal Guidance System for Satellite Rendezvous," *Journal of the Aerospace Sciences*, vol. 27, no. 9, pp. 653–658, Sep. 1960.

[24] G. Inalhan, M. Tillerson, and J. P. How, "Relative Dynamics and Control of Spacecraft Formations in Eccentric Orbits," *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 1, pp. 48–59, Jan. 2002. [Online]. Available: https://arc.aiaa.org/doi/10.2514/2.4874.

[25] K. Yamanaka and F. Ankersen, "New State Transition Matrix for Relative Motion on an Arbitrary Elliptical Orbit," *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 1, pp. 60–66, Jan. 2002. [Online]. Available: https://arc.aiaa.org/doi/10.2514/2.4875.

[26] Z. Dang, "Solutions of Tschauner–Hempel Equations," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 11, pp. 2956–2960, Nov. 2017. [Online]. Available: https://arc.aiaa.org/doi/10.2514/1.G002774.

[27] T. McKechnie *et al.*, "Green Monopropellant 100mN Thruster," in *AIAA Propulsion and Energy 2021 Forum*, VIRTUAL EVENT: American Institute of Aeronautics and Astronautics, Aug. 2021, ISBN: 978-1-62410-611-8.

[28] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model predictive control: theory, computation, and design*, 2nd edition. Madison, Wisconsin: Nob Hill Publishing, 2017.

[29] B. Kouvaritakis and M. Cannon, *Model Predictive Control*, ser. Advanced Textbooks in Control and Signal Processing. Cham: Springer International Publishing, 2016.

[30] E. T. Stoneking. "42 - Spacecraft Simulation." (Dec. 2021), [Online]. Available: https://github.com/ericstoneking/42 (visited on 01/25/2022).

[31] J. Mattingley and S. Boyd, "CVXGEN: A code generator for embedded convex optimization," *Optimization and Engineering*, vol. 13, no. 1, Mar. 2012.

[32] *QEMU*, Sep. 2022. [Online]. Available: https://www.qemu.org/.

[33] A. Fear and E. G. Lightsey, "Implementation of Small Satellite Autonomous Rendezvous Using Model Predictive Control," in *AIAA SCITECH 2022 Forum*, San Diego, CA & Virtual: American Institute of Aeronautics and Astronautics, Jan. 2022, ISBN: 978-1-62410-631-6.

[34] ——, "Robustness and Sensitivity Analysis of a CubeSat Autonomous Rendezvous and Docking MPC Algorithm," in *AIAA SCITECH 2023 Forum*, National Harbor, MD & Online: American Institute of Aeronautics and Astronautics, Jan. 2023, ISBN: 978-1-62410-699-6.

[35] C. Zagaris, H. Park, J. Virgili-Llop, R. Zappulla, M. Romano, and I. Kolmanovsky, “Model Predictive Control of Spacecraft Relative Motion with Convexified Keep-Out-Zone Constraints,” *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 9, pp. 2054–2062, Sep. 2018. [Online]. Available: https://arc.aiaa.org/doi/10.2514/1.G003549.

[36] M. Mammarella, E. Capello, and G. Guglieri, “Robust Model Predictive Control for Automated Rendezvous Maneuvers in Near-Earth and Moon Proximity,” in *2018 AIAA SPACE and Astronautics Forum and Exposition*, ser. AIAA SPACE Forum, American Institute of Aeronautics and Astronautics, Sep. 2018.

[37] S. A. Schweighart and R. J. Sedwick, “High-Fidelity Linearized J Model for Satellite Formation Flight,” *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 6, pp. 1073–1080, Nov. 2002, Publisher: American Institute of Aeronautics and Astronautics.

[38] M. Mammarella, E. Capello, M. Lorenzen, F. Dabbene, and F. Allgower, “A general sampling-based SMPC approach to spacecraft proximity operations,” in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, Dec. 2017, pp. 4521–4526.

[39] K. Dong, J. Luo, Z. Dang, and L. Wei, “Tube-based robust output feedback model predictive control for autonomous rendezvous and docking with a tumbling target,” *Advances in Space Research*, vol. 65, no. 4, pp. 1158–1181, Feb. 2020.

[40] R. Carpenter and T. Lee, “Stable Clock error model using first and second order gauss markov processes,” 2008.

[41] A. Fear and E. G. Lightsey, “Performance of a Model Predictive Control Based Autonomous Rendezvous and Docking Algorithm for CubeSats using Hardware Emulation,” in *45th Rocky Mountain AAS GN&C Conference 2023*, Breckenridge, CO: American Astronautical Society, Feb. 2023.

[42] *Q7S - Satellite Command & Data Handling — AAC SpaceQuest*. [Online]. Available: https://www.aac-clyde.space/what-we-do/space-products-components/command-data-handling/q7s.

[43] *Small Satellite On-board Computer (OBC) — satsearch*. [Online]. Available: https://satsearch.co/products/ati-space-small-satellite-on-board-computer-obc.

[44] *ISIS On Board Computer — satsearch*. [Online]. Available: https://satsearch.co/products/isis-isis-on-board-computer.

[45] *Cortex-A9*. [Online]. Available: https://developer.arm.com/Processors/Cortex-A9.

[46] *BeagleBone Black — BeagleBoard Documentation*. [Online]. Available: https://docs.beagleboard.org/latest/boards/beaglebone/black/.

[47] *AM3358 data sheet, product information and support — TI.com*. [Online]. Available: https://www.ti.com/product/AM3358.

[48] Y. Luo, Y. Li, X. Yuan, and R. Yin, "QSim: Framework for Cycle-accurate Simulation on Out-of-Order Processors based on QEMU," in *2012 Second International Conference on Instrumentation, Measurement, Computer, Communication and Control*, Dec. 2012, pp. 1010–1015.

[49] T.-C. Yeh, G.-F. Tseng, and M.-C. Chiang, "A fast cycle-accurate instruction set simulator based on QEMU and SystemC for SoC development," in *Melecon 2010 - 2010 15th IEEE Mediterranean Electrotechnical Conference*, ISSN: 2158-8481, Apr. 2010, pp. 1033–1038.