AIAA SciTech
4-8 January 2016, San Diego, California, USA
AIAA Guidance, Navigation, and Control Conference

AIAA 2016-0088

Formulation of Torque-Optimal Guidance Trajectories for a CubeSat with Degraded Reaction Wheels

Siddharth S. Kedare* and Steve Ulrich†
Carleton University, Ottawa, Ontario K1S 5B6, Canada

Downloaded by CARLETON UNIVERSITY on January 25, 2016 | http://arc.aiaa.org | DOI: 10.2514/6.2016-0088

This paper presents the theory and design of a torque-optimal guidance algorithm for CubeSat applications. CubeSats and nano-satellites provide mission-flexible low-cost platforms for the academic and scientific communities to conduct cutting-edge research in the harsh environment of space. The mission life of nano-satellites may be limited by the attitude actuators, and it is therefore beneficial to reduce torque and angular momentum usage during reorientation maneuvers. The algorithm focuses on being computationally lightweight and robust, while including the effects of gyroscopic moments, environmental torques, and degraded reaction wheels. Results indicate that this torque-optimal guidance algorithm demonstrates substantial improvements in performance and pointing accuracy over an Eigenaxis controller for similar maneuvers, with low to moderate computational overhead. In doing so, it presents a significant advancement towards the development of intelligent GN&C systems for small satellites.

I. Introduction

For a space mission to be successful, a spacecraft must be capable of accurately pointing its payload at the required target.[1] The associated slewing maneuvers are typically executed by a series of motorized rotating masses such as reaction wheels, momentum wheels, or control moment gyroscopes (CMG), which provide maneuvering torque and angular momentum storage.[2] Over the duration of a mission, these devices accumulate angular momentum while countering persistent external disturbance torques from the space environment. This momentum accumulation necessitates the use of additional momentum control systems, such as magnetorquers or thrusters, to desaturate the wheels in a process known as momentum dumping. However, a full-size attitude control system (ACS) is too large and expensive to be installed on nano-satellites or CubeSats,[3] leading to the widespread use of pico-satellite reaction wheels[4] and passive control systems.[5,6]

Magnetorquers are often used alongside an active ACS for momentum dumping tasks,[7] due to their low cost, reliability, and simplicity. Though highly efficient, magnetorquers are not without their disadvantages. The Michigan Exploration Laboratory suspects that in October 2011, their M-Cubed CubeSat unintentionally became magnetically conjoined to Explorer-1 Prime, a second CubeSat released simultaneously, via strong onboard magnets used for passive attitude control.[8] Such an occurrence highlights the necessity to explore alternate momentum dumping techniques as applicable to nano-satellites and CubeSat class payloads. Additionally, the magnetic field of the Earth is not well known, and is continuously being affected by solar weather. Near the geomagnetic equator, only roll and yaw momentum can be dumped,[9,10] allowing for the saturation of pitch momentum. Furthermore, the use of magnetic dipoles generates an additional magnetic field on the spacecraft, which may disrupt the normal operation of sensors and spacecraft systems.

Though microthruster attitude control systems are being developed for nanosatellite applications,[11] existing systems are prone to leakage and require bulky pressure vessels for propellant storage. Valve actuation solenoids, though typically reliable, have the potential to be stuck open or closed, leading to mission-threatening scenarios, as was experienced during the Gemini 8 mission[12] in 1966.

This paper focuses on the development and simulation of a near real-time low computational-cost guidance algorithm which exploits gyroscopic (internal) and space environment (external) disturbance torques to minimize the use of torque actuators during attitude maneuvers, hence minimizing the accumulation of

*Research Affiliate, Department of Mechanical and Aerospace Engineering, 1125 Colonel By Drive. Member AIAA.
†Assistant Professor, Department of Mechanical and Aerospace Engineering, 1125 Colonel By Drive. Senior Member AIAA.

angular momentum. The use of the space environment in this manner is comparable to the captain of a naval vessel using the inertia of ship and ocean currents to navigate between ports instead of sailing in a straight line, in an effort to reduce fuel consumption. Such an algorithm could significantly reduce the frequency of momentum dump events, thereby decreasing satellite downtime. Additionally, minimizing the usage of ACS actuators would potentially increase their lifespan. For larger satellites, this technique could reduce Reaction Control System (RCS) propellant usage over the duration of the mission, and hence lower lifetime operational costs. Furthermore, reduced RCS usage would minimize the risk of spacecraft surface contamination by the associated propellants. Similar guidance algorithms have been successfully implemented for large-angle reorientation of the International Space Station (ISS). On November 5, 2006[13] and March 3, 2007,[14] Zero Propellant Maneuver (ZPM) trajectories were utilized for 90° and 180° rotation of the ISS respectively. Combined, the two ZPMs saved approximately US$1,500,000 in propellant costs.[15] However, these ZPM trajectories were generated offline at ground stations before being uploaded to the ISS control system.

II. Previous Work

Typical satellite control systems perform pointing, slewing, and maneuvering by a rotation about an Eigenaxis. An Eigenaxis rotation is the shortest angular path between two attitude states, a rationale that is commonly accepted based on its widespread implementation. However, from quaternion kinematics, it can be shown[16] that Eigenaxis maneuvering is not necessarily time- or torque-optimal. Furthermore, Eigenaxis trajectories require the controller to continually fight against the various environmental disturbance torques experienced by the spacecraft,[9] leading to an increased torque requirement.

It is often beneficial to optimize maneuvers to minimize a specific "cost" – typically maneuver duration or associated actuator usage. Doing so can significantly increase spacecraft agility, reduce the need for bulky control actuators, and extend control system lifespan. The optimal satellite reorientation problem is therefore of significant interest in the field of aerospace engineering. Prior work in the field, which has focused on time optimal and torque optimal guidance laws, shall now be presented. Additionally, the use of pseudospectral methods as a means to obtain numerical solutions for the optimal guidance problem shall be outlined.

II.A. Time Optimal Guidance

The general case of minimum-time reorientation of an asymmetric rigid body was first addressed by Proulx and Ross.[17] For control systems in which the three orthogonal components of control are independently constrained, Bilimoria and Wie[16] discovered that non-Eigenaxis maneuvers require less maneuver time than Eigenaxis maneuvers, as the nutational components can provide additional torque along the reorientation axis. This was subsequently proved via numerical simulations by Xiaoli and Junkins[18] in their investigation of time-optimal spacecraft reorientation. Time-optimal attitude maneuvers have been performed in flight, most recently by NASA's Transition Region and Coronal Explorer (TRACE) spacecraft. The design and flight implementation of the associated guidance algorithms, presented by Karpenko et al.,[19] successfully demonstrated the ability of conventional Eigenaxis controllers to follow time-optimal attitude trajectories.

II.B. Torque Optimal Guidance

Torque-optimal guidance focuses on the reduction of the net angular momentum required for satellite reorientation maneuvers. These maneuvers are typically performed utilizing reaction wheels, CMGs, thrusters, or a combination thereof. However, reaction wheels and CMGs have limited angular momentum capacity. Once the minimum/maximum limit is reached, the actuator is said to be saturated, and can no longer provide reliable control authority. To recover attitude control capability, thrusters are used to maintain spacecraft attitude while the individual actuators are reoriented or spun down to decrease the total momentum magnitude. This process is known as desaturation, and is a routine on-orbit operation for spacecraft.

CubeSat class satellites typically do not possess thrusters due to volume, cost, and mass constraints. They rely solely on passive techniques, such as gravity gradient torque and magnetorquers, to desaturate their momentum exchange devices. Related work on the use of environmental torques for actuator desaturation dates back to the Skylab program, where gravity gradient torque was used to desaturate momentum accumulated during the daylight portion of an orbit.[20, 21] Additional applications regarding the use of gravity gradient torque to desaturate CMG momentum have been previously proposed by Powell[22] and Tong.[23] Tong showed in his study that despite complete momentum dumping not being achieved, a significant attitude maintenance CMG desaturation fuel saving of over 40% was attainable, thus demonstrating the potential for extending the life span of a spacecraft.

Prior research has predominantly focused on applications for space stations and large spacecraft. In 2000, an approach for optimizing the attitude command sequence to the ISS CMG attitude hold controller was proposed by Chamitoff et al.[24] This technique focused on minimizing fuel usage while including nonlinear system dynamics. An ISS 90° yaw maneuver was performed without use of propellant, by propagating a simplified model of the vehicle dynamics and environment which included Euler components and gravity gradient, but excluded aerodynamic effects. However, at the end of the maneuver, the CMGs were saturated.

Expanding upon this existing research, Bedrossian and Bhatt[14] developed the ZPM in collaboration with Draper Laboratory. Its origins can be traced back to the mid-1990's when Bedrossian proposed this approach during the development of a general Centralized Momentum Management (CMM) concept.[25] The goal of the CMM was to increase satellite life by trading off between satellite performance and control variables, while incorporating the ability to "look-ahead" in guidance algorithm decision making. In an effort to achieve these goals, a general optimal control problem framework was proposed.

The ZPM can be defined as the development of an attitude trajectory shaped in a manner that takes advantage of the nonlinear system dynamics and environmental disturbance torques to reduce or eliminate the net cost of a rotation.[14] As described earlier in Sec. II, an Eigenaxis maneuver, though kinematically the shortest past, typically results in an increased "cost". By considering a kinematically longer path, the path dependence of the system dynamics can be utilized to lower this cost. In the early 2000s, the ZPM concept was used to establish the feasibility of general three-axis momentum desaturation without the use of thrusters.[26] The study indicated that by suitably maneuvering the spacecraft in a disturbance field, the CMGs could be desaturated.

From the spacecraft dynamics and kinematics, and the associated environmental models, a nonlinear, constrained optimal control (more accurately *guidance*) problem can be defined. Solving such a problem is still considered difficult, despite the mathematics of dynamical optimization theory being established by Euler in the 18th century.[27] In 1962, Pontryagin's principle revolutionalized optimization theory by providing a method to determine optimal control in a constrained problem. However, it is only with recent advances in pseudospectral (PS) methods that have allowed for efficient solutions to optimal control problems in nonlinear dynamics systems.

II.C. Pseudospectral Methods

Pseudospectral methods are a class of direct collocation wherein the optimal control problem is transcribed into a nonlinear programming problem (NLP) by parameterizing the dynamical states and associated control variables using global polynomials and collocating the differential-algebraic equations using nodes obtained from a Gaussian quadrature.[28] Three of the most common sets of collocation points are the *Legendre-Gauss* (LG), *Legendre-Gauss-Radau* (LGR), and *Legendre-Gauss-Lobatto* (LGL) points. All these sets are defined on the non-dimensionalized time domain $[-1, 1]$, but differ significantly in their handling of endpoints. LG points include neither of the endpoints, the LGR points include one endpoint, and LGL points include both endpoints. Of these, the Lobatto PS method (LPM)[29,30] and the Gauss PS method (GPM)[31–33] have been extensively documented in recent years. A detailed comparison of the performance of the LG, LGR, and LGL schemes in a first-order optimality system was performed by Garg et al.,[28] and the results indicated that the Gauss or Radau collocation methods provided state and control accuracies several orders of magnitude more accurate as compared to the Lobatto method. However, in applications where the exact solution was unknown, all three methods provided similar results, with the Lobatto costate oscillating about the correct costate.

II.D. Existing Optimization Software

Multiple software packages are available to solve the optimal control (or guidance) problems outlined in Sec. II.A and Sec. II.B. MATLAB provides an optimization toolbox, called using the `optimtool` command, which allows the user to solve constrained and unconstrained problems with or without the inclusion of control states.[34] However, there are multiple limitations regarding the use of `optimtool`, including the inability to configure and solve multi-phase problems. An example of such a problem would be to have a spacecraft reorient itself, pause for a specified time, and then execute a subsequent reorientation in quick succession. Dedicated software packages, which combine a pseudospectral collocation scheme with a Non-Linear Problem (NLP) solver, have been developed over the years. Examples of such programs are DIDO,[35] SOCS (Sparse Optimal Control Software),[36] and GPOPS (General Pseudospectral OPtimal Software),[37] all of which allow for the evaluation of nonlinear optimal control or guidance problems with path constraints.[38]

DIDO was developed by Ross and his associates[39,40] at the Naval Postgraduate School in 2001, and implements the Legendre PS collocation method alongside the SNOPT (Sparse Nonlinear OPTimizer) NLP solver.[41] It requires no prior knowledge of optimization techniques, nor does it require the user to provide an initial guess. Due to its ease of use, it is used extensively in academia, industry and government laboratories.[42] DIDO has also been flight-tested by NASA, and played an integral role in the formulation of the ZPM outlined earlier in Sec. II.B.

The GPOPS package, on the other hand, employs a Gaussian pseudospectral method. By utilizing the SNOPT solver, GPOPS simultaneously solves the entire trajectory over a small number of nodes based on the path constraints, initial conditions, and initial guesses provided by the user. Due to its open source nature and free availability, the GPOPS package was selected to generate the optimal trajectories. Furthermore, in the context of this research, the use of GPOPS allowed for the evaluation of time-optimal guidance trajectories against the work of Boyarko et al.[43] This simplified the validation of the optimization environment prior to the development of a torque-optimal guidance algorithm.

III. Reaction Wheel Considerations

Reaction wheels are a class of actuators for spacecraft which provide attitude control using a brushless electric motor attached to a high-inertia flywheel which is free to spin about a fixed axis.[44] They operate on the principle of conservation of angular momentum, presented in Eq. (1) as relevant to a spacecraft without external disturbance torques.

$$\vec{h}_{total} = \vec{h}_{sat} + \vec{h}_{RW} = \mathbf{J}_{sat}\vec{\omega}_{sat} + \sum_{i=1}^{n} \mathbf{J}_{RW_i}\vec{\omega}_{RW_i} \tag{1}$$

From Eq. (1), by changing the angular velocity of one or more reaction wheels, $\vec{\omega}_{RW_i}$ the angular velocity vector of the spacecraft, $\vec{\omega}_{sat}$ can be modified. The rate of change of $\vec{\omega}_{RW_i}$ therefore defines the instantaneous torque provided by the reaction wheel for spacecraft maneuvers. This rate of change is typically limited by the driving motor and the condition of the wheel bearings, with degraded bearings resulting in reduced available torque or increased actuator power draw in order to overcome friction. In addition, the maximum angular momentum of each reaction wheel is limited by structural constraints. During the course of a mission, the harsh space environment may degrade bearings and associated lubricants. Increased friction can then lead to reaction wheel failure or seizure. Such an event can severely limit or require restructuring of a space mission, as evidenced during ongoing missions by the Kepler Space Telescope,[45] Hubble Space Telescope,[46] and Hayabusa[47] spacecraft. As part of this research, it was essential to consider the worst-case scenario of performing torque-optimal attitude maneuvers with the reaction wheels providing reduced torque. Minimizing actuator usage in such a scenario would preserve spacecraft maneuverability while reducing the probability of continued degradation and/or complete failure of damaged components.

IV. Optimal Guidance Algorithm

The torque-optimal guidance problem was defined and solved within the MATLAB-Simulink environement, coupled with GPOPS and a Sparse Nonlinear Optimizer (SNOPT). After benchmarking GPOPS to validate its trajectory optimization capabilities, a cost function based on the applied torques was implemented. An on-board orbit propagator provided the initial and final states required by GPOPS, while the controller bounding box was defined based on actuator limits, i.e., torque and angular momentum capacity.

IV.A. GPOPS

The General Pseudospectral Optimal Software (GPOPS) tool was implemented alongside a spacecraft dynamics model to obtain benchmark results, and validate the capability of the MATLAB-Simulink environment to accurately optimize guidance trajectories. Following validation against an existing time-optimal guidance scenario,[16,43,48] GPOPS was utilized for the generation of the torque-optimal guidance trajectory. The software, which is integrated into the MATLAB environment, requires the user to define the dynamics of the problem using differential equations, an associated cost function, connections between phases (if present), and finally the limits and initial guesses for each of the states and controls.[37] In its general form, any optimal guidance/control problem can be formulated so as to minimize the cost function

$$J = \Phi\left(\mathbf{x}\left(t_f\right), t_f\right) + \int_{t_0}^{t_f} g\left(\mathbf{x}\left(t\right), \mathbf{u}\left(t\right), t\right) dt \tag{2}$$

In Eq. (2), Φ is the Mayer and g is the Lagrange component. The system is subject to the dynamic constraints given by

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \tag{3}$$

and the associated boundary conditions

$$\phi\left(\mathbf{x}\left(t_0\right), t_0, \mathbf{x}\left(t_f\right), t_f\right) = 0 \tag{4}$$

In this study, the optimal guidance problem defined in Eqs. (2) - (4) is solved using a direct transcription method called the *Gauss pseudospectral method*.[31] Though a detailed explanation regarding the development and mathematics behind the Gauss pseudospectral method is beyond the scope of this paper, an overview based on work by Huntington and Rao[49] shall be provided to familiarize the reader with the associated algorithms. For a single-phase optimal control problem, the Gauss pseudospectral method can be summarized as follows. First, the original time interval $t \in [t_0, t_f]$, is transformed to the non-dimensionalized time interval $\tau \in [-1, 1]$ as

$$t = \frac{(t_f - t_0)\tau + (t_f + t_0)}{2} \tag{5}$$

Following the temporal transformation in Eq. (5), the cost function in Eq. (2) can be rewritten in terms of τ as

$$J = \Phi(\mathbf{x}(1), t_f) + \frac{(t_f - t_0)}{2} \int_{-1}^{1} g(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau) d\tau \tag{6}$$

In the same manner, the dynamic constraints of Eq. (3) are given in terms of τ

$$\frac{2}{(t_f - t_0)} \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\tau} = \mathbf{f}(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau) \tag{7}$$

The boundary conditions of Eq. (4) in terms of τ are given as

$$\phi(\mathbf{x}(-1), t_0, \mathbf{x}(1), t_f) = 0 \tag{8}$$

We approximate the state, $\mathbf{x}(t)$, in terms of a basis $N+1$ Lagrange interpolating polynomial on the interval from $[-1, 1]$ as

$$\mathbf{x}(t) \approx \mathbf{X}(t) = \sum_{k=0}^{N} \mathbf{X}(t_k) L_k(t) \tag{9}$$

Furthermore, suppose we select the following $N+1$ points on $\tau \in [-1, 1]$ for discretization of the continuous-time problem: the initial point $\tau = -1$ and N Gauss points, τ_k, $k = 1, \cdots, N$, which lie within the interval $[-1, 1]$. The approximation of the state derivative at the Gauss points is then expressed as

$$\left[\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}\right]_{t_i} \approx \left[\frac{\mathrm{d}\mathbf{X}^N}{\mathrm{d}t}\right]_{t_i} = \sum_{k=0}^{N} \mathbf{X}_k \left(\frac{\mathrm{d}L_k}{\mathrm{d}t}\right)_{t_i} = \sum_{k=0}^{N} D_{ik} \mathbf{X}_k = \mathbf{f}(\mathbf{X}_i, \mathbf{U}_i, t_i) \tag{10}$$

where D_{ik} is called the *Differentiation Matrix.* The differential operators $D \in \mathbb{R}^{N \times N}$ and $\bar{D} \in \mathbb{R}^N$ are computed through the exact derivative of the Lagrange interpolating polynomial, $L_k(t)$ as

$$\left[\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}\right]_{t_i} = \dot{\mathbf{x}}(t_i) \approx \dot{\mathbf{X}}(t_i) = \mathbf{X}(t_0) \cdot \bar{D}_i + \sum_{k=1}^{N} \mathbf{X}(t_k) \cdot D_{ik} \tag{11}$$

where $D_{ik} = \dot{L}_k(t_i)$ and $\bar{D}_{ik} = \dot{L}_0(t_i)$ are the differential operators. The continuous-time optimal control problem can then discretized into a nonlinear programming problem (NLP) using the variables $\mathbf{X}_k \in \mathbb{R}^n$ and $\mathbf{U}_k \in \mathbb{R}^m$ for the states and control at the Gauss points respectively, for $k = 1, \cdots, N$. The initial and final states, $\mathbf{X}(t_0) \in \mathbb{R}^n$ and $\mathbf{X}(t_f) \in \mathbb{R}^n$ respectively, are also included as variables. The continuous-time cost function in Eq. (6) is first approximated using a Gauss quadrature.

$$J = \Phi(\mathbf{X}(t_f), t_f) + \frac{t_f - t_0}{2} \sum_{k=1}^{N} w_k g(\mathbf{X}_k, \mathbf{U}_k, t_k) \tag{12}$$

where w_k, the Gauss weight at each Gauss point, is given by

$$w_k = \frac{2}{(1 - \tau_k^2)[P_N']^2} \tag{13}$$

In Eq. (13), P_N' represents the N$^{\text{th}}$-degree Legendre polynomial defining the Gauss points. The differential equation constraints in Eq. (7) are also discretized at the Gauss points as

$$\frac{2}{t_f - t_0} \bar{D}_i \mathbf{X}(t_0) + \frac{2}{t_f - t_0} \sum_{k=1}^{N} D_{ik} \mathbf{X}_k = \mathbf{f}(\mathbf{X}_i, \mathbf{U}_i, t_i), \quad i = 1, \cdots, N \tag{14}$$

where

$$\left[\quad \bar{D}_i \quad D_{ik} \quad \right] = \left[\quad (\mathrm{d}L_0/\mathrm{d}\tau)_{\tau_i} \quad (\mathrm{d}L_k/\mathrm{d}\tau)_{\tau_i} \quad \right] \in \mathbb{R}^{(N+1)\times(N+1)} \tag{15}$$

Huntington and Rao[49] emphasize that unlike previously developed pseudospectral methods,[29,50] the differential equations are collocated only at the Gauss points, not at the boundary points. The discretized boundary conditions are given as

$$\phi(\mathbf{X}(t_0), t_0, \mathbf{X}(t_f), t_f) = 0 \tag{16}$$

Lastly, the terminal state, $\mathbf{X}(t_f)$, is defined using a quadrature approximation to the dynamics as

$$\mathbf{X}(t_f) = \mathbf{X}(t_0) + \frac{t_f - t_0}{2} \sum_{k=1}^{N} w_k \mathbf{f}(\mathbf{X}_k, \mathbf{U}_k, t_k) \tag{17}$$

The cost function of Eq. (12) along with the constraints of Eq. (14) and Eq. (16) define an NLP. The solution of this NLP is an approximate solution to the continuous-time optimal control problem.

IV.B. Sparse Nonlinear Optimizer (SNOPT)

The NLP described in Sec. IV.A was solved with the MATLAB `mex` interface of the NLP solver SNOPT using analytic first-order derivatives for the constraint Jacobian and the gradient of the objective function. A large number of NLP solvers, such as IPOPT,[51] KNITRO,[52] and MINOS,[53] are available to academic and industrial users. However, SNOPT was selected based on availability and prior implementation within the GPOPS environment.

SNOPT is a general purpose system for constrained optimization. It minimizes a linear or nonlinear function subject to bounds on the variables and sparse linear or nonlinear constraints.[41] SNOPT uses a sequential quadratic programming (SQP) algorithm to solve an NLP. The search direction are obtained through QP subproblems which minimize a quadratic model of a Lagrangian function subject to linearized constraints. In order to ensure that convergence is independent of the starting point, an augmented Lagrangian merit function is reduced along each search direction. An in-depth explanation of the SNOPT algorithm is beyond the scope of this paper, and details regarding the implementation of SNOPT may be found in the SNOPT User's Guide.[41] However, it suffices to state that SNOPT is suitable for solving nonlinear programs of the form

$$\text{minimize } \mathbf{x} \text{ for } f_0(\mathbf{x}) \text{ subject to } l \leq \begin{pmatrix} \mathbf{x} \\ f(\mathbf{x}) \\ A_L \mathbf{x} \end{pmatrix} \leq u$$

where $\mathbf{x}$ is an n-vector of variables, l and u are constant lower and upper bounds, $f_0(\mathbf{x})$ is a smooth scalar objective function, A_L is a sparse matrix, and $f(\mathbf{x})$ is a vector of smooth nonlinear constraint functions $f_i(\mathbf{x})$. In the ideal case, the first derivatives of $f_0(\mathbf{x})$ and $f_i(\mathbf{x})$ should be coded by the user. Upper and lower bounds are specified for all variables and constraints.

V. Implementation and Validation

This section presents the structure of the guidance algorithm alongside simplifications made to the dynamics model within the on-board propagator and optimization routine in an effort to reduce computational requirements. The technique utilized to validate the torque-optimal trajectory is described along with the properties of the satellite under consideration, and justification regarding the selected maneuver duration.

V.A. Guidance Algorithm Outline

Figure 1 illustrates the flow of information within the torque-optimal guidance algorithm.

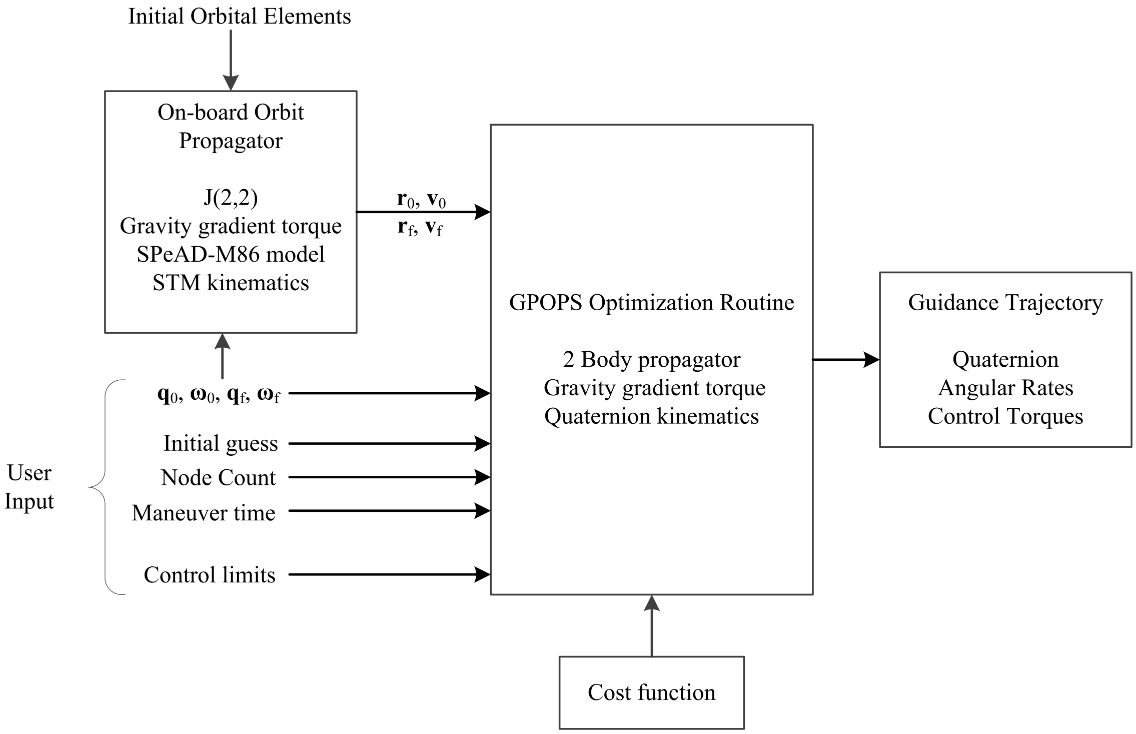


Figure 1. Guidance Algorithm Overview

The core of the guidance algorithm is the GPOPS optimization routine, which generates an optimal trajectory based on the user-defined cost function. The on-board propagator provides the initial and final orbital states required by the optimization routine. Additional user inputs include the desired initial and final attitude states, node count, maneuver time, control torque limits, and an initial guess for the states. The properties of the on-board and optimization propagation environments introduced in Fig. 1 are summarized in Table 1.

Parameter	On-board	Optimization
Time step, Δt (sec)	1	Gauss points
Gravity model	Spherical harmonic	Keplerian
Order of gravity model	2	N.A.
Atmospheric Model	SPeAD-M86	None
Orbit Integrator	ode3	Gaussian quadrature
Attitude propagation	STM	Quaternion
Attitude Integrator	Forward Euler	Gaussian quadrature

Table 1: Orbit and Attitude Propagators within the Guidance Algorithm

Figure 17 in the Appendix presents the Simulink block diagram of the on-board propagation environment. The development and justification of specific aspects of the on-board and optimization propagation environments shall now be discussed.

V.B. State Transition Matrix

In an effort to reduce the computational power required by the on-board model, a state transition matrix (STM), as presented by Whitmore,[54] was implemented for attitude propagation. Details regarding the formulation of this technique can be found in U.S. Patent 6,061,611 "Closed-form integrator for the quaternion (Euler angle) kinematics equations". An abridged derivation of the STM shall now be presented, along with a discrete-time implementation technique. From quaternion kinematics, the rate of change of the quaternion state, $\dot{\mathbf{q}}$, is given by

$$\dot{\mathbf{q}} = \Psi \mathbf{q} \quad \text{where} \quad \Psi = \frac{1}{2}\boldsymbol{\Omega} \tag{18}$$

As Eq. (18) is of the form $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$, it can be integrated to obtain an expression for the continuous state.

$$\mathbf{x}(t) = e^{\int_{t_0}^{t} \mathbf{A}\, \mathrm{d}t} \mathbf{x}_0 \tag{19}$$

Next, let

$$\frac{U}{2} = \int_{t_0}^{t} \frac{\omega_x}{2}\, \mathrm{d}t, \quad \frac{V}{2} = \int_{t_0}^{t} \frac{\omega_y}{2}\, \mathrm{d}t, \quad \frac{W}{2} = \int_{t_0}^{t} \frac{\omega_z}{2}\, \mathrm{d}t \tag{20}$$

Substituting the variables from Eq. (20) into the skew-symmetric matrix $\boldsymbol{\Omega}$, and applying Eq. (19), we obtain a modified expression for the continuous quaternion state.

$$\mathbf{q}(t) = e^{\boldsymbol{\rho}} \mathbf{q}_0 \quad \text{where} \quad \boldsymbol{\rho} = \begin{bmatrix} 0 & \frac{W}{2} & -\frac{V}{2} & \frac{U}{2} \\ -\frac{W}{2} & 0 & \frac{U}{2} & \frac{V}{2} \\ \frac{V}{2} & -\frac{U}{2} & 0 & \frac{W}{2} \\ -\frac{U}{2} & -\frac{V}{2} & -\frac{W}{2} & 0 \end{bmatrix} \tag{21}$$

The state transition matrix, $\boldsymbol{\Phi}(t, t_0)$, is obtained as a function of the angular velocity states by expanding $e^{\boldsymbol{\rho}}$ using the Maclaurin series.

$$\boldsymbol{\Phi}(t, t_0) = \mathbf{I}_4 \cos\left(\frac{\|\omega\|}{2}\right) + \boldsymbol{\rho}\frac{2}{\|\omega\|} \sin\left(\frac{\|\omega\|}{2}\right) \quad \text{where} \quad \|\omega\| = \sqrt{U^2 + V^2 + W^2} \tag{22}$$

The STM is then substituted into Eq. (21) to obtain the quaternion state change equation.

$$\mathbf{q}(t) = \boldsymbol{\Phi}(t, t_0)\mathbf{q}_0 \tag{23}$$

With the STM and associated equation of state defined, we now construct an implementation technique by discretization of the variables presented in Eqs. (20) - (23) at n temporal nodes, with $1 < k < n$.

$$U = \int_{t_0}^{t} \omega_x\, \mathrm{d}t = \frac{\omega_{x_k} + \omega_{x_{k+1}}}{2} \Delta t \tag{24a}$$

$$V = \int_{t_0}^{t} \omega_y\, \mathrm{d}t = \frac{\omega_{y_k} + \omega_{y_{k+1}}}{2} \Delta t \tag{24b}$$

$$W = \int_{t_0}^{t} \omega_z\, \mathrm{d}t = \frac{\omega_{z_k} + \omega_{z_{k+1}}}{2} \Delta t \tag{24c}$$

$$\boldsymbol{\rho}_{k+1,k} = \frac{1}{2} \begin{bmatrix} 0 & W & -V & U \\ -W & 0 & U & V \\ V & -U & 0 & W \\ -U & -V & -W & 0 \end{bmatrix} \tag{24d}$$

$$\|\omega_{k+1,k}\| = \sqrt{U^2 + V^2 + W^2} \tag{24e}$$

The state transition matrix can now be defined in the discrete-time domain as

$$\boldsymbol{\Phi}_{k+1,k} = \mathbf{I}_4 cos\left(\frac{\|\omega_{k+1,k}\|}{2}\right) + \boldsymbol{\rho}_{k+1,k} \frac{2}{\|\omega_{k+1,k}\|} sin\left(\frac{\|\omega_{k+1,k}\|}{2}\right) \tag{25}$$

Finally, the discrete time STM formulation for quaternion propagation becomes

$$\mathbf{q}_{k+1} = \mathbf{\Phi}_{k+1,k}\mathbf{q}_k \tag{26}$$

The quaternion states obtained through the STM propagation will tend to denormalize over time as a result of errors in evaluating trignometric terms. As a result, it is recommended that the quaternion be normalized periodically to avoid round-off error accumulation. Note that for the case of $\|\omega\| = 0$, i.e., no rotation, the STM is singular, and the algorithm assigns $\mathbf{\Phi}_{k+1,k} = \mathbf{I}_4$ to prevent numerical instability.

V.C. Environmental Torque Evaluation for a 3U CubeSat

From the requirement to minimize computational overhead, it was essential to determine the extent to which perturbation torques had to be modelled. Though prior studies appeared to have established gravity gradient and atmospheric effect as the dominant perturbing torques, these studies focused on generic spacecraft and not specifically on the CubeSat geometry. A 3U CubeSat with an inertia tensor identical to that presented in Sec. V.F was simulated in a real-world environment (see Fig. 18 in the Appendix) while the on-board controller maintained an orientation of $[45^\circ \;\; 45^\circ \;\; 45^\circ]_B^{RPY}$ for a period of one day. The ratio of the perturbing torque magnitudes, T_{gg} and T_{aero}, was calculated at various orbital altitudes to allow for an order of magnitude analysis. Figure 2 presents the results of this study.

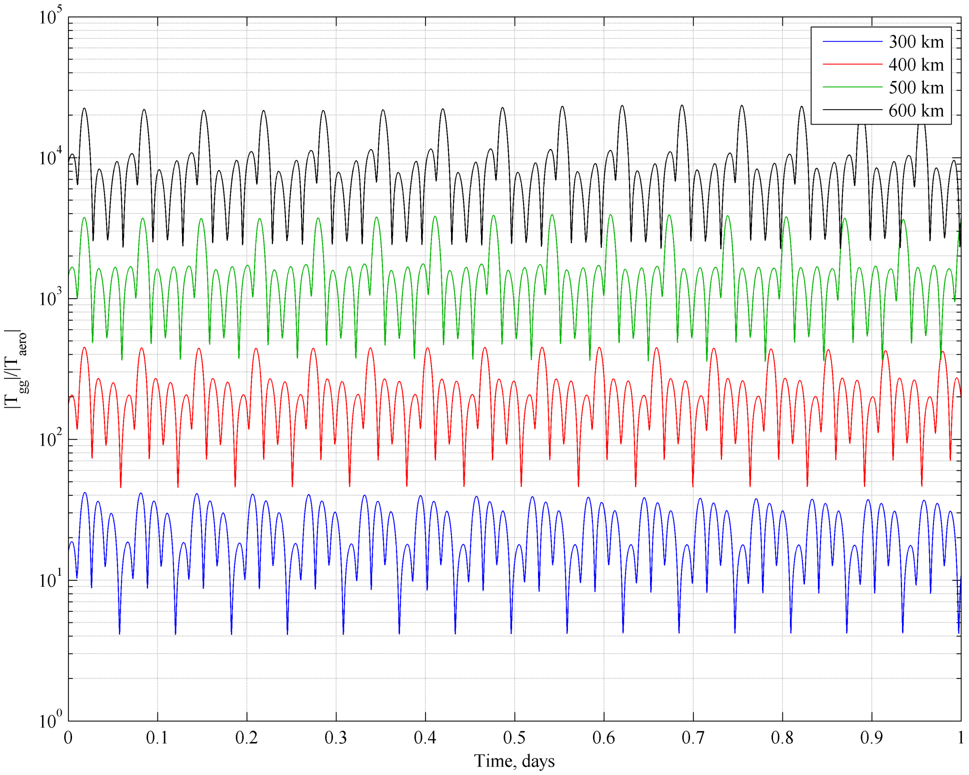


Figure 2. Magnitude comparison of dominant perturbing torques with varying orbit altitude for a 3U CubeSat in LEO

The order of magnitude analysis illustrates that for orbit altitudes of 400 km and higher, the gravity gradient torque $\vec{T}_{gg}$ is the dominant environmental perturbation affecting the spacecraft attitude. Therefore, in order to minimize computational cost, $\vec{T}_{gg}$ was the solitary environmental perturbation modeled within the optimization routine.

V.D. Quaternion Formulation of Gravity Gradient Torque

Initial attempts to implement gravity gradient perturbation torque using Euler angles into the optimization routine demonstrated severe numerical instability and non-convergent solutions. The trigonometric functions within the quaternion-to-DCM function were suspected to trigger these instabilities. An alternative formulation was therefore necessary. Junfeng et al.[55] utilize a technique to calculate the gravity gradient torque directly from the quaternion state and spacecraft position. This formulation is specifically for a spacecraft with principal moments of inertia, but could be easily extended to accommodate any inertia tensor.

$$T_{gg_x} = 3\frac{\mu_\oplus}{r^3}(I_{zz} - I_{yy})(2q_2q_1 - 2q_4q_3)(2q_3q_1 + 2q_4q_2) \tag{27a}$$

$$T_{gg_y} = 3\frac{\mu_\oplus}{r^3}(I_{xx} - I_{zz})(2q_3q_1 + 2q_4q_2)(2q_4^2 + 2q_1^2 - 1) \tag{27b}$$

$$T_{gg_z} = 3\frac{\mu_\oplus}{r^3}(I_{yy} - I_{xx})(2q_2q_1 - 2q_4q_3)(2q_4^2 + 2q_1^2 - 1) \tag{27c}$$

A similar formulation is incorporated by Bender[56] for the calculation of gravity gradient torque. However, Junfeng et al.[55] and Bender[56] both approximate the $\mu_\oplus/r^3$ term by the mean motion of the orbit, which is only a valid assumption for perfectly circular orbits. The formulation presented in Eq. (27) is the general case, where r is the magnitude of the position vector $\vec{r}_G$.

V.E. Validation Routine

Following the generation of torque-optimal trajectories, it was necessary to validate them in a real-world propagation environment. To accomplish this, the time history of torques, as generated by GPOPS, was passed to an open-loop controller running within the real-world propagator. Specifically, the open-loop control law is given by

$$\mathbf{u} = \mathbf{T}_{GPOPS} \tag{28}$$

where $\mathbf{u}$ is the control torque and $\mathbf{T}_{GPOPS}$ is the generated torque profile. This open loop controller is not provided any information regarding the target attitude state - it simply applies the commanded profile. Though a PID controller could be designed to follow the generated $\mathbf{q}$ and $\boldsymbol{\omega}$ profiles, such an implementation is beyond the scope of this research.

As the optimal trajectory is discretized at the Gauss nodes, the time intervals between the guidance 'waypoints' are not constant. However, the real-world simulator and typical On-Board Computer (OBC) modules operate at a constant time-step. It was therefore necessary to sample the guidance trajectory at a constant frequency of 10 Hz using the MATLAB `resample` function with linear interpolation.

V.F. Simulation Properties

A 3U CubeSat (refer Fig. 19 and Table 5 in the Appendix for spacecraft configuration and geometry) having a mass of 4.0 kg and an inertia tensor

$$\mathbf{J} = \begin{bmatrix} 0.059 & 0 & 0 \\ 0 & 0.114 & 0 \\ 0 & 0 & 0.126 \end{bmatrix} \text{kg} \cdot \text{m}^2$$

with initial Keplerian orbital elements

$$[a \quad e \quad i \quad \Omega \quad \omega \quad \nu]_0 = [6778 \text{ km} \quad 0.005 \quad 0.1^\circ \quad 270^\circ \quad 90^\circ \quad 0^\circ]$$

was considered for this study. This corresponds to a 366.1 km × 433.9 km orbit having a period of 92.6 minutes, with a low inclination of 0.1° to the equatorial plane of the Earth. The cost function for the optimization routine was defined in Lagrangian form as

$$J = \int_{t_0}^{t_f} \sqrt{u_x^2 + u_y^2 + u_z^2}\, \mathrm{d}t. \tag{29}$$

which is representative of the sum of the control torque vector magnitude over time. Table 2 presents the initial and final states for the guidance algorithm.

Parameter	Value	Unit
$\mathbf{r}_0$	$[6744.1 \quad -1.652\text{E-}12 \quad 11.77]_G$	km
$\mathbf{r}_f$	$[6700.3 \quad 769.03 \quad 11.69]_G$	km
$\mathbf{v}_0$	$[1.88\text{E-}15 \quad 7.707 \quad 8.237\text{E-}19]_G$	km·s^{-1}
$\mathbf{v}_f$	$[-8.757\text{E-}1 \quad 7.657 \quad 1.533\text{E-}3]_G$	km·s^{-1}
$\boldsymbol{\omega}_0$	$[0 \quad 0 \quad 0]_B$	rad·s^{-1}
$\boldsymbol{\omega}_f$	$[0 \quad 0 \quad 0]_B$	rad·s^{-1}
$\mathbf{q}_0$	$[0 \quad 0 \quad 0 \quad 1]$	–
$\mathbf{q}_f$	Case 1: $[0 \quad 0 \quad \sqrt{2}/2 \quad \sqrt{2}/2]$	–
	Case 2: $[0 \quad 0 \quad 1 \quad 0]$	–
Nodes	Case 1: 20	–
	Case 2: 60	–
t_0	0	sec
t_f	100	sec

Table 2: Summary of Initial and Final States for Optimization Routine

Case 1 and Case 2 in $\mathbf{q}_f$ correspond to a 90° and 180° rotation about the body z-axis respectively, while $\mathbf{q}_0$ represents an initial orientation aligned with the geocentric inertial frame. The node count for each case was selected iteratively based on numerical stability and the generation of a valid solution. The control torques were limited to 0.1 mN·m and the angular momentum limit set to 10.3 mN·m·s, which correspond to 16% and 100% of the relevant limits of the MAI-201 Miniature 3-Axis Reaction Wheel system.[57] The reduced torque capability was enforced to simulate a scenario with severely degraded reaction wheels, as discussed earlier in Sec. III. The selection of t_0 and t_f was based on a preliminary scaling of time-optimal and Eigenaxis results using the maneuver duration correlation in Eq. (30) as outlined by Bilimoria and Wie.[16]

$$t = \sqrt{\frac{\mathbf{J}_{ii}}{u_{max}}}\hat{t} \tag{30}$$

where $\hat{t}$ is the non-dimensional maneuver time and u_{max} is the maximum allowable control torque. A range of maneuver durations, presented in Table 3 was computed based on the minimum and maximum satellite inertia components.

Yaw angle	Maneuver type	t_{min} (sec.)	t_{max} (sec.)
90°	Eigenaxis	60.88	88.90
	Time-optimal	52.80	85.94
180°	Eigenaxis	86.11	125.8
	Time-optimal	78.77	115.1

Table 3: Estimated Maneuver Duration

Based on these estimates, a maneuver duration of 100 seconds was selected for the 90° and 180° yaw maneuvers. The possibility that the satellite may be incapable of performing a 180° rotation about the yaw axis within the enforced time limit is acknowledged. However, such a scenario tests the robustness of the optimal guidance routine to generate a valid trajectory when faced with maneuvers at or beyond its dynamic capabilities.

VI. Results and Discussion

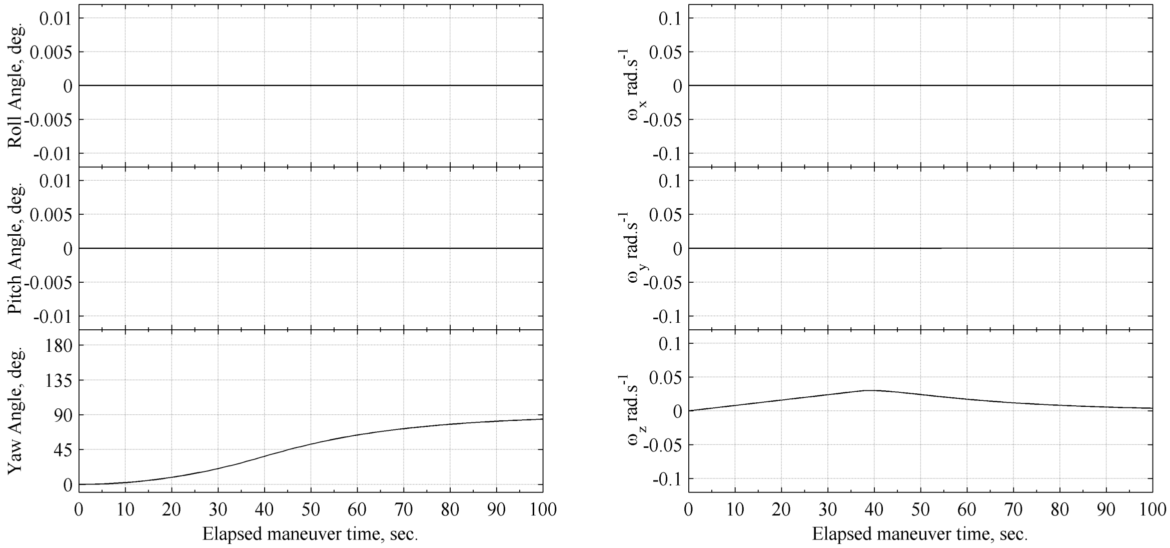
The trajectories generated by the torque-optimal guidance algorithm and associated validation results shall now be presented, along with the performance for Eigenaxis maneuvers in identical scenarios.

VI.A. Large-Angle Eigenaxis Maneuvers

Having defined the spacecraft maneuver parameters, the first step was to examine large-angle rest-to-rest Eigenaxis maneuvers for a satellite with the properties presented in Sec. V.F. This study evaluates the performance of an Eigenaxis controller for a 3U CubeSat equipped with realistic torque and angular momentum constraints. The real-world model was used alongside a basic PD controller with damping ratio $\zeta = 1$ and natural frequency $\omega_n = 0.13$.

90° Rest-to-rest Yaw Maneuver

Figure 3 presents the spacecraft orientation during a 90° rest-to-rest yaw maneuver. In 100 seconds, the Eigenaxis guidance and control law is able to yaw the spacecraft to an angle of 84.21°, undershooting the commanded attitude by 5.79°.



(a) RPY angles

(b) Angular rates

Figure 3. Attitude states: 90° Eigenaxis yaw maneuver, Δt_{mv} = 100 sec.

During this maneuver, the roll and pitch angles remain at 0.00°, indicating that the z-axis is indeed the Eigenaxis. At the end of the maneuver, the spacecraft has a small residual ω_z of 0.22 deg$\cdot$s^{-1}, indicating that the Eigenaxis maneuver was unable to fully satisfy the rest-to-rest constraint within the allotted maneuver duration. Figure 4 and Figure 5 present the actuator usage and trajectory followed by the body axis during the maneuver respectively.

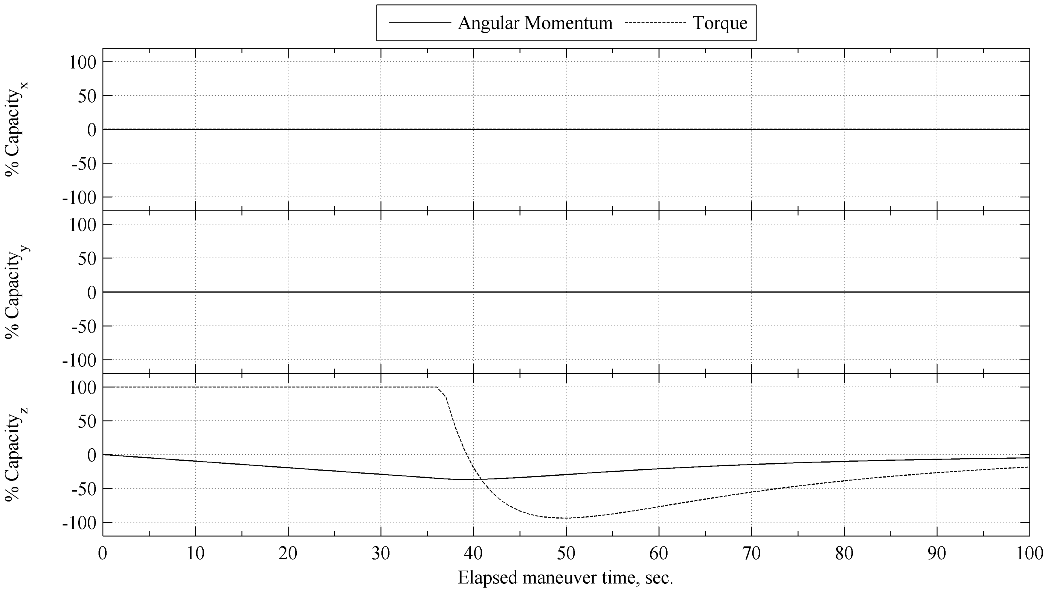


Figure 4. Actuator usage: 90° Eigenaxis yaw maneuver, Δt_{mv} = 100 sec.

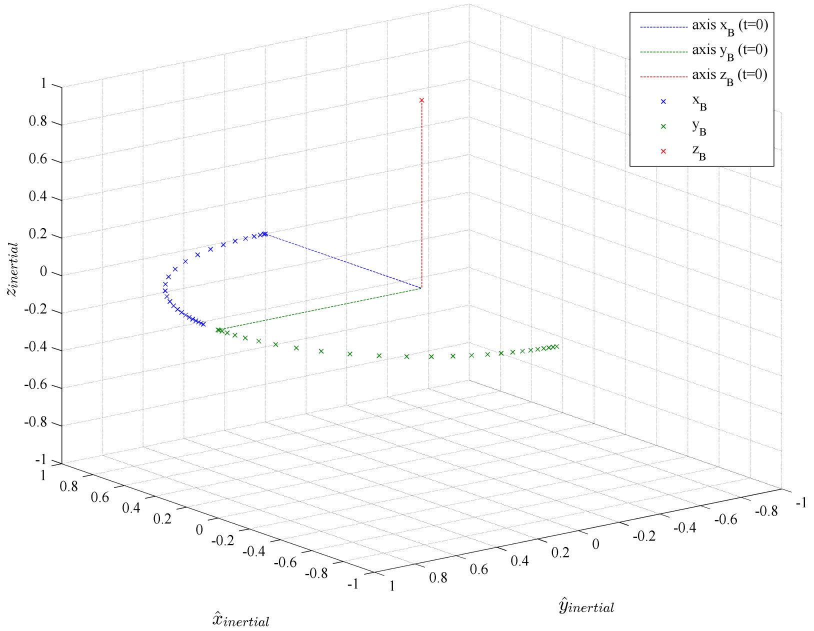
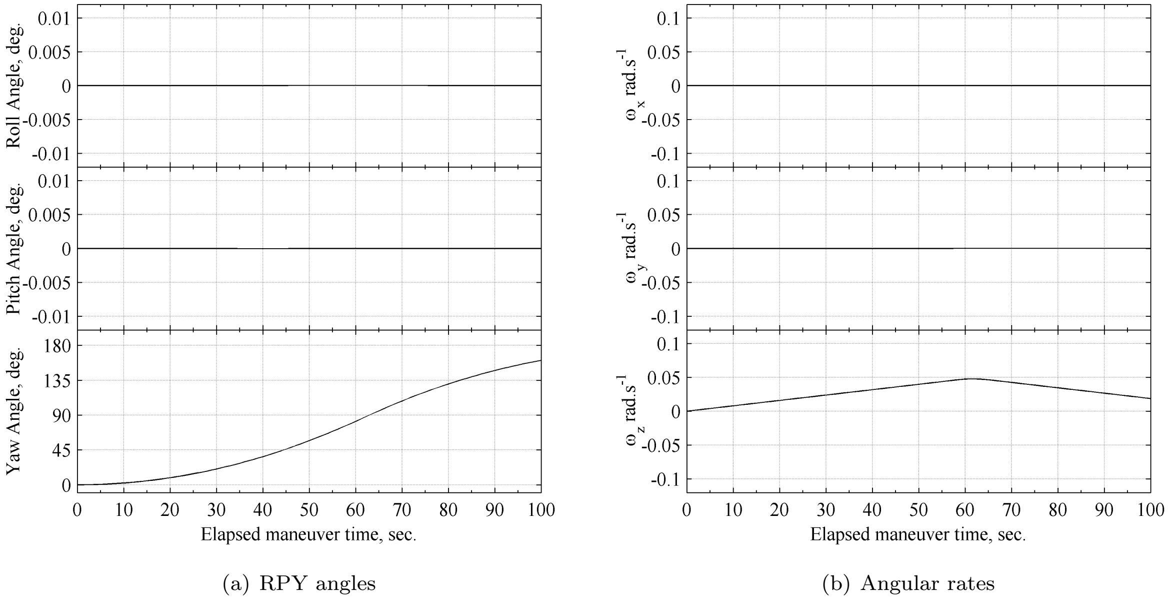


Figure 5. 3D Representation: 90° Eigenaxis yaw maneuver, Δt_{mv} = 100 sec.

From Fig. 4, we observe that rotating about the z-axis results in sole usage of the z-axis actuator. The maximum angular momentum usage during the maneuver was 38.9%. At the end of the maneuver, momentum usage in the z-axis accumulates to 4.76%. The trajectory in Fig. 5 verifies the use of the z-axis as the Eigenaxis, and illustrates that the spacecraft did not achieve the commanded 90° yaw angle.

180° Rest-to-rest Yaw Maneuver

Figure 6 presents the spacecraft orientation during a 180° yaw maneuver.



(a) RPY angles

(b) Angular rates

Figure 6. Attitude states: 180° Eigenaxis yaw maneuver, Δt_{mv} = 100 sec.

In 100 seconds, the Eigenaxis guidance and control law is able to yaw the spacecraft to an angle of 160.6°, undershooting the commanded attitude by 19.4°. Similar to the 90° Eigenaxis maneuver, the roll and pitch angles remain at 0.00°. At the end of the maneuver, the spacecraft has a residual ω_z of 1.07 deg·s^{-1}, which indicates that the Eigenaxis maneuver did not satisfy the rest-to-rest constraint. Figure 7 and Figure 8 present the actuator usage and trajectory followed by the body axis during the maneuver respectively.

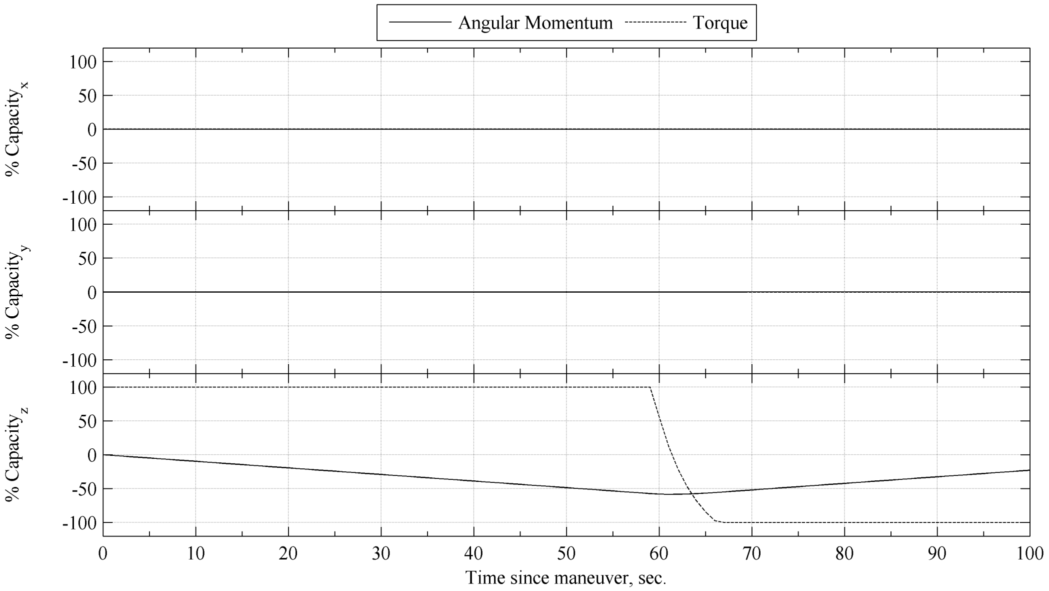


Figure 7. Actuator usage: 180° Eigenaxis yaw maneuver, Δt_{mv} = 100 sec.

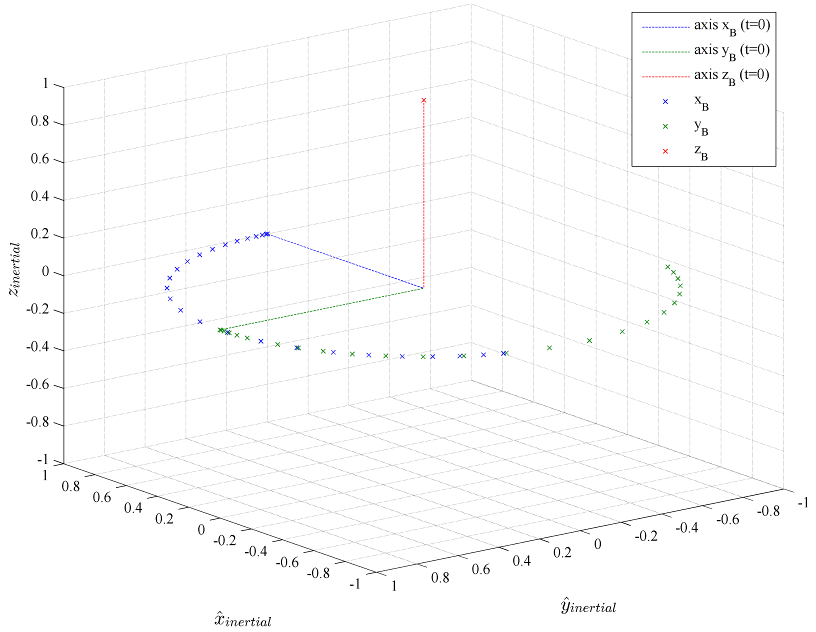


Figure 8. 3D Representation: 180° Eigenaxis yaw maneuver, Δt_{mv} = 100 sec.

Figure 8 confirms the use of the z-body axis as the Eigenaxis during the maneuver. The maximum angular momentum usage during the maneuver was 58.4%. At the end of the maneuver, momentum usage about the z-axis was 22.8%. The trajectory indicates that the spacecraft did not achieve the commanded attitude in the specified maneuver duration, despite significant utilization of the available torque and angular momentum.

VI.B. Torque-Optimal Solutions and Validation

Having examined the performance of an Eigenaxis guidance law, we now present the torque-optimal guidance formulation for the 90° and 180° yaw maneuvers. It should be noted that the trajectories presented in this section are not representative of global minima for the associated torque cost-function, but rather of local minima for the specified maneuver duration.

90° Rest-to-rest Yaw Maneuver

Figure 9 presents the attitude states and control variables generated by the torque-optimal guidance routine for a 90° yaw maneuver.

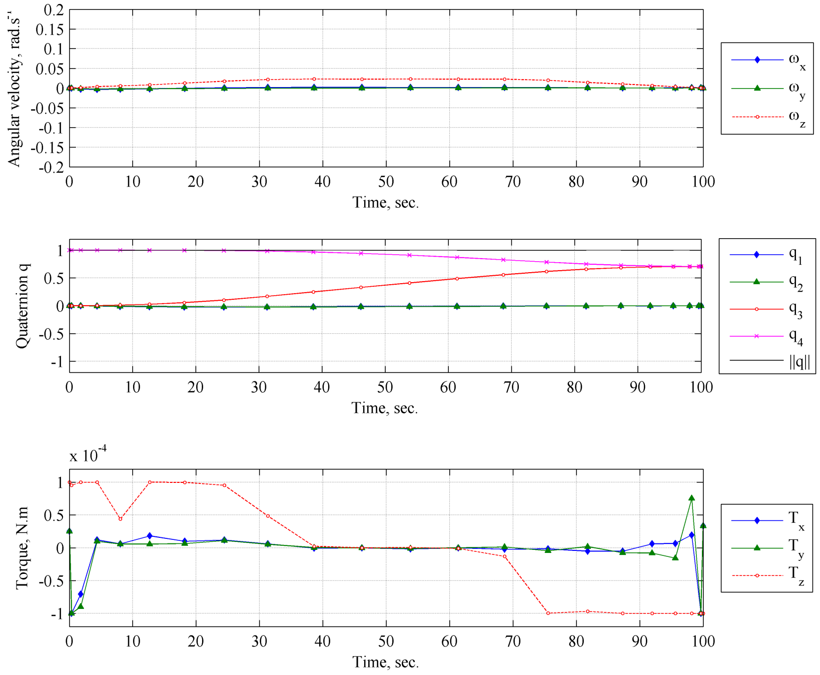
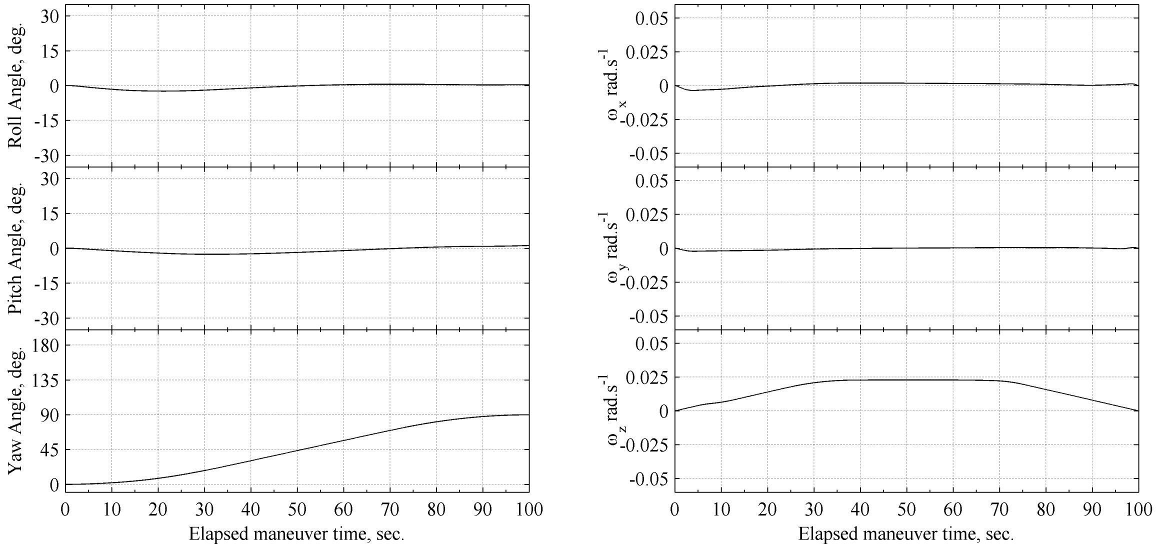


Figure 9. Time history: 90° yaw torque-optimal trajectory

The torque-optimal guidance routine generates a non-linear path from the initial to the final attitude states in quaternion space, using internal and external disturbance torques to minimize the torque-cost function. By utilizing internal and external torques produced during the rotation, and implementing a "look-ahead" approach, the algorithm reduces actuator usage. Figure 10 presents the spacecraft orientation while following the guidance torque profile for the 90° yaw maneuver.



(a) RPY angles

(b) Angular rates

Figure 10. Attitude states: 90° torque-optimal yaw maneuver validation

By following the trajectory described by the torque-optimal guidance algorithm, the spacecraft achieves a yaw angle of 89.88° within the specified maneuver duration. The associated roll and pitch angles are 0.33° and 1.16° respectively. The torque-optimal guidance routine therefore achieves greater yaw-pointing accuracy than an Eigenaxis controller, but introduces small errors in roll and pitch. At the end of the maneuver, the angular rates about all three axis are <0.01 deg·s^{-1}. The torque-optimal guidance law thus satisfies the rest-to-rest constraint for the maneuver, unlike an Eigenaxis controller. Figure 11 and Figure 12 present the actuator usage and maneuver trajectory respectively.

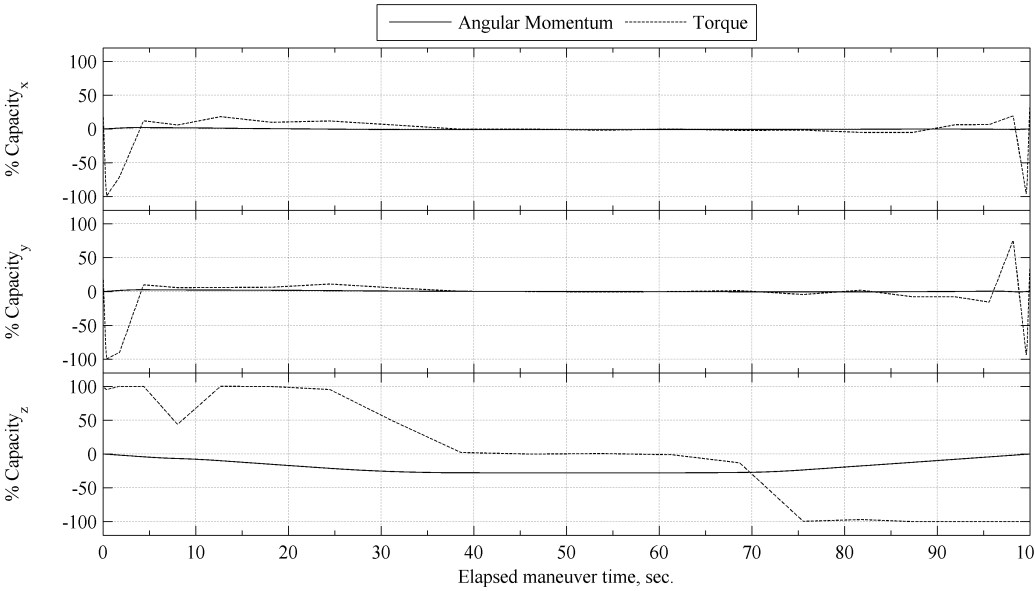


Figure 11. Actuator usage: 90° torque-optimal yaw maneuver validation

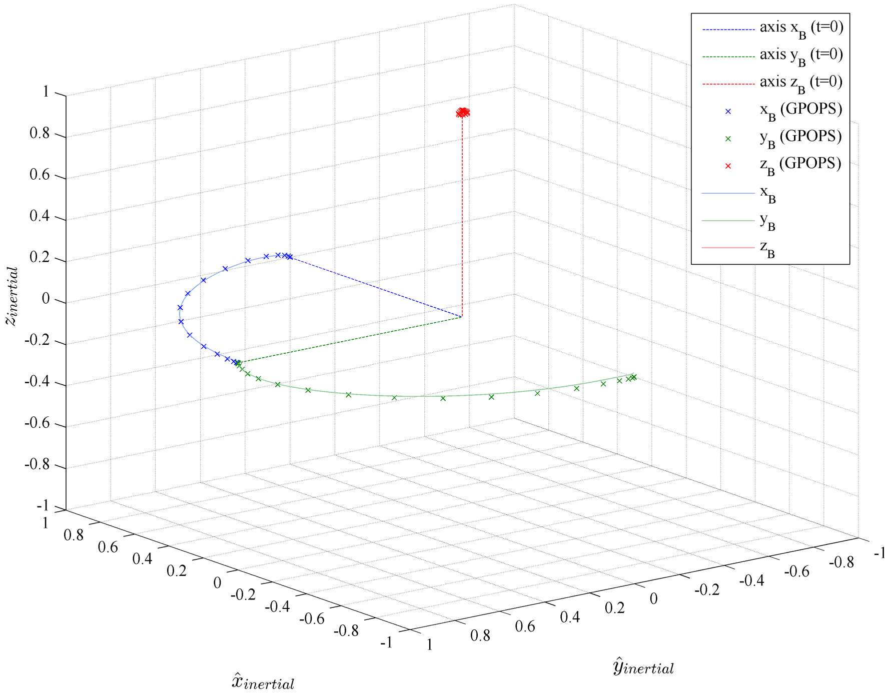


Figure 12. 3D Representation: 90° torque-optimal yaw maneuver validation

From Fig. 11, we observe the utilization of actuators about all three axes. The maximum angular momentum usage during the maneuver is 2.08%, 2.49%, and 27.9% in the x-, y-, and z-axis respectively. At the end of the maneuver, accumulated momentum usage is <0.1% about each axis. This is a significant improvement over the actuator usage for the equivalent Eigenaxis maneuver, presented in Fig. 4. By utilizing the nutational torque components, the guidance algorithm significantly reduces torque usage about the z-axis, with minimal additional momentum usage in the x- and y-axis. The spacecraft closely follows the torque-optimal guidance waypoints, as seen in Fig. 12. For a 90° yaw maneuver within the specified duration, the torque-optimal solution is near-Eigenaxis. The minor deviations add small nutational components based on the inertia tensor of an asymmetric spacecraft, effectively increasing the torque available about each axis.

For comparison purposes, when the 100-second time limit was removed, the Eigenaxis controller required 150 seconds to achieve a comparable yaw angle and angular rates – significantly longer than the torque-optimal routine. The Eigenaxis control scheme however did not introduce any measurable pointing errors about the roll and pitch axes.

180° Rest-to-rest Yaw Maneuver

Figure 13 presents the attitude states and control variables generated by the torque-optimal guidance routine for a 180° yaw maneuver.

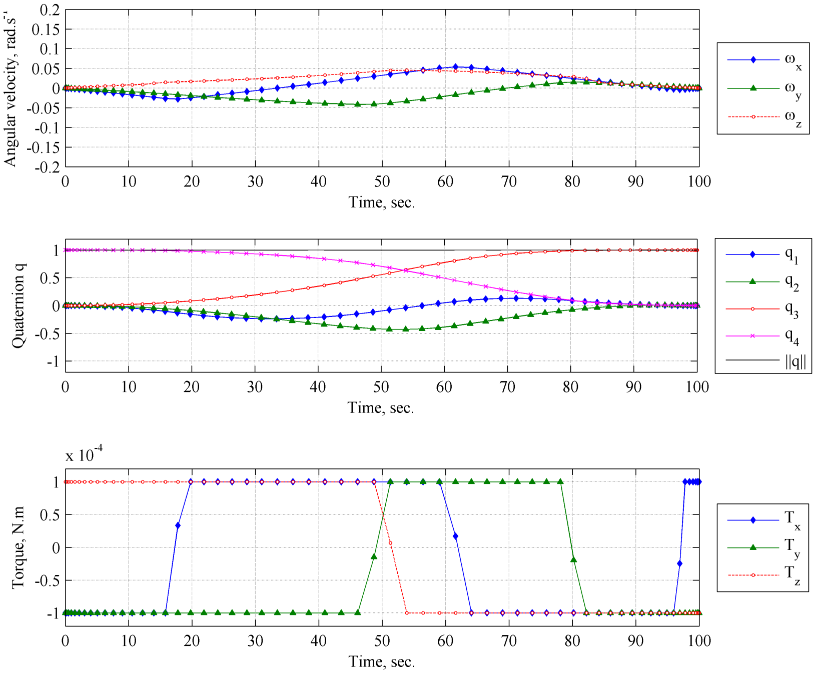
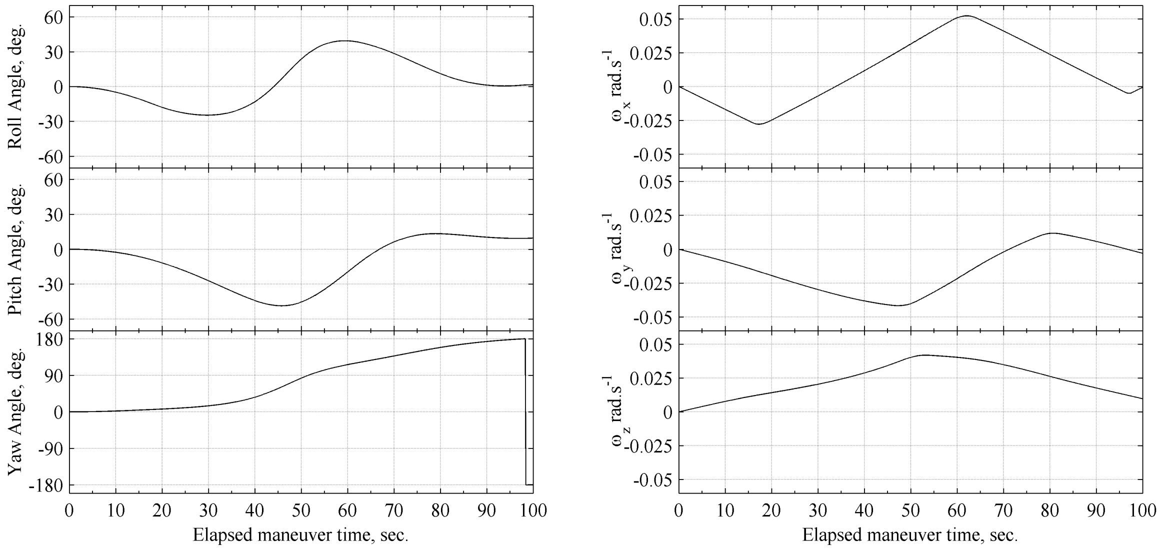


Figure 13. Time history: 180° yaw torque-optimal trajectory

As in the case of the 90° yaw maneuver, the torque-optimal guidance routine generates a non-linear path from the initial to the final attitude states in quaternion space. However, unlike the 90° case, the actuators continuously apply torque about each axis for the duration of the maneuver. This is due to the larger angular rates required to accomplish the maneuver within the specified time. Figure 14 presents the spacecraft orientation while following the guidance waypoints for the 180° yaw maneuver.



(a) RPY angles

(b) Angular rates

Figure 14. Attitude states: 180° torque-optimal yaw maneuver validation

Following the trajectory described by the torque-optimal guidance algorithm, the spacecraft achieves a yaw angle of -179.1° within the specified maneuver duration. This corresponds to a yaw-pointing error of 0.9°, which is significantly lower than that achieved in the Eigenaxis case. The associated roll and pitch angles are 1.56° and 9.56° respectively. The torque-optimal guidance routine therefore achieves greater yaw-pointing accuracy than an Eigenaxis controller, but introduces moderate errors in roll and pitch. At the end of the maneuver, ω_x, ω_y, and ω_z are 0.04, 0.17 and 0.56 deg·s^{-1} respectively. Though the rest-to-rest constraint is not satisfied, ω_z is nearly half that achieved by the Eigenaxis controller, underlining the robustness of the guidance algorithm. Figure 15 and Figure 16 present the actuator usage and maneuver trajectory respectively.

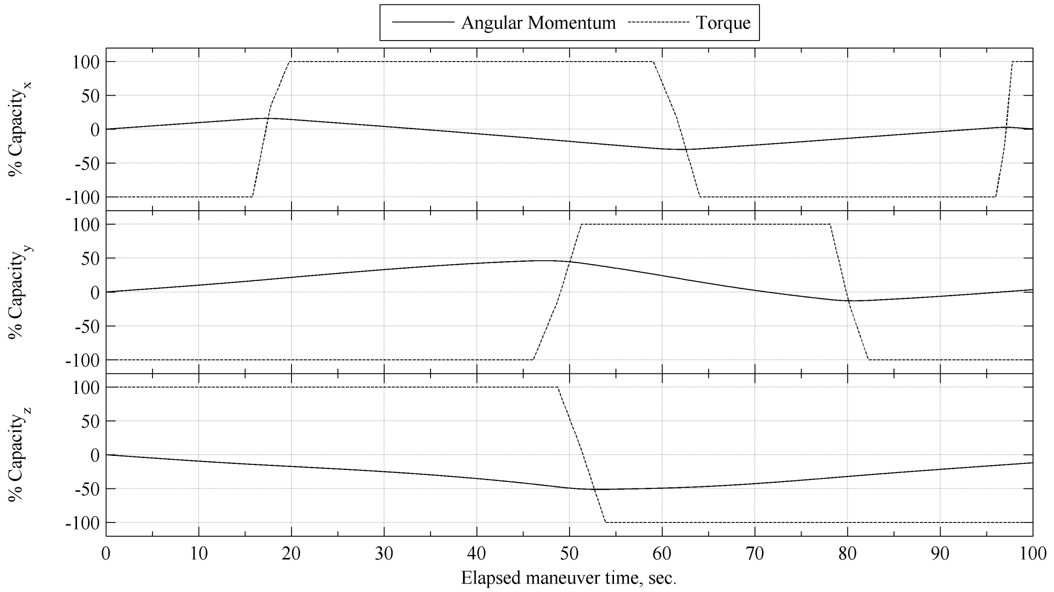


Figure 15. Actuator usage: 180° torque-optimal yaw maneuver validation

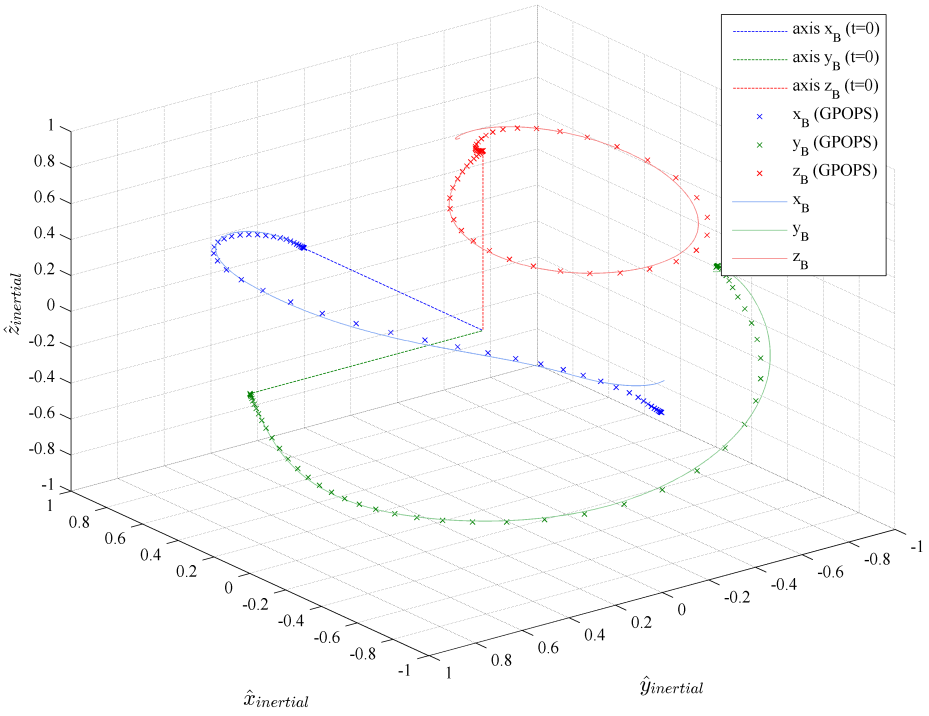


Figure 16. 3D Representation: 180° torque-optimal yaw maneuver validation

In Fig. 15, we observe the utilization of actuators about all three axes. The maximum angular momentum usage during the maneuver is 29.9%, 46.1%, and 51.2% in the x-, y-, and z-axis respectively. By the end of the maneuver, accumulated momentum usage is 0.35%, 3.32%, and 11.9% about these axes respectively. By utilizing the nutational torque components, the guidance algorithm significantly reduces torque usage about the z-axis, with moderate additional momentum usage in the x- and y-axis.

For a 180° yaw maneuver within the specified duration, the torque-optimal solution is clearly non-Eigenaxis. The z-axis deviates significantly from the vertical, tracing a path resembling a cardioid. As a result, both the x- and y-axis deviate from the perfect arcs seen for the Eigenaxis maneuver in Fig. 8. It can be deduced that the axis of rotation is not constant – a characteristic typical of both time- and torque-optimal guidance. The algorithm clusters the Gauss points at the beginning and end of the trajectory in an effort to accurately capture the large rates of change near these temporal nodes. The spacecraft closely follows the torque-optimal guidance waypoints for the majority of the maneuver, as seen in Fig. 16. However, as the spacecraft nears the endpoint, the control system is unable to fully counter the gyroscopic torques, resulting in a deviation from the predicted trajectory.

For comparison purposes, when the 100-second time limit was removed, the Eigenaxis controller required 170 seconds to achieve a comparable yaw angle and angular rates – significantly longer than the torque-optimal routine. The Eigenaxis control scheme however did not introduce any measurable pointing errors about the roll and pitch axes.

VI.C. Computational Cost

The results presented in Sec. VI.B indicate that the trajectory generated by the torque-optimal guidance algorithm can be closely followed by an open-loop controller in a real-world environment. However, it is necessary to consider the CPU run-time required to generate these trajectories. Also important is the time required to generate the orbital states required by the guidance algorithm using the on-board propagator. Table 4 summarizes the algorithm run time on two different processors, averaged over ten executions.

Processor specifications were obtained from official Intel documentation[a,b].

					Run Time (sec.)		
Yaw Angle	Nodes	Processor	GFLOP	RAM	*Propagator*	*Guidance*	*Total*
90°	20	i7 2700K	112	16 GB	0.267	5.134	5.401
		Core2Duo T7700	19.2	4 GB	0.690	9.531	10.22
180°	60	i7 2700K	112	16 GB	0.267	63.72	63.99
		Core2Duo T7700	19.2	4 GB	0.690	114.4	115.1

Table 4: CPU Run Times for Torque-optimal Trajectory Generation

The on-board propagator requires <1 second to compute the initial and final states required by the guidance algorithm. The CPU time required for on-board propagation is independent of the node count, and scales with maneuver duration - in this case, 100 seconds. From Table 4, the guidance routine is clearly the dominant computational load, utilizing 92% - 99.6% of the total run time. Computation times are similar to those presented by Boyarko et al. for time-optimal trajectory generation using fifth-order and seventh-order IDVD methods,[43] further supporting the validity of the results.

In order to put the run times presented in Table 4 into perspective, the computational power available on a Snapdragon 800 System on Chip (SoC), which powers the Google Nexus 5[c], is outlined. Equipped with a 2.26 GHz Quad-core Krait 400 CPU and a Qualcomm® Adreno$^{\text{TM}}$ 330 GPU, this SoC is capable of 129.6 GFLOPS. It uses the ARM architecture, in contrast to the x86-64 architecture in modern-day laptops and desktops. ARM does not support hyperthreading, but has the advantage of operating at lower temperatures and with lower power consumption than mainstream processors. Various missions have established precedence regarding the integration of a smartphone SoC on board a CubeSat. In 2013, STRaND-1 was launched into orbit, carrying a classic CubeSat computer and a Google Nexus One smartphone[d]. Additionally, as part of the Small Spacecraft Technology Program, NASA has launched five PhoneSats into LEO, the most recent in April 2014 aboard a SpaceX Falcon IX launch vehicle[e]. These missions demonstrated the ability to successfully incorporate smartphone technology into CubeSats while providing a significant improvement in computational power over existing CubeSat processors.

Since optimal guidance is a predictive procedure, not real-time, the guidance algorithm would be executed off-line prior to the planned maneuver, and the resulting trajectory data stored until required by the controller. Therefore, the algorithm would not continuously utilize valuable computational power which is necessary for scientific observations.

VII. Conclusion

This torque-optimal guidance algorithm presents a significant step towards the development of intelligent GN&C systems for small satellites. Simulation results indicate that the torque-optimal guidance algorithm outperforms an Eigenaxis control law in both yaw-pointing accuracy and angular momentum usage for identical maneuvers, with low to moderate computational overhead. For larger spacecraft, the increased size and the presence of structural booms for antennae and instrumentation would necessitate the inclusion of atmospheric and magnetic torques within the guidance routine. Given the modular structure of the algorithm, these additional perturbations could be integrated as future improvements to the optimization code.

[a] Intel Inc., Intel®Core2Duo$^{\text{TM}}$Processor T7000 Series, 2011 (accessed September 12, 2014).
URL: http://download.intel.com/support/processors/core2duo/sb/core_T7000.pdf

[b] Intel Inc., Intel®i7$^{\text{TM}}$2700 Desktop Processor Series, 2012 (accessed Sept. 19, 2014).
URL: http://download.intel.com/support/processors/corei7/sb/core_i7-2700_d.pdf

[c] Google Inc., Nexus 5 Technical Specifications. 2014 (accessed Sept. 21, 2014), http://www.google.ca/nexus/5

[d] Surrey Satellite Technology Limited, STRaND-1 Smartphone Nanosatellite, 2014 (accessed May 6, 2014), http://www.sstl.co.uk/Missions/STRaND-1–Launched-2013/STRaND-1/

[e] Hall, L. and Dunbar, B., Small Spacecraft Technology Program, 2014 (accessed May 15, 2014), http://www.nasa.gov/directorates/spacetech/small_spacecraft/#.U4KVjPldXXQ

Appendix

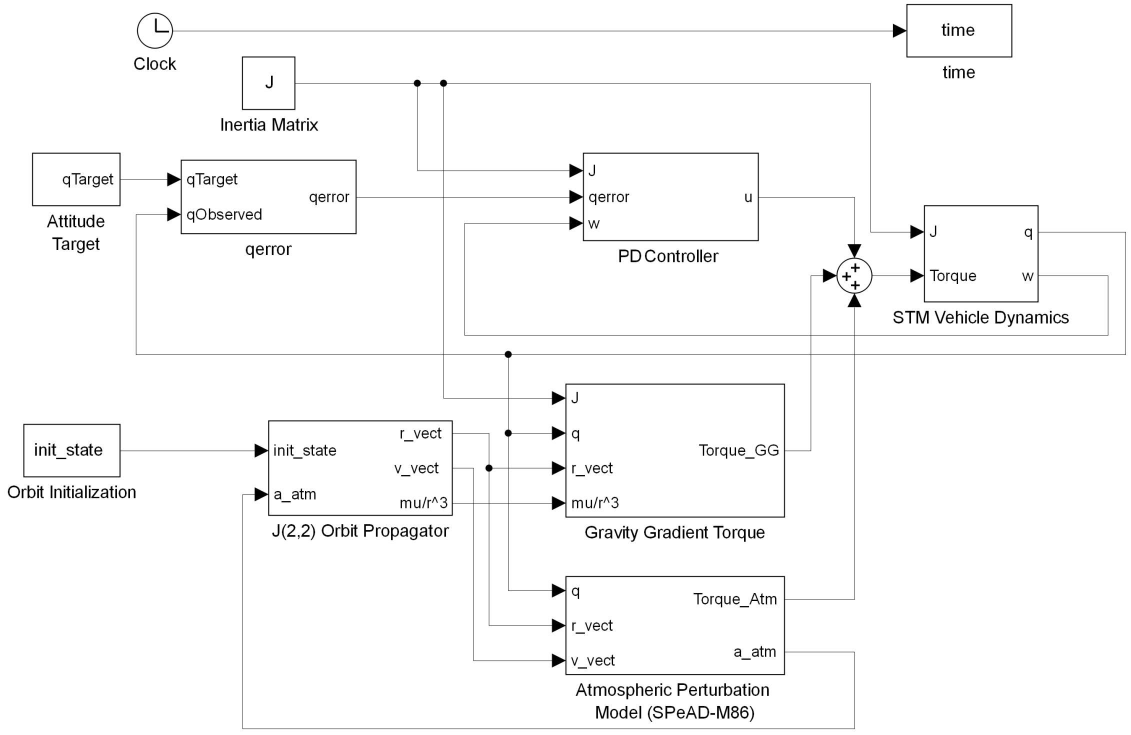


Figure 17. Top-level Simulink Block Diagram of Onboard Propagator

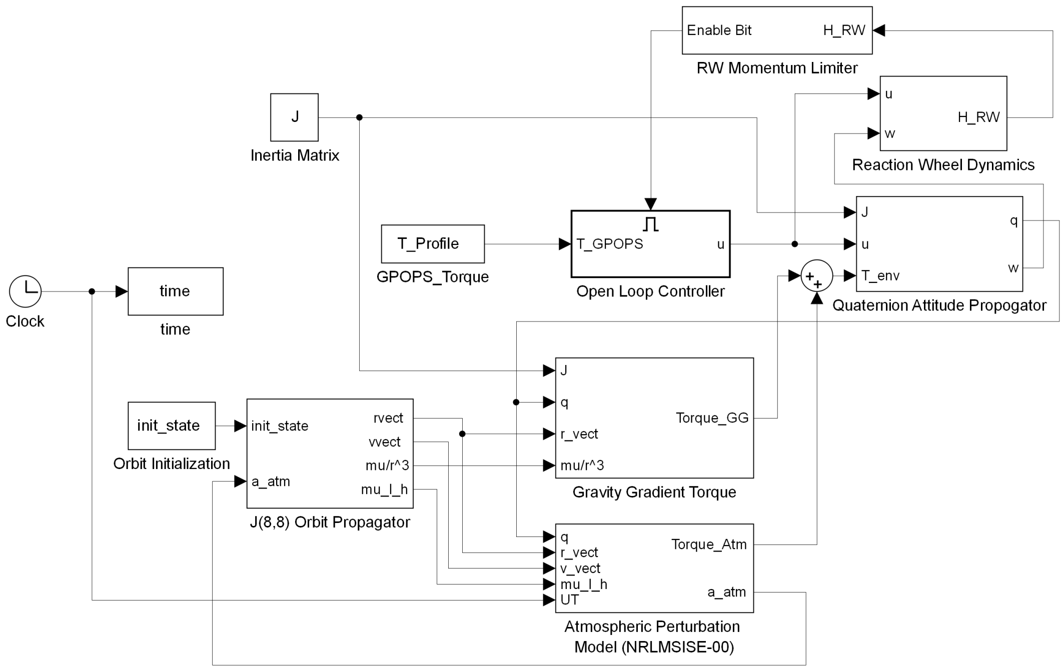


Figure 18. Top-level Simulink Block Diagram of Real-world Propagator for Optimal Trajectory Validation

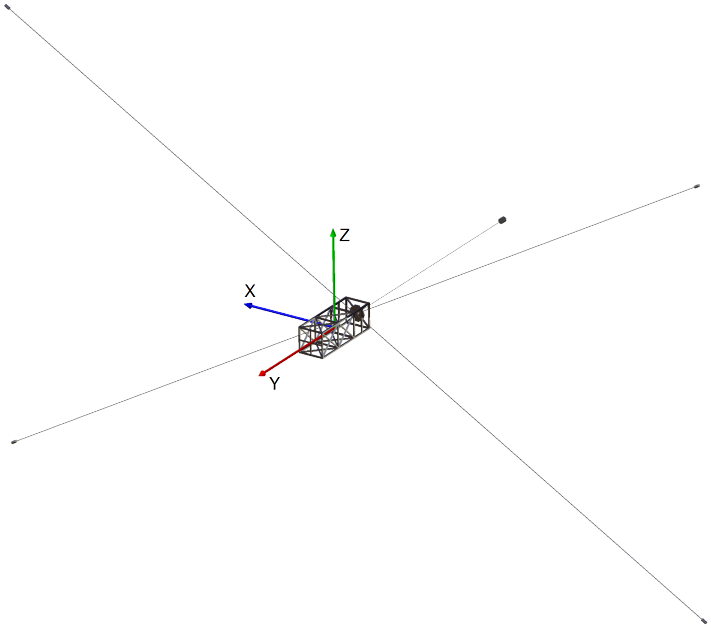


Figure 19. 3U CubeSat configuration with deployed gravity gradient booms

Positive roll, pitch, and yaw correspond to counterclockwise rotations about the x-, y-, and z-axis respectively. The 6 faces of the CubeSat and their centers of pressure were individually defined in the body coordinate frame as area vectors in order to account for atmospheric perturbations. Table 5 provides the mathematical definition of the satellite geometry.

Panel ID	Label	Area, A (m^2)	Area normal vector, $\hat{A}$	C_P location, $\vec{r}_{cp}$
1	Starboard panel	0.03	$+\hat{x}$	$+(b/2)\hat{x} - (l/3000)\hat{y}$
2	Fore panel	0.01	$+\hat{y}$	$+(l/2)\hat{y}$
3	Zenith panel	0.03	$+\hat{z}$	$-(l/3000)\hat{y} + (h/2)\hat{z}$
4	Port panel	0.03	$-\hat{x}$	$-(b/2)\hat{x} - (l/3000)\hat{y}$
5	Aft panel	0.01	$-\hat{y}$	$-(l/2)\hat{y}$
6	Nadir panel	0.03	$-\hat{z}$	$-(l/3000)\hat{y} - (h/2)\hat{z}$

Table 5: CubeSat Geometry Definition

The dimensions of the CubeSat are $l = 0.03$ m, $b = 0.01$ m, and $h = 0.01$ m.

Acknowledgments

The authors would like to thank Dr. Anil V. Rao and his team for their insights into the internal workings and development of GPOPS. They also thank Dr. Philip Gill at the University of California, San Diego for his assistance in obtaining the SNOPT solver.

References

[1] Wertz, J. R. and Larson, W. J., *Space Mission Analysis and Design, 3rd ed.*, Microcosm and Springer, Hawthorne, CA, 2008, pp. 354–380.

[2] Proper, I., *Reaction Wheel Design, Construction and Qualification Testing*, Master's thesis, York University, Toronto, Canada, 2010.

[3] Jan, Y. W., and Chiou, J. C., "Attitude Control System for ROCSAT-3 Microsatellite: A Conceptual Design," *Acta Astronautica*, Vol. 56, No. 4, 2005, pp. 439–452.

[4] Candini, G. P., Piergentili, F., and Santoni, F., "Miniaturized Attitude Control System for Nanosatellites," *Acta Astronautica*, Vol. 81, 2005, pp. 325–334.

[5] Ovchinnikov, M., Penkov, V., Norberg, O., and Barabash, S., "Attitude Control System for the First Swedish Nanosatellite MUNIN," *Acta Astronautica*, Vol. 46, No. 2, 2000, pp. 319–326.

[6] Martinelli, M. I., and Peña, R. S. S., "Passive 3 Axis Attitude Control of MSU-1 Pico-satellite," *Acta Astronautica*, Vol. 56, No. 5, 2005, pp. 507–517.

[7] Grassi, M. and Pastena, M., "Minimum Power Optimum Control of Microsatellite Attitude Dynamics," *Journal of Guidance, Control and Dynamics*, Vol. 23, No. 5, 2000, pp. 798–804.

[8] Grayzeck, E., *National Space Science Data Center COSPAR ID: 2013-072H*, NASA, 2013 (accessed March 24, 2014), http://nssdc.gsfc.nasa.gov/nmc/spacecraftDisplay.do?id=2013-072H.

[9] Hughes, P. C., *Spacecraft Attitude Dynamics*, John Wiley and Sons, New York, 1986.

[10] Wie, B., *Space Vehicle Dynamics and Control*, AIAA Education Series, Ohio, 1998.

[11] Platt, D., "A Monopropellant Milli-Newton Thruster System for Attitude Control of Nanosatellites," *16th Annual AIAA/USU Conference on Small Satellites, SSC02-III-5*, Logan, Utah, August 2002.

[12] G.E. Mueller, C. W. Mathews, a., *National Space Science Data Center COSPAR ID:1966-020A*, NASA, 2013 (accessed March 30, 2014), http://http://nssdc.gsfc.nasa.gov/nmc/spacecraftDisplay.do?id=1966-020A.

[13] Bedrossian, N., Bhatt, S., Lammers, M., Nguyen, L., and Zhang, Y., "First Ever Flight Demonstration of Zero Propellant Maneuver Attitude Control Concept," *Proc. 2007 AIAA GN&C Conf.*, Hilton Head, SC, Aug. 20–23 2007, pp. 1–12.

[14] Bedrossian, N., Bhatt, S., Lammers, M., and Nguyen, L., "Zero Propellant Maneuver Flight Results for 180° ISS Rotation," *Proc. 2007 Int. Symp. Space Flight Dynamics*, Annapolis, MD, Sept. 24–28 2007, pp. 1–10, NASA/CP-2007-214158.

[15] Bedrossian, N., Bhatt, S., and Ross, I. M., "Zero Propellant Maneuver Guidance: Rotating the International Space Station," *IEEE Control Systems Magazine*, Vol. 29, No. 5, 2009, pp. 53–73.

[16] Bilimoria, K. D., and Wie, B., "Time-Optimal Three-Axis Reorientation of Rigid Spacecraft," *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 3, 1993, pp. 446–452.

[17] Proulx, R., and Ross, I. M., "Time-Optimal Reorientation of Asymmetric Rigid Bodies," *Advances in the Astronautical Sciences*, Vol. 109, 2001, pp. 1207–1227.

[18] Xiaoli, B., and Junkins, J. L., "New Results for Time-Optimal Three-Axis Reorientation of a Rigid Spacecraft," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 4, July - Aug 2009, pp. 1071–1076.

[19] Karpenko, M., Bhatt, S., Bedrossian, N., Fleming, A., and Ross, I. M., "First Flight Results on Time-Optimal Spacecraft Slews," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 2, March - April 2012, pp. 367–376.

[20] Coon, T. R., and Irby, J. E., "Skylab Attitude Control System," *IBM Journal of Research and Development*, Vol. 20, No. 1, 1976, pp. 58–66.

[21] Chubb, W. B., and Seltzer, S. M., "Skylab Attitude and Pointing Control System," Tech. rep., Feb. 1971, NASA TN D-6068.

[22] Powell, B. K., "Gravity Gradient Desaturation of a Momentum Exchange Attitude Control System," *AIAA Guidance, Control and Flight Mechanics Conference*, Hempstead, NY, August 16-18 1971, AIAA Paper 71-940.

[23] Tong, D., "Spacecraft Momentum Dumping Using Gravity Gradient," *Journal of Spacecraft and Rockets*, Vol. 35, No. 5, Sept-Oct 1998, pp. 714–717.

[24] Chamitoff G. E., Dershowitz, A. L., and Bryson, A. L., "Command Level Maneuver Optimization for the International Space Station," *Advances in the Astronautical Sciences*, Vol. 104, 2000, pp. 311–326, AAS Paper 00-027.

[25] Bedrossian, N., Metzinger, R., and Adams, N., "Centralized Momentum Management," *Draper Lab Presentation*, January 31 1996.

[26] Pietz, J. and Bedrossian, N., "Momentum Dumping Using Only CMGs," *AIAA GN&C Conference*, Austin, TX, 2003.

[27] Euler, L., "Elementa Calculi variationum," *Originally published in Novi Commentarii academiae scientiarum Petropolitanae 10*, 1766, pp. 51–93.

[28] Garg, D., Patterson, M. A., Hager, W. W.,Rao, A. V.,Benson, D. A., and Huntington, G. T., "An Overview of Three Pseudospectral Method for the Numerical Solution of Optimal Control Problems," *Advances in the Astronautical Sciences*, Univelt Inc., San Diego, 2010, pp. 475–487.

[29] Elnagar, G., Kazemi, M., and Razzaghi, M., "The Pseudospectral Legendre Method for Discretizing Optimal Control Problems," *IEEE Transactions on Automatic Control*, Vol. 40, No. 10, October 1995, pp. 1793–1796.

[30] Fahroo, F. and Ross, I. M., "Costate Estimation by a Legendre Pseudospectral Method," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 2, 2001, pp. 270–277.

[31] Benson, D. A., *A Gauss Pseudospectral Transcription for Optimal Control*, Ph.D. thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, November 2004.

[32] Benson, D. A., Huntington, G. T., Thorvaldsen, T. P., and Rao, A. V., "Direct Trajectory Optimization and Costate Estimation via an Orthogonal Collocation Method," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 6, 2006, pp. 1435–1440.

[33] Huntington, G. T., *Advancement and Analysis of a Gauss Pseudospectral Transcription for Optimal Control*, Ph.D. thesis, Department of Aeronautics and Astronautics,Massachusetts Institute of Technology, 2007.

[34] Mathworks Inc., *Optimization Toolbox^{TM}User's Guide for R2013b*, September 2013.

[35] Ross, M., *User's Manual For DIDO: A MATLAB^{TM} Application Package for Solving Optimal Control Problems*, Naval Postgraduate School, Monterey, CA, February 2004, Version PR.1.

[36] Holmström, K., Martinsen, F., and Edvall, M., *User's Guide for TOMLAB/SOCS*, Tomlab Optimization Inc.

[37] Rao, A. V., *User's Manual for GPOPS Version 4.x: A MATLAB(R) Software for Solving Multiple-Phase Optimal Control Problems Using hp-Adaptive Pseudospectral Methods*, 2011.

[38] Ross, I. M. and D'Souza, C. N., "A Hybrid Optimal Control Framework for Mission Planning," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 4, July 2005, pp. 686–697.

[39] Rea, J. R., *A Legendre Pseudospectral Method for Rapid Optimization of Launch Vehicle Trajectories*, Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, 2001.

[40] Ross, I. M., and Karpenko, M., "A Review of Pseudospectral Optimal Control: From Theory to Flight," *Annual Reviews in Control*, Vol. 36, 2012, pp. 182–197.

[41] Gill, P. E., Murray, W., and Saunders, M. A., *User's Guide for SNOPT Version 7: Software for Large-Scale Nonlinear Programming*, Department of Mathematics University of California, San Diego, La Jolla, CA.

[42] Gong, Q., Kang, W., Bedrossian, N., Fahroo, F., Sekhavat, P., and Bollino, K., "Pseudospectral Optimal Control for Military and Industrial Applications," *46th IEEE Conference on Decision and Control*, New Orleans, LA, Dec. 2007, pp. 4128–4142.

[43] Boyarko, G. A., Romano, M., and Yakimenko, O. A., "Time-Optimal Reorientation of a Spacecraft Using an Inverse Dynamics Optimization Method," *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 4, July-August 2011, pp. 1197–1208.

[44] Votel, R. and Sinclair, D., "Comparision of Control Moment Gyros and Reaction Wheels for Small Earth-Observing Satellite," *26th Annual AIAA/USU Conference on Small Satellites*, Logan, UT, 13-16 August 2012, SSC12-X-1.

[45] NASA, *NASA Ends Attempts to Fully Recover Kepler Spacecraft, Potential New Missions Considered*, 15 August 2013 (accessed March. 10, 2015), Release: M13-58.

[46] Hur-Diaz, S., Wirzburger, J., and and Smith, D., "Three Axis Control of the Hubble Space Telescope Using Two Reaction Wheels and Magnetic Torquer Bars for Science Observations," *F. Landis Markley Astronautics Symposium*, Cambridge, MD, 29 June - 2 July 2008, Report No. AAS-08-279.

[47] Kawaguchi, J., Kominato, T., and Shirakawa, K., "Attitude Control Flight Experience: Coping with Solar Radiation and Ion Engines Leak Thrust in Hayabusa (MUSES-C)," *20th ISSFD Conference*, Annapolis, MD, 24-28 September 2007.

[48] Fleming, A., *Real-time Optimal Slew Maneuver Design and Control*, Master's thesis, U.S. Naval Postgraduate School, Monterey, CA, 2004.

[49] Huntington, G. T., and Rao, A. V., "Optimal Configuration of Spacecraft Formations via a Gauss Pseudospectral Method," *2005 AAS/AIAA Astrodynamics Specialist Conference*, Lake Tahoe, CA, 7-11 August 2005.

[50] Fahroo, F. and Ross, I. M., "Direct Trajectory Optimization by a Chebyshev Pseudospectral Method," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 1, January-February 2002, pp. 160–166.

[51] Kawajir, Y., Laird, C., Vigerske, S., and Wächter, A., *Introduction to IPOPT: A Tutorial for Downloading, Installing, and Using IPOPT*, Eclipse Public License, August 2014, Revision: 2500.

[52] Waltz, R. A. and Plantenga, T. D., *KNITRO User's Manual Version 7.0*, Ziena Optimization, Inc., September 2010.

[53] Murtagh, B. A. and Saunders, M. A., *MINOS 5.51 User's Guide*, Systems Optimization Laboratory, Stanford University, Stanford CA, September 2003.

[54] Whitmore, S. A., "Closed-form Integrator for the Quaternion (Euler Angle) Kinematics equations," May 9 2000, US Patent 6,061,611.

[55] Junfeng, L., Min, X., Zhaolin, W., and Shimin, W., "Minimum-torque Earth Off-nadir Pointing Control of Gravity Gradient Stabilized Small Satellites," *Tsinghua Science and Tenchnology*, Vol. 5, No. 1, March 2000, pp. 31–33.

[56] Bender, E., *An Analysis of Stabilizing 3U CubeSats Using Gravity Gradient Techniques and a Low Power Reaction Wheel*, California Polytechnic State University, San Luis Obispo, CA, 2011.

[57] CubeSatShop.com, *MAI-201 Miniature 3-Axis Reaction Wheel*, 2014 (accessed Aug 24, 2014).