12-2017

Optimized Thruster Allocation Utilizing Dual Quaternions for the Asteroid Sample Return Mission (OSIRIS-REx)

Asher Smith

OPTIMIZED THRUSTER ALLOCATION UTILIZING DUAL QUATERNIONS FOR THE ASTEROID SAMPLE RETURN MISSION (OSIRIS-REx)

A Thesis

Submitted to the Faculty

of

Embry-Riddle Aeronautical University

by

Asher Smith

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Aerospace Engineering

December 2017

Embry-Riddle Aeronautical University

Daytona Beach, Florida

OPTIMIZED THRUSTER ALLOCATION UTILIZING DUAL QUATERNIONS FOR THE ASTEROID SAMPLE RETURN MISSION (OSIRIS-REx)

by

Asher Smith

A Thesis prepared under the direction of the candidate's committee chairman, Dr. Dongeun Seo, Department of Aerospace Engineering, and has been approved by the members of the thesis committee. It was submitted to the School of Graduate Studies and Research and was accepted in partial fulfillment of the requirements for the degree of Master of Science in Aerospace Engineering.

THESIS COMMITTEE

Chairman, Dr. Dongeun Seo

Member, Dr. Troy Henderson

Member, Dr. Richard Prazenica

Graduate Program Coordinator, Dr. Magdy Attia

12·1·2017
Date

Dean of College of Engineering, Dr. Maj Mirmirani

12/1/2017
Date

Vice Chancellor, Academic Support, Dr. Christopher Grant

12/1/2017
Date

ACKNOWLEDGMENTS

Author thanks Landis Francis Markley from NASA Goddard Space Flight Center for help on the initial optimized distribution matrix theory, Paul Mason from NASA Goddard Space Flight Center, Attitude Controls Systems Branch, for the help with the simulation development, thruster distribution theory, and thesis editing, and Eric T. Stoneking from NASA Goddard Space Flight Center, Attitude Controls System Branch, for the base development of the 42 simulation software.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SYMBOLS

μ	Gravitation constant of the orbiting body
τ	Thruster force vector
τ_{pos}	Thruster force vector shifted to purely positive values
τ_{burn}	Thruster fire time vector
$\widehat{\omega}$	Dual velocity quaternion
D	Thruster distribution matrix
F_{des}	Desired force and torque vector
F_{pos}	Desired force and torque vector in purely positive form
$\hat{f}^B$	External forces and torques in the body frame
$\hat{f}_g^B$	External force due to gravity
$\hat{f}_{gg}^B$	External force due to gravity gradient
$\hat{f}_c^B$	Control input force
I^B	Mass moment of inertia matrix
m	Mass of body
M	Non-optimized thruster distribution matrix
M^B	Dual inertia matrix
q	Quaternion with vector and scalar component in the form $[\vec{q}, q_4]$
$\hat{q}$	Dual position quaternion denoted as $[q_r, q_d]$
q_r	Real quaternion relating to position
q_d	Quaternion relating to attitude
$\vec{r}$	Position vector
T_{dir}	Thruster direction matrix
T_N	Torques produced by the thrusters resulting from vector cross product of attitudes and locations
V_{null}	Matrix whose columns span the null space of the non-optimized distribution matrix

ABBREVIATIONS

AD&CS	Attitude Dynamics and Controls System
DCM	Direction Cosine Matrix
DOF	Degrees of Freedom
OSIRIS-REx	Origins Spectral Interpretation Resource Identification Security Regolith Explorer
TAG	Touch and Go

ABSTRACT

Smith, Asher MSAE, Embry-Riddle Aeronautical University, December 2017. Optimized Thruster Allocation Utilizing Dual Quaternions for the Asteroid Sample Return Mission (OSIRIS-REx).

As spacecraft require higher positional accuracy from the attitude control systems, new algorithm developments, along with sensor and actuator resolution and range improvements are necessary to achieve the desired science accuracies. For agile 6-Degrees of freedom (6-DOF) spacecraft with redundancy, the actuators are usually oversized or overpopulated to meet the desired slew requirements. Currently, most spacecraft utilize an over-actuated thruster system to produce 6-DOF control. This thesis presents a simulation of the OSIRIS-REx mission during the descent phase to the asteroid Bennu, with a focus on utilizing dual quaternion dynamics and a newly developed thruster allocation method. The dual quaternion based dynamics are chosen in order to demonstrate its feasibility in real-time applications. Contrary to typical plant dynamics, which decouple the spacecraft orbit and attitude dynamics, the dual quaternion description provides a compact and coupled dynamics system. Due to the coupled nature of dual quaternions, a newly developed thruster distribution matrix is implemented to take both the coupled command body forces and torques and transform them into the individual thruster frames. The developed method is based on a min-max optimization that results in a constant thruster distribution matrix. From the optimization, a minimum thrust solution is calculated for the coupled position and attitude commands. Therefore, its integration into the dual quaternion dynamics is intuitive and simplistic. The final result is a computationally fast thruster allocation solution for real-time applications.

1. Introduction

In most NASA class B or higher missions, the spacecraft bus is required to have some (if not all) actuator redundancy. This is particularly true for pose missions (attitude and positional control). Due to their capability to provide simultaneous positional and attitude control, the primary actuator that is used within large agile spacecraft today is thrusters. Additionally, thrusters are easily used in combination with one another, providing an easy solution to the necessity of redundancy. If enough thrusters are utilized, full six degree-of-freedom (6-DOF) control is possible.

When designing the majority of current spacecraft control problems, there are two separate control requirements that must be accounted for: the positional control, and the attitude control. Strictly speaking, the positional control problem deals with translational motion such as orbital maintenance or injection, and the attitude control problem deals with the rotational requirements of the spacecraft. For most current design processes, the dynamics are assumed uncoupled to reduce complexity and allow for a simple solution. However, the true physics of the problem are coupled and do not separate out the positional and attitude dynamics into separate systems. Therefore, a design based on the coupled dynamics can produce a more realistic solution to meet the tighter requirements of future missions. Since the attitude dynamics are inherently non-linear, the ability to design and verify a coupled dynamics control system is a function of a higher fidelity simulation. The coupled positional and attitude dynamics system lead to the dual quaternion description due to its compactness and relative simplicity when describing the dynamics (Seo, 2015).

A typical dual quaternion is composed of two separate quaternions combined into a single dual quaternion. One of these quaternions describes the translational motion of the spacecraft, while the second describes the attitude based motion. This dual quaternion description then inherits all the general benefits provided by a regular quaternion such as its non-singular implementation of three-dimensional attitude dynamics. From this, the dual quaternion is able to simultaneously describe both the translational and rotational dynamics, which results in it being a viable alternative 6-DOF representation of the spacecraft control problem. Additionally, it has been shown to be the most compact and computationally efficient tool when compared to others (Aspragathos & Dimitros, 1998), (Funda & Paul, 1990), and (Funda, Taylor, & Paul, 1990).

Inherent within the 6-DOF dual quaternion description is a requirement of a coupled 6-DOF command input. To properly utilize thrusters to complete this 6-DOF approach, a process known as control allocation is utilized, which is the connecting link between the control laws to the individual actuators.

Currently, there is a division of three main methods for completing the thruster allocation. The first is a selection matrix method, also known as the decoupled method. This method is based on having multiple tables of thruster combinations for different commands, such as a slew about a single axis. These tables can then be combined and proportionally scaled depending on the input command. Most current spacecraft utilize this design approach due to its intuitiveness, simplicity, and computational efficiency. However, this method is unable to account for a non-regular thruster configuration, such as a non-axisymmetric layout due to the inherent axis coupling effect (Wang M., 2009).

The second method is a process of Linear Programming. These routines generally use an optimizing scenario such as the Simplex method. This process provides an optimal thruster allocation given a command input (Bodson, 2002) and (Wang M., 2009). However, the Simplex method is rarely used in flight computers for thruster-based allocation due to the optimization routine running in the control loop algorithm. Therefore, the on-board processing power required is generally too high to be practical (Ankersen, Aleshin, Vankov, & Volochinov, 2005), (Crawford, 1969), (Doman, Gambel, & Ngo, 2007), and (Wang M., 2009).

This thesis proposes a third thruster allocation method, which is a combination of the first two methods in order to reap the benefits of both methods while mitigating many of their individual drawbacks. The proposed method does this by calculating a single optimized thruster distribution matrix. It provides the benefits of being an optimized thruster allocation method and therefore is more efficient that the typical thruster selection method.

Additionally, since the proposed method results in a single optimized thruster distribution matrix, it requires less memory storage on board the spacecraft computers when compared to the thruster selection method. This is due to the thruster selection method requiring multiple lookup tables for the various spacecraft maneuvers, which all must be individually stored on the spacecraft computers.

Due to the optimization and the single resulting distribution matrix, the axis coupling effects of the thrusters are inherently included in the thruster firing calculations. In addition, because of the calculations used to form the distribution matrix, it accounts for asymmetries of the thrusters about the center of mass and its effect on the thruster torques. In comparison to the selection matrices, which require additional post processing to account for asymmetries about the center of mass, the distribution matrix automatically accounts for all of these characteristics.

The basis of the proposed method is a pseudo inverse distribution matrix with an additional optimization. Non-optimized pseudo inverse methods have been proposed previously in propulsion systems such as underwater vehicles (Garus, 2004) and aircraft (Johansen & Fossen, 2013). However, the proposed method optimizes the given pseudo inverse allocation matrix which results in a constant distribution matrix. Additionally, it does not require calculations within the control loop. Therefore, it is very applicable to real-time applications due to its simplicity and computational speed (Smith & Seo, 2017).

2. Dual Quaternion Dynamics

The driving purpose of utilizing the dual quaternion is to present the most compact and computationally efficient 6-DOF simulation. Additionally, due to its dual nature, the kinematics and dynamics of the system are coupled and therefore allow for a higher fidelity simulation (Seo, 2015). Furthermore, the dual format still inherits all the traditional benefits of a regular quaternion such as avoiding a gimbal lock situation.

The original concept of the dual quaternion was derived from the idea of describing the dynamics of a system with a screw motion which includes a rotation and translation along the rotation axis. Typically, the 6-DOF system is described with a separate translation vector and Euler angles or a quaternion. However, one method to combine these two components into a single entity is through the idea of a screw motion.

Originally characterized by Chasles' theorem, the screw motion can be described in full by utilizing two characteristics, the screw axis and the screw pitch. These characteristics directly correspond to the rotation and translation of a system (Wu, Hu, Hu, Li, & Lian, 2005). Utilizing this concept to express velocities in a 3D space was originally conceptualized to derive a mathematical method to describe the screw motion; the result is known as the dual quaternion.

2.1. Quaternions Overview

The concept of quaternions was originally created by Hamilton as a rotational transformation utilizing a redundant set in order to avoid singularities generally posed by full sets such as Euler angles. The general formulation of a quaternion is as follows (Wu, Hu, Hu, Li, & Lian, 2005).

$$q = \begin{bmatrix} \vec{q} \\ q_4 \end{bmatrix} \qquad (1)$$

where $\vec{q} = [q_1, q_2, q_3]^T \in \mathbb{R}^3$ is the vector component of the quaternion and $q_4 \in \mathbb{R}$ is the scalar component. Herein, a vector quaternion refers to a quaternion with a zero-scalar component, and a scalar quaternion refers to a quaternion with a zero-vector component.

To properly utilize the quaternion in the equations of motion, a set of binary operators defined in Table (2.1) is utilized to manipulate the values in the context of quaternion algebra.

Table 2.1 Binary quaternion operators where a, b are quaternions (Seo, 2015)

Addition	$a + b = \left[\vec{a} + \vec{b}, a_4 + b_4\right]$
Scalar Product	$\lambda a = [\lambda\vec{a}, \lambda a_4]$
Product	$ab = \left[a_4\vec{b} + b_4\vec{a} + \vec{a}\text{x}\vec{b}, a_4b_4 - \vec{a}\cdot\vec{b}\right]$
Conjugate	$a^* = [-\vec{a}, a_4]$
Dot Product	$a \cdot b = \left[\vec{0}, \vec{a}\cdot\vec{b} + a_4b_4\right]$
Cross Product	$a \text{ x } b = \left[b_4\vec{a} + a_4\vec{b} + \vec{a} \text{ x } \vec{b}, 0\right]$
Norm	$\lVert a \rVert^2 = aa^* = a^*a = aa$

The purpose of quaternions is to describe a 3 dimensional rotation in 3-sphere. For a frame rotation about a generic unit axis n, describing a rotation from on frame denoted as B, to another frame, D, it can be described by the following

$$q_{B/D} = \left[\sin\left(\frac{\theta}{2}\right) n, \cos\left(\frac{\theta}{2}\right)\right] \tag{2}$$

To describe a rotation from one frame such as r^B to another frame such as r^D, the quaternion can be used as by pre-multiplying by the quaternion conjugate and post multiplying by the quaternion utilizing the quaternion product defined previously (Wu, Hu, Hu, Li, & Lian, 2005).

$$r^D = q_{B/D}{}^{*} r^B q_{B/D} \tag{3}$$

By taking the time derivative of the quaternion, the following kinematic equations can be derived.

$$\dot{q} = \left[-\frac{1}{2}(\vec{q}^{x}\omega + q_4\omega), -\frac{1}{2}\vec{q}^{T}\omega\right] \tag{4}$$

where $\vec{q}^{x}$ is defined as the skew symmetric matrix and ω is the angular velocity of the system.

$$\vec{q}^{x} = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix} \tag{5}$$

Additionally, ω is the rotational dynamics of the system. Typically, it is defined using Euler's Equations as follows

$$I\dot{\omega} + \omega \,\mathrm{x}\, I\omega = \tau \tag{6}$$

where I is the moment of inertia matrix, and τ is any external torque applied to the system.

2.2. Dual Numbers

The concept of a dual number is defined and used to develop the dual quaternion. They have the following characteristics. The nomenclature and derivations have been taken from (Wu, Hu, Hu, Li, & Lian, 2005).

$$\hat{z} = a + \varepsilon b \tag{7}$$

where $\varepsilon^2 = 0$ but $\varepsilon \neq 0$. In general, a is denoted as the real part of the dual number and b is denoted as the dual part. The dual numbers have a distinct set of operators as defined in (Wu, Hu, Hu, Li, & Lian, 2005).

Table 2.2 Dual number operators

Addition	$\hat{z}_1 + \hat{z}_2 = a_1 + b_1 + \varepsilon(b_1 + b_2)$
Scalar Product	$\lambda\hat{z} = \lambda a + \varepsilon\, \lambda b$
Product	$\hat{z}_1\hat{z}_2 = a_1 a_2 + \varepsilon(a_1 b_2 + a_2 b_1)$

2.3. Dual Vectors

The concept of a dual vector is very similar to the dual numbers. However, instead of containing two separate scalar values, it contains two vectors. The real part of the dual vector denotes the unit direction and the dual part denotes the rotation with respect to the origin of the coordinate frame. Herein the "dual" form is denoted with the caret, ^, such as $\hat{q}$.

A typical unit dual vector is known as a Plücker line (Wu, Hu, Hu, Li, & Lian, 2005). In general, line I can be described as a Plücker line $\hat{I} = I + \varepsilon m$. Geometrically, I is a unit vector, and m is the rotation about the axis defined by p x I, and is normal to the plane containing the line I and the frame's origin. This is directly linked to Chasles theorem, which states the general displacement of a rigid body can be described by a rotation about and axis, typically known as the screw axis, and a translation parallel to that axis. For the defined Plücker line, the screw axis is m and the translation is p as seen in Figure 2.1 (Wu, Hu, Hu, Li, & Lian, 2005)

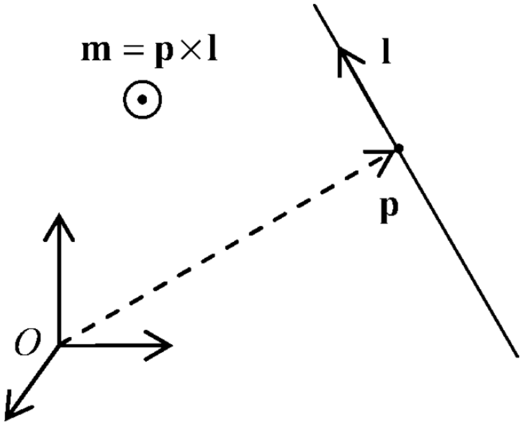

Figure 2.1 Plücker line with rotation pointing out of the page (Wu, Hu, Hu, Li, & Lian, 2005)

Additionally, the utilization of Plücker lines can describe the relationship between two separate lines. One such example shares many similarities to the dot product of two unit vectors. The scalar product between two separate Plücker lines can be shown to be equal to the cosine of the dual angle.

$$\hat{\theta} = \theta + \varepsilon d \tag{8}$$

where θ is the crossing angle and d is the common perpendicular distance between the two lines.

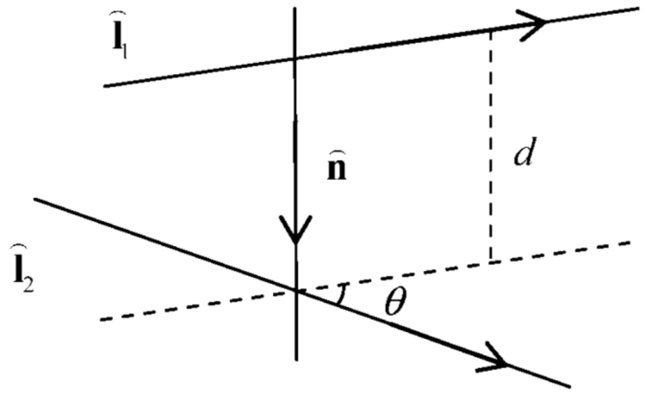

Figure 2.2 Relationship between Plücker lines (Wu, Hu, Hu, Li, & Lian, 2005)

In general, this relationship can be formed by the following.

$$\hat{\text{I}}_1\hat{\text{I}}_2 = \cos(\hat{\theta}) \tag{9}$$

where $\hat{I}_1 = I_1 + \varepsilon m_1$ and $\hat{I}_2 = I_2 + \varepsilon m_2$. Lastly, $\hat{I}_1 x\ \hat{I}_1 = sin(\hat{\theta})\,\hat{n}$. The new Plücker $\hat{n}$ is the common perpendicular intersecting line.

2.4. Dual Quaternions

At its fundamental level, a dual quaternion is a regular quaternion but with dual number components included.

$$\hat{q} = [\hat{\vec{q}}, \hat{q}_4] \tag{10}$$

where $\hat{\vec{q}}$ is a dual vector and $\hat{q}_4$ is a dual number. Additionally, it can be written as follows

$$\hat{q} = q_r + \varepsilon q_d \tag{11}$$

where q_r denotes the real part (rotation) of the system, and q_d denotes the dual part (translation) of the system. In general, the dual vector can be considered the same as a dual quaternion with a 0 scalar component. Also, due to the formulation of the dual quaternion utilizing two separate quaternions, it inherits all of the benefits of the traditional single quaternion (Wu, Hu, Hu, Li, & Lian, 2005). However, as with quaternions, the dual quaternion has specific operators as follows.

Table 2.3 Binary quaternion operators where $\hat{a}$, $\hat{b}$ are dual quaternions and $\lambda \in \mathbb{R}$ (Seo, 2015)

Addition	$\hat{a} + \hat{b} = [a_r + b_r, a_d + b_d]$
Scalar Multiplication	$\lambda\hat{a} = \lambda a_r + \lambda a_d$
Product	$\hat{a} *_d \hat{b} = \left[a_r *_q b_r, a_r *_q b_d + a_d *_q b_r\right]$
Swap	$\hat{a}^\dagger = [a_d, a_r]$
Conjugate	$\hat{a}^* = [a_r^*, a_d^*]$
Dot Product	$\hat{a} \cdot \hat{b} = [a_r \cdot b_r, a_r \cdot b_d + a_d \cdot b_r]$
Cross Product	$\hat{a}\ \mathrm{x}\ \hat{\mathrm{b}} = [a_r\ \mathrm{x}\ b_r, a_r\ \mathrm{x}\ b_d + a_d\ \mathrm{x}\ b_r]$
Norm	$\|\hat{a}\| = \hat{a}\hat{a}^*$

By taking two separate frames, D and B, the relationship between them can be described by either a rotation, q, followed by a translation, r^D, or the opposite with a translation, r^B, followed by a rotation, q. From here it can be shown using Eqn. (3), that $r^D = q^* r^B q$, and using this, a Plücker line can be defined which satisfies $\hat{l}^D = \hat{q}^* \hat{l}^B \hat{q}$. Herein, the unit dual quaternion is a function of both the rotation q and the translation r^D, or the opposite translation r^B and rotation q.

To derive this, the following steps were adapted from (Wu, Hu, Hu, Li, & Lian, 2005). Initially, two Plücker lines expressed in the separate D and B frames are taken as follows. $\hat{I}^D = I^D + \varepsilon m^D$ and $\hat{I}^B = I^B + \varepsilon m^B$,

$$\begin{aligned} I^B &= q^* I^D q \\ m^D &= p^D \text{ x } I^D \\ &= (q^* \, p^B q - r^D) \text{ x } (q^* I^B q) \\ &= q^* \, m^B q - r^D \text{ x } (q^* I^B q) \\ &= q^* m^B q + \frac{1}{2}(r^{D*} q^* I^B q + q^* I^B q r^D) \end{aligned} \tag{12}$$

If a new quaternion is defined as $q^t = \frac{1}{2} q r^D$, and a dual quaternion is defined as $\hat{q} = q + \varepsilon q^t$. It is possible to show that Eqn. (11) is equivalent to the following:

$$\hat{I}^B + \varepsilon m^b = (q + \varepsilon q^t)^* *_d (I^D + \varepsilon m^D) *_d (q + \varepsilon q^t) \tag{13}$$

So, $\hat{I}^B = \hat{q}^* *_d \hat{I}^D *_d \hat{q}$, where the dual quaternion from here is the following:

$$\begin{aligned} \hat{q} &= q + \varepsilon q^t \\ &= q + \varepsilon \frac{1}{2} q r^D \\ &= q + \varepsilon q q^* r^B q \\ &= q + \varepsilon \frac{1}{2} r^D q \end{aligned} \tag{14}$$

As stated previously, the dual quaternion motion is tied directly to the concept of screw motion. Therefore, it can be written directly as a function of the individual screw elements.

$$\hat{q} = \left[\sin\left(\frac{\hat{\theta}}{2}\right)n, \cos\left(\frac{\hat{\theta}}{2}\right)\right] \tag{15}$$

where n denotes the screw axis and $\hat{\theta}$ is the dual angle of the screw motion. The derivation for this form can be found geometrically from Figure 2.3, and it was also adapted from (Wu, Hu, Hu, Li, & Lian, 2005).

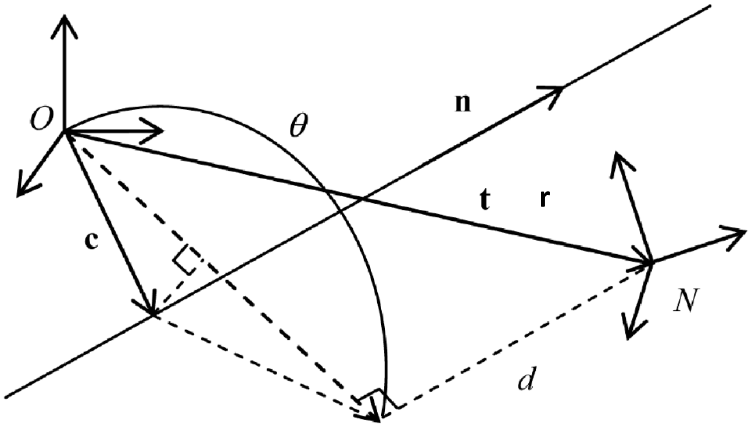

Figure 2.3 Visualization of the screw motion with a rotation θ about the n axis at point c. Followed by a translation r (Wu, Hu, Hu, Li, & Lian, 2005).

To derive Eqn. (14), initially a dual quaternion is defined.

$$\hat{q} = q + \varepsilon\frac{1}{2}q \tag{16}$$

Using Figure 2.3, and solving for some of the geometry the segment c can found

$$c = \frac{1}{2}\left(r - dn + \cot\left(\frac{\theta}{2}\right) n \text{ x } r\right) \tag{17}$$

where d is the pitch of the screw motion derived as

$$d = r \cdot n \tag{18}$$

Using Eqn. (16), it can be shown that the following statement is true.

$$\sin\left(\frac{\theta}{2}\right) c \text{ x } n + \frac{d}{2}\cos\left(\frac{\theta}{2}\right) n = \frac{1}{2}\left(\sin\left(\frac{\theta}{2}\right) r \text{ x } n + \cos\left(\frac{\theta}{2}\right) r\right) \tag{19}$$

Taking the quaternion $q = \left[\sin\left(\frac{\theta}{2}\right) n, \cos\left(\frac{\theta}{2}\right)\right]$ into consideration with Eqn. (19), the following steps can be taken

$$\begin{aligned}\hat{q} &= q + \varepsilon \frac{1}{2} q \\ &= \left[\sin\left(\frac{\theta}{2}\right) n, \cos\left(\frac{\theta}{2}\right)\right] + \varepsilon \frac{1}{2}\left[\cos\left(\frac{\theta}{2}\right) r + \sin\left(\frac{\theta}{2}\right) r \text{ x } n, -d \sin\left(\frac{\theta}{2}\right)\right] \\ &= \left[\sin\left(\frac{\theta}{2}\right) n + \varepsilon\left(\frac{d}{2}\cos\left(\frac{\theta}{2}\right) n + \sin\left(\frac{\theta}{2}\right) c \text{ x } n\right), \cos\left(\frac{\theta}{2}\right) - \varepsilon \frac{d}{2}\sin\left(\frac{\theta}{2}\right)\right] \\ &= \left[\sin\left(\frac{\hat{\theta}}{2}\right) \hat{n}, \cos\left(\frac{\hat{\theta}}{2}\right)\right]\end{aligned}$$

To better understand the concept of how a dual quaternion for formed, a simple example for a spacecraft's states in a circular orbit with orbital radius of $r_{B/I}^{I}$, about a generic body with gravitation constant μ is formed.

Initially the dual quaternion as seen in Eqn. (20) is taken. For this example it is assumed that the desired frame is equal to the inertial frame, so $q_{B/D} = q_{B/I}$.

$$\hat{q}_{B/D} = q_{B/D} + \varepsilon \frac{1}{2} r_{B/D}^{D} q_{B/D} \tag{20}$$

If the spacecraft is aligned with the inertial axis, $q_{B/D}$ will be the unit quaternion, and $r_{B/D}^{D}$ will be the position vector of the spacecraft in inertial space with a zero scalar component.

$$q_{B/D} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$r_{B/D}^{D} = \begin{bmatrix} r_x^I \\ r_y^I \\ r_z^I \\ 0 \end{bmatrix}$$

When this is all combined, the dual quaternion can be formed. Note that the multiplication is a quaternion operator as defined previously.

$$\hat{q}_{B/D} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} + \varepsilon \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} r_x^I \\ r_y^I \\ r_z^I \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} + \varepsilon \begin{bmatrix} r_x^I \\ r_y^I \\ r_z^I \\ 0 \end{bmatrix}$$

2.5. Dual Quaternion Equations of Motion

For the following analysis, three frame indices are defined. The body-fixed frame, B, reference/desired frame, D, and the inertial frame, I. Furthermore, the individual vector quaternions are described in the following format in terms of the previously defined reference frames. The vector quaternion $a_{y/z}^{x}$ is a vector a of the y frame relative to the z frame represented in the x frame.

Initially, the definition of a dual quaternion is taken as follows:

$$\hat{q}_{B/D} = q_{B/D} + \varepsilon \frac{1}{2} r_{B/D}^{D} q_{B/D} \tag{21}$$

where the rotation is described by $\hat{q}_{B/D}$ and the translation is described by $r_{B/D}^{B}$. Continuing, the definition of $\hat{q}_{D/I}$ can also be formed by a similar process. Therefore, the 6-DOF relative error between the body frame and the reference/desired frame can be defined by the following (Wu, Hu, Hu, Li, & Lian, 2005).

$$\hat{q}_{B/D} = \hat{q}_{D/I}^{*} *_{d} \hat{q}_{B/I} = \left[q_{B/D}, \frac{1}{2} q_{B/D} *_{q} r_{B/D}^{B} \right] \tag{22}$$

where $r_{B/D}^{B} = r_{B/I}^{B} - r_{D/I}^{B}$is the relative position vector quaternion between the body frame and the desired/reference frame, expressed in the body frame. Throughout the entirety of the calculations, it is assumed that the desired states are known and their derivatives are bounded. Taking the time derivative of the dual quaternion in Eqn. (20) results in the following derivation adapted from (Wu, Hu, Hu, Li, & Lian, 2005).

$$
\begin{aligned}
2\dot{\hat{q}}_{B/D} &= 2\dot{q}_{B/D} + \varepsilon\left(\dot{r}^D q_{B/D} + r^D \dot{q}_{B/D}\right) \\
&= \omega^D_{B/D} q + \varepsilon\left(\dot{r}^D q_{B/D} + \frac{1}{2} r^D \omega^D_{B/D} q_{B/D}\right) \\
&= \ \omega^D_{B/D} q + \varepsilon\left(\dot{r}^D q + \left(r \text{ x } \omega^D_{B/D}\right) q + \frac{1}{2} \omega^D_{B/D} r^D q_{B/D}\right) \\
&= \left(\omega^D_{B/D} + \varepsilon\left(\dot{r}^D + r \text{ x } \omega^D_{B/D}\right)\right)\left(q + \varepsilon \frac{1}{2} r^D_{B/D} q_{B/D}\right)
\end{aligned}
$$

$$
\begin{aligned}
2\dot{\hat{q}}_{B/D} &= \widehat{\omega}^D_{B/D} *_d \hat{q}_{B/D} \\
&= \hat{q}_{B/D} *_d \widehat{\omega}^B_{B/D} *_d \hat{q}_{B/D}{}^{*} *_d \hat{q}_{B/D}
\end{aligned}
$$

$$
\dot{\hat{q}}_{B/D} = \tfrac{1}{2} \hat{q}_{B/D} *_d \widehat{\omega}^B_{B/D} \qquad (23)
$$

Next, a dual relative velocity can defined.

$$
\widehat{\omega}^B_{B/D} = \widehat{\omega}^B_{B/I} - \widehat{\omega}^B_{D/I} \qquad (24)
$$

Additionally, the body dual velocity is defined as the following.

$$
\begin{aligned}
\widehat{\omega}^B_{B/D} &= \hat{q}^*_{B/D} *_d \widehat{\omega}^D_{B/D} *_d \hat{q}_{B/D} \\
&= \left(q_{B/D} + \varepsilon \frac{1}{2} r^D q_{B/D}\right)^{*} *_d \left(\widehat{\omega}^D_{B/D} + \varepsilon\left(\dot{r}^D + r^D \text{ x } \widehat{\omega}^D_{B/D}\right)\right) \\
&\quad *_d \left(q_{B/D} + \varepsilon \tfrac{1}{2} r^D q_{B/D}\right)
\end{aligned}
$$

$$
\widehat{\omega}^B_{B/D} = \widehat{\omega}^B_{B/D} + \varepsilon\left(q_{B/D}{}^{*}\left(\dot{r}^D + r^D \text{ x } \widehat{\omega}^D_{B/D}\right) q_{B/D} + \ \widehat{\omega}^B_{B/D} \text{ x } r^D\right) \qquad (25)
$$

Utilizing the quaternion equations of motion defined in Eqn. (4) and the quaternion property stating $qq^* = 1$, the following equation can be formed.

$$\dot{r}^B = \frac{d{q_{B/D}}^*}{dt} r^D q_{B/D} + {q_{B/D}}^* r^D q_{B/D} + {q_{B/D}}^* r^D \dot{q}_{B/D}$$

$$= \ -q^*_{B/D} \dot{q}_{B/D} q^*_{B/D} q_{B/D} + q^*_{B/D} \dot{r}^D q_{B/D} + {q_{B/D}}^* r^D \dot{q}_{B/D}$$

$$r^B = q^*_{B/D}\left(\dot{t}^D + r^D \text{ x } \omega^D_{B/D}\right) q \tag{26}$$

By substituting Eqn. (25) into Eqn. (26), it results in an equation for dual rates.

$$\widehat{\omega}^B_{B/D} = \widehat{\omega}^B_{B/D} + \varepsilon\left(\dot{r}^B + \widehat{\omega}^B_{B/D} \text{ x } r^B\right) \tag{27}$$

To derive the equation of motion for the dual acceleration of the system, Eqn. (25) can be rewritten in terms of the dual quaternion with dual body frame rates in reference to the inertial reference frame. This derivation is an adaption of one presented in (Wang & Sun, 2012).

$$\widehat{\omega}^B_{B/D} = \widehat{\omega}^B_{B/I} - \widehat{\omega}^B_{D/I}$$

$$\widehat{\omega}^B_{B/D} = \widehat{\omega}^B_{B/I} - \hat{q}^*_{B/D} *_d \widehat{\omega}^D_{D/I} *_d \hat{q}_{B/D} \tag{28}$$

Taking the time rate of change derivative of Eqn. (28) results in the dual acceleration of the system.

$$\dot{\widehat{\omega}}^B_{B/D} = \dot{\widehat{\omega}}^B_{B/I} - \hat{q}^*_{B/D} *_d \dot{\widehat{\omega}}^D_{D/I} *_d \hat{q}_{B/D} - \hat{q}^*_{B/D} *_d \widehat{\omega}^D_{D/I} *_d \dot{\hat{q}}_{B/D}$$

$$- \dot{\hat{q}}^*_{B/D} *_d \widehat{\omega}^D_{D/I} *_d \hat{q}_{B/D}$$

$$= \dot{\widehat{\omega}}^B_{B/I} - \hat{q}^*_{B/D} *_d \dot{\widehat{\omega}}^D_{D/I} *_d \hat{q}_{B/D} - \hat{q}^*_{B/D} *_d \widehat{\omega}^D_{D/I} *_d \frac{1}{2} \hat{q}_{B/D} *_d \widehat{\omega}^B_{B/D}$$

$$+ \frac{1}{2} \widehat{\omega}^B_{B/D} *_d \hat{q}^*_{B/D} *_d \widehat{\omega}^D_{D/I} *_d \hat{q}_{B/D}$$

$$\dot{\widehat{\omega}}^B_{B/D} = \dot{\widehat{\omega}}^B_{B/I} - \hat{q}^*_{B/D} *_d \dot{\widehat{\omega}}^D_{D/I} *_d \hat{q}_{B/D} + \widehat{\omega}^B_{B/D} \text{ x } \left(\hat{q}^*_{B/D} *_d \widehat{\omega}^D_{D/I} *_d \hat{q}_{B/D}\right) \tag{29}$$

From Newton’s Second Law, a dual form of $F = ma$ is formed using a dual inertia matrix, M^B and the dual acceleration in Eqn. (29).

$$\hat{f}^B = \widehat{M}^B *_d \dot{\widehat{\omega}}^B_{B/D} \tag{30}$$

$$\widehat{M}^B = \begin{bmatrix} m & 0 & 0 & 0 & 0_{1\text{x}3} & 0 \\ 0 & m & 0 & 0 & 0_{1\text{x}3} & 0 \\ 0 & 0 & m & 0 & 0_{1\text{x}3} & 0 \\ 0 & 0 & 0 & 1 & 0_{1\text{x}3} & 0 \\ 0_{3\text{x}1} & 0_{3\text{x}1} & 0_{3\text{x}1} & 0_{3\text{x}1} & I^B & 0_{3\text{x}1} \\ 0 & 0 & 0 & 0 & 0_{1\text{x}3} & 1 \end{bmatrix} \tag{31}$$

where m is the mass of the body, and I^B is the moment of inertia matrix for the body in the body frame. From this point, the following equation of motion is formed and organized to fit the format as seen in (Seo, 2015), which is the format used during the simulations contained herein.

$$\begin{aligned}\left(\dot{\widehat{\omega}}^B_{B/D}\right)^{\dagger} = \left(\widehat{M}^B\right)^{-1}\Big(\hat{f}^B - \widehat{\omega}^B_{B/I}\, x \,\Big(M^B\left(\widehat{\omega}^B_{B/I}\right)^{\dagger}\Big) - M^B\left(\hat{q}_{B/D} *_d \dot{\widehat{\omega}}^D_{D/I} *_d \hat{q}_{B/D}\right)^{\dagger} \\ - M^B\left(\widehat{\omega}^{\mathrm{B}}_{D/I}\,\mathrm{x}\,\widehat{\omega}^B_{D/I}\right)^{\dagger}\Big)\end{aligned} \tag{32}$$

where $\hat{f}^B = f^B + \varepsilon\tau^B$ which are the external forces and torques on the body frame.

2.6. External Forces for Dual Quaternion Dynamics

The external forces on the body frame, $\hat{f}^B$, are broken up into individual disturbances as follows.

$$\hat{f}^B = \hat{f}^B_g + \hat{f}^B_{gg} + \hat{f}^B_c \tag{33}$$

where the individual subscripts denote the separate external disturbances on the body frame. Subscript g denotes the gravitational force, gg denotes the gravity-gradient, and c is for the control input.

2.6.1. Gravitational Force

Due to the focus of this thesis being on satellite dynamics, gravity is the primary external perturbation being applied to the body of interest. Therefore, to derive the forces caused on a body due to gravity, Newton's Law of Gravitation is taken as follows (Curtis, 2010)

$$\vec{f_\iota} = -\frac{Gm_1m_2}{||\vec{r_\iota}||^3}\vec{r_\iota} \tag{34}$$

where $\vec{f_\iota}$ is the force due to gravity on body, ι, G is the universal gravitational constant, m_1 is the mass of the first body, m_2 is the mass of the second body, and $\vec{r_\iota}$ is the positional vector of the i body. So, writing this equation in terms of the acceleration due to gravity for two separate bodies results in the following

$$\ddot{\vec{r}}_1 = -\frac{Gm_2}{||\vec{r}_1||^3}\vec{r}_1 \tag{35}$$

$$\ddot{\vec{r}}_2 = -\frac{Gm_1}{||\vec{r}_2||^3}\vec{r}_2 \tag{36}$$

By evaluating the relative position of these two bodies, $r = r_2 - r_1$, and by extension their relative accelerations, $\ddot{\vec{r}} = \ddot{\vec{r}}_2 - \ddot{\vec{r}}_1$, the relative motion of two bodies is described. This is also typically known as the two body problem.

$$\ddot{\vec{r}} = -\frac{Gm_1}{||\vec{r}_2||^3}\vec{r}_2 - \frac{Gm_2}{||\vec{r}_1||^3}\vec{r}_1$$

$$= -\frac{G(m_1 + m_2)}{\|\vec{r}\|^3}\,\vec{r}$$

$$\ddot{\vec{r}} = -\frac{\mu}{\|\vec{r}\|^3}\vec{r} \tag{37}$$

where μ is the gravitational parameter for the central body. Since the central body is generally significantly more massive than the satellite, m_2 can be neglected. Therefore, μ is considered as a constant for each central body. Eqn. (37) is typically known as the orbit equation and is the base form for the dual gravity force vector and it describes the forces due to gravity on the body. From this point, the Eqn. (37) must be converted into dual form so it can be utilized within the dual quaternion equations. Since the force due to gravity is a pure translational force without any induced torques, in the dual form it contains a zero vector for the dual component of the dual quaternion resulting in the following equation adapted from (Seo, 2015).

$$\hat{f}_g^B = \left[m a_g^B, \vec{0}\right], a_g^B = -\mu \frac{r_{B/I}^B}{\left\|r_{B/I}^B\right\|^3} \tag{38}$$

where a_g^B is the acceleration due to gravity which is equivalent to $\ddot{\vec{r}}$.

2.6.2. Gravity Gradient Torque

Gravity gradient torques are caused by the difference of gravity on separate parts of an orbiting body. Therefore, if an orbiting body is broken up into individual differential masses, dm, the force due to gravity on the individual differential masses that are furthest away from the orbited body are actually less than the force due to gravity on the other differential masses that are closer to the orbited body. Although typically small, these differences in gravitational forces can cause a torque about the center of mass of the orbiting body. The following figure displays the geometry of the gravity gradient torque problem, and the derivation is adapted from (Curtis, 2010).

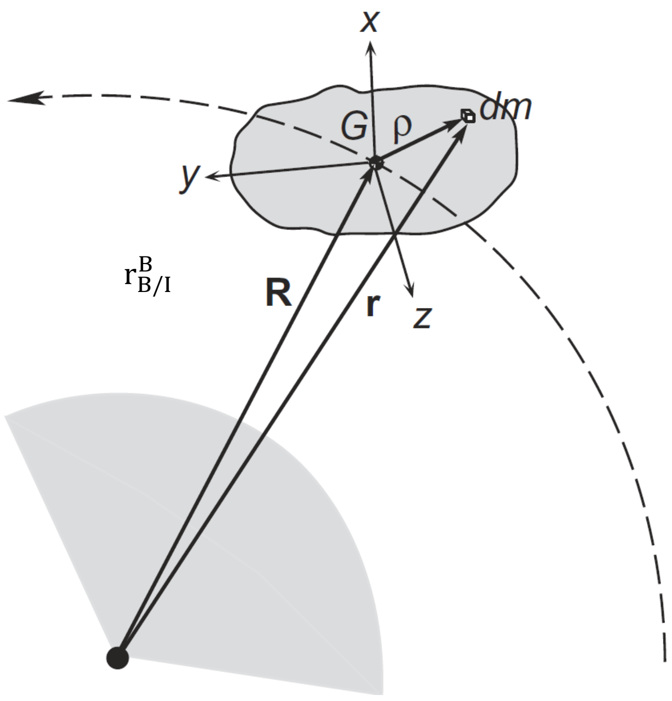

Figure 2.3 Gravity gradient torque geometry definitions (Curtis, 2010)

Typically, the torque about the center of mass of an object is calculated by completing the cross product of the position vector, ρ, with the force vector applied at dm. Therefore, in terms of the differential mass, dm, and the corresponding differential force, df, applied to that differential mass, the torque applied at the center of mass can be calculated. Since there is an infinite number of these differential masses and forces, a body integral is completed to sum up all of the torques caused by the differential masses resulting in Eqn. (38).

$$f_{gg}^{B} = \int_{B} \rho \text{ x } df_{g} dm \tag{39}$$

where the integral B is the body integral which represents a triple integral about each of the body axis.

The acceleration due to gravity derived earlier in Eqn. (37) can be substituted into Eqn. (39) in combination with dm, resulting in the following

$$f_{gg}^{B} = \int_{B} -\frac{\mu}{\|\vec{r}\|^{3}} \rho \text{ x } (\vec{r}_{B/I}^{B} + \rho) dm \tag{40}$$

To further simplify the equation, $\|\vec{r}\|^{3}$ is expanded and simplified using the assumption that $\left\|\vec{r}_{B/I}^{B}\right\| \gg \|\rho\|$.

$$\begin{aligned} \|\vec{r}\|^{3} &= \frac{1}{(\vec{r} \cdot \vec{r})^{\frac{3}{2}}} \\ &= \frac{1}{\left(\left(\vec{r}_{B/I}^{B} + \rho\right) \cdot \left(\vec{r}_{B/I}^{B} + \rho\right)\right)^{\frac{3}{2}}} \\ &\cong \frac{1}{\vec{r}_{B/I}^{B} \cdot \vec{r}_{B/I}^{B} + 2\vec{r}_{B/I}^{B} \cdot \rho} \\ &\cong \frac{1}{\left(\vec{r}_{B/I}^{B}\vec{r}_{B/I}^{B}\right)^{\frac{3}{2}} \left(1 + \frac{2\vec{r}_{B/I}^{B}\rho}{\vec{r}_{B/I}^{B} \cdot \vec{r}_{B/I}^{B}}\right)^{\frac{3}{2}}} \\ &\cong \frac{1}{\left(\vec{r}_{B/I}^{B}\vec{r}_{B/I}^{B}\right)^{\frac{3}{2}}} \left(1 - \frac{3\vec{r}_{B/I}^{B} \cdot \rho}{\vec{r}_{B/I}^{B} \cdot \vec{r}_{B/I}^{B}}\right) \end{aligned} \tag{41}$$

This is then substituted into Eqn. (40)

$$f_{gg}^{B} = \frac{-\mu}{\left\|\vec{r}_{B/I}^{B}\right\|^{3}} \int_{B} \rho \text{ x } \left(1 - \frac{3\vec{r}_{B/I}^{B} \cdot \rho}{\vec{r}_{B/I}^{B} \cdot \vec{r}_{B/I}^{B}}\right)\left(\vec{r}_{B/I}^{B} + \rho\right) dm$$

$$= \frac{-3\mu}{\left\|\vec{r}_{B/I}^{B}\right\|^{3}} \int_{B} \rho \text{ x } \frac{\vec{r}_{B/I}^{B} \cdot \rho}{\vec{r}_{B/I}^{B} \cdot \vec{r}_{B/I}^{B}}\left(\vec{r}_{B/I}^{B} + \rho\right) dm$$

$$= \frac{-3\mu}{\left\|\vec{r}_{B/I}^{B}\right\|^{3}} \int_{B} \rho \text{ x } \frac{\vec{r}_{B/I}^{B} \cdot \rho}{\vec{r}_{B/I}^{B} \cdot \vec{r}_{B/I}^{B}} \vec{r}_{B/I}^{B} dm$$

$$= 3\mu \frac{r_{B/I}^{B} \text{ x } \left(I^{B} r_{B/I}^{B}\right)}{\left\|r_{B/I}^{B}\right\|^{5}}$$

where I^{B} is the same moment of inertia matrix of the body in the body frame from previously. Due to the gravity gradient being a pure torque disturbance, the real part of the dual quaternion will be a $\vec{0}$. The dual formulation of gravity gradient is as follows

$$\hat{f}_{gg}^{B} = \left[\vec{0}, f_{gg}^{B}\right], f_{gg}^{B} = 3\mu \frac{r_{B/I}^{B} \text{ x } \left(I^{B} r_{B/I}^{B}\right)}{\left\|r_{B/I}^{B}\right\|^{5}} \tag{42}$$

2.6.3. Control Force

The control force $\hat{f}_{c}^{B}$ is a dual quaternion based controller, which requires a coupled force and torque input. In a typical utilization, where it drives the errors to zero, it will cause the following limit to occur

$$\lim_{t \to \infty} \{ \hat{q}_{B/D}, \hat{\omega}_{B/D}^{B}\} = \{\hat{1}, \hat{0}\} \tag{43}$$

This is a single coupled controller in comparison to other typical simulations which have a decoupled position and attitude controllers. Therefore, its integration and optimization will be different. However, this is one of the primary benefits of utilizing dual quaternions because it simplifies the control design for a system due to its coupled nature. Typical coupled systems that are not dual quaternions require a constraint equation to tie the positional and attitude controller. However, with the dual quaternions the constraints are inherent within the control design.

Additionally, it leads directly into the thruster distribution matrix optimization which also requires a coupled force and torque input as seen in the following sections. Investigation into controllers within the dual quaternion space is not a focus of this thesis. However, it is an area of future research.

3. Control Allocation: Pseudo Inverse Optimization Method

The primary objective behind the newly developed thruster allocation method was to develop an optimized thruster distribution matrix that will account for both the desired body forces and torques simultaneously. This was accomplished by developing a constant, optimized thruster distribution matrix which is an improvement on the typically utilized thruster selection logic and more computationally efficient when compared to other optimized allocation methods such as the Simplex method (Ankersen, Aleshin, Vankov, & Volochinov, 2005), (Crawford, 1969), (Doman, Gambel, & Ngo, 2007), and (Wang M., 2009).

3.1. Control Allocation Overview

For simplistic simulations, the concept of actuator allocation is typically disregarded. For these systems, a general controls diagram of the system and controller can be created as in Figure 3.1.

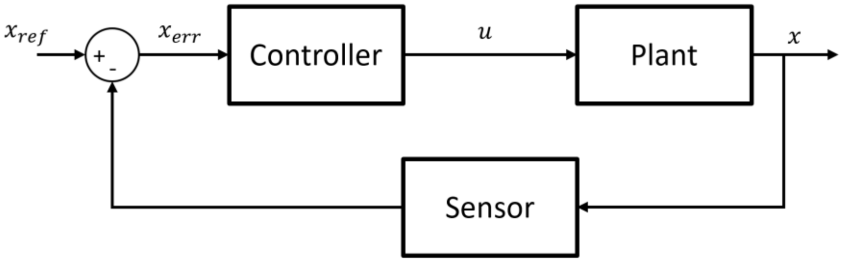

Figure 3.1 Traditional controls block diagram

where the Controller block contains the designed controller which produces and desired control output, u, which is then led into the Plant block, which contains all of the system kinematics and dynamics equations. These equations are used to calculate the system state variable x, which is then read by the Sensor block which contains any of the onboard sensors such as gyroscopes, start trackers, etc. The output of the sensor is then compared to the desired reference state, x_{ref}, to form the state error, x_{err}.

However, for more realistic simulations, actuators are utilized to create the forces and torques about the body frame. Therefore, an allocation process is required to take the desired command input u from the controller and distribute it to the appropriate actuators. Typically, this is completed after the controller calculates the command input, but prior to implementation into the plant dynamics as seen in Figure 3.2.

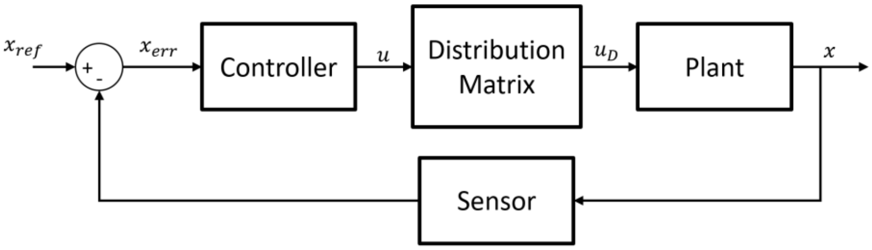

Figure 3.2 Controls block diagram with distribution matrix included

Once the controller allocation is completed within the distribution matrix block, it results in an actuator command u_D. Typically, this is a command vector for all the contained actuators. This is then inputted into the plant dynamics, where models for the actuators are contained. Therefore, the actuator commands will cause the actual physical body forces and torques, which depending on the accuracy and type of actuators, will differ slightly from the ideal forces and torques calculated by the controller.

3.2. Traditional Thruster Selection Method

The most commonly utilized thruster allocation method in spacecraft today is a thruster selection matrix. Also known as the decoupled method, it is based on individual lookup tables for each of the degrees of freedom of a system. This method is used due to its simplicity and ease of understanding. For a spacecraft using thrusters for attitude only based control, 3-DOF, a total of six tables are used.

As an example, the generic spacecraft thruster layout is taken as in the following figure with a total of 8 thrusters.

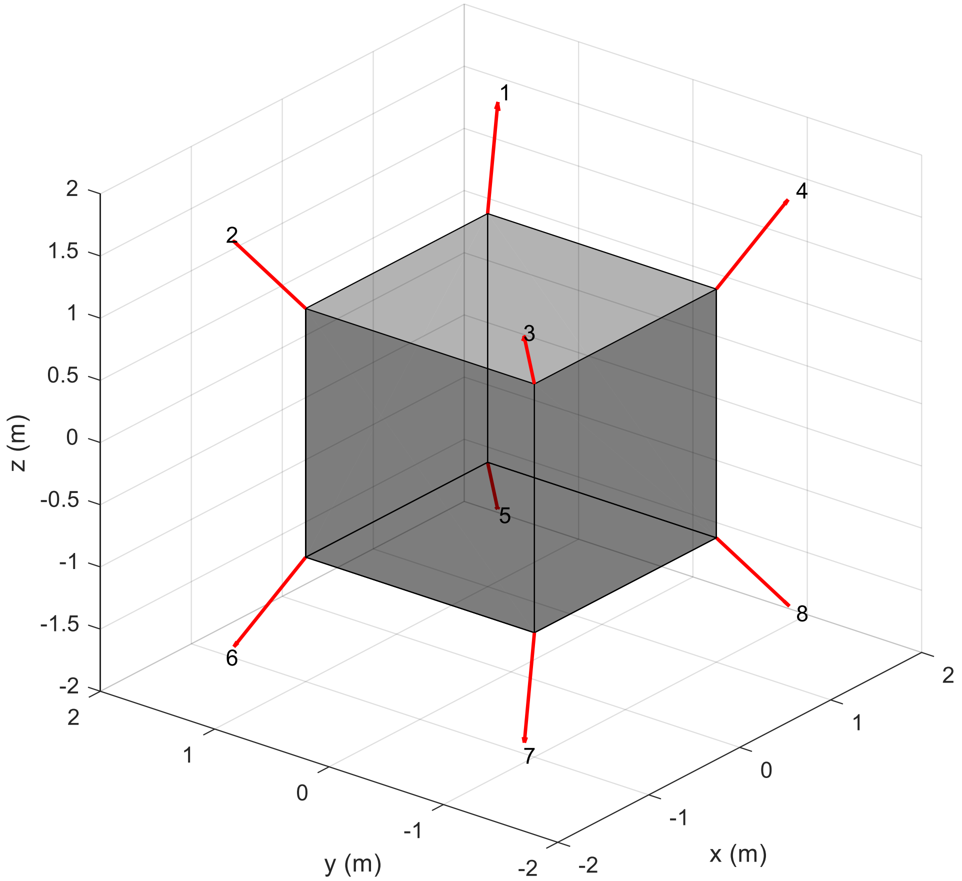

Figure 3.3 Generic spacecraft example thruster layout

By taking the thruster layout, the selection tables can be created for each of the 3-DOF. For example, a positive rotation about the x-axis can be formed as follows. For this example, it is assumed that the corresponding thrusters on opposite sides of the center of mass are capable of producing and equal and opposite torque about the center of mass. Therefore, the thruster layout as a whole is perfectly symmetric about the center of mass of the body.

$$\begin{aligned} x_{rot} &= a\tau_{+x} \\ &= a\begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \end{aligned} \tag{44}$$

where a is a scaling value chosen by the controller, and τ_{+x} is which thrusters must fire to complete the a slew maneuver about the positive x axis. By extension, a vector and scaling value can be created for each of the 3-DOF resulting in the following

$$\begin{gathered} \tau = a\tau_{+x} + b\tau_{-x} + c\tau_{+y} + d\tau_{-y} + e\tau_{+z} + f\tau_{-z} \\ \tau = a\begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} + b\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} + c\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} + d\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} + e\begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} + f\begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \end{gathered} \tag{45}$$

where $b, c, d, e,$ and f are all individual scaling factors for each of the 3-DOF. τ is the compiled thruster command vector. This concept can easily be extended out to a full 6-DOF if desired, using the same method.

As an example, if the constants are chosen to vary from 0-1, and it is desired to have a slew about the positive x-axis and a slew of half the rate about the positive y-axis, Eqn. (45) would be formed as follows

$$\tau = 1\begin{bmatrix}0\\0\\1\\1\\1\\1\\0\\0\end{bmatrix} + 0\begin{bmatrix}1\\1\\0\\0\\0\\0\\1\\1\end{bmatrix} + 0.5\begin{bmatrix}0\\1\\1\\0\\1\\0\\0\\1\end{bmatrix} + 0\begin{bmatrix}1\\0\\0\\1\\0\\1\\1\\0\end{bmatrix} + 0\begin{bmatrix}0\\1\\0\\1\\0\\1\\0\\1\end{bmatrix} + 0\begin{bmatrix}1\\0\\1\\0\\1\\0\\1\\0\end{bmatrix}$$

$$= \begin{bmatrix}0\\0.5\\1.5\\1\\1.5\\1\\0\\0.5\end{bmatrix}$$

In addition, as previously stated, this current example assumes perfect symmetry of the thrusters about the center of mass. If any asymmetries were in place, additional calculations would be needed to form thruster selection tables that retain pure translation and rotation maneuvers, which would then be scaled in the same manner as in Eqn. (45).

3.3. Pseudo Inverse Optimization Method

The concept of the pseudo inverse optimized distribution matrix is to create constant matrix that completes the control allocation similar to the selection logic. However, it will also optimize the resulting thruster force vector and it will require only a single matrix product calculation instead of multiple scalar multiplications and vector additions for each of the selection logic tables. Additionally, the result accounts for any asymmetry of the thrusters about the center of mass.

The following distribution matrix method is an adapted process to utilize a coupled force and torque input and optimally distribute it to all the necessary thrusters (Smith & Seo, 2017). The optimization process is performed to calculate the minimum thrust required for an individual command input, through the use of a min-max solver. The desired final product of the pseudo inverse optimization method is a purely positive constant thruster distribution matrix, $D \in \mathbb{R}^{nx12}$, which when multiplied by a vector of desired forces and torques, F_{pos}, produces a N dimensional vector of thruster fire ratios, τ.

$$\tau = D\, F_{pos} \tag{46}$$

To calculate this optimized thruster distribution matrix, the following process is completed. For any 6DOF system, a desired command force and torque 6x1 vector $F_{des} \in \mathbb{R}^6$ is assembled by separating the vector F_{des} into the body axis forces F and torques N.

$$F_{des} = \begin{bmatrix} F \\ N/a \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ X \\ Y \\ Z \end{bmatrix}_{6x1} \tag{47}$$

where a is a scaling factor with units of length resulting in the vector F_{des} having units of force. The scaling factor a of the torque components in F_{des} is chosen for mathematical convenience. The desired force vector F_{des} is translated to a non-negative 12x1 vector $F_{pos} \in \mathbb{R}^{12} \mid F_{pos} \geq 0$ with the positive and negative components of each section separated. This variation comes from F_{pos} being composed of a 12x1 vector which is strictly positive by taking the positive and negative values of the desired forces and torques input F_{des} and rearranging the negative values to the bottom 6 vector locations and taking their absolute value. While they are no longer negative, their location within the bottom half of F_{pos} denotes that they are negative values.

$$F_{pos} = \begin{bmatrix} F_{des}(positive) \\ |F_{des}(negative)| \end{bmatrix} = \begin{bmatrix} x_{pos} \\ y_{pos} \\ z_{pos} \\ X_{pos} \\ Y_{pos} \\ Z_{pos} \\ |x_{neg}| \\ |y_{neg}| \\ |z_{neg}| \\ |X_{neg}| \\ |Y_{neg}| \\ |Z_{neg}| \end{bmatrix}_{12x1} \tag{48}$$

The necessity of a purely positive desired force and torque vector is inherent from the thrusters, which can only be fired in a single direction along single axis thrusters. This vector is multiplied with the distribution matrix as seen in Eqn. (43). Since the distribution matrix is also a purely positive matrix, the resulting thruster force vector will be the force required for each thruster along its positive axis direction.

Next, a set of $n \geq 6$ thrusters, to ensure 6 DOF, is compiled into a n dimensional vector τ.

$$\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \vdots \\ \tau_n \end{bmatrix}_{nx1} \tag{49}$$

where τ_i for $i = 1, 2, \ldots n$ is the thrust provided by each individual thruster along its own axis. Due to thruster's single directional force, τ must be a non-negative vector. Additionally, a matrix $M \in \mathbb{R}^{6xn}$ is formed by the thruster locations and relative attitudes based on the body frame located at the center of mass of the spacecraft. It is a basic kinematic transformation matrix. These locations and relative attitudes, compiled into the matrix $T_{dir} \in \mathbb{R}^{3xn}$, are used to calculate the torque capabilities of each thruster about the body axis, $T_N \in \mathbb{R}^{3xn}$, by computing the cross product of each thruster attitude with its corresponding location relative to the spacecraft center of mass.

$$M = \begin{bmatrix} T_{dir} \\ T_N \end{bmatrix} \tag{50}$$

This matrix can then be used to describe the relation between the desired forces and torques vector, F_{des}, and the thruster force vector τ.

$$F_{des} = M\tau \tag{51}$$

From this point, to get it into a distribution matrix form, the pseudo inverse of M is calculated by completing a singular value decomposition as follows:

$$M^{+} = V_0 S_{6x6}^{-1} U^T \tag{52}$$

$$M = USV^T = U\left[S_{6x6}\ 0_{6x(n-6)}\right][V_0\ V_{null}]^T \tag{53}$$

where S is a 6x6 positive-definite diagonal matrix, V_0 is a nx6 matrix which satisfies the following condition in Eq. (54), and V_{null} is a nx(n-6) matrix whose columns span the null space of M (Smith & Seo, 2017).

$$V_0^T V_0 = I_{6x6} \tag{54}$$

From this point, the cost function that is utilized for the min-max optimization is formed as follows in Eq. (55). The cost function's result is a vector solution, which corresponds to one of the components that make up F_{pos}. Therefore, since $F_{pos} \in \mathbb{R}^{12}$, $i = 1, 2, \dots 12$, there will be a total of 12 optimized vector solutions, as explained later.

$$F_i(x) = M^{+} b_i + V_{null} x \tag{55}$$

where b_i is a 6x1 unit vector with all 0 components apart from a 1 placed in the i location of b when $i = 1, 2, \ldots 6$, or a -1 placed in the $i - 6$ location of b when $i = 7, 8, \ldots 12$. These correspond to the positive and negative parts of F_{pos} shown previously.

To complete the optimization process using cost function in Eqn. (55), MATLAB's min-max optimization function, *fminimax*, is called, and is based on the following equation (Optimization Toolbox 4 User's Guide, 2008).

$$D_i = \min \max F_i(x) \tag{56}$$

where $F_i(x)$ is the selected cost function being optimized from Eqn. (55), and $D_i \in \mathbb{R}^{\mathrm{nx1}}$ is the resulting optimized solution vector. The min-max optimization minimizes the maximum element of each D_i vector. Therefore, it results in a minimum thrust solution for each vector D_i (Smith & Seo, 2017). The solver starts with an initial condition of an nx1 zero vector, x_0, and solves for the min-max solution, x, based on the cost function in Eqn. (55). This optimization is run individually for each component of F_{pos}, so $i = 1, 2, \ldots 12$, so a set of 12 D_i vectors is created, which are then compiled into the complete, optimized, distribution matrix D.

$$D = [D_1 \; D_2 \ldots D_{12}] \tag{57}$$

Additionally, the following constraint is applied to the optimization.

$$Ax \leq c \tag{58}$$

where A is chosen to be $-V_{null}$, $c = M^{+}b_i$, and b_i is the same variable from the cost function in Eqn. (55). When this constraint is applied to the optimization, it will restrict the resulting D_i vectors to be purely positive. Therefore, when the compiled thruster distribution matrix D is used in conjunction with F_{pos}, as seen in Eqn. (46), the thruster solution vector, τ, will be purely positive. This is a requirement due to thrusters only being able to fire in a single direction. Therefore, the positive thrust vector, τ, will already be aligned with the individual thruster axis' and thrust directions.

4. OSIRIS-REx 8 Thruster Example

The OSIRIS-REx scenario, simulated as a testbed, for the dual quaternion and thruster distribution matrix was one of the sections for the landing maneuver where OSIRIS-REx is descending down towards the asteroid Bennu to gather a sample of the regolith. It simulates a total descent of 30 meters, over 500 seconds during which a basic slew maneuver is preformed to align the spacecraft axes with Bennu's inertial axes.

This was all simulated utilizing dual quaternion dynamics within NASA Goddard's open source flight software known as "42". For completeness, the dual quaternion equations of motion are included again as follows

$$\hat{q}_{B/D} = q_{B/D} + \varepsilon \frac{1}{2} r^{D}_{B/D} q_{B/D}$$

$$\left(\dot{\widehat{\omega}}^{B}_{B/D}\right)^{\dagger} = \left(\widehat{M}^{B}\right)^{-1} \left(\hat{f}^{B} - \widehat{\omega}^{B}_{B/I}\, x \left(M^{B}\left(\widehat{\omega}^{B}_{B/I}\right)^{\dagger}\right) - M^{B}\left(\hat{q}_{B/D} *_{d} \dot{\widehat{\omega}}^{D}_{D/I} *_{d} \hat{q}_{B/D}\right)^{\dagger} - M^{B}\left(\widehat{\omega}^{\mathrm{B}}_{D/I}\, \mathrm{x}\, \widehat{\omega}^{B}_{D/I}\right)^{\dagger}\right)$$

4.1. OSIRIS-REx Mission Overview

The Origins, Spectral Interpretation, Resource Identification, Security, Regolith Explorer (OSIRIS-REx) is also known as the Asteroid Sample Return Mission. The mission objective for OSIRIS_REx is to collect a sample of regolith from the surface of the asteroid Bennu and return it in a capsule back to Earth for analysis. The driving motivation behind the mission objective is the desire to find clues and answers to many of the questions regarding the origins of the solar system, since many asteroids are remnants of the debris from the solar system formation process.

The asteroid 101955 Bennu was chosen due to its carbonaceous composition, which may contain some information regarding the early history of the solar system. Therefore, it is also possible that the asteroid contains the molecular precursors to the origins of life on Earth. On a separate note, due to its relative orbit to Earth, it is also potentially a very hazardous asteroid which might impact the Earth in the late 22nd century. Therefore, OSIRIS-REx will provide necessary details to mitigate the possibility of a collision in the future. The primary science objectives are the following, taken from the OSIRIS-REx mission overview (Mission Objectives, 2017).

Table 4.1 OSIRIS-REx's key science objectives (Mission Objectives, 2017)

Objective	Description
1	Return and analyze a sample of Bennu’s surface
2	Map the asteroid
3	Document the sample site
4	Measure the orbit deviation caused by non-gravitational forces (the Yarkovsky effect)
5	Compare observations at the asteroid to ground-based observations

The OSIRIS-REx mission contains multitudes of different trajectory sections for the duration of its life. However, for the simulations contained herein, a section of the Touch and Go (TAG) maneuver is analyzed. During the TAG maneuver, the primary objective is to touch down safely on the asteroid for a long enough period of time to gather a sample of regolith from the surface. Once the sample is gathered, the mass of the sample will be tested to determine if it meets the mission requirements. If the sample is deemed to be not be sufficient, then additional TAG maneuvers may be attempted. The section of the

TAG maneuver that is analyzed is a small 30-meter section of the descent from approximately 475 meters to 445 meters.

4.2. External Forces: Gravity and Gravity Gradient

For spacecraft based simulations, the force due to gravity is generally the primary external force on the body. However, since the exact shape and properties of Bennu are not known, its irregular gravity field would be very difficult to model correctly. Therefore, for simplicity of analysis, it is modeled as a point source, so the irregularities of mass distribution were not taken into account for the contained simulations. One additional benefit with the simplified gravity model is it significantly simplifies and shortens the simulation analysis time. The gravity force on the spacecraft is implemented into the dual quaternion simulation utilizing the Eqn. (38), repeated here for completeness.

$$\hat{f}_g^B = \left[m a_g^B, \vec{0}\right], a_g^B = -\mu \frac{r_{B/I}^B}{\left\|r_{B/I}^B\right\|^3}$$

The other external perturbation that was taken into account during the simulation was the gravity gradient as seen in Eqn. (42). Due to the gravity of the asteroid being modeled as a point source, and its total mass being relatively small, the torque produced from the gravity gradient perturbation is also relatively small. However, it was still included for simulation accuracy. The equation utilized is repeated below.

$$\hat{f}_{gg}^B = \left[\vec{0}, f_{gg}^B\right], f_{gg}^B = 3\mu \frac{r_{B/I}^B \text{ x } \left(I^B r_{B/I}^B\right)}{\left\|r_{B/I}^B\right\|^5}$$

4.3. Controller

Since the focus of this thesis is the dual quaternion dynamics and optimal distribution matrix, a non-dual quaternion based, heritage optimal ramp coast glide controller was utilized. This controller was taken from the OSIRIS-REx simulation utilized in (Smith & Seo, 2017). Therefore, the focus will not be on the controller but instead on the dual quaternion dynamics and the effectiveness of the optimal distribution matrix. However, a basic description of the controller is given.

At its basic form, the ramp coast glide controller contains two separate components, and it will switch between these two components depending on the spacecraft's state relative to the desired state. When this error state passes a below predetermined error maximum the control type is switched to the second component.

The simpler of the two components is when the error is below the error maximum, where it represents a critically damped Proportional Derivative (PD) controller. When the error state is above the error maximum, it is controlled by the first component, which is an optimal bang bang controller with a maximum imposed rate. The concept behind an optimal bang bang controller is to minimize the amount of time it takes to perform a maneuver. Generally, this is accomplished by setting the actuators to their maximum value towards the desired state resulting in an acceleration, and then half way through the maneuver set the actuators to the negative max value to cause the system to brake and stop at the desired state. However, for the current implementation, there is a coasting period at a predetermined maximum rate between the accelerating and braking sections.

The error maximum cutoff point, x_c, is based on the maximum desired acceleration, a_{max} , and the natural frequency of the controlled system, ω_0.

$$x_c = \frac{a_{max}}{\omega_0^2} \tag{58}$$

Therefore, when the state x is less than x_c the control is based off a PD controller as follows

$$a = -K_p x - K_d v \tag{59}$$

where K_p is the proportional gain, K_d is the derivative gain, and v is the velocity of the system. These gain are chosen to create a critically damped response.

$$\begin{aligned} K_d &= 2m\omega_0 \\ K_p &= m\omega_0^2 \end{aligned} \tag{60}$$

Since the gains are calculated to great a critically damped response, the natural motion of the system for this control segment is

$$\begin{aligned} x(t) &= (x_0 + (v_0 + \omega_0 x_0)t)e_0^{-\omega} t \\ v(t) &= (v_0 - \ \omega_0(v_0 + \omega_0 x_0)t)e^{-\omega_0 t} \end{aligned} \tag{61}$$

where t is the time, and v_0 is the initial velocity. Therefore, when $t \rightarrow \infty$

$$\begin{aligned} x &\approx (x_0 + \omega_0 x_0)te^{-\omega_0 t} \\ v &\approx -\omega_0(v_0 + \omega_0 x_0)te^{-\omega_0 t} \approx -\omega_0 x \end{aligned} \tag{62}$$

Next, when the state is larger than the cutoff point, it takes the form of the following bang bang controller

$$a = a_{max} sgn(v_{cmd} - v) \tag{63}$$

where v_{cmd} is the bang bang switching line defined as

$$v_{cmd} = \ - sgn(x)\sqrt{2a_{max}(|x| - x_0)} \tag{64}$$

The $sgn(x)$ function is known as a sign function and is represented as the following

$$sgn(x) = \begin{cases} -1 \text{ if } x < 0 \\ \ \ 1 \text{ if } x > 0 \end{cases} \tag{65}$$

Additionally, the braking point, x_b, is calculated based on the system characteristics and the maximum chosen rate v_{max}.

$$x_b = x_0 + \frac{1}{2a_{max}} v_{max}^2 \tag{66}$$

where $x_0 = \frac{1}{2}x_c$. All of these characteristics are combined into the following figure for clarity.

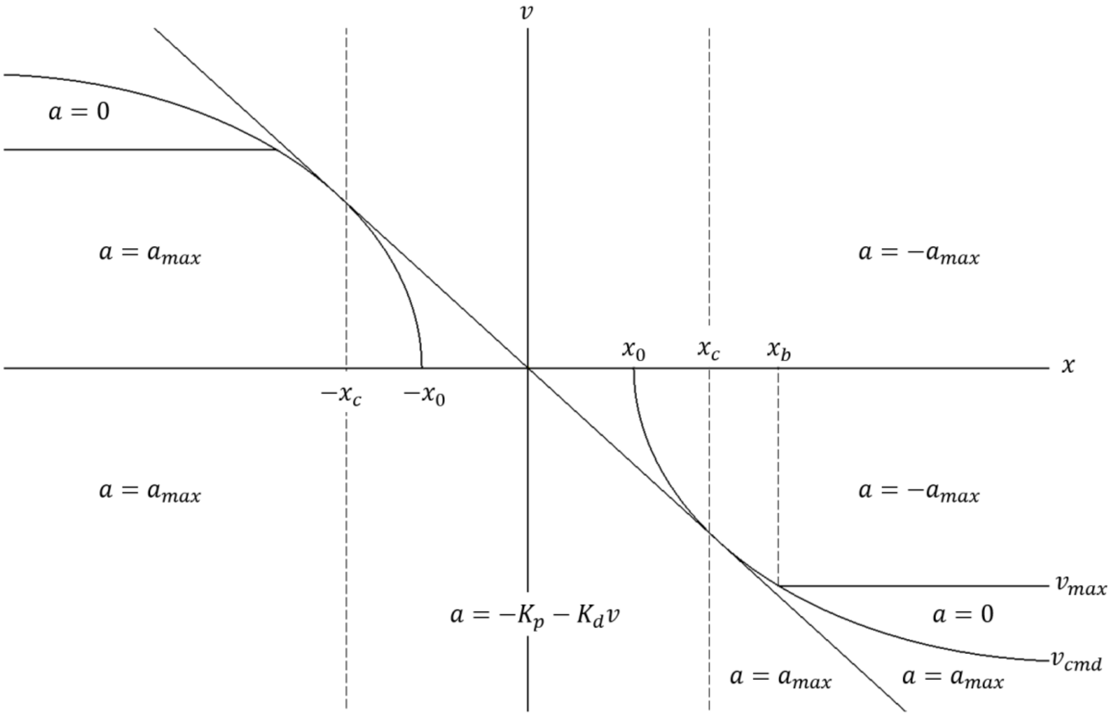

Figure 4.1 Visual representation of the ramp coast glide controller

There is a current known problem with this control design if the state passes the cutoff point x_c prior to reaching the switching line v_{cmd}. This would cause the controller to not switch to the PD control smoothly since there would be a velocity error between the expected velocity of v_{cmd} and the true velocity at x_c. However, this error does not occur in the contained simulations. Therefore, it is noted but disregarded for the remainder of the analysis.

To further the accuracy of the simulation, a fixed control cycle time step was included of ten milliseconds. This results in a non-continuous time controller, similar to what is implemented on in-flight spacecraft such as OSIRIS-REx. Due to the set control cycle, the thruster fire times are quantized to match this control cycle. Therefore, they can only perform one set of commands during each control cycle and cannot not switch prior to the next control cycle command.

4.4. Thruster Specifications

Initially, the actuators that are used in the simulation are defined. The only actuator utilized for the duration of the simulation are thrusters. On OSIRIS-REx, there are two separate sets of thrusters denoted as the A set and B set. Since they are decoupled and only a single set is used for nominal mission operations, there will be a separate optimized distribution matrix for each. Therefore, for the following simulation, only the A set of thrusters is utilized. The A set of thruster locations and attitudes were set up and utilized to form the following M matrix of thruster attitudes and torque capabilities about the center of mass as seen from Eqn. (50). The center of mass of the spacecraft in reference to the arbitrary reference point is as follows.

$$CM = [-0.006\ 0.003\ 0.770]$$

$$M = \begin{bmatrix} 0.4330 & -0.4330 & -0.4330 & 0.4330 & 0.4330 & -0.4330 & -0.4330 & 0.4330 \\ 0.25 & 0.25 & -0.25 & -0.25 & 0.25 & 0.25 & -0.25 & -0.25 \\ 0.8660 & 0.8660 & 0.8660 & 0.8660 & -0.8660 & -0.8660 & -0.8660 & -0.8660 \\ 0.7606 & 0.7606 & -0.7606 & -0.7606 & -0.9122 & -0.9122 & 0.9174 & 0.9174 \\ -0.6543 & 0.6439 & 0.6439 & -0.6543 & 0.9168 & -0.9065 & -0.9065 & 0.9168 \\ -0.1914 & 0.1944 & -0.1970 & 0.1940 & -0.1914 & 0.1944 & -0.1970 & 0.1940 \end{bmatrix}$$

Graphically, these values depict the thruster layout below.

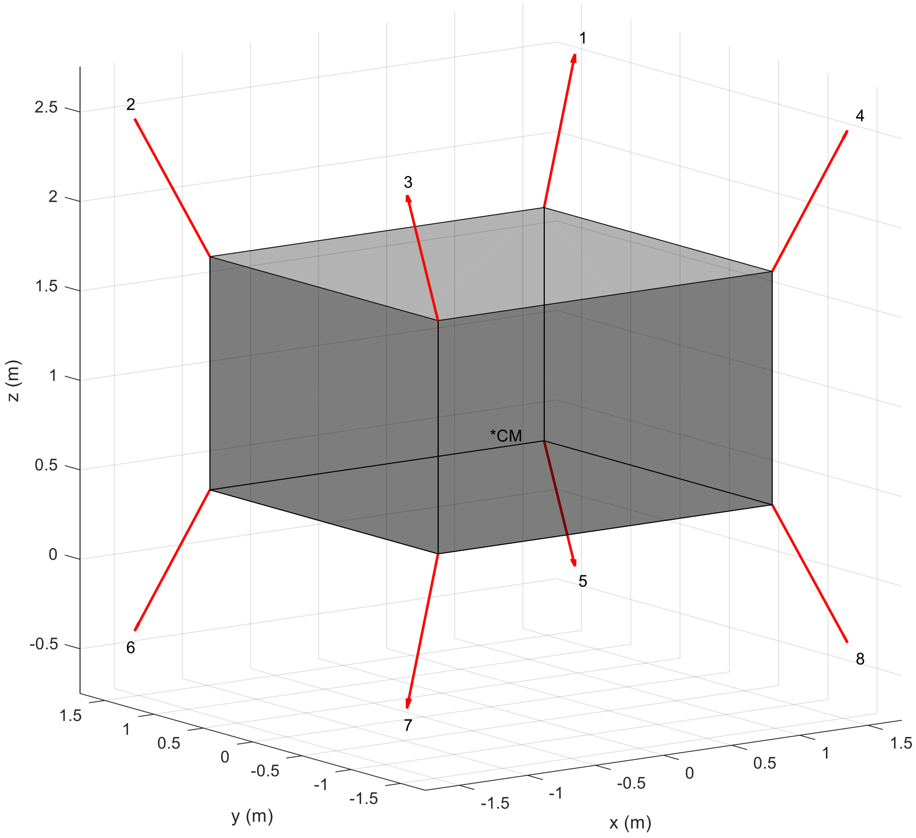

Figure 4.2 OSIRIS-REx thruster layout

Additionally, each of the thrusters described herein are identical small attitude based thrusters capable of producing 4.45 Newtons of thrust. Therefore, it is possible to have thruster saturation if the command forces and torques require a higher thrust than is possible for an individual thruster. When this occurs, the entire thruster force vector is scaled down proportionally. Therefore, the attitude maneuver is retained but the magnitude is lowered so the previously saturated thruster is firing at the maximum value. However, it is noted that a stability analysis of the saturation has not been completed for all cases. The contained simulations where the thrusters become saturated are stable and therefore it is not a current area of focus. However, it has been suggested as an investigation point for the future.

4.5. Thruster Distribution Matrix Calculation

Following the steps laid out in section 3, the singular value decomposition of the M matrix is completed along with evaluating MATLAB's *fminimax* function optimization, resulting in the following optimized distribution matrix.

$$\mathrm{D} = \begin{bmatrix} 0.0000 & 0.0000 & 0.0010 & 0.0000 & 0.3204 & 1.2870 & 0.6745 & 1.0903 & 0.2896 & 0.2980 & 0.0000 & 0.0000 \\ 0.6745 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0903 & 0.2906 & 0.2980 & 0.3204 & 1.2870 \\ 0.6745 & 1.0903 & 0.0000 & 0.2980 & 0.0000 & 1.2870 & 0.0000 & 0.0000 & 0.2906 & 0.0000 & 0.3204 & 0.0000 \\ 0.0000 & 1.0903 & 0.0028 & 0.2980 & 0.3204 & 0.0000 & 0.6745 & 0.0000 & 0.2878 & 0.0000 & 0.0000 & 1.2870 \\ 0.0933 & 0.0826 & 0.2896 & 0.2980 & 0.0000 & 1.2870 & 0.5812 & 1.0077 & 0.0010 & 0.0000 & 0.3204 & 0.0000 \\ 0.5812 & 0.0981 & 0.2906 & 0.2980 & 0.3204 & 0.0000 & 0.0933 & 0.9923 & 0.0000 & 0.0000 & 0.0000 & 1.2870 \\ 0.5735 & 0.9923 & 0.2906 & 0.0000 & 0.3204 & 1.2870 & 0.1010 & 0.0981 & 0.0000 & 0.2980 & 0.0000 & 0.0000 \\ 0.1010 & 1.0077 & 0.2878 & 0.0000 & 0.0000 & 0.0000 & 0.5735 & 0.0826 & 0.0028 & 0.2980 & 0.3204 & 1.2870 \end{bmatrix}$$

This distribution matrix provides the capabilities to transform any 6-DOF body command input to an optimized thruster command as shown in Eqn. (46). Additionally, the distribution matrix fulfills the strictly positive constraint imposed on the optimization to produce a strictly positive thruster force vector. Once this optimal distribution matrix is obtained, it is integrated into the dual quaternion based OSIRIS-REx simulation. In this simulation, the heritage ramp coast glide controller, discussed previously, provides the ideal forces and torques command vector, which are then used in combination to the distribution matrix in order to calculate the individual thruster firing times. The actual forces and torques produced by the thrusters is then compared to the ideal forces and torques to determine the effectiveness of the optimal distribution matrix.

As a quick verification and example, if a F_{des} vector is taken as follows, representing a multiple body axis translation and rotation, it will be transformed into the

F_{pos} vector to be used in combination with the thruster distribution matrix.

$$F_{des} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ -1 \\ -1 \\ 0 \end{bmatrix} \quad \Rightarrow \quad F_{pos} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

This vector is then pre-multiplied by the thruster distribution matrix as seen in Eqn. (46) resulting in the following.

$$\tau = D\ F_{pos} = \begin{bmatrix} 0.2980 \\ 1.2980 \\ 2.0852 \\ 1.0903 \\ 0.4963 \\ 0.6793 \\ 1.8637 \\ 1.7271 \end{bmatrix}$$

This is the resting thruster force vector τ. In order to confirm its validity the kinematic transformation equation, Eqn. (51), can be utilized.

$$F_{des} = M\tau$$

$$= M \begin{bmatrix} 0.2980 \\ 1.2980 \\ 2.0852 \\ 1.0903 \\ 0.4963 \\ 0.6793 \\ 1.8637 \\ 1.7271 \end{bmatrix}$$

$$= \begin{bmatrix} 1 \\ 1 \\ 0 \\ -1 \\ -1 \\ 0 \end{bmatrix}$$

The resulting F_{des} vector is identical to the F_{des} vector used to form F_{pos}. Therefore, it the distribution matrix effectively converts the desired body forces and torques into the individual thruster frames.

4.6. OSIRIS-REx TAG Maneuver Simulation Results

During the simulation, the spacecraft maneuver includes multiple simultaneous axis rotations, in addition to a positional change. The control objective of the simulation is to align the spacecraft body axis to Bennu's initially fixed axis system, and to perform a translational maneuver of the spacecraft along an arc towards Bennu's surface. Utilizing the thruster distribution matrix and controller, the control objective was met as seen in Figures 4.1, and 4.2. These two figures are the dual and real components of the dual quaternion respectively.

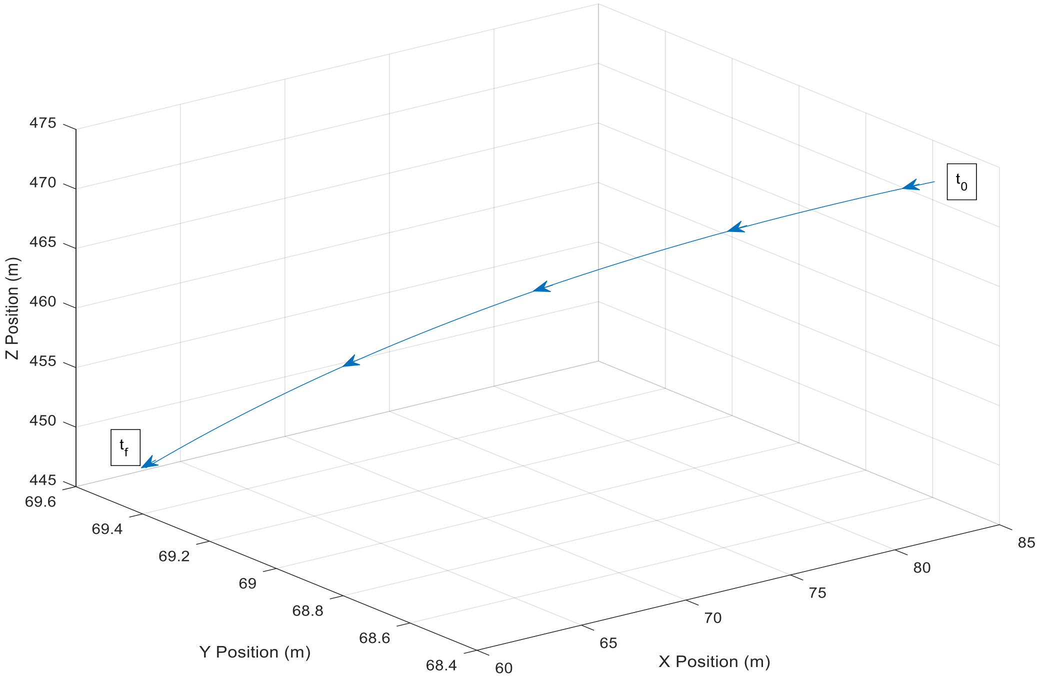

Figure 4.3 Spacecraft Position in Inertial Space

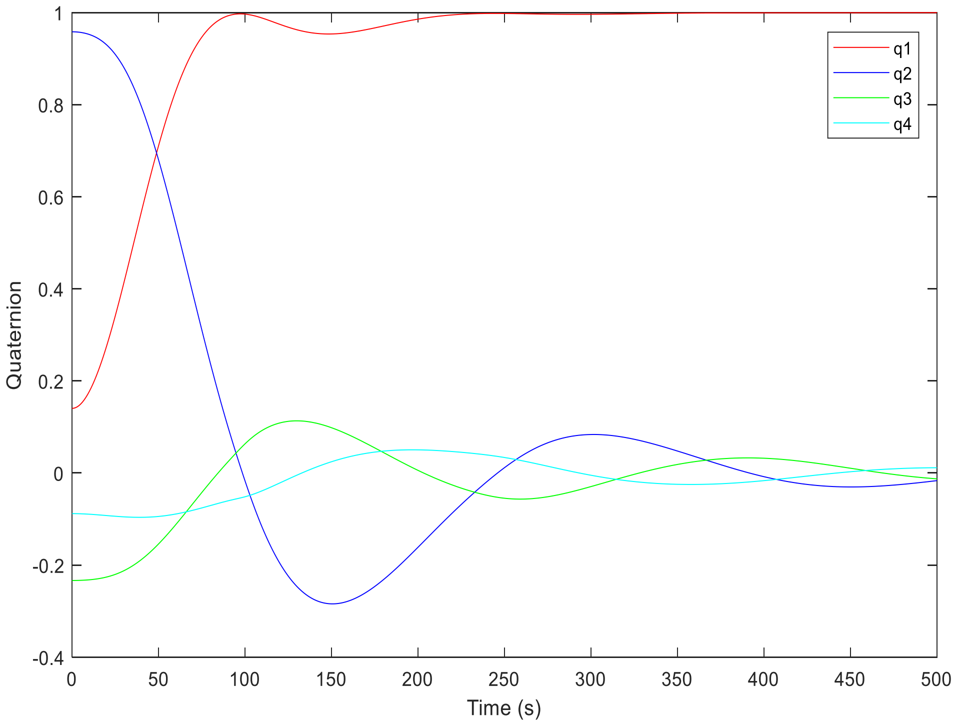

Figure 4.4 Spacecraft Quaternion Based Attitude in Inertial Space

To determine the effectiveness of the thruster distribution matrix, the command force and torque vector calculated by the control law was utilized as the F_{des} vector for each control step. This was inputted into Eqn. (33) with the thruster distribution to calculate the thruster force vector for each command cycle as seen in Figure (4.3). The thruster force vector was then converted back to body forces and torques to provide a straightforward comparison to the produced body forces and torques and the ideal body forces and torques as seen in Figure (4.4).

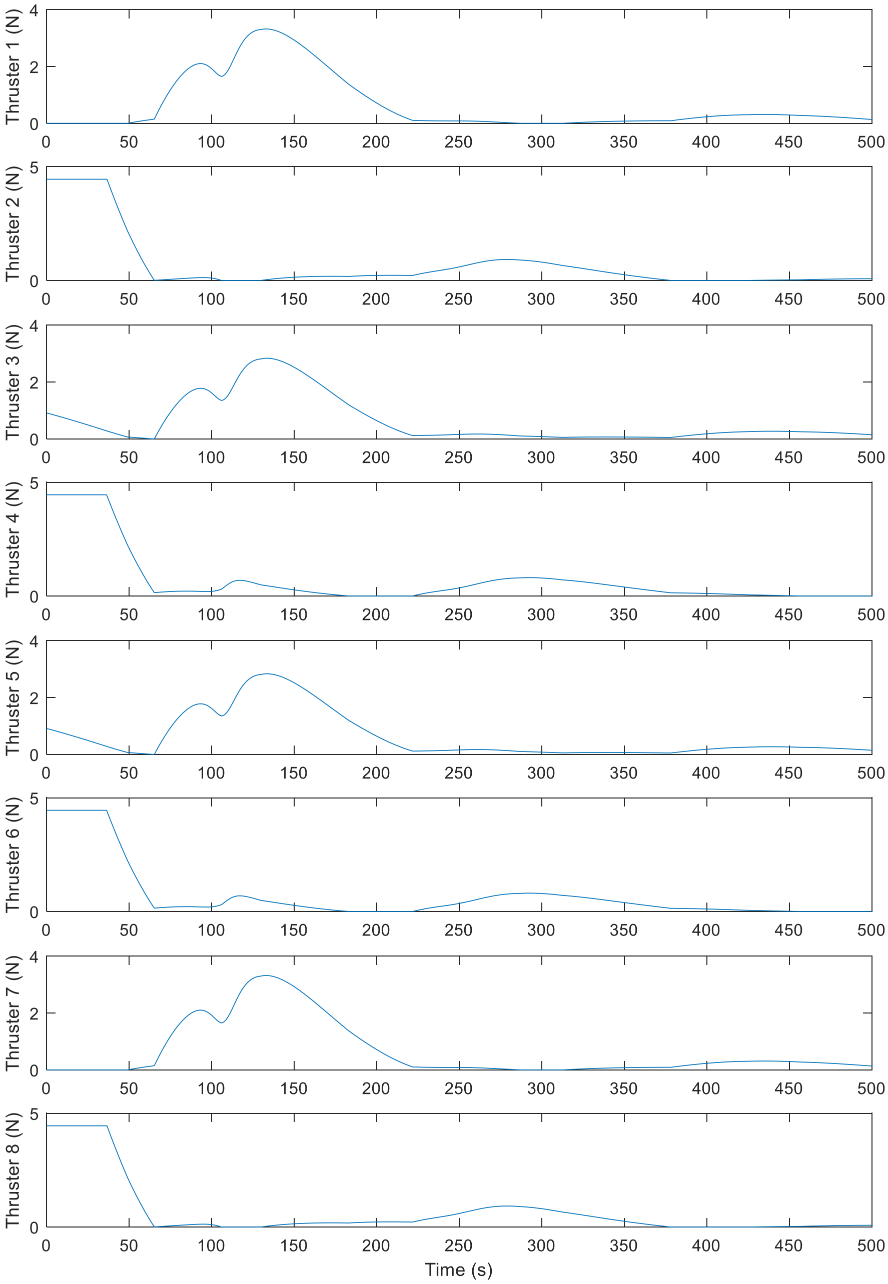

Figure 4.5 Thruster forces over the Bennu orbit simulation

The inclusion of thruster saturation is observable from these results as it is clearly seen in Figure 4.5 for thrusters 2, 4, 6, and 8, which are saturated for approximately the first 40 seconds of the simulation. This directly corresponds to the difference in the ideal body torque and the commanded resulting body torque as seen in Figure 4.6.

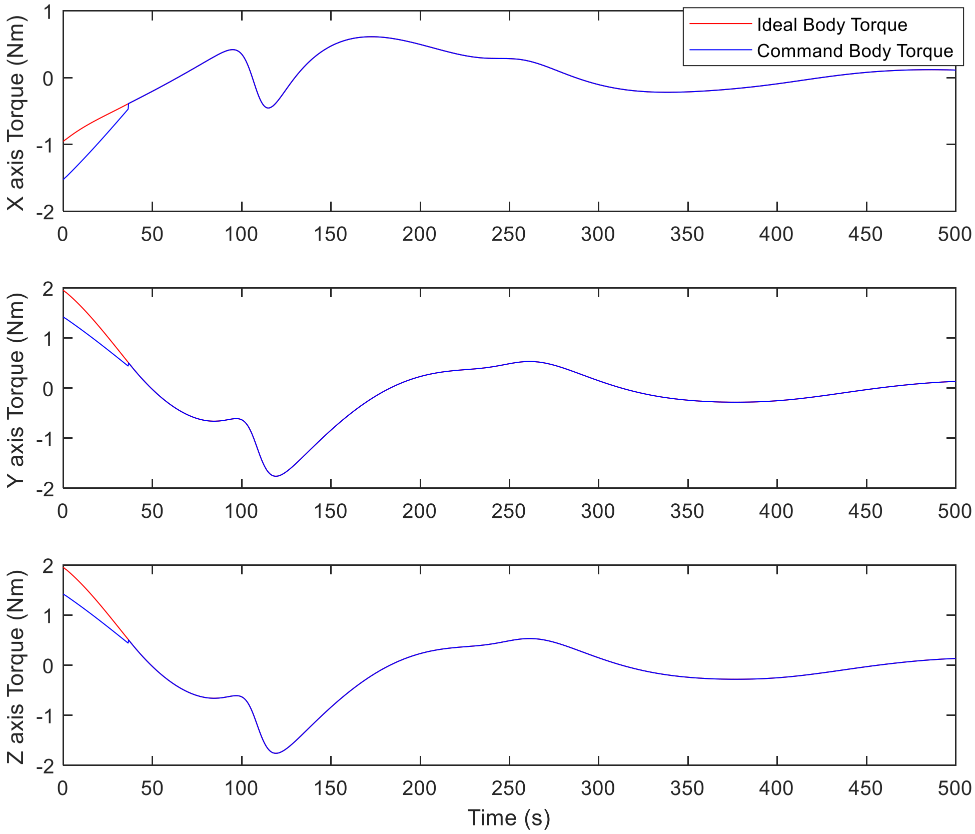

Figure 4.6 Body axis torque commands versus ideal torque commands

The precision of the command body torques in relation to the ideal body torques was analyzed per control cycle and there was an initial high percent error due to four of the thruster being saturated for the beginning of the simulation. Once the thrusters stopped being saturated, all of the percent errors were all contained within a 0.10 percent error for the remaining duration of the simulation as seen in Figure 4.7. Therefore, it is concluded that the pseudo inverse method of thruster distribution matches very closely to the ideal torque scenario.

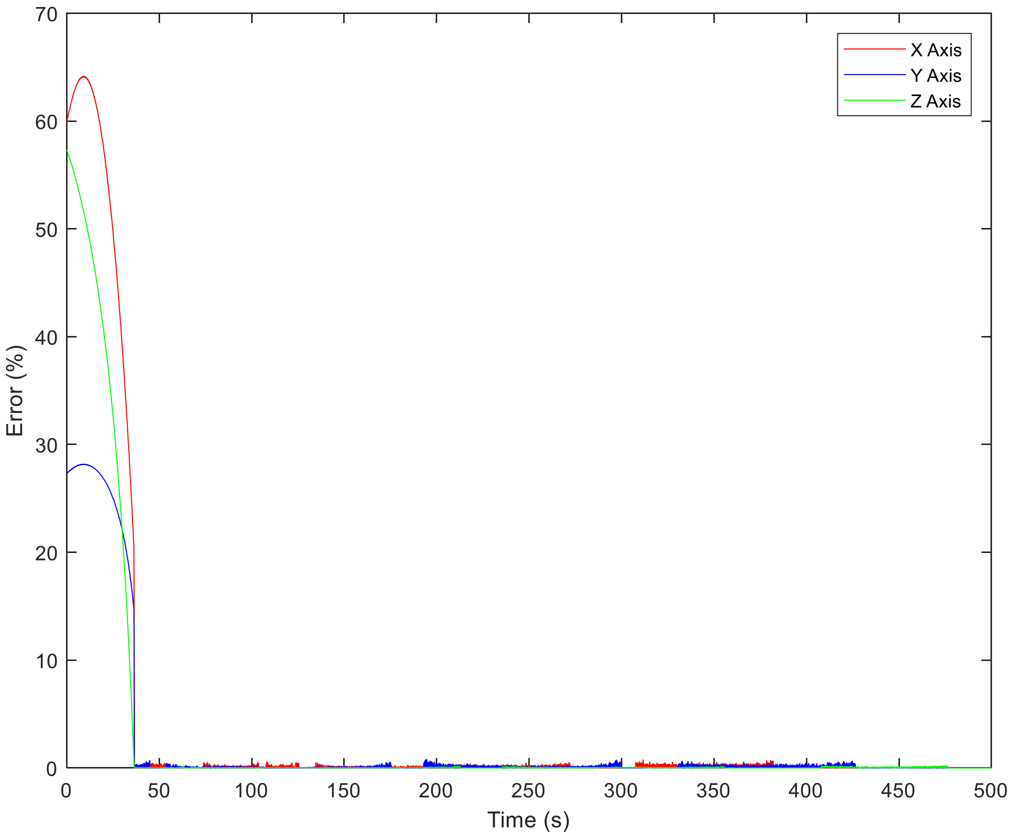

Figure 4.7 Body axis command torque error

The last area of analysis for the efficiency of the distribution logic is the angular rate and the attitude errors of the body frame. These errors are the actual spacecraft rates and angles produced by the thrusters in comparison to the desired rates and angles. The plots show the body angles and rates trending towards a steady state of zero error, and therefore further prove the distribution logic is preforming as expected.

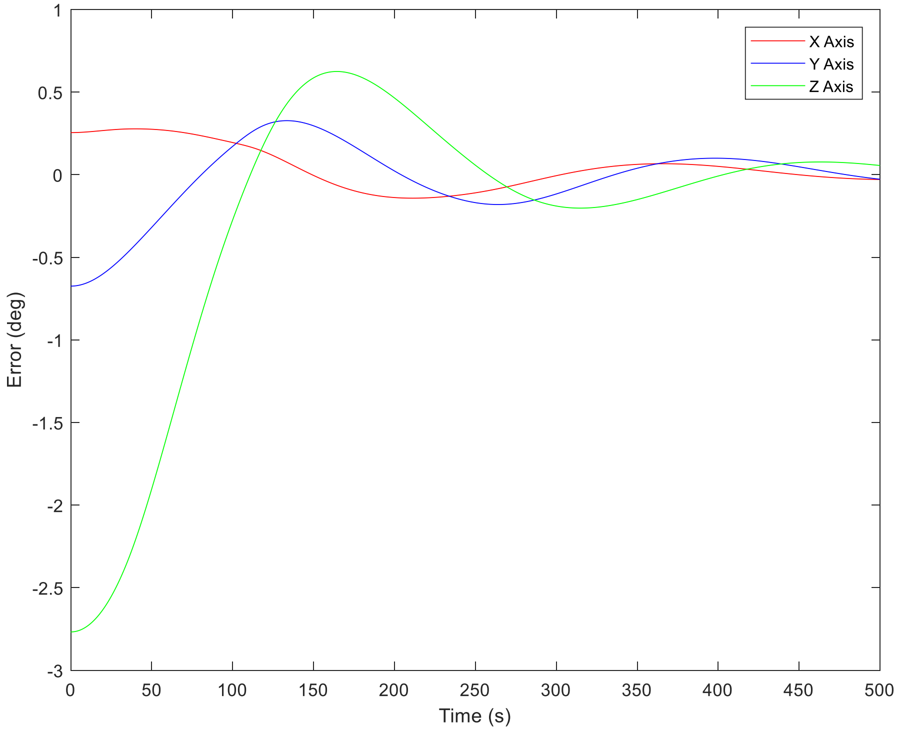

Figure 4.8 Body frame angle errors

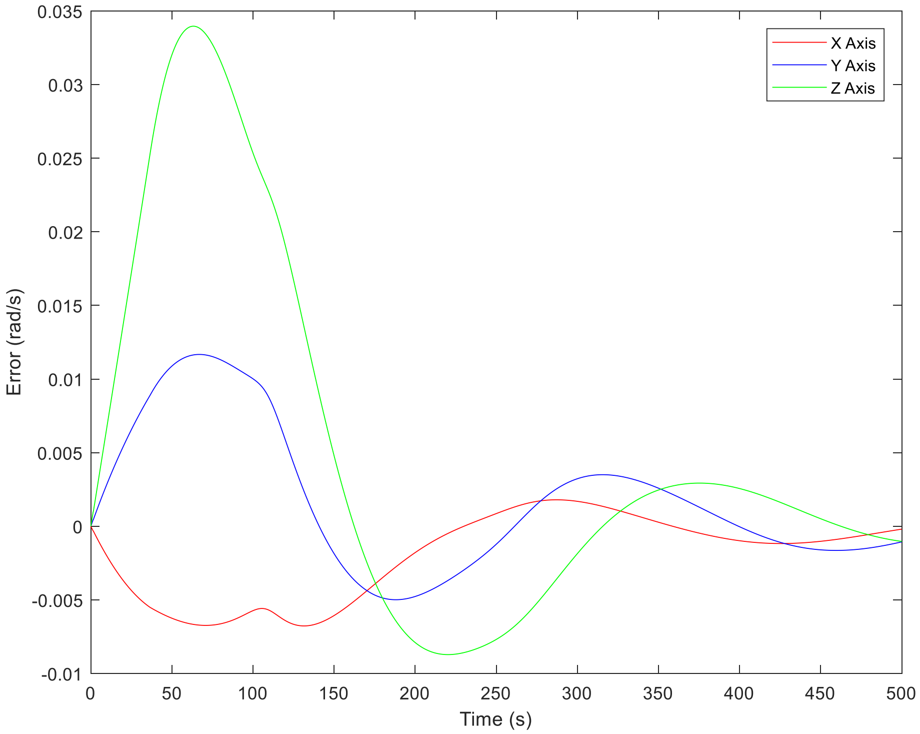

Figure 4.9 Body frame angular rotation rate errors

4.7. Non-Quantized and Non-Saturated Thruster Simulation Results

To further confirm the effectiveness of the thruster distribution matrix, the quantization and saturation of the thrusters were turned off for a simulation run in order to analyze the percent errors that the distribution matrix solution produces when compared to the ideal command input.

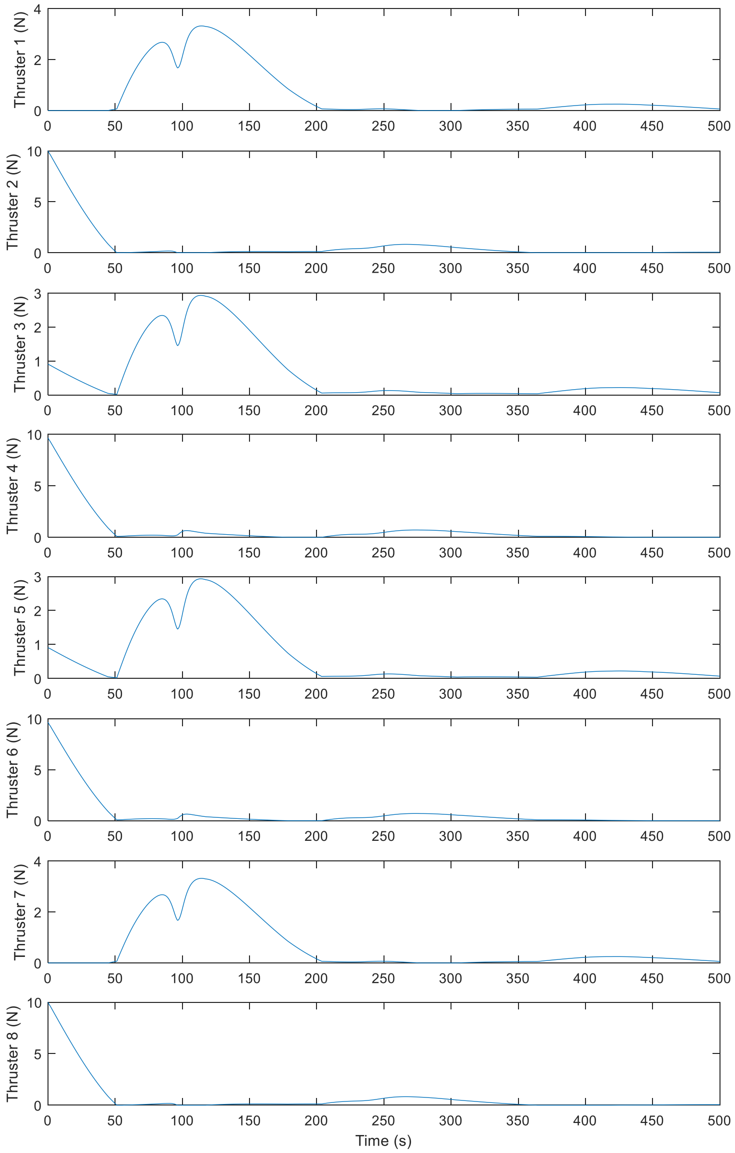

Figure 4.10 Non-quantized and non-saturated thruster forces over the Bennu orbit simulation with non-saturated and non-quantized thrusters

The differences can be immediately noticed due to the thrusters 2, 4, 6, and 8 no longer being saturated. Therefore, the resulting plots of the ideal body torque when compared to the command torque, showed little to no error.

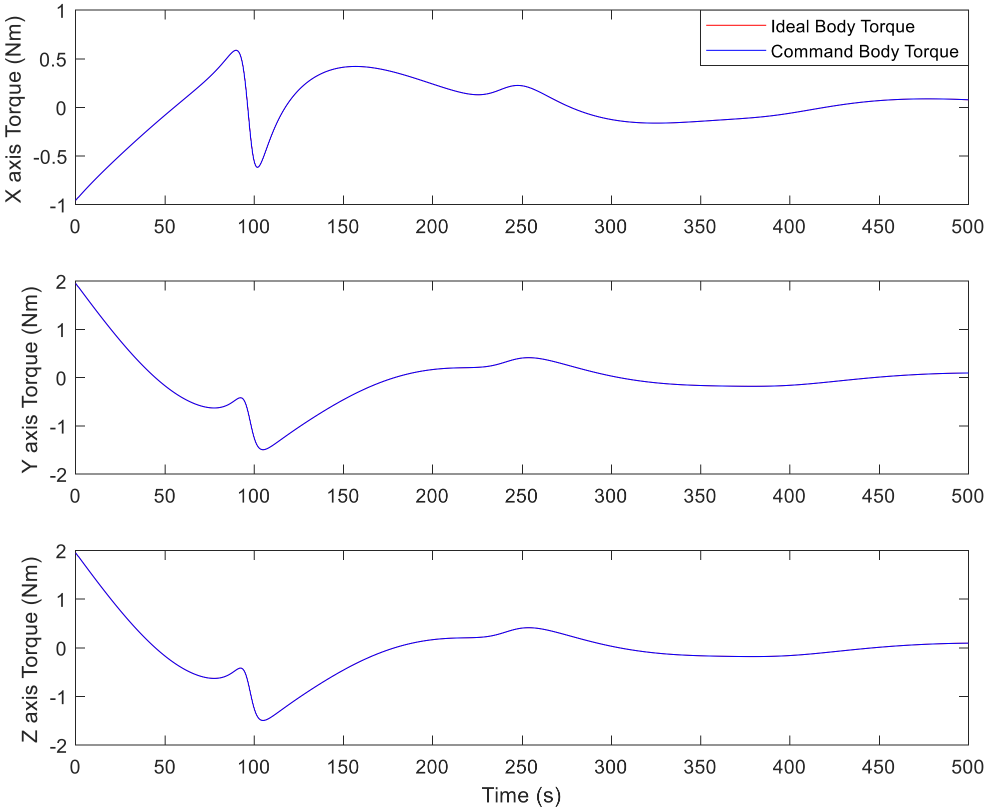

Figure 4.11 Body axis torque commands versus ideal torque commands with non-saturated and non-quantized thrusters

This is even more apparent when analyzing the graph of the individual control cycle command errors, which analyzed the error of the command torque compared to the ideal toque. The command errors had a maximum peak of just less than 0.1 percent about a single axis. The remaining two axis of the spacecraft all remained at a significantly lower percent error, which further emphasizes the distribution matrix's ability to efficiently distribute the control torques to all axis of the spacecraft.

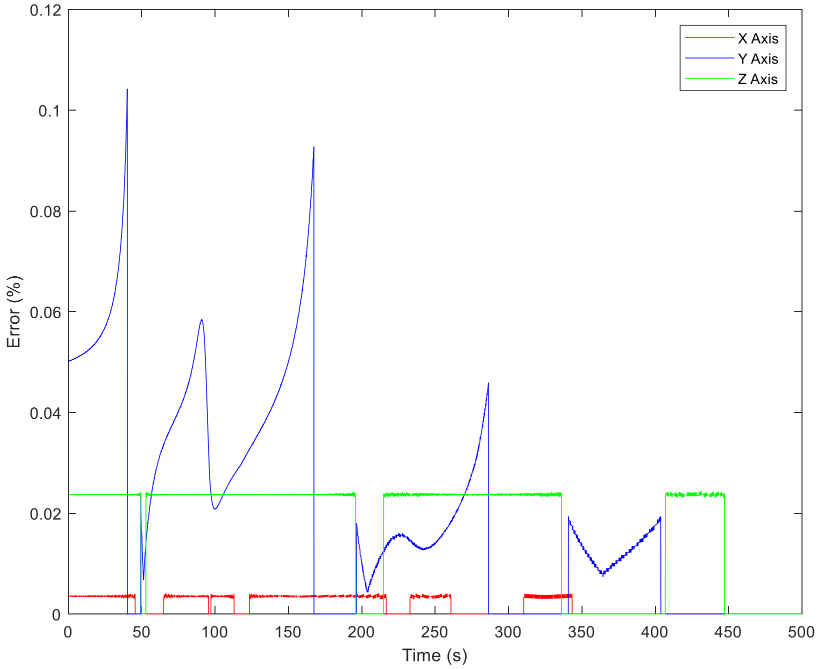

Figure 4.12 Body axis command torque error

For completeness, the following figures are included for the non-quantized and non-saturated thruster simulation. As expected the simulation still obtains the desired control object with slightly faster settling time and slightly higher rotational rates due to the increased torque capabilities. While the direct improvements of the results are not as apparent as they are in the previous figures, the trends are still able to be observed.

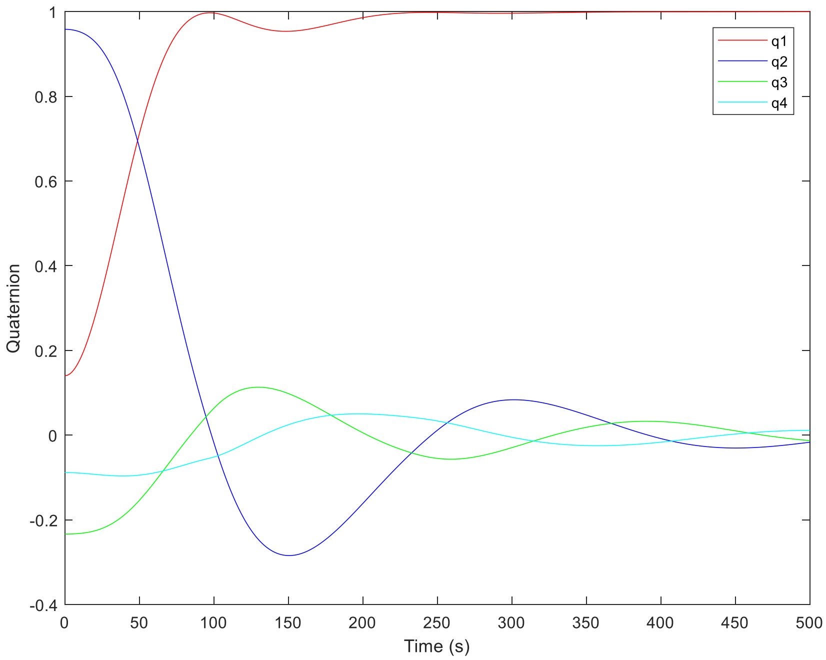

Figure 4.13 Spacecraft Quaternion Based Attitude in Inertial Space with non-saturated and non-quantized thrusters

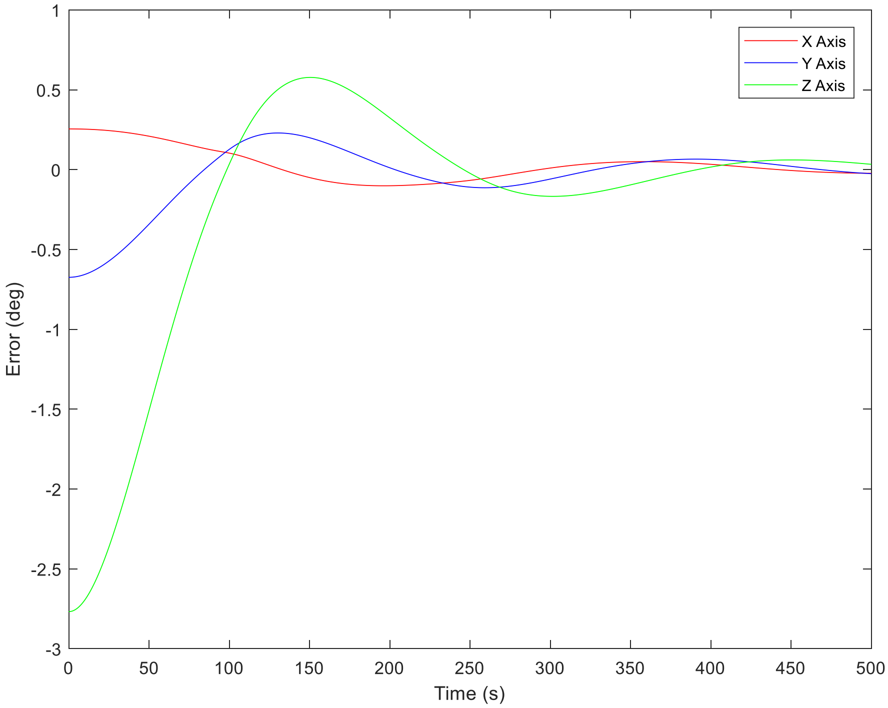

Figure 4.14 Body frame angle errors with non-saturated and non-quantized thrusters

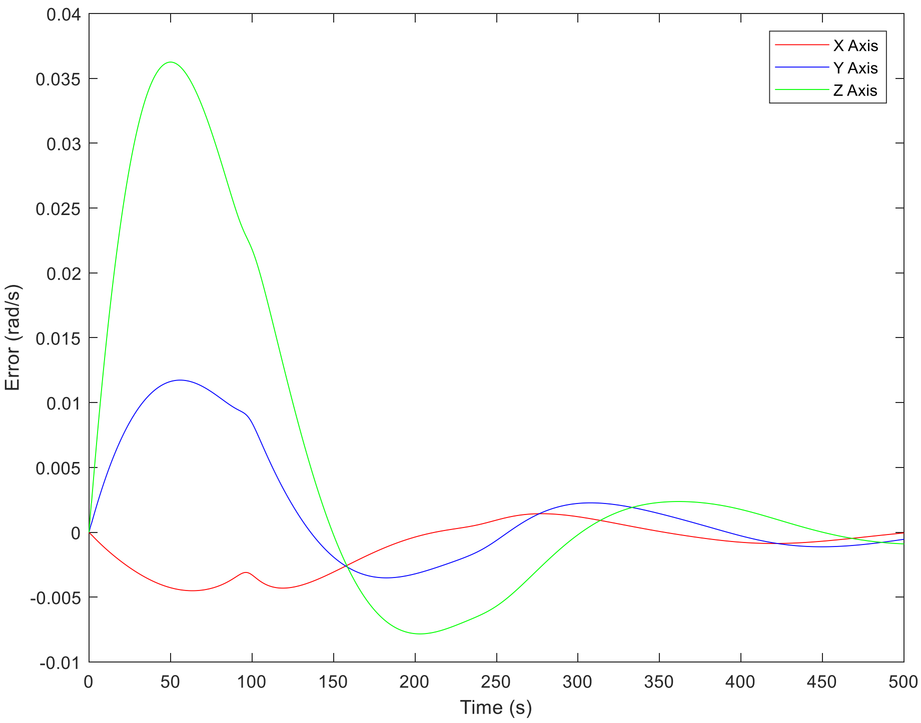

Figure 4.15 Body frame angle errors with non-saturated and non-quantized thrusters

5. Conclusion

The application of a dual quaternion dynamics simulation in combination with an optimized thruster distribution matrix is analyzed for this thesis. The decision to utilize the dual quaternion instead of a more traditional 3-dimension position vector and attitude quaternion is made due to the inherent benefits of the coupled system while utilizing dual quaternion. The dual quaternion provides a simulation in which the position and attitude equations of motion are coupled. Therefore, when compared to a traditional system where the translational orbit and attitude motion are decoupled, the interconnection between the two types of motion will be included. With this framework in place, the optimized thruster distribution matrix was utilized to provide a minimum thrust solution to the thruster allocation problem. This solution is more compact and computationally efficient than a typical thruster allocation problem utilized in most spacecraft today. Additionally, it resulted in a very accurate and precise solution when the resulting body forces and torques from the distribution matrix was compared to the ideal body forces and torques.

6. Recommendations

The primary area of future work that is recommended to investigate is the robustness of the thruster distribution matrix. Since the calculation of the optimized thruster distribution matrix requires exact model knowledge for the thruster locations and attitudes, the effect of slight variations in those numbers needs to be addressed. Additionally, the topic of failed thrusters and how the efficiency of the distribution matrix including the failed thruster compares to full distribution matrix without any failed thrusters is an area of interest. Next, investigation into an optimal controller that is optimized within the dual quaternion space is desired. Therefore, the entirety of the simulation will be within the dual quaternion format. Lastly, a stability analysis of the thruster saturation for cases outside of the analyzed simulation would be beneficial to the research's robustness.

REFERENCES

Ankersen, F., Aleshin, A., Vankov, A., & Volochinov, V. (2005). Optimization of Spacecraft Thruster Management Function. *Journal of Guidance, Control, and Dynamics, 28*(6), 1283-1290.

Aspragathos, N., & Dimitros, J. (1998). A comparative study of three methods for robotic kinematics. *IEEE Trans. Syst. Man Cybern. Part B: Cybern, 28*(2), 135-145.

Bodson, M. (2002). Evaluation of Optimization Methods for Control Allocation. *Journal of Guidance, Control, and Dynamics, 25*(4), 703-711.

Crawford, B. (1969). Configuration Design and Efficient Operation of Redundant Multi-Jet Systems. *AIAA Guidance, Control and Flight Mechanics Conference* (pp. 69-845). Princeton, New Jersey: AIAA.

Curtis, H. (2010). *Orbital Mechanics for Engineering Students* (Vol. 2). Elsevier Ltd.

Doman, D., Gambel, B., & Ngo, A. (2007). Control Allocation of Reaction Control Jets and Aerodynamic Surfaces for Entry Vehicles. *AIAA Guidance, Navigation and Control Conference and Exhibit* (pp. 1-24). Hilton Head, SC: AIAA.

Funda, J., & Paul, R. (1990). A computational analysis of screw transformations in robotics. *IEEE Trans. Robot Autom., 6*(3), 348-356.

Funda, J., Taylor, R., & Paul, R. (1990). On homogeneous transforms, quaternions and computational efficiency. *IEEE Trans. Robot. Autom, 6*(3), 382-388.

Garus, J. (2004). Optimization of Thrust Allocation in the Propulsion System of an Underwater Vehicle. *AMCS International Journal of Applied Mathematics and Computer Science, 14*(4), 461-467.

Johansen, T., & Fossen, T. (2013). Control Allocaiton-A Survey. *Automatica, 49*(5), 1087-1103.

Optimization Toolbox 4 User's Guide. (2008). Chap. 4, Massachusetts: The Mathworks.

Seo, D. (2015). Fast adaptive pose tracking control for satellites via dual quaternion upon non-certainty equivalence principle. *Acta Astronautica 115*, 32-39.

Smith, A., & Seo, D. (2017). Spacecraft Thruster Distribution Matrix for Precision 6DOF Control. *AIAA SPACE and Astronautics Forum and Exposition*.

Wang, J., & Sun, Z. (2012). 6-dof robust adaptive terminal sliding mode control for spacecraft formation flying. *Acta Astronautica, 73*(0), 76-87.

Wang, M. (2009). Design of the Optimal Thruster Combinations Table for the Real Time Control Allocation of Spacecraft Thrusters. *Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference* (pp. 5063-5068). IEEE.

Wu, Y., Hu, X., Hu, D., Li, T., & Lian, J. (2005). Strapdown inertial navigation system algorithms based on dual quaternions. *IEEE Trans*. *Aerospace Electronic Systems*, *41*(1), 110-132.

A. Pseudo Inverse Optimization MATLAB Code

```
 %% ///////////////////////////////////////////////////////////////////
% File: [Min, MinId, Max, MaxId, y] = CalcExtrema(data)
% Authors: Asher Smith
% Revision: 8/18/2017
% Purpose: Computes the thrust distribution matrix for nt = 8 thusters
%          in 2 clusters of 4.
% Output: Vector [nt x 12]  thrust distribution matrix 'Dist'.
%         The 12 columns of Dist are:
%         [+Fx +Fy +Fz +Tx +Ty +Tz -Fx -Fy -Fz -Tx -Ty -Tz]
%               where +Fi (-Fi) denotes a positive (negative) unit
%               force along axis i
%                     +Ti (-Ti) denotes a positive (negative) unit
%                     torque around axis i.
%
%//////////////////////////////////////////////////////////////////////

clc
clear
close all
set(0,'defaultfigurecolor',[1 1 1])

global Minv b Vnull

%% Set Initial Conditions and Variables
nt = 8;                         % number of thrusters
X  = 0.5;                       % distance of clusters from center of
spacecraft
d  = 0.03;                      % distance of thrusters from center of
cluster
cm = [-0.006; 0.003; 0.77];     % center-of-mass location
r2 = sqrt(2);
r3 = sqrt(3);
a=1;

% Thrust directions
M = [0.433013   -0.433013   -0.433013   0.433013    0.433013
    -0.433013   -0.433013   0.433013;...
     0.25       0.25        -0.25       -0.25       0.25        0.25
    -0.25       -0.25;...
     0.866025   0.866025    0.866025    0.866025    -0.866025
    -0.866025   -0.866025   -0.866025];

% Thruster locations
R = [1.227      -1.227      -1.227      1.227       1.227       -1.227
    -1.227     1.227;...
     1.157      1.157       -1.157      -1.157      1.157       1.157
    -1.157     -1.157;
     1.725056   1.725056    1.725056    1.725056    0.421366
     0.421366    0.421366   0.421366];
```

```
% Adds three rows to M for the torques, given by the cross products of
the lever arms (R-cm) and the thrusts.
M(4:6,:) = cross(R(1:3,:)-kron(cm, ones(1,nt)),M(1:3,:));

%% Compute pseudoinverse and null space basis
[U,S,V] = svd(M);
Minv = V(:,1:6)*S(:,1:6)^-1*U';
Vnull = V(:,7:nt);

%% Find force/torque distribution matrix. The index j cycles over unit
% positive and negative forces and torques for each axis.
% The general thrust solution is Minv*b+Vnull*x. 'minimax'
options1=optimoptions('fminimax','MinAbsMax',6); % Minimize abs values
f = ones(nt,1);
z = zeros(6,1);
for j = 1:12
    b = z;
    x = zeros(nt-6,1);
    if j>6
        b(j-6,:) = -1;
    else
        b(j) = 1;
    end
    constr = Minv*b;
    x = fminimax('dfun',x,-Vnull,constr,[],[],[],[],[],options1);
    Dist(:,j) = Minv*b + Vnull*x;
end
disp('Distibution Matrix = ')
disp(Dist)

% Numerical 0 Elimination
for a = 1:8
    for b = 1:12
        if Dist(a,b) < 1e-10
           Dist(a,b) = 0;
        end
    end
end
disp(Dist)



function [f,g] = dfun(x)
global Minv b Vnull

f = Minv*b + Vnull*x;
g = -f;
end
```