NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

DESIGN AND EXPERIMENTAL IMPLEMENTATION OF OPTIMAL SPACECRAFT ANTENNA SLEWS

by

Adam G. Sears

December 2013

Thesis Co-Advisors: Mark Karpenko
I. Michael Ross

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704–0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202–4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704–0188) Washington DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** December 2013	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE DESIGN AND EXPERIMENTAL IMPLEMENTATION OF OPTIMAL SPACECRAFT ANTENNA SLEWS		**5. FUNDING NUMBERS**	
6. AUTHOR(S) Adam G. Sears			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943–5000		**8. PERFORMING ORGANIZATION REPORT NUMBER**	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A		**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB protocol number ____N/A____.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited		**12b. DISTRIBUTION CODE**	

13. ABSTRACT (maximum 200 words)

This thesis investigates the development and implementation of optimal slew trajectories for positioning a spacecraft antenna. Conventional maneuvers are developed by considering each gimbal independently. Consequently, maneuver design is simple, but may be highly sub-optimal and cause significant torques to be imposed on the spacecraft body. This work explores the impact of implementing optimal slew paths that best utilize system dynamics with the objective of increasing available customer time on communications links and enabling new missions. Accomplishing this required the development of a detailed multibody system model that can be easily tailored to any spacecraft antenna configuration. Various software suites were used to perform thorough validation and verification of the Newton-Euler formulation developed herein. The antenna model was then utilized to solve an optimal control problem for a geostationary communications satellite. The developed maneuvers not only reduce the antenna slew time, but also reduce the impact of the antenna motion on the spacecraft attitude. This reduces reliance on the spacecraft attitude control system to maintain pointing, and minimizes the impact of antenna motion on the operation of other payloads. Successful implementation of the designed maneuvers on a laboratory testbed validate the approach in a real hardware environment.

14. SUBJECT TERMS Optimal Slew Maneuvers, Spacecraft, Antenna, Communications Satellite, Multibody Dynamics			**15. NUMBER OF PAGES** 171
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540–01–280–5500 Standard Form 298 (Rev. 2–89)
Prescribed by ANSI Std. 239–18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

DESIGN AND EXPERIMENTAL IMPLEMENTATION OF OPTIMAL SPACECRAFT ANTENNA SLEWS

Adam G. Sears
Captain, United States Air Force
B.S., Embry-Riddle Aeronautical University, 2006

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ASTRONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 2013

Author: Adam G. Sears

Approved by: Mark Karpenko
Co-Advisor

I. Michael Ross
Co-Advisor

Knox T. Millsaps
Chair, Department of Mechanical and Aerospace Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This thesis investigates the development and implementation of optimal slew trajectories for positioning a spacecraft antenna. Conventional maneuvers are developed by considering each gimbal independently. Consequently, maneuver design is simple, but may be highly sub-optimal and cause significant torques to be imposed on the spacecraft body. This work explores the impact of implementing optimal slew paths that best utilize system dynamics with the objective of increasing available customer time on communications links and enabling new missions. Accomplishing this required the development of a detailed multibody system model that can be easily tailored to any spacecraft antenna configuration. Various software suites were used to perform thorough validation and verification of the Newton-Euler formulation developed herein. The antenna model was then utilized to solve an optimal control problem for a geostationary communications satellite. The developed maneuvers not only reduce the antenna slew time, but also reduce the impact of the antenna motion on the spacecraft attitude. This reduces reliance on the spacecraft attitude control system to maintain pointing, and minimizes the impact of antenna motion on the operation of other payloads. Successful implementation of the designed maneuvers on a laboratory testbed validate the approach in a real hardware environment.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

ADCS	attitude determination and control system
BVP	boundary value problem
CoM	center of mass
DoD	Department of Defense
DOF	degrees of freedom
DCM	direction cosine matrix
EFOV	extended field of view
FOV	field of view
GEO	geosynchronous earth orbit
GRGT	Guam Remote Ground Terminal
HEO	highly elliptical orbit
ISS	International Space Station
LEO	low earth orbit
MA	multiple access
MEO	medium earth orbit
MOC	Missions Operations Center
MMS	Magnetospheric Multiscale Mission
NASA	National Aeronautics and Space Administration
NOAA	National Oceanic and Atmospheric Administration
NPS	Naval Postgraduate School
PD	Proportional-Derivative
RBSP	Radiation Belt Storm Probes
SA	single access
SAC	single access compartment
SAR	Schedule Add Request
SNAS	Space Network Access System
STK	Systems Tool Kit
TDRS	tracking data relay satellite
TDRSS	tracking data relay satellite system
TRACE	Transitional Regional Coronal Explorer

TT&C	Telemetry, Tracking, and Command
WSC	White Sands Complex

I. INTRODUCTION

A. MOTIVATION

Since 1983, the Tracking and Data Relay Satellite System (TDRSS) constellation has been providing Department of Defense (DoD), civil, and commercial users with communications and data relay services for a variety of space-based platforms in low earth orbit (LEO), medium earth orbit (MEO), highly elliptical orbits (HEO) and even geosynchronous orbits (GEO) [1]. The constellation is now in its third generation with the recent launch of Tracking and Data Relay Satellite-K (TDRS-K) in January, 2013 [2]. The current constellation consists of eight satellites, TDRS-D through TDRS-K, as seen in Figure 1.

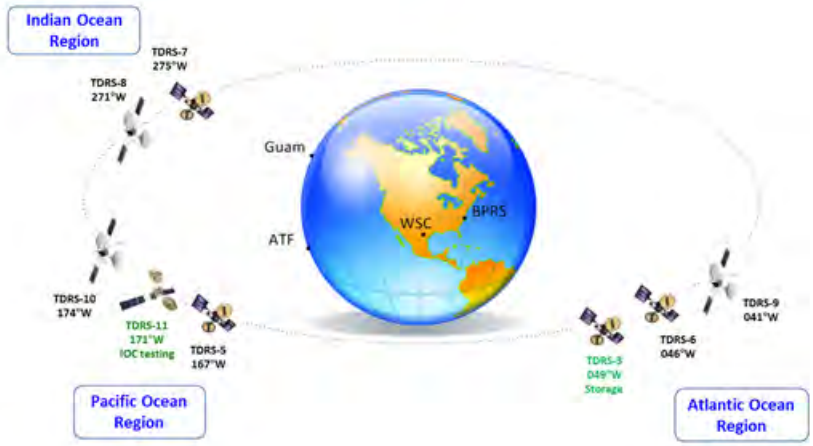


Figure 1. TDRS Constellation (from [3])

As shown in Figure 2, the current generation of TDRS consists of four communications links. The multiple access (MA) antenna provides S-band links to LEO users in a ±14° field-of-view (FOV) [4]. The 2.4 m Space-Ground Link Antenna provides a ground link to the White Sands Complex (WSC) or Guam Remote Ground Terminal (GRGT) [5]. Forward and aft omni antennas provide S-band Telemetry, Tracking, and Command (TT&C) [4]. Lastly, two steerable single access (SA) antennas provide links to

users in all orbit regimes with S-band, Ku-band, and Ka-band links [4]. Optimization of the slewing control for these 4.9 m dishes is the focus of this thesis.

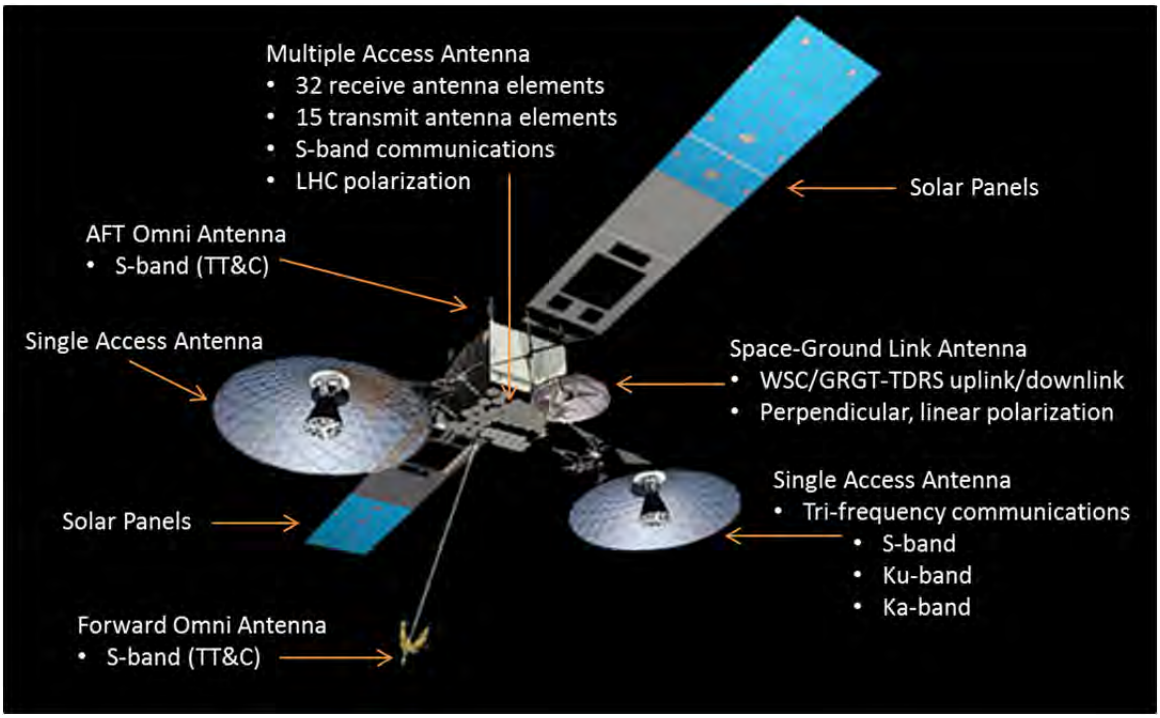


Figure 2. Third Generation TDRS Overview (from [4])

B. TDRS SINGLE ACCESS ANTENNA POINTING

While the TDRSS has proven to be a very effective and capable constellation, there are limitations to its ability to meet all customer needs. One of the often overlooked limitations is the efficiency of the SA antenna slews. The 0.17° beamwidth of the SA antenna, when used in Ka-band, drives stringent pointing requirements [6]. The SA antenna tracks user satellites via an azimuth and elevation gimbal along the pitch axis (South direction) and roll axis (orbit direction) of the spacecraft, as shown in Figure 3.

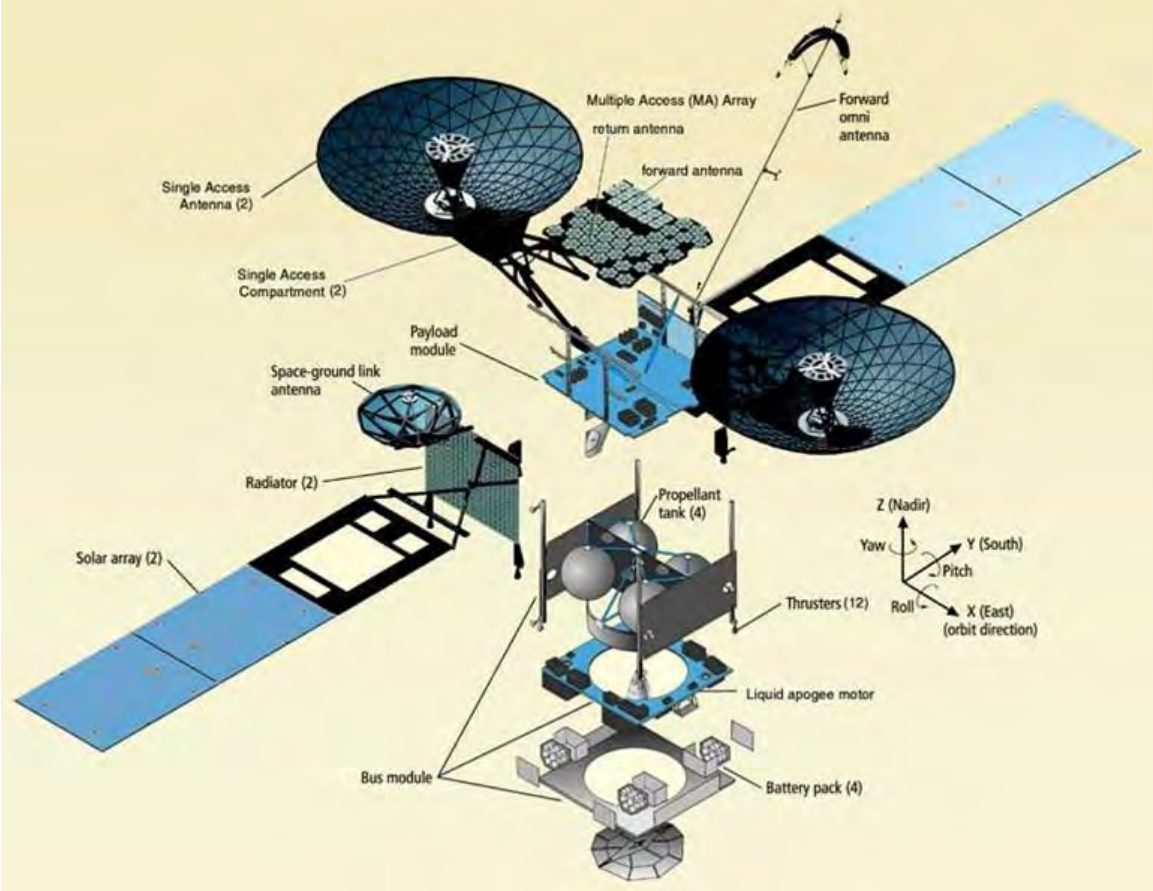


Figure 3. Third Generation TDRS Exploded View (from [7])

TDRS operates using three distinct fields of view. The LEO FOV consists of a ±14° box centered on nadir, and services 95% of all TDRS customers [8]. The primary field of view covers an ellipse where the major axis extends ±32° in roll (elevation), and ±24° in pitch (azimuth) [8]. Lastly, an extended field of view (EFOV) extends the primary ellipse out to 72° outboard, allowing for the servicing of highly elliptical, or even geosynchronous customers [8].

In any operating region, when a contact is broken with a departing customer, however, it takes considerable time for the antenna to slew before a new customer can be acquired. While a nominal slew is designed to take three minutes, slews can take well under three minutes for small slews within LEO, to over seven minutes for a large slew to or from the outboard limits of the antenna FOV [8]. Unfortunately, even if the size of the slew is minimal and can be performed under three minutes, the flight software inserts

dead time in order to increase the slew time to three minutes [8]. This is due to current limitations with the scheduling software.

C. TDRS SCHEDULING PROBLEM

Presently, the contact schedule for TDRSS is scheduled three weeks in advance [8]. Customers submit requests for contact, including desired start and stop times in a Schedule Add Request (SAR) [8, 9]. Scheduling is performed using a combination of three systems, including the Space Network Access System (SNAS), as shown in Figure 4 [9]. Since the schedule can change over three weeks and due to the difficulty of accurately predicting TDRS and target satellite ephemeris three weeks in advance, a standardized three minute block of time is scheduled for each slew, based on the maximum slew time within the primary FOV [8]. If the antenna slew will take less than three minutes to complete, an appropriate amount of dead time is inserted following cutoff from the departing customer [8]. For example, if a slew is calculated to take two minutes, the SA antenna will remain pointed at the last position of the departing satellite for one minute before beginning the two minute slew maneuver to the next customer. On completion of the maneuver, the antenna will be pointing at the desired position and be moving at the rate desired to intercept the new satellite. Slews that take longer than three minutes unexpectedly (i.e., slight increases in data time due to ephemeris error during initial scheduling) result in an alarm at the mission operations center (MOC) and a reduction in user operational time. Large slews that are known ahead of time to exceed three minutes (a seven minute slew to or from the EFOV for the Magnetospheric Multiscale Mission (MMS) or Radiation Belt Storm Probes (RBSP) missions, for example) are scheduled in advance in order to provide the customer with an accurate start time [8].

The scheduling approach outlined above may lead to inefficient use of the TDRS system and even result in dropped users. First, there is the obvious inefficiency of inserting dead time for small slews. While there may in fact be hours of dead time between contacts due to a lack of access requests, this small delay could greatly impact customer operational time and satisfaction during periods of peak congestion.

Furthermore, it may be possible to improve on larger slews, such as the seven minute EFOV slews, in order to provide more operational time. Lastly, due to the scheduling constraints, certain users may be dropped because of an inability to meet timeliness.

One such example of this type of dropped mission is in servicing a train of earth sciences satellites. Multiple earth-sciences satellites are placed in a coordinated sun-synchronous orbit in order to provide correlated passes of a variety of data. The satellites fly in relatively close proximity, and in order to meet their requests, the SA antenna has to slew rapidly to catch each passing satellite. However, since the maximum time to slew between customers is less than three minutes, the current system cannot perform this mission.

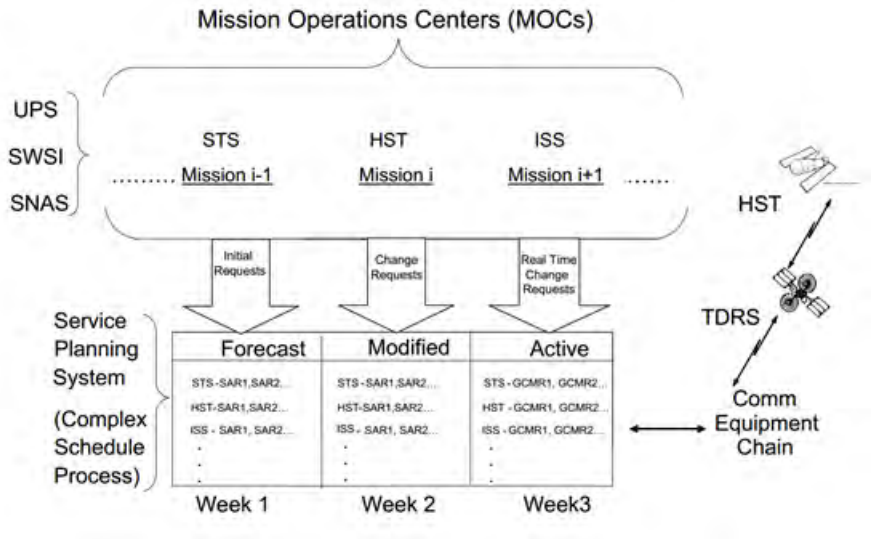


Figure 4. TDRS Scheduling Process (from [9])

While the three-week scheduling constraint limits the options available to improve maneuver efficiency, certain methods can be explored to yield measurable benefits. Primarily, if it can be shown that traditional maneuvers can be completed in less time using optimal control theory, then the reduced-time slews can be implemented to obtain a new scheduled slew time, which can replace the current three minute slew. While this method may still suffer from the inefficiency of downtime (due to the scheduling problem), improving slew time will increase operational availability,

particularly during periods of congestion. Furthermore, the timeliness of the largest slews could be improved, increasing availability for manually scheduled slews as well. Lastly, if the time can be reduced enough, missions like the Earth science train could be satisfied, thereby further extending the utility of the TDRSS.

D. THESIS OBJECTIVE AND SCOPE

The objective of this thesis is to apply optimal control theory to improve the design of spacecraft antenna slews. The goal will be to reduce the amount of time required to slew between customers. It was also found during preliminary research that the slew rate during program track (30 steps/s or 0.225°/s) was the most limiting factor on antenna slews [10, 11].[1] It was assumed that this limit was not a hard limit driven by gimbal design, but a soft limit designed to ensure that the impact of the antenna motion on the spacecraft body, and thus the other communications antennas, is minimal. Therefore, another objective of this thesis is to minimize the effect of antenna slews on the accumulation of spacecraft angular momentum. Improvements to this area could impact a wide variety of areas for all spacecraft with moving appendages or systems, improving the momentum space of control systems, increasing pointing accuracy, and possibly decreasing fuel and mass requirements. The result of applying optimal control theory to the problem of antenna slew design can therefore improve the availability of these systems and also have a profound affect on the design of future bus systems.

In order to accomplish optimization, the spacecraft had to be modeled. The dynamics of the TDRS spacecraft and antenna couple can be described succinctly by Equation (1.1) and Equation (1.2), where I_b is the inertia dyadic of the spacecraft, I_a is the inertia dyadic of the SA antenna, R_a is the influence matrix of the main body on the antenna, ω_a and ω_b are the antenna and spacecraft rotation rates, H_b is the angular momentum of the satellite, and T_b is torque on the satellite, T_a is torque on the antenna. While these equations can be expanded in order to include sloshing and flexible modes,

[1] It is assumed that the rates described in [10], which are given in steps per minute and degrees per minute, are actually values for steps per second and degrees per second, which would correlate with data from [11].

these impacts were determined to be outside the scope of this thesis and assumed to be zero.

$$I_b\dot{\omega}_b + R_a\dot{\omega}_a + \omega_b \times H_b = T_b \tag{1.1}$$

$$I_a\dot{\omega}_a + R_a{}^T\dot{\omega}_b = T_a \tag{1.2}$$

Many of the parameters that make up the spacecraft and antenna inertia and the influence matrix in Equation (1.1) and Equation (1.2) depend on the configuration of the antenna system. Therefore, Equation (1.1) and Equation (1.2) cannot be used directly for analysis. Instead a complete dynamic model of the multibody system has to be developed which also includes the influence of the motion of the spacecraft body. This analysis is outlined in Chapter II through Chapter IV. Furthermore, the model was validated to ensure accuracy of the dynamic response. Validation and verification of the dynamic model is described in Chapter V.

Lastly, an optimal control method had to be chosen. The tools developed at the Naval Postgraduate School (NPS) for attitude maneuver optimization are the prime candidates for approaching the TDRS slew problem. Faculty and researchers have explored a wide variety of optimal control problems and have had particular success in improving the timeliness and efficiency of spacecraft attitude maneuvers. In 2006, the National Aeronautics and Space Administration (NASA) performed a zero-propellant maneuver on the International Space Station (ISS), rotating 90° and 180° without the use of propellant, by utilizing a maneuver designed at NPS [12]. In 2010, NPS was given the opportunity to perform optimal maneuvers on NASA's Transitional Regional and Coronal Explorer (TRACE) spacecraft, where slew times were improved by approximately 20% over traditional eigenaxis maneuvers [13].

These optimal maneuvers were developed using the DIDO software designed by NPS professor Dr. Isaac M. Ross [14]. This software will be utilized here to develop optimal spacecraft antenna slew maneuvers. Details of the underlying optimal control theory utilized by DIDO and how DIDO was implemented are presented in Chapter VI.

Using the developed TDRS model, slew scenarios were created for use as optimal control problems. Each scenario was solved using the DIDO software to generate both

minimum-time, and minimum-momentum slews. The designed minimum time trajectories were implemented on a laboratory testbed to llustrate an approach by which these types of maneuvers can be applied to a physical system. This process and the results are presented in Chapter VII.

II. DEVELOPING A MULTI-BODY DYNAMIC MODEL

A. THE NEWTON-EULER APPROACH

In order to investigate the optimal performance of antenna slew maneuvers, a suitable dynamic model must be developed. Due to the complex and elaborate nature of spacecraft, consisting of multiple hinged or gimbaled joints fixed to lightweight structures, a multibody approach must be utilized. Three methodologies were explored to develop and validate a multibody dynamic model for the spacecraft antenna system. The first is the Newton-Euler approach described by Eric Stoneking [15]. This was the main approach used here to generate a general multibody model in MATLAB. This model was then tailored to fit a variety of systems, including a satellite test case, and a laboratory model, both using TDRS as an operational baseline. A variety of other more simple models were developed to illustratre the process, and to troubleshoot and validate the MATLAB code. Each system was also modeled in SimMechanics software suite of Simulink in order to perform additional verification and validation on the MATLAB model. Lastly, the SymPy program, which implements Kane's method, was also utilized for verification and validation purposes. These latter two methods will be discussed in Chapters III and IV. It should be noted that many software packages exist for simulating a multi-body system. However, it is difficult to use their output to support optimal control design. Hence, a suitable model for this purpose had to be developed as part of this thesis.

The procedure outlined in [15] allows for the straightforward derivation of a very complex, multibody system that can be tailored to any jointed-type spacecraft. A general procedure is described to construct equations of motion in matrix form for an N link system, where N is the number of independently movable bodies, or links, in the system. The method is focused around utilizing a set of state variables, organized as a column vector. These state variables can be selected according to the specific problem being analyzed, but it is most convenient to utilize variables such as the body's rotational velocity, ω, or the translational velocity, v. It will be shown that this selection of primary state variables will allow for the easiest propagation of the model in order to solve for the dynamics of the system. Other state variables are utilized and will be

discussed further as the approach is detailed. These state variables lead directly do a selection of dynamics variables. For example, ω to $\dot{\omega}$, and v to $\dot{v}$. It is these dynamics variables that will be solved for using this process. The principle behind selecting the dynamics variables is that each one can be utilized to solve for other desired states by integrating a set of differential equations. For example, $\dot{\omega}$ could be used to calculate rotational velocity, which could then be used to calculate rotational position. Once determined, the dynamics variables (populating a state column vector) are left multiplied by a mass matrix and set equal to a set of constraint terms in the solution vector. Together, the state vector, mass matrix, and constraint vector make up the equations of motion of the system for translation and attitude.

In [15], Stoneking outlines procedures for handling both spherical joints, and gimbaled joints. The processes are similar, but utilize different details for including joint or gimbal constraints, and for reducing the joint forces in the dynamics equations. One key aspect is that as joints and links are added to the system, the complexity rapidly grows. The spherical joint system results in $9N-3$ equations, whereas the gimbaled system results in $6N$ equations. This is because the gimbaled system already includes some reduction of redundant terms simply by virtue of its construction. One of the key features of this procedure is the ability to build the system through inspection. To enable this, the mass matrix and constraint vectors are partitioned into smaller submatrices and subvectors. Each submatrix and subvector collects terms that serve similar functions and are structured in such a way that each can be constructed simply with knowledge of how the system links and joints are configured. The process is designed in such a way that generalized equations can be used to fill out each element of these submatrices and subvectors.

Furthermore, row operations are utilized to reduce the dimensionality of the system to $3N+3$ equations for both the spherical and gimbaled system. The mass matrix and constraint vector are restructured, and the state vector partitioned into subvectors as well. These new matrices and vectors are then manipulated to eliminate the equations involving redundant constraint forces and torques at the joints.

Overall, the method allows for straightforward construction of the equations of motion of a complex system, and provides a way to reduce the dimensionality and solve for the state variables of the system. This is all possible without the need to perform a lengthy derivation of individual equations of motion. While the following section describes the derivation of the equations of motion, it is meant as an aid to describe how the matrices are constructed and solved, but full derivation is not necessary to execute the method itself.

B. DERIVING THE EQUATIONS OF MOTION

While a system can be described by inspection using the generalized equations provided in [15], the equations of motion will be developed in full here to help explain the procedure. A generic three-link model was developed for use in both the test case and the laboratory model. This model was then restructured or truncated as necessary to suit various system configurations under investigation.

The first step is to define the system configuration and reference frames. The first link is considered the base of the system, while the subsequent links are connected by gimbaled joints as shown in Figure 5. An inertial reference frame is declared outside the model, while individual local reference frames are placed at the center of mass (CoM) of each link. These reference frames are aligned to each link in the manner most convenient for the problem. For example, it may be convenient to align the frames to the principle axes of inertia for each link, or it may be convenient to align frames to match gimbal structure or frames for spacecraft sensors.

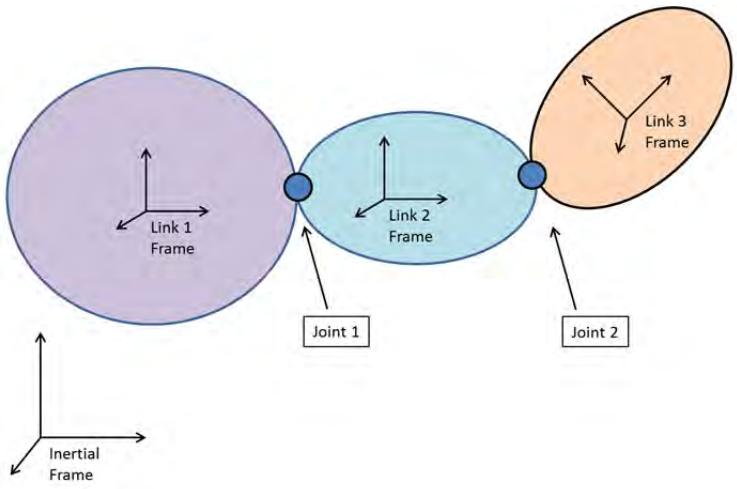


Figure 5. Link Frames for Generic Three-Body System

Once the frames are established, forces and torques can be assigned to each body and gimbal. External forces (F_1, F_2, F_3) and external torques (T_1, T_2, T_3) are applied at the center of mass of each body, as shown in Figure 6.

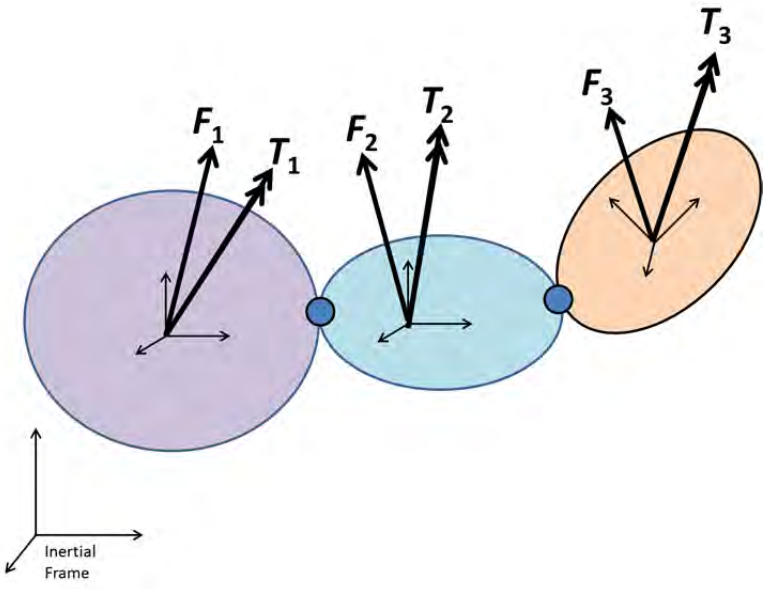


Figure 6. Link External Forces and Torques

Each external force and torque is assumed positive in the respective body frames. Furthermore, joint torques (T_{G1}, T_{G2}) and joint forces (F_{G1}, F_{G2}) are assigned to each

gimbaled joint. The torques and forces act equal in magnitude and opposite in sign on each body attached to that joint, as shown in Figure 7 through Figure 9. In order to generate the equations of motion, it is assumed that these forces act positive in the link frame farthest from the base (outer link) and act negative on the link frame closest to the base (inner link). For example in Figure 7, F_{G1} and T_{G1} act in the negative link one direction since it is the innermost link. However, in Figure 8, F_{G1} and T_{G1} act in the positive link two direction, since it is the outermost link. Notice also that the forces and torques are equal and opposite in magnitude. This sign orientation will also become evident when the equations of motion are constructed.

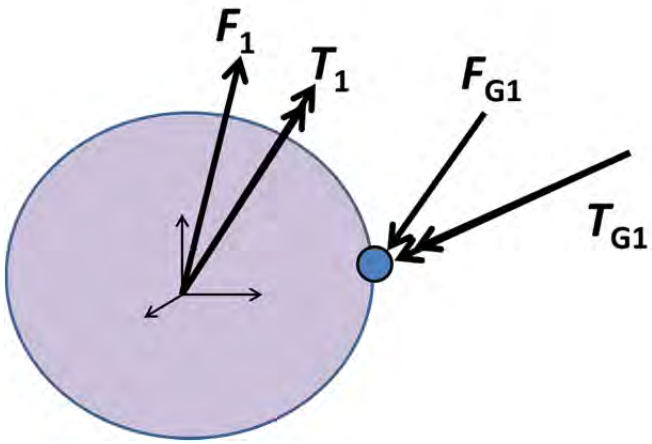


Figure 7. Link One Forces and Torques

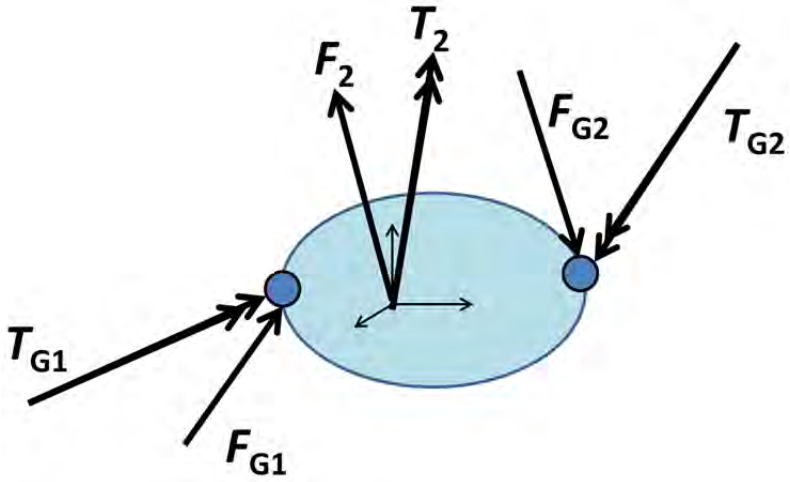


Figure 8. Link Two Forces and Torques

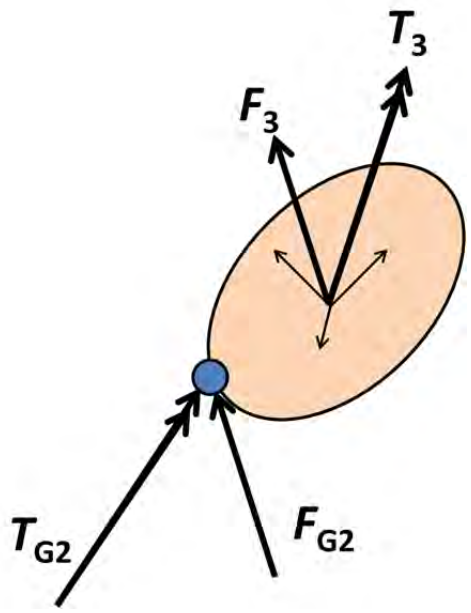


Figure 9. Link Three Forces and Torques

Each of these forces and torques is a 3x1 column vector of components acting along the *x*, *y*, and *z* axes. Frames of reference for forces, torques, and other variables will be fully explored later, but this stage of the derivation does require assumptions to be made for these basic directions. It should also be noted that some different direction assumptions are used in [15]. For example, [15] assumes the gimbal forces and torques to be positive on the inner link and negative on the outer link. While the selection is arbitrary at this point, the choice must be kept track of as it affects how the equations of motion are written and does result in final equations of motion expressed in a slightly different from those given in [15].

In order to generate the equations of motion of the system, each link is inspected individually using external and joint forces and torques. Two equations of motion are developed for each link, a translation equation and a rotation equation. Translational motion derivation begins with Newton's second law, where m is the scalar mass of the link, $\dot{v}$ is the local time rate of change of velocity of the link as a 3x1 column vector, and F is the sum of forces exerted on the link, also as a 3x1 column vector:

$$m\dot{v} = F \tag{2.1}$$

Summing the forces for each body results in the following equations:

$$m_1\dot{v}_1 = F_1 - F_{G1} \tag{2.2}$$

$$m_2\dot{v}_2 = F_2 + F_{G1} - F_{G2} \tag{2.3}$$

$$m_3\dot{v}_3 = F_3 + F_{G2}. \tag{2.4}$$

Rotation equations of motion can be similarly implemented by starting with Euler's equation, where I is the inertia tensor of the link (as a 3x3 matrix), ω and $\dot{\omega}$ are the local angular velocity and time rate of change of angular velocity of the link as 3x1 column vectors, T is the sum of torques exerted on the link as a 3x1 column vector, and $I\omega$ is the angular momentum of the link as a 3x1 column vector [16].

$$I\dot{\omega} = T - \omega \times I\omega \tag{2.5}$$

It is important to note that Equation (2.5) assumes there is no momentum storage in the system, such as reaction wheels. The torque term in Equation (2.5) can be further decomposed into three different torques: external torque, joint torque, and the moment exerted by forces on each link due to the joint. Assuming the external forces (F_1, F_2, F_3) are applied at the center of mass of the link, there are no couples generated by these forces. However, each link is still subject to the moments created by the joint constraint forces (F_{G1}, F_{G2}). Computing these torques requires the addition of a moment arm variable, r. There is a moment arm for each joint of each link. Link one and three each connect to one gimbal, while link two connects to two gimbals, resulting in a total of four moment arms. The moment arms are measured from the center of mass of each link to the respective gimbal and are measured in the local link frame. Each variable is denoted by a two number subscript. The first number in the subscript is the link and the second number is the gimbal. For example, the moment arm for link two to gimbal one is r_{21}. This results in the four moment arms $(r_{11}, r_{21}, r_{22}, r_{32})$, each as a 3x1 column vector, as shown in Figure 10.

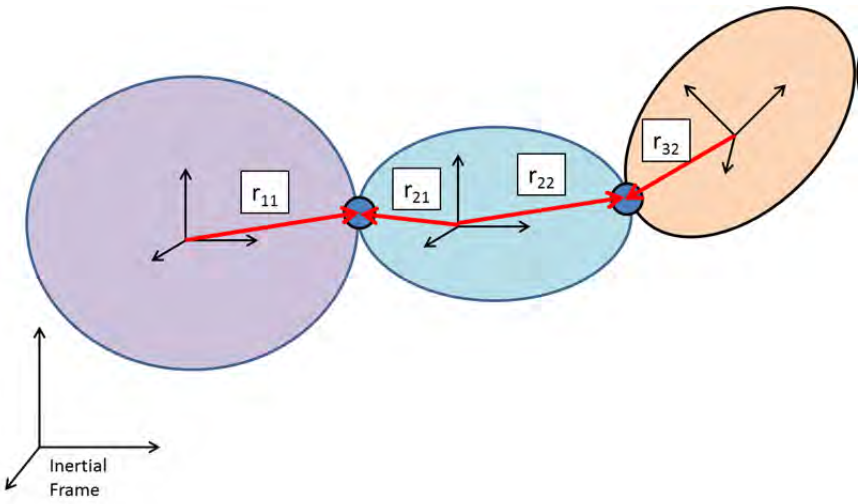


Figure 10. Link Moment Arms

With these variables in place, the rotation equations of motion can be generated.

$$I_1\dot{\omega}_1 = T_1 - T_{G1} - \left(r_{11} \times F_{G1}\right) - \left(\omega_1 \times I_1\omega_1\right) \tag{2.6}$$

$$I_2\dot{\omega}_2 = T_2 + T_{G1} - T_{G2} + \left(r_{21} \times F_{G1}\right) - \left(r_{22} \times F_{G2}\right) - \left(\omega_2 \times I_2\omega_2\right) \tag{2.7}$$

$$I_3\dot{\omega}_3 = T_3 + T_{G2} + \left(r_{32} \times F_{G2}\right) - \left(\omega_3 \times I_3\omega_3\right) \tag{2.8}$$

Notice the sign change on the joint torques between the equations of motion for each body. It is also important to note that each moment arm is established in the local link frame; this can result in some confusion when using these terms in conjunction with gimbal forces to generate moments. Notice also the duplication of joint forces and joint torques in Equation (2.6) through Equation (2.8). Each joint force and torque is applied equal and opposite on each link, as discussed previously. However, in order to properly orient each force and torque to each link as the system moves, coordinate transformations must be applied. These coordinate transformations will be necessary for a variety of components, not just forces and torques. However, for this stage of the derivation, it is easiest to leave the equations in a generic reference frame. The equations of motion will be derived fully in accordance with the procedure in [15], and then proper coordinate transformations will be applied where necessary. This delay in applying coordinate transformations makes it doubly important that proper attention is given to tracking reference frame orientations and sign assumptions throughout the derivation.

For example, suppose link two is a simple rod, one meter long with a gimbal at each end. The x-axis of link two is aligned to the length of the rod, positive toward gimbal two, with a center of mass at 0.5 meters along the length. This results in a r_{21} of [-0.5; 0; 0] meters since gimbal one is in the negative direction from the center of mass. Alternatively, r_{22} is [0.5; 0; 0] meters since gimbal two is in the positive direction from the center of mass. The moment arm r_{21} generates a moment when crossed with F_{G1}. Since F_{G1} is assumed positive, the torque generated by this cross product is assumed positive, $(r_{21} \times F_{G1})$. However, in reality, the generated torque would be negative in the link frame due to the negative first element value of r_{21}. Conversely, $(r_{22} \times F_{G2})$ is assumed negative due to F_{G2} being negative in the link 2 frame. Since the elements of r_{22} are positive, this results in a negative torque in the link frame. Errors in the assumptions on the signs of the forces will become apparent when the system is simulated.

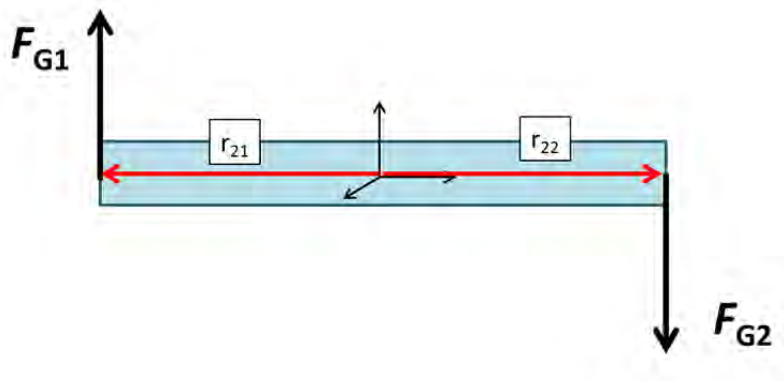


Figure 11. Example Layout of Link Moment Arms (Link 2)

Throughout the derivation of equations of motion, many of these types of arbitrary assumptions must be made regarding directions and frames. While the decisions can be made to accommodate specific problems or setups, they will not change the functionality of the final equations of motion as long as they are properly tracked and used uniformly throughout the process. As mentioned above, any errors in the assumed

signs of the internal forces and moments will be obvious when the equations of motion are solved.

While Equations (2.2) through (2.4) and (2.6) through (2.8) are suitable for each link individually, they do not describe the system as a whole. To consider these links as a series of connected bodies, additional constraint equations are required. The translational velocities of the gimbals as points in space (not the internal angular rates of the gimbals) are solved for both links connected to that gimbal and set equal to one another. In other words, the velocity of the coincident point on two different links is the same since the links are joined at that point. To compare the two values, however, both velocities must be solved and set in the same frame. First, the velocities of each gimbal are calculated using the velocity of the center of mass, v, the rotational velocities of the link, ω, and the position of the gimbal, r [15].

$$V_{G1} = v_1 + (\omega_1 \times r_{11}) \tag{2.9}$$

$$V_{G1} = v_2 + (\omega_2 \times r_{21}) \tag{2.10}$$

$$V_{G1} = v_1 + (\omega_1 \times r_{11}) = v_2 + (\omega_2 \times r_{21}) \tag{2.11}$$

$$V_{G2} = v_2 + (\omega_2 \times r_{22}) \tag{2.12}$$

$$V_{G2} = v_2 + (\omega_2 \times r_{32}) \tag{2.13}$$

$$V_{G2} = v_2 + (\omega_2 \times r_{22}) = v_3 + (\omega_3 \times r_{32}) \tag{2.14}$$

The time derivative of Equations (2.11) and (2.14) are then taken. However, they must be compared in the same frame; the inertial frame will be assumed for this derivation. This leads to the generation of some additional terms due to centripetal acceleration as seen in Equation (2.15) [17].

$$^{N}\frac{d}{dt}(x) = {}^{1}\frac{d}{dt}(x) + {}^{N}\omega^{1} \times x \tag{2.15}$$

Equation (2.15) and the chain rule can be used to transform Equation (2.11) and Equation (2.14) into equations:

$$\begin{aligned}&\dot{v}_1+\left(\dot{\omega}_1\times r_{11}\right)+\left(\omega_1\times\dot{r}_{11}\right)+\left(\omega_1\times\left(\omega_1\times r_{11}\right)\right)=...\\&...\dot{v}_2+\left(\dot{\omega}_2\times r_{21}\right)+\left(\omega_2\times\dot{r}_{21}\right)+\left(\omega_2\times\left(\omega_2\times r_{21}\right)\right)\end{aligned}\tag{2.16}$$

$$\begin{aligned}&\dot{v}_2+\left(\dot{\omega}_2\times r_{22}\right)+\left(\omega_2\times\dot{r}_{22}\right)+\left(\omega_2\times\left(\omega_2\times r_{22}\right)\right)=...\\&...\dot{v}_3+\left(\dot{\omega}_3\times r_{32}\right)+\left(\omega_3\times\dot{r}_{32}\right)+\left(\omega_3\times\left(\omega_3\times r_{32}\right)\right)\end{aligned}.\tag{2.17}$$

However, since the $\dot{r}_{ij}=0$ for non-prismatic joints, these terms can be removed.

$$\dot{v}_1+\left(\dot{\omega}_1\times r_{11}\right)+\left(\omega_1\times\left(\omega_1\times r_{11}\right)\right)=\dot{v}_2+\left(\dot{\omega}_2\times r_{21}\right)+\left(\omega_2\times\left(\omega_2\times r_{21}\right)\right)\tag{2.18}$$

$$\dot{v}_2+\left(\dot{\omega}_2\times r_{22}\right)+\left(\omega_2\times\left(\omega_2\times r_{22}\right)\right)=\dot{v}_3+\left(\dot{\omega}_3\times r_{32}\right)+\left(\omega_3\times\left(\omega_3\times r_{32}\right)\right)\tag{2.19}$$

At this point, it is convenient to introduce some notation to simplify expression of the cross products and double cross products. Each vector cross product can be represented by transforming the left vector into a 3x3 skew symmetric matrix described in Equations (2.20) and (2.21) [15]. The matrices are constructed such that when the matrix is left multiplied by a vector, it is equivalent to the cross product (or double cross product) of the two vectors.

$$\tilde{x}=\begin{bmatrix}0&-x_3&x_2\\x_3&0&x_1\\x_2&x_1&0\end{bmatrix}\tag{2.20}$$

$$\bar{x}=\begin{bmatrix}-x_2{}^2-x_3{}^2&x_1x_2&x_1x_3\\x_2x_1&-x_3{}^2-x_1{}^2&x_2x_3\\x_3x_1&x_3x_2&-x_1{}^2-x_2{}^2\end{bmatrix}\tag{2.21}$$

Furthermore, in order set up the equations for future steps, the order of the cross products is reversed, which changes the sign of each cross product term. This simplifies Equations (2.16) and (2.17) into Equations (2.22) and (2.23).

$$\dot{v}_1-\tilde{r}_{11}\dot{\omega}_1+\bar{\omega}_1r_{11}=\dot{v}_2-\tilde{r}_{21}\dot{\omega}_2+\bar{\omega}_2r_{21}\tag{2.22}$$

$$\dot{v}_2-\tilde{r}_{22}\dot{\omega}_2+\bar{\omega}_2r_{22}=\dot{v}_3-\tilde{r}_{32}\dot{\omega}_3+\bar{\omega}_3r_{32}\tag{2.23}$$

Using Equations (2.2) through (2.4); Equations (2.6) through (2.8); and Equations (2.22) and (2.23), the dynamics of a three-link system can be described using a set of ordinary differential equations based on the variables, $\left(\dot{\omega}_1,\dot{\omega}_2,\dot{\omega}_3,\dot{\mathrm{v}}_1,\dot{\mathrm{v}}_2,\dot{\mathrm{v}}_3,\mathrm{F}_{G1},\mathrm{F}_{G2}\right)$

. External torques, external forces, joint torques and system configuration variables such as inertia and moment arms are considered inputs into the problem. Additional states, including rotational velocity $(\omega_1, \omega_2, \omega_3)$, rotational position $(\delta_1, \delta_2, \delta_3)$, translational velocity (v_1, v_2, v_3), and translational position (d_1, d_2, d_3), ay be obtained by integrating the base state variables.

C. TAILORING TO GIMBALED JOINTS

The equations of motion described in Section B can be further tailored to the desired gimbaled system by utilizing the gimbal angle rates to determine link angular rates. First, two new variables are introduced, θ and Γ. Each gimbal has a column vector, θ, which describes the Euler angles for the gimbaled joint. These angles are used to build kinematic differential equations, codified in the Γ matrix, which relates gimbal rates, $\dot{\theta}$, to link angular rotation rates, ω, as shown in Equation (2.24) and Equation (2.25) [18].

$$\omega_2 = \omega_1 + \Gamma_1 \dot{\theta}_1 \tag{2.24}$$

$$\omega_3 = \omega_2 + \Gamma_2 \dot{\theta}_2 = \omega_1 + \Gamma_1 \dot{\theta}_1 + \Gamma_2 \dot{\theta}_2 \tag{2.25}$$

Each Γ matrix will utilize a specific Euler sequence that the gimbal rotates through. The Euler sequences can be described as either body-fixed, or space-fixed and can use any number of sequential rotations (see [18] for details). For example, a body-three x-y-z sequence utilizes three successive rotations, each about the body axes. First, the body is rotated about its x-axis by θ_x. Next it is rotated about its y-axis by θ_y. Lastly it is rotated about the body z-axis by θ_z. Similarly, a body-three y-x-z rotation would rotate about the y-axis by θ_y, then about the x-axis by θ_x, then about the z-axis by θ_z. Rotations could also be made about a non-body fixed axis, the inertial axes for example. This is referred to as a space sequence. For example, a space-three x-y-z using the inertial frame would rotate the body around the inertial x-axis by θ_x, then the inertial y-axis by θ_y, and finally the inertial z-axis by θ_z. Which sequence is utilized, the number of rotations, and whether or not to use body or space sequences, is dependent on the

problem being solved. In this thesis, body-three x-y-z rotations were implemented for the gimbaled system. This choice results in the Γ matrix in Equation (2.26) [18].

$$\Gamma = \begin{bmatrix} \cos\theta_y \cos\theta_z & \sin\theta_z & 0 \\ -\cos\theta_y \sin\theta_z & \cos\theta_z & 0 \\ \sin\theta_y & 0 & 1 \end{bmatrix} \tag{2.26}$$

A variety of kimenatic differential equations based on body and space rotations are provided in [18]. However, their chosen notation is different from what is used here and is worth clarifying. An x-y-z rotation is assumed, either body fixed or space fixed. Here, subscripts for the angles are maintained as x, y, and z components. Therefore, the sequence would first rotate about the x-axis by θ_x, then the y-axis by θ_y, then the z-axis by θ_z. However, most literature, including [18] does not utilize the x,y,z notation; instead they use a 1,2,3 notation. However, the angles $(\theta_1, \theta_2, \theta_3)$ given in [18] are listed in sequential order of operations, not according to their relative axis. For example, for a 1–2-3 rotation, the sequence would rotate about the first axis by θ_1, then the second axis by θ_2, then the third axis by θ_3. However, for a 2–1-3 rotation, the sequence would rotate about the second axis by θ_1, then the first axis by θ_2, then the third by θ_3. This difference in notation can lead to confusion, so it is important that any assumptions are properly carried through the entire derivation.

Equation (2.24) and Equation (2.25) are differentiated using the chain rule to derive Equations (2.27) and Equation (2.28), which give the angular accelerations. These derivatives generate additional terms according to Equation (2.15). It is important to note that the additional cross product seems to have been incorrectly omitted in [15].

$$\dot{\omega}_2 = \dot{\omega}_1 + \dot{\Gamma}_1\dot{\theta}_1 + \Gamma_1\ddot{\theta}_1 + \omega_2 \times \Gamma_1\dot{\theta} \tag{2.27}$$

$$\begin{aligned} &\dot{\omega}_3 = \dot{\omega}_2 + \dot{\Gamma}_2\dot{\theta}_2 + \Gamma_2\ddot{\theta}_2 + \omega_3 \times \Gamma_2\dot{\theta}_2 = \ldots \\ &\ldots\dot{\omega}_1 + \dot{\Gamma}_1\dot{\theta}_1 + \Gamma_1\ddot{\theta}_1 + \omega_2 \times \Gamma_1\dot{\theta} + \dot{\Gamma}_2\dot{\theta}_2 + \Gamma_2\ddot{\theta}_2 + \omega_3 \times \Gamma_2\dot{\theta}_2 \end{aligned} \tag{2.28}$$

The equations can be further expanded by inserting the rates for link two and three (Equation (2.24) and Equation (2.25)) and the time rate of change of angular velocity of link two (Equation (2.27)).

$$\dot{\omega}_2 = \dot{\omega}_1 + \dot{\Gamma}_1\dot{\theta}_1 + \Gamma_1\ddot{\theta}_1 + (\omega_1 + \Gamma_1\dot{\theta}) \times \Gamma_1\dot{\theta} = \dot{\omega}_1 + \dot{\Gamma}_1\dot{\theta}_1 + \Gamma_1\ddot{\theta}_1 + \tilde{\omega}_1\Gamma_1\dot{\theta} \tag{2.29}$$

$$\begin{aligned}\dot{\omega}_3 &= \dot{\omega}_1 + \dot{\Gamma}_1\dot{\theta}_1 + \Gamma_1\ddot{\theta}_1 + \tilde{\omega}_1\Gamma_1\dot{\theta}_1 + \dot{\Gamma}_2\dot{\theta}_2 + \Gamma_2\ddot{\theta}_2 + (\omega_1 + \Gamma_1\dot{\theta}_1 + \Gamma_2\dot{\theta}_2) \times \Gamma_2\dot{\theta}_2 = \ldots \\ &\dot{\omega}_1 + \dot{\Gamma}_1\dot{\theta}_1 + \Gamma_1\ddot{\theta}_1 + \dot{\Gamma}_2\dot{\theta}_2 + \Gamma_2\ddot{\theta}_2 + \tilde{\omega}_1(\Gamma_1\dot{\theta}_1 + \Gamma_2\dot{\theta}_2) + (\Gamma_1\dot{\theta}_1 \times \Gamma_2\dot{\theta}_2)\end{aligned} \tag{2.30}$$

Inserting these equations back into Equations (2.22) and (2.23) yields Equations (2.31) and (2.32).

$$\dot{v}_1 - \tilde{r}_{11}\dot{\omega}_1 + \bar{\omega}_1 r_{11} = \dot{v}_2 - \tilde{r}_{21}\left(\dot{\omega}_1 + \dot{\Gamma}_1\dot{\theta}_1 + \Gamma_1\ddot{\theta}_1 + \omega_1 \times \Gamma_1\dot{\theta}\right) + \bar{\omega}_2 r_{21} \tag{2.31}$$

$$\begin{aligned}&\dot{v}_2 - \tilde{r}_{22}\left(\dot{\omega}_1 + \dot{\Gamma}_1\dot{\theta}_1 + \Gamma_1\ddot{\theta}_1 + \omega_1 \times \Gamma_1\dot{\theta}\right) + \bar{\omega}_2 r_{22} = \ldots \\ &\dot{v}_3 - \tilde{r}_{32}(\dot{\omega}_1 + \dot{\Gamma}_1\dot{\theta}_1 + \Gamma_1\ddot{\theta}_1 + \dot{\Gamma}_2\dot{\theta}_2 + \Gamma_2\ddot{\theta}_2 + \tilde{\omega}_1(\Gamma_1\dot{\theta}_1 + \Gamma_2\dot{\theta}_2) + (\Gamma_1\dot{\theta}_1 \times \Gamma_2\dot{\theta}_2)) + \bar{\omega}_3 r_{32}\end{aligned} \tag{2.32}$$

Rearranging the terms, equations for the time rate of change of translational velocities of link two and three can be derived.

$$\dot{v}_2 = \dot{v}_1 + (\tilde{r}_{21} - \tilde{r}_{11})\dot{\omega}_1 + \tilde{r}_{21}\Gamma_1\ddot{\theta}_1 + \tilde{r}_{21}\dot{\Gamma}_1\dot{\theta}_1 + \tilde{r}_{21}\tilde{\omega}_1\Gamma_1\dot{\theta} + \bar{\omega}_1 r_{11} - \bar{\omega}_2 r_{21} \tag{2.33}$$

$$\begin{aligned}\dot{v}_3 &= \dot{v}_2 + (\tilde{r}_{32} - \tilde{r}_{22})\dot{\omega}_1 + (\tilde{r}_{32}\Gamma_1 - \tilde{r}_{22}\Gamma_1)\ddot{\theta}_1 + \tilde{r}_{32}\Gamma_2\ddot{\theta}_2 \ldots \\ &\ldots - \tilde{r}_{22}\tilde{\omega}_1\Gamma_1\dot{\theta}_1 + \tilde{r}_{32}\dot{\Gamma}_1\dot{\theta}_1 - \tilde{r}_{22}\dot{\Gamma}_1\dot{\theta}_1 + \tilde{r}_{32}\dot{\Gamma}_2\dot{\theta}_2 + \tilde{r}_{32}\tilde{\omega}_1(\Gamma_1\dot{\theta}_1 + \Gamma_2\dot{\theta}_2) \ldots \\ &\ldots + \tilde{r}_{32}(\Gamma_1\dot{\theta}_1 \times \Gamma_2\dot{\theta}_2) + \bar{\omega}_2 r_{22} - \bar{\omega}_3 r_{32}\end{aligned} \tag{2.34}$$

The equation for link three can be further expanded by inserting the time rate of change of velocity of link two.

$$\begin{aligned}\dot{v}_3 &= \dot{v}_1 + (\tilde{r}_{32} - \tilde{r}_{22} + \tilde{r}_{21} - \tilde{r}_{11})\dot{\omega}_1 + (\tilde{r}_{32}\Gamma_1 - \tilde{r}_{22}\Gamma_1 + \tilde{r}_{21}\Gamma_1)\ddot{\theta}_1 + \tilde{r}_{32}\Gamma_2\ddot{\theta}_2 + \ldots \\ &\ldots(\tilde{r}_{32}\tilde{\omega}_1 + \tilde{r}_{21}\tilde{\omega}_1 - \tilde{r}_{22}\tilde{\omega}_1)\Gamma_1\dot{\theta} + (\tilde{r}_{32} - \tilde{r}_{22} + \tilde{r}_{21})\dot{\Gamma}_1\dot{\theta}_1 + \tilde{r}_{32}\dot{\Gamma}_2\dot{\theta}_2 + \ldots \\ &\ldots\tilde{r}_{32}\tilde{\omega}_1(\Gamma_2\dot{\theta}_2) + \tilde{r}_{32}(\Gamma_1\dot{\theta}_1 \times \Gamma_2\dot{\theta}_2) + \bar{\omega}_2 r_{22} - \bar{\omega}_3 r_{32} + \bar{\omega}_1 r_{11} - \bar{\omega}_2 r_{21}\end{aligned} \tag{2.35}$$

These equations can now be inserted into the equations of motion for translation and rotation for link two and three (link one Equations (2.2) and (2.6) are unaffected).

$$\begin{aligned}&m_2\dot{v}_1 + m_2(\tilde{r}_{21} - \tilde{r}_{11})\dot{\omega}_1 + m_2\tilde{r}_{21}\Gamma_1\ddot{\theta}_1 + m_2\tilde{r}_{21}\dot{\Gamma}_1\dot{\theta}_1 + m_2\tilde{r}_{21}\tilde{\omega}_1\Gamma_1\dot{\theta} + m_2\bar{\omega}_1 r_{11} - m_2\bar{\omega}_2 r_{21} = \ldots \\ &\ldots F_2 + F_{G1} - F_{G2}\end{aligned} \tag{2.36}$$

$$
\begin{aligned}
&m_3\dot{v}_1 + m_3(\tilde{r}_{32} - \tilde{r}_{22} + \tilde{r}_{21} - \tilde{r}_{11})\dot{\omega}_1 + m_3(\tilde{r}_{32}\Gamma_1 - \tilde{r}_{22}\Gamma_1 + \tilde{r}_{21}\Gamma_1)\ddot{\theta}_1 + m_3\tilde{r}_{32}\Gamma_2\ddot{\theta}_2 + ... \\
&...m_3(\tilde{r}_{32}\tilde{\omega}_1 + \tilde{r}_{21}\tilde{\omega}_1 - \tilde{r}_{22}\tilde{\omega}_1)\Gamma_1\dot{\theta} + m_3(\tilde{r}_{32} - \tilde{r}_{22} + \tilde{r}_{21})\dot{\Gamma}_1\dot{\theta}_1 + m_3\tilde{r}_{32}\dot{\Gamma}_2\dot{\theta}_2 + ... \\
&...m_3\tilde{r}_{32}(\tilde{\omega}_1\Gamma_2\dot{\theta}_2) + m_3\tilde{r}_{32}(\Gamma_1\dot{\theta}_1 \times \Gamma_2\dot{\theta}_2) + m_3(\bar{\omega}_2 r_{22} - \bar{\omega}_3 r_{32} + \bar{\omega}_1 r_{11} - \bar{\omega}_2 r_{21}) = ... \\
&...F_3 + F_{G2}
\end{aligned}
\tag{2.37}
$$

$$
I_2\dot{\omega}_1 + I_2\dot{\Gamma}_1\dot{\theta}_1 + I_2\Gamma_1\ddot{\theta}_1 + I_2\tilde{\omega}_1\Gamma_1\dot{\theta} = T_2 + T_{G1} - T_{G2} + \tilde{r}_{21}F_{G1} - \tilde{r}_{22}F_{G2} - \tilde{\omega}_2\left(I_2\omega_2\right) \tag{2.38}
$$

$$
\begin{aligned}
&I_3\dot{\omega}_1 + I_3\dot{\Gamma}_1\dot{\theta}_1 + I_3\Gamma_1\ddot{\theta}_1 + I_3\dot{\Gamma}_2\dot{\theta}_2 + I_3\Gamma_2\ddot{\theta}_2 + I_3(\tilde{\omega}_1\Gamma_2\dot{\theta}_2) + I_3(\Gamma_1\dot{\theta}_1 \times \Gamma_2\dot{\theta}_2) = ... \\
&...T_3 + T_{G2} + \tilde{r}_{32}F_{G2} - \tilde{\omega}_3\left(I_3\omega_3\right)
\end{aligned}
\tag{2.39}
$$

Equations (2.2), (2.6), and (2.36) through (2.39) fully describe the system using the translational and angular acceleration for link one, the gimbal acceleration, and the gimbal constraint forces $\left(\dot{\omega}_1, \ddot{\theta}_1, \ddot{\theta}_2, \dot{\mathrm{v}}_1, \mathrm{F}_{\mathrm{G1}}, \mathrm{F}_{\mathrm{G2}}\right)$. External forces; external torques; gimbal torques; position and attitude velocities of link two and three; and gimbal positions and rates are all considered inputs or known at each instant in time. In order to solve this system of six equations, a matrix based approach is outlined in [15], which is described next in Chapter III.

THIS PAGE INTENTIONALLY LEFT BLANK

III. SOLVING THE EQUATIONS OF MOTION

A. IMPLEMENTING EQUATIONS OF MOTION IN MATRIX FORM

First, the six equations of motion developed in Chapter II are rearranged to place the terms that include the dynamics variables and joint forces $(\dot{\omega}_1, \ddot{\theta}_1, \ddot{\theta}_2, \dot{\mathrm{v}}_1, \mathrm{F}_{G1}, \mathrm{F}_{G2})$ on one side, and the terms that include the inputs or knowns on the other.

$$m_1\dot{v}_1 + F_{G1} = F_1 \tag{3.1}$$

$$\begin{aligned} &m_2\dot{v}_1 + m_2(\tilde{r}_{21} - \tilde{r}_{11})\dot{\omega}_1 + m_2\tilde{r}_{21}\Gamma_1\ddot{\theta}_1 - F_{G1} + F_{G2} = \ldots \\ &-m_2\tilde{r}_{21}\dot{\Gamma}_1\dot{\theta}_1 - m_2\tilde{r}_{21}\tilde{\omega}_1\Gamma_1\dot{\theta} - m_2\bar{\omega}_1 r_{11} + m_2\bar{\omega}_2 r_{21} + F_2 \end{aligned} \tag{3.2}$$

$$\begin{aligned} &m_3\dot{v}_1 + m_3(\tilde{r}_{21} - \tilde{r}_{11} + \tilde{r}_{32} - \tilde{r}_{22})\dot{\omega}_1 + m_3(\tilde{r}_{21}\Gamma_1 + \tilde{r}_{32}\Gamma_1 - \tilde{r}_{22}\Gamma_1)\ddot{\theta}_1 + m_3\tilde{r}_{32}\Gamma_2\ddot{\theta}_2 \ldots \\ &\ldots - F_{G2} = -m_3(\tilde{r}_{21}\dot{\Gamma}_1 - \tilde{r}_{22}\dot{\Gamma}_1 + \tilde{r}_{32}\dot{\Gamma}_1)\dot{\theta}_1 - m_3(\tilde{r}_{32}\tilde{\omega}_1 + \tilde{r}_{21}\tilde{\omega}_1 - \tilde{r}_{22}\tilde{\omega}_1)\Gamma_1\dot{\theta}_1 \ldots \\ &\ldots - m_3\tilde{r}_{32}\dot{\Gamma}_2\dot{\theta}_2 - m_3\tilde{r}_{32}(\tilde{\omega}_1\Gamma_2\dot{\theta}_2) - m_3\tilde{r}_{32}(\Gamma_1\dot{\theta}_1 \times \Gamma_2\dot{\theta}_2) \ldots \\ &\ldots - m_3(\bar{\omega}_1 r_{11} - \bar{\omega}_2 r_{21} + \bar{\omega}_2 r_{22} - \bar{\omega}_3 r_{32}) + F_3 \end{aligned} \tag{3.3}$$

$$I_1\dot{\omega}_1 + \tilde{r}_{11}F_{G1} = T_1 - T_{G1} - \tilde{\omega}_1\left(I_1\omega_1\right) \tag{3.4}$$

$$I_2\dot{\omega}_1 + I_2\Gamma_1\ddot{\theta}_1 - \tilde{r}_{21}F_{G1} + \tilde{r}_{22}F_{G2} = T_2 + T_{G1} - T_{G2} - \tilde{\omega}_2\left(I_2\omega_2\right) - I_2\dot{\Gamma}_1\dot{\theta}_1 - I_2\tilde{\omega}_1\Gamma_1\dot{\theta} \tag{3.5}$$

$$\begin{aligned} &I_3\dot{\omega}_1 + I_3\Gamma_1\ddot{\theta}_1 + I_3\Gamma_2\ddot{\theta}_2 - \tilde{r}_{32}F_{G2} = \ldots \\ &T_3 + T_{G2} - \tilde{\omega}_3\left(I_3\omega_3\right) + I_3\dot{\Gamma}_1\dot{\theta}_1 + I_3\dot{\Gamma}_2\dot{\theta}_2 - I_3(\tilde{\omega}_1\Gamma_2\dot{\theta}_2) - I_3(\Gamma_1\dot{\theta}_1 \times \Gamma_2\dot{\theta}_2) \end{aligned} \tag{3.6}$$

Note, that while there are some time-rate-of-change components on the right hand side, these do not include the defined dynamics variables and will be solved for at each instant in time by integrating the equations of motion. The dynamics variables are then pulled out and placed into a dynamics variable column vector. The remaining terms on the left compose a mass matrix, while the terms to the right side of the equation are appropriately structured as a column vector of constraint terms.

$$\begin{bmatrix} I_1 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \tilde{r}_{11} & \boldsymbol{0} \\ I_2 & I_2\Gamma_1 & \boldsymbol{0} & \boldsymbol{0} & -\tilde{r}_{21} & \tilde{r}_{22} \\ I_3 & I_3\Gamma_1 & I_3\Gamma_2 & \boldsymbol{0} & \boldsymbol{0} & -\tilde{r}_{32} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{m}_1 & \boldsymbol{1} & \boldsymbol{0} \\ m_2(\tilde{r}_{21}-\tilde{r}_{11}) & m_2(\tilde{r}_{21}\Gamma_1) & \boldsymbol{0} & \boldsymbol{m}_2 & \boldsymbol{-1} & \boldsymbol{1} \\ m_3(\tilde{r}_{21}-\tilde{r}_{11}+\tilde{r}_{32}-\tilde{r}_{22}) & m_3(\tilde{r}_{21}\Gamma_1+\tilde{r}_{32}\Gamma_1-\tilde{r}_{22}\Gamma_1) & m_3\tilde{r}_{32}\Gamma_2 & \boldsymbol{m}_3 & \boldsymbol{0} & \boldsymbol{-1} \end{bmatrix} \begin{Bmatrix} \dot{\omega}_1 \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \dot{v}_1 \\ F_{G1} \\ F_{G2} \end{Bmatrix} = \ldots$$

$$\ldots \begin{Bmatrix} T_1 - T_{G1} - \tilde{\omega}_1\left(I_1\omega_1\right) \\ T_2 + T_{G1} - T_{G2} - \tilde{\omega}_2\left(I_2\omega_2\right) - I_2(\dot{\Gamma}_1\dot{\theta}_1) - I_2\tilde{\omega}_1\Gamma_1\dot{\theta} \\ T_3 + T_{G2} - \tilde{\omega}_3\left(I_3\omega_3\right) + I_3(\dot{\Gamma}_1\dot{\theta}_1) + I_3(\dot{\Gamma}_2\dot{\theta}_2) - I_3(\tilde{\omega}_1\Gamma_2\dot{\theta}_2) - I_3(\Gamma_1\dot{\theta}_1 \times \Gamma_2\dot{\theta}_2) \\ F_1 \\ -m_2\tilde{r}_{21}\dot{\Gamma}_1\dot{\theta}_1 - m_2\tilde{r}_{21}\tilde{\omega}_1\Gamma_1\dot{\theta} - m_2\bar{\omega}_1 r_{11} + m_2\bar{\omega}_2 r_{21} + F_2 \\ -m_3(\tilde{r}_{21}\dot{\Gamma}_1 - \tilde{r}_{22}\dot{\Gamma}_1 + \tilde{r}_{32}\dot{\Gamma}_1)\dot{\theta}_1 - m_3(\tilde{r}_{32}\tilde{\omega}_1 + \tilde{r}_{21}\tilde{\omega}_1 - \tilde{r}_{22}\tilde{\omega}_1)\Gamma_1\dot{\theta}_1 - m_3\tilde{r}_{32}\dot{\Gamma}_2\dot{\theta}_2 - m_3\tilde{r}_{32}(\tilde{\omega}_1\Gamma_2\dot{\theta}_2) + \ldots \\ -m_3\tilde{r}_{32}(\Gamma_1\dot{\theta}_1 \times \Gamma_2\dot{\theta}_2) - m_3(\bar{\omega}_1 r_{11} - \bar{\omega}_2 r_{21} + \bar{\omega}_2 r_{22} - \bar{\omega}_3 r_{32}) + F_3 \end{Bmatrix} \quad (3.7)$$

Each term in the mass matrix of Equation (3.7) is itself a 3x3 matrix. The mass terms $(\boldsymbol{m}_1, \boldsymbol{m}_2, \boldsymbol{m}_3)$ are effectively the three mass scalars of each link multiplied by a 3x3 identity matrix to yield a 3x3 mass matrix. Similarly, the **1**, **-1**, and **0** terms are all scalar values multiplied by a 3x3 identity matrix. This yields an 18x18 matrix. Each variable $(\dot{\omega}_1, \ddot{\theta}_1, \ddot{\theta}_2, \dot{v}_1, F_{G1}, F_{G2})$ consists of a 3x1 column vector, resulting in a total state vector of 18x1 terms. Likewise, each element in the rightmost column vector represents a 3x1 vector, resulting in a total constraint vector of 18x1 terms. This matrix system represents 18 dynamics equations based on 18 variables.

Each matrix may be further partitioned into submatrices. This is useful for two reasons. First, one of the main goals is to outline a process to describe complex systems and build their dynamics by inspection. By partitioning the larger dynamics and constraints into submatrices, it is easier to build each component without first describing the full equations of motion for each link as was performed here. Stoneking outlines a process generalized to *N* bodies and *N*-1 joints, and then provides general equations to fill out each submatrix by inspection [15]. The mass matrix is partitioned into five submatrices, (I, P, Π, μ, J) as well as a 3*N*x3 matrix populated with zeros. The constraint matrix is broken into two submatrices, (T, F). These two sets of partitions will be discussed in further detail in Section C. The complexity of the equations of motion, and thus the submatrices, grows rapidly with each additional gimbal and link, hence the need for a simple, straightforward approach based on building each submatrix individually. The three-body, two-gimbal system is simple enough to describe in equation form before matrix implementation, therefore the generalized equations were not utilized here. Instead the equations were derived fully, and then translated into matrix form. The generalized equations as well as a derivation for a similar three-body system using spherical joints instead of gimbaled joints can be found in [15].

The second reason for using submatrices is to reduce the state vector by eliminating the joint equations. The process described thus far has resulted in a system of 6*N* equations, or 18 total equations for a three body system. However, it is assumed that knowledge of the joint forces $(\mathrm{F}_{G1}, \mathrm{F}_{G2})$ is not required and these forces can be

incorporated into the overall problem by row operations. This eliminates the 3*N*-3 joint forces, resulting in a 3*N*+3 system. In order to perform this operation, Stoneking further partitions the mass matrix into four submatrices, (A,R,S,U), and the constraint vector into two subvectors, (T',F'). Furthermore, the state vector is partitioned into two subvectors, $(\dot{x},F_G)$. These three partitions will be discussed in further detail in Section D. While the matrices and vectors of Equation (3.7) can be converted directly into these submatrices and subvectors, it is convenient to first partition the matrices and vectors into the (I,P,Π,μ,J) submatrices. This allows for an easier understanding of what each term in the mass matrix contributes to the overall equations of motion, as well as how each term interacts with the state variables.

B. REFERENCE FRAMES

Before partitioning can begin, decisions need to be made regarding coordinate frames of each variable and of each equation of motion as a whole. For simplicity, Equation (3.7) was provided in general form. No declarations have been made as to what coordinate frame each term is in, so the equations are presented in a purely general nature, with no regard to coordinate frame de-confliction. In practice, however, the coordinate frames of each variable must be considered carefully and managed so that each equation of motion, codified in each row, utilizes a consistent coordinate frame. The selection of coordinate frames is ultimately arbitrary as long as proper conversion procedures are followed; a vector expressed in frame A is equivalent to that same vector expressed in frame B. However, some frames prove to be more convenient than others and will be selected as such. To begin, it is most convenient to express all forces as well as the translational velocity and acceleration of link one in the inertial frame. The remaining terms are expressed in the local frame of each link (see Table 1). Gimbal torques are defined in the outer link frame, as discussed in Chapter II.

Inertial Frame	Link One Frame	Link Two Frame	Link Three Frame
$v_1, \dot{v}_1$	I_1	I_2	I_3
F_1, F_2, F_3	$r_{11}, \tilde{r}_{11}$	$r_{21}, \tilde{r}_{21}, r_{22}, \tilde{r}_{22}$	$r_{32}, \tilde{r}_{32}$
F_{G1}, F_{G2}	$\omega_1, \dot{\omega}_1, \tilde{\omega}_1, \bar{\omega}_1$	$\omega_2, \dot{\omega}_2, \tilde{\omega}_2, \bar{\omega}_2$	$\omega_3, \dot{\omega}_3, \tilde{\omega}_3, \bar{\omega}_3$
		$\Gamma_1, \dot{\Gamma}_1, \theta_1, \dot{\theta}_1, \ddot{\theta}_1$	$\Gamma_2, \dot{\Gamma}_2, \theta_2, \dot{\theta}_2, \ddot{\theta}_2$
	T_1	T_2, T_{G1}	T_3, T_{G2}

Table 1. Variable Reference Frames.

While these frames were chosen because of the convenience, each equation of motion uses a number of terms from each frame. Therefore, the vectors must be placed in a common coordinate frame for each equation of motion, or row of Equation (3.7). Each row can utilize a different frame, but the frames must be consistent across that row. To ensure the frames match, direction cosine matrices (DCM) will be used to convert between frames. A DCM is an orthogonal 3x3 matrix that converts a vector from one frame to another utilizing a series of subsequent rotations based on relative angles between the two frames. Literature utilizes a vast array of notation schemes for DCMs, but the notation that will be used here is ${}^{a}C^{b}$ where C indicates a DCM, b is the original frame of the given vector, and a is the desired frame in which the vector is to be expressed. For example, if vector x is originally expressed in frame b, indicated by ${}^{b}x$, and it is desired to utilize vector x in frame a, indicated by ${}^{a}x$, then the DCM would be used as follows [16]:

$$ {}^{a}x = \left[{}^{a}C^{b} \right] {}^{b}x . \tag{3.8} $$

The same DCM can be utilized to transform from frame a back to frame b utilizing Equation (3.9), as shown in Equations (3.10) and (3.11) [16].

$$ C^{T}C = I = CC^{T} \tag{3.9} $$

$$ \left[{}^{a}C^{b} \right]^{T} {}^{a}x = \left[{}^{a}C^{b} \right]^{T} \left[{}^{a}C^{b} \right] {}^{b}x \tag{3.10} $$

$$ \left[{}^{a}C^{b} \right]^{T} {}^{a}x = \left[{}^{b}C^{a} \right] {}^{a}x = {}^{b}x \tag{3.11} $$

Similar to the Euler sequences previously used to construct the Γ matrix, the rotations of a DCM can be expressed as a sequence of rotations in either body-fixed or space-fixed. Here, it is easiest to use a body-fixed x-y-z rotation when converting back and forth between each link due to the nature of the gimbaled joints and the configuration of the system. This is based on the assumption that the multiple axis gimbals physically perform one rotation at a time, each rotation built upon the last, using a system of subsequent single axis rotations. This assumption would not hold for a spherical joint, for example a ball joint, which can perform multiple rotations simultaneously. The gimbal angles within the vectors θ_1 and θ_2 will be used to form a DCM to convert between frames one and two, and between frames two and three, respectively. This yields the following DCMs [18]

$$ {}^{2}C^{1} = \begin{bmatrix} \cos\theta_{1y}\cos\theta_{1z} & \sin\theta_{1x}\sin\theta_{1y}\cos\theta_{1z}+\cos\theta_{1x}\sin\theta_{1z} & -\cos\theta_{1x}\sin\theta_{1y}\cos\theta_{1z}+\sin\theta_{1x}\sin\theta_{1z} \\ -\cos\theta_{1y}\sin\theta_{1z} & -\sin\theta_{1x}\sin\theta_{1y}\sin\theta_{1z}+\cos\theta_{1x}\cos\theta_{z} & \cos\theta_{1x}\sin\theta_{1y}\sin\theta_{1z}+\sin\theta_{1x}\cos\theta_{1z} \\ \sin\theta_{1y} & -\sin\theta_{1x}\cos\theta_{1y} & \cos\theta_{1x}\cos\theta_{1y} \end{bmatrix} \tag{3.12} $$

$$ {}^{3}C^{2} = \begin{bmatrix} \cos\theta_{2y}\cos\theta_{2z} & \sin\theta_{2x}\sin\theta_{2y}\cos\theta_{2z}+\cos\theta_{2x}\sin\theta_{2z} & -\cos\theta_{2x}\sin\theta_{2y}\cos\theta_{1z}+\sin\theta_{2x}\sin\theta_{2z} \\ -\cos\theta_{2y}\sin\theta_{2z} & -\sin\theta_{2x}\sin\theta_{2y}\sin\theta_{2z}+\cos\theta_{2x}\cos\theta_{2z} & \cos\theta_{2x}\sin\theta_{2y}\sin\theta_{2z}+\sin\theta_{2x}\cos\theta_{2z} \\ \sin\theta_{2y} & -\sin\theta_{2x}\cos\theta_{2y} & \cos\theta_{2x}\cos\theta_{2y} \end{bmatrix} \tag{3.13} $$

Kane, Likins, and Levinson provide DCMs for any combination of body or space conversions [18]. However, each vector in [18] is considered to be a row vector, whereas vectors are treated here as column vectors. Therefore, the DCMs provided in [18] must be transposed to give the same in the DCMs as Equations (3.12) and (3.13). Furthermore, the same notational discrepancy noted before in Chapter II for the kinematic differential equations applies to the DCMs provided in [18].

While the gimbal angles are ideal for building the transformations between each link, the transformation between the inertial frame and link one still remains. Since link one serves as the main body of the system, it will rotate about each axis simultaneously rather than one rotation performed after another is complete. This does not lend itself easily to the classical build described in Equations (3.12) and (3.13). Furthermore, in order to avoid singularities that could result from application of a traditional Euler angle

(roll, pitch, yaw) approach, a set of four quaternion parameters (q_1, q_2, q_3, q_4) are utilized instead. A thorough explanation of quaternions will not be given here. Instead, equations and principles useful to the specific system and problem of interest in this thesis will be discussed and implemented. A full discussion of quaternions is presented in [16].

Unlike Euler angles, quaternions are not independent; they are constrained by Equation (3.14) [16]. This additional constraint term is what allows quaternions to avoid many of the geometric singularities that arise when performing unrestrained rotations using Euler angles [16].

$$q_1^{\,2} + q_2^{\,2} + q_3^{\,2} + q_4^{\,2} = 1 \tag{3.14}$$

Quaternions are derived from a single rotation of the system about an eigenaxis. The eigenaxis is a vector that is fixed in two frames. For the purpose of this system, the frames are the body frame of link one and the inertial reference frame. This is represented in Equation (3.15), where e represents the eigenaxis and subsequent components, L represents the components of the link one frame, and N the components of the inertial frame [16].

$$e = e_1 L_1 + e_2 L_2 + e_3 L_3 = e_1 N_1 + e_2 N_2 + e_3 N_3 \tag{3.15}$$

The quaternions are defined in Equation (3.16), where e_1, e_2, and e_3, are the vector components of the eigenaxis of rotation, and θ is the angle by which the system is rotated about that axis [16].

$$\begin{Bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{Bmatrix} = \begin{Bmatrix} \mathrm{e}_1 \sin\dfrac{\theta}{2} \\ \mathrm{e}_2 \sin\dfrac{\theta}{2} \\ \mathrm{e}_3 \sin\dfrac{\theta}{2} \\ \cos\dfrac{\theta}{2} \end{Bmatrix} \tag{3.16}$$

Consider a body frame, L, initially coincident with the inertial frame, N, then rotated about only the x-axis. Since the x-axis of the body and the x-axis of the inertial frame remain coincident through the rotation, the eigenaxis is $[1,0,0]^{\mathrm{T}}$. This results in the four quaternions:

$$q_x = \begin{bmatrix} q_{x1} \\ q_{x2} \\ q_{x3} \\ q_{x4} \end{bmatrix} = \begin{Bmatrix} \sin\frac{\theta_x}{2} \\ 0 \\ 0 \\ \cos\frac{\theta_x}{2} \end{Bmatrix}. \tag{3.17}$$

The same process could be made for individual rotations about the y and z axes.

$$q_y = \begin{bmatrix} q_{y1} \\ q_{y2} \\ q_{y3} \\ q_{y4} \end{bmatrix} = \begin{Bmatrix} 0 \\ \sin\frac{\theta_y}{2} \\ 0 \\ \cos\frac{\theta_y}{2} \end{Bmatrix} \tag{3.18}$$

$$q_z = \begin{bmatrix} q_{z1} \\ q_{z2} \\ q_{z3} \\ q_{z4} \end{bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ \sin\frac{\theta_z}{2} \\ \cos\frac{\theta_z}{2} \end{Bmatrix} \tag{3.19}$$

Furthermore, successive rotations of quaternions can be determined using the quaternion multiplication as shown in Equation (3.20), where q is the first rotation and q' is the second rotation, resulting in a total rotation described by q'' [16]:

$$q'' = \begin{bmatrix} q'_4 & q'_3 & -q'_2 & q'_1 \\ -q'_3 & q'_4 & q'_1 & q'_2 \\ q'_2 & -q'_1 & q'_4 & q'_3 \\ -q'_1 & -q'_2 & -q'_3 & q'_4 \end{bmatrix} \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{Bmatrix}. \tag{3.20}$$

Equations (3.17), (3.18), and (3.19) can be inserted one at a time into Equation (3.20) to yield the total x-y-z rotation. First, the rotations about the x and y axes are used to find the quaternions for an x-y rotation.

$$q_{xy} = \begin{bmatrix} \cos\frac{\theta_y}{2} & 0 & -\sin\frac{\theta_y}{2} & 0_1 \\ 0 & \cos\frac{\theta_y}{2} & 0 & \sin\frac{\theta_y}{2} \\ \sin\frac{\theta_y}{2} & 0 & \cos\frac{\theta_y}{2} & 0 \\ 0 & -\sin\frac{\theta_y}{2} & 0 & \cos\frac{\theta_y}{2} \end{bmatrix} \begin{Bmatrix} \sin\frac{\theta_x}{2} \\ 0 \\ 0 \\ \cos\frac{\theta_x}{2} \end{Bmatrix} = \begin{Bmatrix} \sin\frac{\theta_x}{2}\cos\frac{\theta_y}{2} \\ \cos\frac{\theta_x}{2}\sin\frac{\theta_y}{2} \\ \sin\frac{\theta_x}{2}\sin\frac{\theta_y}{2} \\ \cos\frac{\theta_x}{2}\cos\frac{\theta_y}{2} \end{Bmatrix} \quad (3.21)$$

Then the x-y quaternions are used with the z rotation quaternions to yield the quaternions for the total rotation.

$$q_{xyz} = \begin{bmatrix} \cos\frac{\theta_z}{2} & \sin\frac{\theta_z}{2} & 0 & 0 \\ -\sin\frac{\theta_z}{2} & \cos\frac{\theta_z}{2} & 0 & 0 \\ 0 & 0 & \cos\frac{\theta_z}{2} & \sin\frac{\theta_z}{2} \\ 0 & 0 & -\sin\frac{\theta_z}{2} & \cos\frac{\theta_z}{2} \end{bmatrix} \begin{Bmatrix} \sin\frac{\theta_x}{2}\cos\frac{\theta_y}{2} \\ \cos\frac{\theta_x}{2}\sin\frac{\theta_y}{2} \\ \sin\frac{\theta_x}{2}\sin\frac{\theta_y}{2} \\ \cos\frac{\theta_x}{2}\cos\frac{\theta_y}{2} \end{Bmatrix} = \begin{Bmatrix} \sin\frac{\theta_x}{2}\cos\frac{\theta_y}{2}\cos\frac{\theta_z}{2} + \cos\frac{\theta_x}{2}\sin\frac{\theta_y}{2}\sin\frac{\theta_z}{2} \\ \cos\frac{\theta_x}{2}\sin\frac{\theta_y}{2}\cos\frac{\theta_z}{2} - \sin\frac{\theta_x}{2}\cos\frac{\theta_y}{2}\sin\frac{\theta_z}{2} \\ \sin\frac{\theta_x}{2}\sin\frac{\theta_y}{2}\cos\frac{\theta_z}{2} + \cos\frac{\theta_x}{2}\cos\frac{\theta_y}{2}\sin\frac{\theta_z}{2} \\ \cos\frac{\theta_x}{2}\cos\frac{\theta_y}{2}\cos\frac{\theta_z}{2} - \sin\frac{\theta_x}{2}\sin\frac{\theta_y}{2}\sin\frac{\theta_z}{2} \end{Bmatrix} \quad (3.22)$$

Conversely, quaternions can be used to determine the components of a DCM, as shown in Equation (3.23) [16].

$$C = \begin{bmatrix} 1-2(q_2^{\,2}+q_3^{\,2}) & 2(q_1q_2+q_3q_4) & 2(q_1q_3-q_2q_4) \\ 2(q_2q_1-q_3q_4) & 1-2(q_1^{\,2}+q_3^{\,2}) & 2(q_2q_3+q_1q_4) \\ 2(q_3q_1+q_2q_4) & 2(q_3q_2-q_1q_4) & 1-2(q_1^{\,2}+q_3^{\,2}) \end{bmatrix} \quad (3.23)$$

Lastly, the time rate of change of the quaternion vector can be solved for using the current quaternions and the body angular velocity as shown in Equation (3.24) [16]. This will be used later to solve for the time history of the quaternion vector.

$$\dot{q} = \begin{Bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{Bmatrix} = \frac{1}{2}\begin{bmatrix} q_4 & -q_3 & q_2 & q_1 \\ q_3 & q_4 & -q_1 & q_2 \\ -q_2 & q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{bmatrix} \begin{Bmatrix} \omega 1 \\ \omega 2 \\ \omega 3 \\ 0 \end{Bmatrix} \quad (3.24)$$

Overall, Equation (3.12), Equation (3.13), and Equation (3.23) yield three DCMs to transform back and forth between the inertial frame and the body frames of link one,

two and three ${}^1C^N, {}^2C^1, {}^3C^2$. These DCMs can be transposed to yield DCMs to convert in the other direction, or stacked to yield DCMs between non-consecutive frames as follows:

$$ {}^N C^1 = \left[{}^1C^N \right]^T \tag{3.25} $$

$$ {}^1C^2 = \left[{}^2C^1 \right]^T \tag{3.26} $$

$$ {}^2C^3 = \left[{}^3C^2 \right]^T \tag{3.27} $$

$$ {}^2C^N = {}^2C^1 {}^1C^N \tag{3.28} $$

$$ {}^3C^N = {}^3C^2 {}^2C^1 {}^1C^N \tag{3.29} $$

$$ {}^3C^1 = {}^3C^2 {}^2C^1 . \tag{3.30} $$

With these DCMs defined, the dynamics and constraint matrices can be easily decomposed into submatrices and converted to convenient common frames.

C. SUBMATRIX PARTITIONING

To further cast the equations of motion into a form suitable for numerical simulation, the mass matrix is first broken into the five submatrices and a zero matrix.

$$ \left[\begin{array}{c|c|c} I & 0 & P \\ \hline \Pi & \mu & J \end{array}\right] = \left[\begin{array}{ccc|ccc} I_1 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \tilde{r}_{11} & \boldsymbol{0} \\ I_2 & I_2\Gamma_1 & \boldsymbol{0} & \boldsymbol{0} & -\tilde{r}_{21} & \tilde{r}_{22} \\ I_3 & I_3\Gamma_1 & I_3\Gamma_2 & \boldsymbol{0} & \boldsymbol{0} & -\tilde{r}_{32} \\ \hline \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{m}_1 & \boldsymbol{1} & \boldsymbol{0} \\ m_2(\tilde{r}_{21} - \tilde{r}_{11}) & m_2(\tilde{r}_{21}\Gamma_1) & \boldsymbol{0} & \boldsymbol{m}_2 & \boldsymbol{-1} & \boldsymbol{1} \\ m_3(\tilde{r}_{21} - \tilde{r}_{11} + \tilde{r}_{32} - \tilde{r}_{22}) & m_3(\tilde{r}_{21}\Gamma_1 + \tilde{r}_{32}\Gamma_1 - \tilde{r}_{22}\Gamma_1) & m_3\tilde{r}_{32}\Gamma_2 & \boldsymbol{m}_3 & \boldsymbol{0} & \boldsymbol{-1} \end{array}\right] \tag{3.31} $$

The I matrix relates the inertia of each link to the attitude dynamics of that link. The first column is multiplied by the time rate of change of angular velocity and denotes how that link is affected by the motion of link one, the base. The second and third columns multiply by the time rate of change of gimbal velocities, and account for the

additional rotation of each link due to the motion of the gimbal. Still referenced to an arbitrary general frame, this results in Equation (3.32), where each element is a 3x3 matrix, resulting in a 9x9 matrix.

$$I = \begin{bmatrix} I_1 & \boldsymbol{0} & \boldsymbol{0} \\ I_2 & I_2\Gamma_1 & \boldsymbol{0} \\ I_3 & I_3\Gamma_1 & I_3\Gamma_2 \end{bmatrix} \tag{3.32}$$

The most convenient frame for the I matrix is the body frame for each row. Row one will be standardized to the body frame of link one, row two to link two, and row three to link three. Each 3x3 inertia matrix is natively in the correct frame, but the first column requires a DCM to convert $\dot{\omega}_1$ (that each row will be multiplied by) to the correct frame. Since the dynamics variables and joint forces are multiplied through each row during matrix multiplication, and each row requires a different frame, DCMs cannot be applied directly in the state vector. Therefore, DCMs required to convert the dynamics variables and joint forces will also be set in place in the dynamics submatrices. Similarly, the Γ terms, which become $\Gamma\ddot{\theta}$ when matrix multiplication is performed, are naturally given in the outer link frame. Therefore, DCMs must be applied here as well. This results in the form of the I matrix for computation as shown in Equation (3.33).

$$I = \begin{bmatrix} I_1 & \boldsymbol{0} & \boldsymbol{0} \\ I_2\,{}^2C^1 & I_2\Gamma_1 & \boldsymbol{0} \\ I_3\,{}^3C^1 & I_3\,{}^3C^2\Gamma_1 & I_3\Gamma_2 \end{bmatrix} \tag{3.33}$$

A 9x3 matrix of zeros follows the I matrix, since the time rate of change of positional velocity has no impact on the attitude equations of motion.

The P matrix converts the gimbal constraint forces to torques. Each row of the matrix has a moment arm term for each gimbal it is connected to and the sign of the element ensures proper alignment of F_{G1} and F_{G2} relative to the link frame. Each element of the P matrix in Equation (3.34) is a 3x3 matrix, resulting in a total size of 9x6.

$$P = \begin{bmatrix} \tilde{r}_{11} & \boldsymbol{0} \\ -\tilde{r}_{21} & \tilde{r}_{22} \\ \boldsymbol{0} & -\tilde{r}_{32} \end{bmatrix} \tag{3.34}$$

The rows of the P matrix must also be put into the relative link body frame. While each $\tilde{r}$ term is already in the native frame, the gimbal constraint forces that these terms will be multiplied by are defined in the inertial frame and must therefore be converted.

$$P = \begin{bmatrix} \tilde{r}_{11}\,{}^{1}C^{N} & \boldsymbol{0} \\ -\tilde{r}_{21}\,{}^{2}C^{N} & \tilde{r}_{22}\,{}^{2}C^{N} \\ \boldsymbol{0} & -\tilde{r}_{32}\,{}^{3}C^{N} \end{bmatrix} \tag{3.35}$$

The Π matrix collects the cross product terms generated from the gimbal constraint described in Equations (2.22) and (2.23). Each row correlates to a link. The first column multiplies by $\dot{\omega}_1$ and accounts for the translational acceleration due to the rotation of link one relative to the inertial frame. The second and third columns account for the translational acceleration due to the gimbal rotations. Each element of the Π matrix in Equation (3.36) is a 3x3 matrix, resulting in a 9x9 Π matrix.

$$\Pi = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ m_2(\tilde{r}_{21} - \tilde{r}_{11}) & m_2(\tilde{r}_{21}\Gamma_1) & \boldsymbol{0} \\ m_3(\tilde{r}_{21} - \tilde{r}_{11} + \tilde{r}_{32} - \tilde{r}_{22}) & m_3(\tilde{r}_{21}\Gamma_1 + \tilde{r}_{32}\Gamma_1 - \tilde{r}_{22}\Gamma_1) & m_3\tilde{r}_{32}\Gamma_2 \end{bmatrix} \tag{3.36}$$

The Π matrix is best evaluated in the inertial frame, but this requires the use of DCMs to convert the various vector quantities to the correct frame. First, each $\tilde{r}_{ij}$ term must be converted to the inertial frame. Furthermore, the Γ_i terms, which become $\Gamma_i\ddot{\theta}_j$ when matrix multiplication is performed, are in the link two and three frames, and must also be converted to inertial. Lastly, the $\dot{\omega}_1$, $\ddot{\theta}_1$, and $\ddot{\theta}_2$ terms that will be multiplied through need to be transformed into the inertial frame. This results in the final Π matrix in Equation (3.37).

$$\Pi = \begin{bmatrix} 0 & 0 & 0 \\ m_2({}^{N}C^{2}\tilde{r}_{21} - {}^{N}C^{1}\tilde{r}_{11})\,{}^{N}C^{1} & m_2\,{}^{N}C^{2}(\tilde{r}_{21}\Gamma_1) & 0 \\ m_3({}^{N}C^{2}\tilde{r}_{21} - {}^{N}C^{1}\tilde{r}_{11} + {}^{N}C^{3}\tilde{r}_{32} - {}^{N}C^{2}\tilde{r}_{22}) & m_3\,{}^{N}C^{2}(\tilde{r}_{21}\Gamma_1 + ({}^{2}C^{3}\tilde{r}_{32})\Gamma_1 - \tilde{r}_{22}\Gamma_1) & m_3\,{}^{N}C^{3}\tilde{r}_{32}\Gamma_2 \end{bmatrix} \tag{3.37}$$

The μ matrix collects the mass terms.

$$\mu = \begin{bmatrix} \boldsymbol{m}_1 \\ \boldsymbol{m}_2 \\ \boldsymbol{m}_3 \end{bmatrix} \tag{3.38}$$

Each element of Equation (3.38) is a 3x3 diagonal matrix, where the mass of each link is copied across the diagonal. The off-diagonal terms are zero, as shown in Equation (3.39). This results in a 9x3 matrix.

$$\mu = \begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_1 & 0 \\ 0 & 0 & m_1 \\ m_2 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_2 \\ m_3 & 0 & 0 \\ 0 & m_3 & 0 \\ 0 & 0 & m_3 \end{bmatrix} \tag{3.39}$$

While the mass terms are agnostic to the frame, the acceleration of link one that will right multiply the mass terms will need to be transformed to the inertial frame, as shown in Equation (3.40).

$$\mu = \begin{bmatrix} \boldsymbol{m}_1 {}^N C^1 \\ \boldsymbol{m}_2 {}^N C^1 \\ \boldsymbol{m}_3 {}^N C^1 \end{bmatrix} \tag{3.40}$$

Lastly, the J matrix collects the gimbal constraint forces to be used in the translational equations of motion. Each element of Equation (3.41) is a 3x3 matrix, where the diagonal is populated by either zeros, or positive or negative unity, and the off-diagonal terms are zero, as shown in Equation (3.42). The sign of each element accounts for proper orientation of the gimbal constraint forces relative to each link. Since the constraint forces are defined in the inertial frame, no DCMs are necessary.

$$J = \begin{bmatrix} \boldsymbol{1} & \boldsymbol{0} \\ \boldsymbol{-1} & \boldsymbol{1} \\ \boldsymbol{0} & \boldsymbol{-1} \end{bmatrix} \tag{3.41}$$

$$J = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \tag{3.42}$$

Similarly, the constraint vector in Equation (3.7) can be partitioned into two subvectors, F and T. The constraints for the orientation equations of motion are collected in T, shown in Equation (3.43) where each row represents a 3x1 column vector, resulting in a 9x1 column vector.

$$T = \begin{Bmatrix} T_1 - T_{G1} - \tilde{\omega}_1 \left(I_1 \omega_1 \right) \\ T_2 + T_{G1} - T_{G2} - \tilde{\omega}_2 \left(I_2 \omega_2 \right) - I_2 (\dot{\Gamma}_1 \dot{\theta}_1) - I_2 \tilde{\omega}_1 \Gamma_1 \dot{\theta} \\ T_3 + T_{G2} - \tilde{\omega}_3 \left(I_3 \omega_3 \right) + I_3 (\dot{\Gamma}_1 \dot{\theta}_1) + I_3 (\dot{\Gamma}_2 \dot{\theta}_2) - I_3 (\tilde{\omega}_1 \Gamma_2 \dot{\theta}_2) - I_3 (\Gamma_1 \dot{\theta}_1 \times \Gamma_2 \dot{\theta}_2) \end{Bmatrix} \tag{3.43}$$

Similar to the Π or P submatrix of the dynamics matrices, each row of the T vector corresponds to a link, and the local link frame is utilized as the reference frame for each row. Therefore, DCMs are required to transform gimbal torques, (T_{G1}, T_{G2}), as well as some of the dynamics terms involved in gimbal rotation $(\dot{\Gamma}_1 \dot{\theta}_1)$.

$$T = \begin{Bmatrix} T_1 - {}^1C^2 T_{G1} - \tilde{\omega}_1 \left(I_1 \omega_1 \right) \\ T_2 + T_{G1} - {}^2C^3 T_{G2} - \tilde{\omega}_2 \left(I_2 \omega_2 \right) - I_2 (\dot{\Gamma}_1 \dot{\theta}_1) - I_2 \, {}^2C^1 \tilde{\omega}_1 \, {}^1C^2 \Gamma_1 \dot{\theta}_1 \\ T_3 + T_{G2} - \tilde{\omega}_3 \left(I_3 \omega_3 \right) + I_3 ({}^3C^2 \dot{\Gamma}_1 \dot{\theta}_1) + I_3 (\dot{\Gamma}_2 \dot{\theta}_2) - I_3 ({}^3C^1 \tilde{\omega}_1 \, {}^1C^3 \Gamma_2 \dot{\theta}_2) - I_3 ({}^3C^2 \Gamma_1 \dot{\theta}_1 \times \Gamma_2 \dot{\theta}_2) \end{Bmatrix} \tag{3.44}$$

The constraints for the translation equations of motion are collected in F, shown in Equation (3.45) where each row represents a 3x1 column vector, resulting in a 9x1 column vector.

$$F = \begin{Bmatrix} F_1 \\ -m_2\tilde{r}_{21}\dot{\Gamma}_1\dot{\theta}_1 - m_2\tilde{r}_{21}\tilde{\omega}_1\Gamma_1\dot{\theta} - m_2\bar{\omega}_1 r_{11} + m_2\bar{\omega}_2 r_{21} + F_2 \\ -m_3(\tilde{r}_{21}\dot{\Gamma}_1 - \tilde{r}_{22}\dot{\Gamma}_1 + \tilde{r}_{32}\dot{\Gamma}_1)\dot{\theta}_1 - m_3(\tilde{r}_{32}\tilde{\omega}_1 + \tilde{r}_{21}\tilde{\omega}_1 - \tilde{r}_{22}\tilde{\omega}_1)\Gamma_1\dot{\theta}_1 - m_3\tilde{r}_{32}\dot{\Gamma}_2\dot{\theta}_2 ... \\ ... - m_3\tilde{r}_{32}(\tilde{\omega}_1\Gamma_2\dot{\theta}_2) - m_3\tilde{r}_{32}(\Gamma_1\dot{\theta}_1 \times \Gamma_2\dot{\theta}_2) ... \\ ... - m_3(\bar{\omega}_1 r_{11} - \bar{\omega}_2 r_{21} + \bar{\omega}_2 r_{22} - \bar{\omega}_3 r_{32}) + F_3 \end{Bmatrix} \quad (3.45)$$

Similarly to the T subvector, each row of the F vector corresponds to a link. However, since the equations of motion for translation were placed in the inertial reference frame in the dynamics submatrices, F must also be given in the inertial frame. This results in the need for a variety of transformations since each term except the link forces are defined in local link frames.

$$F = \begin{Bmatrix} F_1 \\ -m_2\,{}^N C^2 \tilde{r}_{21}\dot{\Gamma}_1\dot{\theta}_1 - m_2\,{}^N C^2 \tilde{r}_{21}\,{}^2 C^1 \tilde{\omega}_1\,{}^1 C^2 \Gamma_1\dot{\theta}_1 - m_2\,{}^N C^1 \bar{\omega}_1 r_{11} + m_2\,{}^N C^2 \bar{\omega}_2 r_{21} + F_2 \\ -m_3\,{}^N C^2(\tilde{r}_{21}\dot{\Gamma}_1 - \tilde{r}_{22}\dot{\Gamma}_1 + {}^2 C^3 \tilde{r}_{32}\dot{\Gamma}_1)\dot{\theta}_1 - m_3({}^N C^3 \tilde{r}_{32}\,{}^3 C^1 \tilde{\omega}_1\,{}^1 C^2 + {}^N C^2 \tilde{r}_{21}\,{}^2 C^1 \tilde{\omega}_1\,{}^1 C^2 - {}^N C^2 \tilde{r}_{22}\,{}^2 C^1 \tilde{\omega}_1\,{}^1 C^2)\Gamma_1\dot{\theta}_1 ... \\ ... - m_3\,{}^N C^3 \tilde{r}_{32}\dot{\Gamma}_2\dot{\theta}_2 - m_3\,{}^N C^3 \tilde{r}_{32}({}^3 C^1 \tilde{\omega}_1\,{}^1 C^3 \Gamma_2\dot{\theta}_2) - m_3\,{}^N C^3 \tilde{r}_{32}({}^3 C^2 \Gamma_1\dot{\theta}_1 \times \Gamma_2\dot{\theta}_2) - m_3({}^N C^1 \bar{\omega}_1 r_{11} - {}^N C^2 \bar{\omega}_2 r_{21} ... \\ ... + {}^N C^2 \bar{\omega}_2 r_{22} - {}^N C^3 \bar{\omega}_3 r_{32}) + F_3 \end{Bmatrix} \quad (3.46)$$

Reassembling these submatrices and subvectors back into the mass matrix and constraint subvector results in the a form of Equation (3.7), that is relevant for numerical simulation:

$$\begin{bmatrix} I_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \tilde{r}_{11}{}^1C^N & \mathbf{0} \\ I_2{}^2C^1 & I_2\Gamma_1 & \mathbf{0} & \mathbf{0} & -\tilde{r}_{21}{}^2C^N & \tilde{r}_{22}{}^2C^N \\ I_3{}^3C^1 & I_3{}^3C^2\Gamma_1 & I_3\Gamma_2 & \mathbf{0} & \mathbf{0} & -\tilde{r}_{32}{}^3C^N \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{m}_1 & \mathbf{1} & \mathbf{0} \\ m_2({}^NC^2\tilde{r}_{21} - {}^NC^1\tilde{r}_{11}) & m_2{}^NC^2(\tilde{r}_{21}\Gamma_1) & \mathbf{0} & \boldsymbol{m}_2 & \mathbf{-1} & \mathbf{1} \\ m_3({}^NC^2\tilde{r}_{21} - {}^NC^1\tilde{r}_{11} + {}^NC^3\tilde{r}_{32} - {}^NC^2\tilde{r}_{22}) & m_3{}^NC^2(\tilde{r}_{21}\Gamma_1 + ({}^2C^3\tilde{r}_{32})\Gamma_1 - \tilde{r}_{22}\Gamma_1) & m_3{}^NC^3\tilde{r}_{32}\Gamma_2 & \boldsymbol{m}_3 & \mathbf{0} & \mathbf{-1} \end{bmatrix} \begin{Bmatrix} \dot{\omega}_1 \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \dot{v}_1 \\ F_{G1} \\ F_{G2} \end{Bmatrix} = \ldots$$

(3.47)

$$\ldots \begin{Bmatrix} T_1 - T_{G1} - \tilde{\omega}_1(I_1\omega_1) \\ T_2 + T_{G1} - {}^2C^3T_{G2} - \tilde{\omega}_2(I_2\omega_2) - I_2(\dot{\Gamma}_1\dot{\theta}_1) - I_2{}^2C^1\tilde{\omega}_1{}^1C^2\Gamma_1\dot{\theta}_1 \\ T_3 + T_{G2} - \tilde{\omega}_3(I_3\omega_3) + I_3({}^3C^2\dot{\Gamma}_1\dot{\theta}_1) + I_3(\dot{\Gamma}_2\dot{\theta}_2) - I_3({}^3C^1\tilde{\omega}_1{}^1C^3\Gamma_2\dot{\theta}_2) - I_3({}^3C^2\Gamma_1\dot{\theta}_1 \times \Gamma_2\dot{\theta}_2) \\ F_1 \\ -m_2{}^NC^2\tilde{r}_{21}\dot{\Gamma}_1\dot{\theta}_1 - m_2{}^NC^2\tilde{r}_{21}{}^2C^1\tilde{\omega}_1{}^1C^2\Gamma_1\dot{\theta}_1 - m_2{}^NC^1\overline{\omega}_1r_{11} + m_2{}^NC^2\overline{\omega}_2r_{21} + F_2 \\ -m_3{}^NC^2(\tilde{r}_{21}\dot{\Gamma}_1 - \tilde{r}_{22}\dot{\Gamma}_1 + {}^2C^3\tilde{r}_{32}\dot{\Gamma}_1)\dot{\theta}_1 - m_3{}^NC^2(\tilde{r}_{21}{}^2C^1\tilde{\omega}_1{}^1C^2 - \tilde{r}_{22}{}^2C^1\tilde{\omega}_1{}^1C^2)\Gamma_1\dot{\theta}_1 - m_3{}^NC^3\tilde{r}_{32}\dot{\Gamma}_2\dot{\theta}_2 + \ldots \\ \ldots - m_3{}^NC^3\tilde{r}_{32}({}^3C^1\tilde{\omega}_1{}^1C^3\Gamma_2\dot{\theta}_2) - m_3{}^NC^3\tilde{r}_{32}({}^3C^2\Gamma_1\dot{\theta}_1 \times \Gamma_2\dot{\theta}_2) - m_3({}^NC^1\overline{\omega}_1r_{11} - {}^NC^2\overline{\omega}_2r_{21} + {}^NC^2\overline{\omega}_2r_{22} - {}^NC^3\overline{\omega}_3r_{32}) + F_3 \end{Bmatrix}$$

D. ELIMINATING JOINT CONSTRAINT FORCES

While the submatrices and subvectors discussed are convenient for a greater understanding of the interactions between terms and are useful for building the system through inspection, further partitioning must be completed before they can be efficiently solved. In order to reduce the complexity of the system, it is desired to perform matrix operations to eliminate the joint constraint forces, as knowledge of these forces is not required to simulate the dynamics. They can, however, be reconstructed later if required.

To perform the reduction operation, the state vector, mass matrix, and constraint vector are re-partitioned. The state vector is split between the dynamics variables and the gimbal constraint forces. The dynamics variables are collected in the $\dot{x}$ column vector in Equation (3.48), where each term is a 3x1 column vector, resulting in a 12x1 column vector.

$$\dot{x} = \begin{Bmatrix} \dot{\omega}_1 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{v}_1 \end{Bmatrix} \tag{3.48}$$

The remaining gimbal constraint forces are collected in the F_G column vector in Equation (3.49), where each term is a 3x1 column vector, resulting in a 6x1 column vector.

$$F_G = \begin{Bmatrix} F_{G1} \\ F_{G2} \end{Bmatrix} \tag{3.49}$$

The mass matrix is next split into the A, R, S, and U submatrices. The A submatrix collects the first 12 rows of the first 12 columns of the mass matrix. Therefore, it collects the entire I submatrix and corresponding zero matrices, as well as the first three rows of the Π and μ submatrices. The A submatrix is shown in Equation (3.50), where each term is a 3x3 matrix, resulting in a 12x12 matrix.

$$A = \begin{bmatrix} I_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ I_2\,{}^2C^1 & I_2\Gamma_1 & \mathbf{0} & \mathbf{0} \\ I_3\,{}^3C^1 & I_3\,{}^3C^2\Gamma_1 & I_3\Gamma_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{m}_1 \end{bmatrix} \tag{3.50}$$

The R submatrix collects the remaining six rows of the first 12 columns of the mass matrix. Effectively, it collects the P matrix and first three rows of the J matrix, as shown in Equation (3.51), where each term is a 3x3 matrix, resulting in a 12x6 matrix.

$$R = \begin{bmatrix} \tilde{r}_{11}\,{}^1C^N & \mathbf{0} \\ -\tilde{r}_{21}\,{}^2C^N & \tilde{r}_{22}\,{}^2C^N \\ \mathbf{0} & -\tilde{r}_{32}\,{}^3C^N \\ \mathbf{1} & \mathbf{0} \end{bmatrix} \tag{3.51}$$

The S submatrix collects the remaining 6 rows of the first 12 columns of the mass matrix. It collects the rows of Π and μ not contained in the A matrix, as shown in Equation (3.52), where each element is a 3x3 matrix, resulting in a 6x12 matrix.

$$S = \begin{bmatrix} m_2({}^NC^2\tilde{r}_{21} - {}^NC^1\tilde{r}_{11}) & m_2\,{}^NC^2(\tilde{r}_{21}\Gamma_1) & \mathbf{0} & \boldsymbol{m}_2 \\ m_3({}^NC^2\tilde{r}_{21} - {}^NC^1\tilde{r}_{11} + {}^NC^3\tilde{r}_{32} - {}^NC^2\tilde{r}_{22}) & m_3\,{}^NC^2(\tilde{r}_{21}\Gamma_1 + ({}^2C^3\tilde{r}_{32})\Gamma_1 - \tilde{r}_{22}\Gamma_1) & m_3\,{}^NC^3\tilde{r}_{32}\Gamma_2 & \boldsymbol{m}_3 \end{bmatrix} \tag{3.52}$$

Lastly, the U matrix collects the remaining 6 rows of the remaining 6 columns of the mass matrix. It collects the J terms not included in the R matrix, as shown in Equation (3.53), where each element is a 3x3 matrix, resulting in a 6x6 matrix.

$$U = \begin{bmatrix} -\mathbf{1} & \mathbf{1} \\ \mathbf{0} & -\mathbf{1} \end{bmatrix} \tag{3.53}$$

Since the dynamics and state vectors were partitioned as described, the constraint vector must also be restructured into the T' and F' subvectors. The T' subvector collects the first 12 elements of the constraint vector, or the entire T vector and first three elements of the F vector, resulting in a 12x1 column vector, shown in Equation (3.54).

$$T' = \begin{Bmatrix} T_1 - {}^1C^2 T_{G1} - \tilde{\omega}_1 \left(I_1 \omega_1 \right) \\ T_2 + T_{G1} - {}^2C^3 T_{G2} - \tilde{\omega}_2 \left(I_2 \omega_2 \right) - I_2 (\dot{\Gamma}_1 \dot{\theta}_1) \\ T_3 + T_{G2} - \tilde{\omega}_3 \left(I_3 \omega_3 \right) + I_3 ({}^3C^2 \dot{\Gamma}_1 \dot{\theta}_1) + I_3 (\dot{\Gamma}_2 \dot{\theta}_2) \\ F_1 \end{Bmatrix} \tag{3.54}$$

The F' subvector collects the remaining constraint terms, or the remaining six elements of the F vector, resulting in a 6x1 column vector, shown in Equation (3.55).

$$F' = \begin{Bmatrix} -m_2 {}^N C^2 \tilde{r}_{21} \dot{\Gamma}_1 \dot{\theta}_1 - m_2 {}^N C^1 \bar{\omega}_1 r_{11} + m_2 {}^N C^2 \bar{\omega}_2 r_{21} + F_2 \\ -m_3 {}^N C^2 (\tilde{r}_{21} \dot{\Gamma}_1 - \tilde{r}_{22} \dot{\Gamma}_1 + ({}^2C^3 \tilde{r}_{32}) \dot{\Gamma}_1) \dot{\theta}_1 - m_3 {}^N C^3 \tilde{r}_{32} \dot{\Gamma}_2 \dot{\theta}_2 \ldots \\ \ldots - m_3 {}^N C^2 ({}^2C^1 \bar{\omega}_1 r_{11} - \bar{\omega}_2 r_{21} + \bar{\omega}_2 r_{22} - {}^2C^3 \bar{\omega}_3 r_{32}) + F_3 \end{Bmatrix} \tag{3.55}$$

When assembled, the *A*,*R*,*S* and *U* submatrices yield Equation (3.56), expanded for clarity in Equation (3.57).

$$\begin{bmatrix} A & R \\ S & U \end{bmatrix} \begin{Bmatrix} \dot{x} \\ F_G \end{Bmatrix} = \begin{Bmatrix} T' \\ F' \end{Bmatrix} \tag{3.56}$$

$$\left[\begin{array}{c|c} A & R \\ \hline S & U \end{array}\right]\left\{\begin{array}{c} \dot{x} \\ \hline F_G \end{array}\right\}=\left\{\begin{array}{c} T' \\ \hline F' \end{array}\right\}$$

$$\left[\begin{array}{cccc|cc} I_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \tilde{r}_{11}\,{}^1C^N & \mathbf{0} \\ I_2\,{}^2C^1 & I_2\Gamma_1 & \mathbf{0} & \mathbf{0} & -\tilde{r}_{21}\,{}^2C^N & \tilde{r}_{22}\,{}^2C^N \\ I_3\,{}^3C^1 & I_3\,{}^3C^2\Gamma_1 & I_3\Gamma_2 & \mathbf{0} & \mathbf{0} & -\tilde{r}_{32}\,{}^3C^N \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{m}_1 & \mathbf{1} & \mathbf{0} \\ \hline m_2({}^NC^2\tilde{r}_{21}-{}^NC^1\tilde{r}_{11}) & m_2\,{}^NC^2(\tilde{r}_{21}\Gamma_1) & \mathbf{0} & \boldsymbol{m}_2 & -\mathbf{1} & \mathbf{1} \\ m_3({}^NC^2\tilde{r}_{21}-{}^NC^1\tilde{r}_{11}+{}^NC^3\tilde{r}_{32}-{}^NC^2\tilde{r}_{22}) & m_3\,{}^NC^2(\tilde{r}_{21}\Gamma_1+({}^2C^3\tilde{r}_{32})\Gamma_1-\tilde{r}_{22}\Gamma_1) & m_3\,{}^NC^3\tilde{r}_{32}\Gamma_2 & \boldsymbol{m}_3 & \mathbf{0} & -\mathbf{1} \end{array}\right]\left\{\begin{array}{c} \dot{\omega}_1 \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \dot{v}_1 \\ \hline F_{G1} \\ F_{G2} \end{array}\right\}= \qquad (3.57)$$

$$\ldots\left\{\begin{array}{c} T_1-T_{G1}-\tilde{\omega}_1(I_1\omega_1) \\ T_2+T_{G1}-{}^2C^3T_{G2}-\tilde{\omega}_2(I_2\omega_2)-I_2(\dot{\Gamma}_1\dot{\theta}_1)-I_2\,{}^2C^1\tilde{\omega}_1\,{}^1C^2\Gamma_1\dot{\theta}_1 \\ T_3+T_{G2}-\tilde{\omega}_3(I_3\omega_3)+I_3({}^3C^2\dot{\Gamma}_1\dot{\theta}_1)+I_3(\dot{\Gamma}_2\dot{\theta}_2)-I_3({}^3C^1\tilde{\omega}_1\,{}^1C^3\Gamma_2\dot{\theta}_2)-I_3({}^3C^2\Gamma_1\dot{\theta}_1\times\Gamma_2\dot{\theta}_2) \\ \hline F_1 \\ -m_2\,{}^NC^2\tilde{r}_{21}\dot{\Gamma}_1\dot{\theta}_1-m_2\,{}^NC^2\tilde{r}_{21}\,{}^2C^1\tilde{\omega}_1\,{}^1C^2\Gamma_1\dot{\theta}_1-m_2\,{}^NC^1\bar{\omega}_1r_{11}+m_2\,{}^NC^2\bar{\omega}_2r_{21}+F_2 \\ -m_3\,{}^NC^2(\tilde{r}_{21}\dot{\Gamma}_1-\tilde{r}_{22}\dot{\Gamma}_1+{}^2C^3\tilde{r}_{32}\dot{\Gamma}_1)\dot{\theta}_1-m_3\,{}^NC^2(\tilde{r}_{21}\,{}^2C^1\tilde{\omega}_1\,{}^1C^2-\tilde{r}_{22}\,{}^2C^1\tilde{\omega}_1\,{}^1C^2)\Gamma_1\dot{\theta}_1-m_3\,{}^NC^3\tilde{r}_{32}\dot{\Gamma}_2\dot{\theta}_2+\ldots \\ \ldots-m_3\,{}^NC^3\tilde{r}_{32}({}^3C^1\tilde{\omega}_1\,{}^1C^3\Gamma_2\dot{\theta}_2)-m_3\,{}^NC^3\tilde{r}_{32}({}^3C^2\Gamma_1\dot{\theta}_1\times\Gamma_2\dot{\theta}_2)-m_3({}^NC^1\bar{\omega}_1r_{11}-{}^NC^2\bar{\omega}_2r_{21}+{}^NC^2\bar{\omega}_2r_{22}-{}^NC^3\bar{\omega}_3r_{32})+F_3 \end{array}\right\}$$

Multiplied out, Equation (3.56) gives:

$$A\dot{x} + RF_G = T' \tag{3.58}$$

$$S\dot{x} + UF_G = F' . \tag{3.59}$$

Solving Equation (3.59) for F_G yields:

$$F_G = U^{-1}(F' - S\dot{x}) . \tag{3.60}$$

Reinserting Equation (3.60) back into Equation (3.58) yields a form where F_G has been eliminated as an independent variable.

$$A\dot{x} + R(U^{-1}(F' - S\dot{x})) = T' \tag{3.61}$$

Consolidating terms yields:

$$(A - RU^{-1}S)\dot{x} = T' - RU^{-1}F' . \tag{3.62}$$

Lastly, Equation (3.62) can be solved for $\dot{x}$ to yield the values of the desired variables, $\dot{\omega}_1$, $\ddot{\theta}_1$, $\ddot{\theta}_2$, and $\dot{v}$ as shown in Equation (3.63).

$$\dot{x} = \begin{Bmatrix} \dot{\omega}_1 \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \dot{v}_1 \end{Bmatrix} = (A - RU^{-1}S)^{-1}(T' - RU^{-1}F') \tag{3.63}$$

Equation (3.63) is the final deliverable for the dynamic simulation model. For the general three-body model, the other state variables (ω_1, ω_2, ω_3, $\dot{q}$, q, $\dot{\theta}_1$, $\dot{\theta}_2$, θ_1, θ_2, v, d) can be solved for by augmenting the state vector and integrating Equation (3.63). A MATLAB script was developed and utilized to solve for these values, as described in Chapter IV.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. IMPLEMENTING THE SIMULATION IN MATLAB

A. GENERIC MATLAB CODE

In order to perform simulations of the system, the equations from Section 2.3 were implemented in a general MATLAB code. The code was written for any three-body gimbaled system as previously configured, and as written in such a way as to facilitate easy tailoring to various implementations of the described system, (i.e., different gimbal or body constraints).

The code, provided in Appendix A, uses a symbolic architecture that utilizes numerical data for constants or knowns such as mass, lengths, inertias, etc., and uses placeholder symbols for unknowns or variables expected to change such as gimbal angles and rates. The equations and matrices are then solved symbolically to yield a final $\dot{x}$ in terms of the symbolic variables. However, this vector only includes the state variables, $\dot{x} = \{\dot{\omega}_1, \ddot{\theta}_1, \ddot{\theta}_2, \dot{v}_1\}^T$. In order to step the simulation forward in time, the state vector must be expanded to describe the entire set of state variables for the system under consideration. Furthermore, q is reestablished as the column vector of four quaternions, and d is introduced as a 3x1 column vector to describe the translation of link one.

With these variables established, a new state vector, x', is defined in Equation (4.1).

$$x' = \begin{Bmatrix} \omega_1 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \\ v_1 \\ \theta_1 \\ \theta_2 \\ d_1 \\ q \end{Bmatrix} \tag{4.1}$$

Taking the derivative of x' yields Equation (4.2).

$$\dot{x}' = \begin{Bmatrix} \dot{\omega}_1 \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \dot{v}_1 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{d}_1 \\ \dot{q} \end{Bmatrix} \tag{4.2}$$

The augmented state vector is used in the MATLAB function ODE45 along with initial values for each state and a start and end time. The MATLAB ODE45 function integrates the system over the specified timeframe and returns a time history of the states in x'.

It was found early in the development of the MATLAB Code that the symbolic implementation is an inefficient approach for solving the matrix equations. The first MATLAB code was written to provide equations of motion in purely symbolic terms, even those where numerical values can be expected as knowns (i.e., mass, lengths, inertias, etc.). However, it was found that with such large matrices and with so many symbolic variables, MATLAB was unable to compute the symbolic solution on the available hardware. The number of symbolic variables was reduced by utilizing values for known quantities and relegating symbolic use to only unknown variables. Further order reduction of the system was made by implementing gimbal or link constraints. This effectively reduces the system dimensionality to less than 3N+3, driving the system to become solvable and far more efficient. However, if a large order system is required, one with full range of motion in each gimbal or with a large number of links, for example, a numerical approach is recommended due to the complexities of the symbolic computation.

This more efficient numerical approach was utilized by LCDR Rich Gargano [19]. Whereas the method described here solves for the time rate of change of the state variable as an equation, then passes these equations to ODE45 to be solved, Gargano solves the matrix equations within ODE45 numerically. Gargano utilizes a

dynamics function within ODE45 to complete the matrix calculations. Initial values are passed to the dynamics equation, which populates the matrices with numerical values, thereby avoiding the use of symbolic variables. The dynamics equations are then solved numerically. MATLAB is able to solve the numerical matrices far more efficiently than the symbolic matrices, resulting in orders of magnitude improvement in computational efficiency.

B. SYSTEM TAILORING

In order to troubleshoot the MATLAB code during development, and in order to complete validation and verification of the code and the derived Newton-Euler equations, the general system was tailored to describe a simple two-link pendulum.

One of the benefits of the approach described in [15] is the ease of tailoring the general equations to a specified system. First, by manipulating the structure of the I, Π, μ and particularly the P and J matrices, the layout of the system can be easily altered. For example, body three could be moved to connect to body one instead of body two. This approach is covered in [15] for both the spherical joint and gimbaled procedures and should be implemented when building the submatrices. The other way to tailor the system is to constrain body motion or gimbal motion. For example, body one could be constrained in one or more axes for translation or attitude, or certain gimbal axes could be constrained to yield a double or single axis gimbal, as opposed to the three axes gimbal utilized in the general model. Constraints could be implemented when building the submatrices but it was found to be easier to consider the full system, then apply constraints to the $A, R, S,$ and U matrices. Since the basic layout of the general case is applicable to the desired test cases, delaying constraint application as late as possible allowed for the easiest tailoring to each specific system, while minimizing changes to the default code, and thus minimizing the possibility of introducing errors. However, the earlier these constraints are implemented, the more efficient the code is since operations are not being performed on superfluous variables. Nonetheless, the operations that would be eliminated by implementing constraints early are simple operations that MATLAB can perform quickly. The largest savings are achieved by constraining the system prior to

taking the inverse of any matrices. Waiting to constrain the system until the $A, R, S,$ and U matrices are constructed minimizes the efficiency losses, while maintaining an easy to tailor system.

To tailor the system using constraints, the axes of movement for body one and the axes of movement for each gimbal must be investigated. For example, if the motion of joint one is constrained in the x-axis to create a two-axes gimbaled joint in the y and z axes, then the state variables θ_{1x} is constant and $\dot{\theta}_{1x}$ and $\ddot{\theta}_{1x}$ are both zero by definition.

To begin the tailoring process, the impact the constrained variable would have on other equations of motion if unconstrained must be eliminated. This is completed by replacing the angle or velocity symbol for the constrained axis with a static numerical value or zero, respectively, then proceeding with the simulation as normal. For example, if body one is considered inertial, body rates for ω_1 are set to zero, resulting in zeros for $\tilde{\omega}_1$ and $\bar{\omega}_1$. Therefore, any terms multiplied by these matrices will be set to zero, thus eliminating the impact ω_1 would have on the motion of the other links. Then the quaternions vector for body one, $q,$ would be set to some numerical value to indicate a constant orientation of body one, and $\dot{q}$ is set to a constant {0,0,0,0} to negate rotation.

Next, the system of equations must be reduced so that the zeroed variables are not solved. Continuing with the constraint example, and starting with Equation (3.56), it can be seen that any column multiplied by $\ddot{\theta}_{1x}$ will be zero since motion is constrained in that axis. Since $\ddot{\theta}_{1x}$ is the fourth element in $\dot{x}$, it can be eliminated along with the fourth column of the A and S submatrices. However, this creates a problem in the size of the system of equations.

With the constrained columns eliminated, the system becomes overdetermined since the row corresponding to the eliminated constraint acts as a redundant constraint equation for the gimbal torque. In order to re-create a square syste, an equivalent number of rows must be eliminated in order to drive the number of equations equal to the number

of unknowns. However, an additional procedure must be implemented first in order to retain the joint constraint terms that will be eliminated in those rows.

For unconstrained joint axes, the torque is treated as a control input and provided by the user. However, for constrained axes, an internal resultant torque is generated in the joint to counteract any torques applied to the outer link. This can be easily implemented in the equations by replacing the total joint torque terms $(\mathrm{T}_{\mathrm{G1}}, \mathrm{T}_{\mathrm{G2}})$ with a combination of joint control torques $(\mathrm{T}_{\mathrm{G1C}}, T_{G2C})$ and joint reaction torques $(\mathrm{T}_{\mathrm{G1R}}, T_{G2R})$. Each axis of the total joint torque term will have either a control torque, or a reaction torque, but cannot have both. It is important to note that in [15], joint torque also includes damping terms. However, for the scope of this thesis, damping was assumed minimal and not modeled.

For example, consider a free floating link one, free to move in all six degrees of freedom (DOF), attached to a link two by a two axes gimbal, free to move in the x and z axes, but constrained in the y-axis. In practice, a torque would be applied in the x and z axes to drive link motion. Therefore, the torque in the x and z axes would be comprised of a control torque. However, the internal torque in the y-axis is driven by the y-axis torques exerted by the second link. This would generate a reaction torque for the y-axis of the joint. Suppose an external torque is applied to link two in the y-axis. This torque would not drive motion in the gimbal, as it is constrained and cannot be moved in the y-axis. However, the external torque does generate a resultant torque at the gimbaled joint that translates from link two, through the joint, to link one, and will drive motion in link one. This reaction torque would be the combination of the external torque, and any torques generated by the motion of link two. In this case, the total torque in gimbal one would be as follows:

$$TG1 = \begin{Bmatrix} T_{G1Cx} \\ T_{G1Ry} \\ T_{G1Cz} \end{Bmatrix} \tag{4.3}$$

In order to implement propagation of torques from outer links to inner links via resultant torques, the resultant torque must first be solved. In the case of joint one, starting with Equation (3.5) and exploring only the y-axis terms yields Equation (4.4).

$$\begin{aligned}&\left[I_2\dot{\omega}_1\right]_y+\left[I_2\Gamma_1\ddot{\theta}_1\right]_y-\left[\tilde{r}_{21}F_{G1}\right]_y+\left[\tilde{r}_{22}F_{G2}\right]_y=\ldots\\&\ldots \mathrm{T}_{2y}+\mathrm{T}_{G1Ry}-\mathrm{T}_{G2y}-\left[\tilde{\omega}_2\left(I_2\omega_2\right)\right]_y-\left[I_2\dot{\Gamma}_1\dot{\theta}_1\right]_y-\left[I_2\tilde{\omega}_1\Gamma_1\dot{\theta}\right]_y\end{aligned} \tag{4.4}$$

This is then solved for T_{G1Ry}.

$$\begin{aligned}&\mathrm{T}_{G1Ry}=\left[I_2\dot{\omega}_1\right]_y+\left[I_2\Gamma_1\ddot{\theta}_1\right]_y-\left[\tilde{r}_{21}F_{G1}\right]_y+\left[\tilde{r}_{22}F_{G2}\right]_y-\mathrm{T}_{2y}+\mathrm{T}_{G2y}\ldots\\&\ldots+\left[\tilde{\omega}_2\left(I_2\omega_2\right)\right]_y+\left[I_2\dot{\Gamma}_1\dot{\theta}_1\right]_y-\left[I_2\tilde{\omega}_1\Gamma_1\dot{\theta}\right]_y\end{aligned} \tag{4.5}$$

The result, given in Equation (4.5), can then be inserted in to the y component of Equation (3.4), as shown in Equation (4.6)

$$\begin{aligned}&\left[I_1\dot{\omega}_1\right]+\left[\tilde{r}_{11}F_{G1}\right]=T_1-(\left[I_2\dot{\omega}_1\right]_y+\left[I_2\Gamma_1\ddot{\theta}_1\right]_y-\left[\tilde{r}_{21}F_{G1}\right]_y+\left[\tilde{r}_{22}F_{G2}\right]_y-\mathrm{T}_{2y}+\mathrm{T}_{G2y}\ldots\\&\ldots+\left[\tilde{\omega}_2\left(I_2\omega_2\right)\right]_y+\left[I_2\dot{\Gamma}_1\dot{\theta}_1\right]_y-\left[I_2\tilde{\omega}_1\Gamma_1\dot{\theta}\right]_y)-\left[\tilde{\omega}_1\left(I_1\omega_1\right)\right]\end{aligned} \tag{4.6}$$

$$\begin{aligned}&\left[(I_1+I_{2y})\dot{\omega}_1\right]+\left[I_2\Gamma_1\ddot{\theta}_1\right]_y+\left[(\tilde{r}_{11}-\tilde{r}_{21}\,\mathrm{y})F_{G1}\right]+\left[\tilde{r}_{22}F_{G2}\right]_y=\ldots\\&\ldots T_1+\mathrm{T}_{2y}-\mathrm{T}_{G2y}-\left[\tilde{\omega}_2\left(I_2\omega_2\right)\right]_y-\left[I_2\dot{\Gamma}_1\dot{\theta}_1\right]_y-\left[\tilde{\omega}_1\left(I_1\omega_1\right)\right]-\left[I_2\tilde{\omega}_1\Gamma_1\dot{\theta}\right]_y\end{aligned} \tag{4.7}$$

In practice, the process of reducing the system order can be implemented anywhere from deriving the equations of motion to equation to the construction of the *A R* and *T'* matrices [15]. It can be noted that the process of solving for a resultant torque and applying it to the inner body is effectively accomplished by adding the constrained equation of motion of the outer to the inner body equations of motion. Propagating the terms is as simple as converting the constrained row of the outer body to the inner body frame using a DCM and adding the result to the corresponding inner body. This is illustrated in Equation (4.8), where the subscript i denotes the three rows of the inner body and the subscript o denotes the constrained outer body row. It is important to note that a row populated by zero must be added to the constrained outer body row for each axis that is not constrained (i.e., if an axis is controlled instead of constrained, the row is replaced with zeroes) in order to give it the proper dimensions for the DCM.

$$(A-RU^{-1}S)_i+{}^i\mathrm{C}^o(A-RU^{-1}S)_o=(T'-RU^{-1}F')_i+{}^i\mathrm{C}^o(T'-RU^{-1}F')_o \tag{4.8}$$

For example, start with the general system previously described and implement constraints in Equation (3.62). Suppose gimbal two is constrained in the x-axis, meaning

θ_{1x}, $\dot{\theta}_{1x}$, and $\ddot{\theta}_{1x}$ are all zero by definition. Therefore, column four of the $(A - RU^{-1}S)$ matrix, which corresponds to $\ddot{\theta}_{1x}$ can be eliminated since it multiplies by zero. Then, row four, which corresponds to the motion of link two in the x-axis, which is now a torque constraint, can be added to row one through three, which corresponds to the rotation of link one. While the constrained row in the outer body may only be a single row, once the necessary DCM is applied to convert it to the inner body frame, the terms may apply to all three rows of the inner body.

$$(A - RU^{-1}S)_{1-3} + {}^{1}\mathrm{C}^{2}(A - RU^{-1}S)_{4} = (T' - RU^{-1}F')_{1-3} + {}^{1}\mathrm{C}^{2}(T' - RU^{-1}F')_{4} \quad (4.9)$$

Furthermore, if more than one outer link is constrained in the same axis, both constrained rows will need to be propagated inward, starting with the outermost link. For example, start with the general case and assume gimbals one and two are both constrained in the x-axis. The seventh row (x-axis rotation of link three) is left multiplied by a DCM to convert it to the body two frame, then added to rows four through six. Then row four, which now also includes terms from the link three equation of motion, is left multiplied by a DCM to convert it the body one frame, and added to rows one through three.

$$(A - RU^{-1}S)_{4-6} + {}^{2}\mathrm{C}^{3}(A - RU^{-1}S)_{7} = (T' - RU^{-1}F')_{4-6} + {}^{2}\mathrm{C}^{3}(T' - RU^{-1}F')_{7} \quad (4.10)$$

$$\begin{aligned}&(A - RU^{-1}S)_{1-3} + {}^{1}\mathrm{C}^{2}((A - RU^{-1}S)_{4} + {}^{2}\mathrm{C}^{3}(A - RU^{-1}S)_{7}) = \ldots \\ &\ldots(T' - RU^{-1}F')_{1-3} + {}^{1}\mathrm{C}^{2}((T' - RU^{-1}F')_{4} + {}^{2}\mathrm{C}^{3}(T' - RU^{-1}F')_{7})\end{aligned} \quad (4.11)$$

Once this addition is complete and the terms are propagated to the innermost free link, the redundant outer row can then be eliminated, resulting in a properly sized system of equations.

This process is repeated for any constrained gimbal axes, by eliminating columns and combining and eliminating rows as necessary to eliminate gimbal motion and translation or rotation of link one.

The elimination of translation or rotation of link one is more straightforward. As described, the columns that are multiplied by $\dot{\omega}_1$ and $\dot{v}_1$ are eliminated as usual.

However, since link one is the innermost body, the additional process described for constrained gimbals does not apply. Link one is instead modeled as an immovable point in space. The joint force, F_{G1} is solved in such a way as to lock the joint to inertial space. Another way to view this setup is to assume the external torques and forces on link one are dynamic and always have the proper orientation and magnitude to keep the link stationary. Therefore, the rows that correspond to the eliminated degrees of freedom for body one can simply be eliminated.

V. VALIDATION AND VERIFICATION

A. SIMMECHANICS AS A VALIDATION AND VERIFICATION TOOL

In order to validate the accuracy of the MATLAB model developed in Chapter IV, independent tools were used to check the dynamics equations and overall system response. The SymPy program, which utilizes Python scripts and system configuration data and inputs, was utilized to generate equations of motion for each model. The equations were checked against symbolic outputs from MATLAB in order to verify no dynamics components were missing. This was particularly vital to identifying and correcting errors in that specific terms could be identified and corrected. The other tool used for this was the SimMechanics software. SimMechanics is a MATLAB environment that allows for easy construction and simulation of multibody systems. The program uses a block-diagram approach which allows the user to add bodies, joints, and forces, and torques in any desired configuration. The program then evaluates the system dynamics, solves the equations of motion, and propagates the dynamics for a specified period. Lastly, the program generates an animation of the system in order to visualize the results.

SimMechanics was utilized because it is simple to learn and operate, flexible enough to facilitate construction of a wide variety of system configurations, and operates natively with MATLAB, providing for easy output and correlation of results with the developed MATLAB model. One drawback to the SimMechanics suite is that it solves for the dynamics directly and does not provide the equations of motion as an output. This shortcoming would make it difficult to use the SimMechanics results directly for optimization.

The model is generated in the standard Simulink workspace, as shown in Figure 12. Each of the primary blocks represents either the mass properties of a link, a joint, or a translation to properly configure the system.

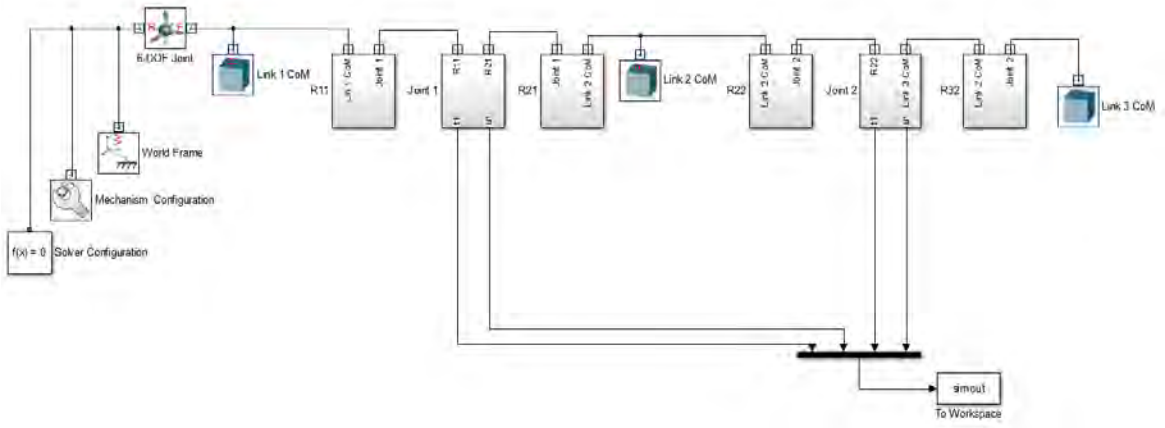


Figure 12. General SimMechanics Model for Three Link System

Mass properties, displayed in Figure 12 as the “Link 1 CoM,” “Link 2 CoM,” and “Link 3 CoM,” specify the mass of each link as well as the moments of inertia. These blocks also specify the shapes and dimensions of the links to be used in generating the animation. While these shapes and dimensions in themselves are not utilized to solve for the dynamics, they can be utilized to automatically generate appropriate inertia tensors. This option was not utilized for this work and custom inertia tensors were manually inserted in order to ensure properties matched those used in the MATLAB simulations.

Each joint, labeled “Joint 1” and “Joint 2” in Figure 12, is specified by a subsystem of blocks. This subsystem contains three individual blocks. The main block, labeled “Joint 1” or “Joint 2,” represents a revolute joint about a single axis. While every system explored in this thesis consists of single axis joints, multiple blocks could be used to construct a dual or triple axis joint. One property of the revolute joint is that it always rotates about the z-axis. Therefore, if the joint is desired to rotate in the x or y axes, the joint must first be rotated accordingly. Two blocks in the subsystem, labeled “Align Z for Joint” and “Unalign Z for Joint” perform the required 90 degree rotations about the x or y axes in order to align the z axis of the joint to the desired axis of the system, as seen in Figure 13. If a z-axis joint is desired, these blocks become dummy blocks and perform no function.

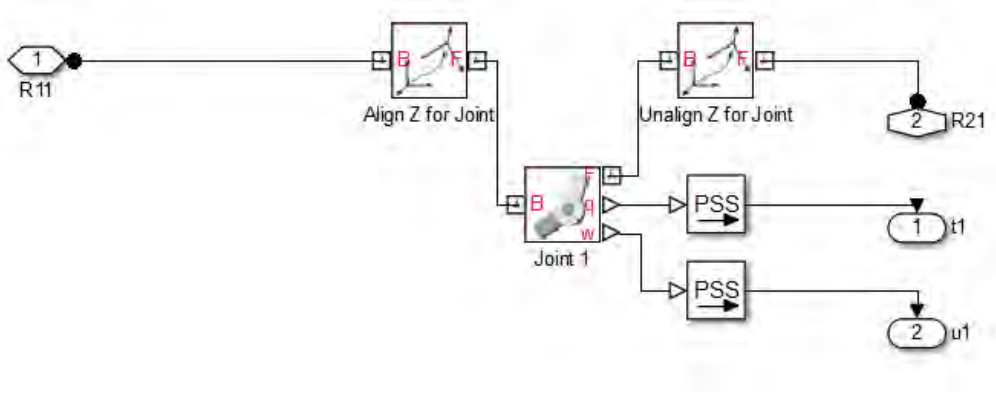


Figure 13. SimMechanics Joint Subsystem

The translations between joints and link centers of mass are performed by the subsystems labeled "R11," "R21," "R22," and "R32." Each of these subsystems contain translation blocks for the x, y, and z components of the moment arms $\left(r_{11}, r_{21}, r_{22}, r_{32}\right)$, as shown in Figure 14.

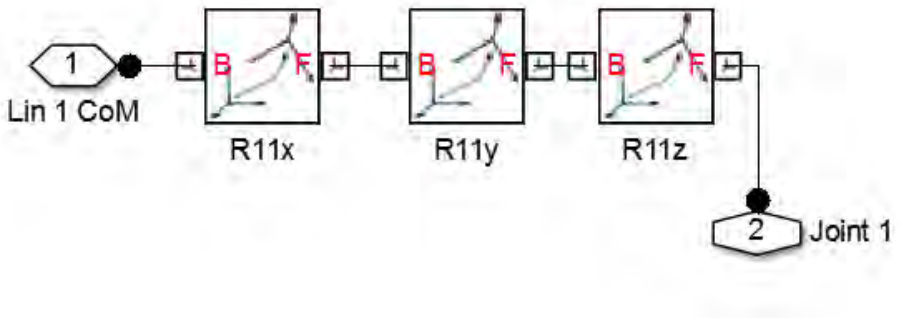


Figure 14. SimMechanics Translation Subsystem

Other blocks in the primary workspace include the Solver Configuration block, which defines solver settings for the simulation; the Mechanism Configuration block, which establishes the global gravity acceleration; the World Frame, which establishes an inertial reference frame; and a 6-DOF Joint, which allows for translation and rotation of link one relative to the inertial reference frame (this block is removed for a system with link one fixed).

B. TWO-LINK PENDULUM MODEL

The first test case utilized to debug the developed dynamic model and to perform validation and verification on was a simple two-link pendulum system. In order to tailor the general model to a simple two-link pendulum, the desired two-link pendulum must first be described. Link one is fixed for motion in both translation and attitude. This effectively removes the link from the system, resulting in gimbal one being fixed to an inertial point in space. Link two and three are modeled as rods connected by single axis gimbals, rotating in body local z-axes. Gravity is established along the inertial y -xis as $-9.81 \, m/s^2$. Arbitrary masses, lengths and inertias are provided for each link. This results in the system shown in Figure 15.

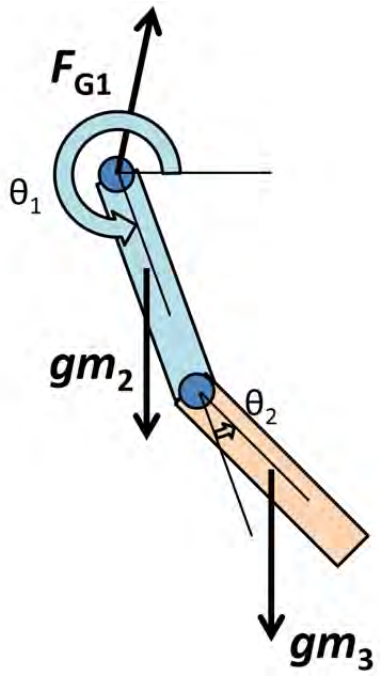


Figure 15. Two-link Pendulum

First, the masses, lengths, and inertias are provided or calculated for each link. The parameters for link one can be left as symbols since these values will be eliminated once constraints are applied. For the two-link pendulum system, masses of 10 kg were assumed for link two and three. Each link was assumed to have a length, l, of one meter

and width, r, of 10 centimeters, with mass evenly distributed so the center of mass resides at the volumetric center of the link. Inertias were assumed as follows:

$$I_2 = I_3 = \begin{bmatrix} \frac{1}{2}mr^2 & 0 & 0 \\ 0 & \frac{1}{12}ml_2 & 0 \\ 0 & 0 & \frac{1}{12}ml_2 \end{bmatrix} = \begin{bmatrix} 0.050 & 0 & 0 \\ 0 & 0.833 & 0 \\ 0 & 0 & 0.833 \end{bmatrix} kg \cdot m^2 \ . \qquad (5.1)$$

Constraints were applied in the MATLAB model starting with constraining link one to the inertial frame as described. First, the orientation of link one was aligned to the inertial frame by setting q = [0,0,0,1]$^{\mathrm{T}}$. Next, the rotation of link one was zeroed by setting $\dot{q}$ = [0,0,0,0]$^{\mathrm{T}}$. This negated the impact any rotation of link one would have on the subsequent links. Next, rows one through three of the $A, R,$ and T' submatrices, as well as columns one through three of the A and S submatrices were eliminated to constrain attitude motion of body one. Rows 10 through 12 of the $A, R,$ and T' submatrices, as well as columns 10 through 12 of the A and S submatrices were also eliminated to constrain translational motion of body one.

Joint constraints were implemented by setting $\theta_{1x}, \theta_{1y}, \dot{\theta}_{1x}, \dot{\theta}_{1y}, \theta_{2x}, \theta_{2y}, \dot{\theta}_{2x},$ and $\dot{\theta}_{2y}$ to zero, while leaving $\theta_{1z}, \dot{\theta}_{1z}, \theta_{2z},$ and $\dot{\theta}_{2z}$ as symbolic variables. Lastly, columns four, five, seven and eight of the A and S submatrices were eliminated. Next rows seven and eight (corresponding to the constrained axes for link three) were converted to the link two frame and added to rows four through six (corresponding to link two), as described in Chapter IV. Then rows four and five were converted to the link one frame and added to rows three through four. Note that in this case the constrained gimbal torques that get propagated through are eventually eliminated altogether. This is specific to a case where none of the links can move in the constrained degree of freedom. The result is a two dimension system to be solved using the Newton-Euler approach, where $\dot{x}$ was simplified from to only two variables.

$$\dot{x} = \begin{Bmatrix} \ddot{\theta}_{1z} \\ \ddot{\theta}_{2z} \end{Bmatrix} \qquad (5.2)$$

This system was expanded to the full state dynamics model:

$$\dot{x}' = \begin{Bmatrix} \ddot{\theta}_{1z} \\ \ddot{\theta}_{2z} \\ \dot{\theta}_{1z} \\ \dot{\theta}_{2z} \end{Bmatrix} \tag{5.3}$$

$$x' = \begin{Bmatrix} \dot{\theta}_{1z} \\ \dot{\theta}_{2z} \\ \theta_{1z} \\ \theta_{2z} \end{Bmatrix} . \tag{5.4}$$

The full state model was then solved for a 10 second simulation, where $\dot{\theta}_1, \dot{\theta}_{2,}$ and θ_2 were initialized as zero, and θ_1 was initialized at 300°, as depicted in Figure 16.

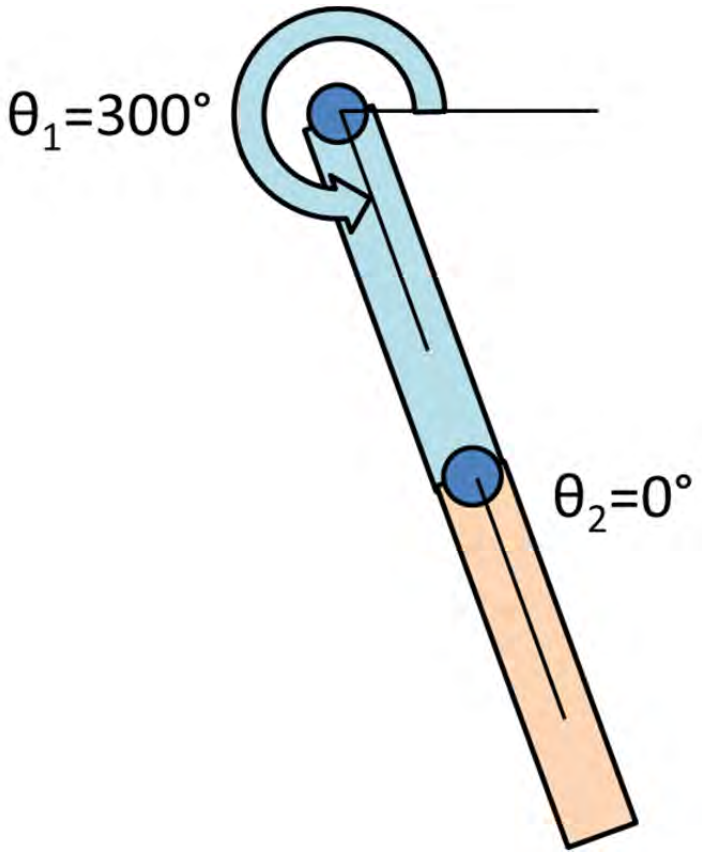


Figure 16. Double Pendulum Initial Configuration

The resulting gimbal angles and rates were plotted, as shown in Figure 17.

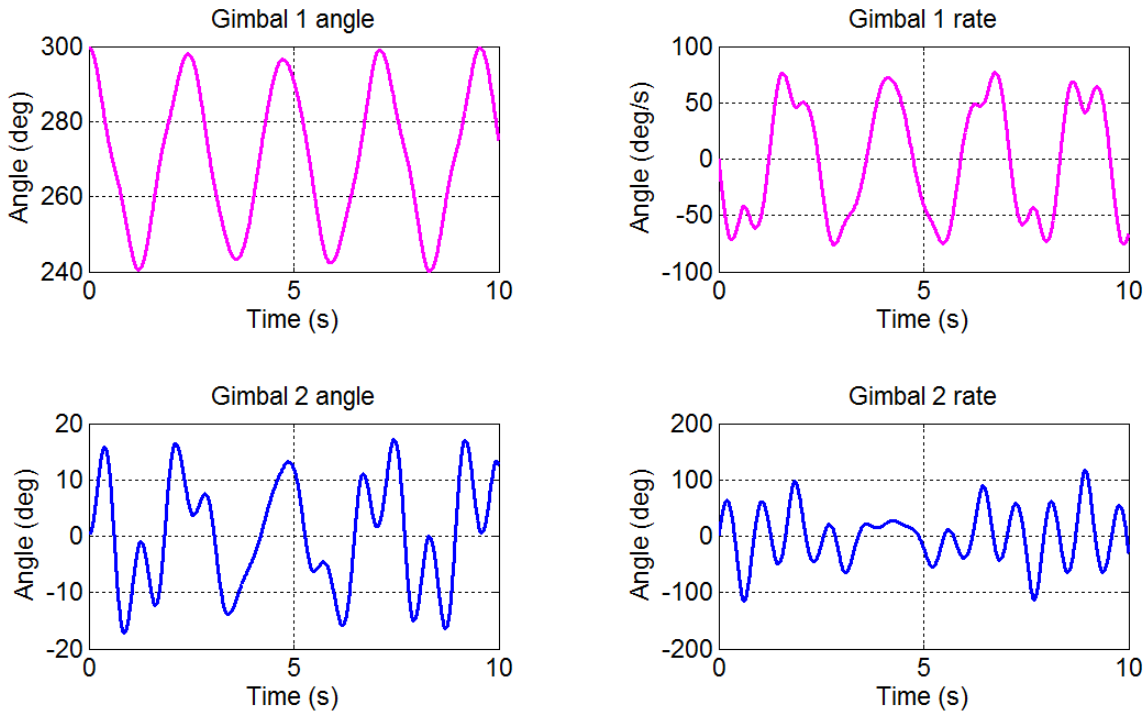


Figure 17. Double Pendulum Results from Newton-Euler Simulation

The full code for the MATLAB model of the double pendulum can be found in Appendix C.

Once the response was obtained from the MATLAB model, it needed to be verified. The general SimMechanics system was tailored to the double pendulum model. This model can be found in Appendix D. The simulation was then performed for a 10 second interval to match the simulation that was run for the MATLAB model. The results of this simulation can be seen in Figure 18.

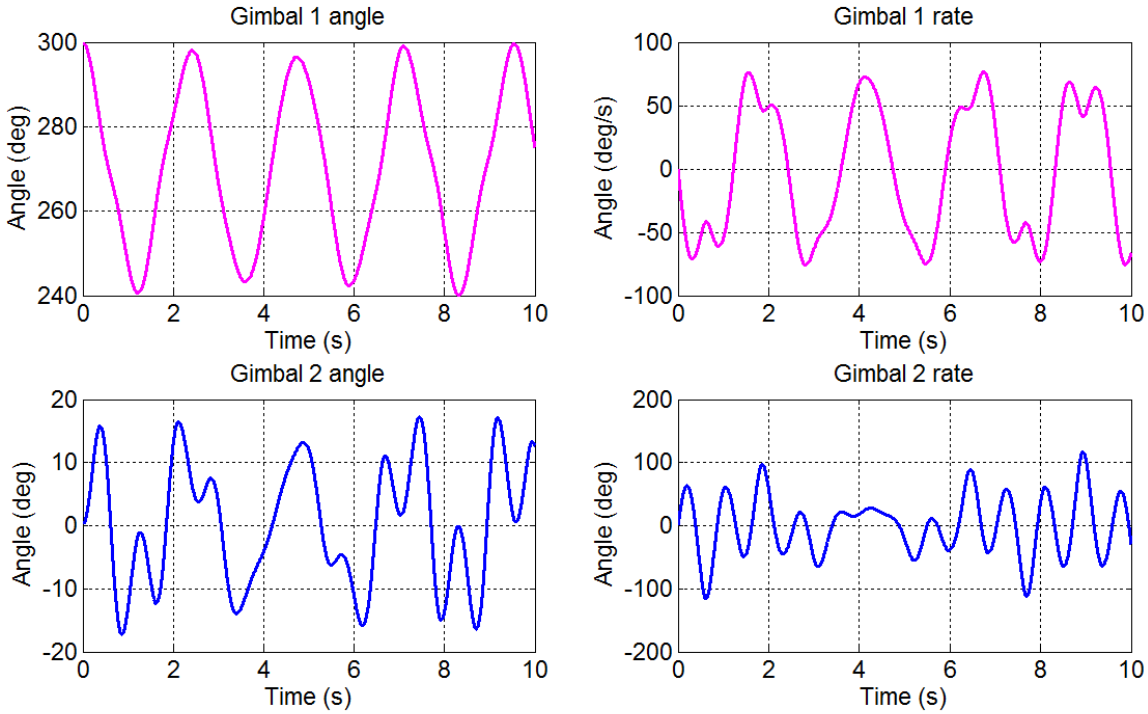


Figure 18. Double Pendulum Results from SimMechanics

Comparing the SimMechanics results to those of the developed simulation code, it is observed that the two results are identical. Absolute differences for gimbaled joint angles and rates can be seen in Figure 19. For the double pendulum case, the differences are seen to be negligible. Hence, the two models are considered to be in agreement for the tested double pendulum system.

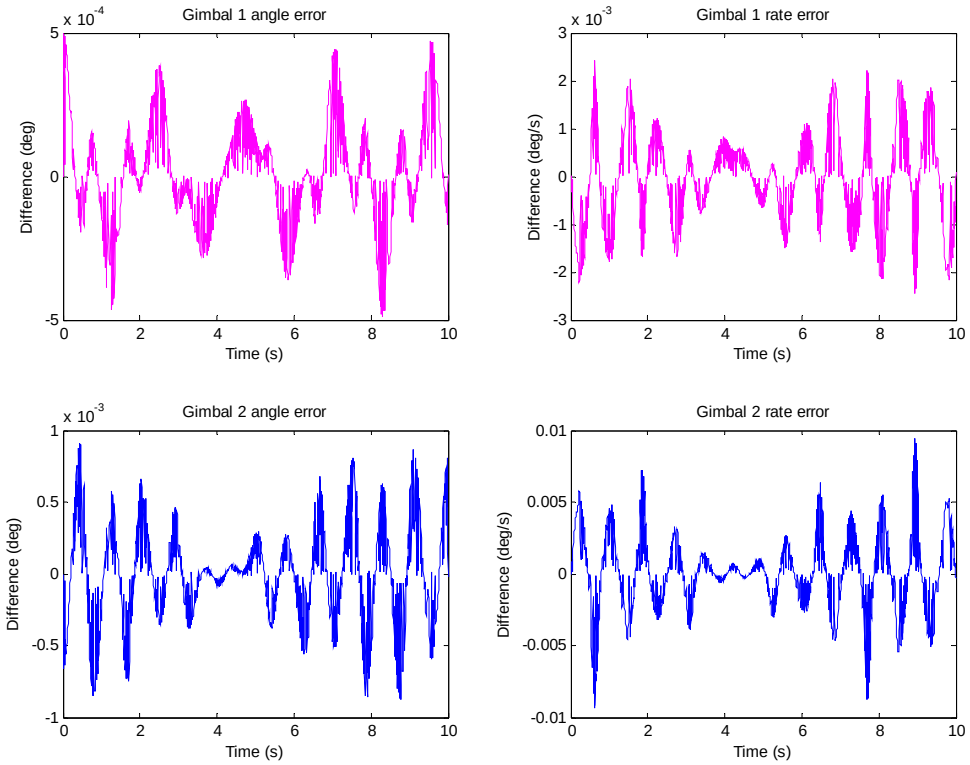


Figure 19. Double Pendulum Simulation Residuals

Once the simulations for the two-link pendulum confirmed the viability of the process, further tests were conducted on a more relevant system configuration.

C. AZIMUTH-ELEVATION SYSTEM.

A more applicable system configuration to approximate a gimbaled antenna is an azimuth-elevation setup. In order to maintain a stepwise evolution of the system, the double pendulum configuration was adapted to the azimuth-elevation setup. Properties such as link dimensions, link mass, and link inertia were transferred directly from the previous double pendulum model. The system was easily converted to an azimuth-elevation system by changing the first gimbaled joint from a z-axis joint, to a y-axis joint. This created an azimuth gimbal. The second joint was maintained as a z-axis joint in order to establish an elevation joint. Lastly, an inertial base was established, effectively creating a two link system where joint one is fixed in inertial space, as seen in Figure 20.

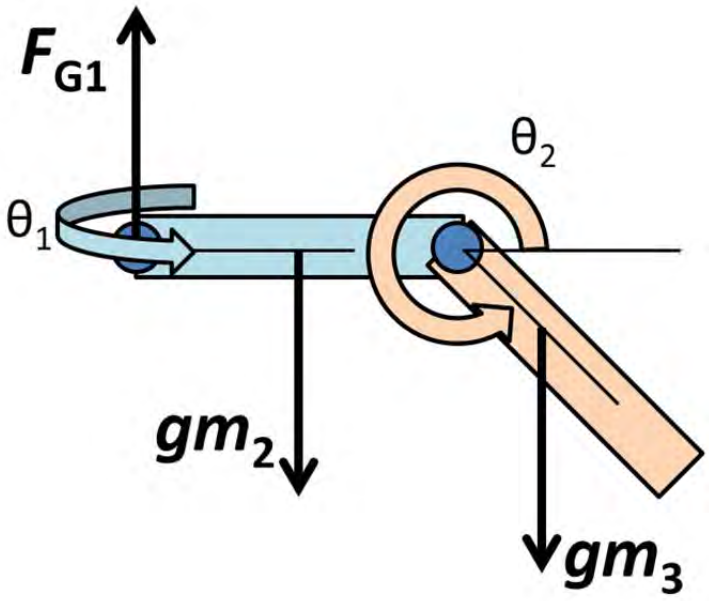


Figure 20. Azimuth-Elevation Configuration

Since the MATLAB code was written in a general form, only small changes to the double pendulum code were required to implement the az-el system. No changes were necessary to link one, so the body remained constrained in both translation and orientation.

Joint constraints were implemented by setting $\theta_{1x}, \theta_{1z}, \dot{\theta}_{1x}, \dot{\theta}_{1z}, \theta_{2x}, \theta_{2y}, \dot{\theta}_{2x},$ and $\dot{\theta}_{2y}$ to zero, while leaving $\theta_{1y}, \dot{\theta}_{1y}, \theta_{2z},$ and $\dot{\theta}_{2z}$ as symbolic variables. Column elimination proceeded in the same fashion as the double pendulum, but instead of eliminating column five, column six was eliminated, changing the constraint on joint one from the y-axis to the z-axis. This was accomplished by simply changing a few indices in the MATLAB code. Similarly, row addition was completed the same as for the double pendulum, except row six was added instead of row five. Unlike the double pendulum case, where an off-axis reaction in joint two does not create out of plane motion about joint one, any off axis reaction torque in joint two for the az-el system (created by the motion of body three) has the potential to induce motion in body two, at least about the gimbaled y-axis. Again, this configuration change was as simple as modifying a few indices in the MATLAB code.

This resulted in the following system:

$$\dot{x} = \begin{Bmatrix} \ddot{\theta}_{1y} \\ \ddot{\theta}_{2z} \end{Bmatrix} . \tag{5.5}$$

Expanding this system to the augmented state model yielded:

$$\dot{x}' = \begin{Bmatrix} \ddot{\theta}_{1y} \\ \ddot{\theta}_{2z} \\ \dot{\theta}_{1y} \\ \dot{\theta}_{2z} \end{Bmatrix} . \tag{5.6}$$

The augmented model was solved for a 10 second simulation, where initial conditions on $\dot{\theta}_1, \dot{\theta}_2, \theta_1,$ and θ_2 were all taken as zero (see Figure 21).

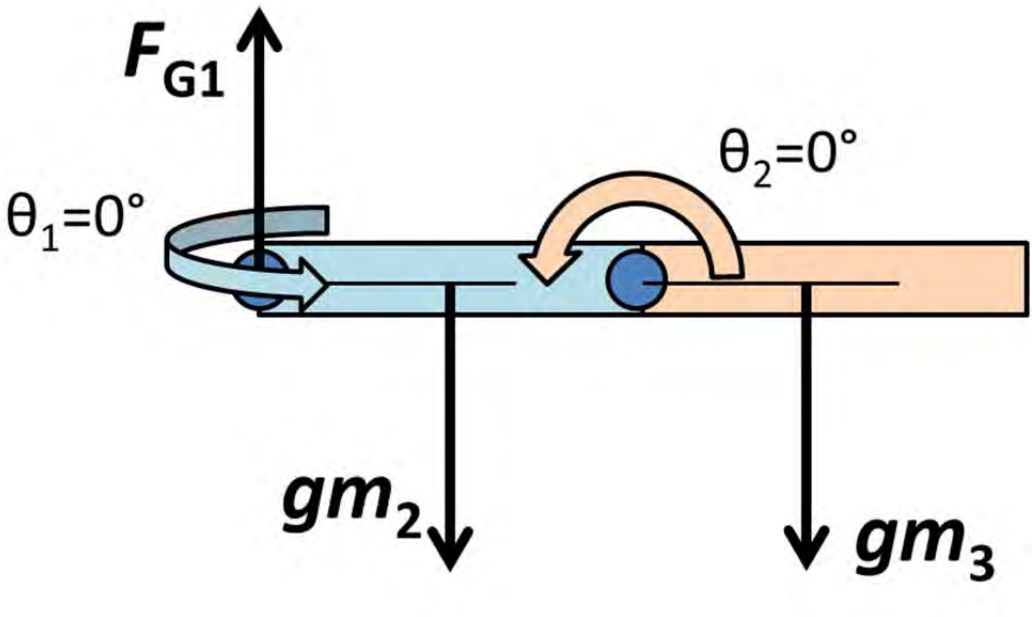


Figure 21. Azimuth-Elevation Initial Configuration

The resulting gimbal angles and rates were plotted, as shown in Figure 22.

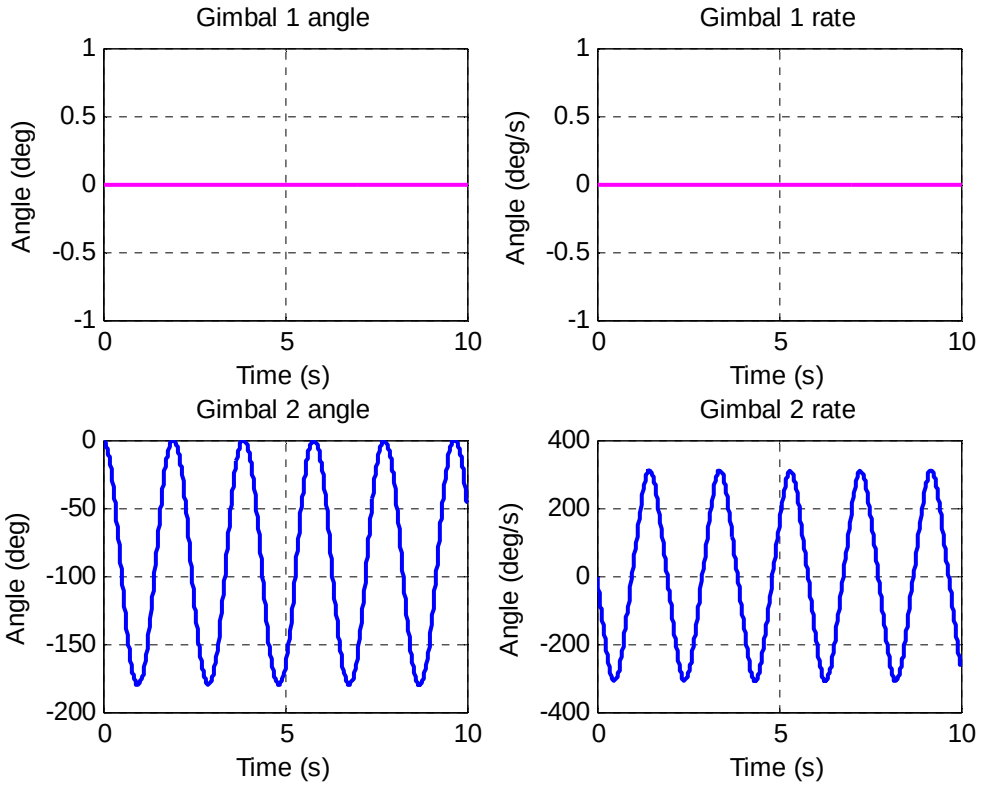


Figure 22. Azimuth-Elevation Results from Newton-Euler Simulation

Since body three has no off axis inertia components in this case, it does not generate any off axis torque about gimbal one. Therefore, no reaction torque is created in joint two, and no movement is induced into body two. While there are joint reaction forces in joint two, they are in the x and y axes, which are constrained in joint one, so they cannot create movement of body one or body two. The system thus operates similar to a single pendulum.

In order to investigate proper propagation of off axis forces and torques in the code, arbitrary off axis components were added to the inertia matrix for bodies one and two.

$$I2 = I3 = \begin{bmatrix} \frac{1}{2}mr^2 & \frac{m}{100} & -\frac{m}{100} \\ \frac{m}{100} & \frac{1}{12}ml_2 & \frac{m}{100} \\ -\frac{m}{100} & \frac{m}{100} & \frac{1}{12}ml_2 \end{bmatrix} = \begin{bmatrix} 0.050 & 0.100 & -0.100 \\ 0.100 & 0.833 & 0.100 \\ -0.100 & 0.100 & 0.833 \end{bmatrix} kg \cdot m^2 \qquad (5.7)$$

No further changes to the configuration or the code were made. The system was then simulated for 10 seconds. The results can be seen in Figure 23.

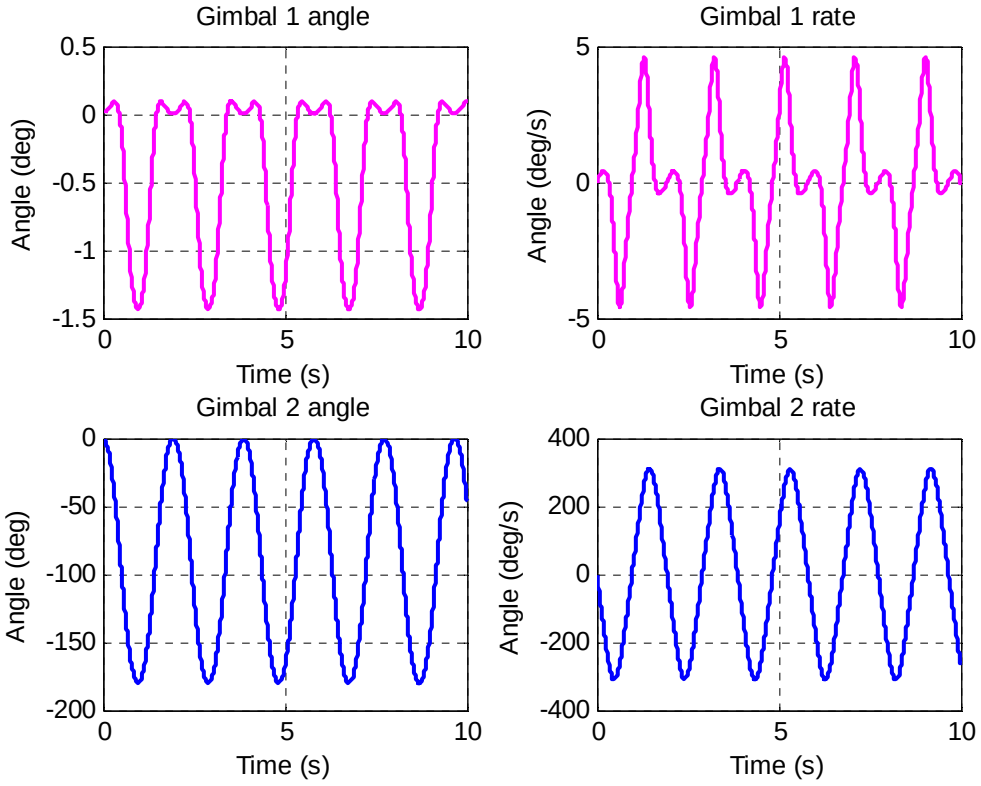


Figure 23. Azimuth-Elevation Results from Newton-Euler Simulation (Full Inertia Tensor)

The az-el configuration with full inertia tensor was then replicated in the SimMechanics workspace. The only change required to the double pendulum system was to add rotations to align joint one to the y-axis (-90° rotation in the x-axis before the joint to align the z-axis to the y-axis, and a 90° rotation after the joint to realign the axes), and

to add the off-axis inertia components. The model was simulated for 10 seconds, resulting in the response shown in Figure 24.

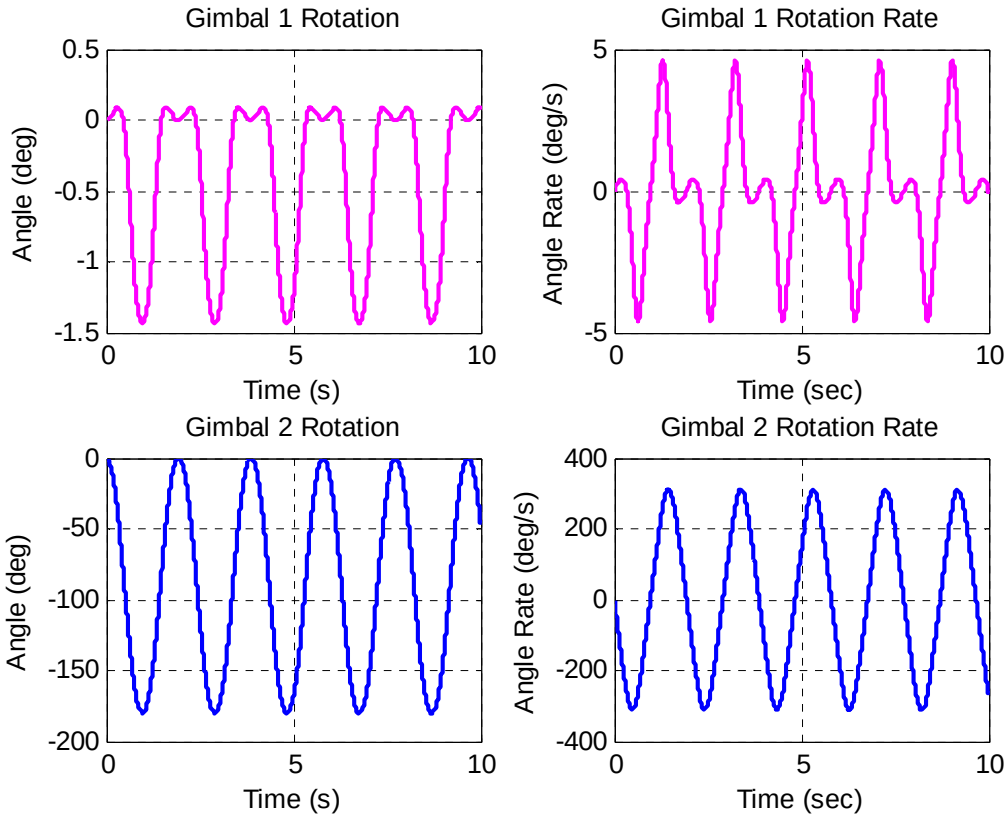


Figure 24. Azimuth-Elevation Results in SimMechanics (Full Inertia Tensor)

The absolute difference between the MATLAB and SimMechanics results can be seen in Figure 25. As in the case of the double pendulum, minimal differences are observed and these can be attributed to numerical errors in the ODE45 solver.

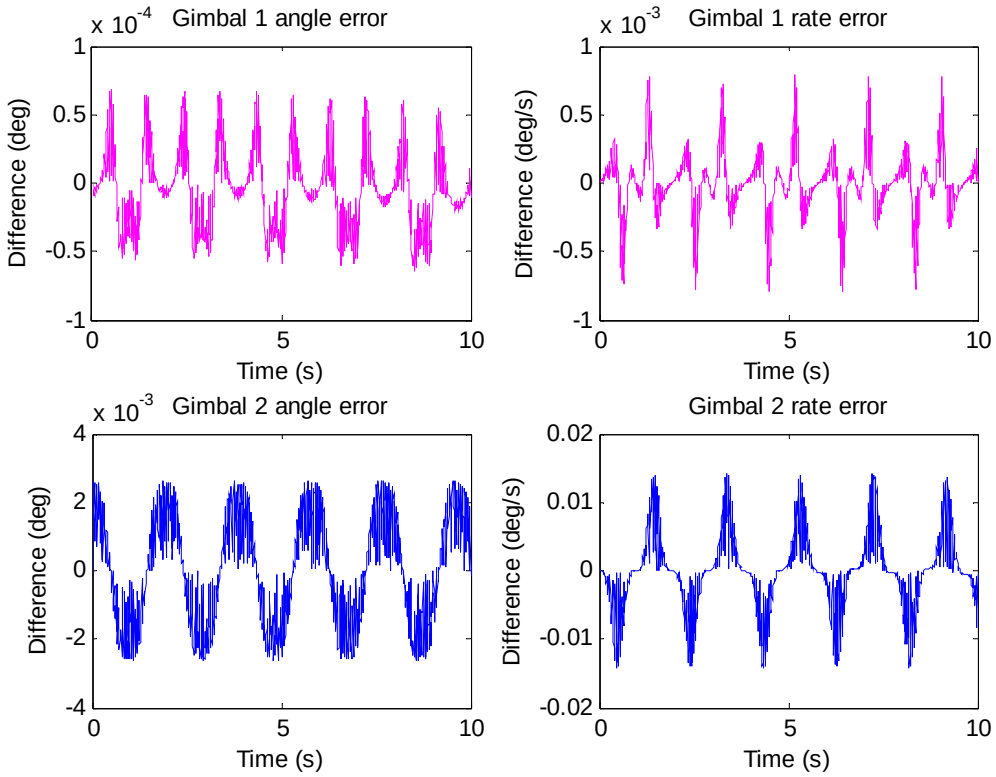


Figure 25. Azimuth-Elevation Simulation Residuals

D. AZIMUTH-ELEVATION SYSTEM WITH PD CONTROLLER

The final validation and verification test evolution was the addition of a control system to the model to investigate the effects of a dynamically applied control torque. A Proportional-Derivative (PD) controller was chosen for its simplicity and ease of insertion into the model. The PD control system only requires knowledge inputs for position and velocity. Both of these values are readily available in both the MATLAB and SimMechanics model as the joint angle and angle rate.

In a simple PD controller, the current system position and velocity are compared to a desired endpoint position and velocity. The difference between current and desired values creates an error signal. These position and velocity error signals are then multiplied by a proportional gain and a derivative gain, Kp and Kd, respectively. Finally, the gained signals are summed and the result becomes the control torque applied to the joint, as shown in Figure 26.

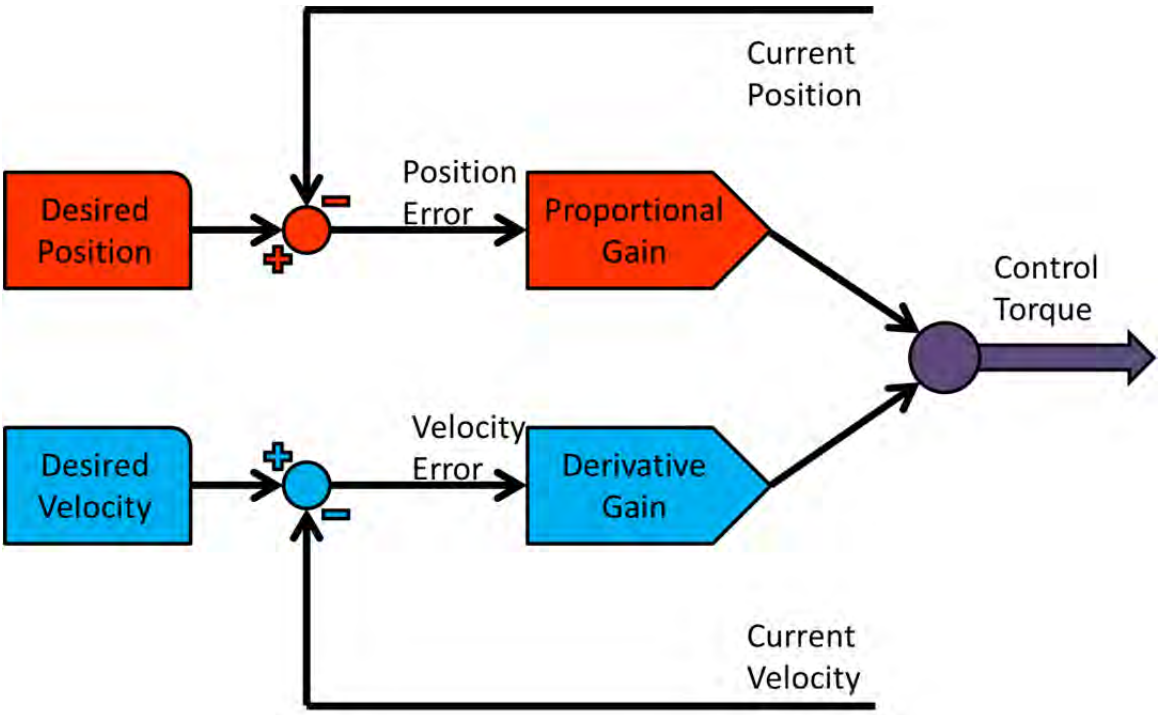


Figure 26. PD Controller

The shape of the system response is controlled by adjusting the proportional and derivative gain. By tuning these two variables, a system can be driven to an over-damped, critically damped, or underdamped response, with any desired settling time, natural frequency, overshoot, etc. For the simulation, a critically damped system was utilized, where the joint would rotate to the desired position as quickly as possible without overshooting the endpoint. In order to develop this response, the following procedure was used . This procedure is not intended to cover the derivation or the full explanation of a PD controller. Instead it is meant as a straightforward shortcut to implementing a rudimentary PD controller. For a full discussion of controller derivations and customization, see [20].

Starting with the settling time, t_s, of 20 seconds, and utilizing a damping ratio, ζ, of unity (necessary for a critically damped system), a natural frequency, ω_n, was determined by utilizing Equation (5.8).

$$\omega_n = \frac{4}{\zeta t_s} \tag{5.8}$$

$$\omega_n = \frac{4}{\zeta t_s} = \frac{4}{20} = 0.2 \tag{5.9}$$

From here, the required gains could be solved using Equation (5.10) and Equation (5.11), where I is the effective inertia of the body about the gimbal axis

$$Kp = I\omega_n^{\,2} = 0.04I \tag{5.10}$$

$$Kd = 2I\zeta\omega_n = 0.4I \tag{5.11}$$

Now, a gimbal control torque can be generated using Equation (5.12), where θ_d is the desired final joint angle, and $\dot{\theta}_d$ is the desired final joint rate.

$$T_{Gc} = Kp(\theta - \theta_{\mathrm{d}}) + Kd(\dot{\theta} - \dot{\theta}_{\mathrm{d}}) \tag{5.12}$$

This process was applied to both the azimuth and elevation gimbal using the same configuration as the full inertia az-el model. The inertia value in Equation (5.10) and Equation (5.11) were set to the sum of the principle inertia value of each link in each respective axis (I_{yy} for azimuth, I_{zz} for elevation), and adjusted using the parallel axis theorem as appropriate. Gravity was disabled in order to investigate the effects of the control logic and applied torque without external disturbances. The desired azimuth angle was set to 45°, and the desired elevation angle was set to 15°. The desired settling point rates for both azimuth and elevation rates were set to zero. This setup resulted in the response shown in Figure 27.

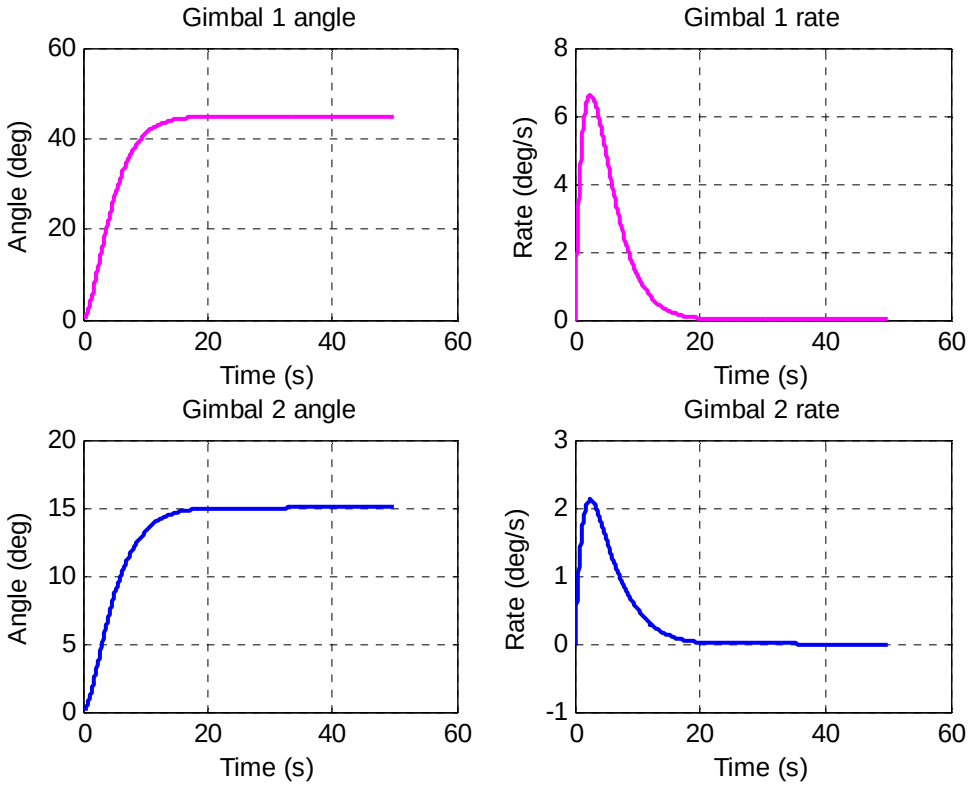


Figure 27. Azimuth-Elevation Results from Newton-Euler Simulation (PD controller)

The PD controller was implemented in the SimMechanics model using additional blocks in the joint subsystems as shown in Figure 28 and Figure 29. Controller gains were solved using the same code as the MATLAB model.

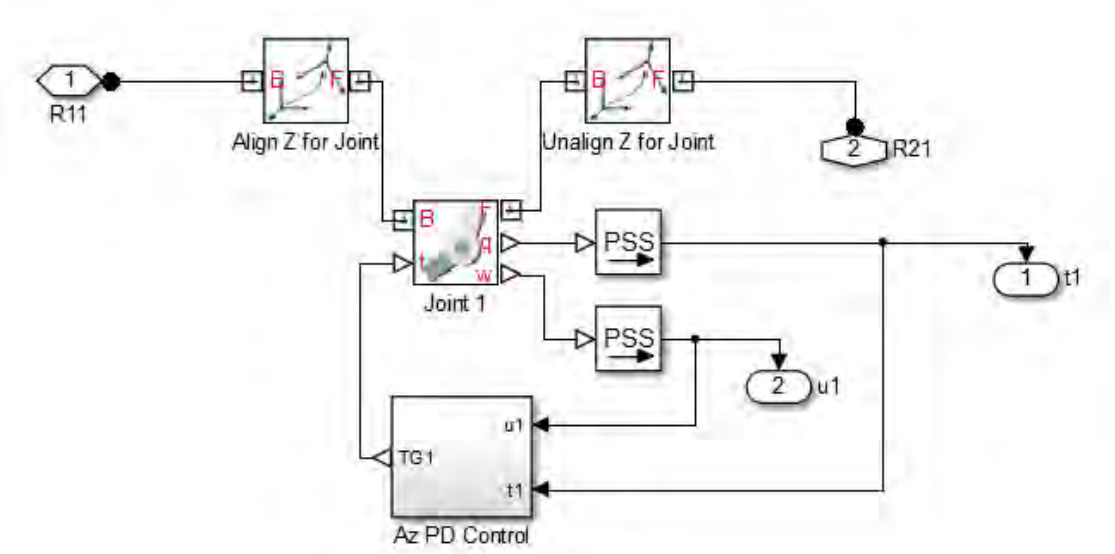


Figure 28. PD Control Feeds in Joint Subsystem

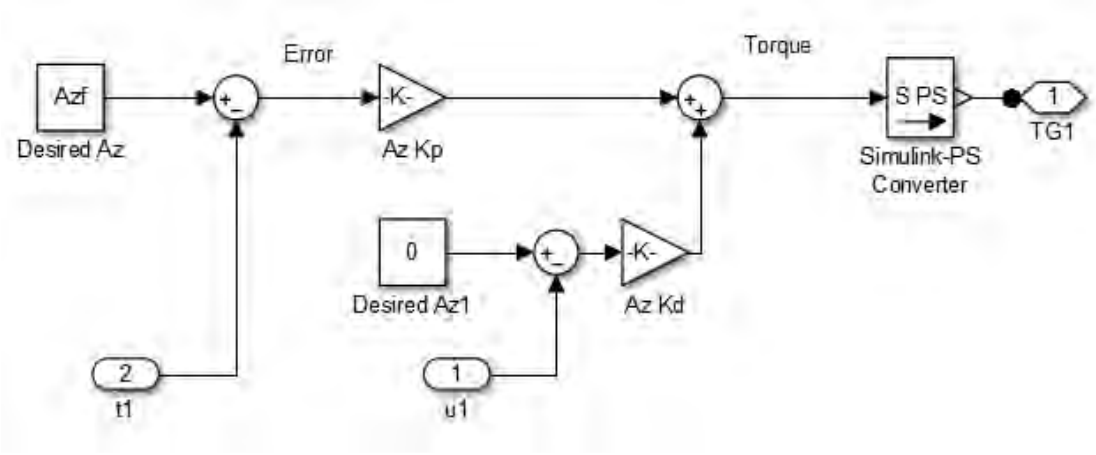


Figure 29. PD Controller Subsystem

The same initial conditions and simulation time were utilized for the SimMechanics model. This resulted in the response depicted in Figure 30.

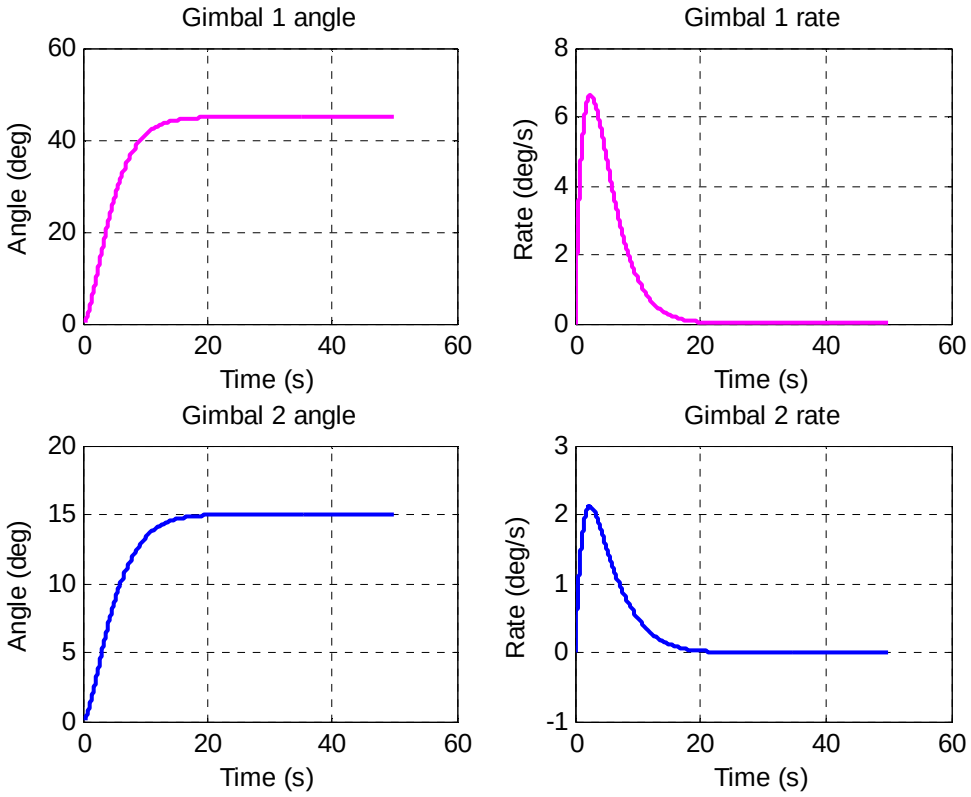


Figure 30. Azimuth-Elevation Results from SimMechanics (PD controller)

Comparing the response from SimMechanics to the response given by the developed code, the differences were negligible, as shown in Figure 31.

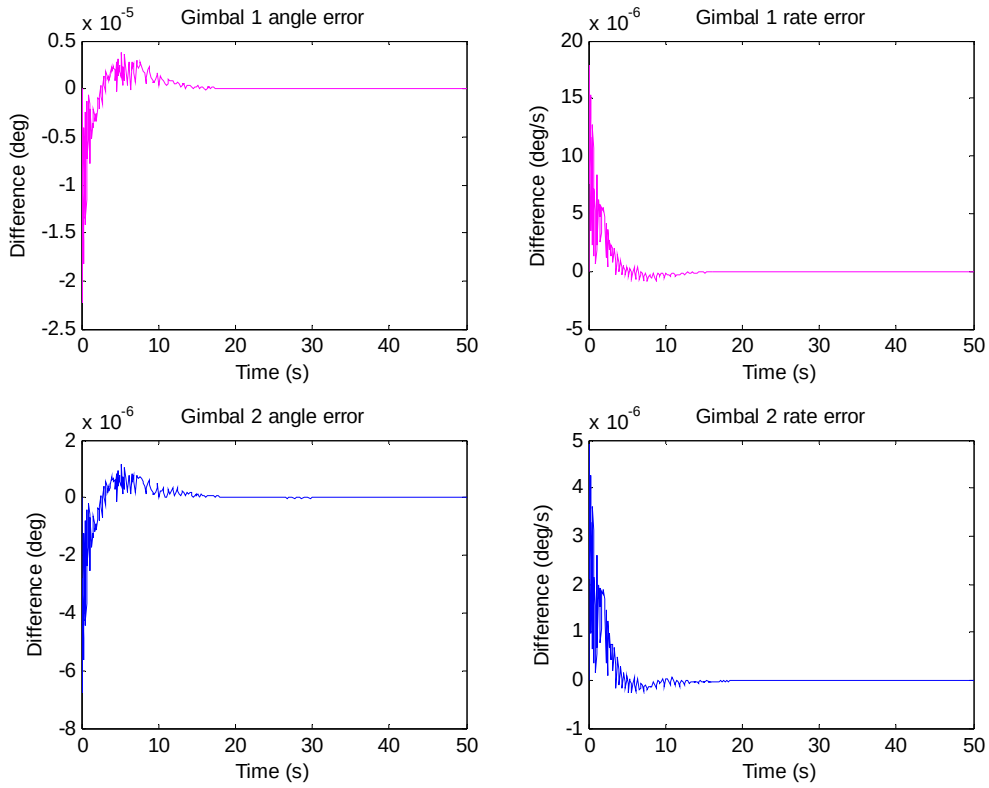


Figure 31. Azimuth-Elevation PD Controller Difference

E. VALIDATION AND VERIFICATION SUMMARY

While only four complete validation and verification tests were described in this chapter, a myriad of other tests were run in order to thoroughly vet the dynamics model. These included unique inertia tensors, various system dimensions and configurations, and even the addition of external forces and torques. This plethora of simulations was essential to identification and resolution of errors in the code for the dynamics model. Although they are not presented here due to the number of simulations, the similarity between each simulation, and the minutia involved, they were critical to ensuring an accurate dynamics response.

Most importantly, this model validated the row addition and reduction process described in Chapter IV. Tests were performed where the rows were reduced without the proper resultant torque propagation and the results whoed that the torques applied to the outer bodies did not influence the motion of inner bodies. Therefore, without the addition,

or some other method of solving and propagating a reaction torque, the system will not respond properly. However, for each case investigated, the proposed method worked flawlessly, and was able to produce the correct result.

VI. OPTIMAL CONTROL PROCESS

A. INTRODUCTION TO BOUNDARY VALUE PROBLEM

One approach to obtain an optimal control solution is to formulate a boundary value problem (BVP). The BVP is a very powerful tool for solving optimal control problems. With regard to this thesis, the BVP methodology was selected to explore the solution to the problem.

To solve the BVP, Pontryagin's Principle was applied [21]. This process begins with the definition of a problem statement. The scope of the problem is reduced by implementing constraints and boundary values. The problem includes a control function, u, and a state space model, $\dot{x} = f(x,u,t)$, to describe the system. The control function is the input to the state model that drives a response. This problem statement is used to develop and minimize a function called the Hamiltonian, *H*. Pontryagin's Principle states that the optimal control function minimizes the Hamiltonian for every instant in time [21]. Therefore, if the Hamiltonian can be minimized, the control that drives this minimization will be the optimal solution to the stated problem. To begin, a problem statement is formulated for the BVP. This consists of three parts: a function or objective to minimize, the dynamics of the system, and a set of boundary conditions and constraints.

The function or objective to minimize is the cost function, *J*. The cost function is a function of the system states and the control and can have two components, an endpoint cost (Mayer cost), *E*, and a running cost (Lagrange cost), *F*. The cost can consist of either or both of these two components [21]. The endpoint cost is a function of the states at the final time, and possibly the final time itself. Minimizing an endpoint cost will minimize some desired value at the final time of the problem. For example, minimizing the final time will yield the time optimal solution. Minimizing values such as position or velocity error can maximize accuracy by minimizing final endpoint error. The running cost is a function of the entire state trajectory, the control function, and the time. The running cost is calculated over the entire path of the solution. A running cost can be used to minimize

energy, control effort, etc. For example, by using spacecraft body angular momentum as a running cost, an optimal solution to minimize the disturbance caused by moving the antenna can be found. The entire cost can be represented by the Bolza cost functional, depicted in Equation [21].

$$J = E + \int_{t_0}^{t_f} F dt \tag{6.1}$$

The dynamics of the system are represented by the dynamic equations developed in Chapter II and Chapter III, or their simplified form. The system is described as a state space model, where a state and its time derivative are established. For the general three link system, the dynamics equations would be:

$$x = \begin{Bmatrix} \omega_1 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \\ v_1 \\ \theta_1 \\ \theta_2 \\ d_1 \\ \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{Bmatrix} \end{Bmatrix}; \quad \dot{x}' = \begin{Bmatrix} \dot{\omega}_1 \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \dot{v}_1 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{d}_1 \\ \begin{Bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{Bmatrix} \end{Bmatrix} = f(x', u, t) = \begin{Bmatrix} \begin{Bmatrix} (A - RU^{-1}S)^{-1}(T' - RU^{-1}F') \end{Bmatrix} \\ \dot{\theta}_1 \\ \dot{\theta}_2 \\ v1 \\ \begin{Bmatrix} \frac{1}{2} \begin{bmatrix} q_4 & -q_3 & q_2 & q_1 \\ q_3 & q_4 & -q_1 & q_2 \\ -q_2 & q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{bmatrix} \begin{Bmatrix} \omega 1 \\ \omega 2 \\ \omega 3 \\ 0 \end{Bmatrix} \end{Bmatrix} \end{Bmatrix} \tag{6.2}$$

$$u = \begin{Bmatrix} T_{G1c} \\ T_{G2c} \end{Bmatrix} . \tag{6.3}$$

Lastly, the boundaries and constraints of the problem are defined. Each state can be given any combination of fixed values, boundary conditions, or functions. For example, an initial position and velocity can be defined, or a lower and upper boundary can be defined in order to specify a range of initial values to let the solution of the dictate the optimal value. Boundaries can be placed on values over the entire path of the problem, forcing a constraint on the solution. For example, lower and upper angle constraints could be set to model soft or hard stops for joints, or limits could be set on

control torque to constrain the maximum motor output. Lastly, fixed values or boundary conditions could also be functions of (i.e., functions of time or other states), as opposed to explicit quantities.

Putting all these components together, the problem statement can be written in a convenient form [21].

$$x = \begin{Bmatrix} \omega_1 & \dot{\theta}_1 & \dot{\theta}_2 & v_1 & q & \theta_1 & \theta_2 & d_1 \end{Bmatrix}^T \quad u = \begin{Bmatrix} T_{G1c} \\ T_{G2c} \end{Bmatrix}$$

Minimize $\quad J = E + \int_{t_0}^{t_f} F dt$

Subject to:

$$\dot{x}' = \begin{Bmatrix} \dot{\omega}_1 \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \dot{v}_1 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{d}_1 \\ \begin{Bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{Bmatrix} \end{Bmatrix} = f(x', u, t) = \begin{Bmatrix} \begin{Bmatrix} (A - RU^{-1}S)^{-1}(T' - RU^{-1}F') \end{Bmatrix} \\ \dot{\theta}_1 \\ \dot{\theta}_2 \\ v_1 \\ \begin{Bmatrix} \frac{1}{2} \begin{bmatrix} q_4 & -q_3 & q_2 & q_1 \\ q_3 & q_4 & -q_1 & q_2 \\ -q_2 & q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{bmatrix} \begin{Bmatrix} \omega 1 \\ \omega 2 \\ \omega 3 \\ 0 \end{Bmatrix} \end{Bmatrix} \end{Bmatrix}$$

$$\begin{Bmatrix} \omega_1(t_0) \\ \dot{\theta}_1(t_0) \\ \dot{\theta}_2(t_0) \\ v_1(t_0) \\ d_1(t_0) \\ \theta_1(t_0) \\ \theta_2(t_0) \\ q(t_0) \end{Bmatrix} = \begin{Bmatrix} [0,0,0]^{\mathrm{T}} \\ [0,0,0]^{\mathrm{T}} \\ [0,0,0]^{\mathrm{T}} \\ [0,0,0]^{\mathrm{T}} \\ [0,0,0]^{\mathrm{T}} \\ [0,0,0]^{\mathrm{T}} \\ [0,0,0]^{\mathrm{T}} \\ [0,0,0,1]^{T} \end{Bmatrix} \quad t_0 = 0 \quad \begin{Bmatrix} \theta_1(t_f) \\ \theta_2(t_f) \\ \dot{\theta}_1(t_f) \\ \dot{\theta}_2(t_f) \end{Bmatrix} = \begin{Bmatrix} [\theta_{1xf}, \theta_{1yf}, \theta_{1zf}]^T \\ [\theta_{2xf}, \theta_{2yf}, \theta_{2zf}]^T \\ [\dot{\theta}_{1xf}, \dot{\theta}_{1yf}, \dot{\theta}_{1zf}]^T \\ [\dot{\theta}_{2xf}, \dot{\theta}_{2yf}, \dot{\theta}_{2zf}]^T \end{Bmatrix} \tag{6.4}$$

This problem setup defines initial conditions for the state, initial time, final conditions for only joint orientation, and boundaries for joint control torque. Other final states and intermediate states (states at any time between initial and final) are not bounded. This will be the basic foundational problem statement for all subsequent optimal control solutions. Slight additions or modifications will be made on a case-by case basis to fit the desired model or objective. Specifically different cost functions will be implemented along with various other constraints.

With the problem formulation established, the process of formulating the BVP can begin. First, the adjoint convector is introduced. The adjoint convector is a vector of costates (λ), where every costate is associated with a state, and effectively mirrors the time history of the state in a non-linear way [21]. The costates essentially provide a reference for the cost of each state that is not obstructed by the relative magnitudes of the states themselves [21].

$$\lambda = \begin{Bmatrix} \lambda_{\omega_1} \\ \lambda_{\dot{\theta}_1} \\ \lambda_{\dot{\theta}_2} \\ \lambda_{v_1} \\ \lambda_{\theta_1} \\ \lambda_{\theta_2} \\ \lambda_{d_1} \\ \lambda_{\mathbf{q}} \end{Bmatrix} \tag{6.5}$$

Using this costate vector, the Hamiltonian is defined:

$$H = F + \lambda^T f \ . \tag{6.6}$$

The Hamiltonian must be minimized with respect to control, and the control trajectory that achieves this result is considered the optimal control trajectory.

The solution is obtained by with the Hamiltonian Minimization Condition, which in the absence of constraints on the controls is also known as the Euler-Lagrange equation.

$$\frac{\partial H}{\partial u} = 0 \tag{6.7}$$

However, the application of the Hamiltonian Minimization Condition depends on knowledge of the dynamics of the costates. Therefore, the costates must be solved first using a series of differential equations.

Each costate is defined by an adjoint equation, expressed in Equation (6.8) [21].

$$-\dot{\lambda} = \frac{\partial H}{\partial x} \tag{6.8}$$

The costate vector and the state dynamics each comprise systems of differential equations that must be solved to express the Hamiltonian so that the Hamiltonian Minimization Condition can be applied. The introduction of the adjoint equation results in a system of *2N* variables (*N* states and *N* costates), and *2N* differential equations, requiring *2N* known point conditions to generate a unique solution [21]. For the three body problem statement in Equation (6.4), there are 25 unique states. The problem statement includes 25 unique boundary conditions at the initial time, but only six unique boundary conditions at the final time. Therefore, there are 50 differential equations with 31 boundary conditions specified. If every state and costate were provided with an endpoint condition, there would be 50 equations with 50 boundary conditions, and a solution could be found. It is also important to note that if final time is not explicitly defined, this results in an additional boundary condition, resulting in a total of 32 necessary conditions for the general problem statement.

In order to solve for the additional boundary conditions, the Endpoint Lagrangian, $\bar{E}$, the vector of Lagrange multipliers, *v*, and the terminal Transversality Condition are introduced [21]. First, the vector of Lagrange multipliers contains one variable for every state that has an explicit endpoint condition. For example, the general three body problem statement provides two endpoint conditions; therefore there are two associated Lagrange multipliers.

$$v = \begin{Bmatrix} v_1 \\ v_2 \end{Bmatrix} \tag{6.9}$$

This vector is used in the definition of the Endpoint Lagrangian, where *e* is a vector of state endpoint conditions [21].

$$\bar{E} = E + v^T e \tag{6.10}$$

Lastly, the terminal Transversality Condition is introduced [21].

$$\lambda(t_f) = \frac{\partial \bar{E}}{\partial x(\mathrm{t}_f)} \tag{6.11}$$

Together, Equation (6.10) and Equation (6.11) provide the missing conditions. For example, application of the Transversality Condition introduces an additional 24 equations, but only introduces 6 additional variables. This results in producing 18 of the missing conditions since many of the partial derivatives are zero. If every state were provided with an endpoint condition, *N* additional Lagrange multipliers would be introduced along with *N* additional differential equations. Therefore, since the problem would already have *2N* point conditions before calculating terminal Transversality, this step becomes superfluous and no additional information would be obtained.

One missing condition remains since final time was not defined. The Hamiltonian Value Condition is therefore introduced to satisfy the final necessary point condition, as defined in Equation (6.12) [21]. This step is also superfluous if a final time is specified.

$$H(t_f) = -\frac{\partial \bar{E}}{\partial \mathrm{t}_f} \tag{6.12}$$

It can be noted that if there are constraints to the problem, particularly with respect to control constraints, additional terms are required in the Hamiltonian and the Euler-Lagrange equation no longer applies. Instead, the partial derivative, ∂H/∂u must be interpreted as a switching function.

This overview of optimal control theory is meant to introduce the reader to some of the comcepts involved, but is not meant to be an exhaustive treatment For a more thorough discussion, see [14] and [21].

B. DOUBLE PENDULUM EXAMPLE

Most optimal control problems quickly become so complex that they require numerical techniques to solve. Nonetheless, the double pendulum is an example of a system that is simple enough to provide a useful example of the steps involved in this

process. To begin, the same double pendulum system as used previously in validation and verification can be modified to provide an example application of the approach. Eliminating gravity further reduces the complexity, and assuming the availability of a control torque for each gimbal provides a controlled system to optimize. The dimensions, masses, and inertias of the system were maintained from the validation and verification model.

After performing system tailoring and solving for the dynamic equations of the system, the state space in Equation (6.13) is produced.

$$x = \begin{Bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \theta_1 \\ \theta_2 \end{Bmatrix}$$

$$\dot{x} = \begin{Bmatrix} \dfrac{-6T_{G1} + 6T_{G2} - 30\dot{\theta}_1^2 sin(\theta_2) - 30\dot{\theta}_2^2 sin(\theta_2) + 9T_{G2} cos(\theta_2) - 60\dot{\theta}_1\dot{\theta}_2 sin(\theta_2) - 45\dot{\theta}_1^2 cos(\theta_2) sin(\theta_2)}{(45cos(\theta_2)^2 - 80)} \\ \dfrac{\begin{array}{c}(12T_{G1} - 60T_{G2} + 300\dot{\theta}_1^2 sin(\theta_2) + 60\dot{\theta}_2^2 sin(\theta_2) + 90\dot{\theta}_1^2 sin(2\theta_2) + 45\dot{\theta}_2^2 sin(2\theta_2) + ... \\ ...18T_{G1} cos(\theta_2) - 36T_{G2} cos(\theta_2) + 120\dot{\theta}_1\dot{\theta}_2 sin(\theta_2) + 90\dot{\theta}_1\dot{\theta}_2 sin(2\theta_2))\end{array}}{(45\cos(2\theta_2) - 115)} \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{Bmatrix} \quad (6.13)$$

$$u = \begin{Bmatrix} T_{G1} \\ T_{G2} \end{Bmatrix}$$

Suppose the goal of the system is to point the outer pendulum link at some angle with respect to the inertial frame, and suppose the desired optimization objective is a combination of the minimum time with the minimum amount of torque. This gives the cost function represented in Equation (6.14).

$$J = t_f + \frac{1}{2}\int_{t_0}^{t_f} (\mathrm{T_{G1}}^2 + \mathrm{T_{G2}}^2) dt \quad (6.14)$$

The point constraints of the problem are then introduced. Suppose the starting joint angles and rates are set to zero, the initial time is zero, the final rates are zero, and the final combined angle is 90° (any combination of joint one and joint two angles that will result in the outer link pointing at 90° relative to the inertial).

Lastly, some control and state boundaries are defined, effectively limiting the solution space available to the solver. The minimum and maximum torque limit for both joints is set to -1 Nm and 1 Nm, respectively. Joint angles are allowed to move freely between -180° and 180°. Finally, the joint rates are constrained to between -5 °/s and 5 °/s.

This results in a relatively straightforward problem definition.

$$x = \begin{Bmatrix} \dot{\theta}_1 & \dot{\theta}_2 & \theta_1 & \theta_2 \end{Bmatrix} \qquad u = \begin{Bmatrix} T_{G1} & T_{G2} \end{Bmatrix}$$

$$Minimize \qquad J = t_f + \frac{1}{2}\int_{t_0}^{t_f} (\mathrm{T_{G1}}^2 + \mathrm{T_{G2}}^2)dt$$

$$Subject\ to:$$

$$\ddot{\theta}_1 = \frac{-6T_{G1} + 6T_{G2} - 30\dot{\theta}_1^2 sin(\theta_2) - 30\dot{\theta}_2^2 sin(\theta_2) + 9T_{G2}cos(\theta_2) - 60\dot{\theta}_1\dot{\theta}_2 sin(\theta_2) - 45\dot{\theta}_1^2 cos(\theta_2)sin(\theta_2)}{(45cos(\theta_2)^2 - 80)}$$

$$\ddot{\theta}_2 = \frac{\begin{array}{l}(12T_{G1} - 60T_{G2} + 300\dot{\theta}_1^2 sin(\theta_2) + 60\dot{\theta}_2^2 sin(\theta_2) + 90\dot{\theta}_1^2 sin(2\theta_2) + 45\dot{\theta}_2^2 sin(2\theta_2) + ... \\ ...18T_{G1}cos(\theta_2) - 36T_{G2}cos(\theta_2) + 120\dot{\theta}_1\dot{\theta}_2 sin(\theta_2) + 90\dot{\theta}_1\dot{\theta}_2 sin(2\theta_2))\end{array}}{(45\cos(2\theta_2) - 115)}$$

$$\left(\dot{\theta}_1(t_0), \dot{\theta}_2(t_0), \theta_1(t_0), \theta_2(t_0)\right) = (0,0,0,0)$$

$$(\dot{\theta}_1(t_f), \dot{\theta}_2(t_f), \theta_1(t_f)) = (0,0,\frac{\pi}{2} - \theta_2(t_f))$$

Figure 32. Double Pendulum Optimization Problem Definition

Notice that since it is irrelevant how the final angle is apportioned between the two joint angles, a single endpoint for joint one was defined in terms of the final total

angle and final endpoint for joint two. This choice is completely arbitrary, but is necessary in order to set a final boundary condition.

Starting the solution process, the Hamiltonian is first calculated.

$$H = F + \lambda^T \dot{x}$$
$$H = \frac{1}{2}(\mathrm{T_{G1}}^2 + \mathrm{T_{G2}}^2) + \lambda_{\dot{\theta}_1}\ddot{\theta}_1 + \lambda_{\dot{\theta}_2}\ddot{\theta}_2 + \lambda_{\theta_1}\dot{\theta}_1 + \lambda_{\theta_2}\dot{\theta}_2 \tag{6.15}$$

The Hamiltonian Minimization Condition is applied next to give:

$$H = \frac{1}{2}(\mathrm{T_{G1}}^2 + \mathrm{T_{G2}}^2) + \lambda_{\dot{\theta}_1}\ddot{\theta}_1 + \lambda_{\dot{\theta}_2}\ddot{\theta}_2 + \lambda_{\theta_1}\dot{\theta}_1 + \lambda_{\theta_2}\dot{\theta}_2$$
$$\frac{\partial H}{\partial \mathrm{T_{G1}}} = \mathrm{T_{G1}} + \lambda_{\dot{\theta}_1}\left(\frac{-6T_{G1}}{(45cos(\theta_2)^2 - 80)}\right) + \lambda_{\dot{\theta}_2}\left(\frac{12T_{G1} + 18T_{G1}cos(\theta_2)}{(45\cos(2\theta_2)\text{ - }115)}\right) = 0 \tag{6.16}$$
$$\frac{\partial H}{\partial \mathrm{T_{G2}}} = \mathrm{T_{G2}} + \lambda_{\dot{\theta}_1}\left(\frac{6T_{G2} + 9T_{G2}cos(\theta_2)}{(45cos(\theta_2)^2 - 80)}\right) + \lambda_{\dot{\theta}_2}\left(\frac{60T_{G2} - 36T_{G2}cos(\theta_2)}{(45\cos(2\theta_2)\text{ - }115)}\right) = 0.$$

Solving for the controls yields:

$$\mathrm{T_{G1}} = -1 + \lambda_{\dot{\theta}_1}\left(\frac{6}{(45cos(\theta_2)^2 - 80)}\right) - \lambda_{\dot{\theta}_2}\left(\frac{12 + 18cos(\theta_2)}{(45\cos(2\theta_2)\text{ - }115)}\right)$$
$$\mathrm{T_{G2}} = -1 - \lambda_{\dot{\theta}_1}\left(\frac{6 + 9cos(\theta_2)}{(45cos(\theta_2)^2 - 80)}\right) - \lambda_{\dot{\theta}_2}\left(\frac{60 - 36cos(\theta_2)}{(45\cos(2\theta_2)\text{ - }115)}\right). \tag{6.17}$$

Next, the adjoint equations are calculated.

$$-\dot{\lambda}_{\dot{\theta}_1} = \frac{\partial H}{\partial u_1} = \left(\begin{array}{c} \frac{\lambda_{\dot{\theta}_1}(-60\dot{\theta}_1 sin(\theta_2) - 60\dot{\theta}_2 sin(\theta_2) - 90\dot{\theta}_1 cos(\theta_2) sin(\theta_2))}{(45cos(\theta_2)^2 - 80)} \ldots \\ \ldots + \frac{\lambda_{\dot{\theta}_2}(600\dot{\theta}_1 sin(\theta_2) + 180\dot{\theta}_1 sin(2\theta_2) + 120\dot{\theta}_2 sin(\theta_2) + 90\dot{\theta}_2 sin(2\theta_2))}{(45\cos(2\theta_2) - 115)} + \lambda_{\theta_1} \end{array} \right)$$

$$-\dot{\lambda}_{\dot{\theta}_2} = \frac{\partial H}{\partial u_2} = \left(\begin{array}{c} \frac{\lambda_{\dot{\theta}_1}(-60\dot{\theta}_2 sin(\theta_2) - 60\dot{\theta}_1 sin(\theta_2))}{(45cos(\theta_2)^2 - 80)} \ldots \\ \ldots + \frac{\lambda_{\dot{\theta}_2}(120\dot{\theta}_2 sin(\theta_2) + 90\dot{\theta}_2 sin(2\theta_2) + 120\dot{\theta}_1 sin(\theta_2) + 90\dot{\theta}_1 sin(2\theta_2))}{(45\cos(2\theta_2) - 115)} + \lambda_{\theta_2} \end{array} \right)$$

$$-\dot{\lambda}_{\theta_1} = \frac{\partial H}{\partial \theta_1} = 0$$

$$-\dot{\lambda}_{\theta_2} = \frac{\partial H}{\partial \theta_1} = \left(\begin{array}{c} \frac{\lambda_{\dot{\theta}_1}(-30\dot{\theta}_1^2 cos(\theta_2) - 30\dot{\theta}_2^2 cos(\theta_2) - 45\dot{\theta}_1^2 cos(\theta_2)^2 + 45\dot{\theta}_1^2 sin(\theta_2)^2 - 9T_{G2} sin(\theta_2) - 60\dot{\theta}_1\dot{\theta}_2 cos(\theta_2))}{(45cos(\theta_2)^2 - 80)} \ldots \\ \ldots + \frac{\lambda_{\dot{\theta}_1}(90\cos(\theta_2)\sin(\theta_2))(-6T_{G1} + 6T_{G2} - 30\dot{\theta}_1^2 sin(\theta_2) - 30\dot{\theta}_2^2 sin(\theta_2) + 9T_{G2} cos(\theta_2) - 60\dot{\theta}_1\dot{\theta}_2 sin(\theta_2) - 45\dot{\theta}_1^2 cos(\theta_2) sin(\theta_2))}{(45cos(\theta_2)^2 - 80)^2} \\ \lambda_{\dot{\theta}_2}(300\dot{\theta}_1^2 \cos(\theta_2) + 60\dot{\theta}_1^2 \cos(\theta_2) + 180\dot{\theta}_1^2 \cos(2\theta_2) + 90\dot{\theta}_2^2 \cos(2\theta_2) \ldots \\ \ldots + \frac{\ldots - 18T_{G1}\sin(\theta_2) + 36T_{G2}\sin(\theta_2) + 120\dot{\theta}_1\dot{\theta}_2 \cos(\theta_2) + 180\dot{\theta}_1\dot{\theta}_2 \cos(2\theta_2))}{(45\cos(2\theta_2) - 115)} \ldots \\ \lambda_{\dot{\theta}_2}(90\sin(2\theta_2))(12T_{G1} - 60T_{G2} + 300\dot{\theta}_1^2 sin(\theta_2) + 60\dot{\theta}_2^2 sin(\theta_2) + 90\dot{\theta}_1^2 sin(2\theta_2) + 45\dot{\theta}_2^2 sin(2\theta_2) \ldots \\ \ldots + \frac{\ldots 18T_{G1} cos(\theta_2) - 36T_{G2} cos(\theta_2) + 120\dot{\theta}_1\dot{\theta}_2 sin(\theta_2) + 90\dot{\theta}_1\dot{\theta}_2 sin(2\theta_2))}{(45\cos(2\theta_2) - 115)^2} \end{array} \right) \quad (6.18)$$

At this point, the problem contains eight differential equations, but only seven point conditions (four point conditions for initial conditions, and three defined final conditions). This leaves two additional endpoint conditions, including one required since final time is not specified. Therefore, the Terminal Transversality and the Hamiltonian Value Condition are both necessary steps.

First, the Endpoint Lagrangian is calculated:

$$\bar{E} = E + v^T e = t_f + v_1(\dot{\theta}_{1f}) + v_2(\dot{\theta}_{2f}) + v_3(\frac{\pi}{2} - \theta_{2f}) \tag{6.19}$$

Using this, Terminal Transversality is applied to yield four additional equations, while only adding three additional variables.

$$\begin{aligned}
\lambda_{\dot{\theta}_1}(t_f) &= \frac{\partial E}{\partial(\dot{\theta}_{1f})} = v_1 \\
\lambda_{\dot{\theta}_2}(t_f) &= \frac{\partial E}{\partial(\dot{\theta}_{2f})} = v_2 \\
\lambda_{\theta_1}(t_f) &= \frac{\partial E}{\partial(\theta_{1f})} = 0 \\
\lambda_{\theta_2}(t_f) &= \frac{\partial E}{\partial(\theta_{2f})} = -v_3
\end{aligned} \tag{6.20}$$

Lastly, the Hamiltonian Value Condition yields the final condition necessary for a unique solution.

$$H(t_f) = -\frac{\partial \bar{E}}{\partial t_f} = -1 \tag{6.21}$$

It can be seen that the equations yielded by this process can become difficult to solve, even for a simple problem like the double pendulum. To solve the BVP, various methods can be utilized, such as a shooting or collocation technique. Shooting uses a guess and check method to hone in on the optimal solution. However, shooting suffers from the curse of sensitivity [21], in which the Hamiltonian BVP has poles on both halves of the imaginary plane which causes issues in propagating the equations.

To circumvent the curse of sensitivity, collocation can be used. Using this approach, the timeframe is split into discrete time intervals and the entire solution is

obtained at once using linear algebra of the form Ax=b. However, this type of solution suffers from the curse of dimensionality, in that the size of the discretized system grows exponentially as smaller time intervals are utilized. A system of N intervals requires N^2 elements. For example, if a system is split into 10 intervals, a square matrix solution of 100 elements is required. For systems with quick dymanics, more time divisions are necessary. This can at best slow down the solver, and at worst break the solver. This problem has, however, been alleviated over the last decades with the implementation of more complex and more powerful computing platforms. One additional way to help improve solution efficiency and accuracy is to first solve the system using a coarse grid, and then use this solution as a guess and solve the system again on a finer grid. Building up a solution in this way, an accurate solution can be obtained faster without the need for extensive computational hardware.

C. SOLVING THE OPTIMAL CONTROL PROBLEM IN DIDO

Even though the equations to solve the BVP for optimal control of the double pendulum model can be derived, they are quite complex. Thus, the solution must be obtained using an optimal control software package. One such optimal control program is DIDO [14]. DIDO is a MATLAB based software suite that is specifically designed to easily and quickly solve optimal control problems using the theory described in Section A. DIDO only requires the dynamics equations and the bounds on the system, then solves the optimal control problem using the Legendre pseudospectral method [22]. Most problem definitions are easy to implement in DIDO and the software allows for easy variation of the problem in order to explore a wide range of problem definitions. Furthermore, DIDO provides tools to allow insight into controls, states, covectors, etc., in order to glean deeper understanding of a system response or to troubleshoot the problem specification.

To begin, the user must define the problem. This involves specifying five MATLAB files: a dynamics file, a cost file, an events file, an optional path file, and a problem file. The dynamics file describes the state dynamics of the system. The states are provided for each interval in time and the dynamics are returned for each instant. The

cost file describes the cost function of the system as both an endpoint cost and a running cost. The events file describes the known boundary conditions on the system states. The path function provides additional constraints on the solution at each node, as required. Lastly, the problem file describes the scales and boundaries of the problem, and coordinates the solution. For example, the problem file establishes how many states and what state variables make up the problem, as well as what boundaries are specified for each state variable, limiting the solution space. The problem file also establishes the values necessary for the event conditions, i.e., the initial and final values of each state.

Once these MATLAB files are provided, DIDO automatically solves the conditions necessary for Pontryagin's Principle and finds an extremal solution by applying the equations described in Section A. The solution for a version of the double pendulum problem with state and control constraints can be seen in Figure 33 and Figure 34. In the solution shown, the gimbal rates and gimbal torques were constrained to be less than 5°/s and 0.25 Nm, respectively.

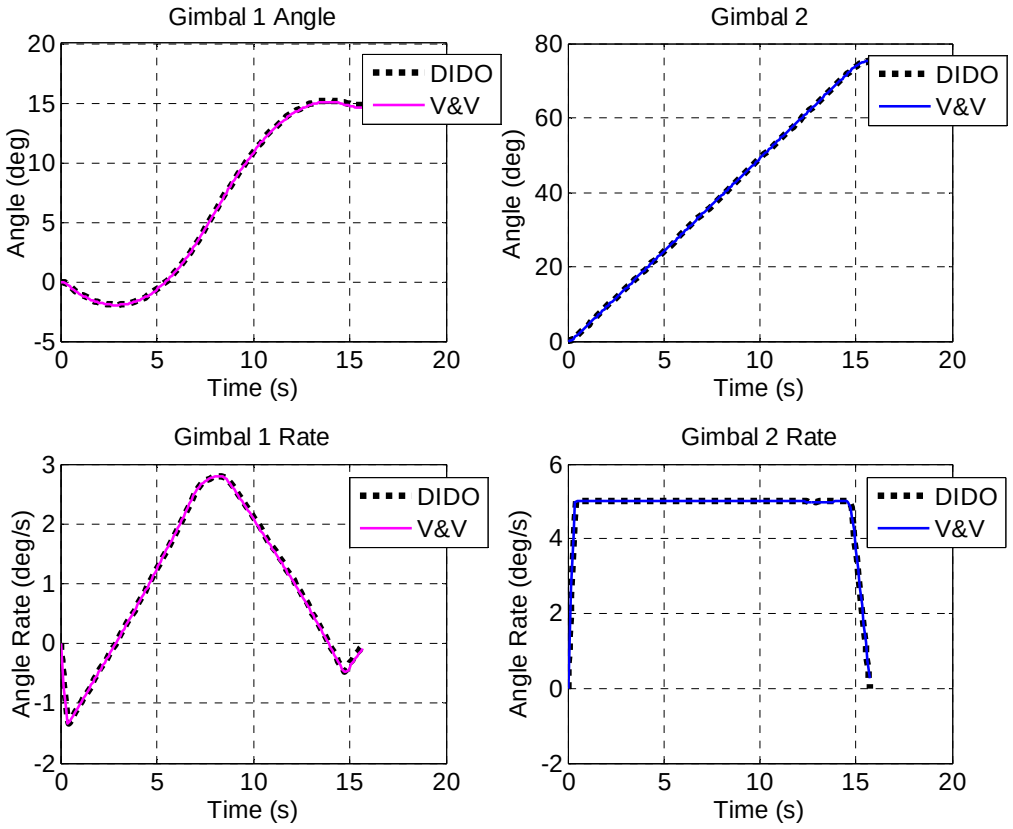


Figure 33. Optimal Control of the Double Pendulum (State Trajectories)

In order to ensure the solution provided by DIDO is feasible, the dynamics of the system are driven by the controls solved by DIDO. This propagation can be done using ODE45. The results of this validation and verification test are seen as the solid colored line in Figure 33, whereas the DIDO trajectory is gived with the dotted black line. With the approach to optimal control outlined and validated, the DIDO program was utilized to investigate problems relevant to slew control of satellite antennas.

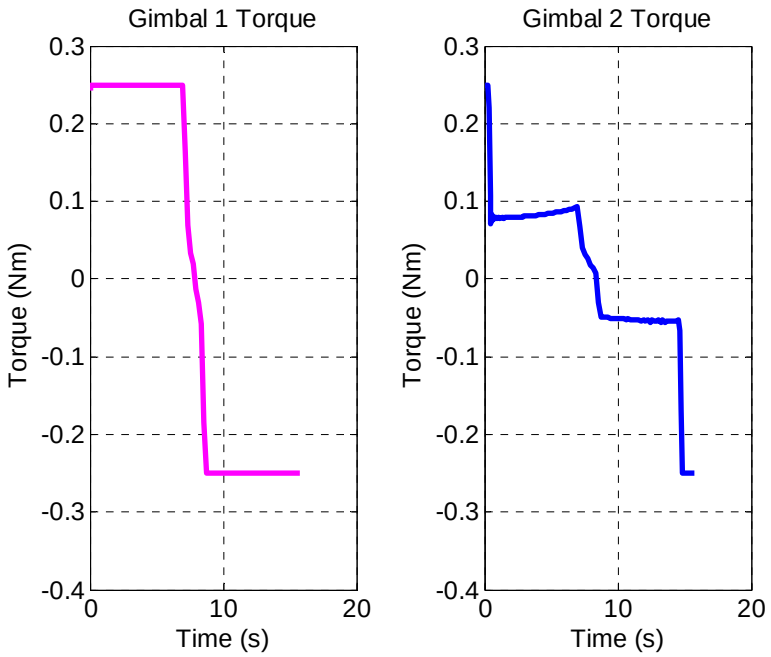


Figure 34. Optimal Control of the Double Pendulum (Control Trajectories)

THIS PAGE INTENTIONALLY LEFT BLANK

VII. TDRS OPTIMIZATION

A. TDRS SLEW CONTROL

The primary goal of this thesis is to explore the optimization of TDRS antenna slews. However, in order to properly formulate the problem statement and in order to compare optimum slew improvements over current slews, the conventional approach to controlling the TDRS SA antenna must be understood. Current slew control of the SA antenna is described in this section.

A slew and track consists of three parts, an open loop program track in order to move the antenna from one user to another, a pull in phase, and an autotrack phase [1]. The program track gets the antenna pointing close enough that a signal feedback loop can be implemented so that the pull in phase can reduce the error from around 0.22° to 0.03°. Next, the system enters the autotrack phase where it will remain for the extend of the relay [11, 10]. Once in autotrack, the beamwidth of the communications signal drives stringent pointing requirements, as shown in Table 2. Extensive study has been given to improving the closed loop pull in and autotrack phases of the slew in order to achieve these high accuracies. However, minimal work has been done on the open loop program track phase. Furthermore, no evidence was found of any optimization efforts for this portion of the maneuver. Therefore, this chapter will focus on the modeling and optimization of large program track maneuvers.

	Field of View (degrees)		Maximum Antenna Pointing Error (degrees)					
	E-W (AZ)	N-S (EL)	SSAF	SSAR	KuSAF	KuSAR	KaSAF	KaSAR
PT, LEO	+/-10.5° Conical		-----	-----	-----	-----	0.105°	0.102°
PT PEFOV	+/-22.5°	+/-31.0°	0.360°	0.360°	0.155°	0.155°	0.114°	0.114°
AT PEFOV	+/-22.5°	+/-31.0°	-----	-----	0.087°	0.061°	0.073°	0.045°
Nominal Antenna Beamwidth			~2.1°	~2.0°	~0.31°	~0.28°	~0.18°	~0.17°
PEFOV	Primary Elliptical Field of View							

Table 2. LEO and Primary FOV Pointing for TDRS H,I,J (from [6])

Six minutes prior to service start time, WSC computes the slew maneuver and uploads it to the satellite as coefficients for a third-order polynomial to control gimbal position [1, 8, 10]. TDRS uses a gimbal drive assembly to follow the desired polynomial [10]. While the program track is considered an open loop input, it does use a resolver signal to provide position feedback [1]. A rate command derived from the position profile is directly applied to the stepper motor and an estimated torque command is fed forward to the system as well [1]. The gimbal drive assembly uses a two-phase permanent magnetic stepper motor [10]. The motor has a 1.5° step size, which is stepped down by a harmonic drive speed reducer with a reduction ratio of 200:1 [10]. This yields a resolution of 0.0075°, small enough to ensure the minimum 0.03° pointing is met during autotrack. The program track slew consists of three segments: a five second acceleration, a constant rate coast, and a five second deceleration [10]. For maneuvers where one slew axis is larger than the other, the constant slew rate is adjusted so that the slews end concurrently [11].

B. TDRS MODEL

A general multibody model was developed in order to simulate the dynamics of TDRS and SA antenna during slew maneuvers. However, this process requires masses, inertias, and geometric dimensions in order to properly model the spacecraft. Unfortunately, even though effort was made to obtain the necessary data from Boeing, the release of this data was not approved in time for this thesis. Nevertheless, assumptions were made based on open source TDRS data, TDRS drawings, and scaled representations from a laboratory testbed [23].

The total mass of TDRS-K is 3,454 kg at launch, including fuel [24]. An estimated operation mass of 3,310 was assumed for fuel expenditure for geostationary orbit raising and circularization. This was allocated as 3,200 kg for the spacecraft body, including bus, solar panels, Space-Ground Link Antenna, forward and rear omni TT&C antennas, and the Western SA antenna, and Eastern SA antenna deployment boom. These systems were combined into one rigid body and considered body one of the system. The gimbal assembly from azimuth gimbal to elevation gimbal was allocated 10 kg and

considered body two. Lastly, the SA antenna itself, including the payload electronics and wiring in the SA compartment (SAC) were allocated 100 kg and considered body 3. A coordinate system was assigned in order to align the gimbal axes of TDRS with the gimbal axes of the general model. The x-axis was aligned with nadir, the y-axis was aligned with the North vector, and the z-axis was aligned to the positive velocity or East vector. This configuration can be seen in Figure 35.

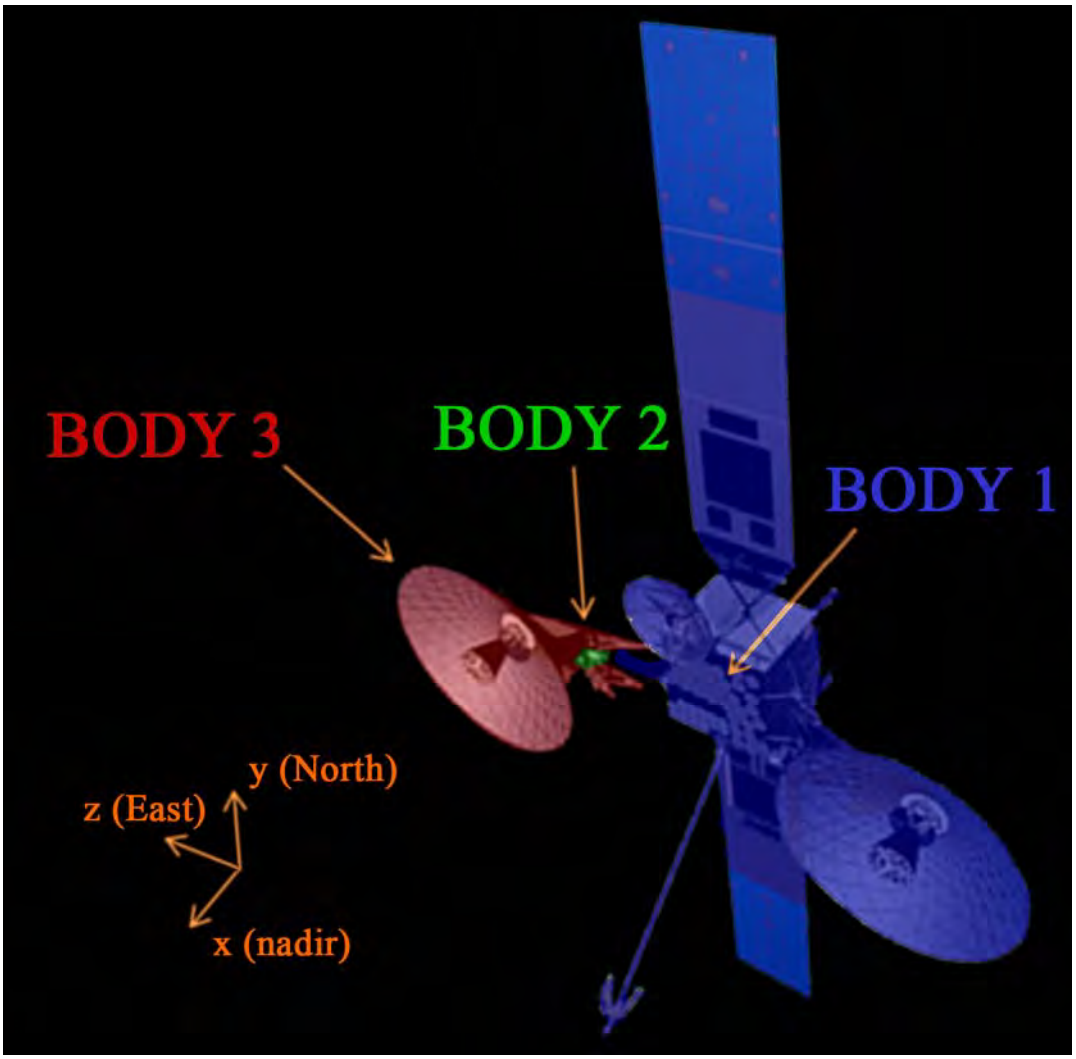


Figure 35. TDRS Model Configuration (after [4])

Inertia estimates for the body were made by assuming a constant density of the bus and calculating the moments of inertia of a 3x3 meter box. Mass was then distributed

unevenly to account for the solar arrays, Western SA antenna, Space-Ground Link antenna. Then the inertia was shifted slightly toward the Western SA antenna in order to account for CoM of the amalgamated system. This resulted in the following inertia matrix estimate:

$$I_1 = \begin{bmatrix} 5000 & -50 & -250 \\ -50 & 5000 & -100 \\ -250 & -100 & 4800 \end{bmatrix} kgm^2$$

The distance from the CoM of body one to the azimuth gimbal (joint one) was estimated.

$$r_{11} = \begin{Bmatrix} 0.5 \\ 0.1 \\ 5.0 \end{Bmatrix} m$$

The azimuth gimbal was aligned to the positive y-axis. Any outboard (East) angles measure negative and inboard angles (West) measure positive. The distance from the CoM of body two to the azimuth gimbal was estimated.

$$r_{21} = \begin{Bmatrix} -0.1 \\ 0.0 \\ -0.5 \end{Bmatrix} m$$

Body two, which simulates the gimbal assembly system was estimated as a 1 meter by .25 meter box, and small off axis inertia components added as necessary. The relative size and mass of this body are negligible compared to bodies one and three, and the system could arguably be modeled as a two-body system with a two-axis gimbal. However, this system was maintained in a three-body configuration to increase accuracy

and align the system with the validation and verification work already performed on a three-link azimuth-elevation system.

$$I_2 = \begin{bmatrix} 0.885 & -0.05 & -0.01 \\ -0.05 & 0.885 & -0.02 \\ -0.01 & -0.02 & 0.104 \end{bmatrix} kgm^2$$

The distance between the CoM of body two and the elevation gimbal (joint two) were estimated, as well as the distance between the CoM of body three and the elevation gimbal.

$$r_{22} = \begin{Bmatrix} 0.1 \\ 0.0 \\ 0.5 \end{Bmatrix} m$$

$$r_{32} = \begin{Bmatrix} -0.2 \\ 0.0 \\ -0.2 \end{Bmatrix} m$$

Lastly, the inertia of the SA antenna was estimated. This started by scaling the mass and dimension of the testbed antenna, then adding mass as necessary to simulate the SAC and the launch support structure. During launch, a truss assembly connects the stowed SA antenna to the body in order to alleviate launch stresses. During antenna deployment, this structure is separated from the spacecraft body and remains connected to the SAC. This process resulted in the inertia estimate in Figure 42.

$$I_3 = \begin{bmatrix} 180.0 & 8.0 & 10.0 \\ 8.0 & 205.0 & 0.5 \\ 10.0 & 0.5 & 215.0 \end{bmatrix} kgm^2$$

This data was input into the general model in order to generate a TDRS dynamics model. However, more assumptions were made to the model as a whole to simplify the scope and complexity of the problem to this thesis as a feasibility study. The validation and verification steps taken in Section B were vital for justifying these assumptions.

First, the gimbal angles about each joint were considered the azimuth and elevation pointing angles necessary to point the antenna at the user spacecraft. However, as the spacecraft body rotates during a slew, the spacecraft angle changes the necessary pointing angles. On TDRS, the spacecraft body rotation and attitude determination and control system (ADCS) coordinates with the SA controller to account for this deviation. In order to scope the problem, the ADCS was not modeled. Instead, the spacecraft is assumed to maintain nadir pointing with enough accuracy to nullify the SA error. In order to simulate this, a maneuver was performed using the maximum gimbal rate (0.225°/s) and the spacecraft allowed to drift. Maximum spacecraft rates were measured. These rates were taken as limits in the antenna slew optimization. It is assumed that if the optimal control maneuver can adhere to these constraints, then the TDRS control system can minimize the rotational impact of the maneuver to an acceptable margin in the same way as is done for a conventional slew.

Next, the general model to this point has assumed a body one frame rotating directly about the inertial axis. In other words, if the SA antenna is stationary, there is no rotation about body one relative to the inertial axis. This essentially indicates an inertial axis aligned to the orbital frame. However, the orbital frame is rotating about the Earth at 360° per sidereal day. The Earth is subsequently orbiting around the sun, which is moving about space, etc. Coupled motion is induced by each subsequent rotation about an ideal true inertial frame. However, as scope is increased from body frame to orbital frame, to Earth fixed frame, etc., the impact of each rotation becomes smaller and smaller to a point where the effects are minimal. For most Earth orbiting satellites, this point is

considered the Earth Centered Inertial frame. A frame is fixed to the center of the earth and aligned to the First Point of Aries, or the Vernal Equinox. If this is chosen as the inertial frame, the orbital frame rotates about it at a rate of 0.0042 °/s. In effect, this adds one more rotation in order to go from the inertial frame to the orbital frame to the body frame. However, since this rotation is small relative to the timeframe of the desired slews, it was assumed that the impact of this rotation was negligible and assumed zero.

As described in Chapter I, the flexible modes of the spacecraft and fuel sloshing was not modeled. It was also determined that friction in the gimbal could be ignored for the scope of this thesis. Since gimbal torque limits were unknown, estimated maximums and minimums had to be developed. In order to do this, the acceleration of the antenna was analyzed. Accelerations from zero to max gimbal rate were observed to vary from approximately 1.5 seconds to 5 seconds depending on the source [10, 11]. The differences between these two estimates may be due to different generation of spacecraft, or scope of the explanation (i.e., the five second acceleration time may include other delays). In order to maintain conservatism, a five second acceleration time to maximum rate was assumed. The model was generated and torque varied in order to gain insight into the response. It was determined that approximately 0.2 Nm of torque resulted in the proper five second acceleration from 0°/s to 0.225°/s. Friction torque was then calculated based on a viscous friction coefficient of $5e^{-3}$ Nms/rad [10]. Using this friction coefficient and the peak gimbal rate, a friction torque of $1.96e^{-5}$ Nm was calculated. Although during optimization, the gimbal rate was allowed to reach beyond 0.225°/s, the resulting friction torque was always two or more orders of magnitude less than the gimbal torque. Therefore, the friction was assumed negligible for the scope of this thesis.

C. TDRS MODEL VALIDATION AND VERIFICATION

In order to ensure basic functionality and accuracy of the TDRS model, as well as to justify the assumptions in Section B, validation and verification was performed on the Newton-Euler simulation using the SimMechanics model.

Three tests were run to ensure functionality and accuracy of the dynamics. First, an initial rate of 0.225 was induced in each gimbal and the system allowed to drift for

120 seconds. The gimbal responses, body rotation rates, and body rotation angles were compared between both models.

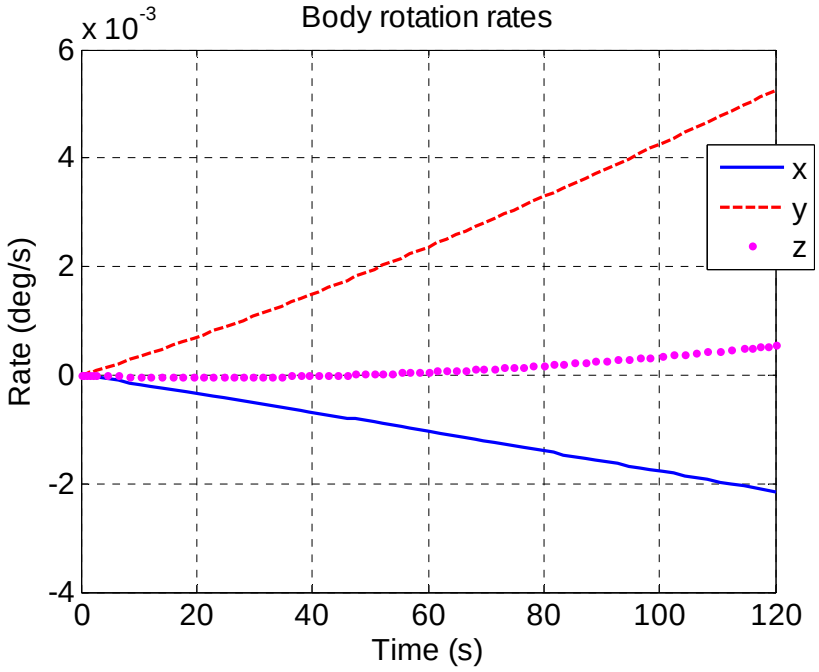


Figure 36. TDRS Newton-Euler Simulation Spacecraft Body Rates for V&V Test 1

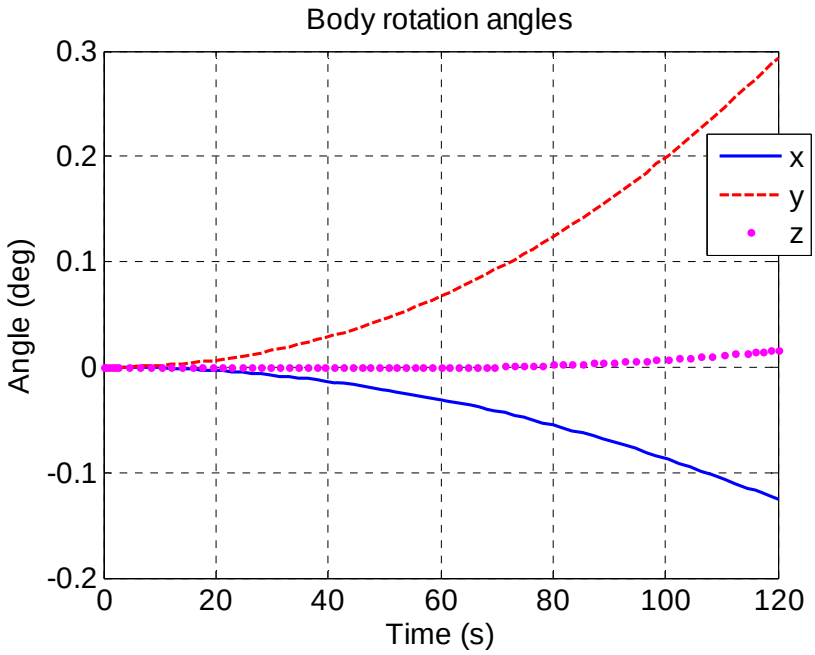


Figure 37. TDRS Newton-Euler Spacecraft Body Angles for V&V Test 1

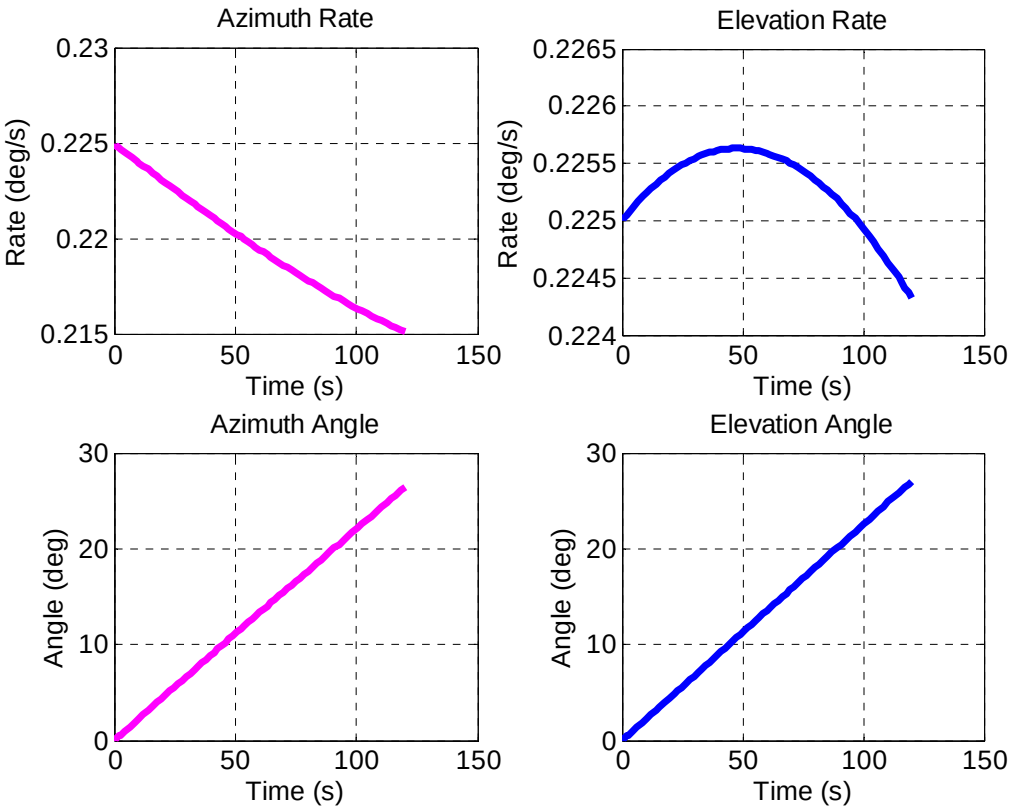


Figure 38. TDRS Newton-Euler Gimbal Response for V&V Test 1

The same system was then modeled in SimMechanics, and used to verify the MATLAB response.

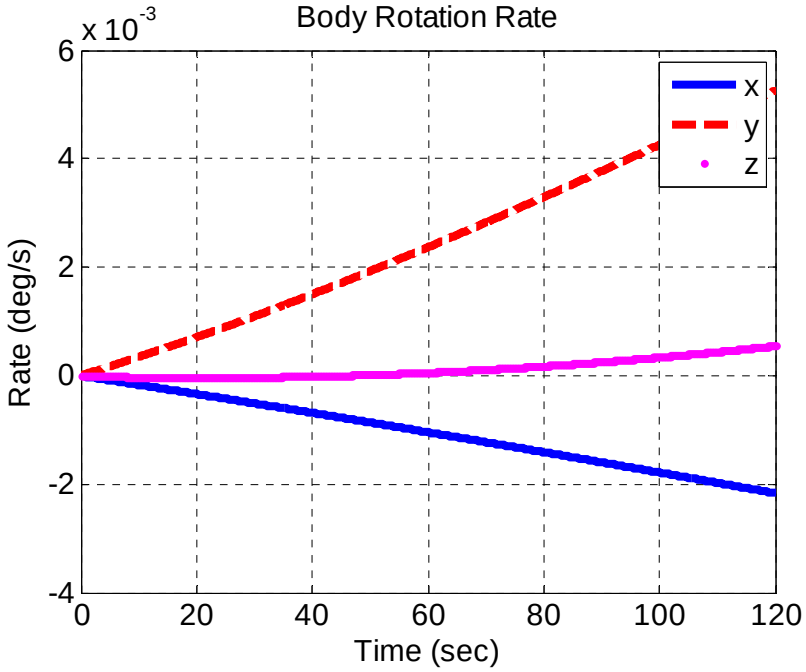


Figure 39. TDRS SimMechanics Spacecraft Body Rates for V&V Test 1

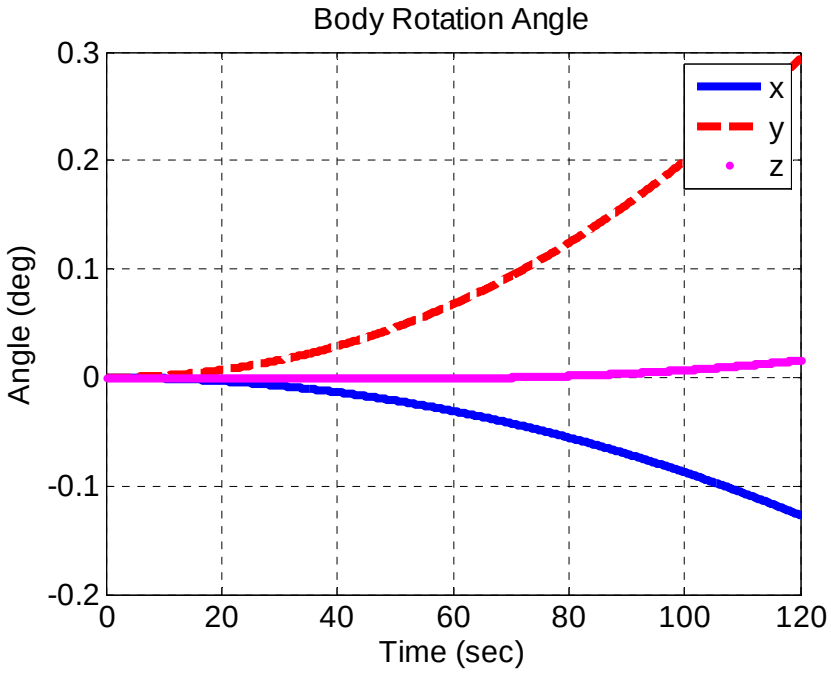


Figure 40. TDRS SimMechanics Spacecraft Body Angles for V&V Test 1

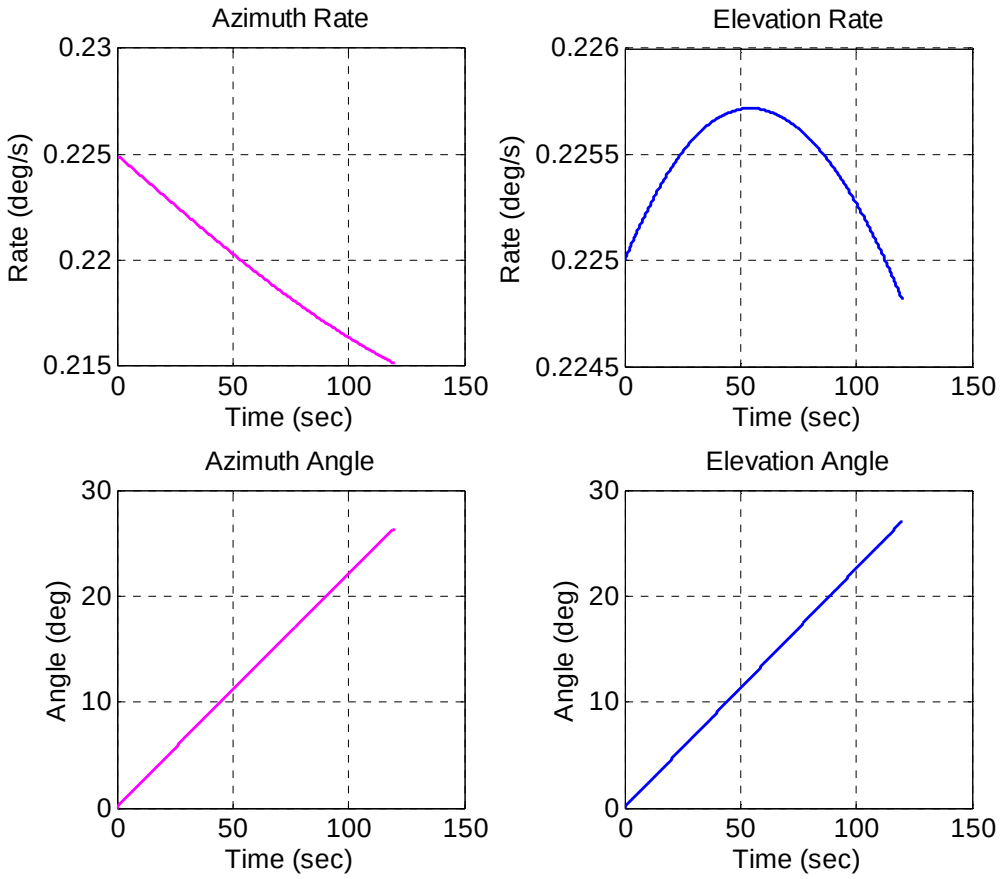


Figure 41. TDRS SimMechanics Gimbal Response for V&V Test 1

While there are slight deviations of the response, namely with respect to elevation gimbal response, the deviations are on the magnitude of $1e^{-4}$ for angle and rate. Since these responses will be dynamically driven and maintained by the torquers for the optimal simulations, these errors were considered negligible. The more important response was the spacecraft body angular rate, which matched very closely. This was vital to the assumption regarding the TDRS ADCS.

In a second test, a constant torque of 0.1 Nm was applied to each gimbal from the rest position and the system response measured for 20 seconds. The response can be seen in Figure 42 through Figure 44.

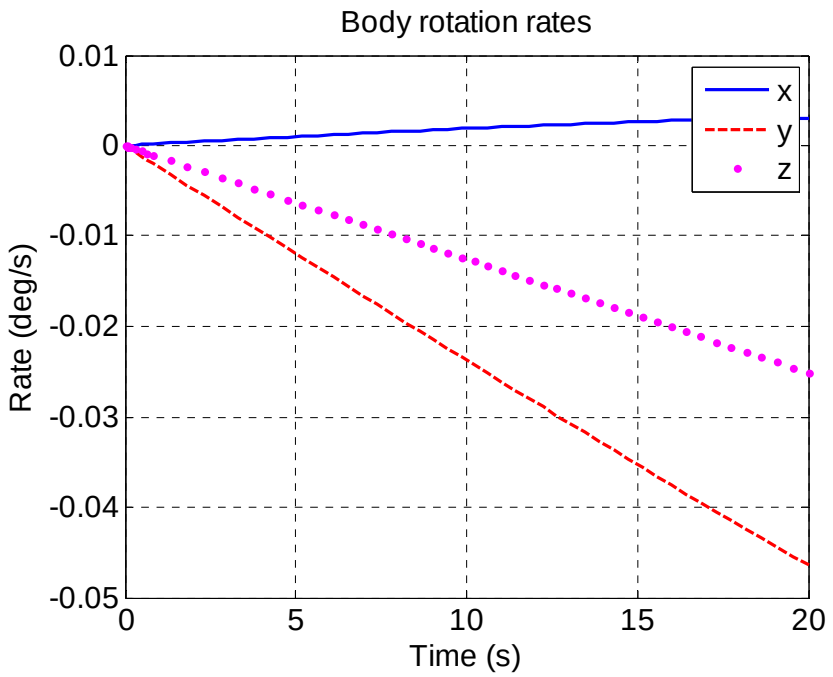


Figure 42. TDRS Newton-Euler Spacecraft Body Rate for V&V Test 2

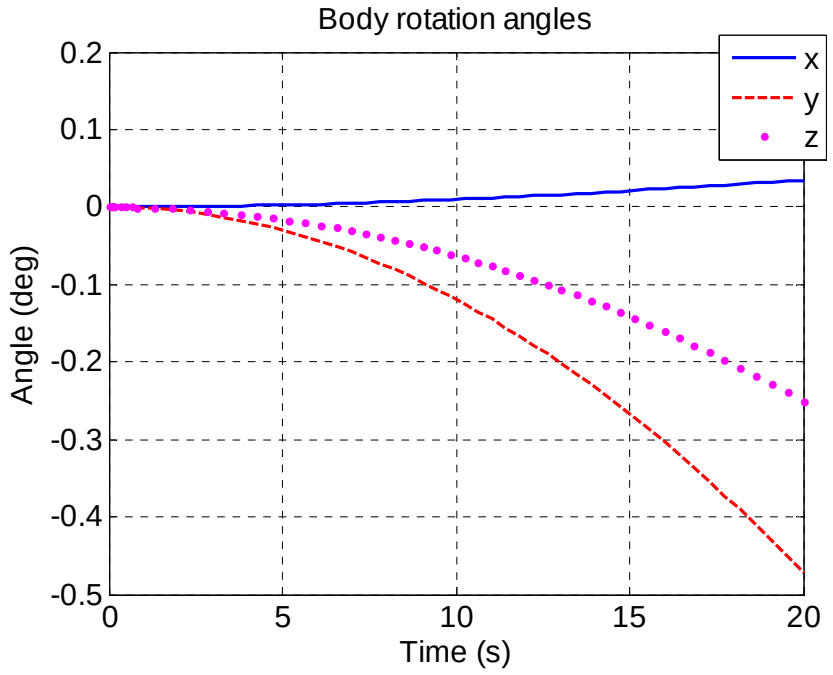


Figure 43. TDRS Newton-Euler Spacecraft Body angle for V&V Test 2

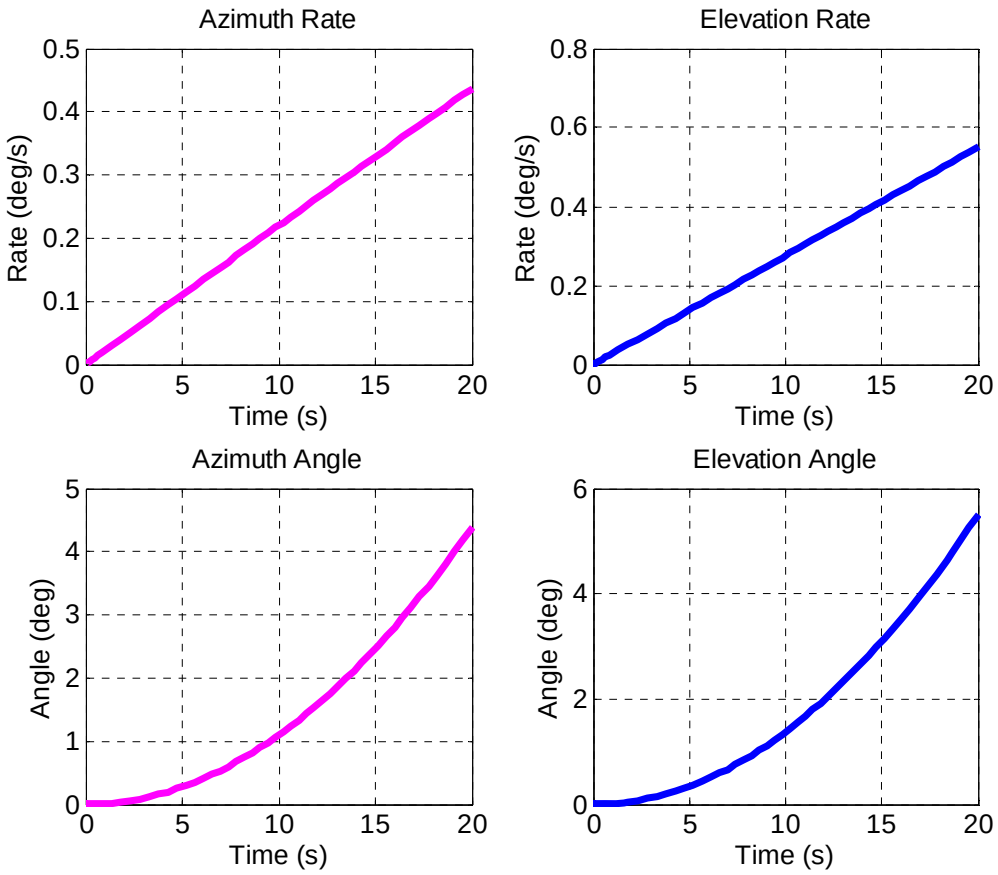


Figure 44. TDRS Newton-Euler Gimbal Response for V&V Test 2

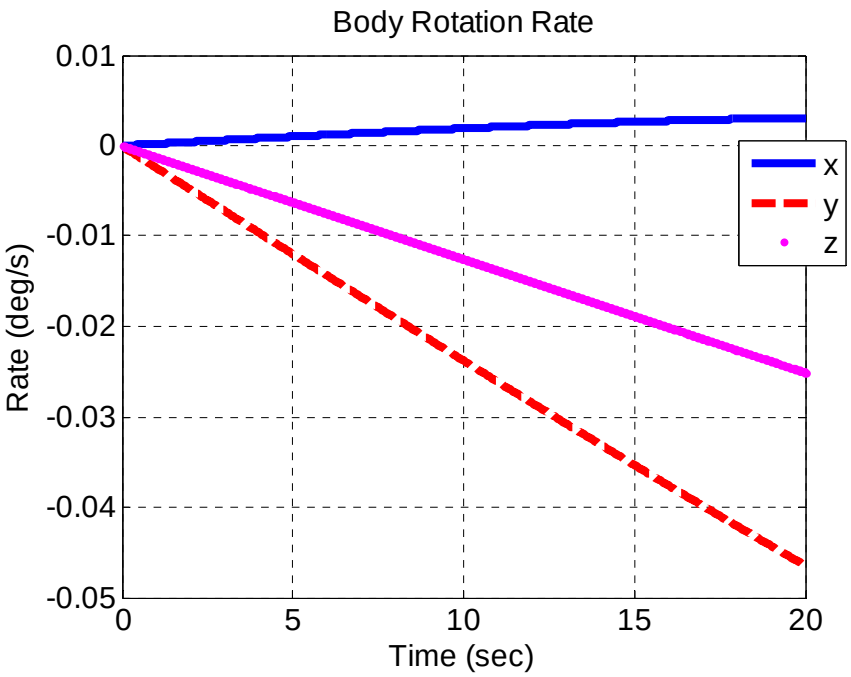


Figure 45. TDRS SimMechanics Spacecraft Body Rate for V&V Test 2

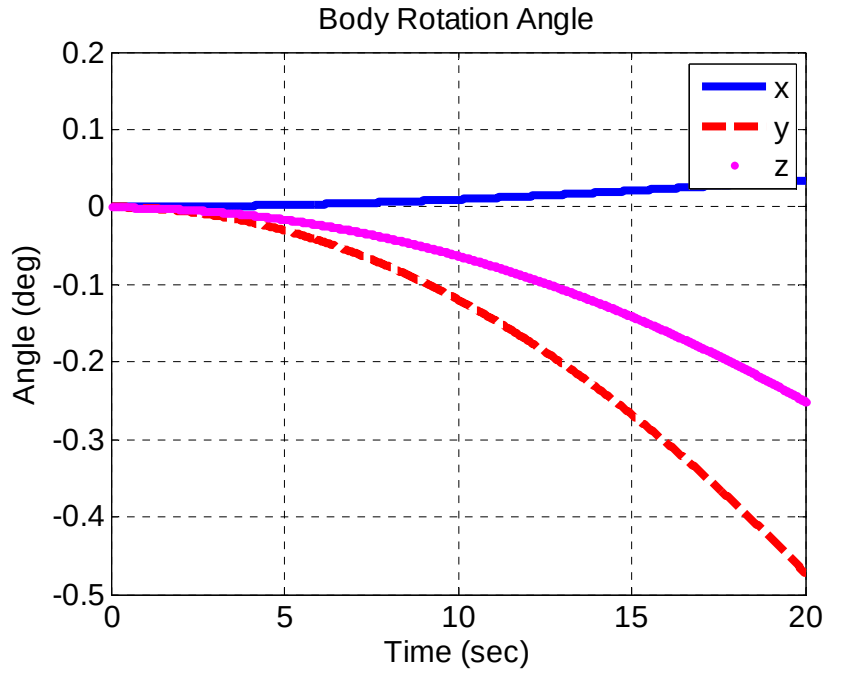


Figure 46. TDRS SimMechanics Spacecraft Body Angle for V&V Test 2

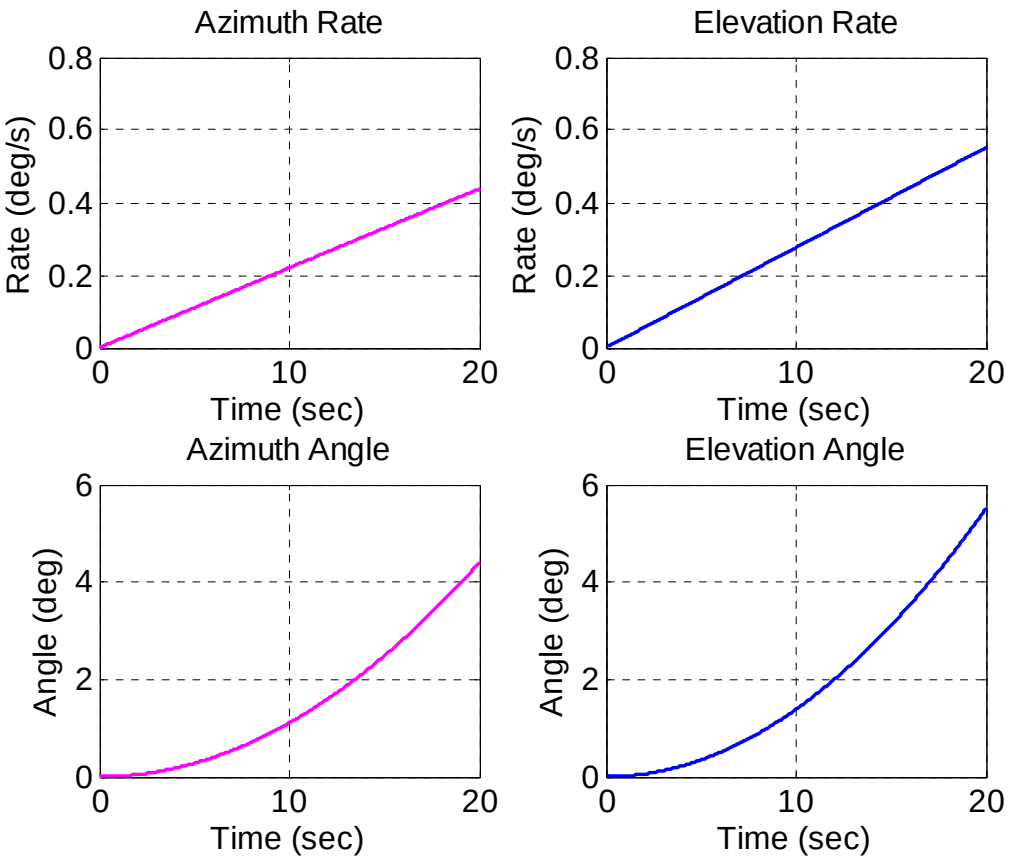


Figure 47. TDRS SimMechanics Gimbal Response for V&V Test 2

As expected, under driven conditions, the responses correspond much better than under free rotation.

A third validation and verification test performed to ensure dynamic accuracy was a simulated antenna slew. This test was also utilized to obtain bounds on the disturbance induced on the spacecraft body for a standard antenna slew. A 0.2 Nm torque was applied to a resting system for five seconds. At five seconds, the torque was turned off and the system allowed to drift for a remaining three minutes. This test not only validated a change in torque, but also provided the maximum body rates required for optimal control analysis. Results can be seen in Figures 48 through 50.

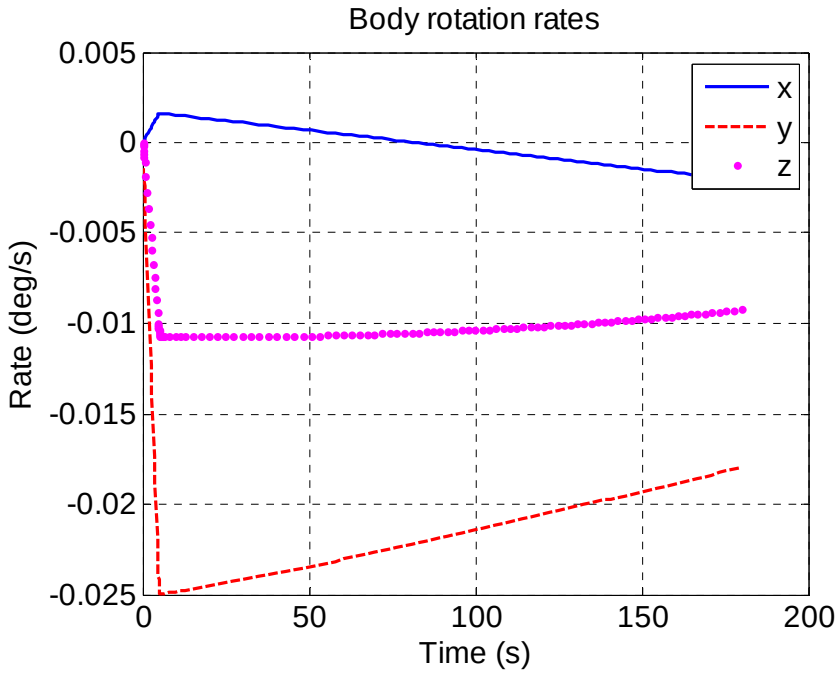


Figure 48. TDRS Newton-Euler Spacecraft Body Rate for V&V Test 3

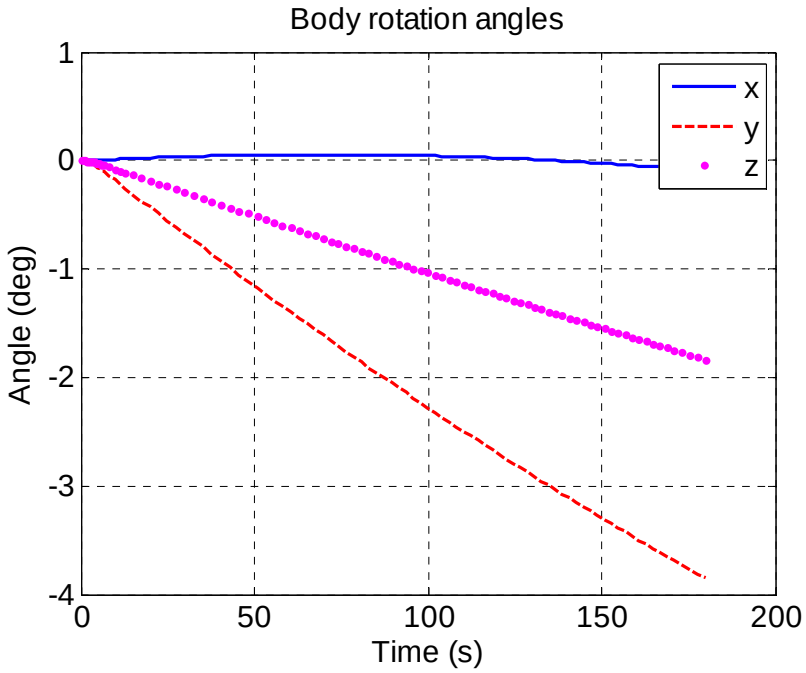


Figure 49. TDRS Newton-Euler Spacecraft Body Angle for V&V Test 3

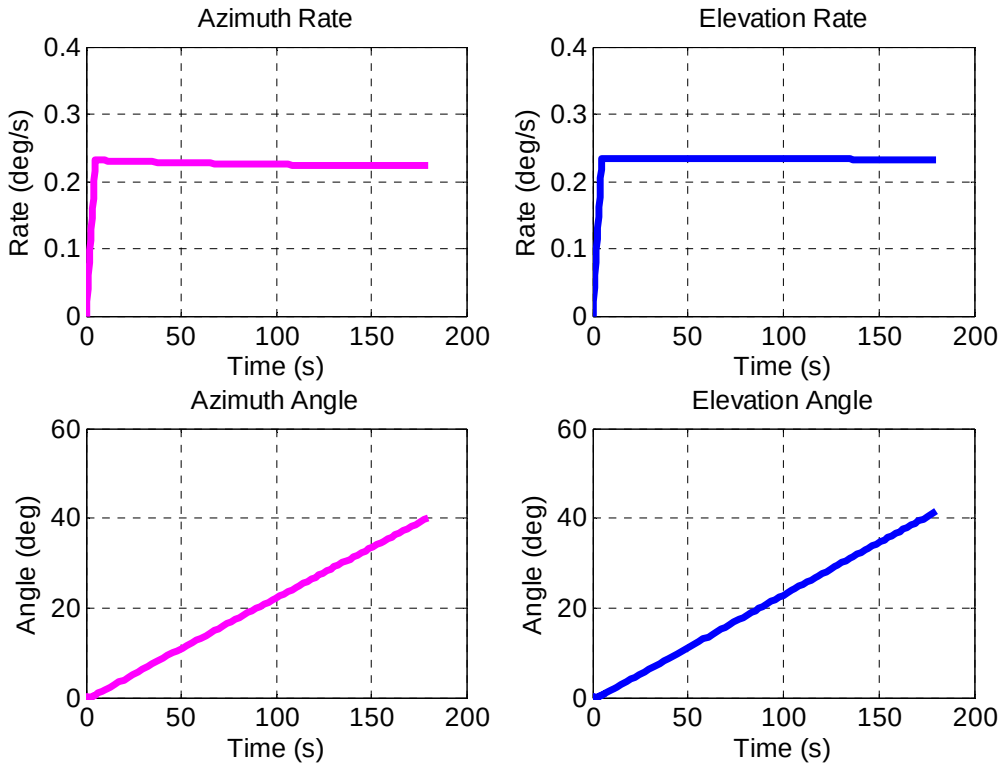


Figure 50. TDRS Newton-Euler Gimbal Response for V&V Test 3

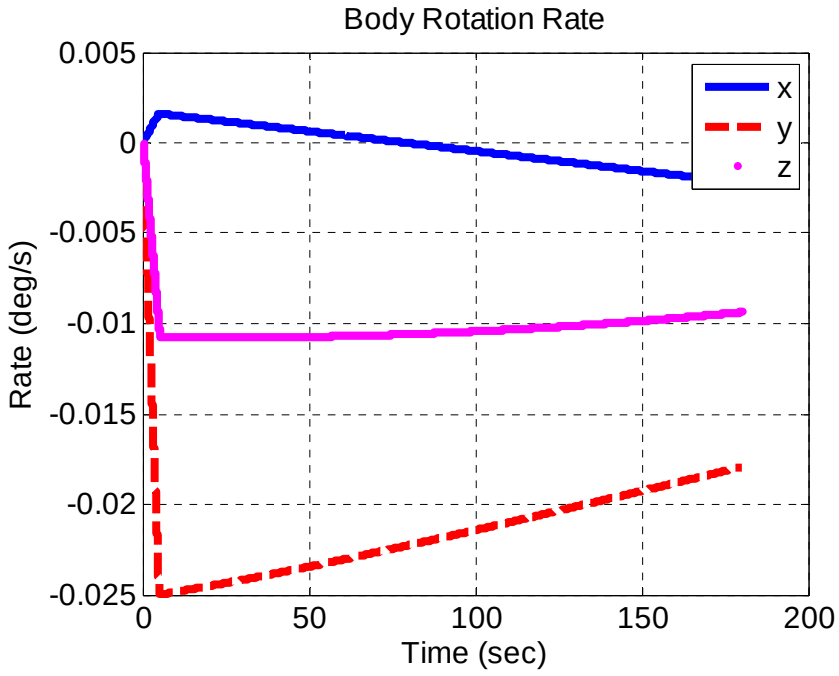


Figure 51. TDRS SimMechanics Spacecraft Body Rate for V&V Test 3

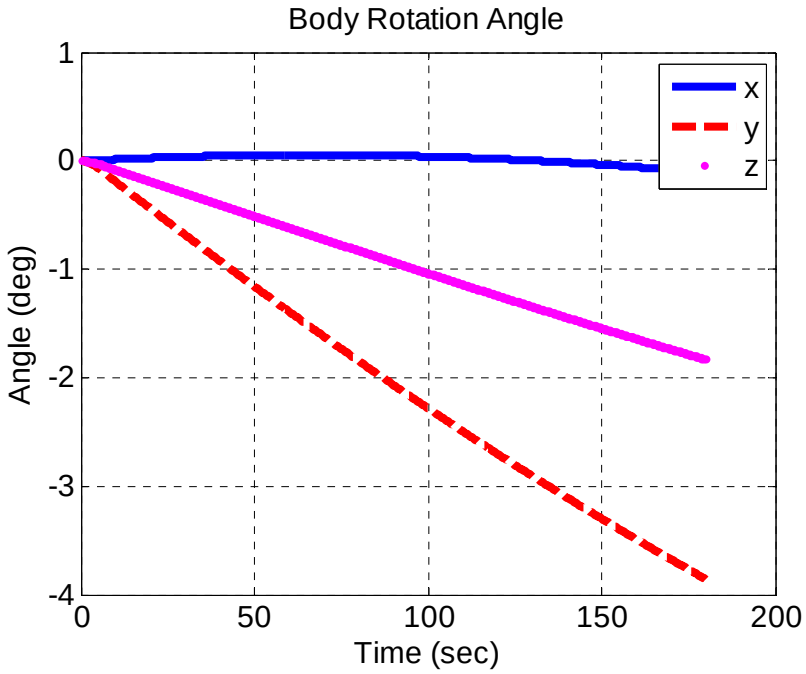


Figure 52. TDRS SimMechanics Spacecraft Body Angle for V&V Test 3

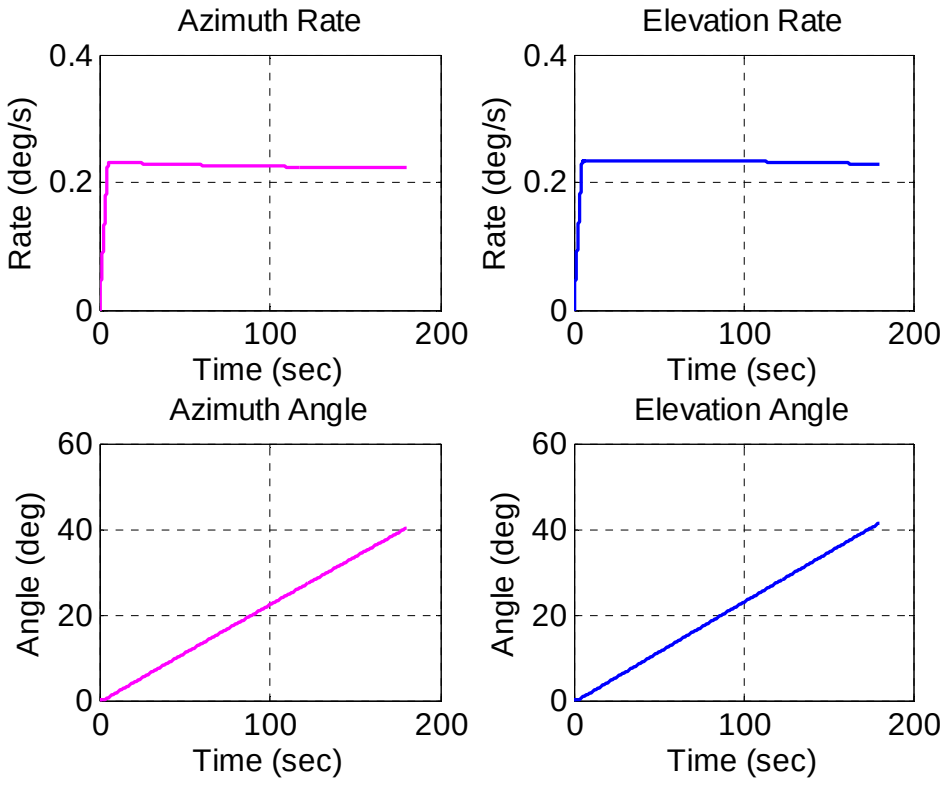


Figure 53. TDRS SimMechanics Gimbal Response for V&V Test 3

The maximum spacecraft body rotational rates were measured and recorded for use as optimization bounds:

$$\omega_{1x\max} = 0.0022 \, {}^{\circ}/_{s^2}$$
$$\omega_{1y\max} = 0.0250 \, {}^{\circ}/_{s^2}$$
$$\omega_{1z\max} = 0.0126 \, {}^{\circ}/_{s^2}.$$

The final validation and verification test was to ensure the geostationary rate of the orbital frame had negligible impact on the system and the orbital frame could be considered inertial for the limited timeframe of the antenna slew. The same five second acceleration maneuver above was performed, except the spacecraft was initialized at 0.0042°/s about the y-axis. The system was then propagated for 180 seconds. The results can be seen in Figure 54 through Figure 56, where the plots depict motion relative to the orbital frame (i.e., 0.0042°/s was subtracted from the spacecraft body rate about the y-axis).

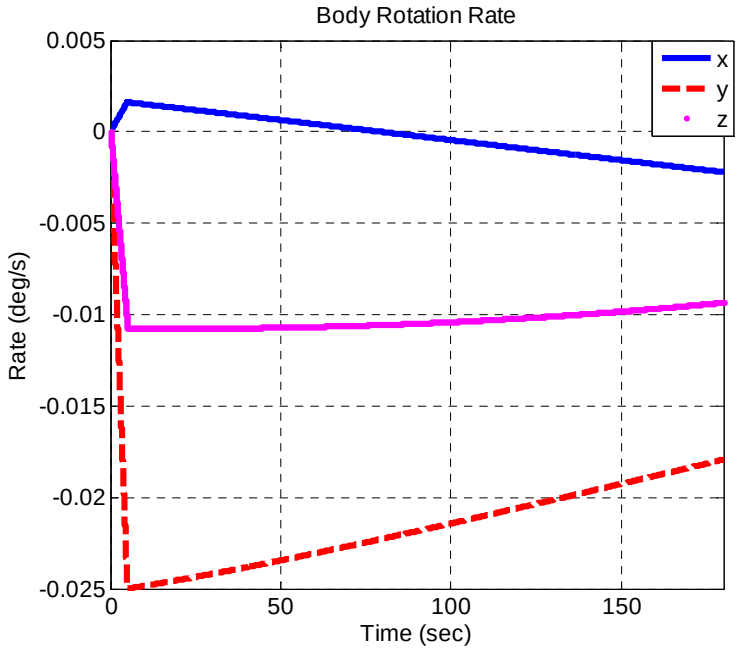


Figure 54. TDRS SimMechanics Spacecraft Body Rate for V&V Test 4

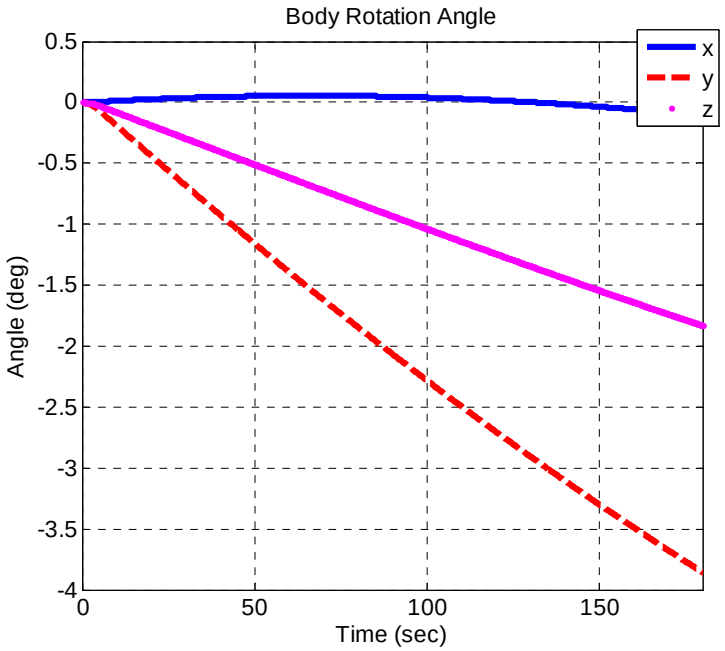
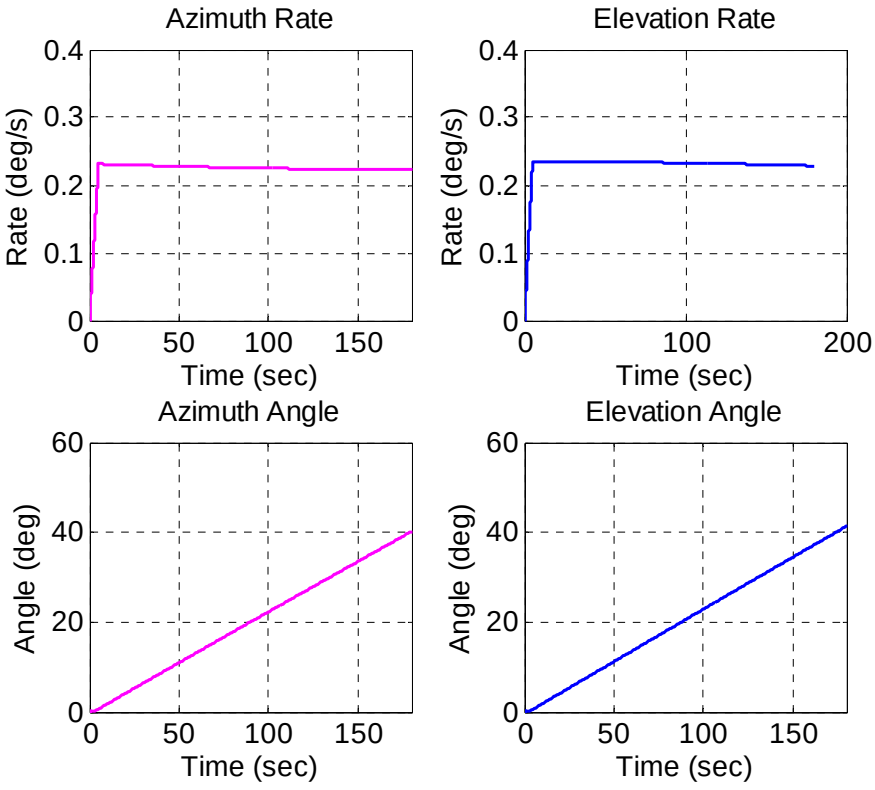


Figure 55. TDRS SimMechanics Spacecraft Body Angle for V&V Test 4



v

Figure 56. TDRS SimMechanics Gimbal Response for V&V Test 4

It can be easily seen that the differences between the tests with and without the orbital frame motion are negligible. The rate responses are similar enough to discount the impact of non-inertial dynamics and assume the orbital frame to be inertial. It should be noted that the position plot response for the y-axis does not have the additional rotation rate removed. Therefore, it is 0.756° off, which is the expected amount following 180 seconds of an additional 0.0042°/s.

With the model validated, and useful bounds for body rate and torque gleaned from the simulations, the model is implemented next in DIDO so that the slew maneuver can be optimized.

D. TDRS OPTIMIZATION IN DIDO

Before DIDO could be used to determine optimal slew paths, the initial and final conditions for the maneuvers had to be established. The azimuth, elevation, azimuth rate, and elevation rate of a departing spacecraft were used as the initial conditions for the problem. The azimuth, elevation, azimuth rate, and elevation rate of the future customer were set as the target end conditions. These data points, however, were not available from the literature. In order to generate them, the Systems Tool Kit (STK) software was used. STK is a powerful orbit propagator and could easily provide azimuth and elevation data required for this analysis.

First, five spacecraft were placed into an orbital model in STK: TDRS, the International Space Station, National Oceanic and Atmospheric Administration (NOAA) 15, Worldview 2, and a fictional MEO spacecraft. Orbital data for these spacecraft can be found in Table 3. A sixth spacecraft was added, a geostationary spacecraft at 72° outboard in azimuth, was added to test the EFOV, but did not require STK simulation due to its known position and rate relative to TDRS. The assumption was also made that both spacecraft were in the same inclination orbit and at the ascending node during the maneuver (required to nullify azimuth rate since TDRS is geosynchronous, not geostationary).

	TDRS	ISS	NOAA-15	Worldview-2	MEO
Semi-Major Axis (km)	42166.41	6787.36	7173.81	7150.06	16678.14
Eccentricity	0.001014	0.001563	0.002720	0.001403	0.500000
Inclination (deg)	2.730	51.595	98.708	98.504	20.000
[1]RAAN (deg)	64.113	330.089	330.583	58.213	0
[2]Arg of Perigee (deg)	187.147	98.819	65.831	74.292	360
[1]Right Ascension of Ascending Node					
[2]Argument of Perigee					

Table 3. Target Satellite Orbital Elements

Six scenarios were identified over the course of 24 hours wherein one spacecraft was losing view of TDRS, while another was coming into view. These included points where the MEO spacecraft was at apogee in order to simulate larger slews. Azimuth and elevation data were calculated for each of these times and azimuth and elevation rates determined by simple average of last known angles over time. This resulted in the scenarios listed in Table 5.

	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6
Starting Azimuth (deg)	-5.67	-9.07	7.52	-29.42	-8.84	-72.00
Starting Azimuth rate (deg/s)	-0.0006	0.0065	-0.0070	0.0002	0.0017	0.0000
Ending Azimuth (deg)	-8.27	-3.83	8.24	5.19	-24.19	8.24
Ending Azimuth rate (deg/s)	0.0054	-0.0063	-0.0053	0.0035	-0.0007	-0.0053
Starting Elevation (deg)	-7.35	-2.05	6.12	-7.21	-0.86	0.00
Starting Elevation rate (deg/s)	0.0005	0.0030	0.0071	-0.0010	-0.0077	0.0000
Ending Elevation (deg)	-5.16	8.92	5.16	6.97	1.42	5.16
Ending Elevation rate (deg/s)	-0.0086	-0.0038	0.0085	0.0011	-0.0012	0.0085
Scenario 1: ISS to NOAA-15						
Scenario 2: MEO to Worldview-2						
Scenario 3: NOAA-15 to Worldview-2						
Scenario 4: MEO to ISS						
Scenario 5: ISS to MEO						
Scenario 6: GEO to Worldview-2 (Same final conditions as Scenario 3)						

Table 4. Boundary Conditions for Slew Optimization

The data points given in Table 4 were then utilized in the following problem definition:

$$x = \{\omega_1 \;\; \dot{\theta}_1 \;\; \dot{\theta}_2 \;\; \theta_1 \;\; \theta_2\} \qquad u = \{T_{G1} \;\; T_{G2}\}$$

$$\textit{Minimize} \qquad J = \alpha t_f + \beta \int_{t_0}^{t_f} (\mathrm{T_{G1}}^2 + \mathrm{T_{G2}}^2) dt$$

$$\textit{Subject to}:$$

$$\dot{\omega}_1 = \left[(A - RU^{-1}S)^{-1}(T' - RU^{-1}F') \right]_1$$

$$\ddot{\theta}_1 = \left[(A - RU^{-1}S)^{-1}(T' - RU^{-1}F') \right]_2$$

$$\ddot{\theta}_2 = \left[(A - RU^{-1}S)^{-1}(T' - RU^{-1}F') \right]_3$$

$$\left(\omega_1(t_0), \dot{\theta}_1(t_0), \dot{\theta}_2(t_0), \theta_1(t_0), \theta_2(t_0) \right) = (0, \dot{A}z_0, \dot{E}l_0, Az_0, El_0)$$

$$(\dot{\theta}_1(t_f), \dot{\theta}_2(t_f), \theta_1(t_f), \theta_2(t_f)) = (\dot{A}z_f, \dot{E}l_f, Az_f, El_f)$$

$$-1 \le \dot{\theta}_1(t) \le 1 \quad \text{deg/}s$$

$$-1 \le \dot{\theta}_2(t) \le 1 \quad \text{deg/}s$$

$$-72 \le \theta_1(t) \le 24 \quad \text{deg}$$

$$-32 \le \theta_2(t) \le 32 \quad \text{deg}$$

$$-0.0022 \le \omega_{1x}(t) \le 0.0022 \quad \text{deg/}s$$

$$-0.025 \le \omega_{1y}(t) \le 0.025 \quad \text{deg/}s$$

$$-0.0126 \le \omega_{1z}(t) \le 0.0126 \quad \text{deg/}s$$

$$0.2 \le T_{G1c}(t) \le 0.2 \quad Nm$$

$$0.2 \le T_{G2c}(t) \le 0.2 \quad Nm$$

Figure 57. TDRS Optimal Control Problem Definition

The bounds for the gimbal angles were taken directly from the TDRS FOV limits. Spacecraft body rate bounds and torque limits were gleaned from simulation data as explained in Section C. However, the gimbal rate limits were increased beyond 0.225°/s. If the TDRS gimbal rate limits were maintained, there would be almost no room for slew time improvement. The only possible decrease in slew time would come from the ability to slew the antenna in such a way as to assist acceleration and deceleration. However, this only leaves approximately 10 seconds of a maneuver for optimization and it proves very difficult for the system to optimize during these accelerations. Simulations were conducted to try to optimize the slew while maintaining gimbal rate limits and the improvements were negligible.

However, if it is assumed that the 0.225°/s rate limit is not a hard limit (i.e., driven by the physical maximum rate of the gimbal hardware), but is instead a soft limit (i.e., put in place to minimize rotational impact on the spacecraft), then this limit can be increased as long as the condition it is meant to alleviate is not violated. In other words, the system has a requirement to minimize rotation rate of the spacecraft to some degree. It is assumed that the gimbal limits were established to meet this requirement. If the requirement can be met in some other manner, slewing along an optimal path for example, then the rate and torque limits can be increased and the original requirement to minimize spacecraft rotation maintained. This expands the solution space of the system and provides the problem with ample resources for optimization. Max gimbal rates were set to 1°/s in order to provide a wide solution space. For almost every maneuver, this boundary was sufficient to allow the antenna to maneuver optimally. However, for larger slews, this boundary was increased to 2°/s in order to allow room to smooth out the trajectory profile, as explained in the following paragraph.

Furthermore, this problem formulation is similar to previous definitions, with one exception. The addition of weighting factors, α and β, allowed for tailoring between simulations. The weighting factors allow for different emphasis to be placed on the endpoint cost or the running cost. Each simulation was performed twice at two different weighting factors. First, α was set to 1 and β set to 0. This returned the time optimal slew. However, the control sometimes had large impulses which may cause unacceptable vibrations to the spacecraft. In order to smooth out the control profile, a second optimization was performed. For this second optimization, the minimum slew time from the first simulation was set as a final time for a boundary condition. Then α was set to zero to eliminate the endpoint cost, and β was set to one. This forced DIDO to minimize the amount of torque applied throughout the minimum-time maneuver. This resulted in a much smoother torque trajectory. However, less torque resulted in larger slew rates. While these larger slew rates did not adversely affect the output (i.e., the spacecraft body rotation limits were still met), it was still desired not to increase them unnecessarily. For one scenario, however, the gimbal rate did increase beyond the default 1°/s, so the rate

limit was set to 2°/s in order to allow sufficient solution space while maintaining a smooth torque trajectory.

Scenario 6 was also utilized to examine the ability to minimize the impact the antenna slew has on spacecraft pointing. An estimate for the time of the conventional maneuver was used as a fixed final time and the cost function adjusted to minimize spacecraft spin rates. This scenario is indicative of a case where timeliness is not a driving concern. Minimizing the spacecraft spin rates would help increase the accuracy of the other payloads as well as minimizing fuel consumption. If designed around an optimal maneuver, it is possible that the required momentum space of a spacecraft could be reduced, decreasing fuel and mass requirements.

E. RESULTS

An optimal maneuver was solved for each scenario using DIDO. Estimates were made for timeliness of conventional slews based on a maximum acceleration for five seconds to maximum rate, then maximum deceleration over five seconds to the final desired rate. The longer axis, azimuth or elevation, drove an estimated conventional maneuver time. It was assumed that the shorter axis would adjust the peak gimbal rate so that both axes would conclude the maneuver concurrently. These values were then compared to the minimal time returned by DIDO and slew time improvements were calculated. The maximum azimuth and elevation rates were also noted. The data are summarized in Table 6.

	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6
Estimated Conventoinal Time (s)	17.13	54.32	9.80	159.40	73.78	362.17
Optimal Slew time (s)	16.44	40.12	8.66	87.56	56.03	122.60
Maximum Azimuth rate (deg/s)	0.2347	0.2351	0.1576	0.4688	0.4053	1.0214
Maximum Elevation rate (deg/s)	0.1806	0.3294	0.2051	0.2786	0.0617	0.0776
% reduction in slew time	4.03	26.14	11.62	45.07	24.05	66.15

Table 5. Optimal Antenna Slew Results

The best reductions in time come from scenarios where there is a large difference in scale between azimuth and elevation, and the larger the dominant slew, the better. For

a conventional maneuver, the gimbal rates are limited in order to minimize spacecraft disturbance due to the antenna motion. However, since one axis is smaller than the other, the smaller axis has available room to slew in a non-traditional manner. This allows the smaller axis to reduce the impact of the antenna slew on the spacecraft, allowing the donimant axis to increase gimbal rate without violating spacecraft pointing. With the exception of scenario three, the dominant axis gimbal rate always exceeded the conventional limit, and in some cases, the smaller axis gimbal rate exceeded it as well. Scenario three is an exception, but still follows the same characteristics as the other optimal slews. In scenario three, the largest slew magnitude is very small, less than one degree, so the antenna doesn't have time to accelerate to the maximum gimbal rate. Instead the antenna accelerates for approximately half the maneuver, then decelerates for the other half. However, since the smaller axis doesn't need as much acceleration as the dominant axis, it still has room to assist the dominant axis. For each case, the torque for the dominant axis acts for the most part as a bang-bang maneuver, providing maximum torque to accelerate then maximum torque to decelerate, with smaller deviations during the middle duration of the slew. Alternatively, the torque for the smaller axis is changing constantly in order to maintain the spacecraft spin rates within requirements.

The full response of scenario six is provided in Figure 58 through Figure 61. The characteristics of this response are typical of each scenario; only the sizes of the individual maneuvers and the details of the torque profile vary, particularly with respect to the smaller axis. For this simulation, final time was set to 122.6, α was set to zero, and β was set to one. The particular note is the plot of the spacecraft body rates, in Figure 61. Notice that the y-axis, which is the major axis for this maneuver, pushes the limits of its boundary envelope. This was the case for every maneuver. The spacecraft rate was always the limiting factor in the maneuver. Furthermore, in some cases two axes reached the rate limit, not just the spacecraft axis aligned with the dominant antenna slew axis. This would indicate that the minor axis is performing spin minimization of the major axis to its best ability, but is limited by its own spin requirement.

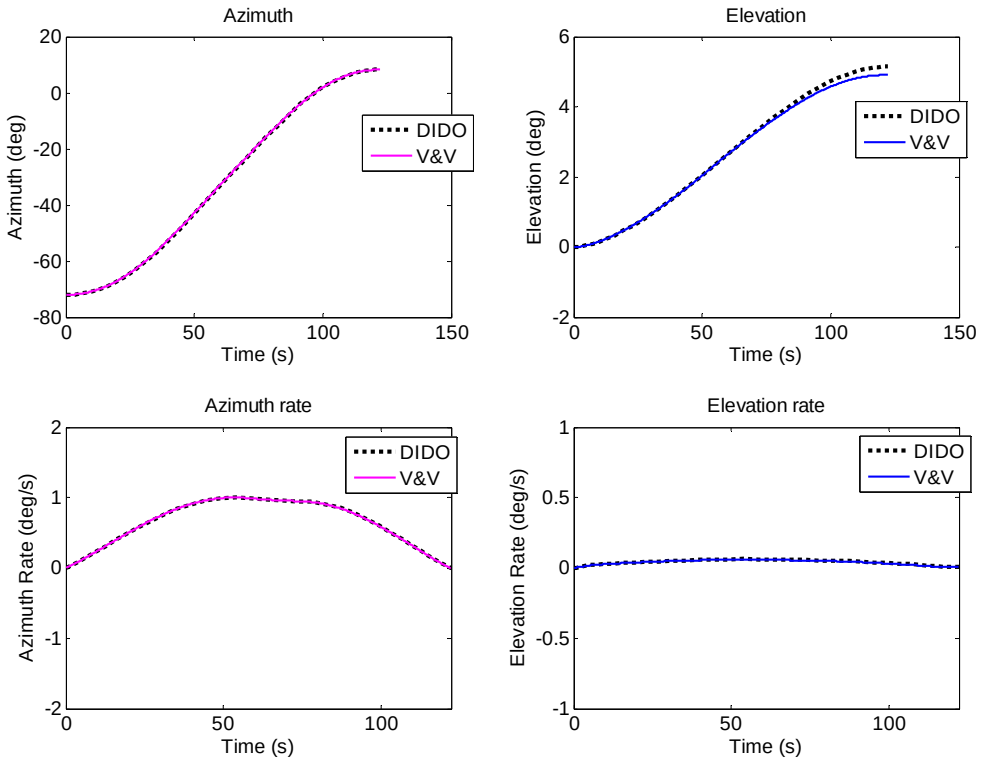


Figure 58. Scenario 6 Optimal Gimbal Response

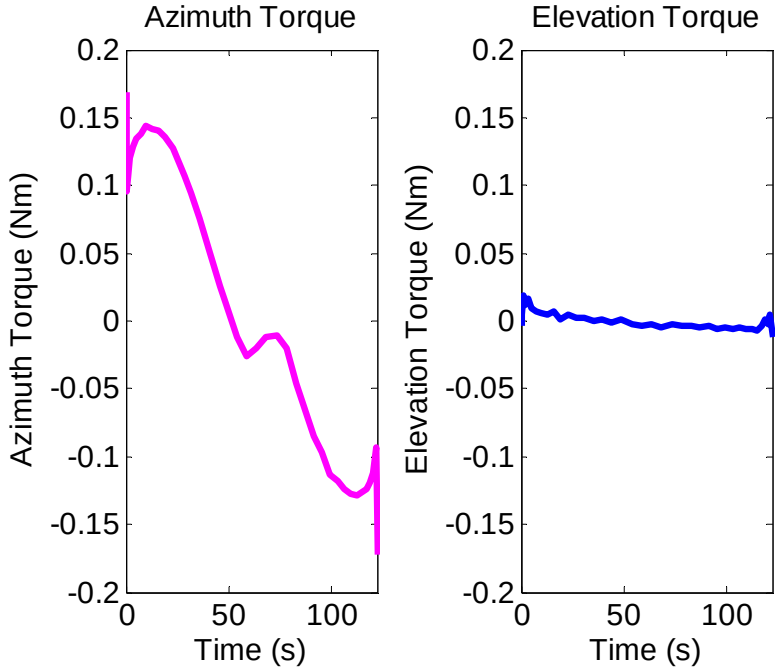


Figure 59. Scenario 6 Optimal Gimbal Torque

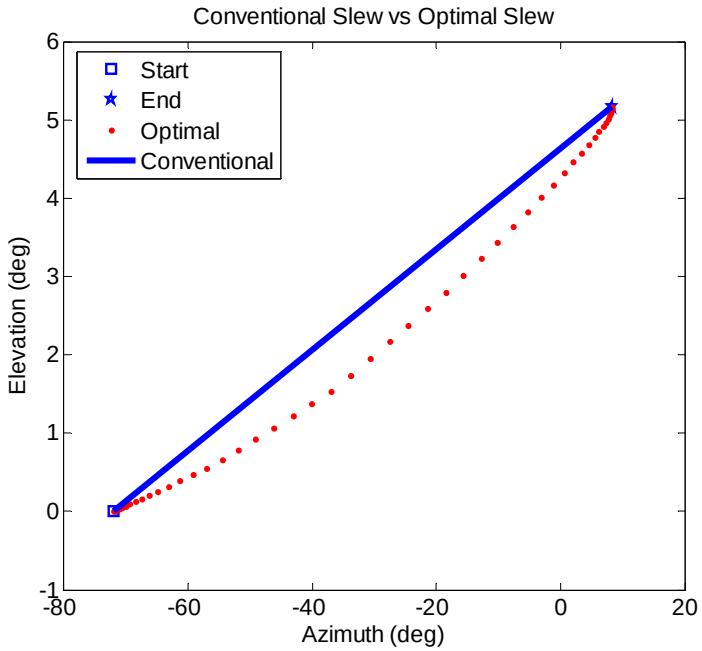


Figure 60. Combined Az-El Trajectory

Notice in Figure 60 how the optimal path drifts away from the conventional path. This illustrates how the smaller axis moves in order to generate additional capability along the dominant axis.

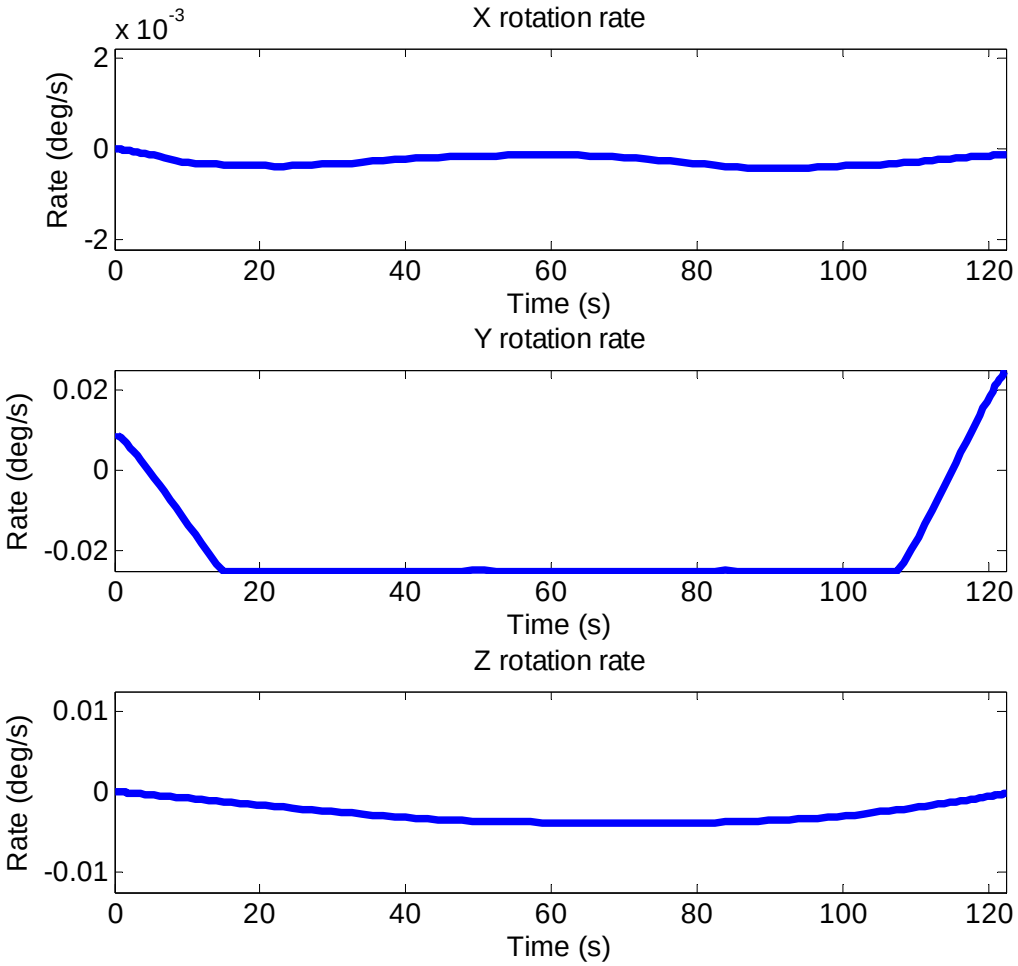


Figure 61. Scenario 6 Spacecraft Body Spin Rates

In order to test the slew's ability to minimize the induced spacecraft rate, the problem definition was altered slightly, as seen in Figure 62.

$$x = \left\{ \omega_1 \quad \dot{\theta}_1 \quad \dot{\theta}_2 \quad \theta_1 \quad \theta_2 \right\} \qquad u = \left\{ T_{G1} \quad T_{G2} \right\}$$

$$\textit{Minimize} \qquad J = \alpha t_f + \beta \int_{t_0}^{t_f} (\mathrm{T_{G1}}^2 + \mathrm{T_{G2}}^2) dt + \int_{t_0}^{t_f} (\varsigma_x \omega_{1x}{}^2 + \varsigma_y \omega_{1y}{}^2 + \varsigma_z \omega_{1z}{}^2) dt$$

$$\textit{Subject to}:$$

$$\dot{\omega}_1 = \left[(A - RU^{-1}S)^{-1} (T' - RU^{-1}F') \right]_1$$

$$\ddot{\theta}_1 = \left[(A - RU^{-1}S)^{-1} (T' - RU^{-1}F') \right]_2$$

$$\ddot{\theta}_2 = \left[(A - RU^{-1}S)^{-1} (T' - RU^{-1}F') \right]_3$$

$$\left(\omega_1(t_0), \dot{\theta}_1(t_0), \dot{\theta}_2(t_0), \theta_1(t_0), \theta_2(t_0) \right) = (0, \dot{A}z_0, \dot{E}l_0, Az_0, El_0)$$

$$(\dot{\theta}_1(t_f), \dot{\theta}_2(t_f), \theta_1(t_f), \theta_2(t_f)) = (\dot{A}z_f, \dot{E}l_f, Az_f, El_f)$$

$$t_f = \textit{Conventional Slew Time}$$

$$-1 \le \dot{\theta}_1(t) \le 1 \quad \text{deg/}s$$

$$-1 \le \dot{\theta}_2(t) \le 1 \quad \text{deg/}s$$

$$-72 \le \theta_1(t) \le 24 \quad \text{deg}$$

$$-32 \le \theta_2(t) \le 32 \quad \text{deg}$$

$$-0.0022 \le \omega_{1x}(t) \le 0.0022 \quad \text{deg/}s$$

$$-0.025 \le \omega_{1y}(t) \le 0.025 \quad \text{deg/}s$$

$$-0.0126 \le \omega_{1z}(t) \le 0.0126 \quad \text{deg/}s$$

$$0.2 \le T_{G1c}(t) \le 0.2 \quad Nm$$

$$0.2 \le T_{G2c}(t) \le 0.2 \quad Nm$$

Figure 62. TDRS Minimum Disturbance Problem Definition

Scenario six was used, and the final time was set to the estimated conventional slew time. Although the final time could be set anywhere from the conventional slew time to the optimal slew time, it was set to the conventional slew time in order to provide the problem with the maximum room for optimizatoin. Also, an additional running cost and three associated weighting factors were added to minimize spacecraft spin rates. For this simulation, α was set to zero, β was set to 0.01 in order to smooth torque without impacting the minimization of disturbances. Lastly, the disturbance weighting factor, ζ, was set to different values in order to tune on each axis. The weighting factor for the x

and z axes was set to 1 and the weighting factor for the y axis was set to 10 in order to minimize disturbances on the axis impacted most by the motion of the antenna. The optimized response can be seen in Figure 63 through Figure 65. The modulation of the control torques, as seen in Figure 64, is necessary to counteract the motion induced on the spacecraft body.

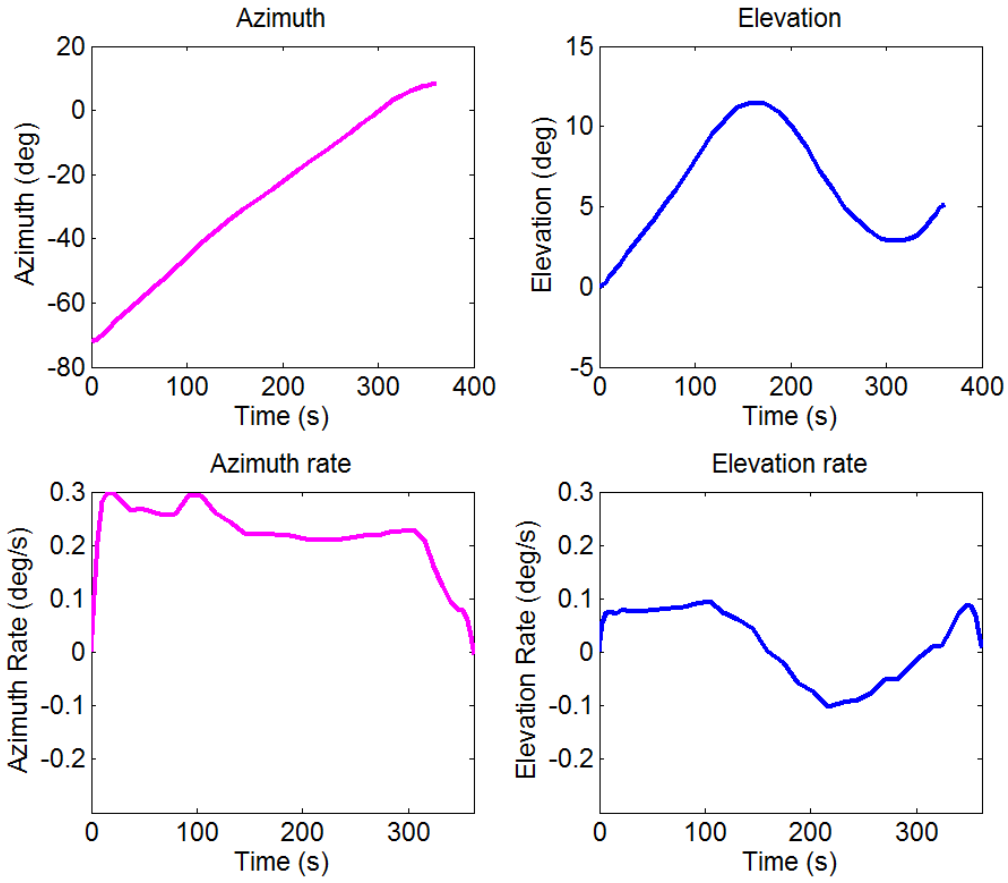


Figure 63. TDRS Scenario 6 Minimum Disturbance Gimbal Response

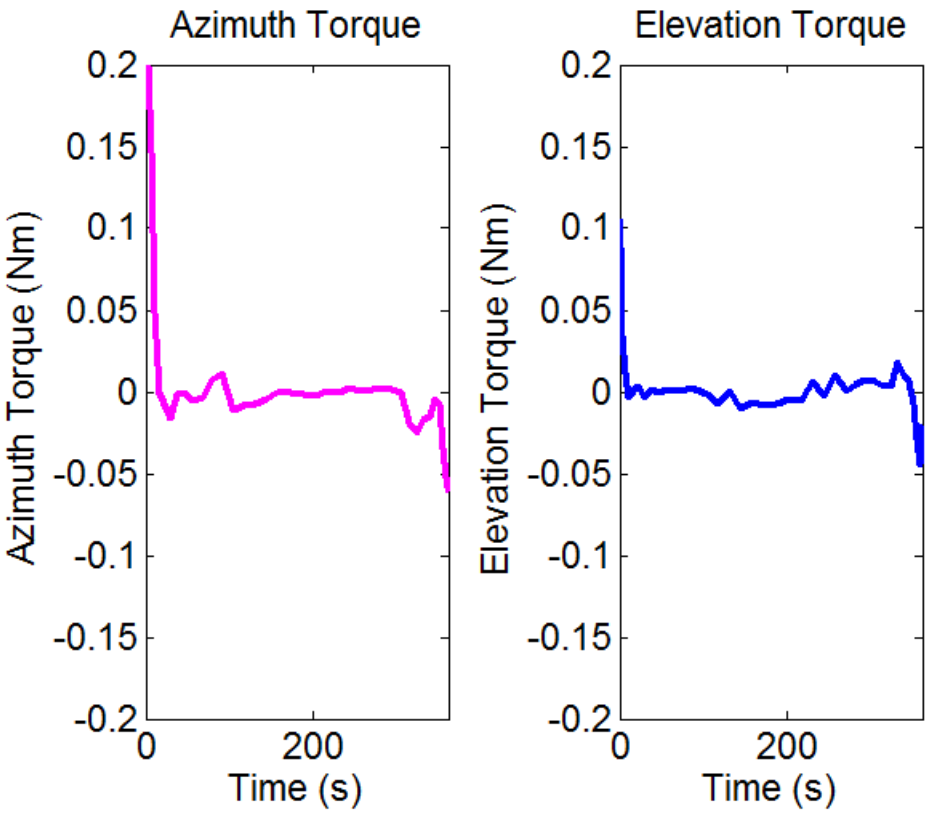


Figure 64. TDRS Scenario 6 Minimum Disturbance Torque Profile

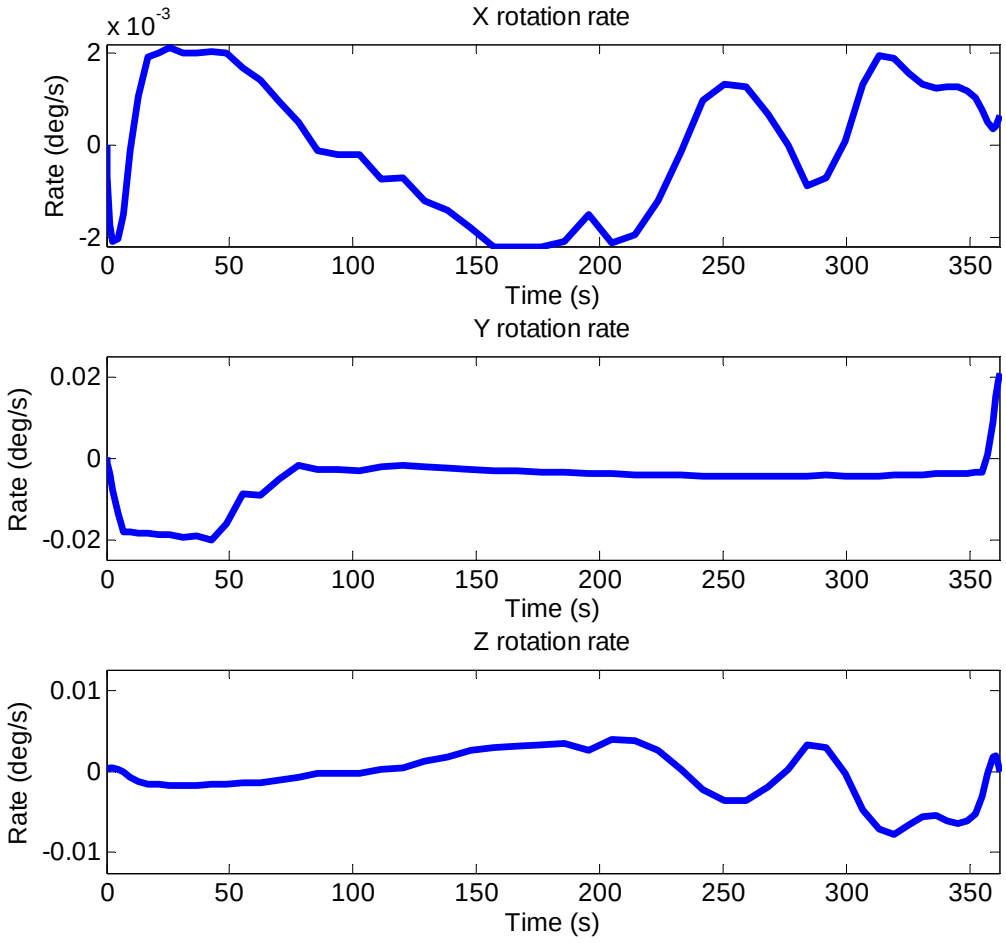


Figure 65. TDRS Scenario 6 Minimum Disturbance Spacecraft Spin Rate

Even with optimization, the spin rate does reach rates that approach the boundary rate (0.0208 °/s). As opposed to the minimum time maneuvers, the system never reaches the rate constraint and recovers to reduce the total spin rate throughout the entire maneuver. However, this is performed at the cost of additional spin rate in the x and z axes. Adjustments to the cost function and weighting factors could reduce this imbalance to yield a desired momentum minimization. This response can be compared to the conventional maneuver, as seen in Figure 66.

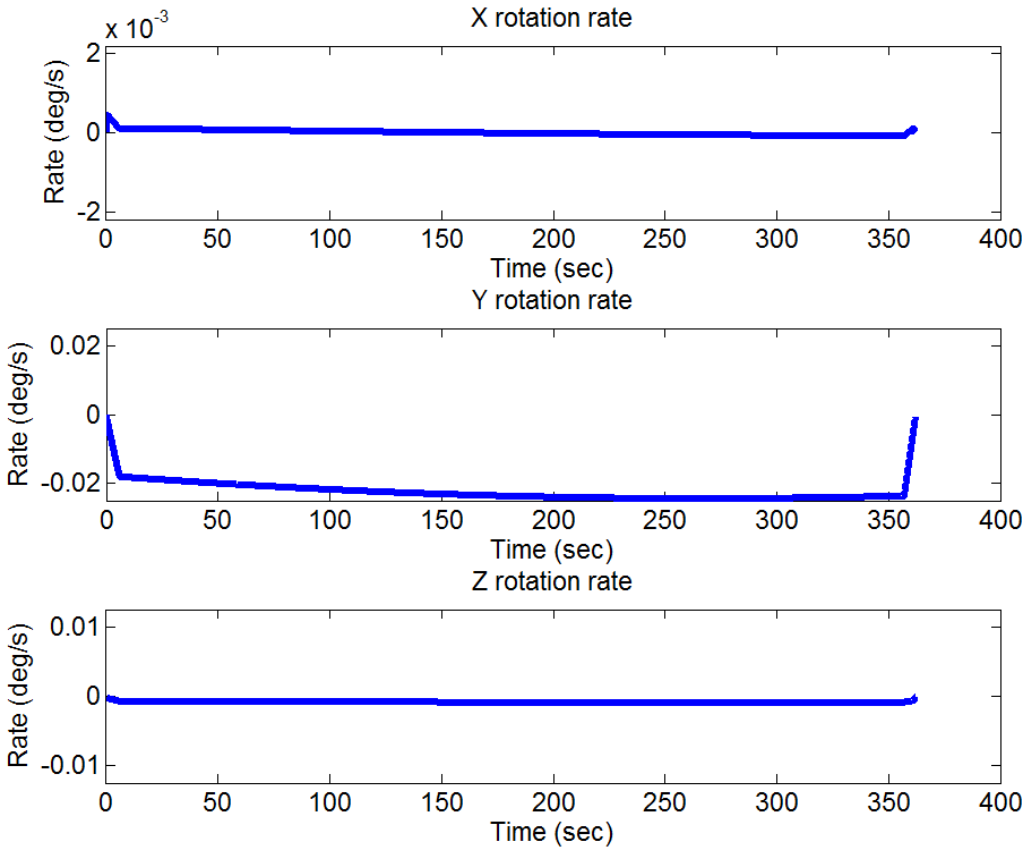


Figure 66. TDRS Scenario 6 Conventional Slew

The conventional slew imparts much less spin on the x and z axes, but the spin rate along the y-axis is near or at the limit for the majority of the maneuver. Figure 67 and Figure 68 show the slew magnitudes of the spacecraft rates. It is evident that the optimal maneuver greatly reduces the disturbance on the spacecraft.

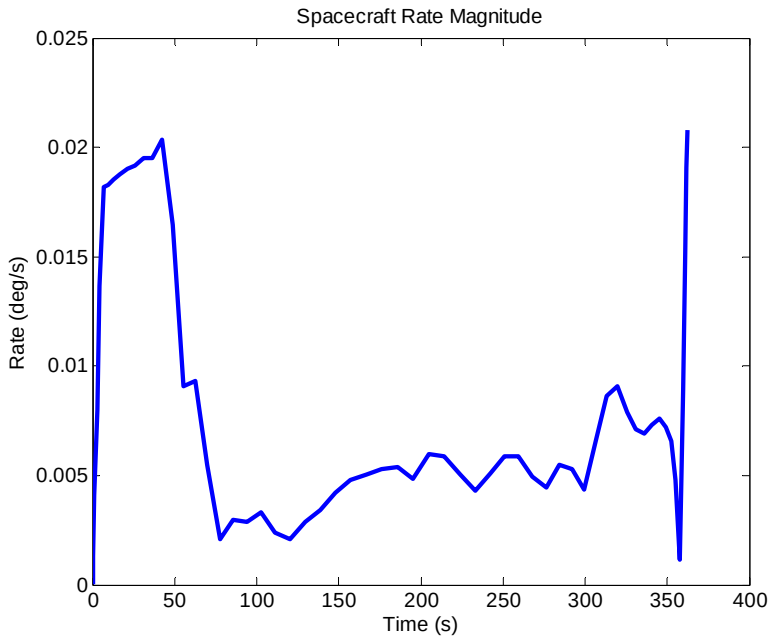


Figure 67. TDRS Scenario 6 Minimum Disturbance Spacecraft Rate Magnitude

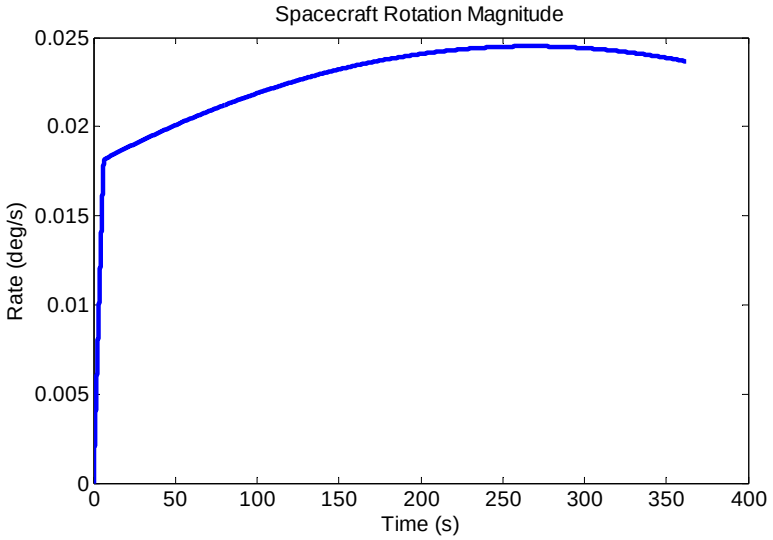


Figure 68. TDRS Scenario 6 Conventional Slew Spacecraft Rate Magnitude

F. IMPLEMENTATION ON LABORATORY TESTBED

In order to test optimal control slews in a controlled environment, a modular testbed was designed, built, and tested by Lt Greg Contreras, USN [23]. The testbed consists of a lightweight, transportable base, a support arm, small antenna dish, and

associated power and control systems. The dish can rotate about the elevation axis, and the support arm rotates to provide azimuth pointing. Both axes are driven by brushless DC motors built by Maxon. Azimuth is driven by a gear connected to the bottom of the support arm. Elevation is driven directly by a gimbal support piece rigidly fixed to the antenna. The motors controlled by Maxon EPOS2 controllers and the entire system can be controlled in realtime. The testbed was designed to be completely modular, and various end effectors could be replaced in order to simulate a multitude of designs or missions, including communications antennas, telescopes, or weapon systems. Lastly, the system is lightweight, portable, and can be operated remotely. Further details of the system, its design, and its operation can be found in [23].

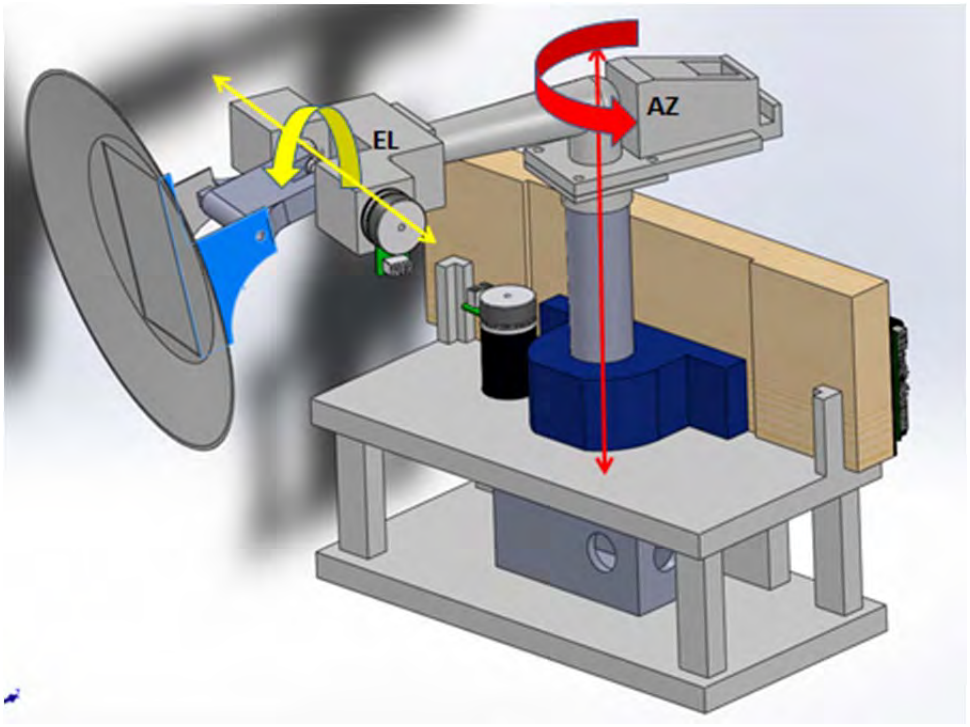


Figure 69. Azimuth-Elevation Laboratory Testbed [23]

The optimal trajectrory provided by DIDO for scenario 3 was implemented on the laboratory testbed. The scenario called for a 0.7183° change in azimuth and 0.9548° change in elevation and was optimized for a slew time of 8.66 seconds, as depicted in Figure 70.

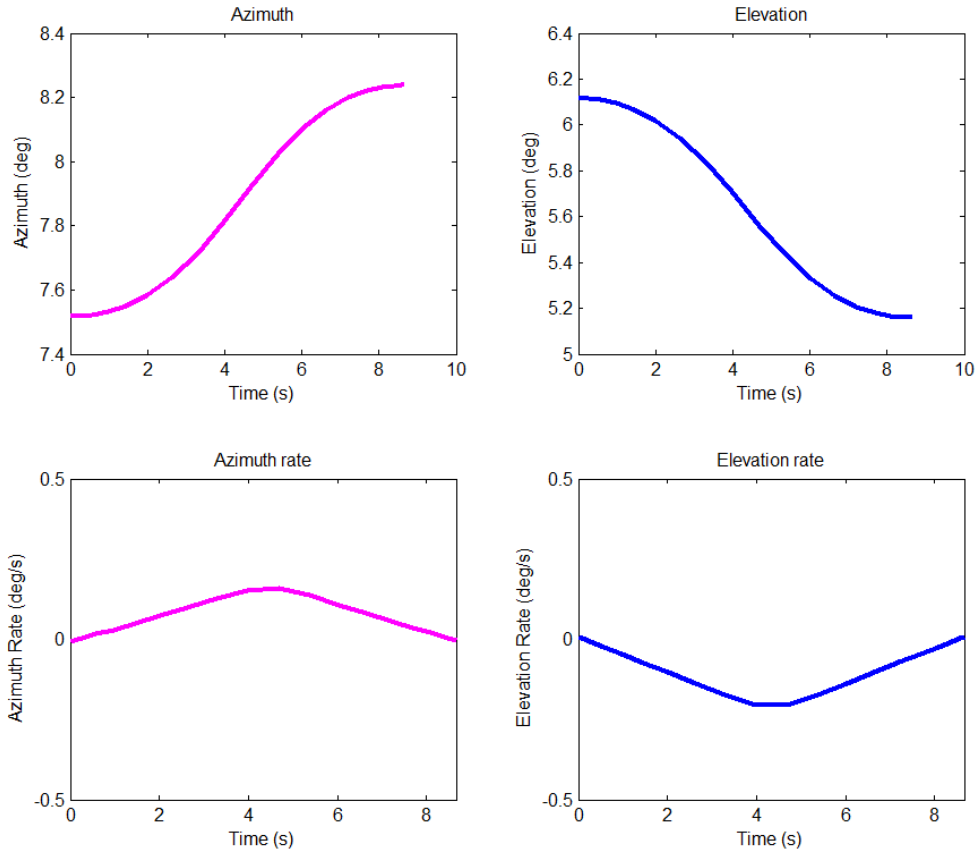


Figure 70. TDRS Scenario 3 Optimal Gimbal Trajectory

In order to implement this slew on TDRS, the trajectory would have to be converted to a format that the TDRS control system can process. This involves fitting the position data to a third order polynomial. The position data for azimuth and elevation were fit to a polynomial using the MATLAB function, polyfit. This resulted in two equations:

$$\theta_1 = -0.0025t^3 + 0.0331t^2 - 0.0182t + 7.5201\,(\text{deg}) \tag{7.1}$$

$$\theta_2 = 0.033t^3 - 0.0451t^2 + 0.0346t + 6.1091\,(\text{deg}) \tag{7.2}$$

Comparing the optimal trajectory with polynomial trajectories, shown in Figure 71, some differences can be found. There are losses in fidelity that occur when the system is fit using a third order polynomial. Further work would need to be done to explore ways to mitigate this loss. This could include adjusting the cost function to obtain

smoother functions, implementing a path file that could drive a third order polynomial response, or finding innovative ways to provide TDRS with the more complex trajectory.

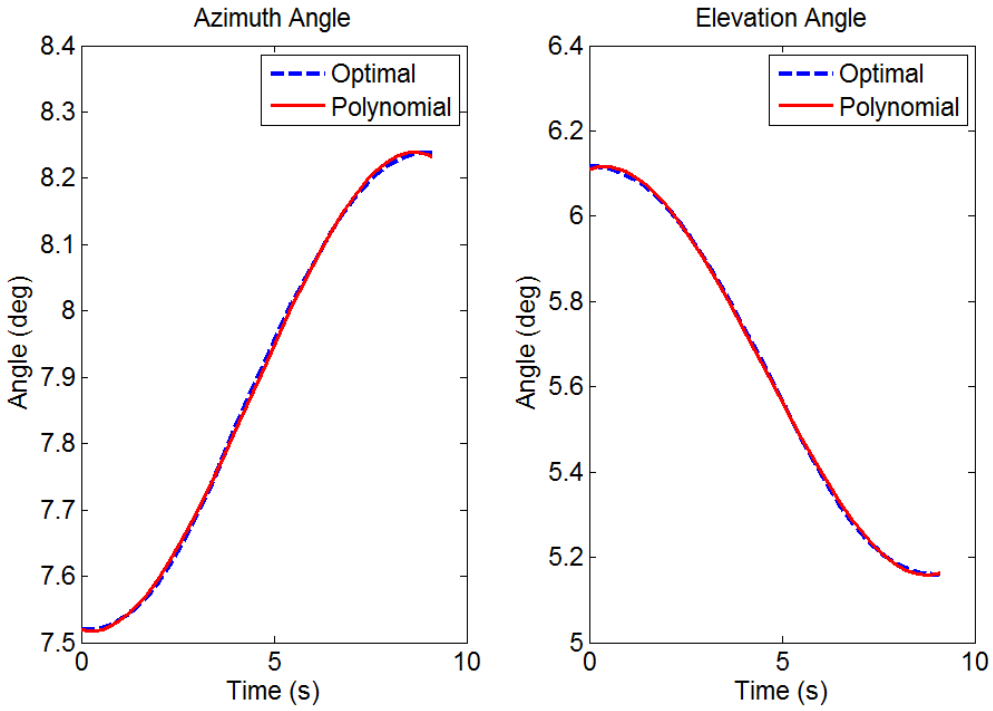


Figure 71. TDRS Scenario 6 Optimal & Polynomial Slew Comparison

These equations were then evaluated over the slew time at discrete intervals of 0.05 seconds. This was the required position data format for the testbed. This data was converted to quad counts (462848 quad counts per 360°) and saved into a .csv (comma separated values) file, which could be read by the testbed controllers.

The testbed was configured to the position control mode and provided the optimal control path as described in [23]. This resulted in the responses in Figure 72 and Figure 73. The red lines indicate the demanded position. The black lines, overlaid over the demanded position, represent the actual position. Both demanded and actual position is measured in quad counts on the vertical axis.

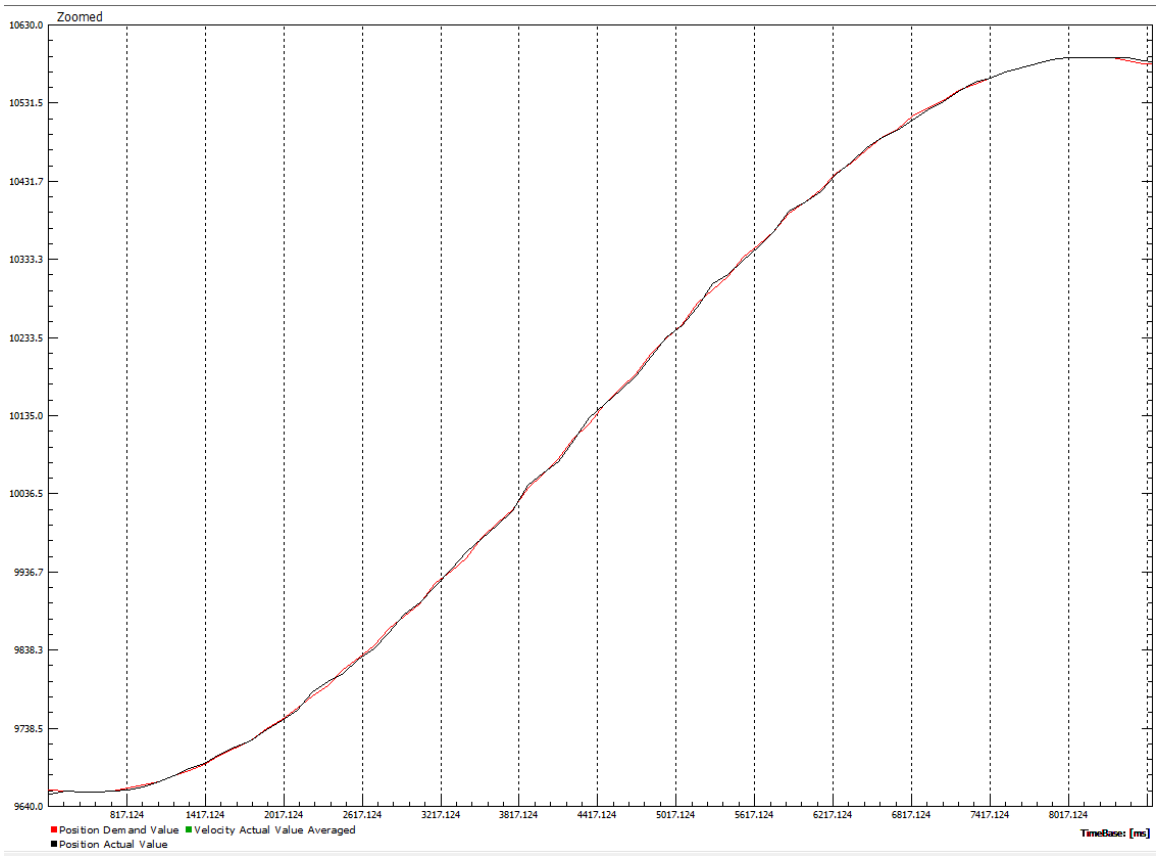


Figure 72. Experimental Implementation of Optimal Antenna Slew (Azimuth)

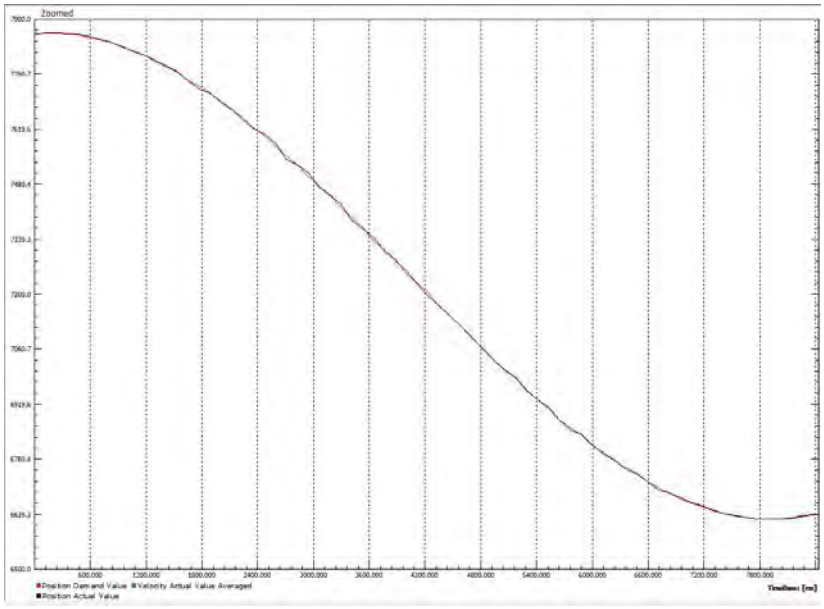


Figure 73. Experimental Implementation of Optimal Antenna Slew (Elevation)

The red line in Figure 79 and Figure 80 depicts the demanded position fed to the testbed. The black line represents the actual response generated by the testbed. As can be seen, the actual response closely matches the desired trajectory. Overall, this test demonstrated the ability to generate an optimal slew maneuver and to steps required to successfully implement this maneuver on a physical system.

VIII. CONCLUSION

A. CONCLUSION

This thesis demonstrated the improvements to spacecraft antenna slew maneuvers that can be obtained by utilizing optimal control theory. The primary driver of these improvements is a shift in problem formulation. Traditionally, the antenna control system is designed in such a way as to minimize its impact on spacecraft pointing. In order to meet spacecraft pointing requirements, conservatism is often built into the maneuver design in order to limit the impact on the spacecraft. For TDRS this resulted in a gimbal rate limit that greatly reduced the system's responsiveness. However, if the problem is attacked from a different angle, the unnecessary conservative can be removed. Simply by a change in the design point of view, additional capability can be gleaned from the system.

It was discovered that the greatest benefit arose for large maneuvers and maneuvers with a large size difference between each axis. This result particularly benefits maneuvers in the EFOV where user time is currently being minimized or lost. Nonetheless, even small maneuvers can be improved. By applying optimal control, it is possible to reduce the worst-case slew time, meaning that additional capacity can be attained with minimal impact to scheduling. This could drastically reduce the inefficiency of slew times and increase operational availability of TDRS while minimizing costly software and hardware updates.

Lastly, the results show that minimizing momentum transfer to the spacecraft from the antenna can be achieved as part of the optimal control solution. This can have impact on the design of future TDRS bus systems, specifically by relaxing the requirements on the ADCS.

These results do not apply solely to antenna pointing. In effect, this theory can apply to any pointing system: robotic manipulators, optics, even weapon systems. Furthermore, it can be applied on systems other than spacecraft.

B. FUTURE WORK

The scope of this thesis was to investigate the feasibility of optimal antenna slews. This limited the extent that thorough analysis could be performed, especially with the available spacecraft data. Furthermore, while the basic concepts and functionality were thoroughly vetted, many assumptions were made to minimize the complexity of the model. In order to implement this type of maneuver on a spacecraft, a fully complete and high fidelity model would need to be utilized. This would include flexibility, sloshing, vibration, static and dynamic friction, non-inertial affects, external disturbances, etc.

Furthermore, more detailed knowledge of the conventional slew maneuver profile could lead to further benefits. The slew method indicated in literature does not account for the nominal three minute slew window that TDRS utilizes. For example, the three minute TDRS slew window is based on the maximum maneuver size within the LEO FOV, 28°. However, for maximum acceleration to maximum gimbal rate, this would only account for a 130 second slew. This leaves 50 seconds of slew time that is being utilized by TDRS in some unknown way. Because of this, it is very possible that even more improvements could be found within the full slew maneuver.

Above all, the spacecraft ADCS would need to be accurately modeled. TDRS ADCS was not modeled in this thesis due to unknowns in its sizing and control algorithms. However, if these details were known and modeled, then assumptions could be minimized and more accurate ADCS requirements incorporated into the problem definition to yield a more applicable result. Furthermore, the minimum time and minimum momentum maneuvers could be combined to yield a single problem definition that could provide more system availability, while maintaining or exceeding spacecraft pointing requirements.

Once thoroughly vetted on a high fidelity spacecraft model, operational analysis would need to be conducted. Currently, TDRS is meeting its defined mission requirements. Additionally, even though TDRS performs upwards of 50 slews a day, there are certain periods where the system is not being utilized due to lack of user requests. Therefore, if the operational analysis were performed over an entire day or a

long period of time, the benefits would be lost if averaged out. It is imperative that operational analysis focus on missions that are not being met or operations impacted by extended slew times that could be solved with the application of optimal control. Further focus should be paid to times of high congestion where the need for timeliness is highest and the mission impact of delayed acquisition the most critical.

Lastly, the ability of the desired system to perform these types of maneuvers must be evaluated. This will be a system specific. Optimal maneuvers may be easier to implement on higher order polynomials or different control schemes altogether. TDRS utilizes a third order polynomial, and this led to a loss of some precision regarding the response. While it is possible the system could still perform the required optimal maneuver under these constraints, it cannot be discounted that the very characteristics that assist the maneuver could be lost. There are ways around this problem, such as incorporating the control scheme in problem formulation, or innovate ways to trick the system into accepting a more complex command, as demonstrated in the TRACE flight experiment [25]. For systems in development, control schemes necessary for optimal maneuvers should be implemented early in design. This would allow for the best use of the hardware and for the best vehicle performance.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. GENERAL MATLAB CODE FOR THREE LINK SYSTEM

The following is an example MATLAB script that implements the Netwon-Euler process described in this thesis. The code illustrates how to implement a double-pendulum model. As was mentioned in the text, it is straightforward to modify the code for different configurations. Placeholder values have been inserted for necessary parameters such as mass and inertia. This script can be used both to simulate the system and as part of an optimization routine.

```
% Simple three body system

function Three-Body-Newton-Euler-Code
%%
clear all; clc;         %close all;

tic;                    %start program timer
%%Establish main variables
g = 9.81;               %gravity m/s^2

%ACRONYMS
%TROC = TIME RATE OF CHANGE
%DCM = Direction Cosine Matrix

%%-------------------------------------------------------------------
----
%TO TAILOR SYSTEM, COMMENT OUT NON-CONSTRAINED ANGLES AND ANGLE RATES
%CONSTRAINED ANGLE RATES SHOULD BE SET TO ZERO
%All gimbal variables defined in outer link and assumed (+) in outer
link
%comment out unconstrained axes
syms t1x t1y t1z;           %orientation angles of gimbal 1 (rad)
syms u1x u1y u1z;           %rate of change of orientation angle of
gimbal 1 (rad/s)
t1x = 0;                    %angle of gimbal 1 in x direction
t1y = 0;                    %angle of gimbal 1 in y direction
%t1z = 0;                   %angle of gimbal 1 in z direction

u1x = 0;                    %TROC of angle of gimbal 1 in x direction
u1y = 0;                    %TROC of angle of gimbal 1 in Y direction
%u1z = 0;                   %TROC of angle of gimbal 1 in Z direction

%Kinematic differential equation for link 1 (Kane, Likins, Levinson pg
427
Gamma1 = [cos(t1y)*cos(t1z), sin(t1z), 0;
         -cos(t1y)*sin(t1z), cos(t1z), 0;
          sin(t1y)         ,        0, 1];
```

```
%Time rate of change of kinematic differential equation for link 1
Gamma1dot = [(-sin(t1y)*cos(t1z)*u1y) - (cos(t1y)*sin(t1z)*u1z),
cos(t1z)*u1z, 0;
             (sin(t1y)*sin(t1z)*u1y) - (cos(t1y)*cos(t1z)*u1z) , -
sin(t1z)*u1z, 0;
              cos(t1y)*u1y                                      ,
0, 0];

u1 = [u1x; u1y; u1z];       %TROC of angle of gimble 1 in vector form


%%-------------------------------------------------------------------------
----
%TO TAILOR SYSTEM, COMMENT OUT NON-CONSTRAINED ANGLES AND ANGLE RATES
%CONSTRAINED ANGLE RATES SHOULD BE SET TO ZERO
%All gimbal variables defined in outer link and assumed (+) in outer
link
%comment out unconstrained axes
syms t2x t2y t2z;           %orientation angles of gimbal 2 (rad)
syms u2x u2y u2z;           %rate of change of orientation angle of
gimbal 2 (rad/s)
t2x = 0;                    %angle of gimbal 2 in x direction
t2y = 0;                    %angle of gimbal 2 in y direction
%t2z = 0;                    %angle of gimbal 2 in z direction

u2x = 0;                    %TROC of angle of gimbal 2 in x direction
u2y = 0;                    %TROC of angle of gimbal 2 in Y direction
%u2z = 0;                   %TROC of angle of gimbal 2 in Z direction

%Kinematic differential equation for link 2 (Kane, Likins, Levinson pg
427
Gamma2 = [cos(t2y)*cos(t2z), sin(t2z), 0;
         -cos(t2y)*sin(t2z), cos(t2z), 0;
          sin(t2y)         ,        0, 1];

%Time rate of change of kinematic differential equation for link 2
Gamma2dot = [(-sin(t2y)*cos(t2z)*u2y) - (cos(t2y)*sin(t2z)*u2z),
cos(t2z)*u2z, 0;
             (sin(t2y)*sin(t2z)*u2y) - (cos(t2y)*cos(t2z)*u2z) , -
sin(t2z)*u2z, 0;
              cos(t2y)*u2y                                      ,
0, 0];

u2 = [u2x; u2y; u2z];       %TROC of angle of gimble 1 in vector form

%%Direction Cosine matrices
%ONLY APPlICABLE FOR THIS PROBLEM, NOT GENERALIZED
%DCMS 12 and 23 ARE FROM KANE,LIKINS,LEVINSON PG 423

%Body-three: 1–2-3
%Transformation matrix from 1 to 2
DCM_21 = [cos(t1y)*cos(t1z) ,  sin(t1x)*sin(t1y)*cos(t1z) +
cos(t1x)*sin(t1z), -cos(t1x)*sin(t1y)*cos(t1z) + sin(t1x)*sin(t1z);
```

```
          -cos(t1y)*sin(t1z), -sin(t1x)*sin(t1y)*sin(t1z) +
cos(t1z)*cos(t1x),  cos(t1x)*sin(t1y)*sin(t1z) + sin(t1x)*cos(t1z);
          sin(t1y)          ,                                 -
sin(t1x)*cos(t1y),                               cos(t1x)*cos(t1y)];

%Transformation matrix from from 2 to 3
DCM_32 = [cos(t2y)*cos(t2z) ,  sin(t2x)*sin(t2y)*cos(t2z) +
cos(t2x)*sin(t2z), -cos(t2x)*sin(t2y)*cos(t2z) + sin(t2x)*sin(t2z);
          -cos(t2y)*sin(t2z), -sin(t2x)*sin(t2y)*sin(t2z) +
cos(t2z)*cos(t2x),  cos(t2x)*sin(t2y)*sin(t2z) + sin(t2x)*cos(t2z);
          sin(t2y)          ,                                 -
sin(t2x)*cos(t2y),                               cos(t2x)*cos(t2y)];

syms q1 q2 q3 q4
q1 = 0;                 %Quaternion 1; comment if non-inertial body 1
q2 = 0;                 %Quaternion 2; comment if non-inertial body 1
q3 = 0;                 %Quaternion 3; comment if non-inertial body 1
q4 = 1;                 %Quaternion 4; comment if non-inertial body 1
q = [q1;q2;q3;q4];

syms qdot1 qdot2 qdot3 qdot3
qdot1 = 0;              %TROC of q1; comment if non-inertial body 1
qdot2 = 0;              %TROC of q2; comment if non-inertial body 1
qdot3 = 0;              %TROC of q3; comment if non-inertial body 1
qdot4 = 0;              %TROC of q4; comment if non-inertial body 1
qdot = [qdot1;qdot2;qdot3;qdot3];

%Transformatrion matrix from N to 1 (BONG WIE pg 335)
DCM_1N = [1-2*(q2^2+q3^2), 2*(q1*q2+q3*q4), 2*(q1*q3+q2*q4);
         2*(q2*q1+q3*q4), 1-2*(q1^2+q3^2), 2*(q2*q3+q1*q4);
         2*(q3*q1+q2*q4), 2*(q3*q2+q1*q4), 1-2*(q1^2+q2^2)];


%Additional DCMs based on primary DCMs
DCM_12 = DCM_21.';
DCM_23 = DCM_32.';
DCM_N1 = DCM_1N.';

DCM_2N = DCM_21*DCM_1N;
DCM_N2 = DCM_2N.';

DCM_3N = DCM_32*DCM_2N;
DCM_N3 = DCM_3N.';

DCM_31 = DCM_32*DCM_21;
DCM_13 = DCM_31.';


%%Rotational Velocities

%Angular rate of body 1 relative to inertial in _____ frame
(rad/s)????????? FRAME
```

```
%??????????????????????????????????????????????????????????????????????????
??????????????????????????????????????????????????????????????????????????
????????????????????????????????????????
syms w1x w1y w1z

w1 = [w1x; w1y; w1z];       %angular rate of body 1 in vector form

%cross product matrix for (w1 X _)
w1c = [0  , -w1z,   w1y;
       w1z,    0,  -w1x;
      -w1y,  w1x,     0];

%double cross product matrix for (w1 X (w1 X _))
w1cc = [-(w1y^2)-(w1z^2),          w1x*w1y,          w1x*w1z;
                 w1y*w1x, -(w1z^2)-(w1x^2),          w1y*w1z;
                 w1z*w1x,          w1z*w1y, -(w1x^2)-(w1y^2)];

%%-------------------------------------------------------------------------
----
%Angular rate of body 2 relative to body 1 in body 2 frame (rad/s)
w2 = DCM_21*w1 + Gamma1*u1;
w2x = w2(1); w2y = w2(2);  w2z = w2(3);

%cross product matrix for (w2 X _)
w2c = [0  , -w2z,   w2y;
       w2z,    0,  -w2x;
      -w2y,  w2x,     0];

%double cross product matrix for (w2 X (w2 X _))
w2cc = [-(w2y^2)-(w2z^2),          w2x*w2y,             w2x*w2z;
                 w2y*w2x, -(w2z^2)-(w2x^2),          w2y*w2z;
                 w2z*w2x,          w2z*w2y, -(w2x^2)-(w2y^2)];


%%-------------------------------------------------------------------------
----
%Angular rate of body 2 relative to body 1 in body 2 frame (rad/s)
w3 = DCM_32*w2 + Gamma2*u2;
w3x = w3(1); w3y = w3(2);  w3z = w3(3);

%cross product matrix for (w3 X _)
w3c = [0  , -w3z,   w3y;
       w3z,    0,  -w3x;
      -w3y,  w3x,     0];

%double cross product matrix for (w3 X (w3 X _))
w3cc = [-(w3y^2)-(w3z^2),          w3x*w3y,          w3x*w3z;
                 w3y*w3x, -(w3z^2)-(w3x^2),          w3y*w3z;
                 w3z*w3x,          w3z*w3y, -(w3x^2)-(w3y^2)];


%%Inertia Properties
```

```
%%------------------------------ ------------------------------------
-----

m1 = 10;                  %mass of body 1 (kg)
m2 = 10;                  %mass of body 2 (kg)
m3 = 10;                  %mass of body 3 (kg)


M1 = m1*eye(3);          %mass matrix for body 1
M2 = m2*eye(3);          %mass matrix for body 2
M3 = m3*eye(3);          %mass matrix for body 3

zero =0*eye(3);          %3x3 zero placeholder for matrices

I1 = eye(3); %replace with inertia matrix
%moments of inertia of body 1 in frame 1 (kg m^2)

%moments of inertia of 2nd body in frame 2
I2 = eye(3); %replace with inertia matrix
%moments of inertia of body 2 in frame 2 (kg m^2)


%moments of inertia of 3rd body in fame 3
I3 = eye(3); %replace with inertia matrix
%moments of inertia of body 3 in frame 3 (kg m^2)


%%------------------------------------------------------------------------
----
H1 = I1*w1;              %Angular momentum of body 1 in frame 1
H2 = I2*w2;              %Angular momentum of body 2 in frame 2
H3 = I3*w3;              %Angular momentum of body 3 in frame 3

%syms l1 l2
%%Radii
%radius of CG of body 1 to gimbal 1 in frame 1
r11x = 0; r11y = 0; r11z = 0;
r11 = [r11x; r11y; r11z];   %radius of CG of body 1 to gimbal 1

r11c = [0  , -r11z,  r11y;
      r11z,    0,  -r11x;
     -r11y,  r11x,     0];
%radius from cm1 to gimbal 1 cross product


%radius of CG of body 2 to gimbal 1 in frame 2
syms r21x r21y r21z
r21x= R21(1); r21y = R21(2); r21z = R21(3);
r21 = [r21x; r21y; r21z];   %radius of CG of body 2 to gimbal 1

r21c = [0  , -r21z,  r21y;
      r21z,     0, -r21x;
```

```
     -r21y,  r21x,     0];
%radius from cm2 to gimbal 1 cross product


%radius of CG of body 2 to gimbal 2 in frame 2
syms r22x r22y r22z
r22x = R22(1); r22y = R22(2); r22z = R22(3);
r22 = [r22x; r22y; r22z];   %radius of CG of body 2 to gimbal 2

r22c = [0  , -r22z,  r22y;
      r22z,     0, -r22x;
     -r22y,  r22x,     0];
%radius from cm2 to gimbal 2 cross product


%radius of CG of body 3 to gimbal 2 in frame 3
syms r32x r32y r32z
r32x = R32(1); r32y = R32(2); r32z = R32(3);
r32 = [r32x; r32y; r32z];   %radius of CG of body 3 to gimbal 2

r32c = [0  , -r32z,  r32y;
      r32z,    0,  -r32x;
     -r32y,  r32x,     0];
%radius from cm3 to gimbal 2 cross product

%%-------------------------------------------------------------------------
----
%External forces on body 1
syms F1x F1y F1z
F1x = 0; F1y = 0; F1z = 0;
F1 = [F1x; F1y; F1z];

%TO SOLVE FOR BASE FORCE, USE SYMS F1X AND F1Y, THEN SOLVE FOR XDOT
BELOW
%AND PULL OUT VX AND VY.
%THEN USE THE FOLLOWING LINE:
%solutions = solve(Vx == 0, Vy == 0, F1x, F1y)
%TO SOLVE FOR F1X AND F1Y, THEN COPY PASTE


%External forces on body 2
F2x = 0; F2y = -g*m2; F2z = 0;
F2 = [F2x; F2y; F2z];

%External forces on body 3
F3x = 0; F3y = -g*m3; F3z = 0;
F3 = [F3x; F3y; F3z];

%External torques on body 1
T1x = 0; T1y = 0; T1z = 0;
T1 = [T1x; T1y; T1z];

%External torques on body 2
```

```
T2x = 0; T2y = 0; T2z = 0;
T2 = [T2x; T2y; T2z];

%External torques on body 3
T3x = 0; T3y = 0; T3z = 0;
T3 = [T3x; T3y; T3z];

syms Fg1x Fg1y Fg1z;        %Forces at gimbal 1
Fg1 = [Fg1x; Fg1y; Fg1z];

syms Fg2x Fg2y Fg2z;        %Forces at gimbal 2
Fg2 = [Fg2x; Fg2y; Fg2z];

%Torques at gimbal 1
Tg1x = 0;
Tg1y = 0;
Tg1z = 0;
Tg1 = [Tg1x; Tg1y; Tg1z];

%Torques at gimbal 2
Tg2x = 0;
Tg2y = 0;
Tg2z = 0;
Tg2 = [Tg2x; Tg2y; Tg2z];


%%Establish main matrices
%Each row of I uses the frame corresponding to the leading column:
%i.e., row 1 is frame 1, row 2 is frame 2, row 3 is frame 3, etc...
I = [I1,      zero,      zero;
     I2*DCM_21, I2*Gamma1,      zero;
     I3*DCM_31, I3*DCM_32*Gamma1, I3*Gamma2];

%Each row of u uses the inertial reference frame
u = [M1*DCM_N1; M2*DCM_N1; M3*DCM_N1];

%Each row of PI uses the inertial reference frame
PI = [zero
,
zero,                   zero;
      m2*DCM_N2*r21c*DCM_21 - m2*DCM_N1*r11c
,
m2*DCM_N2*r21c*Gamma1,                   zero;
      m3*DCM_N3*r32c*DCM_N1 - m3*DCM_N2*r22c*DCM_21 +
m3*DCM_N2*r21c*DCM_21 - m3*DCM_N1*r11c, m3*DCM_N3*r32c*DCM_32*Gamma1 -
m3*DCM_N2*r22c*Gamma1 + m3*DCM_N2*r21c*Gamma1, m3*DCM_N3*r32c*Gamma2];

%Each row of T uses the frame corresponding to that row
T = [T1 - w1c*H1 - DCM_12*Tg1;
     T2 - w2c*H2 + Tg1 - (DCM_23*Tg2) - I2*Gamma1dot*u1 -
I2*DCM_21*w1c*DCM_12*Gamma1*u1;
```

```
     T3 - w3c*H3 + Tg2 - I3*DCM_32*Gamma1dot*u1 - I3*Gamma2dot*u2 -
I3*DCM_31*w1c*DCM_13*Gamma2*u2 - I3*cross(DCM_32*Gamma1*u1,
Gamma2*u2)];

%Each row of F uses the inertial reference frame
F = [F1;
     F2 - m2*DCM_N2*r21c*Gamma1dot*u1 - m2*DCM_N1*w1cc*r11 +
m2*DCM_N2*w2cc*r21 - m2*DCM_N2*r21c*DCM_21*w1c*DCM_12*Gamma1*u1;
     F3 - m3*DCM_N2*r21c*Gamma1dot*u1 + m3*DCM_N2*r22c*Gamma1dot*u1 -
m3*DCM_N3*r32c*DCM_32*Gamma1dot*u1 - m3*(DCM_N3*r32c*Gamma2dot*u2) -
m3*(DCM_N1*w1cc*r11) + m3*(DCM_N2*w2cc*r21) - m3*(DCM_N2*w2cc*r22) +
m3*(DCM_N3*w3cc*r32) - m3*(DCM_N3*r32c*DCM_31*w1c +
DCM_N2*r21c*DCM_21*w1c - DCM_N2*r22c*DCM_21*w1c)*DCM_12*Gamma1*u1 -
m3*DCM_N3*r32c*DCM_31*w1c*DCM_13*Gamma2*u2 -
m3*DCM_N3*r32c*cross(DCM_32*Gamma1*u1,Gamma2*u2)];

%Each row of P uses the frame corresponding to that row
P = [r11c*DCM_1N ,  zero;
     -r21c*DCM_2N,  r22c*DCM_2N;
     zero        , -r32c*DCM_3N];

J = [eye(3) ,   zero;
     -eye(3),  eye(3);
        zero, -eye(3)];

%Restructured A matrix
A = [I, [zero;zero;zero];
    PI(1:3,:), u(1:3,:)];


S = [PI(4:end,:), u(4:end,:)];

R = [P;
     J(1:3,:)];

U = J(4:end, :);

Tprime = [T;F(1:3,:)];

Fprime = F(4:end,:);

%ELEM vector denotes only the unconstrained variables - see numbers
below
% % % w1x = 1
% % % w1y = 2
% % % w1z = 3
% % % u1x = 4
% % % u1y = 5
% % % u1z = 6
% % % u2x = 7
% % % u2y = 8
% % % u2z = 9
% % % v1x = 10
```

```
% % % v1y = 11
% % % v1z = 12
%example, an inertial link 1 (w1,v1=0), and only az (y) in gimbal 1 and
%el (z) in gimbal two would be ELEM = [5,9]
%example, inertial link 1 with double pendulum (both gimbals in z)
% ELEM = [6,9]
ELEM = [6,9];
[m,n] = size(ELEM);
%-------------------------------------------------------------
Tprime2 = Tprime;
Aprime = A(:,ELEM);
Rprime = R;
Cnst3A = [Aprime(7,:); Aprime(8,:); zeros(1,n)];
Cnst3R = [Rprime(7,:); Rprime(8,:); zeros(1,6)];
Cnst3T = [Tprime2(7); Tprime2(8); 0];

Aprime(4:6,:) = Aprime(4:6,:) + DCM_23*Cnst3A;
Rprime(4:6,:) = Rprime(4:6,:) + DCM_23*Cnst3R;
Tprime2(4:6)  = Tprime2(4:6)  + DCM_23*Cnst3T;

Cnst2A = [Aprime(4,:); Aprime(5,:); zeros(1,n);];
Cnst2R = [Rprime(4,:); Rprime(5,:); zeros(1,6)];
Cnst2T = [Tprime2(4);  Tprime2(5);           0];

Aprime(1:3,:) = Aprime(1:3,:) + DCM_12*Cnst2A;
Rprime(1:3,:) = Rprime(1:3,:) + DCM_12*Cnst2R;
Tprime2(1:3)  = Tprime2(1:3)  + DCM_12*Cnst2T;

Aprime = Aprime(ELEM,:);
Rprime = Rprime(ELEM,:);
Sprime = S(:,ELEM);
Tprime2 = Tprime2(ELEM);

alpha2 = Rprime*eye(6)/U;
right = Tprime2 - alpha2*Fprime;
left = Aprime - alpha2*Sprime;

global xdot xdotprime
xdot = left\right;

%%
syms x1x x1y x1z
syms v1x v1y v1z

xdotempty = sym('xdotempty',[12 1]);
xdotempty(1:12) = zeros(12,1);
for i = 1:n
    j = ELEM(i);
    xdotempty(j) = xdot(i);
end

%xdotprime = [xdotempty;u1x;u1y;u1z;u2x;u2y;u2z;v1x;v1y;v1z];
%expanded xdot to include total states to be solved in ODE45
xdotprime = [xdotempty(1:12); %will yield w1, u1, u2, v1
```

```
            u1x;   %will yield t1x
            u1y;   %will yield t1y
            u1z;   %will yield t1z
            u2x;   %will yield t2x
            u2y;   %will yield t2y
            u2z;   %will yield t2z
            v1x;   %will yield d1x
            v1y;   %will yield d1y
            v1z;   %will yield d1z
            qdot;  %will yield q
            ];

xdotprime = simplify(xdotprime);

%STARTING CONDITIONS
tf = 10;
w1xinit = 0; w1yinit = 0; w1zinit = 0;
u1xinit = 0; u1yinit = 0; u1zinit = 0;
u2xinit = 0; u2yinit = 0; u2zinit = 0;
v1xinit = 0; v1yinit = 0; v1zinit = 0;
t1xinit = 0; t1yinit = 0; t1zinit = 0;
t2xinit = 0; t2yinit = 0; t2zinit = 0;
d1xinit = 0; d1yinit = 0; d1zinit = 0;
q1init = 0;  q2init = 0;  q3init = 0;  q4init = 1;



xinit = [
        w1xinit;
        w1yinit;
        w1zinit;
        u1xinit;
        u1yinit;
        u1zinit;
        u2xinit;
        u2yinit;
        u2zinit;
        v1xinit;
        v1yinit;
        v1zinit;
        t1xinit;
        t1yinit;
        t1zinit;
        t2xinit;
        t2yinit;
        t2zinit;
        d1xinit;
        d1yinit;
        d1zinit;
        q1init;
        q2init;
        q3init;
        q4init];
```

```
options = odeset('RelTol',1e-6,'AbsTol',1e-9);
[time,response] = ode45(@dynamics, [0 tf], xinit, options);
w1x_ans = response(:,1);
w1y_ans = response(:,2);
w1z_ans = response(:,3);
u1x_ans = response(:,4);
u1y_ans = response(:,5);
u1z_ans = response(:,6);
u2x_ans = response(:,7);
u2y_ans = response(:,8);
u2z_ans = response(:,9);
v1x_ans = response(:,10);
v1y_ans = response(:,11);
v1z_ans = response(:,12);
t1x_ans = response(:,13);
t1y_ans = response(:,14);
t1z_ans = response(:,15);
t2x_ans = response(:,16);
t2y_ans = response(:,17);
t2z_ans = response(:,18);
d1x_ans = response(:,19);
d1y_ans = response(:,20);
d1z_ans = response(:,21);
q1_ans = response(:,22);
q2_ans = response(:,23);
q3_ans = response(:,24);
q4_ans = response(:,25);

figure
subplot(2,2,1)
plot(time,t1y_ans*(180/pi),'m', 'LineWidth',2);
grid on
title('Gimbal 1 angle')
ylabel('Angle (deg)')
xlabel('Time (s)')

subplot(2,2,2)
plot(time,u1y_ans*(180/pi),'m', 'LineWidth',2);
grid on
title('Gimbal 1 rate')
ylabel('Angle (deg/s)')
xlabel('Time (s)')

subplot(2,2,3)
plot(time,t2z_ans*(180/pi),'b', 'LineWidth',2);
grid on
title('Gimbal 2 angle')
ylabel('Angle (deg)')
xlabel('Time (s)')

subplot(2,2,4)
plot(time,u2z_ans*(180/pi),'b', 'LineWidth',2);
grid on
title('Gimbal 2 rate')
ylabel('Angle (deg)')
```

```
xlabel('Time (s)')

save three_body_data.mat;

end

function [dx] = dynamics(t,y)
global xdotprime
w1x = y(1);
w1y = y(2);
w1z = y(3);
u1x = y(4);
u1y = y(5);
u1z = y(6);
u2x = y(7);
u2y = y(8);
u2z = y(9);
v1x = y(10);
v1y = y(11);
v1z = y(12);
t1x = y(13);
t1y = y(14);
t1z = y(15);
t2x = y(16);
t2y = y(17);
t2z = y(18);
d1x = y(19);
d1y = y(20);
d1z = y(21);
q1 = y(22);
q2 = y(23);
q3 = y(24);
q4 = y(25);
%assings input values to state variables

qdot = .5*[q4 , -q3,  q2, q1;
          q3 ,  q4, -q1, q2;
          -q2,  q1,  q4, q3;
          -q1, -q2, -q3, q4]*[w1x;w1y;w1z;0];
%calculate quaternion TROC based on quaternions and w1
%Bong Wie pg 344

dx = [eval(xdotprime)]; %calculates and returns TROCs
t
end
```

LIST OF REFERENCES

[1] Q. Tham et al., “Robust Antenna Pointing Control for TDRS Spacecraft,” in *Proc.36th Conf. on Decision & Control*, San Diego, CA, 1997.

[2] B. Dunbar, “Tracking and Data Relay Satellite (TDRS),” NASA, 13 November 2013. [Online]. Available: http://www.nasa.gov/content/tracking-and-data-relay-satellite-tdrs/#.UqS5j8RDt8E. [Accessed Nobember 2013].

[3] B. Dunbar, “Tracking and Data relay Satellite (TDRS) Fleet,” NASA, 13 November 2013. [Online]. Available: http://www.nasa.gov/directorates/heo/scan/services/networks/txt_tdrs_fleet.html#.UqS_hcRDt8F. [Accessed December 2013].

[4] B. Dunbar, “Tracking and Data Relay Satellite (TDRS) Third Generation,” NASA, 23 August 2013. [Online]. Available: http://www.nasa.gov/content/tracking-and-data-relay-satellite-tdrs-third-generation/#.UqS_vsRDt8E. [Accessed December 2013].

[5] B. Dunbar, “Tracking and Data Relay Satellite (TDRS) Second Generation Capabilities,” NASA, 9 October 2012. [Online]. Available: http://www.nasa.gov/content/tracking-and-data-relay-satellite-tdrs-second-generation-capabilities/#.UqTAgsRDt8E. [Accessed December 2013].

[6] M. Toral et al., “Payload On-Orbit Performance Verification of TDRS HIJ,” in *22nd AIAA International Communications Satellite Systems Conf. & Exhibit 2004*, Monterey, CA, 2004.

[7] B. Dunbar, “Tracking and Data Relay Satellite (TDRS) Characteristics,” NASA, 25 January 2013. [Online]. Available: http://www.nasa.gov/directorates/heo/scan/services/networks/txt_tdrs_characteristics.html#.UqTGBMRDt8H. [Accessed December 2013].

[8] M. Rackley and M. Bielucki, *Interview Regarding TDRS Schedule & Control.* [Interview]. 15 August 2013.

[9] K. Tasaki, “Automation in Space Network Access System (SNAS) for NASA’s Tracking and Data Relay Satellite System (TDRSS),” 27 March 2007. [Online]. Available: http://csse.usc.edu/GSAW/gsaw2007/s3/pajerski.pdf. [Accessed December 2013].

[10] J. Yuan et al., “Single Access Antenna Pointing Control System Design of TDRS,” in *1st International Symposium on Systems and Control in Aerospace and Astronautics*, Harbin, 2006.

[11] H. Schmeichel and T. McElroy, “TDRSS Single-Access Antenna Control System,” in *Guidance and Control 1980, Volume 42: Advances in the Astronautical Sciences*, San Diego, CA, Univelt Inc., 1980, pp. 115–146.

[12] N. Bedrossian, “International Space Station Zero-Propellant Maneuver (ZPM) Demonstration,” NASA, 03 December 2013. [Online]. Available: http://www.nasa.gov/mission_pages/station/research/experiments/274.html. [Accessed December 2013].

[13] A. Stein, “NPS Teams Up with NASA, Industry to Test Groundbreaking Maneuvers on Spacecraft in Orbit,” NPS, 2010. [Online]. Available: http://www.nps.edu/About/News/NPS-Teams-Up-with-NASA-Industry-to-Test-Groundbreaking-Maneuvers-on-Spacecraft-in-Orbit.html. [Accessed December 2013].

[14] I. M. Ross, “A Beginner’s Guide to DIDO,” Ellisar, LLC, Monterey, CA, 1998–2007.

[15] E. Stoneking, “Newton-Euler dynamic equations of motion for a multi-body spacecraft,” in *Proc. 2007 AIAA GNC Conf*, AIAA-2007–6441.

[16] B. Wie, *Space Vehicle Dynamics and Control*, 2nd ed. Reston, VA: AIAA, 2008.

[17] W. Wiesel, *Spaceflight Dynamics* 3rd ed. Beavercreek, OH: Aphelion Press, 2010.

[18] T. R. Kane et al., *Spacecraft Dynamics*, New York: McGraw-Hill, 1983.

[19] R. Gargano, *Conversation Regarding Netwon-Euler Implementation.* [Interview]. 10 June 2013.

[20] N. Nise, *Control Systems Engineering* (6th Ed.). Jefferson City, John Wiley & Sons Inc., 2011.

[21] I. M. Ross, *A Primer of Pontryagin’s Principle in Optimal Control*, San Francisco: Collegiate Publishers, 2009.

[22] I. M. Ross and M. Karpenko, “A Review of Pseudospectral Optimal Control: From Theory to Flight,” *Annual Reviews in Control*, Vol. 36, 2012.

[23] C. Calkings, “TDRS-K Media Kit,” January 2013. [Online]. Available: http://tdrs.gsfc.nasa.gov/assets/files/TDRSKMediaGuide_FINAL.pdf. [Accessed December 2013].

[24] G. Contreras, “Design and Prototyping of a Satellite Antenna Slew Testbed,” M.S. thesis, Dept. of Space Systems, Naval Postgraduate School, Monterey, 2013.

[25] M. Karpenko et al., “First Flight Results on Time-Optimal Spacecraft Slews,” Journal of Guidance, Control, and Dynamics, Vol. 35, No. 2, 2012

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California