Modeling and Control of a 6-DOF Multi-rotorcraft: Air-Ball

by

Jerry Ja Wang

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

Master of Science
In
Aerospace Engineering

Melbourne, Florida
May 2018
We the undersigned committee hereby approve the attached thesis, “Modeling and Control of A 6-DOF Multi-rotorcraft: Air-Ball” by Jerry J. Wang.

Markus Wilde, Ph.D.
Thesis Advisor and Assistant Professor
Department of Mechanical and Aerospace Engineering

Tiauw Go, Ph.D.
Associate Professor
Department of Mechanical and Aerospace Engineering

James Brenner, Ph.D.
Assistant Professor
Department of Chemical Engineering

Hamid Hefazi, Ph.D.
Department Head
Department of Mechanical and Aerospace Engineering
Abstract

Modeling and Control of A 6-DOF Multi-rotorcraft: Air-Ball

Author: Jerry Ja Wang

Advisor: Markus Wilde, Ph.D.

The Air-Ball project seeks to develop an omnidirectional, unmanned aerial vehicle with six degrees of freedom. The Air-Ball vehicle uses six propellers powered by six independent electric motors for propulsion and control. Each motor is attached to a hollow shaft that can rotate ±360° actuated by electric servos. Four of these propulsion shafts are configured in an orientation resembling a quad-copter, where the shafts are 90° apart. The other two propulsion shafts extrude perpendicular from each side of the plane of the quad-copter configuration. Under this unique configuration, Air-Ball is capable of decoupling the motion in the translational and rotational degrees of freedom, which brings great potential in spacecraft controls research and in using Air-Ball as a mapping or search tool. The highly nonlinear and over-actuated system poses a challenge for system modeling and controller design. To cope with this challenge, a modeling software, SimMechanics, was used to develop a flight simulation based on multibody physics. In addition, a mathematical model of Air-Ball was developed for designing a sliding mode controller. The sliding mode controller was integrated with the SimMechanics model to simulate the controller performance. Overall, the combination of sliding mode controller and the SimMechanics plant model produced multiple translation and rotation maneuver simulations. Uncertainties in the mathematical model were identified from the SimMechanics model and addressed appropriately. The following thesis provides a literature search on other six degrees of freedom vehicles, the mathematical model development, the SimMechanics model development, controller choice and design, and flight simulation results.
# Table of Contents

Table of Contents ................................................................. iv

List of Figures ................................................................. vii

List of Tables ........................................................................... ix

Acknowledgement .................................................................... x

Chapter 1 Introduction .......................................................... 1

  1.1 Previous Work and Project Background ........................................ 1
  1.2 Project Objectives .................................................................. 3
  1.3 Thesis Structure .................................................................... 4

Chapter 2 Literature Review ....................................................... 5

  2.1 SPHERES Testbed and Astrobess ............................................. 5
  2.2 Omni-Directional Aerial Vehicle .............................................. 7
  2.3 Spherical VTOL Aerial Vehicles .............................................. 8

Chapter 3 Dynamics, Kinematics and Force Modeling .................. 10

  3.1 Reference Frames and Transformation ....................................... 10
  3.2 Rigid Body Translation Dynamics ........................................... 11
  3.3 Rigid Body Rotation Dynamics ............................................... 12
  3.4 Equations of Motion ............................................................. 13
  3.5 Translational and Rotational Kinematic Mapping ...................... 14
  3.6 Motor Force Model ............................................................... 15
  3.7 Propeller Frame to Body Frame Transformation ...................... 16
  3.8 Control Distribution ............................................................. 18
  3.9 Summary ............................................................................. 22
Appendix A Thrust Stand Data.................................................................90
Appendix B SMC in Simulink ..................................................................93
Appendix C SimMechanics Model Tree.......................................................94
Appendix D PID Servo Controller in Simulink .........................................95
List of Figures

Figure 1: Air-Ball assembly ..........................................................2
Figure 2: Propulsion-shaft ............................................................2
Figure 3: Center box .................................................................3
Figure 4: SPHERES ................................................................5
Figure 5: Astrobes [3] ...............................................................6
Figure 6: Omni-Directional Aerial Vehicle [5] ............................7
Figure 7: Japanese Flying Sphere ...............................................8
Figure 8: Thrust Vectoring Spherical VTOL Aerial Vehicle [7] .......9
Figure 9: Inertial fixed frame using NED notation [8] ..................10
Figure 10: Inertial fixed frame to Air-Ball Body frame ...............11
Figure 11: Turnigy Thrust Stand [15] ..........................................16
Figure 12: Air-Ball body frame (yellow) and propeller frames (red) .17
Figure 13: Propulsion shaft angle ..............................................18
Figure 14: Simulink and SimMechanics connection diagram ......25
Figure 15: Exporting CAD model into a physical modeling XML format [15] .................................26
Figure 16: Import physical modeling XML file into SimMechanics [15] ........................................27
Figure 17: Mass propertyless cube defined at the center of Air-Ball.28
Figure 18: SimMechanics visualization window .........................29
Figure 19: Z Velocity for Case 1 ................................................30
Figure 20: Roll rate for Case 2 ..................................................31
Figure 21: Air-Ball Motor Configuration ....................................42
Figure 22: Air-Ball side view .....................................................43
Figure 23: Position of Air-Ball (Hover) ......................................44
Figure 24: Motor 1 and Motor 3 RPM (Hover) ............................45
Figure 25: Reference signal vs. model response ........................46
Figure 26: X translation trajectory ..............................................47
Figure 27: Air-Ball position (X-Translation) ..............................48
Figure 28: Air-Ball Attitude (X-Translation) ..............................48
Figure 29: Air-Ball linear velocity (X-Translation) .......................49
Figure 30: Air-Ball rotation rate (X-Translation) .........................49
Figure 31: Motor 1 and Motor 3 RPM (X-Translation) ...............51
Figure 32: Motor 2 and Motor 3 RPM (X-Translation) ...............51
Figure 33: Motor 5 and Motor 6 RPM (X-Translation) ...............52
Figure 34: Reference signal vs. model response .........................53
Figure 35: Y translation trajectory ..............................................54
Figure 36: Air-Ball position (Y-Translation) ..............................55
Figure 37: Air-Ball Attitude (Y-Translation) ..............................55
Figure 38: Air-Ball linear velocity (Y-Translation) .......................56
Figure 39: Air-Ball rotation rate (Y-Translation) .............................................. 57
Figure 40: Motor 1 and Motor 3 RPM (Y-Translation) ........................................ 58
Figure 41: Motor 2 and Motor 4 RPM (Y-Translation) ........................................ 58
Figure 42: Motor 5 and Motor 6 RPM (Y-Translation) ........................................ 59
Figure 43: Reference signal vs. model response .................................................. 60
Figure 44: Z translation trajectory ................................................................... 60
Figure 45: Air-Ball position (Z-Translation) ..................................................... 61
Figure 46: Air-Ball Attitude (Z-Translation) ..................................................... 62
Figure 47: Air-Ball linear velocity (Z-Translation) ............................................. 62
Figure 48: Air-Ball rotation rate (Z-Translation) ............................................. 63
Figure 49: Motor 1 and Motor 3 RPM (Z-Translation) ........................................ 63
Figure 50: Motor 2 and Motor 4 RPM (Z-Translation) ........................................ 64
Figure 51: Motor 5 and Motor 6 RPM (Z-Translation) ........................................ 65
Figure 52: Reference signal vs. model response .................................................. 66
Figure 53: Air-Ball position (Pitch-Rotation) ...................................................... 67
Figure 54: Air-Ball Attitude (Pitch-Rotation) ..................................................... 67
Figure 55: Air-Ball linear velocity (Pitch-Rotation) ............................................. 68
Figure 56: Air-Ball Angular Velocity (Pitch-Rotation) ....................................... 69
Figure 57: Motor 1 and Motor 3 RPM (Pitch-Rotation) ....................................... 70
Figure 58: Motor 2 and Motor 4 RPM (Pitch-Rotation) ....................................... 70
Figure 59: Motor 5 and Motor 6 RPM (Pitch-Rotation) ....................................... 71
Figure 60: Reference signal vs. model response .................................................. 72
Figure 61: Air-Ball position (Roll-Rotation) ....................................................... 73
Figure 62: Air-Ball yaw angle (Roll-Rotation) .................................................... 73
Figure 63: Air-Ball Attitude (Roll-Rotation) ....................................................... 74
Figure 64: Air-Ball linear velocity (Roll-Rotation) ............................................. 74
Figure 65: Air-Ball angular velocity (Roll-Rotation) ........................................... 75
Figure 66: Motor 1 and Motor 3 RPM (Roll-Rotation) ........................................ 76
Figure 67: Motor 2 and Motor 4 RPM (Roll-Rotation) ........................................ 77
Figure 68: Motor 5 and Motor 6 RPM (Roll-Rotation) ........................................ 77
Figure 69: Air-Ball attitude (Yaw-Rotation) ........................................................ 78
Figure 70: Air-Ball position (Yaw-Rotation) ....................................................... 79
Figure 71: Air-Ball position (Translation-Rotation) ............................................. 80
Figure 72: Air-Ball attitude (Translation-Rotation) ............................................. 80
Figure 73: Air-Ball linear velocity (Translation-Rotation) .................................... 81
Figure 74: Air-Ball angular velocity (Translation-Rotation) .................................. 82
Figure 75: Motor 1 and Motor 3 RPM (Translation-Rotation) ............................. 82
Figure 76: Motor 2 and Motor 4 RPM (Translation-Rotation) ............................. 83
Figure 77: Motor 5 and Motor 6 RPM (Translation-Rotation) ............................. 84
List of Tables

Table 1: Propeller Frame Transformation Matrix ..............................................17
Table 2: CAD components to corresponding SimMechanics blocks ....................27
Acknowledgement

I would like to thank my advisors: Dr. Markus Wilde, Dr. Tiauw Go, and Dr. James Brenner for their patience, guidance, and support throughout my academic career at Florida Tech. This thesis would not have been possible without their expertise. I would like to give Dr. Wilde a special thank you for guiding this project since junior design, all the way through senior design, and turning it into a thesis project. This project has given me a great opportunity to grow as an engineer.

I would also like to thank my matrix supervisor Dr. Jose Nunez for taking a chance on a fresh engineering graduate student and for giving me the opportunity to work at NASA KSC. A special thank you to my NASA mentors: Mike DuPuis and Michael Johansen for their patience and their extensive knowledge on modeling and control. Of course, I would like to extend my gratitude to everyone in the Flight Technology Branch at NASA KSC.

Finally, I would like to thank my friends James (Jimmy) Byrnes, Andrew Czap, Juliette Bido, and Charles (Joe) Berry for the support and input throughout the duration of this thesis. I am proud to say that I was in the same class with them.
Chapter 1
Introduction

1.1 Previous Work and Project Background

Air-Ball was a senior design graduation project focused on developing an omnidirectional unmanned aerial vehicle (UAV) capable of flying with six degrees of freedom (6 DOF) at Florida Institute of Technology’s Orbital Robotic Interaction, On-orbit servicing, and Navigation (ORION) laboratory. The project seek to demonstrate an alternative method to reproduce satellite kinematics and dynamics in a controlled laboratory environment. For this, Air-Ball must be capable of decoupling its rotational and translational motion, and thus be able to translate in any direction while rotating about any body axis at the same time. A vehicle with such capabilities can be called an omnidirectional flight vehicle (OFV). Due to its unique capabilities, Air-Ball also has potential applications in fields such as search and rescue or civil infrastructure inspections.

Air-Ball is enclosed in an outer cage to absorb compression forces generated by collisions and act as a propeller enclosure. The outer cage consists of 3-D printed junctions connected by carbon fiber rods. Six propulsion-shafts connect the outer cage to the center cage that houses electronic components. Slip-rings are used to route the motor wires from the avionics box into the shafts, enabling unlimited rotation of the shafts Four of these propulsion-shafts are configured 90° apart from each other, resembling a quadcopter, while the other two propulsion-shafts extrude perpendicular from the plane of the quadcopter assembly, as shown in Figure 1. Each shaft is free to rotate ±360° through actuation by electric servos, as shown in Figure 2.

Air-Ball’s propulsion system consists of six electric multi-rotorcraft motors along with six propellers. All six propulsion-shafts are mounted to a box around the center of gravity of Air-Ball. As shown in Figure 3, this center box houses the power supply, electronic speed controllers (ESCs), the onboard computer, the internal measurements unit (IMU), servos to rotate each propulsion-shaft, and a Wi-Fi antenna for communication. Air-Ball
weights at 4.14 kg without any electronics and 4.98 kg with electronics. The propellers are 0.3302 m in diameter and the propeller motor combination produces up to 34.1 N of thrust.
1.2 Project Objectives

The three primary objectives for this thesis are as follows:

- To develop a firm understanding of the kinematics and dynamics of Air-Ball through classical mechanics. The kinematics of Air-Ball are formalized in the body and inertial frame where the position and velocity of Air-Ball are defined. The dynamics of Air-Ball are defined in the body frame and propeller frame where the sum of the external forces acting on the body is accounted.

- To develop a model based controller capable of controlling Air-Ball to hover, translate, rotate, and perform combined translation and rotation maneuvers. The
controller uses the equations of motion to estimate a nominal control to achieve stability.

- To assess controller performance by comparing the command input with the SimMechanics plant output.

1.3 Thesis Structure

The thesis is organized into six chapters as follows:

Chapter 2 provides an overview of the existing literature on the current state-of-the-art of 6 DOF flight vehicles and compares some similarities between existing vehicles and Air-Ball.

Chapter 3 describes the kinematics and dynamics modeling of Air-Ball and develops the equations of motion needed for the controller design.

Chapter 4 introduces the SimMechanics software used to develop the flight simulator and the integration of SimMechanics with Computer Aided Design (CAD) and MATLAB/Simulink.

Chapter 5 describes the design of the sliding mode controller (SMC) by using the EOM derived in Chapter 3 in the multi-input and multi-output form.

Chapter 6 presents the flight simulation result for stable hover, linear translation, angular rotation, and combination of translation and rotation on the same axis.

Chapter 7 presents a conclusion section and recommendations for future work.
Chapter 2
Literature Review

2.1 SPHERES Testbed and Astrobees

Simulating satellite dynamics in a ground-based environment presents complications due to the presence of gravity. In order to simulate in an actual microgravity environment, the National Aeronautics and Space Administration and the United States military used the small spacecraft testbeds known as the Synchronized Position Hold Engage and Reorient Experimental Satellites (SPHERES) as shown in Figure 4 [1].

![SPHERES](image)

Figure 4: SPHERES

The SPHERES facility on the International Space Station (ISS) consisted of three self-contained free-floating satellites actuated by twelve CO₂ cold gas thrusters [1]. SPHERES was used to mature the capability of control algorithms for future formation flights in micro-gravity with 6 DOF, demonstrate rendezvous and docking maneuvers, and improve the field of human and tele-robotic operation and control [2]. Although SPHERES
is among the most used ISS capability for experiments, the opportunity to run SPHERES activities is limited. SPHERES requires substantial maintenance from astronauts to setup, supervise, and replace consumables [3]. This led to the development of a next generation free-flying robot, Astrobees as shown in Figure 5.

![Figure 5: Astrobees [3]](image)

Astrobees’ primary purpose is to continue the capability of SPHERES by providing a more flexible testbed for research in a micro-gravity environment [3]. The upgrades in this new generation testbed are: no consumables other than rechargeable batteries which Astrobees can dock and recharge autonomously, further range, faster movement speed, better sensing hardware, faster computation time, and can perform most operations without astronaut supervision [3]. Astrobees also have new capability to support robotic manipulators for human-robot interaction.

Although SPHERES and Astrobees are both flexible platforms that will enable future micro-gravity research, with the upcoming retirement of the ISS in 2025 [4] requires alternatives to test micro-gravity on the ground. Inspired by both SPHERES and Astrobbees, Air-Ball seeks to serve as a spacecraft maneuver testbed on the ground using multiple variable thrusters to perform spacecraft maneuvers.
2.2 Omni-Directional Aerial Vehicle

Traditional rotorcraft such as quad-copters are commonly used for a wide variety of applications due to their agility and simplicity; however, these traditional rotor-craft are under-actuated that result in coupled translation and rotation [5]. If such constraint is broken, the rotorcraft can experience an increase in performance, flight duration, payload, and robustness [5].

The Omni-Directional Aerial Vehicle (ODAV) uses the configuration as shown in Figure 6 to break the limitation in traditional rotorcraft. This configuration requires eight actuators fixed to the vertices of a cube with reversible fixed-pitch motor-propeller. The reversible mechanism allows for both positive and negative thrust to be produced that evenly distribute the load of the ODAV. The control strategy for the ODAV was done in two separate control loops by assuming translation and rotation are decoupled. Position control was chosen for the ODAV’s translational dynamics and a cascaded control structure was used to track the attitude.

![Figure 6: Omni-Directional Aerial Vehicle [5]](image)
Although the experiment demonstrated the feasibility of using reversible motor-propeller actuators to independently generate thrust and torque in all directions, the proposed controller was not able to fully decouple the ODAV’s translational and rotational dynamics [5]. The force mapping of the ODAV and the control allocation strategy was not fully defined in literature, and the proposed thruster allocation strategy was not able to recover all of the available thrust and torque. In contrast, Air-Ball’s control allocation strategy is defined through the summation of thrust and moment between the propeller frame and body frame. The control strategy of Air-Ball uses one control loop that governs both the translational and the rotational dynamics and controls each dynamics separately.

2.3 Spherical VTOL Aerial Vehicles

With the rapid development of UAVs in recent years, the engineering community has experienced an increase in the utilization of small rotorcraft for scientific research. The combination of inexpensive hardware and ease of customization has led to some unique flight configurations such as spherical VTOL aerial vehicles. The initial effort can be traced back to the Japan Flying Sphere (JFS), shown in Figure 7, which the Japanese Defense Ministry funded in 2011 to study the dynamics of vertical take-off in fixed wing aircraft [6].

![Figure 7: Japanese Flying Sphere](image)
The JFS prototype uses a singular propeller and eight control vanes enclosed in a spherical frame [6]. The control vanes redirect the flow generated by the propeller to create thrust vectors for maneuvers. Based on similar principles, other iterations exist with a simplified setup such as a single propeller with a two-axis swash-plate and four control surfaces [7], as shown in Figure 8. The idea of thrust vector manipulation in Air-Ball was extracted from the context of these spherical VTOL vehicles.

Figure 8: Thrust Vectoring Spherical VTOL Aerial Vehicle [7]
Chapter 3  
Dynamics, Kinematics and Force Modeling

Mathematical modeling of Air-Ball is necessary to understand the kinematics and dynamics behavior. Furthermore, the resulting system of equations of motion is crucial for model based controller. Two reference frames are defined in the beginning that allows for defining the kinematics of Air-Ball. The Newton-Euler method is used to describe the dynamics of Air-Ball. The motor force and torque orientations are formulated in the body and propeller frame by direction cosine matrices. The model assumes Air-Ball to be a rigid body consisting of six rotors grouped into three counter-rotating fixed pitch propeller pairs. The aim is to provide the mathematical equations that describes the behavior of Air-Ball by similar methods presented in [9, 10, 11, 12].

3.1 Reference Frames and Transformation

Two reference systems are needed to derive Air-Ball’s rigid body dynamics: a fixed frame \( \{F\} \) shown in Figure 9 and a body frame \( \{B\} \). It is useful to define the fixed frame tangent to the Earth surface and to use the North, East, and Down (NED) coordinates. The body frame is attached to the center of gravity (CG) of Air-Ball shown in Figure 10.

![Figure 9: Inertial fixed frame using NED notation [8]](image-url)
3.2 Rigid Body Translation Dynamics

Start by defining the force vector in the body frame as $F^B$ and the velocity vector in the body frame as $V^B$ that represents the velocity of CG of Air-Ball with respect to the fixed frame. The translation dynamics can be expressed by Newton’s second law as

$$F^B = \left(\frac{d}{dt} (mV^B)\right)^F$$

(1)

where $(\cdot)^F$ is the derivative vector with respect to the inertial frame [9].

Equation 1 is currently in the fixed frame and must be expressed in the body frame. This can be done by defining an arbitrary vector $V$, such as a position or velocity vector, that can be changed in both the fixed frame and the body frame by an angular velocity $\Omega$ with respect to the fixed frame [10] such as
\[ \vec{V} = \frac{dV}{dt} = \vec{v}^B + \Omega^B \times \vec{v}^B \tag{2} \]

where \( \vec{v}^B \) is the time derivative of vector \( \vec{V} \) as seen by the body frame.

\[ \vec{F} = \frac{dV}{dt} = \vec{v}^B + \Omega^B \times \vec{v}^B \tag{2} \]

Equation 2 represents the Coriolis theorem that relates the derivative of a vector in the fixed frame with the derivative of the same vector in the rotating body frame [10].

Applying the Coriolis theorem to Equation 2 yields:

\[ \vec{F}^B = m \left( \frac{d}{dt} \vec{v}^B + \Omega^B \times \vec{v}^B \right) \tag{3} \]

The total force is the sum of all thrusts generated by all six electrical motors and the gravitational force acting on the CG expressed as

\[ \vec{F}^B = \sum (F_{motor})^B + \vec{G}^B \tag{4} \]

where vector \((F_{motor})^B\) is the thrust generated by an electric motor in the body frame and \( \vec{G}^B \) is the gravitational force in the body frame. Substituting Equation 4 into Equation 3 gives

\[ \frac{d}{dt} \vec{v}^B = \frac{1}{m} \left( \sum (F_{motor})^B + \vec{G}^B \right) - \Omega^B \times \vec{v}^B \tag{5} \]

### 3.3 Rigid Body Rotation Dynamics

The moment vector in the body frame can be expressed as \( \vec{M}^B \) and the inertial tensor matrix about the CG of Air-Ball with respect to the body frame is expressed as
\[ I = \begin{bmatrix} l_x & -l_{xy} & -l_{xz} \\ -l_{xy} & l_y & -l_{yz} \\ -l_{xz} & -l_{yz} & l_z \end{bmatrix} \]  

The rotational dynamics of Air-Ball can be derived by Euler’s Equation given as

\[ \mathbf{M}^B = I \dot{\mathbf{\Omega}}^B + \mathbf{\Omega}^B \times I \mathbf{\Omega}^B \]  

Take into account that the total torque acting on Air-Ball is the sum of all torques produced by all six motors expressed as

\[ \mathbf{M}^B = \sum T_{\text{motor}} \]  

Combine Equation 7 and Equation 8 and rearrange the equation yields

\[ \frac{d}{dt} \mathbf{\Omega}^B = I^{-1} (\sum T_{\text{motor}} - \mathbf{\Omega}^B \times I \mathbf{\Omega}^B) \]  

3.4 Equations of Motion

Grouping Equation 5 and Equation 9 into vector form yields

\[
\begin{bmatrix}
\frac{d}{dt} \mathbf{V}^B \\
\frac{d}{dt} \mathbf{\Omega}^B
\end{bmatrix} = \begin{bmatrix}
\frac{1}{m} (\sum (\mathbf{F}_{\text{motor}})^B + \mathbf{G}^B) - \mathbf{\Omega}^B \times \mathbf{V}^B \\
I^{-1} (\sum T_{\text{motor}} - \mathbf{\Omega}^B \times I \mathbf{\Omega}^B)
\end{bmatrix}
\]  

These are the equations of motion (EOM) describing the motion of a physical system with respect to time and given a set of initial conditions. The EOM are highly nonlinear and are not separable; therefore, an analytical solution is not possible. Instead, numerical methods can be employed to solve the problem.
3.5 Translational and Rotational Kinematic Mapping

The EOM describe how total thrust and total moments affects the translational and rotational velocity of Air-Ball in its body frame. In order to use the position and orientation information in the fixed frame for controller feedback, a kinematic mapping is required. The kinematic mapping is considered in two parts: translational and rotational. The two transformations are taken from [11] and [12] and then merged together into a compact matrix at the end.

The translational mapping is taken from [13] directly using the Euler angle rotation order yaw, $\psi$, on the z-axis, followed up pitch, $\theta$, on the y-axis, and finally roll, $\phi$, on the x-axis. The complete transformation matrix from body frame to fixed frame is given as

$$
R_B^F = \begin{bmatrix}
\cos\theta\cos\psi & \sin\psi\sin\theta\cos\psi - \cos\psi\sin\theta & \cos\theta\sin\psi\cos\psi + \sin\psi\sin\theta \\
\cos\theta\sin\psi & \sin\psi\sin\theta\sin\psi + \cos\psi\cos\theta & \cos\theta\sin\psi\sin\theta\sin\psi - \sin\psi\cos\theta \\
-\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta
\end{bmatrix} \tag{11}
$$

the inverse of $R_B^F$ is equivalent to the transpose of itself. This is a useful quality that can be used to transform from the fixed frame to the body frame.

The transformation matrix for the angular velocities from body frame to fixed frame is taken from [14] and expressed as

$$
W_B^F = \begin{bmatrix}
1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\
0 & \cos\phi & -\sin\phi \\
0 & \sin\phi\sec\theta & \cos\phi\sec\theta
\end{bmatrix} \tag{12}
$$

To merge Equation 11 and Equation 12, a vector for position and orientation of the CG of Air-Ball in the fixed frame must be defined as
\[ q^F = \begin{bmatrix} X \\ Y \\ Z \\ \phi \\ \theta \\ \psi \end{bmatrix} \]  

(13)

and equivalently a derivative of this vector for linear velocity and angular velocity in the fixed frame as

\[ \dot{q}^F = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \]  

(14)

The resulting compact matrix takes the form of

\[ \dot{q}^F = \begin{bmatrix} R^F_B & 0 \\ 0 & W^F_B \end{bmatrix} \begin{bmatrix} V^B \\ \Omega^B \end{bmatrix} \]  

(15)

where \( V^B = [u \ v \ w]^T \) is the velocity vector in the body frame and \( \Omega^B = [p \ q \ r]^T \) is the angular velocity vector in the body frame. Equation 15 is used to simulate the dynamics of Air-Ball with respect to the fixed frame.

### 3.6 Motor Force Model

Brushless motors are common for rotor-craft propulsion systems due to their fast response time and flexibility; however, motor dynamics are difficult to model in simulation. By assuming the motor dynamics response is significantly faster than the rest of the system response, each motor is treated as a black box with thrust curve data obtained from a commercial off-the-shelf thrust stand shown in Figure 11. The motor data is listed in Appendix A.
A duty cycle sweep between 0% to 100% was conducted, and the corresponding thrust value was collected and processed. The thrust stand was calibrated by a spring scale, and a correction factor was applied to all measured thrust to determine the true value. The rotation per minute (RPM) of the motor can be calculated by multiplying the duty cycle (%) by the maximum RPM. A motor constant $k$ was calculated by dividing the thrust by the motor RPM.

### 3.7 Propeller Frame to Body Frame Transformation

Since the thrust generated by each motor is in the propeller frame, $T^p$, transformation between the propeller frame to the body frame is necessary to determine the force distribution. By using geometric relations, transformation matrices are developed by using direction cosine matrices (DCM). Since the orientation of each propeller are different from the layout of Air-Ball, the DCM rotation orders differ. The rotation angles for x-y-z axes are denoted as $\alpha, \beta, \gamma$ respectively. Table 1 provides DCM rotation for each motor.
Figure 12: Air-Ball body frame (yellow) and propeller frames (red)

Table 1: Propeller Frame Transformation Matrix

<table>
<thead>
<tr>
<th>Motor Number</th>
<th>DCM Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motor 1</td>
<td>$R_{P_1}^B = R(\beta)R(\alpha)R(y)$</td>
</tr>
<tr>
<td>Motor 2</td>
<td>$R_{P_2}^B = R(\beta)R(\alpha)R(y)$</td>
</tr>
<tr>
<td>Motor 3</td>
<td>$R_{P_3}^B = R(\beta)R(\alpha)R(y)$</td>
</tr>
<tr>
<td>Motor 4</td>
<td>$R_{P_4}^B = R(\beta)R(\alpha)R(y)$</td>
</tr>
<tr>
<td>Motor 5</td>
<td>$R_{P_5}^B = R(\beta)R(y)R(\alpha)$</td>
</tr>
<tr>
<td>Motor 6</td>
<td>$R_{P_6}^B = R(\beta)R(y)R(\alpha)$</td>
</tr>
</tbody>
</table>
3.8 Control Distribution

![Figure 13: Propulsion shaft angle](image)

From the geometric relationship in Figure 12, the distance vectors between the propeller axes and the CG of Air-Ball is given as following

\[ d_1^B = [d \ 0 \ 0]^T \]
\[ d_2^B = [0 \ -d \ 0]^T \]
\[ d_3^B = [-d \ 0 \ 0]^T \]
\[ d_4^B = [0 \ d \ 0]^T \]
\[ d_5^B = [0 \ 0 \ d]^T \]
\[ d_6^B = [0 \ 0 \ -d]^T \]
The thrust vector by each motor is given as follows:

\[
T_1^P = k \omega_1 = k \begin{bmatrix} 0 \\ \omega_1 \cos \delta_1 \\ \omega_1 \sin \delta_1 \end{bmatrix}
\]

\[
T_2^P = k \omega_2 = k \begin{bmatrix} 0 \\ \omega_2 \cos \delta_2 \\ \omega_2 \sin \delta_2 \end{bmatrix}
\]

\[
T_3^P = k \omega_3 = k \begin{bmatrix} 0 \\ \omega_3 \cos \delta_3 \\ \omega_3 \sin \delta_3 \end{bmatrix}
\]

\[
T_4^P = k \omega_4 = k \begin{bmatrix} 0 \\ \omega_4 \cos \delta_4 \\ \omega_4 \sin \delta_4 \end{bmatrix}
\]

\[
T_5^P = k \omega_5 = k \begin{bmatrix} 0 \\ \omega_5 \cos \delta_5 \\ \omega_5 \sin \delta_5 \end{bmatrix}
\]

\[
T_6^P = k \omega_6 = k \begin{bmatrix} 0 \\ \omega_6 \cos \delta_6 \\ \omega_6 \sin \delta_6 \end{bmatrix}
\]

where \( \delta \) angles are defined as the propulsion-shaft rotation angle shown in Figure 13.

Considering the distance vectors and the transformations from propeller frame to body frame, the total thrust and torque expressed in Equation 4 and Equation 8 respectively can be written in the propeller frame as

\[
\sum F^B = R_{P1}^B T_1^P + R_{P2}^B T_2^P + R_{P3}^B T_3^P + R_{P4}^B T_4^P + R_{P5}^B T_5^P + R_{P6}^B T_6^P
\]  

(16)

\[
\sum T^B = d_1^B \times (R_{P1}^B T_1^P) + d_2^B \times (R_{P2}^B T_2^P) + d_3^B \times (R_{P3}^B T_3^P) + d_4^B \times (R_{P4}^B T_4^P) + d_5^B \times (R_{P5}^B T_5^P) + d_6^B \times (R_{P6}^B T_6^P)
\]  

(17)
By combining Equation 16 and Equation 17 into a compact matrix yields the form

\[
\begin{bmatrix}
F_x^B \\
F_y \\
F_z \\
T_x \\
T_y \\
T_z
\end{bmatrix} =
\begin{bmatrix}
D_{11} & D_{12} & D_{13} & D_{14} & D_{15} & D_{16} & k\omega_1 \\
D_{21} & D_{22} & D_{23} & D_{24} & D_{25} & D_{26} & k\omega_2 \\
D_{31} & D_{32} & D_{33} & D_{34} & D_{35} & D_{36} & k\omega_3 \\
D_{41} & D_{42} & D_{43} & D_{44} & D_{45} & D_{46} & k\omega_4 \\
D_{51} & D_{52} & D_{53} & D_{54} & D_{55} & D_{56} & k\omega_5 \\
D_{61} & D_{62} & D_{63} & D_{64} & D_{65} & D_{66} & k\omega_6
\end{bmatrix}
\]

(18)

where the elements of matrix \(D\) are given as

\[
D_{11} = -k(\sin(\gamma) \sin(\delta_1) + \sin(\beta) \cos(\delta_1))
\]

\[
D_{12} = k(\sin(\gamma) \sin(\delta_2) + \sin(\beta) \cos(\delta_2) + \sin(\alpha) \sin(\delta_2))
\]

\[
D_{13} = -k(\sin(\gamma) \sin(\delta_3) + \sin(\beta) \cos(\delta_3))
\]

\[
D_{14} = k(\sin(\gamma) \sin(\delta_4) + \sin(\beta) \cos(\delta_4) + \sin(\alpha) \sin(\delta_4))
\]

\[
D_{15} = k(\sin(\alpha) \cos(\delta_5) + \cos(\beta) \cos(\delta_5) + \sin(\gamma) \sin(\delta_5))
\]

\[
D_{16} = k(\sin(\alpha) \cos(\delta_6) + \cos(\beta) \cos(\delta_6) + \sin(\gamma) \sin(\delta_6))
\]

\[
D_{21} = -k(\sin(\alpha) \cos(\delta_1) + \cos(\gamma) \sin(\delta_1) + \sin(\beta) \sin(\delta_1))
\]

\[
D_{22} = k(\sin(\gamma) \sin(\delta_2) + \sin(\alpha) \cos(\delta_2))
\]

\[
D_{23} = -k(\sin(\alpha) \cos(\delta_3) + \cos(\gamma) \sin(\delta_3) + \sin(\beta) \sin(\delta_3))
\]

\[
D_{24} = k(\sin(\gamma) \sin(\delta_4) + \sin(\alpha) \cos(\delta_4))
\]

\[
D_{25} = -k(\cos(\alpha) \sin(\delta_5) + \sin(\gamma) \cos(\delta_5) + \sin(\beta) \sin(\delta_5))
\]

\[
D_{26} = -k(\cos(\alpha) \sin(\delta_6) + \sin(\gamma) \cos(\delta_6) + \sin(\beta) \sin(\delta_6))
\]

\[
D_{31} = -k(\cos(\beta) \cos(\delta_1) + \sin(\alpha) \sin(\delta_1))
\]
\[ D_{32} = -k(\cos(\alpha) \cos(\delta_2)) \]
\[ D_{33} = -k(\cos(\beta) \cos(\delta_3) + \sin(\alpha) \sin(\delta_3)) \]
\[ D_{34} = -k(\cos(\alpha) \cos(\delta_4)) \]
\[ D_{35} = -k(\sin(\beta) \cos(\delta_5) + \sin(\alpha) \sin(\delta_5)) \]
\[ D_{36} = -k(\sin(\beta) \cos(\delta_6) + \sin(\alpha) \sin(\delta_6)) \]
\[ D_{41} = dk(\sin(\beta) \sin(\delta_1) + \sin(\gamma) \cos(\delta_1)) \]
\[ D_{42} = dk(\sin(\beta) \sin(\delta_2) + \cos(\beta) \cos(\gamma) \cos(\delta_2)) \]
\[ D_{43} = -dk(\sin(\beta) \sin(\delta_3) + \sin(\gamma) \cos(\delta_3)) \]
\[ D_{44} = -dk(\sin(\beta) \sin(\delta_4) + \cos(\beta) \cos(\gamma) \cos(\delta_4)) \]
\[ D_{45} = -dk(\cos(\beta) \sin(\delta_5) + \sin(\gamma) \cos(\delta_5)) \]
\[ D_{46} = dk(\cos(\beta) \sin(\delta_6) + \sin(\gamma) \cos(\delta_6)) \]
\[ D_{51} = dk(\sin(\alpha) \sin(\delta_1) + \cos(\gamma) \cos(\delta_1)) \]
\[ D_{52} = dk(\sin(\alpha) \sin(\delta_2) + \sin(\gamma) \cos(\delta_2)) \]
\[ D_{53} = -dk(\sin(\alpha) \sin(\delta_3) + \cos(\gamma) \cos(\delta_3)) \]
\[ D_{54} = dk(\sin(\alpha) \sin(\delta_4) + \sin(\gamma) \cos(\delta_4)) \]
\[ D_{55} = -dk(\sin(\gamma) \sin(\delta_5) + \cos(\alpha) \cos(\delta_5)) \]
\[ D_{56} = dk(\sin(\gamma) \sin(\delta_6) + \cos(\alpha) \cos(\delta_6)) \]
\[ D_{61} = dk(\cos(\beta) \sin(\delta_1)) \]
\[ D_{62} = dk(\sin(\beta) \cos(\delta_2)) \]
\[ D_{63} = dk(\cos(\beta) \sin(\delta_3)) \]
\[ D_{64} = -dk(\sin(\beta) \cos(\delta_4)) \]
\[ D_{65} = dk(\sin(\beta) \sin(\delta_5) + \sin(\alpha) \cos(\delta_5)) \]
\[ D_{66} = dk(\sin(\beta) \sin(\delta_6) + \sin(\alpha) \cos(\delta_6)) \]

Taking the inverse of Equation 18 yields

\[
\begin{bmatrix}
\omega_1 \\
\omega_2 \\
\omega_3 \\
\omega_4 \\
\omega_5 \\
\omega_6
\end{bmatrix}^o = \frac{1}{k} \begin{bmatrix}
D_{11} & D_{12} & D_{13} & D_{14} & D_{15} & D_{16} \\
D_{21} & D_{22} & D_{23} & D_{24} & D_{25} & D_{26} \\
D_{31} & D_{32} & D_{33} & D_{34} & D_{35} & D_{36} \\
D_{41} & D_{42} & D_{43} & D_{44} & D_{45} & D_{46} \\
D_{51} & D_{52} & D_{53} & D_{54} & D_{55} & D_{56} \\
D_{61} & D_{62} & D_{63} & D_{64} & D_{65} & D_{66}
\end{bmatrix}^{-1}
\begin{bmatrix}
F_x^B \\
F_y^B \\
F_z^B \\
T_x^B \\
T_y^B \\
T_z^B
\end{bmatrix}
\]

Equation 19 can be used to convert the required external forces in the body frame to requested RPMs in the propeller frame. The matrix \( D \) varies with time and is dependent on the motor constant \( k \), vehicle geometry, and vehicle orientation.

### 3.9 Summary

The mathematical model of Air-Ball was derived from first principles by splitting the system into linear translation and angular motion components and then taking the summation of linear forces and angular forces, respectively. The control action distribution matrix was also derived based on Air-Ball’s motor configuration in different orientations and the transformation matrix between the propeller frame and the body frame. In summary, the set of equations of motion is long and difficult to express on paper, so instead a simplified form is given here instead for illustration.

The assumptions are as follows:
• Stable hover using motor 1, motor 2, motor 3, motor 4. \( \delta_1 = \delta_2 = \delta_3 = \delta_4 = 0 \)

• No propeller force in y direction. \( \delta_5 = \delta_6 = 0 \). Thrust vectors five and six aligned with the x direction.

• Air-Ball is fully symmetric, \( l_x = l_y = l_z \neq 0 \), \( l_{xy} = l_{yx} = l_{zx} = 0 \)

• Small angle assumptions in Air-Ball orientation

\[
D_{\text{hover}} = \begin{bmatrix}
0 & \sin(\delta_2) & 0 & -\sin(\delta_4) & \cos(\delta_5) & \cos(\delta_6) \\
\sin(\delta_1) & 0 & -\sin(\delta_3) & 0 & -\sin(\delta_5) & \sin(\delta_6) \\
-\cos(\delta_1) & -\cos(\delta_2) & -\cos(\delta_3) & -\cos(\delta_4) & 0 & 0 \\
0 & \cos(\delta_2) & 0 & -\cos(\delta_4) & -\sin(\delta_5) & -\sin(\delta_6) \\
\cos(\delta_1) & 0 & -\cos(\delta_3) & 0 & -\cos(\delta_5) & \cos(\delta_6) \\
\sin(\delta_1) & \sin(\delta_2) & \sin(\delta_3) & \sin(\delta_4) & 0 & 0
\end{bmatrix}
\]

(20)

\[
\dot{u} = v r - w q + \frac{k}{m} (\omega_5 + \omega_6)
\]

(21)

\[
\dot{v} = w p - u r
\]

(22)

\[
\dot{w} = u q - v p + \frac{k}{m} (-\omega_1 - \omega_2 - \omega_3 - \omega_4) + g
\]

(23)

\[
\dot{p} = \frac{1}{l_x} [qr(l_y - l_z) + lr q (\omega_1 + \omega_3) + (lr q + dk) \omega_2 + (lr q - dk) \omega_4 - lr (\omega_6 + \omega_5)]
\]

(24)

\[
\dot{q} = \frac{1}{l_y} [(pr(l_x - l_z) + lr p (\omega_2 + \omega_4) + (lr r - dk) \omega_5 + (lr r + dk) \omega_6 \\
+ (lr p + dk) \omega_1 + (lr p - dk) \omega_3]
\]

(25)

\[
\dot{r} = \frac{1}{l_z} [pq(l_x - l_y) + lr q (\omega_5 + \omega_6) + lr \dot{\omega}_1 + lr \dot{\omega}_2 + lr \dot{\omega}_3 + lr \dot{\omega}_4]
\]

(26)
In the end, the complexity of the model is driven by the control action distribution matrix in Equation 18. A simplified version of the model was given above for illustration purpose. The controller design and flight simulator is based on the fully expressed mathematical model that uses the result Equation 19 to calculate the control input, which will be discussed in the later chapter.
Chapter 4
Flight Simulation

The flight simulator is constructed by combining the SMC in Simulink and the plant model from SimMechanics as shown in Figure 14. SimMechanics is a product under Simulink’s physical modeling family based on Simscape, using standard Newtonian dynamics of forces and torques to assist engineering design and simulation of rigid body mechanics and their motion [16]. With SimMechanics, a plant model can be produced to simulate mechanical systems with a suite of options. These options enable the user to specify mass properties, kinematic constraints, and coordinate systems of each component of the entire system [16]. Similar to Simulink models, the mechanical system in SimMechanics is represented by connecting block diagrams and can be interfaced with Simulink to design and test a controller’s performance.

![Figure 14: Simulink and SimMechanics connection diagram](image)

An ordinary Simulink model represents the mathematics of a system’s motion, such as the algebraic and the differential equations that describes the system’s future motion based on its current state [16]. However, as mentioned previously, in deriving such model through...
first principles, some model dynamics may be neglected due to mathematical complexity, unknown parameters, and underestimation of the system’s order. The presence of uncertainty as can be the cause of inaccurate controller performance or simulation solver failure.

Similarly, a SimMechanics model represents the physical structure of a system by the geometric and kinematic relationships of its components [16]. SimMechanics converts this structure representation to an internal equivalent mathematical model. The geometric and kinematic relationships can be self-described in the SimMechanics workspace or imported from a Computer Aided Design (CAD) assembly. The CAD importation method was used to construct Air-Ball’s SimMechanics model, and modifications were made in the SimMechanics workspace to define actuators interfacing with the SMC designed in Simulink.

4.1 Importing CAD Geometry into SimMechanics

CAD software allows the user to model a system geometrically through a collection of components, otherwise known as an assembly. The assembly information can be used to construct the SimMechanics block diagram. The transition from the CAD assembly to the SimMechanics diagram model uses an XML file in a physical modeling format that contains all of the material properties and inertia information [16], and the process is shown in Figure 15.

![Figure 15: Exporting CAD model into a physical modeling XML format [15]](image)

The XML file is portable and independent of SimMechanics. From the XML file the user can fully construct the SimMechanics diagram following the process shown in Figure 16.
4.2 Modeling Using Sim-Mechanics

To fully utilize the XML file feature in SimMechanics, the CAD assembly must satisfy some constraints to produce a valid model. These constraints differ depending on the CAD assembly components and its corresponding SimMechanics counterpart. A table is given to relate the two nomenclatures below.

Table 2: CAD components to corresponding SimMechanics blocks

<table>
<thead>
<tr>
<th>CAD Assembly Component</th>
<th>Corresponding SimMechanics Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part</td>
<td>Body</td>
</tr>
<tr>
<td>Constraints</td>
<td>Joints</td>
</tr>
<tr>
<td>Fundamental Root</td>
<td>Ground/Root Weld/Root Body</td>
</tr>
<tr>
<td>Subassembly</td>
<td>Subsystem</td>
</tr>
<tr>
<td>Subassembly Root</td>
<td>Joint-Subsystem</td>
</tr>
<tr>
<td>Fixed Part</td>
<td>Root Body</td>
</tr>
</tbody>
</table>

All CAD assemblies require a single fundamental root, which is a fixed point that does not move. All positions and orientations of other subassemblies refer directly or indirectly to this fundamental root. In the case of Air-Ball, the fundamental root is also located where the vehicle’s coordinate system is defined, which is translated to root body in SimMechanics. Since Air-Ball’s center is an empty cage, a zero-mass, zero-inertial 1 mm x 1 mm x 1 mm cube was used as a reference as shown in Figure 17. A root body does not affect the model’s dynamics and is necessary as it represents a fixed anchor for part constraints. This root body can carry multiple coordinate systems for this purpose [16].
Subassemblies can be used in the CAD model to isolate collections of components and their corresponding constraints. The translation of subassemblies to subsystems in SimMechanics improves the model organization in the assembly hierarchy. For instance, each of Air-Ball’s propulsion shafts are organized into individual subsystem in the assembly hierarchy. This allows for a clear visualization when conducting simulations and debugging of the software.

The constraints in the CAD assembly restrict how each individual part moves with respect to its reference body. Without specifying constraints between bodies, the CAD model can move in six unrestricted DOF relative to its reference body. The DOF in each constraint must be reduced to accurately simulate the system.

In SimMechanics, these constraints are transitioned into joints. Some of these joints can become actuated such as the joints connecting the servo motor to the propulsion-shaft. A simple PID controller was implemented to control the servo motor position.

Mass properties and inertia tensors are required to be part of the CAD assembly. When the SimMechanics model is generated from the XML file, the mass properties information is transitioned to the corresponding body block in each subsystem. The CAD
platform in the case of Air-Ball calculated mass properties and inertial values from the mass
density and geometric volume with respect to the part’s center of gravity. The mass density
is specified in each components’ part file prior to assembly. The CAD software and the XML
translator computes the center of gravity of each individual part automatically based on the
geometric information.

The SimMechanics model tree for Air-Ball can be found in Appendix C.

4.3 Visualization and Animation through SimMechanics

One powerful aspect of SimMechanics is real-time the visualization and animation
window running simultaneously with the dynamics solver. The visual aid facilitates
debugging of the system during the building phase and provides a better understanding of
the system response as shown in Figure 18.

Figure 18: SimMechanics visualization window
4.4 Comparison of Motion Due to Analytical Equations of Motion with SimMechanics

Verification is needed to determine if the SimMechanics model matches with the EOM derived in Chapter 3. This verification is an important step because the EOM from classical mechanics is used to design the controller for stabilizing the SimMechanics model. Two cases were performed to ensure the accuracy. Case 1 looks at both models’ behavior under an applied linear force, in this case the force is the gravitational acceleration. Case 2 looks at behavior under an applied torque in the x-axis.

The EOM from Chapter 3 are solved in MATLAB script using the ordinary differential equation solver (ode113). The error tolerance of the MATLAB solver was set at $1e^{-3}$ for both relative tolerance and absolute tolerance. The SimMechanics solver tolerance is set to be default as $1e^{-8}$ time-step for both relative and absolute tolerance.

![Figure 19: Z Velocity for Case 1](image)
Figure 19 above shows the velocity in the z-axis from the SimMechanics model compared with the analytical EOM model, with no differences. According to Figure 20, the SimMechanics rotational behavior also compared well against the EOM; however, some deviation between the two models can be observed after the six seconds mark. This deviation is due to the difference in solver error tolerance setting.

Overall the two models compared well with each other, which suggest that the EOM from classical mechanics derivation is a good baseline for designing a model based controller for the SimMechanics model.
Chapter 5
Multi-Input-Multi-Output Controller Design

Often, in nonlinear control, it is difficult to derive a mathematical model that covers the actual system with total accuracy. From a control system point of view, modeling inaccuracies can be classified into structured (parametric) uncertainty or unstructured uncertainties (unmodeled) [17]. Structured uncertainty comes from inaccuracies that are included in the model, whereas unstructured uncertainty comes from underestimation of the system order [17]. Failure to accurately model the system can lead to negative effects on the control system, thus efforts to counter these inaccuracies have been developed in two forms: robust control and adaptive control. Robust control is composed of a nominal part and additional terms to address the uncertainty while adaptive control is similar but with the addition that the model is constantly updating the terms base on measured performance during operation [17].

A robust control is chosen to stabilize Air-Ball in the form of Sliding Mode Control (SMC). The controller is designed through a systematic approach outlined in [17] using the equation of motion derived in the previous chapter.

5.1 Sliding Mode Control

SMC is widely used in applications such as advanced underwater vehicles, high-performance aircraft, and space vehicles due to system parameter uncertainty [17]. The methodology of SMC is based on the idea that it is easier to control a 1st-order system than some nth order system, regardless of nonlinearity or not. Thus a notational simplification is needed to reduce the nth system to be a 1st order system, often in the form of a tracking error. After the simplification, it becomes intuitive to show that the newly transformed system can achieve perfect performance regardless of the uncertainty value; however, the cost of this perfect performance comes at the price of high control cost [17]. This typically leads to a modification of the control law, which exchanges tracking performance to reduce the cost of control.
5.2 Sliding Mode Control in General Form

The equivalent control approach for SMC solves for a control input that will ensure the system dynamics stays on the sliding surface; furthermore, by setting the derivative of the Lyapunov candidate function to be less than a negative value, this will ensure the sliding surface will be reached within a finite amount of time. The equivalent control is the control algorithm calculated from the mathematical model, and stability is reached once the equivalent control is equal to the actual control. Such control scheme is based on a Lyapunov approach which provides robustness in the presence of uncertainty.

A discontinuity function is required across the sliding surface to take into account modeling imprecision and disturbances [17]. However, this discontinuity function can induce a chattering phenomenon across the sliding surface. This is due to the fact that the switching action is not instantaneous which can lead to the controller compensating for the delay. Chattering is undesirable in practice due to the possibility of exciting high frequency motion that is neglected in the course of modeling and requires high control activity [17]. The remedy for chattering is to modify the discontinuity function to have a smoothed trade-off between the tracking performance and the control bandwidth. A thin boundary layer is applied around the switching surface. When the system is outside of the boundary layer the control law is made to guarantee attraction to the boundary layer. When the system is inside the boundary layer, the control law is interpolated as a smooth function.

First by defining the state vector in body frame as

\[
\dot{q}^B = \begin{bmatrix} V^B \\ \Omega^B \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix}
\]  

(27)

using the standard aircraft notation and assuming that Air-Ball is fully symmetrical, the equations of motion given in Equation 10 can be rewritten as
\[
\begin{bmatrix}
\dot{u} \\
\dot{v} \\
\dot{w} \\
\dot{p} \\
\dot{q} \\
\dot{r}
\end{bmatrix}
= 
\frac{1}{m} \left( \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix}^B + \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \right) - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} u \\ v \\ w \end{bmatrix} 
\]

(28)

Carrying out the multiplication yields

\[
\begin{bmatrix}
\dot{u} \\
\dot{v} \\
\dot{w} \\
\dot{p} \\
\dot{q} \\
\dot{r}
\end{bmatrix}
= 
\begin{bmatrix}
vr - wq + \frac{1}{m} F_x^B \\
wp - ur + \frac{1}{m} F_y^B \\
uq - vp + \frac{1}{m} F_z^B + mg \\
- \frac{1}{I_x} \left( l_y - l_z \right) qr + T_x^B \\
- \frac{1}{I_y} \left( l_z - l_x \right) pr + T_y^B \\
- \frac{1}{I_z} \left( l_x - l_y \right) pq + T_z^B
\end{bmatrix}
\]

(29)

Rearrange Equation 29 into a general form as follows

\[
H(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau
\]

(30)

where

\[
H(q) = 
\begin{bmatrix}
m & 0 & 0 & 0 & 0 & 0 \\
0 & m & 0 & 0 & 0 & 0 \\
0 & 0 & m & 0 & 0 & 0 \\
0 & 0 & 0 & l_x & 0 & 0 \\
0 & 0 & 0 & l_y & 0 & 0 \\
0 & 0 & 0 & 0 & l_z
\end{bmatrix}
\]

(31)
\[ C(q, \dot{q}) = \begin{bmatrix} 0 & r & q & 0 & 0 & 0 \\ -r & 0 & p & 0 & 0 & 0 \\ \theta & -p & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -l_x r & l_y q & 0 \\ 0 & 0 & 0 & l_x r & 0 & -l_x p \\ 0 & 0 & 0 & -l_y q & l_x p & 0 \end{bmatrix} \] (32)

\[ g(q) = \begin{bmatrix} -\frac{1}{2} \rho |u| u C_d A_x \\ -\frac{1}{2} \rho |v| v C_d A_y \\ -\frac{1}{2} \rho |w| w C_d A_z + g \end{bmatrix} \] (33)

where \( \rho \) is atmospheric density, \( C_d \) is the drag coefficient, \( A_x, A_y, A_z \) are the cross-sectional areas of Air-Ball in x-y-z plane, and \( g \) is the gravitational acceleration constant.

\[ \tau = \begin{bmatrix} F^B_x \\ F^B_y \\ F^B_z \\ \tau^B_x \\ \tau^B_y \\ \tau^B_z \end{bmatrix} \] (34)

Combining Equation 31 through Equation 34 into Equation 30 yields
as the equations of motion in general form that can be used for SMC design. The \( H(q) \) is the inertia matrix of Air-Ball, \( C(q, \dot{q}) \) is the Coriolis matrix that describes the coupling between the axes, \( g(q) \) is the vector containing external forces such as vehicle drag and gravity, \( \tau \) is the apply forces and torque.

The SMC requires a notation simplification to reduce the tracking states to tracking errors. Position tracking errors are defined as

\[
\ddot{q} = q - q_d
\]  \( (36) \)

where \( q_d \) is the desired position input, also known as position reference input for the controller. The velocity error can be expressed as the derivative of Equation 36 as

\[
\dot{q} = \dot{q} - q_d
\]  \( (37) \)

Next, the sliding surface is given as

\[
s = \ddot{q} + \Lambda \dot{q}
\]  \( (38) \)

such that \( \Lambda \) is the control bandwidth matrix that is a symmetric positive definite matrix given as
\[
\Lambda = \begin{bmatrix}
\lambda_1 & 0 & 0 & 0 \\
0 & \lambda_1 & 0 & 0 \\
0 & 0 & \lambda_3 & 0 \\
0 & 0 & 0 & \lambda_4 \\
0_{3\times3} & 0 & \lambda_5 & 0 \\
0 & 0 & 0 & \lambda_6
\end{bmatrix}
\]

Furthermore, Equation 38 can be rewritten with a reference velocity \( \dot{q}_r \) as

\[
s = \dot{\tilde{q}} + \Lambda \tilde{q} = \dot{q} - \dot{q}_r
\]

where \( \dot{q}_r \) is the reference velocity defined as

\[
\dot{q}_r = \dot{q}_d - \Lambda \tilde{q}
\]

Finally, the sliding condition which ensures the system reaches the sliding surface in a finite amount of time can be specified as

\[
\frac{1}{2} \frac{d}{dt} s^T s \leq -\eta (s^T s)^{\frac{1}{2}}
\]

The \( s^T s \) can be interpreted as the square norm of the distance to the sliding surface where \( s = 0 \) is the equilibrium point. By satisfying the sliding condition, the surface becomes an invariant set, meaning that even with some disturbances or uncertainties, the trajectory will remain on the sliding surface and thus reach the desired state over time.

The SMC control law is given as

\[
\tau = \dot{\tilde{r}} - K \text{sat}(s)
\]

where \( \dot{\tilde{r}} \) is equivalent control, \( K \) is the controller gain that is positive definite, and \( \text{sat}() \) is a saturation function with known bounds neighboring the sliding surface.

The estimation of the equivalent control can be determined based on specifying the error bound for the inertial matrix, Coriolis matrix, and external force vector.
The equivalent control given as

$$\hat{r} = \hat{H}\dot{q}_r + \hat{C}\dot{q}_r + \hat{g}$$  \hspace{1cm} (44)

where $\hat{H}$, $\hat{C}$, and $\hat{g}$ are the estimation parameters based on the modeling error bound. The modeling error bounds are given as

$$\tilde{H} = \hat{H} - H$$  \hspace{1cm} (45)

$$\tilde{C} = \hat{C} - C$$  \hspace{1cm} (46)

$$\tilde{g} = \hat{g} - g$$  \hspace{1cm} (47)

The implementation of the SMC into Simulink can be found in Appendix B.

5.3 PID Control for the Propulsion-Shafts

The propulsion-shaft position is controlled by a proportional-integral-derivative position feedback controller (PID). First by defining the position vector of each propulsion-shaft angle as

$$q_{\text{servo}} = \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \\ \delta_5 \\ \delta_6 \end{bmatrix}$$  \hspace{1cm} (48)

the position error is

$$\bar{q} = q_{\text{servo}} - q_{\text{servo}_d}$$  \hspace{1cm} (49)

where $q_{\text{servo}_d}$ is the desired rotation angle.

The control law of the PID is given as
\[ \tau_{shaft} = -k_p \ddot{q} - k_d \dot{q} - k_i \int \dot{q} \, dt \]  

(50)

where \( k_p \), \( k_d \), and \( k_i \) are the proportional gain, derivative gain, and integral gain respectively.

The implementation of the PID controller for the propulsion-shaft can be found in Appendix D.

5.4 Combining SMC with SimMechanics

As mentioned above, the flight simulator can be treated as the combination of two separate components. Component one is the SMC controller formulated in this Chapter using the equations of motion formulated in Chapter 3. The SMC controller is implemented in a Simulink block diagram using standard mathematics blocks and feedback from the plant model. Component two of the flight simulator is the plant model from SimMechanics. To couple the two components, the inputs and outputs of each of the components must be identified.

The SMC block diagram takes in the desired state from user commands and receives current state data from the plant model to calculate the state error. The SMC block diagram’s output is a \([6 \times 1]\) control vector, \( \tau \), as the right hand-side of Equation 35. The output control vector represents the required amount of force needed to achieve the desired state. The control vector information is converted to meaningful physical properties such as force and torque values by Equation 19. These values are accepted by the SimMechanics plant model as inputs used to calculate the next state of the system. The new state is passed back to the SMC block and the process continues to iterate.

A force element block is assigned to each motor. The block functions as an interface between normal Simulink blocks and SimMechanics blocks. The force element block accepts the result of Equation 19 as an input and assigns the force in the positive y direction of each motor. In the case of the motor CAD drawing, the thrust is generated in the positive
y direction due to the orientation of the drawing, the force resolution frame is set to ‘Attached Frame’ which ensures the applied force follows the coordinate system of the motor. The force vector is then transformed to the body frame of Air-Ball through the constraints set in the CAD assembly, the transformation is expressed mathematically through matrix $D$ given in Equation 18.
Chapter 6
Simulation Results and Discussion

The flight simulator is used to simulate Air-Ball performing different types of maneuvers. The flight simulation accounts for the thrust and torque balancing produced by each motor and uses the SMC and control action distribution matrix to control Air-Ball. The system dynamics due to motor reaction torque were not included in this simulation since motor responses are significantly faster when compared to the rest of the system dynamics and do not substantially affect the long-term motion of the system. Furthermore, motor dynamics add another layer of computation cost to the simulation solver.

The simulation solver uses a variable time step automatically picked by Simulink to determine the best solver configuration. This allows the Simulink solver to determine the fastest way to solve the system while maintaining an accuracy within the default relative and absolute tolerance of $1 \cdot 10^{-8}$. The average runtime for a 200 to 300 second simulation is about 10 to 15 minutes. Note that while the stiff solver settings such as ode23s, ode23t, and ode23tb reduce the computation time down to 20 seconds, the simulation results are inaccurate and cause the solver to terminate prematurely due to diverging solutions. Stiff solvers are designed to perform more computation work per step that leads to a larger time-step; this allows for the solver to solve the problem quickly with the cost of numerical accuracy.

6.1 Hovering

Starting with the motor configuration shown in Figure 21 (motors 1 - 4 produce the thrust for hover while motors 5 and 6 are idle), stable hover is easy to accomplish. However, during the hovering phase, a small downward pitching moment was induced by the weight of motors 5 as shown in Figure 22.
Figure 21: Air-Ball Motor Configuration
Figure 22: Air-Ball side view
The top motor in the motor pair induced a translation in the x-axis and a loss of altitude as shown in Figure 23, because the plane of motors 1-4 was pitches forward.

Figure 23: Position of Air-Ball (Hover)
Although the disturbance due to the weight of motor 5 was not modeled into the SMC’s estimation control, the controller was able to correct the pitching by increasing the RPM in motor 1 as shown in Figure 24.

6.2 Linear Translation

Each translation and attitude maneuver required different sets of command inputs from the signal generation block available in MATLAB/Simulink. All maneuvers started from the motor configuration shown in Figure 21.

6.2.1 Translation x-direction:

For translations along the x-axis the translation force was produced by motors 5 and 6, while motors 1 - 4 remained in the gravity off-loading position. No signal delay was needed in the beginning since motor 5 and motor 6 were already in the correct configuration with the thrust vectors pointed in the position x-direction. As shown in Figure 25, Air-Ball
was given the command to translate 1 meter in the positive x direction at the 10 seconds mark and arrived at the target distance at the 50 seconds mark.

\[ \text{Figure 25: Reference signal vs. model response} \]
The full x-translation trajectory is shown in Figure 26. During the translation time, motor 5 and motor 6 were rotated by 180° to change the thrust vector in the opposite direction. This was to allow Air-Ball to have the correct thrust orientation to stop moving once it has reached the target distance. After arriving at the target distance, Air-Ball was commanded to hold for 25 seconds before returning back to the origin.
Figure 27: Air-Ball position (X-Translation)

Figure 28: Air-Ball Attitude (X-Translation)
Disturbances in the x and y direction can be observed in Figure 27. At time 53 s, the x-position of Air-Ball shifted in the negative direction while the y-position started to shift at the 15 s mark. The shift in the y-direction corresponds to the rotation of motor 5 and motor 6 that induced a yaw rotation as shown in Figure 28 and Figure 30. The yaw motion was uncorrected for the remaining simulation and caused the thrust vectors of motor 5 and motor 6 to be generate a force component in the y-direction that led to a velocity change in the y-axis as shown in Figure 29.

Figure 29: Air-Ball linear velocity (X-Translation)
The spikes in the roll and pitch rotation rate correspond to the shift in the weight of motor 5. As motor 5 rotated for the thrust vector change, the moment created by the weight mentioned in the hovering section was moved from the pitch axis to the roll axis. From Figure 31 and Figure 32, motor RPM changes can be observed as motor 1-4 compensated for the disturbance caused by the weight of motor 5.
Figure 31: Motor 1 and Motor 3 RPM (X-Translation)

Figure 32: Motor 2 and Motor 3 RPM (X-Translation)
The spikes in RPM seen in Figure 33 were due to the instantaneous motor RPM impulses. Each spike occurred at the same time when the x-reference input changed. Since the translational velocity was not specified, the motor impulses started the maneuver, and Air-Ball drifted in the direction of the specified command.

No major disturbance was observed in the z-direction in the case of the x-translation. However, significant disturbances were presence for both x-direction and y-direction. These disturbances were due to the induced yaw rotation caused by turning motor 5 and motor 6 as well as the rotation induced by the weight of motor 5. Unwanted thrust components as the result of these disturbances led the translation in the x-direction to deviate away from the reference command by negative 0.05 m and moved Air-Ball in the y-direction by positive 0.20 m.
6.2.2 Translation in y-direction

Translation along the y-axis was similar to the x-axis case. Starting from the standard configuration, motor 5 and motor 6 were rotated by 90° to point the thrust vectors in the positive y-direction. This rotation was done in between the 5 s and 25 s marks before the y-translation command was given at the 30 s mark. The translation command as shown in Figure 34 commanded Air-Ball to translate 1 m in the positive y direction and holding at the destination for 50 s before returning to the initial position.

![Figure 34: Reference signal vs. model response](image)
The y-translation trajectory is shown in Figure 35. The same technique of repositioning motor 5 and motor 6 thrust vectors has been used to execute the translation in y-direction. The two motors were rotated by 90° to align both thrust vectors in the positive y-direction that will allow Air-Ball to initiate the translation motion. Then, the two motors were rotated by 180° to align both thrust vectors in the negative y-direction. This was to ensure that Air-Ball can be stopped at the target distance and returning to the initial position. The last servo rotation maneuver rotated both motors back to aligning with the positive y-direction to allow Air-Ball to stop at the origin.
Figure 36: Air-Ball position (Y-Translation)

Figure 37: Air-Ball Attitude (Y-Translation)
In Figure 36, similar disturbances occurred in the x-translation can be observed in the y-translation. The x-translation was disturbed by the yaw induced by repositioning motor 5 and motor 6. Air-Ball moved in the negative x-direction by 0.20 m similar to the x-translation where it moved in the y-direction by 0.20 m as seen in Figure 27.

Figure 38: Air-Ball linear velocity (Y-Translation)
Figure 39: Air-Ball rotation rate (Y-Translation)

The x-axis disturbance induced by the yaw motion can be observed Figure 38 where the velocity profile in the x-direction is similar to the y-axis velocity profile in the x-translation case. The spikes in roll angular rate in Figure 39 was due to the weight of motor 5 as it rotated to reconfgure the thrust vector.
Figure 40: Motor 1 and Motor 3 RPM (Y-Translation)

Figure 41: Motor 2 and Motor 4 RPM (Y-Translation)
The same motor RPM changes in the x-translation can be observed here for the y-translation in Figure 40 and Figure 41; as motor 1-4 compensates for the disturbance caused by the weight shift of motor 5.

![Figure 42: Motor 5 and Motor 6 RPM (Y-Translation)](image)

The RPM spikes for motor 5 and motor 6 are observed in Figure 42, similar to Figure 33 for the x-translation maneuver.

### 6.2.3 Translation in z-direction

For translation along the z-axis, the translation force was produced by motors 1 - 4 while motors 5 and 6 were not active throughout the simulation. Air-Ball was able to reach the desires position at 1 meter in the positive z-direction and return to its initial position at the end of the simulation, as shown in Figure 43. The z-translation trajectory is shown in Figure 44.
Figure 43: Reference signal vs. model response

Figure 44: Z translation trajectory
Some disturbances in the x-direction were present during the z-direction translation. This was due to the induced pitch by the weight of motor 5 as mentioned above. The plane of motors 1-4 was pitched forward as shown in Figure 46 and created an x velocity component as shown in Figure 47.
Figure 46: Air-Ball Attitude (Z-Translation)

Figure 47: Air-Ball linear velocity (Z-Translation)
Figure 48: Air-Ball rotation rate (Z-Translation)

Figure 49: Motor 1 and Motor 3 RPM (Z-Translation)
The spikes of pitch angular rate shown in Figure 48 correspond to the spikes in motor 1 motor 3 shown in Figure 49. These spikes correspond to the impulses generated from the reference command changes in Figure 43. The same behavior can be seen in Figure 50 for motor 2 and motor 4.

![Figure 50: Motor 2 and Motor 4 RPM (Z-Translation)]
In Figure 51, no RPMs were generated for motor 5 and motor 6 since none of their thrust vectors had an influence in the z-direction.

### 6.3 Angular Rotation

The same principle of delay input was implemented for the angular rotation simulation. During the pitch and roll rotation maneuver, the motor pair that is on the rotation axis must rotate in the opposite direction of the vehicle rotation at the same rate. This is to ensure that this particular motor pair will remain pointing in the opposite direction of gravity.

#### 6.3.1 Pitch Rotation

As shown in Figure 52, Air-Ball was given the command to start the pitch maneuver at the 10 s mark to reach the desired pitch angle of $85^\circ$ at the 100 s mark. After reaching $85^\circ$, Air-Ball holds the pitch position for 100 s before returning to the initial attitude orientation.
As shown in Figure 53 below, a loss of 1 cm in altitude can be observed as well as a small translation in the x-direction by less than 0.005 m. The largest translation disturbance was in the y-direction by 0.03 m. Although disturbances were presence in all three axes, the magnitudes were small and did not affect the tracking performance of the pitch command and other attitude orientations as seen in Figure 52 above and Figure 54 below.

Figure 52: Reference signal vs. model response
Figure 53: Air-Ball position (Pitch-Rotation)

Figure 54: Air-Ball Attitude (Pitch-Rotation)
The velocity spike in the x-direction in Figure 55 was caused by the spike of RPM in motor 1-3 and motor 5-6 shown in Figure 57 and Figure 59 to start the pitch maneuver. The other velocity spikes in the x-direction are also attributed from the RPM spikes in motor 1-3 and motor 5-6 for stopping the pitch rotation at 85° and initiating the pitch maneuver back to the initial orientation. The y-direction velocity profile was caused by the chattering in motor 2-4 RPM the 10 s to 100 s mark as well as the 200 s to 300 s mark.

![Figure 55: Air-Ball linear velocity (Pitch-Rotation)](image)

Figure 55: Air-Ball linear velocity (Pitch-Rotation)
Figure 56: Air-Ball Angular Velocity (Pitch-Rotation)

Following the motor configuration in Figure 21, Air-Ball uses the differential thrust in motor 1-3 and motor 5-6 to generate the torque for the pitch motion. The angular rate of propulsion-shaft 2-4 were set to be the same as the pitch rotation rate, as shown in Figure 56, but in the opposite direction. This is to ensure the thrust vectors from motor 2-4 remains pointed in the negative z-direction for gravity off-loading.

The RPM from motor 1-3 decreases as the pitch angle increases, as shown in Figure 57; while the RPM from motor 5-6 increases to compensate for the loss of thrust from motor 1-3, as shown in Figure 59.
Figure 57: Motor 1 and Motor 3 RPM (Pitch-Rotation)

Figure 58: Motor 2 and Motor 4 RPM (Pitch-Rotation)
The RPM profile in motor 2-4 were constant throughout the simulation, with some exception of spikes at timestamps where the pitch input command changes.

Figure 59: Motor 5 and Motor 6 RPM (Pitch-Rotation)

6.3.2 Roll Rotation

As shown in Figure 60, the roll input command required a delay of 30 s for motor 5-6 to reposition the thrust vectors to align in the negative y-direction. The remaining roll command input is similar to the pitch command input.
From Figure 61, the x-position of Air-Ball was deviated to 0.05 m at around the 150 s mark, the y-position was deviated in the negative direction by 0.04 m, and a loss of altitude by less than 0.02 m. The linear velocity profiles in Figure 64 are due to the repositioning of motor 5 and motor 6 happening more frequently that caused the yaw angle to be disturbed more as shown in Figure 62. Although the translational disturbances were presence, the magnitudes were within 0.05 m and did not affect the overall rotation maneuver as shown in Figure 60 and Figure 63.
Figure 61: Air-Ball position (Roll-Rotation)

Figure 62: Air-Ball yaw angle (Roll-Rotation)
Figure 63: Air-Ball Attitude (Roll-Rotation)

Figure 64: Air-Ball linear velocity (Roll-Rotation)
By following the motor configuration in Figure 21, for the roll maneuver Air-Ball uses the differential thrust in motor 2-4 and motor 5-6 to generate the torque for the roll motion. The angular rate of propulsion-shaft 1-3 were set to be the same as the roll rotation rate, as shown in Figure 65, but in the opposite direction that ensures the thrust vector of motor 1-3 remains aligned with the negative z-direction for gravity off-loading.

![Figure 65: Air-Ball angular velocity (Roll-Rotation)]
The RPM from motor 1 started out higher than motor 3 to compensate for the pitch down motion caused by the weight of motor 5 as shown in Figure 66. As motor 5 repositions to align in the negative y-direction, motor 1-3 RPM converges to the same value; at the same time motor 2 increases its RPM to account for the weight of motor 5, shown in Figure 67. Motor 2-4 RPM decreases as the roll angle increases; while motor 5-6 RPM increases to compensate for the loss of thrust from motor 2-4, as shown in Figure 68 below. The same RPM change fashion can be observed as Air-Ball returns from the 85° roll angle back to the initial orientation.
Figure 67: Motor 2 and Motor 4 RPM (Roll-Rotation)

Figure 68: Motor 5 and Motor 6 RPM (Roll-Rotation)
6.3.3 Yaw Rotation

Conventional multi-rotorcraft use the motor acceleration to control the yaw rotation. Since motor dynamics were not included in the model, another approach was attempted to control yaw rotation. Propulsion-shafts 2 and 4 are offset by 5° to produce thrust vectors for the yaw motion. By rotating the propulsion shaft of motor 2 and motor 4 in the opposite direction, the thrust in motor 2-4 generate a second component in the horizontal direction. These horizontal components yield a rotation about the z-axis as shown in Figure 69 below.

![Figure 69: Air-Ball attitude (Yaw-Rotation)](image_url)

Although the yaw rotation was possible by turning propulsion-shaft 2-4, the yaw motion was uncontrollable and no command follow was possible. The yaw procedure was challenging and adequate timing of the propulsion shaft rotation input was required to ensure Air-Ball does not exceed the 90° rotation limit. Moreover, the disturbances in the x and y translation are largely due to the lack of control in both directions as shown in Figure 70.
6.4 Translation and Rotation on the same Axis

The last simulation combines the linear translation and the angular rotation on the same axis. Since Air-Ball is Omni-directional, the case of linear translation in the x-direction and roll rotation were simulated as shown in Figure 71 and Figure 72.
Figure 71: Air-Ball position (Translation-Rotation)

Figure 72: Air-Ball attitude (Translation-Rotation)
The simulation starts with Air-Ball in the initial hover configuration shown in Figure 11. Motors 5 and 6 were pointed in the positive x-axis and motors 1 - 4 provided the hovering thrust. Motors 5 and 6 provided the impulsive thrust at the 10 s mark to initiate the x-translation. As seen in Figure 73, the velocity gradient in the x-direction was due to the drag force in the x-direction. The velocity spike in the y-direction was due to the induced yaw motion from repositioning of motor 5-6, as shown in Figure 74 between the 25 s to 45 s mark. Motor 5 and motor 6 were repositioned to point along the negative y-axis so the roll maneuver can be performed.
Figure 74: Air-Ball angular velocity (Translation-Rotation)

Figure 75: Motor 1 and Motor 3 RPM (Translation-Rotation)
As Air-Ball translates in the x-direction and rotates on the x-axis simultaneously, some gravity off-loading thrust in motor 2-4 were lost due to performing the roll rotation; thus an increase in motor 1-3 RPM was observed between the 50 s mark to the 200 s mark, as shown in Figure 75 above.

The roll command was given after motor 5 and motor 6 are fixed in their commanded positions and using the same principle mentioned in the roll rotation section above, the differential thrust in motor 5-6 and motor 2-4 performed the rolling motion, as shown in Figure 76 and Figure 77.

![Figure 76: Motor 2 and Motor 4 RPM (Translation-Rotation)](image-url)
Figure 77: Motor 5 and Motor 6 RPM (Translation-Rotation)
Chapter 7 Conclusion and Future Work

7.1 Conclusion

Rotorcraft have become popular research and prototyping instruments among industry and academia due to their flexibility and affordability. The unique multi-rotorcraft capable of six DOF, Air-Ball, was developed to address missing capabilities such as micro-gravity flight experiment in a ground facility, Omni-directional flight for multi-rotorcraft, and decoupled translation and rotation flight. The complexity of Air-Ball posed challenges in terms of modeling the vehicle mathematically and developing an adequate controller for flight. This thesis document addresses these problems by introducing an alternative method for model based design by using SimMechanics. The combined utilization of SimMechanics model and mathematical model led to the development of a flight simulator that was used to understand Air-Ball’s vehicle dynamics and kinematics. The plant model generated by SimMechanics compared well with the mathematical model derived through first principles as shown in Figure 19 and Figure 20 in Chapter 4.

The error between the reference command and the plant output falls within 0.05 m for all pure translation cases and less than 1° for pitch and roll rotation cases. The translational disturbances in the rotational case were under 0.05 m, despite the unintended yaw motion from repositioning each motor. Although the pure yaw rotation was possible by offsetting propulsion-shaft angle by 5°, there was no command following in the yaw rotation due to the horizontal thrust components of motor 2-4 being unregulated as the result of shaft angles being an user-input.

7.2 Future Work

7.2.1 Hardware Implementation to Validate Simulation

More effort is required in hardware development to ensure the accuracy of the flight simulator. One example of this hardware adjustment would be counterweights on each motor to prevent unwanted rotation. Another development would be angular position feedback and
rotation speed control on the servo motors. This is crucial in the angular rotation maneuver since the shaft rotation speed must match the rotation speed of the vehicle. The position information feedback of each servo is also important for the SMC and control action matrix. To prevent weight shifting and inertia changes during flight, a new design of the electronics component box is needed to secure all of the electronic components inside. The current design zip-ties down the electronic components and can come loose during flight. Similar to the ODAV, reversible fixed-pitch propellers can be added to provide positive and negative thrust vector to reduce the unintended yaw motion caused by rotating the propulsion-shaft. Finally, the simulation performance of the SMC should be compared to the hardware performance to ensure the SMC can stabilize the actual vehicle.

7.2.2 Modeling of Motor Dynamics

Previous assumptions stated that motor the dynamics are instantaneous when compared to vehicle dynamics; therefore, the modeling of the motors was simplified. Although this assumption is true, the torque difference among each set of motor pairs has significant impact on vehicle dynamics. The torque differential can even be use for yaw rotation control and stabilization. Including motor dynamics into the flight simulation will increase the simulation validity at the cost of higher computation power requirement.

7.2.3 Using Quaternions to Describe Attitude Information

As mentioned in the modeling chapter, the simulation cannot exceed over ± 90° due to gimbal lock. A quaternion based model, although not as intuitive as Euler angle, will be able to eliminate the gimbal lock effect and improve simulation runtime. Quaternions are more efficient to solve algebraically and therefore reduce the computation power requirement.

7.2.4 Alternative Controller Designs

Alternative control laws such as PID control, linear feedback control, and adaptive control should be investigated to determine the best controller for Air-Ball. An adaptive controller is the next step up in development from SMC because of its adaptive feature in
addressing model uncertainty. The SMC operates with a constant inertia matrix, but in reality the matrix can change based on vehicle states. This is because electronic the component placement in the center box is not symmetrical and can shift during flight. Therefore, a controller that can adapt to the changing inertia matrix is highly recommended.

7.2.5 Closed-loop Control and Optimization

The presented flight simulation requires the mid-flight motor configuration to be predetermined by the user. The motor configuration information is used in the control action distribution matrix to compute the force for each motor to perform a certain maneuver. Although the above method is sufficient in studying Air-Ball’s flight behavior, a closed-loop controller and optimization feature are needed for future studies. The closed-loop controller and optimization feature will replace the user input step, which predetermines the motor configuration. This optimization procedure will require an additional control algorithm and an expanded distribution matrix to account for additional state feedback. The proposed optimized controller, if successful, will determine the best method to achieve a certain maneuver given the current state of the vehicle.
References


## Appendix A Thrust Stand Data

<table>
<thead>
<tr>
<th>Duty Cycle</th>
<th>RPM</th>
<th>Angular speed (rad/s)</th>
<th>Frequency (Hz)</th>
<th>Corrected Force (g)</th>
<th>Corrected Force (N)</th>
<th>K constant</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>550</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
</tr>
<tr>
<td>1</td>
<td>73.72</td>
<td>7.719940347</td>
<td>550</td>
<td>0</td>
<td>0</td>
<td>0.1</td>
</tr>
<tr>
<td>2</td>
<td>147.44</td>
<td>14.3988609</td>
<td>550</td>
<td>50</td>
<td>69.6</td>
<td>0.2</td>
</tr>
<tr>
<td>3</td>
<td>221.16</td>
<td>23.35982104</td>
<td>550</td>
<td>75</td>
<td>104.4</td>
<td>0.2</td>
</tr>
<tr>
<td>4</td>
<td>294.88</td>
<td>30.87976139</td>
<td>550</td>
<td>100</td>
<td>139.2</td>
<td>0.3</td>
</tr>
<tr>
<td>5</td>
<td>368.6</td>
<td>38.59970174</td>
<td>550</td>
<td>125</td>
<td>174.0</td>
<td>0.4</td>
</tr>
<tr>
<td>6</td>
<td>442.32</td>
<td>46.31964208</td>
<td>550</td>
<td>150</td>
<td>208.8</td>
<td>0.5</td>
</tr>
<tr>
<td>7</td>
<td>516.04</td>
<td>54.03958243</td>
<td>550</td>
<td>175</td>
<td>243.5</td>
<td>0.5</td>
</tr>
<tr>
<td>8</td>
<td>589.76</td>
<td>61.75952278</td>
<td>550</td>
<td>200</td>
<td>278.3</td>
<td>0.6</td>
</tr>
<tr>
<td>9</td>
<td>663.48</td>
<td>69.47946313</td>
<td>550</td>
<td>225</td>
<td>313.1</td>
<td>0.7</td>
</tr>
<tr>
<td>10</td>
<td>737.2</td>
<td>77.19940347</td>
<td>550</td>
<td>250</td>
<td>347.9</td>
<td>0.8</td>
</tr>
<tr>
<td>11</td>
<td>810.92</td>
<td>84.91934382</td>
<td>550</td>
<td>275</td>
<td>382.7</td>
<td>0.8</td>
</tr>
<tr>
<td>12</td>
<td>884.64</td>
<td>92.63928417</td>
<td>550</td>
<td>300</td>
<td>417.5</td>
<td>0.9</td>
</tr>
<tr>
<td>13</td>
<td>958.36</td>
<td>100.3592245</td>
<td>550</td>
<td>325</td>
<td>452.3</td>
<td>1.0</td>
</tr>
<tr>
<td>14</td>
<td>1032.08</td>
<td>108.0791649</td>
<td>550</td>
<td>350</td>
<td>487.1</td>
<td>1.1</td>
</tr>
<tr>
<td>15</td>
<td>1105.8</td>
<td>115.7991052</td>
<td>550</td>
<td>375</td>
<td>521.9</td>
<td>1.2</td>
</tr>
<tr>
<td>16</td>
<td>1179.52</td>
<td>123.5190456</td>
<td>550</td>
<td>400</td>
<td>556.7</td>
<td>1.2</td>
</tr>
<tr>
<td>17</td>
<td>1253.24</td>
<td>131.2389859</td>
<td>550</td>
<td>425</td>
<td>591.5</td>
<td>1.3</td>
</tr>
<tr>
<td>18</td>
<td>1326.96</td>
<td>138.9589263</td>
<td>550</td>
<td>450</td>
<td>626.3</td>
<td>1.4</td>
</tr>
<tr>
<td>19</td>
<td>1400.68</td>
<td>146.6788666</td>
<td>550</td>
<td>475</td>
<td>661.1</td>
<td>1.5</td>
</tr>
<tr>
<td>20</td>
<td>1474.4</td>
<td>154.3988069</td>
<td>550</td>
<td>500</td>
<td>695.9</td>
<td>1.5</td>
</tr>
<tr>
<td>21</td>
<td>1548.12</td>
<td>162.1187473</td>
<td>550</td>
<td>525</td>
<td>730.6</td>
<td>1.6</td>
</tr>
<tr>
<td>22</td>
<td>1621.84</td>
<td>169.8386876</td>
<td>550</td>
<td>550</td>
<td>765.4</td>
<td>1.7</td>
</tr>
<tr>
<td>23</td>
<td>1695.56</td>
<td>177.558628</td>
<td>550</td>
<td>575</td>
<td>800.2</td>
<td>1.8</td>
</tr>
<tr>
<td>24</td>
<td>1769.28</td>
<td>185.2785683</td>
<td>550</td>
<td>600</td>
<td>835.0</td>
<td>1.8</td>
</tr>
<tr>
<td>25</td>
<td>1843.01</td>
<td>192.9880587</td>
<td>550</td>
<td>625</td>
<td>869.8</td>
<td>1.9</td>
</tr>
<tr>
<td>26</td>
<td>1916.72</td>
<td>200.718449</td>
<td>550</td>
<td>650</td>
<td>904.6</td>
<td>2.0</td>
</tr>
<tr>
<td>27</td>
<td>1990.44</td>
<td>208.4383894</td>
<td>550</td>
<td>675</td>
<td>939.4</td>
<td>2.1</td>
</tr>
<tr>
<td>28</td>
<td>2064.16</td>
<td>216.1583297</td>
<td>550</td>
<td>700</td>
<td>974.2</td>
<td>2.1</td>
</tr>
<tr>
<td>29</td>
<td>2137.88</td>
<td>223.8787201</td>
<td>550</td>
<td>725</td>
<td>1009.0</td>
<td>2.2</td>
</tr>
<tr>
<td>30</td>
<td>2211.6</td>
<td>231.5982104</td>
<td>550</td>
<td>750</td>
<td>1043.8</td>
<td>2.3</td>
</tr>
<tr>
<td>31</td>
<td>2285.32</td>
<td>239.3181508</td>
<td>550</td>
<td>775</td>
<td>1078.6</td>
<td>2.4</td>
</tr>
<tr>
<td>32</td>
<td>2359.04</td>
<td>247.0380911</td>
<td>550</td>
<td>800</td>
<td>1113.4</td>
<td>2.5</td>
</tr>
<tr>
<td>33</td>
<td>2432.76</td>
<td>254.7580315</td>
<td>550</td>
<td>825</td>
<td>1148.2</td>
<td>2.5</td>
</tr>
<tr>
<td>34</td>
<td>2506.48</td>
<td>262.4779718</td>
<td>550</td>
<td>850</td>
<td>1182.9</td>
<td>2.6</td>
</tr>
<tr>
<td>35</td>
<td>2580.2</td>
<td>270.1997122</td>
<td>550</td>
<td>875</td>
<td>1217.7</td>
<td>2.7</td>
</tr>
<tr>
<td>36</td>
<td>2653.92</td>
<td>277.9178525</td>
<td>550</td>
<td>900</td>
<td>1252.5</td>
<td>2.8</td>
</tr>
<tr>
<td>37</td>
<td>2727.64</td>
<td>285.6377929</td>
<td>550</td>
<td>925</td>
<td>1287.3</td>
<td>2.8</td>
</tr>
<tr>
<td>38</td>
<td>2801.36</td>
<td>293.3577332</td>
<td>550</td>
<td>950</td>
<td>1322.1</td>
<td>2.9</td>
</tr>
<tr>
<td>39</td>
<td>2875.08</td>
<td>301.0776735</td>
<td>550</td>
<td>975</td>
<td>1356.9</td>
<td>3.0</td>
</tr>
<tr>
<td>40</td>
<td>2948.8</td>
<td>308.7976139</td>
<td>550</td>
<td>1000</td>
<td>1391.7</td>
<td>3.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>41</td>
<td>3022.52</td>
<td>316.5175542</td>
<td>550</td>
<td>1025</td>
<td>1426.5</td>
<td>14.0</td>
</tr>
<tr>
<td>42</td>
<td>3096.24</td>
<td>324.2374946</td>
<td>550</td>
<td>1050</td>
<td>1461.3</td>
<td>14.3</td>
</tr>
<tr>
<td>43</td>
<td>3169.96</td>
<td>331.9573439</td>
<td>550</td>
<td>1075</td>
<td>1496.1</td>
<td>14.7</td>
</tr>
<tr>
<td>44</td>
<td>3243.68</td>
<td>339.6773753</td>
<td>550</td>
<td>1100</td>
<td>1530.9</td>
<td>15.0</td>
</tr>
<tr>
<td>45</td>
<td>3317.4</td>
<td>347.3973156</td>
<td>550</td>
<td>1125</td>
<td>1565.7</td>
<td>15.4</td>
</tr>
<tr>
<td>46</td>
<td>3391.12</td>
<td>355.1172859</td>
<td>550</td>
<td>1150</td>
<td>1600.5</td>
<td>15.7</td>
</tr>
<tr>
<td>47</td>
<td>3464.84</td>
<td>362.8372262</td>
<td>550</td>
<td>1175</td>
<td>1635.2</td>
<td>16.0</td>
</tr>
<tr>
<td>48</td>
<td>3538.56</td>
<td>370.5571665</td>
<td>550</td>
<td>1200</td>
<td>1670.0</td>
<td>16.4</td>
</tr>
<tr>
<td>49</td>
<td>3612.28</td>
<td>378.2771068</td>
<td>550</td>
<td>1225</td>
<td>1704.8</td>
<td>16.7</td>
</tr>
</tbody>
</table>

...
<table>
<thead>
<tr>
<th></th>
<th>5971.32</th>
<th>625.3151681</th>
<th>550</th>
<th>2025</th>
<th>2818.2</th>
<th>27.6</th>
<th>6.2</th>
<th>0.00463</th>
</tr>
</thead>
<tbody>
<tr>
<td>82</td>
<td>6045.04</td>
<td>633.0351085</td>
<td>550</td>
<td>2050</td>
<td>2853.0</td>
<td>28.0</td>
<td>6.3</td>
<td>0.00463</td>
</tr>
<tr>
<td>83</td>
<td>6118.76</td>
<td>640.7550488</td>
<td>550</td>
<td>2075</td>
<td>2887.8</td>
<td>28.3</td>
<td>6.4</td>
<td>0.00463</td>
</tr>
<tr>
<td>84</td>
<td>6192.48</td>
<td>648.4749892</td>
<td>550</td>
<td>2100</td>
<td>2922.6</td>
<td>28.7</td>
<td>6.4</td>
<td>0.00463</td>
</tr>
<tr>
<td>85</td>
<td>6266.20</td>
<td>656.1949295</td>
<td>550</td>
<td>2125</td>
<td>2957.4</td>
<td>29.0</td>
<td>6.5</td>
<td>0.00463</td>
</tr>
<tr>
<td>86</td>
<td>6339.92</td>
<td>663.9148699</td>
<td>550</td>
<td>2150</td>
<td>2992.2</td>
<td>29.4</td>
<td>6.6</td>
<td>0.00463</td>
</tr>
<tr>
<td>87</td>
<td>6413.64</td>
<td>671.6348102</td>
<td>550</td>
<td>2175</td>
<td>3026.9</td>
<td>29.7</td>
<td>6.7</td>
<td>0.00463</td>
</tr>
<tr>
<td>88</td>
<td>6487.36</td>
<td>679.3547506</td>
<td>550</td>
<td>2200</td>
<td>3061.7</td>
<td>30.0</td>
<td>6.8</td>
<td>0.00463</td>
</tr>
<tr>
<td>89</td>
<td>6561.08</td>
<td>687.0746909</td>
<td>550</td>
<td>2225</td>
<td>3096.5</td>
<td>30.4</td>
<td>6.8</td>
<td>0.00463</td>
</tr>
<tr>
<td>90</td>
<td>6634.80</td>
<td>694.7946313</td>
<td>550</td>
<td>2250</td>
<td>3131.3</td>
<td>30.7</td>
<td>6.9</td>
<td>0.00463</td>
</tr>
<tr>
<td>91</td>
<td>6708.52</td>
<td>702.5145716</td>
<td>550</td>
<td>2275</td>
<td>3166.1</td>
<td>31.1</td>
<td>7.0</td>
<td>0.00463</td>
</tr>
<tr>
<td>92</td>
<td>6782.24</td>
<td>710.2345112</td>
<td>550</td>
<td>2300</td>
<td>3200.9</td>
<td>31.4</td>
<td>7.1</td>
<td>0.00463</td>
</tr>
<tr>
<td>93</td>
<td>6855.96</td>
<td>717.9544523</td>
<td>550</td>
<td>2325</td>
<td>3235.7</td>
<td>31.7</td>
<td>7.1</td>
<td>0.00463</td>
</tr>
<tr>
<td>94</td>
<td>6929.68</td>
<td>725.6743927</td>
<td>550</td>
<td>2350</td>
<td>3270.5</td>
<td>32.1</td>
<td>7.2</td>
<td>0.00463</td>
</tr>
<tr>
<td>95</td>
<td>7003.40</td>
<td>733.3943333</td>
<td>550</td>
<td>2375</td>
<td>3305.3</td>
<td>32.4</td>
<td>7.3</td>
<td>0.00463</td>
</tr>
<tr>
<td>96</td>
<td>7077.12</td>
<td>741.1142734</td>
<td>550</td>
<td>2400</td>
<td>3340.1</td>
<td>32.8</td>
<td>7.4</td>
<td>0.00463</td>
</tr>
<tr>
<td>97</td>
<td>7150.84</td>
<td>748.8342137</td>
<td>550</td>
<td>2425</td>
<td>3374.9</td>
<td>33.1</td>
<td>7.4</td>
<td>0.00463</td>
</tr>
<tr>
<td>98</td>
<td>7224.56</td>
<td>756.5541544</td>
<td>550</td>
<td>2450</td>
<td>3409.7</td>
<td>33.4</td>
<td>7.5</td>
<td>0.00463</td>
</tr>
<tr>
<td>99</td>
<td>7298.28</td>
<td>764.2740944</td>
<td>550</td>
<td>2475</td>
<td>3444.5</td>
<td>33.8</td>
<td>7.6</td>
<td>0.00463</td>
</tr>
<tr>
<td>100</td>
<td>7372.00</td>
<td>771.9940347</td>
<td>550</td>
<td>2500</td>
<td>3479.3</td>
<td>34.1</td>
<td>7.7</td>
<td>0.00463</td>
</tr>
</tbody>
</table>
Appendix B SMC in Simulink
Appendix C SimMechanics Model Tree
Appendix D PID Servo Controller in Simulink