3-2022

Coupled Orbit-Attitude Dynamics and Control of a CubeSat Equipped with a Robotic Manipulator

Charles M. Carr

COUPLED ORBIT-ATTITUDE DYNAMICS AND CONTROL OF A CUBESAT EQUIPPED WITH A ROBOTIC MANIPULATOR

THESIS

Charles M. Carr, Captain, USAF

AFIT-ENY-MS-22-M-283

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENY-MS-22-M-283

COUPLED ORBIT-ATTITUDE DYNAMICS AND CONTROL OF A CUBESAT EQUIPPED WITH A ROBOTIC MANIPULATOR

THESIS

Presented to the Faculty

Department of Aeronautics and Astronautics

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Astronautical Engineering

Charles M. Carr, B.A.

Captain, USAF

March 24, 2022

AFIT-ENY-MS-22-M-283

COUPLED ORBIT-ATTITUDE DYNAMICS AND CONTROL OF A CUBESAT EQUIPPED WITH A ROBOTIC MANIPULATOR

THESIS

Charles M. Carr, B.A.
Captain, USAF

Committee Membership:

Maj Costantinos Zagaris, Ph.D
Chair

Lt Col Bryan D. Little, Ph.D
Member

Maj Robert A. Bettinger, Ph.D
Member

Abstract

This research investigates the utility and expected performance of a robotic servicing CubeSat. The coupled orbit-attitude dynamics of a 6U CubeSat equipped with a four-link serial manipulator are derived. A proportional-integral-derivative controller is implemented to guide the robot through a series of orbital scenarios, including rendezvous and docking following ejection from a chief spacecraft, repositioning the end effector to a desired location, and tracing a desired path with the end effector. Various techniques involving path planning and inverse differential kinematics are leveraged. Simulation results are presented and performance metrics such as settling time, state errors, control use, and system robustness are analyzed. Recommended improvements to the system design and the model are discussed.

Acknowledgements

I would like to thank my advisor, Major Costantinos Zagaris, for introducing me to this topic, encouraging me to pursue it despite the steep learning curve, and guiding me throughout the process. This research would not have been possible without him. I would also like to thank my committee members, whose support enabled me to complete this thesis.

Each of my AFIT professors equipped me with the tools I needed to complete this research. I am grateful for their instruction and the privilege of carrying these tools with me after graduation.

I am indebted to the professors and teachers throughout my life, who gave me a passion for science that brought me to this point. I thank them for opening my mind to the universe.

Finally, I thank my families, both given and chosen, for supporting me throughout this program and giving me everything else that I needed to succeed.

Charles M. Carr

Table of Contents

Page

List of Figures

List of Tables

COUPLED ORBIT-ATTITUDE DYNAMICS AND CONTROL OF A CUBESAT EQUIPPED WITH A ROBOTIC MANIPULATOR

I. Introduction

1.1 Motivation

The proliferation of low Earth orbit (LEO) and the expansion of national space architectures into the cis-lunar domain necessitates the advancement of space robotics for a variety of missions [1, 2]. Robotic systems with varying degrees of autonomy are necessary for in-space manufacturing and assembly, on-orbit servicing, and maintenance of human exploration spacecraft, which are critical mission areas in the creation of a permanent presence in space, on the Moon, and eventually on Mars. In recent years NASA has matured automation and robotics technologies in support of these mission areas [3, 4, 5, 6].

A particular concept of interest is the use of small robotic spacecraft for on-orbit servicing missions. Small satellites, or smallsats, typically defined as those with less than 600 kg of mass, are generally faster and cheaper to build as they can increasingly leverage commercial-off-the-shelf (COTS) components. Launch opportunities for smallsats have also increased with numerous commercial start ups entering the market in recent years. Smallsats can therefore be proliferated in orbit and replaced frequently as technology improves.

Smallsat capabilities have continued to improve in recent years with countless COTS options for both fully integrated buses and for each subsystem [7]. The use of smallsats has also increased dramatically over the last decade. Nearly 40% of all

smallsats launched between 2010 and 2020 were launched in 2020, where 94% of all spacecraft launched into orbit were smallsats, accounting for 43% of the total upmass [8]. While the surge in smallsats is largely attributable to commercial broadband constellations in LEO, smallsats are used for a variety of missions including remote sensing, science, and technology demonstration.

Very large satellites, or space stations, could benefit significantly from small robotic servicing spacecraft as routine maintenance is typically necessary, especially if these systems could alleviate the burden on astronauts having to perform risky space-walks. Small free-flying spacecraft could offer significant power savings, increased mobility, and could potentially access more regions of the space station compared to existing robotic systems on the International Space Station (ISS). For example, the Space Station Remote Manipulator System (Canadarm2) is 17 m in length, has a mass of 1,497 kg, and must be attached to a power and data grappling fixture on the exterior of the Station [9]. Similarly, the Special Purpose Dexterous Manipulator (SPDM), also known as Dextre, draws an average power of 600 W with a peak of 2,000 W, and has a stopping distance of nearly 6 inches [10]. Therefore, a small free-flying satellite may be better suited to perform delicate maintenance tasks.

CubeSats are a particular subset of smallsats that have gained popularity over the last several years. With their standardized form factors, CubeSats can be designed, built, and launched very quickly and can leverage a wealth of COTS subsystems. Historically, CubeSats have been mostly used for academic purposes due to their relative simplicity and low cost. However, enabled by the miniaturization of critical space technologies, current and future CubeSat missions demonstrate their increasing capability and utility.

NASA, as part of the Artemis program, plans to launch two CubeSats into lunar orbit ahead of the Lunar Gateway module. The Lunar Flashlight mission will search

for ice deposits at the Moon's south pole with a pulsed laser sensor on a 6U CubeSat [11]. The Cislunar Autonomous Positioning System Technology Operations and Navigation Experiment, or CAPSTONE, will measure the amount of fuel needed for station-keeping in the intended orbit of the Lunar Gateway using a 12U CubeSat [12]. Both missions will leverage advancements in CubeSat propulsion systems. Research conducted by the United States Naval Academy aims to demonstrate a 3U robotic servicing CubeSat in the interior of the ISS [13, 14]. Other research efforts investigate the use of CubeSats for asteroid sampling missions [15]. These missions demonstrate the increasing capabilities of CubeSats.

In light of the need for autonomous robotic servicing technology and the potential benefits of using smallsats for this mission, the question remains as to how small a useful robotic servicing satellite can be. Can a CubeSat be used for robotic servicing? NASA's 2020 Small Spacecraft Technology State of the Art report identifies mission modelling and simulation as a burgeoning area in the field of small satellite research that requires further development [7]. This research models a notional robotic servicing CubeSat and simulates the system in a series of mission scenarios to determine its feasibility.

1.2 Problem Statement

This research aims to determine the feasibility of a robotic servicing CubeSat by analyzing the expected performance of a notional system. In order to complete the analysis, high fidelity, coupled orbit-attitude dynamics of the system are derived, and the nonlinear relative orbital dynamics with respect to a chief spacecraft are modelled. A controller is implemented to regulate errors of the nonlinear system.

The main challenge faced by a CubeSat with a robotic manipulator is the high degree of coupling between the manipulator and base motion. Additionally, CubeSats

have relatively small propulsive capabilities. To address these challenges, this research focuses on the following questions.

Given the state of the art in CubeSat propulsion and attitude determination and control (ADACS) subsystems,

1. can a CubeSat successfully rendezvous and dock with a client spacecraft using standard control methods?

2. can a CubeSat robot stabilize its base as the manipulator performs various maneuvers?

3. can a CubeSat robot perform various tasks with its manipulator within acceptable levels of error?

In modelling, simulating, and analyzing a notional system this research answers the question of whether or not a robotic servicing CubeSat is feasible, and estimates the performance of such a system. This research can also advise the mechanical design of a future CubeSat robot as well as its trajectory planning and control strategies.

This work presents a significant contribution as many studies related to rendezvous and proximity operations (RPO) use linear equations of motion [16, 17], or rely on linearization techniques for control [18, 19]. Many studies also treat the orbital reference frame as inertial, since the duration of engagement scenarios tends to be short relative to orbital periods, and the robot typically operates in very close proximity to the serviced satellite [18, 20]. Several studies dealing with the trajectory planning and control of space robotics systems focus on novel kinematic control methods and thus ignore system dynamics altogether [13, 21]. The treatment of gravity in the dynamics of multibody systems is inconsistent and often not thoroughly addressed. This work addresses gravity directly through Kane's Method by applying a Newtonian gravitational force to each rigid body in the system. Finally, few studies that derive

the dynamics of free-flying satellites with robotic manipulators focus on CubeSats [19, 22], where the high degree of coupling between the manipulator and base motion presents a unique challenge. This work applies well known methods of control to demonstrate the feasibility of a novel system: a CubeSat with a robotic manipulator.

1.2.1 Research Scope and Assumptions

The notional system used in this research is a 6U CubeSat with a four-link serial manipulator. Justification for this choice and for other characteristics of the system is provided in Chapter III. The methods used in this research can be applied to alternative system designs, but the results are specific to this system.

It is assumed that the propulsive capabilities of CubeSats limit their utility to close proximity operations. In other words, CubeSats cannot rapidly perform large orbital maneuvers to rendezvous with an arbitrary client spacecraft. Therefore the mission scenarios simulated assume that the CubeSat robot begins in close proximity to the client, eliminating the need for large translational maneuvers. However, the Keplerian orbital motion of both the client and the CubeSat robot is fully incorporated in all simulations. Orbital perturbations are ignored.

This research is limited to the case of servicing a cooperative target. Some robotics missions, such as grappling a tumbling debris object or others involving large loads on the manipulator, are clearly not suited to a CubeSat system. Therefore the client in all scenarios is assumed to be three axis stabilized with no propulsion.

This research is limited to servicing spacecraft in Earth orbit. All mission scenarios involve a client spacecraft in LEO, specifically the orbit of the ISS. Alternative Earth orbits can be simulated, including highly eccentric orbits and those in different orbital regimes. To investigate servicing missions in orbits much higher than GEO would require a higher fidelity gravity model that incorporates the Moon.

The dynamics of the CubeSat robot ignores contact forces on the manipulator. The model only incorporates the coupled orbit-attitude motion of the CubeSat in the free-flying case. Contact forces are an essential part of the dynamics in an on-orbit servicing mission, and understanding the magnitudes of forces and torques capable of being imparted by the manipulator is necessary to determine the types of servicing missions the system is suited for. However, the ability to control motion in the free-flying case is prerequisite to performing a servicing mission with physical contact, and therefore this research is a necessary step in determining the feasibility of the system.

Finally, state estimation is not incorporated in the model and simulations. The system has perfect knowledge of both its own state and that of the client. Any spacecraft requires state estimation to perform its mission, especially those involving RPO. Estimation is even more crucial for a robotics mission as trajectory planning for the manipulator depends heavily on its configuration. The accuracy and precision of the manipulator motion is likely determined by that of the state measurements. Therefore estimation and sensors should be incorporated in future high-fidelity system models. This research focuses solely on the system dynamics, motion planning, and well-established methods of control. However, the robustness of the controller is investigated by injecting random disturbance forces into the feedback control loop and observing the system response. The incorporation of contact forces to the CubeSat robot and state estimation are necessary future improvements to the model that are discussed, along with other recommended follow-on research efforts, in Chapter V.

1.3 Organization of the Thesis

This research investigates the performance of a 6U CubeSat robot through a series of representative orbital scenarios to determine the feasibility of such a system. Chapter II presents, in greater detail, past and current robotics missions in space,

along with present research in the field. It also introduces a requisite level of technical information to understand the research methodology.

Chapter III details the CubeSat robot system used in the model, and outlines the process of deriving the system dynamics. The simulation framework is thoroughly described, including the dynamics, calculation of the relative states and state errors, and the controller. Each scenario simulated is explained in detail, including the initial states and the scenario objectives. The full simulation campaign is also explained. Performance metrics used for each scenario are listed.

Chapter IV presents the results of the simulations and the analysis of the data collected. The performance metrics and other relevant summarizing statistics are presented and discussed. The effects of both the system design and the scenario assumptions on the results are explored. Additional results are provided in Appendix A. The thesis concludes with chapter V which summarizes the research conducted and provides recommendations for improving the system design and the model. Beneficial follow-on research efforts are also discussed.

II. Background and Literature Review

This chapter begins with a summary of past and present robotics missions in space to understand the conventions in the field and the state of the art. Current research in the field as published in various conference proceedings and journal articles is then discussed to determine the contributions of this research to the body of knowledge. Specialized topics involving multibody kinematics, dynamics, and trajectory planning are presented to prime the reader for the research methodology in Chapter III.

2.1 Past and Present Missions

Several robotic systems have already been demonstrated on orbit. The Japanese Space Agency launched the Engineering Test Satellite VII (ETS-7) in 1997 and successfully demonstrated an autonomous rendezvous and docking. ETS-7 also performed several other tests that validated theoretical techniques for robot trajectory planning [23]. The Shuttle Remote Manipulator System (SRMS), also known as Canadarm, operated in space for over 25 years and was instrumental in the capture and repair of numerous satellites, remote inspection of the Shuttle prior to re-entry, and construction of the ISS in cooperation with its successor, Canadarm2 [24]. The latter is still used on the ISS today for routine maintenance and capturing of unpiloted cargo ships. The SPDM is another ISS platform that performs maintenance missions on the exterior of the Station that require dexterity and precision. The SPDM has 15 degrees of freedom (DOF) and four interchangeable servicing tools [10]. It can work in tandem with the Canadarm2 to reach different areas of the ISS, and is part of NASA's ongoing Robotic Refueling Mission (RRM), designed to mature technologies needed for on-orbit refueling of satellites [3]. Both the Canadarm2 and Dextre are teleoperated, either by astronauts inside the ISS or by operators on the ground.

The interior of the ISS has also served as a valuable testbed for free-flying robotics technology. The Synchronized Position Hold Engage and Reorient Experimental Satellites (SPHERES) are a series of miniature satellites that operate in the interior of the ISS and are used to test metrology, formation flying, rendezvous and docking, and autonomy algorithms [25]. Astrobee, the successor platform to the SPHERES with the same mission, comes equipped with electric fans for maneuvering and perching arms for grasping handrails to conserve energy while stationary. The Astrobee platform will be used to test novel concepts such as automated assembly in the presence of obstacles and uncertainties [19], and thrust-free maneuvering through robotic spacecraft "hopping" [22].

In a 2007 mission established by the Defense Advanced Research Projects Agency (DARPA) named Orbital Express, a spacecraft equipped with a six DOF robotic arm demonstrated autonomous propellant transfer between spacecraft, replacement of a battery module and backup computer, and rendezvous in LEO [26].

Many more missions involving space robotics are in the planning and development stages. NASA's On-Orbit Servicing, Assembly, and Manufacturing (OSAM) 1 satellite, previously known as Restore-L, plans to rendezvous, grasp, and refuel Landsat 7, thereby extending its mission life. The OSAM-1 satellite will also host the Space Infrastructure Dexterous Robot (SPIDER), which will assemble a functional, three meter communications antenna, as well as a 10 meter lightweight beam to demonstrate the ability to assemble large structures in orbit [4, 6]. The OSAM-2 mission will take this concept one step further by assembling two 10-meter beams that will unfurl functional solar panels [27].

2.2 Present Research

Despite the progress made over the past several decades in space robotics, the kinematics, dynamics, and guidance, navigation, and control (GNC) of free-flying satellites with robotic manipulators remains a heavily researched topic due to the complexity of such systems and their associated mission concepts. A wealth of research has been published addressing one or more of the problems related to the use of free-flying robotic satellites. Particularly, multiple studies address the dynamics and control of such systems. Typical methods for deriving the dynamics include Kane's Method [28, 29, 30], the Lagrangian formulation [31, 32], direct application of the conservation of momentum [20, 33], or by applying Newton's Law [34]. Various methods are employed to control spacecraft with robotic manipulators. Existing systems typically use some combination of proportional-integral-derivative (PID) control [35, 36, 37], and many experimental or modelled systems analyze the expected performance of PID control [17, 19, 28, 34, 38, 39, 40]. When sensor measurements, or uncertainty in either the system dynamics or its environment, are considered, adaptive control methods [17, 20, 28, 41, 42] and robust control methods [19, 43] are used. Several studies investigate the use of more novel control methods such as convex optimization [18, 44, 45, 46], pseudospectral methods of optimal control [31], genetic algorithms [47, 48], or model predictive control (MPC) [25, 49].

Many of the aforementioned studies and existing orbital systems focus on larger servicing satellites on the order of several hundred kilograms. These systems benefit from the relatively small degree of coupling between the base and manipulator motion. This study aims to analyze the feasibility of a robotic servicing CubeSat. The small mass of the base relative to the manipulator makes the task of nulling out disturbance forces and torques on the base due to the manipulator motion particularly challenging. To analyze the feasibility of a robotic servicing CubeSat, this

research focuses on a notional 6U CubeSat model equipped with a four-link serial manipulator. The dynamics of the system are derived using Kane's method. Standard techniques for manipulator trajectory planning in physical and joint space are applied, and a feedback PID controller is implemented. Three representative orbital scenarios are simulated with the goal of determining the expected performance of a robotic servicing CubeSat, and the magnitude of controls required for such a system. The results can be used to determine the appropriate actuators for a real-world system. Recommended improvements to both the system design, and the model are discussed.

2.3 Technical Primer

Several concepts often used in robotics are leveraged for this research and require introduction. These include Kane's Method for deriving the dynamics of multibody systems, methods for solving for the system kinematics, manipulability, and trajectory planning. Each of these concepts is summarized in this section.

2.3.1 Kane's Method

Kane's Method is a well-established method for deriving the equations of motion of a dynamical system [30, 50]. It is particularly useful for multibody systems subject to kinematic and dynamic constraints [51]. Consider a system of N rigid bodies subject to m constraints. Rather than computing the inertial acceleration of the center of mass (COM) of each rigid body in the system and applying Newton's Second Law, resulting in $3N + m$ differential equations, Kane's Method leverages a minimal set of n generalized coordinates, as well as cardinal vectors, or directions of allowable motion, to reduce the dynamics to just n second-order differential equations. This is achieved by projecting Newton's Second Law along the directions of the cardinal vectors, also known as partial velocities.

Let q_k $(k = 1, \ldots, n)$ be the minimal set of generalized coordinates describing the configuration of a system of $j = 1, \ldots, N$ rigid bodies, $\vec{r}^{\,j}(t, q_k)$ and $\vec{v}^{\,j}(t, q_k, \dot{q}_k)$ be the position and velocity of the COM of body j, and $\vec{\omega}^{\,j}(t, q_k, \dot{q}_k)$ its angular velocity. The $n + 1$ translational cardinal vectors for body j are:

$$\begin{aligned} \vec{\alpha}_0^{\,j}(t, q_k) &= \frac{\partial \vec{r}^{\,j}}{\partial t}\,, \\ \vec{\alpha}_k^{\,j}(t, q_k) &= \frac{\partial \vec{r}^{\,j}}{\partial q_k}\,. \end{aligned} \tag{1}$$

And the $n + 1$ rotational cardinal vectors for body j are:

$$\begin{aligned} \vec{\beta}_k^{\,j}(t, q_k) &= \frac{\partial \vec{\omega}^{\,j}}{\partial \dot{q}_k}\,, \\ \vec{\beta}_0^{\,j}(t, q_k) &= \vec{\omega}^{\,j} - \sum_{k=1}^{n} \vec{\beta}_k^{\,j} \dot{q}_k\,. \end{aligned} \tag{2}$$

Now one can compute a set of n scalar momentum rates, $K_k^\sharp$, also known as generalized inertia forces, by projecting the time derivative of linear and angular momentum for each body along their respective cardinal vectors, as shown below:

$$K_k^\sharp(t, q_k, \dot{q}_k, \ddot{q}_k) = \sum_{j=1}^{N} m_j \dot{\vec{v}}^{\,j} \cdot \vec{\alpha}_k^{\,j} + \dot{\vec{L}}^{\,j} \cdot \vec{\beta}_k^{\,j}\,, \tag{3}$$

where $\vec{L}^{\,j}$ is the angular momentum of body j. The next step is to project the external forces and moments along the cardinal vectors for each generalized coordinate. Let $\vec{F}^{\,j}$ be the sum of external forces on body j and $\vec{T}^{\,j}$ be the sum of external torques about that body's COM, then the n generalized active forces are:

$$K_k^\flat(t, q_k, \dot{q}_k) = \sum_{j=1}^{N} \vec{F}^{\,j} \cdot \vec{\alpha}_k^{\,j} + \vec{T}^{\,j} \cdot \vec{\beta}_k^{\,j}\,. \tag{4}$$

The n equations of motion are then simply:

$$K_k^\sharp = K_k^\flat \,. \tag{5}$$

Equation 5 is one form of Kane's dynamical equations. A detailed derivation, starting from Newton's Second Law, can be found in Kane's original work [30, 50].

This method has reduced the number of differential equations from $3N + m$ to n, and greatly simplified the treatment of the m constraints. The tedious part of this method is identifying the resultant forces and torques on each body in the system and computing the system kinematics.

2.3.2 Robot Kinematics

A useful concept for deriving the kinematics of coupled, multibody systems is the generalized transformation matrix, which transforms a vector from one frame to another through a relative rotation and translation. Consider the vector $\vec{v}^{\,A}$ expressed in the A frame: $\vec{v}^{\,A}$ is transformed into the B frame as shown in Equation 6 below, where $\widetilde{v}^{\,i}$ is the augmented vector formed by appending a 1 to the end of $\vec{v}^{\,i}$, $\mathbf{T}_{BA}$ and $\mathbf{C}_{BA}$ are the transformation and rotation matrices from the A to the B frame, respectively, and $\vec{p}_A^{\,B}$ is the position of the A origin expressed in the B frame.

$$\widetilde{v}^{\,B} = \begin{bmatrix} \vec{v}^{\,B} \\ 1 \end{bmatrix} = \mathbf{T}_{BA}\, \widetilde{v}^{\,A} = \begin{bmatrix} \mathbf{C}_{BA} & \vec{p}_A^{\,B} \\ \vec{0}^{\,T} & 1 \end{bmatrix} \begin{bmatrix} \vec{v}^{\,A} \\ 1 \end{bmatrix} \tag{6}$$

The transformation matrices can be used to determine the pose of the end effector in the frame of any preceding joint, or in the base or inertial frames. Computing the end effector pose from the base pose and manipulator joint angles is known as forward or direct kinematics, and is a straightforward problem with a unique solution. The reverse of this problem, determining the system configuration from the end effector pose, is not straightforward and often has multiple or infinitely many solutions.

However, given some starting configuration at time t_0, the configuration space trajectory that will produce some desired end effector trajectory can be determined through inverse differential kinematics [52].

Differential kinematics begins with the end effector position in the inertial frame, $\vec{r}_{EE}$, or the solution to the forward kinematics problem. Taking the partial derivative with respect to the generalized coordinates yields the analytical Jacobian matrix:

$$\mathbf{J} = \frac{\partial \vec{r}_{EE}}{\partial \vec{q}} = \begin{bmatrix} \frac{\partial r_1}{\partial q_1} & \frac{\partial r_1}{\partial q_2} & \cdots & \frac{\partial r_1}{\partial q_n} \\ \frac{\partial r_2}{\partial q_1} & \frac{\partial r_2}{\partial q_2} & \cdots & \frac{\partial r_2}{\partial q_n} \\ \frac{\partial r_3}{\partial q_1} & \frac{\partial r_3}{\partial q_2} & \cdots & \frac{\partial r_3}{\partial q_n} \end{bmatrix}, \tag{7}$$

where $\vec{q}$ is the column vector of generalized coordinates. The Jacobian maps changes in the generalized coordinates to changes in the end effector velocity in the current configuration:

$$\vec{v}_{EE} = \mathbf{J}\dot{\vec{q}}. \tag{8}$$

Equation 8 is known as the differential kinematic equation, and can be inverted to yield:

$$\dot{\vec{q}} = \mathbf{J}^{-1}\vec{v}_{EE}\,, \tag{9}$$

or the inverse differential kinematic equation. In the case of a system with redundant DOF, or more generalized coordinates than spatial dimensions, the Jacobian is not square and therefore cannot be directly invertible. In this case a pseudo-inverse of the Jacobian can be used to yield an acceptable solution that also satisfies some desirable criteria, such as singularity avoidance. The Moore-Penrose pseudo-inverse, $\mathbf{J}^{\dagger}$, shown in Equation 10 minimizes the $\left|\dot{\vec{q}}\right|$ that achieves the desired $\vec{v}_{EE}$:

$$\dot{\vec{q}} = \mathbf{J}^{\dagger}\vec{v}_{EE} = \left(\mathbf{V}\mathbf{S}^{\dagger}\mathbf{U}\right)\vec{v}_{EE}\,, \tag{10}$$

where the singular value decomposition of the Jacobian is $\mathbf{J} = \mathbf{USV}^T$. The pseudo-inverse of $\mathbf{S}$ is determined by taking the reciprocal of the non-zero elements, leaving the zero elements in place, then transposing the matrix.

Note that the Jacobian in Equation 7 only accounts for the translational velocity of the end effector. However, this concept easily extends to include the angular velocity if both the position and orientation of the end effector are of interest. With $\dot{\vec{q}}$ known, the required generalized coordinates are then:

$$\vec{q} = \vec{q}_0 + \int_0^t \dot{\vec{q}}\, dt\,. \tag{11}$$

2.3.3 Manipulability

Manipulability is a robotics concept that is useful for evaluating manipulator performance. For a fixed-base manipulator, the reachable workspace includes every point that the end effector can reach. At the edge of the reachable workspace, the manipulator cannot move the end effector in a certain direction with changes in its joints, namely across the boundary of the workspace. Therefore this configuration is not manipulable. In a manipulable configuration, the manipulator can arbitrarily change the pose of the end effector with changes in its joints. It is desired to determine how manipulable a given configuration is. This information is captured in the velocity manipulability ellipsoid [52], which represents the end effector velocities attainable by a manipulator in a given configuration with certain magnitudes of joint speeds.

Consider a set of joint speeds with constant unit norm:

$$\dot{\vec{q}}^T \dot{\vec{q}} = 1\,, \tag{12}$$

which defines a sphere in joint velocity space. Note that the sphere's radius can be any positive number depending on the capabilities of the manipulator. Unity is used for simplicity as the goal is to determine the relative manipulability of a configuration.

Substituting Equation 10 for $\dot{\vec{q}}$ yields:

$$\vec{v}_{EE}^{\,T} \left(\mathbf{J}^{\dagger T} \mathbf{J}^{\dagger} \right) \vec{v}_{EE} = 1 \,. \tag{13}$$

Equation 13 describes the points on the surface of an ellipsoid in end effector velocity space that the manipulator can achieve with constant unit norm joint speeds. The volume of this ellipsoid indicates the relative manipulability of the configuration. In a singular configuration this three dimensional ellipsoid collapses down to a planar ellipse or a line.

Note that in the case of a redundant manipulator the Jacobian has more columns than rows, and its pseudo-inverse can be expressed as:

$$\mathbf{J}_r^{\dagger} = \mathbf{J}^T \left(\mathbf{J}\mathbf{J}^T \right)^{-1} , \tag{14}$$

or the right pseudo-inverse. Substituting this expression into Equation 13 yields:

$$\vec{v}_{EE}^{\,T} \left(\mathbf{J}\mathbf{J}^T \right)^{-1} \vec{v}_{EE} = 1 \,. \tag{15}$$

The volume of the ellipsoid is thus proportional to:

$$w = \sqrt{\det \left(\mathbf{J}\mathbf{J}^T \right)} \,, \tag{16}$$

which is termed the manipulability measure. Configurations with larger w are desired as they have increased manipulability. A singular configuration will have $w = 0$.

This concept is not only useful for robot kinematics but for statics and dynamics as well. The relation between joint forces and end effector forces is mapped through the same Jacobian that relates joint and end effector speeds [52]. Therefore configurations that are highly manipulable are also beneficial for imparting relatively large forces to the end effector.

2.3.4 Trajectory Planning

Trajectory planning is an important concept in robotics as any task requires, at a minimum, the manipulator move from one configuration to another. This can either be point-to-point motion, or motion through a series of points. In both cases a suitable trajectory must be established, either in configuration (joint) or operational (Cartesian) space, both of which have been heavily researched for decades [53]. Suitable trajectories for robotic manipulators might be ones that avoid obstacles in the workspace and singular configurations, and prevent the joint actuators from saturating. The latter quality generally requires that the joint positions and velocities are continuous functions of time, and the trajectories are smooth [52].

Many methods exist for forming smooth trajectories between points or through a series of points. These typically involve polynomial or spline interpolation [54, 55]. In classical Euclidean spaces, Bézier curves are often used to generate smooth curves through a series of points. The De Casteljau algorithm is a stable recursive method for generating Bézier curves [56]. To generate a curve of degree n, the algorithm is as follows:

1. Define $n + 1$ control points (that determine the shape of the curve).
2. Connect the control points with straight lines to form a control polygon.
3. For some parameter $t \in [0, 1]$ subdivide each segment of the polygon by the ratio $t : (1 - t)$.
4. Connect each subdivided point to form a new polygon with one less side.
5. Repeat Steps $3 - 4$ until there is a single point. This point lies on the Bézier curve.
6. Vary the parameter t from 0 to 1 for the point to trace out the Bézier curve.

This method has been extended to the group of rigid body motions in three dimensional space, also known as the special Euclidean group $SE(3)$, to form smooth trajectories between two poses with boundary conditions on translational and angular velocity [57]. The algorithm is reproduced here for completeness, with the theoretical underpinnings omitted. More details can be found in references [57] and [58].

Consider the pose of a rigid body expressed as a generalized transformation matrix:

$$\mathbf{T} = \begin{bmatrix} \mathbf{C} & \vec{p} \\ \vec{0}^T & 1 \end{bmatrix} , \tag{17}$$

where $\mathbf{C}$ is the direction cosine matrix representing the body's attitude, typically in the inertial frame, and $\vec{p}$ is the position of the body COM. The generalized velocities of the body can be represented as follows:

$$\mathbf{V} = \begin{bmatrix} [\vec{\omega}^{\times}] & \vec{v} \\ \vec{0}^T & 0 \end{bmatrix} , \tag{18}$$

where $[\vec{\omega}^{\times}]$ is the skew-symmetric matrix associated with the angular velocity of the body, and $\vec{v}$ is the inertial velocity of the body COM. The following kinematic equation holds:

$$\dot{\mathbf{T}} = \mathbf{T}\mathbf{V} . \tag{19}$$

It is desired to generate a smooth curve in $SE(3)$ connecting an initial and final pose, $\mathbf{T}_0$ and $\mathbf{T}_f$, with boundary conditions on the velocities, $\mathbf{V}_0$ and $\mathbf{V}_f$. For simplicity assume $t \in [t_0, t_f] = [0, 1]$. For different t_0 and t_f the time vector and velocities can be scaled. The De Casteljau algorithm on $SE(3)$ is as follows:

1. Calculate boundary conditions on $\dot{\mathbf{T}}$:

$$\begin{aligned} \dot{\mathbf{T}}_0 &= \mathbf{T}_0 \mathbf{V}_0 \\ \dot{\mathbf{T}}_f &= \mathbf{T}_f \mathbf{V}_f \end{aligned} \tag{20}$$

2. Transform boundary conditions into infinitesimal generators:

$$\begin{aligned} \mathbf{V}_0^1 &= \frac{1}{3} \mathbf{T}_0^{-1} \dot{\mathbf{T}}_0 \\ \mathbf{V}_2^1 &= \frac{1}{3} \mathbf{T}_f^{-1} \dot{\mathbf{T}}_f \end{aligned} \tag{21}$$

3. Compute control points:

$$\begin{aligned} \mathbf{T}_1 &= \mathbf{T}_0 e^{\mathbf{V}_0^1} \\ \mathbf{T}_2 &= \mathbf{T}_f e^{-\mathbf{V}_2^1} \end{aligned} \tag{22}$$

4. Find interior velocity that maps $\mathbf{T}_1$ to $\mathbf{T}_2$:

$$\mathbf{V}_1^1 = \log(\mathbf{T}_1^{-1} \mathbf{T}_2) \tag{23}$$

5. Find the time-varying "polynomial" generating velocities:

$$\begin{aligned} \mathbf{V}_0^2(t) &= \log(e^{\mathbf{V}_0^1 (1-t)} e^{\mathbf{V}_1^1 t}) \\ \mathbf{V}_1^2(t) &= \log(e^{\mathbf{V}_1^1 (1-t)} e^{\mathbf{V}_2^1 t}) \\ \mathbf{V}_0^3(t) &= \log(e^{\mathbf{V}_0^2(t) (1-t)} e^{\mathbf{V}_1^2(t) t}) \end{aligned} \tag{24}$$

6. Compute the pose matrix:

$$\mathbf{T}(t) = \mathbf{T}_0 e^{\mathbf{V}_0^1 t} e^{\mathbf{V}_0^2(t)\, t} e^{\mathbf{V}_0^3(t)\, t} \tag{25}$$

This algorithm leverages the matrix exponential and logarithm to yield a smooth curve in $SE(3)$ between two arbitrary poses. The time history of the generalized velocities can be computed with the kinematic relation in Equation 19. Note that the resulting trajectory is smooth in terms of pose and velocities, but the accelerations are only continuous - not necessarily smooth. Some robot applications desire trajectories with continuous jerk, and therefore must use higher-order polynomial splines [55].

2.4 Summary

In this chapter several past and present space robotics missions were discussed, along with current research in the field to highlight the unique contributions of this research: the focus on CubeSat platforms, and the rigorous treatment of coupled orbit-attitude motion in the dynamics model of the system. Several topics pertinent to this research were introduced including Kane's Method, robot kinematics, and trajectory planning in order to understand the Methodology.

III. Methodology

This chapter begins with a description of the CubeSat robotic system used in simulations and subsequent analysis. The process of deriving the dynamics is then discussed, followed by a description of the simulation framework used to evaluate the system, with emphasis on the controller. Three types of scenarios are simulated to demonstrate the system's performance. Each scenario is presented along with the performance metrics used in the analysis.

3.1 The System

To investigate the feasibility of a robotic servicing CubeSat, a notional 6U CubeSat with a four-link serial manipulator is used. The 6U format is chosen as it is the smallest base platform that could reasonably be expected to house both the manipulator, and all additional subsystems needed for successful operation of the system, particularly propulsion, ADACS, and navigation sensors. A computer-aided design (CAD) model of the system structure is shown in Figure 1.

The four-link serial manipulator is inspired by the anthropomorphic geometry, a common manipulator structure described in Section 1.2 of Siciliano's textbook [52]. A diagram of the manipulator with its workspace, taken from this text, is shown in Figure 2. Typically comprised of three links with revolute joints, where the axis of rotation of the first is perpendicular to that of the other two, which are parallel, the anthropomorphic manipulator is so named due to its similarity to the human arm: the first two joints comprise the shoulder, while the third functions as an elbow [52]. The fourth joint and link are added to this model to give the manipulator additional reach. Notice the void at the top of the manipulator workspace in Figure 2. This is due the fact that the manipulator has a kinematic singularity whenever the end effector lies on the line extending out from the first link. This singularity has serious implications for the system performance and trajectory planning, as discussed in Chapter IV.

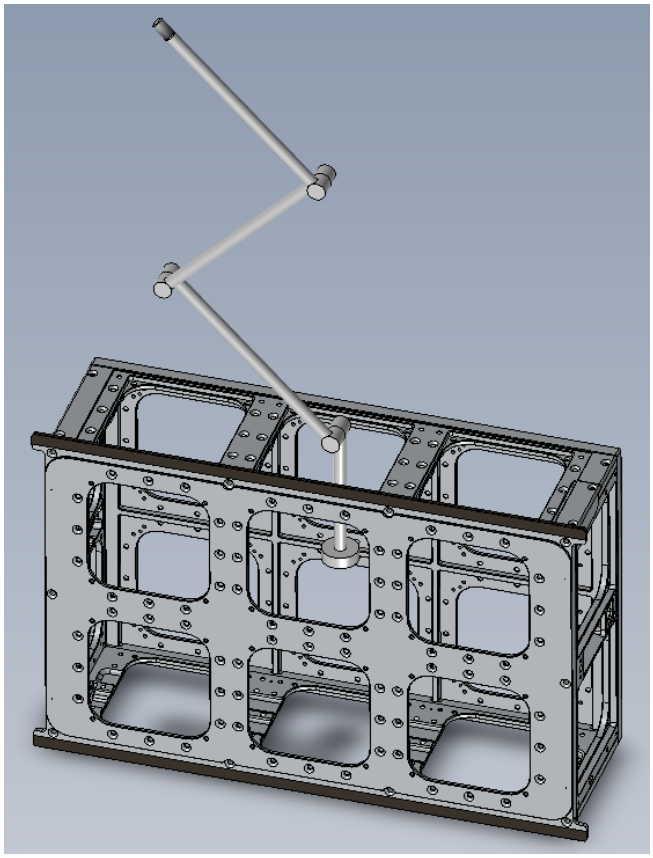

Figure 1: CAD model of 6U CubeSat robot structure

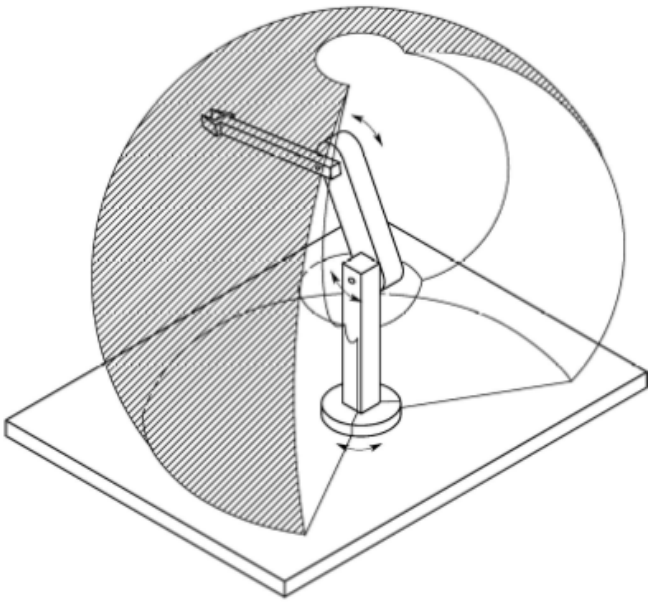

Figure 2: Anthropomorphic manipulator with workspace

As the manipulator contains only four DOF, it cannot achieve an arbitrary pose for the end effector in three dimensional space, as that would require six DOF. Therefore, throughout the simulations the orientation of the end effector is ignored and only its position and velocity is considered.

The CubeSat base and each of four links are modelled as solid, rigid bodies of uniform density that are kinematically coupled. The base is a rectangular prism and the links are thin cylindrical rods. The joint dimensions and masses are assumed to be negligible. This is a reasonable assumption for a low fidelity robot model, as the joints in a typical manipulator are small and light compared to the links. The mass, dimensions, and inertia properties of each body in the system is shown in Table 1. The inertia matrices are expressed in the principal frames of each body.

Table 1: 6U CubeSat Robot Specifications

Body	**Mass (kg)**	**Dimensions (cm)**	**Inertia Matrix ($\mathbf{kg \cdot m^2}$)**
Base	10	$10 \times 20 \times 30$	$10^{-2} \cdot \text{diag}\,[4.17,\ 8.03,\ 10.83]$
Link 1	0.5	10	$10^{-4} \cdot \text{diag}\,[4.17,\ 0.25,\ 4.17]$
Link 2	1	15	$10^{-4} \cdot \text{diag}\,[0.50,\ 18.75,\ 18.75]$
Link 3	1	15	$10^{-4} \cdot \text{diag}\,[0.50,\ 18.75,\ 18.75]$
Link 4	1	15	$10^{-4} \cdot \text{diag}\,[0.50,\ 18.75,\ 18.75]$

Figure 3 shows the relevant coordinate frames in the system, where $\mathcal{N}$ is the Earth-centered inertial frame. The red vectors point in the direction of the x-axes of each frame, y-axes are shown in green, and z-axes in blue. Joint 1 is coincident with the base COM so that link 1 extends to the 10×30 cm face of the CubeSat that is perpendicular to the $\hat{y}$ axis of the base frame, $\mathcal{B}$. Joint 1 rotates about the $\hat{y}$ axis of its frame, $\mathcal{J}_1$, while joints 2 - 4 rotate about $\hat{z}$ of their respective frames.

The lengths of links 2 - 4 were chosen such that the manipulator can be stowed alongside the CubeSat as shown in Figure 4. This configuration reduces the footprint of the system, allowing it to be stored in and deployed from existing operational systems, such as the NanoRacks CubeSat Deployer [59].

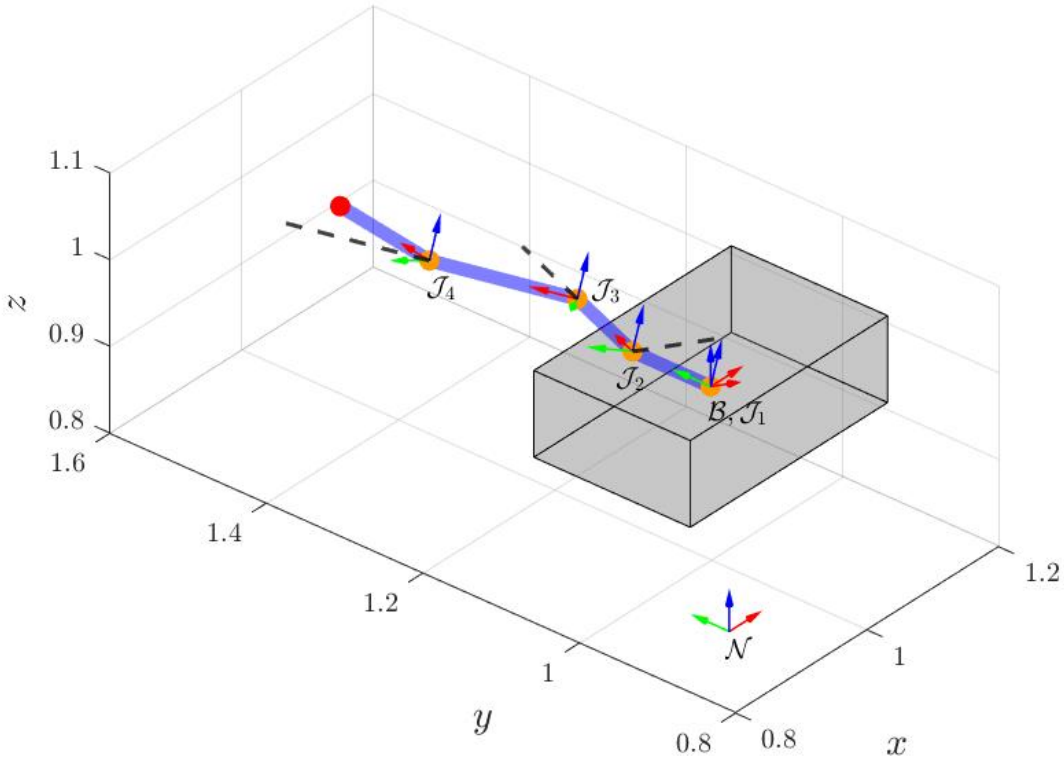

Figure 3: 6U CubeSat robot reference frames

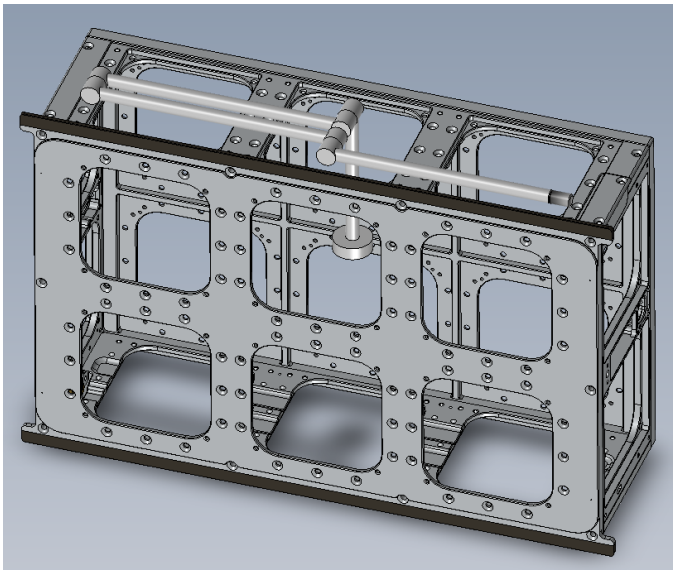

Figure 4: 6U CubeSat robot in stowed configuration

The system is composed of 21 states, described below.

- $\vec{r}_B^N = [x, y, z]^T$ is the base position expressed in the inertial frame.
- $\underline{e}_{BN} = [e_0, \vec{e}^T] = [e_0, e_1, e_2, e_3]^T$ is the quaternion describing the base attitude with respect to the inertial frame, where e_0 is the scalar portion.
- $\vec{\Theta} = [\theta_1, \theta_2, \theta_3, \theta_4]^T$ is the vector of joint angles.
- $\dot{\vec{r}}_B^N = [\dot{x}, \dot{y}, \dot{z}]^T$ is the base velocity expressed in the inertial frame.
- $\vec{\omega}_{BN}^B = [\omega_x, \omega_y, \omega_z]^T$ is the base angular velocity relative to the inertial frame expressed in the base frame.
- $\dot{\vec{\Theta}} = [\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, \dot{\theta}_4]^T$ is the vector of joint angular speeds.

The base is three-axis stabilized and has thrust capability in each axis of the $\mathcal{B}$ frame. Additionally, each joint has an associated torque, for a total of 10 controls in the system. The torques and forces on the base are assumed to be applied at the base COM. The minimum and maximum value of each control is shown in Table 2, where f_B and τ_B are forces and torques on the base, respectively, and τ_i is the torque on joint i. The controls can be applied continuously and assume arbitrarily small values.

Table 2: Control Bounds

Control	**Range**
f_B	± 0.1 N
τ_B	± 0.1 N·m
τ_i	± 0.05 N·m

These bounds were chosen because they are within the ranges of thrust and torque provided by many commercial off the shelf (COTS) subsystems for CubeSats. According to the NASA's 2020 State-of-the-Art Small Spacecraft Technology report,

there are over 10 propulsion systems, either commercially available or in development, that provide at least 0.1 N of thrust in the 1U to 2U form factor. The same report identifies several reaction wheel systems for CubeSats with maximum torques of 0.1 N·m. [7]. A different study investigating the feasibility of on-orbit robotic assembly of small satellites identifies a commercially available motor commonly used in robotic arms that delivers 0.05 N·m of torque with a mass of 0.11 kg [47], further justifying the massless joint assumption.

3.2 Derivation of System Dynamics

The equations of motion of the system are generated with MotionGenesis Kane 5.9, a software package that applies Kane's Method to derive the dynamics of a multibody system [60]. The user inputs simple commands to define the system and the relevant coordinate frames, the generalized coordinates and kinematics governing the system configuration, and the forces acting on the system. MotionGenesis applies Kane's Method to derive the dynamics, and auto-generates fast, memory-efficient MATLAB code [61, 62, 63]. The list of commands used to generate the MATLAB dynamics function is shown in Algorithm 1, where repeated commands are omitted.

The resulting MATLAB function takes in the system state and controls and outputs the acceleration-level variables:

$$\ddot{\vec{q}} = f(\vec{X}, \vec{U})\,. \tag{26}$$

Algorithm 1 MotionGenesis Commands

1: NewtonianFrame N ▷ Define inertial frame
2: RigidBody B, L1, L2... ▷ Declare rigid bodies
3: Point J1(B) ▷ Declare joint 1, coincident with base
4: Repeat for all joints
5: Constant G $= 6.6732 \times 10^{-11}$ $\text{Nm}^2\text{kg}^{-2}$ ▷ Define all system constants (masses, lengths, etc.)
6: B.SetMass(m_B) ▷ Set the mass of each body
7: Constant $I_{1,xx} = \frac{1}{12} m_1 l_1^2$ ▷ Compute principal moments for each body
8: B.SetInertia($B_{CM}, I_{B,xx}, I_{B,yy}, I_{B,zz}, I_{B,xy}, I_{B,yz}, I_{B,xz}$) ▷ Set inertia matrix for each body about its COM
9: Variable $x'', y'', ...$ ▷ Declare all generalized coordinates at acceleration level
10: SetGeneralizedSpeeds($x', y', ...$) ▷ Set generalized speeds
11: Define rotational kinematics (Steps 12-15):
12: $\mathcal{B}$.SetAngularVelocityAcceleration($\mathcal{N}, \omega_x \hat{b}_x + \omega_z \hat{b}_y + \omega_z \hat{b}_z$) ▷ Set base angular velocity w.r.t. inertial frame
13: $\mathcal{B}$.SetRotationMatrixODE($\mathcal{N}$,Quaternion,e_0,e_1,e_2,e_3) ▷ Set quaternion kinematic equation
14: L1.RotateY($\mathcal{B}, \theta_1$) ▷ Rotate Link 1 about $\hat{b}_y$ by θ_1
15: Repeat for all links
16: Define translational kinematics (Steps 17-19):
17: $\mathcal{B}^{\mathcal{O}}$.Translate($\mathcal{N}^{\mathcal{O}}, x\,\hat{n}_x + y\,\hat{n}_y + z\,\hat{n}_z$) ▷ Translate origin of base frame from inertial origin
18: J2.Translate($\mathcal{L}_1^{\mathcal{O}}, l_1\,\hat{l}_{1,x}$) ▷ Translate Joint 2 from Link 1 origin
19: Repeat for all link origins, COMs, and joints
20: $\vec{r}_B =$ Express(B_{CM}.GetPosition($\mathcal{N}^{\mathcal{O}}$), $\mathcal{N}$) ▷ Get inertial position vector of the COM of each body
21: B_{CM}.AddForce$\left(\dfrac{-Gm_E m_B}{|r_B|^3} \vec{r}_B \right)$ ▷ Add gravitational force to each body
22: Specified $f_x, f_y, ..., \tau_x, \tau_y, ..., \tau_1, \tau_2, ...$ ▷ Declare control forces and torques as input variables
23: B_{CM}.AddForce($f_x\,\hat{b}_x + f_y\,\hat{b}_y + f_z\,\hat{b}_z$) ▷ Add control force to base
24: B.AddTorque($\tau_x\,\hat{b}_x + \tau_y\,\hat{b}_y + \tau_z\,\hat{b}_z$) ▷ Add control torque to each body
25: Kane = System.GetDynamicsKane() ▷ Get system dynamics
26: Code algebraic(Kane, [x'' , y'' , z'' , ω_x' , ω_y' , ω_z' , θ_1'' , θ_2'' , θ_3'' , θ_4'']) Dynamics.m ▷ Generate MATLAB Dynamics.m function with the specified outputs

3.3 Simulation Framework

The dynamics function produced with MotionGenesis is incorporated into the Robot Motion block of the Simulink model shown in Figure 5, which is used to run several orbital scenarios. The Simulink model is broken into five subsystems: the space station, or chief, orbital dynamics, the CubeSat robot dynamics and kinematics, the relative states, the state errors, and the controller.

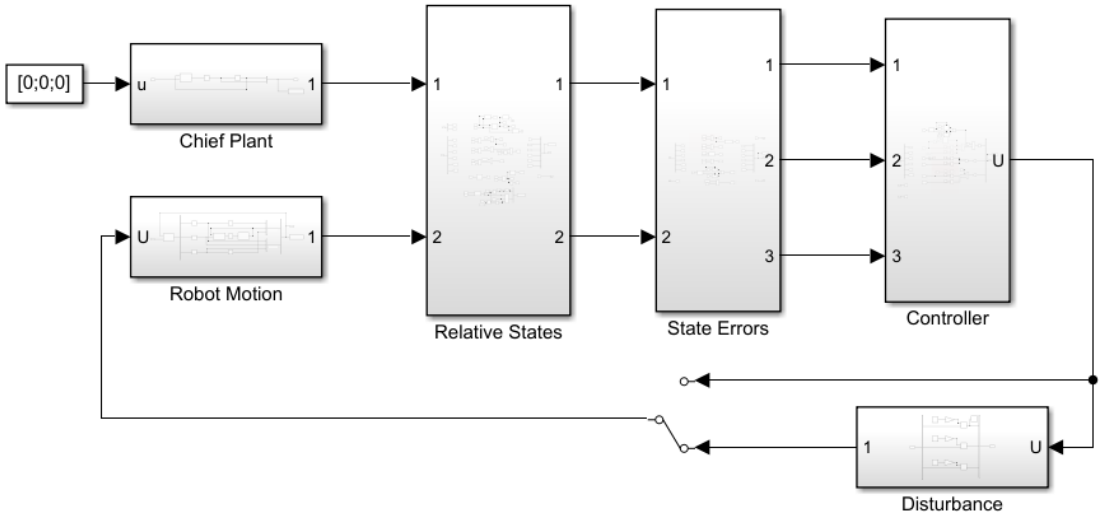

Figure 5: Simulink model flow diagram

The model uses the ode45 solver with relative tolerance 10^{-12} and absolute tolerance 10^{-14}. Lengths are expressed in meters rather than the more common unit of kilometers used for orbital dynamics for several reasons. The first is that all scenarios investigated involve close proximity operations, where relative positions and velocities are small and more intuitively expressed in meters and meters per second. Additionally, the use of meters for lengths avoids small physical dimensions for the CubeSat robot on the order of 10^{-4}. Finally, the use of meters avoids confusion with the dimensions of the feedback controls, where Newtons and Newton-meters are desired for forces and torques, respectfully.

3.3.1 Space Station Dynamics

The chief spacecraft is modelled as a point source that moves under the gravitational attraction of a spherical Earth, as shown in Equation 27:

$$\ddot{\vec{r}}_c = \frac{-Gm_e}{r_c^3}\vec{r}_c\,, \tag{27}$$

where $r_c = |\vec{r}_c|$. The chief acceleration is integrated twice to yield its position and velocity. These can be used to determine the rotation between the inertial and LVLH frames, $\mathbf{C}_{LN}$, as shown in Equation 28:

$$\mathbf{C}_{LN} = \begin{bmatrix} \hat{r}_c^{\,T} \\ \hat{t}^{\,T} \\ \hat{h}^{\,T} \end{bmatrix}\,, \tag{28}$$

where $\hat{r}_c = \dfrac{\vec{r}_c}{|\vec{r}_c|}$ is the radial unit vector, $\vec{h} = \vec{r}_c \times \dot{\vec{r}}_c$ is the chief's specific angular momentum vector, and $\hat{t} = \hat{h} \times \hat{r}_c$ is the orbital tangent, or in-track, vector that completes the right-handed coordinate system.

Non-spherical effects such as J_2 are ignored, as the duration of the simulations are short with respect to any Earth orbital period, and since the chief and CubeSat remain in close proximity, any ellipticity effects would perturb each to a similar degree. The chief spacecraft is not modelled as an extended rigid body for simplicity. However, this does not reduce the generality of the results, since the operational concept assumes the chief spacecraft is three-axis stabilized and aligned with the LVLH frame, and therefore any location on the spacecraft exterior that requires servicing would be stationary with respect to this frame. In this case, any servicing location on the spacecraft would serve as a stationary target for the CubeSat to rendezvous and dock with, no different than the LVLH origin.

For chief spacecraft with a specific structure, the rigid body motion could easily be incorporated into the model. As the orbital dynamics are modelled with Newtonian gravity, perturbing forces such as aerodynamic drag, solar radiation pressure, and J_2 would be straightforward to include.

3.3.2 Dynamics and Kinematics

As mentioned in Section 3.3, the dynamics function for the CubeSat robot takes in the state and controls and outputs the acceleration-level variables:

$$\ddot{\vec{q}} = \left[\ddot{x},\, \ddot{y},\, \ddot{z},\, \dot{\omega}_x,\, \dot{\omega}_y,\, \dot{\omega}_z,\, \ddot{\theta}_1,\, \ddot{\theta}_1,\, \ddot{\theta}_3,\, \ddot{\theta}_4\right] . \tag{29}$$

The translational acceleration and joint angular accelerations are integrated twice to yield their respective velocity- and position-level states. The base angular acceleration is integrated once to yield the base angular speed, which is fed into Equation 30 to yield the time derivative of the quaternions.

$$\dot{\underline{e}}_{BN} = \frac{1}{2}\begin{bmatrix} -e_1 & -e_2 & -e_3 \\ e_0 & -e_3 & e_2 \\ e_3 & e_0 & -e_1 \\ -e_2 & e_1 & e_0 \end{bmatrix} \vec{\omega}^B_{BN} = \frac{1}{2}\begin{bmatrix} -\vec{e}^T \\ [\vec{e}^{\times}] + e_0\mathbf{I} \end{bmatrix} \vec{\omega}^B_{BN} \tag{30}$$

$\dot{\underline{e}}_{BN}$ is then integrated and fed back into Equation 30. All CubeSat states are fed back into the dynamics function, and passed on to a subsequent block to calculate the relative states with respect to the chief.

3.3.3 Relative States

The states of interest are those relative to the orbital frame of the chief, or the LVLH frame, and therefore the CubeSat states must be converted to this frame. The Relative States block takes in the inertial states of the chief, and the states of the CubeSat from its dynamics and kinematics block and outputs the relative states of the CubeSat. The relative translational states of the CubeSat expressed in the LVLH frame are calculated with Equations 31 and 32.

$$\vec{\rho}^{\,L} = \mathbf{C}_{LN}\left(\vec{r}_c^{\,N} - \vec{r}_B^{\,N}\right) \tag{31}$$

$$\dot{\vec{\rho}}^{\,L} = \mathbf{C}_{LN}\left(\dot{\vec{r}}_c^{\,N} - \dot{\vec{r}}_B^{\,N}\right) - \vec{\omega}_{LN}^{\,L} \times \vec{\rho}^{\,L} \tag{32}$$

The relative quaternion between the CubeSat base and the LVLH frame is calculated as follows:

$$\underline{e}_{BN} \rightarrow \mathbf{C}_{BN} \rightarrow \mathbf{C}_{BL} = \mathbf{C}_{BN}\mathbf{C}_{NL} \rightarrow \underline{e}_{BL}\,, \tag{33}$$

where Simulink's built-in functions are used to convert between quaternions and direction cosine matrices.

The relative angular velocity between the CubeSat base and LVLH frame, expressed in the base frame, is:

$$\vec{\omega}_{BL}^{\,B} = \vec{\omega}_{BN}^{\,B} - \mathbf{C}_{BL}\vec{\omega}_{LN}^{\,L}\,. \tag{34}$$

The joint angles and angular speeds simply pass through this block, as they are measured with respect to the base frame. Given the other relative states, the position and velocity of the end effector in the LVLH frame can be calculated through forward kinematics.

3.3.4 State Errors

The relative states are passed into a subsystem block to calculate the state errors, defined as the difference between the desired state, X_d, and the current state, X:

$$X_e = X_d - X\,. \tag{35}$$

Equation 35 applies to all states except the quaternion. The quaternion error is determined by Equation 36.

$$\underline{e}_e = \begin{bmatrix} e_0 & -e_1 & -e_2 & -e_3 \\ e_1 & e_0 & e_3 & -e_2 \\ e_2 & -e_3 & e_0 & e_1 \\ e_3 & e_2 & -e_1 & e_0 \end{bmatrix} \begin{bmatrix} e_{d,0} \\ -e_{d,1} \\ -e_{d,2} \\ -e_{d,3} \end{bmatrix}, \tag{36}$$

where $\underline{e}_d$ is the desired quaternion. The prescription of the desired states depends on the state in question and on the scenario, which will be discussed in Section 3.4.

3.3.5 Controller

The state errors are passed into the Controller block, where they are converted to generalized forces requested of the actuators. As the actuators are not modelled, the requested controls are fed directly back into the CubeSat dynamics. The translational state errors are regulated with a PID controller of the following form:

$$\vec{f}^{\,L} = \mathbf{K}_{p,t}\vec{\rho}_e^{\,L} + \mathbf{K}_{i,t}\int_0^t \vec{\rho}_e^{\,L}\,dt + \mathbf{K}_{d,t}\dot{\vec{\rho}}_e^{\,L}\,, \tag{37}$$

where $\mathbf{K}_{p,t}$, $\mathbf{K}_{i,t}$, and $\mathbf{K}_{d,t}$ are 3×3 diagonal, positive definite gain matrices on the position, integral of position, and velocity errors, respectfully. Notice the resulting force is expressed in the LVLH frame. The CubeSat thrusters are assumed to be fixed in the base frame, and therefore $\vec{f}^{\,L}$ must be rotated back into the base frame: $\vec{f}^{\,B} = \mathbf{C}_{BL}\vec{f}^{\,L}$.

The attitude errors are regulated with a quaternion feedback control law:

$$\vec{\tau}^{B} = -\mathbf{K}_e \vec{e}_{BL,e} + \mathbf{K}_\omega \vec{\omega}^{B}_{BL,e}\,, \tag{38}$$

where $\mathbf{K}_e$ and $\mathbf{K}_\omega$ are 3×3 diagonal, positive definite gain matrices on the vector portion of the quaternion error, and the angular velocity error, respectfully.

The errors in the joint states are also regulated with a PID controller:

$$\vec{\tau}_J = \mathbf{K}_{p,j}\vec{\Theta}_e + \mathbf{K}_{i,j}\int_0^t \vec{\Theta}_e\, dt + \mathbf{K}_{d,j}\dot{\vec{\Theta}}_e\,, \tag{39}$$

where $\mathbf{K}_{p,j}$, $\mathbf{K}_{i,j}$, and $\mathbf{K}_{d,j}$ are 4×4 diagonal, positive definite gain matrices on the joint angle, integral of joint angle, and joint angular speed errors, respectfully. As mentioned in Section 3.1, the controls are bounded, and therefore the requested controls pass through a saturation block before feeding into the dynamics.

3.3.5.1 Controller Tuning

The values of the controller gains vary with the scenario, although effort is made to minimize the number of different sets of gains. Ultimately, one set of gains is used when relatively large translations are performed, and another set is used when the CubeSat is within one meter of having the LVLH origin in its reachable workspace. In both cases the gains are tuned manually. In some instances the response is very sensitive to changes in the gain. In others, one set of gains that produced an acceptable response for a given set of initial conditions yields an unacceptable response when the initial conditions change. In these instances, the process requires extensive trial and error. The general process follows the steps below:

1. $\mathbf{K}_p$ and $\mathbf{K}_d$ are set to identity, while $\mathbf{K}_i$ is set to zero.

2. Simulation is ran to observe the error responses.

3. $\mathbf{K}_p$ and $\mathbf{K}_d$ are increased or decreased by one order of magnitude. Generally $\mathbf{K}_d$ is one order of magnitude greater than $\mathbf{K}_p$ to yield responses with no overshoot.

4. Repeat Steps 2 and 3 until the error responses are acceptable in terms of stability, settling time, and overshoot.

5. Set the integral gain to roughly $\mathbf{K}_d \times 10^{-6}$ and repeat Step 2.

6. Adjust $\mathbf{K}_i$ until the steady state error is acceptable.

7. If settling time or overshoot have worsened, make small adjustments to $\mathbf{K}_p$ and $\mathbf{K}_d$ to retain those response characteristics.

8. Make small adjustments to all gains to improve the error response.

Step 8 can be repeated indefinitely. However, in this research the gains are only fine-tuned a handful of times as the intent is to estimate the level of controls needed for a robotic servicing CubeSat, rather than to optimize the performance of a notional system.

3.4 Simulation Scenarios

Three different scenarios are simulated to analyze the performance of the system. The first scenario models the process of deploying the CubeSat robot from the chief spacecraft, stabilizing the attitude of the CubeSat, rendezvousing with the chief, and deploying the manipulator to dock with the chief. The second and third scenarios involve performing prescribed motions with the manipulator while attempting to keep the CubeSat base stationary in the LVLH frame.

In all scenarios the chief reference orbit is that of the ISS. The orbital parameters are based off the following two-line element (TLE) set taken from the Amateur Radio on the ISS live tracker [64]:

ISS (ZARYA)
1 25544U 98067A 21313.73829955 .00004352 00000-0 87154-4 0 9991
2 25544 51.6454 343.0140 0003216 189.4812 283.0096 15.48986629311154

The mean motion, eccentricity, inclination, right ascension of the ascending node, and argument of perigee are taken directly from the TLE. The mean anomaly at $t = 0$ is selected randomly from a uniform distribution over 0 to 2π.

3.4.1 Scenario 1: Rendezvous and Docking

In the first scenario, the CubeSat robot is deployed from the chief spacecraft, and must match the position and velocity of the chief while placing the end effector on the LVLH origin. The initial state of the CubeSat robot is as follows:

- $\vec{\rho}^{L} = [0, 0, 0]^T$ m (base begins at the LVLH origin)
- $\underline{e}_{BL} = [1, 0, 0, 0]$ (base aligned with LVLH frame)
- $\vec{\Theta} = [0,\ \pi,\ -\pi,\ 0]^T$ rads (the stowed manipulator configuration from Figure 4)
- $\dot{\vec{\rho}}^{L} = [0, \dot{y}, 0]^T$ m/s (non-zero relative velocity along the in-track direction of the LVLH frame)
- $\vec{\omega}^{B}_{BL} = [\omega_x, \omega_y, \omega_z]^T$ rads/s (non-zero relative angular velocity between the base and LVLH frames)
- $\dot{\vec{\Theta}} = [0, 0, 0, 0]^T$ rads/s (manipulator is stationary)

Notice that the initial attitude and position are expressed relative to the LVLH frame. Necessary transformations are performed to express these in the inertial frame as required by the dynamics function.

$\dot{y}_0$ varies between -0.5 and -2 m/s, and the components of $\vec{\omega}_{BL}^{B}$ vary from 2 to 5°/s. These values are chosen as they are the quoted deployment velocities and tip-off rates in the Nanoracks CubeSat Deployer Interface Definition Document [59].

The rendezvous and docking maneuver is split into two phases. In the first phase the manipulator remains in the stowed configuration while the base stabilizes its attitude and approaches the rendezvous position. When the base is within 1 m of the desired final position and remains there for 10 seconds, it switches to phase two, deploying the manipulator to a predetermined docking configuration and regulating all remaining errors in the states.

The desired final states of the CubeSat robot for phase two in this scenario are:

- $\vec{\rho}^{L} = [0.0207, -0.4138, -0.0207]^T$ m (predetermined position)
- $\underline{e}_{BL} = [1, 0, 0, 0]$ (base aligned with LVLH frame)
- $\vec{\Theta} = [45°, 146.25°, -45°, -67.5°]^T$ (predetermined docking configuration)
- $\dot{\vec{\rho}}^{L} = [0, 0, 0]^T$ m/s (stationary in the LVLH frame)
- $\vec{\omega}_{BL}^{B} = [0, 0, 0]^T$ rads/s (zero relative angular velocity between the base and LVLH frames)
- $\dot{\vec{\Theta}} = [0, 0, 0, 0]^T$ rads/s (manipulator is stationary)

The final state of the CubeSat robot is chosen to place the end effector on the LVLH origin, with the base attitude aligned with the LVLH frame. The docking configuration for the manipulator, $\vec{\Theta}$, is chosen to maximize the manipulability while minimizing the distance of the end effector from the y-axis of the base frame, d. Maximizing manipulability is desired so that the CubeSat robot can achieve relatively fast velocities and large forces for the end effector. Minimizing d is desired because for subsequent scenarios the base is assumed fixed in the LVLH frame, and the end

effector reachable workspace is considered the circle in the xz-plane, centered on the LVLH origin, with maximum radius. In other words, docking configurations with end effector positions far from the y-axis of the base frame have smaller reachable workspaces in the LVLH frame.

In order to determine the docking configuration for the manipulator, over one million configurations are considered, spanning all combinations of joint angles. For each configuration, the forward kinematics are calculated to determine if the configuration is acceptable, defined as meeting the following criteria:

1. $y_{EE}^B \geq y_{EE,min}^B = 0.3\,m$
2. $y_{EE}^B > y_{J3}^B$
3. $y_{EE}^B > y_{J4}^B$

These criteria omit configurations where the end effector is too close to the CubeSat base, or is closer to the base than either joint 3 or 4. The edge of the chief spacecraft is assumed to lie in the $+y$ hemisphere of the LVLH frame, and therefore no portion of the CubeSat robot can extend beyond the $y = 0$ plane. Thus, end effector positions with $y_{EE}^B < 30$ cm would place the leading edge of CubeSat base < 20 cm from the border of the chief, which is only 5 cm more than the lengths of links 2 - 4, making it difficult for the manipulator to move freely. If 2. and 3. above were not true, that would permit docking configurations with either link 2, 3, or 4 penetrating the wall of the chief spacecraft.

For all acceptable configurations, the distance of the end effector from the y axis of the base frame, d, is computed. Those with distances of $1 - 15$ cm are selected, and the manipulability measure is computed from Equation 16, reproduced here:

$$w = \sqrt{\det\left(\mathbf{J}\mathbf{J}^T\right)}\,. \tag{40}$$

Given the design of the manipulator, configurations with the end effector exactly on the y axis of the base frame have a singular Jacobian, and therefore zero volume for the manipulability ellipsoid as shown in Figure 6. This makes intuitive sense since, in such a configuration, rotation of joint 1 would produce zero velocity of the end effector. For this reason, any configuration with $d < 1$ cm is omitted. Finally, a score is calculated for each remaining configuration as:

$$S = \frac{w}{d}\,. \tag{41}$$

Equation 41 is chosen for simplicity, since large w and small d increase the score, and the configuration with the maximum score is selected. Figures 6 - 8 below show the CubeSat robot in several configurations with the resulting manipulability ellipsoid in green. The ellipsoids are computed in the same manner discussed in Section 2.3.3, assuming constant unit norm joint speeds. The physical dimensions of the CubeSat robot are expressed in meters, while the ellipsoid has dimensions of meters per second.

Notice in Figure 6 the manipulability ellipse lies in the xy-plane, since no joint movement can induce end effector velocity in the z direction. The manipulability measure tends to increase as the manipulator moves further from the base y-axis, as seen in Figure 7, since joint 1 speed can then induce larger end effector speeds.

Given that manipulability and distance from the base y axis are proportional, a balance must be struck for the docking configuration, as seen in Figure 8. This configuration maximized the score in Equation 41, having decent manipulability and a short end effector distance from the base y axis, and thus a large workspace.

The controller gains for each phase of the scenario are shown in Table 3. Notice the gains for the attitude and joint controllers are the same in each phase, and the translational controls are all larger in phase 2 than in phase 1. This is expected since the magnitude of the translational errors are much larger in phase 1, and therefore a smaller gain is required to induce a sensible control response.

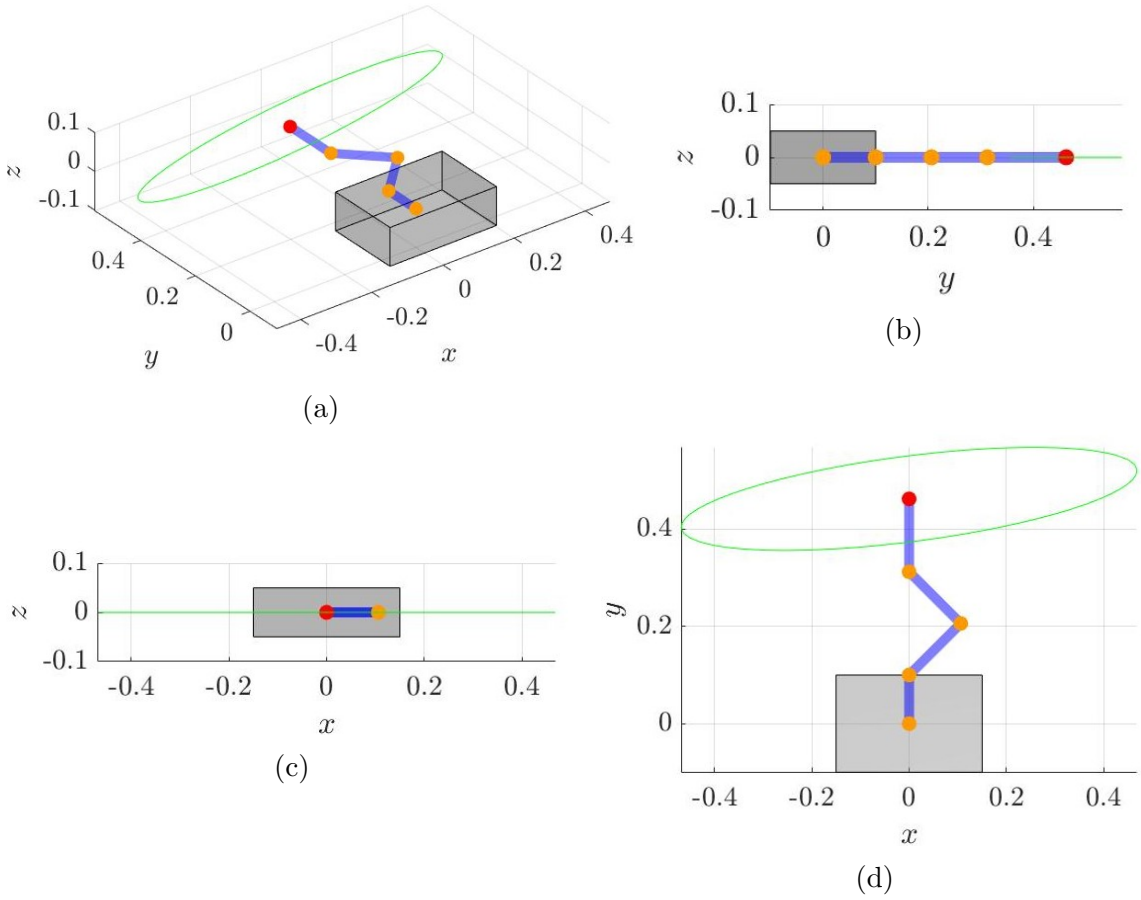

Figure 6: CubeSat robot with manipulability ellipsoid in a singular configuration: $\vec{\Theta}^T = [0°, 45°, 90°, -45°]$

Table 3: Controller Gains for Scenario 1

Gain	**Phase 1**	**Phase 2**
$\mathbf{K}_{p,t}$	$0.4 \times \mathbf{I}_{(3\times3)}$	$20 \times \mathbf{I}_{(3\times3)}$
$\mathbf{K}_{i,t}$	$10^{-6} \times \mathbf{I}_{(3\times3)}$	$10^{-3} \times \mathbf{I}_{(3\times3)}$
$\mathbf{K}_{d,t}$	$40 \times \mathbf{I}_{(3\times3)}$	$200 \times \mathbf{I}_{(3\times3)}$
$\mathbf{K}_{e}$	$\mathbf{I}_{(3\times3)}$	$\mathbf{I}_{(3\times3)}$
$\mathbf{K}_{\omega}$	$\mathbf{I}_{(3\times3)}$	$\mathbf{I}_{(3\times3)}$
$\mathbf{K}_{p,j}$	$\mathbf{I}_{(4\times4)}$	$\mathbf{I}_{(4\times4)}$
$\mathbf{K}_{i,j}$	$10^{-3} \times \mathbf{I}_{(4\times4)}$	$10^{-3} \times \mathbf{I}_{(4\times4)}$
$\mathbf{K}_{d,j}$	$\mathbf{I}_{(4\times4)}$	$\mathbf{I}_{(4\times4)}$

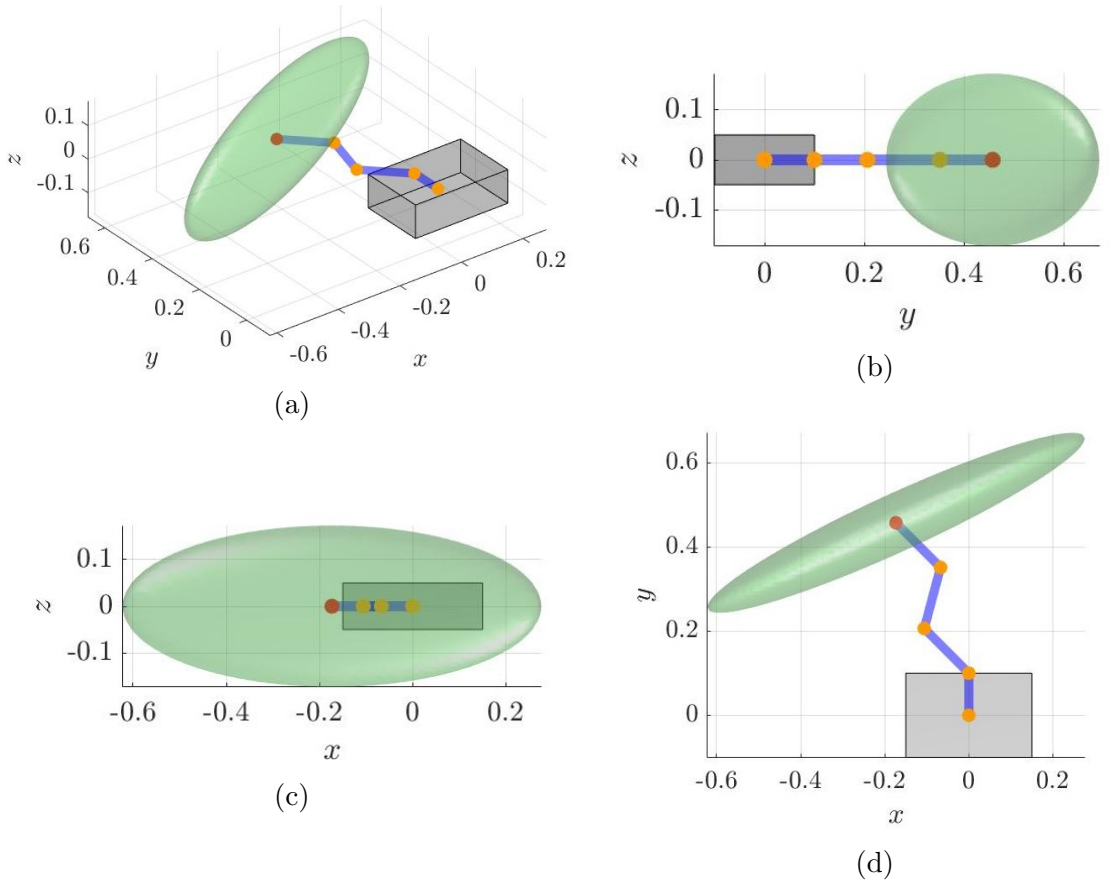

Figure 7: CubeSat robot in a manipulable configuration: $\vec{\Theta}^T = [0°, 135°, -60°, 60°]$

Ten simulations are ran for scenario 1, each with different values of ejection speeds and tip-off rates for the CubeSat. In the best case scenario, $\dot{y}_0 = -0.5\,m/s$, and $\vec{\omega}^B_{BL} = 2°/s \times [1,1,1]^T$, which are the lower bounds for the Nanoracks CubeSat Deployer [59]. In the worst case scenario, $\dot{y}_0 = -2\,m/s$, and $\vec{\omega}^B_{BL} = 5°/s \times [1,1,1]^T$. Values between the best and worst cases, with equal increments, are simulated.

The best and worst case scenarios are also re-run with random disturbance forces and torques added to each individual feedback control. The random disturbances are sampled from a standard normal distribution at a rate of $t_s = 0.1$ s, and scaled by either 1, 5 or 10% of the respective maximum control. For example, in the 10% case the random disturbance in each of the thrust axes is sampled from $\mathcal{N}(\mu, \sigma^2) = \mathcal{N}(0, 1)$, then multiplied by 0.1×0.1 N $= 0.01$ N, with the effect of sampling from $\mathcal{N}(0, 10^{-4})$.

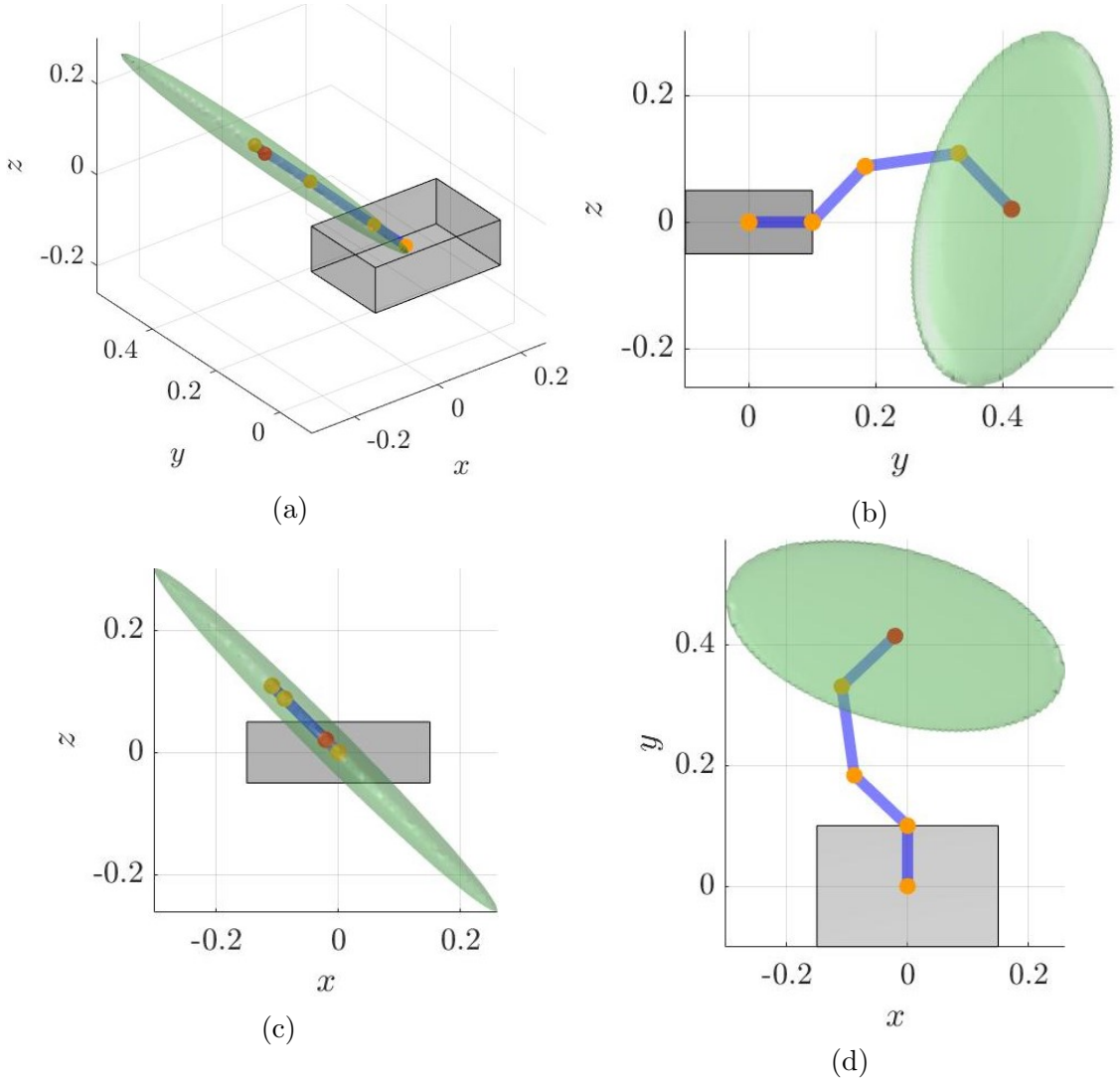

Figure 8: CubeSat robot with manipulability ellipsoid in docking configuration: $\vec{\Theta}^T = [45°, 146.25°, -45°, -67.5°]$

This is a generous estimate of the magnitude of disturbance forces expected on orbit. Solar radiation pressure and atmospheric drag would impart forces on the order of 10^{-6} and 10^{-5} N, respectively, to a CubeSat in LEO, according to approximate models [65]. $J2$ perturbations are larger, with magnitudes as high as 10^{-2} N according to calculations from $J2$ models [66, 67]. However, both the chief and CubeSat would experience similar effects from $J2$ perturbations due to their close proximity, so the effect over the duration of the engagement scenario would be negligible.

The metrics used to analyze the system performance in this scenario are the settling time, defined as the time required for the end effector to reach the origin of the LVLH frame and remain within ± 1 mm of error, and total control usage from $t = 0$ to t_{settle}. For the disturbed cases, statistics on state errors and controls after t_{settle} are computed.

3.4.2 Scenario 2: End Effector Placement

In scenario 2 the CubeSat robot begins in the docking configuration, or final state, from scenario 1, with the end effector at the origin of the LVLH frame, and is tasked with moving the end effector to a random location in its reachable workspace while keeping the base attitude and position fixed in the LVLH frame.

The reachable workspace is considered to be a circle in the $y = 0$ plane of the LVLH frame with maximum radius that the end effector can reach with the base fixed at the docking position. The maximum radius is calculated with the Pythagorean theorem, using the maximum reach of links $2 - 4$ and the base LVLH position:

$$\begin{aligned} R_{max} &= \sqrt{(0.45)^2 - (y_B^L - 0.1)^2} - \sqrt{(x_B^L)^2 + (z_B^L)^2}\,, \\ R_{max} &= \sqrt{(0.45)^2 - (0.3138)^2} - \sqrt{(0.0207)^2 + (0.0207)^2}\,, \\ R_{max} &= 0.2933\,\mathrm{m}\,. \end{aligned} \tag{42}$$

The new desired position of the end effector is chosen by selecting a random radius R from a uniform distribution between 0 and R_{max}, and a random phase angle ϕ from a uniform distribution between $-\pi$ and π. Then the desired end effector position is:

$$\vec{r}_{EE,d}^{\,L} = \begin{bmatrix} R\cos\phi \\ 0 \\ R\sin\phi \end{bmatrix}. \tag{43}$$

The objective of the task is to place the end effector at the desired location with the system at rest. Rather than specifying the new joint angles for the manipulator, the end effector trajectory is generated using the De Casteljau algorithm on $SE(3)$ described in Chapter II, and inverse differential kinematics is used to determine the joint angles and angular speeds required to achieve the end effector trajectory. As mentioned in Section 3.1, the orientation of the end effector is ignored, and only the position portion of the interpolating curve from the De Casteljau algorithm is used. The desired initial and final velocity for the algorithm is zero.

Once the end effector trajectory is determined, the desired joint speeds are computed at each time step by:

$$\dot{\vec{\Theta}} = \mathbf{J}^{\dagger}\vec{v}_{EE}\,, \tag{44}$$

where $\mathbf{J}$ is the 3×4 analytical Jacobian that converts joint speeds to end effector velocities in the base frame, and $\mathbf{J}^{\dagger}$ is the Moore-Penrose pseudo-inverse. The 3×4 Jacobian assumes that the base is fixed in the LVLH frame, and thus velocities in the LVLH frame are identical to those in the base frame. This assumption is valid as long as the controller can maintain the desired base pose. Once $\dot{\vec{\Theta}}$ is calculated, the desired joint angles are simply:

$$\vec{\Theta} = \vec{\Theta}_0 + \int_0^t \dot{\vec{\Theta}}\, dt\,. \tag{45}$$

The controller gains used for this scenario are shown in Table 4. The translational gains are the same as those from phase 2 of scenario 1. The gains on the attitude and joint controllers are all greater than those of scenario 1. This makes intuitive sense since the errors in every state are expected to be smaller than those in scenario 1, and so the gains must be higher to produce an adequate control response and maintain near-zero error in these states.

Table 4: Controller Gains for Scenario 2

Gain	**Value**
$\mathbf{K}_{p,t}$	$20 \times \mathbf{I}_{(3\times3)}$
$\mathbf{K}_{i,t}$	$10^{-3} \times \mathbf{I}_{(3\times3)}$
$\mathbf{K}_{d,t}$	$200 \times \mathbf{I}_{(3\times3)}$
$\mathbf{K}_{e}$	$10 \times \mathbf{I}_{(3\times3)}$
$\mathbf{K}_{\omega}$	$300 \times \mathbf{I}_{(3\times3)}$
$\mathbf{K}_{p,j}$	$100 \times \mathbf{I}_{(4\times4)}$
$\mathbf{K}_{i,j}$	$10^{-2} \times \mathbf{I}_{(4\times4)}$
$\mathbf{K}_{d,j}$	$100 \times \mathbf{I}_{(4\times4)}$

Ten different points in the workspace are selected, and three simulations are ran for each point. The three simulations differ in the desired time to perform the maneuver, as summarized in Table 5. The manipulator attempts to traverse the interpolated curve connecting the initial position of the end effector to its desired position in either 5, 10, or 15 seconds, and an additional 5 seconds is granted for the system to null out any disturbances to the base caused by the manipulator motion.

Table 5: Scenario 2 Simulation Times

Simulation #	**Maneuver Time (s)**	**Settling Time (s)**	**Total Time (s)**
1	5	5	10
2	10	5	15
3	15	5	20

For the 10 second cases, the simulations are re-ran with random disturbance forces produced in the same manner as in scenario 1, except the sampling time is decreased to $t_{samp} = 0.01$ s. The metrics used to analyze the system performance in this scenario are total control usage and the magnitude of the end effector position error at t_{final}.

3.4.3 Scenario 3: End Effector Path Tracing

As in scenario 2, scenario 3 begins with the CubeSat robot in the docking configuration, or final state, from scenario 1, and it is then tasked with tracing a circular path, counterclockwise at a constant speed, in the $y = 0$ plane of the LVLH frame with the end effector while keeping the base attitude and position fixed. The circular path is centered on the origin, and its radius, R, is selected randomly from a uniform distribution between 10 and 29 cm, or just under the maximum possible radius.

In order to trace the prescribed circular path, the end effector must first be moved from the origin to the path. To accomplish this an entry point is determined randomly by selecting a phase angle, ϕ, from a uniform distribution between $-\pi$ and π. The entry point is then:

$$\vec{r}_{entry} = \begin{bmatrix} R\cos\phi \\ 0 \\ R\sin\phi \end{bmatrix}. \tag{46}$$

To eliminate instantaneous changes in the desired velocity of the end effector, it is desired to match the velocity at the entry point between the circular path and the entry trajectory. The desired position of the end effector during the circular motion is:

$$\vec{r}_{circ} = \begin{bmatrix} R\cos\left[\omega(t - t_{entry}) + \phi\right] \\ 0 \\ R\sin\left[\omega(t - t_{entry}) + \phi\right] \end{bmatrix} = \begin{bmatrix} R\cos\left[\dfrac{2\pi}{T_{circ}}(t - t_{entry}) + \phi\right] \\ 0 \\ R\sin\left[\dfrac{2\pi}{T_{circ}}(t - t_{entry}) + \phi\right] \end{bmatrix}, \tag{47}$$

where t_{entry} is the time it takes for the end effector to move from the origin to the circular path, and T_{circ} is the desired period of the circular motion. The velocity of the end effector at the entry point ($t = t_{entry}$) is then:

$$\vec{v}_{entry} = \begin{bmatrix} -R\left(\frac{2\pi}{T_{circ}}\right)\sin\phi \\ 0 \\ R\left(\frac{2\pi}{T_{circ}}\right)\cos\phi \end{bmatrix}. \tag{48}$$

The initial state of the end effector and the desired entry position and velocity are used in the De Casteljau algorithm to produce a smooth entry trajectory from the origin to the circular path. The resulting trajectory is combined with the circular path and the desired joint speeds and angles are then computed with the Jacobian pseudo-inverse, as in scenario 2. The controller gains used in scenario 3 are identical to those in scenario 2.

Ten different circular paths are selected, and three simulations are ran for each path. The three simulations differ in the desired time to perform the maneuver, as summarized in Table 6. For the 30 second cases, the simulations are re-ran with random disturbance forces as in scenario 2. The metrics used to analyze the system performance in this scenario are total control usage, and the average effector position error over the length of the simulation.

Table 6: Scenario 3 Simulation Times

Simulation #	**Entry Maneuver Time (s)**	**Circular Motion Period (s)**	**Total Time (s)**
1	5	15	20
2	7.5	22.5	30
3	10	30	40

3.5 Summary

In this chapter the CubeSat robotic system used in subsequent simulations is described and illustrated. The process of deriving the system dynamics is outlined. The simulation framework is introduced, and the various components of the Simulink model are detailed, including the chief orbital dynamics, the dynamics and kinematics of the CubeSat robot, the relative states and state errors, and the controller. Three representative scenarios that are simulated in this study are described in detail, including the overall concept of each, the assumptions and goals, how the desired states are determined, the controller gains used, and how the simulation campaign is conducted. The results of the simulations are presented and discussed in the following chapter.

IV. Results and Analysis

4.1 Scenario 1: Rendezvous and Docking

In the absence of disturbance forces the settling time and total control usage for each simulation are shown in Table 7, where F_B is the total impulse and is calculated as follows:

$$F_B = \int_0^{t_{settle}} |f_{B,x}| + |f_{B,y}| + |f_{B,z}| \, dt \,, \tag{49}$$

and T_B is the total angular impulse, calculated the same as Equation 49 except with τ_B replacing f_B.

Table 7: Rendezvous and Docking Results

$\dot{y}_0$ (m/s)	$\omega^B_{BL,0}$ (°/s/ax)	t_{settle} (s)	F_B (N·s)	T_B (N·m·s)	T_1 (N·m·s)	T_2 (N·m·s)	T_3 (N·m·s)	T_4 (N·m·s)
-0.5000	2.0000	431.31	13.9372	0.2931	0.0294	0.1594	0.3413	0.0980
-0.6667	2.3333	520.08	19.7873	0.3387	0.0296	0.1873	0.4532	0.1260
-0.8333	2.6667	596.37	26.2911	0.3891	0.0299	0.2176	0.5745	0.1563
-1.0000	3.0000	665.10	33.3188	0.4427	0.0303	0.2496	0.7024	0.1883
-1.1667	3.3333	728.68	40.7984	0.4989	0.0306	0.2927	0.8351	0.2555
-1.3333	3.6667	788.45	48.6581	0.5567	0.0309	0.3167	0.9712	0.2555
-1.5000	4.0000	844.15	56.8528	0.6184	0.0313	0.3513	1.1099	0.2901
-1.6667	4.3333	892.30	65.3841	0.6860	0.0317	0.3864	1.2505	0.3253
-1.8333	4.6667	924.34	74.1956	0.7588	0.0321	0.4219	1.3926	0.3608
-2.0000	5.0000	903.76	83.2816	0.8371	0.0324	0.4576	1.5359	0.3966

The settling times vary from about 7.2 minutes in the best case scenario to about 15.4 minutes in the worst. Interestingly, the worst case settling time occurred for the second to largest ejection speed. This is likely attributable to the increased errors inducing a larger control response for the largest ejection speed, causing the system to settle faster. These times are reasonable but may be longer than desired in a real world system. However, these ejection speeds were taken from the NanoRacks CubeSat Deployer, which is meant to eject the CubeSat so that it drifts a safe distance

from the ISS. A berthing system for a CubeSat meant to rendezvous with the chief spacecraft could produce a much smaller ejection speed, allowing the CubeSat to rendezvous and dock faster and with less fuel.

The total impulse, F_B, and total angular impulse, T_B, increase almost linearly from the best to worst cases, as shown in Figure 9. The values are well within the capabilities of COTS propulsion and ADACS subsystems for CubeSats [7].

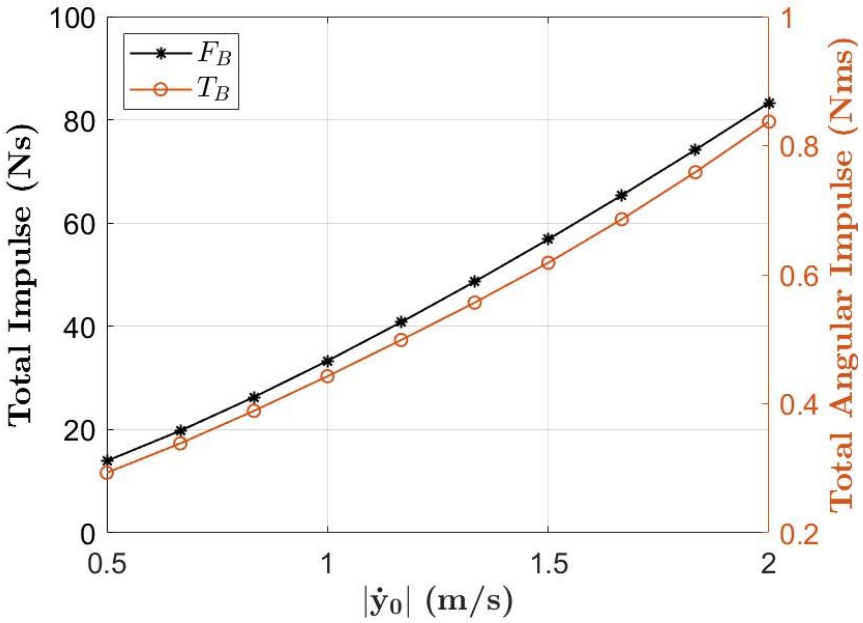

Figure 9: Total impulse and angular impulse applied to the base versus ejection speed.

The total angular impulses applied to the joints also increase almost linearly with ejection speed, as shown in Figure 10. The increase in total angular impulse on joint 1 is nearly indistinguishable, as the reaction torque on joint 1 due to the in-track acceleration of the base is zero. Therefore the increase is attributable solely to the increase in the base tip-off rate at ejection, which is slight. Joints 2 - 4 see a much steeper increase in total angular impulse because as thrust is applied to the base in the in-track direction, these joints must apply torque in order to maintain the stowed configuration from Figure 4. As the thrust duration increases with increased ejection speed, the total angular impulse to these joints increases proportionately.

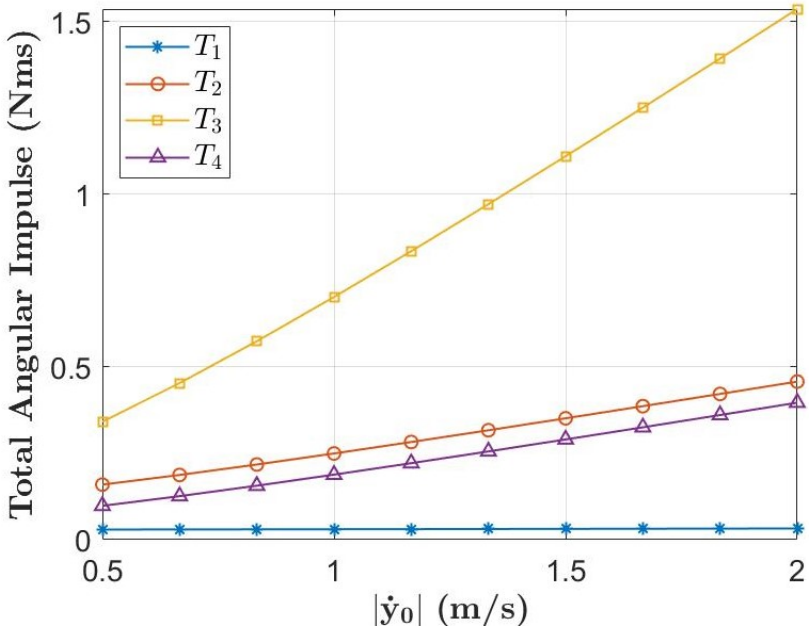

Figure 10: Total angular impulse applied to the joints versus ejection speed.

For all cases in this scenario the state error responses are reminiscent of a first order system, where after a short period of growth the state errors decay to zero with no overshoot. This is intentional as overshoot in this scenario is akin to colliding with the chief spacecraft. Figure 11 shows the end effector position error in the LVLH frame over time for the best and worst case ejection speeds. In all cases the CubeSat base does not drift far from the chief spacecraft - less than 300 m in the worst case scenario. However, the rendezvous takes a significant amount of time due to the bounds on the thrust and the zero-overshoot requirement.

The translational states take the longest time to settle, as expected. In all cases the initial angular velocity at ejection is stabilized within 20 seconds, with no significant effect on the base attitude error. Once the CubeSat crosses the threshold into phase 2 of the maneuver, the manipulator is deployed and the joint angles settle to within 1° of error in less than 8 seconds. The manipulator deployment imparts a torque on the CubeSat base, but the attitude controllers are able to counteract the disturbance with little to no effect on the attitude error.

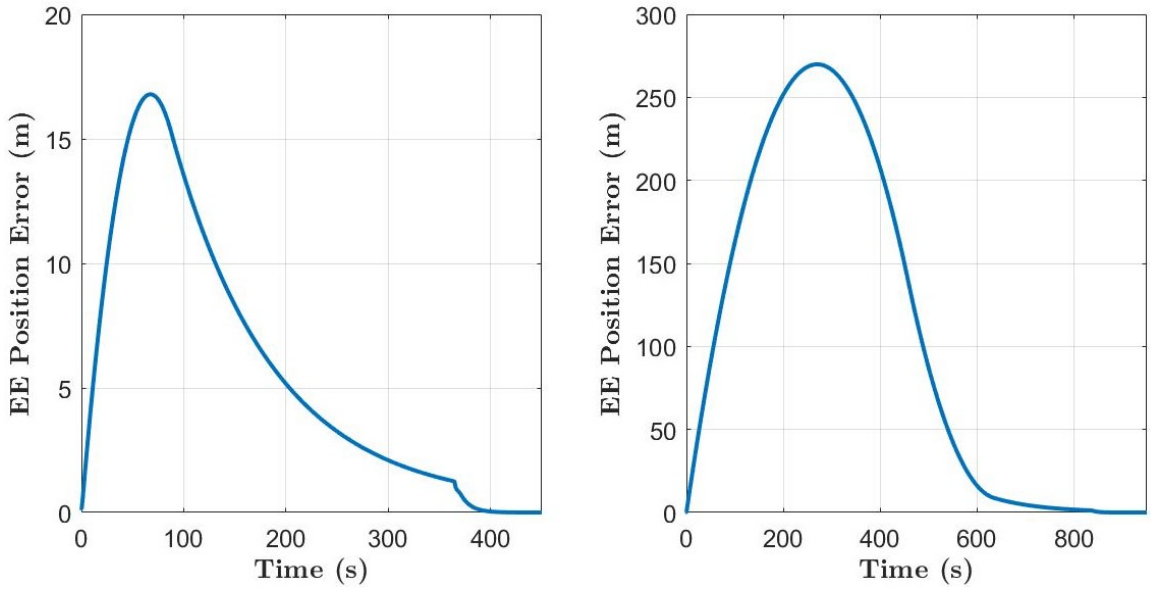

Figure 11: End effector position error in the LVLH frame for the best (left) and worst (right) case ejection speeds

The control histories that produce these responses are reminiscent of bang-bang control, as shown in Figures 12 - 14 for the worst case ejection speed. The control histories and state errors for an intermediate case, with $\dot{y}_0 = -1.667$ m/s, are shown in Appendix A. Each figure includes the control history for the entire rendezvous and docking on the left, with a closer look at the transition into phase 2 of the maneuver on the right.

Figure 12 shows that the thrust is often saturated, particularly in the in-track direction, during the rendezvous. The disturbance imparted to the base from the manipulator motion also causes the thrusters to saturate temporarily. Figure 13 shows that the attitude controls never saturate during the maneuver. A small amount of torque is applied during the rendezvous to counteract the disturbance caused by the thrusters. In phase 2, the magnitude of torque applied to the base to counteract the manipulator motion is relatively much larger, but still only about half the saturation limit.

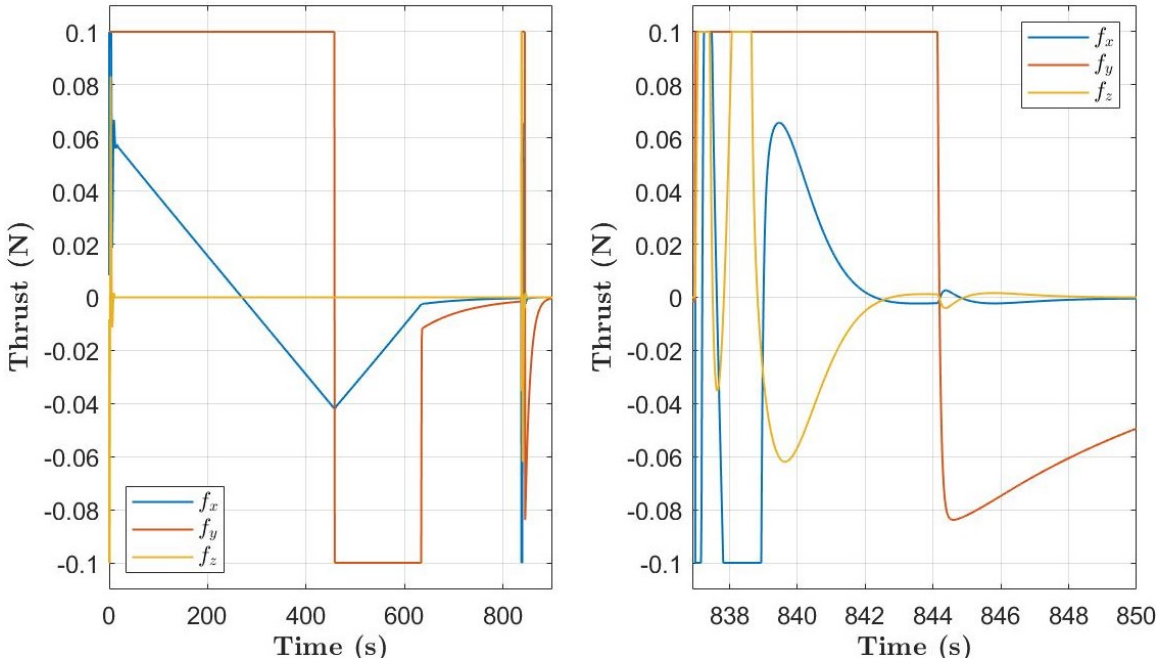

Figure 12: Thrust profile for $\dot{y}_0 = -2$ m/s.

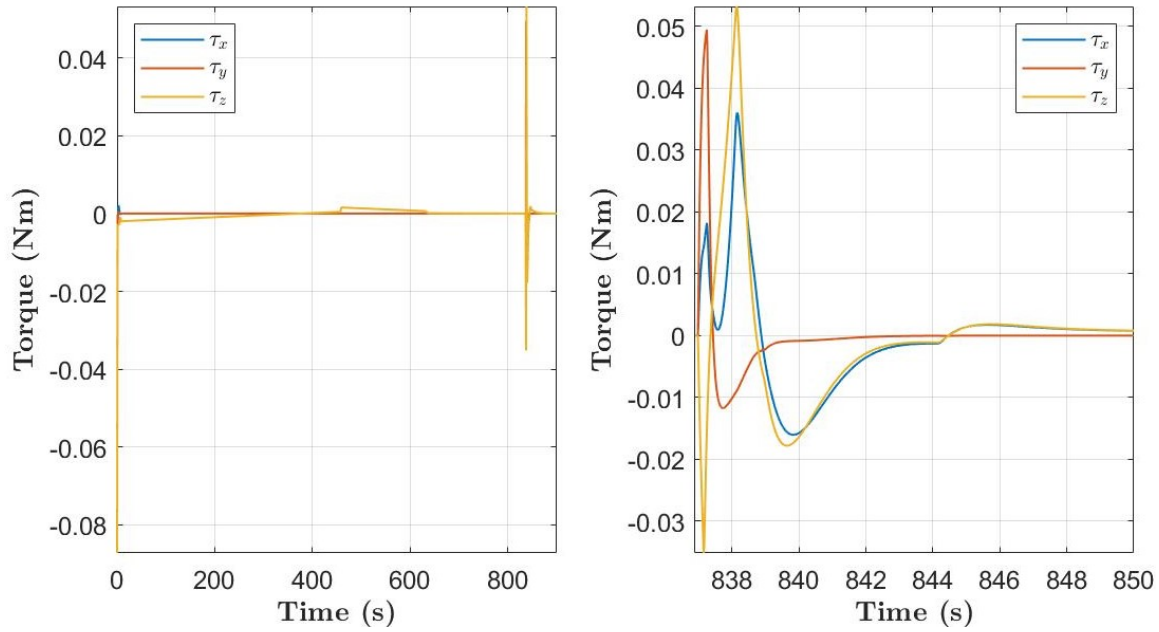

Figure 13: Base torque profile for $\dot{y}_0 = -2$ m/s.

The joint torque profile during the rendezvous in Figure 14 is similar to the attitude controls. Some of the joint torques saturate for several seconds at the beginning of phase 2 before quickly settling.

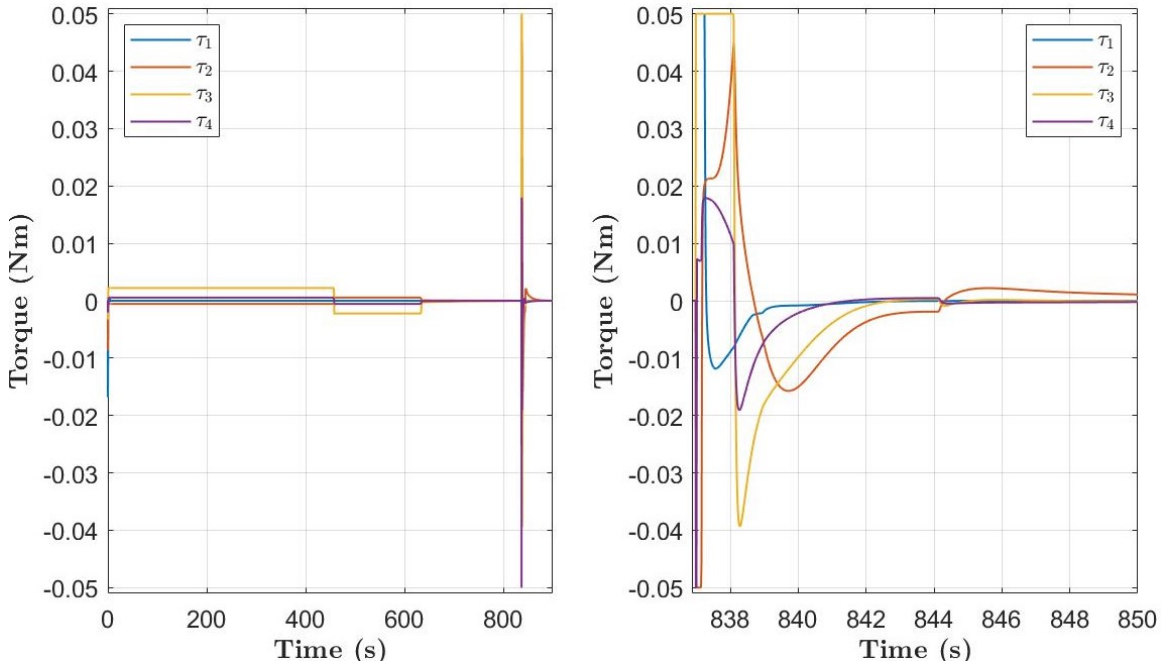

Figure 14: Joint torque profile for $\dot{y}_0 = -2$ m/s.

In the presence of disturbance forces and torques the system remains stable and completes the rendezvous and docking maneuver in nearly the same time as the undisturbed case. Figure 15 shows the end effector position error in the LVLH frame from the best case ejection speed for both the undisturbed and the most heavily disturbed case. Notice on the left side of Figure 15 the error response appears the same between the two cases. As state estimation in the model is ignored, the continuous controller is able to respond to changes in the state errors induced by the disturbance forces near instantaneously, and therefore counteract their effect. This does, however, require more control than in the undisturbed cases.

Table 8 below shows the total control usage for each disturbed case. The percent disturbance column refers to the percent of the maximum control that the random disturbance signal is scaled by. For both cases, increasing this scaling leads to an increase in total control usage for all actuators.

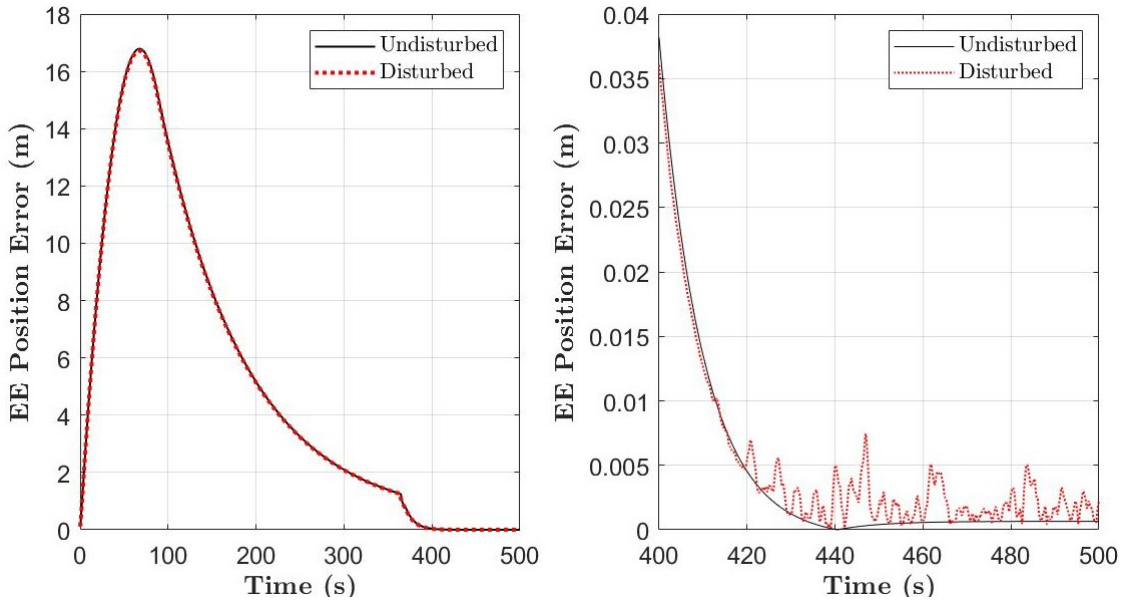

Figure 15: End effector position error for $\dot{y}_0 = -0.5$ m/s.

Table 8: Undisturbed Versus Disturbed Rendezvous and Docking Results

$\dot{y}_0$ (m/s)	$\omega^B_{BL,0}$ (°/s/ax)	% dist	F_B (N·s)	T_B (N·m·s)	T_1 (N·m·s)	T_2 (N·m·s)	T_3 (N·m·s)	T_4 (N·m·s)
-0.500	2.000	0	13.9372	0.2931	0.0294	0.1594	0.3413	0.0980
-0.500	2.000	1	14.3552	0.8095	0.1413	0.2316	0.3903	0.1903
-0.500	2.000	5	16.6159	3.1003	0.5922	0.6440	0.7174	0.6939
-0.500	2.000	10	19.9932	5.9995	1.1565	1.1868	1.1895	1.3466
-2.000	5.000	0	83.2816	0.8371	0.0324	0.4576	1.5359	0.3966
-2.000	5.000	1	95.2071	2.0177	0.2842	0.5197	1.5768	0.4822
-2.000	5.000	5	127.4348	6.9999	1.2426	1.3039	1.8883	1.4324
-2.000	5.000	10	142.1345	13.0859	2.4275	2.4407	2.6907	2.7693

While the settling time is largely unaffected by the disturbances, the system requires non-negligible control after settling in order to counteract the disturbances. Additionally, the error in the end effector position exhibits random fluctuations in the steady state as large as 7 mm, as seen on the right side of Figure 15. Therefore, the total controls in Table 8 are computed with the undisturbed settling times.

The mean and standard deviation of the controls and of the end effector position error after t_{settle} for each of the disturbed cases are shown in Table 9. All axes are considered together in computing mean thrust and mean torque on the base.

Similarly, all joints are considered together when computing the mean torque on the joints. In other words, $\overline{f_B}$ is the mean thrust in each base axis, $\overline{\tau_B}$ is the mean torque about each base axis, and $\overline{\tau_J}$ is the mean torque in each joint.

Table 9: Mean Controls and End Effector Error After t_{settle}

$\dot{y}_0$ (m/s)	$\omega^B_{BL,0}$ (°/s/ax)	% dist	$\overline{f_B} \pm \sigma$ (mN)	$\overline{\tau_B} \pm \sigma$ (mN·m)	$\overline{\tau_J} \pm \sigma$ (mN·m)	$\overline{\lvert\rho_e\rvert} \pm \sigma$ (mm)
-0.500	2.000	1	0.0360 ± 1.5466	0.0418 ± 0.4605	0.0208 ± 0.3691	0.6525 ± 0.1232
-0.500	2.000	5	0.1951 ± 7.7541	0.2060 ± 2.3085	0.1028 ± 1.8422	1.2095 ± 0.6392
-0.500	2.000	10	0.3935 ± 15.5673	0.4115 ± 4.6308	0.2048 ± 3.6837	2.0897 ± 1.4092
-2.000	5.000	1	0.0245 ± 1.6095	0.0283 ± 0.4754	0.0138 ± 0.3821	0.6168 ± 0.1184
-2.000	5.000	5	0.2932 ± 8.1944	0.3041 ± 2.4301	0.1504 ± 1.9338	1.1350 ± 0.5997
-2.000	5.000	10	0.5797 ± 16.4181	0.6028 ± 4.8692	0.2997 ± 3.8640	2.0119 ± 1.2898

Clearly as the disturbances increase in magnitude more control is needed in the steady state to counteract them. The values for $\overline{|\rho_e|}$ indicate that there is some steady state error in the end effector position both with and without disturbances, mostly attributable to errors in the base position. These could be mitigated by increasing the integral gain on the base position error. The standard deviations on the end effector position error are indicative of the errors one could expect for different magnitudes of disturbance forces on orbit.

4.2 Scenario 2: End Effector Placement

Figure 16 shows the randomly-generated desired positions for the end effector, along with the workspace boundary. Each position is given a case number as shown in the legend. Figure 17 shows a typical end effector trajectory generated by the De Casteljau algorithm, in this case for the target position $(x, z) = (0.2691, -0.0592)$ m. Notice the velocity curves are continuous and the position curves are smooth.

Figure 18 shows the end effector position error for each case as a function of the maneuver time. The cases can be traced back to the positions in Figure 16. Recall that for all cases the system was given an additional 5 seconds of settling time. In four out of ten cases the system is able to place the end effector on the desired point within 1 mm of error with a 5 second maneuver. In all but two cases, this error

decreases to 0.1 mm for 10 and 15 second maneuvers. Notice the logarithmic y-axis scale, which is necessary to display all of the results due to the wide range of values, especially for the 5 second maneuvers. For each target location the final end effector position error decreases when the system is given more time to perform the maneuver. The improvement in performance is greater for the cases with relatively high errors.

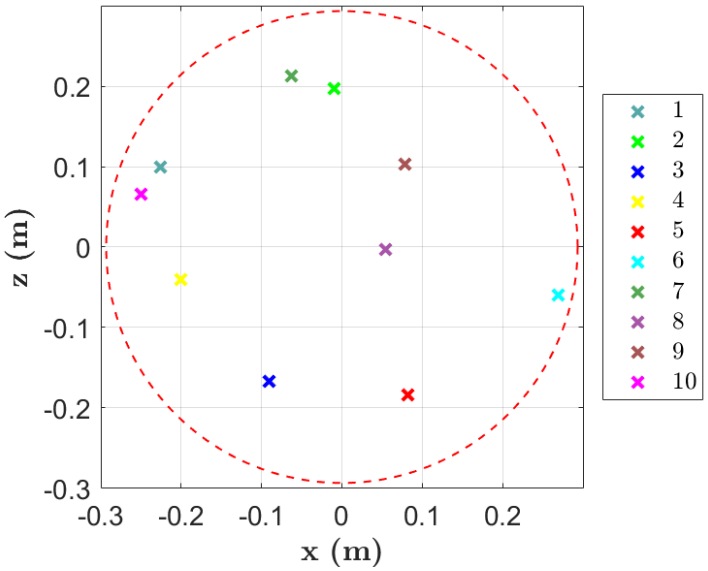

Figure 16: Random locations in the workspace used for scenario 2.

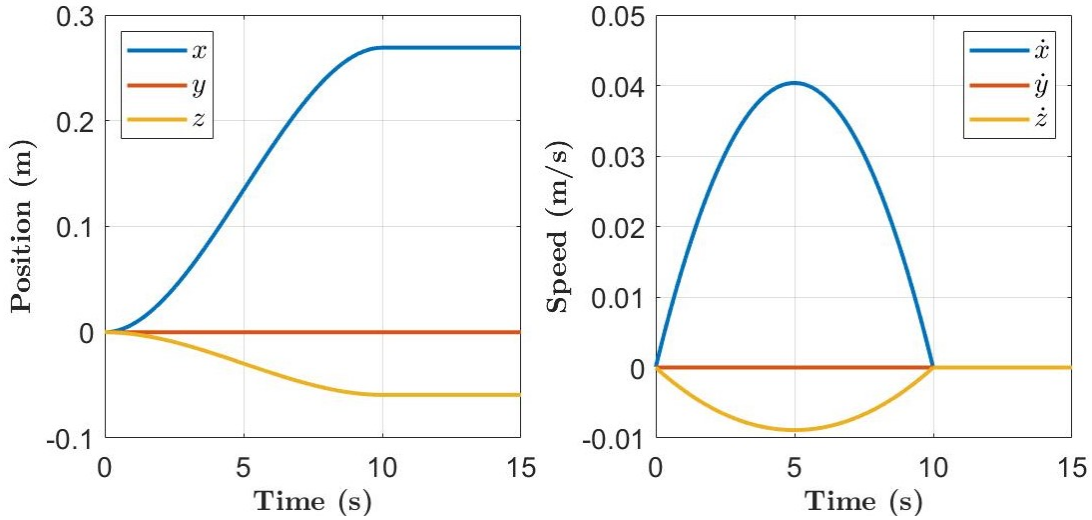

Figure 17: Typical position (left) and velocity (right) profile generated by the De Casteljau algorithm.

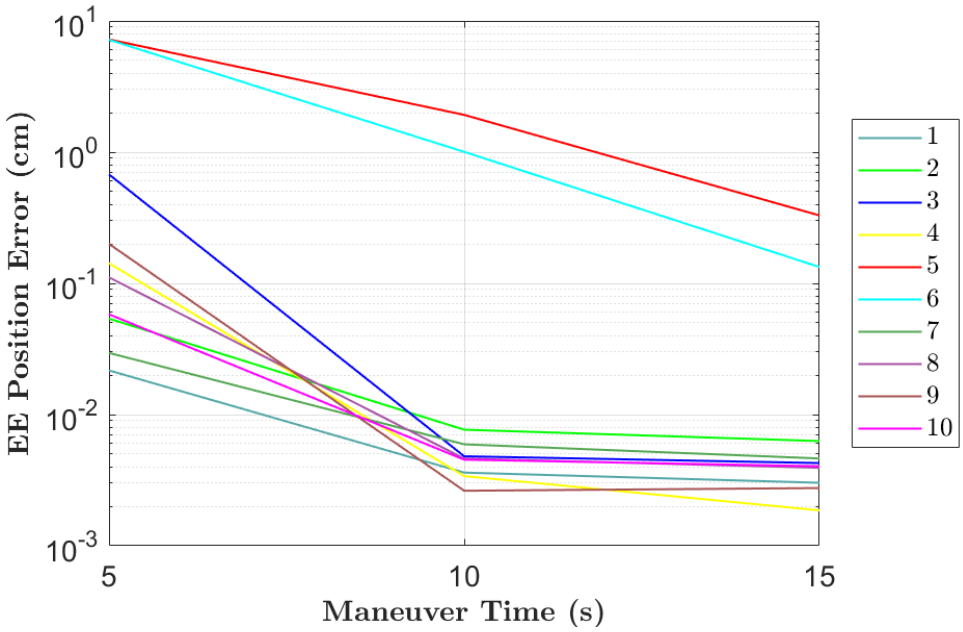

Figure 18: End effector position error versus maneuver time.

Two cases stand out with over 7 cm of position error after the 5 second maneuver. In these cases the joint speeds required to perform the maneuver, given the initial configuration of the system, are much higher than in the other cases. Figure 19 shows the joint speeds for the 5 second repositioning in case six. Joint 1 reaches speeds of over 2.5 rads/s in less than 2 seconds.

The controls required to achieve this maneuver are shown in Figure 20. Nearly all controls are saturated for the first several seconds of the maneuver. The rapid manipulator motion imparts significant disturbances to the base, leading to large control usage. The controllers are unable to completely regulate the state errors, as shown in Figure 21. The base position errors in particular account for several centimeters of error in the end effector position by the end of the maneuver. The base attitude also does not recover from the initial disturbance.

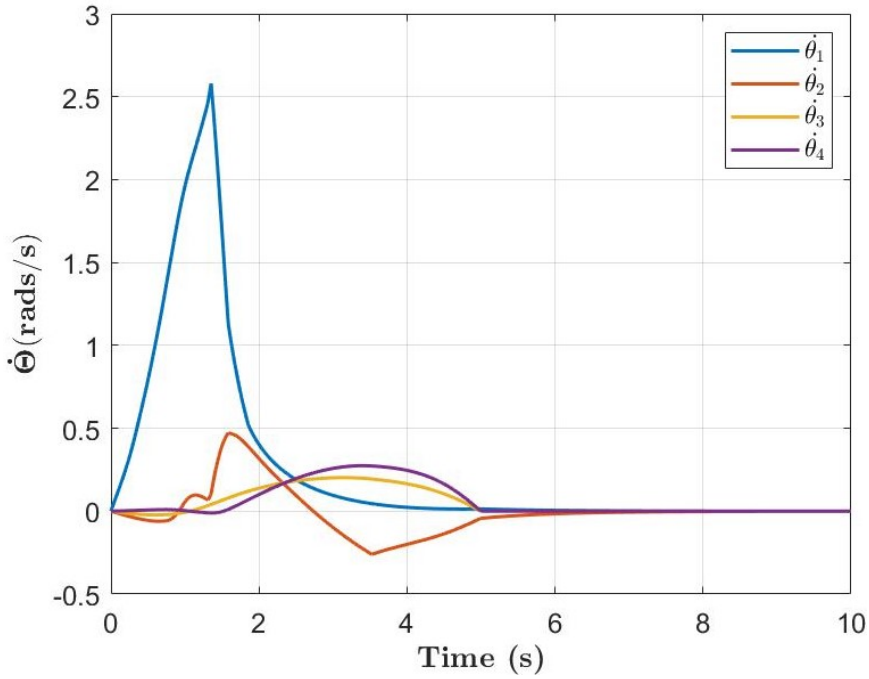

Figure 19: Joint speeds for the 5 second repositioning in case six.

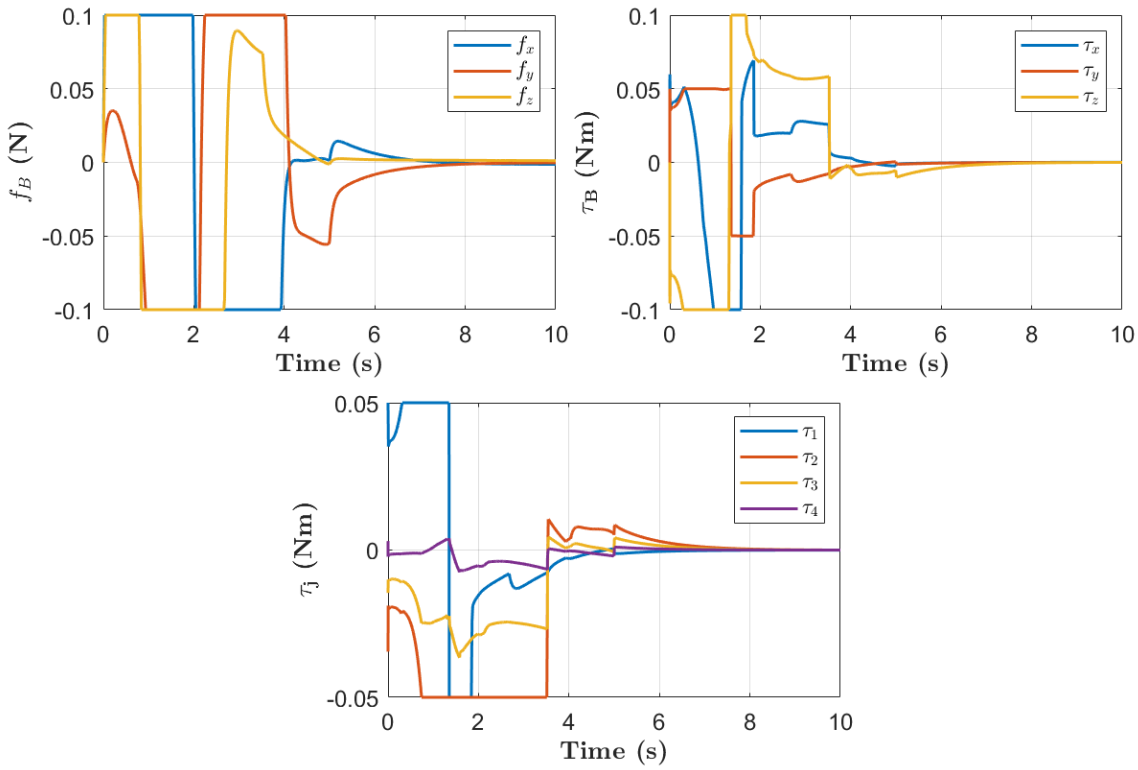

Figure 20: Controls for the 5 second repositioning in case six.

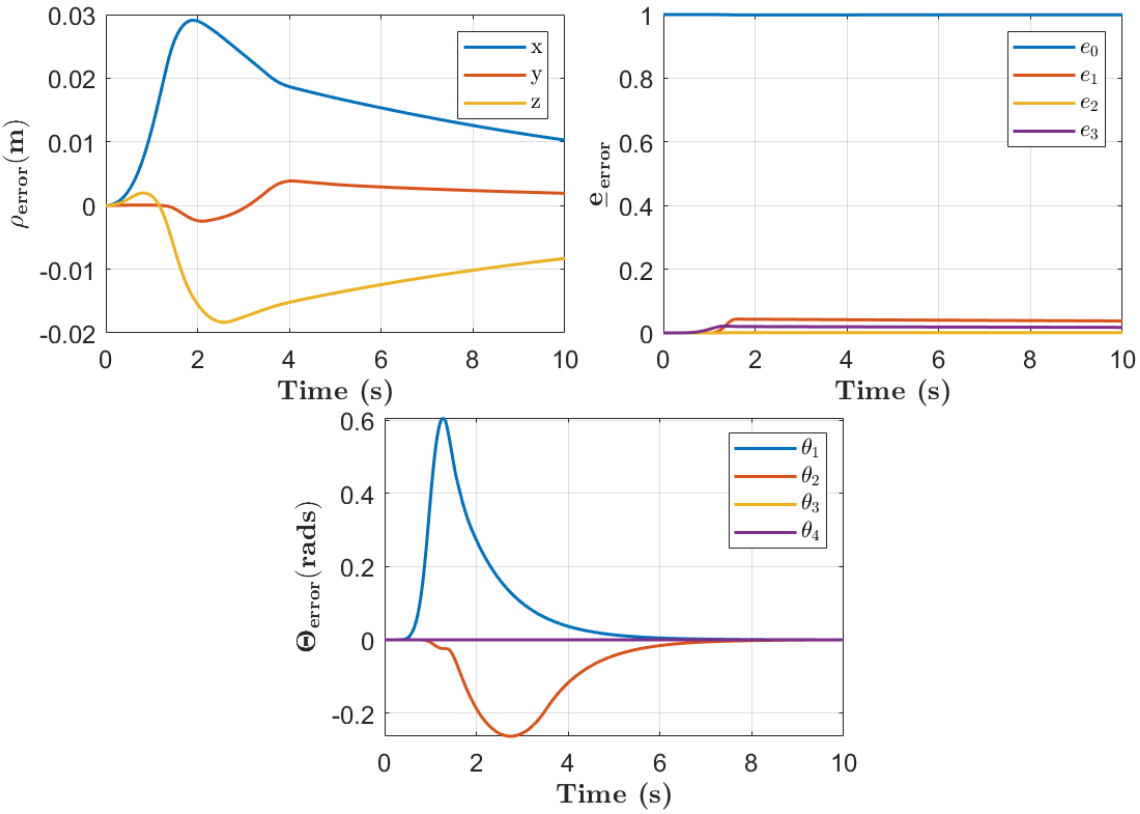

Figure 21: State errors for the 5 second repositioning in case six.

Part of the reason for the aggressive joint trajectories is that the manipulator passes close to a singular configuration near the start of the maneuver, namely a configuration with the end effector lying on the y-axis of the CubeSat body frame. This negative effect could be mitigated by introducing an alternative Jacobian psuedo-inverse that maximizes distance from singular configurations, or by redesigning the system to remove kinematic singularities in the desired workspace.

Ultimately these results are informative for the trajectory planning problem - end effector trajectories that require excessive joint speeds should be avoided in order to achieve high accuracy in end effector positioning. For a specific desired position, ample time should be given to perform the maneuver to limit the joint speeds. The control histories and state errors for a typical case, a 10 second maneuver to the location $\vec{r}_{EE} = [0.0784,\ 0,\ 0.1028]^T$, are shown in Appendix A.

As expected, the control requirements decrease as the maneuver time increases. Table 10 shows the mean and standard deviation of the percent decrease in total impulse and angular impulse relative to the 5 second maneuver. Notice the percent decrease from 5 to 10 seconds is much larger than that from 10 to 15 seconds. This indicates that there is a diminishing marginal benefit of increasing the maneuver time.

Table 10: Mean Percentage Decrease in Control Use Relative to 5 Second Maneuver

	Mean % Decrease		
Maneuver Time (s)	$\overline{\Delta F_B} \pm \sigma$	$\overline{\Delta T_B} \pm \sigma$	$\overline{\Delta T_J} \pm \sigma$
10	40.69 ± 12.06	44.71 ± 6.80	47.38 ± 3.10
15	56.24 ± 15.40	60.90 ± 9.25	63.03 ± 5.87

The final end effector position error is correlated with the total control use, as shown in Figure 22. The total impulse is for all thrust axes, and the total angular impulses are either for all base axes or for all joints. The maneuver time is incorporated in the color and shape of the data points.

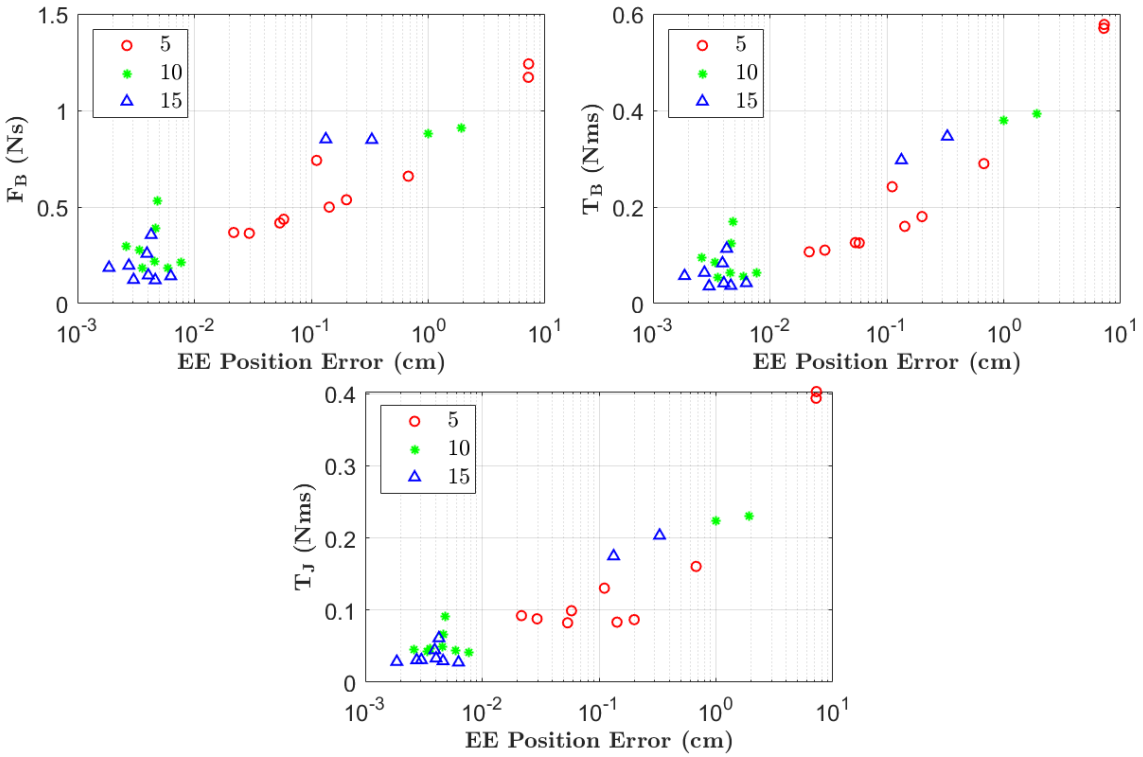

Figure 22: Control use versus end effector position error.

The end effector position error tends to increase exponentially as the control requirements for the maneuver increase. The trend is less noticeable when the control requirements are relatively small, as seen by the cluster of data points for the 10 and 15 second maneuvers. The two outlying cases (cases 5 and 6) indicate that control use is a stronger determinant of end effector position errors than maneuver time. The 10 and 15 second maneuvers for these cases still exhibit larger errors than the 5 second maneuvers for most other cases, since they require more control. Ultimately, maneuvers that require large amounts of control to achieve rapid joint accelerations will induce the largest errors.

Disturbance forces have little effect on the end effector position error at the end of the maneuver. Table 11 shows the average and standard deviation of the difference in end effector position error between the disturbed and undisturbed cases for each magnitude of disturbance. Recall that the disturbances are included for the 10 second maneuvers.

Table 11: Disturbed Versus Undisturbed End Effector Position Error

Disturbance Magnitude (%)	Mean Difference (μm)
1	0.6279 ± 35.6374
5	13.9123 ± 23.0355
10	52.3851 ± 111.6607

The disturbed cases tend to have higher position error but by an insignificant amount, on the order of 10^{-5} m. Generally the greater the disturbance the greater the position error, but again the increase is negligible. In some instances the disturbed case even have a smaller error than the undisturbed case. Therefore the effect of disturbances on the end effector position error is more random than definitively negative.

However, as with the rendezvous and docking scenario, disturbances increase the controls required to perform the maneuvers. Figure 23 shows the total control use in the presence of disturbances relative to the undisturbed case. The top left plot is translational control, top right is attitude control, and the bottom is total angular impulse on all joints. The three magnitudes of disturbances are shown for each case.

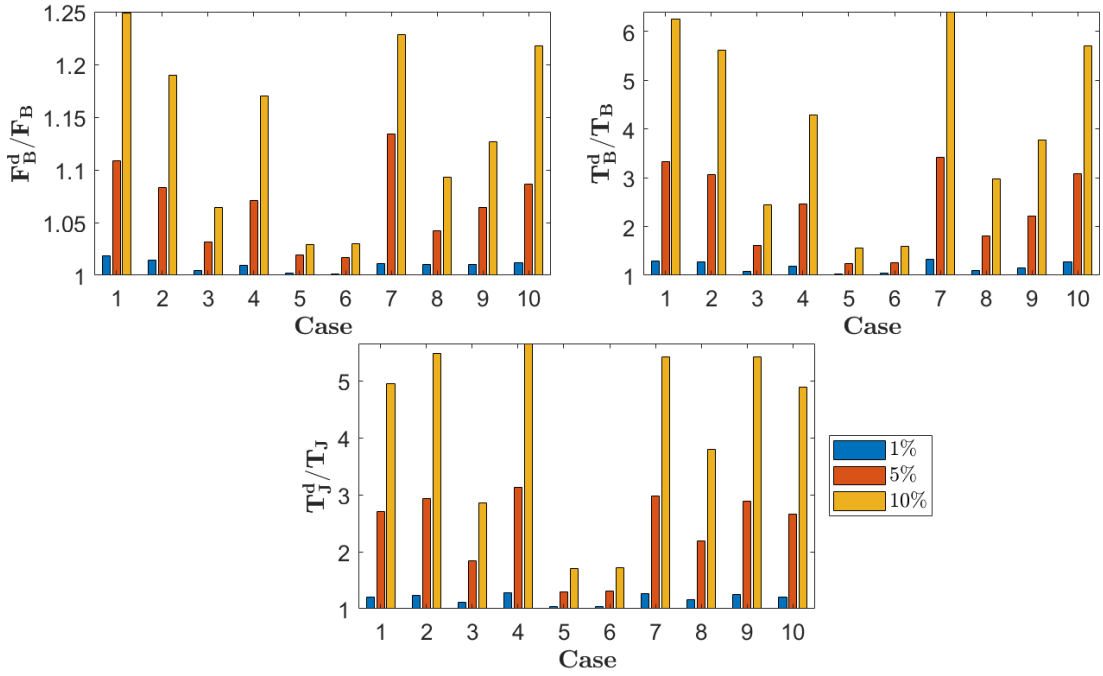

Figure 23: Control use with disturbances relative to undisturbed cases.

Across all cases the inclusion of disturbances increases the required control. The relative increase in control use when increasing the magnitude of the disturbances is also very consistent. Notice the different y-axis scales in each plot. Disturbances have a much smaller effect on the translational control than on the attitude and joint controls. This suggests that the manipulator motion has a large effect on the base translation that eclipses that of the external disturbances. In contrast, disturbances with magnitudes of 5% and 10% of the maximum torques often dominate the total angular impulse for both the attitude and joint control. This suggests that the maneuvers themselves are not very strenuous on the attitude and joint actuators, at least for the 10 second maneuvers.

Cases five and six stand out as the inclusion of disturbances has a relatively much smaller effect on total control use than in all other cases. The reason is that these maneuvers cost significantly more control in the undisturbed case. Not coincidentally, these cases are the same as those with large end effector position errors. In sum, disturbances have a smaller effect on maneuvers that require more control.

4.3 Scenario 3: End Effector Path Tracing

Figure 24 shows a typical desired end effector trajectory in the LVLH frame for the path tracing scenario, along with the associated velocity profile. The end effector begins at the origin and moves to trace the circular path counterclockwise. Notice the velocity profiles are continuous but not smooth at the entry point to the circular path, which occurs here at $t = 5$ seconds.

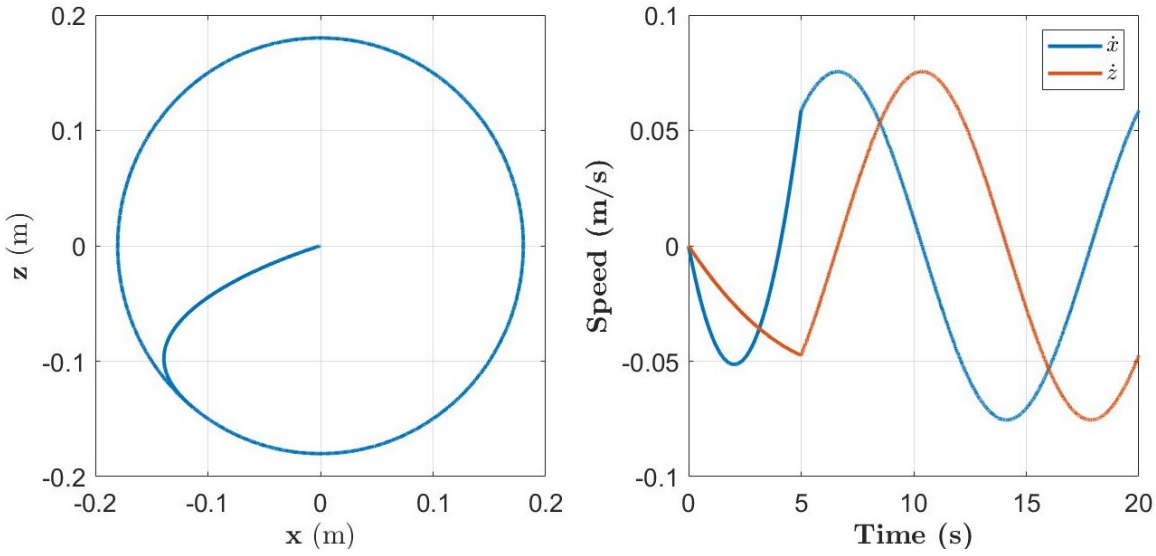

Figure 24: Desired end effector position (left) and velocity (right) profiles.

In nearly all cases the system is able to achieve the desired end effector trajectory within 5 mm of position error. The mean and standard deviation of the end effector position error for each case is shown in Table 12.

Table 12: Mean End Effector Position Error

Case #	$\overline{r_e} \pm \sigma$ (mm) $t_f = 20$ s	$t_f = 30$ s	$t_f = 40$ s
1	2.5636 ± 1.4472	0.5840 ± 0.2156	0.3788 ± 0.1836
2	3.7947 ± 1.9222	1.4075 ± 0.7589	0.4974 ± 0.2636
3	2.2891 ± 0.9396	0.6919 ± 0.3573	0.4404 ± 0.2465
4	4.3867 ± 2.5254	1.6951 ± 1.0348	0.4577 ± 0.2362
5	5.7738 ± 3.3583	1.3409 ± 0.8756	0.3620 ± 0.1399
6	0.7649 ± 0.2998	0.4698 ± 0.2433	0.3309 ± 0.1934
7	39.6515 ± 9.9465	9.3064 ± 5.2563	3.0419 ± 2.9539
8	22.8916 ± 7.0979	7.5288 ± 3.3507	2.0061 ± 1.9595
9	1.0216 ± 0.4929	0.4582 ± 0.2130	0.3244 ± 0.1676
10	19.9560 ± 5.3930	4.1065 ± 3.0247	1.4040 ± 1.1555

The mean and standard deviation decrease as the maneuver time increases, as expected. Several cases exhibit significantly higher errors than the rest, especially for the 20 second maneuvers. In these cases the error spikes during the entry trajectory onto the circular path, and the system is unable to fully dampen out the error while performing the circular maneuver. This is demonstrated in Figure 25 which shows the end effector position error over time for case seven.

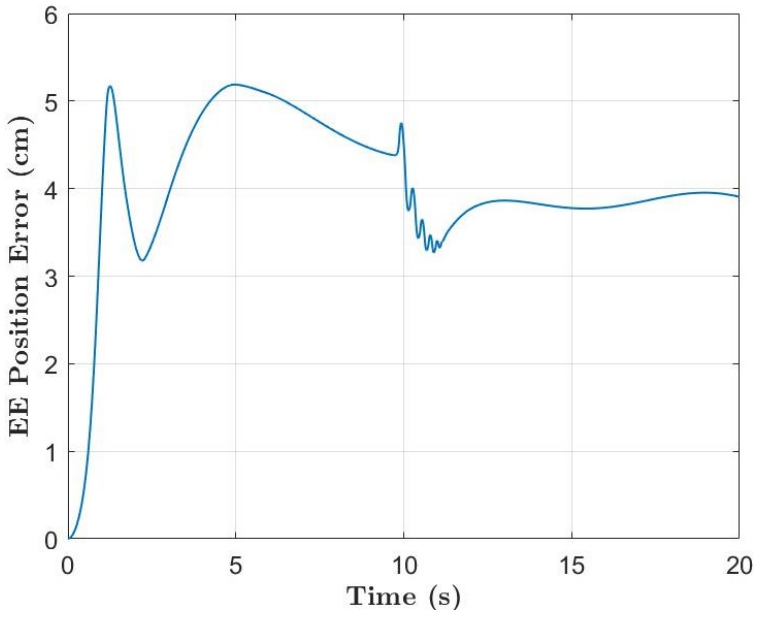

Figure 25: End effector position error versus time with large errors.

As the end effector moves onto the circular path its position error jumps to over 5 cm. The disturbances imparted to the base are not fully dampened and the position error remains above 3 cm throughout the circular motion. An interesting effect occurs around 10 seconds when, given the errors in the base pose, the desired end effector velocity forces the manipulator to the reachable workspace boundary. This causes the manipulator to oscillate until the desired trajectory re-enters the workspace.

The state errors and controls resulting in this end effector trajectory are shown in Figures 26 and 27. The large jump in all state errors in the first 5 seconds is clearly evident. The base attitude and joint errors largely dissipate by the end of the 20 second maneuver, while those in the base position remain relatively high. The control use is very chaotic during this maneuver, with many of the controls saturating consistently for the first 12 seconds. The faulty trajectory planning is evident in Figure 27 between 10 and 12 seconds, where all controls oscillate between their maximum and minimum values as the manipulator traces its workspace boundary.

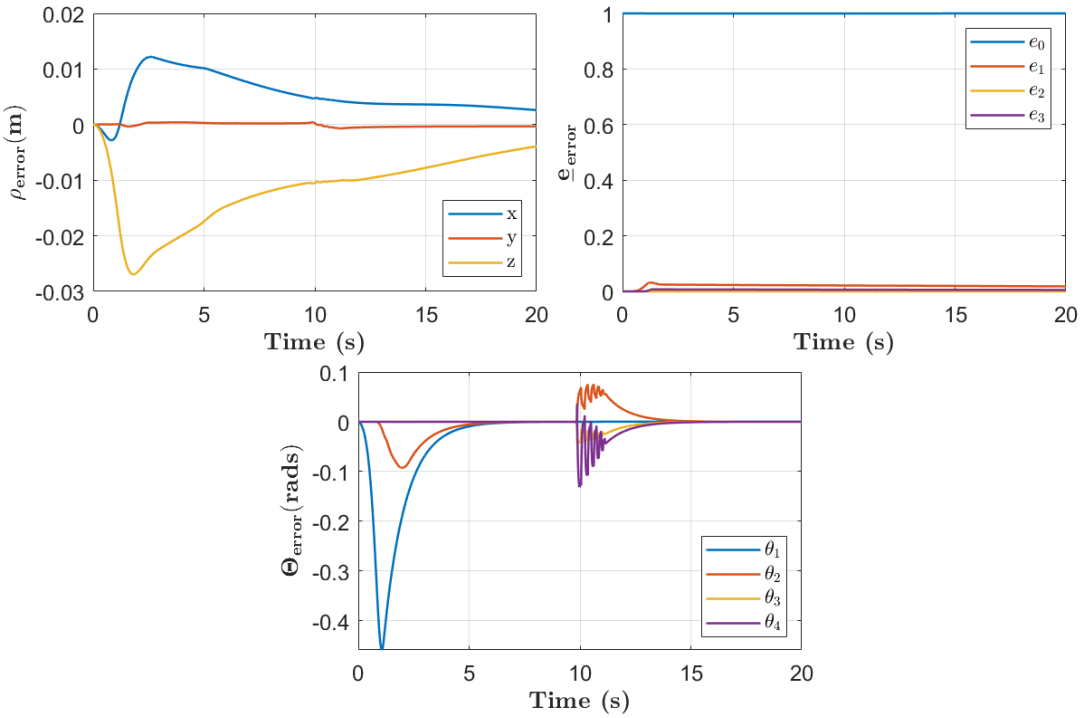

Figure 26: State errors for case seven shown in Figure 25

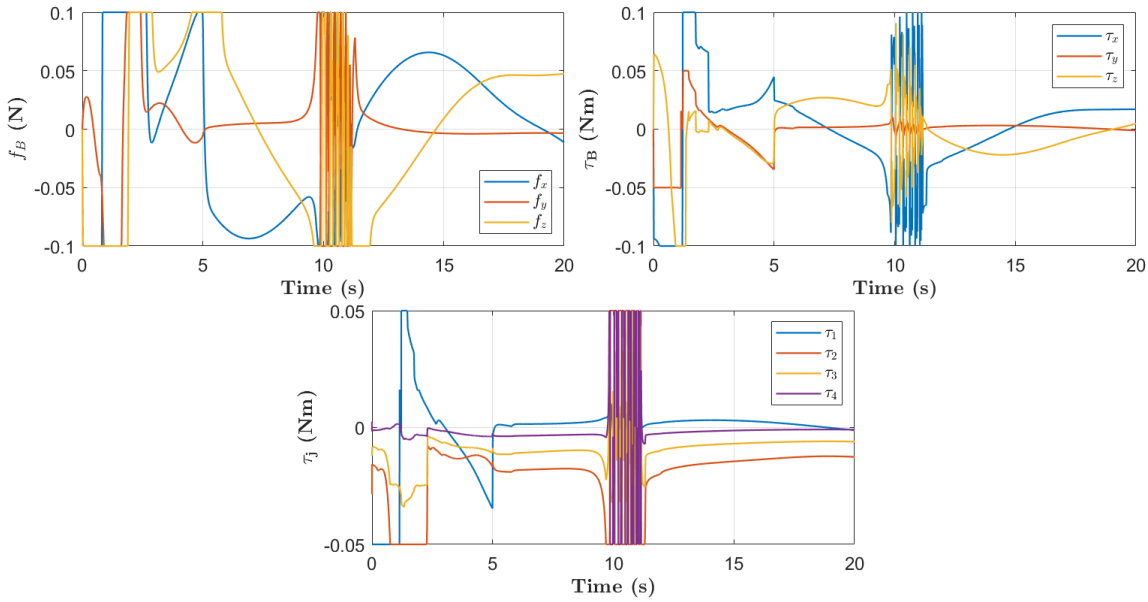

Figure 27: Controls for case seven shown in Figure 25

This phenomenon can be avoided in several ways. The simplest would be to limit the joint speeds and accelerations in the trajectory planning. If the manipulator motion was less aggressive, the base would be able to regulate its errors successfully, and the end effector would not be driven to the edge of its workspace. Another simple mitigation strategy would be to implement singularity avoidance in the Jacobian pseudo-inverse differential kinematics algorithm. As the manipulability is zero at the edge of the workspace, these configurations would be avoided as long as the relative importance of singularity avoidance is high. Finally, the base states could be included in the Jacobian, and therefore included in the inverse differential kinematics. The effect would be to allow both the base and the manipulator to move to achieve a desired end effector trajectory. This may be difficult in practice since the base translational motion is not very precise. However, allowing the base to move slightly may benefit overall performance and would certainly avoid issues with the manipulator reaching the workspace boundary. This may also be necessary for real-world tasks that require large movements extending beyond the workspace of a stationary-base system.

Across all cases the end effector position error is almost entirely due to errors in the position of the CubeSat base. Joint angle errors are typically bounded by $\pm 10^{-5}$ rads, and the elements of the base quaternion deviate from the desired attitude by a similar amount. Meanwhile, the order of magnitude of the base position error matches that of the end effector throughout the simulations. This can be seen in Figures 50 - 56 in Appendix A, which shows the control histories and state errors for a typical case, the 30 second maneuver for case two.

The total control usage consistently decreases with maneuver time, as shown in Figure 28. The relative decrease in total impulse with time is very similar in each case due to the similarity of the maneuvers. While the different entry trajectories onto the circular paths require varying amounts of control, the circular path constitutes the majority of the maneuver and requires a similar control profile for each case, which serves to equalize the relative control use. For this reason, the correlation between total control use and end effector position error is not as strong as in scenario 2. For this scenario the end effector position error is predominately driven by control use during the entry trajectory.

Disturbance forces once again have little to no effect on the average end effector position error, but did increase the required total control use. Figure 29 shows the total control use with disturbances relative to the undisturbed cases. Notice the increases across all actuators are much smaller than those seen in scenario 2, where the 10% disturbances resulted in as much as a six-fold increase in total angular impulse imparted to the CubeSat base. The relatively small increases shown in Figure 29 are in agreement with the previous finding that the larger the control use required for the maneuver, the smaller the effect of disturbances. A comparison of Figures 22 and 29 shows that the path tracing maneuvers require about twice as much control as the simple end effector repositioning maneuvers of scenario 2.

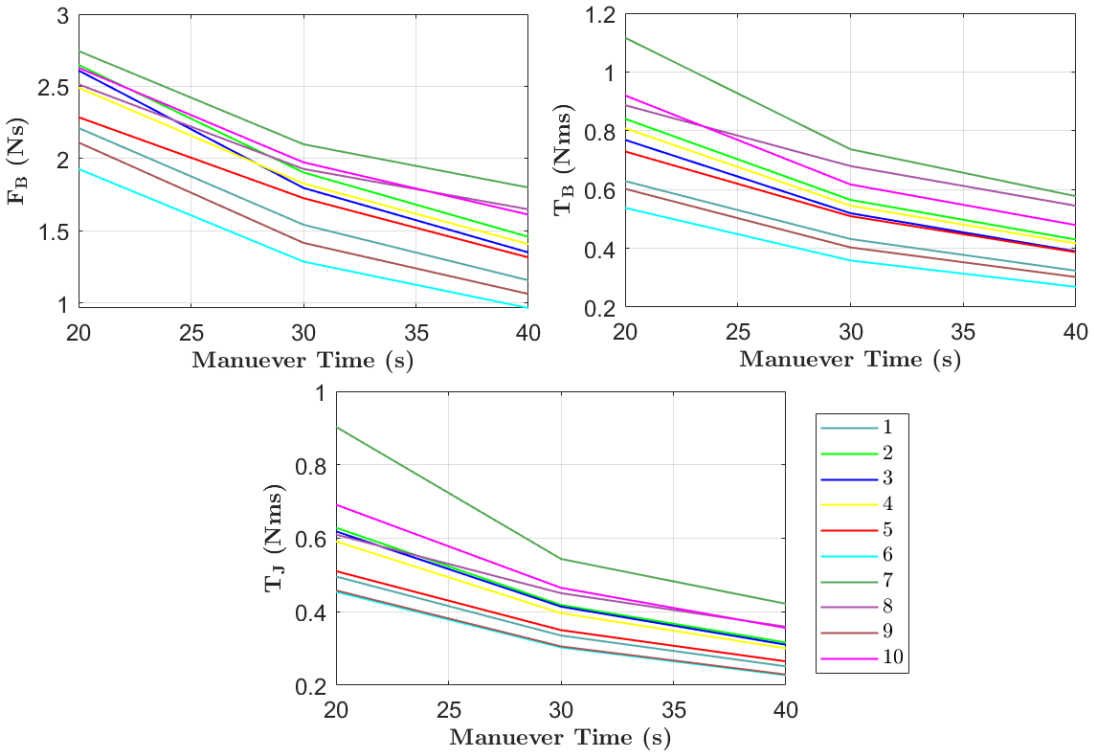

Figure 28: Total control use versus maneuver time.

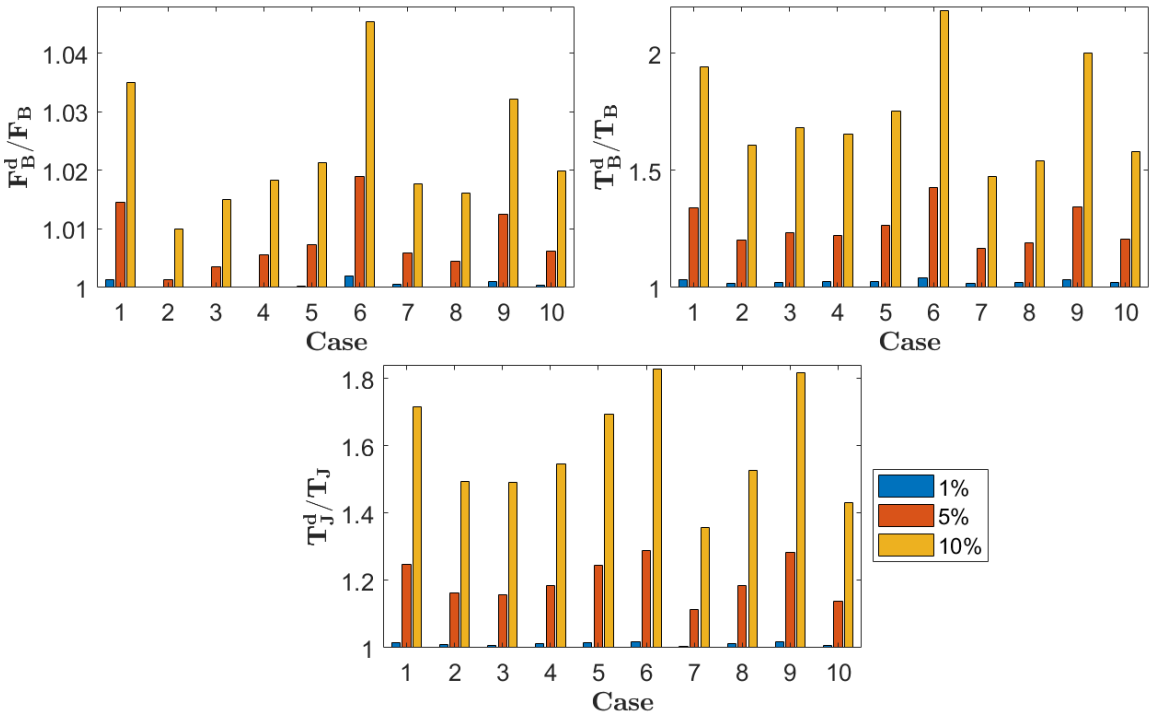

Figure 29: Control use with disturbances relative to undisturbed cases.

4.4 Summary

The CubeSat robot was able to perform each orbital scenario, indicating that the system is feasible in the free-flying case. The rendezvous and docking maneuvers were lengthy, but used achievable levels of control and settled within an acceptable error. The stationary tasks were successful in general, with end effector position errors less than 1 mm in many cases. Across all scenarios disturbance forces had little effect on the error responses, as the controllers were able to counteract the effects. Disturbances did, however, increase the control required to perform the maneuvers.

The most common issue with performance seen in the stationary tasks involved kinematic singularities. When the desired end effector trajectory brought the manipulator near a singularity, the Jacobian pseudo-inverse algorithm often resulted in excessively large joint speeds. In attempting these joint trajectories, the manipulator imparted large disturbance forces on the CubeSat base that the base controllers were unable to counteract, resulting in excessive control use and poor overall performance.

These cases are informative for the trajectory planning of a real-world system. A common thread in all cases with large errors was large joint accelerations, particularly in joint 1. In the rendezvous configuration with the manipulator extended, movement in joint 1 imparts the largest disturbance forces on the base. But joint 1 movement is necessary for all motions spanning three dimensions. One solution is to design the system to remove kinematic singularities in the workspace. Another is to bound the joint accelerations. All of the poor performance cases exhibited joint 1 accelerations in excess of 0.75 rad/s^2, with the worst cases approaching 2 rad/s^2. A reasonable mitigation strategy would be to limit both the joint speeds and accelerations. More simulations and analysis would need to be performed to determine appropriate bounds on each. Many other recommendations for follow-on research efforts, as well as potential alternative investigations, are discussion in Section 5.1.

V. Conclusions

In this research the coupled orbit-attitude dynamics of a 6U CubeSat with a four-link serial manipulator were derived. Relative orbital dynamics between the CubeSat and a chief spacecraft were implemented in a Simulink model, and a PID controller was used to guide the CubeSat through a series of scenarios representative of an on-orbit servicing system. The scenarios included a rendezvous and docking following ejection from a chief spacecraft, repositioning the manipulator to place the end effector on a desired location, and tracing a prescribed path with the end effector.

The simulation results indicate that, for the free-flying case, a CubeSat with a robotic manipulator is feasible. In other words, given the state of the art in CubeSat ADACS and propulsion subsystems, a CubeSat is capable of both rendezvousing and docking with a chief spacecraft after ejection, and controlling the base motion while maneuvering the manipulator.

Rendezvous and docking required a significant amount of time, anywhere between 7.2 and 15.4 minutes depending on the ejection speed. However, as discussing in Section 4.1, these times could be reduced significantly by using a berthing system more appropriate for this application. In the stationary tasks, the CubeSat was able to regulate errors in the base states and follow the prescribed motion of the end effector, often with sub-millimeter accuracy. In a handful of outlying cases the system exhibited large errors in the states due to kinematic singularities. These cases could easily be mitigated with several measures discussed in Sections 4.2 and 4.3. Across all scenarios the inclusion of random disturbance forces had little effect on the system error responses, but did increase the control required to perform the maneuvers.

The results of this research are limited to the system detailed in Section 3.1 and to the scenarios simulated - namely RPO with a three-axis stabilized, cooperative chief spacecraft in LEO. However, reasonable assumptions regarding similar systems

and different orbits can be drawn from these results. For example, it can be assumed that a 12U system could exhibit performance similar to, or even better than, the 6U system in this study. A 12U system would have a more massive base and therefore less coupling between the base and manipulator motion, and it would have a larger volume budget for more sophisticated actuators. Additionally, it can be assumed that this system could perform similarly in all Earth orbital regimes. Highly eccentric orbits may negatively impact the performance, particularly that in the rendezvous and docking scenario where orbital motion is most important. However, the chief orbit in this study, or the ISS orbit, is one of the lowest operational orbits with the largest gravitational forces and orbital speeds. Therefore the orbital motion in any other regime would likely have a smaller impact on the coupled base-manipulator motion, potentially improving the system stability and performance.

These results are also limited to a free-flying CubeSat robot, as contact dynamics were not included in the model. However, the success of the system in performing several free-flying tasks warrants further investigation into the expected capabilities of a CubeSat robot, to include the addition of contact dynamics in the model and simulations.

Designing a real-world system would be a significant systems engineering challenge, as this work assumed six thrusters each capable of 0.1 N, which is currently available in the 1U form factor [7]. A real-world system would also require significant volume budget for power, navigation, structures, thermal control, command and data handling, flight software, and communications subsystems. Therefore, advances in miniaturized propulsion systems would be necessary to realize the performance seen in these results.

The contribution of this work is significant for many reasons. Few studies in space robotics have investigated the dynamics and control of a nanosat-scale system. Most research in space robotics focus on larger systems and do not thoroughly address gravity or coupled orbit-attitude motion, as discussed in Section 1.2. Not only is the concept of a robotic servicing CubeSat novel, but the derivation of its dynamics using Kane's Method, with the inclusion of Newtonian gravity, and the direct control of the nonlinear system presents a unique contribution to the present body of research.

The methods of deriving the system dynamics, integrating the dynamics into an RPO simulation framework, and implementing trajectory planning techniques for on-orbit servicing tasks can be applied to any other robotic servicing spacecraft. The results of this research can inform the mechanical design of a future real-world system, specifically designing a manipulator with no kinematic singularities in the spacecraft's desired workspace.

Ultimately this research indicates that a robotic servicing CubeSat is feasible, and follow-on research, discussed in the next section, should be conducted to advance this concept. If realized, robotic servicing CubeSats could contribute to the space logistics infrastructure, which is a key component of the DOD and NASA's vision of a permanent presence in cis-lunar space [1, 6].

5.1 Recommendations

Several follow-on research efforts are recommended to more thoroughly investigate this concept. Perhaps most important is improving the fidelity of the model, which can be accomplished in several ways. First and foremost, the chief spacecraft can be modelled as its own multibody system with representative features such as solar panels and communications antennas. This would enable more challenging scenarios, such as docking with a specific point on the spacecraft while avoiding all other structures.

Next, contact dynamics between the manipulator and a chief spacecraft should be incorporated in the model to accurately simulate docking and servicing tasks. Contact dynamics are necessary to determine the types of servicing tasks a CubeSat could perform, and can leverage existing techniques in force compliance control [37].

Estimation is another critical aspect of any real world system that should be incorporated in the model to more accurately analyze the expected performance and robustness of the system. Traditional spacecraft sensors, such as those for navigation and attitude determination, can be incorporated to capture uncertainty in the base states. Sensors unique to robotic systems, namely proprioceptive sensors to measure the manipulator internal states, and exteroceptive sensors to characterize the manipulator interaction with the orbital environment, such as force, distance, and visual sensors, can be added to thoroughly capture all uncertainties related to the manipulator and the client spacecraft. Numerous studies are dedicated to both the design of space sensors for autonomous rendezvous and docking [68], and pose estimation of spacecraft during RPO [69]. Incorporation of the sensor dynamics into the simulation framework would require accurate sensor models, as well as a hybrid Simulink model to mate the continuous CubeSat dynamics with the discrete sensor measurements.

Finally, actuator models can be included to improve the fidelity of the simulations. Pulsed thrusters with minimum impulse bits rather than continuous, arbitrarily-small thrusts would greatly improve the fidelity of the translational controls and likely decrease the overall system performance. Modelling the base reaction wheels or control moment gyroscopes as distinct bodies in the multibody system would improve the fidelity of the attitude control scheme. Incorporating joint actuation systems would capture the true dynamics of the manipulator and likely improve the stability of the system. Dynamic parameters such as friction in the joint servomotors can be deliberately chosen to decouple the joints and linearize the manipulator motion [52].

Note that incorporating accurate sensor and actuator models, to include their power requirements, in the overall system model would enable systems engineering and design studies to determine the solar array size and battery storage requirements for a CubeSat robot's power subsystem.

In addition to improving the fidelity of the model, several additional capabilities could be added to the simulation framework to enable more thorough and complete investigations of system performance. For example, allowing for alternative Jacobian pseudo-inverses in the inverse differential kinematics algorithm could create new metrics for performance analysis, such as obstacle and singularity avoidance, and distance from mechanical joint limits. Incorporating the full end effector pose into the trajectory planning algorithm, rather then solely its position, would enable the simulation of more specific manipulator tasks.

As discussed in Section 4.3, allowing the system to leverage its base DOF to achieve desired end effector trajectories would increase the flexibility and maneuverability of the system, and enable the simulation of a broad range of tasks spanning large, complex motions. This would involve expanding the Jacobian to include the base states, and devising an appropriate Jacobian pseudo-inverse. Modelling the actuators would increase the benefits of this addition as one could accurately weight the relative costs of using each DOF of the system to achieve a task based on the power requirements of the controllers, thereby producing a power-optimized trajectory.

The task of tuning the controller gains for the simulations in this research was laborious. Therefore, it would be worthwhile to develop an optimizing tool that could automatically compute optimal gains for the controller given a specific task and the performance metrics of interest. Developing this tool would be a challenge given the sensitivity of the system error responses to changes in the gains. Balancing the various performance metrics, such as settling time, control use, and steady-state error, while

ensuring an acceptable response with zero overshoot across the full range of initial conditions, would require significant logic. However, an auto-optimizing tool would greatly reduce the burden on the researcher when simulating new scenarios.

Beyond improvements to the model and additions to the simulation, there are several alternative studies that can be conducted to advance the robotic servicing CubeSat concept. For example, given the long settling times for the rendezvous and docking maneuvers in this research, it would be worthwhile to investigate the design parameters of a berthing system for this specific application. This investigation could seek to determine the range of ejection speeds that allows time- and control-optimal rendezvous and docking maneuvers while maintaining system safety.

Given the issues with kinematic singularities encountered in this study, it would be useful to investigate the performance of systems with different designs. These could involve different form factors for the CubeSat base, manipulators with more DOF, or even a dual-manipulator system. Different manipulator designs could enable the inclusion of end effector orientation in the path planning algorithms and alleviate issues with singularities. A trade study aimed at determining the designs best suited to perform specific tasks would greatly benefit the state of the art.

Given the systems engineering challenge associated with designing the CubeSat robot of this research, it would be useful to investigate the performance of under-actuated systems. For example, instead of using six thrusters aligned with the CubeSat base axes, perhaps the system could have two thrusters oriented at different angles to achieve controllability in two dimensions, or a single thruster with an adjustable thrust angle. This study could leverage existing work in the autonomous rendezvous and docking of an under-actuated system where attitude control is used to allow a single, body-fixed thruster to achieve arbitrary three dimensional translations [49].

Finally, countless investigations into alternative control methods can be conducted to evaluate the impacts on performance. For example, adaptive control is a very common method used when uncertainties exist in the dynamics, and applying it to this system would be a worthwhile study. While PID control has heritage in space robotic systems and was successful in this research, it would be valuable to compare its performance to optimal controllers with both minimum time and minimum control objectives. However, formulating the scenarios in this research as optimal control problems would be challenging given the number of variables and constraints in the system, and the complexity of the dynamics. Additionally, an on-orbit servicing system would likely use some form of feedback control, rather than feed forward, given the reliance on noisy sensor measurements, uncertainty in client spacecraft dynamics, and the likelihood of disturbances on orbit, both from unmodelled effects on the dynamics and from obstacles that must be avoided. Regardless of the control methods used, it would be valuable to accurately model the system actuators in order to intelligently weight the various controls.

Appendix A. Additional Results

1.1 Scenario 1: Rendezvous and Docking

$$\dot{\mathbf{y}}_0 = -1.667 \text{ m/s}; \; \omega^{\mathbf{B}}_{\mathbf{BL},0} = 3.3333 \text{ °/s/axis}$$

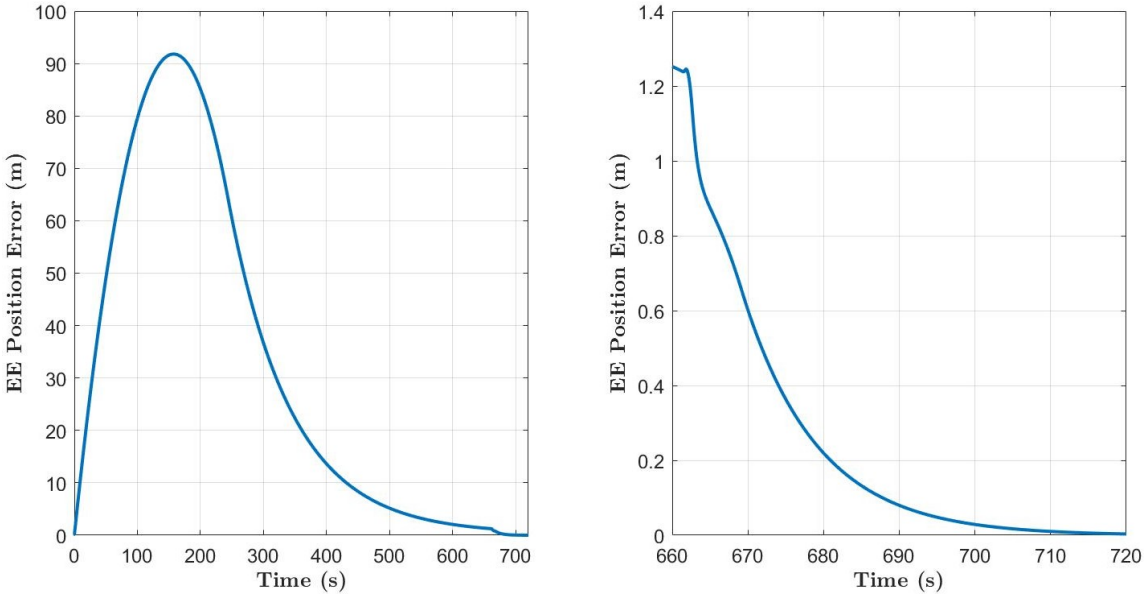

Figure 30: Scenario 1 End effector position error in the LVLH frame.

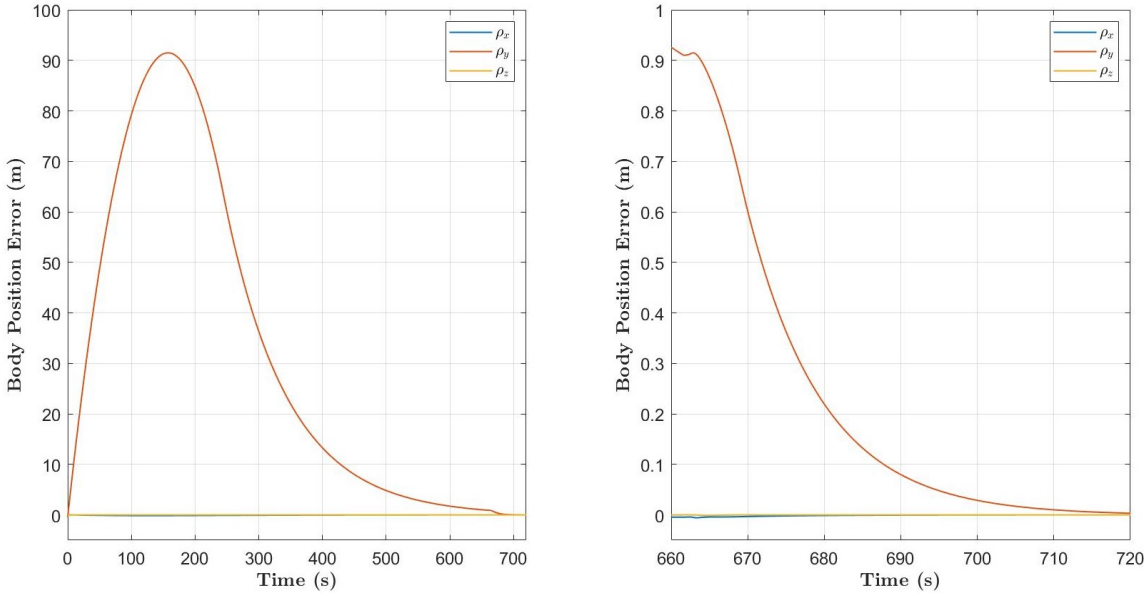

Figure 31: Scenario 1 CubeSat base position error in the LVLH frame.

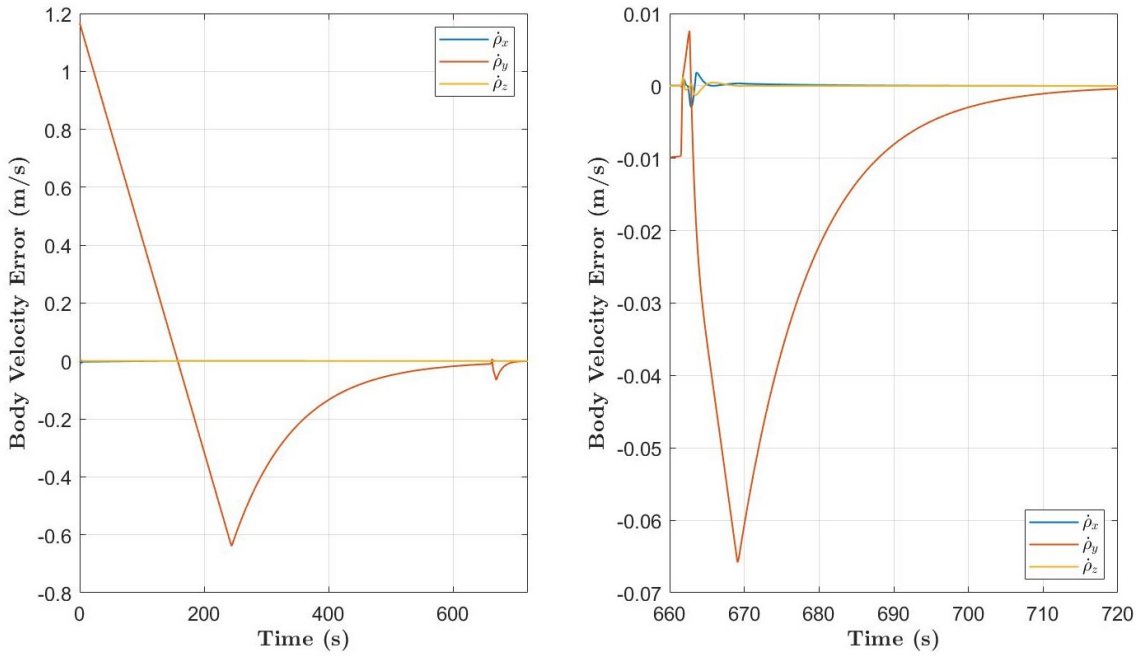

Figure 32: Scenario 1 CubeSat base velocity error in the LVLH frame.

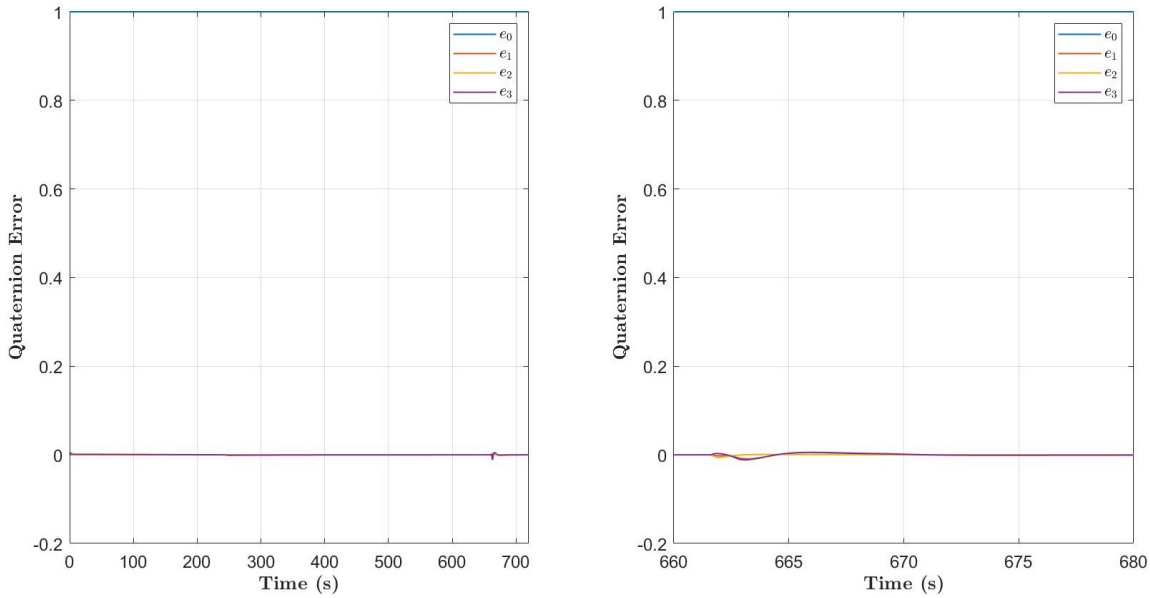

Figure 33: Scenario 1 CubeSat quaternion error relative to the LVLH frame.

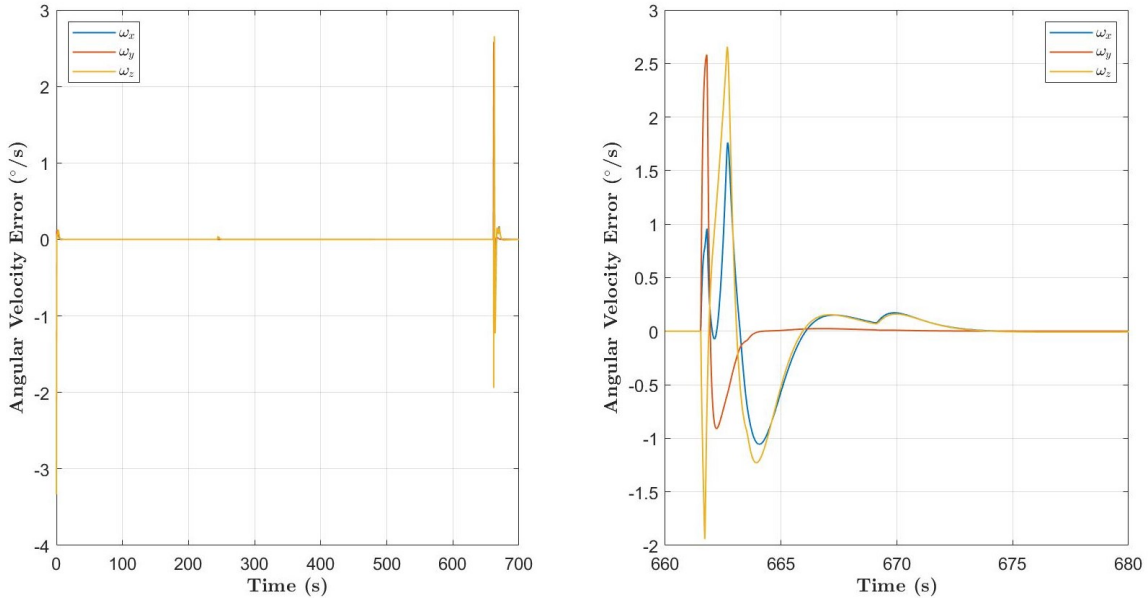

Figure 34: Scenario 1 CubeSat base angular velocity error relative to the LVLH frame.

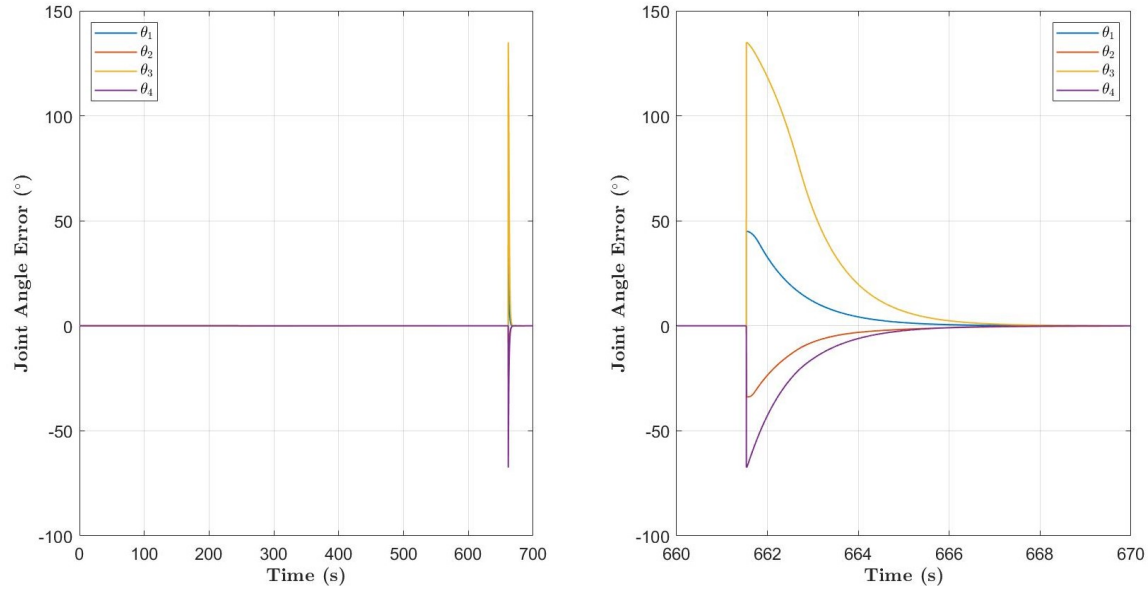

Figure 35: Scenario 1 Joint angle errors.

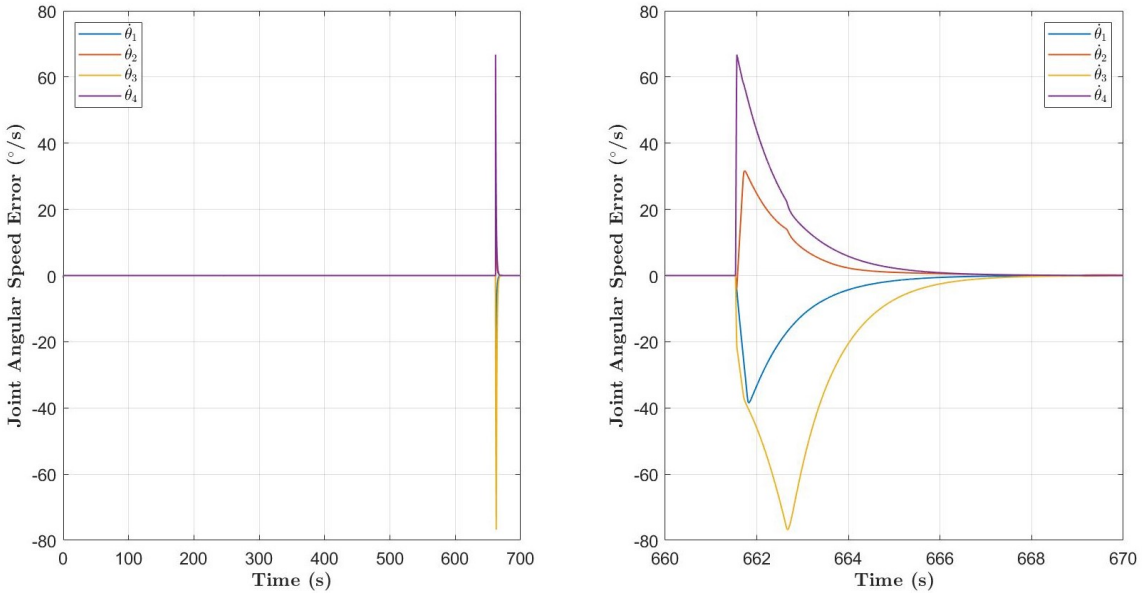

Figure 36: Scenario 1 Joint angular speed errors.

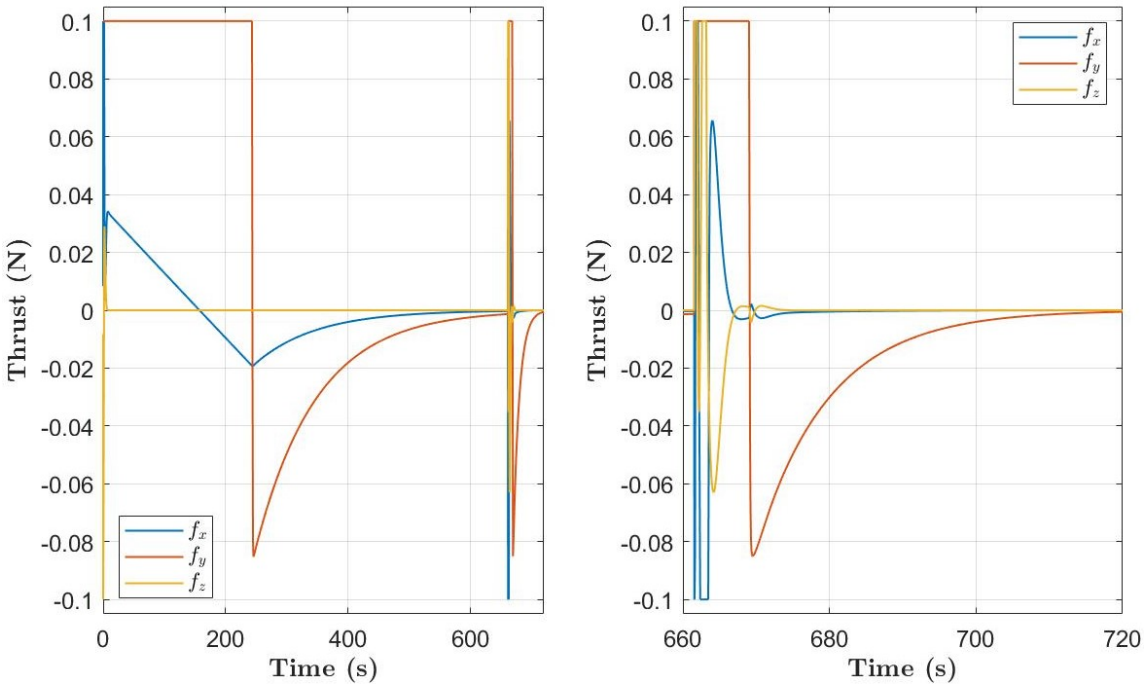

Figure 37: Scenario 1 Thrust profiles.

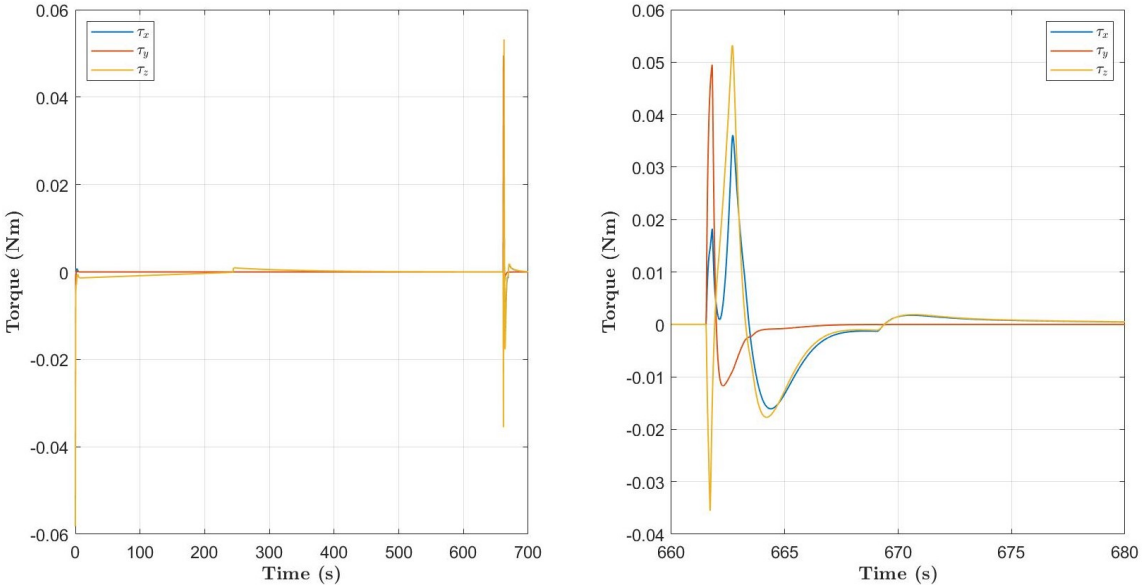

Figure 38: Scenario 1 Base torque profiles.

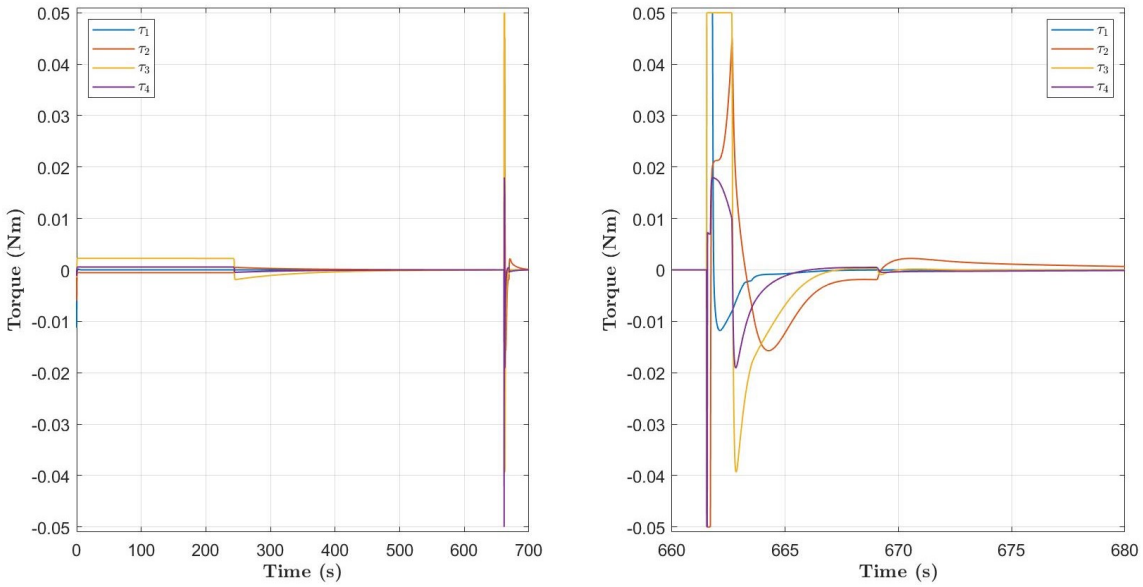

Figure 39: Scenario 1 Joint torque profiles.

1.2 Scenario 2: End Effector Placement

$$\tilde{\mathbf{r}}_{\mathbf{EE,des}} = [\mathbf{0.0784},\ \mathbf{0},\ \mathbf{0.1028}]^{\mathbf{T}}\ \mathbf{m},\ \mathbf{t}_{\mathbf{maneuver}} = \mathbf{10}\ \mathbf{s}$$

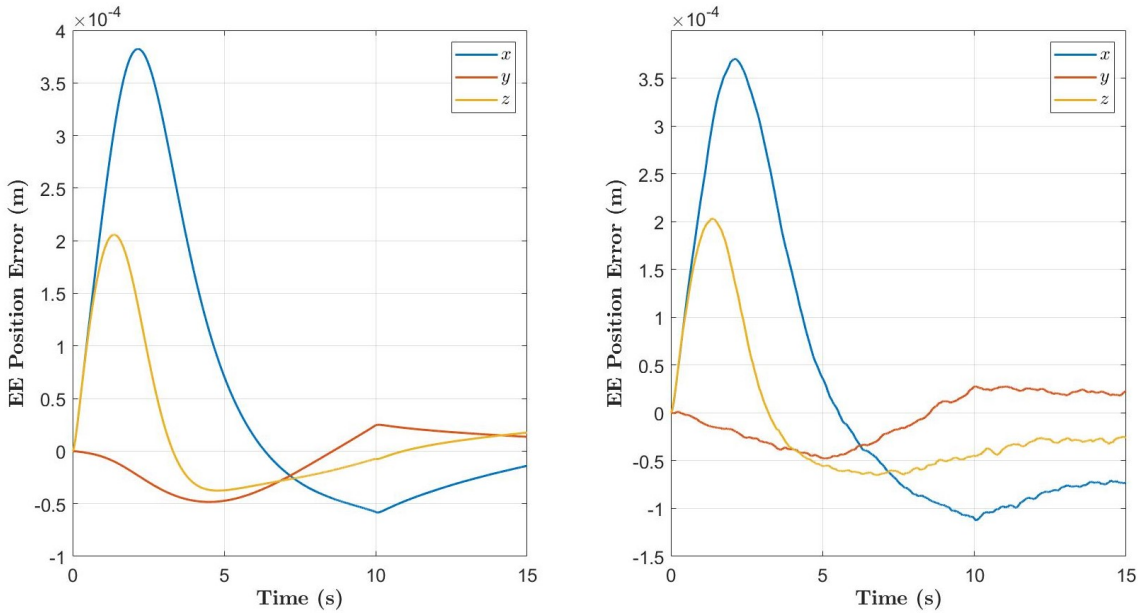

Figure 40: Scenario 2 End effector position error in the LVLH frame. Undisturbed (left), maximum disturbances (right).

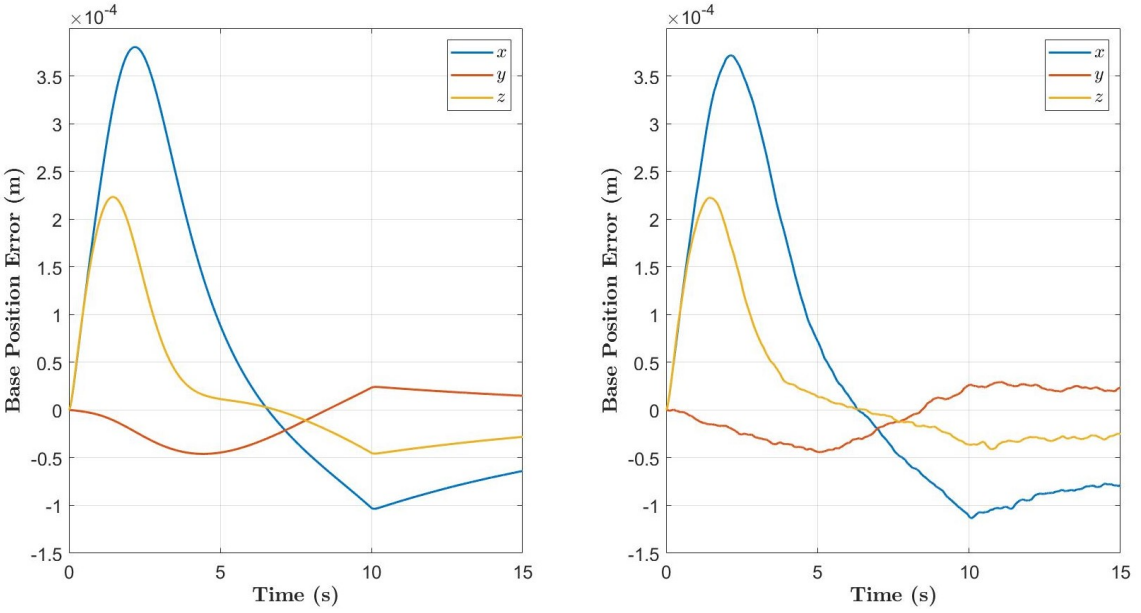

Figure 41: Scenario 2 CubeSat base position error in the LVLH frame. Undisturbed (left), maximum disturbances (right).

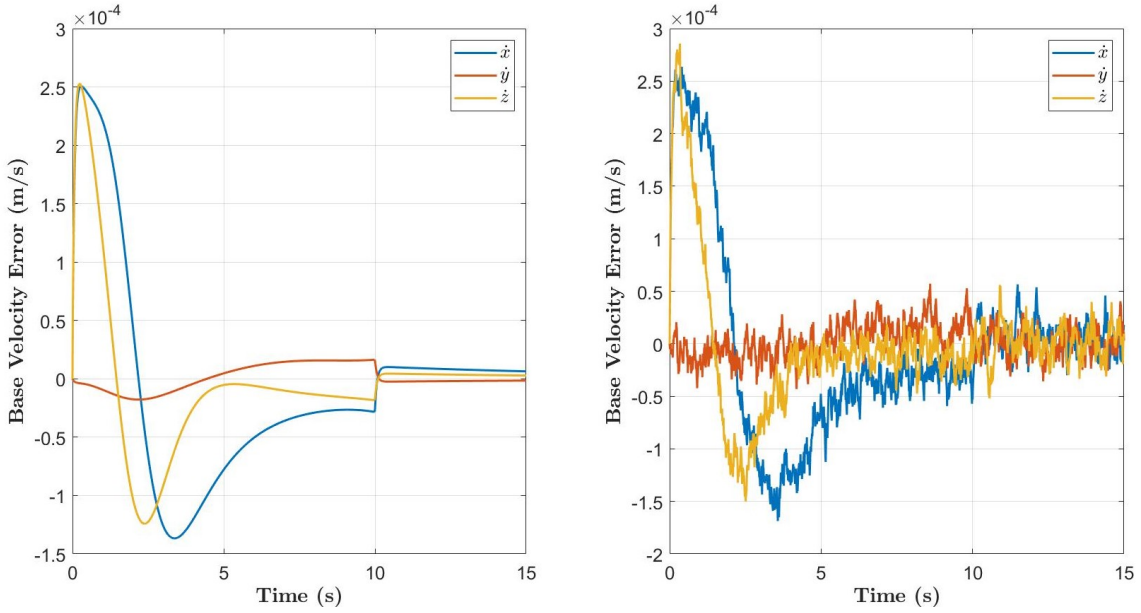

Figure 42: Scenario 2 CubeSat base velocity error in the LVLH frame. Undisturbed (left), maximum disturbances (right).

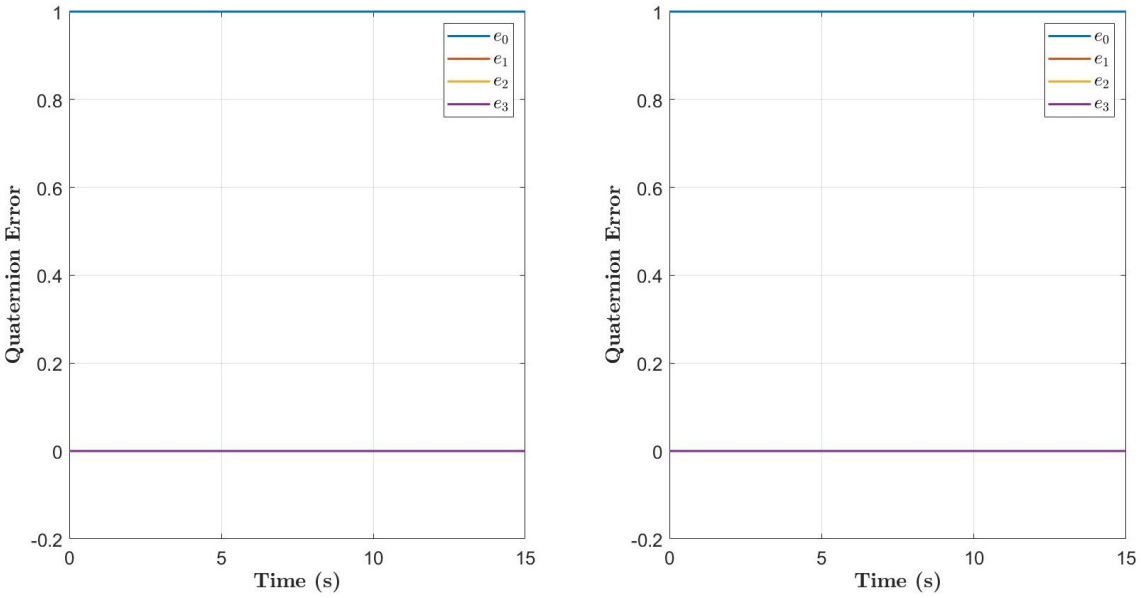

Figure 43: Scenario 2 CubeSat quaternion error relative to the LVLH frame. Undisturbed (left), maximum disturbances (right).

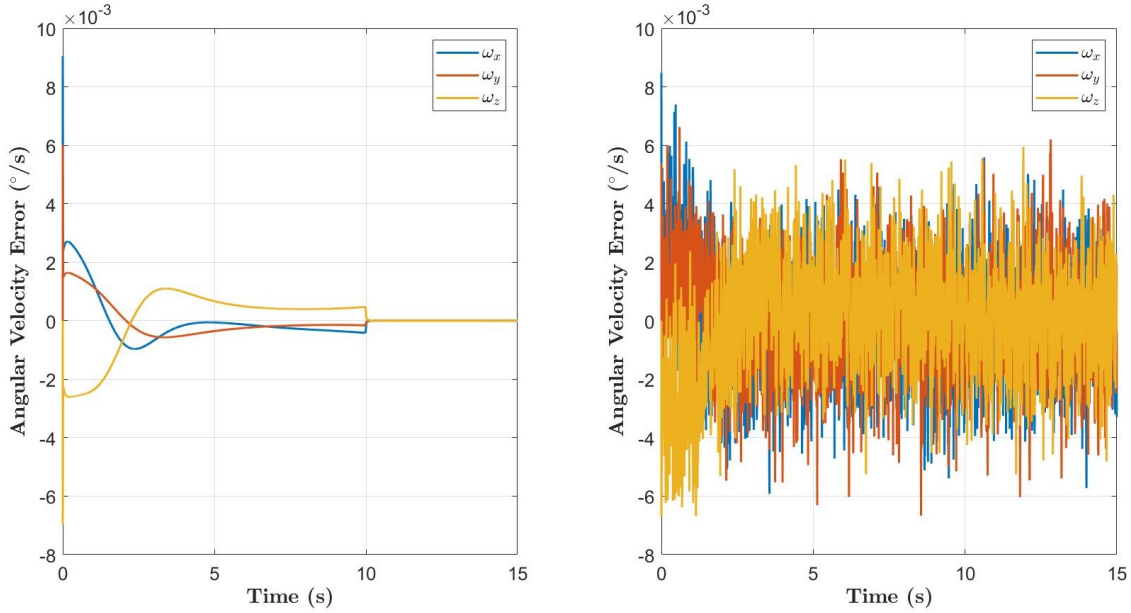

Figure 44: Scenario 2 CubeSat base angular velocity error relative to the LVLH frame. Undisturbed (left), maximum disturbances (right).

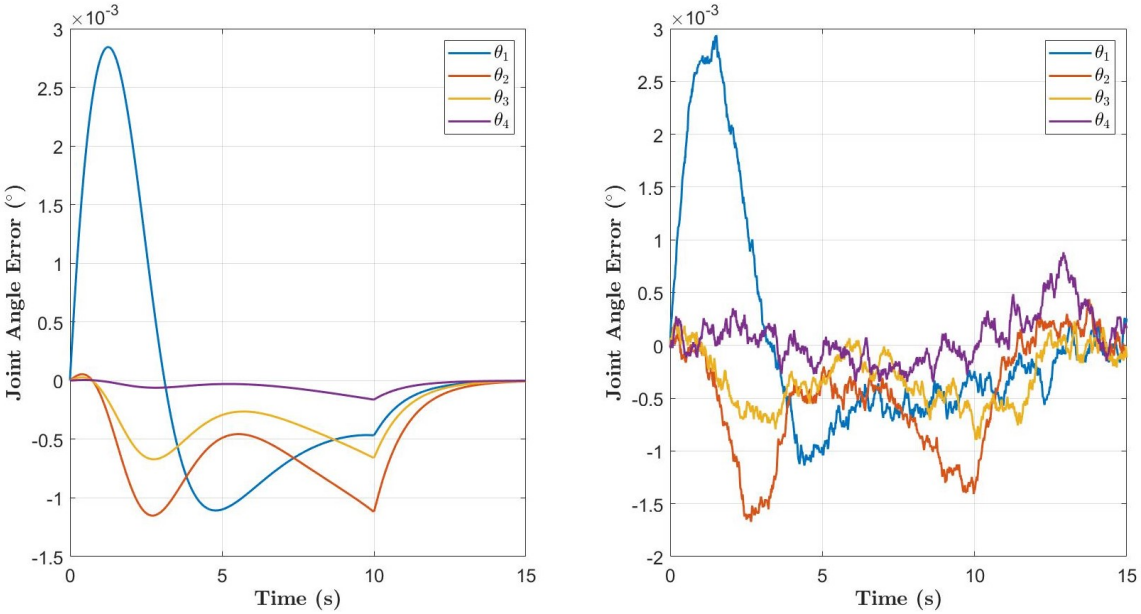

Figure 45: Scenario 2 Joint angle errors. Undisturbed (left), maximum disturbances (right).

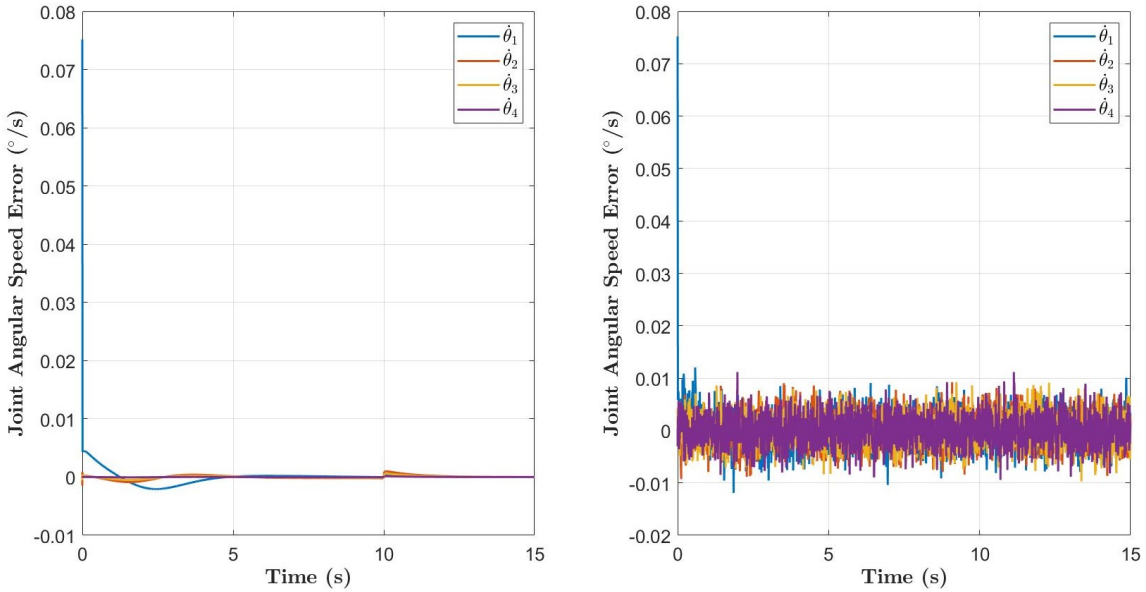

Figure 46: Scenario 2 Joint angular speed errors. Undisturbed (left), maximum disturbances (right).

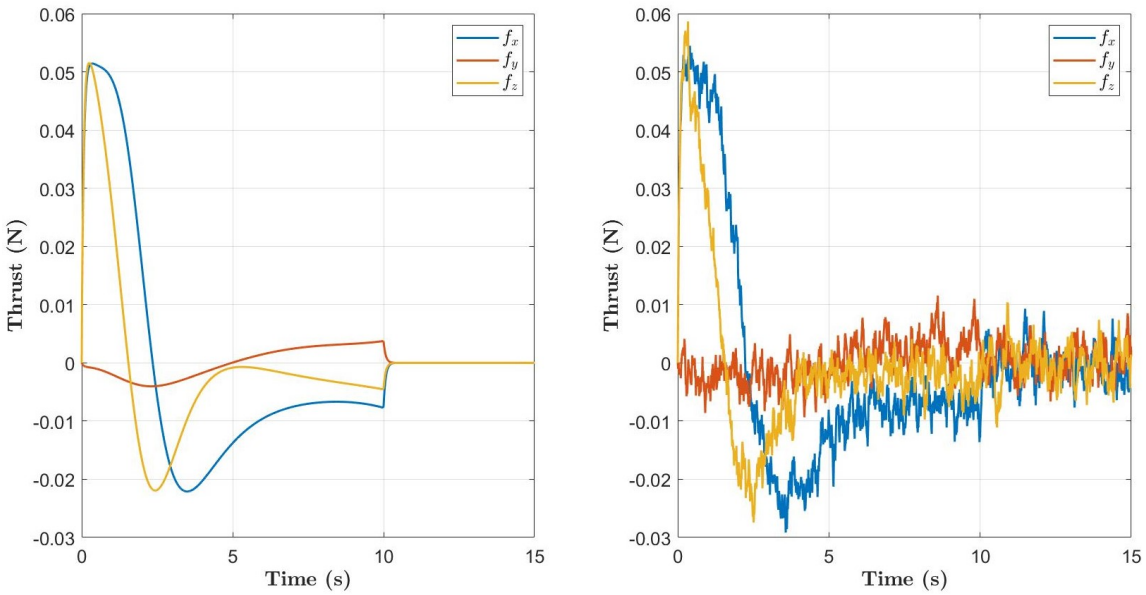

Figure 47: Scenario 2 Thrust profiles. Undisturbed (left), maximum disturbances (right).

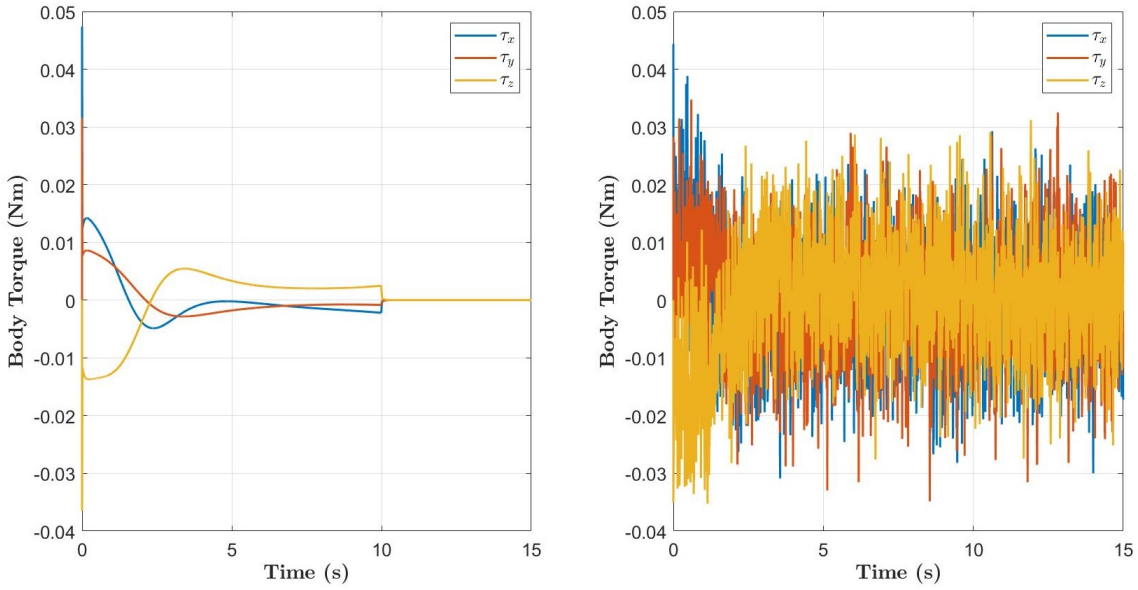

Figure 48: Scenario 2 Base torque profiles. Undisturbed (left), maximum disturbances (right).

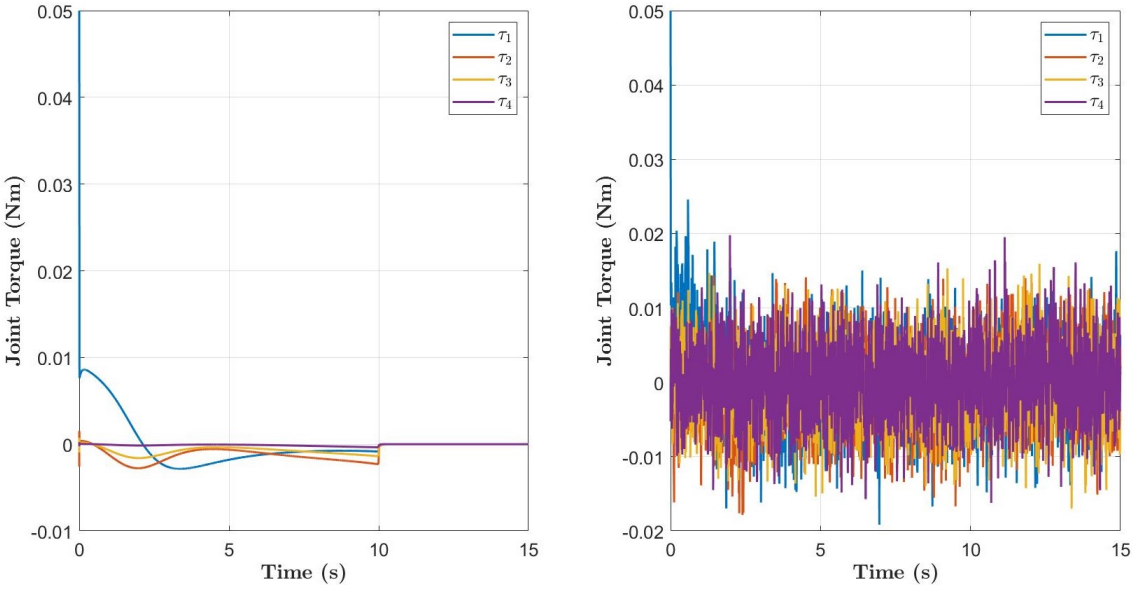

Figure 49: Scenario 2: Joint torque profiles. Undisturbed (left), maximum disturbances (right).

1.3 Scenario 3: End Effector Path Tracing

$$\tilde{\mathbf{r}}_{\mathbf{EE,entry}} = [-\mathbf{0.0740},\ \mathbf{0},\ \mathbf{0.2724}]^{\mathbf{T}}\ \mathbf{m},\ \mathbf{t}_{\mathbf{maneuver}} = \mathbf{30}\ \mathbf{s}$$

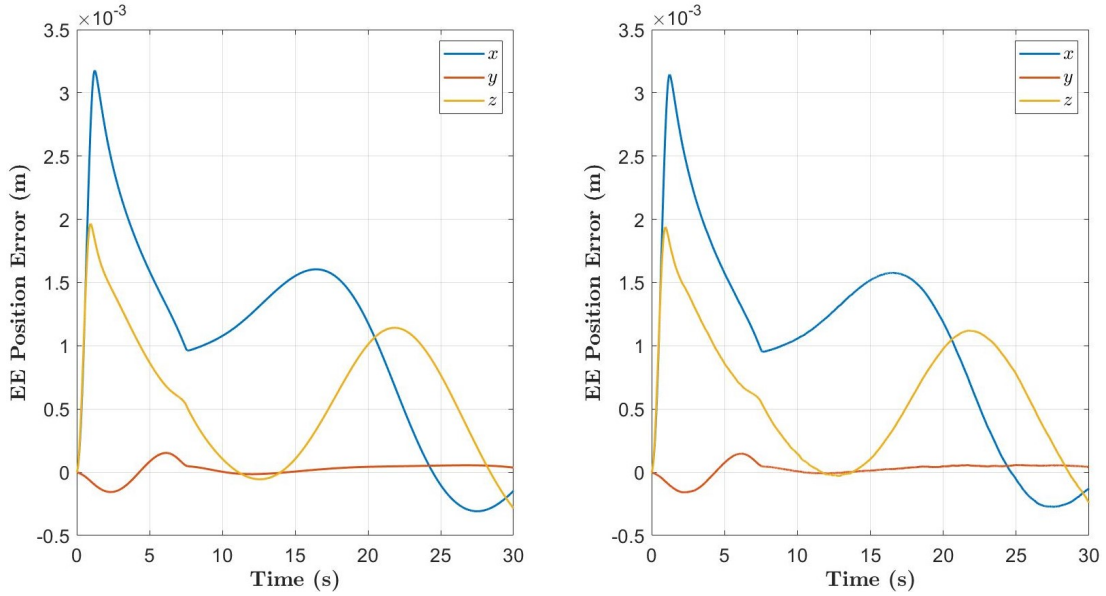

Figure 50: Scenario 3 End effector position error in the LVLH frame. Undisturbed (left), maximum disturbances (right).

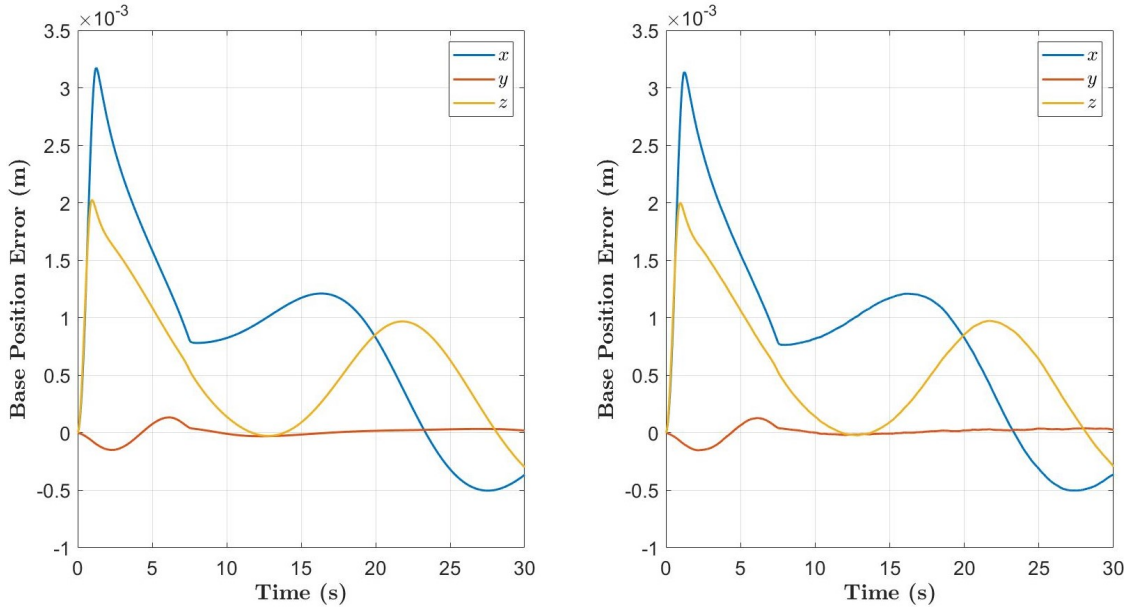

Figure 51: Scenario 3 CubeSat base position error in the LVLH frame. Undisturbed (left), maximum disturbances (right).

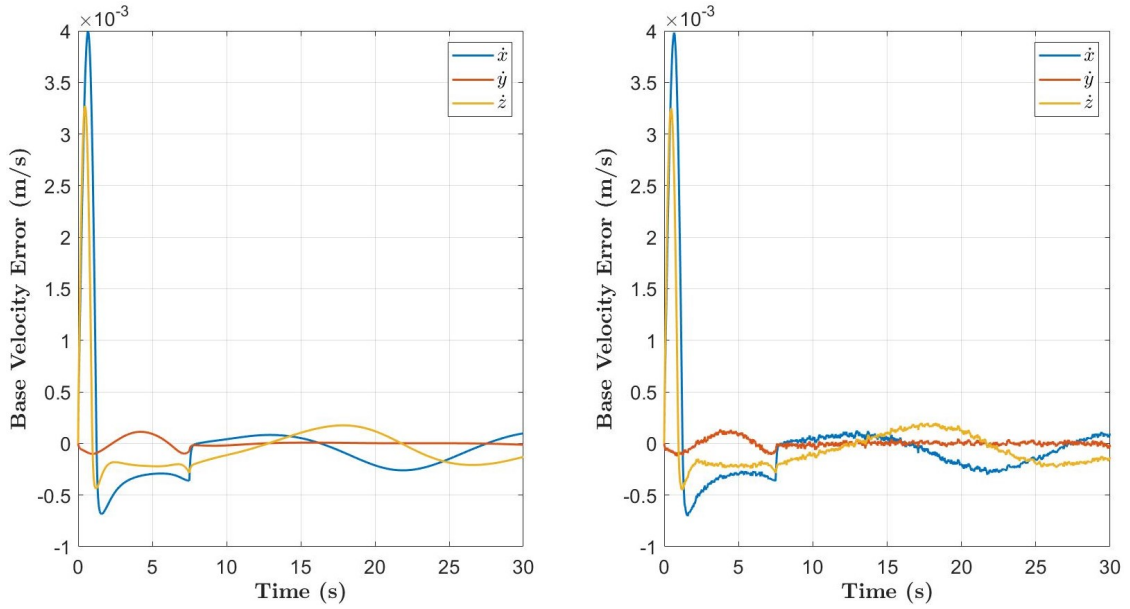

Figure 52: Scenario 3 CubeSat base velocity error in the LVLH frame. Undisturbed (left), maximum disturbances (right).

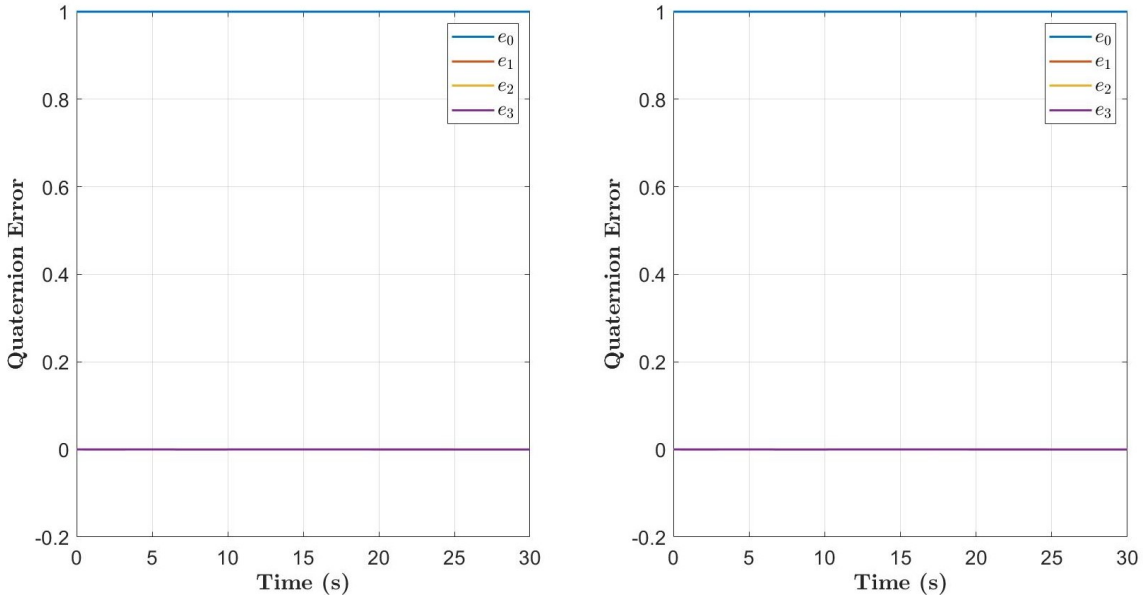

Figure 53: Scenario 3 CubeSat quaternion error relative to the LVLH frame. Undisturbed (left), maximum disturbances (right).

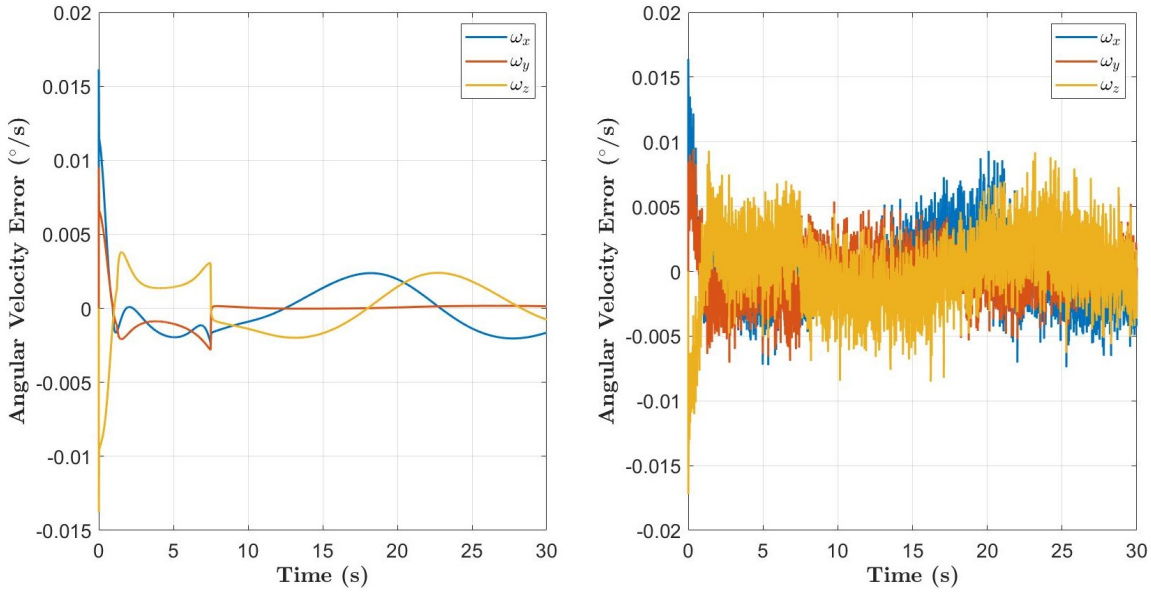

Figure 54: Scenario 3 CubeSat base angular velocity error relative to the LVLH frame. Undisturbed (left), maximum disturbances (right).

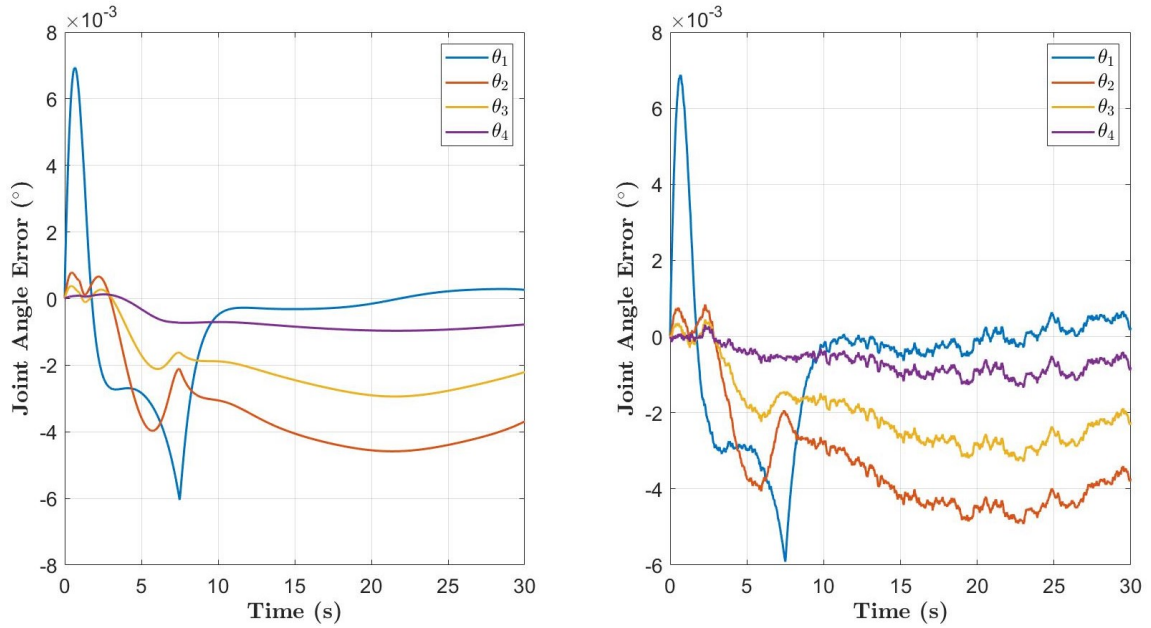

Figure 55: Scenario 3 Joint angle errors. Undisturbed (left), maximum disturbances (right).

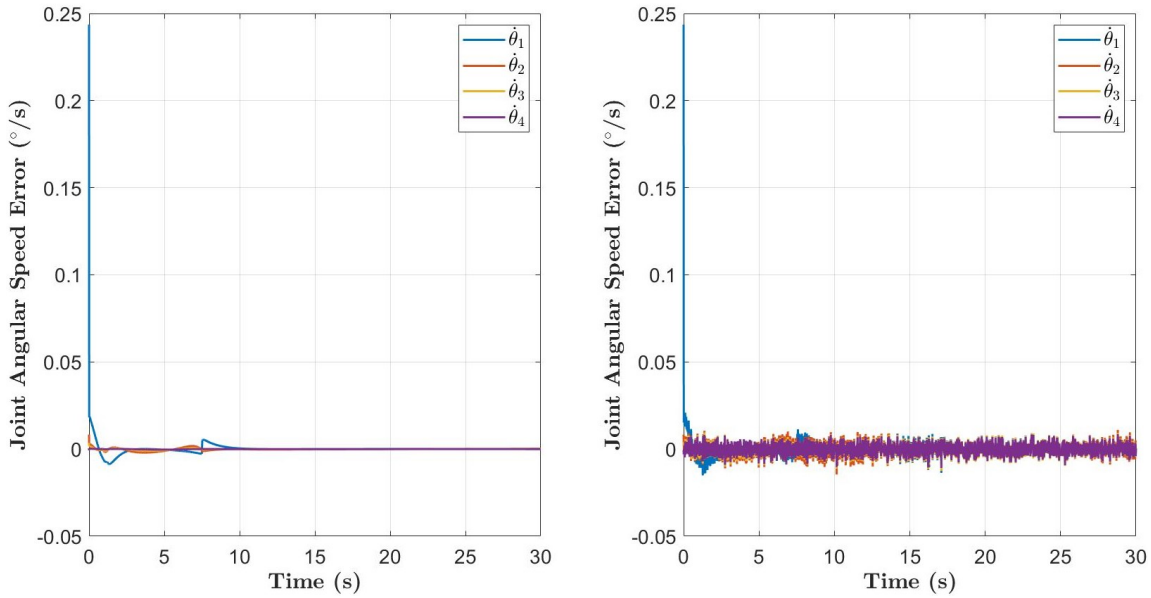

Figure 56: Scenario 3 Joint angular speed errors. Undisturbed (left), maximum disturbances (right).

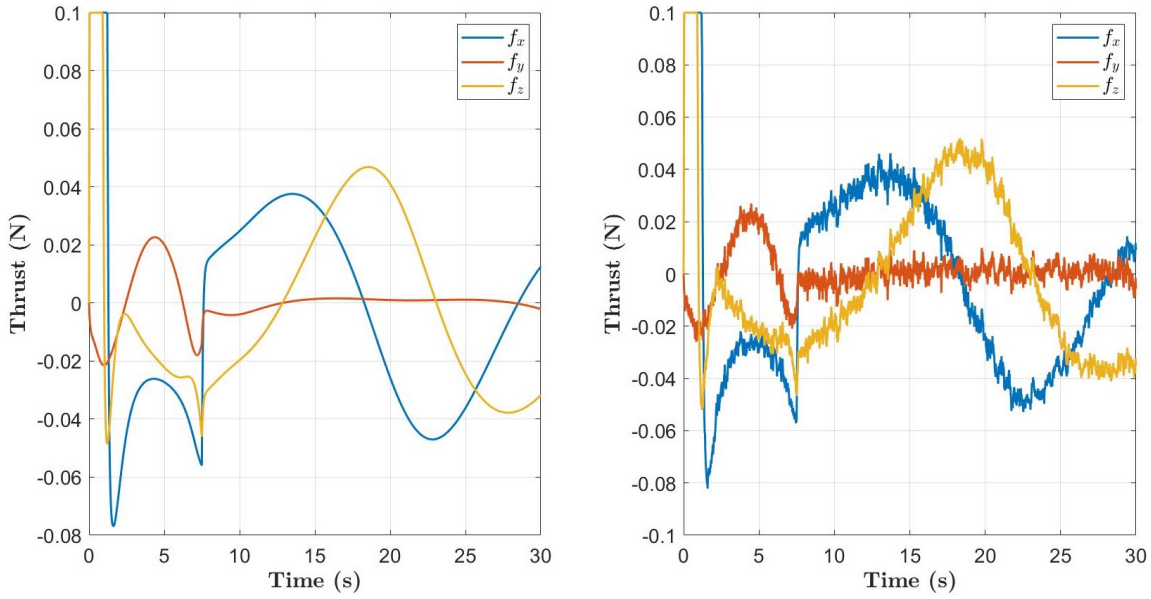

Figure 57: Scenario 3 Thrust profiles. Undisturbed (left), maximum disturbances (right).

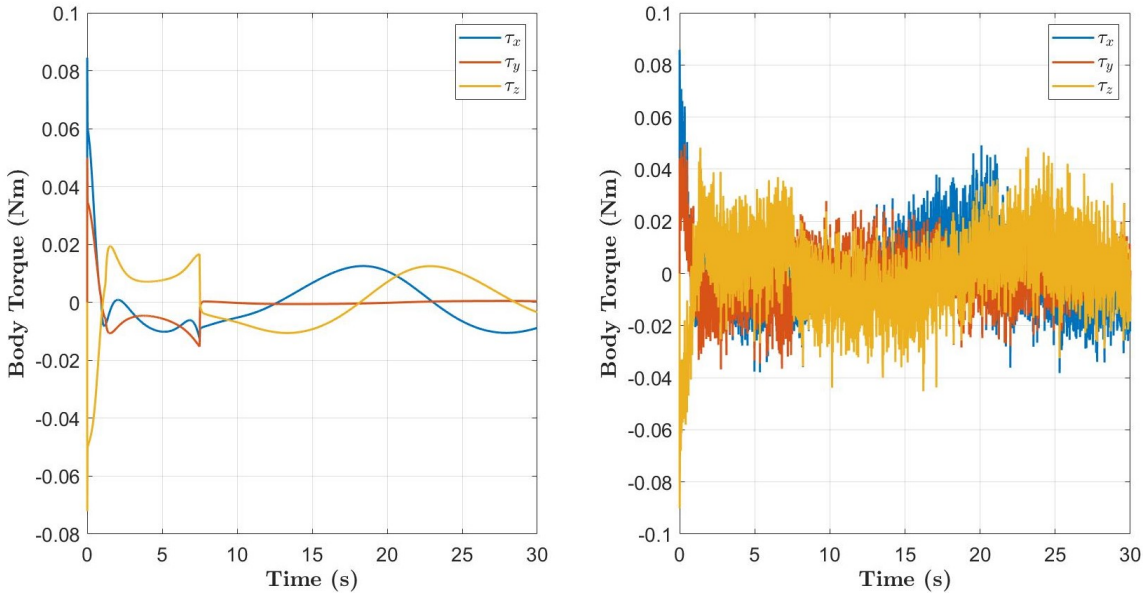

Figure 58: Scenario 3 Base torque profiles. Undisturbed (left), maximum disturbances (right).

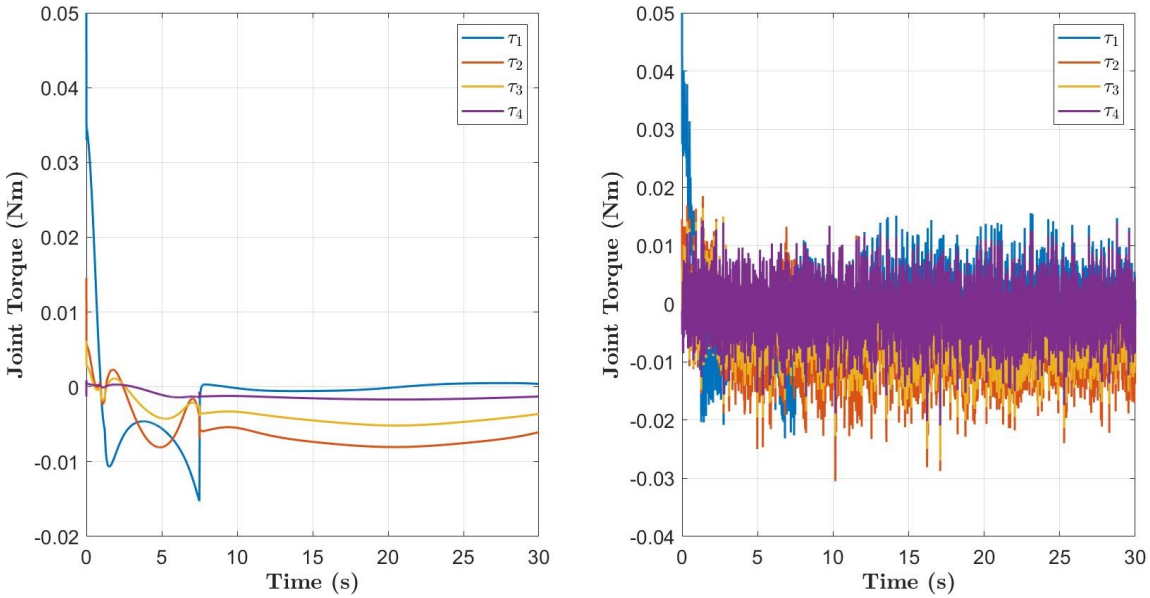

Figure 59: Scenario 3 Joint torque profiles. Undisturbed (left), maximum disturbances (right).

Bibliography

1. John Olson, Steven Butow, Eric Felt, Thomas Cooley, and Joel Mozer. State of the Space Industrial Base 2021: Infrastructure and Services for Economic Growth and National Security. Technical Report November, United States Space Force, Defense Innovation Unit, and Air Force Research Laboratory, 2021.

2. Michael A. Luu and Daniel E. Hastings. Valuation of on-orbit servicing in proliferated low-earth orbit constellations. *Accelerating Space Commerce, Exploration, and New Discovery Conference, ASCEND 2020*, pages 1–14, 2020.

3. Gardell G. Gefke, Alex Janas, Joseph Pellegrino, Matthew Sammons, and Benjamin Reed. Advances in robotic servicing technology development. *AIAA SPACE 2015 Conference and Exposition*, pages 1–9, 2015.

4. Benjamin B. Reed, Robert C. Smith, Bo Naasz, Joseph Pellegrino, and Charles Bacon. The Restore-L servicing mission. *AIAA Space and Astronautics Forum and Exposition, SPACE 2016*, (September):1–8, 2016.

5. Benjamin B. Reed, Charles E. Bacon, and Bo Naasz. Designing spacecraft to enable robotic servicing. *AIAA SPACE and Astronautics Forum and Exposition, SPACE 2017*, (203999):1–9, 2017.

6. Justin C. Brannan, Craig R. Carignan, and Brian J. Roberts. Hybrid strategy for evaluating on-orbit servicing, assembly, and manufacturing technologies. *Accelerating Space Commerce, Exploration, and New Discovery Conference, ASCEND 2020*, pages 1–10, 2020.

7. Small Spacecraft Technology. State - of - the - Art Small Spacecraft Technology. *NASA Technical Report*, (October):NASA/TP–2020–5008734, 2020.

8. BryceTech. Smallsats by the Numbers 2021. Technical report, BryceTech, 2021.

9. Canadian Space Agency. Canadarm2 ' s data sheet, 2019.

10. Mark Garcia. Special Purpose Dexterous Manipulator, 2018.

11. Grayson Huggins, Ali Talaksi, E. Glenn Lightsey, Dawn Andrews, Daniel Cavender, Carlos Diaz, Donald McQueen, Hunter Williams, John Baker, and Matthew Kowalkowski. Development of a cubesat-scale green monopropellant propulsion system for nasa's lunar flashlight mission. *AIAA Scitech 2021 Forum*, (January):1–15, 2021.

12. Bradley Cheetham, Thomas Gardner, and Alec Forsman. Cislunar autonomous positioning system technology operations and navigation experiment (Capstone). *Accelerating Space Commerce, Exploration, and New Discovery Conference, ASCEND 2020*, pages 1–8, 2020.

13. Dakota Wenberg, Michael Kutzer, Levi Devries, John Gregory, Michael Sanders, and Jin S. Kang. Development of On-orbit Assembly Demonstrator in 3U CubeSat Form Factor. *IEEE Aerospace Conference Proceedings*, pages 1–11, 2020.

14. Ansley N Knight, Trent R Tetterton, Alec J Engl, Peter M Sinkovitz, Bailee J Ward, and Jin S Kang. Design and Development of On-orbit Servicing CubeSat-class Satellite. In *Small Satellite Conference*, 2020.

15. Michael C.F. Bazzocchi and Houman Hakima. A cubesat-based robotic asteroid sampling mission. *AIAA Scitech 2020 Forum*, 1 PartF(January):1–12, 2020.

16. Eric A Butcher and Chad Wenn. Lyapunov-Based Control of Kinematically Coupled Close-Proximity Spacecraft Feature Point Dynamics. *AAS/AIAA Space Flight Mechanics Meeting*, pages 1–20, 2021.

17. Steve Ulrich, Alvar Saenz-Otero, and Itzhak Barkana. Passivity-based adaptive control of robotic spacecraft for proximity operations under uncertainties. *Journal of Guidance, Control, and Dynamics*, 39(6):1441–1450, 2016.

18. Josep Virgili-Llop, Costantinos Zagaris, Richard Zappulla, Andrew Bradstreet, and Marcello Romano. A convex-programming-based guidance algorithm to capture a tumbling object on orbit using a spacecraft equipped with a robotic manipulator. *International Journal of Robotics Research*, 38(1):40–72, 2019.

19. Bryce Doerr, Keenan Albee, Monica Ekal, Richard Linares, and Rodrigo Ventura. Safe and Uncertainty-Aware Robotic Motion Planning Techniques for Agile On-Orbit Assembly. In *AAS/AIAA Astrodynamics Specialist Conference*, pages 1–21, 2021.

20. Thai Chau Nguyen-Huynh and Inna Sharf. Adaptive reactionless motion and parameter identification in postcapture of space debris. *Journal of Guidance, Control, and Dynamics*, 36(2):404–414, 2013.

21. Gangqi Dong and Z. H. Zhu. Autonomous robotic capture of Non-cooperative target by Vision-based kinematic control. *AIAA SPACE 2015 Conference and Exposition*, pages 6–11, 2015.

22. Katrina Alsup. *Robotic Spacecraft Hopping: Application and Analysis*. Master's thesis, Naval Postgraduate School, 2018.

23. Kazuya Yoshida. Engineering test satellite VII flight experiments for space robot dynamics and control: Theories on laboratory test beds ten years ago, now in orbit. *International Journal of Robotics Research*, 22(5):321–335, 2003.

24. Brandi Dean. Space Shuttle Canadarm Robotic Arm Marks 25 Years in Space, 2006.

25. Simon Nolet, Edmund Kong, and David W. Miller. Design of an algorithm for autonomous docking with a freely tumbling target. *Modeling, Simulation, and Verification of Space-based Systems II*, 5799(May 2005):123, 2005.

26. Tom A. Mulder. Orbital Express autonomous rendezvous and capture flight operations Part 2 of 2: ARC exercises 4, 5, and end-of-life. *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, (August):1–24, 2008.

27. Jennifer Harbaugh. On-Orbit Servicing, Assembly, and Manufacturing 2 (OSAM-2), 2021.

28. Fei Liu, Quan Hu, Lei Liu, Xiaohui Li, Weihui Liu, Wen Wen, and Jingchao Zhao. Dynamics and Control for Trajectory Tracking of Parallel Space Capturing Robot. In *AAS/AIAA Astrodynamics Specialist Conference*, pages 1–16, 2020.

29. Yinghong Jia and Arun K. Misra. Trajectory planning for a space robot actuated by control moment gyroscopes. *Journal of Guidance, Control, and Dynamics*, 41(8):1833–1841, 2018.

30. Thomas R Kane and David A Levinson. Formulation of Equations of Motion for Complex Spacecraft. *Journal of Guidance, Control, and Dynamics*, 3(2):99–112, 1980.

31. Alexander Crain and Steve Ulrich. Experimental validation of pseudospectral-based optimal trajectory planning for free-floating robots. *Journal of Guidance, Control, and Dynamics*, 42(8):1726–1742, 2019.

32. Y. Cao, V. J. Modi, C. W. de Silva, and A. K. Misra. Dynamics of a class of novel manipulators with deployable links. *Astrodynamics Specialist Conference*, (c):623–633, 2000.

33. Jessica S. Friz, Jason Neuhaus, and P. Sean Kenney. Modeling conservation of angular momentum for robotic in-space assembly systems. *AIAA Scitech 2020 Forum*, 1 PartF(January):1–34, 2020.

34. Randy Spicer and Jonathan Black. Simulating the dynamics and control of a free-flying small satellite with a robotic manipulator for 3d printing. *AIAA Scitech 2020 Forum*, 1 PartF(January):1–19, 2020.

35. Frederick Tasker and Carl Henshaw. Managing contact dynamics for orbital robotic servicing missions. *Space 2008 Conference*, (September), 2008.

36. Elisabeth A. Gambone. Seeker cubesat control system. *AIAA Scitech 2020 Forum*, 1 PartF(January):1–7, 2020.

37. Richard Volpe. Techniques For Collision Prevention, Impact Stability, And Force Control By Space Manipulators. *Teleoperation and Robotics in Space*, pages 175–212, 1994.

38. Carl Glen Henshaw. The DARPA Phoenix Spacecraft Servicing Program : Overview and Plans for Risk Reduction. In *The 12th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, pages 1–9, 2014.

39. Erik E. Komendera, William R. Doggett, John T. Dorsey, Thomas J. Debus, Kris Holub, and Sean P. Dougherty. Control system design implementation and preliminary demonstration for a tendon-actuated lightweight in-space MANipulator (TALISMAN). *AIAA SPACE 2015 Conference and Exposition*, 2015.

40. Dongsheng Guo, Feng Xu, and Laicheng Yan. New pseudoinverse-based path-planning scheme with PID characteristic for redundant robot manipulators in the presence of noise. *IEEE Transactions on Control Systems Technology*, 26(6):2008–2019, 2018.

41. Romeo Ortega and Mark W. Spong. Adaptive motion control of rigid robots: A tutorial. *Automatica*, 25(6):877–888, 1989.

42. Harry Berghuis, Romeo Ortega, and Henk Nijmeijer. A Robust Adaptive Controller for Robot Manipulators. In *IEEE International Conference on Robotics and Automation*, pages 1876–1881, Nice, 1992.

43. Hiranya S. Jayakody, Lingling Shi, Jay Katupitiya, and Nathan Kinkaid. Robust adaptive coordination controller for a spacecraft equipped with a robotic manipulator. *Journal of Guidance, Control, and Dynamics*, 39(12):2699–2711, 2016.

44. Gaurav Misra. *Convex Optimization Based Planning and Control Methods for Space-Robotic Systems*. Dissertation, Rutgers University, 2019.

45. Riccardo Bonalli, Abhishek Cauligi, Andrew Bylard, and Marco Pavone. GuSTO: Guaranteed sequential trajectory optimization via sequential convex programming. *Proceedings - IEEE International Conference on Robotics and Automation*, 2019-May:6741–6747, 2019.

46. Gaurav Misra and Xiaoli Bai. Task-constrained trajectory planning of free-floating space-robotic systems using convex optimization. *Journal of Guidance, Control, and Dynamics*, 40(11):2857–2870, 2017.

47. Ezinne E. Uzo-Okoro, Prakash Manandhar, Daniel Erkel, Mary Dahl, Emily Kiley, Olivier Deweck, and Kerri Cahoy. Optimization of on-orbit robot assembly of small satellites. *Accelerating Space Commerce, Exploration, and New Discovery Conference, ASCEND 2020*, pages 1–18, 2020.

48. Andrew W. Verstraete, Nicole M. St.Louis, and Jennifer Hudson. Geo robotic servicer trajectory optimization. *AIAA Space and Astronautics Forum and Exposition, SPACE 2016*, (September):1–9, 2016.

49. Alexander A. Soderlund, Sean Phillips, Anonto Zaman, and Christopher Petersen. Autonomous satellite rendezvous and proximity operations via geometric control methods. *AIAA Scitech 2021 Forum*, pages 1–13, 2021.

50. Thomas R. Kane, Peter W. Likins, and David A. Levinson. *Spacecraft Dynamics*. McGraw-Hill, 1983.

51. Thomas R. Kane and David A. Levinson. Multibody dynamics. *Journal of Applied Mechanics*, 50(4):1071–1078, 1983.

52. Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics: Modelling, Planning and Control*. Number 9781846286414. Springer, 2009.

53. Richard Paul. Manipulator Cartesian Path Control. *IEEE Transactions on Systems, Man and Cybernetics*, 9(11):702–711, 1979.

54. Chun Shin Lin, Po Rong Chang, and J. Y.S. Luh. Formulation and Optimization of Cubic Polynomial Joint Trajectories for Industrial Robots. *IEEE Transactions on Automatic Control*, 28(12):1066–1074, 1983.

55. Jincan Li, Mingyu Gao, Zhiwei He, and Yuxiang Yang. Circular trajectory planning with pose control for Six-DOF manipulator. *Proceedings of 2018 IEEE 4th Information Technology and Mechatronics Engineering Conference, ITOEC 2018*, (Itoec):1950–1955, 2018.

56. Q. J. Ge and B. Ravani. Geometric construction of Bézier motions. *Journal of Mechanical Design, Transactions of the ASME*, 116(3):749–755, 1994.

57. Claudio Altafini. The de casteljau algorithm on SE(3). *Nonlinear control in the Year 2000*, (November 2007):23–34, 2007.

58. Claudio Altafini. Inverse kinematics along a geometric spline for a holonomic mobile manipulator. In *IEEE International Conference on Robotics and Automation*, pages 1265–1270, Seoul, 2001.

59. L L C NanoRacks. NanoRacks CubeSat Deployer (NRCSD) Interface Definition Document (IDD) NRCSD List of Revisions. Technical Report Idd, Nanoracks, 2013.

60. Motion Genesis LLC. MotionGenesis Kane Tutorial, 2016.

61. Cody J Allard. *Modular Software Architecture for Complex Multi- Body Fully-Coupled Spacecraft Dynamics*. Dissertation, University of Colorado, 2018.

62. Cody Allard, Manuel Diaz Ramos, Hanspeter Schaub, Patrick Kenneally, and Scott Piggott. Modular software architecture for fully coupled spacecraft simulations. *Journal of Aerospace Information Systems*, 15(12):670–683, 2018.

63. Ke Yang, Xu Yang Wang, Tong Ge, and Chao Wu. Simulation platform of underwater quadruped walking robot based on MotionGenesis Kane 5.3 and central pattern generator. *Journal of Harbin Institute of Technology (New Series)*, 21(3):24–29, 2014.

64. Amateur Radio on the International Space Station. International Space Station Two-Line Element, 2021.

65. James R. Wertz. Orbits and Astrodynamics. In James R. Wertz, Davis F. Everett, and Jeffery J. Puschell, editors, *Space Mission Engineering: The New SMAD*, chapter 9, pages 212–214. Microcosm Press, Torrance, 2011.

66. Marshall H. Kaplan. *Modern Spacecraft Dynamics and Control*. John Wiley Sons, New York, 1976.

67. Dennis C Pak. *Linearized equations for J2 perturbed motion relative to an elliptical orbit*. Master of science, San Jose State University, 2005.

68. Xiang Zhu. Optical design of space cameras for automated rendezvous and docking systems. In Khanh D. Pham and Genshe Chen, editors, *Sensors and Systems for Space Applications XI*, volume 1064106, pages 1–12. SPIE, 2018.

69. David H. Curtis and Richard G. Cobb. Satellite articulation tracking using computer vision. *Journal of Spacecraft and Rockets*, 56(5):1478–1491, 2019.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
14–03–2022	Master's Thesis	Sept 2020 — Mar 2022

4. TITLE AND SUBTITLE

COUPLED ORBIT-ATTITUDE DYNAMICS AND CONTROL OF A CUBESAT EQUIPPED WITH A ROBOTIC MANIPULATOR

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Charles M. Carr

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

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

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

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENY-MS-22-M-283

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

AFXX/XXXX
Building XXX
WPAFB OH 45433-7765
DSN XXX-XXXX, COMM 937-XXX-XXXX
Email: first.last@us.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

XXXX/XXXX

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

12. DISTRIBUTION / AVAILABILITY STATEMENT

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

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This research investigates the utility and expected performance of a robotic servicing CubeSat. The coupled orbit-attitude dynamics of a 6U CubeSat equipped with a four-link serial manipulator are derived. A proportional-integral-derivative controller is implemented to guide the robot through a series of orbital scenarios, including rendezvous and docking following ejection from a chief spacecraft, repositioning the end effector to a desired location, and tracing a desired path with the end effector. Various techniques involving path planning and inverse differential kinematics are leveraged. Simulation results are presented and performance metrics such as settling time, state errors, control use, and system robustness are analyzed. Recommended improvements to the system design and the model are discussed.

15. SUBJECT TERMS

subject terms here

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Captain First M. Last, AFIT/ENG
U	U	U	UU	XXX	19b. TELEPHONE NUMBER *(include area code)* (937) 255-3636, ext XXXX; first.last@afit.edu