Simulation Development and Analysis of Attitude-Control System Architectures for an Astronaut Mobility Unit

by

Jared Philip Rize

B.S., Lehigh University (2012)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautical and Astronautical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2014

Signature redacted

Author .....
Department of Aeronautics and Astronautics
May 9, 2014

Certified by ..... Signature redacted .....
Jeffrey A. Hoffman
Professor of the Practice of Aerospace Engineering
Thesis Supervisor

Certified by ..... Signature redacted .....
Michele D. Carpenter
Senior Member of the Technical Staff
Signature redacted Thesis Supervisor

Accepted by .....
Paulo C. Lozano
Associate Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

Simulation Development and Analysis of Attitude-Control System Architectures for an Astronaut Mobility Unit

by

Jared Philip Rize

Submitted to the Department of Aeronautics and Astronautics
on May 9, 2014, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautical and Astronautical Engineering

Abstract

Control-moment gyroscopes (CMGs) are spacecraft attitude-control actuators which control the spacecraft's orientation and pointing. CMGs operate on electrical power and therefore obey the the conservation of angular momentum. Single-gimbal CMGs are equipped with a high-speed flywheel which can be gimbaled to impart gyroscopic torques. The net reaction torques are observed by the spacecraft resulting in pure rotation. A CMG based attitude control system (ACS) is favorable compared to a cold gas thruster ACS because of fundamental differences in how the reaction torques are produced. CMGs provide a continuous range of motion while RCS thrusters are limited by the minimum on-off time for the thruster valves. This minimum open-close time leads to a bang-bang response as opposed to the smoother CMG response. Furthermore, CMGs are powered using batteries and can therefore be recharged, while RCS thrusters use propellant which depletes over time. CMG sizing, the act of designing and choosing the electrical and mechanical parameters for a given spacecraft ACS, is studied in this thesis. The CMG sizing tool analyzes the specific system configuration (i.e. mass properties, thruster location and placement, CMG architecture, etc.) and the mission and system requirements to provide an "idealized" CMG model. Detailed simulation results and recommendations are presented for the design and analysis of the Mobility Augmenting Jetpack with Integrated CMGs (MAJIC) system. The CMG sizing software acts as a parametric tool which can be adopted to any spacecraft system.

Thesis Supervisor: Jeffrey A. Hoffman
Title: Professor of the Practice of Aerospace Engineering

Thesis Supervisor: Michele D. Carpenter
Title: Senior Member of the Technical Staff

Acknowledgments

I would like to thank Draper Laboratory and the numerous advisors for all of their technical support and guidance throughout the Next-Generate EVA project. In particular, I would like to thank Dr. Michele Carpenter and Lee Yang for their expertise in CMGs and spacecraft ACS development and deployment. They assisted with not only the theoretical aspects of the research, but also with the implementation in MATLAB/Simulink. Dr. Bobby Cohanim provided great insight into the overall system development and organization. Without his pertinent oversight, the overhead of developing an entire real-time virtual reality simulation would have been much more difficult. Also, without the IR&D and UR&D funding, this project would not have been possible. The resources Draper provides to it's Fellows is outstanding and greatly help motivate the work for this research effort.

NASA and, in particular, the members of JSCs VR Lab were instrumental in connecting the MATLAB/Simulink with the visualization tool, EDGE. Their expertise proved extremely useful in the piloted human-in-the-loop evaluations conducted on-site at JSC. I would also like to thank James Tinch for his effort in coordinating the NASA crew-members to evaluate the simulation. Brad Bell, James Hobblit, and Frank Graffagnino helped extensively with interfacing the multiple simulation environments and developing the three-dimensional models used in the simulation. The experience of working at NASA and with current and former crew-members is one I will never forget. I can only hope that one day this work will directly benefit them and NASA's overall pursuit of exploring the far reaches of the universe.

Lastly, I would like to thank my parents and family for all of their support during what I am sure was a trying time. They believed in this effort and wished success before ever learning about the specific research objectives. My parents have always been the most supportive and willing to help whether its regarding professional or academic interests. They have taught me to be humble and approach the world with compassion and not cynicism. They have installed in me the valuable lessons that only a caring parent can and have instilled in me the idea that I try and live my life

according to:

Good, Better, Best

Never Let it Rest Until Your Good is Better

And Your Better is Best

- *St. Jerome*

Contents

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 1

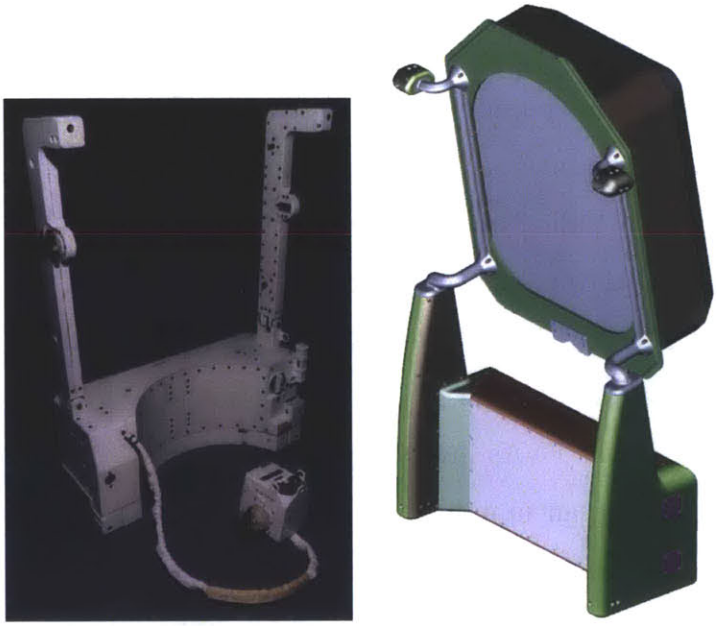
Introduction

1.1 Background

Current safety regulations require that an astronaut performing an extra-vehicular activity (EVA) have a stainless-steel umbilical tether that connects the astronaut to a hard-point such as a railing on the International Space Station (ISS). The tether is in place in case the astronaut becomes disoriented or is struck by a piece of debris moving away from the ISS. The dependency on using a tether requires that the astronaut performs work in a known, cooperative environment. However, this strategy fails when thinking about EVAs involving uncooperative targets such as a tumbling satellite where attaching an umbilical cable might not be feasible. Also, the use of a tether might be hindered during an EVA exploration of an unknown target such as an asteroid where man-made hand-holds and railings are unavailable. To address the current limitations placed on EVA tasks while still ensuring that the astronaut is working in a safe environment, NASA has developed and flown the Manned Maneuvering Unit (MMU) and Simplified Aid for EVA Rescue (SAFER) devices. Both devices use reaction control system (RCS) thrusters to allow the astronaut to translate and work untethered. The SAFER system is currently worn on tethered ISS EVAs. In the event that the tether breaks or becomes detached, the astronaut can still use the propulsive SAFER system to return to the spacecraft. The SAFER system has only been approved for use as an emergency rescue device and not an operational

unit. If the SAFER were an operational unit, it would allow the astronauts the flexibility and freedom to perform work at unknown, uncooperative target locations while still maintaining its effectiveness as a self-rescue device. NASA has recently begun working on a Jetpack design that would be an operational successor to the SAFER device.

NASA's current plan for the Jetpack follows very much the same design as the SAFER: utilizing a hand-held controller to input incremental $\Delta\mathbf{V}$ changes to the RCS thrusters. The major design revision is the inclusion of the suitport interface on the rear of the Jetpack. Fig. 1-1a shows the current SAFER design and the proposed Jetpack design.



(a) SAFER device (b) Jetpack device

Figure 1-1: NASA's SAFER device and proposed Jetpack

The remainder of this thesis will be dedicated to exploring the trade space when adding control moment gyroscopes (CMGs) to the NASA Jetpack design. The new design will be referred to as the Mobility Augmented Jetpack with Integrated CMGs (MAJIC). The trade space will be limited to single-gimbal CMGs. A blended attitude-control system (ACS) with CMGs and RCS thrusters provides several benefits over an ACS with only RCS thrusters. For instance, CMGs use electrical power and do

not consume propellant. This feature provides a substantial benefit when considering using the MAJIC system as an operational unit. Because CMGs use electrical power, they can be run using batteries, and the potential for rechargeable solar batteries increases the longevity of a CMG-augmented Jetpack. Also, CMGs provide continuous attitude actuation, whereas thrusters operate on a discrete cycle defined by the minimum on-time of the thruster. The minimum on-time of a thruster is the length of time taken to open and close the thruster's solenoid. The continuous nature of the CMGs means that zero (near zero) attitude error is possible. For applications which have strict pointing requirements and therefore need an extremely low attitude error, only the blended (e.g. combined RCS thrusters and CMGs) ACS can meet the pointing requirements. Furthermore, when working in potentially foreign and uncooperative environments, the plumes associated with thruster firings may be undesirable. Consider an astronaut conducting an exploration EVA near the surface of an asteroid: each time one of the forward-facing thrusters fire, debris scattering from the surface of the asteroid can cause potentially dangerous visibility conditions. The CMGs do not produce any such plume effect and therefore can be used for attitude actuation at the surface of an asteroid without worrying about creating a debris field. CMGs provide several obvious benefits over RCS thrusters for attitude-control; however, the decision to use CMGs must also consider some of their shortcomings, which will be discussed in detail in Sec. 2.4.1 and 2.4.3.

The baseline design for the MAJIC is NASA's preliminary Jetpack design. NASA has provided system dimensions and properties for the current Jetpack design. The Jetpack has 24 RCS thrusters, which are arranged in four pods. Fig. 1-2 shows the thruster locations (expressed in the Jetpack body-fixed coordinate frame, $\mathcal{F}_B$). The thrusters are labeled 0 through 23 and the arrows indicate the direction of motion associated with firing each thruster. For instance, firing thrusters 0,1,2 and 3 would result in a net motion along the Jetpack's $-\hat{e}_{1B}$ axis. Tab. 1.1 provides the thruster locations in $\mathcal{F}_B$, as well as the direction of motion associated with firing each thruster.

NASA has indicated that the Jetpack system will use either Nitrogen gas (GN_2) or Tridyne (91% GN_2, 6% H_2, 3% O_2 molar) [3]. Nitrogen gas typically has a specific

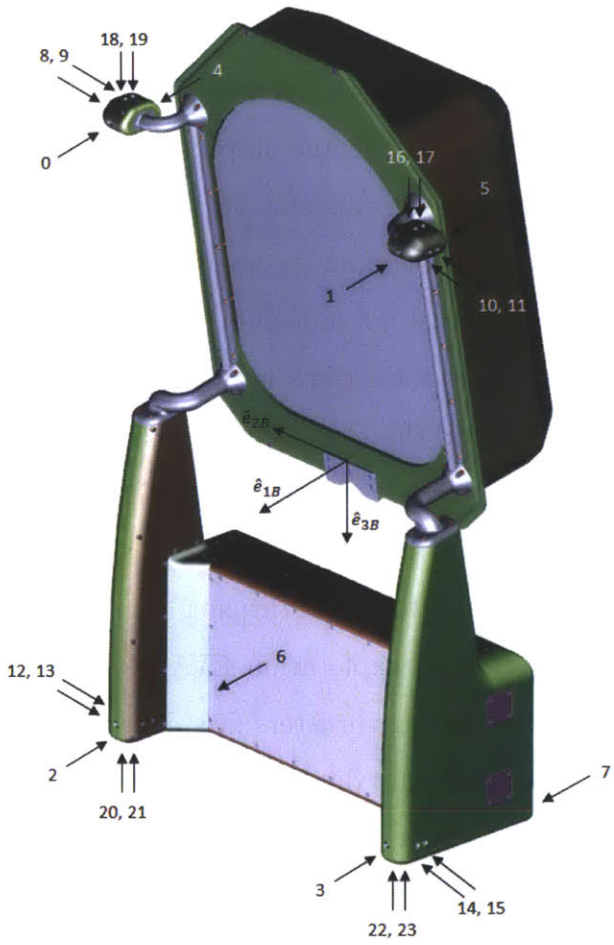


Figure 1-2: Jetpack RCS thruster locations

impulse, or Isp, of 72s, while Tridyne has an estimated Isp of 135s. The increased Isp of Tridyne will allow a Jetpack outfitted with Tridyne to last approximately twice as long as a system outfitted with pure Nitrogen gas assuming the same $\Delta\mathbf{V}$ requirements. The 24 RCS thrusters are each capable of producing 3.56 N (0.8 lbf) of thrust. Knowing the baseline Jetpack configuration, a parametric spacecraft simulation environment (Sec. 3) was developed to allow for rapidly changing and testing control algorithms on the overall MAJIC system performance. The simulation environment can also be used to size the CMGs by simulating an ensemble of mission scenarios using a Monte Carlo approach. The mission trajectories chosen for the CMG sizing study were developed to address EVA missions that cannot currently be completed or are made more challenging because of the requirement to use a tether. The missions,

Table 1.1: RCS Thruster Locations ($\mathcal{F}_B$)

Thruster	X (in)	Y (in)	Z (in)	Direction
0	3.510	14.070	-19.870	−X
1	3.510	-14.070	-19.870	−X
2	3.397	13.143	23.659	−X
3	3.397	-13.140	23.659	+X
4	-0.066	14.070	-19.870	+X
5	-0.069	-14.070	-19.870	+X
6	-8.600	13.750	23.986	+X
7	-8.600	-13.750	26.986	+X
8	2.060	15.400	-19.870	−Y
9	1.380	15.400	-19.870	−Y
10	2.060	-15.400	-19.860	+Y
11	1.380	-15.400	-19.860	+Y
12	1.946	14.455	23.659	−Y
13	1.267	14.455	23.659	−Y
14	1.947	-14.500	23.659	+Y
15	1.267	-14.500	23.659	+Y
16	2.060	-14.070	-21.194	+Z
17	1.383	-14.070	-21.194	+Z
18	2.060	14.070	-21.197	+Z
19	1.383	14.070	-21.197	+Z
20	1.947	13.143	24.986	−Z
21	1.267	13.143	24.986	−Z
22	1.947	-13.140	24.986	−Z
23	1.267	-13.143	24.986	−Z

as well as the parametrically varied parameters, of the Monte-Carlo case study are presented in Sec. 3.3. The overall goal of the Monte-Carlo simulation is to determine what size CMGs, for a given CMG architecture, are required to complete the mission trajectories of interest and to determine the power and energy consumption so that approximations on the batteries required (i.e. type, number, power limit, etc.) can be made. Knowing the electrical subsystem requirements for the batteries will allow insight into the overall CMG system mass and footprint, which can be used to determine whether or not the CMGs can "buy" their way on-board. In other words, whether the additional mass of the new subsystem (i.e. CMGs, batteries, controller,

etc.) is greater than, less than, or equal to the propellant mass savings.

1.2 Literature Review

CMGs have long been considered for use in spacecraft ACS because of their reusable power sources and the large output torques created by relatively small gimbal inputs. The main drawback of using CMGs is that additional attitude-control actuators must be used to desaturate the CMGs when encountering a singularity (see Sec. 2.4.1). Several different papers have suggested using environmental disturbance torques for desaturation. Gravity gradient, aerodynamic, magnetic, solar radiation, and other environmental torques can be used to lower the angular momentum of the CMG architecture [4]. Additionally, magnetic torquers [5] can also be used to desaturate the CMGs without using any propellant from the RCS thrusters.

CMGs can be classified according to either single- or dual-gimbal depending on the physical design of the system. Single-gimbal CMGs use a single gimbal and motor to rotate the constant angular momentum vector of the flywheel to create gyroscopic torques. Dual-gimbal CMGs have an additional gimbal and motor and are capable of pointing the angular momentum vector in any direction. The versatility of the dual-gimbal CMG can be extremely beneficial in singularity avoidance, which is further discussed in Sec. 1.2.2. The majority of CMGs will have a constant angular momentum vector with a constant rotor speed; however, variable-speed CMGs can change the rotor speed similar to a reaction wheel (RW). RWs, commonly referred to as momentum wheels, act as attitude actuators by varying the flywheel speed to impart torques through the conservation of angular momentum. The additional degree of freedom can be used to avoid singularities in a similar manner to dual-gimbal CMGs.

The notion of using CMGs as attitude-control actuators on mobility units is itself not a novel idea [6]. The two automatic control modes for the Astronaut Maneuvering Research Vehicle (AMRV) are compared in terms of attitude and attitude rate during maneuvering [6]. However, due to the state of CMG technology and mission concept of

operations (CONOPS) at the time of the Skylab experiment, CMGs were not chosen for further consideration. With increased efficiency and small-scale CMGs capable of producing larger torques, the idea of using CMGs on a mobility unit is once again being considered. The addition of CMGs provides a reduction of fuel consumption, attitude error, and an extension of the system's life cycle attributable to the CMG's rechargeable power source.

1.2.1 CMG Sizing

CMG sizing refers to the process of sizing the spacecraft's ACS given the prospective mission trajectories of the spacecraft. Depending on the manner in which the ACS will be used, the sizing requirements for the CMGs will differ. Traditionally, single-axis maneuvers are considered for a first-order sizing method [7]. Given information regarding the spacecraft's mass and moment of inertia (MOI) properties, as well as a desired maneuver rate, a required torque can be established for the particular CMG architecture. The process of sizing a CMG for a given single-axis maneuver is:

$$\tau_{\text{req}} = \mathrm{I}_{s/c}\ddot{\theta} \tag{1.1}$$

where τ_{req} is the required torque for the given maneuver, $\mathrm{I}_{s/c}$ is the spacecraft's MOI for the particular axis, and $\ddot{\theta}$ is the angular acceleration of the spacecraft. Integrating Eq. 1.1 with respect to time (assuming constant τ_{req}) produces

$$\theta = \frac{1}{2}\frac{\tau_{\text{req}}}{\mathrm{I}_{s/c}}t^2 \tag{1.2}$$

which can be re-written in the following form

$$\tau_{\text{req}} = \frac{2\theta \mathrm{I}_{s/c}}{t^2} \tag{1.3}$$

where θ is the desired attitude change and t is the time allotted for the attitude change. Given a single CMG, the torque output on the spacecraft can be written

$$\tau_{\text{CMG}} = |h \times \dot{\delta}_{\text{max}}| \tag{1.4}$$

where h is the magnitude of the angular momentum vector and $\dot{\delta}_{\text{max}}$ is the maximum gimbal rate. Simplifying the previous equation yields

$$\tau_{\text{CMG}} = k|\text{I}_{\text{rotor}}\omega_{\text{rotor}} \times \dot{\delta}_{\text{max}}| \tag{1.5}$$

which shows that the torque output of the CMG is a function of its rotor inertia I_{rotor}, rotor speed ω_{rotor}, and maximum theoretical gimbal rate $\dot{\delta}_{\text{max}}$, as well as scalar constant k which is related to the CMG architecture chosen for the final design. CMG sizing will be investigated in greater detail in Sec. 2.4.2.

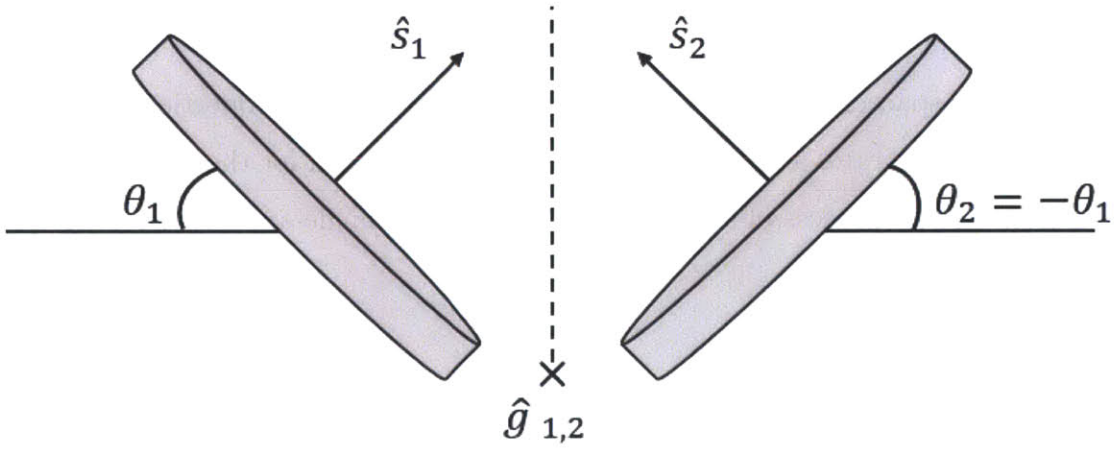


Figure 1-3: Scissored Pair - Vector Diagram

Fig. 1-3 shows that a scissored-pair CMG configuration provides a varying-magnitude, constant-direction torque. The configuration is arranged such that the sum of the CMGs angular momentum vectors ($\mathbf{h} = \sum_{i=1}^{2} \mathbf{s}_i$) is projected along a single axis. Since the gimbal axes are fixed, the resulting torque is constrained to a single axis in the spacecraft's body-fixed frame. The magnitude of the torque is related to the vector sum of the CMGs angular momentum vectors and is bounded by $\left[-|2h \times \dot{\delta}_{\text{max}}|, |2h \times \dot{\delta}_{\text{max}}| \right]$. Figs. 1-4 and 1-5 illustrate the gimbal configurations that result in minimal and maximmal angular momentum projections, respectively.

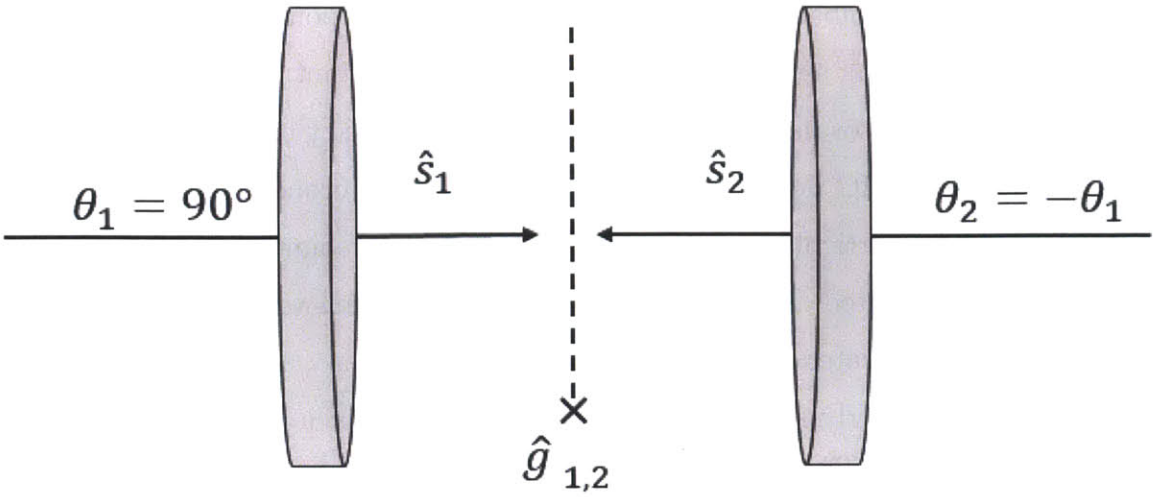


Figure 1-4: Scissored Pair - Minimal Projection

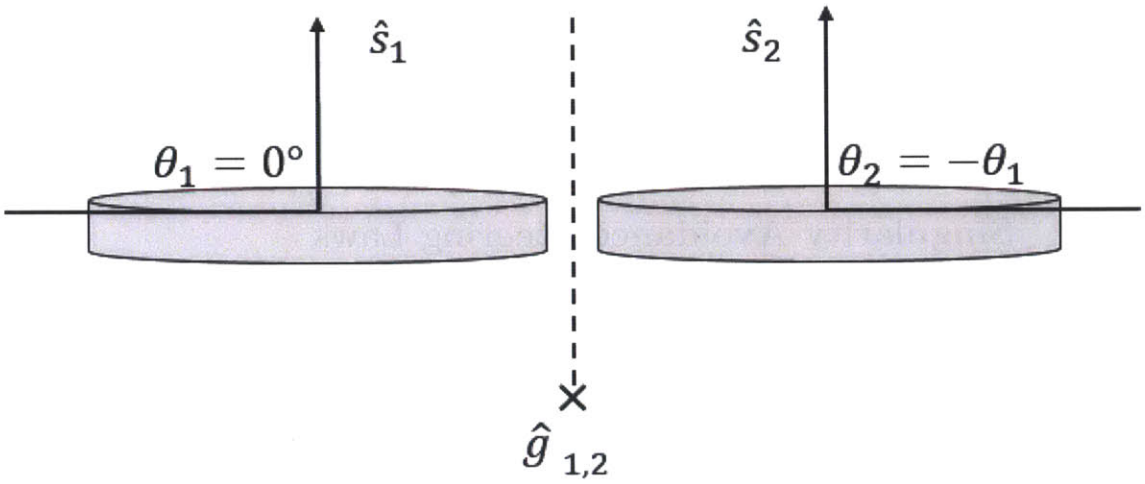


Figure 1-5: Scissored Pair - Maximal Projection

Eq. 1.5 illustrates that requiring the output torque of the CMG architecture to be equal to the required torque for the given maneuver will ensure that the CMG architecture is capable of performing the desired maneuver. The CMG parameters can be chosen in any such manner that produces the desired output torque. For this reason, Eq. 1.5 proves that there are many different possible CMG choices capable of producing the desired single-axis maneuver. Since the CMG parameters necessary to perform the desired maneuver are not unique, other considerations regarding final design choices must be used to specify the final CMG parameters. The rotor speed

and/or maximum gimbal rate can be varied depending upon the available motors and efficiencies as long as the ratio of the two remain constant. Additionally, choices that influence the rotor inertia such as material and geometry can be made to ensure that the specific CMG design can produce the required torque. These calculations are valid given a particular attitude change in a fixed amount of time. Therefore, the equations presented above will not provide sufficient answers for missions which involve more complicated maneuvers.

In order to properly size a CMG architecture for a given mission and/or range of missions, another technique is developed for optimizing the CMG size for a given mission. This thesis will use Monte-Carlo simulation techniques (Sec. 3.3) to size various CMG architectures for a given ensemble of mission profiles. The collection of missions used in the numerical simulations are developed to cover the entire spectrum of potential trajectories the MAJIC system could experience during a typical EVA or exploration activity.

1.2.2 Singularity Avoidance Steering Laws

There has been a significant amount of research into the steering laws associated with various CMG architectures. CMGs are prone to saturation and singularity problems, similar in nature to singular states associated with robotic linkages [8]. Saturation can be thought of as the maximum boundary of the momentum workspace for the CMG architecture. Fig. 1-6 shows a saturated state for a 3-link robotic manipulator. The robotic arm is at the the maximum extent of it's workspace and cannot reach beyond this point. CMGs encounter similar saturation effects, which are highly dependent upon the architecture. Saturation is considered an external singularity since the three-dimensional saturation surface forms the outer envelope of the CMG architecture's singular surface.

Internal singularities occur when the CMGs cannot produce a torque in a given direction. The mechanical analog of this happens when the links of the robotic arm are arranged such that their positioning prevents them from reaching a particular state. Fig. 1-7 demonstrates this by showing the links bent back upon each other

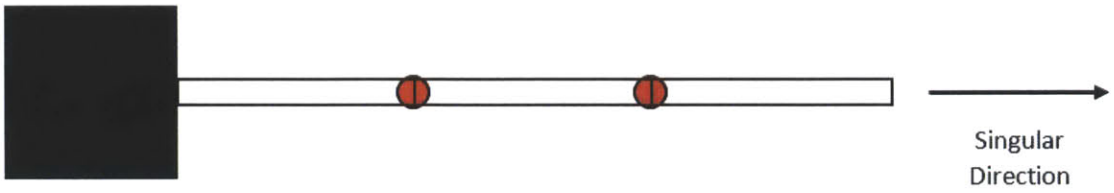


Figure 1-6: Robotic Manipulator - Saturation (External)

and preventing the manipulator from being able to move any further. Depending upon the number of CMGs on-board the spacecraft, internal singularities may or may not be escaped using null-motion. Null-motion is a singularity avoidance technique which refers to using the null-space of a matrix to generate additional solutions. This technique is similar to finding the homogeneous solution to an ordinary differential equation, which when paired with the particular solution, will provide the exact steering law solution. Null-motion is only possible when the number of CMGs is greater than 3. CMG saturation and singularities are discussed in further detail in Sec. 2.4.1.

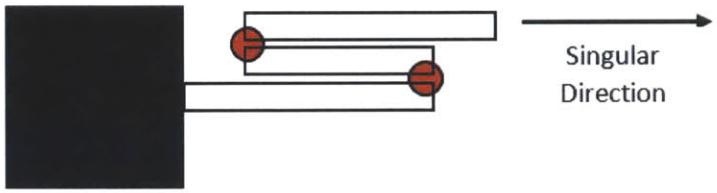


Figure 1-7: Robotic Manipulator - Singularity (Internal)

Singularity robust (SR) steering laws have been developed which avoid singular states by introducing output torque errors [9] [10]. The output torque errors are introduced by modifying the Jacobian using in the computation of the gimbal rate commands of the steering law (Sec. 2.4.3). This can be achieved by adding a slight disturbance to the requested output torque when the Jacobian approaches a singularity. These algorithms are typically effective at avoiding singularities; however, depending on the size of the disturbances added, they can have undesirable effects if a high-level of pointing precision is required.

Another technique that has been investigated for avoiding CMG singular states is determining a subset of initial gimbal angles based on pre-computed state data for a particular mission [11]. The main drawback of this technique is that the state data for the entire mission must be known ahead of time to properly integrate the dynamics backwards. This would be ideal for simplistic maneuvers of large-scale spacecraft such as the ISS. On the contrary, for a crew-member performing a precision EVA or exploring an unfamiliar location, this method would not be favorable. Of the two singularity avoidance strategies discussed above, the SR steering laws is preferred if small torque errors are acceptable when nearing singular states. The fact that the entire mission (e.g. torque and momentum states) must be known in advance removes the latter technique from consideration in the MAJIC ACS.

1.3 Novel Contributions

Throughout this thesis, several novel contributions have been developed for studying, analyzing, and optimizing blended ACS systems consisting of single-gimbal CMGs. The techniques are a combination of analytical, numerical and geometric algorithms primarily developed in MATLAB/Simulink r2013a. The following is a brief description of the techniques. More detailed information can be found throughout this thesis:

- Analytical technique for CMG sizing for spacecraft single-axis slew maneuvers
  - Works in conjunction with singular surface mapping software
- Analytical technique to geometrically design and optimize pyramidal CMG architecture
  - Specify relationship for maximum angular momentum values between axes to determine optimal skew angle(s)
- Complete singular surface mapping software that works with any CMG architecture
- Dynamic, real-time simulation environment for MAJIC system

  - Easily adaptable to any spacecraft with ACS containing RCS thrusters and/or CMGs
  - Ability to define relevant inertial and local coordinate frames
- Empirical sizing tool based on commercially available CMG specifications
- Interface between NASA's Engineering DOUG Graphics for Exploration (EDGE) and MATLAB/Simulink
  - Enables two-way communication between EDGE and MATLAB/Simulink to allow for triggering simulation events
  - Updated EDGE DCOMM source code for quick connection to MATLAB/Simulink
- Desaturation algorithm for RCS thrusters with user-defined saturation thresholds for entering and exiting desaturation mode
- Incorporated three-dimensional virtual reality helmet with active head and arm tracking

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 2

Spacecraft Dynamics

2.1 Coordinate Frame Conventions

Several coordinate frames will be discussed throughout the work, especially when dealing with simulation techniques and analyzing simulation results in Sec. 3.2.3. An inertial coordinate frame, denoted by $\mathcal{F}_{\mathrm{N}}$, is fixed in space. In an inertial frame of reference, the laws of mechanics are simplest. For instance, consider Newton's first law applied to an inertial frame of reference:

$$\mathbf{F} = m\mathbf{a} \tag{2.1}$$

Now, consider the same law applied to a non-inertial frame of reference (derived using the Transport theorem [12]):

$$\mathbf{F}' = m\left(\mathbf{a}_B + \frac{\mathrm{d}\boldsymbol{\Omega}}{\mathrm{dt}} \times \mathbf{r}_B + 2(\boldsymbol{\Omega} \times \mathbf{v}_B) + \boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \boldsymbol{r}_B)\right) \tag{2.2}$$

The preceding example shows that in an inertial frame, the laws of mechanics are their simplest. It also shows that different choices of local coordinate frames may greatly impact the complexity of the resulting equations. Local coordinate frames include the spacecraft's body frame $\mathcal{F}_{\mathrm{B}}$, the CMG-fixed frame $\mathcal{F}_{\mathrm{CMG}}$, and any frame introduced by adding additional hardware and/or sensors $\mathcal{F}_{\mathrm{sensor}}$. Fig. 2-1 shows the

typical coordinate frames for a spacecraft application including the inertial, body, and CMG gimbal frames. There could be several CMG gimbal frames depending on how many CMGs are used on-board the spacecraft.

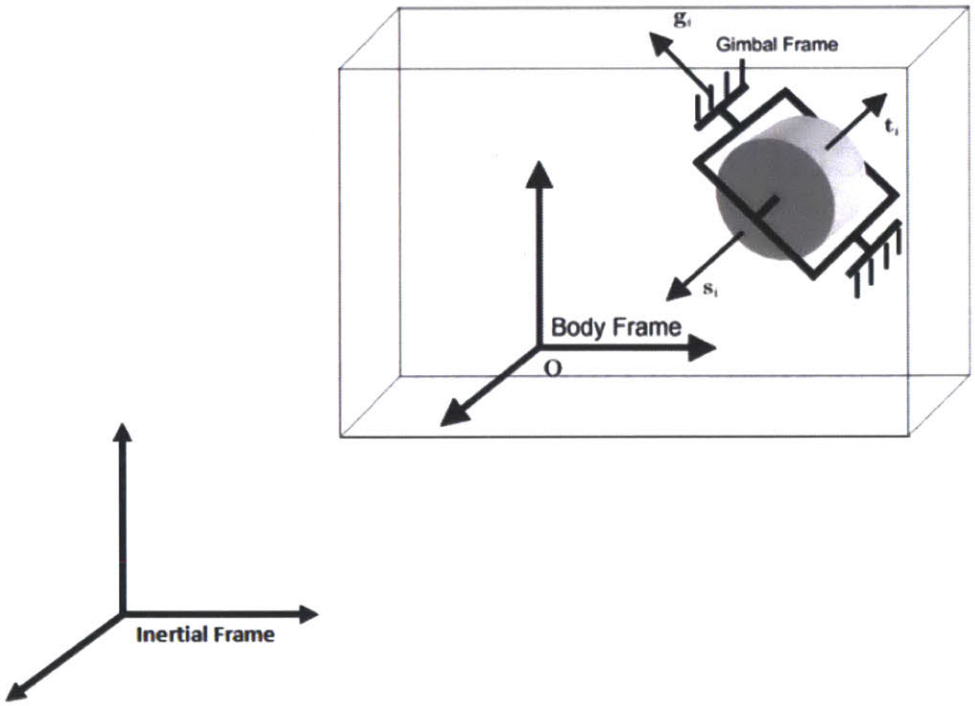


Figure 2-1: Relative Frames of Reference [1]

The local coordinate frame for a CMG is depicted in Fig. 2-2. The CMG-fixed coordinate frame uses a right-handed convention (as all coordinate frames in this thesis) and has three orthogonal basis vectors: g_i, s_i, t_i. The indices i denote that each CMG has it's own local coordinate frame. Therefore, if there are a total of N CMGs on-board the spacecraft, then $i \in \{1, 2, \ldots, N\}$. The meaning of each of the basis vectors is explained below.

- $\mathbf{g}_i \rightarrow$ gimbal axis (fixed in the spacecraft body coordinate frame $\mathcal{F}_\mathrm{B}$)
- $\mathbf{s}_i \rightarrow$ spin axis
- $\mathbf{t}_i \rightarrow$ transverse axis (torque axis). Given by $\mathbf{t}_i = -\mathbf{s}_i \times \mathbf{g}_i$

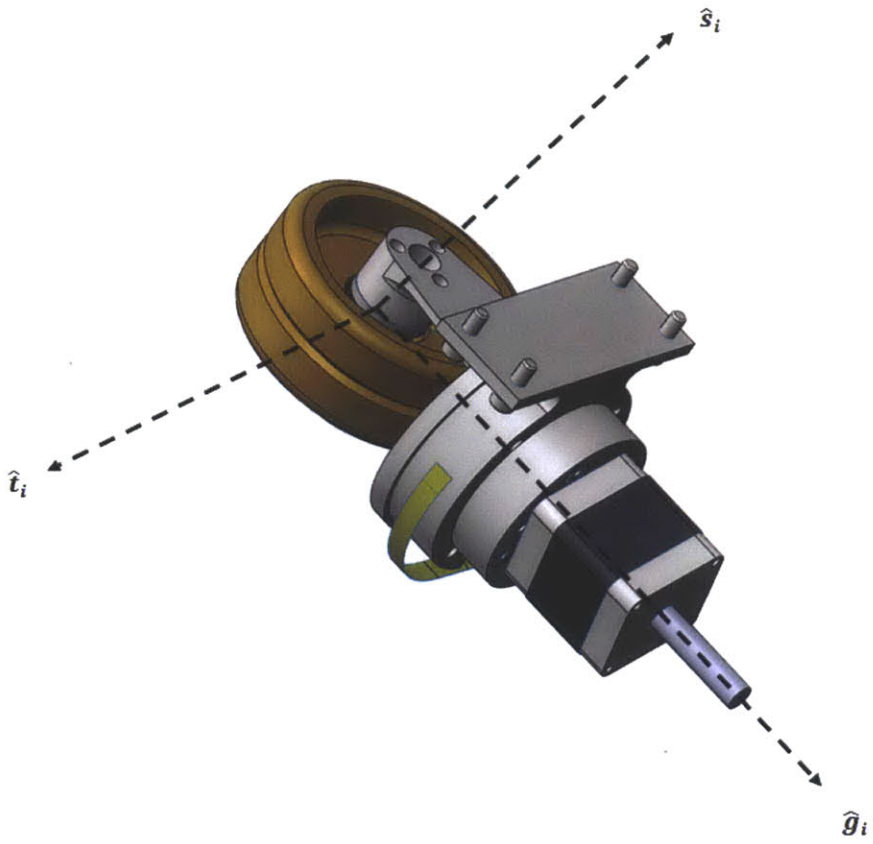


Figure 2-2: CMG Local Coordinate Frame

The rotation of the gimbal motor, and the condition that the gimbal axis ($\mathbf{g}_i$) of i-th CMG remains fixed in the spacecraft frame of reference, the spin and transverse axis are allowed to rotate. The rotation of the spin axis introduces a gyroscopic torque that acts on the CMG and spacecraft in equal and opposite magnitude and direction. The gyroscopic torque is induced because of the constant angular momentum of the flywheel spinning about the spin axis. Fig. 2-3 shows the gyroscopic torque, $\Delta\tau_i$, induced on the CMG by gimbaling the CMG through an angle δ. The gyroscopic torque experienced by the spacecraft would be in the opposite direction and have the same magnitude ($\Delta\tau_{S/C}$). Attitude representations are covered in Sec. 2.2.

Since CMGs generate torque by rotating the spin and transverse axis, no external torques are introduced to the system. The fact that no external torques are applied to the system means that a spacecraft operating with CMGs as attitude-control actuators obeys the law of the conservation of angular momentum:

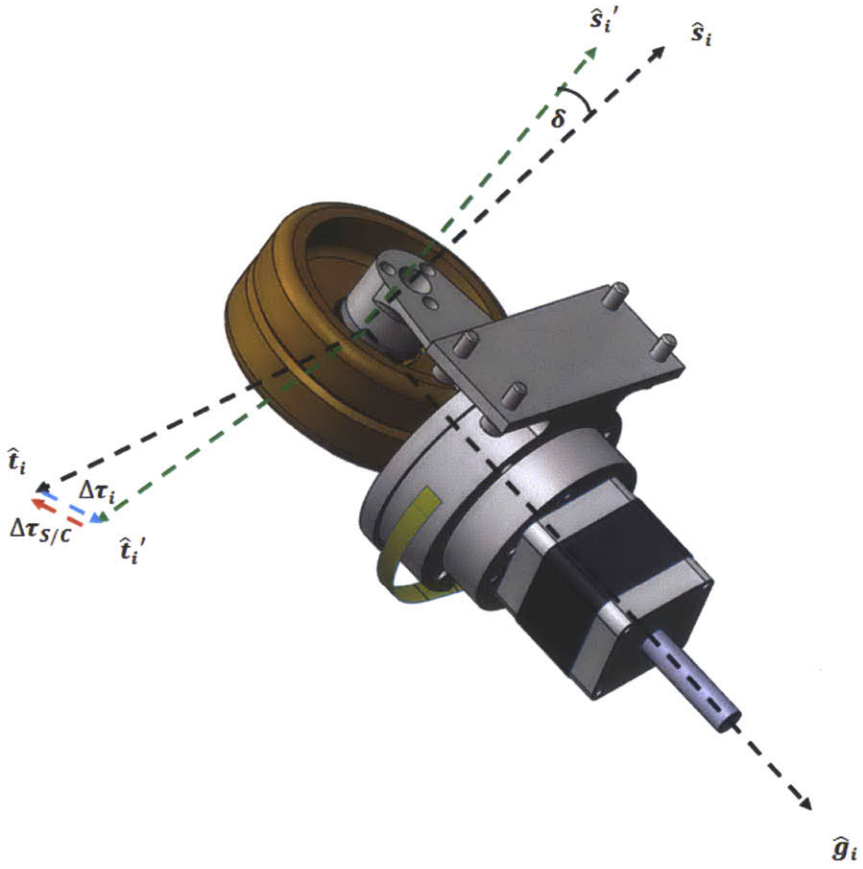


Figure 2-3: CMG Rotated Coordinate Frame

$$\sum \boldsymbol{\tau}_{\text{ext}} = \frac{\mathrm{d}\mathbf{H}}{\mathrm{dt}} = 0 \tag{2.3}$$

where $\boldsymbol{\tau}_{\text{ext}}$ represents the external torque acting on the system and $\mathbf{H}$ represents the system's angular momentum. The idea of the conservation of angular momentum will be revisited in Sec. 2.3.

2.2 Angle Representation

In order to characterize the attitude, or three-dimensional orientation of a spacecraft, a set of parameters must be given. There are several different formalisms for attitude representation in three-dimensions, such as Euler angles, Modified Rodrigues parameters (MRP), quaternions and direction cosine matrices (DCM). Euler angles and

quaternions will be discussed further in sections 2.2.1 and 2.2.2, respectively. There exist transformations between each of the formalisms such that given a spacecraft attitude defined by Euler angles and a rotation sequence, it is possible to determine the corresponding DCM or quaternion representation. The strengths and weaknesses of each formalism are discussed in the upcoming subsections.

2.2.1 Euler

In most spacecraft and aircraft notations, Euler angles are a set of three rotation angles and a particular sequence which define the relative orientation of one reference frame to another. In general, Euler angles describe the attitude of one reference frame relative to another through three successive rotation angles (θ_1, θ_2, θ_3) [12]. A rotation about the third body axis by an angle θ_1, the second body axis by an angle θ_2, and the first body axis by an angle θ_3 can be described as a 3-2-1 or Z-Y-X rotation. A 1-2-3 rotation through the same rotation angles will not yield the same orientation; therefore, the order of the axes about which the reference frame is rotated is important here. The subscript on the rotation angles do not refer to which axis the rotation occurs about, but the order in which it occurs. To avoid uncertainty when defining an Euler sequence, the set of three Euler angles will be referred to as pitch, yaw, roll (θ, ψ, ϕ).

The rotation sequence that will be used throughout the simulation is a pitch, yaw, roll (2-3-1) sequence. The rotation matrix which describes the orientation of the spacecraft body frame $\mathcal{F}_\mathrm{B}$ relative to the inertial frame $\mathcal{F}_\mathrm{N}$ can be found by composing the three rotations

$$R = R_1(\phi)R_3(\psi)R_2(\theta) \tag{2.4}$$

where R_2, R_3, R_1 are the pitch, yaw, and roll rotation matrices ($R \in \mathbb{R}^{3\times3}$). Since matrix multiplication is typically not commutative, the order of the rotation matrices is indeed important. The 2-3-1 rotation sequence is an extrinsic rotation since it occurs about a fixed axis, $\mathcal{F}_\mathrm{N}$. An intrinsic rotation occurs about the axes of rotation coordinate system which is initially aligned with the fixed one. A composite rotation

matrix R can be found symbolically by multiplying the DCMs associated with the specified rotation. The DCM associated with each elementary rotation operation is presented below, parametrized by θ:

$$R_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \tag{2.5}$$

$$R_2 = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \tag{2.6}$$

$$R_3 = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.7}$$

Referencing Eq. 2.4, the composite rotation matrix can be written as

$$R_{231}(\theta,\psi,\phi) = \begin{bmatrix} \cos\psi\cos\theta & \sin\phi & -\cos\psi\sin\theta \\ -\cos\phi\sin\psi\cos\theta + \sin\phi\sin\theta & \cos\phi\cos\psi & \cos\phi\sin\psi\sin\theta + \sin\phi\cos\theta \\ \sin\phi\sin\psi\cos\theta + \cos\phi\sin\theta & -\sin\phi\cos\psi & -\sin\phi\sin\psi\sin\theta + \cos\phi\cos\theta \end{bmatrix} \tag{2.8}$$

The inverse transformations from the rotation matrix R_{231} to the Euler angles (θ, ψ, ϕ) are

$$\begin{aligned} \theta &= \arctan\left(-\frac{R(1,3)}{R(1,1)}\right) \\ \psi &= \arcsin\left(-R(1,2)\right) \\ \phi &= \arctan\left(-\frac{R(3,2)}{R(2,2)}\right) \end{aligned} \tag{2.9}$$

The following diagram illustrates a 2-3-1 Euler angle rotation sequence in sequential steps with $\theta = \psi = \phi = 30°$.

Euler angles are a very clear and concise way to deal with three-dimensional rotation and attitude orientation. However, depending upon the sequence of rotation,

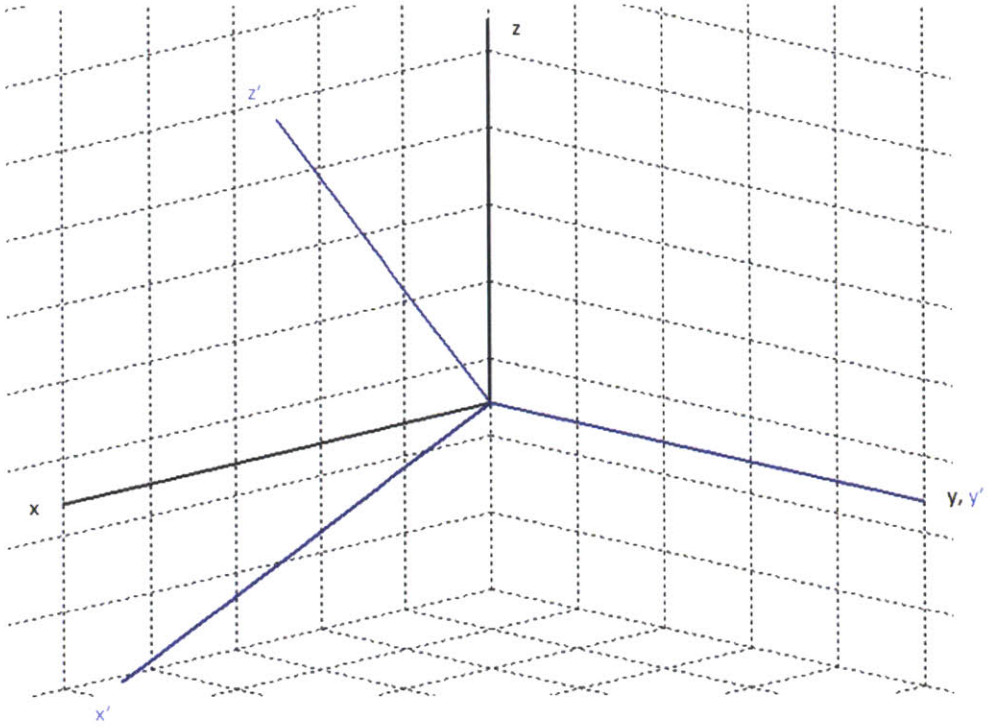


Figure 2-4: 2-3-1 Sequence: Rotation about Y

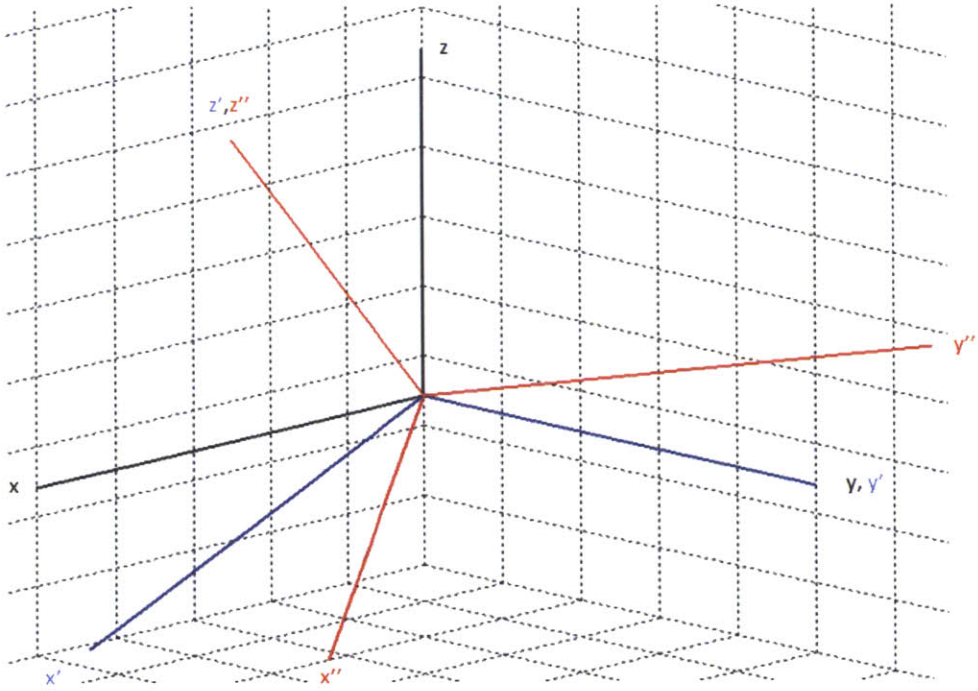


Figure 2-5: 2-3-1 Sequence: Rotation about Z′

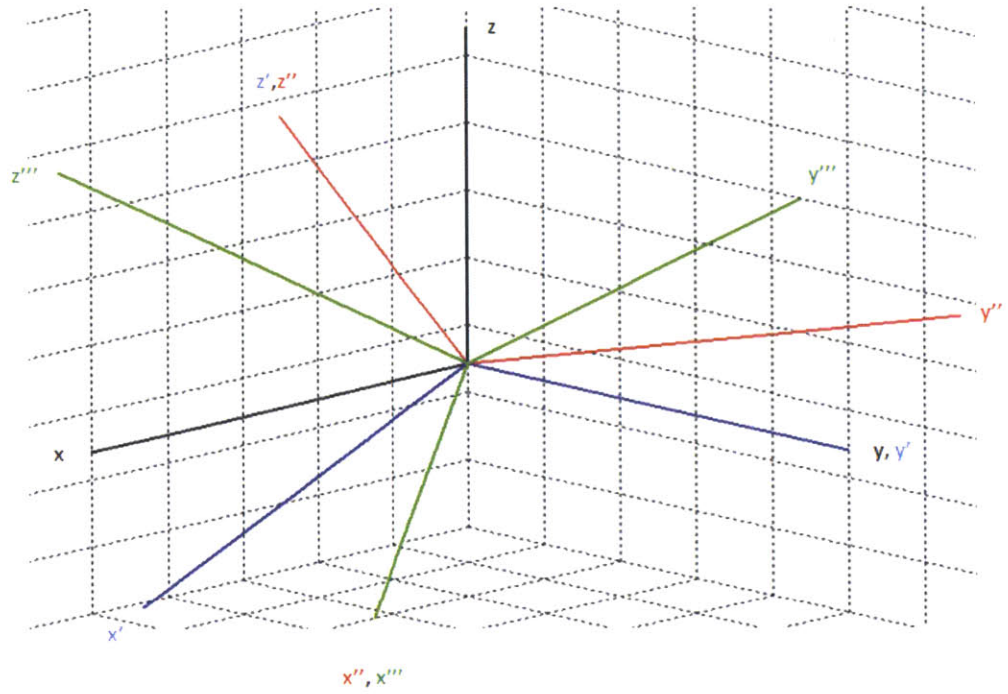


Figure 2-6: 2-3-1 Sequence: Rotation about X″

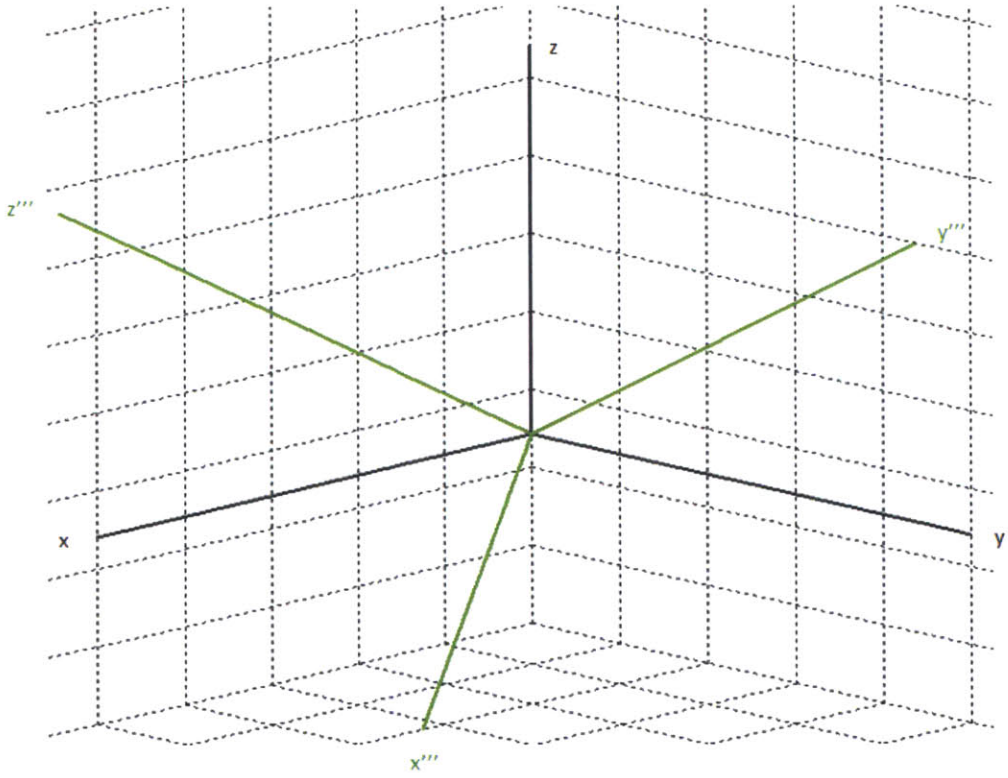


Figure 2-7: 2-3-1 Sequence: Initial and Final Reference Frames

numerical instabilities and singularities can arise. The singularities can be thought of as a situation in which two rotation angles are not uniquely defined [13]. For the 2-3-1 rotation sequence, a singularity occurs when $\psi = \pm\frac{\pi}{2}$. This implies that if $\psi = \pm\frac{\pi}{2}$, the rotational sequence is no longer unique, since interchanging θ and ϕ will give the same resulting transformation. This can be shown through a simple example. The rotational sequence is once again 2-3-1 and the rotation angles are $\theta = \frac{\pi}{8}$, $\psi = \frac{\pi}{2}$, and $\phi = \frac{\pi}{4}$:

$$R_1\left(\frac{\pi}{4}\right)R_3\left(\frac{\pi}{2}\right)R_2\left(\frac{\pi}{8}\right) = \begin{bmatrix} 0 & 1 & 0 \\ -0.3827 & 0 & 0.9239 \\ 0.9239 & 0 & 0.3827 \end{bmatrix} \tag{2.10}$$

$$R_1\left(\frac{\pi}{8}\right)R_3\left(\frac{\pi}{2}\right)R_2\left(\frac{\pi}{4}\right) = \begin{bmatrix} 0 & 1 & 0 \\ -0.3827 & 0 & 0.9239 \\ 0.9239 & 0 & 0.3827 \end{bmatrix} \tag{2.11}$$

The sum of ψ and ϕ is however unique. Different values of $\psi + \phi$ will result in different transformations. In order to avoid the problem of singularities associated with Euler angles, quaternions are commonly used to represent attitude in aeronautical and astronautical situations.

2.2.2 Quaternion

Quaternions[1] can be thought of as: a four-dimensional vector, a scalar plus a vector with three components, or a complex number with three imaginary parts[2]. An example quaternion is shown below:

$$\check{q} = (r, \mathbf{v}) = a1 + bi + cj + dk \tag{2.12}$$

The basis elements of a quaternion are defined by 1, i, j, k. When multiplying basis elements together, multiplying by 1 does not change basis elements. Multiplying the other basis elements together results in different elements. Some basic multiplication

[1] represented with an overhead $\check{}$

[2] The second definition will be used throughout

of basis elements have been provided

$$i^2 = j^2 = k^2 = ijk = -1 \tag{2.13}$$

$$\begin{aligned} ij &= k \\ jk &= i \\ ki &= j \end{aligned} \tag{2.14}$$

Any other basis element combination can be found by manipulating Eq. 2.13. Because of the scalar plus vector notation adopted in this work, one can prove that quaternion multiplication (like matrix multiplication) is not commutative. For instance, consider multiplying two quaternions $\check{a} = (a, \mathbf{a})$ and $\check{b} = (b, \mathbf{b})$, where a represents the real component of the quaternion and $\mathbf{a}$ represents the vector components of the quaternion. The resulting Hamilton product is shown below:

$$\check{c} = (ab - \mathbf{a} \cdot \mathbf{b}, a\mathbf{b} + b\mathbf{a} + \mathbf{a} \times \mathbf{b}) \tag{2.15}$$

The presence of the cross product in the resulting vector component shows that quaternion multiplication is not commutative. The negative sign associated with the dot product in the real component of c is due to the multiplication of the basis elements derived in Eq. 2.13. One can define the conjugate of a quaternion $q = (q, \mathbf{q})$ as follows

$$\check{q}^\star = (q, -\mathbf{q}) \tag{2.16}$$

The product of a quaternion ($\check{q} = (q, \mathbf{q})$) and its conjugate ($\check{q}^\star$) can be used to find the inverse of a quaternion:

$$\check{q}\check{q}^\star = qq + \mathbf{q}\mathbf{q} = \check{q} \cdot \check{q} \tag{2.17}$$

defining the above quantity as the dot-product of two quaternions $(\check{q} \cdot \check{q})$, the inverse can be written as such

$$\check{q}^{-1} = \frac{\check{q}^{\star}}{\check{q} \cdot \check{q}} \tag{2.18}$$

The norm of a quaternion is the product of the quaternion and its conjugate which results in a Euclidean norm. A quaternion can be unitized (norm set equal to one) by dividing the quaternion by its norm. Since the norm of a quaternion is a scalar, the unit norm is simply the individual components of the quaternion divided by its norm. Consider the case where $\check{q} = (a, b\imath + c\jmath + dk)$ is an arbitrary quaternion, the unit quaternion can be found as such

$$\check{q} = \frac{\check{q}}{||\check{q}||} = \frac{(a, b\imath + c\jmath + dk)}{\sqrt{a^2 + b^2 + c^2 + d^2}} \tag{2.19}$$

A unit quaternion can be used to rotate a vector through the transformation

$$\mathbf{r}' = \check{q}\mathbf{r}\check{q}^{\star} \tag{2.20}$$

A vector can be thought of as a purely imaginary quaternion (i.e. $\check{r} = (0, \mathbf{r})$) so the above equation can be re-written as

$$\check{r}' = \check{q}\check{r}\check{q}^{\star} \tag{2.21}$$

After computing the two Hamilton products [14], the resulting quaternion is

$$\check{r}' = \Big(q(\mathbf{r} \cdot \mathbf{q}) - \mathbf{q} \cdot (q\mathbf{r} + \mathbf{q} \times \mathbf{r}), q(q\mathbf{r} + \mathbf{q} \times \mathbf{r}) + (\mathbf{r} \cdot \mathbf{q})\mathbf{q} + \mathbf{q} \times (q\mathbf{r} + \mathbf{q} \times \mathbf{r})\Big) \tag{2.22}$$

which after some simplification yields

$$\check{r}' = \Big(0, (q^2 - \mathbf{q} \cdot \mathbf{q})\mathbf{r} + 2q(\mathbf{q} \times \mathbf{r}) + 2(\mathbf{q} \cdot \mathbf{r})\mathbf{q}\Big) \tag{2.23}$$

This result shows that after the rotation of $\check{r}$, a purely imaginary quaternion, another purely imaginary quaternion, $\check{r}'$, is recovered. It can be shown that the rotation in-

duced by the quaternion is equivalent to rotating the vector $\mathbf{r}$ through an angle θ about the axis $\hat{\boldsymbol{\omega}}$. The following relationships can be made between the unit quaternion and the axis-angle rotation induced by the transformation shown above:

$$\begin{aligned} q &= \pm\cos\left(\frac{\theta}{2}\right) \\ \mathbf{q} &= \pm\sin\left(\frac{\theta}{2}\right)\hat{\boldsymbol{\omega}} \end{aligned} \tag{2.24}$$

The quaternion kinematic differential equations (shown below in Eq. 2.25), which can be integrated forward in time, can be used to express an object's orientation in three space without the singularities accompanying Euler angles.

$$\begin{aligned} \dot{q} &= -\frac{1}{2}\boldsymbol{\omega}^{\top}\mathbf{q} \\ \dot{\mathbf{q}} &= \frac{1}{2}(q\boldsymbol{\omega} - \boldsymbol{\omega}\times\mathbf{q}) \end{aligned} \tag{2.25}$$

2.3 Overall System Dynamics

As previously mentioned in Sec. 2.1, the dynamics of a spacecraft containing N CMGs obeys the law of the conservation of angular momentum. This will simplify the kinematic and dynamic equations of motion for the composite system. The total inertia of the combined system (spacecraft and N CMGs) can be written as [15]:

$$\bar{J}_{S/C} = \bar{J} + m_i(\mathbf{r}_i^{\top}\mathbf{r}_i\bar{1} - \mathbf{r}_i\mathbf{r}_i^{\top}) + \sum_{i=1}^{N}\bar{I}_{gw} \tag{2.26}$$

where $\bar{J} \in \mathbb{R}^{3\times 3}$ is the inertia of the spacecraft and CMG architecture about the center of mass (COM), $m_i(\mathbf{r}_i^{\top}\mathbf{r}_i\bar{1} - \mathbf{r}_i\mathbf{r}_i^{\top})$ are the contributions due to the parallel-axis theorem, and $\sum_{i=1}^{N}\bar{I}_{gw}$ are the time-varying inertia contributions of the gimbal and flywheel. The total angular momentum of the system is expressed below:

$$\mathbf{H} = \bar{J}_{S/C}\boldsymbol{\omega} + \sum_{i=1}^{N}\mathbf{h}_i \tag{2.27}$$

The angular momentum contribution from the i-th CMG (expressed in the i-th CMG reference frame with basis vectors $[\mathbf{s}_i, \mathbf{t}_i, \mathbf{g}_i]$, $\mathcal{F}_{\text{CMG}}$) is

$$\mathbf{h}_i = \begin{bmatrix} \bar{I}_{\text{fly}}\Omega \\ 0 \\ \bar{I}_{gw}\dot{\delta}_i \end{bmatrix} \tag{2.28}$$

where $\bar{I}_{\text{fly}}$ is the flywheel inertia, Ω is the flywheel spin speed, $\bar{I}_{gw}$ is the combined inertia of the gimbal and flyhweel and $\dot{\delta}_i$ is the gimbal spin speed for the i-th CMG.

Differentiation of Eq. 2.27 gives the time rate of change of the angular momentum of the system, which is known to equal zero since the spacecraft and CMG composite system obey the law of the conservation of angular momentum:

$$0 = \frac{\mathrm{d}}{\mathrm{dt}}\left(\bar{J}_{S/C}\boldsymbol{\omega} + \sum_{i=1}^{N}\mathbf{h}_i\right) + \boldsymbol{\omega} \times \left(\bar{J}_{S/C}\boldsymbol{\omega} + \sum_{i=1}^{N}\mathbf{h}_i\right) \tag{2.29}$$

The equation above was derived using the Transport theorem for analytical mechanics. The individual time rate of change terms are derived below keeping in mind that $\bar{J}_{S/C}$ and $\mathbf{h}$ are functions of flywheel speed $\boldsymbol{\Omega} \in \mathbb{R}^{N\times 1}$, gimbal angle $\boldsymbol{\delta} \in \mathbb{R}^{N\times 1}$ and gimbal rate $\dot{\boldsymbol{\delta}} \in \mathbb{R}^{N\times 1}$.

$$\frac{\overset{B}{\mathrm{d}}(\bar{J}_{S/C}\boldsymbol{\omega})}{\mathrm{dt}} = \bar{J}_{S/C}\dot{\boldsymbol{\omega}} + \frac{\partial(\bar{J}_{S/C}\boldsymbol{\omega})}{\partial\boldsymbol{\delta}}\frac{\mathrm{d}\boldsymbol{\delta}}{\mathrm{dt}}\boldsymbol{\omega} \tag{2.30}$$

$$\frac{\overset{B}{\mathrm{d}}\mathbf{h}}{\mathrm{dt}} = \frac{\partial\mathbf{h}}{\partial\boldsymbol{\delta}}\frac{\mathrm{d}\boldsymbol{\delta}}{\mathrm{dt}} + \frac{\partial\mathbf{h}}{\partial\dot{\boldsymbol{\delta}}}\frac{\mathrm{d}\dot{\boldsymbol{\delta}}}{\mathrm{dt}} + \frac{\partial\mathbf{h}}{\partial\boldsymbol{\Omega}}\frac{\mathrm{d}\boldsymbol{\Omega}}{\mathrm{dt}} \tag{2.31}$$

Because only single-gimbal CMGs (SGCMG) are considered with constant flywheel speed ($\dot{\boldsymbol{\Omega}} = 0$), Eq. 2.31 can be simplified to

$$\frac{\mathrm{d}\mathbf{h}}{\mathrm{dt}} = \frac{\partial\mathbf{h}}{\partial\boldsymbol{\delta}}\frac{\mathrm{d}\boldsymbol{\delta}}{\mathrm{dt}} + \frac{\partial\mathbf{h}}{\partial\dot{\boldsymbol{\delta}}}\frac{\mathrm{d}\dot{\boldsymbol{\delta}}}{\mathrm{dt}} \tag{2.32}$$

Eq. 2.29 can be re-written, substituting in the expressions derived above, as:

$$0 = \bar{J}_{S/C}\dot{\boldsymbol{\omega}} + \dot{\bar{J}}_{S/C}\boldsymbol{\omega} + \dot{\mathbf{h}} + \boldsymbol{\omega}^{\times}\bar{J}_{S/C}\boldsymbol{\omega} + \boldsymbol{\omega}^{\times}\mathbf{h} \tag{2.33}$$

where $\boldsymbol{\omega}^{\times}$ indicates a skew-symmetric matrix. A Skew-symmetric matrix can replace a vector cross-product by a matrix multiplication. The general form a skew-symmetric matrix derived from a three-dimensional vector is given below:

$$\boldsymbol{\omega}^{\times} = \begin{bmatrix} 0 & -\boldsymbol{\omega}(3) & \boldsymbol{\omega}(2) \\ \boldsymbol{\omega}(3) & 0 & -\boldsymbol{\omega}(1) \\ -\boldsymbol{\omega}(2) & \boldsymbol{\omega}(1) & 0 \end{bmatrix} \tag{2.34}$$

By grouping terms that are dependent upon CMG parameters, one can simplify the above expression by introducing an internal control torque ($\boldsymbol{\tau}$) such that Eq. 2.33 becomes:

$$\boldsymbol{\tau} = \bar{J}_{S/C}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega}^{\times}\bar{J}_{S/C}\boldsymbol{\omega} \tag{2.35}$$

where $\boldsymbol{\tau}$ is defined as follows

$$\boldsymbol{\tau} = -\dot{\bar{J}}_{S/C}\boldsymbol{\omega} - \dot{\mathbf{h}} - \boldsymbol{\omega}^{\times}\mathbf{h} \tag{2.36}$$

The problem can be further simplified by assuming the contributions from the gimbal-wheel is insignificant. The equations then simplify down to

$$\begin{aligned} \boldsymbol{\tau} &= -(\cancelto{0}{\dot{\bar{J}}_{S/C}\boldsymbol{\omega}} + \dot{\mathbf{h}}) - \boldsymbol{\omega}^{\times}\mathbf{h} \\ \boldsymbol{\tau} &= -\frac{\partial \mathbf{h}}{\partial \boldsymbol{\delta}}\frac{\mathrm{d}\boldsymbol{\delta}}{\mathrm{dt}} - \boldsymbol{\omega}^{\times}\mathbf{h} \\ \frac{\partial \mathbf{h}}{\partial \boldsymbol{\delta}}\dot{\boldsymbol{\delta}} &= -\boldsymbol{\tau} - \boldsymbol{\omega}^{\times}\mathbf{h} \end{aligned} \tag{2.37}$$

The internal control torque, provided by the CMGs, can be found by solving Eq. 2.37 for $\dot{\boldsymbol{\delta}}$. The solution to Eq. 2.37 is shown below

$$\dot{\boldsymbol{\delta}} = -\left[\frac{\partial \mathbf{h}}{\partial \boldsymbol{\delta}}\right]^{-1}(\boldsymbol{\tau} + \boldsymbol{\omega}^{\times}\mathbf{h}) \tag{2.38}$$

It is important to note that solution derived above involves inverting a Jacobian

matrix, $\bar{J} = \dfrac{\partial \mathbf{h}}{\partial \boldsymbol{\delta}} \in \mathbb{R}^{3\times N}$. Since the Jacobian matrix will most likely not be square (i.e. $N \neq 3$), then a traditional matrix inverse cannot be computed. Because of this, the CMG steering laws presented in Sec. 2.4.3 are based on pseudoinverses. Also, if the Jacobian is singular and an inverse cannot be computed, the above equation for the commanded control torque $\boldsymbol{\tau}$ cannot be evaluated either. A singular Jacobian implies that the CMGs are at a singular gimbal orientation. CMG singularities are reviewed in Sec. 2.4.1.

2.4 Attitude Control System Design

Attitude control system (ACS) design involves choosing a particular set of actuators and defining the input-output behavior to achieve the desired attitude results. For instance, a spacecraft equipped with only RCS thrusters would have a different ACS than a system with both RCS thrusters and CMGs. In order to determine the input-output behavior of the spacecraft, the overall system dynamics are derived in Sec. 2.3 with the actuators dynamics included. The steering laws for the actuators are then derived which output command torques to achieve the proper pointing requirements.

2.4.1 CMG Singularities

ACS that use CMGs as actuators must be aware of singular gimbal configurations which occurs when the Jacobian loses rank, (i.e. $\text{rank}\left(\dfrac{\partial \mathbf{h}}{\partial \boldsymbol{\delta}}\right) < 3$). When the rank of the Jacobian becomes singular, the CMG architecture is no longer able to produce a torque in a given direction. This can be visualized by considering the case when the output torques for all N CMGs are constrained to a plane ($\text{rank}(\bar{J}) = 2$). Fig. 2-8 represents a situation in which a CMG architecture has entered a singularity since the CMG architecture is no longer able to project torque along a specified direction. In a singular state ($\text{rank}(\bar{J}) = 2$), all the unit vectors $\mathbf{t}_i$ lay in the same plane (i.e. the torque output can only span two dimensions). Defining a unit vector $\mathbf{u}$ normal to this plane, one can express the geometric relationship as shown below

$$\mathbf{u} \cdot \mathbf{t}_i = 0 \quad \forall \quad i = 1, \ldots, N \tag{2.39}$$

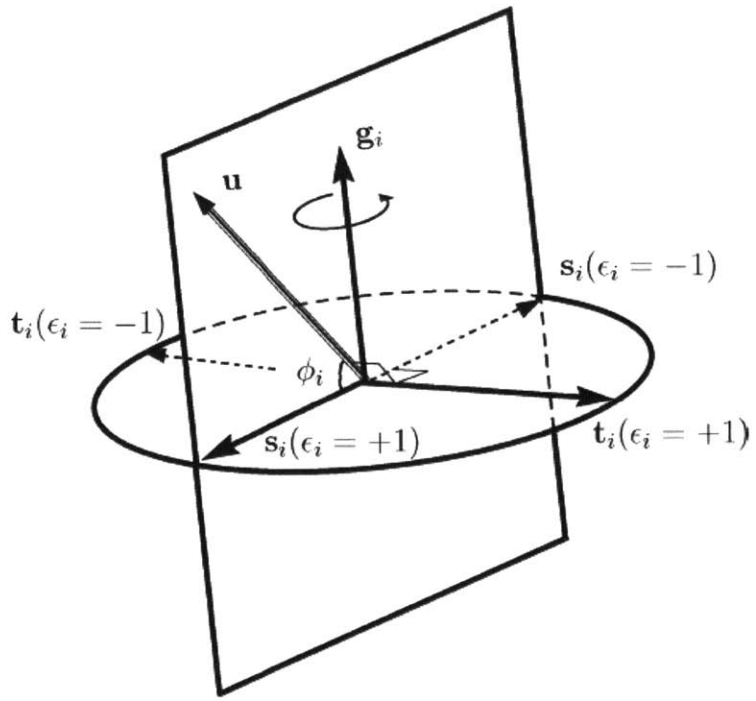


Figure 2-8: CMG Singularity Diagram [1]

Also, $\mathbf{t}_i$ is normal to $\mathbf{g}_i$ by definition of the CMGs orthogonal coordinate frame. Fig. 2-8 gives some geometrical insight into what happens during a singularity: $\mathbf{t}_i$ is normal to the plane spanned by $\mathbf{u}$ and $\mathbf{g}_i$. As a result, the projection of $\mathbf{s}_i$ onto the singular direction $\mathbf{u}$ is either maximum or minimum. The two possibilities can be expressed mathematically as shown below:

$$\begin{gathered} \mathbf{u} \cdot \mathbf{t}_i = 0 \quad \mathbf{u} \cdot \mathbf{s}_i > 0 \\ \\ \mathbf{u} \cdot \mathbf{t}_i = 0 \quad \mathbf{u} \cdot \mathbf{s}_i < 0 \end{gathered} \tag{2.40}$$

One can define $\epsilon_i \triangleq \text{sign}(\mathbf{u} \cdot \mathbf{s}_i)$ to represent whether or not the angular momentum projection is at a maximum ($\epsilon_i = +1$) or minimum ($\epsilon_i = -1$). Using the principle of a singular direction ($\mathbf{u}$), one can express the torque and spin axes as

$$\mathbf{t}_i = \epsilon_i \frac{\mathbf{g}_i \times \mathbf{u}}{|\mathbf{g}_i \times \mathbf{u}|} \quad \forall \quad i = 1, \ldots, N \tag{2.41}$$

$$\mathbf{s}_i = \epsilon_i \frac{(\mathbf{g}_i \times \mathbf{u}) \times \mathbf{g}_i}{|\mathbf{g}_i \times \mathbf{u}|} \quad \forall \quad i = 1, \ldots, N \tag{2.42}$$

The total angular momentum at a particular singular state can be found by summing the individual contributions to the overall angular momentum of the CMG architecture. The total angular momentum ($\mathbf{H}$) is computed in the following manner:

$$\mathbf{H} = \sum_{i=1}^{N} \epsilon_i \frac{(\mathbf{g}_i \times \mathbf{u}) \times \mathbf{g}_i}{|\mathbf{g}_i \times \mathbf{u}|} \tag{2.43}$$

The shapes of the singular surfaces depends on the orientation of the gimbal axes, as well as the sign of ϵ_i. Given a particular set of ϵ values ($\epsilon_i = \pm 1 \quad \forall \quad i = 1, 2, \ldots, N$), one can compute the singular surfaces of the CMG architecture by running a Monte-Carlo simulation. The simulation generates unit-magnitude singular direction vectors $\mathbf{u}$ at random and computes the total angular momentum at each of the singular states. A point-cloud map of the singular surface over the interval ϵ_i is shown in Figs. 2-9 - 2-12 for two pyramidal CMG architectures ($\beta = 54.7°$ and $90°$). The complete singular surface for a particular CMG architecture can be found by superimposing the results from all possible ϵ_i profiles. The total number of ϵ profiles can be found by permuting the valid ϵ values for the number of CMGs present in the given architecture. For instance, for a pyramidal architecture ($N = 4$), the number of valid permutations is $2^4 = 16$. Although due to the symmetry of certain architectures, not all permutations may be necessary to compute the entire singular surface. Because of the symmetric nature of the pyramidal architecture, the complete singular surface can be generated by setting $\epsilon_i = +1 \quad \forall \quad i = 1, \ldots, N$, which is shown in Fig. 2-9, and by modifying the ϵ-parameter such that one CMG has a minimal projection and the rest have maximal projections (i.e. $\epsilon_1 = \epsilon_2 = \epsilon_4 = +1, \epsilon_3 = -1$), which is shown in Fig. 2-10.

The maximum angular momentum envelope appears to have four holes on the surface of the point cloud. However, the four holes connect smoothly to the the

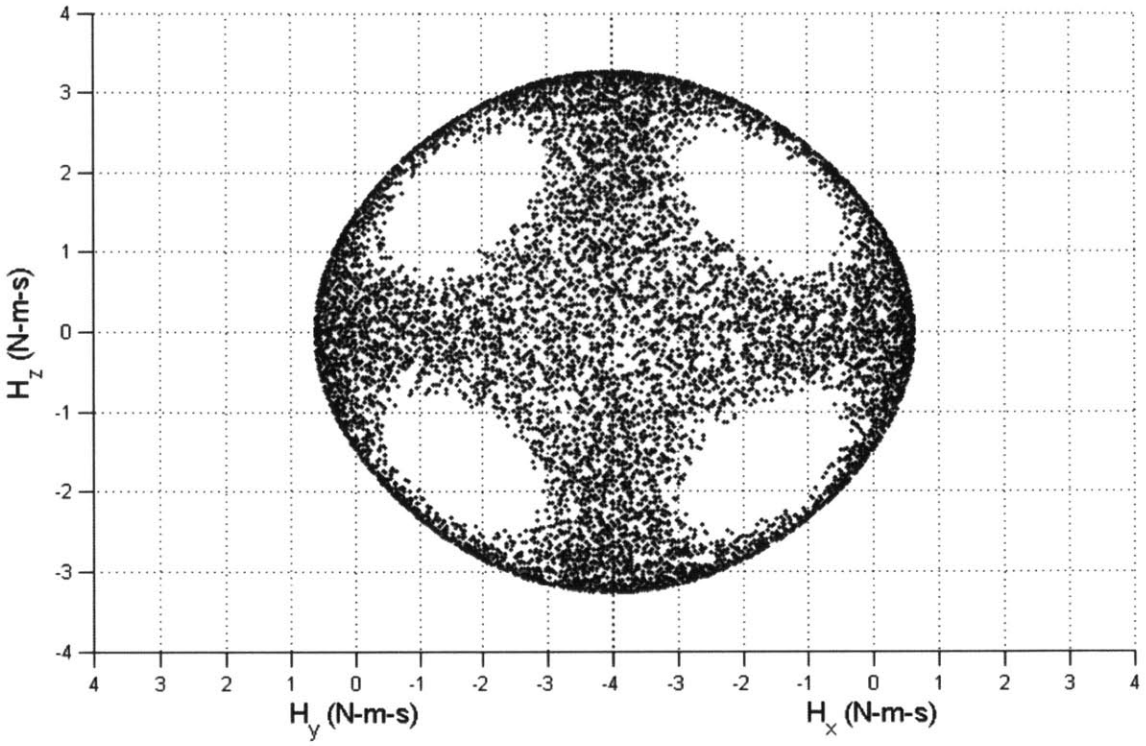


Figure 2-9: External Singular Surface Envelope $\beta = 54.7°$ ($\epsilon_i = +1 \quad \forall \quad i = 1, \ldots, N$)

trumpet-like singular surfaces generated by the internal singularities of the pyramidal architecture. Fig. 2-11 shows the superposition of the two types of singularities. Singularities can be classified as either external (maximum angular momentum projections) or internal (any singularity that resides within the external boundaries). Internal singularities can be further classified as hyperbolic or elliptic [16]. Hyperbolic internal singularities are singularities which can be escaped through null motion, meanwhile elliptic singularities cannot. Fig. 2-12 shows the singular surfaces of the pyramid configuration with $\beta = 90°$.

From Fig. 2-12, it is immediately apparent that the singular surfaces of the architecture with $\beta = 90°$ are much simpler than with $\beta = 54.7°$. However, the maximum angular momentum envelope for the latter configuration is much more symmetric and can therefore provide a more uniform torque distribution for a given CMG size. This result is summarized in Tables 2.1 and 2.2

Because of the extra degree-of-freedom (DoF) in this particular architecture ($N = 4$), additional solutions, which do not alter the torque output of the CMGs, can

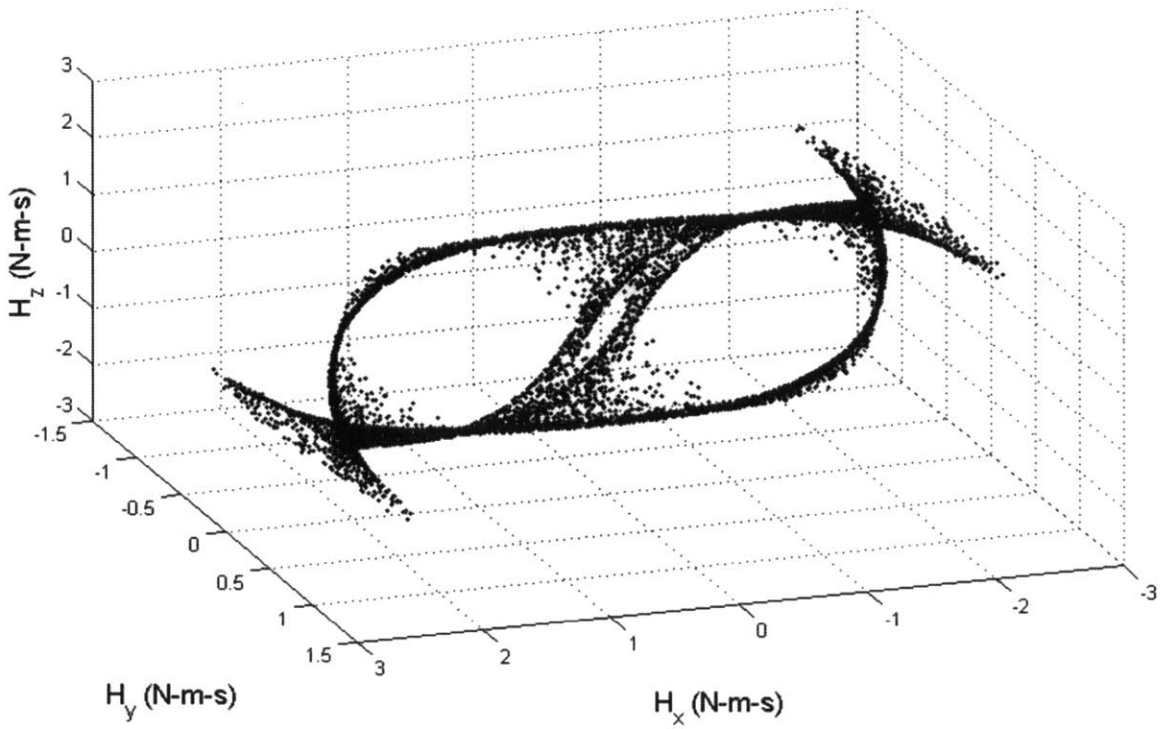


Figure 2-10: Internal Singular Surface Envelope $\beta = 54.7°$ ($\epsilon_1 = \epsilon_2 = \epsilon_4 = +1, \epsilon_3 = -1$)

be found by using the null-space associated with the current configuration. This approach is analogous to the idea of a homogeneous and particular solution to a differential equation. The pseudoinverse is used to calculate the particular solution and the null-space is used to modify the homogeneous solution into a more favorable gimbal configuration. This can be expressed mathematically as such:

$$\dot{\boldsymbol{\delta}}_{\text{CMD}} = \dot{\boldsymbol{\delta}}_u + \dot{\boldsymbol{\delta}}_{\mathcal{N}} \tag{2.44}$$

where $\dot{\boldsymbol{\delta}}_u$ are the gimbal rates found by solving Eq. 2.38 and $\dot{\boldsymbol{\delta}}_{\mathcal{N}}$ are the gimbal rates associated with the null-space. The idea of using a null-space searching algorithm to place the CMGs in a more favorable gimbal configuration will be further investigated in Sec. 3. When dealing with various CMG architectures, it is very important to understand how the singular surfaces are distributed. Different CMG configurations may be more appropriate for different spacecraft applications depending on the types of attitude slews and maneuvers that would be commonly requested.

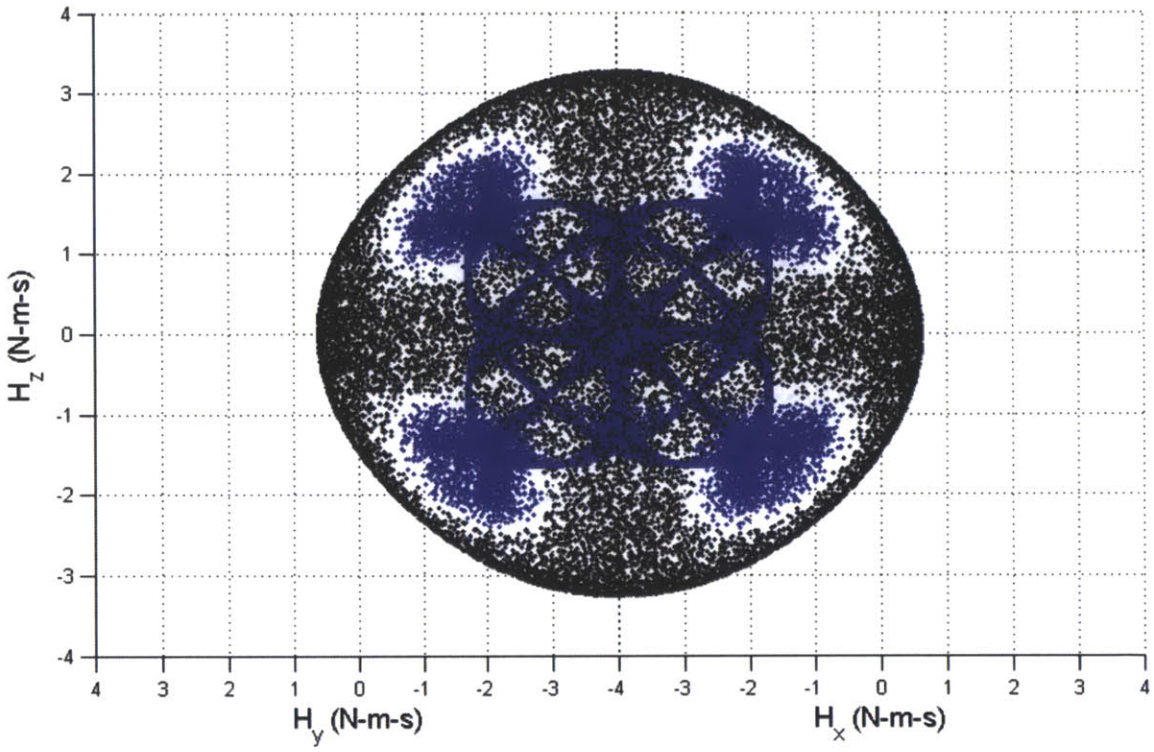


Figure 2-11: Maximum Angular Momentum Envelope $\beta = 54.7°$

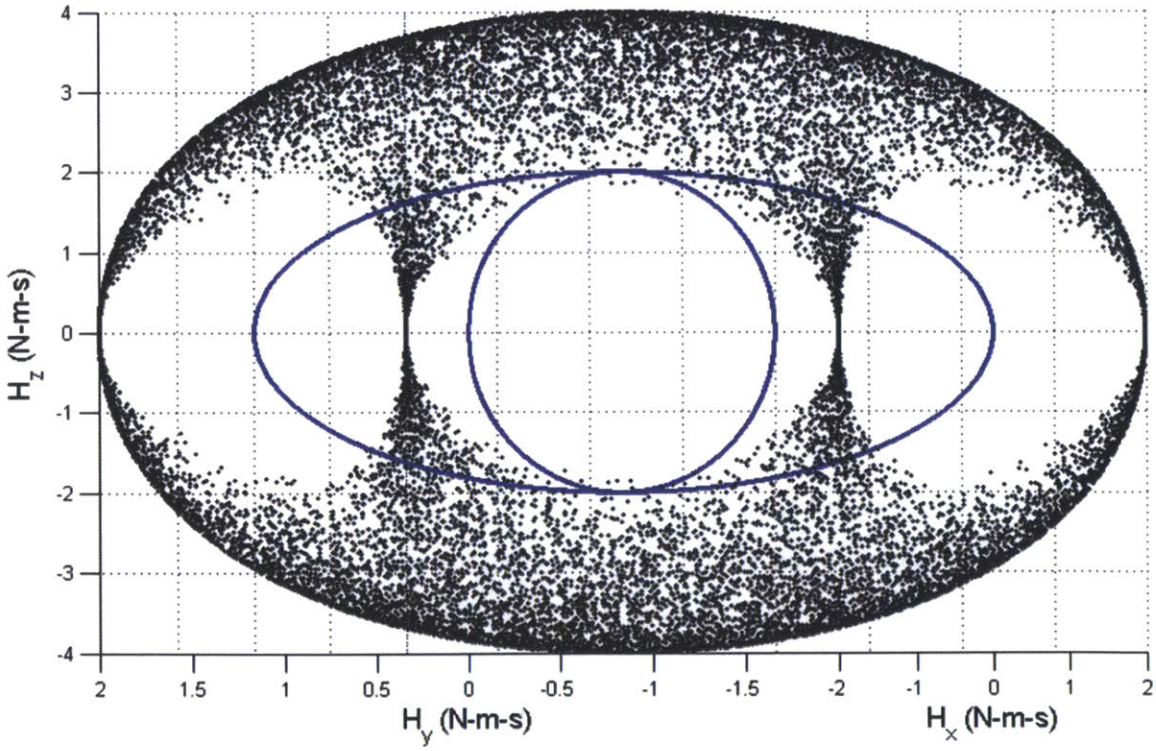


Figure 2-12: Maximum Angular Momentum Envelope $\beta = 90°$

Table 2.1: Max Ang. Momentum for Pyramid Architecture ($\beta = 54.7°$)

Component	Value
$\mathbf{H}_x$	$3.16\,h_0$
$\mathbf{H}_y$	$3.16\,h_0$
$\mathbf{H}_z$	$3.27\,h_0$

Table 2.2: Max Ang. Momentum for Pyramid Architecture ($\beta = 90°$)

Component	Value
$\mathbf{H}_x$	$2\,h_0$
$\mathbf{H}_y$	$2\,h_0$
$\mathbf{H}_z$	$4\,h_0$

The maximum external angular momentum values (expressed in terms of the constant angular momentum, h_0) of the two CMG architectures outlined above are summarized below in Tables 2.1 and 2.2. Another benefit of the current configuration is that it uses four CMGs. Using four CMGs provides one-level of redundancy in case of a CMG failure. In the case of a CMG failure, the remaining CMGs will be able to provide attitude actuation about all three directions.

Because of the extra DoF, as well as the nearly spherical maximum angular momentum envelope, the pyramidal architecture with a skew angle of $\beta = 54.7°$ has been chosen as the CMG configuration for the MAJIC ACS. The presence of the extra DoF allows null-motion solutions to help escape hyperbolic internal singularities.

2.4.2 Typical CMG Architectures

There are various CMG configurations that are commonly used in spacecraft applications. The following subsections will review the most common CMG architectures[3] and provide a theoretical background on their strengths and weaknesses.

[3]Only CMG architectures that provide three-axis attitude control are considered

Pyramid ($\beta = 54.7°$)

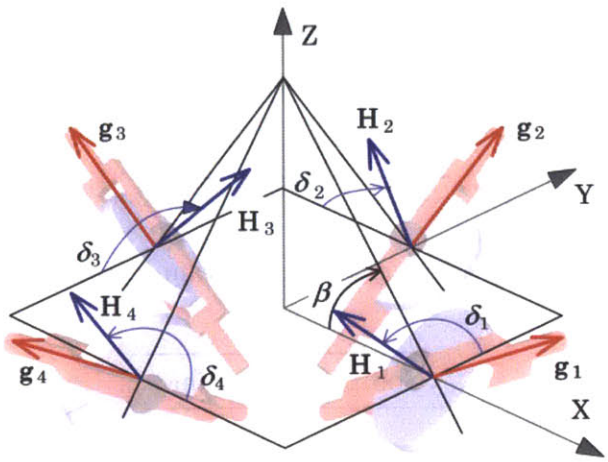


Figure 2-13: Pyramid CMG Architecture ($\beta = 54.7°$)

One of the most common architectures is the four CMG pyramidal configuration parametrized by the skew angle, β, which is defined as the inclination angle of the CMG architecture to the spacecraft's X-Y plane (Fig. 2-13). The additional CMG provides a level of redundancy in case of a failure during operation and can also be used to avoid singularities by allowing null motion which is further described in Sec. 2.4.1. The pyramidal configuration has a skew angle of $\beta = 54.7°$ because this value provides a nearly symmetric maximum angular momentum envelope in all directions. The symmetric angular momentum envelope is related to the geometric properties of a regular octahedron. The octahedron has eight faces which are all equilateral triangles and can also be decomposed into two unit square-based pyramids.

Fig. 2-14 shows a regular octahedron. The skew angle (β) corresponding to aligning a CMG at the midpoint of each base can be found by projecting the surface normals of two faces and solving for the angles of the resulting interior right triangle created by the surface normal. Fig. 2-15 illustrates this by considering the regular octahedron formed with vertices located at $\begin{bmatrix}\pm 1 & 0 & 0\end{bmatrix}$, $\begin{bmatrix}0 & \pm 1 & 0\end{bmatrix}$, and $\begin{bmatrix}0 & 0 & \pm 1\end{bmatrix}$:

The surface normals n_1 and n_2 can be written

$$\begin{aligned} n_1 &= (B - A) \times (C - B) \\ n_2 &= (D - C) \times (A - D) \end{aligned} \tag{2.45}$$

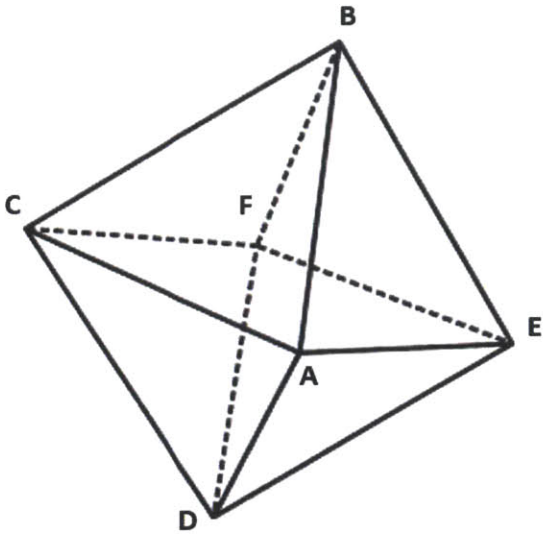


Figure 2-14: Regular Octahedron

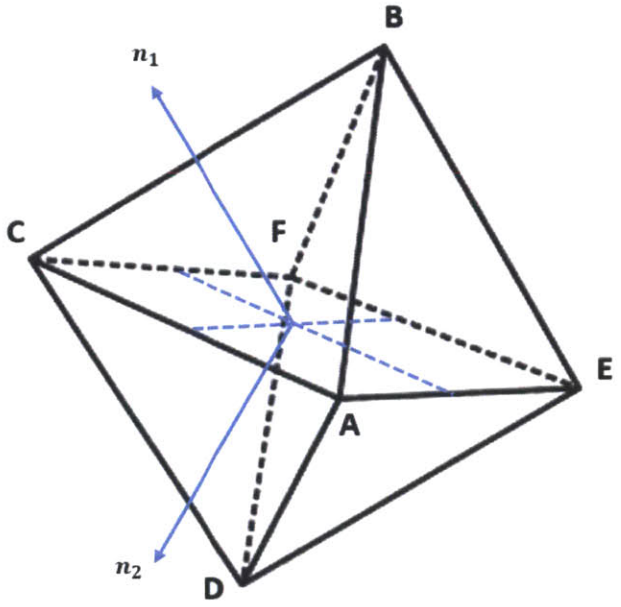


Figure 2-15: Regular Octahedron - Calculating Skew Angle (1)

and the angle between the two normal vectors can be computed using the scalar product of the normal vectors.

$$\cos(\theta) = \frac{n_1 \cdot n_2}{||n_1||\,||n_2||} \tag{2.46}$$

Once the angle has been calculated, the skew angle can be computed using certain

basic identities: (1) the sum of the angles of a triangle must be 180° and (2) the surface normal forms a right-angle with the surface, which can be seen in Fig. 2-16.

$$\beta = \frac{\pi}{2} - \frac{\theta}{2} \tag{2.47}$$

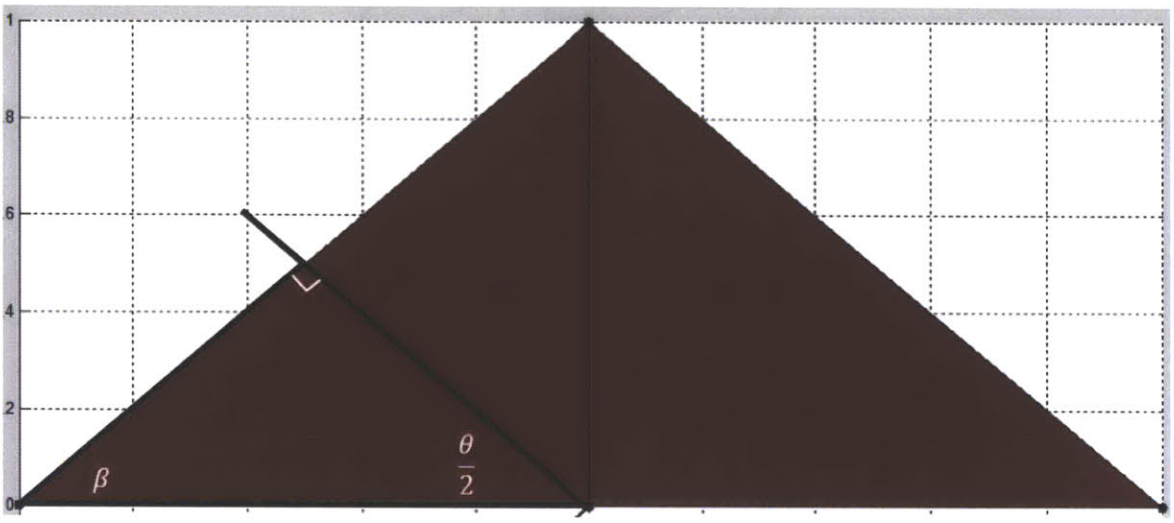


Figure 2-16: Regular Octahedron - Calculating Skew Angle (2)

The numerical solution can be found to be $\beta = 54.7°$; however, an analytical solution can also be found where $\theta = \arccos\left(\frac{1}{3}\right)$.

$$\beta = \frac{\pi - \arccos\left(\frac{1}{3}\right)}{2} \tag{2.48}$$

The analytical value for β is equal to half the dihedral angle ($\phi \approx 109.5°$) for an octahedron. This result not only agrees with the literature, but also provides an analytical solution for the skew angle.

Pyramid ($\beta = 53.1°$)

Another common pyramidal configuration is parametrized by $\beta = 53.1°$ [17]. This particular skew angle choice provides an identical maximum angular momentum in all directions and can be analytically derived using the properties of Table 2.3. By requiring that the maximum angular momentum be equal in all directions, the following

constraint is introduced:

$$2 + 2\cos(\beta) = 4\sin(\beta) \tag{2.49}$$

The solution to this equation will provide the necessary skew angle. Fig. 2-17 is a plot of both curves and one can see that Eq. 2.49 has two solutions in the range $0 \leq \beta \leq \frac{\pi}{2}$. Using MATLAB, the two solutions correspond to $\beta = 53.1°$ and $180°$; however, the latter solution is omitted because it corresponds to an angular momentum of zero. When a skew angle of $53.1°$ is chosen, the theoretical maximum angular momentum for each axis is $3.2\,h_0$, where h_0 is the constant angular momentum of each CMG.

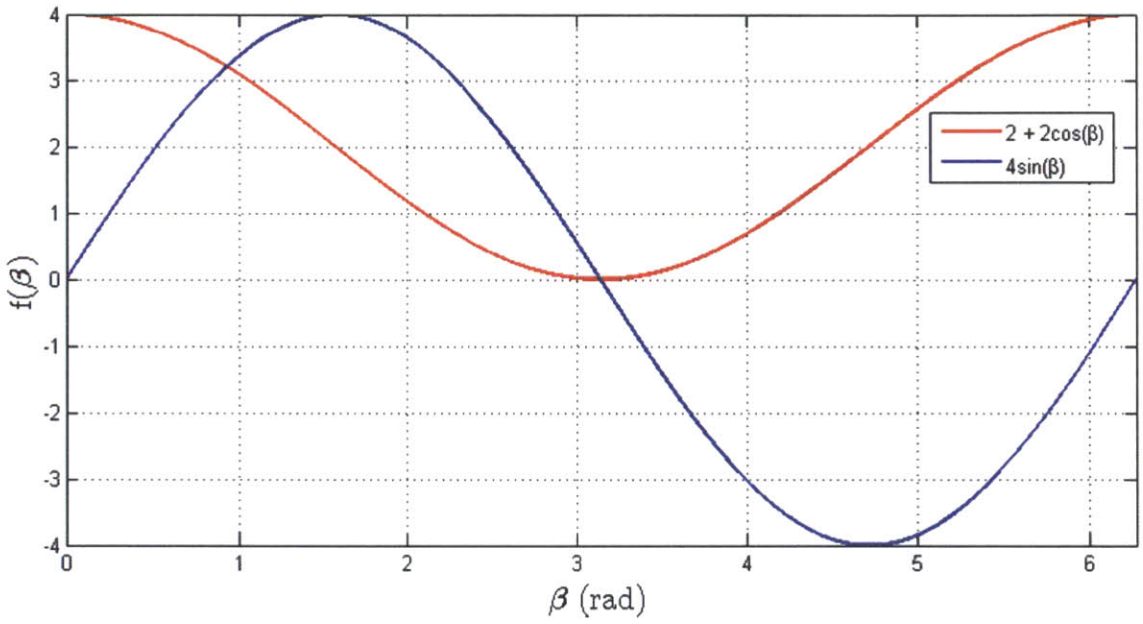


Figure 2-17: Plot of $2 + 2\cos(\beta)$ and $4\sin(\beta)$ vs. β

Box-90 ($\beta = 90°$)

The Box-90 is another typical pyramid configuration where $\beta = 90°$. Although this particular choice of skew angle does not provide a symmetrical momentum envelope, it has fewer singularities than the previous two pyramid configurations. The reduction in singularities allows for a simpler control law to be implemented, which results in a much less computationally complex steering algorithm. The singularity surfaces for

the Box-90 configuration are shown in Fig. 2-12. Since the internal singular surfaces are confined to particular arcs, simpler steering laws can be used that specifically avoid these singular regions.

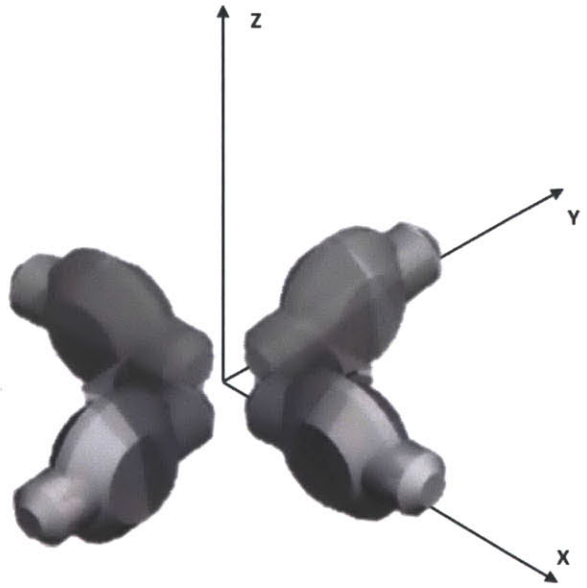


Figure 2-18: Box-90 CMG Architecture ($\beta = 90°$)

Scissored-Pair

The scissored-pair architecture utilizes six CMGs to provide three-axis attitude control. The scissored-pairs are arranged along the spacecraft's body frame such that each scissored-pair controls the torque about one axis. The major drawback of using scissored-pairs is that two additional CMGs must be used (six instead of four) to achieve three-axis attitude control, and there is no redundancy in case of equipment failure during the system's lifetime. Scissored-pairs have the simplest control law, since there are only external singularities present, which in turn causes scissored-pair CMGs to behave very much like reaction wheels, which vary the angular momentum vector by changing their flywheel's rotation rate. Since the only type of singularity present is an external singularity, the only constraint for the control law is when the scissored-pair is outputting maximum torque along a particular axis.

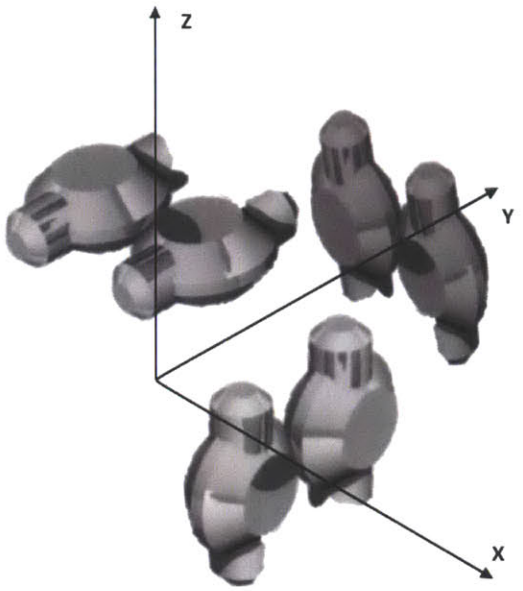


Figure 2-19: Scissored-Pair CMG Architecture

The tradeoff between steering law complexity and mass must be considered whenever CMGs are used for spacecraft ACS. Depending on the spacecraft's allowable payload and on-board computing, the choice between CMG architectures is a critical design decision. CMG sizing for a single-axis maneuver can be revisited in further detail considering the particular CMG architecture. Table 2.3 shows particular k values for some theoretical CMG architectures where β is the inclination angle of the CMGs

Table 2.3: CMG Sizing Coefficients

Architecture	$\boldsymbol{k}$
Single	1
Pyramid ($\beta = 54.7°$)	
X	$2 + 2\cos(\beta)$
Y	$2 + 2\cos(\beta)$
Z	$4\sin(\beta)$
Box-90 ($\beta = 90°$)	
X	$2 + 2\cos(\beta)$
Y	$2 + 2\cos(\beta)$
Z	$4\sin(\beta)$
Scissored-Pair	2

in the pyramid/box-90 architecture and the axes are defined as shown in Figs. 2-13 and 2-18, respectively. The coefficient for the scissored-pair configuration (Fig. 2-19) is intuitive, since it provides an additional constraint that the two CMGs have equal-magnitude and opposite-sign gimbal angles. This constraint allows the scissored-pair to apply torque about a single-axis in the same manner a reaction wheel does.

2.4.3 CMG Steering Laws

Before revisiting the control law shown in Eq. 2.37, one must consider a way in which to take matrix inverses of non-square matrices. One of the most widely used and well-known ways of doing this is the Moore-Penrose pseudoinverse [18]. The Moore-Penrose pseudoinverse can be computed using the singular value decomposition (SVD) of a matrix:

$$\bar{A} = \bar{U}\bar{\Sigma}\bar{V}^* \tag{2.50}$$

where $\bar{U} \in \mathbb{R}^{m\times m}$, $\bar{\Sigma} \in \mathbb{R}^{m\times n}$, and $\bar{V} \in \mathbb{R}^{n\times n}$. $\bar{V}^*$ is the Hermitian transpose (conjugate transpose) of the matrix $\bar{V}$. Both $\bar{U}$ and $\bar{V}$ are unitary matrices (orthogonal if $\bar{A} \in \mathbb{R}$). The columns of the unitary matrices, $\bar{U}$ and $\bar{V}$, are found by computing the eigenvectors of $\bar{A}\bar{A}^\top$ and $\bar{A}^\top\bar{A}$, respectively. Also, the singular values in $\bar{\Sigma}$ are found by taking the square-root of the eigenvalues[4] from $\bar{A}\bar{A}^\top$ or $\bar{A}^\top\bar{A}$. The pseudoinverse can be found by imposing the property that

$$\bar{A}\bar{A}^\dagger = \bar{I} \tag{2.51}$$

Plugging in the known expression for $\bar{A}$, one can compute the pseudoinverse such that $\bar{A}^\dagger$ satisfies Eq. 2.51:

$$\bar{A}^\dagger = \bar{V}\bar{\Sigma}^\dagger\bar{U}^* \tag{2.52}$$

[4]Require that $\bar{A}\bar{A}^\top$ is invertible since the eigenvalues and eigenvectors must be calculated

It is important to note that since $\bar{\Sigma}$ is a diagonal rectangular matrix, the pseudoinverse can be computed by simply taking the reciprocal of the non-zero diagonal elements (singular values) and leaving the zero elements.

$$\bar{\Sigma}^{\dagger} = \begin{bmatrix} \frac{1}{\sigma_1} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\sigma_2} & 0 & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\sigma_m} & 0 \end{bmatrix} \tag{2.53}$$

The pseudoinverse can also be rewritten in the following form

$$\bar{A}^{\dagger} = (\bar{A}^{*}\bar{A})^{-1}\bar{A}^{*} \tag{2.54}$$

Eqs. 2.52 and 2.54 can be shown to be equivalent by substituting the SVD of $\bar{A}$ into Eq. 2.54. The benefit of expressing the pseudoinverse in this manner is that there is no need to calculate the eigenvalues or eigenvectors. The pseudoinverse can be expressed in terms of the matrix $\bar{A}$ and its Hermitian transpose $\bar{A}^{*}$.

The concept of a pseudoinverse is typically required to solve for the commanded gimbal rates to produce the desired control torque. The pseudoinverse is necessary because of the dimensions of the Jacobian ($\bar{J} \in \mathbb{R}^{3\times N}$) which must be inverted. The dimensions of the Jacobian are determined by the number of CMGs; therefore, any architecture that does not contain 3 CMGs will need to make use of the pseudoinverse in computing the commanded gimbal rates. By defining $\bar{A} \triangleq \dfrac{\partial \mathbf{h}}{\partial \boldsymbol{\delta}}$, Eq. 2.38 can be re-written in a more compact notation

$$\dot{\boldsymbol{\delta}} = \bar{A}^{\dagger}\mathbf{u} \tag{2.55}$$

Bedrossian [10] has suggested an alternate form of the the proposed steering law to help avoid singular CMG configurations. The proposed steering law minimizes the tradeoff problem between minimizing the gimbal angles and solving the torque equation. The original form of the new singularity robust (SR) pseudoinverse had

two weighting matrices $\bar{K}_1$ and $\bar{K}_2$; however, by setting $\bar{K}_1 = \bar{I}$ and $\bar{K}_2 = k\bar{I}$, the SR inverse can be expressed as:

$$\bar{J}^{\dagger} = [\bar{J}^{\top}\bar{J} + k\bar{I}]^{-1}\bar{J}^{\top} \tag{2.56}$$

With $k = 0$, the solution of the SR pseudoinverse is equivalent to the original pseudoinverse shown in Eq. 2.54. Near singular gimbal configurations, a larger value of k is ideal since it will perturb the pseudoinverse away from the singularity. However, when the CMGs are far from a singular gimbal configuration, a small or zero value of k is ideal since it will reduce the torque output error. Because of the complex requirements on the gain value, k, Bedrossian suggests an algorithm which scales the value of k depending on the distance from a singularity, which is computed as:

$$m = \sqrt{\det(\bar{J}\bar{J}^{\top})} \tag{2.57}$$

The singularity measure m can be thought of as the distance from a singularity because as $m \to 0$, the matrix $\bar{J}$ becomes nearly singular and loses full rank. The algorithm assigns the gain based on the distance from a singularity:

- $k = 0 \rightarrow$ when the CMGs are far from a singularity ($m > m_{cr}$)
- $k = \dfrac{k_0}{m} \rightarrow$ when the CMGs are in the vicinity of a singularity ($m < m_{cr} \quad \&\& \quad \dfrac{k_0}{m} < k_{\max}$)
- $k = k_{max} \rightarrow$ when the CMGs are near a singularity ($m < m_{cr} \quad \&\& \quad \dfrac{k_0}{m} > k_{\max}$)

where m_{cr} is the critical value of the singularity measure, k_0 is an arbitrary constant, and $k_{\max}$ is the maximum allowable value for k.

2.4.4 RCS Thruster Jet Select Logic

The jet selection algorithms used for spacecraft's equipped with RCS thrusters determine which thrusters should fire and for how long given both translational ($\Delta\mathbf{V}$) and rotational commands ($\Delta\boldsymbol{\omega}$). It is important to consider the physical differences

between the two types of actuators: CMGs and RCS thrusters. As stated in Sec. 1.1, CMGs provide continuous attitude-actuation; whereas, thrusters are discrete actuators. Spacecraft thrusters are limited by their minimum on-time. The minimum on-time refers to how quickly the thrusters can be opened and closed and is directly related to the minimum torque which can be applied by the actuators. In order to avoid unwanted thruster firings for small attitude drifts caused by the discrete nature of the thrusters, an attitude dead-band is typically chosen depending upon the attitude precision requirements for the mission. A smaller dead-band will require much finer control and therefore will use more propellant in a given amount of time. Meanwhile, a larger dead-band will use less propellant but the attitude precision will be less than that associated with the smaller dead-band.

Two algorithms were initially considered when programming the thruster selection logic: phase plane [2] and simplex controller. The phase plane controller can be represented graphically by looking at a plot of the phase plane error (shown in Fig. 2-20) where $\dot{\theta}_{\mathrm{CW}}$ is the cushion width, θ_{DB} is the attitude dead-band, $\dot{\theta}_{\mathrm{RL}}$ is the rate limit, $\dot{\theta}_{\mathrm{DR}}$ is the drift rate, and θ_{DR} is the attitude error corresponding to entry into the drift channel.

The area shown in green corresponds to a situation in which positive thruster commands would be issued. Consider a spacecraft with a negative attitude error (θ_{ERR}) and negative attitude error rate ($\dot{\theta}_{\mathrm{ERR}}$). If no thrusters were fired, then the spacecraft's attitude error would continue to increase. Firing the positive thrusters would reduce the attitude error rate until the spacecraft enters the drift channel. Phase plane controllers have been widely studied and used on spacecrafts equipped with RCS and cold gas thrusters. A six DoF jet select algorithm can be achieved by superimposing three single-axis phase plane controllers; however, for spacecrafts that have fixed-location thrusters and time-varying inertia properties, a more robust controller is desired.

The jet selection algorithm used for the trade space analysis and design of the MAJIC system is a revised-simplex controller. The simplex controller solves a linear programming problem at each time step to minimize the cost while meeting the

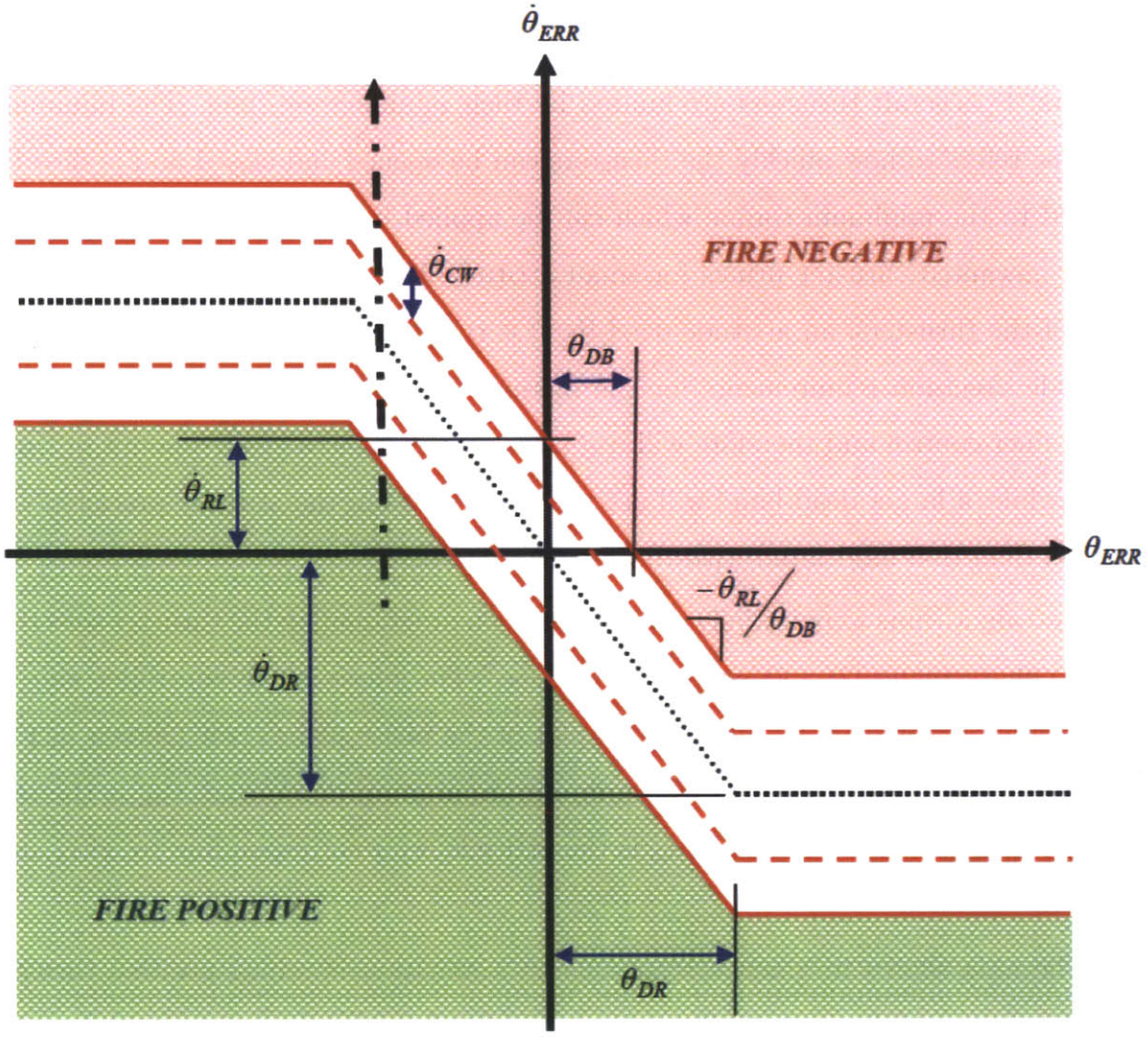


Figure 2-20: Single-Axis Phase Plane Controller - Graphical Representation [2]

constraints. The general framework for a linear programming problem is shown below:

$$\begin{aligned} \text{minimize} \quad & z &=& \mathbf{c}^\top \mathbf{x} \\ \text{subject to} \quad & \bar{A}\mathbf{x} &=& \mathbf{b} \\ & \mathbf{x} &\geq& 0 \end{aligned} \tag{2.58}$$

where $\mathbf{x} \in \mathbb{R}^{N\times 1}$ are the problem variables, $\mathbf{c} \in \mathbb{R}^{N\times 1}$ are the objective function constants, and $\bar{A} \in \mathbb{R}^{N\times N}$ corresponds to the constraint equations. The constraint equation can be modified to handle inequality constraints by adding slack variables which turn the inequality constraint into an equality constraint. The following exam-

ple problem adopted from [19] shows the general procedure for applying the simplex method to a linear programming problem with inequality constraints.

$$\begin{aligned} \text{minimize} \quad & z = -x_1 - 2x_2 \\ \text{subject to} \quad & -2x_1 + x_2 \leq 2 \\ & -x_1 + 2x_2 \leq 7 \\ & x_1 \leq 3 \\ & x_1, x_2 \geq 0 \end{aligned} \tag{2.59}$$

which can be re-written in standard form using slack variables (x_3, x_4, x_5)

$$\begin{aligned} \text{minimize} \quad & z = -x_1 - 2x_2 \\ \text{subject to} \quad & -2x_1 + x_2 + x_3 = 2 \\ & -x_1 + 2x_2 + x_4 = 7 \\ & x_1 + x_5 = 3 \\ & x_1, x_2, x_3, x_4, x_5 \geq 0 \end{aligned} \tag{2.60}$$

If the initial basic feasible solution is found using the slack variables, the following Simplex tableau can be set up

Table 2.4: Simplex Tableau - Initial

Basis	x_1	x_2	x_3	x_4	x_5	**RHS**
-z	-1	-2	0	0	0	0
x_3	-2	1	1	0	0	2
x_4	-1	2	0	1	0	7
x_5	1	0	0	0	1	3

First, the optimality of the basic feasible solution is assessed. Since the reduced costs for the non-basic variables are negative, the current basis is not optimal. The entering variable is chosen as the non-basic variable with the largest reduced cost (x_2). The leaving variable is found by taking the smallest non-negative ratio, where ratio refers to the RHS value divide by the non-zero, column-wise values of the entering variable. After the first iteration, the entering variable is x_2 and the leaving variable

is x_3 (denoted by the shading of the table). The tableau is transformed by pivoting and performing elementary row operations. After pivoting, the resulting tableau is shown in Table 2.5:

Table 2.5: Simplex Tableau - First Iteration

Basis	x_1	x_2	x_3	x_4	x_5	**RHS**
-z	-5	0	2	0	0	4
x_2	-2	1	1	0	0	2
x_4	3	0	-2	1	0	3
x_5	1	0	0	0	1	3

The entering variable is x_1 and the leaving variable, found by taking the non-negative minimum value of the ratio of the RHS and column-wise entries of the entering variable, is x_4. The same procedure is repeated until the reduced costs of the non-basic variables are non-negative. The final Simplex tableau is shown in Tab. 2.6. The final solution can be interpreted directly from the table as $x_2 = 5$, $x_1 = 3$, and $x_3 = 3$. The non-basic variables are zero and the final cost is $z = -13$.

Table 2.6: Simplex Tableau - Final Iteration

Basis	x_1	x_2	x_3	x_4	x_5	**RHS**
-z	0	0	0	1	2	13
x_2	0	1	0	$\frac{1}{2}$	$\frac{1}{2}$	5
x_1	1	0	0	0	1	3
x_3	0	0	1	$-\frac{1}{2}$	$\frac{3}{2}$	3

The same linear programming framework can be applied to the RCS jet selection logic. Eq. 2.58 is readily adapted such that the problem variables ($\mathbf{x}$) represent the thruster on-times, the objective function constants ($\mathbf{c}$) are the relative cost associated with firing each thruster, and the constraint equations ($\bar{A}$) relate the problem variables to the desired $\Delta\mathbf{V}$ and $\Delta\boldsymbol{\omega}$ commands. Since the Simplex controller solves the linear minimum fuel consumption problem at each time step, the flexibility in defining the

objective function constants allows for simulating various thruster failure modes. For instance, the objective function constant for a particular thruster can be magnified to simulate a damaged thruster. Also, different sets of constants can be utilized if there are multi-level thrusters with different operating regimes. For instance, if a spacecraft is outfitted with two types of thrusters: one that provides an extremely high-level of thrust and another that provides a low thrust, fuel efficient burn, then the objective weights can be modified to achieve the desired performance. The input vector ($\mathbf{b}$) for the linear programming problem can be defined as:

$$\mathbf{b} = \begin{bmatrix} \Delta\mathbf{V} \\ \Delta\boldsymbol{\omega} \end{bmatrix} \tag{2.61}$$

Therefore, the constraint equations must relate the thruster on-times to the corresponding instantaneous $\Delta\mathbf{V}$ and $\Delta\boldsymbol{\omega}$ associated with firing each thruster. This can be readily computed ahead of time using the simple relationships (expressed in the spacecraft's body coordinate frame, $\mathcal{F}_\mathrm{B}$):

$$\mathbf{a}_{\mathrm{trans},i} = \frac{\mathbf{F}_i}{m} \tag{2.62}$$

$$\mathbf{a}_{\mathrm{rot},i} = \bar{J}_{S/C}^{-1}\boldsymbol{\tau}_i \tag{2.63}$$

where $\mathbf{a}_{\mathrm{trans}}$ is the instantaneous translational acceleration associated with firing the i-th thruster and $\mathbf{a}_{\mathrm{rot}}$ is the instantaneous rotational acceleration induced by firing the i-th thruster. The constraint equations can be grouped into matrix form such that the modified problem exactly resembles Eq. 2.58:

$$\bar{A} = \begin{bmatrix} \mathbf{a}_{\mathrm{trans},1} & \mathbf{a}_{\mathrm{trans},2} & \cdots & \mathbf{a}_{\mathrm{trans},N} \\ \mathbf{a}_{\mathrm{rot},1} & \mathbf{a}_{\mathrm{rot},2} & \cdots & \mathbf{a}_{\mathrm{rot},N} \end{bmatrix} \tag{2.64}$$

where $\bar{A} \in \mathbb{R}^{6\times N}$. This can be verified by checking the units associated with the matrix multiplication.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 3

Six Degree-of-Freedom Simulation Environment

3.1 Background

The MAJIC simulation environment was developed in MATLAB/Simulink r2013a and is cross-compatible with Windows and Linux operating systems. A detailed front-end graphics driver is integrated such that human-in-the-loop simulations can be used to compare ACS performance and preference. The simulation is used for comparing ACS performance across various mission trajectories and profiles with automated translation and attitude commands. The automated commands are the same across the various control systems which allows for detailed comparisons between control mode performance. The simulation, integrated with the front-end graphics program, is used to determine user preference between ACS, as outlined in Sec. 3.2. The program utilized for the graphics portion of the simulation is NASA's Engineering Doug Graphics for Exploration (EDGE) package. EDGE provides a real-time graphical interface which allows for precise representation of position and attitude data for all objects in a scene. The integration of EDGE and MATLAB/Simulink is a novel concept developed for this research since most of NASA's simulations are developed using the Trick Simulation Development Toolkit [20]. The MATLAB/Simulink interface is further discussed in Sec. 3.2. The outline below summarizes the different responsi-

bilities of each of the software components involved in the simulation environment:

- MATLAB → initialization files, mathematical utilities EML code, post-processing and plotting scripts
- Simulink → numerical integration of dynamics, ACS logic, data logging
- EDGE → graphical interface for MATLAB/Simulink

The mathematical (quaternion) utilities developed for the simulation are adopted from another Draper simulation [21]. Using the same mathematical functions ensures that the simulation and results are cross-compatible with other internally developed simulation environments. Another benefit of using Draper's mathematical utilities was that it did not require MATLAB's Aerospace Toolbox[1] and could therefore be run on a basic installation of MATLAB with purchasing additional licenses. The simulation uses a fixed-step, fourth-order Runge-Kutta (RK4) method for integrating the dynamics forward in time. The time-step chosen for the simulation is 50 Hz. The controller frequency is specified as 25 Hz to appropriately model the discrete-time nature of a real spacecraft ACS.

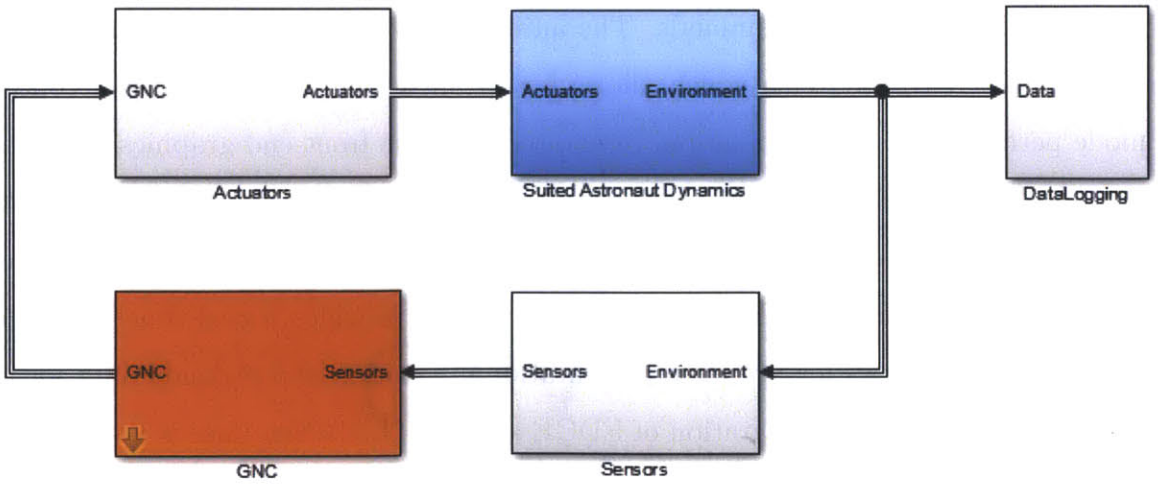


Figure 3-1: Top-Level Simulink Block Diagram

[1]The Aerospace Toolbox contains the functions required to convert between quaternions and DCM's and Euler angles and vice versa

Fig. 3-1 shows the top-level Simulink block diagram of the MAJIC simulation. The blocks are separated into the major MAJIC subsystems: actuators, dynamics, sensors, and guidance, navigation, and control (GNC). The actuators block maps the input commands to the actuators and outputs the corresponding forces and torques, which are then fed into the dynamics block. The output of the actuators block imposes saturation effects and minimum thruster on-time limits, as well as tracking propellant consumption. The dynamics block receives the forces and torques fed in from the actuators block, as well as any environmental torques (i.e. gravity gradient, solar wind, aerodynamic drag, etc.) and disturbance forces and torques. The dynamics block propagates the quaternion states forward in time using the RK4 method. Traditionally, an astronaut and spacesuit are modeled as a rigid-body; however, the MAJIC simulation environment models the astronaut and spacesuit as a composition of several rigid segments which consist of human-model, spacesuit and mass properties of the baseline NASA Jetpack design. Each segment can be varied for crew-member size and mass properties. The torques, forces, and time-varying inertia properties associated with an astronaut moving are included for further investigation into ACS performance. The output of the dynamics block is then passed to the sensors block which models the actual signal that would be received by the GNC subsystem. Perfect sensor data is assumed, since sensors are not currently modeled in the simulation. However, this can be easily modified assuming an accurate stochastic model is known for a particular sensor. The sensor data is received by the GNC block which generates the appropriate $\Delta\mathbf{V}$ and $\Delta\boldsymbol{\omega}$ commands for the current stage of the mission. The data logging block saves the time history of the state signals, as well as other relevant vehicle information such as fuel consumption, attitude error, and commanded RCS thruster on-times.

The simulation is a tool to explore the trade space between different ACS designs which involves not only the types of actuators, but also the number and orientation of the actuators (particularly in the case of the CMG architecture). Since varying the number of on-board actuators requires changing the dimensions of particular simulation parameters (e.g. $\bar{J} \in \mathbb{R}^{3\times N}$), an initial condition file assigns the size of dependent

signals based on the particular ACS design. The simulation includes several control and pointing modes to allow the most flexibility when designing representative missions. Control modes refer to which actuators are being used for translation and attitude control. The six basic control modes are summarized below:

- 0 → CMGs only (can only be used on missions that do not require translation)
- 1 → RCS thrusters only (translation & attitude)
- 2 → RCS thrusters (translation) and CMGs (attitude)
- 3 → Human-in-the-Loop (subject controls translation inputs)
- 4 → Human-in-the-Loop (subject controls translation and attitude inputs)
- 5 → Desaturation

Pointing mode refers to the state of the system's attitude as a function of time. For example, in an attitude hold maneuver, the attitude of the vehicle should remain constant over time; whereas, during a free drift the attitude of the vehicle changes but not with any definitive purpose. The three pointing modes are summarized below:

- 0 → Free drift (no specific pointing requirements)
- 1 → Attitude hold (spacecraft holds its current attitude and nulls any body rate)
- 2 → Target track (spacecraft aligns with particular vector in $\mathcal{F}_N$)

The control and pointing mode are initialized in the MATLAB scripts, but can be changed during a mission. This feature is particularly important in the event of a CMG saturation, where the control mode would need to be briefly changed to control mode 5 (desaturation mode) before returning to its previous mode. Control mode 0 is rarely used and has been included for completeness. For the analysis in Sec. 3.3, the control mode will be limited to either 1, 2, or 5 and the pointing mode will be restricted to either 1 or 2.

3.2 Virtual Reality Lab Testing

The real-time graphics program used by the MAJIC simulation environment was developed and is currently maintained by the Virtual Reality (VR) Lab at NASA's Johnson Space Center. There are two versions of the visualization software: Dynamic On-board Ubiquitous Graphics (DOUG) and EDGE (Engineering DOUG Graphics for Exploration). EDGE is the publicly available version of NASA's DOUG program and and is identical in nature except the default scene which features the ISS has less detailed models due to ITAR restrictions. Sec. 3.2.1 presents a realistic EDGE scene and describes how the graphics program is structured. Also, the MATLAB/Simulink interface will be presented which is used to transmit dynamic, real-time information to EDGE such that the scene portrays the simulation environment. EDGE can be used with scripted procedures written in Tcl [22].

3.2.1 EDGE

EDGE utilizes a tree-based hierarchy to determine how nodes are placed into a scene. The topmost node is Null, which parents each underlying node. Every node has a position (x, y, z) and orientation defined by a Pitch-Yaw- Roll (3-2-1) rotation sequence. The positions and orientations are defined relative to one another such that a node with position (0,0,0) and orientation (0,0,0) would be in the exact same position as its parent[2]. Nodes that do not have a 3D Computer-aided design (CAD) model associated with them are defined as system nodes. Fig. 3-2 exhibits a typical organizational structure for an EDGE scene load.

The entries of the tree are color-coordinated by type of node. For instance, system nodes are purple, camera nodes are red, light nodes are yellow, and model nodes are blue. Although it is not necessary to create two separate nodes (i.e. ITOKAWA_SIM_POS and ITOKAWA_SIM_ATT) for a model's position and attitude, it can be extremely useful when placing cameras throughout the scene. If a camera is parented to a model's position node, then the camera will remain at the

[2]EDGE's default units are inches (in) and degrees (deg)

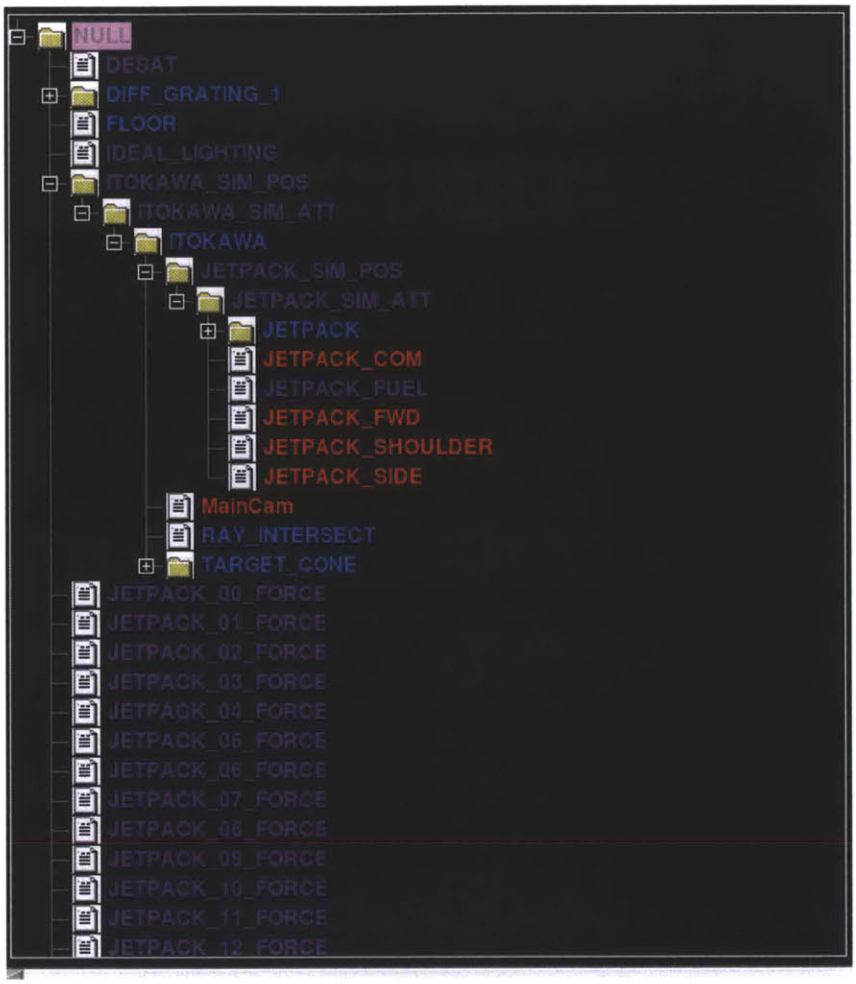


Figure 3-2: EDGE Tree Organizational Structure

same relative distance and orientation regardless of the model nodes attitude. This can be useful when trying to image certain aspects of a scene such as a rocket launching or a spacecraft berthing with the ISS, where attitude changes must be shown. It follows that a camera parented to a model's attitude node, will remain at the same distance and orientation irregardless of the model's position and attitude. These types of cameras can be extremely useful for showing a viewpoint from inside a crew-member's EVA suit since the camera will track with the crew-members position and attitude. It is recommended that all files pertaining to a particular EDGE scene load are kept in a single userdata directory. By keeping all of the project files organized in one location, the userdata directory allows the specific EDGE installation to be upgraded without altering or modifying any behaviors of the particular scene. In

csh, the userdata environmental variable, which controls what userdata directory is sourced for the scene, can be changed by entering

```
setenv USERDATA userdata.project
```

where *userdata.project_name* is the name of the directory that contains the specific scripts and files for the project. The userdata directory is organized in such a way that inexperienced users can quickly learn to create and personalize a new project. The itemized list below shows the most relevant directories and corresponding files:

- ./userdata.project_name/configs/user_models.cfg → Define nodes, cameras and light positions, orientations, and settings.
- ./userdata.project_name/models/ → Organizes CAD files for relevant models in project.
- ./userdata.project_name/sim_data/ → Organizes data files for playback and recording (.trk, .csv, etc.).
- ./userdata.project_name/states/ → Define base states for particular scenarios. Overrides user_models.cfg settings.
- ./userdata.project_name/textures/ → Define texture files which can be applied to model surfaces.
- ./userdata.project_name/user.cfg → Main configuration file for project. Loads specific plugins and terminal commands.

Nodes can be added to a scene by editing the configuration (user_models.cfg) file. The node class is defined below:

```
node(node_name);
        model(file_name);
        xyz(x_val, y_val, z_val);
        pyr(pitch_val, yaw_val, roll_val);
        order(PYR/PRY/YPR/YRP/RPY/RYP);
```

```
parent(node_name);
layer(BACKGROUND/SCENE/FOREGROUND);
priority(val);
lod(SIZE/DISTANCE/SELECT/TIME, ...
    size/dist/obj_id/duration, file_name);
```

While running EDGE, a user can access and modify a node's position and orientation values by accessing the individual elements (e.g. x,y,z and pitch, yaw, roll), as well as the specific node's parent and visibility. This property lets you modify the nodes position and orientation by changing one or more values at a time.

The user can use a subset of the fields above and the default values will be applied; however, the node and parent node's name must be filled in for any node placed into the scene. As many nodes can be added to a scene as necessary. When interfacing with MATLAB/Simulink, it is often very useful to use system nodes (nodes without a model attached) to store other pertinent simulation information. For instance, if one created a heads-up-display (HUD) that used the actual propellant consumption of the vehicle to estimate the mission time remaining, then the propellant consumption data for the vehicle can be sent to EDGE through a system node called VEHICLE_FUEL. Since each node has a position and attitude, six pieces of data can be stored in the VEHICLE_FUEL node to display other critical mission parameters.

The default screen for the MAJIC simulation is shown below in Fig. 3-3. The tree view and node dialog boxes are available through the toolbar (Edit). Also, different display settings can be chosen such as having one view open on the window or having twin left and right viewing windows. The default cameras assigned to these windows can be modified by editing the main configuration file (user.cfg). The toolbar and right-click menu can also be modified and custom utilities can be added; however, that is beyond the scope of a beginning user and is omitted for brevity.

In order to interface MATLAB/Simulink with EDGE, a communication protocol was written such that data could be transmitted back and forth between MATLAB/Simulink and EDGE in real-time. The reason for requiring two-way communication was because EDGE scripts and/or plugins can also modify node values.

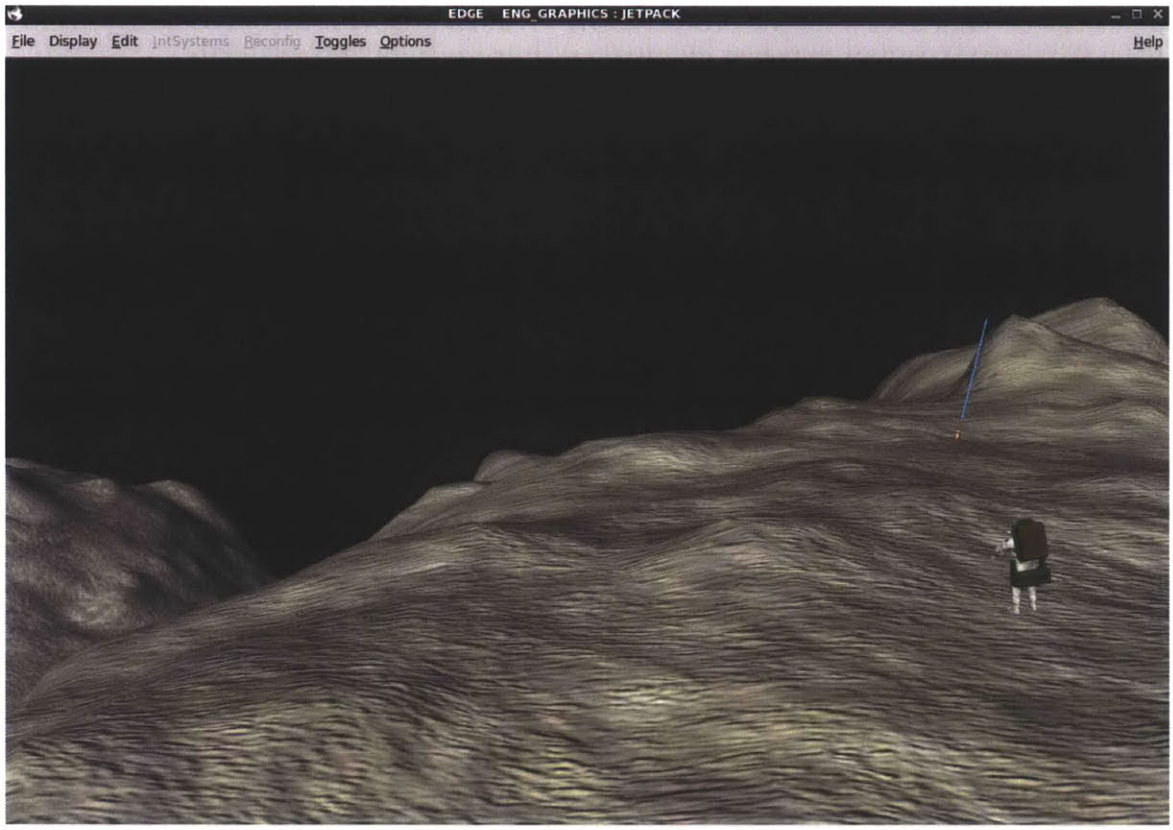


Figure 3-3: Default EDGE Scene

Therefore, if a particular script being run causes an object to change orientation and MATLAB/Simulink is trying to track that very same object, it is important to get accurate orientation data from EDGE. EDGE has the unique ability to be able to run in a standalone mode or as a manager and client(s) pair. When running in standalone mode, one simply launches EDGE and interacts directly with the one window. Conversely, when running in a manager-client mode, one must initialize a non-graphical manager followed by one or more clients. The manager's main responsibility is to determine who can access the default tree for the current project and moderate the information transmitted and received by each client. The clients all have access to the system nodes and can manipulate camera angles, lighting parameters, etc. without adversely affecting the other clients. This is a particularly useful feature if multiple monitors are used to display a common EDGE rendering. Each monitor can have pre-defined settings, camera angles, and peripherals depending on the purpose of that specific monitor. EDGE's default C/C++ communication protocol (DCOMM) has

been extensively tested and has hundreds of functions already defined.

In order to facilitate communication between MATLAB/Simulink and EDGE, a Simulink Level 2 C-MEX S-function was created that includes the DCOMM header file and all the additional functions. A C-MEX S-function is a Simulink wrapper which allows the execution of C/C++ code within a simulation. Directly integrating the communication protocol between Simulink and EDGE easily allows transferring real-time data for many variables. Once the .mex file has been generated and MATLAB establishes connection with EDGE, the incoming Simulink signals are mapped to nodal quantities in EDGE. Since the default units for EDGE are inches and degrees, some unit conversions may also have to take place depending on which functions are implemented from the DCOMM library. The mapping between signals and nodal quantities is done using an application programming interface (API) file. This file specifies which signal belongs to which node and nodal quantity, as well as the specific units for each. The API file is parsed by the C/C++ file to determine how to handle the signals entering the S-function block in the the Simulink diagram. Fig. 3-4 shows an example Simulink implementation of the DCOMM library. The signals are muxed into one aggregate signal and the S-Function parses the API file to determine which nodes to send the data to. The figure also shows data being received by Simulink regarding the position of a target cone. This data can then be used to trigger certain events or modify simulation behavior depending on the value.

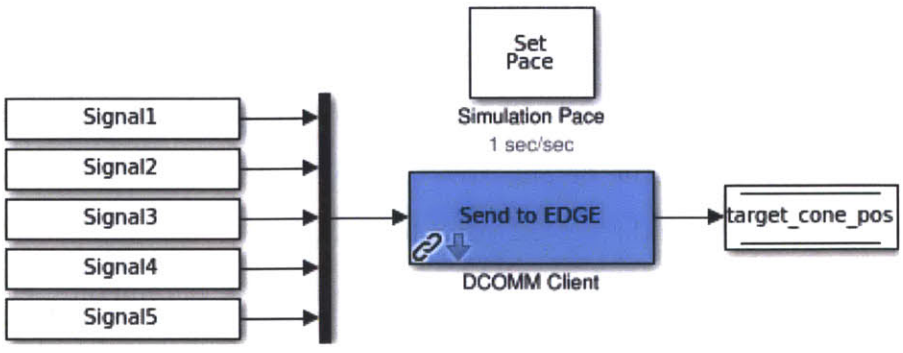


Figure 3-4: MATLAB/Simulink Connection to EDGE

Writing a reliable parsing function is critical to transmitting data accurately between MATLAB/Simulink and EDGE. The parsing tool developed for this implemen-

tation is modeled after the API functions built-in for Trick simulations.

<NodeName> <SignalName> <DoF> <Units> <Write> <Scale>

where *NodeName* is the node name in EDGE (case-sensitive), *SignalName* is the signal name associated with Simulink (optional), *DoF* is the position or attitude component that the data is being sent to, *Units* are the natural units of the simulation (in = inches and d = degrees), *Write* defines data that will be written to EDGE, and *Scale* is a separate scaling parameter (optional). The matching API file for the Simulink diagram presented in Fig. 3-4 is shown below for reference:

```
JETPACK_SIM_POS signal1 x     in write 1.0
JETPACK_SIM_POS signal2 y     in write 1.0
JETPACK_SIM_POS signal3 z     in write 1.0
JETPACK_SIM_ATT signal4 pitch d  write 1.0
JETPACK_SIM_ATT signal5 yaw   d  write 1.0
```

3.2.2 Simulation Validation

The ability to portray the simulation environment with a front-end graphics software package is extremely useful for simulation troubleshooting and validation. Simulation data can be sent between the graphics program and vice versa so that the scene provides a real-time look into the mechanics of the simulation software. Additionally, data is transmitted back and forth through the use of system nodes.

System nodes, or nodes that do not have a model attribute associated with them, are used to store relevant information and simulation parameters. For instance, the simulation time is transmitted to EDGE and displayed on the screen so that the pilot can pace himself/herself. This is done by creating a system node called SIM_TIME and storing the simulation's current time as the node's x-value. This technique is used to display the simulation time on the HUD. The driver script for the HUD uses a built-in Tcl command that checks whether a node has been updated. The following code shows how the command is used to display the simulation time:

```
doug.callback add update -nodes "SIM_TIME" {
```

```
        set sim_time [doug.node SIM_TIME get -x]
}
```

where SIM_TIME is the system node name used to store the relevant data and sim_time is a local variable used inside the script to display the current simulation time. Furthermore, system nodes are used with model nodes to trigger particular behaviors such as displaying thruster plume models. A system node keep tracks of the number of on-off commands associated with each individual thruster model node. A Tcl script checks the system nodes for updates and displays or hides the plume model depending on whether the counter is even or odd. Two nodes are necessary because the plume model has a particular location and orientation in the MAJIC body-fixed coordinate frame. Therefore, another node must be used to track the on-off commands received by the ACS. Indicating which thrusters are being commanded provides insight into the behavior of the ACS.

During the initial interfacing of the simulation environment, two discoveries were made that were directly attributable to an error in the simulation. Although the automated translation commands were allowing the MAJIC system to navigate from way-point to way-point, the wrong thrusters were being fired. After further investigation, it was also discovered that depending on the inertial position of the MAJIC system, the thrusters would sometimes fire correctly and sometimes incorrectly. This would be caused by an improper representation of the spacecraft's position and/or attitude in the spacecraft's body-fixed ($\mathcal{F}_B$) and inertial ($\mathcal{F}_N$) coordinate frames. If the conversion between the spacecraft's body-fixed coordinate frame conflicts with its orientation in the inertial coordinate frame, then the wrong thrusters will be commanded.

As predicted, the source of the error were EML blocks which utilized mathematical utilities to convert between Euler angles and quaternions and conversely. Draper quaternion utilities are written for both left- and right-hand conventions. Before the simulation initializes, a parameter is defined which tells the compiler whether to use left- or right-handed conventions. The default is right-handed unless otherwise specified by the user. The MATLAB function genpath(*FolderName*) is used to recursively

create a path string that contains *FolderName* and all of the subdirectories underneath it. The path string is searched for subdirectories that contain the particular convention and the remaining subdirectories are removed from the path. The naming convention for subdirectories that contain left- and right-hand convention-specific functions is *subdirectory_LH* and *subdirectory_RH*, respectively. Since each directory contained the exact same function names, it is important to remove the appropriate directories before compile time. If both directories (*subdirectory_LH* and *subdirectory_RH*) were included during compilation, MATLAB would search the path for the particular function and choose the left-handed directory because it occurs first alphabetically. This would mean that the wrong functions are being used and that the outputs of these functions would be incorrect for the particular convention. The error was not with the original script used to detect and remove the miscellaneous subdirectories. The error occurred when porting the simulation to be cross-platform. The MATLAB function genpath(*FolderName*) uses a different delimiter to separate the recursive path string depending on the OS. Therefore, when the simulation was ported to Linux from Windows, the script could no longer parse the path string properly and the resulting subdirectories seen by the compiler did not exist and were ignored. Once the cause of the error had been found, it was relatively simple to fix. First, the compiler detects which type of OS architecture is being used and then searches the path string with the appropriate delimiter to remove the necessary subdirectories. Once the necessary changes were implemented into the simulation, it was extremely easy to see that the correct thrusters were firing to navigate the MAJIC system using autopilot. The thruster plume visualization made troubleshooting the error much simpler, in contrast to looking at plots of commanded on-times for 24 individual thrusters.

3.2.3 Preliminary Human-in-the-Loop Test Results

Since a CMG is an electrically powered device, it follows that a combined ACS consumes less propellant than a thrusters-only ACS. Attitude stability is also improved due to the fact that CMGs are a continuous attitude actuator and thrusters are

discrete and limited by the minimum on-time and attitude dead-band set by the controller. The simulation environment answers these quantitative questions automatically; however, it does not answer qualitative questions as easily, such as:

- Can a human pilot tell the difference between the ACS visually?
  - If so, does the pilot have a strong preference for one ACS over another?
  - If not, why not?
- Are there any specific HUD notifications the pilot would require for a given ACS?
- Does a human pilot's performance change when using a different ACS?

To answer these questions, a human-in-the-loop (HITL) subject evaluation was conducted at JSC. The evaluation takes place on the surface of an asteroid (e.g. Itokawa) and each subject must navigate from way-point to way-point. The way-points are indicated by a traffic cone on the surface of the asteroid. Only one target cone is visible at a time and only moves to its new location when the subject is within the cone's sphere of influence, which has been arbitrarily set at 12 ft. The subject is responsible for commanding translation to maneuver the MAJIC system from cone to cone. Meanwhile, the vehicle's attitude is automatically controlled using target tracking. The subjects each flew two ACS architectures: RCS only and a combined ACS using oversized CMGs. The CMGs used in the obstacle course were chosen such that the CMG arrays would not saturate during the mission since the desaturation algorithm had not been implemented yet. Another round of HITL testing was conducted at a later date which also included an ACS design with small-scale CMGs and momentum desaturation capability (Sec. 3.4).

Fig. 3-5 illustrates a subject approaching a target cone. The blue cylindrical shape originating from the top of the traffic cone allows the subjects to see the cone from far distances when his/her view of the cone may be obscured by the terrain. The evaluations are presented in a single-blind format, where the subject is unaware

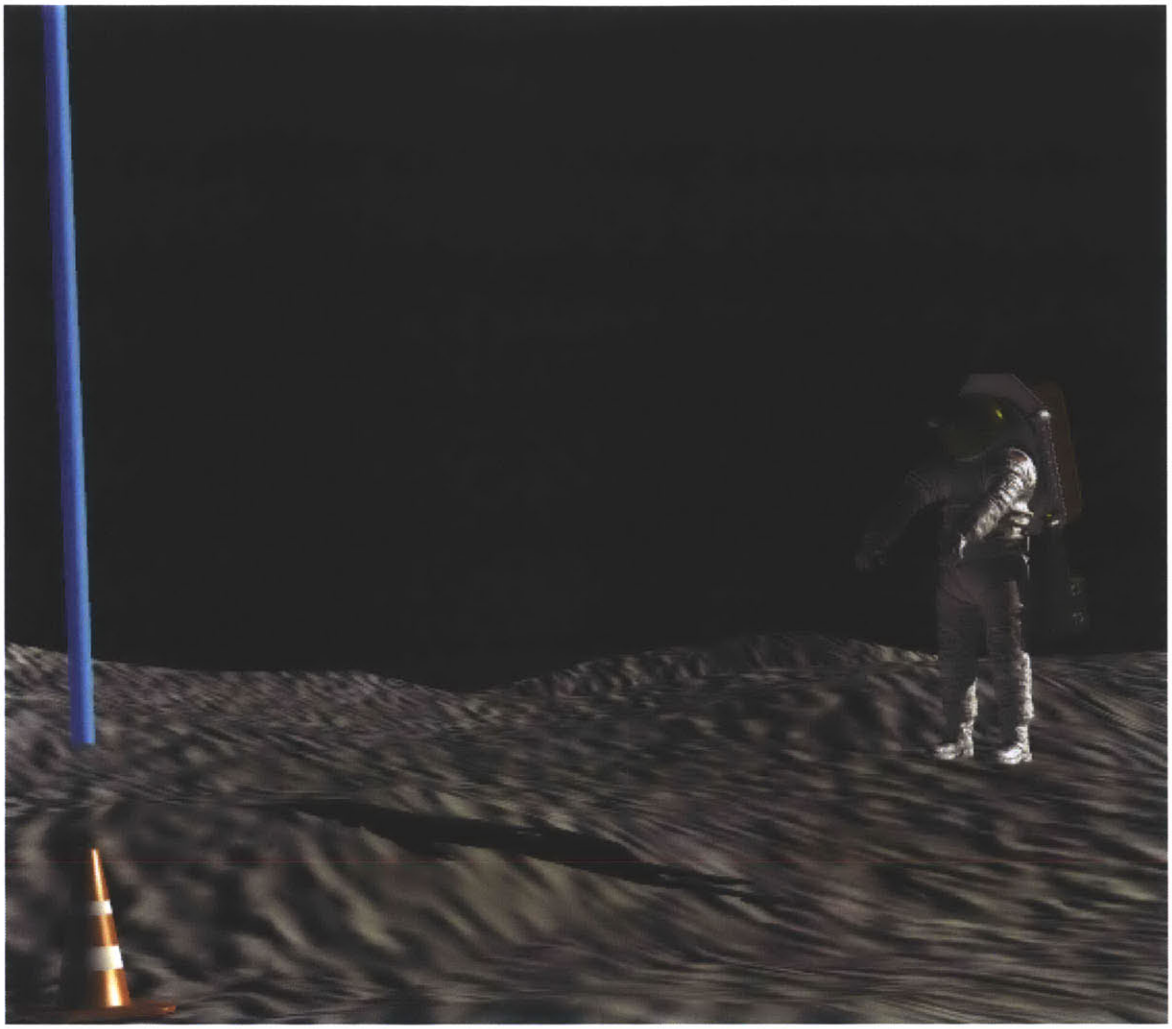


Figure 3-5: HITL Evaluation - Target Cone

which ACS he/she is flying and in what order but the person(s) administering the evaluation are aware. The experience range of the subjects varied from very highly qualified with several hundred hours of flying time (e.g. current/former astronauts with EVA experience, simulation developers, etc.) to medium qualified with only a couple hours of flying time. The subjects are given a few minute practice run before the actual evaluations begin to ensure that they are familiar with the hand controller, which can be seen in Fig. 3-6.

The labels for the hand controller indicate which ΔV commands are input when pressing each particular button. The commands are entered in the MAJIC system's body-fixed coordinate frame, $\mathcal{F}_{\mathrm{B}}$. The minimum controller input is calculated using

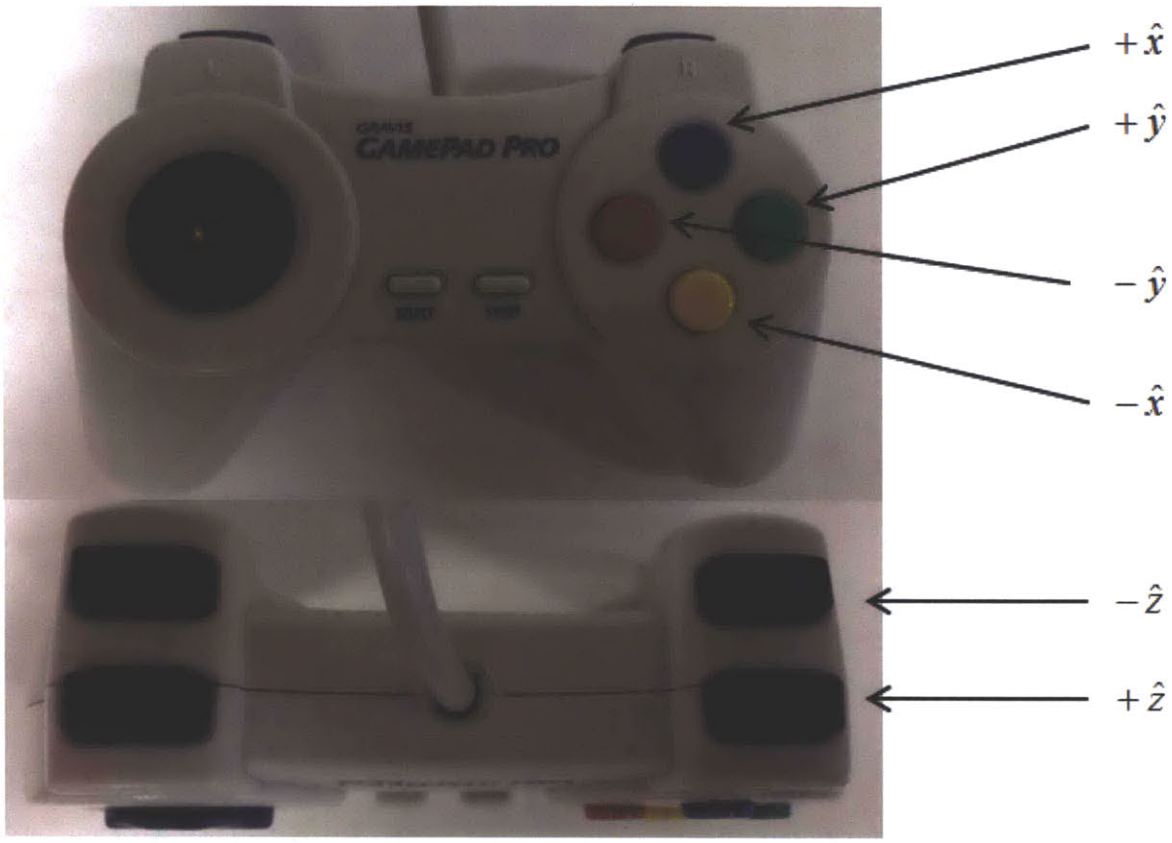


Figure 3-6: HITL Evaluation - Hand Controller

Eq. 3.1

$$\Delta V = \frac{4\mathrm{F}\Delta t}{m} \tag{3.1}$$

where F is the force of each thruster (0.8 lbf) , Δt is the minimum on-time of each thruster, and m is the initial mass of the MAJIC system. The scalar factor of 4 in Eq. 3.1 is due to the fact that typically four thrusters will fire for a pure translation input. Since the controller provides ΔV inputs, after a command is received the vehicle maintains that velocity until another command is input. This specific controller input style was selected because it matches the original SAFER's hand controller design and the majority of the subjects have had experience flying the SAFER simulation.

Fig. 3-7 shows the results of the piloted evaluations from JSC. There were a total of 13 subjects with varying degrees of experience. The plot shows the mean fuel consumption data for the task, as well as one standard-deviation, for each control

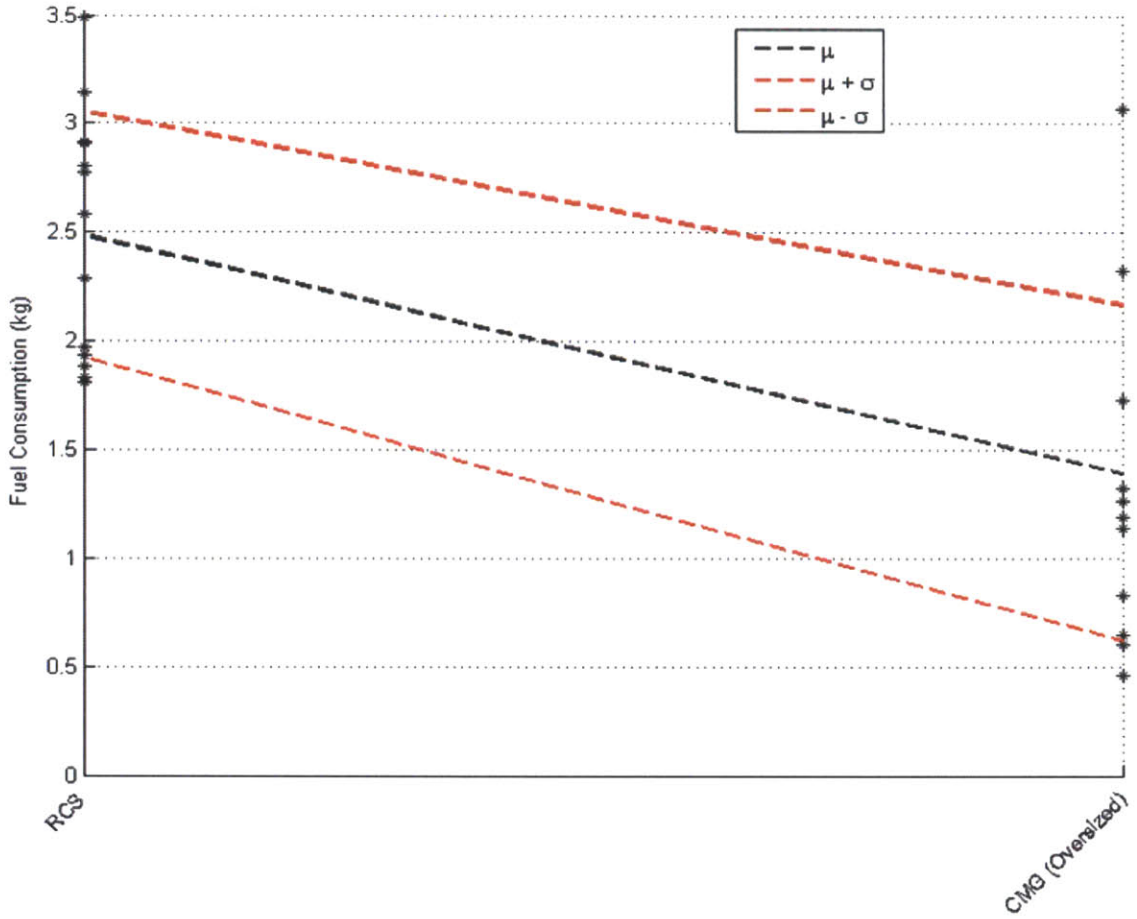


Figure 3-7: HITL Evaluation - Subject Data

mode. From the data, it is apparent that the average fuel consumption for the CMG ACS ($\mu_{\text{CMG}} = 1.39$ kg) is much lower than the average fuel consumption for the RCS thrusters only ($\mu_{\text{RCS}} = 2.49$ kg). In fact, there is a 44.1% difference in the mean fuel consumption among the different control modes. For this particular mission profile, that results in a net fuel savings of 1.1 kg. The fuel savings associated with this mission profile may not seem like a definitive reason to incorporate CMGs into the MAJIC system; however, considering the lifetime of the operational unit, the fuel savings over many EVAs will be substantial.

3.2.4 CMG Momentum Desaturation

Momentum desaturation is a critical factor in determining whether or not CMGs should be added to any spacecraft or vehicle. Since CMGs operate under the purview of the law of conservation of angular momentum, the spacecraft must be able to "dump" angular momentum without affecting mission performance. As previously mentioned, there are several techniques that can be used to dump angular momentum such as magnetic torquers and known external torques (i.e. gravity gradient, solar wind, etc.). For the MAJIC system, the primary method of reducing angular momentum and desaturating will be using the RCS thrusters. During a desaturation maneuver, four critical things must occur:

- Activate desaturation when nearing CMG saturation
- Utilize thrusters and switch pointing mode to attitude hold
- Send specific command to allow CMGs to unwind
- Deactivate desaturation, switch back to desired pointing mode

The MAJIC system must be able to determine when the CMGs are approaching saturation about a particular axis. When this occurs, the CMGs will no longer be responsible for pointing the spacecraft. Instead, the thrusters will take over primary control and the pointing mode will automatically switch to attitude hold. Switching to attitude hold will ensure that no desirable motion will occur when the CMGs are desaturating. This will also prevent the crew-members attitude from drifting due to any non-zero attitude rates at the time of saturation. As soon as the thrusters take over in attitude hold, the CMGs will begin to unwind. The unwinding is caused by sending a series of rotational commands to the CMG, which are computed using Eq. 3.4.

$$\boldsymbol{\tau} = \bar{I}\boldsymbol{\alpha} \tag{3.2}$$

which can be re-written as

$$\frac{\mathbf{h}}{\Delta t} = \bar{I}\frac{\Delta\boldsymbol{\omega}_{\text{CMD}}}{\Delta t} \tag{3.3}$$

$$\Delta\boldsymbol{\omega}_{\text{CMD}} = \bar{I}^{-1}\mathbf{h} \tag{3.4}$$

where $\Delta\boldsymbol{\omega}_{\text{CMD}}$ is the command sent to the CMG controller, $\bar{I}$ is the MAJIC composite inertia matrix and $\mathbf{h}$ is the current angular momentum of the CMG array.

Once the CMGs have finished desaturating, the system returns to a normal operational state by deactivating the thrusters and switching the pointing mode back to what it was initially before desaturation began. The thresholds for initiating and finishing desaturation are user-defined; however, for the MAJIC system, the values are set as 90% and 10% saturation, respectively. The values mean that once the CMGs are approximately 90% saturated about any particular axis, desaturation begins. Then, the desaturation mode remains operational until the CMGs fall below 10% saturation. This safeguards the system from hastily exiting desaturation until the CMGs are brought to within a safe operating level once again, while also minimizing the time spent in desaturation mode. For instance, if the upper and lower limits were set at (80%, 20%), the CMGs will saturate before nearing the surface of the singularity map and also exit desaturation mode before dumping enough angular momentum.

3.3 Preliminary Monte-Carlo Case Study

The concept of using a Monte-Carlo brute-force style technique to simulate complex systems with many dependent parameters has been around for a long time. However, applying this method to the sizing of a spacecraft's ACS is a much less explored area. The basic premise of a Monte-Carlo simulation is to assume a known distribution for a set of independent parameters and carry out the simulation. This process is repeated numerous times to obtain the distribution of the quantities of interest. Much care must be taken to ensure that the random number generator (RNG) used does indeed

produce quality pseudo-random numbers [23]. A Mersenne Twister RNG is used for the Monte-Carlo trials described below [24]. The Mersenne Twister produces uniform pseudo-random numbers with a period of $2^{19937} - 1$. The seed for the RNG used in the Monte-Carlo simulation is given by the current system clock time multiplied by a scalar constant, which is recorded in a log-file so that the results can be replicated if necessary.

The general framework for a Monte-Carlo simulation is presented in Fig. 3-8. The

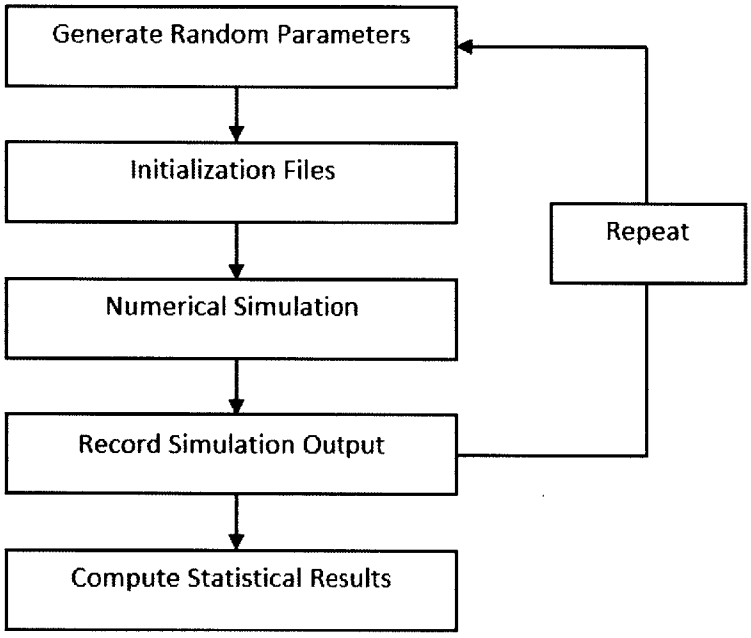


Figure 3-8: Monte-Carlo Framework

main steps for each iteration of the Monte-Carlo simulation are: (1) generate random parameters, (2) initialize system parameters with MATLAB initialization scripts, (3) execute numerical simulation (Simulink), and (4) record simulation output. These steps are repeated until the desired number of iterations has completed and then the statistical results are computed. In order to size the CMG system, a representative profile of missions was designed. The four missions are listed below:

1. Single-Axis Slew Maneuver

2. Solar Array Inspection (ISS)

3. Rescue of Incapacitated Crew-member (ISS)

4. Surface Sampling (Asteroid)

The four missions were chosen to represent realistic situations that an EVA crew-member could experience during a given EVA. The setting for all of the missions takes place around the ISS or on the surface of an asteroid (Itokawa has been chosen as the representative asteroid). It should be noted that for each of the missions there will be two phases to the Monte-Carlo simulation: an initial bounding phase and the actual Monte-Carlo simulation. During the initial bounding phase, the mission profiles will be run twice: once with the RCS-only control mode and once with the blended control mode using oversized CMGs. Running these initial missions will provide an upper and lower bound on the fuel consumption profile of each mission. Since the CMGs do not consume any propellant, the system fuel consumption when using oversized CMGs (no desaturation maneuvers) will be the minimum amount of fuel necessary to complete the given mission. On the other hand, using only the RCS thrusters for both translational and attitude control will provide the maximum amount of fuel consumed.

Mission 1 is a single-axis slew maneuver of 30°. The slew maneuver is about the astronaut's $-\hat{e}_{3B}$ axis in the body local coordinate frame $\mathcal{F}_{\mathrm{B}}$ (Fig. 1-2). The total duration of the mission is 5 minutes; however, the target tracking algorithm positions the crew-member in a minimum amount of time. The additional mission duration provides an indication of how well the ACS can maintain the vehicle's attitude. The translational commands for the mission are to remain in a fixed location while the system slews to its new heading. Since the thrusters have to occasionally fire to null out any accelerations introduced by the gravitational field near the ISS, one would expect the amount of fuel consumed to be extremely small. Fig. 3-9 shows the initial and final states for Mission 1.

This mission was chosen to represent a simple maneuver that would be encountered during most EVA operations or exploration activities. Furthermore, this mission was used for testing the desaturation algorithm since it requires a large change in angular

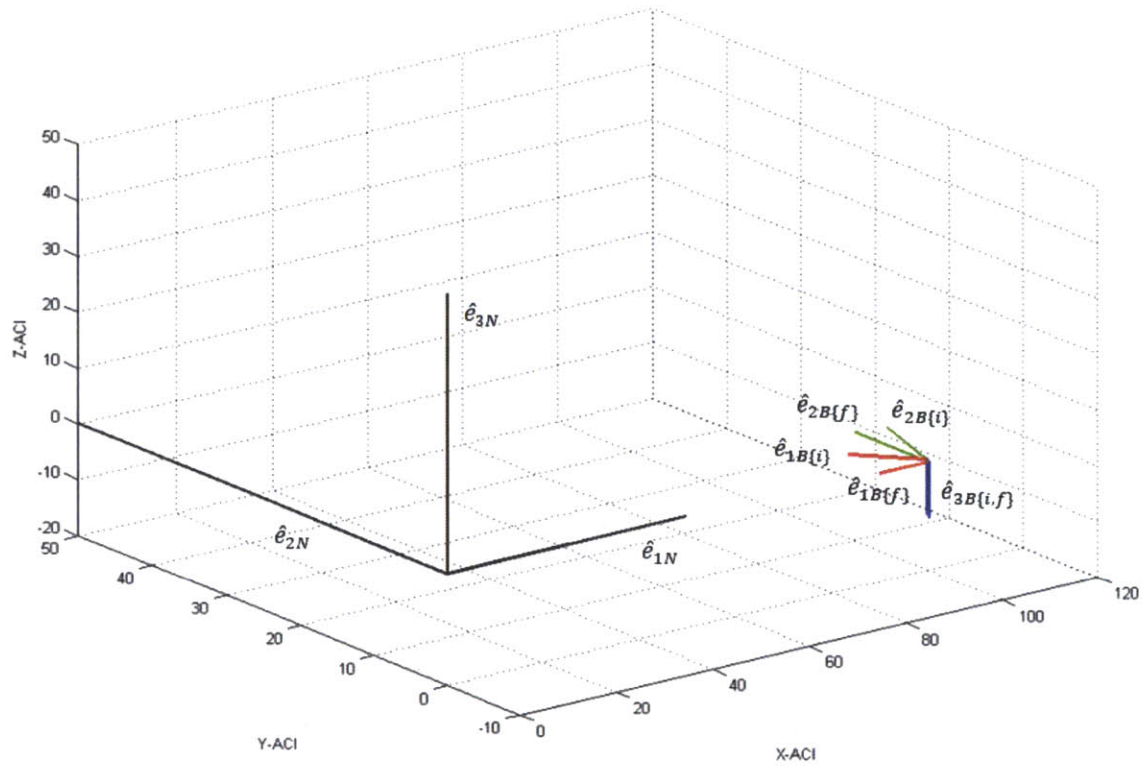


Figure 3-9: Mission 1: Single-Axis Slew

momentum about a particular axis. The crew-member starts with an initial attitude error and a zero initial velocity and must turn towards the desired pointing direction. The black coordinate frame is the Asteroid Centered Inertial (ACI) frame ($\mathcal{F}_{\mathrm{N}}$) and the other coordinate frame is the MAJIC body coordinate frame ($\mathcal{F}_{\mathrm{B}}$). The crew-member is attempting to align his/her line of sight (LOS) vector $\hat{e}_{1B}$ with the ACI frames $-\hat{e}_{1N}$ such that the crew-member would be orienting him/herself with the center of the inertial coordinate frame.

Mission 2 is designed to replicate an EVA in which the ISS's solar arrays are being inspected. The dimensions of the trajectory are based on the actual ISS solar array dimensions. The duration of the mission is 10 minutes. Given the amount of time necessary to reach the solar arrays, this mock-mission is not feasible given the current technology and devices available to EVA crew-members. During the EVA, the crew-member will translate along the length of the solar array inspecting for any disturbances and/or damage. The MAJIC system controls the vehicle's attitude such

that the crew-member has a clear LOS to the center of the solar array during the entire inspection. Fig 3-10 shows the baseline trajectory for the solar array inspection mission. The black arrows indicate the crew-members direction of travel along the solar array.

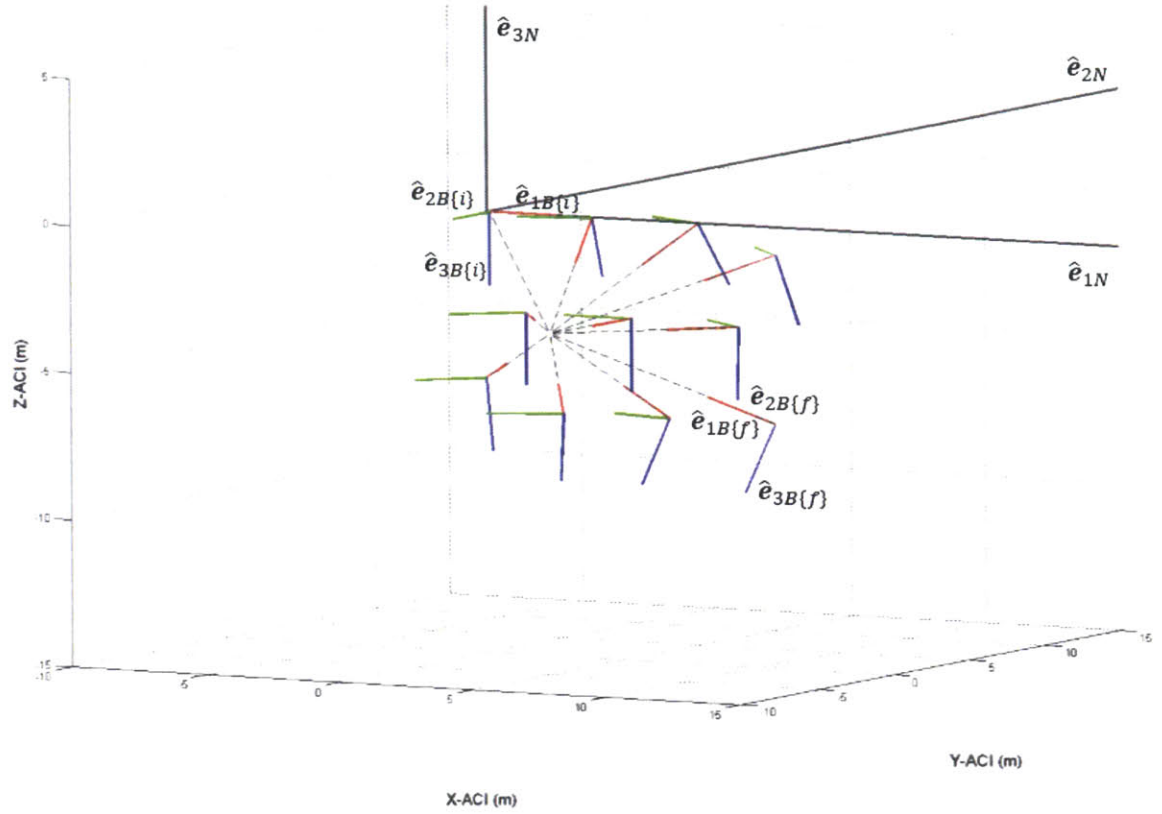


Figure 3-10: Mission 2: Solar Array Inspection (ISS)

The length and width of the solar arrays are 10.7 m and 3.4 m, respectively. The mission begins with the crew-member at the beginning of the solar array. The crew-member translates along the array until reaching the end and then translates along the width of the array before translating back towards the ISS. This pattern is repeated again until the crew-member reaches the end of the solar array.

Mission 3 simulates the rescue of an incapacitated crew-member. The baseline scenario for this mission is that during an EVA, a crew-member becomes impaired and begins floating away from the ISS. The duration of the mission is 23 minutes. In order to replicate the environment around the ISS, there is no gravitational force acting on the crew-members during the mission. Given the current state of the SAFER system,

this would be an extremely difficult task because of the system's small amount of $\Delta \mathbf{V}$. Additionally, the combined mass properties of the incapacitated crew-member and the original system would create an offset in the combined system's center of mass (COM). The MAJIC system will assist with this particular EVA mission because of the reduced fuel consumption, which provides more translational authority, and improved attitude stability, which will ensure that the crew-member can successfully navigate back to the air-lock. The mission begins when the first crew-member reaches the second crew-member and is beginning to make his/her way back to the ISS. In order to make the mission more realistic, the crew-member must avoid a cylindrical section of the ISS when returning to the air-lock.

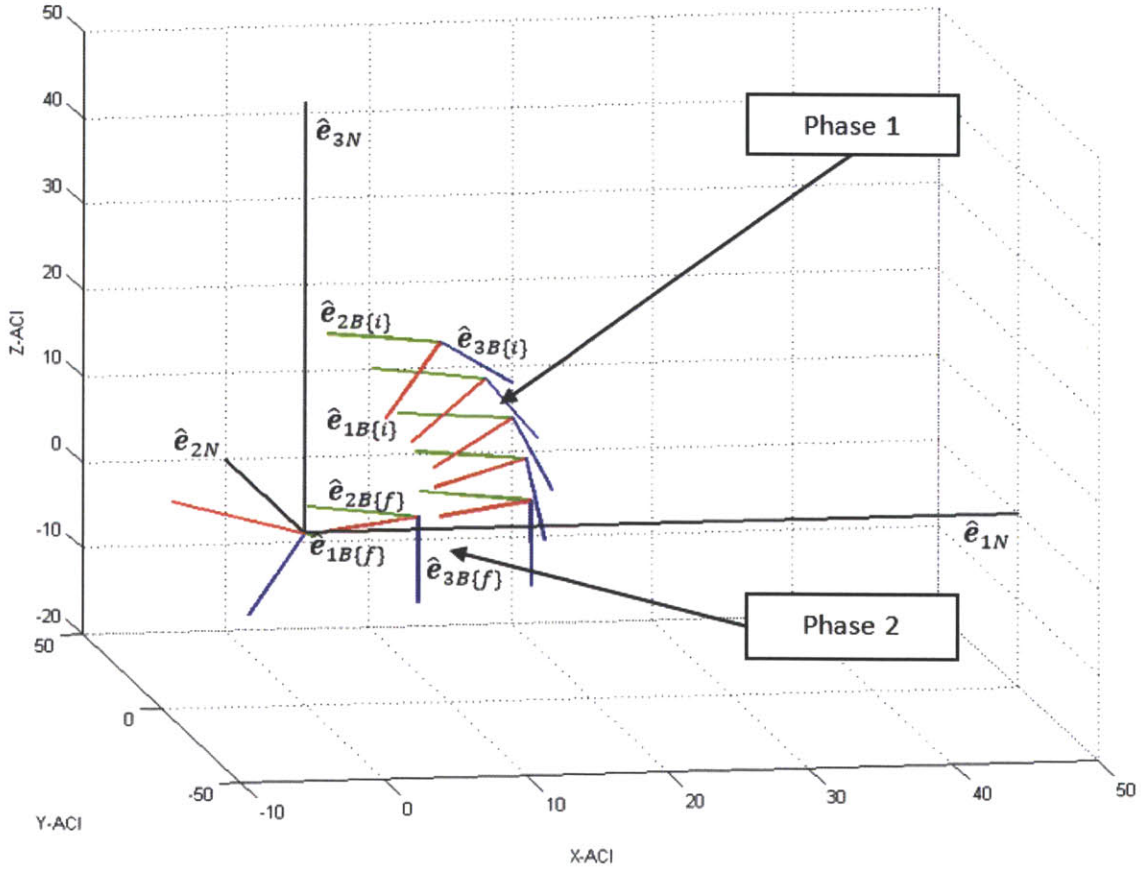


Figure 3-11: Mission 3: Incapacitated Crew-Member Rescue (ISS)

Fig. 3-11 shows the trajectory of the two crew-members. The mission's profile is composed of two distinct sections: (1) translating around the cylindrical section in a circular arc and (2) linearly translating towards the air-lock. The crew-member's

LOS, $\hat{\mathbf{e}}_{1B}$ in the MAJIC system's body-fixed coordinate frame, is directed toward the air-lock for the entirety of the mission. This ensures that the crew-member performing the rescue can navigate back towards the air-lock safely.

Mission 4 is a representative sampling mission on the surface of an asteroid[3]. The duration of the mission is 50 minutes and consists of several phases: (1) approach, (2) orbit raising, (3) orbiting, and (4) sampling. The phases can be seen labeled in Fig. 3-12. This mission is intended to simulate the CMG performance in the presence of a non-zero gravitational field during which an EVA crew-member must leave the main spacecraft and use the MAJIC system to inspect and sample the asteroid's surface. The non-zero gravitational field provides a constant disturbance the CMGs must account for and the approach and orbit raising phases involve significant RCS thruster firing which would perturb the crew-members pointing. Fig. 3-13 shows the same mission profile from within EDGE.

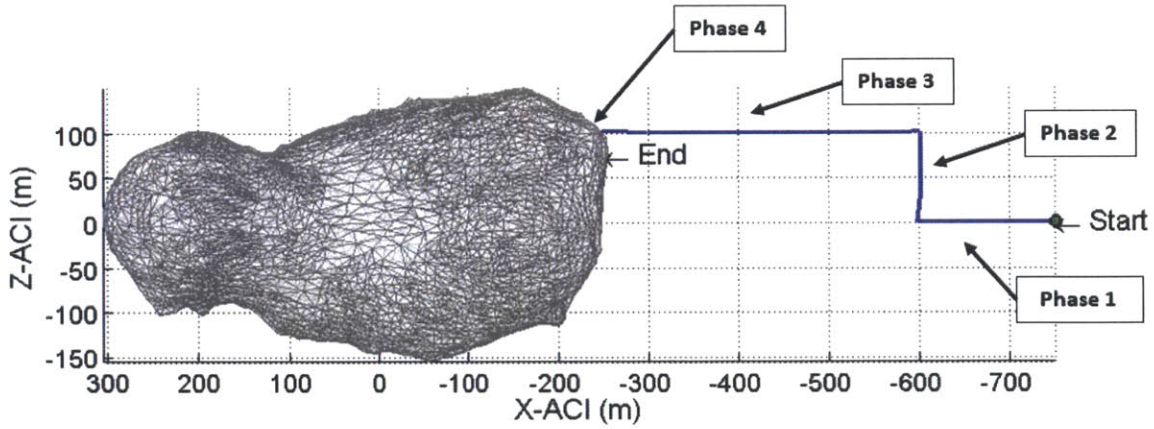


Figure 3-12: Mission 4: Asteroid Surface Sampling

3.3.1 Simulation Parameters

In order to initially determine the optimal CMG sizes for the various missions, an empirical sizing tool was developed. The sizing tool uses commercially available CMGs that currently exist such as the Honeywell M50 and Honeybee Microsat CMG.

[3]The particular asteroid is not important; however, Itokawa was used because extensive data and models were previously constructed of the asteroid for another project

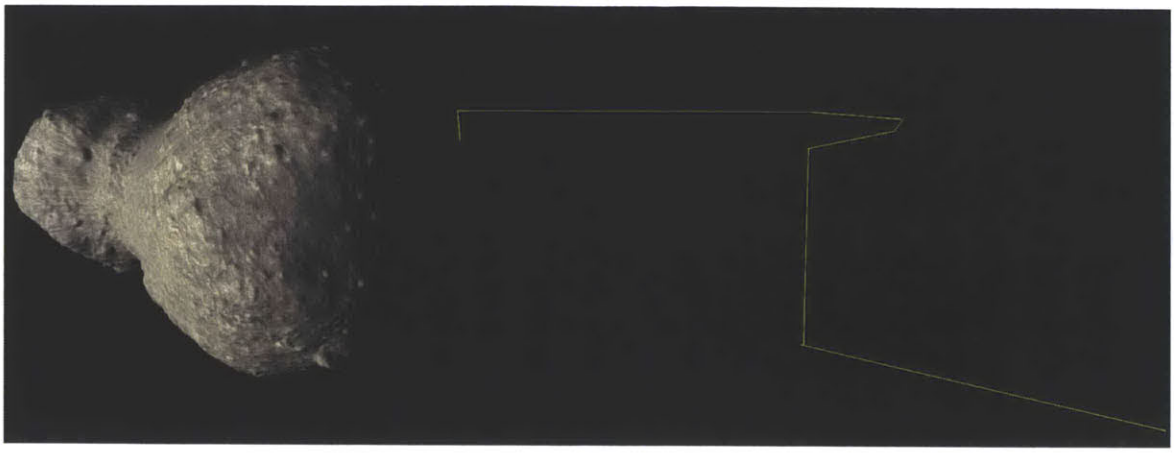


Figure 3-13: Mission 4: Asteroid Surface Sampling (EDGE)

Mass, angular momentum, torque, and power data was compiled for a total of 13 different CMG models. Once the data had been collected, a MATLAB function was designed to take as inputs a parameter (e.g. mass, angular momentum, torque or power) and value. The function then performs one-dimensional piece-wise cubic interpolation on the commercial data to output the other parameter values based on the input. Figs. 3-14 - 3-16 shows a sample output of the empirical sizing program with the input being a CMG mass of 10 kg.

The blue data points represent the commercially available CMG specifications and the red dot represents the interpolated CMG parameters. The empirical sizing tool would be used to generate parametric CMG data given a randomly generated mass. However, after testing the interpolation for a few sample cases, it was immediately apparent that the output sizes were skewed toward larger CMG sizes than desired. The imbalance in the data is caused by the greater availability of information regarding large CMGs (Mass $>$ 5 kg).

In order to generate data for small-scale CMGs, another method had to be developed that would not tailor to large-scale CMGs. For the MAJIC system, the parameters of interest involve sizing the CMG architecture for an ensemble of potential mission trajectories. In order to parametrically vary the CMG size parameters, four independent variables were chosen: rotor inertia (I_{rotor}), rotor angular rate (ω_{rotor}), maximum gimbal rate ($\dot{\delta}_{\text{max}}$), and flywheel density (ρ). By making careful assump-

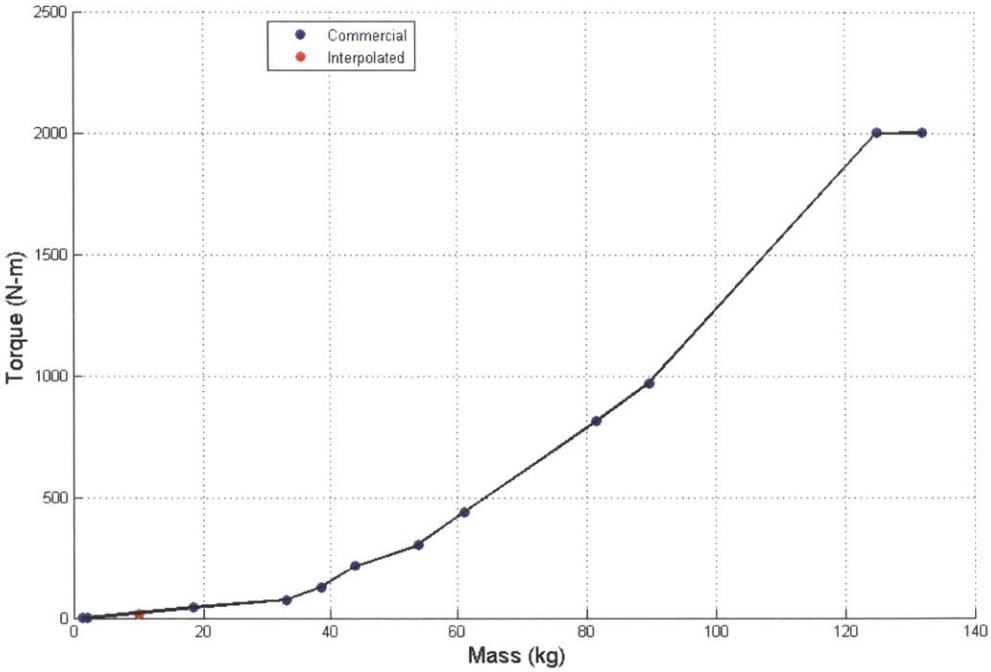


Figure 3-14: CMG Mass (kg) vs. Torque (N-m)

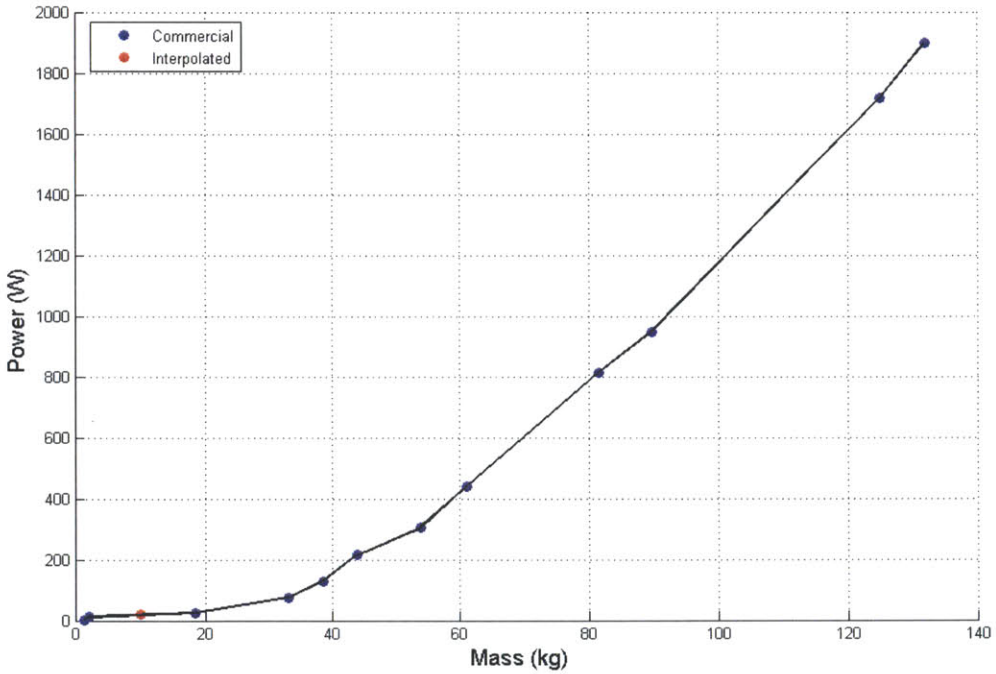


Figure 3-15: CMG Mass (kg) vs. Power (W)

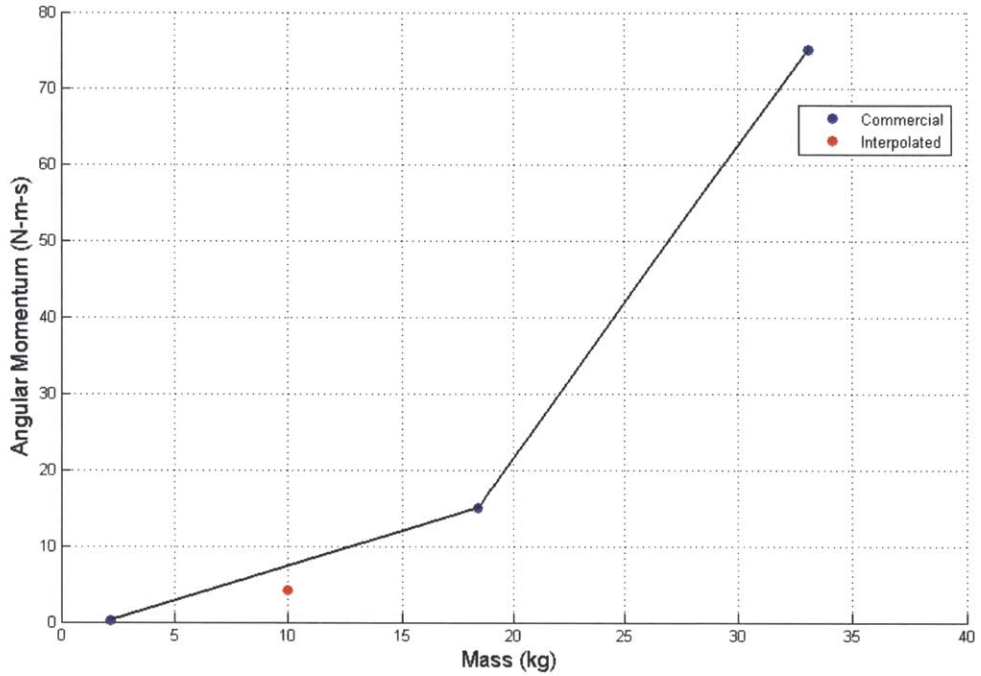


Figure 3-16: CMG Mass (kg) vs. Angular Momentum (N-m-s)

tions about the CMG rotor dimensions and shape, these four parameters can be used to calculate all of the necessary data for the CMG architecture. The following outline shows the a priori assumptions made about the CMGs and the constitutive equations relating the independent and dependent CMG parameters:

CMG parameter calculation:

- Given: $\mathrm{I}_{\mathrm{rotor}}$, ω_{rotor}, $\dot{\delta}_{\mathrm{max}}$, ρ

- Assumptions: cylindrical flywheel $\mathrm{I} = \frac{1}{2}mr^2$, $r = 2h$, $3m_{\mathrm{rotor}} = m_{\mathrm{CMG}}$

$$h = \mathrm{I}_{\mathrm{rotor}}\omega_{\mathrm{rotor}} \tag{3.5}$$

$$|\tau_{\mathrm{max}}| = h \times \dot{\delta}_{\mathrm{max}} \tag{3.6}$$

$$
\begin{aligned}
\rho &= \frac{m}{\mathrm{V}} \\
\rho &= \frac{m}{\pi r^2 h} \\
\rho &= \frac{m}{\pi r^2 \frac{r}{2}} \\
\rho &= \frac{2m}{\pi r^3} \\
r &= \left(\frac{2m}{\pi\rho}\right)^{\frac{1}{3}}
\end{aligned}
\tag{3.7}
$$

$$
\begin{aligned}
\mathrm{I_{rotor}} &= \frac{1}{2} m r^2 \\
\mathrm{I_{rotor}} &= \frac{1}{2} m \left(\frac{2m}{\pi\rho}\right)^{\frac{2}{3}} \\
m &= \left(\frac{\pi\rho(2\mathrm{I_{rotor}})^{\frac{3}{2}}}{2}\right)^{\frac{2}{5}}
\end{aligned}
\tag{3.8}
$$

The bounds for the given parameters (I_{rotor}, ω_{rotor}, $\dot{\delta}_{\mathrm{max}}$, ρ) were either derived from actual CMG data gathered or were chosen such that the CMG had the desired torque and angular momentum bounds. The upper and lower bounds are shown in Table 3.1. The $\dot{\delta}_{\mathrm{min}}$ and $\dot{\delta}_{\mathrm{max}}$ bounds were chosen by reviewing typical gimbal speeds for commercial CMG gimbal motors.

The flywheel density (ρ) is assumed to be made of Aluminum T6-6061, Steel, Brass (60/40) or Tungsten and the corresponding values in Table 3.1 relate the material density to the material, respectively. Varying the flywheel material provides an extra degree of freedom in sizing the CMGs as opposed to only using the rotor inertia and spin speed. Also, it completely defines the set of equations necessary to calculate

Table 3.1: Monte-Carlo Simulation Parameter Bounds

Parameter	Lower Bound	Upper Bound	Units
I_{rotor}	9.1189e-05	1.8238e-04	kg-m^2
ω_{rotor}	10,000	50,000	rpm
$\dot{\delta}_{\text{max}}$	25	100	rpm
ρ	{ 2700, 7850, 8520, 19600 }		kg/m^3

the remaining dependent CMG parameters (Eqs. 3.5-3.8). The rotor's inertia were calculated to ensure that the torque output of the CMGs was between: $0.25 \leq \tau \leq 10$ N-m. This condition ensured that only small satellite CMGs would be considered during the Monte-Carlo simulations.

A performance score J is derived to evaluate the CMGs performance on a particular mission. The performance score for the Monte-Carlo case studies is:

$$\min J = \sum_{i=1}^{n} W_i p_i \tag{3.9}$$

where W_i are the performance weights, p_i are the performance metrics, and n is the number of metrics being evaluated. The performance metrics of interest are:

- $p_1 \rightarrow$ total propellant consumption
- $p_2 \rightarrow$ peak CMG power consumption
- $p_3 \rightarrow$ total CMG energy consumption
- $p_4 \rightarrow$ RMS[4] attitude error

Total propellant consumption is important because it provides a measure of how much additional propellant is used for desaturation maneuvers. p_2 and p_3 can be used to size the electrical subsystem required for powering the CMGs. Depending on the peak power consumption and total energy consumption, batteries can also be investigated

[4] $x_{RMS} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} |x_i|^2}$

for the electrical subsystem. p_4 determines whether or not the particular CMG size can achieve the desired pointing precision. The performance metrics are scaled by the performance weights and summed to calculate J. The performance weights can be used as knobs to adjust the relative importance of one metric over another. For instance, if all four performance metrics are just as important as one another, then the performance weights would all be equal $W = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$. However, if the first and fourth performance metrics were twice as important as the second and third, then the performance weights might be defined as $W = \begin{bmatrix} 2 & 1 & 1 & 2 \end{bmatrix}$. After all Monte Carlo trials have been completed, the minimum score provides the optimal CMG size for the particular mission. Depending upon the particular choice of W_i, different CMG sizes could be optimal for the same mission. Therefore, it is extremely important to determine the correct weighting for the performance metrics before choosing an optimal CMG size.

Since the performance metrics have different units, the results are normalized before computing the final performance scores. If the theoretical maximum value is not known ahead of time for a particular metric, the normalization can be done once all of the Monte Carlo trials have been finished. Eq. 3.10 shows how to normalize a particular metric assuming all trials have been finished and the data is available:

$$p_{1,i} = \frac{p_{1,i}}{\max\{p_1\}} \quad \forall \quad i = 1, \ldots, N \tag{3.10}$$

where N is the total number of Monte-Carlo simulations. Another useful simplification can be used to ensure that the performance score J is always bounded by $(0, 1]$. Eq. 3.11 shows how the weights can be chosen to ensure that the maximum possible score is 1 and the theoretical minimum is 0; however, achieving a score of 0 is improbable since it would imply that no propellant was consumed, the CMGs consumed no power and/or energy, and there was no attitude error.

$$W_i = \frac{1}{n} \quad \forall \quad i = 1, \ldots, n \tag{3.11}$$

where n is the number of metrics being evaluated.

3.3.2 Simulation Results

The results for the Monte-Carlo simulations are displayed below for each of the four missions. The five CMGs sizes with the lowest (i.e. best) normalized score are presented in tabular form, along with the corresponding normalized score. The plots representing the absolute attitude error (rad), fuel consumption (kg), power (W) and total energy (J) versus time (s) for each of the missions are in Appendix A.

Mission 1 - Single-Axis Slew Maneuver

The results from the first mission are summarized in Table 3.2. The CMG with the lowest performance score has a mass of 1.1 kg; however, due to the low fuel consumption requirement of this mission, the second lowest CMG is chosen as the optimal. This particular mission's low fuel consumption implies that a CMG with a smaller mass is more desirable to ensure that the CMGs can buy their way on-board. The second set of CMG parameters have the minimum overall mass of the CMGs with minimum performance scores.

Table 3.2: Mission 1 - Top 5 CMG Parameters

$\mathbf{I_{rotor}}$	$\boldsymbol{\omega_{rotor}}$	$\boldsymbol{\dot{\delta}_{max}}$	$\boldsymbol{\rho}$	**m**	**h**	$\boldsymbol{\tau}$	**J**
kg-m^2	rpm	rpm	$\frac{\text{kg}}{\text{m}^3}$	kg	N-m-s	N-m	-
1.74×10^{-4}	10470	29	7850	1.09	0.19	0.58	2.15×10^{-1}
9.37×10^{-5}	21638	53	2700	0.49	0.21	1.18	2.16×10^{-1}
1.22×10^{-4}	12800	46	8520	0.91	0.16	0.79	2.23×10^{-1}
1.77×10^{-4}	15205	27	7850	1.10	0.28	0.81	2.23×10^{-1}
1.16×10^{-4}	13267	51	2700	0.56	0.16	0.85	2.24×10^{-1}

The optimal CMG has a theoretical mass of 0.50 kg and a flywheel speed of 22000 rpm. The maximum gimbal rate is 50 rpm, which results in an angular momentum magnitude of 0.21 N-m-s and a output torque of 1.20 N-m. The high gimbal rate allows the relatively small CMG to produce large output torques, which are critical for rapid slew maneuvers.

Mission 2 - Solar Array Inspection (ISS)

The top five CMGs from the ISS solar array inspection mission are presented in Table 3.3. The optimal CMG has a mass of 1.10 kg, a flywheel speed of 14000 rpm and a maximum gimbal rate of 46 rpm. The computed angular momentum is 0.14 N-m-s, which may seem extremely low; however, the attitude maneuvers involved in this particular mission are distributed about all three axes. The fact that the attitude maneuvers are distributed in a more even manner than mission 1, means that the angular momentum requirements aren't as critical. Additionally, the output torque of the optimal CMG is 0.69 N-m.

Table 3.3: Mission 2 - Top 5 CMG Parameters

$\mathbf{I_{rotor}}$	$\boldsymbol{\omega_{rotor}}$	$\boldsymbol{\dot{\delta}_{max}}$	$\boldsymbol{\rho}$	**m**	**h**	$\boldsymbol{\tau}$	**J**
kg-m^2	rpm	rpm	$\frac{\text{kg}}{\text{m}^3}$	kg	N-m-s	N-m	-
9.59×10^{-5}	14216	46	19600	1.10	0.14	0.69	1.86×10^{-1}
1.05×10^{-4}	31534	38	2700	0.53	0.35	1.37	1.94×10^{-1}
1.08×10^{-4}	49838	54	8520	0.85	0.56	3.16	2.03×10^{-1}
1.08×10^{-4}	23567	62	8520	0.85	0.27	1.74	2.06×10^{-1}
1.62×10^{-4}	18142	55	19600	1.51	0.31	1.77	2.07×10^{-1}

Mission 3 - Crew-Member Rescue (ISS)

The five optimal performance scores for the the ISS crew-member rescue mission are presented in Table 3.4. This mission involved translating from an arbitrary point on the ISS to the air-lock while carry an incapacitated crew-member. The mass properties of the additional crew-member were added to the MAJIC system to form an aggregate system consisting of the actual MAJIC hardware and two crew-members. The CMG with the best performance score has an overall mass of 1.48 kg, a flywheel speed of 19000 rpm and a maximum gimbal rate of 52 rpm. This particular CMGs flywheel is composed of Tungsten which has an extremely high density (ρ); therefore, with a relatively low flywheel speed it is capable of producing large output torques.

The angular momentum magnitude of the CMG is 0.31 N-m-s and the output torque is 1.68 N-m, which is the largest of all the missions analyzed. Physically, this makes sense since the composite system and mass properties of this mission would require larger torques to perform the same slew maneuvers and maintain the same pointing precision.

Table 3.4: Mission 3 - Top 5 CMG Parameters

$\mathbf{I_{rotor}}$	$\boldsymbol{\omega_{rotor}}$	$\boldsymbol{\dot{\delta}_{max}}$	$\boldsymbol{\rho}$	**m**	**h**	$\boldsymbol{\tau}$	**J**
kg-m^2	rpm	rpm	$\frac{\text{kg}}{\text{m}^3}$	kg	N-m-s	N-m	-
1.57×10^{-4}	18926	52	19600	1.48	0.31	1.68	1.75×10^{-1}
1.06×10^{-4}	13151	68	8520	0.84	0.15	1.04	1.77×10^{-1}
9.53×10^{-5}	15847	52	2700	0.50	0.16	0.85	1.78×10^{-1}
1.45×10^{-4}	26639	26	2700	0.64	0.41	1.12	1.79×10^{-1}
1.51×10^{-4}	20419	59	8520	1.04	0.32	1.99	1.80×10^{-1}

Figs. A-9 - A-12 plot the attitude error (rad), fuel consumption (kg), power (W) and energy (J) versus time (s) for the top five CMG parameters generated. It can be seen that the optimal CMG (iteration 45) maintains a very low attitude error for the entire mission trajectory. Also, the CMGs fuel consumption is the second lowest compared to the other top CMG sizes. The power and energy plots also show that this particular CMG size minimizes power and energy consumption during the crew-member rescue. Given the CMGs parametric model, the flywheel density is strongly connected to the overall efficiency of the final design. The higher density allows for lower flywheel speeds, while still maintaining the high angular momentum magnitude.

Mission 4 - Surface Sampling (Asteroid)

The Monte-Carlo results for the asteroid surface sampling mission are shown in Table 3.5. It can be shown that none of the small-scale CMGs were able to complete the mission successfully since each of the top performing CMGs remained in desaturation mode for the duration of the mission. Fig. A-13 illustrates this because none of the small-scale CMGs can reach zero steady-state attitude error. Additionally, each

set of CMG parameters appear to oscillate at a vary small attitude error, which is characteristic of the dead-band associated with the RCS thrusters only control mode. The CMG with the optimal performance score has a mass of 0.98 kg, flywheel speed of 25000 rpm and a maximum gimbal rate of 49 rpm. Using the relationships developed in Eqs. 3.5 - 3.8, the calculated angular momentum magnitude is 0.35 N-m-s and the output torque is 1.81 N-m.

Table 3.5: Mission 4 - Top 5 CMG Parameters

$\mathbf{I_{rotor}}$	$\boldsymbol{\omega_{rotor}}$	$\boldsymbol{\dot{\delta}_{max}}$	$\boldsymbol{\rho}$	**m**	**h**	$\boldsymbol{\tau}$	**J**
kg-m^2	rpm	rpm	$\frac{\text{kg}}{\text{m}^3}$	kg	N-m-s	N-m	-
1.37×10^{-4}	24739	49	8520	0.98	0.35	1.81	1.52×10^{-1}
1.34×10^{-4}	13725	58	8520	0.97	0.20	1.16	1.55×10^{-1}
1.64×10^{-4}	35555	56	7850	1.06	0.61	3.60	1.56×10^{-1}
9.52×10^{-5}	12428	58	8520	0.79	0.12	0.76	1.57×10^{-1}
1.44×10^{-4}	13967	58	8520	1.01	0.21	1.29	1.67×10^{-1}

The optimal CMG size for this particular mission makes physical sense because of the larger angular momentum magnitude and output torque. The gravitational effect of the asteroid was modeled which meant that throughout the mission the MAJIC ACS had to correct for gravitational torques to maintain the pointing requirements. The pyramidal CMG architecture ($\beta = 54.7°$) defines the maximum angular momentum envelope and significantly enhances the maximum angular momentum per axis. This effect is quantified by the CMG sizing coefficients in Table 2.3. The CMG sizing coefficients define the three-dimensional saturation envelope of the pyramidal architecture.

3.4 Final Human-in-the-Loop Test Results

After performing the Monte-Carlo study with the newly implemented momentum desaturation algorithm to appropriately size the CMGs, another round of piloted evaluations was conducted at NASA JSC. The structure of the evaluations was very

similar to the first round of evaluations described in Sec. 3.2.3. Once again, the subjects were instructed to fly from way-point to way-point; however, the task they were asked to perform was modified. The subjects input rotational commands to align themselves with the blue line emanating from the target cone while the translational commands were delivered by the ACS guidance solution. Also, instead of only flying two ACS (RCS only and oversized CMGs), the subjects operated three ACS designs: (1) RCS thrusters only, (2) RCS thrusters with oversized CMGs and (3) RCS thrusters with small-scale CMGs optimally sized for mass from the Monte-Carlo study. A total of 10 subjects flew the obstacle course.

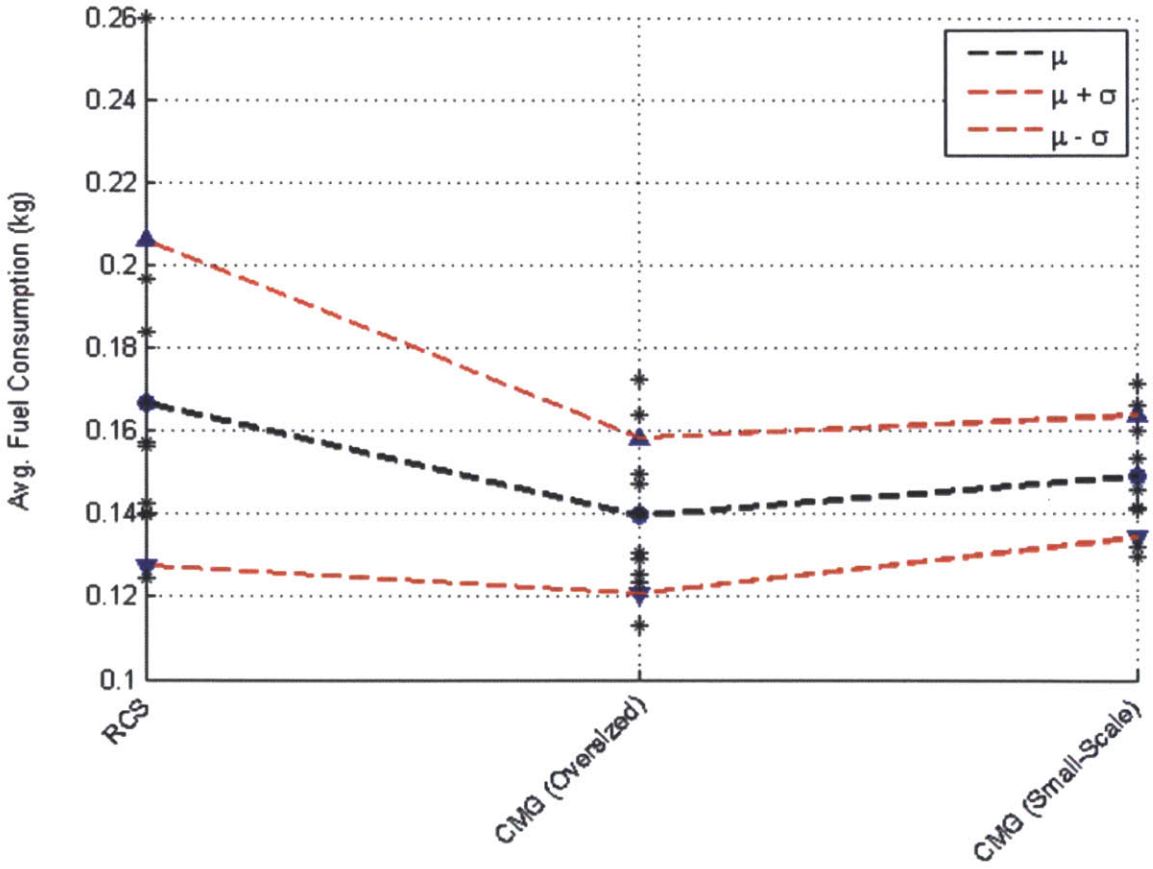


Figure 3-17: HITL Evaluation - Subject Fuel Consumption Data

Fig. 3-17 shows the fuel consumption data from the 10 subjects, as well as the mean fuel consumption and one-standard deviation values for each ACS. It is important to note that since the subjects were only controlling rotational commands, the deviation in fuel consumption is much smaller than the first round of evaluations,

shown in Fig. 3-7. The results from the asteroid mission validate the notion that adding CMGs results in less fuel consumption per mission. The fuel consumption of the thrusters only ACS was the highest followed by the small-scale CMGs and finally the oversized CMGs. The results follow the general intuition since the small-scale CMGs encounter momentum saturation more frequently during the evaluation, directly resulting in a more frequent activation of the RCS thrusters to hold the attitude during desaturation.

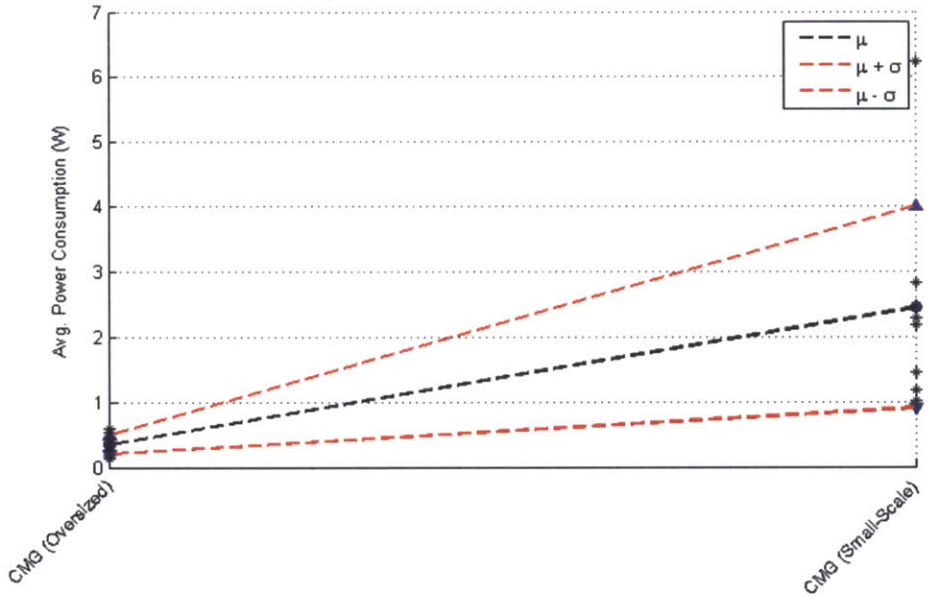


Figure 3-18: HITL Evaluation - Subject Power Data

Figs. 3-18 - 3-19 plot the CMGs power and energy data during the piloted evaluations, respectively. The small-scale CMGs consume more power and energy than the oversized CMGs which makes physical sense since the smaller CMGs must work harder to achieve the same desired torque outputs as the larger CMGs. The average power consumption is approximately 2.45 Watts and will be used for sizing the MAJIC system's electrical subsystem. The maximum power consumed by the small-scale CMGs is 6.23 Watts and can be used a worst case upper bound for sizing the MAJIC batteries.

It is important to note that the physical parameters used for the final round of HITL evaluations were obtained from the Monte-Carlo analysis of the asteroid sur-

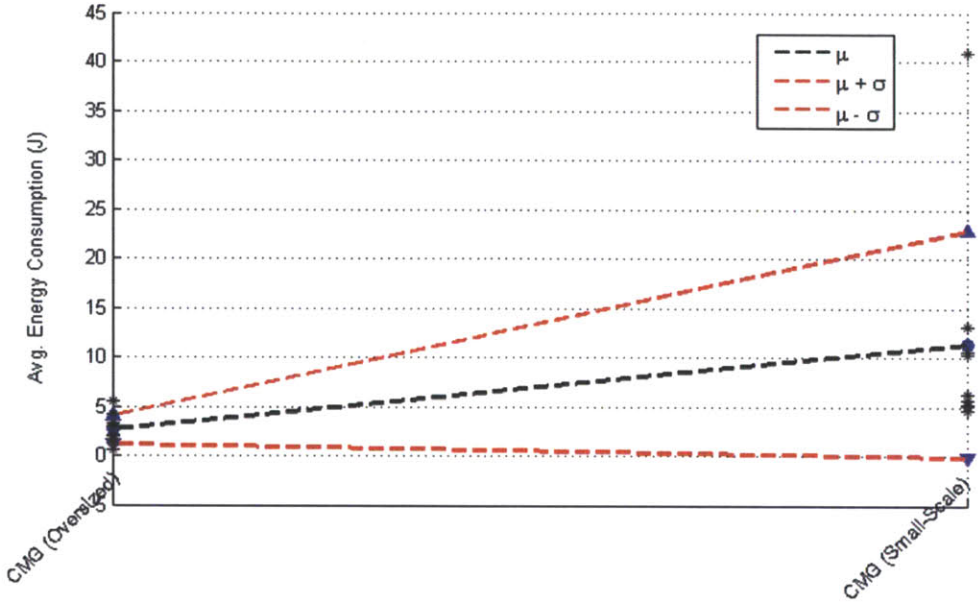


Figure 3-19: HITL Evaluation - Subject Energy Data

face sampling mission. These CMG parameters were selected because of the similarity between the HITL and asteroid sampling mission. The pilots were given questionnaires, which asked which control mode he/she preferred and the relative level of difficulty associated with completing the task given the varying control modes. The vast majority of the pilots preferred the RCS thrusters only and RCS thrusters and oversized CMG control modes over the RCS thrusters and small-scale CMG control mode. This preference can be directly correlated to the amount of time spent in desaturation mode for the small-scale CMGs. Fig. 3-20 shows the time spent in desaturation for each of the 10 subjects. The average time spent in desaturation for the HITL missions was approximately 2.67 minutes. However, the maximum and minimum time spent in desaturation was 8.87 and 0.45 minutes, respectively. The large variance associated with the time spent in desaturation can be attributed to the differences in how aggressively the pilot corrected for attitude errors. For instance, if a pilot attempted to correct a perceived attitude error by slewing the MAJIC system about one axis, then the CMGs could saturate depending on the magnitude of the CMGs angular momentum vector.

Since the majority of the human subjects were disconcerted by the amount of

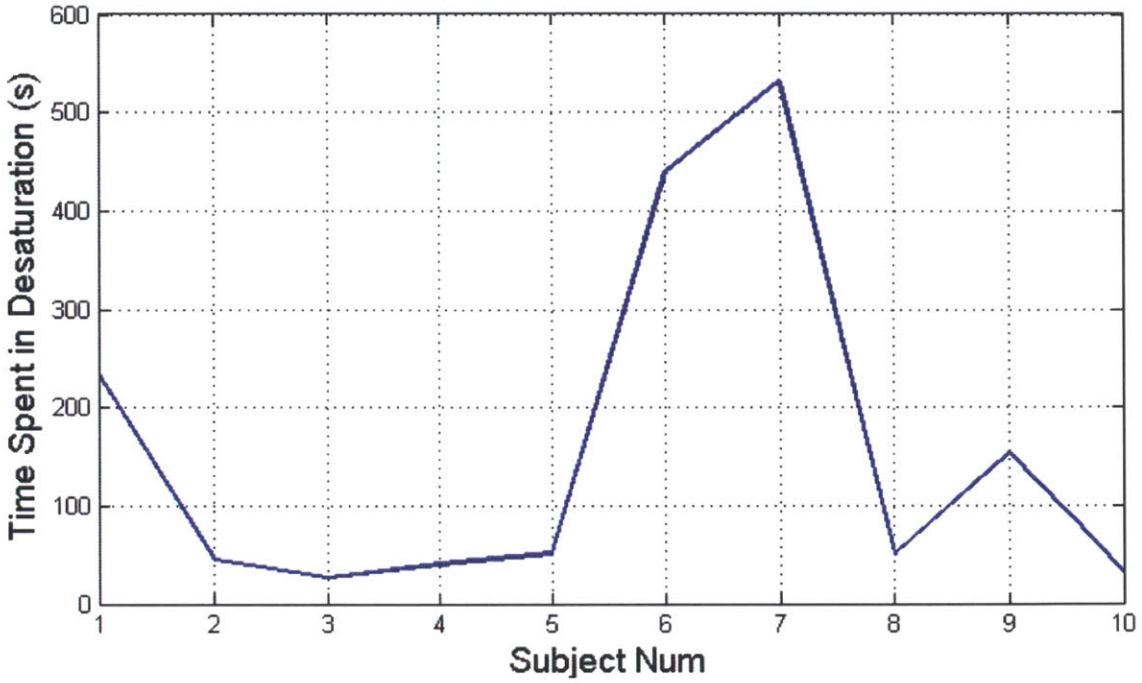


Figure 3-20: HITL Evaluation - Time Spent in Saturation (s) vs. Subject Number

time spent in desaturation mode and the lack of control authority during this time, another Monte-Carlo analysis was conducted with larger-scale CMGs (Sec. 3.5). The same parameters from the previous Monte-Carlo analysis were used (see Table 3.1), except the rotor's inertia (I_{rotor}) was adjusted such that a single CMGs output torque was bounded by $5 \leq \tau \leq 25$ N-m. Increasing the minimum output torque from 0.25 N-m to 5 N-m and the maximum torque from 10 N-m to 25 N-m ensures that only larger-scale CMGs are considered. The larger CMGs will result in less time spent in desaturation mode and will allow human pilots to complete EVA's without saturating the CMGs, while also minimizing the mass associated with more massive, commercially available CMGs.

3.5 Final Monte-Carlo Case Study

The results of the final round of HITL testing at JSC (Sec. 3.4) show that the previous bounds on the CMGs physical parameters were unrealistic given the duration and frequency of saturation events encountered by the human subjects. These bounds

were derived from the notion that the CMGs has to "buy" their way on-board. This meant that the addition of the CMGs and batteries had to weigh less than the fuel savings associated with using a blended ACS, as opposed to a thrusters only ACS. Extending the bounds to include larger CMGs will result in simulation cases with less saturation events. Although the mass of the larger CMGs will be greater than the mass savings associated with switching ACS, the mass savings over the course of the MAJIC system's lifetime will far surpass the additional mass of the larger CMGs.

The mission for the final Monte-Carlo analysis is the same as the asteroid surface sampling mission (i.e. Sec. 3.3 - Mission 4). The simulation results are shown in Appendix B. Furthermore, Fig. B-2 shows an enlarged plot of the absolute attitude error versus time for the top performing CMGs. Table 3.6 shows the top five CMG configurations with the lowest performance scores.

Table 3.6: Top 5 CMG Parameters ($5 \leq \tau \leq 25$ N-m)

$\mathbf{I_{rotor}}$	$\boldsymbol{\omega_{rotor}}$	$\boldsymbol{\dot{\delta}_{max}}$	$\boldsymbol{\rho}$	$\mathbf{m}$	$\mathbf{h}$	$\boldsymbol{\tau}$	$\mathbf{J}$
kg-m^2	rpm	rpm	$\frac{\text{kg}}{\text{m}^3}$	kg	N-m-s	N-m	-
5.70×10^{-4}	49073	39	7850	2.23	2.93	12.00	5.02×10^{-2}
2.81×10^{-3}	20763	32	7850	5.80	6.10	20.55	5.30×10^{-2}
9.58×10^{-4}	25337	27	2700	1.98	2.54	7.24	5.31×10^{-2}
1.57×10^{-3}	31282	33	2700	2.67	5.15	18.03	5.33×10^{-2}
5.23×10^{-3}	12502	33	2700	5.49	6.85	23.76	5.35×10^{-2}

The enlarged attitude error plot shows that the only two CMGs able to completely null the attitude error are the second and third scoring configurations. The other configurations maintain the attitude error at approximately 3×10^{-5} rad. The top performing CMG is chosen for the optimal design since it also minimizes the fuel consumption throughout the mission duration. The CMG parameters generated during the Monte-Carlo mission are ideal for an asteroid exploration and/or sampling mission since it has the highest flywheel speed ($\sim$ 50000 rpm) and density (7850 kg-m^2) of the top scoring CMGs. Additionally, since these CMG parameters produced the lowest (i.e. best) performance score of all the Monte-Carlo missions run, this set

of CMG parameters is chosen as the default configuration for the MAJIC system.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 4

MAJIC System Design

4.1 Overall System Description

The MAJIC system derives some of its basic components from the baseline NASA Jetpack design. However, the main fundamental differences are in the ACS and electrical subsystem, which are defined in Sec. 4.2 and 4.3, respectively.

The addition of four CMGs in a pyramidal architecture ($\beta = 54.7°$) and new electrical components to handle the power and energy needs of the CMGs means that structural changes have to be made to the baseline Jetpack design. Fig. 4-1 is a model of how the upgraded MAJIC system may accommodate the additional mass and volume constraints of the CMGs. The CMG array will be placed inside the suitport housing to ensure that they are protected from the harsh conditions of space, while also protecting the crew-member in the event of a mechanical failure. The batteries and other electrical equipment are stored in the in the bottom rectangular box along with the propellant tanks. The wiring harnesses and other cabling are run through the bottom of the MAJIC system and to the CMGs in the suitport section.

4.2 Attitude Control Subsystem

The finalized ACS design for the MAJIC system utilizes a blended control algorithm for the two sets of actuators on-board: RCS thrusters and CMGs. The thrusters

Figure 4-1: MAJIC System Prototype

will use Tridyne as the propellant because of the increased Isp which will help further extend the mission length of the MAJIC system. The blended control algorithm allocates the rotational $\Delta\boldsymbol{\omega}$ commands to the CMGs and the translational $\Delta\mathbf{V}$ commands to the thrusters. This ensures that the system uses the minimal amount of fuel necessary during the mission. For a mission with no translational component (e.g. orbiting an asteroid), it may or may not be possible to complete the mission without using any propellant depending on whether or not the CMGs saturate. Missions with a translational component must expend fuel since CMGs cannot provide $\Delta\mathbf{V}$ actuation.

The blended control algorithm has desaturation logic built-in, which identifies when a saturation and/or singular state is near, and can either switch to using

thrusters to desaturate the CMGs or use null-motion to steer the CMGs away from the singularity.

The recommended CMG sizes for the MAJIC system are shown below in Table 4.1. These CMG parameters are derived from the simulated Monte-Carlo mission profiles. The chosen parameters are from the larger-scale representative asteroid surface sampling mission. This particular CMG configuration had the overall lowest performance score of all other CMG configurations across all missions. Also, hese CMGs provide a large output torque (12 N-m), as well as a substantial angular momentum ($\sim$ 3 N-m-s), which allows for larger single-axis maneuvers without saturating. The mass penalty associated with incorporating four of these CMGs for use with a pyramidal configuration ($\beta = 54.7°$) is approximately 9 kgs. Although the CMGs cannot directly buy their way on-board for a single mission, over the course of the MAJIC system's lifetime, the benefit of adding the CMGs demonstrates a significant improvement over not incorporating them.

Table 4.1: MAJIC System - Recommended CMG Parameters

$\mathbf{I_{rotor}}$	$\boldsymbol{\omega_{rotor}}$	$\boldsymbol{\dot{\delta}_{max}}$	$\boldsymbol{\rho}$	**m**	**h**	$\boldsymbol{\tau}$
kg-m^2	rpm	rpm	$\frac{\text{kg}}{\text{m}^3}$	kg	N-m-s	N-m
5.70×10^{-4}	49073	39	7850	2.23	2.93	12.00

It should be noted that during the final round of human-in-the-loop testing (Sec. 3.4), several of the pilots mentioned that the desaturation of the small-scale CMGs was unexpected and that they would like an indication of when they were approaching a saturation and/or singularity. The pilots also mentioned that they would feel comfortable using the small-scale CMGs during an actual EVA with the proper training. Training using the MAJIC system will be an extremely important part of preparing the crew-members for use of the MAJIC system. The pilots also mentioned that more education on what control inputs and tasks would likely cause the small-scale CMGs to enter into desaturation mode would be useful in helping the pilots avoid certain

behaviors that might induce saturation or singularity CMG states.

4.3 Electrical Subsystem

The electrical subsystem, in this context, refers to the selection of batteries for the MAJIC blended ACS. Only secondary (rechargeable) batteries are considered when sizing the electrical subsystem. When initially selecting batteries for a space mission, it is important to consider several factors [25]:

- Mission environment (i.e. eclipse period, frequency, temperature, radiation, etc.)
- Mission duration
- Power and energy requirements
- Overhead constraints
  - Mass
  - Volume
  - Budget

The mission environment is extremely important when selecting batteries. If solar powered batteries are used, then the orbital parameters governing the missions are critical to determining how often and when the batteries can be recharged. For non-solar powered batteries, the orbital parameters are not as essential. The thermal operational environment is also important in battery selection. There are a multitude of possible thermal environments in space ranging from the ambient temperature of space to extremely hot temperatures when the batteries remain in the sun's focus. Also, thermal regulatory subsystems can be used to maintain a traditional operating environment; however, that requires additional hardware and software components. For the MAJIC system design, it is assumed that there is no thermal subsystem and therefore the batteries must be able to operate in extreme temperatures.

The duration of the mission refers to the length of a typical EVA which is assumed to be < 7 hours. Although the length of the simulations run have all been under an hour, the electrical and power data can be extrapolated to provide estimates for the battery power and energy requirements. These requirements are imperative to ensuring that the CMGs are sustainable for the duration of the EVAs. The CMGs have a start-up power associated with spinning the flywheel up to speed. The start-up power is heavily dependent upon the physical design of the CMGs and is therefore omitted until a final model and specification has been reached. Once the CMGs have been spun-up, the majority of the power consumption occurs from gimbaling the CMGs to achieve the desired output torques. For battery sizing, the start-up power cost is ignored and only the nominal operating power is considered. Therefore, the power requirement is approximately 2.45 W for the duration of the EVAs (~ 7 hours). Given these requirements and assuming that the CMGs will run off 8 - 12 Volts, the battery capacity can be computed using Eq. 4.1

$$\text{Capacity} = \frac{P\Delta t}{\eta V} \tag{4.1}$$

where P is the power requirement of the device in Watts, Δt is the mission length in hours, V is the expected voltage required to power the device in Volts, and η is the efficiency of the inverter. The battery capacity necessary for the MAJIC ACS is 2.52 Ah and 1.68 Ah, respectively for 8 and 12 Volts ($\eta = 85\%$).

Overhead constraints refers to the the mass, volume and budget of the batteries necessary to power the blended ACS. The mass and volume constraints should be minimal to ensure that the CMGs can buy their way on-board. Because of the prototypical approach to the design, the budget constraints for the MAJIC system are relaxed and assumed to be non-critical.

Table 4.2 provides information on batteries typically used in space applications. NiCd batteries are known for being rugged and forgiving but require proper care to achieve longevity. Economically, NiCd batteries are also the lowest in terms of cost per cycle. NiCd batteries, while having several advantages, are environmentally un-

friendly due to the cadmium and cannot be disposed of in landfills. NiMH batteries typically have 30-40% higher capacity than standard NiCd batteries. Similar to NiCd batteries, NiMH batteries are simple to store and transport. On the contrary, NiMH batteries have a limited lifetime due to deep discharges and require complex charging algorithms to avoid overcharging. Li-Ion batteries have an exceptionally high energy density and easy maintenance since no periodic discharges are required. The main disadvantages of Li-Ion batteries are the transportation regulations and aging restrictions caused by the chemistry of the battery. Li-Polymer batteries have the highest energy density, but are also the most expensive due to the manufacturing cost. Both Li-Ion and Li-Polymer batteries are subject to the same transportation regulations.

Table 4.2: Typical Batteries used in Space Applications

Category	**Abbrev.**	**Energy Density** $\frac{\text{Wh}}{\text{kg}}$	**Operating Temp.** °C	**Life Cycle**
Nickel Cadmium	NiCd	40	0 - 25	1000
Nickel Metal-Hydride	NiMH	60	0 - 25	300 - 500
Nickel Hydrogen	NiH2	65	0 - 20	20000+
Silver-Zinc	-	100	−20 - 75	100
Lithium Ion	Li-Ion	120+	0 - 40	500 - 1000
Lithium Polymer	Li-Polymer	150+	−10 - 40	30000+

Given the expected load of the CMG architecture over a typical EVA mission and the life cycle of a single MAJIC system, a Li-Polymer power source is recommended because of the high energy density and life cycle. Given the energy density of Li-Polymer batteries a mass of 0.10 kg can be expected. Since the bus voltage of the final CMG design is not yet known, the number of cells required is also uknown and therefore the mass of the batteries could vary depending on the number of cells. The high energy density will ensure that the mass of the batteries is minimized, while the vast life cycle will allow for thousands of charge and discharge cycles. For EVA missions, the battery charger will remain inside the ISS. The batteries are sized for a single EVA and therefore will only be charged after the EVA is complete. Given

the typical charging time for Li-Ion and Li-Polymer batteries, a charging time of 2-3 hours is expected for the MAJIC batteries.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 5

Conclusion

5.1 Discussion

Throughout this thesis, the development and analysis of a real-time simulation environment for parametrically sizing spacecraft ACS have been presented. The cross-platform simulation is extremely adaptable and can be easily modified to support a variety of ACS: RCS thrusters only, CMGs only or blended (both thrusters and CMGs). The number of CMGs, as well as the particular architecture, are also able to interchanged relatively easily. While developing the simulation environment, several novel contributions were made to interface with NASA's EDGE graphics visualization software. Additionally, asteroid surface sampling missions were piloted by highly experiences subjects several which had first-hand EVA experience (Secs. 3.2.3 and 3.4).

In Sec. 3.3.2, the optimal CMG sizes for a particular ensemble of missions is presented and a final system design is recommended which includes structural, ACS, and electrical subsystem upgrades to the current NASA baseline Jetpack in Secs. 4.2 and 4.3, respectively. The MAJIC system would greatly benefit from pre-constructed, interchangeable CMG architectures with various angular momentum capacities and output torques. This would allow the crew-members to swap CMG arrays for a particular EVA task. The ability to tune the blended ACS in real-time would improve EVA precision, as well as greatly reduce mission length and fuel consumption. The

Li-Polymer batteries are sized appropriately for a 7 hour EVA mission duration and can be recharged within 3 hours.

The development of the blended ACS and desaturation algorithm (Sec. 3.2.4) allow small-scale CMGs to be used in place of traditionally larger CMGs. The reduction in mass and significant improvement in pointing precision ensure that the CMGs are a valuable upgrade to the RCS thruster only ACS. The desaturation thresholds are easily modifiable, which ensure the CMGs sufficiently desaturate before resuming attitude control. The small-scale CMGs optimized for the four mission trajectories presented above cannot strictly buy their way on-board due to their effective mass: mass of the four CMGs, batteries and structural modifications that must be made to accomodate the new actuators. However, considering the tremendous increase in attitude control performance and the notion that the CMGs would be used for numerous EVAs over their lifetime, the CMGs are clearly desirable and should be incorporated in the MAJIC system design. Furthermore, the addition of the CMGs and fuel consumption savings over a long period of time allow the MAJIC system to become an operational unit instead of a contingency one. The operational benefit of the system allows crew-members to conduct EVAs in environments previously thought to be unreachable such as along the solar arrays and in foreign and/or uncooperative environments.

5.2 Recommendations for Future Work

The virtual-reality simulation environment for the MAJIC system shows that small-scale CMGs are valuable assets for NASA's baseline Jetpack. The CMGs significantly reduce the fuel consumption during an EVA mission, as well as increasing pointing accuracy for crew-members during precision tasks. The increased $\Delta\mathbf{V}$ increases the translational capability of the MAJIC system and permits operational use, as opposed to the emergency self-rescue SAFER. In order to verify the simulation results, a scaled demonstration with a prototype would provide meaningful corroboration. An air-bearing hardware-in-the-loop (HWIL) test-bed would allow various size CMGs to

be tested with different steering laws. NASA currently has an air-bearing set-up for the baseline Jetpack with thrusters. With slight modifications, the existing test-bed could be used to test the MAJIC system.

Additionally, the structural placement of the CMG architecture and electrical subsystem could be optimized to improve the system's overall mass and MOI properties. By rearranging the location of the CMGs and batteries, the system's MOI can be modified to allow for more control about a particular axis. Currently, there are no hardware analogs for the simulated small-scale CMGs. However, if a final CMG specification is determined through HWIL testing, the specifications can be sent to CMG manufacturers who can develop the appropriately sized CMGs.

The simulation can be updated with physical sensor models, which would allow investigations into the stochastic nature of the blended ACS. The sensor models will include non-linear effects and more advanced controllers can be derived to account for the non-linearities. In the current simulation, all measurements are assumed to be known and noiseless; however, in any real space system, some measurements may not be available and/or to varying degrees of precision. Depending upon which measurements are known, advanced filters can be used to estimate the state of the system such as an extended Kalman Filter (EKF) or an unscented Kalman Filter (UKF). EKFs linearize about the previous estimate through Jacobian approximations of the non-linear measurement and dynamic equations. UKFs deal with non-linearities by transforming the measurement and state data through modeled non-linear transformations to lower estimation covariance and improve accuracy.

The positive results from the current simulation environment can be bolstered through higher fidelity simulations that include more accurate sensor models, as well as HWIL demonstrations of the upgraded MAJIC system.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix A

Preliminary Monte-Carlo Simulation Figures

A.1 Mission 1 - Single-Axis Slew Maneuver

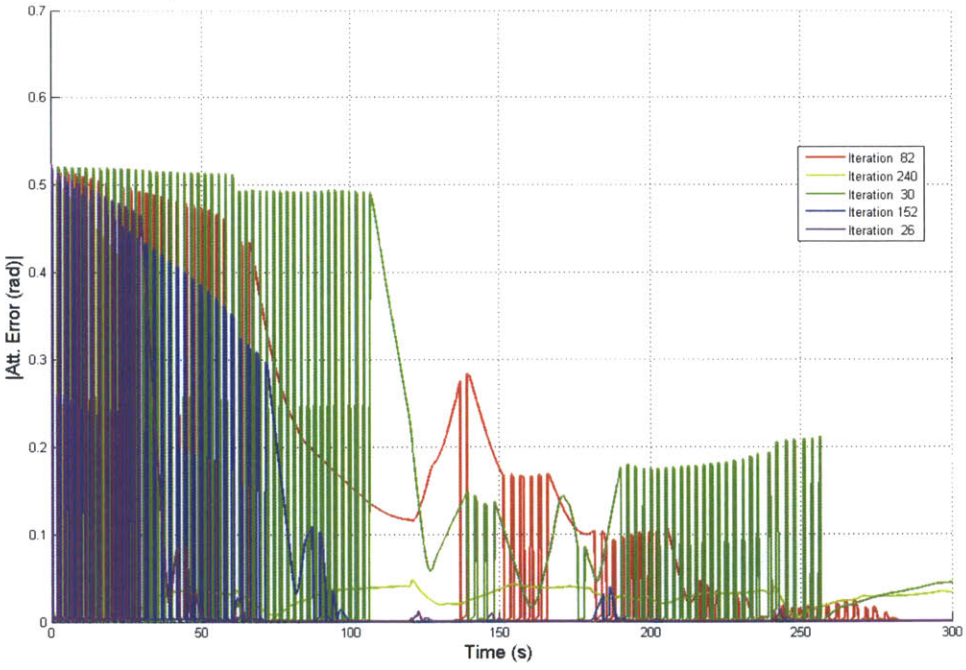


Figure A-1: Mission 1: Attitude Error (rad) vs. Time (s)

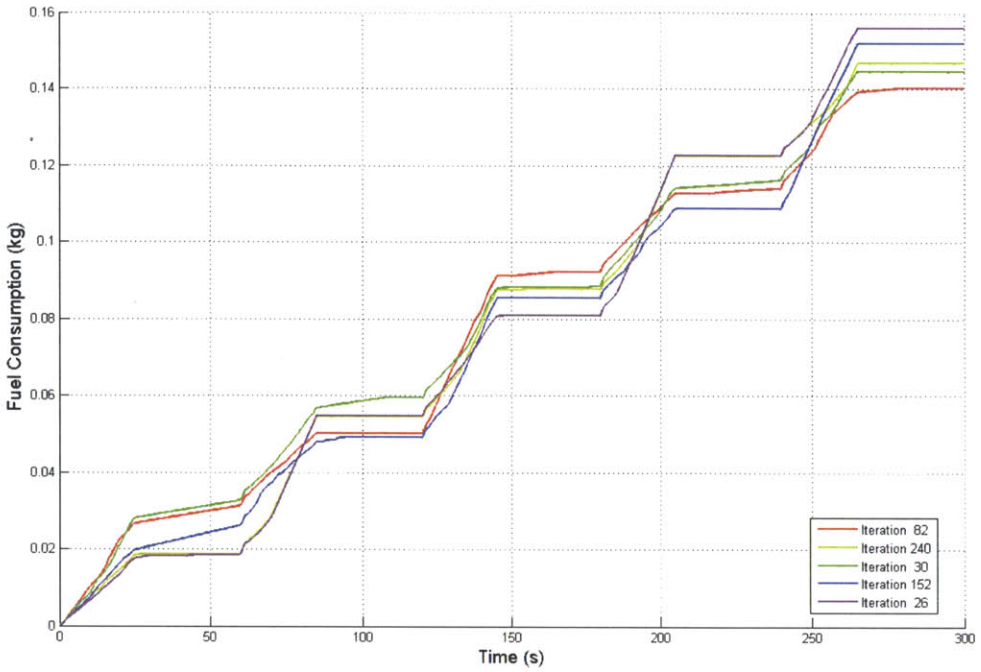


Figure A-2: Mission 1: Fuel Consumption (kg) vs. Time (s)

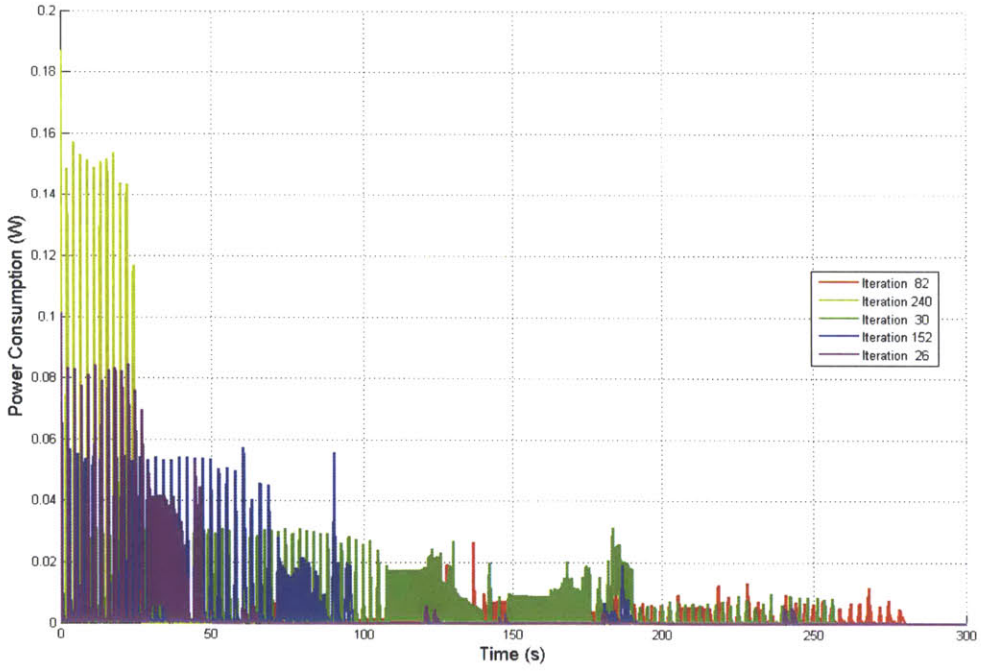


Figure A-3: Mission 1: Power (W) vs. Time (s)

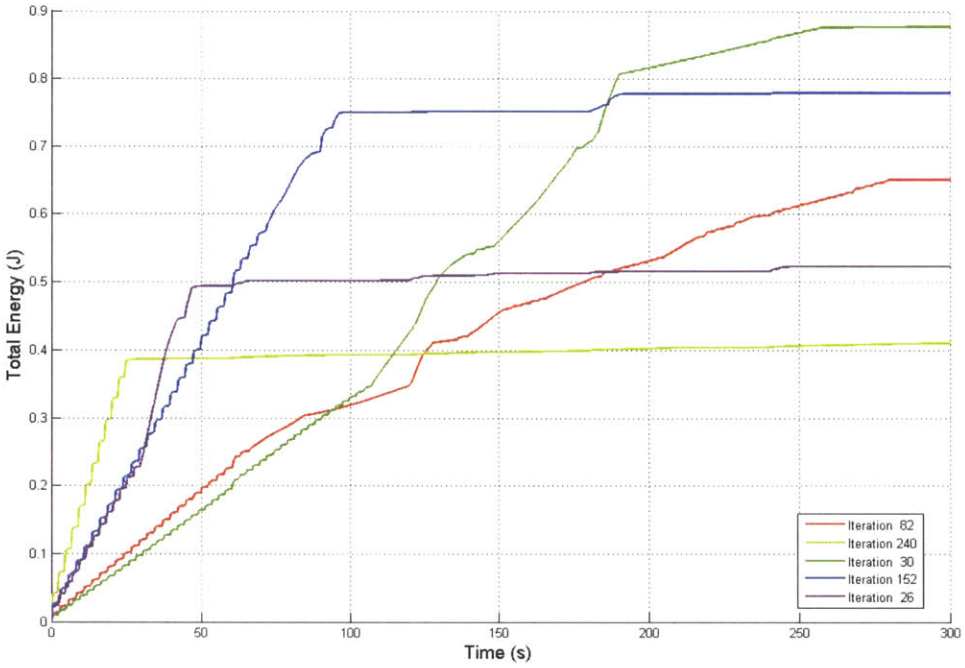


Figure A-4: Mission 1: Energy (J) vs. Time (s)

A.2 Mission 2 - Solar Array Inspection (ISS)

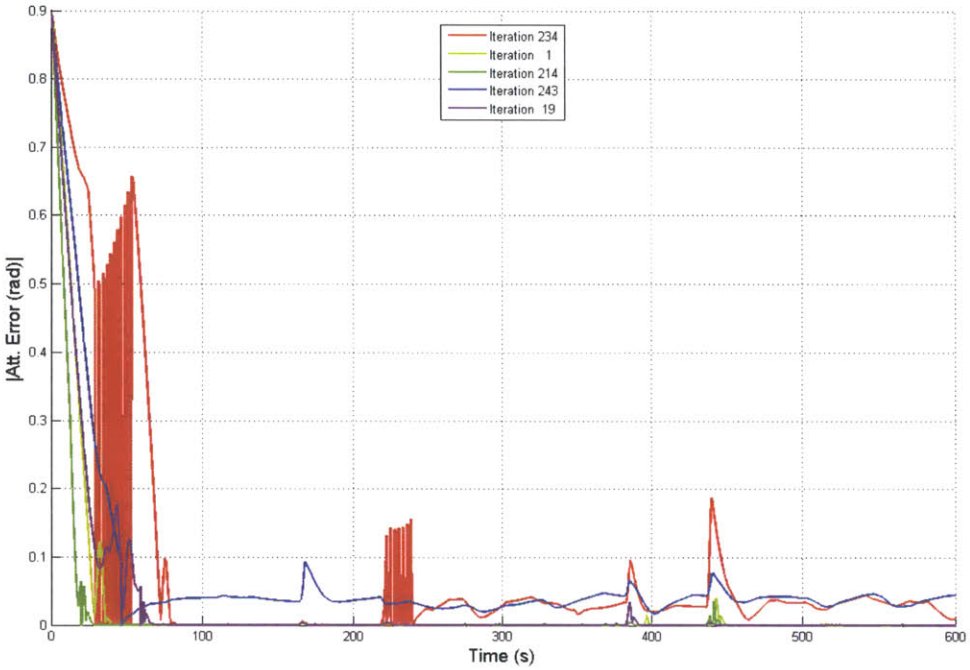


Figure A-5: Mission 2: Attitude Error (rad) vs. Time (s)

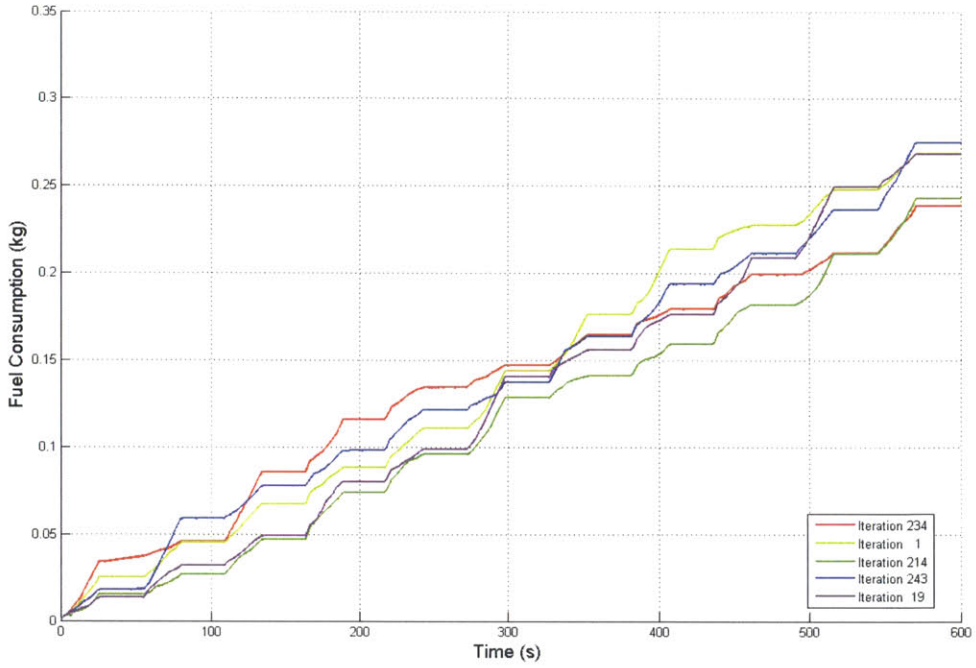


Figure A-6: Mission 2: Fuel Consumption (kg) vs. Time (s)

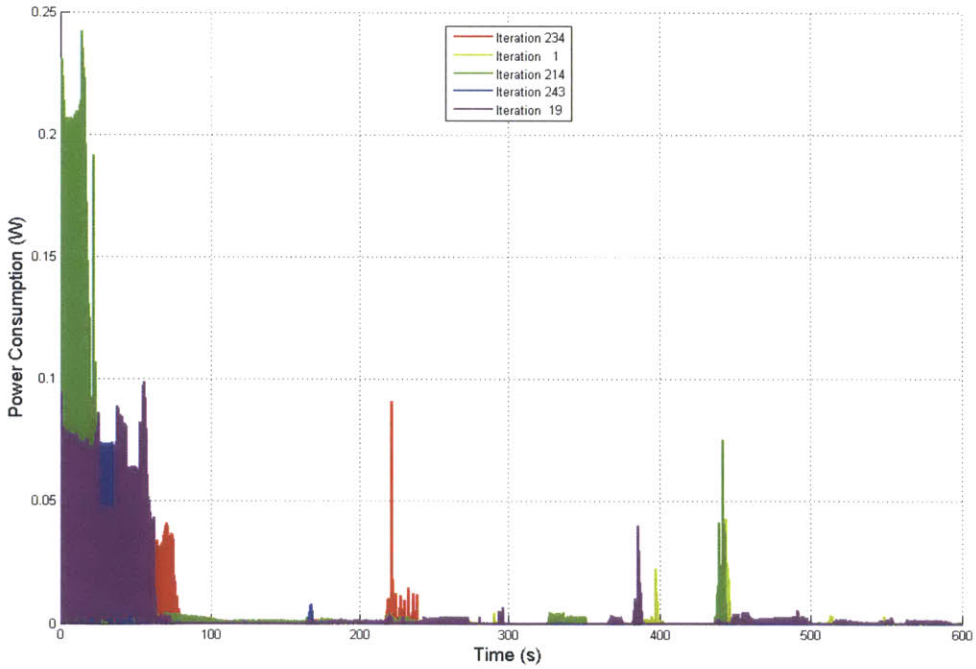


Figure A-7: Mission 2: Power (W) vs. Time (s)

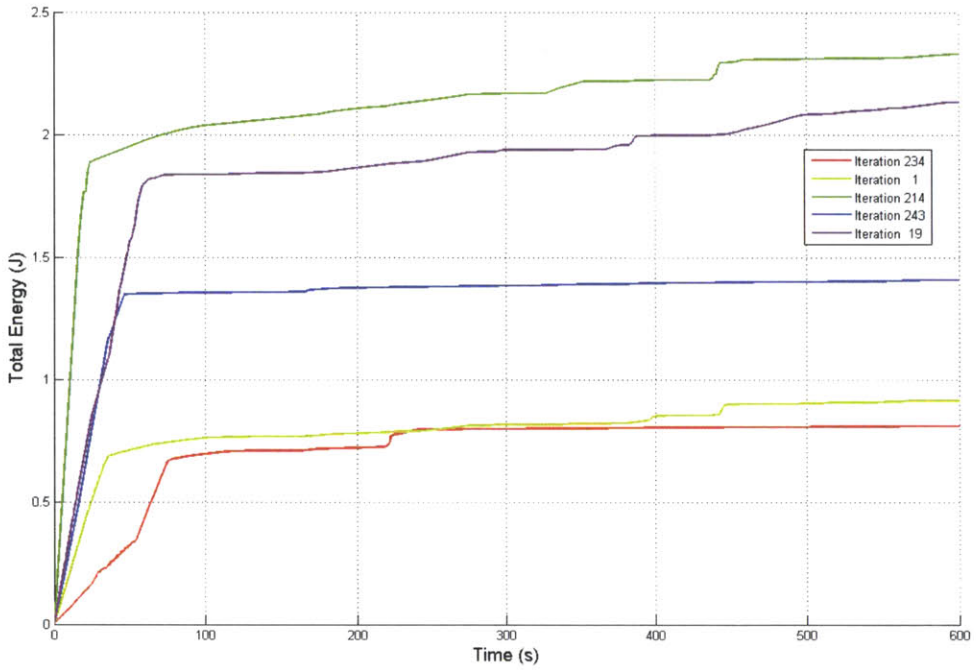


Figure A-8: Mission 2: Energy (J) vs. Time (s)

A.3 Mission 3 - Crew-member Rescue (ISS)

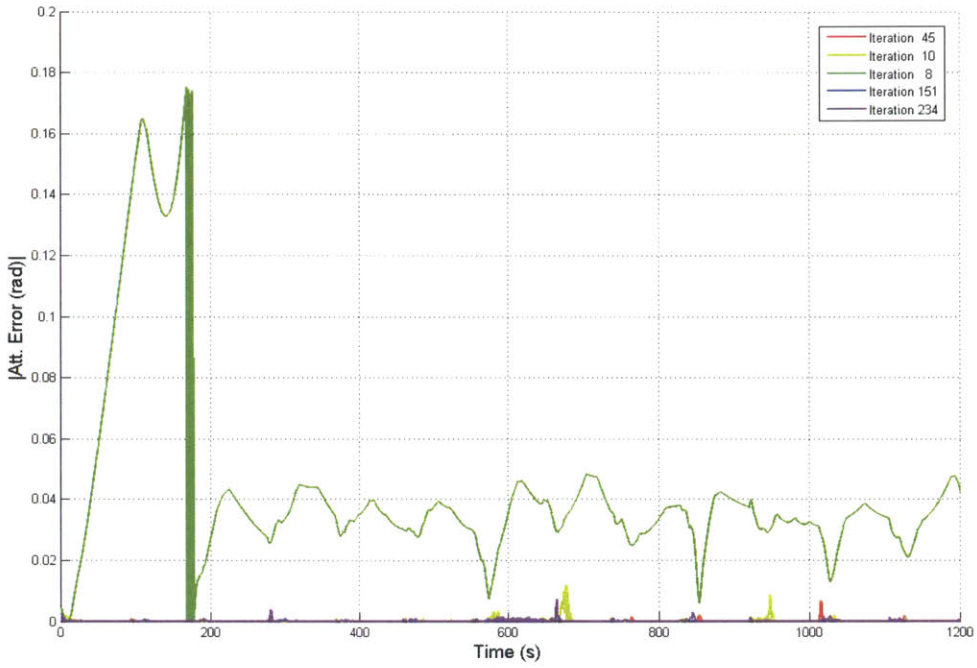


Figure A-9: Mission 3: Attitude Error (rad) vs. Time (s)

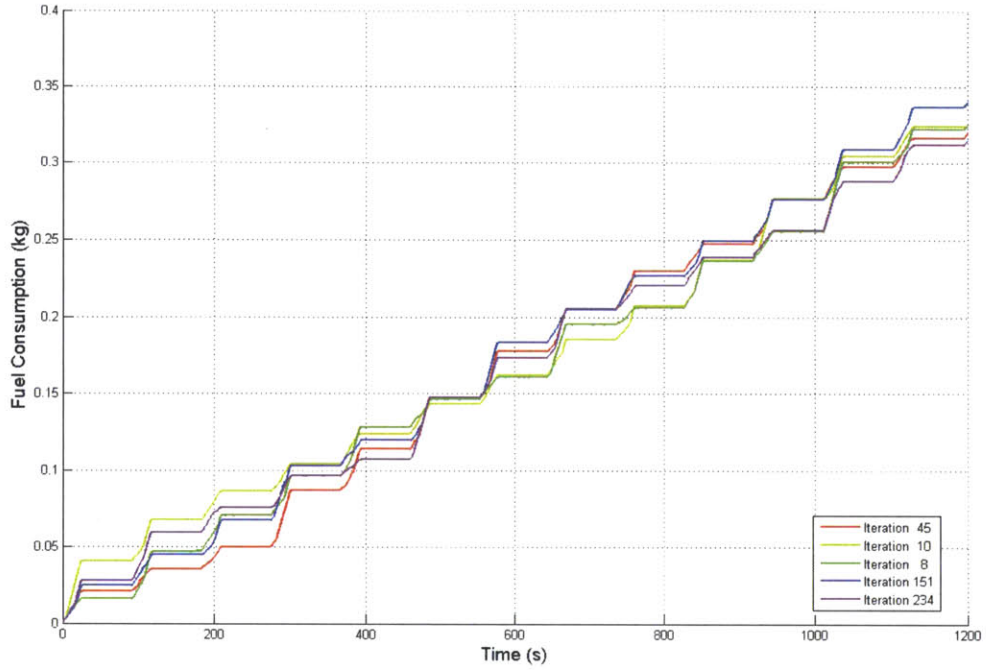


Figure A-10: Mission 3: Fuel Consumption (kg) vs. Time (s)

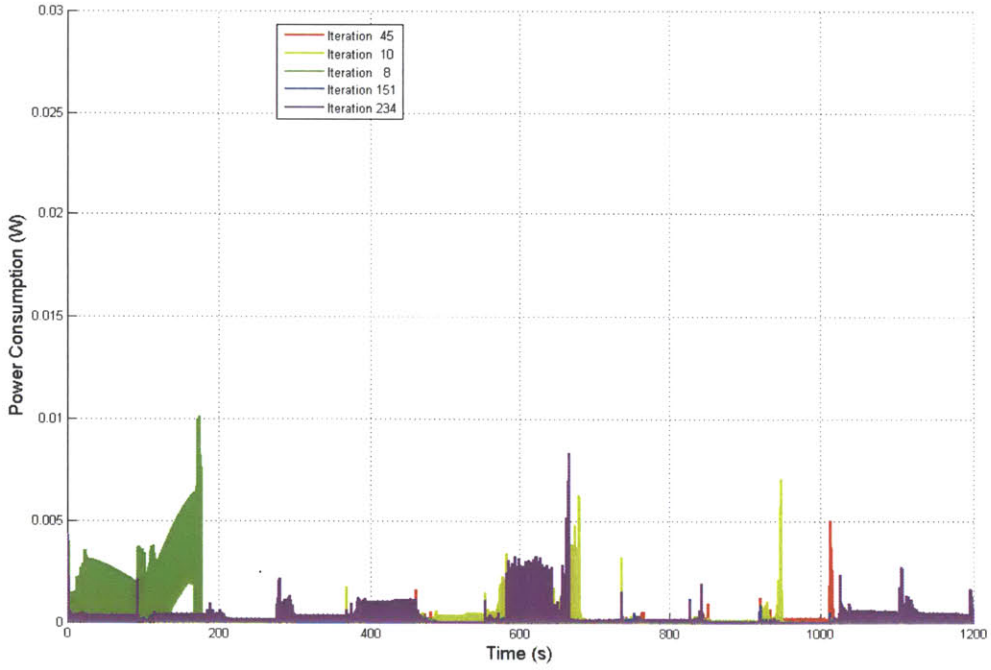


Figure A-11: Mission 3: Power (W) vs. Time (s)

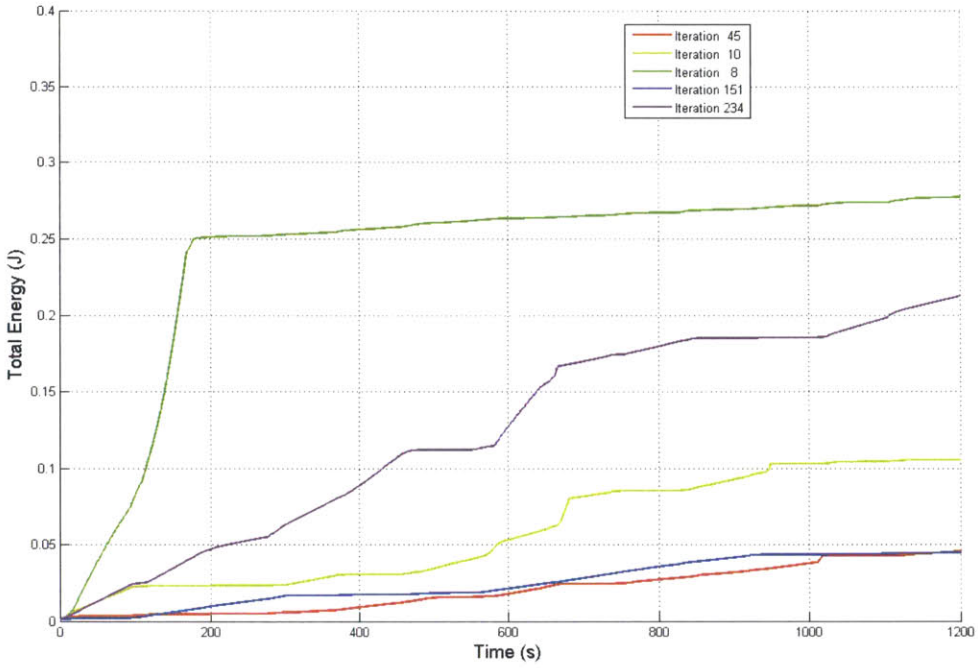


Figure A-12: Mission 3: Energy (J) vs. Time (s)

A.4 Mission 4 - Surface Sampling (Asteroid)

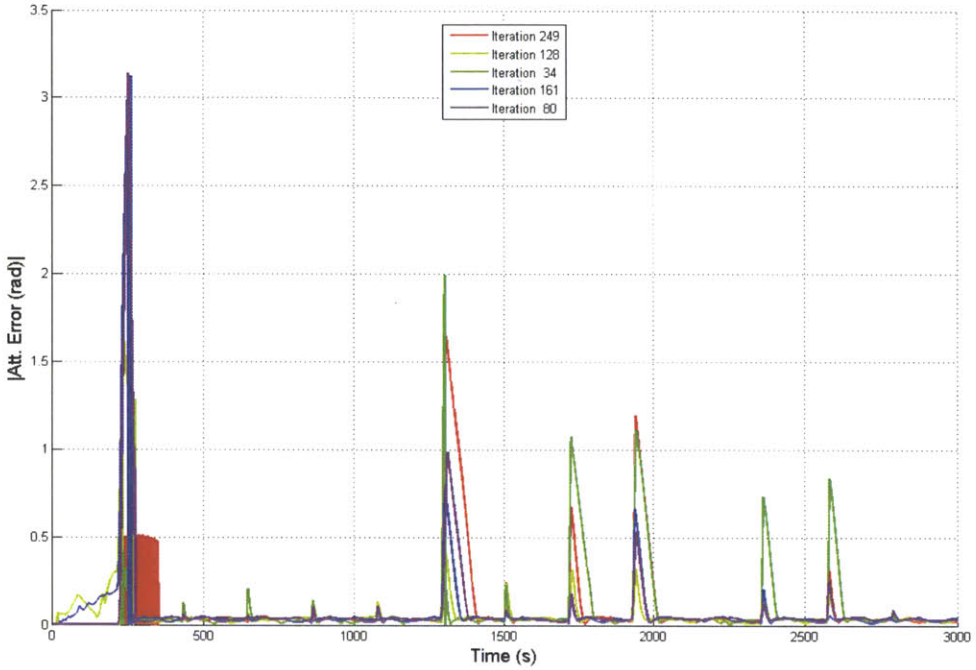


Figure A-13: Mission 4: Attitude Error (rad) vs. Time (s)

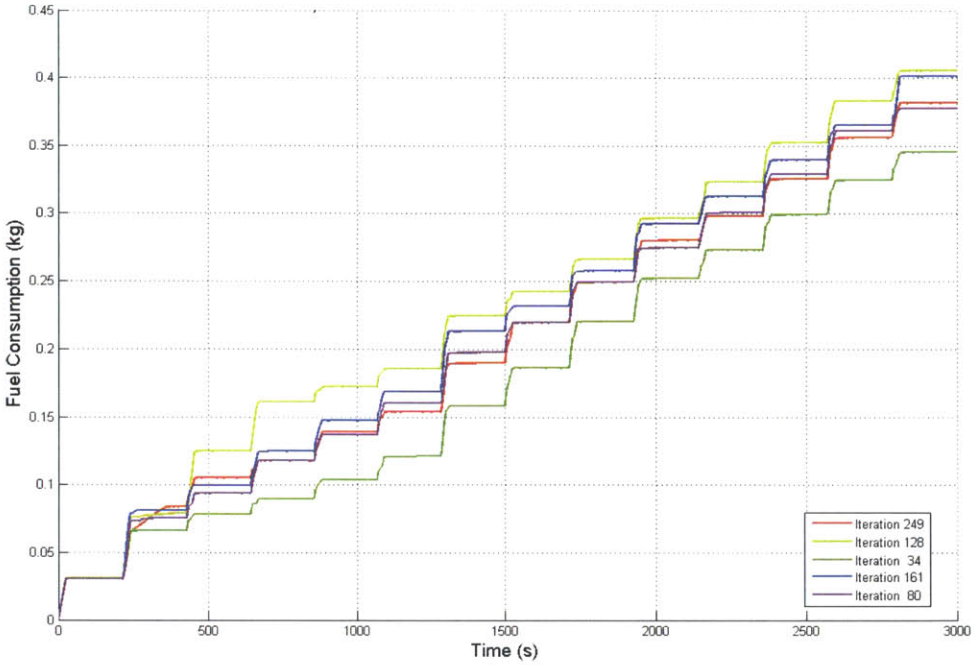


Figure A-14: Mission 4: Fuel Consumption (kg) vs. Time (s)

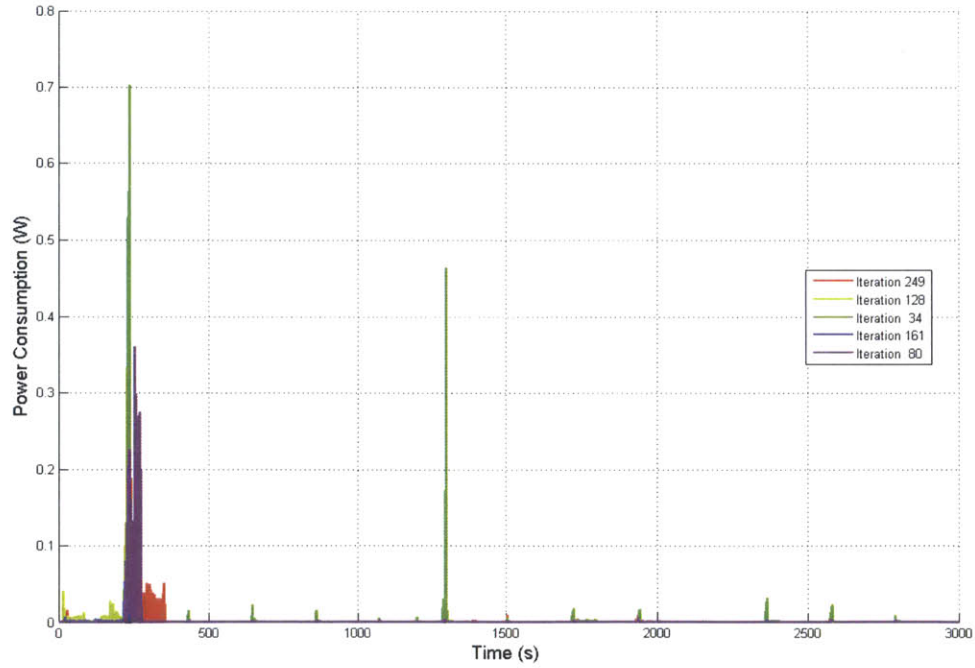


Figure A-15: Mission 4: Power (W) vs. Time (s)

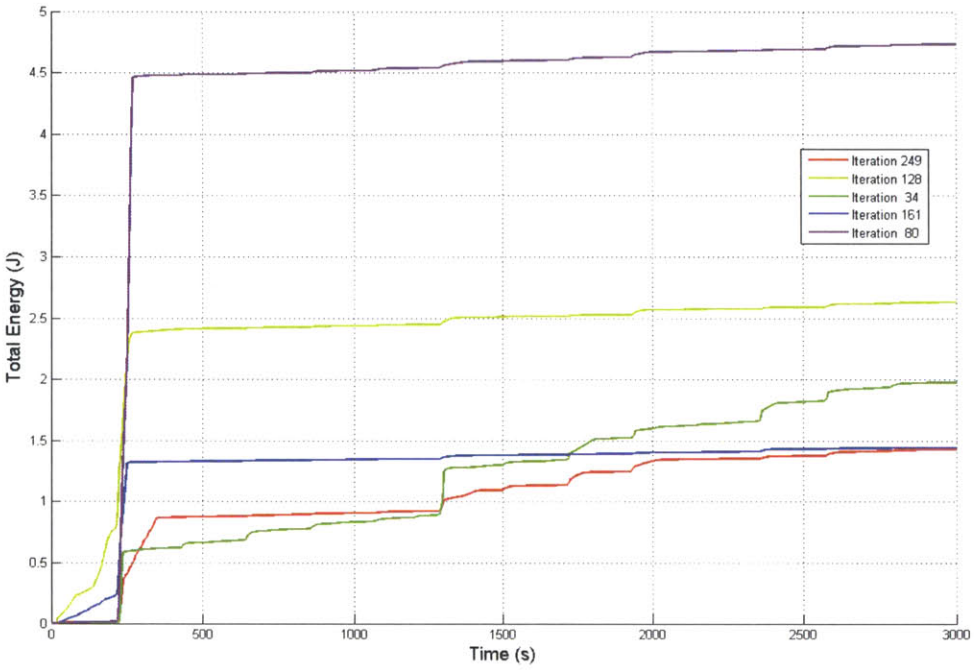


Figure A-16: Mission 4: Energy (J) vs. Time (s)

Appendix B

Final Monte-Carlo Simulation Figures

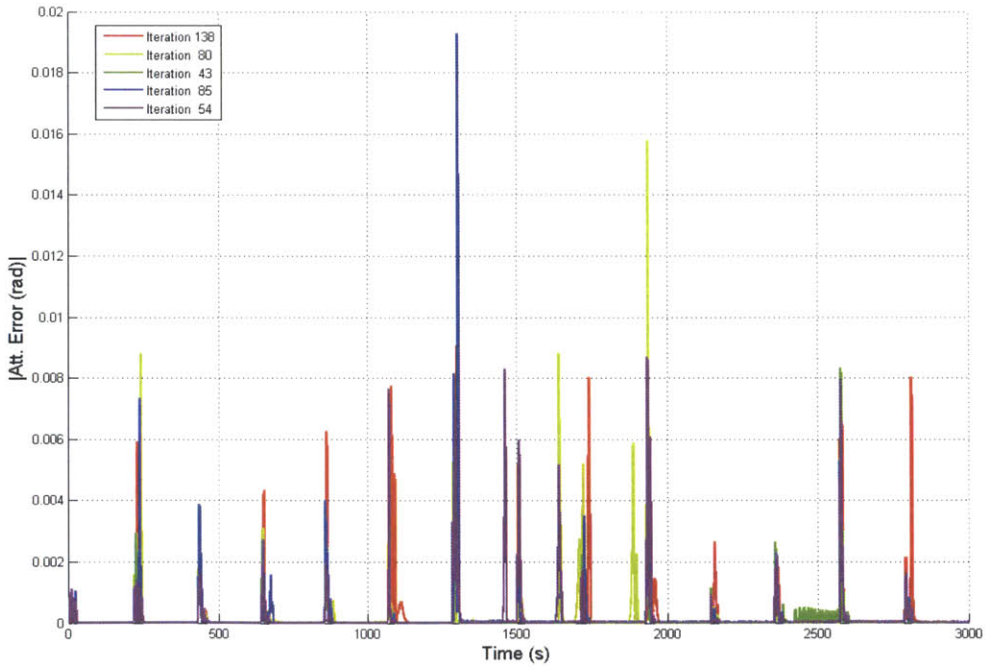


Figure B-1: Attitude Error (rad) vs. Time (s)

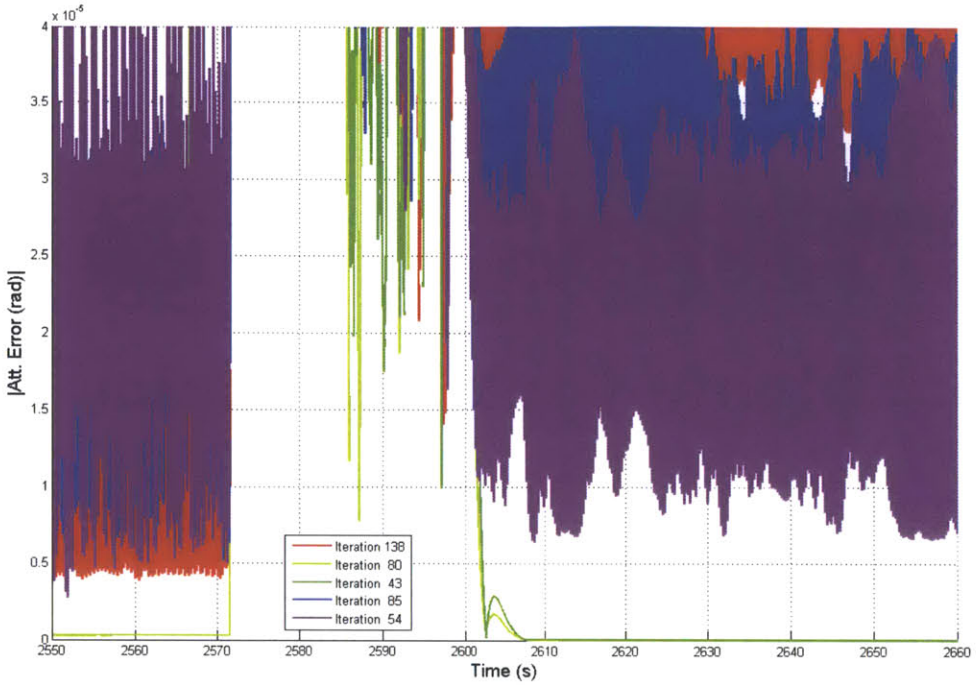


Figure B-2: Attitude Error (rad) vs. Time (s) (Enlarged)

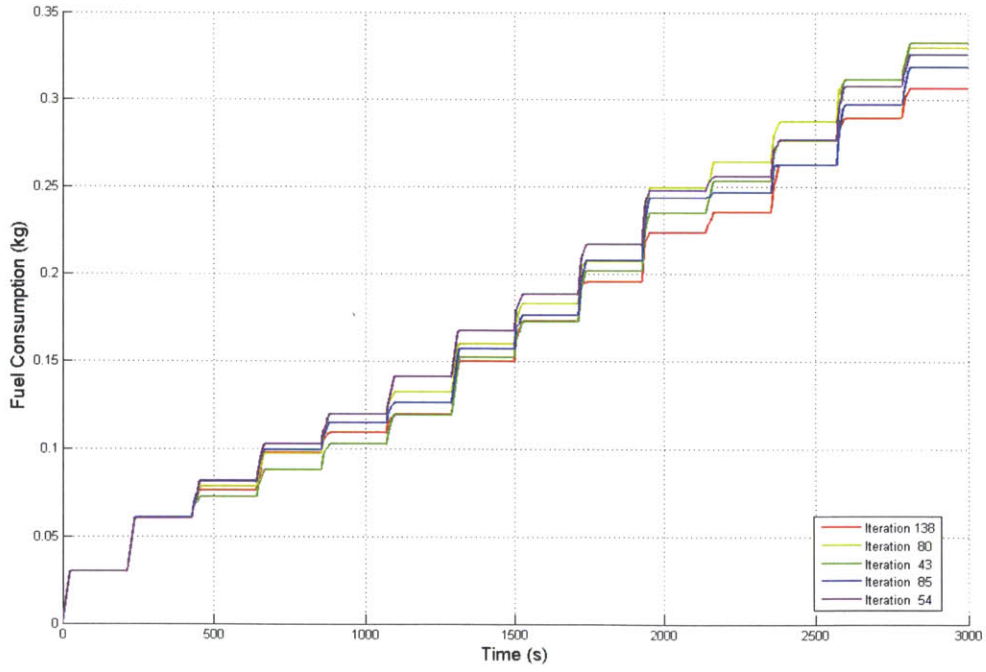


Figure B-3: Fuel Consumption (kg) vs. Time (s)

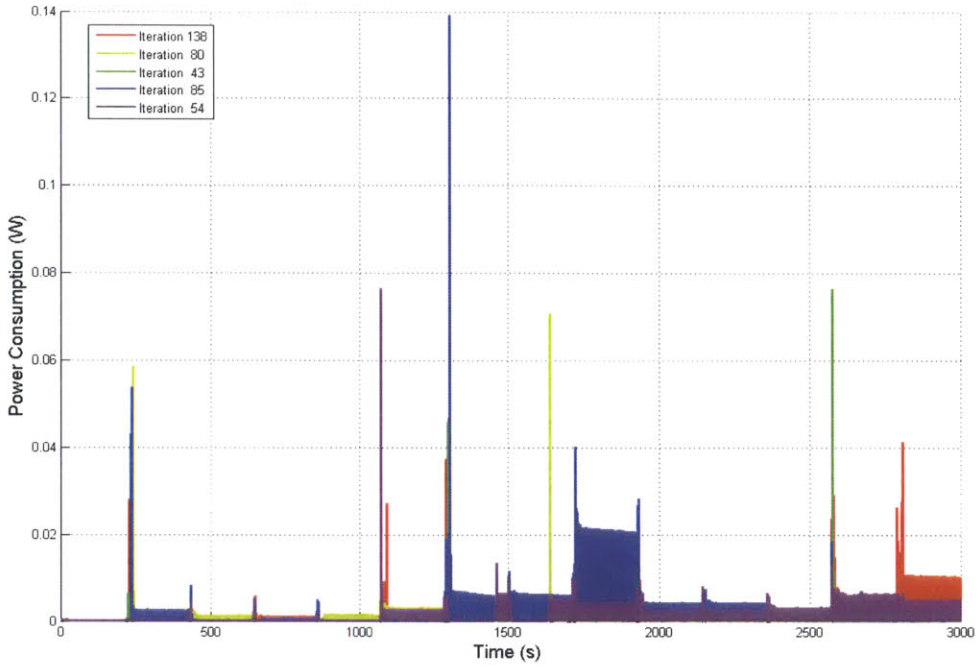


Figure B-4: Power (W) vs. Time (s)

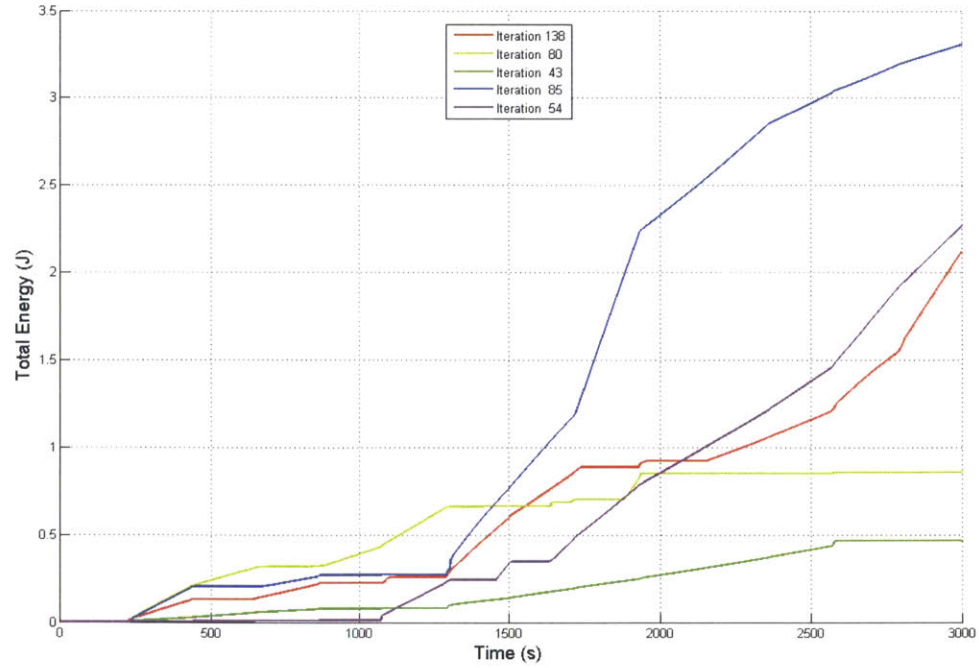


Figure B-5: Energy (J) vs. Time (s)

THIS PAGE INTENTIONALLY LEFT BLANK

Bibliography

[1] H. Yoon and P. Tsiotras, "Singularity analysis of variable speed control moment gyros," *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 3, pp. 374–386, 2004.

[2] M. C. Johnson, "A parameterized approach to the design of lunar lander attitude controllers," 2006.

[3] H. Barber, C. Buell, G. Falkenstein, and R. Gurnitz, "Microthrusters employing catalytically reacted n2-o2-h2 gas mixtures, tridyne," *Journal of Spacecraft and Rockets*, vol. 8, no. 2, pp. 111–116, 1971.

[4] P. Hughes, *Spacecraft Attitude Dynamics*. Courier Dover Publications, 2012.

[5] A. Buckingham and J. Braumiller, "Magnetic torquers for momentum desaturation of space station control moment gyros," *Journal of Spacecraft and Rockets*, vol. 9, no. 6, pp. 387–388, 1972.

[6] T. Murtagh, C. Whitsett, M. Goodwin, and J. Greenlee, "Automatic control of the skylab astronaut maneuvering research vehicle," *Journal of Spacecraft and Rockets*, vol. 11, no. 5, pp. 321 326, 1974.

[7] C. McChesney, "Design of attitude control actuators for a simulated spacecraft," DTIC Document, Tech. Rep., 2011.

[8] N. S. Bedrossian, J. Paradiso, E. V. Bergmann, and D. Rowell, "Redundant single gimbal control moment gyroscope singularity analysis," *Journal of Guidance, Control, and Dynamics*, vol. 13, no. 6, pp. 1096–1101, 1990.

[9] B. Wie, D. Bailey, and C. Heiberg, "Singularity robust steering logic for redundant single-gimbal control moment gyros," *Journal of Guidance, Control, and Dynamics*, vol. 24, no. 5, pp. 865–872, 2001.

[10] N. S. Bedrossian, J. Paradiso, E. V. Bergmann, and D. Rowell, "Steering law design for redundant single-gimbal control moment gyroscopes," *Journal of Guidance, Control, and Dynamics*, vol. 13, no. 6, pp. 1083 1089, 1990.

[11] S. Vadali, S. Walker, and H.-S. Oh, "Preferred gimbal angles for single gimbal control moment gyros," *Journal of Guidance, Control, and Dynamics*, vol. 13, no. 6, pp. 1090–1095, 1990.

[12] H. Schaub and J. L. Junkins, *Analytical Mechanics of Space Systems.* American Institute of Aeronautics and Astronautics, 2003.

[13] P. Singla, D. Mortari, and J. L. Junkins, "How to avoid singularity when using euler angles," *Advances in the Astronautical Sciences. San Diego, California: American Astronautical Society*, pp. 1409–1426, 2004.

[14] J. B. Kuipers, *Quaternions and Rotation Sequences.* Princeton university press Princeton, 1999.

[15] F. A. Leve, "Novel steering and control algorithms for single-gimbal control moment gyroscopes," Ph.D. dissertation, University of Florida, 2010.

[16] G. Margulies and J. Aubrun, "Geometric theory of single-gimbal control moment gyro systems," *Journal of the Astronautical Sciences*, vol. 26, no. 2, pp. 159–191, 1978.

[17] J. Dominguez and B. Wie, "Computation and visualization of control moment gyroscope singularities," in *AIAA Guidance, Navigation, and Control Conference*, 2002.

[18] A. Ben-Israel and T. N. Greville, *Generalized Inverses.* Wiley-Interscience, 1974.

[19] I. Griva, S. G. Nash, and A. Sofer, *Linear and Nonlinear Optimization.* Siam, 2009.

[20] E. J. Paddock, A. Lin, K. Vetter, and E. Z. Crues, "Trick!: A simulation development toolkit," in *Proceedings of AIAA Modeling and Simulation Technologies Conference and Exhibit*, 2003.

[21] M. Jackson and R. McDonald, "Draper simulation analysis tool (dsat): Graphical object simulation techniques and tools for simulink," in *2004 AIAA Conference on Simulation Tools and Techniques*, 2004.

[22] J. K. Ousterhout, *Tcl and the Tk Toolkit.* Pearson Education India, 2006.

[23] G. Marsaglia, "Dichard statistical tests," 1995.

[24] M. Matsumoto and T. Nishimura, "Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 8, no. 1, pp. 3–30, 1998.

[25] D. Linden and T. Reddy, "Handbook of batteries, 2002."