CONTROL ALLOCATION METHODS FOR CONSTRAINED AND OVER ACTUATED SATELLITE ATTITUDE CONTROL SYSTEMS

THESIS

Jonathan M. Childress

AFIT-ENY-MS-17-M-251

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this document are those of the author and do not reflect the official policy or position of the United States Air Force, the United States Department of Defense or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT-ENY-MS-17-M-251

CONTROL ALLOCATION METHODS FOR CONSTRAINED AND OVER ACTUATED SATELLITE ATTITUDE CONTROL SYSTEMS

THESIS

Presented to the Faculty
Department of Astronautical Engineering
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Astronautical Engineering

Jonathan M. Childress, B.S.M.E.

March 3, 2017

AFIT-ENY-MS-17-M-251

CONTROL ALLOCATION METHODS FOR CONSTRAINED AND OVER ACTUATED SATELLITE ATTITUDE CONTROL SYSTEMS

Jonathan M. Childress, B.S.M.E.

Committee Membership:

Dr. Eric Swenson,
Chair

Lt Col Kirk Johnson,
Member

Capt Joshuah Hess,
Member

1Lt Dylan Penn
Member

AFIT-ENY-MS-17-M-251

Abstract

The research presented in this thesis compares the numerically simulated performance of various control allocation methods applied to the constrained and over actuated control system of the Air Force Research Laboratory, Space Vehicle Directorate's satellite simulator, the Resilient Bus Experiment Laboratory (REBEL). The REBEL prototype reaction wheel array design has not yet evaluated the implementation of control allocation methods in its attitude control system. As a result, three different control allocation methods are formulated and applied to the simulation: the simple pseudo inverse, the redistributed pseudo inverse, and the redistributed pseudo inverse with adaptive weighting (a newly developed control allocation variant.) The equations of motion (kinematics and kinetics) for a satellite simulator utilizing a reaction wheel array as an attitude control system are also derived and implemented in the simulation. The control law for a basic high level controller is defined and applied to the simulation as well. A single simulated maneuver is executed with the control system utilizing each of the control allocation methods independently. This single maneuver is evaluated in detail to provide insight into the control allocation methods' functionality. A set of multiple maneuvers is also executed for each control allocation method and the resulting performance metrics are averaged. Based on these results, conclusions are drawn about the different control allocation methods' potential for implementation on REBEL. The results indicate that redistributed pseudo inverse and the redistributed pseudo inverse with adaptive weighting both show promise as potential control allocation methods REBEL could utilize.

Acknowledgments

Dr. Eric Swenson served as my research advisor in this endeavor. His knowledge, resources, guidance, and encouragement were invaluable. Without his help, this research effort could not have been completed. He was always available to provide the assistance I needed, or offer kind words of encouragement.

A special thanks is in order for Lt Col Kirk Johnson and Capt Joshuah Hess who have offered their valuable time to serve as committee members for this thesis.

Dr. Richard Cobb and Col Tim Sands both served as excellent instructors in several controls courses I completed at AFIT. Their knowledge and enthusiasm of the material they taught helped to spark my interest in the field.

1Lt Dylann Penn served as my contact point with AFRL/RV, and provided a wealth of information about the REBEL satellite simulator. He also provided advice that helped to focus the topic of my research into something manageable.

Mr. Phillip Smith and Mr. James Herner were excellent sources of information on the mechanical and electrical aspects of the REBEL satellite simulator reaction wheel design. Their input helped to demystify some of the more convoluted aspects of their operation.

Dr. Michael Bohun, my supervisor at the WPAFB Landing Gear Test Facility, gave me the encouragement, guidance, and time I needed to pursue my interests and perform this research.

Mr. Andrew Zakrajsek, a friend and colleague, provided a great deal of advice and guidance. Without his assistance I would not have had the opportunity to attend graduate school at AFIT.

Jonathan M. Childress

Table of Contents

Page

List of Figures

Figure Page

List of Tables

Table Page

List of Acronyms

ACS Attitude Control System

AFIT Air Force Institute of Technology

AFRL The Air Force Research Laboratory

EOM Equation of Motion

MAMS Maximum Achievable Moment Set

MOI Mass Moment of Inertia

PD Proportional Differential

REBEL Resilient Bus Experiment Laboratory

RPI Redistributed Pseudo Inverse

RPIW Redistributed Pseudo Inverse with Adaptive Weighting

RW Reaction Wheel

RWA Reaction Wheel Array

SPI Simple Pseudo Inverse

SVD Singular Value Decomposition

List of Symbols

α	Adaptive Actuator Weight
$\bar{\phi}$	Upper Saturation Boundary
$\bar{\psi}_0$	Initial Reaction Wheel Speeds
$\bar{q}$	Quaternion Array
$\bar{q}_0$	Initial Quaternion
$\bar{q}_f$	Final Quaternion
$\bar{q}_{error}$	Quaternion Orientation Error
$\bar{u}$	Array of Actuator Torques
$\bar{u}_c$	Constrained Actuator Subset
$\bar{u}_u$	Unconstrained Actuator Subset
$\bar{x}$	State Vector
$\bar{x}_0$	Initial Conditions of Simulation
Δt	Simulation Time Step / Controller Update Rate
$\dot{\bar{q}}$	Quaternion Rates
$\dot{\bar{x}}$	Time Derivative of State Vector
$\dot{\psi}$	Scalar Axial Angular Acceleration of Reaction Wheel
$\dot{\vec{\omega}}$	Angular Acceleration
$\mathbb{U}$	Actuator Constraint Space
ψ	Scalar Axial Angular Velocity of Reaction Wheel
ψ_{max}	Maximum Reaction Wheel Speed
τ_{max}	Maximum Reaction Wheel Torque
τ_{nom}	Nominal Motor Torque (Max Continuous)
$\underline{\phi}$	Lower Saturation Boundary
$\vec{\omega}$	Angular Rate
$\vec{\omega}_0$	Initial Angular Rates
$\vec{\tau}$	Allocated Torques
$\vec{\tau}_c$	Torque Produced by the Constrained Actuators
$\vec{\tau}_{com}$	Commanded Torques
$\vec{H}$	Angular Momentum Vector
$\vec{H}$	Angular Momentum Vector
$\vec{M}$	External Moments
B	Linear Relation Matrix of Actuator Torques to Applied Torques
b	Spacecraft Body Reference Frame
B^+	Moore-Penrose Pseudo Inverse
B_c	Constrained Linear Relation Matrix
B_u	Unconstrained Linear Relation Matrix
C_{MS}	Motor Speed Constant
C_{MT}	Motor Torque Constant
D	Reaction Wheel Axial Principal MOI
E	Reaction Wheel Radial Principal MOI
I	MOI Tensor in the Body Frame
i	Inertial Reference Frame
I_{NL}	Motor No Load Current
I_{nom}	Nominal Motor Current (Max Continuous)
I_{xx}	X Axis Principal MOI
I_{yy}	Y Axis Principal MOI
I_{zz}	Z Axis Principal MOI
K_D	Differential Gain Matrix
K_P	Proportional Gain Matrix
n_{max}	Maximum Motor Speed

P_{nom}	Nominal Continuous Motor Power
Q	Quaternion Rate Transformation Matrix
R	Motor Terminal Resistance
r	Reaction Wheel Body Reference Frame
R^{mn}	Direction Cosine Matrix Between the m and n Reference Frame
R_{ij}^{mn}	Direction Cosine Between i and j Unit Vectors of the m and n Reference Frame
S	Linear Relation Matrix of Actuator Velocity/Acceleration to Angular Momentum/Torque
V_{nom}	Nominal Motor Voltage

CONTROL ALLOCATION METHODS FOR CONSTRAINED AND OVER ACTUATED SATELLITE ATTITUDE CONTROL SYSTEMS

I. Problem Statement

1.1 Introduction

The Air Force Research Laboratory (AFRL), Space Vehicle Directorate is developing a Reaction Wheel Array (RWA) for its Resilient Bus Experiment Laboratory (REBEL) satellite simulator. The RWA is an over actuated and constrained Attitude Control System (ACS). Such ACSs require some method of allocating commands from the high level controller to the actuators. The research presented in this thesis develops several control allocation methods and evaluates their performance in a numerical simulation based on the REBEL satellite simulator. The results of the simulation are analyzed in order to provide evidence of which control allocation methods may show promise for use on REBEL and potentially other over actuated and constrained ACSs.

This chapter discusses in general terms the purpose and need for control allocation, the motivations behind the research, spacecraft simulators, the modeling used in this research, and the objectives for the research.

1.2 The Control Allocation Problem

Control allocation is the mapping of some set of generalized command forces generated by a high level controller into low level actuator commands. When a control system contains more actuators than absolutely necessary for controllability, it is considered an over actuated control system. In unconstrained over actuated systems, the control allocation problem is an ill posed problem because there is generally not a unique solution. Furthermore, unconstrained actuators are not realistic. Control allocation with constrained actuators generally does not have a unique solution either, but the set of generalized control forces that can be exactly produced by the actuators becomes bounded.[1][5] The lack of a unique solution is an advantage rather than a disadvantage because it offers the opportunity to include secondary optimization objectives such as the minimization of control effort.[5] Control allocation can be accomplished within the design of the high level controller or separately as an independent control allocation algorithm.[7] There are three main goals of the control allocation algorithm:[5]

- Determine a unique allocation solution when multiple solutions exist by introducing some secondary performance objective.
- Ensure the solution is subject to the physical limitations, or constraints, of the actuators.
- Determine a "best" configuration of control allocation when no exact solution exists.

This research focuses on control allocation algorithms that are independent from the controller. By removing the control allocation from the high level controller, several benefits could potentially be gained. First, the high level controller can be developed with little knowledge of the actuators.[1] Second, the allocation algorithm can include real-time mitigation for actuator constraints, which cannot be easily accomplished when the control allocation is integrated into the high level controller.[5] Figure 1.1 shows the block diagram layout of a closed loop control system that includes an independent control allocation algorithm where $\vec{\tau}_{com}$ is the vector of generalized forces

[1] Section 2.5.1 provides an example allocation problem that helps clarify this concept.

commanded by the high level controller, $\bar{u}$ is the array of allocated generalized forces, and $\vec{\tau}$ is the vector of applied generalized forces applied to the mechanical system.

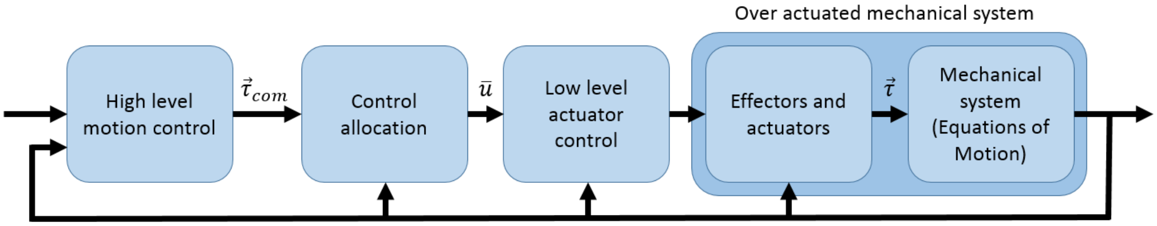
Over actuated mechanical system
High level motion control
$\vec{\tau}_{com}$
Control allocation
$\bar{u}$
Low level actuator control
Effectors and actuators
$\vec{\tau}$
Mechanical system (Equations of Motion)

Figure 1.1. Block diagram layout of closed loop control system with independent control allocation algorithm[1]

There are numerous reasons why the designer of a control system would choose an over actuated design. First, it may be desirable to choose a particular set of actuators rather than a smaller set of actuators for reasons such as but not limited to: cost, standardization, size, accuracy, dynamic response, flexibility, maintenance, and mechanical design.[8] Second, certain actuators can be shared among several control systems with different objectives, and therefore be redundant for the given motion control system.[9] Third, and most importantly, over actuation satisfies the need for actuator redundancy in order to meet fault tolerance and control reconfiguration requirements.[1] The overwhelming cost and complexity of repair missions for on-orbit space vehicles generally make them infeasible. For example, there have only been four recorded space missions to date with the primary goal of repairing on-orbit space vehicles.[10] On-orbit space vehicles that do not utilize redundant actuators suffer from increased risk of premature failure, as attitude control becomes problematic if control authority about one of the vehicle's axes is lost. For example, the Cassini spacecraft has been in operation for over 19 years. Cassini utilizes an over actuated RWA that consists of 4 Reaction Wheels (RWs). These RWs have accumulated tens of thousands of hours of operation. As a result, Cassini's third RW began to show evidence of bearing failure in the years leading up to the spacecraft's arrival at Saturn and was turned off. If the RWA had not been designed as an over actuated system with redundant actuators, controllability of Cassini could have been lost when the RW failed, causing premature mission termination.[11]

1.3 Motivation

AFRL is working in conjunction with the Air Force Institute of Technology (AFIT) to develop a RWA for the REBEL satellite simulator. The RWA design will be an over actuated control system consisting of a set of RWs subject to physical constraints. Currently, the design does not include a control allocation strategy that includes mitigation for RW constraints (torque constraints and saturation speed limitations.) Although the control allocation methods researched in this thesis are applicable to any application, they have been incorporated into an attitude control simulation based on the physical properties of REBEL for performance evaluation. Using REBEL's physical properties in the simulation not only provides realistic properties, but also provides some evidence of merit for the use of these techniques on the REBEL RWA design.

Two of the three control allocation methods studied in this research have been examined in spacecraft applications. The redistributed pseudo inverse with adaptive weighting method is a newly developed control allocation variant based on the redistributed pseudo inverse method. This newly developed allocation method attempts to prevent actuator saturation as a secondary objective. Little has been written on actuator saturation prevention as an objective of control allocation. To the author's knowledge, research on RW saturation speed prevention as an objective of control allocation has not been specifically published. This thesis provides additional insight into how these control allocation methods benefit RWA ACSs in satellite simulators (and by extension satellites themselves) that are subject to the saturation speed and torque constraints of the RWs.

1.4 Spacecraft Simulators

Spacecraft attitude control algorithms generally require physical testing prior to launch to verify performance. Computer simulations are typically used to predict and evaluate the performance of a spacecraft ACS, but physical testing is required for validation. Furthermore, there exists physical properties intrinsic to the spacecraft and its mechanisms that are not easily characterized in computer simulations such as structural flexing and fluid slosh.[12]

Spacecraft simulators[2] are a hardware solution that allow the ACSs to be physically tested before launch. A common type of spacecraft simulator approximates the dynamics experienced by a space vehicle by allowing a test platform complete with ACS to rotate about a spherical air bearing. The center of mass of the platform is controlled to be at the center of rotation of the air bearing to eliminate gravitational torques. The air bearing has such low friction that the test platform experiences nearly negligible rotational drag from the bearing. Disturbance torques not representative of on-orbit conditions are also present and should be considered in some regard. For instance, air-drag applies an external drag torque that opposes the rotation of the simulator. Experiments are typically carried out at low slew rates to mitigate against such drag torques. These test platforms have the ability to incorporate payloads to further improve the accuracy of their dynamics. Spacecraft simulators have been developed and used successfully by AFIT, the Naval Postgraduate School, NASA, Honeywell Corporation, and various universities.[13] Figure 1.2 depicts a CAD model representation of the REBEL satellite simulator, which is discussed in more detail in Section 2.2. As another example, Fig. 1.3 depicts Honeywell Corporation's Momentum Control System and Line of Sight (MCS/LOS) spacecraft simulator.

[2] The term "simulator" is commonly used within the field to refer these types of test platforms. A more accurate term would be "emulator", but it does not receive usage in literature.

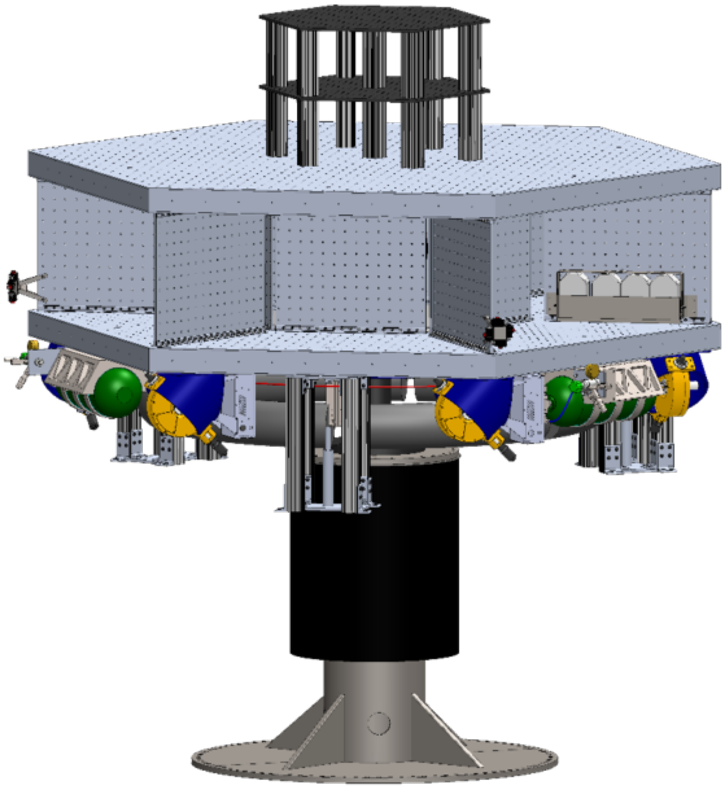

Figure 1.2. AFRL Spacecraft simulator (REBEL)[2]

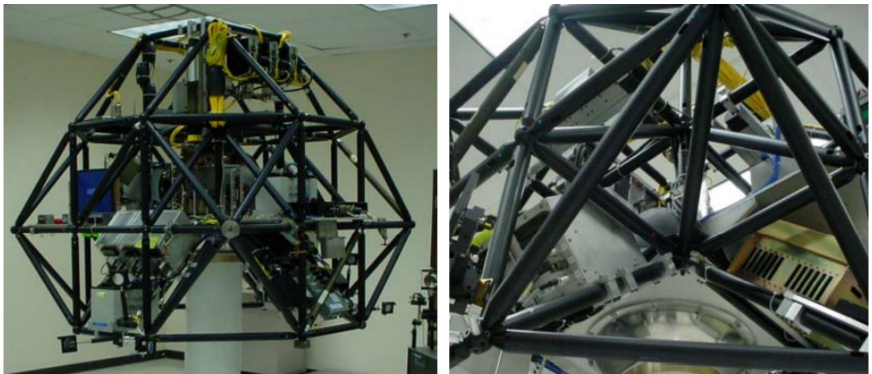

Figure 1.3. Two Views of Honeywell's 2003 MCS/LOS Testbed[3]

1.5 Modeling

Generally, the first step in testing spacecraft control algorithm performance is done with computer simulation. Computer simulation is typically more cost effective than physical testing and can identify issues with the control algorithm before physical testing is performed. A simulation of the non-linear dynamics of the REBEL satellite simulator were developed in previous research to determine the effectiveness of reaction wheel placement, motor selection, and momentum wheel design. This simulation is the foundation upon which this research has been performed. This simulation is later covered in detail in Section 3.3. Various control allocation algorithms and other performance parameter calculations have been added to the original simulation for the purpose of this research.

1.6 Objectives

This research evaluates the numerically simulated performance of various control allocation methods as applied to the REBEL satellite simulator while keeping all other aspects of the simulation identical (high level controller, disturbance torques or lack of, REBEL's physical parameters, etc.). The control allocation methods being examined attempt to improve the performance of the control system when it is subject to actuator constraints. The intention is to reduce the error between the commanded torque from the high level controller $\vec{\tau}_{com}$ and the applied torque to the system $\vec{\tau}$. This research does not focus on developing the high level controller itself or other aspects of the closed loop control system, so the absolute performance of the control system is not of concern. Rather, the relative performance of the control system while utilizing different control allocation methods is observed and compared to draw conclusions about the performance and viability of each approach.

The control allocation methods that are evaluated are the simple Moore-Penrose pseudo inverse (which serves as a baseline for comparison), the redistributed pseudo inverse, and the redistributed pseudo inverse with adaptive weighting. The methodology of these control allocation methods is described further in Section 3.8. A detailed review of the performance of these allocation methods is presented, and recommendations for their application in regards to REBEL and other constrained, over actuated RWA ACSs are provided.

1.7 Preview

Chapter II Provides some background information on the AFRL REBEL satellite simulator, reaction wheels, angular momentum based equations of motion, control allocation, and briefly reviews some relevant literature and research. Chapter III Describes the methodology used to perform this thesis research. In particular, specifics about the RWA being used in the simulation, the simulation itself, the simulated maneuver, the high level controller, the low level motor controller, and the different control allocation algorithms are presented. Chapter IV Presents the results of this thesis research. Chapter V Discusses the results, makes recommendations about the appropriate situations where the presented control allocation methods might be appropriately utilized, and discusses recommendations for future research.

II. Background

2.1 Introduction

This chapter presents background information relevant to the research performed in this thesis. First, information about the REBEL spacecraft simulator is detailed. Next, basic information about reaction wheels are presented. The momentum based equations of motion are derived for a satellite (or satellite simulator) equipped with reaction wheels. An example control allocation problem for a simple constrained and over actuated system is evaluated. A basic control allocation method is then outlined that will be heavily built upon in later chapters. Finally, a literature review of the research topic is discussed.

2.2 AFRL Spacecraft Simulator (REBEL)

The simulation used in this research uses physical properties based on the REBEL spacecraft simulator (depicted previously in Fig. 1.2.) REBEL was developed by AFRL Space Vehicle Directorate for use in testing satellite control algorithms. The REBEL test platform is approximately nine feet in diameter, can maneuver $\pm 30^o$ in pitch and roll as well as $\pm 360^o$ yaw, and utilizes a large air bearing.

REBEL also features its own suite of attitude determination and control components. For attitude determination, REBEL utilizes an on-board inertia measurement unit, a set of six accelerometers, and an external motion capture system. For attitude control, REBEL utilizes a set of twelve pneumatic thrusters, a set of six control moment gyroscopes, and a RWA. Both the control moment gyroscopes and RWA are still under development and have been a joint effort between AFRL and AFIT.

In order to isolate the test platform from external torques, all attitude determination and control systems are self-contained on REBEL. This self-containment requires

REBEL to be battery powered, carry pneumatic air tanks for the thrusters, and be controlled via Wi-Fi connection. REBEL also contains an automatic mass balancing system (still under development) that places the center of mass at the axis of rotation of the air bearing.

2.3 Reaction Wheels

Reaction wheels serve as a method of attitude control by functioning as momentum storage and exchange devices. They consist of an axisymmetric rotational mass affixed to the rotor of an electric motor. The motor is then affixed to the body of the space vehicle. The rate of change of momentum of the rotational mass is proportional to the torque produced by the motor. The mathematical representation of this momentum change, and its exchange with the space vehicle is further detailed Section 2.4.

Because reaction wheels are driven by electric motors, their ability to exchange momentum with the space vehicle is limited by the constraints of the motor. Since the rate of change of momentum of the rotational mass is proportional to the torque produced by the motor, it is constrained by the amount of torque that the motor can deliver. Furthermore, electric motors have a range of operational speeds which cannot be exceeded without risking damage to the motor (typically due to excessive heat generation.) If the motor reaches this maximum speed, it is typically prevented from accelerating further as a safety precaution (i.e. the torque available becomes zero.) This constraint is referred to as an upper saturation speed constraint. Electric motors also tend to have poor performance when operating at low speeds. This poor performance is a result of the motor controller which requires a speed sensor of some form for feedback. These speed sensors typically function poorly at low speeds (due to reduced encoder sampling rates) resulting in poor motor controller performance at low speeds. As a result, reaction wheel operational speeds typically range from some small non-zero value to the maximum motor speed. This small non-zero motor speed is referred to as the lower saturation speed constraint.

Reaction wheels are used in this study as the only means of attitude control. The decision to rely solely on RWs was made for several reasons. First, RWs suffer from several constraints that impede attitude control performance when they are reached. The control allocation methods being researched attempt to mitigate the effects of these constraints, so it is desirable to have them present. Second, because the direction of the angular momentum vector of each RW in the array is fixed relative to the body of the space vehicle, the direction in which it exchanges momentum is constant. The fixed direction of the angular momentum vectors simplifies several calculations, whereas other momentum exchange device (i.e. control moment gyroscopes) require more complex computations.

Figure 2.1 depicts one of AFIT's preliminary RW designs for REBEL. Detailed information on the specification and constraints of the motors selected for the reaction wheels used in this simulation are detailed further in Section 3.2

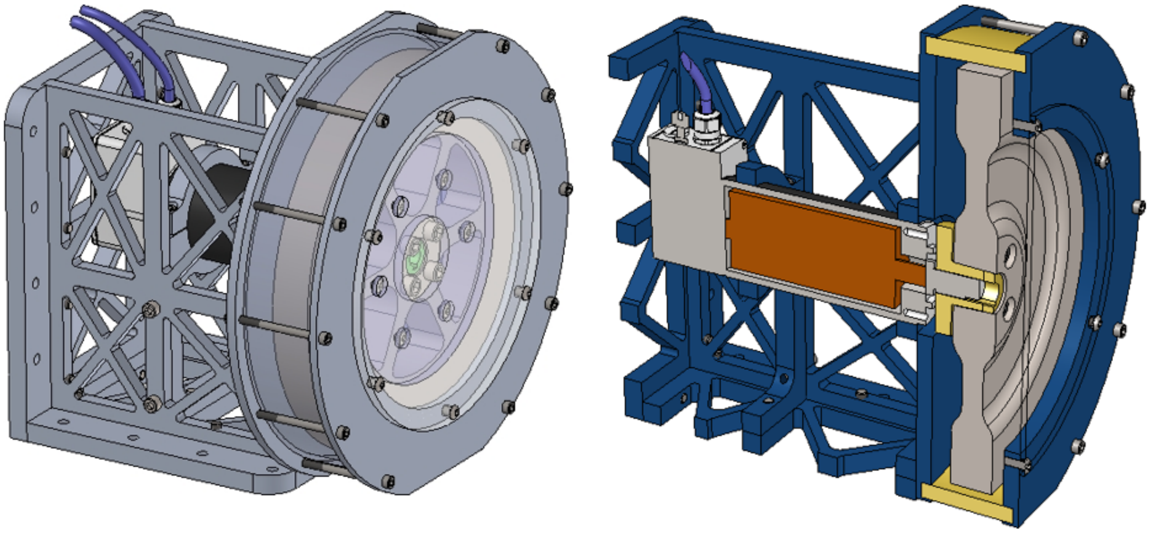

Figure 2.1. AFIT reaction wheel prototype for REBEL[4]

2.4 Angular Momentum Based Equations of Motion

2.4.1 Euler Equation for Satellite

A satellite's rotational Equation of Motion (EOM) can be derived utilizing the principal of the conservation of angular momentum. Using Newton's laws, it can be shown that the angular momentum of a rigid body written in the body frame is equivalent to the product of the Mass Moment of Inertia (MOI) matrix and the angular rate.[14]

$$\vec{H}^b = I\vec{\omega}^b \tag{2.1}$$

It can also be shown through mathematical manipulation that the external moments applied to a rigid body are equivalent to the rate of change of angular momentum.[14] It is important to note that both the rate of change of angular momentum and the external moments are written in the body frame, but the time derivative of the angular momentum must be inertial.

$$\frac{d^i}{dt}\vec{H}^b = \vec{M}^b \tag{2.2}$$

Substituting Eq. (2.1) into Eq. (2.2) yields Eq. (2.3), where the MOI and the angular rates are written in the body frame

$$\vec{M}^b = \frac{d^i}{dt}I\vec{\omega}^b. \tag{2.3}$$

In order to take the inertial derivative in the body reference frame, the transport theorem can be invoked to produce Eq. (2.4).[14] Here, all calculations are performed in the body frame, but the b subscripts have been omitted to simplify the expressions. Unless denoted otherwise, all parameters are considered to be expressed in the body reference frame.

$$\vec{M} = I\dot{\vec{\omega}} + \vec{\omega} \times I\vec{\omega} \tag{2.4}$$

where

$$\vec{M} = \begin{bmatrix} M_1 & M_2 & M_3 \end{bmatrix}^T \qquad \text{External Moments Acting on the Body,} \tag{2.5}$$

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \qquad \text{Mass Moment of Inertia Matrix}^1, \tag{2.6}$$

$$\vec{\omega} = \begin{bmatrix} \omega_1 & \omega_2 & \omega_3 \end{bmatrix}^T \qquad \text{Angular Rate,} \tag{2.7}$$

$$\dot{\vec{\omega}} = \begin{bmatrix} \dot{\omega}_1 & \dot{\omega}_2 & \dot{\omega}_3 \end{bmatrix}^T \qquad \text{and Angular Acceleration.} \tag{2.8}$$

Substituting Eq. (2.5), (2.6), (2.7), and (2.8) into Eq. (2.4) allows the equation to be written in matrix form as:

$$\begin{bmatrix} M_1 \\ M_2 \\ M_3 \end{bmatrix} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{\omega}_1 \\ \dot{\omega}_2 \\ \dot{\omega}_3 \end{bmatrix} + \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \tag{2.9}$$

Equation (2.9) is formally known as Euler's equation, and relates the externally applied torques $\vec{M}$ to the body frame angular velocities $\vec{\omega}$ and accelerations $\dot{\vec{\omega}}$ of a rigid body.[15] Though in actuality no physical body is truly rigid, modeling the flexibility of structures within the dynamics is not trivial and is considered outside of the scope of this research. Equation (2.9) is the foundation of the REBEL satellite simulation model.

[1] The body axis frame is typically placed on the principal axis, such that the products of inertia are zero. Though in actuality it is difficult to place the body frame such that the products of inertia are exactly zero, the assumption is made that the values are small enough to be considered negligible.

2.4.2 Inclusion of Reaction Wheel Dynamics

Because reaction wheels are used as the actuators in the control system of the REBEL simulation, it is important to understand their dynamics and how they are incorporated into the Euler Equation model used to represent REBEL's dynamics (Eq. (2.9).) Reaction wheels are axisymmetric rotors (i.e. momentum wheels) that are accelerated by a motor fixed to a rigid body. Their ability to provide attitude control can be easily understood by considering them as momentum storage devices internal to the rigid body they provide control to. Similar to the satellite's angular momentum, a reaction wheel's angular momentum can be expressed using Eq. (2.1).

Figure 2.2 depicts the reaction wheel's body frame r with respect to the body frame of the satellite b and the inertial frame i, as well as its angular momentum $\vec{H}^r_{RW}$. Note that absolute location and orientation of the body frame and the inertial frame are considered to be arbitrarily placed in this derivation. In the r frame, the RW's angular velocity is strictly in the $\hat{r}_3$ direction, and has a scalar magnitude of ψ. Its angular acceleration is also strictly in the $\hat{r}_3$ direction, and has scalar magnitude $\dot{\psi}$. The reaction wheel is affixed to the body of satellite such that the orientation of $\hat{r}_3$ is fixed with respect to the reference frame of the satellite b, and the orientation of $\hat{r}_1$ and $\hat{r}_2$ vary in time with respect to the satellite's body frame b at a rate equivalent to the rotation of the wheel about the $\hat{r}_3$ axis ψ.

Applying Eq. (2.1) and expanding it into matrix form yields

$$\bar{H}^r_{RW} = \begin{bmatrix} E & 0 & 0 \\ 0 & E & 0 \\ 0 & 0 & D \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \psi \end{bmatrix} = D\psi \left[\hat{r}_3\right]. \tag{2.10}$$

Due to the axisymmetric geometry of the rotor, the MOI has been simplified such that $I_{xx} = I_{yy} = E$; and for consistency, let $I_{zz} = D$.

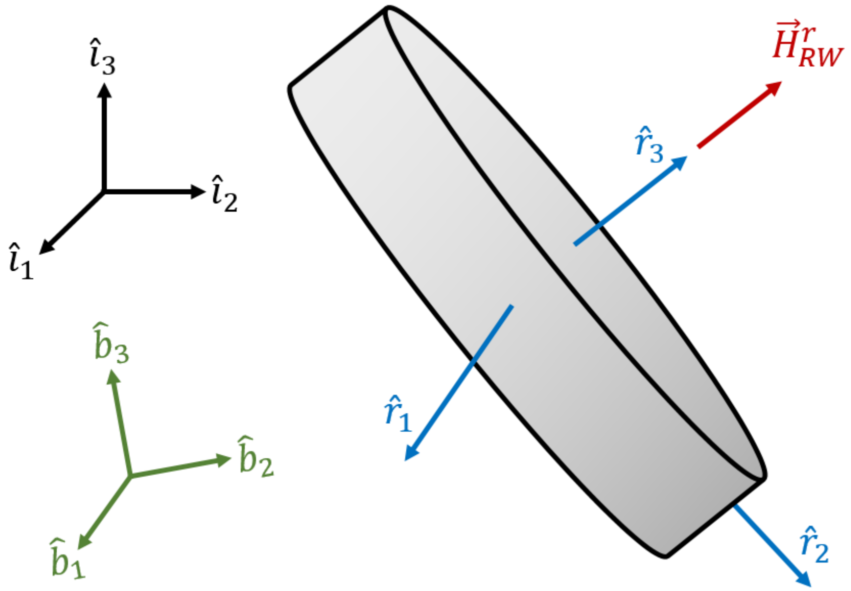

Figure 2.2. Reaction wheel body frame r

The angular momentum of the reaction wheel (Eq. (2.10)) and the angular momentum of the satellite (Eq. (2.1)) can be summed together to produce the total angular momentum of the combined system. However, before the angular momentum of the RW can be summed with the angular momentum of the satellite, it must be expressed in the same reference frame as the satellite. Since the direction of the RW's angular momentum is fixed with respect to the satellite's body reference frame b, a non-time-varying rotation matrix can be used to express the RW's angular momentum vector in the satellite's body reference frame.

$$\vec{H}^b_{RW} = R^{br}\vec{H}^r = \begin{bmatrix} R^{br}_{11} & R^{br}_{12} & R^{br}_{13} \\ R^{br}_{21} & R^{br}_{22} & R^{br}_{23} \\ R^{br}_{31} & R^{br}_{32} & R^{br}_{33} \end{bmatrix} D\psi\,[\hat{r}_3] = D\psi \begin{bmatrix} R^{br}_{13} \\ R^{br}_{23} \\ R^{br}_{33} \end{bmatrix} \tag{2.11}$$

Here, $R^{br}_{ij} = \cos\theta_{ij}$ and θ_{ij} is the angle between the $\hat{b}_i$ unit vector and the $\hat{r}_j$ unit vector.[14]

With the RW's angular momentum expressed in the satellite's body reference frame b, it can be summed with the satellite's angular momentum H^b_{sat} to produce the total angular momentum $\vec{H}_{total}$. This summation can be accomplished for any number n of RWs on the satellite by applying Eq. (2.11) to each RW, and summing the resulting angular momentums.[2][16]

$$\begin{aligned} \vec{H}^b_{total} &= \vec{H}^b_{sat} + \sum_{k=1}^{n} \vec{H}^b_{RW,k} \ , \\ \vec{H}^b_{total} &= \vec{H}^b_{sat} + \sum_{k=1}^{n} D_k \psi_k \begin{bmatrix} R^{br^k}_{13} \\ R^{br^k}_{23} \\ R^{br^k}_{33} \end{bmatrix} \ . \end{aligned} \tag{2.12}$$

The rate of change of total angular momentum can be determined by taking the inertial derivative of Eq. (2.12). The inertial derivative of the satellite's angular momentum $\vec{H}_{sat}$ has already been determined in Eq. (2.9), and the inertial derivative of the RW angular momentum term is determined using the transport theorem

$$\begin{aligned} \frac{d^i}{dt}\sum_{k=1}^{n} \vec{H}^b_{RW,k} = \sum_{k=1}^{n} \frac{d^i}{dt}\vec{H}^b_{RW,k} = \sum_{k=1}^{n}\left(\frac{d^b}{dt}\vec{H}^b_{RW,k} + \vec{\omega}\times\vec{H}^b_{RW,k}\right) , \\ \frac{d^i}{dt}\sum_{k=1}^{n} D_k\psi_k \begin{bmatrix} R^{br^k}_{13} \\ R^{br^k}_{23} \\ R^{br^k}_{33} \end{bmatrix} = \sum_{k=1}^{n}\left(D_k\dot{\psi}_k \begin{bmatrix} R^{br^k}_{13} \\ R^{br^k}_{23} \\ R^{br^k}_{33} \end{bmatrix}\right) + \vec{\omega}\times\sum_{k=1}^{n}\left(D_k\psi_k \begin{bmatrix} R^{br^k}_{13} \\ R^{br^k}_{23} \\ R^{br^k}_{33} \end{bmatrix}\right) . \end{aligned} \tag{2.13}$$

Equation (2.13) becomes clearer when the summations are expressed in matrix form. For example, allow the number of reaction wheels n to be 4, then

[2] The formulation for total angular momentum does not include the RW rotor mass rotating at the satellite's body rate $\vec{\omega}$. It is assumed that either the rotor mass is negligible compared to the satellite mass, or that the MOI of the satellite itself was computed to include all of the mass contribution of the rotors.

$$\frac{d^i}{dt}\sum_{k=1}^{4} D_k\psi_k \begin{bmatrix} R_{13}^{br^k} \\ R_{23}^{br^k} \\ R_{33}^{br^k} \end{bmatrix} = R \begin{bmatrix} D_1\dot{\psi}_1 \\ D_2\dot{\psi}_2 \\ D_3\dot{\psi}_3 \\ D_4\dot{\psi}_4 \end{bmatrix} + \vec{\omega} \times R \begin{bmatrix} D_1\psi_1 \\ D_2\psi_2 \\ D_3\psi_3 \\ D_4\psi_4 \end{bmatrix} \tag{2.14}$$

where

$$R = \begin{bmatrix} R_{13}^{br^1} & R_{13}^{br^2} & R_{13}^{br^3} & R_{13}^{br^4} \\ R_{23}^{br^1} & R_{23}^{br^2} & R_{23}^{br^3} & R_{23}^{br^4} \\ R_{33}^{br^1} & R_{33}^{br^2} & R_{33}^{br^3} & R_{33}^{br^4} \end{bmatrix} \tag{2.15}$$

is a transformation matrix that expresses the array of RW axial angular velocities $\bar{\psi}$ and angular accelerations $\dot{\bar{\psi}}$ in the satellite body reference frame.

The inertial derivative of Eq. (2.12) can then be written in matrix form for the case of 4 reaction wheels by utilizing Eq. (2.9) and (2.14)

$$\begin{bmatrix} M_1 \\ M_2 \\ M_3 \end{bmatrix} = \begin{bmatrix} I_{xx}\dot{\omega}_1 \\ I_{yy}\dot{\omega}_2 \\ I_{zz}\dot{\omega}_3 \end{bmatrix} + R \begin{bmatrix} D_1\dot{\psi}_1 \\ D_2\dot{\psi}_2 \\ D_3\dot{\psi}_3 \\ D_4\dot{\psi}_4 \end{bmatrix} + \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \times \left(\begin{bmatrix} I_{xx}\omega_1 \\ I_{yy}\omega_2 \\ I_{zz}\omega_3 \end{bmatrix} + R \begin{bmatrix} D_1\psi_1 \\ D_2\psi_2 \\ D_3\psi_3 \\ D_4\psi_4 \end{bmatrix} \right) \tag{2.16}$$

where the inertial derivative of total angular momentum is equivalent to the external moments applied to the system $\vec{M}$, and the products of $I\vec{\omega}$ and $I\dot{\vec{\omega}}$ have been condensed to single vectors to conserve space.[3]

Equation (2.16) can also be written as

$$\vec{M} = I\dot{\vec{\omega}} + S\dot{\bar{\psi}} + \vec{\omega} \times \left(I\vec{\omega} + S\bar{\psi} \right) \tag{2.17}$$

[3] This simplification is only applicable for the case where the products of inertia are zero.

where

$$S = R \begin{bmatrix} D_1 & 0 & 0 & 0 \\ 0 & D_2 & 0 & 0 \\ 0 & 0 & D_3 & 0 \\ 0 & 0 & 0 & D_4 \end{bmatrix} . \tag{2.18}$$

Equation (2.16) and (2.17) clearly displays how reaction wheels can influence the angular momentum of a satellite. They represent a set of differential equations, where the external moments $\vec{M}$ (if known) and the angular accelerations $\dot{\vec{\psi}}$ of the reaction wheels can be considered inputs. Note that $\vec{M}$ is usually considered as a disturbance input if it is present in the model. The non-linear set of differential equations can be coupled with a corresponding set of kinematic equations and numerically solved to predict the satellite's rotational motion. The numerical solution process is covered in more detail in Section 3.3, where the details of the simulation used in this research are presented.

2.5 Control Allocation

2.5.1 Example Control Allocation

Control allocation is a concept to which sophisticated solutions can be applied. Fundamentally, control allocation is the practice of distributing some number of commanded control values m across some number of actuators p. The problem becomes of interest when p is greater than m, which leads to infinitely many ways to distribute the m control values across the p actuators. The complexity of the solutions to the control allocation problem can sometimes make the simplicity of the problem's fundamentals unclear. Oppenheimer[5] presents a simple control allocation problem that consists of a scalar command value (m is 1) that is allocated to two control variables

(p is 2.) In Oppenheimer's example, he established a graphical solution to a control allocation problem that elegantly depicts the concept of control allocation.

Let $\bar{u} \in \mathbb{R}^2$ represent an array of generic control forces produced by two actuators such that $\bar{u}$ is $[\delta_1 \;\; \delta_2]^T$. Let $\tau_{com} \in \mathbb{R}^1$ be a scalar value that represents a desired generic control force applied to a system. Let $\bar{u}$ be linearly related to τ_{com} by the matrix B, where

$$\begin{aligned} B &= [3 \;\; 1] \\ B\bar{u} &= \tau_{com} \\ 3\delta_1 + \delta_2 &= \tau_{com}. \end{aligned} \tag{2.19}$$

Equation (2.19) represents a basic algebraic set of equations for which there are more unknowns than there are equations. The result is infinitely many solutions for the unknown control vector $\bar{u}$ given some scalar τ_{com}. Furthermore, the elements of $\bar{u}$ are typically subject to physical constraints (e.g. motor torque limits) which places bounds on the possible solution space. The constrained solution can be described as a subset of the unconstrained solution $\bar{u} \in \mathbb{U}^2 \subset \mathbb{R}^2$. Equation (2.20) reformulates Eq. (2.19) as a constrained control allocation problem.

$$\begin{aligned} &\text{Find } \delta_1,\ \delta_2 \text{ such that } \tau_{com} = 3\delta_1 + \delta_2 \\ &\text{subject to } -1 \leq \delta_1 \leq 1,\ -1 \leq \delta_2 \leq 1 \end{aligned} \tag{2.20}$$

Figure 2.3 depicts a graphical solution to Eq. (2.20) for various values of τ_{com}. The dimensional simplicity of this allocation problem lends itself well to the two dimensional plot format. The diagonal lines represent all possible solutions of $\bar{u} \in \mathbb{R}^2$ for τ_{com} values of 2, 3, 4, and 5. The box around the origin of the plot represents the constraint boundaries of the elements of $\bar{u} \in \mathbb{U}^2 \subset \mathbb{R}^2$. An achievable solution must lie within the constraint "box".

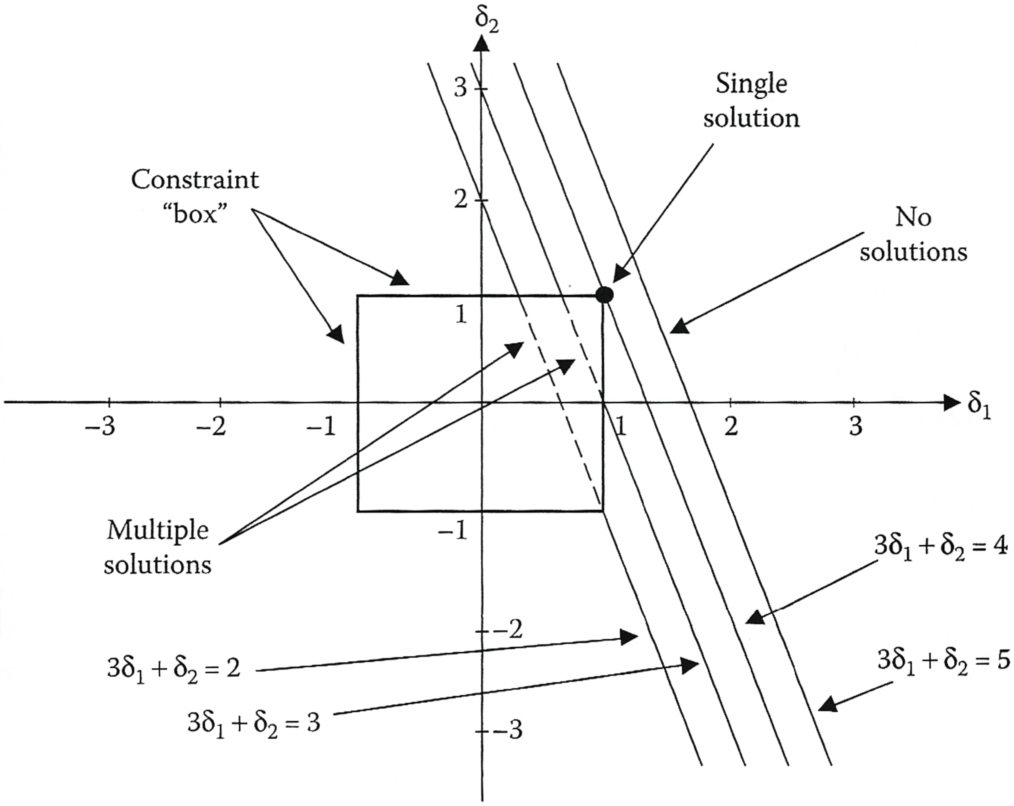

Figure 2.3. One dimensional control allocation problem with two independent control variables [5]

If τ_{com} is 2 or 3, there are multiple solutions for $\bar{u}$ that exist within the constrained solution space $\mathbb{U}^2$. If τ_{com} is 4, there is only one solution that lies on the boundary of the constrained solution space $\mathbb{U}^2$. If τ_{com} is 5, there are no solutions that satisfy $\bar{u} \in \mathbb{U}^2$. The challenging part of the control allocation problem is determining an optimal solution of $\bar{u} \in \mathbb{U}$ when multiple solutions exist, or a solution of $\bar{u} \in \mathbb{U}$ that minimizes the error between some achievable general force τ and the commanded general force τ_{com} when no exact solution exists.

2.5.2 Pseudo-Inverse

When dealing with control allocation in an over actuated system, a common approach for distributing control authority to the actuators is the use of the Moore-Penrose pseudo-inverse.[1] Moore-Penrose pseudo-inverse approach is both simple and produces a unique solution; however, it does not take actuator constraint into consideration. The pseudo-inverse allows for the inversion of the non-square B matrix in Eq. (2.19) so that the allocated generalized forces $\bar{u}$ can be solved directly. The Moore-Penrose pseudo-inverse is formed from

$$B^{+} = B^{T}\left(BB^{T}\right)^{-1}. \tag{2.21}$$

Equation (2.21) is in fact a simplification of the solution to the minimization of the quadratic cost function that measures the cost of control action

$$\min_{\bar{u}\in\mathbb{R}^{p}} \frac{1}{2}\left(\bar{u}-\bar{u}_{p}\right)^{T} W\left(\bar{u}-\bar{u}_{p}\right) \quad \text{subject to } \vec{\tau}_{com} = B\bar{u}, \tag{2.22}$$

where $W \in \mathbb{R}^{p\times p}$ is a positive definite weighting matrix that weights actuator use, and $\bar{u}_p$ is an array of the preferred values of $\bar{u}$.[1] A general solution for $\bar{u}$ can be found using Lagrange multipliers and the optimality conditions of Eq. (2.22).[17]

$$\begin{aligned} \bar{u} &= \left(I - CB\right)\bar{u}_{p} + C\vec{\tau}_{com} \\ C &= W^{-1}B^{T}\left(BW^{-1}B^{T}\right)^{-1} \end{aligned} \tag{2.23}$$

Equation (2.23) provides a generic pseudo-inverse that solves the unconstrained weighted control allocation problem by minimizing each actuator's deviation from some preferred value. For the special case where W is identity and $\bar{u}_p$ is an array of zeros, Eq. (2.23) can be further simplified to Eq. (2.24), where C is the Moore-

Penrose pseudo-inverse B^+ of Eq. (2.21).

$$\begin{aligned} \bar{u} &= C\vec{\tau}_{com} \\ C &= B^+ = B^T \left(BB^T\right)^{-1} \end{aligned} \tag{2.24}$$

The pseudo-inverse technique is an excellent way to formulate a control allocation solution when actuator constraints are not of importance.[1] It introduces a secondary objective, the minimization of control effort, in order to select a single solution when infinitely many are possible. Since physical systems have constraints by nature, more sophisticated methods of control allocation are typically needed. The unconstrained pseudo-inverse is however heavily relied upon in many constrained control allocation solutions. The formulation of the Moore-Penrose pseudo inverse is presented as background information that is drawn upon when the methodology of more sophisticated allocation strategies are discussed in Section 3.8.

2.6 Literature Review

Johansen et al.[1] and Oppenheimer et al.[18] surveyed a variety of independent control allocation strategies. The research summarized the allocation methodologies, benefits, drawbacks, and applications. These surveys served as an invaluable resource for the research performed in this thesis.

Research has shown that the redistributed pseudo inverse allocation method is a viable method for allocating control torques in over actuated systems. Bodson[19] showed through simulation of various over actuated aircraft subject to actuator constraints that the redistributed pseudo inverse control allocation method was a simple yet viable method of distributing control torques. It compared well against other allocation methods at reducing average allocation error $||\tau_{com} - \tau||_2$, average maneuver time, and average control $||u||_2$ (a secondary performance objective.)[19]

In his research, Bodson[19] identified an error in the redistributed pseudo inverse allocation method where non-exact solutions could be produced when exact solutions were attainable. Bordignon et al.[20] later published on the control allocation methodology used for the X-35B aircraft. The publication outlines a modified redistributed pseudo inverse method that was successfully implemented in the controls of the X-35B that mitigates the issue outlined by Bodson. In a subsequent publication, Jin[21] evaluated the performance of this altered method against the original method and produced evidence that it offered improved performance. This altered method is what has been used in this research and is discussed further in Section 3.8.2.

Arun Kishore[22] demonstrated that adaptive weighting matrices can be used with the minimization of the cost function in Eq. (2.22) to provide an allocation method that prevents actuators in an over actuated control system from reaching saturation whenever possible. Numerical results were presented from an aircraft simulation where the actuators were control flaps, but the methodology is applicable to any set of actuators subject to saturation limits. Although the research did not include an iterative method of redistributing unconstrained actuators to compensate for the allocation error induced by the saturation of some subset of actuators, these concepts can be applied nonetheless. The implementation adaptive weighting variation is detailed further in Section 3.8.3.

2.7 Summary

This section has provided important background information that will help the reader better understand the topics discussed in the research methodology. Information about the REBEL spacecraft simulator has been detailed because the simulation used in the research is based on its physical properties. Basic information about reaction wheels have been presented since they are the sole actuators used in the simulation. The momentum based equations of motion have been derived for a satellite (or satellite simulator) equipped with reaction wheels have been derived so the reader will not take these equations for granted when they are presented in the details of the numerical simulation. An example control allocation problem has been outlined to help clarify the concept of control allocation. A basic control allocation method has been outline that are heavily built on when more sophisticated control allocation strategies are discussed. Finally, a literature review has been presented which relates where the topics of this research originated and how others in the field have utilized them.

III. Methodology

3.1 Chapter Overview

This chapter outlines in detail the research methodology used to evaluate several control allocation methods. The topics discussed in this chapter provide technical detail regarding the numerical simulation and control allocation algorithms evaluated in this research. Information regarding the RWA layout, the MOI values for the RW rotor, and the motors used for the RWs is provided. Details about the formulation of simulation, its numerical solution process, its important parameters, and its initial conditions are also treated. The simulated maneuver is discussed, as well as the high level controller that provides command values to the control allocation algorithm. Finally, the control allocation methods that are evaluated are discussed in detail. These methods include the Simple Pseudo Inverse (SPI) method, the Redistributed Pseudo Inverse (RPI) method, and the Redistributed Pseudo Inverse with Adaptive Weighting (RPIW) method.

3.2 Reaction Wheels

Reaction wheels are the exclusive means of attitude control used in this simulation. AFIT is working with AFRL Space Vehicle Directorate to design and implement a RWA for the REBEL test bed, but a finalized design has not been reached yet. The selection of the RWA parameters was made with consideration given to the design work that has already been performed, but is not necessarily reflective of the final design.

3.2.1 Reaction Wheel Array Layout

The RWA layout used in this simulation was one produced by preliminary unpublished AFIT research for the REBEL RWA deign. This design does not necessarily reflect the finalized layout of the REBEL RWA, or any optimized design in particular; however, it was shown to be viable. Figure 3.1 depicts the RWA layout that is used in the simulation performed for this research.

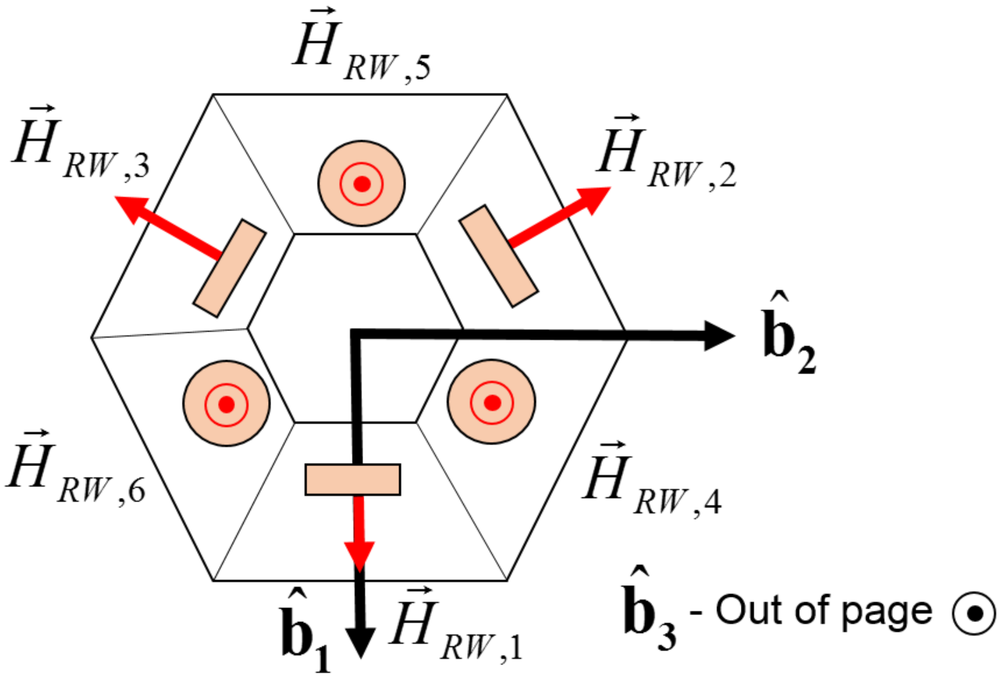

Figure 3.1. Diagram of reaction wheel array configuration used in this research

In this RWA configuration, 6 RWs are used. 3 RWs with their axial direction purely in the $\hat{b}_1$ - $\hat{b}_2$ are spaced equiradially about the about the $\hat{b}_3$ axis. The placement of the first wheel has its axial direction aligned with the $\hat{b}_1$ axis. The angular momentum of these wheels are depicted as $\vec{H}_{RW,i}$ where i is 1, 2, and 3. A second set of 3 wheels are oriented so their axial directions are aligned with the $\hat{b}_3$ axis. The angular momentum of these wheels are depicted as $\vec{H}_{RW,i}$ where i is 4, 5, and 6.

$\vec{H}_{RW,4}$, where each of the 3 angular momentum components are equal in magnitude and direction. This configuration allows the RWA to have angular momentum components in all 3 axes of the REBEL body reference frame b.

The formulation of this RWA design was the result of several design considerations. It was desired that all of the RWs be of identical design. That is, the MOI and motor for each wheel need be the same. These design requirements presented some issue though since the $\hat{b}_3$ principal component of REBEL's MOI is significantly larger the that of the $\hat{b}_1$ and $\hat{b}_2$ components. The larger MOI value of the $\hat{b}_3$ axis corresponds to a need for significantly more momentum storage capacity in the $\hat{b}_3$ direction in order to meet REBEL angular velocity/RW saturation requirements. In lieu of using a single RW with a larger momentum wheel to produce angular momentum in the $\hat{b}_3$ direction, it was determined that using 3 RWs identical to the RWs placed on the $\hat{b}_1$ - $\hat{b}_2$ plane would produce favorable results while meeting the previously mentioned design considerations.

In this configuration, the set of RWs aligned with the $\hat{b}_3$ axis can be considered as a single RW that has 3 times the angular momentum. That is,

$$\vec{H}_{RW,4'} = \vec{H}_{RW,4} + \vec{H}_{RW,5} + \vec{H}_{RW,6} \tag{3.1}$$

where $\vec{H}_{RW,4}$, $\vec{H}_{RW,5}$, and $\vec{H}_{RW,6}$ are equivalent. The simplification of these 3 RWs of course assumes that each wheel is operating at identical speeds for all time. In reality, minor differences in RW drag, motor efficiencies, etc. may cause the wheels to operate at increasingly different speeds over time, even if identical motor torque commands are issued. Since the simulation in this research does not model these small differences between the motors, the assumption of identical RW speeds does not alter the results. It is important to keep this consideration in mind however.

With the RW combination assumption made, the set of 6 reaction wheels can be thought of as a set of 4, where the MOI of the 4th wheel is 3 times larger than wheels 1, 2, and 3. The result is a simplified derivation for the angular momentum derivation performed in Section 2.4.2 where only 4 RWs need to be included. Namely, the transformation matrix presented in Eq. (2.15) that expresses the array of RW axial angular velocities $\bar{\psi}$ and angular accelerations $\dot{\bar{\psi}}$ in the satellite body reference frame can be determined as

$$R = \begin{bmatrix} \cos 0^o & \cos 120^o & \cos 240^o & \cos 90^o \\ \cos 90^o & \cos 30^o & \cos 150^o & \cos 90^o \\ \cos 90^o & \cos 90^o & \cos 90^o & \cos 0^o \end{bmatrix}. \tag{3.2}$$

3.2.2 Reaction Wheel Mass Moment of Inertia

If the identicality of each momentum wheel is taken into account along with the simplification of the 4th wheel being a summation of 3 identical wheels, the axial principal MOI of each wheel D_i becomes D for i is 1, 2, 3 and $3D$ for i is 4. In this simulation, the axial principal MOI for the wheels D was chosen to be 0.065 kg-m^2. This selection was made in order to deliberately under size the MOI of the wheels. This decision was made to help ensure that the reaction wheels would encounter saturation speed constraints in the simulation. This consideration is of course partially depended on the motor's upper saturation limit, which is detailed further in the next section. As previously stated, the control allocation methods being researched are meant to provide mitigation to actuator constraint, so taking steps to ensure that saturation speeds were reached was considered important.

With the principal MOI values D determined, the angular momentum transformation matrix S in Eq. (2.18) can be evaluated as

$$S = 0.065 \begin{bmatrix} \cos 0^o & \cos 120^o & \cos 240^o & 3\cos 90^o \\ \cos 90^o & \cos 30^o & \cos 150^o & 3\cos 90^o \\ \cos 90^o & \cos 90^o & \cos 90^o & 3\cos 0^o \end{bmatrix} \text{kg-m}^2 \tag{3.3}$$

or,

$$S = 0.065 \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} & 0 \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix} \text{kg-m}^2. \tag{3.4}$$

Note that the 4th column of the transformation matrix s has been multiplied by 3 to account for the 3 identical RWs assumed to be operating at identical speeds.

3.2.3 Reaction Wheel Motor

To understand the actuator constraints that the control allocation algorithm must mitigate, it is important to understand the performance parameters and limitations of the RW motors. The motor for each of the RWs are identical per the desired design criteria of the REBEL RWA. Note that the motor used in this simulation does not necessarily reflect the motor that will be used in the final design of the REBEL RWA.

The motor parameters used for the simulation reflect those of the Maxon EC 60∅ mm, Brushless, 400 Watt Motor (P/N: 167132).[6] Some important characteristics of this motor are listed in Table 3.1. Of particular note is the maximum operating speed n_{max} which is 7000 rpm. One of the reasons that this motor was selected was for its maximum speed, which is relatively low compared to other available motors which would suite this application. By limiting the maximum operating speed, RW saturation is reached more easily. Since speed saturation is a constraint that the control allocation algorithms presented in this research have been designed to mitigate against, a lower maximum operating speed n_{max} was desirable.

Table 3.1. Operational Parameters of the Maxon EC 60∅ mm, Brushless, 400 Watt Motor (P/N: 167132) [6]

Symbol	Parameter	Value	Unit
C_{MT}	Motor Torque Constant	0.085	Nm/A
C_{MS}	Motor Speed Constant	113	rpm/A
n_{max}	Motor Max Speed	7000	rpm
τ_{nom}	Nominal Torque (Max Continuous)	0.768	Nm
I_{nom}	Nominal Current (Max Continuous)	9.56	A
V_{nom}	Nominal Voltage	48	V
P_{nom}	Nominal Continuous Motor Power	400	W
I_{NL}	No Load Current	0.67	A
R	Terminal Resistance	0.345	Ω

Another important consideration of the motor is the maximum continuous operating torque that it can produce. This torque maximum is a function of motor speed, and Fig. 3.2 depicts continuous torque operating range as a function of motor speed n. Figure 3.2 can be used to approximate a linear function that approximates the boundary of the continuous torque operating range, which represents the maximum continuous operating torque of the motor as a function of motor speed. This linear approximation was formed as a line connecting the maximum continuous operating torque at 0 rpm and the maximum continuous operating torque at the maximum motor speed of 7000 rpm. This approximation was made in the absence of an equation that defined the boundary of the continuous operation range shown in Fig. 3.2.

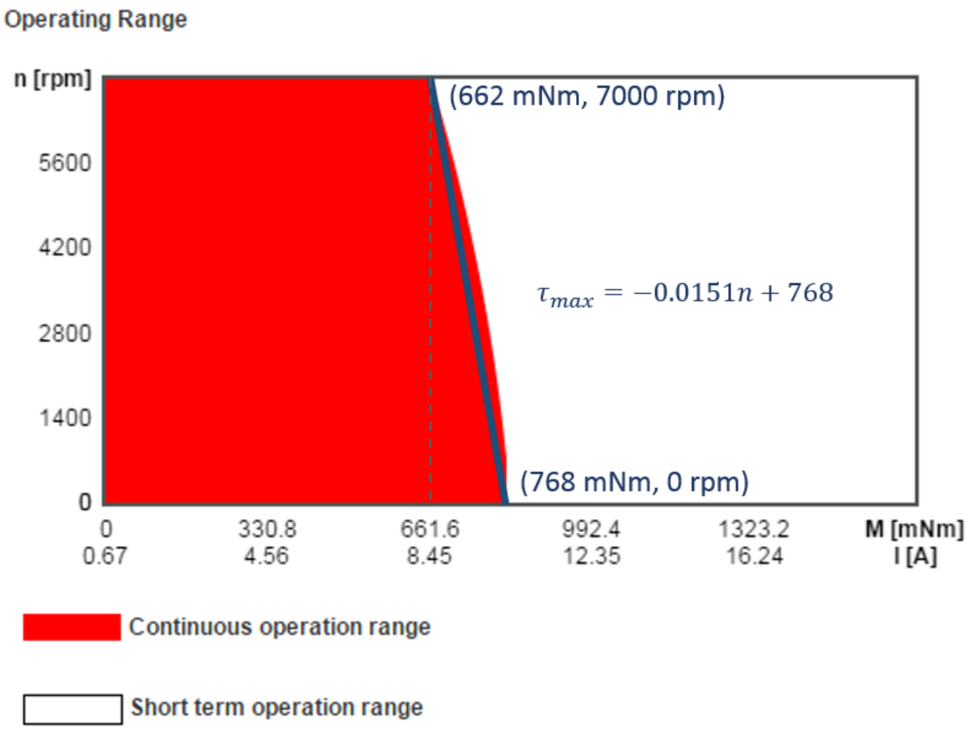

Figure 3.2. Maximum continuous operating torque as a function of motor speed linearly approximated[6]

As Fig. 3.2 shows, the maximum continuous torque available from the motor τ_{max} decreases with motor speed. An interview with an AFIT research assistant experienced with Maxon motors confirmed that the maximum continuous torque of the motors appears to be approximately bi-directional.[23] That is, τ_{max} is equivalent to $-\tau_{max}$ over the operating speed range. The maximum continuous torque can be defined as

$$\tau_{max}[\text{mNm}] = -0.0151n + 768, \quad \text{for}: \ 0 \leq n \leq 7000 \text{ rpm} \tag{3.5}$$

where the motor speed n is in rpm. Although this approximation produces slightly smaller maximum torque values than Fig. 3.2 shows towards the middle of the operating speed range, it never approximates the torque as too high. Eq. (3.5) is used in the simulation to determine the maximum torque that each motor can produce based on its speed. This torque maximum is applied as a torque constraint, where if the torque command allocated to the RW is higher than the maximum continuous operating torque τ_{max}, it is truncated to that value and not permitted to operate in the short term operation range.

It should also be noted that other than the speed dependent torque limits the RW motors are modeled as ideal for the purpose of this simulation. That is, friction and time delay effects are assumed to be non-existent in the motors. In reality, these effects play an important role on the absolute performance of the system. Since the relative performance between the allocation methods is the only concern, these effects have been neglected. Future research should include the characterization and implementation of these motor limitations.

3.3 Model and Simulation

3.3.1 Equation of Motion and Numerical Solution

The attitude of REBEL is tracked throughout the simulation with quaternions. The decision to use quaternions was made to prevent the singularities that are present in other forms of attitude tracking methods from causing simulation error. The use of quaternions is a common approach, and sufficient literature exists on the use of quaternions so it is not be expressly derived here. The quaternion for the angular orientation is expressed as

$$
\begin{aligned}
q_4 &= \pm\frac{1}{2}\sqrt{1 + \mathrm{trace}\left(R^{ib}\right)} \\
\begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} &= \frac{1}{4q_4}\begin{bmatrix} R^{ib}_{23} - R^{ib}_{32} \\ R^{ib}_{31} - R^{ib}_{13} \\ R^{ib}_{12} - R^{ib}_{21} \end{bmatrix} \\
\bar{q} &= \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix}^T
\end{aligned}
\tag{3.6}
$$

where R^{ib} is the transformation matrix of the body reference frame to the inertial reference frame, and R_{ij} is an element of this matrix in the i^{th} row and j^{th} column. The quaternion rates are then derived as a function of the body reference frame angular velocity.

$$
\begin{aligned}
\dot{\bar{q}} &= Q\vec{\omega} \\
Q &= \frac{1}{2}\begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix}
\end{aligned}
\tag{3.7}
$$

Equation (3.7) form the kinematics of the numerical solution. Equation (2.17) (the momentum based EOM for the spacecraft and 4 reaction wheels) can be rearranged to be equated to the angular acceleration of the spacecraft (Eq. (3.8).) This equation forms the dynamics of the numerical simulation.

$$
\dot{\vec{\omega}} = -I^{-1}\vec{\omega} \times I\vec{\omega} - I^{-1}\vec{\omega} \times S\bar{\psi} - I^{-1}S\dot{\bar{\psi}} + I^{-1}\vec{M}
\tag{3.8}
$$

If the external torque $\bar{M}$ is known and the reaction wheel accelerations are considered a controlled input, Eq. (3.7) and (3.8) can be formed into a set of nonlinear differential equations represented in matrix form

$$\dot{\bar{x}} = A\bar{x} + L\vec{M} + G\dot{\bar{\psi}} \tag{3.9}$$

where

$$\dot{\bar{x}} = \begin{bmatrix} \dot{q}_1 & \dot{q}_2 & \dot{q}_3 & \dot{q}_4 & \dot{\omega}_1 & \dot{\omega}_2 & \dot{\omega}_3 & \dot{\psi}_1 & \dot{\psi}_2 & \dot{\psi}_3 & \dot{\psi}_4 \end{bmatrix}^T \tag{3.10}$$

$$\bar{x} = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 & \omega_1 & \omega_2 & \omega_3 & \psi_1 & \psi_2 & \psi_3 & \psi_4 \end{bmatrix}^T \tag{3.11}$$

$$A = \begin{bmatrix} [0]_{4\times4} & Q & [0]_{4\times4} \\ [0]_{3\times4} & -I^{-1}\vec{\omega} \times I & -I^{-1}\vec{\omega} \times S \\ [0]_{4\times4} & [0]_{4\times3} & [0]_{4\times4} \end{bmatrix} \tag{3.12}$$

$$L = \begin{bmatrix} [0]_{4\times3} \\ I^{-1} \\ [0]_{4\times3} \end{bmatrix} \tag{3.13}$$

$$G = \begin{bmatrix} [0]_{7\times4} \\ [1]_{4\times4} \end{bmatrix} \tag{3.14}$$

and $[0]_{m\times n}$ denotes an $m \times n$ matrix of zeros and $[1]_{m\times n}$ represents an $m \times n$ identity matrix. This matrix formulation represents the non linear equations of motion derived previously. The state matrix $\bar{x}$ is comprised of 11 state variables which represent the spacecraft quaternion, angular rates, and RW angular rates. Here, A contains time varying values, while L and G are constant.

It should be noted that the reaction wheel acceleration array $\dot{\bar{\psi}}$ in Eq. (3.9) is considered as the input in the state equation. These values are considered to be defined by the control allocation algorithm based on the torque commands τ_{com}

produced by the high level controller. These values can also be equated to motor torques τ_i as the product of the RW's axial principal MOI D_i and the motor's angular acceleration $\dot{\psi}_i$. For the purpose of this research, it is assumed that there is no delay or error between the commanded motor torques and the achieved motor torques.

Because Eq. (3.9) is nonlinear, the linear tools typically used to solve such state equations cannot be utilized. Instead, numerical ordinary differential equation solvers can be used to solve for the states (Eq. (3.11)) across some arbitrary time span. In this research, MATLAB's ODE45 function was used as a solver for the non-linear EOM presented in Eq. (3.9), although other numerical methods can suffice.

3.3.2 Simulation Parameters

ODE45 integrates a system of differential equations across a designated time step Δt given initial conditions $\bar{x}_0$. ODE45 uses variable time stepping within the time interval Δt. It provided as an output the time and state array $\bar{x}$ that satisfy the set of differential equations at the end of the time step Δt.[24]

The time step Δt was set equal to the update interval of the high level controller. That is, the inputs $\dot{\bar{\psi}}$ are a constant value over the time step Δt that ODE45 solves across. The time step was set as 0.05 s, or 20 Hz. This time step value was chosen as it is a typical on-board controller update rate for such a high level controller. This update rate can be varied, and would impact the performance of the closed loop system. With a faster update rate, the inputs to the system of differential equations would be updated with the controller's commands more frequently. If the high level controller were designed well, it would correlate to better system response. If the time steps grow too large, it would introduce what is essential a time delay between the controller and the RW actuators. This time delay would degrade performance and potentially introduce instability. If the absolute performance of the closed loop system were of importance, the effects of this update rate could be researched in more detail. Since the allocation algorithm is of sole consideration in this research, and the relative performance between the closed loop systems is the only metric of concern, effort was not put into characterizing the effects of update rates as they grew too large. Modeling the impact of controller update rate is left for future research.

3.3.3 Initial Conditions

The initial conditions for the state matrix were selected as

$$\bar{x}_0 = \begin{bmatrix} q_{1,t_0} & q_{2,t_0} & q_{3,t_0} & q_{4,t_0} & \omega_{1,t_0} & \omega_{2,t_0} & \omega_{3,t_0} & \psi_{1,t_0} & \psi_{2,t_0} & \psi_{3,t_0} & \psi_{4,t_0} \end{bmatrix}^T$$

$$\bar{x}_0 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0\text{rpm} & 0\text{rpm} & 0\text{rpm} & 3500\text{rpm} & 3500\text{rpm} & 3500\text{rpm} & 3500\text{rpm} \end{bmatrix}^T \tag{3.15}$$

where the initial quaternion is

$$\bar{q}_0 = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T, \tag{3.16}$$

the initial angular velocity of the REBEL simulation is

$$\vec{\omega}_0 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \text{rpm}, \tag{3.17}$$

and the initial speed of the RWA is

$$\bar{\psi}_0 = \begin{bmatrix} 3500 & 3500 & 3500 & 3500 \end{bmatrix}^T \text{rpm}. \tag{3.18}$$

The initial quaternion selection was made for convenience, and bears no impact on the overall results of the simulation. The initial angular velocity was chosen for simplicity. Non-zero values could be used if desired, but in this case non-zero initial angular velocities would contribute no additional value to the results of the research being conducted. The RW initial angular velocities were set as 3500 rpm, approximately half way between their lower and upper saturation speeds of 100 rpm and 7000 rpm respectively. The initial angular velocity chosen for two reasons. First, this initial condition allows each RW an equal range of speed change in both directions before saturation is reached. Second, as seen in Eq. (3.8) the angular velocities of the RWs have a direct impact on the EOM. Since in most cases the RWs of a control system are not allowed to operate at low speed, it was desirable to include some non-trivial angular velocity as a starting point.

3.3.4 Assumptions

3.3.4.1 Mass Moment of Inertia Tensor

The body reference frame MOI tensor I used for the REBEL test frame in the simulation is

$$I = \begin{bmatrix} 1843 & 0 & 0 \\ 0 & 1843 & 0 \\ 0 & 0 & 2267 \end{bmatrix} \text{kg-m}^2. \tag{3.19}$$

The shape of this MOI tensor represents of a body reference frame that is perfectly aligned with the principal axis. The magnitude of the values in the MOI tensor represent the mass of REBEL while carrying its maximum rated payload mass. Two considerations of this MOI tensor are important to understand.

First, in reality it is impossible to arrange the body reference frame such that the products of inertia are exactly zero. The assumption of zero valued products of inertia introduce some error into the simulation, but it is assumed that through careful placement of the body reference frame in reality, the products of inertia can be minimized. It can also be mentioned once again that the absolute performance of the control system is not under inspection. For the purposes of this research, the product of inertia terms are assumed to be zero.

Second, the inclusion of the maximum payload was chosen to exacerbate reaction wheel saturation speeds. That is, for increased REBEL MOI values, the RWs must undergo a greater change in angular momentum to provide the same change in angular velocity of the satellite. The consequence of this larger angular momentum change is that RW saturation speeds are reached more quickly than if the REBEL MOI values were reduced. Since the allocation methods presented in this research serve as a means of mitigating against these types of actuator constraints, it was desirable to increase the MOI values as much as possible while remaining within the values of reality.

3.3.4.2 External Moments

In this research, the applied external moments in the simulation $\vec{M}$ are assumed to be zero. Removing these assumptions is left as suggested future work, where developers can more accurately simulate absolute system performance. In reality, REBEL experiences a number of external torques while operating. These external torques can include gravitational pull due to an imperfectly positioned center of mass, small air currents in the room, air drag during maneuvers, and drag from the friction in the air bearing during maneuvers. The presence of these external moments would cause increased demands from the high level controller to offset. The implications of these increased demand values is that the RWs would reach torque limits more frequently and saturation speeds more quickly.

The control allocation methods researched in this thesis serve as mitigation against motor torque and saturation speed constraints of the RWs. It is believed that in the presence of these external torques, the relative performance increase of the control systems utilizing these improved allocation methods would be more prevalent. If a relative performance increase can be observed without the presence of these external moments, the results should be sufficient enough to provide evidence that they would also provide enhanced performance under the presence of such external moments. Further research could be performed with the external moments included in the model as physics based equations to verify this claim. Significant research would need to be performed however to accurately characterize these external moments, which are often nonlinear mathematical models.

3.3.4.3 Reaction Wheel Bearing Friction and Air Drag

The RWs on REBEL experience air and bearing drag in reality. This drag would serve as a method of momentum transfer between the spinning wheel and the REBEL frame. To counteract this undesirable momentum exchange, the high level controller would command a small torque to negate this momentum transfer. This small additional torque command would result in increased RW torques throughout the maneuver, effectively decreasing the torque limit of the motors. The drag torques would increase as a function of speed, so these effects would be more pronounced at higher RW speeds where the torque limits are already lower as a result of the motor physics. This effective reduction of maximum torque causes an effect very similar to the exacerbation of actuator constraints due to the application of external moments.

These drag forces are treated as negligible in this simulation for the same reason that the external moments are neglected. The drag forces would create a small additional torque that the RWs must continuously overcome, causing them to reach their torque limits more readily during maneuvers. The inclusion of physics based models of these drag forces would greatly complicate the model while only serving to further pronounce the advantages of the control allocation methods presented in this research, since they are meant to mitigate against torque limit constraints. If performance improvements can be shown without the presence of these drag forces, it should be sufficient to show that their performance improvements would only be more evident with drag forces imposed. The effects of RW drag on the model can be left to later research for verification, where significant effort would be required to accurately model the drag forces experienced by the RWs.

3.3.4.4 Rate Constraints

Finally, there were no rate constraints placed on the angular velocity of the REBEL test frame in the simulation. Such constraints would allow the test frame to accelerate only to some velocity limit, coast toward the final orientation, then decelerate to a stop at the final orientation. Rate constraints are normally implemented to mitigate RWs from accelerating to saturation speeds as well as limiting the control effort needed to complete a maneuver at the expense of increasing the maneuver time. The absence of rate constraints can cause feedback style controllers to behave similar to bang bang controllers where more refined control demands are only commanded when the orientation error becomes small. Since this research seeks to mitigate the effects of actuator constraints, these rate constraints were not implemented as reaching actuator constraint is considered desirable.

3.4 Single Maneuver

The single maneuver (analyzed in detail) used in this research is a 1-2-3 Euler rotation of $15^o, 15^o,$ and 15^o respectively. The corresponding final quaternion for this maneuver is

$$\bar{q}_f = \begin{bmatrix} 0.1452 & 0.1114 & 0.1452 & 0.9723 \end{bmatrix}^T . \tag{3.20}$$

This maneuver was chosen for several reasons. First, it ensures that some amount of rotation occurs on every axis. Second, the values provide a final orientation that lie well within the rotational capabilities of the REBEL test bed. Finally, the rotation is small enough that the high level linear feedback controller used in this research was stable. More information on the high level controller used is be presented in Section 3.6.

A requirement was added to the simulation that the final orientation must be held within ±1^o for 10 s. This requirement was implemented to ensure that the simulated maneuver was not considered complete if the test bed merely passed through the final orientation before it had settled (overshoot). This required hold time is also similar to a hold time that might be placed on an actual imaging or directional communication satellite being oriented towards a target.

3.5 Multiple Maneuvers

In order to provide some level of validation to the results obtained from the single maneuver, an array of 343 different maneuvers is also simulated for each allocation method. This set of multiple maneuvers is performed so that an average of performance metrics can be obtained. This set of maneuvers varies the final orientation of the simulated REBEL maneuver while keeping the initial orientation constant according to Eq. (3.16). The final orientations can be described as a 1-2-3 Euler rotation of x_f, y_f, and z_f, where each parameter of the Euler rotation is varied from 0^o to 15^o by increments of 2.5^o. Every possible combination of x_f, y_f, and z_f were simulated, producing 7^3, or 343 unique maneuvers. The maximum maneuver slew was intentionally kept small to prevent the linear feedback controller used in the control loop from causing the system to exhibit instability. This is a limitation of the simulation that prevents the control allocation methods presented in this research from being applied to larger (and consequently faster) slews.

3.6 High Level Controller

Although the development of a high level controller is not the focus of this research, one is needed nonetheless to produce a vector of command torques $\vec{\tau}_{com}$ for the control allocation algorithm to work with. Therefore, a simple controller that would remain unchanged throughout all of the simulations was desired. One such simple controller is the Proportional Differential (PD) controller. Obviously, such a simple linear controller is a poor choice for controlling the nonlinear dynamics of the REBEL test bed. That is to say there are many more sophisticated controllers that would improve the absolute response of the system in some regard. The absolute performance of the closed loop control system is not of importance though, only the relative performance between simulations that utilize different control allocation algorithms. The control law for this PD controller can be defined as [25]

$$\vec{\tau}_{com} = K_P I \bar{q}_{error}(1:3) + K_D I \vec{\omega} - \vec{\omega} \times \left(I\vec{\omega} + S\bar{\psi} \right). \tag{3.21}$$

where K_P is the proportional gain matrix

$$K_P = \begin{bmatrix} 1.5 & 0 & 0 \\ 0 & 1.5 & 0 \\ 0 & 0 & 1.5 \end{bmatrix}, \tag{3.22}$$

K_D is the differential gain matrix

$$K_D = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix}, \tag{3.23}$$

$\bar{q}_{error}(1:3)$ is the first 3 terms of the orientation error in terms of quaternions

$$\bar{q}_{error} = \begin{bmatrix} q_{f,4} & q_{f,3} & -q_{f,2} & -q_{f,1} \\ -q_{f,3} & q_{f,4} & q_{f,1} & -q_{f,2} \\ q_{f,2} & -q_{f,1} & q_{f,4} & -q_{f,3} \\ q_{f,1} & q_{f,2} & -q_{f,3} & -q_{f,4} \end{bmatrix} \bar{q}, \tag{3.24}$$

I is the MOI tensor of REBEL, $\vec{\omega}$ in the angular velocity of REBEL, and S is the transformation matrix defined in Eq. (3.4).

The proportional and derivative gain matrices were experimentally determined because standard linear analysis tools could not be applied to the nonlinear closed loop system. These gain values were chosen with simple goals in mind. First, the system response needed to be stable for the maneuver being performed. Fortunately, the stability of the system was not overly temperamental to the selection of these gains, largely due to the relatively small maneuver being performed and the large MOI values of the REBEL test frame relative to the RW MOI values which generally keeps the slew rates small. Second, the gain matrices needed to be chosen such that the commanded torques frequently reached values that caused the motors to reach their torque constraint limits. Finally, it was desirable that the high level controller's commands would quickly drive the RWs toward their saturation speed limits (which corresponds well with the second goal.) The gain matrices depicted in Eq. (3.22) and (3.23) met all of these goals when used in the simulation described in the previous sections.

It should be noted that limits are placed on the commands generated by the PD controller in the simulation. The limits are necessary because the PD controller generates arbitrarily large torque commands that do not compare well with what the RWA is capable of achieving. These large torque commands present a problem when analyzing the results of the simulation, as the applied torques do not compare well against the commanded torques a majority of the time. Therefore, a method to determine the maximum possible torque achievable for each individual body axis was devised, referred to as the Maximum Achievable Moment Set (MAMS), and used to impose limits on the torque commands generated by the PD controller. Equation (3.25) shows how the MAMS is determined.

$$\text{MAMS} = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix} \bar{\tau}_{nominal,[4\times1]} \tag{3.25}$$

Here, $\bar{\tau}_{nom}$ is a 4 by 1 array where each element is the nominal maximum continuous torque of the motor (see Table 3.1) and the matrix is the element-wise absolute value of the transformation matrix presented in Eq. (3.4) that transforms the array of RW angular accelerations $\dot{\bar{\psi}}$ into the satellite body reference frame (motor torque can be considered as the product of the motor's angular acceleration and the axial MOI term of the RW.) Essentially, the MAMS determines the maximum torque that the combined RWs can apply to each axis of the REBEL test frame if each RW was operating to produce the maximum torque on that axis and that axis only. These values cannot be achieved simultaneously unless the non-zero elements in their corresponding rows of the transformation matrix do not share non-zero entries in any of the columns where they exist. For instance, according to Eq. (3.25) the $\hat{b}_3$ element of the MAMS can be achieved in conjunction with the $\hat{b}_1$ or $\hat{b}_2$ element, but the $\hat{b}_1$ and $\hat{b}_2$ elements of the MAMS cannot be achieved simultaneously.

The purpose of applying these limits to the high level controller torque commands $\vec{\tau}_{com}$ is simply to limit their values to a point where they are comparable to the applied torques $\vec{\tau}$. It should not be expected that the applied torques meet these values often, which is verified by the simulation results. It should be noted that these torque command limitations do not impact the high level controller's performance, as it represents the maximum torque that could ever be applied by the combined actuators to any single body frame axis. It should also be noted that Eq. (3.25) is only valid as a result of the symmetry of the RW layout. For a non-symmetric RWA layout, the MAMS formulation may need to be reformulated.

3.7 Low Level Motor Controller

The low level motor controllers should also be given some consideration since they form a portion of the closed loop control system. When evaluating the absolute performance of a control system intended for use as flight hardware, its inclusion in the simulation would be critical. This research attempts to isolate the control allocation algorithm and assess the relative performance of the control system based on different control allocation algorithms. For this reason, it was not deemed necessary to model the low level motor controllers. That is, it is assumed that the torque commands assigned to the RWs by the control allocation algorithm are identical to the torque delivered by the RW motors. This assumption was considered valid since this research only aims to compare the relative performance between different control allocation algorithms when the remained of the control system remains constant. In reality the low level motor controllers may introduce some steady state error and phase delay that may influence the closed loop performance of the control system.

3.8 Control Allocation Methods

3.8.1 Simple Pseudo Inverse

The SPI control allocation method is the simplest of the methods used in this research. It is commonly used to allocate controls in over actuated systems when control allocation is not included as a part of the high level controller. In this allocation method, the Moore-Penrose pseudo inverse discussed in Section 2.5.2 is used to allocate the commanded torque $\vec{\tau}_{com}$ as actuator inputs $\bar{u}$.

This allocation method is an excellent choice when it is desirable to minimize the control effort of the actuators and actuator constraints are not of concern. In actuality, actuator constraints are a physical reality of nearly all actuators. Under the SPI method, when actuators become constrained as a result of the allocated commands, they are simply truncated to their constraint values and the allocation error $\vec{\tau}_{com} - \vec{\tau}$ suffers greatly. The detrimental effects of the RW constraints could potentially be mitigated by giving some consideration to the values of the weighting matrix W of Eq. (2.22) that is otherwise identity,; however, it would require apriori knowledge of the maneuvers being performed. Also, designing such a weighting matrix that would limit actuator constraint in every case would be difficult if not impossible.

The SPI allocation method is used as a baseline against which the other two allocation methods discussed in this research are compared against. If another control allocation method shows improved system performance when implemented in the simulation, as compared to the results produced when using the SPI allocation method, it is considered a success.

3.8.2 Redistributed Pseudo Inverse

The RPI control allocation method is an improved version of the SPI method that has been evaluated and utilized in existing research.[1, 5, 19, 20, 21] Unlike the SPI method, it mitigates against direct actuator constraints, such as motor torque limits in RWs. This mitigation is accomplished by iteratively re-solving the SPI whenever an actuator becomes constrained. The iterative solutions use only the remaining unconstrained actuators in an attempt to redistribute the unachieved command torque $\vec{\tau}_{com}$. This iteration is performed until the command torques are achieved, all actuators have become constrained, or no improvement to the applied torque vector $\vec{\tau}$ can be made with the remaining unconstrained actuators.

The RPI algorithm follows the steps outlined below.[1] Here, it is described as it pertains to three axis attitude control utilizing RWs, though it can be generalized for any number or type of generalized command forces.

1. Distribute three axis command torques $\vec{\tau}_{com}$ to the RWs using the SPI distribution method described in Section 3.8.1 and Section 2.5.2. In particular, Eq. (2.24) is used to allocate the command torque $\vec{\tau}_{com}$ as actuator accelerations by replacing the B matrix with the S matrix presented in Eq. (3.4). Here, the B matrix retains its representation as B for consistency with previous sections, but the reader should keep in mind that it is equated to the S matrix.

$$\bar{u} = B^{+}\vec{\tau}_{com} = S^{+}\vec{\tau}_{com} \tag{3.26}$$

2. Determine if $\bar{u} \in \mathbb{U}$. That is, determine if the torque distribution from the solution of Eq. (3.26) has caused any of the RWs to exceed their acceleration rates (a function of their motor torque limits and radial principal MOI components D) or saturation speeds.

   - If $\bar{u} \in \mathbb{U}$, the solution is complete
   - If $\bar{u} \notin \mathbb{U}$, proceed to step 3.

3. Separate the allocated RW acceleration values $\bar{u}$ into constrained $\bar{u}_c$ and unconstrained $\bar{u}_u$ subsets. Separate the columns of the B matrix accordingly:

$$\bar{u} = \text{proj}_{\mathbb{U}}\left(\bar{u}\right)$$

$$\bar{u} = \left(\bar{u}_c^T, \bar{u}_u^T\right)^T \tag{3.27}$$

$$B = \left(B_c, B_u\right) \tag{3.28}$$

4. Determine the three axis torque that is produced by the constrained RWs only, utilizing the constrained portions of separated matrices formed in Eq. (3.27) and (3.28):

$$\vec{\tau}_c = B_c \bar{u}_c \tag{3.29}$$

5. Determine the difference between the commanded three axis torque $\vec{\tau}_{com}$ and the three axis torque produced by the constrained actuators $\vec{\tau}_c$. Redistribute the "remaining" torque to the unconstrained RWs using

$$B_u \bar{u}_u = \vec{\tau}_{com} - \vec{\tau}_c$$

and

$$\bar{u}_u = B_u^+ \left(\tau_{com} - \vec{\tau}_c\right). \tag{3.30}$$

6. Determine if $\bar{u}_u \in \mathbb{U}$. That is, determine if the torque reallocation to the unconstrained RWs performed in step 4 has caused any of them to become constrained.

   - If $\bar{u}_u \in \mathbb{U}$, the solution is complete, where $\bar{u} = \left(\bar{u}_c^T, \bar{u}_u^T\right)^T$.
   - If $\bar{u}_u \notin \mathbb{U}$, return to step 3, where $\bar{u} = \left(\bar{u}_c^T, \bar{u}_u^T\right)^T$.

This algorithm is to be repeated until one of the following cases has been met.

1. The command torque $\vec{\tau}_{com}$ is equivalent to the allocated RW torques subject to their constraints:

$$\vec{\tau}_{com} = B\bar{u}, \quad \bar{u} \in \mathbb{U}$$

2. All RWs have become constrained.

3. No improvement can be made to the allocated torques. That is, the remaining unconstrained RWs have no control authority over one or more axes where the command torque has not been met.

One consideration that must be made when using the RPI method is that the B_u matrix can become rank deficient. Such a case can occur when there are no unconstrained actuators remaining with control authority over one of the primary control axes. This rank deficiency does not imply that further improvements cannot be made to the axes for which control authority remains however, so an attempt to redistribute torque to the unconstrained RWs must still be made. Computing the redistribution is problematic though because the Moore-Penrose pseudo inverse cannot be computed as shown in Eq. (2.23) when B is rank deficient.[1]

One solution to rank deficiency issue is to use an inversion method based on Singular Value Decomposition (SVD).[26] Beginning with the Moore-Penrose pseudo inverse from Eq. (2.23)

$$\bar{u} = C\vec{\tau}_{com}$$
$$C = B^{+} = B^{T}\left(BB^{T}\right)^{-1}$$

the $\left(BB^{T}\right)$ term of the equation can be replaced with its SVD.

$$
\begin{aligned}
\text{SVD}\left(BB^T\right) &= U\Sigma V^T \\
C = B^+ = B^T\left(U\Sigma V^T\right)^{-1} &= B^T U^{-1}\Sigma^{-1}\left(V^T\right)^{-1}
\end{aligned} \tag{3.31}
$$

The properties of the SVD are such that both U and V are orthogonal matrices, so their inverses are equal to their transposes. Σ is a diagonal matrix, which contains a number of nonzero entries along its diagonal equal to the rank of B, or BB^T. Taking advantage of these properties, Eq. (3.31) can be rewritten as

$$
\begin{aligned}
C &= B^T U^T \Sigma^{\text{inv}} V \\
\Sigma^{\text{inv}} &= \text{diag}\left(\sigma_1^{-1}, \sigma_2^{-1}, ..., \sigma_p^{-1}, 0, ..., 0\right)
\end{aligned} \tag{3.32}
$$

where p is the rank of B or BB^T and the number of terms along the diagonal are equal to the number of rows in B or BB^T.

Equation (3.32) permits a solution to be computed for the Moore-Penrose pseudo inverse when B is rank deficient. The SVD method is also valid for cases where B has full rank, so the only downside of using it exclusively is increased computational difficulty in the algorithm.

Though the RPI method is strictly better than the SPI, it still has several downsides. First, the RPI approach does not guarantee that the allocated torques $\vec{\tau}$ equal the commanded torques $\vec{\tau}_{com}$ whenever possible. Second, there is no guarantee that the error between the allocated and commanded torques is minimized in some sense when an exact solution is not possible.[1]

The minimization of error between command and applied torque when an exact solution is not possible cannot be achieved without the inclusion of an allocation error term in the cost function (Eq. (2.22).) The addition of an allocation error term drastically increases the complexity of the minimization solution to a point where a closed form solution is not attainable and linear or quadratic programming methods must be invoked.[1] The study of such allocation methods is left to future research.

The concern that the RPI approach does not guarantee that the allocated torques equal the commanded torques whenever possible is the result of a mitigable error in the algorithm. In Bodson's[19] simulation of over actuated aircraft, this error was noted but its cause was not identified. Bodson went on to provide a simple example where this issue was clear and prominent. To better understand the cause of this error, the example Bodson provided is repeated and commented on here.

Consider the command torque and linear relation matrix:

$$\vec{\tau}_{com} = \begin{bmatrix} 0 \\ 9 \\ 0 \end{bmatrix} , \quad B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \tag{3.33}$$

where the only actuator constraints are:

$$\bar{u}_{max} = -\bar{u}_{min} = \begin{bmatrix} 5 & 10 & 2 & 1 \end{bmatrix}^T . \tag{3.34}$$

Following step 1 of the RPI allocation method yields the unconstrained solution as:

$$\bar{u} = \begin{bmatrix} 0 & 6 & -3 & 3 \end{bmatrix}^T . \tag{3.35}$$

Step 2 of the algorithm applies the constraints of Eq. (3.34) and $\bar{u}$ becomes:

$$\bar{u} = \begin{bmatrix} 0 & 6 & -2 & 1 \end{bmatrix}^T . \tag{3.36}$$

Step 3 of the algorithm then divides $\bar{u}$ into its constrained and unconstrained elements:

$$\begin{aligned} \bar{u}_u &= \begin{bmatrix} 0 & 6 \end{bmatrix}^T, \quad \bar{u}_c = \begin{bmatrix} -2 & 1 \end{bmatrix}^T \\ B_u &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B_c = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}. \end{aligned} \tag{3.37}$$

Step 4 of the algorithm then determines the torque produced by the constrained actuators only:

$$\vec{\tau}_c = \begin{bmatrix} 0 & 1 & -1 \end{bmatrix}^T. \tag{3.38}$$

Step 5 then reallocates the unconstrained actuators $\bar{u}_u$ to the difference between the commanded torque $\vec{\tau}_{com}$ and the constrained torque $\vec{\tau}_c$:

$$\bar{u}_u = \begin{bmatrix} 0 & 8 \end{bmatrix}^T. \tag{3.39}$$

The new solution for $\bar{u}_u$ does not violate the constraints, so the algorithm has completed with final values of:

$$\bar{u} = \begin{bmatrix} 0 & 8 & -2 & 1 \end{bmatrix}^T, \quad \vec{\tau} = \begin{bmatrix} 0 & 9 & -1 \end{bmatrix}^T. \tag{3.40}$$

However, there obviously exists multiple exact solutions. For example:

$$\bar{u} = \begin{bmatrix} 0 & 9 & 0 & 0 \end{bmatrix}^T, \quad \vec{\tau} = \begin{bmatrix} 0 & 9 & 0 \end{bmatrix}^T. \tag{3.41}$$

The solution method breaks down when the constrained actuators are selected in Eq. (3.37). When the two actuators become constrained, their values become "locked" set equal to their applicable constraint values. Also, two columns have become removed from the linear relation matrix B to form the unconstrained matrix B_u. There are two implications of simultaneously removing these two columns.

First, the unconstrained linear relation matrix B_u has been made rank deficient. Rank deficiency implies that the unconstrained actuators do not have control authority over the span of the applied torque $\vec{\tau}$. In this sample case, the unconstrained actuators have no control authority of the third value of $\vec{\tau}$. Consequently, the third value of $\vec{\tau}$ produced by the constrained actuators $\bar{u}_c$ in Eq.(3.38) is no longer available for manipulation in future redistribution iterations. This loss of manipulation presents an obvious error as the third value of $\vec{\tau}$ is not equal to the third value of the command torque $\vec{\tau}_{com}$. Therefore, the redistribution of the unconstrained actuators $\bar{u}_u$ are unable to correct this error.

Second, more than one actuator became constrained in the same iteration of the algorithm. The simultaneous constraint of multiple actuators becomes of consequence when the constrained actuators are coupled in the linear relation matrix B. That is, they both have control authority over one or more of the same applied torque $\vec{\tau}$ values (i.e. they each have nonzero values in the same row of their corresponding columns of B) In such an event, the coupled constrained values are "locked". If however one of the constrained actuators were allowed to be redistributed in the next iteration of the algorithm it may be redefined at a value within the bounds of the constraints that also provides an exact solution.

If instead the maximum number of actuators considered to be constrained per iteration is limited to one rather than the actual number of actuators that have exceeded constraints, both of these issues can be mitigated.

First, the number of columns removed from the B matrix to for the B_u matrix is limited to a maximum of one, reducing the rate at which the row rank of B may decrease as much as possible.[1] Consequently, this limitation leaves the unconstrained matrix with full control authority for as long as possible.

Second, in the case where the constrained elements of the allocated actuator command array $\bar{u}$ are coupled in the linear relation matrix B, when the single constrained element becomes "locked" the other coupled elements are free to be updated in subsequent iterations of the algorithm which may result in unconstrained values that satisfy the exact solution. To facilitate these outcomes, it is generally best to select the constrained actuator with the greatest absolute value as the single actuator which becomes the "locked" constrained value. Jin[21] has proposed other methods for selecting the single actuator to apply the constraint to, but these other methods are left for future research.

By applying the single actuator constraint limitation to the algorithm and re-solving the example problem provided by Bodson produces a more favorable result. The steps of the solution remain unchanged until Eq. (3.37) when the constrained elements of B and $\bar{u}$ are chosen. The new solution to this step then becomes:

$$\begin{gathered}\bar{u}_u = \begin{bmatrix} 0 & 6 & -3 \end{bmatrix}^T, \quad \bar{u}_c = \begin{bmatrix} 1 \end{bmatrix} \\ B_u = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad B_c = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}. \end{gathered} \tag{3.42}$$

Here, the third element of $\bar{u}_u$ violates the constraints presented in Eq. (3.34), but the redistribution iteration is allowed to manipulate it in case its reallocated value may be within the bounds of its constraint. It is also evident that B_u has retained full row rank in this case, allowing the redistribution of the unconstrained actuators

[1] Depending on the structure of the B matrix, it is possible for it to become rank deficient with the removal of a single column even if the initial number of columns is four or greater. It is however assumed that the control system's design results in a B matrix that is initially of full row rank.

$\bar{u}_u$ to have full control authority over the elements of $\vec{\tau}$. Proceeding to step 4 of the solution process:

$$\vec{\tau}_{constraint} = \begin{bmatrix} 0 & 1 & 1 \end{bmatrix}^T . \tag{3.43}$$

Then in step 5, the reallocated unconstrained actuators have the solution:

$$\bar{u}_u = \begin{bmatrix} 0 & 8 & -1 \end{bmatrix}^T . \tag{3.44}$$

The new solution for $\bar{u}_u$ does not violate the constraints, so the algorithm has completed with final values of:

$$\bar{u} = \begin{bmatrix} 0 & 8 & -1 & 1 \end{bmatrix}^T , \quad \vec{\tau} = \begin{bmatrix} 0 & 9 & 0 \end{bmatrix}^T . \tag{3.45}$$

By incorporating the single constraint limitation, the explicit example that Bodson provided to show fault with the RPI method has been alleviated. This result however still does not guarantee that an exact solution will be determined whenever possible. There are still corner cases that result in inexact solutions when exact solutions exist. Jin[21] did show however that the single constraint limitation provided significant mitigation against this issue.

This issue inherent to the RPI method is presented to fully disclose one of its shortcomings, but it should detract from the fact that it is by design a strict improvement to the SPI.[1] The RPI method is in fact identical to the SPI method when no actuator constraints have been reached. The RPI method only redistributes torque among unsaturated actuators when constrained actuators have introduced allocation error.

Another drawback of the RPI method is that it lacks consideration for limiting torque allocation to RWs that are approaching saturation speed. Though it attempts to redistribute torque to other RWs after one has become saturated, it does not take measures to mitigate the RWs from reaching saturation. This can become problematic in RWAs where RW saturation is easily achieved, which is made apparent in the simulation results. A variant of the RPI method is introduced in the next section that helps address this lack of consideration.

3.8.3 Redistributed Pseudo Inverse with Adaptive Weighting

One method of mitigating against saturation speeds in the control allocation process outlined in Section 3.8.2 is to make use of the weighting matrix W which was otherwise considered to be identity. As previously mentioned, the weighting matrix can be an effective way of handling constraints in the control allocation problem. It is difficult however to select constant values for a weighting matrix without apriori knowledge of the planned maneuver and how it impacts the RW speeds. Creating such a constant weighting matrix that works to mitigate RW saturation under all conditions is difficult if not impossible. Research has been performed however that shows adaptive weighting matrices can be used to eliminate the need for apriori knowledge.[22] This research however did not make use of the redistribution process utilized in the RPI method. By replacing the identity weighting matrix used in the solution process of the RPI method with an adaptive weighting matrix a new control allocation variant is formed, termed the Redistributed Pseudo Inverse with Adaptive Weighting allocation method, or RPIW. This method has not been previously studied in research and is thought to be a unique idea.

Consider the weighting matrix

$$W = \begin{bmatrix} \alpha_1 & 0 & \dots & 0 \\ 0 & \alpha_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \alpha_p \end{bmatrix} \tag{3.46}$$

where α_i is a weighting value associated with each of the n RWs. If this weighting matrix is used in Eq. (2.23) (the general solution to the minimization of the quadratic cost function that measures control effort) rather than an identity matrix, the solution becomes

$$\bar{u} = \left(W^{-1}B^T \left(BW^{-1}B^T\right)^{-1}\right) \vec{\tau}_{com}. \tag{3.47}$$

If the values of α_i are chosen based on the current RW speed ψ_i, the reliance on that particular wheel can be limited or enhanced, where the solution is still subject to $\vec{\tau}_{com} = B\bar{u}$. By increasing the weighting of a particular actuator, the reliance on that actuator decreases as the cost function of Eq. (2.22) attempts to minimize the total cost. Similarly, if the weighting of a particular actuator is decreased, the reliance on that actuator increases. Equation (3.48) shows how the weighting values were determined for this simulation.[2]

[2] Considerations should be made to change the conditionality of this equation if the reaction wheels are operating at negative speeds

if $\dot{\psi}_i \geq 0$

$$\alpha_i = \begin{cases} 1 + \frac{99(\psi_i - 0.8\psi_{max})}{\psi_{max} - 0.8\psi_{max}}, & \text{if } \psi_i > \bar{\phi} \\ 1, & \text{if } \underline{\phi} \leq \psi_i \leq \bar{\phi} \\ 0.01 + \frac{0.99*(\psi_i - 100 \text{ rpm})}{0.2\psi_{max}}, & \text{if } \psi_i < \underline{\phi} \end{cases} \tag{3.48a}$$

Otherwise

$$\alpha_i = \begin{cases} 0.01 + \frac{0.99(\psi_{max} - \psi_i)}{\psi_{max} - 0.8\psi_{max}}, & \text{if } \psi_i > \bar{\phi} \\ 1, & \text{if } \underline{\phi} \leq \psi_i \leq \bar{\phi} \\ 1 + \frac{99(0.2*\psi_{max} + 100 \text{ rpm})}{0.2\psi_{max}}, & \text{if } \psi_i < \underline{\phi} \end{cases} \tag{3.48b}$$

These equations represent a linear change in the weighting value α_i when the associated RW speed ψ_i is within some value of the upper of lower saturation speed. These upper and lower bounds are denoted as $\bar{\phi}$ and $\underline{\phi}$ respectively. In this simulation these boundaries were chosen such that $\bar{\phi}$ is $0.8\psi_{max}$ and $\underline{\phi}$ is $0.2\psi_{max} + 100$ rpm. These bounds were chosen to be approximately 20% of the trailing ends of the range of possible RW speed. These boundary values allowed the existence of an ample range in the center of the RW operational range where these weighting values remained as 1. Only when the RW speeds came close to saturation would the weighting values change and begin to mitigate. Conversely, if a RW is within these saturation boundaries and is accelerating such that it is leaving saturation, its corresponding weighting value is lowered in order to increase the use of said RW. Additional research could certainly be performed to evaluate if more appropriate boundaries exist than the ones chosen here, but for this preliminary look at the RPIW technique, these values appeared to work well. Figure 3.3 graphically depicts how these weights change with RW speed, and direction of acceleration

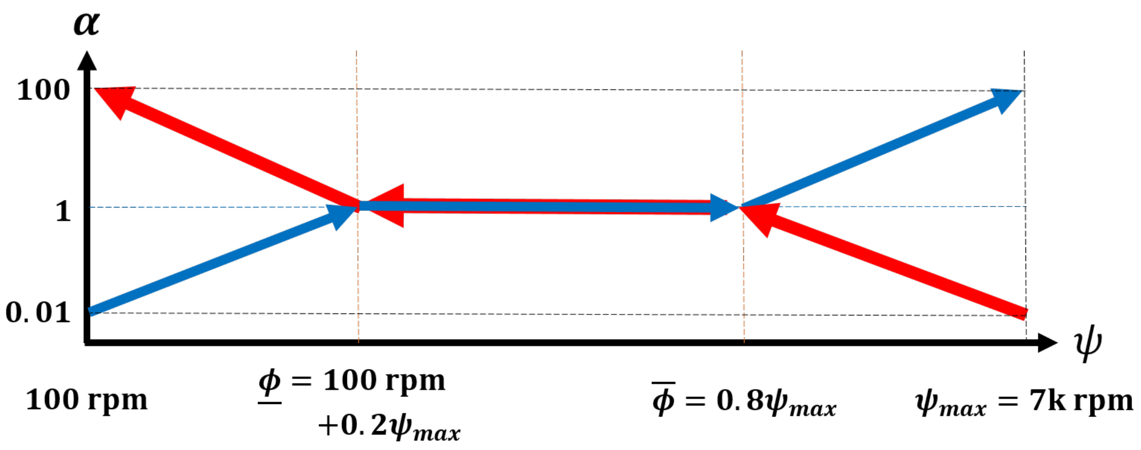

Figure 3.3. Weighting matrix values as a function of wheel speed

The absolute value of these weightings is not of importance, but rather their values relative to each other. Therefore, maximum and minimum values of α_i were chosen as 100 and 0.01 respectively. These values permitted a large relative difference between weighting values. Additional research could be conducted to determine if there are more optimal maximum and minimum values; however, the values used in this research are sufficient to show that the RPIW method has merit.

Not only does this allocation method make use of weighting matrices, it also utilizes the RPI method outline in Section 3.8.2. In theory, introducing adaptive weighting matrices to the RPI allocation method works quite well. On the first iteration, the weighting matrix ensures that the most appropriate actuators (in terms of saturation speed prevention) are most heavily relied on. If that allocation causes the actuator(s) to become torque constrained they are clipped, placed in the subset of constrained actuators, and removed from the remaining unconstrained set of actuators used in the redistribution. Since the primary goal of the allocation algorithm is to reduce allocation error, even if all of the remaining actuators with control authority are near saturation, they are used regardless. The eventual use of all actuators simply means actuators that are departing saturation get first priority during the allocation for a time step, those that are within the saturation bounds get second priority, and those that are approaching saturation receive last priority. The large relative difference between the maximum and minimum weighting values help ensure this priority structure works as intended. In the case where no RWs have exceeded the saturation bounds $\bar{\phi}$ and $p\underline{h}i$, the weighting matrix W is identity and the RPIW method becomes identical to the RPI method. In this regard, the RPIW method only begins to consider the secondary goal of RW saturation speed prevention when one or more RWs exceed the designated speed boundaries.

It should also briefly be noted that the weighting matrix W is always positive definite. As a result, its use in Eq. (3.47) does not cause the directionality of any of the allocated commands $\bar{u}$ to change. Should the directionality of any of these values be altered, it would impact the way the weighting values are calculated in Eq. (3.48). If the directionality of a RW torque was altered in the middle of a redistribution process, it could potentially have undesirable consequences for the weighted control allocation algorithm.

As with the RPI method, this iteration process continues until one of the following cases has been met:

1. The command torque $\vec{\tau}_{com}$ is equivalent to the allocated RW torques subject to their constraints:

$$\vec{\tau}_{com} = B\bar{u}, \quad \bar{u} \in \mathbb{U}$$

2. All RWs have become constrained.

3. No improvement can be made to the allocated torques. That is, the remaining unconstrained RWs have no control authority over one or more axes where the command torque has not been met.

The drawback of the RPIW method is that some control effort reduction may be traded off for saturation speed limit mitigation when the weighting matrix W is adjusted away from identity. This tradeoff is a result of the weighting matrix W altering the solution to the minimization function in Eq. (2.22). This tradeoff can be beneficial though if it prevents an actuator from reaching saturation which could lead to a more severe reduction in the closed loop performance. It may also provide increased RW life expectancy by reducing the amount of time RWs operate at speed extremes. Operating at such speed extremes tends to lower the life expectancy of RWs. The study of RW life expectancy is outside the scope of this research however and left for future work.

3.9 Summary

This chapter has provided an in depth evaluation of the methodology used to generate the results of this research. Information on the RW rotors, motors, and RWA layout have been outlined. The simulation of the REBEL test bed has been discussed in great detail, including the nonlinear dynamics and kinematics used, the numerical solution, the important parameters, and the initial conditions. The simulated maneuver, high level controller, and low level motor controller have been discussed in some detail. Finally, three control allocation algorithms have been discussed in great detail. The results presented in the next chapter provide a comparison of these different allocation methods when applied to the numerical simulation of REBEL described in this chapter.

IV. Results

4.1 Introduction

This chapter presents the results of the simulation described in Chapter III. The simulation is executed with each of the 3 control allocation methods implemented. The detailed analysis of the single maneuver are presented first The results are split into sections based on the allocation method used, and then a fourth section is presented in which the results are compared side by side. Then, a summary of the averaged 343 simulated maneuvers is presented and compared to the results of the single maneuver analysis. Desirable results would indicate that both the RPI and RPIW allocation methods reduce the allocation error between the torque commanded by the high level controller $\vec{\tau}_{com}$ and the applied torque $\vec{\tau}$. The performance of the closed loop system is also considered. The metrics used to gauge the performance and effectiveness of the different control allocation methods are discussed as they are presented. Desirable results would also indicate that the RPIW allocation method was able to mitigate against reaction wheels reaching saturation speed in some measurable way. These results are used to evaluate whether the RPI and RPIW appear to offer improved performance over the SPI method for the particular set of maneuvers examined. This evaluation will be used to draw conclusions about if and when the RPI and RPIW methods may be applicable in Chapter V.

4.2 Single Maneuver: Simple Pseudo Inverse

4.2.1 Simulation State Variables

Figure 4.1 plots the solution of the state matrix (Eq. (3.11)) across the time span of the simulated maneuver. The orientation of REBEL is plotted in terms of quaternions, and shows the orientation travel from its initial state $\bar{q}_0$ to its final state $\bar{q}_f$. The angular rates of REBEL are also plotted which demonstrates that the simulation starts at rest and concludes at rest at the target orientation. Finally, the figure plots the angular rates of the RWA $\bar{\psi}$ across the maneuver. Note that each wheel starts at an initial condition of 3500 rpm, and must stay within the operating range of the motor of 100 rpm and 7000 rpm. Of particular note is the brief periods that wheels 1 and 2 reach their saturation speed limits. The implication of RW saturation speeds are discussed further with the presentation of additional results.

The maneuver was completed in 91.10 seconds. This time includes the 10 second hold that is required on the final target, where the orientation must remain within ±1^o of the final orientation.

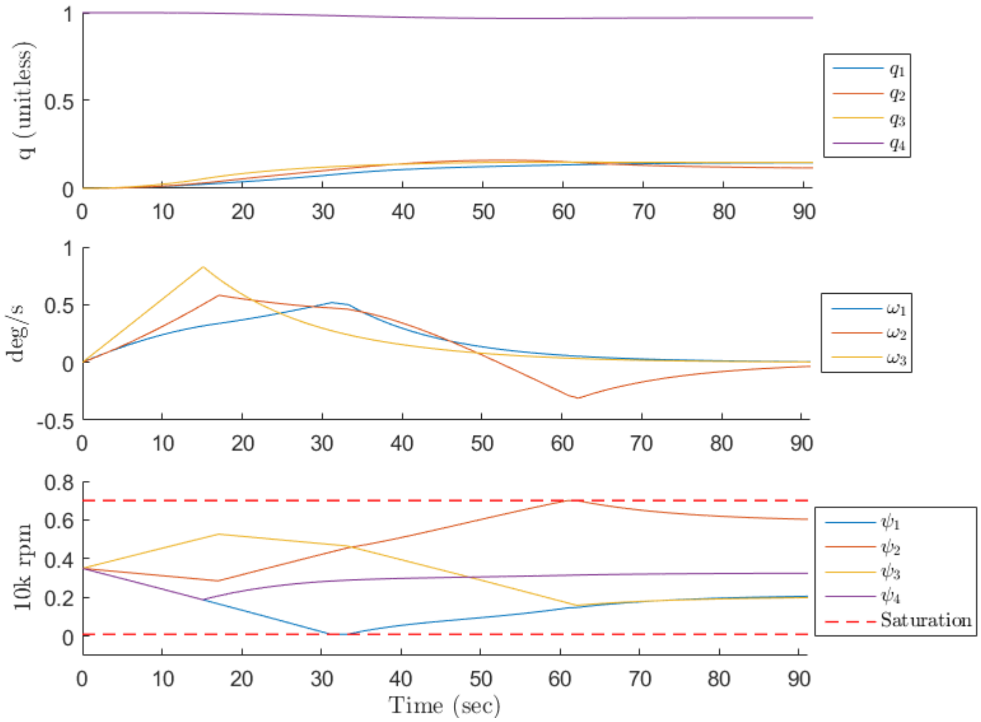

Figure 4.1. Simulation state equation (Eq. (3.11)) solution with simple pseudo inverse control allocation

4.2.2 Angular Momentum Analysis

Figure 4.2 plots the magnitude of the angular momentum of the spacecraft (i.e. REBEL) H_{SC}, the angular momentum of the RWA H_{RWA}, and the combined total angular momentum H_{TOTAL}. This plot is presented as a verification of the dynamics of the simulation. Since there are no external moments applied in the simulation, the rate of change of the total angular momentum should be zero; therefore, the total angular momentum should remain constant (Eq. (2.2).) Figure 4.2 shows that the total angular momentum does in fact remain constant. The figure also confirms that the total angular momentum of REBEL starts at zero (initially at rest) and ends at zero (at rest in its final orientation). The angular momentum of the RWA is exchanged with REBEL to produce the angular velocity required to complete the maneuver while keeping the total angular momentum constant at all time.

4.2.3 Motor Torques

Figure 4.3 plots the torque produced by each RW motor during the maneuver. Remember that RW 4 is a combination of 3 RWs, and the motor torque plotted for it here represents only 1 of those 3 wheels. The motor torque limit τ_{max} has been depicted on each plot as a dashed red line, and the motor torques are clipped at this value if they are commanded to exceed it. Recall that τ_{max} varies as a function of motor speed according to Eq. (3.5), and this variation can be seen in Fig. 4.3 although it is subtle.

A binary indicator of the saturation speed status of each RW has been imposed on the plots. This indicator has only two states, high or low, where the high state indicates that the RW is at either the upper or lower saturation speed. When the indicator is high, the RW motor is not permitted to produce torque that would drive the wheel beyond it saturation speed.

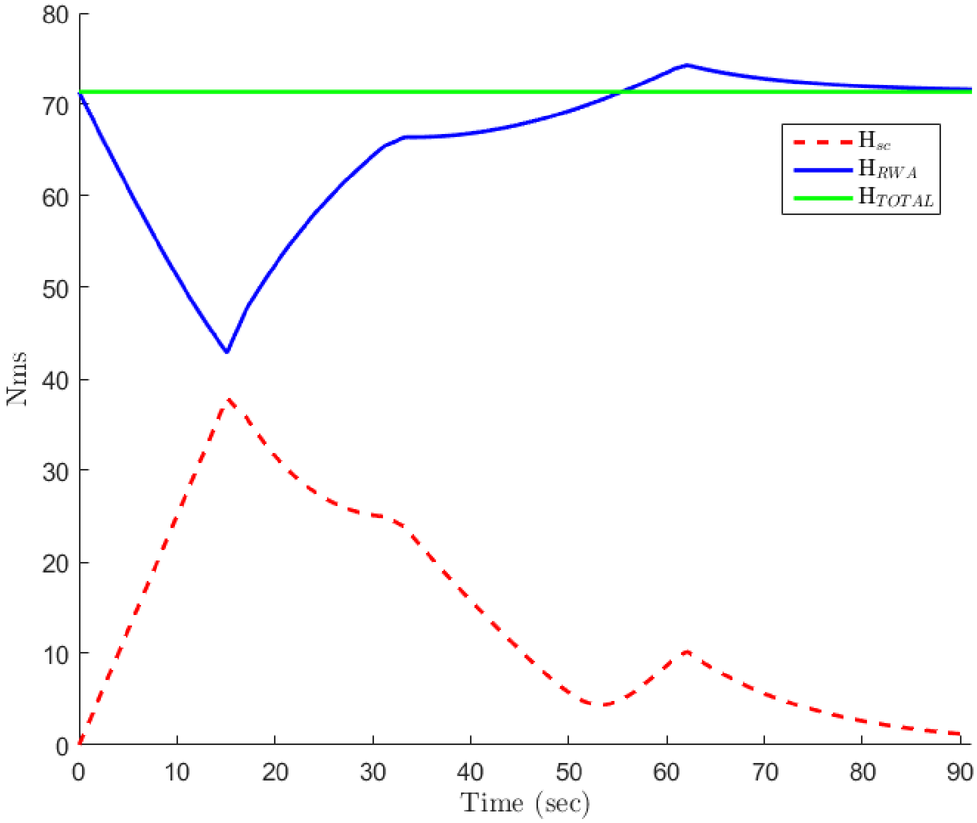

Figure 4.2. Satellite, RWA, and total angular momentum with simple pseudo inverse

This indicator shows that both RW 1 and 2 suffer from saturation speed constraints at some point in the maneuver. RW 1 remains saturated for a total of 1.95 seconds. RW 2 remains saturated for a total of 1.10 seconds. Altogether, the RW are at saturation speed for a total of 3.05 seconds during the maneuver.

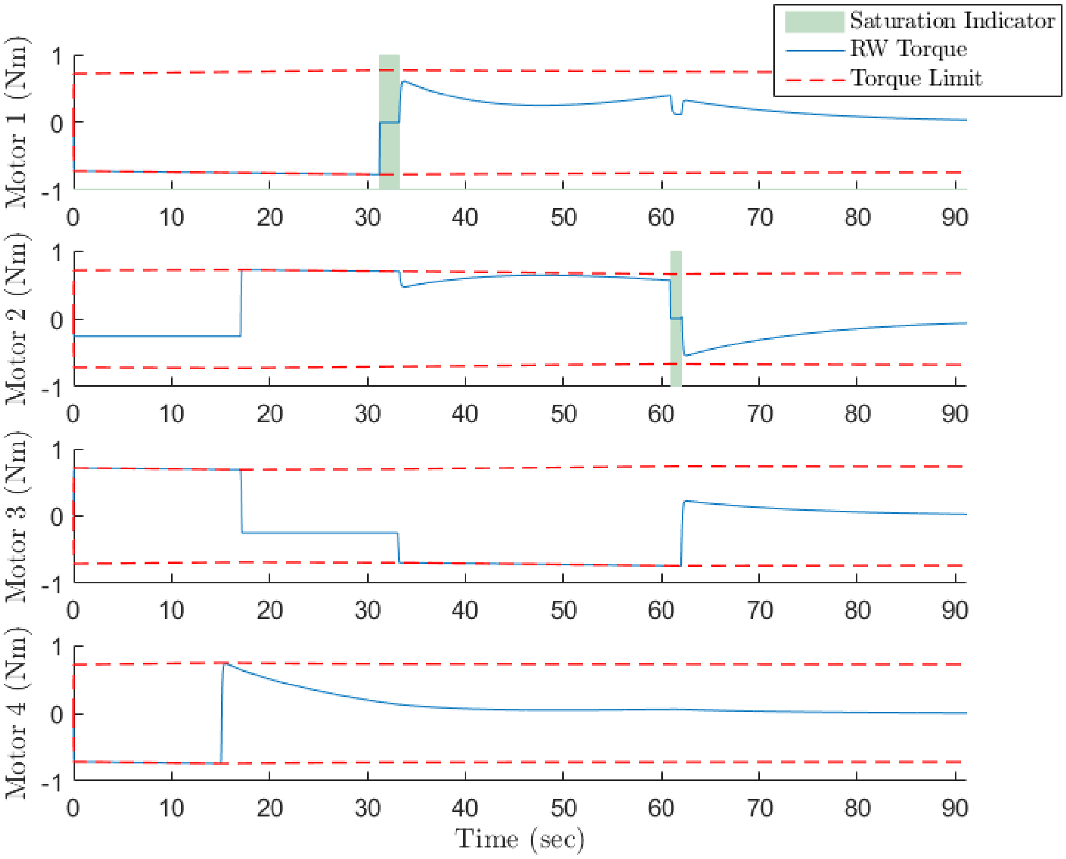

Figure 4.3. RW motor torques and saturation speed constraint indicators (binary) with the simple pseudo inverse allocation method

4.2.4 Commanded vs Applied Torques

Figure 4.4 shows the 3-axis torque commands from the high level controller $\vec{\tau}_{com}$ and the 3-axis torque that is applied to REBEL by the RWA $\vec{\tau}$. The applied torque produced by the RWA attempts to match the command torques, but is limited by actuator constraints and the ability of the allocation algorithm to mitigate against these constraints. The SPI allocation method features no such mitigation methods, so ability to match the applied torque to the command torque is severely limited when actuators become constrained.

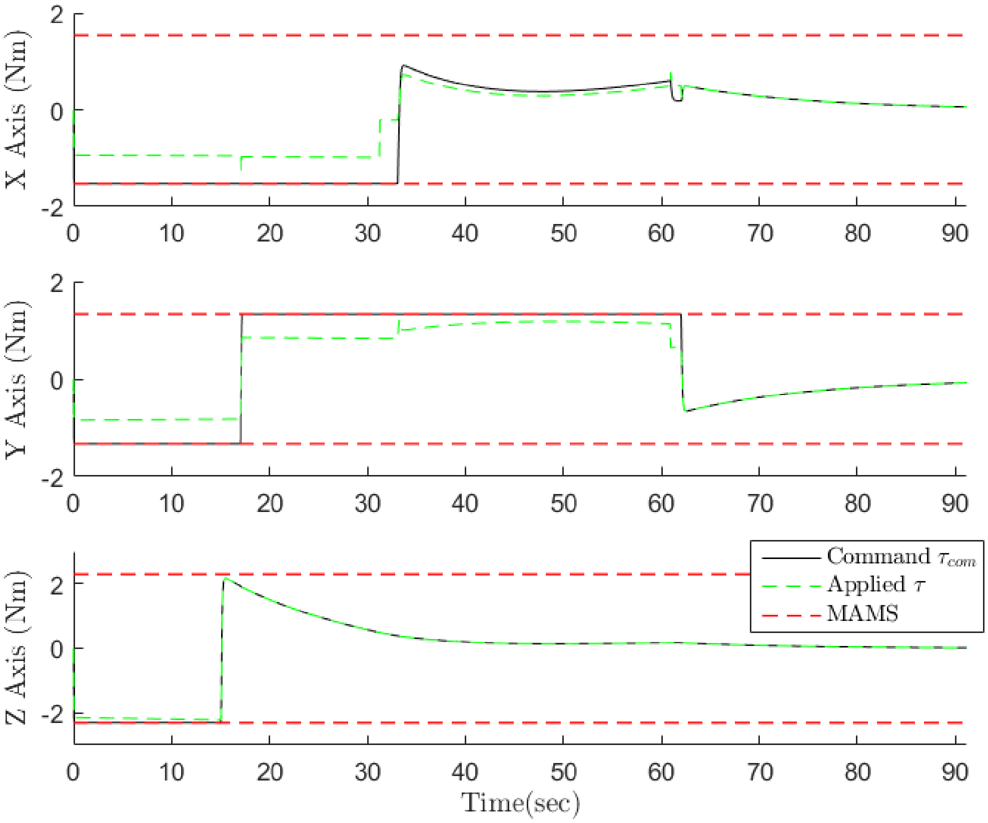

Figure 4.4. Torque commands from high level controller on each axis and torque applied after simple pseudo inverse allocation and constraint application

The high level controller's torque limits are also imposed on the plots in Fig. 4.4 as red dashed lines. These limits are the MAMS that were discussed in Section 3.6. As a reminder, these torque limits represent the maximum torque that could be achieved in any axis if all actuators with control authority in that axis were used to maximize the torque applied to that axis. Typically, due to actuators having control authority over multiple axes, MAMS torque limits cannot be achieved simultaneously. In fact, in many cases these limits should not be expected to be attainable. These limits are applied to the high level controller to prevent torque commands from being arbitrarily large and comparing poorly against the applied torque values. Note that any commands above the MAMS could not be achieved anyway, so this limitation does not impact the controllers performance.

Figure 4.4 provides insight into the control allocation algorithm's ability to match the applied torque $\vec{\tau}$ to the commanded torque $\vec{\tau}_{com}$. Unfortunately, the overall performance of the allocation algorithm can be difficult to discern when comparing the commanded and applied torques across the 3 separate axes. In order to make direct comparison possible, the 2-norm of the command torque vector $||\vec{\tau}_{com}||_2$ and the applied torque vector $||\vec{\tau}||_2$ are computed and plotted in Fig. 4.5.

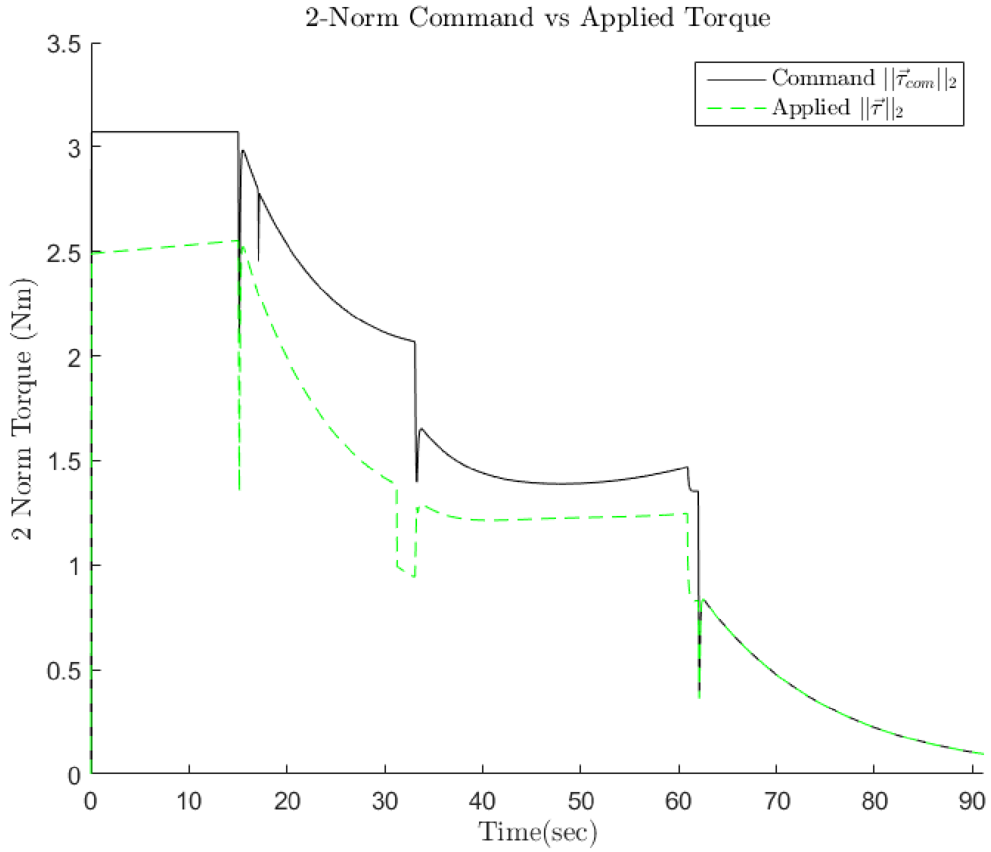

Figure 4.5. 2-norm of torque commands from high level controller and torque applied after simple pseudo inverse allocation and constraint application

Figure 4.5 depicts the magnitude of the commanded torque as compared to the magnitude of the applied torque. As the applied torque $\vec{\tau}_{com}$ approached the applied torque $\vec{\tau}$ the difference between these magnitudes will shrink. It is important to note however that reducing the difference between these magnitudes does not necessarily correspond with a reduction in allocation error $\vec{\tau}_{com} - \vec{\tau}$.

Several useful metrics of allocation performance can be calculated from the data plotted in Fig. 4.4 and 4.5. These metrics are presented in Table 4.1.

Table 4.1. Comparison metrics between commanded torques and applied torques for the control system utilizing the simple pseudo inverse allocation method

Axis	Mean % of τ_{com} Applied	Commanded Momentum Exchange/Control Effort (Nm-s)	Applied Momentum Exchange/Control Effort (Nm-s)	% of Commanded Momentum Exchange Applied
1	80.6	70.9	47.9	67.6
2	81.5	90.4	68.1	75.3
3	99.1	62.5	60.7	97.1
2-norm	86.5	140.7	114.3	81.2

The "Mean % of τ_{com} Applied" represents the time average percentage of the applied torque as it compares to the commanded torque. In other words, the percentage of the commanded torque that is applied is calculated at every time step of the simulation, then averaged together. The "Mean % of τ_{com} Applied" is calculated for each individual axis, and also calculated for the 2-norm of all 3 axes. This calculation is carried out as

$$\text{Mean \% of } \tau_{com} \text{ Applied} = \text{mean}\left(100\frac{\bar{\tau}}{\bar{\tau}_{com}}\right) \tag{4.1}$$

where $\bar{\tau}$ and $\bar{\tau}_{com}$ are time history arrays of the applied torque and command torque respectively for any single axis, or the 2-norm of the 3 axes. The quotient of these vectors is taken element wise.

Columns 3 and 4 of Table 4.1 represent the total control effort commanded by the high level controller and the total control effort delivered by the RWA during the maneuver. The total control effort can also be thought of as the total amount of momentum exchange. These values are calculated as

$$\begin{aligned}\text{Commanded Control Effort} &= \int_{t_0}^{t_f} \bar{\tau}_{com}\, dt \\ \text{Applied Control Effort} &= \int_{t_0}^{t_f} \bar{\tau}\, dt\end{aligned} \tag{4.2}$$

where $\bar{\tau}$ and $\bar{\tau}_{com}$ are time history arrays of the applied torque and command torque respectively for any single axis, or the 2-norm of the 3 axes.

The commanded and applied control effort are not particularly useful on their own, but can provide a measure of the allocation algorithm's performance when compared as a percentage. The percentage of the commanded control effort (or commanded momentum exchange) met by the applied control effort (or applied momentum exchange) populates the final column of Table 4.1 and are calculate with

$$\text{\% of H}_{\text{com}}\text{ Exchange Applied} = 100\left(\frac{Commanded\ \ Control\ \ Effort}{Applied\ \ Control\ \ Effort}\right). \tag{4.3}$$

Table 4.1 provides quantitative metrics that the different allocation algorithm's relative performance can be compared with. Of particular interest are the 2-norm "Mean % of τ_{com}" and 2-norm "% of Commanded Momentum Exchange Applied" as they provide single value metrics of performance. Higher percentages typically indicate better performance of the allocation algorithm, where the applied torque is approaching the commanded torque. In fact, if no torque limits or saturation speeds were encountered in the maneuver, all of the percentage based metrics in Table 4.1 would be 100%.

It should be noted that while these metrics are a valuable means of performance comparison, they can sometimes be deceiving. Both of these calculations are performed as a percentage of 2-norm applied torques to 2-norm command torques. That is to say that if the 2-norm applied torque grows larger and approaches the 2-norm command torque, the percentage value increases. An increase in the 2-norm applied torque does not however guarantee that the allocation error $\vec{\tau}_{com} - \vec{\tau}$ is being reduced. To alleviate this concern, another comparison metric is introduced in Section 4.2.5.

4.2.5 Allocation Error

Another metric for evaluating the performance of a control allocation algorithm is to evaluate the allocation error. The allocation error is perhaps the most important metric to consider when comparing different allocation methods, since it directly measures the error between the commanded torques $\vec{\tau}_{com}$ and the applied torque $\vec{\tau}$. Figure 4.6 shows the allocation error plotted for the maneuver. The allocation error of the individual axes has not been considered here, but rather the 2-norm of the error $||\vec{\tau}_{com} - \vec{\tau}||_2$ only, so that the overall performance can be evaluated which is arguably more important than the per axis performance.

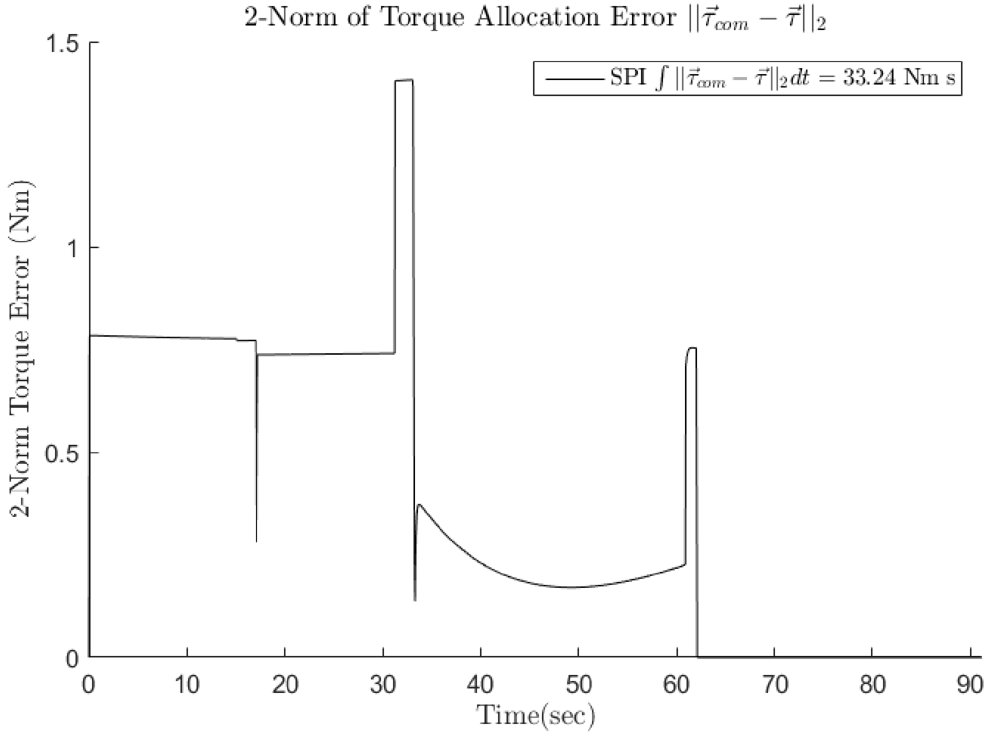

Figure 4.6. 2-norm of the allocation error $\vec{\tau}_{com} - \vec{\tau}$ with simple pseudo inverse allocation

The 2-norm allocation error data shown in Fig. 4.6 can be turned into a single quantitative value that can be used to compare the relative performance of different control allocation algorithms. Such a single value can be computed by integrating the 2-norm allocation error across the maneuver time. The quantity computed from this integration can be thought of as the commanded momentum exchange error, and is calculated as

$$\text{Commanded Momentum Exchange Error} = \int_{t_0}^{t_f} ||\bar{\tau}_{com} - \bar{\tau}||_2 \, dt \tag{4.4}$$

where $\bar{\tau}$ and $\bar{\tau}_{com}$ are time history arrays of the vector of applied torque and command torque respectively. A lower value of this metric indicates better performance of the allocation algorithm. In fact, if not torque constraints or saturation speeds were encountered in the maneuver, the commanded momentum exchange error would be zero. For the SPI allocation method, the commanded momentum exchange error is 33.2 Nm-s.

4.2.6 Power Analysis

A final metric of comparison is an evaluation of the power required by the RWs to complete the maneuver. The power draw metric is not as ideal as some of the other metrics presented as there a numerous variables that can effect these estimations. They can however provide a good qualitative comparison of the performance of the overall system when utilizing the different allocation methods. Table 4.2 contains some important power considerations of the RWA during the maneuver. These estimations are based on calculations provided by the Maxon Motor Company[27]

$$P_{motor} = \frac{n}{C_{MS}}(I) + RI^2 \tag{4.5}$$

$$I = \frac{M}{C_{MT}} + I_{NL} \tag{4.6}$$

where P_{motor} is the power draw of the motor, I is the current draw of the motor, n is the motor speed, M is the motor torque, and C_{MS}, C_{MT}, I_{NL}, and R are various parameters of the motor that are provided in Table 3.1.

Table 4.2. Estimated electrical requirements for the maneuver when utilizing the simple pseudo inverse

Estimated total energy for maneuver	20.5	Watt-Hrs
Estimated total peak power for maneuver	1774.8	Watts
Estimated total peak current for maneuver	48.7	Amps

4.3 Single Maneuver: Redistributed Pseudo Inverse

This section details the results of the simulation using the RPI control allocation method. The results were created in the same way that the results presented in the previous section were created. Therefore, consideration are not given to the methodology of the result generation, as the reader can refer to the previous section for clarification. Important considerations of the results are still discussed, and any newly presented results or metrics are detailed.

4.3.1 Simulation State Variables

Figure 4.7 shows the solution to the state matrix across the duration of the maneuver using the RPI control allocation method. The maneuver was accomplished in 88.25 seconds (3.1% faster than SPI) which includes the 10 second hold on the target within $\pm 1^o$ at the end of the maneuver. Once again, RW 1 and 2 both reached saturation speeds during the maneuver, and are detailed more thoroughly in Section 4.3.3.

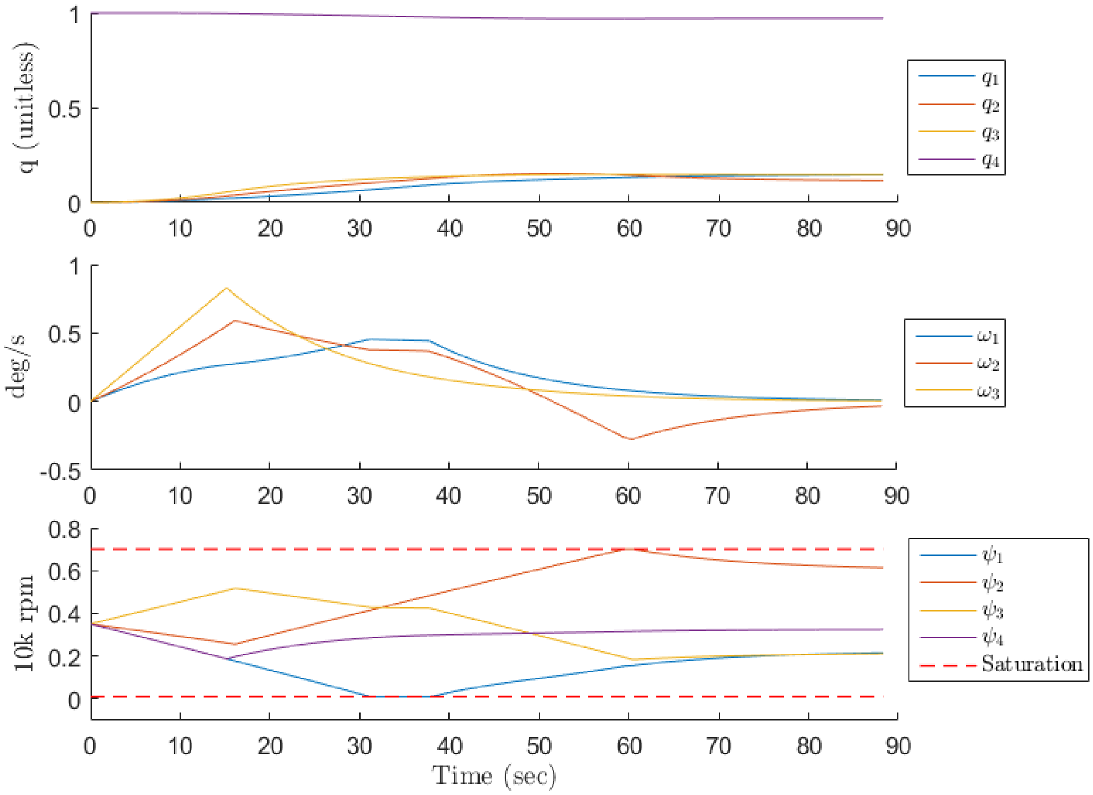

Figure 4.7. Simulation state equation (Eq. (3.11)) solution with redistributed pseudo inverse control allocation

4.3.2 Angular Momentum Analysis

Figure 4.8 plots the magnitude of the angular momentum of the spacecraft (i.e. REBEL) H_{SC}, the angular momentum of the RWA H_{RWA}, and the combined total angular momentum H_{TOTAL}. The total angular momentum remains constant throughout the maneuver, signifying no major issues with the dynamics used in the simulation. The magnitude of the angular momentum of REBEL both begins and ends at zero verifying its initial and final conditions.

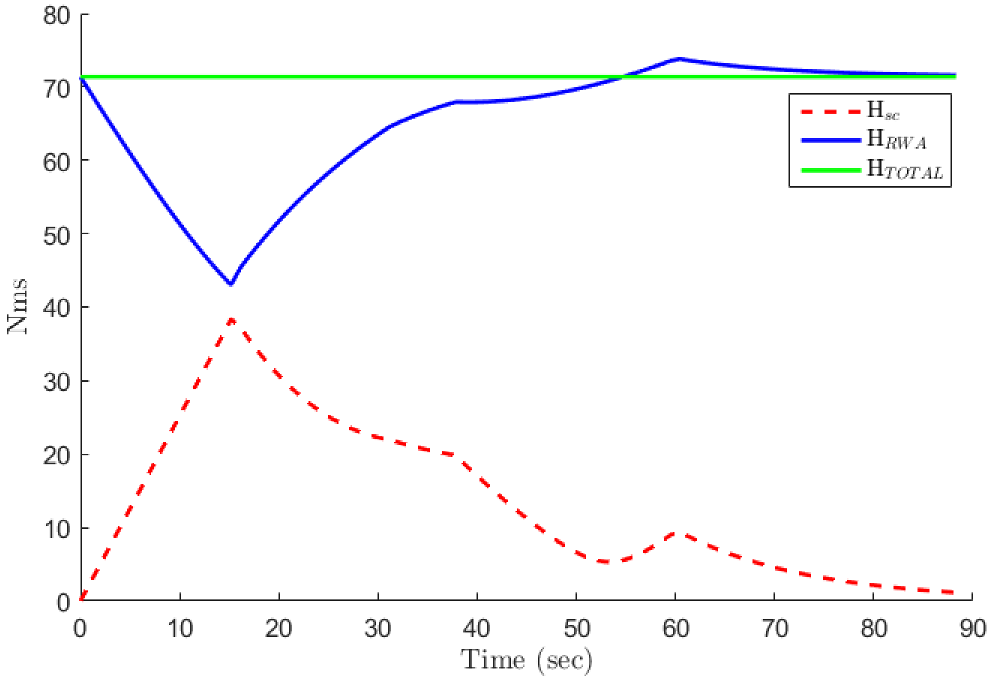

Figure 4.8. Satellite, RWA, and total angular momentum with redistributed simple pseudo inverse

4.3.3 Motor Torques

Figure 4.9 plots the torque produced by each RW motor during the maneuver. Remember that RW 4 is a combination of 3 RWs, and the motor torque plotted for it here represents only 1 of those 3 wheels.

The saturation speed indicator shows that both RW 1 and 2 suffer from saturation speed constraints at some point in the maneuver. RW 1 remains saturated for a total of 6.60 seconds. RW 2 remains saturated for a total of 0.85 seconds. Altogether, the RWs are at saturation speed for a total of 7.45 seconds during the maneuver. This is an increase in the length of time that the RWs are at saturation speed as compared to the SPI method by 144.3%, where the total saturation speed duration was 3.05 seconds.

This result should not be alarming, as the RPI allocation method assigns more torque to RWs than the SPI method does when torque constraints are applied. Figure 4.9 clearly shows that numerous RWs are constrained by their maximum torque limits τ_{max} prior to when the saturation speeds were reached. This increase in allocated torque results in increased wheel acceleration over the time leading up to speed saturation. Increased acceleration rates mean that saturation speeds are reached more readily with the RPI allocation method.

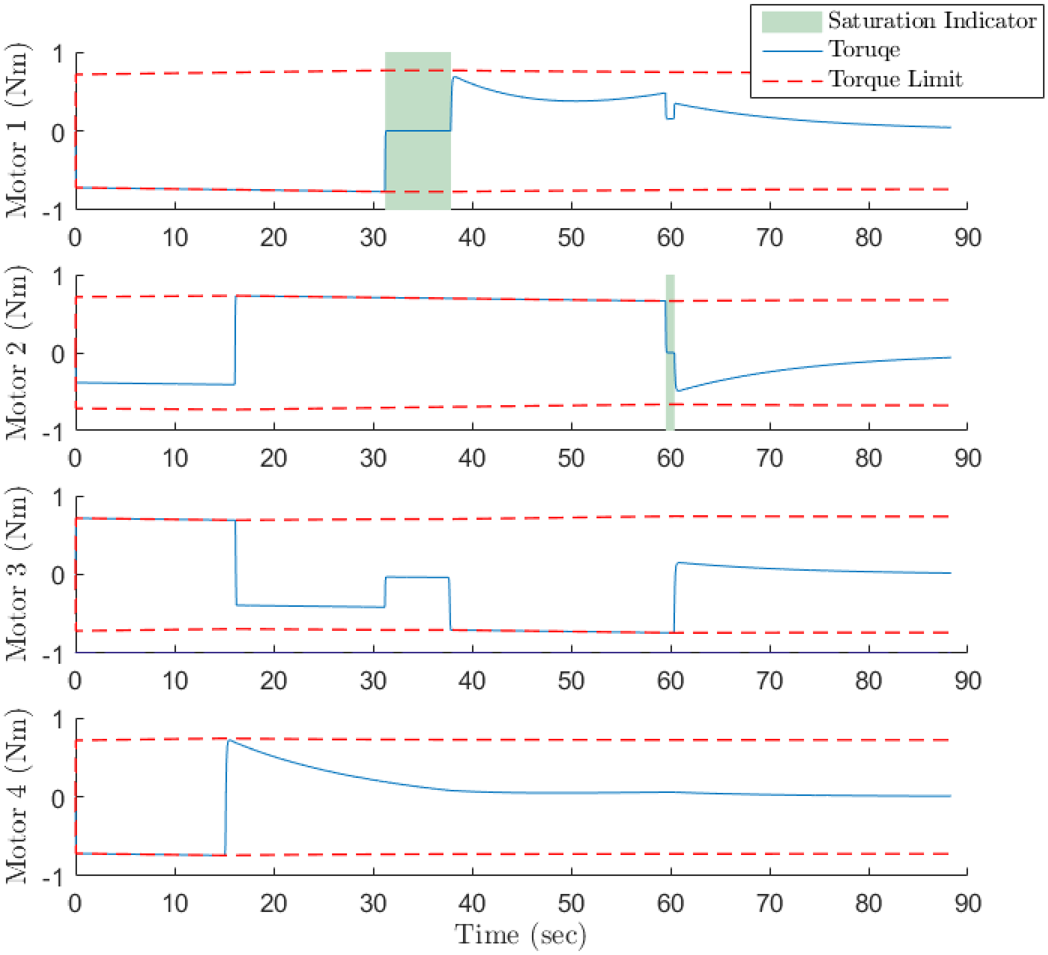

Figure 4.9. RW motor torques and saturation speed constraint indicators (binary) with redistributed pseudo inverse

4.3.4 Commanded vs Applied Torques

Figure 4.10 shows the 3-axis torque commands from the high level controller $\vec{\tau}_{com}$ and the 3-axis torque that is applied to REBEL by the RWA $\vec{\tau}$.

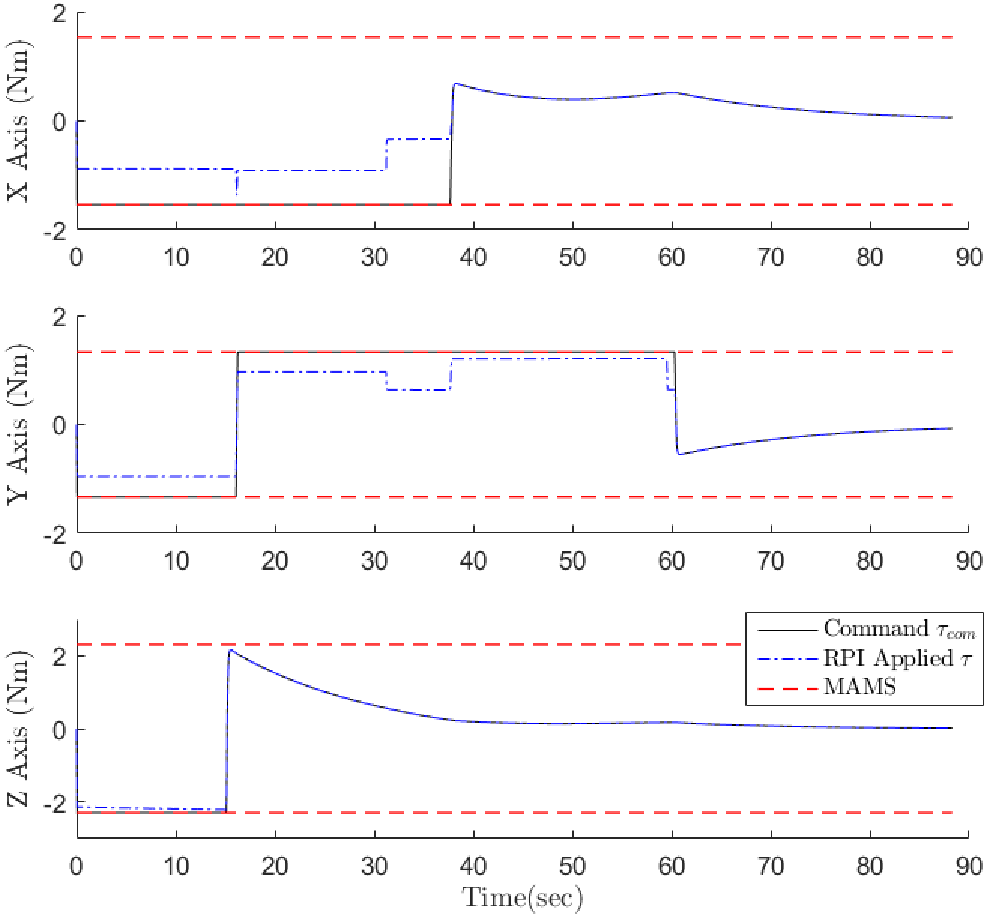

Figure 4.10. Torque commands from high level controller on each axis and torque applied after redistributed pseudo inverse allocation and constraint application

The 2-norm of the command torque vector $||\vec{\tau}_{com}||_2$ and applied torque $||\vec{\tau}||_2$ for the RPI allocation method is computed and plotted in Fig. 4.11.

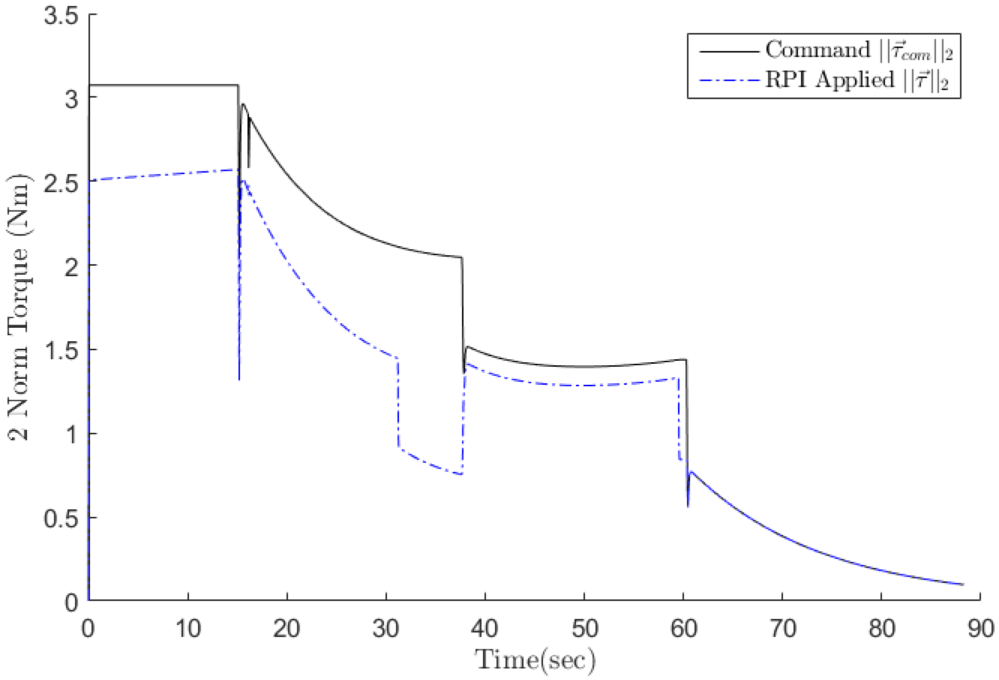

Figure 4.11. 2-norm of torque commands from high level controller and torque applied after redistributed pseudo inverse allocation and constraint application

Table 4.3 shows the important performance metrics calculated from the data in Fig. 4.10 and 4.11. These values were calculated using the same method presented in Section 4.2.4. These metrics are compared against the other allocation methods in Section 4.5.

Table 4.3. Comparison metrics between commanded torques and applied torques for the control system utilizing the redistributed pseudo inverse allocation method

Axis	Mean % of τ_{com} Applied	Commanded Momentum Exchange/Control Effort (Nm-s)	Applied Momentum Exchange/Control Effort (Nm-s)	% of Commanded Momentum Exchange Applied
1	79.33	74.76	46.94	62.79
2	83.59	86.83	67.65	77.91
3	99.04	62.77	60.92	97.06
2-norm	85.87	140.12	111.98	79.91

4.3.5 Allocation Error

The allocation error produced by the RPI method is shown in Fig. 4.12. The methodology used to produce the allocation error metric are covered in Section 4.2.5, but it should be kept in mind that smaller allocation error values indicated better performance of the control allocation algorithm.

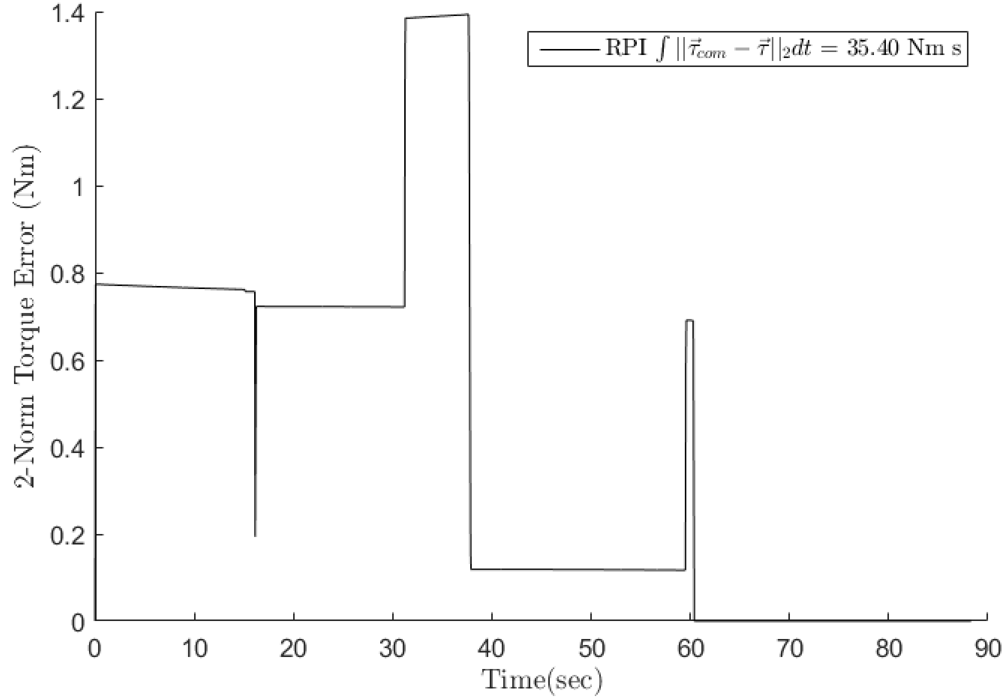

Figure 4.12. 2-norm of the allocation error $\vec{\tau}_{com} - \vec{\tau}$ with redistributed pseudo inverse allocation

The allocation error is integrated across time to form the commanded momentum exchange error, which provides a single valued metrics that can be compared against other allocation methods. The commanded momentum exchange error for the RPI method is 35.4 Nm-s.

4.3.6 Power Analysis

The power analysis introduced in Section 4.2.6 can be applied to the results of the simulated maneuver using the RPI allocation method. These metrics are listed in Table 4.4. These metrics are compared against those produced by the simulated maneuver utilizing other control allocation methods in Section 4.5.

Table 4.4. Estimated electrical requirements for the maneuver when utilizing the redistributed pseudo inverse allocation method

Estimated total energy for maneuver	20.4	Watt-Hrs
Estimated total peak power for maneuver	1826.1	Watts
Estimated total peak current for maneuver	50.5	Amps

4.4 Single Maneuver: Redistributed Pseudo Inverse with Adaptive Weighting

This section details the results of the simulation using the RPIW control allocation method. The results were created in the same way that the results presented Section 4.2 and 4.3 were created. Therefore, consideration are not be given to the methodology of the result generation, as the reader can refer to the previous sections for clarification. Important considerations of the results are still discussed, and any newly presented results or metrics are detailed.

4.4.1 Simulation State Variables

Figure 4.13 shows the solution to the state matrix across the duration of the maneuver using the RPI control allocation method. The maneuver was accomplished in 83.80 seconds (8.0% reduced from the SPI method) which includes the 10 second hold within $\pm 1^o$ of the target at the end of the maneuver. Only RW 1 reached saturation speed during the maneuver, and is detailed more thoroughly in Section 4.4.3.

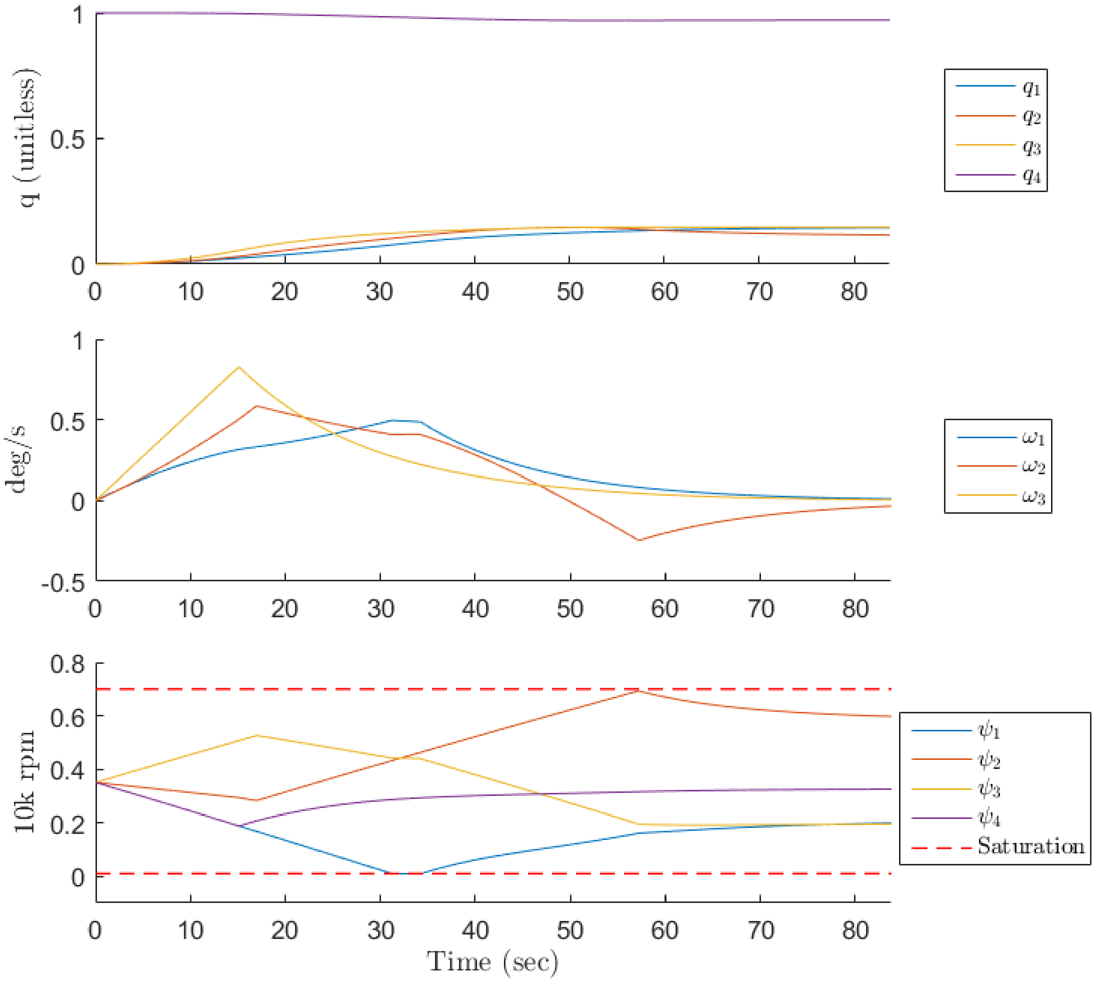

Figure 4.13. Simulation state equation (Eq. (3.11)) solution with adaptively weighted redistributed pseudo inverse control allocation

4.4.2 Angular Momentum Analysis

Figure 4.14 plots the magnitude of the angular momentum of the spacecraft (i.e. REBEL) H_{SC}, the angular momentum of the RWA H_{RWA}, and the combined total angular momentum H_{TOTAL}. The total angular momentum remains constant throughout the maneuver, signifying no major issues with the dynamics used in the simulation. The magnitude of the angular momentum of REBEL both begins and ends at zero verifying its initial and final conditions.

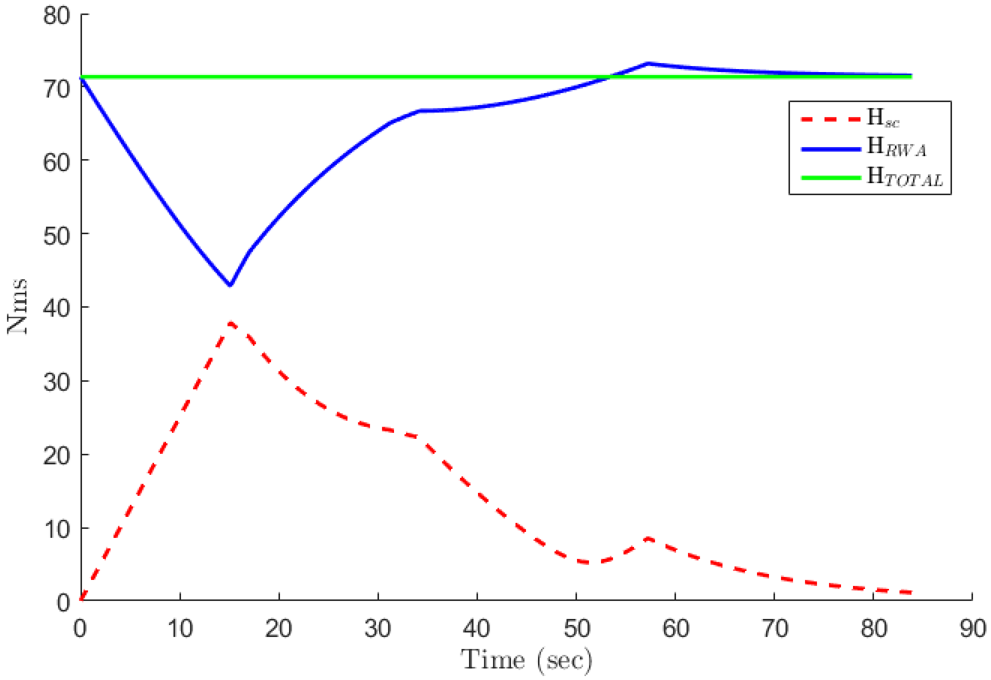

Figure 4.14. Satellite, RWA, and total angular momentum with adaptively weighted redistributed simple pseudo inverse

4.4.3 Motor Torques

Figure 4.15 plots the torque produced by each RW motor during the maneuver. Remember that RW 4 is a combination of 3 RWs, and the motor torque plotted for it here represents only 1 of those 3 wheels.

The saturation speed indicator shows that only RW 1 suffers from saturation speed constraints at some point in the maneuver. RW 1 remains saturated for a total of 2.95 seconds. Altogether, the RWs are at saturation speed for a total of 2.95 seconds during the maneuver. This is clearly a reduction in the length of time that the RWs are at saturation speed as compared to the SPI and RPI method where the total saturation speed duration was 3.05 and 7.45 seconds respectively. The saturation speed duration reduction provides evidence that the weighting matrix is performing as desired, mitigating the amount of time that the RWs are at saturation speed.

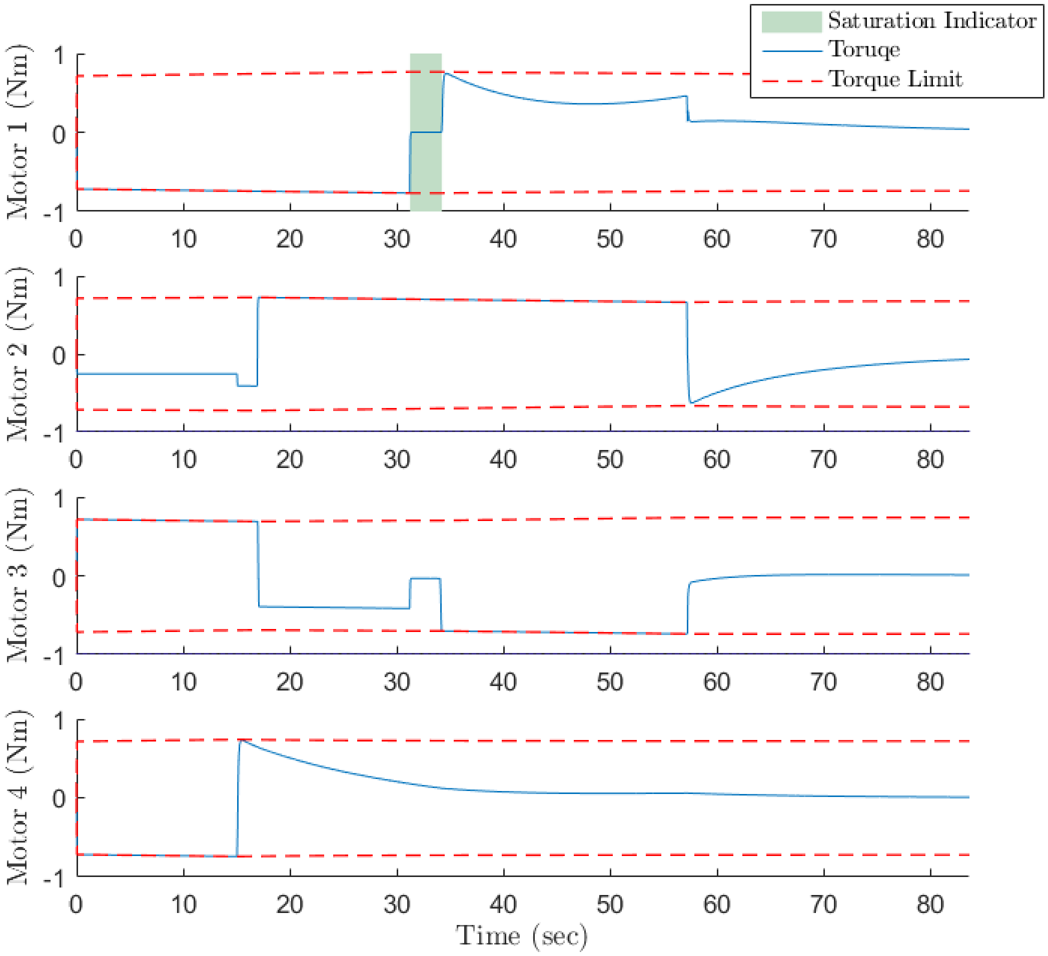

Figure 4.15. RW motor torques and saturation speed constraint indicators (binary) with adaptively weighted redistributed pseudo inverse

4.4.4 Adaptive Weighting Values

Figure 4.16 plots the values of the weighting matrix (Defined by Eq. (3.48)) alongside the corresponding motor speeds. Note that the scale of the weighting matrix is logarithmic. Fig. 4.16 shows that the weighting values are being updated as expected throughout the maneuver. That is, the weights are unity when the corresponding wheel speeds are between their upper and lower saturation boundaries ($\underline{\phi} \leq \psi_i \leq \bar{\phi}$), the weights linearly scale up to 100 when their corresponding RW speeds approach saturation while outside of their saturation bounds, and the weights linearly scale to from 0.01 back to unity when the corresponding RWs are departing saturation while outside of their saturation bounds. The saturation bounds used in this case are $\underline{\phi}$ is $0.2\psi_{max} + 100$ rpm (1500 rpm) and $\bar{\phi}$ is $0.8\psi_{max}$ (5600 rpm.) Recall that a higher weighting value results in less reliance on a particular actuator, while a lower weighting value corresponds to an increased reliance on a particular actuator. For more details on the weighting values, refer back to Section 3.8.3.

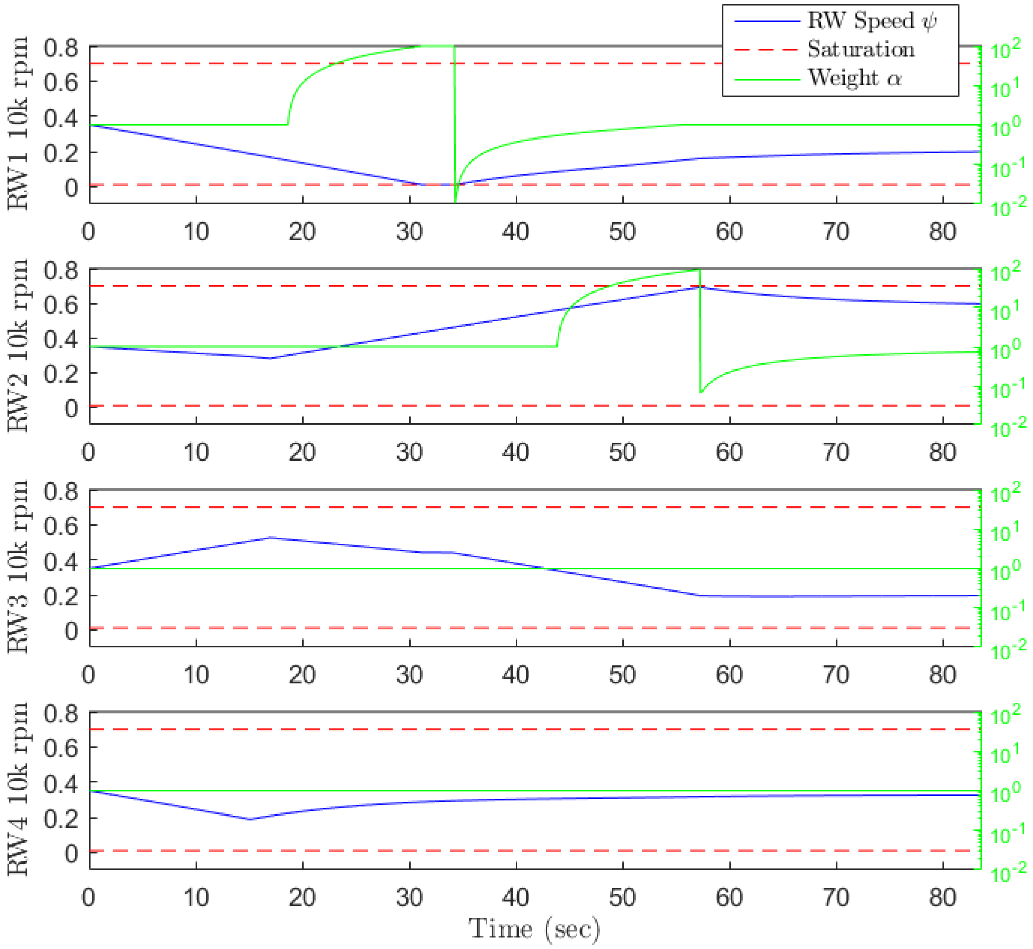

Figure 4.16. Individual RW speeds and their associated weighting values for the adaptively weighted redistributed pseudo inverse allocation method

4.4.5 Commanded vs Applied Torques

Figure 4.17 shows the 3-axis torque commands from the high level controller $\vec{\tau}_{com}$ and the 3-axis torque that is applied to REBEL by the RWA $\vec{\tau}$ for the glsrpiw allocation method.

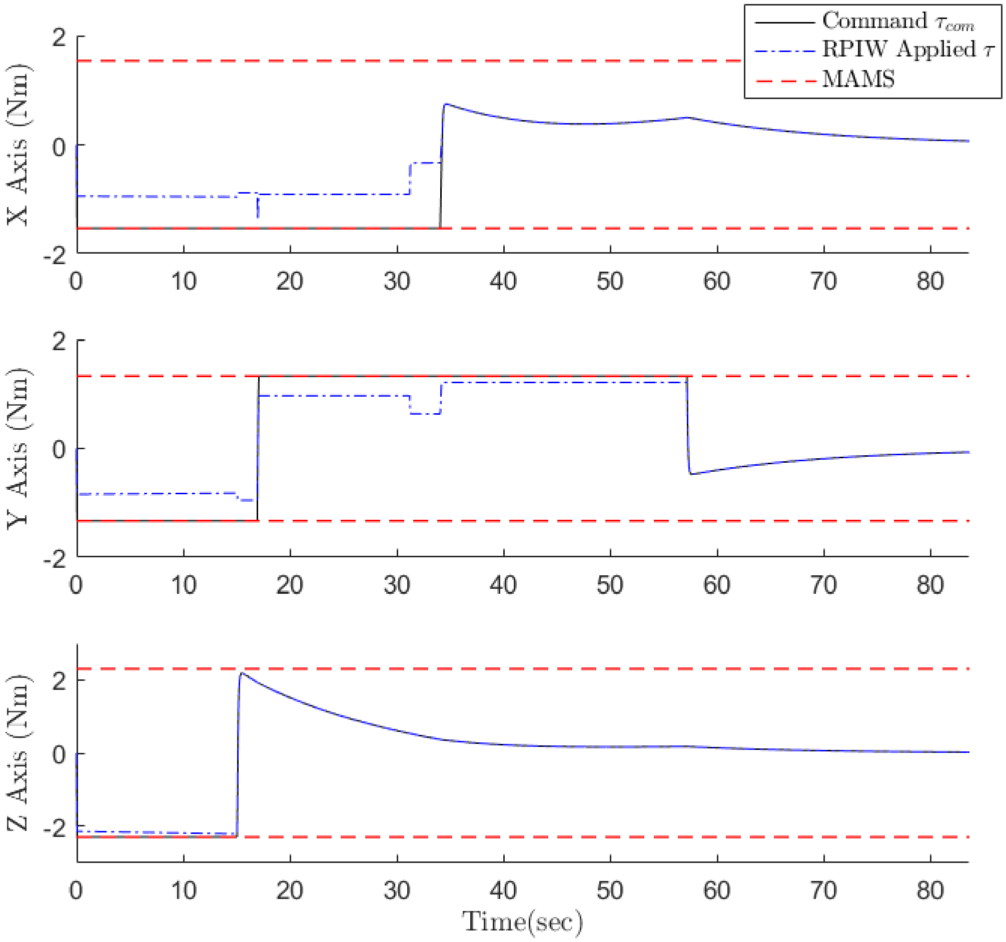

Figure 4.17. Torque commands from high level controller on each axis and torque applied after adaptively weighted redistributed pseudo inverse allocation and constraint application

The 2-norm of the command torque vector $||\vec{\tau}_{com}||_2$ and applied torque $||\vec{\tau}||_2$ for the RPIW allocation method are computed and plotted in Fig. 4.18.

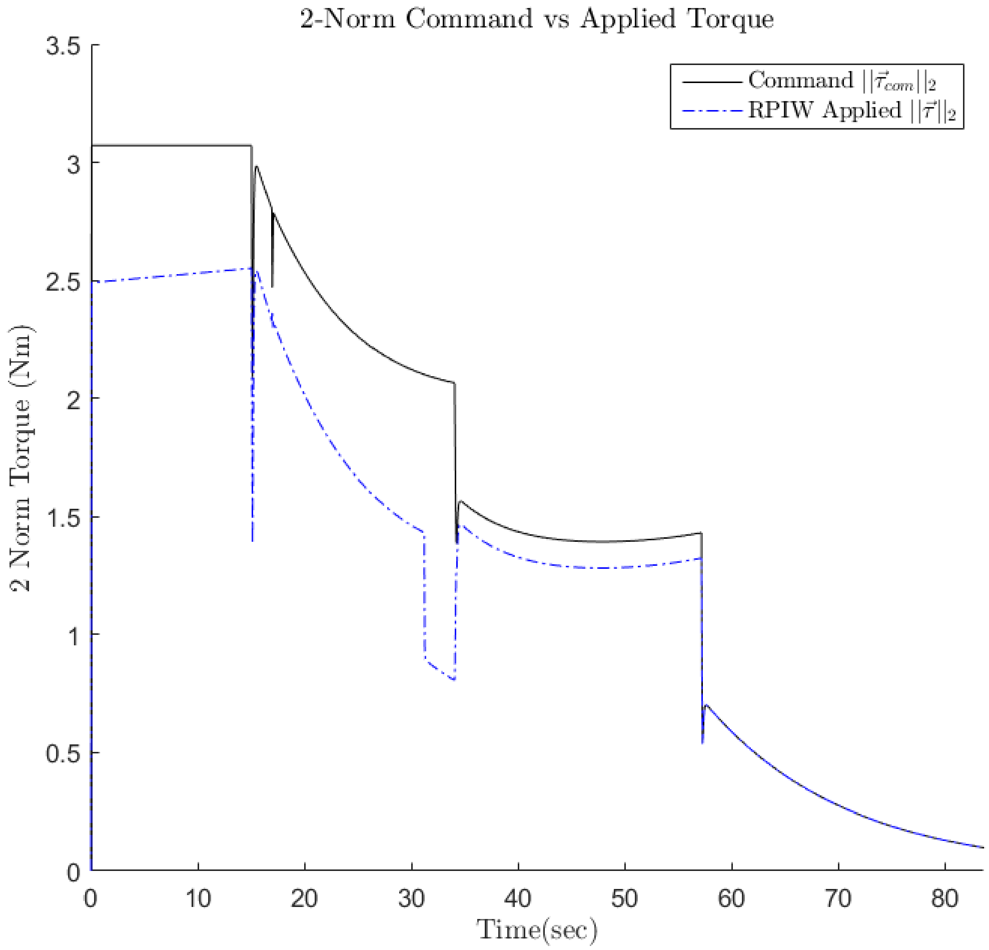

Figure 4.18. 2-norm of torque commands from high level controller and torque applied after adaptively weighted redistributed pseudo inverse allocation and constraint application

Table 4.5 shows the important performance metrics calculated from the data in Fig. 4.17 and 4.18. These values were calculated using the same method presented in Section 4.2.4. These metrics are compared against the other allocation methods in Section 4.5.

Table 4.5. Comparison metrics between commanded torques and applied torques for the control system utilizing the adaptively weighted redistributed pseudo inverse allocation method

Axis	Mean % of τ_{com} Applied	Commanded Momentum Exchange/Control Effort (Nm-s)	Applied Momentum Exchange/Control Effort (Nm-s)	% of Commanded Momentum Exchange Applied
1	82.4	68.9	46.5	67.4
2	83.7	81.6	63.5	77.8
3	99.0	62.5	60.6	97.1
2-norm	87.9	132.4	108.9	82.2

4.4.6 Allocation Error

The allocation error produced by the RPIW method is shown in Fig. 4.19. The methodology used to produce the allocation error metric are covered in Section 4.2.5, but it should be kept in mind that smaller allocation error values indicated better performance of the control allocation algorithm.

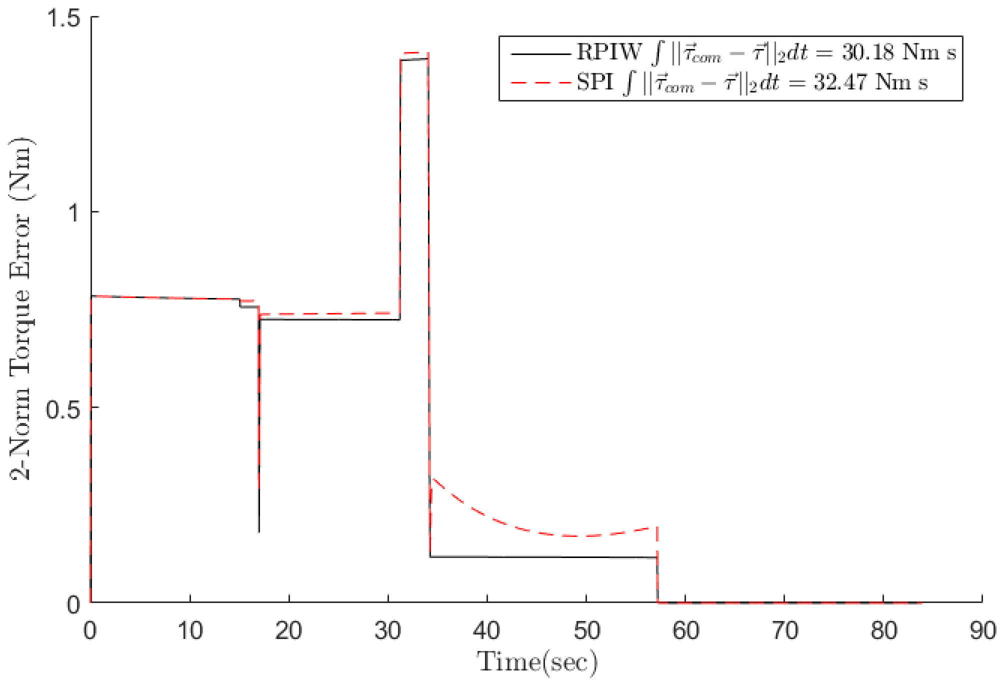

Figure 4.19. 2-norm of the allocation error $\vec{\tau}_{com} - \vec{\tau}$ with adaptively weighted redistributed pseudo inverse allocation

The allocation error is integrated across time to form the commanded momentum exchange error, which provides single valued metric that can be compared against other allocation methods. The commanded momentum exchange error for the RPIW method is 30.2 Nm-s. The commanded momentum exchange error for the RPIW method is compared against the allocation methods in Section 4.5.

4.4.7 Power Analysis

The power analysis introduced in Section 4.2.6 can be applied to the results of the simulated maneuver using the RPIW allocation method. These metrics are listed in Table 4.6. These metrics are compared against those produced by the simulated maneuver utilizing other control allocation methods in Section 4.5.

Table 4.6. Estimated electrical requirements for the maneuver when utilizing the adaptively weighted redistributed pseudo inverse allocation method

Estimated total energy for maneuver	20.0	Watt-Hrs
Estimated total peak power for maneuver	1774.8	Watts
Estimated total peak current for maneuver	50.1	Amps

4.5 Single Maneuver: Allocation Method Comparison

4.5.1 Maneuver Time

Table 4.7 shows the simulated maneuver time for the cases where the simulation used the SPI, RPI, and RPIW control allocation methods. These times include a 10 second hold time at the end of the maneuver. The simulation utilizing the SPI allocation method had the longest maneuver time. The simulation utilizing the RPI method placed second with a 3.1% reduction in time. This reduction in maneuver time appears to be a results of the redistribution process of the RPI method which mitigates against actuator constraints. The RPIW had the fastest maneuver time, which is 8.0% faster than the SPI method and 5.0% faster than the RPI method.

This reduction in maneuver time appears to be due to the RPIW allocation methods ability to both mitigate against actuator constraints, and saturation speeds. The mitigation of saturation speed duration allowed the otherwise constrained RWs to contribute to the applied torque for a larger portion of the maneuver, resulting in better allocated command torques. A variety of maneuvers is later evaluated to help validate this trend.

Table 4.7. Maneuver time including 10 second hold period for each allocation method

SPI	RPI	RPIW
91.10 seconds	88.25 seconds	83.80 seconds

4.5.2 Reaction Wheel Saturation Duration

During the maneuver, all of the control allocation methods produced some amount of reaction wheel saturation on RWs 1 and 2. The presence of reaction wheel saturation should not be interpreted as a poor reflection on the control allocation methods though, since many of the simulation parameters were selected to encourage RW speed saturation. Figure 4.20 shows the speed saturation status of RW 1 and 2 as binary functions (high corresponds to speed saturated wheels, low corresponds to unsaturated wheels.) The time scales have been shortened to better depict the different saturation speed durations. Note that no RW speed saturation took place outside of the depicted timescales. Figure 4.20 is presented as a method of visual comparison of saturation durations observed when using each control allocation method. Conclusions should not be drawn from the particular timing of the RW saturation statuses, as the RW speeds evolve differently during the maneuver when the different control allocation methods are implemented

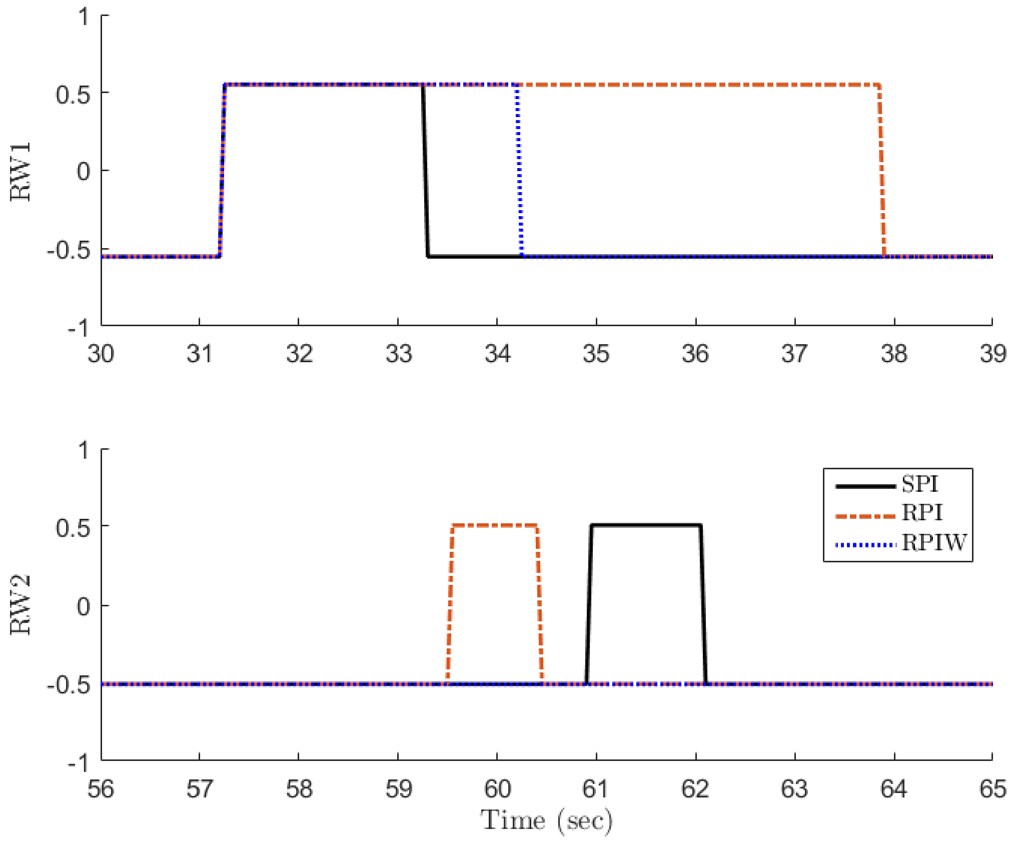

Figure 4.20. Saturation speed status of reaction wheels 1 and 2. Time scales have been matched in length.

Table 4.7 summarizes the speed saturation duration of the RWs depicted in Fig. 4.20. The RPI method had an increase in total duration from the SPI method of 59.1%. Such in increase is not unexpected, as outline in Section 4.3.3. The RPIW method decreased the total speed saturation duration by 3.3% from the SPI method (while generating more applied torque), and by 60.4% from the RPI method.

Table 4.8. Duration of RW saturation speed constraint in seconds.

	SPI	RPI	RPIW	
RW1	1.95	6.60	2.95	sec
RW2	1.10	0.85	0	sec
$\Sigma \mathrm{RW}_i$	3.05	7.45	2.95	sec

4.5.3 Commanded vs Applied Torque

Table 4.9 Compares the "mean % of τ_{com} applied" and the "% of commanded momentum exchange applied" for the simulation while utilizing the SPI, RPI, and RPIW methods for control allocation. Of particular interest is the row of 2-norm values, as it contains single values that can be compared between the methods. These results do not offer as clear of a picture of the performance of the RPI and RPIW methods as compared to the SPI methods, since the primary goal is the reduction of allocation error. A larger "mean % of τ_{com} applied" and "% of commanded momentum exchange applied" do not necessarily correlate with a reduction in allocation error. Furthermore, these results are produced for a single maneuver. An additional set of maneuvers is later evaluated, but these metrics are omitted from the analysis as they likely do not offer a good comparison between different maneuvers.

Table 4.9. Comparison of metrics between commanded torques and applied torques for the different control allocation methods

	SPI		RPI		RPIW	
Axis	Mean % of τ_{com} Applied	% of Commanded Momentum Exchange Applied	Mean % of τ_{com} Applied	% of Commanded Momentum Exchange Applied	Mean % of τ_{com} Applied	% of Commanded Momentum Exchange Applied
1	80.6	67.6	79.3	62.8	82.4	67.4
2	81.5	75.3	83.6	77.9	83.7	77.8
3	99.1	97.1	99.0	97.9	99.0	97.1
2-norm	86.5	81.2	85.9	79.9	87.9	82.2

The simulation utilizing the RPIW allocation method had the best "mean % of τ_{com} applied", meaning the 2-norm of its applied torques matched the 2-norm of the commanded torque better than the other 2 methods for the greatest duration of time. The RPI method came in last in the "mean % of τ_{com} applied" metric, placing the SPI method in second. Recall that the "mean % of τ_{com} applied" metric can be somewhat misleading though, as increasing the value of the 2-norm applied torque closer to the 2-norm value of the commanded torque does not necessarily correlate with a reduction in allocation error.

Similar results were obtained for the comparison of the "% of commanded momentum exchange applied" metric. This metric represents the total momentum exchange that was applied by the control system using the various allocation methods as a percentage of the commanded total momentum exchange. Once again, the RPIW method scored the best, followed by the SPI method, and the RPI method in last. The "% of commanded momentum exchange applied" metric suffers from the same issue the "mean % of τ_{com} applied" metric suffers from. A higher percentage does not necessarily correlate with reduction in allocation error, which is the primary goal of control allocation.

4.5.4 Allocation Error

The reduction of allocation error $||\vec{\tau}_{com} - \vec{\tau}||_2$ is the primary metric for which the control allocation methods are gauged with. By integrating the allocation error over time, the commanded momentum exchange error is produced, which is a single value that can be compared between the allocation methods. Table 4.10 shows the values for this metric produced from the simulation while utilizing the various allocation methods. It should be kept in mind that the values presented are for a single maneuver. A range of maneuvers is later examined from which more authoritative general conclusions can be drawn.

Table 4.10. Comparison of the momentum exchange differences between the different allocation methods

	SPI	RPI	RPIW	
Commanded momentum exchange error	33.2	35.4	30.2	Nm-s

The RPIW method had the lowest commanded momentum exchange error (total allocation error), follwed by the SPI method, and the RPI method. The RPI method's high commanded momentum exchange exchange error (higher the SPI) is likely due to the increased duration of time its RWs were at saturation speed. The increased saturation speed duration caused a degradation of the control system that the redistribution process was not able to effectively overcome. This result indicates that RW speed saturation is a particularly harmful constraint. This is verified in Fig. 4.21, where the 2-norm allocation error of the simulated maneuver utilizing the RPI method is plotted with RW saturation indicators. Notice that when a RW becomes saturated, the allocation error is significantly increased. This indicates a correlation between the amount of time that RWs are saturated and the commanded momentum exchange error, or total allocation error.

Also of note, the relative difference between the commanded momentum exchange error values in Table 4.10 is not particularly large. This lack of contrast should not be of great concern for this particular simulation however. The high level controller was designed to command torques that are largely unachievable due to the RW motor torque constraints (recall the formulation of the MAMS in Section 3.6.) The unachievable command torques leads to inflated allocation errors, since even a perfect allocation method (if one were to exist) would not be able to achieve the commanded torques a majority of the time. A more thoughtfully designed high level controller might provide more realistic torque commands that would bring the overall commanded momentum exchange errors down, and increase their relative differences.

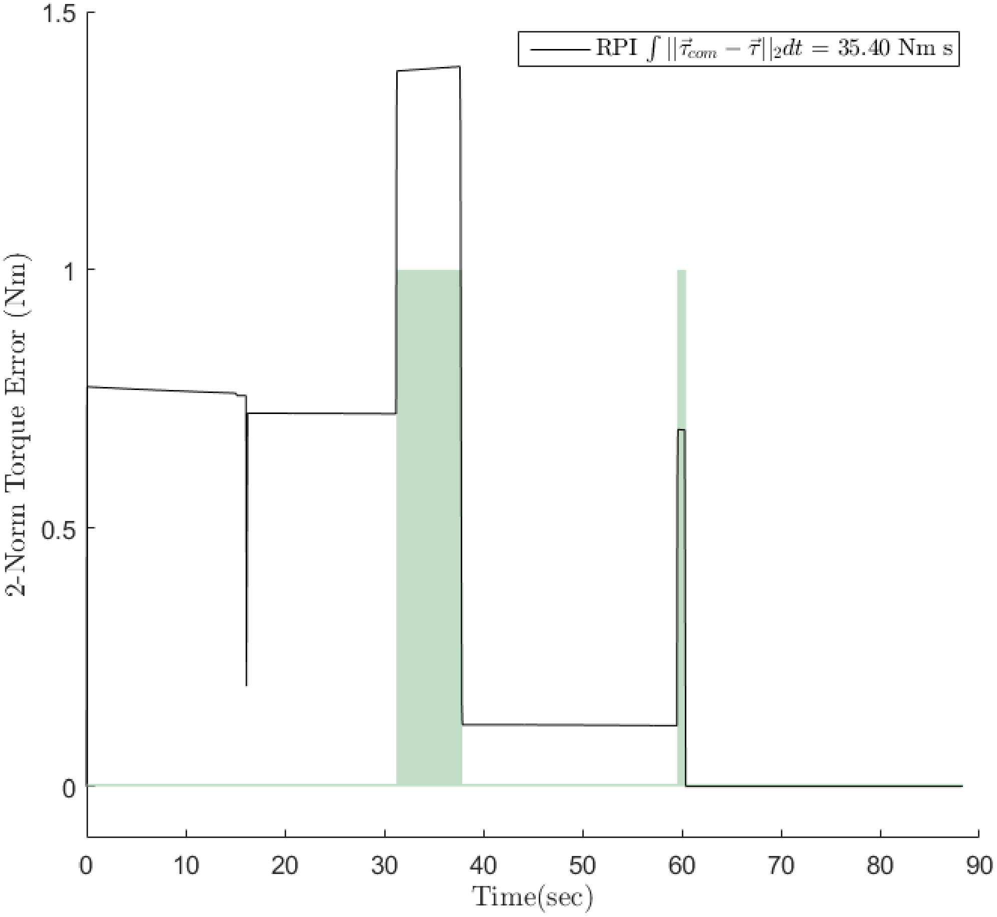

Figure 4.21. 2-norm of the allocation error $\vec{\tau}_{com} - \vec{\tau}$ with redistributed pseudo inverse allocation and binary indicator for saturation speed status of RW 1 and 2

4.5.5 Power Analysis

Finally, Table 4.11 compares the estimated power requirements of the control system utilizing the various control allocation methods. Keep in mind that these values have been produced from a single maneuver, and they will be later evaluated for a larger range of simulated maneuvers.

Table 4.11. Comparison of power and energy requirements of the maneuver for the different control allocation methods

Allocation method	SPI	RPI	RPIW	
Estimated total energy for maneuver	20.5	20.4	20.0	Watt-Hrs
Estimated total peak power for maneuver	1774.8	1826.1	1774.8	Watts
Estimated total peak current for maneuver	48.7	50.5	50.1	Amps

The RPIW method required the least amount of energy to complete the maneuver, followed by the RPI method, and the SPI method in last. The RPI method required the largest peak power draw, followed by the RPIW and SPI methods which had identical peak power draws. The RPI method also had the highest peak current draw, followed by the RPIW method, and the SPI method in last. Even though the RPI and RPIW methods appeared to draw more power than the SPI method, they required less energy to complete their maneuvers. Though the reduction in energy seems contradictory, recall that these methods also greatly shorted the maneuver time. Their maneuver times were shortened sufficiently enough that despite having higher power draws, they still used less energy.

4.6 Multiple Maneuver Summary

A set of 343 maneuvers (described in Section 3.5) was simulated using each of the control allocation methods. Recall that this set of maneuvers is still a somewhat limited analysis, as the slew angles had to be kept small to prevent unstable behavior of the system due to the linear feedback PD controller used in the control loop. The most relevant performance metrics were computed for each allocation method, and the RPI and RPIW method were compared against the SPI method as a percentage. These percentages were averaged across the 343 maneuvers to provide the values shown in Table 4.12. Because the RPI and RPIW methods are intended to improve upon the SPI method (which is considered the baseline allocation method) by mitigating against constraints, this percentage comparison was considered a valid and concise method for presenting the results of the set of maneuvers. The full set of simulation results for the SPI, RPI, and RPIW can be found in Appendix A, B, and C respectively.

Table 4.12. Performance comparison of RPI and RPIW allocation methods as averaged percentage of SPI for 343 simulations

	Maneuver Time	Total RW Saturation Time	Commanded Momentum Exchange Error	Total Consumed Energy
RPI	-0.3%	6.6%	-6.0%	1.5%
RPIW	-0.2%	2.2%	-6.3%	0.9%

On average, there was not a significant enough reduction in maneuver time for the RPI or RPIW to conclude that these methods were able to accomplish their maneuvers faster than the SPI method. This result is likely due to the less than ideal linear PD controller used in the control loop. If the high level controller's command torques are not ideal, reducing the allocation error will not necessarily result in reduced maneuver time.

On average, the RPI and RPIW method slightly increased the duration of time that the RWs were at saturation speed. This results is expected as the redistribution process inherent to the RPI and RPIW methods applies additional torque to the RWs as compared to the SPI when actuator constraints are encountered. On average, the RPIW method was able to limit this increase in saturation duration by 4.4% from the RPI method. This reduction provides some evidence that the RPIW is able to mitigate against RW saturation.

On average, the commanded momentum exchange error of the RPI and RPIW was approximately 6% less than the commanded momentum exchange error produced by the SPI method. This figure appears to be significant enough to conclude that on average, for the limited set of maneuvers examined, the RPI and RPIW methods were able to mitigate against the effects of RW constraints and better match the applied torque $\vec{\tau}$ to the high level controller's command torque $\vec{\tau}_{com}$.

On average, the total energy consumed for the maneuver appeared to be slightly, but not significantly, increased from the energy consumed for the SPI method. Recall that on average, the RPI and RPIW methods reduced the allocation error while failing to reduce the maneuver time significantly (most likely a result of the non ideal controller.) When considering this fact, a small increase in average energy consumed would not be surprising.

4.7 Multiple Maneuver Comparison to Single Maneuver

The single maneuver analyzed in this research provided valuable insight into the RPI and RPIW methods' functionality as compared to the SPI method. The single maneuver did not provide enough evidence however to draw general conclusions about the gross performance of the RPI and RPIW method with any certainty. In order to draw conclusions more confidently, a series of multiple maneuvers was performed and analyzed in the previous section. Table 4.13 compares the same metrics presented for the multiple maneuver analysis against the results of the single analysis. From this comparison, some insight can be drawn about the interaction between the control allocation method and the rest of the control loop.

The reduction in maneuver times for the RPI and RPIW methods over the SPI method observed in the single maneuver was not observed in the averaged set of maneuvers. This indicates that the maneuver time is more than a function of the allocation method alone. The high level controller and other factors in the control loop will cause the maneuver time to vary based on the maneuver in question.

The drastic increase in total RW saturation time observed for the RPI method in the single maneuver is likely not typical. The averaged maneuver set results do indicate however that the RPI method likely does produce some significant significant

Table 4.13. Side by side comparison of the single maneuver and the 343 averaged maneuvers

Single Maneuver

	Maneuver Time	Total RW Saturation Time	Commanded Momentum Exchange Error	Total Consumed Energy
RPI	-3.1%	144.3%	8.9%	-0.5%
RPIW	-8.0%	-3.3%	-7.1%	-2.4%

343 Averaged Maneuvers

	Maneuver Time	Total RW Saturation Time	Commanded Momentum Exchange Error	Total Consumed Energy
RPI	-0.3%	6.6%	-6.0%	1.5%
RPIW	-0.2%	2.2%	-6.3%	0.9%

increase of RW saturation duration - just not nearly as extreme as would have been indicated by the single maneuver result. The single maneuver results does however indicate that large increases in saturation durations are possible.

Although the single maneuver showed an increase in commanded momentum exchange error for the RPI method over the SPI method, the averaged set of results indicated that the RPI method likely decreases the commanded momentum exchange error. Recall from Section 4.5.4 that the allocation error is related to the duration of time the RWs are saturated for. The drastic increase in RW saturation time observed for the RPI method during the single maneuver likely led to the increase in commanded momentum exchange error. This shows that it is possible for the commanded momentum exchange error associated with the RPI method (and possibly the RPIW as well) to be increased if the RWs are saturated for long periods of time, but it is not the average case. It is hoped that the RW saturation prevention technique applied in the RPIW method reduces the chances of this occurrence.

Finally, the slight reduction in total consumed energy observed for the RPI and RPIW method in the single maneuver is not what should be expected on average. The reduction in consumed energy for that particular maneuver was likely a result of the reduced maneuver time. In the averaged results, maneuver time was not significantly reduced for either method, nor was the energy consumed.

4.8 Summary

This chapter presented the results of the simulation described in Chapter III. Results for a single simulated maneuver where the three different allocation methods are utilized were presented both independently, and then compared against each other. Because the single maneuver did not provide sufficient evidence to draw gross conclusions about the control allocation methods, a set of 343 simulated maneuvers was performed for each allocation method and the results were averaged. The results seem to show promise that the RPI and RPIW are viable approaches for reducing allocation error as compared to the SPI method. The RPIW method also shows promise in regards to reducing the amount of time the RWs remained at saturation speed in comparison to the SPI method. It should be noted however that the range of maneuvers examined is somewhat limited. Conclusions drawn from these results are presented in the next chapter.

V. Conclusions and Recommendations

5.1 Introduction

This chapter draws conclusions about the results presented in the previous chapter. The results of the simulations using the SPI, RPI, and RPIW allocation methods are summarized in in their own sections. These sections include a brief discussion of the control allocation methodology, the notable results from the analysis of the single maneuver, and the characteristics determined from the multiple maneuver analysis. Recall that the performance of the SPI method is considered as a baseline of comparison for the RPI and RPIW methods. Brief final remarks are presented about the allocation methods and the research overall. Finally, recommendations for future work are outlined.

5.2 Simple Pseudo Inverse

The SPI control allocation method serves as a basis of comparison for the other control allocation methods. The SPI method is commonly used when an independent control allocation method is desired due to its simplicity. The SPI method calculates a control allocation solution by minimizing the cost of control action. The SPI method does not however include any mitigations against actuator constraints. The allocation method makes no consideration of actuator constraints when allocating the commands from the high level controller. This lack of consideration for actuator constraints results in degraded performance for physical systems, since all actuators have some form of constraints by nature.

The results presented in Section 4.2 show various metrics describing the performance of the simulated control system while utilizing the SPI allocation method. The absolute performance is not of particular consideration, as there are many variables that could be adjusted to improve it. Rather, these metrics are used as a baseline of comparison for the RPI and RPIW allocation methods.

The SPI control allocation method should only be considered in situations where the actuators are not subject to constraints (rare, or non-existent in physical systems), or where the allocated forces are small enough in comparison to the actuator constraints that reaching the constraints is improbable or infrequent. In fact, there is little to be lost by adopting the RPI method in place of the SPI method, since the first step of the RPI method is the solution to the SPI method. The RPI method only continues into the redistribution algorithm if actuators have become constrained. By only using the redistribution algorithm when required, even the computational cost of the RPI method is no greater than that of the SPI method unless actuators reach constraints.

5.3 Redistributed Pseudo Inverse

The RPI method provides a solution to the control allocation problem by minimizing control effort while also providing mitigation against the detrimental effects of actuator constraint with an iterative redistribution process. It first uses the SPI method to generate a solution. Then it performs a check to see if any of the actuators have become constrained. If actuators are constrained, they become isolated and the remaining unconstrained actuators are redistributed in an attempt to eliminate the torque error created by the constrained actuators. This process is repeated until all actuators have become constrained, the commanded torque is met, or no more beneficial changes can be made to the remaining unconstrained actuators.

The RPI method's performance was analyzed in detail for a single simulated maneuver. This analysis provided some insight into the functionality of the RPI method, but the results of the single maneuver were not substantial enough to draw gross conclusions from. Of the more notable conclusions drawn from this single maneuver analysis was that the control system appeared to experience severely degraded performance when its RWs became saturated. This performance impact was observed as an increased commanded momentum exchange error. Since the RPI method makes no consideration of RW speed when allocating command torques, it may not be suitable for applications where RW speeds are easily reached.

To gain a better understanding of the allocation algorithm's gross performance, a series of 343 maneuvers was simulated and the results averaged. It should be noted that even this set of maneuvers is not sufficient to characterize the RPI allocation method's performance for all maneuvers. This is especially true since the slews in the set of maneuvers had to remain small to keep the high level linear feedback controller from growing unstable. The results of this set of maneuvers does however provide a much better estimate of the RPI method's gross behavior than the single maneuver simulation.

On average, the RPI method reduced the maneuver time from the SPI method by 0.3%. This is not a significant reduction in maneuver time, but it should be kept in mind that the high level controller was not ideal, and an improved allocation of its commands does not necessarily translate to reduced maneuver time. The RPI method increased the total RW saturation duration from the SPI method by 6.6% on average. Such in increase is expected, and should be taken into account when considering the RPI method for control allocation applications. On average, the RPI allocation method reduced the allocation error from the SPI method by 6.0%. This is perhaps the most important performance metric, as one of the primary goals of control allocation is the reduction of allocation error. This result shows that the RPI method has promise as an allocation method that could be utilized on REBEL. Finally, the RPI method caused a slight increase in the total energy consumed of 1.5%

over the SPI method on average. This increase is nearly negligible, and it is possible that it could be reduced if a more ideal high level controller were used in place of the linear feedback controller used in this simulation.

5.4 Redistributed Pseudo Inverse with Adaptive Weighting

The RPIW method provides a solution to the control allocation problem by minimizing control effort, mitigating against the detrimental effects of actuator constraints with an iterative redistribution process, and by taking RW speed into consideration when allocating command torques to prevent saturation speeds from being reached when possible. It functions the same as the RPI method, but uses an adaptive weighting matrix, rather than an identity matrix, when computing a solution to the minimization of the control effort cost function (Eq. (2.22).) The values of the weighting matrix are based on the RW speeds, and chosen such that RWs are relied on less when they are approaching saturation and relied on more when they are leaving saturation. In fact, the RPIW becomes identical to the RPI method when none of its RWs are near saturation.

The RPIW method's performance was analyzed in detail for a single simulated maneuver. This analysis provided some insight into the functionality of the RPIW method, but the results of the single maneuver were not substantial enough to draw gross conclusions from. Of the more notable conclusions drawn from this single maneuver analysis was that the RPIW method reduced the amount of time that the RWs were saturated for, as compared to the RPI method performing the same maneuver, by 59%. This result is significantly large enough to suggest that the RPIW method does provide some mitigation against RW saturation in some cases.

To gain a better understanding of the allocation algorithm's gross performance, a series of 343 maneuvers was simulated and the results averaged. It should be noted that even this set of maneuvers is not sufficient to characterize the RPIW allocation method's performance for all maneuvers. This is especially true since the slews in the set of maneuvers had to remain small to keep the high level linear feedback controller from growing unstable. The results of this set of maneuvers does however provide a much better estimate of the RPIW method's gross behavior than the single maneuver simulation.

On average, the RPIW method reduced the maneuver time from the SPI method by 0.2%. This is not a significant reduction in maneuver time, but it should be kept in mind that the high level controller was not ideal, and an improved allocation of its commands does not necessarily translate to reduced maneuver time. The RPIW method increased the total RW saturation duration from the SPI method by only 2.2% on average. This is a reduction of 4.4% from the average increase of RW saturation duration produced by the RPI method. This is a good indication that the RPIW method may be a more appropriate choice as compared the the RPI method when RW saturation is of particular concern. On average, the RPIW allocation method reduced the allocation error from the SPI method by 6.3%. This is perhaps the most important performance metric, as one of the primary goals of control allocation is the reduction of allocation error. This result shows that the RPIW method has promise as an allocation method that could be utilized on REBEL. Finally, the RPIW method caused a slight increase in the total energy consumed of 0.9% over the SPI method on average. This increase is nearly negligible, and it is possible that it could be reduced if a more ideal high level controller were used in place of the linear feedback controller used in this simulation.

It is important to note that the inclusion of the weighting matrix may reduce the effectiveness of the allocation method's ability to minimize control effort. This reduction in control effort minimization occurs because the adaptive weights skew the actuators' contribution to the total cost being minimized. For instance, the adaptive weighting matrix may encourage a less effective actuator to be used more heavily if the more effective actuators are approaching constraint. This is a tradeoff of control effort minimization for reaction wheel saturation speed mitigation. However, if the duration of time that the RWs are saturated for is decreased, it may lead to a performance improvement of the control system since it is able to use the otherwise constrained actuators for a longer duration of time.

It should also be briefly noted that the dynamic weighting values used in this simulation were designed for the case where the lower boundary for RW operating speed is 100 rpm. That is, zero crossings of the RWs were not permitted. Should RW zero crossings be desired the dynamic weighting scheme would need revision. Care should be taken when selecting weighting values at low speeds to prevent oscillatory behavior around a local minimum in the cost function solution. The study of this revised weighting matrix is left for future research. The research presented in this thesis however does provides a proof of concept for the RPIW allocation method.

5.5 Final Remarks

Both the RPI and RPIW methods appear to show some promise as suitable control allocation methods for REBEL and potentially other over actuated spacecraft simulators or satellites utilizing RWAs. Both methods reduced the average allocation error across 343 simulated maneuvers, though those maneuvers were limited to small slew angles. Additionally the RPIW allocation method showed a reduction in the average amount of time its RWs were saturated as compared to the RPI method across the same set of maneuvers. Either method shows promise, but the RPIW method should be given consideration over the RPI method for RWA designs where RW saturation is of greater concern.

5.6 Future Work

It would be valuable to extend the range of the simulations analyzed to cover a broader swath of values. This research only analyzed small slew maneuvers as a result of the high level controller used. With a more appropriate high level controller selection (a non linear controller would likely work particularly well) the range of simulated maneuvers could be greatly expanded. Furthermore, the initial conditions of the RW angular velocities should be varied as well to characterize their impact.

Future research should also include the eventual validation and calibration of the results presented here on the REBEL test bed. Once the REBEL RWA design is implemented, the allocation methods described here could be implemented in REBEL's control system and further evaluated.

Additional research could also focus on the parameters governing the adaptive weighting matrix used in the RPIW method. It is very likely that the saturation bounds $\underline{\phi}$ and $\bar{\phi}$ could be better selected. It is also very likely that Eq. (3.48) could be better formulated. Furthermore, consideration should be given to developing and testing adaptive weighting matrices that can properly handle RW zero crossings.

Jin[21] identified several variations to the RPI allocation method and evaluated their performance in numerical simulation. These variations involve different criteria for selecting a single actuator to constrain when multiple actuators have become constrained in the same solution iteration (see Eq. (3.42).) The evidence presented suggests that other variations of RPI method are feasible and can potentially reduce allocation error. Future work could include the evaluation of these RPI method variations as applied to the REBEL simulation developed in this research.

No disturbance torques were used in this simulation. There are a multitude of disturbance torques that act on the actual REBEL test bed, such as air drag, drag from the air bearing, gravitational pull (center of mass imbalance), and air currents.

Though it is expected that the presence of these disturbance torques would only further highlight the improved performance metrics of the RPI and RPIW method, verification of this assumption would be useful. This research would need to accurately characterize these disturbance torques as physics based equations, which are often non-linear. Much effort would need to be committed to accurately model these disturbance torques.

Other considerations for future work should include evaluation of many of the simplifying assumptions made in this research. Potential topics that should be studied further include measurement noise, controller update rate variations, RW motor response characterization, motor behavior characterization (gear lash, bearing friction, air drag, etc.) and the modeling of low level motor controllers.

The only actuators studied in this research were RWs; however, the concepts presented in this research should work equally well for other sets of actuators. Actuators such as control moment gyroscopes or spacecraft thrusters could be added to the model to verify the applicability of the control allocation concepts presented in this research to other types of constrained and over actuated systems. Additionally, these concepts could also apply equally well to constrained and over actuated control systems on aircraft or other vehicles. The allocation methods presented here could be implemented and evaluated on these vehicle types as well.

Finally, there exist other control allocation methods that could be considered. In particular, there are control allocation methods that not only explicitly minimize the control effort, but also explicitly minimize the allocation error as well. These methods introduce a slack term s into the cost function presented in Eq. (2.22), where s is $\vec{\tau}_{com} - \vec{\tau}$. The solution to the minimization of these cost functions are much more complicated than the ones presented here. In fact, there is no closed form solution for these minimization problems, and linear or quadratic programming methods are often used to solve them.[1] These allocation methods could be compared against the allocation methods presented in this research to determine if they are a feasible method of control allocation for constrained and over actuated spacecraft attitude control systems. While these methods may be feasible, they are certainly more complicated and difficult to implement; while the methods presented in this thesis have been demonstrated as viable options that are arguably more easily implemented.

Appendix A. Multiple Simulation Run Results for SPI

Table A.1. 343 simulation results utilizing SPI allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

Man. No.	x0	y0	z0	xf	yf	zf	Man. t	Sat. t	Hcom Err	Energy
2	0	0	0	2.5	0	0	24.45	0	1.28	1.95
3	0	0	0	5	0	0	35.6	0	2.29	3.47
4	0	0	0	7.5	0	0	42.85	0	3.38	4.68
5	0	0	0	10	0	0	48.4	0	5.18	5.67
6	0	0	0	12.5	0	0	52.8	0	9.48	6.58
7	0	0	0	15	0	0	56.05	0	13.42	7.5
8	0	0	0	0	2.5	0	24.45	0	0.47	1.92
9	0	0	0	2.5	2.5	0	29.8	0	4.22	2.7
10	0	0	0	5	2.5	0	38.1	0	6.26	3.91
11	0	0	0	7.5	2.5	0	45.1	0	8.2	4.91
12	0	0	0	10	2.5	0	50.8	0	10.26	5.79
13	0	0	0	12.5	2.5	0	55.55	0	14.86	6.61
14	0	0	0	15	2.5	0	59.3	0	19.2	7.47
15	0	0	0	0	5	0	35.65	0	1.13	3.45
16	0	0	0	2.5	5	0	37.65	0	4.72	3.85
17	0	0	0	5	5	0	41.7	0	8.41	4.78
18	0	0	0	7.5	5	0	47.1	0	12.06	5.79
19	0	0	0	10	5	0	52.55	0	15.39	6.7
20	0	0	0	12.5	5	0	57.45	0	19.6	7.53
21	0	0	0	15	5	0	61.6	0	24.85	8.3
22	0	0	0	0	7.5	0	42.95	0	1.91	4.75

Table A.1. 343 simulation results utilizing SPI allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
23	0	0	0	2.5	7.5	0	44.15	0	5.12	5.14
24	0	0	0	5	7.5	0	46.2	0	8.72	5.66
25	0	0	0	7.5	7.5	0	49.1	0	13.61	6.69
26	0	0	0	10	7.5	0	53.1	0	17.81	7.66
27	0	0	0	12.5	7.5	0	58.2	0	21.58	8.71
28	0	0	0	15	7.5	0	63.9	0	25.34	9.93
29	0	0	0	0	10	0	48.55	0	3.33	5.91
30	0	0	0	2.5	10	0	49.35	0	6.42	6.31
31	0	0	0	5	10	0	50.6	0	9.47	6.73
32	0	0	0	7.5	10	0	51.95	0	13.94	7.5
33	0	0	0	10	10	0	53.35	0	18.72	8.49
34	0	0	0	12.5	10	0	60.25	0	22.88	9.97
35	0	0	0	15	10	0	69.4	0	26.61	11.51
36	0	0	0	0	12.5	0	53.05	0	5.57	6.99
37	0	0	0	2.5	12.5	0	53.4	0	8.37	7.39
38	0	0	0	5	12.5	0	54.05	0	10.94	7.82
39	0	0	0	7.5	12.5	0	55.1	0	14.16	8.28
40	0	0	0	10	12.5	0	53.55	0	19.07	9.28
41	0	0	0	12.5	12.5	0	64.4	0	23.66	11.2
42	0	0	0	15	12.5	0	76.55	0	27.73	13.16
43	0	0	0	0	15	0	56.35	0	7.79	7.98
44	0	0	0	2.5	15	0	56	0	10.43	8.39

Table A.1. 343 simulation results utilizing SPI allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
45	0	0	0	5	15	0	55.8	0	12.81	8.82
46	0	0	0	7.5	15	0	56.5	0	15.18	9.26
47	0	0	0	10	15	0	53.6	0	19.47	10.01
48	0	0	0	12.5	15	0	69.45	0	24.24	12.44
49	0	0	0	15	15	0	83.7	1.45	29.9	14.87
50	0	0	0	0	0	2.5	23.8	0	0.46	1.92
51	0	0	0	2.5	0	2.5	28.7	0	1.42	3.5
52	0	0	0	5	0	2.5	36.65	0	2.51	4.93
53	0	0	0	7.5	0	2.5	43.2	0	3.62	6.08
54	0	0	0	10	0	2.5	48.55	0	5.6	6.99
55	0	0	0	12.5	0	2.5	52.85	0	9.7	7.77
56	0	0	0	15	0	2.5	55.9	0	13.62	8.55
57	0	0	0	0	2.5	2.5	28.7	0	0.75	3.47
58	0	0	0	2.5	2.5	2.5	32.15	0	4.26	4.22
59	0	0	0	5	2.5	2.5	38.9	0	6.18	5.36
60	0	0	0	7.5	2.5	2.5	45.35	0	7.98	6.3
61	0	0	0	10	2.5	2.5	50.85	0	9.87	7.1
62	0	0	0	12.5	2.5	2.5	55.4	0	14.52	7.81
63	0	0	0	15	2.5	2.5	58.95	0	18.59	8.59
64	0	0	0	0	5	2.5	36.7	0	1.47	4.91
65	0	0	0	2.5	5	2.5	38.5	0	4.94	5.29
66	0	0	0	5	5	2.5	42.25	0	8.61	6.23

Table A.1. 343 simulation results utilizing SPI allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
67	0	0	0	7.5	5	2.5	47.5	0	12.07	7.16
68	0	0	0	10	5	2.5	52.8	0	15.14	7.97
69	0	0	0	12.5	5	2.5	57.55	0	19.82	8.67
70	0	0	0	15	5	2.5	61.45	0	24.49	9.34
71	0	0	0	0	7.5	2.5	43.35	0	2.31	6.15
72	0	0	0	2.5	7.5	2.5	44.45	0	5.41	6.51
73	0	0	0	5	7.5	2.5	46.5	0	9.11	7.05
74	0	0	0	7.5	7.5	2.5	49.4	0	13.96	8.05
75	0	0	0	10	7.5	2.5	53.55	0	18.04	8.94
76	0	0	0	12.5	7.5	2.5	58.5	0	21.68	9.88
77	0	0	0	15	7.5	2.5	63.95	0	25.75	10.95
78	0	0	0	0	10	2.5	48.75	0	3.83	7.24
79	0	0	0	2.5	10	2.5	49.5	0	6.75	7.61
80	0	0	0	5	10	2.5	50.7	0	9.89	7.97
81	0	0	0	7.5	10	2.5	51.8	0	14.58	8.8
82	0	0	0	10	10	2.5	53.9	0	19.19	9.72
83	0	0	0	12.5	10	2.5	61	0	23.2	11.1
84	0	0	0	15	10	2.5	69.55	0	26.8	12.51
85	0	0	0	0	12.5	2.5	53.1	0	6.12	8.2
86	0	0	0	2.5	12.5	2.5	53.45	0	8.74	8.58
87	0	0	0	5	12.5	2.5	54.1	0	11.26	8.96
88	0	0	0	7.5	12.5	2.5	54.95	0	14.84	9.47

Table A.1. 343 simulation results utilizing SPI allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
89	0	0	0	10	12.5	2.5	54.25	0	19.79	10.47
90	0	0	0	12.5	12.5	2.5	66.05	0	24.19	12.35
91	0	0	0	15	12.5	2.5	77.3	0	28.1	14.16
92	0	0	0	0	15	2.5	56.2	0	8.38	9.06
93	0	0	0	2.5	15	2.5	55.8	0	10.84	9.43
94	0	0	0	5	15	2.5	55.7	0	13.16	9.82
95	0	0	0	7.5	15	2.5	56.5	0	15.68	10.24
96	0	0	0	10	15	2.5	55	0	20.31	11.28
97	0	0	0	12.5	15	2.5	72	0	24.97	13.68
98	0	0	0	15	15	2.5	85.1	1.9	30.87	15.93
99	0	0	0	0	0	5	34.45	0	0.81	3.37
100	0	0	0	2.5	0	5	35.75	0	1.72	4.91
101	0	0	0	5	0	5	39.7	0	2.73	6.22
102	0	0	0	7.5	0	5	44.5	0	3.87	7.33
103	0	0	0	10	0	5	49.2	0	5.99	8.24
104	0	0	0	12.5	0	5	53.15	0	9.96	9.01
105	0	0	0	15	0	5	55.9	0	13.86	9.74
106	0	0	0	0	2.5	5	35.75	0	1.09	4.88
107	0	0	0	2.5	2.5	5	37.25	0	4.3	5.56
108	0	0	0	5	2.5	5	41.25	0	6.13	6.6
109	0	0	0	7.5	2.5	5	46.35	0	7.8	7.49
110	0	0	0	10	2.5	5	51.2	0	9.6	8.25

Table A.1. 343 simulation results utilizing SPI allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
111	0	0	0	12.5	2.5	5	55.45	0	14.24	8.97
112	0	0	0	15	2.5	5	58.65	0	18.11	9.76
113	0	0	0	0	5	5	39.7	0	1.76	6.2
114	0	0	0	2.5	5	5	40.95	0	5.13	6.52
115	0	0	0	5	5	5	43.85	0	8.76	7.45
116	0	0	0	7.5	5	5	48.35	0	12.04	8.33
117	0	0	0	10	5	5	53.25	0	14.89	9.08
118	0	0	0	12.5	5	5	57.7	0	19.6	9.73
119	0	0	0	15	5	5	61.4	0	23.74	10.39
120	0	0	0	0	7.5	5	44.65	0	2.65	7.4
121	0	0	0	2.5	7.5	5	45.55	0	5.66	7.72
122	0	0	0	5	7.5	5	47.35	0	9.44	8.27
123	0	0	0	7.5	7.5	5	50.1	0	14.25	9.23
124	0	0	0	10	7.5	5	54.1	0	18.21	10.07
125	0	0	0	12.5	7.5	5	58.9	0	21.76	10.99
126	0	0	0	15	7.5	5	64	0	26.17	11.99
127	0	0	0	0	10	5	49.35	0	4.28	8.48
128	0	0	0	2.5	10	5	50.05	0	7.04	8.8
129	0	0	0	5	10	5	51.15	0	10.29	9.12
130	0	0	0	7.5	10	5	52.05	0	15.12	9.99
131	0	0	0	10	10	5	54.65	0	19.58	10.88
132	0	0	0	12.5	10	5	61.7	0	23.46	12.25

Table A.1. 343 simulation results utilizing SPI allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
133	0	0	0	15	10	5	69.7	0	26.93	13.51
134	0	0	0	0	12.5	5	53.4	0	6.62	9.44
135	0	0	0	2.5	12.5	5	53.7	0	9.05	9.77
136	0	0	0	5	12.5	5	54.4	0	11.54	10.1
137	0	0	0	7.5	12.5	5	54.55	0	15.55	10.68
138	0	0	0	10	12.5	5	55.6	0	20.39	11.76
139	0	0	0	12.5	12.5	5	67.5	0	24.63	13.5
140	0	0	0	15	12.5	5	77.9	0	28.36	15.11
141	0	0	0	0	15	5	56.2	0	8.93	10.28
142	0	0	0	2.5	15	5	55.75	0	11.19	10.6
143	0	0	0	5	15	5	55.8	0	13.46	10.95
144	0	0	0	7.5	15	5	56.5	0	16.3	11.34
145	0	0	0	10	15	5	57.85	0	21.05	12.61
146	0	0	0	12.5	15	5	74.2	0	25.57	14.82
147	0	0	0	15	15	5	86.2	2.1	31.45	16.9
148	0	0	0	0	0	7.5	41.2	0	1.1	4.54
149	0	0	0	2.5	0	7.5	41.7	0	2.08	6.1
150	0	0	0	5	0	7.5	43.6	0	2.96	7.36
151	0	0	0	7.5	0	7.5	46.7	0	4.1	8.43
152	0	0	0	10	0	7.5	50.35	0	6.34	9.32
153	0	0	0	12.5	0	7.5	53.8	0	10.22	10.06
154	0	0	0	15	0	7.5	56.15	0	14.11	10.77

Table A.1. 343 simulation results utilizing SPI allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
155	0	0	0	0	2.5	7.5	41.7	0	1.45	6.06
156	0	0	0	2.5	2.5	7.5	42.4	0	4.55	6.73
157	0	0	0	5	2.5	7.5	44.6	0	6.03	7.7
158	0	0	0	7.5	2.5	7.5	48.1	0	7.58	8.52
159	0	0	0	10	2.5	7.5	52.05	0	9.39	9.25
160	0	0	0	12.5	2.5	7.5	55.7	0	13.92	10.01
161	0	0	0	15	2.5	7.5	58.55	0	17.59	10.79
162	0	0	0	0	5	7.5	43.6	0	1.99	7.34
163	0	0	0	2.5	5	7.5	44.35	0	5.36	7.64
164	0	0	0	5	5	7.5	46.35	0	8.87	8.53
165	0	0	0	7.5	5	7.5	49.8	0	12	9.34
166	0	0	0	10	5	7.5	54	0	14.77	10.04
167	0	0	0	12.5	5	7.5	58.05	0	19.24	10.65
168	0	0	0	15	5	7.5	61.45	0	23.37	11.3
169	0	0	0	0	7.5	7.5	46.8	0	2.92	8.49
170	0	0	0	2.5	7.5	7.5	47.5	0	5.93	8.77
171	0	0	0	5	7.5	7.5	48.95	0	9.8	9.35
172	0	0	0	7.5	7.5	7.5	51.25	0	14.49	10.26
173	0	0	0	10	7.5	7.5	54.95	0	18.35	11.05
174	0	0	0	12.5	7.5	7.5	59.4	0	21.94	11.96
175	0	0	0	15	7.5	7.5	64.2	0	26.59	12.9
176	0	0	0	0	10	7.5	50.5	0	4.67	9.55

Table A.1. 343 simulation results utilizing SPI allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
177	0	0	0	2.5	10	7.5	51.1	0	7.32	9.83
178	0	0	0	5	10	7.5	52.1	0	10.66	10.16
179	0	0	0	7.5	10	7.5	52.8	0	15.57	11.03
180	0	0	0	10	10	7.5	55.7	0	19.9	11.94
181	0	0	0	12.5	10	7.5	62.4	0	23.66	13.25
182	0	0	0	15	10	7.5	70	0	27.12	14.45
183	0	0	0	0	12.5	7.5	54	0	7.07	10.5
184	0	0	0	2.5	12.5	7.5	54.25	0	9.33	10.78
185	0	0	0	5	12.5	7.5	54.95	0	11.84	11.08
186	0	0	0	7.5	12.5	7.5	54.6	0	16.17	11.72
187	0	0	0	10	12.5	7.5	57.3	0	20.88	12.89
188	0	0	0	12.5	12.5	7.5	68.75	0	24.97	14.55
189	0	0	0	15	12.5	7.5	78.5	0	28.57	16.08
190	0	0	0	0	15	7.5	56.35	0	9.4	11.33
191	0	0	0	2.5	15	7.5	55.95	0	11.51	11.62
192	0	0	0	5	15	7.5	56.2	0	13.72	11.93
193	0	0	0	7.5	15	7.5	56.2	0	16.93	12.34
194	0	0	0	10	15	7.5	61	0	21.68	13.85
195	0	0	0	12.5	15	7.5	76.05	0	26.07	15.94
196	0	0	0	15	15	7.5	87.1	2.1	31.71	17.79
197	0	0	0	0	0	10	46.3	0	1.34	5.54
198	0	0	0	2.5	0	10	46.5	0	2.39	7.12

Table A.1. 343 simulation results utilizing SPI allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
199	0	0	0	5	0	10	47.5	0	3.33	8.37
200	0	0	0	7.5	0	10	49.4	0	4.33	9.41
201	0	0	0	10	0	10	52.05	0	6.67	10.28
202	0	0	0	12.5	0	10	54.85	0	10.48	11.02
203	0	0	0	15	0	10	56.75	0	14.38	11.68
204	0	0	0	0	2.5	10	46.5	0	1.77	7.08
205	0	0	0	2.5	2.5	10	46.9	0	4.77	7.74
206	0	0	0	5	2.5	10	48.15	0	5.91	8.69
207	0	0	0	7.5	2.5	10	50.4	0	7.3	9.46
208	0	0	0	10	2.5	10	53.35	0	9.25	10.15
209	0	0	0	12.5	2.5	10	56.35	0	13.52	10.95
210	0	0	0	15	2.5	10	58.6	0	17.05	11.73
211	0	0	0	0	5	10	47.5	0	2.35	8.35
212	0	0	0	2.5	5	10	48	0	5.68	8.65
213	0	0	0	5	5	10	49.3	0	8.95	9.49
214	0	0	0	7.5	5	10	51.75	0	11.93	10.24
215	0	0	0	10	5	10	55.15	0	14.34	10.88
216	0	0	0	12.5	5	10	58.65	0	19.1	11.46
217	0	0	0	15	5	10	61.55	0	23.09	12.12
218	0	0	0	0	7.5	10	49.45	0	3.16	9.48
219	0	0	0	2.5	7.5	10	50	0	6.22	9.72
220	0	0	0	5	7.5	10	51.1	0	10.15	10.33

Table A.1. 343 simulation results utilizing SPI allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
221	0	0	0	7.5	7.5	10	52.95	0	14.71	11.18
222	0	0	0	10	7.5	10	56.05	0	18.49	11.93
223	0	0	0	12.5	7.5	10	60.05	0	22.25	12.83
224	0	0	0	15	7.5	10	64.5	0	27.05	13.73
225	0	0	0	0	10	10	52.2	0	5.02	10.51
226	0	0	0	2.5	10	10	52.65	0	7.63	10.75
227	0	0	0	5	10	10	53.5	0	11.02	11.11
228	0	0	0	7.5	10	10	54.1	0	15.96	11.95
229	0	0	0	10	10	10	56.95	0	20.18	12.88
230	0	0	0	12.5	10	10	63.25	0	23.84	14.15
231	0	0	0	15	10	10	70.55	0	27.44	15.31
232	0	0	0	0	12.5	10	55	0	7.45	11.44
233	0	0	0	2.5	12.5	10	55.2	0	9.66	11.69
234	0	0	0	5	12.5	10	55.85	0	12.19	11.95
235	0	0	0	7.5	12.5	10	55.2	0	16.72	12.65
236	0	0	0	10	12.5	10	59.2	0	21.32	13.91
237	0	0	0	12.5	12.5	10	69.95	0	25.25	15.5
238	0	0	0	15	12.5	10	79.1	0	28.73	16.96
239	0	0	0	0	15	10	56.85	0	9.83	12.27
240	0	0	0	2.5	15	10	56.5	0	11.84	12.52
241	0	0	0	5	15	10	56.95	0	13.96	12.8
242	0	0	0	7.5	15	10	55.95	0	17.58	13.27

Table A.1. 343 simulation results utilizing SPI allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
243	0	0	0	10	15	10	63.95	0	22.25	14.97
244	0	0	0	12.5	15	10	77.7	0	26.47	16.97
245	0	0	0	15	15	10	87.9	1.9	31.76	18.69
246	0	0	0	0	0	12.5	50.45	0	1.54	6.41
247	0	0	0	2.5	0	12.5	50.6	0	2.66	8.01
248	0	0	0	5	0	12.5	51.15	0	3.67	9.27
249	0	0	0	7.5	0	12.5	52.3	0	4.62	10.31
250	0	0	0	10	0	12.5	54.1	0	6.97	11.17
251	0	0	0	12.5	0	12.5	56.2	0	10.73	11.89
252	0	0	0	15	0	12.5	57.65	0	14.62	12.52
253	0	0	0	0	2.5	12.5	50.6	0	2.04	7.97
254	0	0	0	2.5	2.5	12.5	50.8	0	4.93	8.63
255	0	0	0	5	2.5	12.5	51.55	0	5.92	9.57
256	0	0	0	7.5	2.5	12.5	53	0	6.94	10.32
257	0	0	0	10	2.5	12.5	55.05	0	9.18	11.03
258	0	0	0	12.5	2.5	12.5	57.35	0	13.09	11.83
259	0	0	0	15	2.5	12.5	59	0	16.51	12.6
260	0	0	0	0	5	12.5	51.15	0	2.68	9.25
261	0	0	0	2.5	5	12.5	51.45	0	6.01	9.55
262	0	0	0	5	5	12.5	52.3	0	8.98	10.35
263	0	0	0	7.5	5	12.5	54.05	0	11.72	11.04
264	0	0	0	10	5	12.5	56.6	0	14.09	11.64

Table A.1. 343 simulation results utilizing SPI allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
265	0	0	0	12.5	5	12.5	59.45	0	18.97	12.19
266	0	0	0	15	5	12.5	61.85	0	22.75	12.87
267	0	0	0	0	7.5	12.5	52.35	0	3.45	10.37
268	0	0	0	2.5	7.5	12.5	52.75	0	6.52	10.58
269	0	0	0	5	7.5	12.5	53.55	0	10.44	11.21
270	0	0	0	7.5	7.5	12.5	54.95	0	14.9	12.02
271	0	0	0	10	7.5	12.5	57.5	0	18.61	12.73
272	0	0	0	12.5	7.5	12.5	60.9	0	22.65	13.61
273	0	0	0	15	7.5	12.5	64.85	0	27.52	14.45
274	0	0	0	0	10	12.5	54.2	0	5.35	11.38
275	0	0	0	2.5	10	12.5	54.6	0	7.91	11.6
276	0	0	0	5	10	12.5	55.3	0	11.38	11.98
277	0	0	0	7.5	10	12.5	55.75	0	16.32	12.8
278	0	0	0	10	10	12.5	58.45	0	20.43	13.76
279	0	0	0	12.5	10	12.5	64.2	0	24.01	14.98
280	0	0	0	15	10	12.5	71.25	0	27.84	16.11
281	0	0	0	0	12.5	12.5	56.35	0	7.82	12.3
282	0	0	0	2.5	12.5	12.5	56.55	0	9.94	12.53
283	0	0	0	5	12.5	12.5	57.1	0	12.54	12.76
284	0	0	0	7.5	12.5	12.5	56.45	0	17.24	13.51
285	0	0	0	10	12.5	12.5	61.15	0	21.68	14.84
286	0	0	0	12.5	12.5	12.5	71.15	0	25.5	16.38

Table A.1. 343 simulation results utilizing SPI allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
287	0	0	0	15	12.5	12.5	79.8	0	28.87	17.79
288	0	0	0	0	15	12.5	57.65	0	10.23	13.11
289	0	0	0	2.5	15	12.5	57.45	0	12.15	13.34
290	0	0	0	5	15	12.5	58.05	0	14.18	13.6
291	0	0	0	7.5	15	12.5	56.7	0	18.17	14.14
292	0	0	0	10	15	12.5	66.65	0	22.73	15.98
293	0	0	0	12.5	15	12.5	79.2	0	26.8	17.91
294	0	0	0	15	15	12.5	88.7	1.65	31.7	19.53
295	0	0	0	0	0	15	54	0	1.71	7.19
296	0	0	0	2.5	0	15	54.1	0	2.9	8.81
297	0	0	0	5	0	15	54.4	0	3.97	10.08
298	0	0	0	7.5	0	15	55.15	0	4.97	11.13
299	0	0	0	10	0	15	56.4	0	7.25	12
300	0	0	0	12.5	0	15	57.9	0	10.97	12.71
301	0	0	0	15	0	15	58.9	0	14.86	13.31
302	0	0	0	0	2.5	15	54.1	0	2.29	8.76
303	0	0	0	2.5	2.5	15	54.25	0	5.07	9.43
304	0	0	0	5	2.5	15	54.75	0	5.88	10.37
305	0	0	0	7.5	2.5	15	55.65	0	6.65	11.12
306	0	0	0	10	2.5	15	57.05	0	8.97	11.85
307	0	0	0	12.5	2.5	15	58.7	0	12.67	12.65
308	0	0	0	15	2.5	15	59.8	0	16.01	13.41

Table A.1. 343 simulation results utilizing SPI allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
309	0	0	0	0	5	15	54.4	0	2.98	10.06
310	0	0	0	2.5	5	15	54.65	0	6.29	10.36
311	0	0	0	5	5	15	55.25	0	9.14	11.13
312	0	0	0	7.5	5	15	56.45	0	11.49	11.77
313	0	0	0	10	5	15	58.35	0	14	12.33
314	0	0	0	12.5	5	15	60.55	0	18.79	12.86
315	0	0	0	15	5	15	62.4	0	22.37	13.56
316	0	0	0	0	7.5	15	55.2	0	3.79	11.18
317	0	0	0	2.5	7.5	15	55.5	0	6.87	11.36
318	0	0	0	5	7.5	15	56.1	0	10.7	12.02
319	0	0	0	7.5	7.5	15	57.15	0	15.06	12.78
320	0	0	0	10	7.5	15	59.15	0	18.68	13.44
321	0	0	0	12.5	7.5	15	61.95	0	23.08	14.29
322	0	0	0	15	7.5	15	65.2	0	28.04	15.04
323	0	0	0	0	10	15	56.45	0	5.65	12.19
324	0	0	0	2.5	10	15	56.8	0	8.16	12.38
325	0	0	0	5	10	15	57.35	0	11.77	12.78
326	0	0	0	7.5	10	15	57.75	0	16.64	13.59
327	0	0	0	10	10	15	60.15	0	20.67	14.57
328	0	0	0	12.5	10	15	65.25	0	24.16	15.74
329	0	0	0	15	10	15	71.95	0	28.27	16.82
330	0	0	0	0	12.5	15	58	0	8.16	13.1

Table A.1. 343 simulation results utilizing SPI allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
331	0	0	0	2.5	12.5	15	58.2	0	10.2	13.29
332	0	0	0	5	12.5	15	58.65	0	12.91	13.53
333	0	0	0	7.5	12.5	15	58.15	0	17.69	14.31
334	0	0	0	10	12.5	15	63.1	0	22.01	15.7
335	0	0	0	12.5	12.5	15	72.35	0	25.72	17.2
336	0	0	0	15	12.5	15	80.6	0	29.01	18.57
337	0	0	0	0	15	15	58.8	0	10.6	13.9
338	0	0	0	2.5	15	15	58.75	0	12.44	14.11
339	0	0	0	5	15	15	59.3	0	14.45	14.34
340	0	0	0	7.5	15	15	58.45	0	18.71	14.97
341	0	0	0	10	15	15	69.05	0	23.16	16.93
342	0	0	0	12.5	15	15	80.6	0	27.08	18.77
343	0	0	0	15	15	15	89.55	1.3	31.56	20.31

Appendix B. Multiple Simulation Run Results for RPI

Table B.1. 343 simulation results utilizing RPI allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

Man. No.	x0	y0	z0	xf	yf	zf	Man. t	Sat. t	Hcom Err	Energy
2	0	0	0	2.5	0	0	24.45	0	1.28	1.95
3	0	0	0	5	0	0	35.6	0	2.29	3.47
4	0	0	0	7.5	0	0	42.85	0	3.38	4.68
5	0	0	0	10	0	0	48.4	0	5.18	5.67
6	0	0	0	12.5	0	0	52.8	0	9.48	6.58
7	0	0	0	15	0	0	56.05	0	13.42	7.5
8	0	0	0	0	2.5	0	24.45	0	0.47	1.92
9	0	0	0	2.5	2.5	0	29.8	0	4.22	2.7
10	0	0	0	5	2.5	0	38.1	0	6.26	3.91
11	0	0	0	7.5	2.5	0	45.1	0	8.2	4.91
12	0	0	0	10	2.5	0	50.8	0	10.26	5.79
13	0	0	0	12.5	2.5	0	55.55	0	14.86	6.61
14	0	0	0	15	2.5	0	59.3	0	19.2	7.47
15	0	0	0	0	5	0	35.65	0	1.13	3.45
16	0	0	0	2.5	5	0	37.65	0	4.72	3.85
17	0	0	0	5	5	0	41.7	0	8.41	4.78
18	0	0	0	7.5	5	0	47.1	0	12.06	5.79
19	0	0	0	10	5	0	52.55	0	15.39	6.7
20	0	0	0	12.5	5	0	57.45	0	19.6	7.53
21	0	0	0	15	5	0	61.6	0	24.85	8.3
22	0	0	0	0	7.5	0	42.95	0	1.91	4.75

Table B.1. 343 simulation results utilizing RPI allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
23	0	0	0	2.5	7.5	0	44.15	0	5.12	5.14
24	0	0	0	5	7.5	0	46.2	0	8.72	5.66
25	0	0	0	7.5	7.5	0	49.1	0	13.61	6.69
26	0	0	0	10	7.5	0	53.1	0	17.81	7.66
27	0	0	0	12.5	7.5	0	58.2	0	21.58	8.71
28	0	0	0	15	7.5	0	63.9	0	25.34	9.93
29	0	0	0	0	10	0	48.55	0	3.33	5.91
30	0	0	0	2.5	10	0	49.35	0	6.42	6.31
31	0	0	0	5	10	0	50.6	0	9.47	6.73
32	0	0	0	7.5	10	0	51.95	0	13.94	7.5
33	0	0	0	10	10	0	53.35	0	18.72	8.49
34	0	0	0	12.5	10	0	60.25	0	22.88	9.97
35	0	0	0	15	10	0	69.4	0	26.61	11.51
36	0	0	0	0	12.5	0	53.05	0	5.57	6.99
37	0	0	0	2.5	12.5	0	53.4	0	8.37	7.39
38	0	0	0	5	12.5	0	54.05	0	10.94	7.82
39	0	0	0	7.5	12.5	0	55.1	0	14.16	8.28
40	0	0	0	10	12.5	0	53.55	0	19.07	9.28
41	0	0	0	12.5	12.5	0	64.4	0	23.66	11.2
42	0	0	0	15	12.5	0	76.55	0	27.73	13.16
43	0	0	0	0	15	0	56.35	0	7.79	7.98
44	0	0	0	2.5	15	0	56	0	10.43	8.39

Table B.1. 343 simulation results utilizing RPI allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
45	0	0	0	5	15	0	55.8	0	12.81	8.82
46	0	0	0	7.5	15	0	56.5	0	15.18	9.26
47	0	0	0	10	15	0	53.6	0	19.47	10.01
48	0	0	0	12.5	15	0	69.45	0	24.24	12.44
49	0	0	0	15	15	0	83.7	1.45	29.9	14.87
50	0	0	0	0	0	2.5	23.8	0	0.46	1.92
51	0	0	0	2.5	0	2.5	28.7	0	1.42	3.5
52	0	0	0	5	0	2.5	36.65	0	2.51	4.93
53	0	0	0	7.5	0	2.5	43.2	0	3.62	6.08
54	0	0	0	10	0	2.5	48.55	0	5.6	6.99
55	0	0	0	12.5	0	2.5	52.85	0	9.7	7.77
56	0	0	0	15	0	2.5	55.9	0	13.62	8.55
57	0	0	0	0	2.5	2.5	28.7	0	0.75	3.47
58	0	0	0	2.5	2.5	2.5	32.15	0	4.26	4.22
59	0	0	0	5	2.5	2.5	38.9	0	6.18	5.36
60	0	0	0	7.5	2.5	2.5	45.35	0	7.98	6.3
61	0	0	0	10	2.5	2.5	50.85	0	9.87	7.1
62	0	0	0	12.5	2.5	2.5	55.4	0	14.52	7.81
63	0	0	0	15	2.5	2.5	58.95	0	18.59	8.59
64	0	0	0	0	5	2.5	36.7	0	1.47	4.91
65	0	0	0	2.5	5	2.5	38.5	0	4.94	5.29
66	0	0	0	5	5	2.5	42.25	0	8.61	6.23

Table B.1. 343 simulation results utilizing RPI allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
67	0	0	0	7.5	5	2.5	47.5	0	12.07	7.16
68	0	0	0	10	5	2.5	52.8	0	15.14	7.97
69	0	0	0	12.5	5	2.5	57.55	0	19.82	8.67
70	0	0	0	15	5	2.5	61.45	0	24.49	9.34
71	0	0	0	0	7.5	2.5	43.35	0	2.31	6.15
72	0	0	0	2.5	7.5	2.5	44.45	0	5.41	6.51
73	0	0	0	5	7.5	2.5	46.5	0	9.11	7.05
74	0	0	0	7.5	7.5	2.5	49.4	0	13.96	8.05
75	0	0	0	10	7.5	2.5	53.55	0	18.04	8.94
76	0	0	0	12.5	7.5	2.5	58.5	0	21.68	9.88
77	0	0	0	15	7.5	2.5	63.95	0	25.75	10.95
78	0	0	0	0	10	2.5	48.75	0	3.83	7.24
79	0	0	0	2.5	10	2.5	49.5	0	6.75	7.61
80	0	0	0	5	10	2.5	50.7	0	9.89	7.97
81	0	0	0	7.5	10	2.5	51.8	0	14.58	8.8
82	0	0	0	10	10	2.5	53.9	0	19.19	9.72
83	0	0	0	12.5	10	2.5	61	0	23.2	11.1
84	0	0	0	15	10	2.5	69.55	0	26.8	12.51
85	0	0	0	0	12.5	2.5	53.1	0	6.12	8.2
86	0	0	0	2.5	12.5	2.5	53.45	0	8.74	8.58
87	0	0	0	5	12.5	2.5	54.1	0	11.26	8.96
88	0	0	0	7.5	12.5	2.5	54.95	0	14.84	9.47

Table B.1. 343 simulation results utilizing RPI allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
89	0	0	0	10	12.5	2.5	54.25	0	19.79	10.47
90	0	0	0	12.5	12.5	2.5	66.05	0	24.19	12.35
91	0	0	0	15	12.5	2.5	77.3	0	28.1	14.16
92	0	0	0	0	15	2.5	56.2	0	8.38	9.06
93	0	0	0	2.5	15	2.5	55.8	0	10.84	9.43
94	0	0	0	5	15	2.5	55.7	0	13.16	9.82
95	0	0	0	7.5	15	2.5	56.5	0	15.68	10.24
96	0	0	0	10	15	2.5	55	0	20.31	11.28
97	0	0	0	12.5	15	2.5	72	0	24.97	13.68
98	0	0	0	15	15	2.5	85.1	1.9	30.87	15.93
99	0	0	0	0	0	5	34.45	0	0.81	3.37
100	0	0	0	2.5	0	5	35.75	0	1.72	4.91
101	0	0	0	5	0	5	39.7	0	2.73	6.22
102	0	0	0	7.5	0	5	44.5	0	3.87	7.33
103	0	0	0	10	0	5	49.2	0	5.99	8.24
104	0	0	0	12.5	0	5	53.15	0	9.96	9.01
105	0	0	0	15	0	5	55.9	0	13.86	9.74
106	0	0	0	0	2.5	5	35.75	0	1.09	4.88
107	0	0	0	2.5	2.5	5	37.25	0	4.3	5.56
108	0	0	0	5	2.5	5	41.25	0	6.13	6.6
109	0	0	0	7.5	2.5	5	46.35	0	7.8	7.49
110	0	0	0	10	2.5	5	51.2	0	9.6	8.25

Table B.1. 343 simulation results utilizing RPI allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
111	0	0	0	12.5	2.5	5	55.45	0	14.24	8.97
112	0	0	0	15	2.5	5	58.65	0	18.11	9.76
113	0	0	0	0	5	5	39.7	0	1.76	6.2
114	0	0	0	2.5	5	5	40.95	0	5.13	6.52
115	0	0	0	5	5	5	43.85	0	8.76	7.45
116	0	0	0	7.5	5	5	48.35	0	12.04	8.33
117	0	0	0	10	5	5	53.25	0	14.89	9.08
118	0	0	0	12.5	5	5	57.7	0	19.6	9.73
119	0	0	0	15	5	5	61.4	0	23.74	10.39
120	0	0	0	0	7.5	5	44.65	0	2.65	7.4
121	0	0	0	2.5	7.5	5	45.55	0	5.66	7.72
122	0	0	0	5	7.5	5	47.35	0	9.44	8.27
123	0	0	0	7.5	7.5	5	50.1	0	14.25	9.23
124	0	0	0	10	7.5	5	54.1	0	18.21	10.07
125	0	0	0	12.5	7.5	5	58.9	0	21.76	10.99
126	0	0	0	15	7.5	5	64	0	26.17	11.99
127	0	0	0	0	10	5	49.35	0	4.28	8.48
128	0	0	0	2.5	10	5	50.05	0	7.04	8.8
129	0	0	0	5	10	5	51.15	0	10.29	9.12
130	0	0	0	7.5	10	5	52.05	0	15.12	9.99
131	0	0	0	10	10	5	54.65	0	19.58	10.88
132	0	0	0	12.5	10	5	61.7	0	23.46	12.25

Table B.1. 343 simulation results utilizing RPI allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
133	0	0	0	15	10	5	69.7	0	26.93	13.51
134	0	0	0	0	12.5	5	53.4	0	6.62	9.44
135	0	0	0	2.5	12.5	5	53.7	0	9.05	9.77
136	0	0	0	5	12.5	5	54.4	0	11.54	10.1
137	0	0	0	7.5	12.5	5	54.55	0	15.55	10.68
138	0	0	0	10	12.5	5	55.6	0	20.39	11.76
139	0	0	0	12.5	12.5	5	67.5	0	24.63	13.5
140	0	0	0	15	12.5	5	77.9	0	28.36	15.11
141	0	0	0	0	15	5	56.2	0	8.93	10.28
142	0	0	0	2.5	15	5	55.75	0	11.19	10.6
143	0	0	0	5	15	5	55.8	0	13.46	10.95
144	0	0	0	7.5	15	5	56.5	0	16.3	11.34
145	0	0	0	10	15	5	57.85	0	21.05	12.61
146	0	0	0	12.5	15	5	74.2	0	25.57	14.82
147	0	0	0	15	15	5	86.2	2.1	31.45	16.9
148	0	0	0	0	0	7.5	41.2	0	1.1	4.54
149	0	0	0	2.5	0	7.5	41.7	0	2.08	6.1
150	0	0	0	5	0	7.5	43.6	0	2.96	7.36
151	0	0	0	7.5	0	7.5	46.7	0	4.1	8.43
152	0	0	0	10	0	7.5	50.35	0	6.34	9.32
153	0	0	0	12.5	0	7.5	53.8	0	10.22	10.06
154	0	0	0	15	0	7.5	56.15	0	14.11	10.77

Table B.1. 343 simulation results utilizing RPI allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
155	0	0	0	0	2.5	7.5	41.7	0	1.45	6.06
156	0	0	0	2.5	2.5	7.5	42.4	0	4.55	6.73
157	0	0	0	5	2.5	7.5	44.6	0	6.03	7.7
158	0	0	0	7.5	2.5	7.5	48.1	0	7.58	8.52
159	0	0	0	10	2.5	7.5	52.05	0	9.39	9.25
160	0	0	0	12.5	2.5	7.5	55.7	0	13.92	10.01
161	0	0	0	15	2.5	7.5	58.55	0	17.59	10.79
162	0	0	0	0	5	7.5	43.6	0	1.99	7.34
163	0	0	0	2.5	5	7.5	44.35	0	5.36	7.64
164	0	0	0	5	5	7.5	46.35	0	8.87	8.53
165	0	0	0	7.5	5	7.5	49.8	0	12	9.34
166	0	0	0	10	5	7.5	54	0	14.77	10.04
167	0	0	0	12.5	5	7.5	58.05	0	19.24	10.65
168	0	0	0	15	5	7.5	61.45	0	23.37	11.3
169	0	0	0	0	7.5	7.5	46.8	0	2.92	8.49
170	0	0	0	2.5	7.5	7.5	47.5	0	5.93	8.77
171	0	0	0	5	7.5	7.5	48.95	0	9.8	9.35
172	0	0	0	7.5	7.5	7.5	51.25	0	14.49	10.26
173	0	0	0	10	7.5	7.5	54.95	0	18.35	11.05
174	0	0	0	12.5	7.5	7.5	59.4	0	21.94	11.96
175	0	0	0	15	7.5	7.5	64.2	0	26.59	12.9
176	0	0	0	0	10	7.5	50.5	0	4.67	9.55

Table B.1. 343 simulation results utilizing RPI allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
177	0	0	0	2.5	10	7.5	51.1	0	7.32	9.83
178	0	0	0	5	10	7.5	52.1	0	10.66	10.16
179	0	0	0	7.5	10	7.5	52.8	0	15.57	11.03
180	0	0	0	10	10	7.5	55.7	0	19.9	11.94
181	0	0	0	12.5	10	7.5	62.4	0	23.66	13.25
182	0	0	0	15	10	7.5	70	0	27.12	14.45
183	0	0	0	0	12.5	7.5	54	0	7.07	10.5
184	0	0	0	2.5	12.5	7.5	54.25	0	9.33	10.78
185	0	0	0	5	12.5	7.5	54.95	0	11.84	11.08
186	0	0	0	7.5	12.5	7.5	54.6	0	16.17	11.72
187	0	0	0	10	12.5	7.5	57.3	0	20.88	12.89
188	0	0	0	12.5	12.5	7.5	68.75	0	24.97	14.55
189	0	0	0	15	12.5	7.5	78.5	0	28.57	16.08
190	0	0	0	0	15	7.5	56.35	0	9.4	11.33
191	0	0	0	2.5	15	7.5	55.95	0	11.51	11.62
192	0	0	0	5	15	7.5	56.2	0	13.72	11.93
193	0	0	0	7.5	15	7.5	56.2	0	16.93	12.34
194	0	0	0	10	15	7.5	61	0	21.68	13.85
195	0	0	0	12.5	15	7.5	76.05	0	26.07	15.94
196	0	0	0	15	15	7.5	87.1	2.1	31.71	17.79
197	0	0	0	0	0	10	46.3	0	1.34	5.54
198	0	0	0	2.5	0	10	46.5	0	2.39	7.12

Table B.1. 343 simulation results utilizing RPI allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
199	0	0	0	5	0	10	47.5	0	3.33	8.37
200	0	0	0	7.5	0	10	49.4	0	4.33	9.41
201	0	0	0	10	0	10	52.05	0	6.67	10.28
202	0	0	0	12.5	0	10	54.85	0	10.48	11.02
203	0	0	0	15	0	10	56.75	0	14.38	11.68
204	0	0	0	0	2.5	10	46.5	0	1.77	7.08
205	0	0	0	2.5	2.5	10	46.9	0	4.77	7.74
206	0	0	0	5	2.5	10	48.15	0	5.91	8.69
207	0	0	0	7.5	2.5	10	50.4	0	7.3	9.46
208	0	0	0	10	2.5	10	53.35	0	9.25	10.15
209	0	0	0	12.5	2.5	10	56.35	0	13.52	10.95
210	0	0	0	15	2.5	10	58.6	0	17.05	11.73
211	0	0	0	0	5	10	47.5	0	2.35	8.35
212	0	0	0	2.5	5	10	48	0	5.68	8.65
213	0	0	0	5	5	10	49.3	0	8.95	9.49
214	0	0	0	7.5	5	10	51.75	0	11.93	10.24
215	0	0	0	10	5	10	55.15	0	14.34	10.88
216	0	0	0	12.5	5	10	58.65	0	19.1	11.46
217	0	0	0	15	5	10	61.55	0	23.09	12.12
218	0	0	0	0	7.5	10	49.45	0	3.16	9.48
219	0	0	0	2.5	7.5	10	50	0	6.22	9.72
220	0	0	0	5	7.5	10	51.1	0	10.15	10.33

Table B.1. 343 simulation results utilizing RPI allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
221	0	0	0	7.5	7.5	10	52.95	0	14.71	11.18
222	0	0	0	10	7.5	10	56.05	0	18.49	11.93
223	0	0	0	12.5	7.5	10	60.05	0	22.25	12.83
224	0	0	0	15	7.5	10	64.5	0	27.05	13.73
225	0	0	0	0	10	10	52.2	0	5.02	10.51
226	0	0	0	2.5	10	10	52.65	0	7.63	10.75
227	0	0	0	5	10	10	53.5	0	11.02	11.11
228	0	0	0	7.5	10	10	54.1	0	15.96	11.95
229	0	0	0	10	10	10	56.95	0	20.18	12.88
230	0	0	0	12.5	10	10	63.25	0	23.84	14.15
231	0	0	0	15	10	10	70.55	0	27.44	15.31
232	0	0	0	0	12.5	10	55	0	7.45	11.44
233	0	0	0	2.5	12.5	10	55.2	0	9.66	11.69
234	0	0	0	5	12.5	10	55.85	0	12.19	11.95
235	0	0	0	7.5	12.5	10	55.2	0	16.72	12.65
236	0	0	0	10	12.5	10	59.2	0	21.32	13.91
237	0	0	0	12.5	12.5	10	69.95	0	25.25	15.5
238	0	0	0	15	12.5	10	79.1	0	28.73	16.96
239	0	0	0	0	15	10	56.85	0	9.83	12.27
240	0	0	0	2.5	15	10	56.5	0	11.84	12.52
241	0	0	0	5	15	10	56.95	0	13.96	12.8
242	0	0	0	7.5	15	10	55.95	0	17.58	13.27

Table B.1. 343 simulation results utilizing RPI allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
243	0	0	0	10	15	10	63.95	0	22.25	14.97
244	0	0	0	12.5	15	10	77.7	0	26.47	16.97
245	0	0	0	15	15	10	87.9	1.9	31.76	18.69
246	0	0	0	0	0	12.5	50.45	0	1.54	6.41
247	0	0	0	2.5	0	12.5	50.6	0	2.66	8.01
248	0	0	0	5	0	12.5	51.15	0	3.67	9.27
249	0	0	0	7.5	0	12.5	52.3	0	4.62	10.31
250	0	0	0	10	0	12.5	54.1	0	6.97	11.17
251	0	0	0	12.5	0	12.5	56.2	0	10.73	11.89
252	0	0	0	15	0	12.5	57.65	0	14.62	12.52
253	0	0	0	0	2.5	12.5	50.6	0	2.04	7.97
254	0	0	0	2.5	2.5	12.5	50.8	0	4.93	8.63
255	0	0	0	5	2.5	12.5	51.55	0	5.92	9.57
256	0	0	0	7.5	2.5	12.5	53	0	6.94	10.32
257	0	0	0	10	2.5	12.5	55.05	0	9.18	11.03
258	0	0	0	12.5	2.5	12.5	57.35	0	13.09	11.83
259	0	0	0	15	2.5	12.5	59	0	16.51	12.6
260	0	0	0	0	5	12.5	51.15	0	2.68	9.25
261	0	0	0	2.5	5	12.5	51.45	0	6.01	9.55
262	0	0	0	5	5	12.5	52.3	0	8.98	10.35
263	0	0	0	7.5	5	12.5	54.05	0	11.72	11.04
264	0	0	0	10	5	12.5	56.6	0	14.09	11.64

Table B.1. 343 simulation results utilizing RPI allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
265	0	0	0	12.5	5	12.5	59.45	0	18.97	12.19
266	0	0	0	15	5	12.5	61.85	0	22.75	12.87
267	0	0	0	0	7.5	12.5	52.35	0	3.45	10.37
268	0	0	0	2.5	7.5	12.5	52.75	0	6.52	10.58
269	0	0	0	5	7.5	12.5	53.55	0	10.44	11.21
270	0	0	0	7.5	7.5	12.5	54.95	0	14.9	12.02
271	0	0	0	10	7.5	12.5	57.5	0	18.61	12.73
272	0	0	0	12.5	7.5	12.5	60.9	0	22.65	13.61
273	0	0	0	15	7.5	12.5	64.85	0	27.52	14.45
274	0	0	0	0	10	12.5	54.2	0	5.35	11.38
275	0	0	0	2.5	10	12.5	54.6	0	7.91	11.6
276	0	0	0	5	10	12.5	55.3	0	11.38	11.98
277	0	0	0	7.5	10	12.5	55.75	0	16.32	12.8
278	0	0	0	10	10	12.5	58.45	0	20.43	13.76
279	0	0	0	12.5	10	12.5	64.2	0	24.01	14.98
280	0	0	0	15	10	12.5	71.25	0	27.84	16.11
281	0	0	0	0	12.5	12.5	56.35	0	7.82	12.3
282	0	0	0	2.5	12.5	12.5	56.55	0	9.94	12.53
283	0	0	0	5	12.5	12.5	57.1	0	12.54	12.76
284	0	0	0	7.5	12.5	12.5	56.45	0	17.24	13.51
285	0	0	0	10	12.5	12.5	61.15	0	21.68	14.84
286	0	0	0	12.5	12.5	12.5	71.15	0	25.5	16.38

Table B.1. 343 simulation results utilizing RPI allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
287	0	0	0	15	12.5	12.5	79.8	0	28.87	17.79
288	0	0	0	0	15	12.5	57.65	0	10.23	13.11
289	0	0	0	2.5	15	12.5	57.45	0	12.15	13.34
290	0	0	0	5	15	12.5	58.05	0	14.18	13.6
291	0	0	0	7.5	15	12.5	56.7	0	18.17	14.14
292	0	0	0	10	15	12.5	66.65	0	22.73	15.98
293	0	0	0	12.5	15	12.5	79.2	0	26.8	17.91
294	0	0	0	15	15	12.5	88.7	1.65	31.7	19.53
295	0	0	0	0	0	15	54	0	1.71	7.19
296	0	0	0	2.5	0	15	54.1	0	2.9	8.81
297	0	0	0	5	0	15	54.4	0	3.97	10.08
298	0	0	0	7.5	0	15	55.15	0	4.97	11.13
299	0	0	0	10	0	15	56.4	0	7.25	12
300	0	0	0	12.5	0	15	57.9	0	10.97	12.71
301	0	0	0	15	0	15	58.9	0	14.86	13.31
302	0	0	0	0	2.5	15	54.1	0	2.29	8.76
303	0	0	0	2.5	2.5	15	54.25	0	5.07	9.43
304	0	0	0	5	2.5	15	54.75	0	5.88	10.37
305	0	0	0	7.5	2.5	15	55.65	0	6.65	11.12
306	0	0	0	10	2.5	15	57.05	0	8.97	11.85
307	0	0	0	12.5	2.5	15	58.7	0	12.67	12.65
308	0	0	0	15	2.5	15	59.8	0	16.01	13.41

Table B.1. 343 simulation results utilizing RPI allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
309	0	0	0	0	5	15	54.4	0	2.98	10.06
310	0	0	0	2.5	5	15	54.65	0	6.29	10.36
311	0	0	0	5	5	15	55.25	0	9.14	11.13
312	0	0	0	7.5	5	15	56.45	0	11.49	11.77
313	0	0	0	10	5	15	58.35	0	14	12.33
314	0	0	0	12.5	5	15	60.55	0	18.79	12.86
315	0	0	0	15	5	15	62.4	0	22.37	13.56
316	0	0	0	0	7.5	15	55.2	0	3.79	11.18
317	0	0	0	2.5	7.5	15	55.5	0	6.87	11.36
318	0	0	0	5	7.5	15	56.1	0	10.7	12.02
319	0	0	0	7.5	7.5	15	57.15	0	15.06	12.78
320	0	0	0	10	7.5	15	59.15	0	18.68	13.44
321	0	0	0	12.5	7.5	15	61.95	0	23.08	14.29
322	0	0	0	15	7.5	15	65.2	0	28.04	15.04
323	0	0	0	0	10	15	56.45	0	5.65	12.19
324	0	0	0	2.5	10	15	56.8	0	8.16	12.38
325	0	0	0	5	10	15	57.35	0	11.77	12.78
326	0	0	0	7.5	10	15	57.75	0	16.64	13.59
327	0	0	0	10	10	15	60.15	0	20.67	14.57
328	0	0	0	12.5	10	15	65.25	0	24.16	15.74
329	0	0	0	15	10	15	71.95	0	28.27	16.82
330	0	0	0	0	12.5	15	58	0	8.16	13.1

Table B.1. 343 simulation results utilizing RPI allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
331	0	0	0	2.5	12.5	15	58.2	0	10.2	13.29
332	0	0	0	5	12.5	15	58.65	0	12.91	13.53
333	0	0	0	7.5	12.5	15	58.15	0	17.69	14.31
334	0	0	0	10	12.5	15	63.1	0	22.01	15.7
335	0	0	0	12.5	12.5	15	72.35	0	25.72	17.2
336	0	0	0	15	12.5	15	80.6	0	29.01	18.57
337	0	0	0	0	15	15	58.8	0	10.6	13.9
338	0	0	0	2.5	15	15	58.75	0	12.44	14.11
339	0	0	0	5	15	15	59.3	0	14.45	14.34
340	0	0	0	7.5	15	15	58.45	0	18.71	14.97
341	0	0	0	10	15	15	69.05	0	23.16	16.93
342	0	0	0	12.5	15	15	80.6	0	27.08	18.77
343	0	0	0	15	15	15	89.55	1.3	31.56	20.31

Appendix C. Multiple Simulation Run Results for RPIW

Table C.1. 343 simulation results utilizing RPIW allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

Man. No.	x0	y0	z0	xf	yf	zf	Man. t	Sat. t	Hcom Err	Energy
2	0	0	0	2.5	0	0	24.45	0	1.28	1.95
3	0	0	0	5	0	0	35.6	0	2.29	3.47
4	0	0	0	7.5	0	0	42.85	0	3.38	4.68
5	0	0	0	10	0	0	48.4	0	5.18	5.67
6	0	0	0	12.5	0	0	52.8	0	9.48	6.58
7	0	0	0	15	0	0	56.05	0	13.42	7.5
8	0	0	0	0	2.5	0	24.45	0	0.47	1.92
9	0	0	0	2.5	2.5	0	29.8	0	4.22	2.7
10	0	0	0	5	2.5	0	38.1	0	6.26	3.91
11	0	0	0	7.5	2.5	0	45.1	0	8.2	4.91
12	0	0	0	10	2.5	0	50.8	0	10.26	5.79
13	0	0	0	12.5	2.5	0	55.55	0	14.86	6.61
14	0	0	0	15	2.5	0	59.3	0	19.2	7.47
15	0	0	0	0	5	0	35.65	0	1.13	3.45
16	0	0	0	2.5	5	0	37.65	0	4.72	3.85
17	0	0	0	5	5	0	41.7	0	8.41	4.78
18	0	0	0	7.5	5	0	47.1	0	12.06	5.79
19	0	0	0	10	5	0	52.55	0	15.39	6.7
20	0	0	0	12.5	5	0	57.45	0	19.6	7.53
21	0	0	0	15	5	0	61.6	0	24.85	8.3
22	0	0	0	0	7.5	0	42.95	0	1.91	4.75

Table C.1. 343 simulation results utilizing RPIW allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
23	0	0	0	2.5	7.5	0	44.15	0	5.12	5.14
24	0	0	0	5	7.5	0	46.2	0	8.72	5.66
25	0	0	0	7.5	7.5	0	49.1	0	13.61	6.69
26	0	0	0	10	7.5	0	53.1	0	17.81	7.66
27	0	0	0	12.5	7.5	0	58.2	0	21.58	8.71
28	0	0	0	15	7.5	0	63.9	0	25.34	9.93
29	0	0	0	0	10	0	48.55	0	3.33	5.91
30	0	0	0	2.5	10	0	49.35	0	6.42	6.31
31	0	0	0	5	10	0	50.6	0	9.47	6.73
32	0	0	0	7.5	10	0	51.95	0	13.94	7.5
33	0	0	0	10	10	0	53.35	0	18.72	8.49
34	0	0	0	12.5	10	0	60.25	0	22.88	9.97
35	0	0	0	15	10	0	69.4	0	26.61	11.51
36	0	0	0	0	12.5	0	53.05	0	5.57	6.99
37	0	0	0	2.5	12.5	0	53.4	0	8.37	7.39
38	0	0	0	5	12.5	0	54.05	0	10.94	7.82
39	0	0	0	7.5	12.5	0	55.1	0	14.16	8.28
40	0	0	0	10	12.5	0	53.55	0	19.07	9.28
41	0	0	0	12.5	12.5	0	64.4	0	23.66	11.2
42	0	0	0	15	12.5	0	76.55	0	27.73	13.16
43	0	0	0	0	15	0	56.35	0	7.79	7.98
44	0	0	0	2.5	15	0	56	0	10.43	8.39

Table C.1. 343 simulation results utilizing RPIW allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
45	0	0	0	5	15	0	55.8	0	12.81	8.82
46	0	0	0	7.5	15	0	56.5	0	15.18	9.26
47	0	0	0	10	15	0	53.6	0	19.47	10.01
48	0	0	0	12.5	15	0	69.45	0	24.24	12.44
49	0	0	0	15	15	0	83.7	1.45	29.9	14.87
50	0	0	0	0	0	2.5	23.8	0	0.46	1.92
51	0	0	0	2.5	0	2.5	28.7	0	1.42	3.5
52	0	0	0	5	0	2.5	36.65	0	2.51	4.93
53	0	0	0	7.5	0	2.5	43.2	0	3.62	6.08
54	0	0	0	10	0	2.5	48.55	0	5.6	6.99
55	0	0	0	12.5	0	2.5	52.85	0	9.7	7.77
56	0	0	0	15	0	2.5	55.9	0	13.62	8.55
57	0	0	0	0	2.5	2.5	28.7	0	0.75	3.47
58	0	0	0	2.5	2.5	2.5	32.15	0	4.26	4.22
59	0	0	0	5	2.5	2.5	38.9	0	6.18	5.36
60	0	0	0	7.5	2.5	2.5	45.35	0	7.98	6.3
61	0	0	0	10	2.5	2.5	50.85	0	9.87	7.1
62	0	0	0	12.5	2.5	2.5	55.4	0	14.52	7.81
63	0	0	0	15	2.5	2.5	58.95	0	18.59	8.59
64	0	0	0	0	5	2.5	36.7	0	1.47	4.91
65	0	0	0	2.5	5	2.5	38.5	0	4.94	5.29
66	0	0	0	5	5	2.5	42.25	0	8.61	6.23

Table C.1. 343 simulation results utilizing RPIW allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
67	0	0	0	7.5	5	2.5	47.5	0	12.07	7.16
68	0	0	0	10	5	2.5	52.8	0	15.14	7.97
69	0	0	0	12.5	5	2.5	57.55	0	19.82	8.67
70	0	0	0	15	5	2.5	61.45	0	24.49	9.34
71	0	0	0	0	7.5	2.5	43.35	0	2.31	6.15
72	0	0	0	2.5	7.5	2.5	44.45	0	5.41	6.51
73	0	0	0	5	7.5	2.5	46.5	0	9.11	7.05
74	0	0	0	7.5	7.5	2.5	49.4	0	13.96	8.05
75	0	0	0	10	7.5	2.5	53.55	0	18.04	8.94
76	0	0	0	12.5	7.5	2.5	58.5	0	21.68	9.88
77	0	0	0	15	7.5	2.5	63.95	0	25.75	10.95
78	0	0	0	0	10	2.5	48.75	0	3.83	7.24
79	0	0	0	2.5	10	2.5	49.5	0	6.75	7.61
80	0	0	0	5	10	2.5	50.7	0	9.89	7.97
81	0	0	0	7.5	10	2.5	51.8	0	14.58	8.8
82	0	0	0	10	10	2.5	53.9	0	19.19	9.72
83	0	0	0	12.5	10	2.5	61	0	23.2	11.1
84	0	0	0	15	10	2.5	69.55	0	26.8	12.51
85	0	0	0	0	12.5	2.5	53.1	0	6.12	8.2
86	0	0	0	2.5	12.5	2.5	53.45	0	8.74	8.58
87	0	0	0	5	12.5	2.5	54.1	0	11.26	8.96
88	0	0	0	7.5	12.5	2.5	54.95	0	14.84	9.47

Table C.1. 343 simulation results utilizing RPIW allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
89	0	0	0	10	12.5	2.5	54.25	0	19.79	10.47
90	0	0	0	12.5	12.5	2.5	66.05	0	24.19	12.35
91	0	0	0	15	12.5	2.5	77.3	0	28.1	14.16
92	0	0	0	0	15	2.5	56.2	0	8.38	9.06
93	0	0	0	2.5	15	2.5	55.8	0	10.84	9.43
94	0	0	0	5	15	2.5	55.7	0	13.16	9.82
95	0	0	0	7.5	15	2.5	56.5	0	15.68	10.24
96	0	0	0	10	15	2.5	55	0	20.31	11.28
97	0	0	0	12.5	15	2.5	72	0	24.97	13.68
98	0	0	0	15	15	2.5	85.1	1.9	30.87	15.93
99	0	0	0	0	0	5	34.45	0	0.81	3.37
100	0	0	0	2.5	0	5	35.75	0	1.72	4.91
101	0	0	0	5	0	5	39.7	0	2.73	6.22
102	0	0	0	7.5	0	5	44.5	0	3.87	7.33
103	0	0	0	10	0	5	49.2	0	5.99	8.24
104	0	0	0	12.5	0	5	53.15	0	9.96	9.01
105	0	0	0	15	0	5	55.9	0	13.86	9.74
106	0	0	0	0	2.5	5	35.75	0	1.09	4.88
107	0	0	0	2.5	2.5	5	37.25	0	4.3	5.56
108	0	0	0	5	2.5	5	41.25	0	6.13	6.6
109	0	0	0	7.5	2.5	5	46.35	0	7.8	7.49
110	0	0	0	10	2.5	5	51.2	0	9.6	8.25

Table C.1. 343 simulation results utilizing RPIW allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
111	0	0	0	12.5	2.5	5	55.45	0	14.24	8.97
112	0	0	0	15	2.5	5	58.65	0	18.11	9.76
113	0	0	0	0	5	5	39.7	0	1.76	6.2
114	0	0	0	2.5	5	5	40.95	0	5.13	6.52
115	0	0	0	5	5	5	43.85	0	8.76	7.45
116	0	0	0	7.5	5	5	48.35	0	12.04	8.33
117	0	0	0	10	5	5	53.25	0	14.89	9.08
118	0	0	0	12.5	5	5	57.7	0	19.6	9.73
119	0	0	0	15	5	5	61.4	0	23.74	10.39
120	0	0	0	0	7.5	5	44.65	0	2.65	7.4
121	0	0	0	2.5	7.5	5	45.55	0	5.66	7.72
122	0	0	0	5	7.5	5	47.35	0	9.44	8.27
123	0	0	0	7.5	7.5	5	50.1	0	14.25	9.23
124	0	0	0	10	7.5	5	54.1	0	18.21	10.07
125	0	0	0	12.5	7.5	5	58.9	0	21.76	10.99
126	0	0	0	15	7.5	5	64	0	26.17	11.99
127	0	0	0	0	10	5	49.35	0	4.28	8.48
128	0	0	0	2.5	10	5	50.05	0	7.04	8.8
129	0	0	0	5	10	5	51.15	0	10.29	9.12
130	0	0	0	7.5	10	5	52.05	0	15.12	9.99
131	0	0	0	10	10	5	54.65	0	19.58	10.88
132	0	0	0	12.5	10	5	61.7	0	23.46	12.25

Table C.1. 343 simulation results utilizing RPIW allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
133	0	0	0	15	10	5	69.7	0	26.93	13.51
134	0	0	0	0	12.5	5	53.4	0	6.62	9.44
135	0	0	0	2.5	12.5	5	53.7	0	9.05	9.77
136	0	0	0	5	12.5	5	54.4	0	11.54	10.1
137	0	0	0	7.5	12.5	5	54.55	0	15.55	10.68
138	0	0	0	10	12.5	5	55.6	0	20.39	11.76
139	0	0	0	12.5	12.5	5	67.5	0	24.63	13.5
140	0	0	0	15	12.5	5	77.9	0	28.36	15.11
141	0	0	0	0	15	5	56.2	0	8.93	10.28
142	0	0	0	2.5	15	5	55.75	0	11.19	10.6
143	0	0	0	5	15	5	55.8	0	13.46	10.95
144	0	0	0	7.5	15	5	56.5	0	16.3	11.34
145	0	0	0	10	15	5	57.85	0	21.05	12.61
146	0	0	0	12.5	15	5	74.2	0	25.57	14.82
147	0	0	0	15	15	5	86.2	2.1	31.45	16.9
148	0	0	0	0	0	7.5	41.2	0	1.1	4.54
149	0	0	0	2.5	0	7.5	41.7	0	2.08	6.1
150	0	0	0	5	0	7.5	43.6	0	2.96	7.36
151	0	0	0	7.5	0	7.5	46.7	0	4.1	8.43
152	0	0	0	10	0	7.5	50.35	0	6.34	9.32
153	0	0	0	12.5	0	7.5	53.8	0	10.22	10.06
154	0	0	0	15	0	7.5	56.15	0	14.11	10.77

Table C.1. 343 simulation results utilizing RPIW allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
155	0	0	0	0	2.5	7.5	41.7	0	1.45	6.06
156	0	0	0	2.5	2.5	7.5	42.4	0	4.55	6.73
157	0	0	0	5	2.5	7.5	44.6	0	6.03	7.7
158	0	0	0	7.5	2.5	7.5	48.1	0	7.58	8.52
159	0	0	0	10	2.5	7.5	52.05	0	9.39	9.25
160	0	0	0	12.5	2.5	7.5	55.7	0	13.92	10.01
161	0	0	0	15	2.5	7.5	58.55	0	17.59	10.79
162	0	0	0	0	5	7.5	43.6	0	1.99	7.34
163	0	0	0	2.5	5	7.5	44.35	0	5.36	7.64
164	0	0	0	5	5	7.5	46.35	0	8.87	8.53
165	0	0	0	7.5	5	7.5	49.8	0	12	9.34
166	0	0	0	10	5	7.5	54	0	14.77	10.04
167	0	0	0	12.5	5	7.5	58.05	0	19.24	10.65
168	0	0	0	15	5	7.5	61.45	0	23.37	11.3
169	0	0	0	0	7.5	7.5	46.8	0	2.92	8.49
170	0	0	0	2.5	7.5	7.5	47.5	0	5.93	8.77
171	0	0	0	5	7.5	7.5	48.95	0	9.8	9.35
172	0	0	0	7.5	7.5	7.5	51.25	0	14.49	10.26
173	0	0	0	10	7.5	7.5	54.95	0	18.35	11.05
174	0	0	0	12.5	7.5	7.5	59.4	0	21.94	11.96
175	0	0	0	15	7.5	7.5	64.2	0	26.59	12.9
176	0	0	0	0	10	7.5	50.5	0	4.67	9.55

Table C.1. 343 simulation results utilizing RPIW allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
177	0	0	0	2.5	10	7.5	51.1	0	7.32	9.83
178	0	0	0	5	10	7.5	52.1	0	10.66	10.16
179	0	0	0	7.5	10	7.5	52.8	0	15.57	11.03
180	0	0	0	10	10	7.5	55.7	0	19.9	11.94
181	0	0	0	12.5	10	7.5	62.4	0	23.66	13.25
182	0	0	0	15	10	7.5	70	0	27.12	14.45
183	0	0	0	0	12.5	7.5	54	0	7.07	10.5
184	0	0	0	2.5	12.5	7.5	54.25	0	9.33	10.78
185	0	0	0	5	12.5	7.5	54.95	0	11.84	11.08
186	0	0	0	7.5	12.5	7.5	54.6	0	16.17	11.72
187	0	0	0	10	12.5	7.5	57.3	0	20.88	12.89
188	0	0	0	12.5	12.5	7.5	68.75	0	24.97	14.55
189	0	0	0	15	12.5	7.5	78.5	0	28.57	16.08
190	0	0	0	0	15	7.5	56.35	0	9.4	11.33
191	0	0	0	2.5	15	7.5	55.95	0	11.51	11.62
192	0	0	0	5	15	7.5	56.2	0	13.72	11.93
193	0	0	0	7.5	15	7.5	56.2	0	16.93	12.34
194	0	0	0	10	15	7.5	61	0	21.68	13.85
195	0	0	0	12.5	15	7.5	76.05	0	26.07	15.94
196	0	0	0	15	15	7.5	87.1	2.1	31.71	17.79
197	0	0	0	0	0	10	46.3	0	1.34	5.54
198	0	0	0	2.5	0	10	46.5	0	2.39	7.12

Table C.1. 343 simulation results utilizing RPIW allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
199	0	0	0	5	0	10	47.5	0	3.33	8.37
200	0	0	0	7.5	0	10	49.4	0	4.33	9.41
201	0	0	0	10	0	10	52.05	0	6.67	10.28
202	0	0	0	12.5	0	10	54.85	0	10.48	11.02
203	0	0	0	15	0	10	56.75	0	14.38	11.68
204	0	0	0	0	2.5	10	46.5	0	1.77	7.08
205	0	0	0	2.5	2.5	10	46.9	0	4.77	7.74
206	0	0	0	5	2.5	10	48.15	0	5.91	8.69
207	0	0	0	7.5	2.5	10	50.4	0	7.3	9.46
208	0	0	0	10	2.5	10	53.35	0	9.25	10.15
209	0	0	0	12.5	2.5	10	56.35	0	13.52	10.95
210	0	0	0	15	2.5	10	58.6	0	17.05	11.73
211	0	0	0	0	5	10	47.5	0	2.35	8.35
212	0	0	0	2.5	5	10	48	0	5.68	8.65
213	0	0	0	5	5	10	49.3	0	8.95	9.49
214	0	0	0	7.5	5	10	51.75	0	11.93	10.24
215	0	0	0	10	5	10	55.15	0	14.34	10.88
216	0	0	0	12.5	5	10	58.65	0	19.1	11.46
217	0	0	0	15	5	10	61.55	0	23.09	12.12
218	0	0	0	0	7.5	10	49.45	0	3.16	9.48
219	0	0	0	2.5	7.5	10	50	0	6.22	9.72
220	0	0	0	5	7.5	10	51.1	0	10.15	10.33

Table C.1. 343 simulation results utilizing RPIW allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
221	0	0	0	7.5	7.5	10	52.95	0	14.71	11.18
222	0	0	0	10	7.5	10	56.05	0	18.49	11.93
223	0	0	0	12.5	7.5	10	60.05	0	22.25	12.83
224	0	0	0	15	7.5	10	64.5	0	27.05	13.73
225	0	0	0	0	10	10	52.2	0	5.02	10.51
226	0	0	0	2.5	10	10	52.65	0	7.63	10.75
227	0	0	0	5	10	10	53.5	0	11.02	11.11
228	0	0	0	7.5	10	10	54.1	0	15.96	11.95
229	0	0	0	10	10	10	56.95	0	20.18	12.88
230	0	0	0	12.5	10	10	63.25	0	23.84	14.15
231	0	0	0	15	10	10	70.55	0	27.44	15.31
232	0	0	0	0	12.5	10	55	0	7.45	11.44
233	0	0	0	2.5	12.5	10	55.2	0	9.66	11.69
234	0	0	0	5	12.5	10	55.85	0	12.19	11.95
235	0	0	0	7.5	12.5	10	55.2	0	16.72	12.65
236	0	0	0	10	12.5	10	59.2	0	21.32	13.91
237	0	0	0	12.5	12.5	10	69.95	0	25.25	15.5
238	0	0	0	15	12.5	10	79.1	0	28.73	16.96
239	0	0	0	0	15	10	56.85	0	9.83	12.27
240	0	0	0	2.5	15	10	56.5	0	11.84	12.52
241	0	0	0	5	15	10	56.95	0	13.96	12.8
242	0	0	0	7.5	15	10	55.95	0	17.58	13.27

Table C.1. 343 simulation results utilizing RPIW allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
243	0	0	0	10	15	10	63.95	0	22.25	14.97
244	0	0	0	12.5	15	10	77.7	0	26.47	16.97
245	0	0	0	15	15	10	87.9	1.9	31.76	18.69
246	0	0	0	0	0	12.5	50.45	0	1.54	6.41
247	0	0	0	2.5	0	12.5	50.6	0	2.66	8.01
248	0	0	0	5	0	12.5	51.15	0	3.67	9.27
249	0	0	0	7.5	0	12.5	52.3	0	4.62	10.31
250	0	0	0	10	0	12.5	54.1	0	6.97	11.17
251	0	0	0	12.5	0	12.5	56.2	0	10.73	11.89
252	0	0	0	15	0	12.5	57.65	0	14.62	12.52
253	0	0	0	0	2.5	12.5	50.6	0	2.04	7.97
254	0	0	0	2.5	2.5	12.5	50.8	0	4.93	8.63
255	0	0	0	5	2.5	12.5	51.55	0	5.92	9.57
256	0	0	0	7.5	2.5	12.5	53	0	6.94	10.32
257	0	0	0	10	2.5	12.5	55.05	0	9.18	11.03
258	0	0	0	12.5	2.5	12.5	57.35	0	13.09	11.83
259	0	0	0	15	2.5	12.5	59	0	16.51	12.6
260	0	0	0	0	5	12.5	51.15	0	2.68	9.25
261	0	0	0	2.5	5	12.5	51.45	0	6.01	9.55
262	0	0	0	5	5	12.5	52.3	0	8.98	10.35
263	0	0	0	7.5	5	12.5	54.05	0	11.72	11.04
264	0	0	0	10	5	12.5	56.6	0	14.09	11.64

Table C.1. 343 simulation results utilizing RPIW allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
265	0	0	0	12.5	5	12.5	59.45	0	18.97	12.19
266	0	0	0	15	5	12.5	61.85	0	22.75	12.87
267	0	0	0	0	7.5	12.5	52.35	0	3.45	10.37
268	0	0	0	2.5	7.5	12.5	52.75	0	6.52	10.58
269	0	0	0	5	7.5	12.5	53.55	0	10.44	11.21
270	0	0	0	7.5	7.5	12.5	54.95	0	14.9	12.02
271	0	0	0	10	7.5	12.5	57.5	0	18.61	12.73
272	0	0	0	12.5	7.5	12.5	60.9	0	22.65	13.61
273	0	0	0	15	7.5	12.5	64.85	0	27.52	14.45
274	0	0	0	0	10	12.5	54.2	0	5.35	11.38
275	0	0	0	2.5	10	12.5	54.6	0	7.91	11.6
276	0	0	0	5	10	12.5	55.3	0	11.38	11.98
277	0	0	0	7.5	10	12.5	55.75	0	16.32	12.8
278	0	0	0	10	10	12.5	58.45	0	20.43	13.76
279	0	0	0	12.5	10	12.5	64.2	0	24.01	14.98
280	0	0	0	15	10	12.5	71.25	0	27.84	16.11
281	0	0	0	0	12.5	12.5	56.35	0	7.82	12.3
282	0	0	0	2.5	12.5	12.5	56.55	0	9.94	12.53
283	0	0	0	5	12.5	12.5	57.1	0	12.54	12.76
284	0	0	0	7.5	12.5	12.5	56.45	0	17.24	13.51
285	0	0	0	10	12.5	12.5	61.15	0	21.68	14.84
286	0	0	0	12.5	12.5	12.5	71.15	0	25.5	16.38

Table C.1. 343 simulation results utilizing RPIW allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
287	0	0	0	15	12.5	12.5	79.8	0	28.87	17.79
288	0	0	0	0	15	12.5	57.65	0	10.23	13.11
289	0	0	0	2.5	15	12.5	57.45	0	12.15	13.34
290	0	0	0	5	15	12.5	58.05	0	14.18	13.6
291	0	0	0	7.5	15	12.5	56.7	0	18.17	14.14
292	0	0	0	10	15	12.5	66.65	0	22.73	15.98
293	0	0	0	12.5	15	12.5	79.2	0	26.8	17.91
294	0	0	0	15	15	12.5	88.7	1.65	31.7	19.53
295	0	0	0	0	0	15	54	0	1.71	7.19
296	0	0	0	2.5	0	15	54.1	0	2.9	8.81
297	0	0	0	5	0	15	54.4	0	3.97	10.08
298	0	0	0	7.5	0	15	55.15	0	4.97	11.13
299	0	0	0	10	0	15	56.4	0	7.25	12
300	0	0	0	12.5	0	15	57.9	0	10.97	12.71
301	0	0	0	15	0	15	58.9	0	14.86	13.31
302	0	0	0	0	2.5	15	54.1	0	2.29	8.76
303	0	0	0	2.5	2.5	15	54.25	0	5.07	9.43
304	0	0	0	5	2.5	15	54.75	0	5.88	10.37
305	0	0	0	7.5	2.5	15	55.65	0	6.65	11.12
306	0	0	0	10	2.5	15	57.05	0	8.97	11.85
307	0	0	0	12.5	2.5	15	58.7	0	12.67	12.65
308	0	0	0	15	2.5	15	59.8	0	16.01	13.41

Table C.1. 343 simulation results utilizing RPIW allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
309	0	0	0	0	5	15	54.4	0	2.98	10.06
310	0	0	0	2.5	5	15	54.65	0	6.29	10.36
311	0	0	0	5	5	15	55.25	0	9.14	11.13
312	0	0	0	7.5	5	15	56.45	0	11.49	11.77
313	0	0	0	10	5	15	58.35	0	14	12.33
314	0	0	0	12.5	5	15	60.55	0	18.79	12.86
315	0	0	0	15	5	15	62.4	0	22.37	13.56
316	0	0	0	0	7.5	15	55.2	0	3.79	11.18
317	0	0	0	2.5	7.5	15	55.5	0	6.87	11.36
318	0	0	0	5	7.5	15	56.1	0	10.7	12.02
319	0	0	0	7.5	7.5	15	57.15	0	15.06	12.78
320	0	0	0	10	7.5	15	59.15	0	18.68	13.44
321	0	0	0	12.5	7.5	15	61.95	0	23.08	14.29
322	0	0	0	15	7.5	15	65.2	0	28.04	15.04
323	0	0	0	0	10	15	56.45	0	5.65	12.19
324	0	0	0	2.5	10	15	56.8	0	8.16	12.38
325	0	0	0	5	10	15	57.35	0	11.77	12.78
326	0	0	0	7.5	10	15	57.75	0	16.64	13.59
327	0	0	0	10	10	15	60.15	0	20.67	14.57
328	0	0	0	12.5	10	15	65.25	0	24.16	15.74
329	0	0	0	15	10	15	71.95	0	28.27	16.82
330	0	0	0	0	12.5	15	58	0	8.16	13.1

Table C.1. 343 simulation results utilizing RPIW allocation method. From left to right: Maneuver number, x_0 (deg), y_0 (deg), z_0 (deg), x_f (deg), y_f (deg), z_f (deg), maneuver time (sec), saturation duration (sec), commanded momentum exchange error (Nm-s), consumed energy (W-Hr).

1	0	0	0	0	0	0	10	0	0	0.23
331	0	0	0	2.5	12.5	15	58.2	0	10.2	13.29
332	0	0	0	5	12.5	15	58.65	0	12.91	13.53
333	0	0	0	7.5	12.5	15	58.15	0	17.69	14.31
334	0	0	0	10	12.5	15	63.1	0	22.01	15.7
335	0	0	0	12.5	12.5	15	72.35	0	25.72	17.2
336	0	0	0	15	12.5	15	80.6	0	29.01	18.57
337	0	0	0	0	15	15	58.8	0	10.6	13.9
338	0	0	0	2.5	15	15	58.75	0	12.44	14.11
339	0	0	0	5	15	15	59.3	0	14.45	14.34
340	0	0	0	7.5	15	15	58.45	0	18.71	14.97
341	0	0	0	10	15	15	69.05	0	23.16	16.93
342	0	0	0	12.5	15	15	80.6	0	27.08	18.77
343	0	0	0	15	15	15	89.55	1.3	31.56	20.31

Bibliography

[1] T. A. Johansen and T. I. Fossen, "Control allocation - a survey," *Automatica*, vol. 49, pp. 1087–1103, 2013.

[2] AFRL/RV: Guidance, Navigation, and Control Group. Used with permission. Credit: AFRL, 2017.

[3] M. A. Peck and A. R. Cavender, "An airbearing-based testbed for momentum-control systems and spacecraft line of sight," in *Proceedings of the 13 th AAS/AIAA Space Flight Mechanics Winter Meeting, no. AAS 03-127*, 2003.

[4] AFIT: Mechanical Engineering Department. Used with permission. Credit: Phillip Smith, 2017.

[5] M. W. Oppenheimer *et al.*, "Control allocation," in *Control System Applications*, W. S. Levine, Ed. College Park, MD, USA: CRC Press, 2011, ch. 8.

[6] *EC 60∅ mm, Brushless, 400 Watt Motor Spec Sheet*, P/N: 167132, Maxon Motor, April 2016.

[7] O. Harkegard and S. T. Glad, "Resolving actuator redundancy optimal control vs. control allocation," *Automatica*, vol. 41, pp. 137–144, 2005.

[8] D. Huston, "Hierarchical actuator systems," in *Proceedings of the SPIE*, vol. 5762, 2005, pp. 311–319.

[9] M. Valásek, "Design and control of under-actuated and over-actuated mechanical systems - challenges of mechanic and mechatronics," *Vehicle System Dynamics*, vol. 40, pp. 37–49, 2003.

[10] "Introduction," class notes for Chemical Rocket Propulsion, Dept. of Aeronautics and Astronautics , Air Force Institute of Technology, Wright-Patterson AFB, OH, winter 2016.

[11] T. S. Brown, "The cassini reaction wheels: Drag and spin-rate trends from an aging interplanetary spacecraft at saturn," in *AIAA Guidance, Navigation, and Control Conference*, 2016.

[12] D. Penn, "Characterization and modeling of a control moment gyroscope," Master's thesis, Air Force Institute of Technology, Wright Patterson AFB, Ohio, 2015.

[13] J. L. Schwartz *et al.*, "Historical review of air-bearing spacecraft simulators," *Journal of guidance, control, and Dynamics*, vol. 26, pp. 513–522, 2003.

[14] D. L. Kunz, *MECH 521 Intermediate Dynamics Text.* Wright-Patterson AFB, OH: AFIT, 2015.

[15] "Torque free rigid body motion," class notes for Intermediate Spacefilght Dynamics, Dept. of Aeronautics and Astronautics , Air Force Institute of Technology, Wright-Patterson AFB, OH, spring 2016.

[16] "Reaction wheels," class notes for Intermediate Spacefilght Dynamics, Dept. of Aeronautics and Astronautics , Air Force Institute of Technology, Wright-Patterson AFB, OH, spring 2016.

[17] K. A. Bordignon and W. C. Durham, "Closed-form solution to constrained control allocation problem," *Journal of Guidance Control and Dynamics*, vol. 18, pp. 1000–1007, 1995.

[18] M. W. Oppenheimer *et al.*, "Control allocation for over-actuated systems," *2006 IEEE International Conference on Industrial Technology*, pp. 1–6, 2006.

[19] M. Bodson, "Evaluation of optimization methods for control allocation," *Journal of guidance, control, and Dynamics*, vol. 25, pp. 703–711, 2002.

[20] K. A. Bordignon and J. Bessolo, "Control allocation for the x-35b," *Journal of Guidance Control and Dynamics*, vol. 18, pp. 1000–1007, 1995.

[21] J. Jin, "Modified pseudoinverse redistribution methods for redundant controls allocation," *Journal of Guidance, Control, and Dynamics*, vol. 28, pp. 1076–1079, 2005.

[22] S. S. W. C. Arun Kishore and G. Ray, "Disturbance rejection and control allocation of over-actuated systems," *Control and Automation, 2006. MED '06. 14th Mediterranean Conference on*, pp. 1054–1059, 2006.

[23] J. Herner, private communication, 2017.

[24] *ode45 Functin Description*, MathWorks, 2017.

[25] "Attitude control part i and ii," class notes for Intermediate Spacefilght Dynamics, Dept. of Aeronautics and Astronautics , Air Force Institute of Technology, Wright-Patterson AFB, OH, spring 2016.

[26] "Linear algebra and multivariate linear systems," class notes for Robust Control, Dept. of Aeronautics and Astronautics , Air Force Institute of Technology, Wright-Patterson AFB, OH, spring 2016.

[27] *maxon DC motor and maxon EC motor Key information*, Maxon Motor, November 2014.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704–0188*

1. REPORT DATE *(DD–MM–YYYY)* 3 Mar 2017	**2. REPORT TYPE** Master's Thesis	**3. DATES COVERED** *(From — To)* Sept 2016 — Mar 2016

4. TITLE AND SUBTITLE CONTROL ALLOCATION METHODS FOR CONSTRAINED AND OVER ACTUATED SATELLITE ATTITUDE CONTROL SYSTEMS	**5a. CONTRACT NUMBER** **5b. GRANT NUMBER** **5c. PROGRAM ELEMENT NUMBER**
6. AUTHOR(S) Childress, Jonathan M.	**5d. PROJECT NUMBER** **5e. TASK NUMBER** **5f. WORK UNIT NUMBER**
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB OH 45433-7765	**8. PERFORMING ORGANIZATION REPORT NUMBER** AFIT-ENY-MS-17-M-215
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory, Space Vehicles Directorate, Guidance, Navigation, and Controls Group 3550 Aberdeen Ave SE Kirtland AFB NM 87117	**10. SPONSOR/MONITOR'S ACRONYM(S)** AFRL/RVSV **11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION / AVAILABILITY STATEMENT

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

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The research presented in this thesis compares the numerically simulated performance of various control allocation methods applied to the constrained and over actuated control system of the Air Force Research Laboratory, Space Vehicle Directorate's satellite simulator, the REBEL. The REBEL prototype reaction wheel array design has not yet evaluated the implementation of control allocation methods in its attitude control system. As a result, three different control allocation methods are formulated and applied to the simulation: the simple pseudo inverse, the redistributed pseudo inverse, and the redistributed pseudo inverse with adaptive weighting (a newly developed control allocation variant.) The equations of motion (kinematics and kinetics) for a satellite simulator utilizing a reaction wheel array as an attitude control system are also derived and implemented in the simulation. The control law for a basic high level controller is defined and applied to the simulation as well. A single simulated maneuver is executed with the control system utilizing each of the control allocation methods independently. This single maneuver is evaluated in detail to provide insight into the control allocation methods' functionality. A set of multiple maneuvers is also executed for each control allocation method and the resulting performance metrics are averaged. Based on these results, conclusions are drawn about the different control allocation methods' potential for implementation on REBEL. The results indicate that redistributed pseudo inverse and the redistributed pseudo inverse with adaptive weighting both show promise as potential control allocation methods REBEL could utilize.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Dr. Eric Swenson, AFIT/ENY
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(include area code)*
U	U	U	UU	177	(937) 255-3636, x7479; eric.swenson@afit.edu