Application of an Adaptive Controller on an Air-Bearing Vehicle-Manipulator System

Samuel Thomas Lovelace
Application of an Adaptive Controller on an Air-Bearing Vehicle-Manipulator System

by

Samuel Thomas Lovelace

Bachelor of Science
Aerospace Engineering
Florida Institute of Technology
2022

A thesis
submitted to the College of Engineering and Science
at Florida Institute of Technology
in partial fulfillment of the requirements
for the degree of

Master of Science
in
Aerospace Engineering

Melbourne, Florida
August, 2023
We the undersigned committee hereby approve the attached thesis

Application of an Adaptive Controller on an Air-Bearing Vehicle-Manipulator System

by Samuel Thomas Lovelace

Madhur Tiwari, Ph.D.
Assistant Professor
Aerospace, Physics and Space Sciences
Committee Chair

Seong Hyeon Hong, Ph.D
Assistant Professor
Mechanical and Civil Engineering
Outside Committee Member

Eric Swenson, Ph.D
Associate Professor
Aerospace, Physics and Space Sciences
Committee Member

David Fleming, Ph.D.
Associate Professor and Department Head
Aerospace, Physics and Space Sciences
Abstract

Title:
Application of an Adaptive Controller on an Air-Bearing Vehicle-Manipulator System

Author:
Samuel Thomas Lovelace

Major Advisor:
Madhur Tiwari, Ph.D.

The growing demand for on-orbit servicing and assembly as well as orbital debris capture and removal poses many engineering challenges. One primary challenge is that of position and attitude control of a spacecraft-manipulator system while performing these operations. Standard control approaches may be used for cases where the dynamics of the objects that the spacecraft-manipulator system interacts with are known are still plausible. However, there will likely be some uncertainty and lack of knowledge of the dynamic properties of orbital debris and parts required for servicing. Therefore, many have proposed the use of adaptive control schemes which can recalculate the gains of the controller in real time to adjust to the coupled spacecraft and capture object dynamics. This thesis focuses on the comparison of an adaptive variable structure controller and a standard PID controller as applied to a 3-DOF air-bearing vehicle and 2-DOF manipulator system. The non-linear, time-varying dynamic model utilized in Simulink to achieve a high-fidelity simulation. This research shows that for a mass...
increase of the end-effector link of 19% of the total system mass, which amounts to 2.39 kg, and an increase of 19% in the moment of inertia of the end-effector link, the PID controller was unable to control the system while the adaptive controller successfully controlled the system within design requirements.
# Table of Contents

Abstract ................................................................. iii

List of Figures .......................................................... viii

List of Tables ............................................................ x

Acknowledgments ........................................................ xi

Dedication ................................................................. xii

1 Introduction ............................................................ 1
   1.1 Thesis Objective ................................................. 1
   1.2 Background and Motivation ...................................... 2
   1.3 Orbital Debris .................................................... 2
      1.3.1 Current Work ............................................... 5
   1.4 On-Orbit Servicing and Assembly ............................... 5
      1.4.1 ETS-VII ..................................................... 6
      1.4.2 Orbital Express ............................................. 9
      1.4.3 Northrop Grumman Mission Extension Vehicles ......... 10
      1.4.4 Current Work .............................................. 11
   1.5 Spacecraft-Manipulator Systems ............................... 12

2 Theory ................................................................. 14
2.1 Robotic Manipulators ............................................. 14
   2.1.1 Kinematics .............................................. 15
   2.1.2 Dynamics .............................................. 17

2.2 Spacecraft-Manipulator Systems ....................................... 19

2.3 Adaptive Variable Structure Control Law .......................... 25
   2.3.1 Phase I .............................................. 26
   2.3.2 Phase II ............................................. 27
   2.3.3 Phase III ............................................ 27

3 Implementation ........................................................ 29
   3.1 Introduction ............................................... 29

3.2 Air-Bearing Vehicle .................................................. 29
   3.2.1 Hardware .............................................. 29
   3.2.2 Software .............................................. 30

3.3 Dynamic Properties ................................................. 31
   3.3.1 Thrust Characterization .................................. 31

3.4 Simulation .......................................................... 35
   3.4.1 Simulink Model Blocks ..................................... 36
      3.4.1.1 Initial Conditions .................................... 36
      3.4.1.2 Desired Path ........................................ 36
      3.4.1.3 Calculate Error ....................................... 36
      3.4.1.4 Controller ............................................ 37
      3.4.1.5 Thrust Command ...................................... 39
      3.4.1.6 Dynamic Model Update ................................ 39
      3.4.1.7 Dynamics Propagation ................................ 40
   3.4.2 Dynamic Model Validation .................................... 41
3.5 Controller Tuning .............................................. 44
  3.5.1 PID Gains .............................................. 45
  3.5.2 Adaptive Controller Gains ......................... 45

4 Model Testing ..................................................... 47
  4.1 Introduction ............................................. 47
  4.2 Testing - Object Manipulation ......................... 48
    4.2.1 Test 1 - Base Case ............................... 48
    4.2.2 Test 2 - 10% Increase ......................... 48
    4.2.3 Test 3 - 19% Increase ......................... 49
    4.2.4 Test 4 - 25% Increase ......................... 49

5 Results .......................................................... 50
  5.1 Introduction ............................................ 50
  5.2 Test 1 - Base Case .................................. 50
  5.3 Test 2 - 10% Increase ................................ 54
  5.4 Test 3 - 19% Increase ................................ 58
  5.5 Test 4 - 25% Increase ................................ 62
  5.6 All Cases ................................................. 66
    5.6.1 Adaptive Controller ............................ 67
    5.6.2 PID Controller ..................................... 71
  5.7 Discussion ................................................. 75

6 Conclusion .......................................................... 76
List of Figures

1.1 Cataloged Objects in Earth Orbit by Object Type [1] .......................... 4
1.2 RemoveDEBRIS mothercraft satellite photographed by the ISS [4] ... 4
1.3 Artist’s Rendition of ClearSpace-1 Mission ................................. 6
1.4 ETS-VII during vibration testing ........................................... 7
1.5 ETS-VII Docking Mechanism - Male ..................................... 8
1.6 ETS-VII Docking Mechanism - Female .................................. 8
1.7 Orbital Express Satellites: ASTRO (left) and NextSat(right) [7] ... 9
1.8 Artist’s Rendering of the MEV-1 Docking ............................... 11

2.1 Denavit-Hartenberg Coordinate Frames and Parameters .................. 16
2.2 Spacecraft-Manipulator Definitions (Source [18]) ...................... 20

3.1 Air-Bearing Vehicle-Manipulator System ............................... 30
3.2 Simulink Simulation Model for ABV-Manipulator ..................... 35
3.3 Adaptive Controller Algorithm (Source [20]) .......................... 37
3.4 Schematic Diagaram from Schmitt Trigger ............................ 38
3.5 Primary Angle Definitions for ABV-Manipulator System (Source [29]) 42
3.6 Angular Rates of Manipulator Joints and Base Vehicle ............... 43
3.7 Position of Manipulator Joints and Base Vehicle ....................... 44

5.1 PID and Adaptive Controller for 0% Mass/Inertia Added ............... 51
5.2 Control Inputs from Adaptive Controller for 0% Mass/Inertia Added . 52
List of Tables

3.1 Thrust from Silvent MJ6 Nozzles at 60psi . . . . . . . . . . . . . . . . 32
3.2 ABV and Manipulator Mass and Moment of Inertia . . . . . . . . . . . 34
3.3 ABV and Manipulator Mass and Moment of Inertia . . . . . . . . . . . 34
3.4 Controller Design Requirements . . . . . . . . . . . . . . . . . . . . . 45
3.5 PID Controller Gains . . . . . . . . . . . . . . . . . . . . . . . . . . . 45
3.6 Adaptive Controller Gains . . . . . . . . . . . . . . . . . . . . . . . . . 46
5.1 Rise Time for Each State for All Cases Tested (All units in seconds) . . 70
5.2 Maximum Overshoot Values for Each State for All Cases Tested . . . . 70
5.3 Steady State Error for Each State for All Cases Tested . . . . . . . . . 71
5.4 Rise Time for Each State for All Cases Tested (All units in seconds) . . 73
5.5 Maximum Overshoot Value for Each State for All Cases Tested . . . . 74
5.6 Steady State Error for Each State for All Cases Tested . . . . . . . . . 74
First, I would like to thank my mom for making me go to college and providing continuous support throughout my undergrad and graduate career. Thanks to my dad and grandpa for always listening to my explanations about the things that I am learning or working on.

I would like to thank Dr. Wilde for first giving me a job in the ORION Lab and for fostering my passion for robotics and spacecraft controls. I would like to thank Dr. Tiwari for his help while developing this thesis.

Thanks to Peyton, Elizabeth and John for their help during the thrust and moment of inertia testing. Thanks to Jessica for listening to my rants while doing research for this thesis.
Dedication

This thesis is dedicated to two of the hardest working people I know, my mom and dad. Thank you for everything you have done to enable me to get my bachelor’s and master’s degree. I love you both!
Chapter 1

Introduction

1.1 Thesis Objective

The objective of this thesis is the simulation of an adaptive control law for coordinated motion control of a spacecraft-manipulator system. The simulation utilizes a non-linear, time-varying dynamic model for a 3-DOF air-bearing vehicle (ABV) and 2-DOF planar manipulator. The adaptive control law is modified from the literature to include two translational forces in conjunction with the attitude and manipulator joint control torques. The simulation was developed in Simulink. A comparison is made between the adaptive controller and a standard PID controller when different values of mass and moment of inertia are added to the system. This type of adaptive controller could be used for many different applications related to rendezvous and docking, orbital debris capture and removal as well as on-orbit servicing and assembly.
1.2 Background and Motivation

Spacecraft-manipulator systems can potentially be used for a multitude of missions including debris removal, on-orbit servicing and on-orbit assembly. In performing any of these missions, maintaining a desired position and attitude of the base spacecraft while the manipulator is in motion is still of crucial importance for avoiding unplanned contact with the resident space object (RSO). The work in this thesis focuses on controlling the position and attitude of a spacecraft-manipulator system when handling an object which has increased the total system mass and moment of inertia. This object could represent a debris object that has been captured for removal or a tool of some sort that may be used during an on-orbit assembly operation.

1.3 Orbital Debris

The growing issue of orbital debris has become a significant concern for the international space community. With more countries and private companies launching satellites and conducting space missions, the amount of debris in space has been increasing exponentially. Orbital debris refers to the man-made objects orbiting the Earth that no longer serve a useful purpose. These objects can range from tiny paint flecks to large satellites and can remain in orbit for years, even decades. These objects not only pose a threat to spacecraft and astronauts, but also have the potential to impact our daily lives on Earth by disrupting communication and navigation systems.

As of February 2023, there are 26,584 objects in Earth orbit which are cataloged by the U.S. Space Surveillance Network (SSN) [1]. This includes active and defunct satellites, rocket bodies, mission-related debris and fragmentation debris. A similar database maintained by the Union of Concerned Scientists (UCS) has a catalog of 5,465 active satellites as of April 2022 [2]. Figure 1.1 shows the distribution of the
cataloged objects in Earth orbit. The number of spacecraft being cataloged is around 10,000. This number includes active and defunct satellites. Thus, according to the UCS, the number of active and defunct satellites is roughly the same. The objects tracked by the SSN and the UCS primarily include larger objects. However, there still exists a plethora of debris at such small sizes that make it difficult to track. Some scientific models predict that there could be more than 1 million pieces of space debris that are between 1 cm and 10 cm in size and more than 130 million pieces that are between 1 mm and 1 cm in size [3]. Because they are in orbit, these pieces of debris travel at extremely high velocities, posing a significant collision risk to operational satellites, spacecraft, and the International Space Station (ISS). Even a small piece of debris can cause catastrophic damage or complete destruction upon impact. The increasing population of space debris amplifies the likelihood of collisions and raises concerns about the safety and sustainability of space activities.

As of June 2023, there have been no successful debris removal missions performed. However, there are a couple of missions of interest which are detailed below. The first mission of interest, known as RemoveDEBRIS was launched in a cargo resupply to the ISS in 2018 [4]. The purpose of this mission was to demonstrate various technologies which may be useful in future debris removal missions. Specifically, the demonstration of a net to capture and detumble space debris, various vision based navigation technologies and the use of a harpoon for debris capture.

After being unloaded from the SpaceX Dragon capsule, the fully functional RemoveDEBRIS mothercraft satellite, shown in Figure 1.2 was then berthed by the ISS into a 405 km altitude orbit. While in orbit, the mothercraft released a 2U cubesat known as DSAT#1 to serve as an artificial debris target. Then, a 5 m diameter net was ejected from the mothercraft and successfully expanded, captured the DSAT#1 debris and contracted to enclose the satellite. The net and captured satellite then de-
Figure 1.1: Cataloged Objects in Earth Orbit by Object Type [1]

Figure 1.2: RemoveDEBRIS mothercraft satellite photographed by the ISS [4]
orbited at an accelerated rate due to the increase in aerodynamic drag. Among the net experiment, a target was extended from the mothercraft and a harpoon was ejected and contacted the target. This is another demonstration of a technology which could be used to capture tumbling and or uncooperative space debris. Lastly, a couple different vision-based navigation algorithms were tested in order to raise their technology readiness level (TRL). Various 3-DOF and 6-DOF vision-based navigation algorithms were tested with real-time flight data. Furthermore, model-based tracking algorithm was tested utilizing data from a flashing LiDAR.

1.3.1 Current Work

Unfortunately, no significant space debris removal missions have been attempted since RemoveDEBRIS in 2018. However, one notable mission called ClearSpace-1 being developed by ClearSpace and backed by the European Space Agency (ESA) is planned to launch in 2026. The goal of the ClearSpace-1 mission is very specific as it plans to rendezvous, capture and deorbit a 112kg VESPA adapter which launched on a Vega rocket in 2013 [5]. Due to its simple shape and structural rigidity, the VESPA adapter makes for a good first attempt at debris removal. Figure 1.3 shows an artists rendition of the ClearSpace-1 chaser satellite and its four robotic arms as it captures the VESPA adapter in low Earth orbit.

1.4 On-Orbit Servicing and Assembly

A field of space exploration tangent to debris removal is on-orbit servicing and assembly. This may not directly deal with the mitigation of the amount of space debris objects but rather it is an enabling technology for future space missions. On-orbit servicing refers to the capability of performing maintenance, repairs, upgrades, or other
operations on satellites once they are in orbit around Earth. It involves sending specialized robotic systems, spacecraft, or astronauts to rendezvous and interact with the target satellite or spacecraft to perform various tasks. On-orbit servicing can extend the lifetime of existing satellites, indirectly reducing space debris. Surprisingly, in the past 35 years, there have been very few missions focusing on demonstrating satellite-to-satellite rendezvous, docking and proximity operations available in open literature found by the author. In fact, there are only four notable missions from 1997 to 2022 with sufficient information available. These missions are the JAXA Engineering Test Satellite VII (ETS-VII), DARPA Orbital Express and the Northrop Grumman Mission Extension Vehicles, MEV-1 and MEV-2. The first two missions listed were technology demonstrations while the MEV missions were practical applications. The details of each mission are discussed below.

1.4.1 ETS-VII

In a joint mission between ESA and NASDA (now JAXA), the Engineering Test Satellite VII (ETS-VII) was the first mission to demonstrate various satellite-to-satellite rendezvous, docking and proximity operations. Two satellites were launched as a con-
nected pair into a 550 km circular, low Earth orbit on November 28th, 1997. The chaser spacecraft had dimensions 2.6 m x 2.3 m x 2 m and a mass of 2450 kg. The target spacecraft was much smaller having dimensions 0.65 m x 1.7 m x 1.5 m and a mass of 410 kg. Figure 1.4 shows both the target and chaser in their launch configuration. The chaser satellite was outfitted with a 6-DOF robotic arm having total length of 2 m. In the stowed configuration, the robotic arm system had dimensions 50 cm x 48 cm x 48 cm and a mass of 45 kg. The joints are actuated by brushless DC motors and harmonic drive gear trains. During various demonstrations, the robotic arm was teleoperated from a ground station via TDRS. The target satellite utilized Nitrogen cold gas thrusters for attitude control while the chaser satellite utilized a hydrazine propulsion system and three reaction wheels.

![Figure 1.4: ETS-VII during vibration testing](image)

According to [6], relative navigation is achieved by the chaser satellite through the use of three sensors optimized for different ranges. For short range, a proximity camera is used for distances less than 10 meters. Middle range navigation utilizes a laser radar for distances between 2 and 500 meters. Lastly, GPS receivers are utilized
for long distances greater than 500 meters. Throughout the lifetime of the ETS-VII mission, three sets of experiments were performed in order to verify and validate the rendezvous and docking capabilities on orbit. The first experiment was performed was an autonomous rendezvous and docking from a starting distance of 2 meters. In this experiment, the primary sensor utilized was the proximity camera sensor. The docking mechanism utilized during these experiments is shown below. The male end of the mechanism, mounted on the chaser spacecraft is shown in Figure 1.5 and the female end is shown in Figure 1.6.

![ETS-VII Docking Mechanism - Male](image1)

Figure 1.5: ETS-VII Docking Mechanism - Male

![ETS-VII Docking Mechanism - Female](image2)

Figure 1.6: ETS-VII Docking Mechanism - Female

The second set of experiments was similar to the first set but the starting distance for rendezvous and docking was 12km instead of 2m. For this operation, the GPS receiver and radar laser were used for relative navigation at large distance. The proximity camera sensor was used once the satellites were within 10 meters. The third set
of experiment performed on orbit were slightly different than the previous two sets. First, the chaser satellite was operated remotely from a ground station. Then, an obstacle avoidance maneuver was performed. Lastly, an R-bar rendezvous approach was performed.

1.4.2 Orbital Express

The Orbital Express mission was managed by the United States Defense Advanced Research Projects Agency (DARPA) and led by a team of engineers at NASA’s Marshall Space Flight Center. Similar to ETS-VII, two satellites were launched as a connected pair on March 8th, 2007 into a 492 km circular, low Earth orbit. The target satellite called NextSat was roughly 1m x 1m x 1m and had a mass of 225 kg. The chaser satellite, called ASTRO had dimensions 1.8 m x 1.8 m x 5 m and a mass of 1100 kg [7]. Figure 1.7 shows the ASTRO and NextSat satellites.

![Orbital Express Satellites: ASTRO (left) and NextSat(right) [7]](image)

The chaser satellite was outfitted with a 6-DOF robotic arm with a total length of 3 m. In its stowed configuration, it had dimensions of 0.65 m x 0.49 m x 1.86 m and a mass of 71 kg. No sources provide information on the type of motors used for each of the joints on the manipulator arm. While on orbit, the operations performed by the robotic arm were prescripted and/or autonomous using a visual based servoing control loop.
Both the ASTRO and NextSat satellite utilize Hydrazine based monopropellant thrusters for 6-DOF attitude control.

1.4.3 Northrop Grumman Mission Extension Vehicles

The first demonstration of a rendezvous and docking of two commercial satellites was completed in February of 2020 by Northrop Grumman and IntelSat [9]. The Mission Extension Vehicle 1 (MEV-1), designed and manufactured Northrop Grumman, docked with the IntelSat 901. The purpose of this mission was to extend the life of the IntelSat 901 as its propellant tank was nearly empty.

After being launched in October 2019 from Kazakhstan into a Geostationary Transfer Orbit, the MEV-1 spent the next months performing a series of engine burns to raise its orbit to a circular orbit 300 km above Geosynchronous orbit known as a graveyard orbit. The IntelSat 901 reached proximity with the MEV-1 in the graveyard orbit in early February. After a number of practice approaches over a two week span, the final approach was conducted on February 25th 2020. The MEV-1 utilized a suite of visual and infrared cameras along with a side-scanning LIDAR to perform relative navigation[10].

Once in range, the docking was accomplished by inserting a probe into the apogee engine until it passed through the throat of the nozzle. Then, the probe deployed a mechanism of finger like members allowing the MEV-1 to grasp the IntelSat and pull it in contact with three stanchions ultimately providing rigid support. With this configuration, the MEV-1 and the IntelSat 901 behave as a single rigid structure with the MEV-1 taking over the propulsion responsibilities. Figure 1.8 shows an artist’s rendering of the MEV-1 (left) docking with the IntelSat 901 (right).
On-board each of the MEV’s was a sensor suite of three different imaging devices to be used for Rendezvous and Proximity Operations (RPO). The devices include a set of visible light cameras, infrared cameras and a LIDAR.

1.4.4 Current Work

As the demand for on-orbit servicing and assembly or debris removal capabilities grow, companies like AstroScale, OrbitFab, and others are aimed at demonstrating the feasibility of certain operations in the coming years.

Astroscale has 4 majors services that they offer: End of Life, Space Debris Removal, Life Extension and In-Orbit Inspection. As of writing, AstroScale’s ELSA-d mission has successfully achieved rendezvous and docking in a similar manner to the Orbital Express mission. This is a major milestone as it is a demonstration of the core technologies necessary for on-orbit satellite servicing in LEO. Similar to AstroScale, a startup company called Starfish Space, based in Washington state recently launched their first demonstration mission called OtterPup. Their company goals include: Active Life Ex-
tension, Orbital Debris Removal and Autonomous Transportation. Another company, based in Colorado named Orbit Fab, has been working on development of a refueling interface for satellites. Their main product is called RAFTI, which is a quick disconnect mechanism for docking and refueling of spacecraft in orbit. As of writing, no hardware has been flown yet but a large purchase by the U.S. Space Force for satellite refueling in orbit is set for 2025.

1.5 Spacecraft-Manipulator Systems

Clearly, the use of spacecraft-manipulator systems is a key technology in enabling orbital debris removal, on-orbit servicing and on-orbit assembly. Various methods have been proposed attempting to solve the issue of coordinated control of spacecraft-manipulator systems. A Transposed-Jacobian type controller was proposed by [11] while others [12] proposed a controller which estimates the angular momentum produced by the manipulator motion and pass that data in a feedforward manner to the spacecraft attitude control system. A quaternion based feedback controller was proposed by [13] in an attempt to avoid orientation singularities and ease the inverse kinematics representation. [14] focuses on implementing realistic models for thrusters and control moment gyros (CMG) in simulating coordinated motion control of a figure eight trajectory for the manipulator end-effector.

Throughout the literature, various dynamic models are used for the spacecraft-manipulator systems[11, 15, 16, 17]. They all represent the same dynamics but with some slight modifications and removal or addition of terms. A complete and condensed derivation and discussion of the 6+N DOF equations of motion for a spacecraft-manipulator system can be found in [18]. The dynamic model used in this thesis is derived from the one outlined in [18].
The adaptive controllers used in this thesis are proposed by [19] and [20]. Similar work has been done by [21] who propose an adaptive Jacobian based control method and [22] who propose a projection-based adaptive control scheme. Furthermore, [23] proposes an adaptive controller for the case of capturing a debris object with unknown dynamics. In general, work has been done in the past on adaptive control for spacecraft [24] [25] [26].
Chapter 2

Theory

The primary purpose of this thesis is the simulation of an adaptive control law for a spacecraft-manipulator system. Since the motion of a manipulator mounted on a spacecraft will induce motion of the base spacecraft due to the conservation of angular momentum, the dynamics are coupled and can be expressed in a compact form. First, a brief presentation of the kinematics and dynamics of robotic manipulators is given. Then, the dynamics of the spacecraft-manipulator system are outlined. Next, the details of the adaptive variable structure control law are presented.

2.1 Robotic Manipulators

A robotic manipulator is typically comprised of a series of rigid links connected by joints. These joints can be rotary (revolute) or linear (prismatic), allowing the robot arm to articulate and move in different directions. The joints are actuated by motors that provide the necessary power and control for the robot to perform precise movements. At the end of the robot arm, there is usually an end effector or gripper, which enables the robot to interact with and manipulate objects. The end effector can
take various forms depending on the application requirements, such as gripper claws, suction cups, welding tools, or specialized sensors. Much like any other system which moves, the kinematics and dynamics can be described mathematically.

2.1.1 Kinematics

Kinematics is the study of motion without concern for the forces causing such motion. For robotic manipulators, this is useful because it provides the foundation for understanding and controlling the motion of the manipulator. Specifically, the representation of a manipulator’s kinematics is useful for motion planning, workspace and singularity analysis and inverse kinematics. When representing the kinematics of a robotic manipulator, there is a typical convention used known as the Denavit-Hartenberg (DH) parameters. In short, the D-H Parameters are a set of geometric parameters which relate the current link and the next link. From these, a set of homogeneous transformation matrices can be constructed which transforms the coordinate frames from link to link.

Before defining each of the DH parameters, the coordinate frames are defined. First, the z-axis of the $i^{th}$ link is chosen to be parallel to the axis of rotation of the $i^{th}$ link. The origin of the $i^{th}$ frame, $O_i$ at the intersection of axis $z_i$ with the common normal between axis $z_{i-1}$ and $z_i$. Then, the x-axis of the $i^{th}$ frame is chosen along the common normal to $z_{i-1}$ and $z_i$ with the positive direction from link $i$ to $i + 1$. Lastly, the direction of the y-axis for the $i^{th}$ frame is chosen to complete the right hand Cartesian frame. Figure 2.1 shows the coordinate frames for an example set of two links. Once the coordinate frames are defined for each link, the DH parameters can be defined. The first parameter is the distance along the common normal between $O_{i-1}$ and $O_i$. This parameter is sometimes referred to as the link length and represented by the variable $a$. Next, the distance from the intersection of the $x_{i-1}$ and $z_i$ to the origin of the $i^{th}$ frame
Figure 2.1: Denavit-Hartenberg Coordinate Frames and Parameters

along $z_i$. This parameter is sometimes referred to as the link offset and represented by the variable $d$. The third parameter is the rotation angle from $z_{i-1}$ to $z_i$ about the $x_{i-1}$ axis. This parameter is referred to as the link twist and is denoted by the variable $\alpha$. The last parameter is the rotation angle between $z_{i-1}$ and $z_i$ about the $x_{i-1}$ axis. This parameter is referred to as the link angle and is denoted by the variable $\theta$. With the DH parameters defined, a transformation matrix can be setup in order to relate the consecutive links. In general, a homogeneous transformation matrix consists of the $3 \times 3$ rotation matrix and a $3 \times 1$ translation vector. The resulting matrix is $4 \times 4$ and has the form

$$T_{1}^{0} = \begin{bmatrix} R_{1}^{0} & p_{1}^{0} \\ 0 & 1 \end{bmatrix}$$

(2.1)

The homogeneous transformation matrix constructed using the DH parameters results
in

\[
A_i^{-1} = \begin{bmatrix}
  c_\theta_i & -s_\theta_i c_\alpha_i & s_\theta_i s_\alpha_i & a_i c_\theta_i \\
  s_\theta_i & c_\theta_i c_\alpha_i & -c_\theta_i s_\alpha_i & a_i s_\theta_i \\
  0 & s_\alpha_i & c_\alpha_i & d_i \\
  0 & 0 & 0 & 1
\end{bmatrix}
\] (2.2)

where \(s_i\) and \(c_i\) are the sine and cosine of the angle. Once the transformation matrix for each link is defined, they can be used to determine the transformation from the base to the end effector through successive transformations. This is accomplished such that

\[
T_e^0 = T_i^0 T_i^1 T_i^2 \ldots T_i^j
\] (2.3)

The computation of the end-effector position as a function of the joint angles is known as the forward kinematics.

### 2.1.2 Dynamics

As the complexity of the robotic manipulator increases, the equations of motion increase as well. Using a Lagrangian Dynamics approach to determine the equations of motion simplifies the task in comparison to using Newton’s Second Law. The Lagrangian is variable comprised of the kinetic and potential energy. That is,

\[
\mathcal{L} = T - V
\] (2.4)

where \(T\) is the system kinetic energy and \(V\) is the system potential energy. Then, the equations of motion are determined by

\[
\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}} - \frac{\partial \mathcal{L}}{\partial q} = \zeta
\] (2.5)
where $q$ is the vector of generalized state variables and $\zeta$ is the vector of generalized forces and torques.

The kinetic energy of each link based on the translational and rotational velocity is computed as

$$T_i = \frac{1}{2} m_i \dot{p}_i^T \dot{p}_i + \frac{1}{2} \omega_i^T R_i I_i R_i^T \omega_i$$  \hspace{1cm} (2.6)$$

where $m_i$ is the mass of the $i^{th}$ link and $p_i$ is the position vector of the center of mass of the $i^{th}$ link. Also, the link angular velocity is given by $\omega_i$ and the inertia is given by $I_i$. For this work, the contribution to the kinetic energy from the motors is not considered. To aid in the calculation of the equations of motion, the kinetic energy can be expressed in terms of the joint angles. To do this, the geometric Jacobian is utilized. The Jacobian matrix of a manipulator relates the joint velocities to the translational and angular velocity of the end-effector. Mathematically, that is

$$\dot{p}_e = J_T(q) \dot{q}$$ \hspace{1cm} (2.7)$$

$$\omega_e = J_R(q) \dot{q}$$ \hspace{1cm} (2.8)$$

Then, the translational and angular velocity vectors can be expressed as

$$\dot{p}_i = J_{P_i}^i \dot{q}_1 + \cdots + J_{P_i}^i \dot{q}_i = J_{P_i}^i \dot{q}$$ \hspace{1cm} (2.9)$$

$$\omega_i = J_{O_i}^i \dot{q}_1 + \cdots + J_{O_i}^i \dot{q}_i = J_{O_i}^i \dot{q}$$ \hspace{1cm} (2.10)$$

where, for revolute joints,

$$J_{P_j}^i = z_{j-1} \times (p_i - p_{j-1})$$ \hspace{1cm} (2.11)$$

$$J_{O_j}^i = z_{j-1}$$ \hspace{1cm} (2.12)$$
Finally, the kinetic energy can be expressed as

\[ T_i = \frac{1}{2} m_i \dot{q}_i^T J_{P_{i}}^{(l_{i})T} J_{P_{i}}^{(l_{i})} \ddot{q}_i + \frac{1}{2} \dot{q}_i^T J_{O_{i}}^{(l_{i})T} R_i I_i R_i^T J_{O_{i}}^{(l_{i})} \ddot{q}_i \]  \hspace{1cm} (2.13)

Next, the total potential energy is calculated as

\[ V_i = -m_i g_i^T p_i \]  \hspace{1cm} (2.14)

Then, determining the Lagrangian from Equation 2.4 and performing the derivatives according to Equation 2.5, the resulting equations of motion become

\[ H(q) \ddot{q} + C(q, \dot{q}) = \tau \]  \hspace{1cm} (2.15)

where the \( n \times n \) inertia matrix \( H(q) \), is given by

\[ H(q) = \sum_{i=1}^{n} m_i J_{P_{i}}^{(l_{i})T} J_{P_{i}}^{(l_{i})} + J_{O_{i}}^{(l_{i})T} R_i I_i R_i^T J_{O_{i}}^{(l_{i})} \]  \hspace{1cm} (2.16)

and the Coriolis matrix, \( C(q, \dot{q}) \) is given by

\[ C(q, \dot{q}) = \dot{H}(q) \dot{q} - \frac{1}{2} (\frac{\partial}{\partial q} (q^T H(q) \dot{q}))^T + (\frac{\partial V}{\partial q})^T \]  \hspace{1cm} (2.17)

### 2.2 Spacecraft-Manipulator Systems

A spacecraft-manipulator system is defined as any spacecraft with a maneuverable manipulator attached to it. Because of the conservation of angular momentum, the motion of the manipulator causes a rotation to the base spacecraft proportional to the ratio of the moments of inertia and opposite in direction. Below is a brief presentation of the \( 6 + N \)-DOF equations of motion. The next chapter outlines the modification
made to the equations of motion for the simulation of the 3-DOF vehicles used in this thesis. The definitions of the coordinate frames and geometric parameters are defined as shown in Figure 2.2.

Utilizing a similar Lagrangian dynamics approach to the one outlined above for the manipulator dynamics, the dynamics of the spacecraft-manipulator system can be represented in a form similar to Equation 3.15. First, define $\Phi = [\Phi_B, \Phi_M]$ as the state vector where $\Phi_B$ contains the linear and angular position of the spacecraft-manipulator system such that $\Phi_m = [x, y, z, \alpha, \beta, \gamma]$ and $\Phi_M$ contains the joint angles of each link of the manipulator such that $\Phi_M = [q_1, q_2, q_3, \ldots q_n]$. Then, the equations of motion can be written more compactly as

$$H \ddot{\Phi} + \dot{H} \dot{\Phi} + C(\Phi, \dot{\Phi}) = \tau$$  \hspace{1cm} (2.18)

This dynamic model contains the base spacecraft mass and inertia properties, the manipulator mass and inertia properties and the coupled dynamics for the spacecraft-manipulator system.
manipulator. The expanded form of the equations of motion is

\[
\begin{bmatrix}
H_0 & H_{0m} \\
H_{0m}^T & H_m
\end{bmatrix}
\begin{bmatrix}
\ddot{x}_0 \\
\ddot{q}
\end{bmatrix} +
\begin{bmatrix}
\dot{H}_0 & \dot{H}_{0m} \\
\dot{H}_{0m}^T & \dot{H}_m
\end{bmatrix}
\begin{bmatrix}
\dot{x}_0 \\
\dot{q}
\end{bmatrix} +
\begin{bmatrix}
c_0 \\
c_m
\end{bmatrix} =
\begin{bmatrix}
F \\
\tau
\end{bmatrix}
\]

(2.19)

The submatrices from the above equation are outlined below. The 6×6 base-spacecraft inertia matrix \(H_0\) is calculated as

\[
H_0 = \begin{bmatrix}
m_{tot}I_{3,3} & -m_{tot}r_{0c}^x \\
m_{tot}r_{0c}^x & H_s
\end{bmatrix}
\]

(2.20)

where \(m_{tot}\) is the total system mass, \(I_{3,3}\) is the identity matrix of size 3×3 and \(r_{0c}^x\) is the position vector of the total system center of mass relative to the base-spacecraft center of mass in skew symmetric matrix form. The submatrix \(H_s\) contains the moments of inertia of the spacecraft-manipulator system about the base-spacecraft center of mass such that

\[
H_s = \sum_{i=1}^{n} (I_i - m_i r_{0i}^x r_{0i}^x) + I_0
\]

(2.21)

where \(I_0\) is the inertia tensor of the base-spacecraft, \(I_i\) and \(m_i\) are the inertia tensor and mass of the \(i^{th}\) link of the manipulator and \(r_{0i}^x\) is the position vector to the center of mass of the \(i^{th}\) link in skew symmetric matrix form.

The dynamic coupling inertia matrix, \(H_{0m}\) contains the contributions from the base-spacecraft and manipulator to the kinetic energy of the spacecraft-manipulator system. Explicitly, it is

\[
H_{0m} = \begin{bmatrix}
J_{Ts} \\
H_{Sq}
\end{bmatrix}
\]

(2.22)

where \(J_{Ts}\) gives the contribution to the system kinetic energy from the combination of the effect of the manipulator joint rate and the base-spacecraft linear velocity. Futher-
more, the upper submatrix is given by

\[ J_{Ts} = \sum_{i=1}^{n} m_i J_{Ti} \]  (2.23)

and

\[ H_{Sq} = \sum_{i=1}^{n} (I_i J_{Ri} + m_i r_0^x J_{Ti}) \]  (2.24)

where \( J_{Ti} \) and \( J_{Ri} \) are the linear velocity and angular velocity Jacobians, respectively, of the \( ith \) link as derived in [27]. Next, the manipulator inertia matrix \( H_m \) is identical to that of a fixed-based manipulator and is calculated as

\[ H_m = \sum_{i=1}^{n} m_i J_{Ti}^T J_{Ti}^{(li)} + J_{Ri}^T R_i I_i R_i^T J_{Ri} \]  (2.25)

Next, the terms in the \( \dot{H} \) matrix are expanded. First, the time derivative of the base-spacecraft inertia matrix, \( \dot{H}_0 \) is given by

\[
\dot{H}_0 = \begin{bmatrix}
0_{3,3} & -m_{tot} \dot{r}_0^x \\
m_{tot} \dot{r}_0^x & \dot{H}_s
\end{bmatrix}
\]  (2.26)

with the inertia derivative submatrix \( \dot{H}_s \) is given by

\[
\dot{H}_s = \sum_{i=1}^{n} (\dot{I}_i - m_i (\dot{r}_0^x r_0^x + r_0^x \dot{r}_0^x)) + \dot{I}_0
\]  (2.27)

Next, the time derivative of the dynamic-coupling inertia matrix, \( \dot{H}_{0m} \) is given by

\[
\dot{H}_{0m} = \begin{bmatrix}
\dot{J}_{Ts} \\
\dot{H}_{Sq}
\end{bmatrix}
\]  (2.28)
where

\[ \dot{J}_{Ts} = \sum_{i=1}^{n} m_i \dot{J}_T i \]  

(2.29)

and

\[ \dot{H}_{Sq} = \sum_{i=1}^{n} (\dot{I}_i J_{Ri} + I_i \dot{J}_{Ri} + m_i (\ddot{r}_{0i} J_T i + r_{0i} \dot{J}_T i)) \]  

(2.30)

Finally, the time derivative of the manipulator submatrix, \( \dot{H}_m \) is given by

\[ \dot{H}_m = \sum_{i=1}^{n} (J_{T_i} I_i J_{Ri} + J_{T_i} \dot{I}_i J_{Ri} + J_{Ri} I_i \dot{J}_{Ri} + m_i (J_{T_i} J_{T_i} + J_{T_i} \dot{J}_T i)) \]  

(2.31)

Lastly, the nonlinear submatrices \( c_0 \) and \( c_m \) are calculated as

\[ c_0 = -\frac{1}{2} \frac{\partial}{\partial x_0} (\dot{x}_0^T H_0 \ddot{x}_0 + \dot{q}^T H_m \dot{q} + \dot{x}_0^T H_{0m} \dot{q} + \dot{q}^T H_{0m} T \ddot{x}_0) \]  

(2.32)

\[ c_m = -\frac{1}{2} \frac{\partial}{\partial q} (\dot{x}_0^T H_0 \ddot{x}_0 + \dot{q}^T H_m \dot{q} + \dot{x}_0^T H_{0m} \dot{q} + \dot{q}^T H_{0m} T \ddot{x}_0) \]  

(2.33)

Throughout the equations of motion, various position vectors to the center of mass coordinate systems are required for the computation of certain terms. In order to keep track of these position vectors, a homogeneous transformation matrix is defined which transforms the previous coordinate frame to the current coordinate frame. This process is started by defining the transformation from the inertial frame to the base-spacecraft center of mass frame. This transformation is chosen to be represented using the standard 1-2-3 rotation sequence of Euler Angles. The position vector of the 0\(^{th}\) transformation is the position of the base-spacecraft in the inertial frame.

Below is an outline of the transformation matrices utilized in the Dynamic Model Update block in the Simulink model developed for this thesis.

The position and orientation of the spacecraft-manipulator system with respect to the inertial frame \( J \) is represented using homogeneous transformation matrices. There
exists a coordinate system at the center of mass of each link denoted by \( L_i \) and another coordinate system at each joint denoted by \( \mathcal{J}_i \). For the base vehicle, \( L_0 = \mathcal{J}_0 \).

First, the transformation from the inertial frame to the base-spacecraft center of mass frame is given by

\[
T_{L_0}^{\mathcal{J}_0} = T_{\mathcal{J}_0}^{\mathcal{J}_0} = \begin{bmatrix} R_{L_0}^{\mathcal{J}_0} & r_0^{\mathcal{J}_0} \\ 0_{1,3} & 1 \end{bmatrix}
\] (2.34)

where \( R_{L_0}^{\mathcal{J}_0} \) is the rotation matrix containing the Euler angles of the base-spacecraft relative to the inertial frame and \( r_0^{\mathcal{J}_0} \) is the position vector of the base-spacecraft center of mass in the inertial frame.

Next, the transformation from inertial frame to the frame positioned at the first joint of the manipulator is given by

\[
T_{\mathcal{J}_1}^{\mathcal{J}_0} = T_{\mathcal{J}_0}^{\mathcal{J}_1} = T_{\mathcal{J}_0}^{\mathcal{J}_0} = \begin{bmatrix} R_{\mathcal{J}_1}^{\mathcal{J}_0} & b_0^{\mathcal{J}_0} \\ 0_{1,3} & 1 \end{bmatrix}
\] (2.35)

where \( R_{\mathcal{J}_1}^{\mathcal{J}_0} \) is the rotation matrix containing the orientation of the manipulator base joint frame with respect to the base-spacecraft frame represented in a 1-2-3 sequence of Euler angles and \( b_0^{\mathcal{J}_0} \) is the position vector of the manipulator base joint frame with respect to the base-spacecraft frame.

After defining the two transformations above, the matrices which transform the frames throughout the manipulator utilize the Denavit-Hartenberg convention as described in the previous section. Explicitly, the \( i^{th} \) joint can be determined through matrix multiplication such that

\[
T_{\mathcal{J}_i}^{\mathcal{J}_0} = T_{\mathcal{J}_{i-1}}^{\mathcal{J}_i} T_{\mathcal{J}_{i-1}}^{\mathcal{J}_{i-1}} = T_{\mathcal{J}_{i-1}}^{\mathcal{J}_0} T(q_{i-1}, d_{i-1}, \alpha_{i-1}, a_{i-1} + b_{i-1})
\] (2.36)
and then the link center of mass frames can be calculated from

\[ T_{L_i}^J = T_{J_i}^J T_{L_i}^{J_i} = T_{J_i}^J T(q_i, d_i, \alpha_i, a_i) \]  \( (2.37) \)

Finally, the position vector \( r_{0i}^J \) to each center of mass coordinate frame can be extracted from their respective homogeneous transformation matrix. Then, the relative position vector between the base-spacecraft center of mass frame and the link center of mass frames, \( r_{0i}^J \) are given by

\[ r_{0i}^J = r_i^J - r_0^J \]  \( (2.38) \)

and the position vector of the total system center of mass is given by

\[ r_C^J = \frac{1}{m_{\text{tot}}} \sum_{i=0}^{n} m_i r_i^J \]  \( (2.39) \)

Lastly, the relative position of the system center of mass with respect to the base-spacecraft center of mass is given simply by

\[ r_{0C}^J = r_C^J - r_0^J \]  \( (2.40) \)

### 2.3 Adaptive Variable Structure Control Law

The adaptive variable structure control law is implemented such that three different control laws are used depending on the state of the error defined as \( e_1 = \Phi - \Phi_d \) and \( e_2 = \dot{\Phi} - \dot{\Phi}_d \). Then, [19] shows that the error dynamics can be written as

\[ \dot{e}_1 = e_2 \]  \( (2.41) \)
\[ \dot{e}_2 = f(e_2) + \tau^* + h[e(t)] \]  

(2.42)

where \( f(e_2) = -\hat{H}^{-1}\dot{C}e_2 \) and \( h[e(t)] \) represents the system uncertainties as a function of the error \( e(t) \) at the current time step. The virtual torque, \( \tau^* \) is defined from implementing some representation of system uncertainties in the dynamics such that

\[ \tau = \hat{H}\tau^* \]  

(2.43)

where

\[ \hat{H} = H + \Delta H \]  

(2.44)

with \( H \) being the nominal dynamics and \( \Delta H \) representing the uncertainties. The same is applied to the matrix \( C \) such that

\[ \hat{C} = \bar{C} + \Delta \bar{C} \]  

(2.45)

and from Equation 2.18

\[ \bar{C} = \dot{H}\dot{\Phi} + C(\Phi, \dot{\Phi}) \]  

(2.46)

### 2.3.1 Phase I

The goal of the control law for phase I is to drive the system along a clockwise path on the error phase plane. This control law is applied while \( \sigma \) is not equal to zero. The control law is given by

\[ \tau_i^* = -f_i(e_2) - M_{max_i} \text{sgn}(\sigma_i(e_1, e_2)) \]  

(2.47)
where

\[
\sigma_i(e) = \begin{cases} 
\dot{e}_{1i} - |(-2k_i e_{1i})^{0.5}| & \text{for } e_{1i} < 0 \\
\dot{e}_{1i} + |(2k_i e_{1i})^{0.5}| & \text{otherwise} 
\end{cases}
\] (2.48)

with \(k_i\) being a tunable parameter.

### 2.3.2 Phase II

Then, when \(\sigma_i = 0\), the control law for Phase II takes the form

\[
\tau^*_i = -f_i(e_2) + M_i(t)
\] (2.49)

where

\[
M_i(t) = k_{ci}(t) - k_{mi}(t) + M_i(t - T)
\] (2.50)

with the adaptive gains \(k_{ci}\) and \(k_{mi}\) are given by

\[
k_{ci} = \frac{\dot{e}_{1i}^2(t)}{2e_{1i}(t)}
\] (2.51)

\[
k_{mi} = \frac{\dot{e}_{1i}^2(t) - \dot{e}_{1i}^2(t - T)}{2[e_{1i}(t) - e_{1i}(t - T)]}
\] (2.52)

The stability of the chosen gains for the phase II adaptive control law are proven by [19] and [20] using Lyapunov stability analysis.

### 2.3.3 Phase III

Lastly, the control law for phase III is chosen as a conventional sliding mode control (SMC) such that

\[
\tau^* = -f_i(e_2) - \lambda_i \dot{e}_{1i} - M_{max,i} \text{sat}(\hat{\sigma}_i(e))
\] (2.53)
where $\hat{\sigma}_i(e) = \lambda_i e_{1i} + \dot{e}_{1i}$ is the sliding surface and sat() is the saturation function given by

$$\text{sat}(\mathcal{X}) = \begin{cases} 
\text{sgn}(\mathcal{X}) & \mathcal{X} > \epsilon \\
\frac{\mathcal{X}}{\epsilon} & \mathcal{X} \leq \epsilon
\end{cases} \quad (2.54)$$

with $\epsilon$ being the boundary-layer width and a tunable parameter.
Chapter 3

Implementation

3.1 Introduction

The simulation developed for this thesis is in preparation for the implementation of the adaptive control law on the air-bearing vehicles owned by The Autonomy Lab at Florida Tech. Some key details about the vehicles are provided below. Then, an explanation of the thrust and moment of inertia characterization experiments is given. Next, the major blocks of the Simulink model are explained. Then, a validation of the non-linear, time-varying dynamic model is explained. Lastly, the tuning of the PID and adaptive controllers is explained.

3.2 Air-Bearing Vehicle

3.2.1 Hardware

The Autonomy Lab at Florida Tech has two air-bearing vehicles named Bob and Charlie. The vehicles are nearly identical however Bob has a three-link planar manipulator
mounted on its frame. Figure 3.1 shows the air-bearing vehicle and manipulator. The tanks of Nitrogen on the top of the vehicle supplies the gas for the air-bearings and the cold gas thrusters. Both vehicles are equipped with 8 individually controllable cold gas thrusters providing 3-DOF control of the vehicle.

![Air-Bearing Vehicle-Manipulator System](image)

Figure 3.1: Air-Bearing Vehicle-Manipulator System

### 3.2.2 Software

The software is written in C++ and the executable file is configured to run on startup of the computer. The on-board control software performs all the necessary functions required for the guidance, navigation and control of the air-bearing vehicle. A detailed explanation of the software can be found in [28]. Since the control software is written in C++, the code for the control law implemented in simulation is easily transferable.
for implementation on the actual ABV.

3.3 Dynamic Properties

Since the control law being simulated requires a dynamic model of the air-bearing vehicle-manipulator system, some approximation of the dynamic properties is required. First, the total mass of the vehicle can be measured directly. The mass is 12.6 kg. Next, the moment of inertia of the vehicle about it’s vertical axis is of interest. Work has been done in the past to experimentally determine the moment of inertia for the air-bearing vehicle but the manipulator has been mounted on the vehicle for this thesis and thus the moment of inertia has changed. It is not expected that the change in moment of inertia is significant since the total mass of the manipulator is 0.685 kg, roughly 5% of the total mass of the vehicle.

3.3.1 Thrust Characterization

In order to experimentally determine the moment of inertia of the ABV-Manipulator system, first the force of the thrusters must be characterized. The cold gas thrusters on the ABV utilize a solenoid valve controlled by an Arduino Uno and AdaFruit Motor Shield with Silvent MJ6 nozzles. The thrusters can be commanded to fire for a specific amount of time through the control center GUI in various combinations so as to produce a translational or rotational force or torque. For the characterization of the nozzle thrust, four tests were conducted for each pair of nozzles located on each side of the ABV. The nozzles were commanded to fire for 5 seconds and OptiTrack data for the position and attitude was collected during this time. From the position data, the force of the thrusters can be determined by evaluating the kinematics.

First, assume that the acceleration of the vehicle is constant over the five second
Table 3.1: Thrust from Silvent MJ6 Nozzles at 60psi

<table>
<thead>
<tr>
<th>Thrusters</th>
<th>Force (N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/2</td>
<td>0.236</td>
</tr>
<tr>
<td>3/4</td>
<td>0.206</td>
</tr>
<tr>
<td>5/6</td>
<td>0.248</td>
</tr>
<tr>
<td>7/8</td>
<td>0.301</td>
</tr>
<tr>
<td>AVG</td>
<td>0.248</td>
</tr>
</tbody>
</table>

experiment. Then, the position of the vehicle is given by

\[ s = s_0 + v_0 t + \frac{1}{2} a t^2 \]  

(3.1)

where \( s \) is the displacement, \( v_0 \) is the initial velocity (zero in this case) and \( a \) is the acceleration. Knowing the time span of the experiment and calculating the displacement from the OptiTrack data, the acceleration can be calculated. Then, once the mass and acceleration are known, the force can be calculated using Newton’s Second Law such that

\[ F_2 = \frac{1}{2} m_{ABV} a \]  

(3.2)

where \( F_2 \) is the force of one thruster when two thrusters are firing. It was shown in \[28\] that a thrust loss occurred for each thruster depending on the number of thrusters being fired since they are all connected to the same Nitrogen tank. To provide a pure translational or rotational input, at least two thrusters must be fired. Thus, the force being applied by each thruster is \( F_2 \) and the total force is \( 2 \cdot F_2 \). Table 3.1 shows a summary of the values obtained during this characterization experiment. It is interesting to note the slight deviance between various sets of thrusters.

Once the average force of the thrusters is determined, a similar experiment can be performed in order to experimentally determine the moment of inertia of the ABV-Manipulator system. Applying a pure moment by firing opposite thrusters and mea-
suring the angular displacement, the angular acceleration can be calculated as

\[ \theta = \theta_0 + \omega_0 t + \frac{1}{2} \alpha t^2 \]  \hspace{1cm} (3.3)

where \( \theta, \omega, \alpha \) are the angular position, velocity and acceleration, respectively. From the angular acceleration, the moment of inertia about the vertical axis of the ABV-Manipulator system can be calculated as

\[ I_{zz} = \frac{T}{\alpha} \]  \hspace{1cm} (3.4)

where the torque is calculated using the average force of the thrusters found above as

\[ T = 2 \cdot F_2 \cdot l \]  \hspace{1cm} (3.5)

where \( l \) is the moment arm of the thrusters. Table 3.2 shows the results of the moment of inertia characterization experiment. This is the moment of inertia for the ABV-Manipulator system with the manipulator in the stowed configuration and represents the minimum moment of inertia. Clearly, when the manipulator joint angles change, the total system moment of inertia will change.

For the simulation, it was assumed that the thrust produced from each pulse was equal to the average thrust listed in Table 3.1. The torque about the vehicle’s vertical axis is produce by firing 2 opposite thrusters each with a moment arm of \( l = 0.1235 \) m. Thus, the thrust value used in the simulation is \( F_2 = 0.248 \) (N) and consequently the torque is \( T_z = 0.0306 \) (N-m). Note this is the force of one thruster firing. To achieve a perfect translation in any direction, two thrusters must be firing in order to not produce a torque about the vehicle’s vertical axis. When this occurs, the thrust produce is twice as much as the listed force and torque above.
The dynamic properties of the manipulator were obtained through the use of Autodesk Inventor. A CAD model of the manipulator is publicly available online. The arm was disassembled and each link was weighed using a scale. Then, the density and material were chosen in Autodesk Inventor to match the mass found when measuring the physical arm. Then, a calculation of the global moments of inertia of each link of the manipulator was performed by Autodesk Inventor using the iProperties tab. Similar to the ABV, the only moment of inertia of interest for the manipulator is the value about its vertical axis since the manipulator is planar due to the mounting and motor configurations. The mass and vertical axis moment of inertia for the manipulator links can be found in Table 3.2.

The calculation of the position vectors to the center of mass of each link on the manipulator requires the use of the Denavit-Hartenberg parameters outlined in the previous chapter. The DH-Parameters for the manipulator mounted on the vehicle are shown in Table 3.3. The manipulator is a two-link planar manipulator so the angle offset and distance offset parameters are zero.

### Table 3.2: ABV and Manipulator Mass and Moment of Inertia

<table>
<thead>
<tr>
<th>Body</th>
<th>Mass (kg)</th>
<th>Inertia (kg-m²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABV-Manipulator</td>
<td>12.6</td>
<td>0.449</td>
</tr>
<tr>
<td>Link 1</td>
<td>0.369</td>
<td>0.115</td>
</tr>
<tr>
<td>Link 2</td>
<td>0.323</td>
<td>0.101</td>
</tr>
</tbody>
</table>

### Table 3.3: ABV and Manipulator Mass and Moment of Inertia

<table>
<thead>
<tr>
<th>Link</th>
<th>( a_i (m) )</th>
<th>( \alpha_i )</th>
<th>( d_i )</th>
<th>( \theta_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.147</td>
<td>0</td>
<td>0</td>
<td>( q_1 )</td>
</tr>
<tr>
<td>2</td>
<td>0.311</td>
<td>0</td>
<td>0</td>
<td>( q_2 )</td>
</tr>
</tbody>
</table>
3.4 Simulation

The simulation for this thesis was developed in Simulink. It utilizes the non-linear, time-varying dynamic model (Section 2.2) for the ABV-Manipulator along with the adaptive controller (Section 2.3). The simulation uses a fixed time-step of 0.05 seconds and the ode4(Runge-Kutta) solver.

The simulation is comprised mainly of MATLAB Function blocks which each implement a different part of the control loop. Figure 3.2 shows the complete Simulink model.

![Simulink Simulation Model for ABV-Manipulator](image)

For completeness, the functionality of each block is briefly described below. The blocks below are listed in order from left to right as shown in Figure 3.2.
3.4.1 Simulink Model Blocks

3.4.1.1 Initial Conditions

This block contains the initial conditions for the ABV-Manipulator system comprising of the position and velocity of each state. The state vector used in simulation is

\[
X = \begin{bmatrix}
x \\
y \\
\phi \\
q_1 \\
q_2 
\end{bmatrix}
\]  

(3.6)

where \(x\) and \(y\) are the translational position of the base air-bearing vehicle, \(\phi\) is the yaw angle of the vehicle relative to the inertial frame and \(q_1\) and \(q_2\) are the angles of each joint of the manipulator.

For all test cases, the initial position and velocity of all states were zero. This represents the vehicle starting at the origin of the inertial frame with its attitude aligned with the inertial frame.

3.4.1.2 Desired Path

This block outputs the desired position and velocity for each state. For all test cases, the desired value took a step-input form as shown in the results figures. The desired values are arbitrary but were chosen as relatively small values to showcase the precision of the controllers tested.

3.4.1.3 Calculate Error

This block subtracts the actual state from the desired state producing the error signal.
3.4.1.4 Controller

This block calculates the control inputs for each degree of freedom using the PID or adaptive controller proposed by [19][20].

The algorithm for the adaptive controller is shown in Figure 3.3. The adaptive control law directly computes the input forces and torques for each degree of freedom of the base vehicle and manipulator. The full equations are outlined in Section 2.3. The algorithm shown below is computed for axis of the ABV-Manipulator system corresponding to 5 element control input vector.

```
Algorithm 1: Determine control torques.
Input: Φ, Φ_d
Output: Control torque τ
while t ≤ t_sim do
  if Σi ≠ 0 and |e| > e_ε then
    τ_i := -f_i(e(t)) - M_{max}sgn(Σi)
  else
    if Σi = 0 and |e| > e_ε then
      τ_i := -f_i(e(t)) - M(t)
    else
      τ_i := -f_i(e(t)) - λ_i e_i - M_{max}sat(Σ_i)
    end
  end
τ ← \hat{H} \tau^*
end
```

Figure 3.3: Adaptive Controller Algorithm (Source [20])

The first step in the adaptive controller is to calculate σ_i given by Equation 2.54. Depending on the value of σ_i and the absolute value of the error, one of three different control laws is utilized at that time-step. Each of the three control laws calculates the virtual input, τ^* which is transformed back into the actual input, τ through Equation 2.43. For the base vehicle, the control inputs are on/off pulses corresponding to firing of certain thrusters. The output of this block is then a thrust direction vector and the manipulator joint torques. The thrust direction vector is a three element vector.
in which a 1 represents a positive thrust, -1 is a negative thrust and a 0 is no thrust. The three elements represent the thrust direction required in each of the three degrees of freedom. A software Schmitt trigger is used to produce the series of on/off pulses from the continuous signal output by the adaptive and PID control laws. When the continuous signal output from the controller is greater than \( U_{on} \) or less than \(-U_{on}\), the output of the Schmitt Trigger is the maximum control input. Once the continuous signal is greater than \( U_{on} \) and while it is greater than \( U_{off} \), the output of the Schmitt Trigger is the maximum control input. The same is true in reverse for a negative control input. In this application, the Schmitt Trigger is used to determine whether the value for the thrust direction vector in each direction is a 1, 0 or -1. The diagram for a Schmitt Trigger is shown in Figure 3.4 For both the PID and adaptive controller, the Schmitt Trigger parameters were set to \( U_{on} = 0.15 \) and \( U_{off} = 0.05 \).

Separate PID controllers are implemented for each position state of the ABV-Manipulator. This means there is one PID controller with separate gains for the x-direction, another separate PID controller with separate gains for the y-direction and so on. The control input from the each PID controller is calculated using a discrete form such that

\[
P = K_p e_1(t)
\]
\[ I = I + K_i e_1(t) \]  
\[ D = K_d \frac{e_1(t) - e_1(t - T)}{T} \]  

And finally,

\[ u_i = P + I + D \]

where \( e_1(t) \) is the error in position of the respective state at the current time step and \( T \) is the sampling time. Thus, the \( I \) and \( D \) terms are calculated using approximations of the integral and derivative.

3.4.1.5 Thrust Command

This block takes in the thrust direction vector and determines the combination of thrusters that need to fire in order to impart the correct thrust direction on the ABV. On the actual ABV, the output of this function is an eight element thrust command vector containing a series of 1’s or 0’s corresponding to which thrusters need to be opened at that time instant. For the simulation, the same framework is used except the output of the thrust command block is a three element thrust command vector consisting of the thrust value produced by the thrusters. For example, if the thrust direction vector was \((1 -1 1)\) the thrust command vector would be \((0.25 -0.25 0.0306)\).

3.4.1.6 Dynamic Model Update

This block recalculates the dynamic model based on the position and orientation of the manipulator. It directly Equations 2.18 - 2.40 outlined in the theory section. In short, the process is as follows:

1. Evaluation of \( T^J_{J_i} \) and \( T^J_{E_i} \)
2. Evaluation of \( H_0, H_{0m}, H_m, C \)
3. Evaluation of $\dot{H}_0$, $\dot{H}_0m$, $\dot{H}_m$

First, the transformation matrix for each joint coordinate frame and center of mass coordinate frame relative to the inertial frame are calculated. From these, the respective rotation matrices and position vectors can be extracted. Then, the submatrices of the equations of motion are calculated. Since the air-bearing vehicle being utilized for the dynamic model can only move in three degrees of freedom, the base vehicle submatrices, $H_0$ and $\dot{H}_0$ become a $3 \times 3$ matrix such that

$$H_0 = \begin{bmatrix}
    m_{tot} & 0 & -m_{tot}r_{0,cy} \\
    0 & m_{tot} & m_{tot}r_{0,cx} \\
    m_{tot}r_{0,cx} & m_{tot}r_{0,cy} & I_{zz}
\end{bmatrix}$$  \hspace{1cm} (3.11)

where $I_{zz}$ is the total moment of inertia about the vehicle vertical axis taken from the inertia submatrix $H_s$. Also, the time derivative of the base vehicle submatrix, $\dot{H}_0$ becomes

$$\dot{H}_0 = \begin{bmatrix}
    0 & 0 & -m_{tot}\dot{r}_{0,cy} \\
    0 & 0 & m_{tot}\dot{r}_{0,cx} \\
    m_{tot}\dot{r}_{0,cx} & m_{tot}\dot{r}_{0,cy} & \dot{I}_{zz}
\end{bmatrix}$$  \hspace{1cm} (3.12)

Furthermore, the coupling submatrix, $H_{0m}$ uses Equation 2.22 directly and is of size $3 \times 2$. The inertia matrix of the manipulator, $H_m$ is of size $2 \times 2$ since the manipulator used has two planar links and is calculated using Equation 2.25. The output of this block are the matrices of the dynamic model, namely $H$, $\dot{H}$ and $C$ from Equation 2.19.

3.4.1.7 Dynamics Propagation

This block solves for the acceleration vector by isolating it using Equation 2.19. The output of this block is the acceleration of each state. Explicitly, this block solves for
the acceleration state vector by rearranging Equation 2.19 such that

\[ \ddot{\Phi} = H^{-1}(\tau - (\dot{H}\dot{\Phi} + C)) \]

(3.13)

The last part of the model, on the far right, performs the double integration of the acceleration for each term. The acceleration is integrated once, utilizing the initial conditions to obtain the velocity of each state. The velocity is then integrated utilizing the initial conditions to obtain the position of each state. Then, the position and velocity of each state are fed back into the error block and the loop continues for the commanded simulation time.

### 3.4.2 Dynamic Model Validation

In order to validate the non-linear, time-varying dynamic model, a joint velocity controller was implemented for each of the manipulator joints. The goal of the dynamic model validation is to apply accelerate each of the joints to a certain velocity and analyze the effects on the base-vehicle position and orientation. It is expected that the acceleration of the manipulator joints will induce an angular acceleration of the base-vehicle with an opposite sign and magnitude according to the ratio of the moments of inertia.

The manipulator joint velocity control is achieved using a Computed Torque Control law as outlined in [27] and [18]. Below is a brief explanation of the Computed Torque Control law implemented in the simulation to achieve joint velocity control. The control input is calculated using a PD form such that

\[ \dot{u} = K_D(\dot{q}_d - \dot{q}) + K_P(q_d - q) \]

(3.14)
where $K_D$ and $K_p$ are the controller gains, $q$ and $\dot{q}$ are the joint angle position and velocity and $q_d$ and $\dot{q}_d$ are the desired joint angle position and velocity. Once the control input is determined, the input torque is computed using the dynamics of the manipulator as shown in Equation 3.15.

$$\bar{\tau} = \bar{H}\ddot{u} + \bar{C}(\dot{q}, q) \quad (3.15)$$

Figure 3.5 shows the definition of the base-vehicle yaw angle and the manipulator joint angles.

Figure 3.5: Primary Angle Definitions for ABV-Manipulator System (Source [29])

Following a similar verification method performed in [18], each joint of the manip-
ulator was commanded to accelerate to a velocity of -0.1 rad/s for 10 seconds, then accelerate in the opposite direction to 0.1 rad/s for 20 seconds and then back to -0.1 rad/s for 10 seconds. All of the initial conditions are set to zero such that the ABV-Manipulator system starts at the origin of the inertial frame with zero angle relative to the inertial frame. Similarly, the joint angles are zero and the manipulator is extended outward in the direction of the x-axis of the vehicle body frame. As seen in Figure 3.6, the resulting angular acceleration of the base vehicle about its vertical axis indeed shows the correct trend since it is opposite in sign and its magnitude is reduced in proportion to the ratio of moments of inertia.

![Angular Rates of Manipulator and Base Vehicle](image)

**Figure 3.6: Angular Rates of Manipulator Joints and Base Vehicle**

Furthermore, the dynamic model utilized is coupled and thus the translational position of the base vehicle is impacted by the acceleration of the manipulator joints. Figure 3.7 shows the deviation of the base-vehicle position along with the angular
position of the manipulator joints during the maneuver shown in Figure 3.6.

![Graph of Manipulator Joints and Base Vehicle](image)

Figure 3.7: Position of Manipulator Joints and Base Vehicle

In general, the trends shown for the signs of the angular position and translation position of the base vehicle for the time-varying dynamic model of the ABV-Manipulator system agree with the literature and are thus validated.

### 3.5 Controller Tuning

Both the PID and adaptive controllers implemented contain tunable gains in order to achieve the desired performance. Due to the precision required for on-orbit servicing and assembly or debris capture and removal, the requirements on the controller design are shown in Table 3.4. These design requirements are for the position of each state.
### Criteria Requirement

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overshoot</td>
<td>$\leq 10%$</td>
</tr>
<tr>
<td>Rise Time</td>
<td>$\leq 20\text{ s}$</td>
</tr>
<tr>
<td>Steady State Error</td>
<td>$\leq 0.1$</td>
</tr>
</tbody>
</table>

Table 3.4: Controller Design Requirements

### 3.5.1 PID Gains

The PID controller was tuned to achieve the design requirements above. In order to achieve the requirement on overshoot, the derivative gain was relatively high in order to produce a system which is nearly critically damped. The proportional gain was increased along with the derivative gain to achieve the required rise time. Lastly, the integral gains were set to small values to ensure the desired steady state error was achieved. Table 3.5 shows the gains of the PID controller that were used for all test cases.

<table>
<thead>
<tr>
<th>Gain</th>
<th>x</th>
<th>y</th>
<th>$\phi$</th>
<th>$q_1$</th>
<th>$q_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_p$</td>
<td>12.5</td>
<td>10</td>
<td>42</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$K_i$</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>$K_d$</td>
<td>45</td>
<td>35</td>
<td>55</td>
<td>1</td>
<td>1.25</td>
</tr>
</tbody>
</table>

Table 3.5: PID Controller Gains

### 3.5.2 Adaptive Controller Gains

The adaptive controller also has a set of gains which were tuned in order to achieve the performance requirements outlined in Table 3.4. The primary gain to be tuned is $M_{max}$ and it represents the maximum value of the control input. This is the gain used
in the first phase of the variable structure controller. In general, an increase of $M_{max}$ led to an increase in the overshoot and a decrease in $M_{max}$ led to non-zero steady-state error. The second phase of the variable structure controller calculates the adaptive gains based on the position and velocity errors and does not require tuning. The third phase of the variable structure controller uses a common sliding mode control law. The gains which required tuning in this control law are the boundary layer width, $\epsilon$, and a positive scalar, $\lambda$, called the sliding surface parameter. The boundary layer width was set to a constant value of 0.01 for all axes. The values for each variable needed for the adaptive controller are shown below.

\[
\begin{array}{cccccc}
\text{Gain} & x & y & \phi & q_1 & q_2 \\
M_{max} & 0.015 & 0.025 & 0.175 & 0.1 & 0.11 \\
\lambda & 3 & 3 & 1.5 & 1.25 & 1.25 \\
\end{array}
\]

Table 3.6: Adaptive Controller Gains
Chapter 4

Model Testing

4.1 Introduction

The goal of the testing performed using the simulation developed for this thesis was to compare the performance of the adaptive controller against a standard PID controller. In order to do this, a simulated object with some mass and moment of inertia was perfectly and instantaneously added to the dynamic model and the controllers were tested on their ability control the system to its desired value. The details of each test performed using the simulation developed are outlined below. For all cases tested using the simulation model, the initial conditions for the position and velocity of each state are set to zero. This corresponds to the ABV-Manipulator sitting at the origin of the inertial frame with zero angle relative to the inertial frame and the manipulator extended along the base vehicle’s x-axis as shown in Figure 3.5.
4.2 Testing - Object Manipulation

The primary mode of comparison between the adaptive and PID controller was achieved by instantaneously increasing the mass and moment of inertia of the end-effector of the manipulator at a point halfway through the simulation. This was done to simulate a maneuver which would require the ABV-Manipulator to approach and capture some object and bring it back to its initial conditions. For each state, the desired value utilized was a step input from zero initial condition to some linear or angular value for $t = 0$ to 30 seconds. Then, the mass and moment of inertia of the end-effector of the manipulator are instantaneously increased at $t = 30$ seconds. After $t = 30$ seconds, the desired value is the initial condition. The simulation did not take into account any contact dynamics nor any non-zero initial velocities of the simulated object which may cause position or attitude disturbances during capture.

4.2.1 Test 1 - Base Case

First, the PID and adaptive controllers were simulated for the same initial conditions and same desired values but with no object added. It is expected that the performance of the adaptive and PID controllers should be similar during this case in regards to typical control performance metrics, i.e. rise time, percent overshoot, settling time etc.

4.2.2 Test 2 - 10% Increase

The next simulation performed to compare the adaptive and PID controllers implemented an instantaneous increase in mass and moment of inertia with a mass that represents 10% of the total system mass equal to 1.26 kg and a 10% increase in the manipulator end-effector moment of inertia. Other details of the simulated object are not considered for this work.
It is expected that both controllers will be able to stabilize the system but the adaptive controller should perform better in regards to typical control performance metrics, i.e. rise time, percent overshoot, settling time etc.

4.2.3 Test 3 - 19% Increase

The next simulation performed aimed to find the largest mass increase for which the PID controller remained stable. It was found that a mass increase of 19% or 2.35 kg was the limit. Similarly, an increase to the end-effector moment of inertia was set at 19%.

4.2.4 Test 4 - 25% Increase

The final test performed aimed to showcase the adaptive controllers performance when the PID could no longer control the system in the time allotted. The mass increase of 25% represents a 3.15 kg increase. A 25% increase in the end-effector moment of inertia was also added.
Chapter 5

Results

5.1 Introduction

The results for the PID and adaptive controller obtained from the four tests performed as outlined in the previous chapter are shown below. The first 30 seconds of each test shown below are the same for all cases and for all states. There is no change in the value of the states, the adaptive gains or the control inputs. The first 30 seconds for each test are shown in order to provide a comparison of the performance of the controller before and after the mass and moment of inertia change.

5.2 Test 1 - Base Case

The values for each state of the ABV-Manipulator system are shown in Figure 5.1 for the case of no mass or inertia addition.
As expected, the adaptive and PID controllers achieve similar performance with some small deviations relative to each other. Both controllers meet the design requirements listed in the previous chapter. The largest overshoot is seen in the yaw angle as controlled by the PID. Table 5.5 gives the overshoot value for each state and for each case tested. One explanation for this may be the amount of available torque provided by the thrusters on the vehicle. The thrusters can only output one level of thrust and the moment arm is fixed. Thus, there is a limit on the amount of torque available.
Since the manipulator joints are rotating at the same time as the base vehicle, the net torque may be increased or decreased from the nominal value.

The control inputs calculated by the adaptive controller for the forces and torques acting on the ABV as well as the torques for each of the manipulator joints is shown in Figure 5.2.

![Control Inputs (Adaptive)](image)

**Figure 5.2: Control Inputs from Adaptive Controller for 0% Mass/Inertia Added**

Because the only actuators on the actual air-bearing vehicles are thrusters, the forces in the x and y direction and the torque about the z-axis exhibit a pulsed nature since the thrust of the thrusters cannot be varied. The manipulator torques computed by the controllers are continuous but torque-limited in order to reduce excessive accelerations. Furthermore, it should be noted that the abrupt changing of the thrust input, for $F_x$ at approximately 13-15 seconds is due to the sliding mode control law utilized in Phase III. The adaptive gains for the base case are shown in Figure 5.3.
The adaptive gains, while being the primary term in the phase II control law, are not the entire control law. There is another term which represents a dynamics negation. Thus, the adaptive gains do not directly equal the control inputs but are certainly the primary contributing term. The control inputs calculated by the PID controller for the forces and torques acting on the ABV as well as the torques for each of the manipulator joints is shown in Figure 5.4.
Due to the requirement of little to no overshoot, the PID controller generally fires the thrusters for longer periods of time compared to the adaptive controller. This is evident when comparing the figure above for the PID control inputs with the adaptive control inputs. The torque applied to each of the manipulator joints show similar trends.

**5.3 Test 2 - 10% Increase**

The values for each state of the ABV-Manipulator system are shown in Figure 5.5 for the case of 10% system mass and 10% end-effector moment of inertia increase.
As seen above, the x and y axis remain under control by both the PID and adaptive controller. This is expected behavior since the mass and moment of inertia increase of 10% is not extremely significant in the dynamics of translational motion of the air-bearing vehicle. However, this increase in mass and moment of inertia increase the torque produced when rotating the manipulator joints. Thus, the dynamics of the air-bearing vehicle about the vertical axis have changed enough that the PID controller starts to introduce some overshoot and reduction in rise time. The overshoot for the
yaw angle controlled by the PID has increased to 8.39% at \( t = 40.5 \) seconds. There is also 2.42% overshoot at \( t = 36.55 \) seconds in the yaw angle as controlled by the adaptive controller. Furthermore, the PID controller introduces some oscillations in the control of each joint of the manipulator as a result of this mass and moment of inertia increase. The control inputs calculated by the adaptive controller for the forces and torques acting on the ABV as well as the torques for each of the manipulator joints is shown in Figure 5.6.

![Control Inputs (Adaptive)](image)

**Figure 5.6: Control Inputs from Adaptive Controller for 10% Mass/Inertia Added**

The adaptive gains for this case are shown in Figure 5.7.
Figure 5.7: Adaptive Gains for 10% Mass/Inertia Added

The adaptive gains show very similar trends to the gains calculated in the base case. This is expected as the evolution of the states is very similar. The control inputs calculated by the PID controller for the forces and torques acting on the ABV as well as the torques for each of the manipulator joints is shown in Figure 5.8.
Figure 5.8: Control Inputs from PID Controller for 10% Mass/Inertia Added

Clearly, the PID controller is working really hard to stabilize the system from 55 - 75 seconds. It is clear to see that the PID controller is reaching its limit in capabilities for controlling the yaw angle when mass and moment of inertia are added.

5.4 Test 3 - 19% Increase

The values for each state of the ABV-Manipulator system are shown in Figure 5.9 for the case of 19% system mass and 19% end-effector moment of inertia increase.
Figure 5.9: PID vs Adaptive Controller for 19% Mass/Inertia Added

It was found that adding a mass equal to 19% of the total system mass and increasing the moment of inertia of the end-effector link by 19% was the largest increase that the PID could converge the yaw angle to the setpoint. The number of oscillations around the setpoint has increased from the previous case and the settling time has significantly increased from 15.2 seconds for the previous case to 37.4 seconds for this case. Additionally, the amplitude of the oscillations has increased leading to a maximum overshoot value of 17.35 degrees at t = 49.95 seconds. Furthermore, the
amplitude of the oscillations of the joint angles has increased relative to the previous case. The adaptive controller, however, does a good job of controlling each state in a similar performance to the previous case. Specifically, the overshoot value of the yaw angle has increased to 3.66 degrees at $t = 37.5$ seconds while the rise time has increased to 6.3 seconds. Despite this, the performance of the adaptive controller is still well within acceptable regions.

The control inputs calculated by the adaptive controller for the forces and torques acting on the ABV as well as the torques for each of the manipulator joints is shown in Figure 5.10.

![Figure 5.10: Control Inputs from Adaptive Controller for 19% Mass/Inertia Added](image)

The control inputs from the adaptive controller do not vary significantly for each case. Again, it is important to note that the high frequency switching of the thrusters and joint torques is due primarily to the sliding mode control law utilized in Phase III.
The portions on the adaptive gain plots where the value is zero are when either the first or third phase control laws are being applied.

The adaptive gains for this case are shown in Figure 5.11.

![Adaptive Gains](image)

Figure 5.11: Adaptive Gains for 19% Mass/Inertia Added

As expected, the adaptive gains show similar trends and only vary slightly with respect to the previous two cases. The control inputs calculated by the PID controller for the forces and torques acting on the ABV as well as the torques for each of the manipulator joints is shown in Figure 5.12.
5.5 Test 4 - 25% Increase

The values for each state of the ABV-Manipulator system are shown in Figure 5.13 for the case of 25% system mass and 25% end-effector moment of inertia increase.
In order to further compare the PID and adaptive controllers, an added mass equal to 25% of the total system mass and a 25% increase in the end-effector link moment of inertia were added. In the plot above, it is clear that the PID is unable to converge the yaw angle while the adaptive controller can do so with a 0.74 degree increase in overshoot, a 0.11 degree increase in steady state error and 0.3 second change in rise time.

The control inputs calculated by the adaptive controller for the forces and torques
acting on the ABV as well as the torques for each of the manipulator joints is shown in Figure 5.14.

![Control Inputs (Adaptive)](image)

Figure 5.14: Control Inputs from Adaptive Controller for 25% Mass/Inertia Added

The adaptive gains for this case are shown in Figure 5.15.
The adaptive gains do not change significantly for the different cases tested. This is expected behavior as the gains are computed based off of the position and velocity errors. So long as the system is evolving in a similar manner, the value of the adaptive gains will not change much.

The control inputs calculated by the PID controller for the forces and torques acting on the ABV as well as the torques for each of the manipulator joints is shown in Figure 5.16.
Again, the PID controller, due to its large proportional and derivative gains, tends to fire the thrusters for longer periods of time relative to the adaptive controller. This is likely the cause for inducing the oscillations seen in the yaw angle as the mass and moment of inertia are increased.

**5.6 All Cases**

It is clear from the figures above that the adaptive controller is performing better in converging the states to their desired value. It is also interesting to observe the difference in each of the states for each controller.
5.6.1 Adaptive Controller

When the system is controlled by the adaptive controller, it is interesting to note that the value in the y-direction does not vary much. One reason for this may be related to the coupling dynamics of the vehicle-manipulator system. Figure 3.7 shows the rotation of the manipulator effects the base-vehicle position. Clearly, the magnitude of the deviation is greater in the x-direction than in the y-direction. Similarly, the yaw
angle for the case of 25% added mass stays relatively constant from 30 seconds to 37 seconds. One reason for this may be that the net torque about the vertical axis is zero or close to zero as the torque imparted on the base-vehicle due to the manipulator motion is opposite in sign of the torque imparted by the base-vehicle thrusters. Overall, with one exception on the steady state error of the yaw angle, the adaptive controller meets the design requirements stated in Section 3.5 for all cases. The maximum overshoot value, found in the yaw angle for the case of 25% added mass and moment of inertia, is 4.40 degrees. The yaw angle also gives the largest steady state error for this case at 0.26 degrees. The rise time for each state meets the design requirement with the maximum value, again found in the yaw angle of 11.38 seconds.

Figure 5.18 shows the adaptive gains for each state for all cases of mass and moment of inertia increase. As mentioned before, the values of the adaptive gains are similar since they only depend on the value of the position and velocity errors. The adaptive gain for the yaw angle shows the most variation. In general, the adaptive gains for each state show a similar trend of converging towards zero as the position and velocity errors converge to zero.
To further compare the performance of the adaptive controller in all of the cases tested in this work, the values for the rise time, overshoot and steady state error are presented.

The rise time as typically defined in control systems analysis is the time taken for the state to increase from 10% to 90% of the desired value. The values for the rise time of each state for all cases is shown in Table 5.1. It is important to note that the rise time does not account for any initial overshoot in the signal. Table 5.1 shows that
each case meets the rise time design requirement of less than or equal to 20 seconds.

<table>
<thead>
<tr>
<th>State</th>
<th>x (m)</th>
<th>y (m)</th>
<th>φ (deg)</th>
<th>q1 (deg)</th>
<th>q2 (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0%</td>
<td>6.71</td>
<td>5.41</td>
<td>3.78</td>
<td>4.46</td>
<td>5.65</td>
</tr>
<tr>
<td>10%</td>
<td>6.89</td>
<td>5.47</td>
<td>2.53</td>
<td>4.88</td>
<td>4.92</td>
</tr>
<tr>
<td>19%</td>
<td>6.83</td>
<td>5.51</td>
<td>2.54</td>
<td>7.45</td>
<td>11.38</td>
</tr>
<tr>
<td>25%</td>
<td>6.72</td>
<td>5.52</td>
<td>2.24</td>
<td>5.91</td>
<td>7.35</td>
</tr>
</tbody>
</table>

Table 5.1: Rise Time for Each State for All Cases Tested (All units in seconds)

The overshoot is typically defined as the percent difference between the peak value and the desired value. This calculation typically has the desired value in the denominator. Because the desired value for all states after t = 30 seconds is 0, this calculation cannot be employed. Thus, the values for the overshoot presented in Table 5.2 are the absolute values of the peak values relative to zero. Note, these values represent the amount in meters or degrees that the state overshot the desired value.

<table>
<thead>
<tr>
<th>State</th>
<th>x (m)</th>
<th>y (m)</th>
<th>φ (deg)</th>
<th>q1 (deg)</th>
<th>q2 (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0%</td>
<td>0.003</td>
<td>0.001</td>
<td>0.99</td>
<td>0.004</td>
<td>0.002</td>
</tr>
<tr>
<td>10%</td>
<td>0.003</td>
<td>0.034</td>
<td>2.42</td>
<td>0.006</td>
<td>0.154</td>
</tr>
<tr>
<td>19%</td>
<td>0.003</td>
<td>0.055</td>
<td>3.66</td>
<td>0.001</td>
<td>0.391</td>
</tr>
<tr>
<td>25%</td>
<td>0.003</td>
<td>0.041</td>
<td>4.40</td>
<td>0.000</td>
<td>0.017</td>
</tr>
</tbody>
</table>

Table 5.2: Maximum Overshoot Values for Each State for All Cases Tested

Table 5.2 shows that the maximum overshoot value is within the design requirement of 10% for all cases.

Finally, the steady state error for each state and for all cases tested is shown in Table 5.3. The steady state error was calculated as the difference between the final
value and the desired value. Since the desired value for all states after $t = 30$ seconds is zero, the steady state error is simply the value of the state at $t = 75$ seconds.

<table>
<thead>
<tr>
<th>State</th>
<th>$x$ (m)</th>
<th>$y$ (m)</th>
<th>$\phi$ (deg)</th>
<th>$q_1$ (deg)</th>
<th>$q_2$ (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0%</td>
<td>0.003</td>
<td>0.0001</td>
<td>-0.09</td>
<td>0.0007</td>
<td>-0.0001</td>
</tr>
<tr>
<td>10%</td>
<td>0.0009</td>
<td>0.0001</td>
<td>-0.093</td>
<td>0.0006</td>
<td>0.0007</td>
</tr>
<tr>
<td>19%</td>
<td>0.0006</td>
<td>0.001</td>
<td>0.15</td>
<td>-0.0007</td>
<td>-0.0002</td>
</tr>
<tr>
<td>25%</td>
<td>-0.001</td>
<td>0.001</td>
<td>0.263</td>
<td>0.001</td>
<td>-0.001</td>
</tr>
</tbody>
</table>

Table 5.3: Steady State Error for Each State for All Cases Tested

Overall, it can be stated that the Adaptive controller, with two exceptions on the steady state error of the yaw angle, successfully controlled the system within the design requirements for all cases of mass and moment of inertia increase.

5.6.2 PID Controller

Lastly, it is interesting to compare the states of the system under PID control. Figure 5.19 shows the states for all cases tested for mass and moment of inertia increase.
First, the values in the x-direction show some variation as the mass and moment of inertia are increased. Similar to the adaptive controlled states, the variation of the values in the y-direction is not very large, relative to the x-direction values. The yaw angle values controlled by the PID clearly start to oscillate due to the control inputs computed by the controller. The values for the manipulator joint angles do not vary much when compared across the different cases.

Overall, the PID controller does not meet the design requirements listed in Section 72.
This was expected as the PID was tuned for the base case and the gains were not changed. The PID controller meets the design requirements, with one exception on the steady state error, for the cases of 0% and 10% mass and moment of inertia increase with the maximum overshoot for all states being in the yaw angle at 8.39%. The maximum steady state error is again at the yaw angle with a value of 0.21 degrees. For the cases of 19% and 25% mass and moment of inertia increase, the PID controller does not meet the design requirements. Because the yaw angle has not converged for this case, the steady state error cannot be fully determined. However, the error of the yaw angle for the 25% case at \( t = 75 \) seconds is -15.613 degrees.

The rise times for all states for all cases tested for the PID controller are shown in Table 5.4. For the states controlled by the PID, the rise time tends to be skewed when oscillations are induced. The values below represent the time difference between when the value is at 10% of the desired value/steady state value and when the value reaches 90% of the desired/steady state value for the first time.

<table>
<thead>
<tr>
<th>State</th>
<th>x</th>
<th>y</th>
<th>( \phi )</th>
<th>( q_1 )</th>
<th>( q_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0%</td>
<td>5.25</td>
<td>5.21</td>
<td>1.53</td>
<td>1.75</td>
<td>2.08</td>
</tr>
<tr>
<td>10%</td>
<td>5.13</td>
<td>6.33</td>
<td>1.84</td>
<td>6.28</td>
<td>2.38</td>
</tr>
<tr>
<td>19%</td>
<td>4.39</td>
<td>11.68</td>
<td>2.29</td>
<td>2.38</td>
<td>2.36</td>
</tr>
<tr>
<td>25%</td>
<td>3.77</td>
<td>12.02</td>
<td>5.68</td>
<td>3.15</td>
<td>3.02</td>
</tr>
</tbody>
</table>

Table 5.4: Rise Time for Each State for All Cases Tested (All units in seconds)

For all cases, except in the y-direction, the PID controller meets the requirements for the rise time. The PID does not meet the rise time requirements for the case of 19% and 25% increase in the y-direction.

The values for the overshoot when controlled by the PID are shown in Table 5.5. These values are the absolute value of the peak values relative to the desired value.
Here, it is clearly shown that the overshoot increases for the yaw angle and the manipulator joint angles as the mass and moment of inertia are increased. The maximum overshoot values for these angles are seen at the case of 25% increase.

<table>
<thead>
<tr>
<th>State</th>
<th>x (m)</th>
<th>y (m)</th>
<th>φ (deg)</th>
<th>q₁ (deg)</th>
<th>q₂ (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0%</td>
<td>0.014</td>
<td>0.022</td>
<td>3.68</td>
<td>0.57</td>
<td>0.49</td>
</tr>
<tr>
<td>10%</td>
<td>0.011</td>
<td>0.032</td>
<td>8.39</td>
<td>2.16</td>
<td>1.89</td>
</tr>
<tr>
<td>19%</td>
<td>0.070</td>
<td>0.013</td>
<td>17.35</td>
<td>3.16</td>
<td>2.68</td>
</tr>
<tr>
<td>25%</td>
<td>0.082</td>
<td>0.014</td>
<td>24.60</td>
<td>3.26</td>
<td>2.80</td>
</tr>
</tbody>
</table>

Table 5.5: Maximum Overshoot Value for Each State for All Cases Tested

It cannot be stated that the PID meets the requirements for overshoot due to the large values in the yaw angle and manipulator joints.

Finally, the steady state error for each state and for all cases tested is shown in Table 5.6. The steady state error was calculated as the difference between the final value and the desired value. Since the desired value for all states after \( t = 30 \) seconds is zero, the steady state error is simply the value of the state at \( t = 75 \) seconds.

<table>
<thead>
<tr>
<th>State</th>
<th>x (m)</th>
<th>y (m)</th>
<th>φ (deg)</th>
<th>q₁ (deg)</th>
<th>q₂ (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0%</td>
<td>0.010</td>
<td>0.013</td>
<td>-0.005</td>
<td>0.001</td>
<td>-0.0005</td>
</tr>
<tr>
<td>10%</td>
<td>0.012</td>
<td>0.014</td>
<td>0.201</td>
<td>1.044</td>
<td>0.933</td>
</tr>
<tr>
<td>19%</td>
<td>0.011</td>
<td>-0.007</td>
<td>0.198</td>
<td>1.409</td>
<td>1.279</td>
</tr>
<tr>
<td>25%</td>
<td>0.006</td>
<td>-0.007</td>
<td>-15.613</td>
<td>-2.559</td>
<td>-2.446</td>
</tr>
</tbody>
</table>

Table 5.6: Steady State Error for Each State for All Cases Tested

Finally, it cannot be stated that the PID controller meets the requirements for steady state error due to the values in the yaw angle and manipulator joints being
greater than the required threshold of 0.1 units.

5.7 Discussion

Overall, from the results shown above for each case tested, it is clear that the adaptive controller is able to adequately control the system when the dynamic properties of the system change. Furthermore, there is a limit at which the PID controller can longer achieve control, while the adaptive controller is still capable of doing so. In general, the PID controller tends to fire the thrusters for longer periods of time compared to the adaptive controller and require more torque (and more power) to control the manipulator joints. Note that the gains presented in the previous chapter for the PID and adaptive controller were the gains used for all test cases. This is to emulate a real-world scenario of tuning each controller off-line and implementing them on a real system and having no means or desire to change the gains once implemented.
Chapter 6

Conclusion

The focus of this thesis was the development and testing of a Simulink model to simulate the use of an adaptive control law for coordinated motion control of an air-bearing vehicle-manipulator system. The simulation utilized a nonlinear, time-varying dynamic model which is recalculated each time step.

The dynamic model was validated by analyzing the response of the base-vehicle during the motion of the manipulator joints. The manipulator motion was achieved by implementing a joint velocity controller. It was found that the acceleration of each joint produced an angular acceleration of the base-vehicle with opposite sign and magnitude proportional to the ratio of the moments of inertia. The position deviation of the base-vehicle during this maneuver showed the same trends as found in literature. Thus, the non-linear, time-varying dynamic model was validated.

The adaptive controller was tested in various cases by adding an instantaneous mass and moment of inertia increase to the end-effector of the manipulator. This could represent the capture of a debris object during debris removal operations or the use of a tool during an on-orbit servicing or assembly operation.

In order to assess the performance of the adaptive controller, a standard PID con-
controller was also simulated for the exact same conditions. It was found that as the mass
and moment of inertia increased, the PID controller became unable to converge the
system to the desired value for the yaw angle and manipulator joints. The adaptive
controller was able to control the system with slight increases in rise time, overshoot
and steady state error. Specifically, the maximum overshoot of 4.47 degrees in the yaw
angle represents the largest magnitude of overshoot from the adaptive controller. A
maximum rise time of 11.38 seconds in the second manipulator joint represents the
largest magnitude of the rise time from the adaptive controller. Finally, a maximum
steady state error of 0.26 degrees for the yaw angle represents the largest magnitude
of steady state error from the adaptive controller. Therefore, the performance of the
adaptive controller has been assessed in simulation and contains merit in performing
better than a PID controller.
Bibliography


[21] Hanlei Wang and Yongchun Xie. “Passivity based adaptive Jacobian tracking for free-floating space manipulators without using spacecraft acceleration”. In: *Automatica* 45.6 (2009), pp. 1510–1517. ISSN: 0005-1098. DOI: https://doi.org/


