Development of a Generic Guidance Navigation & Control System for Small Satellites: Application to HuskySat-1

Taylor P. Reynolds*, Krish Kaycee†, Bijan Barzgaran‡, Mathias Hudoba de Badyn‡, Sean Rice‡, Emma Hansen‡, Aaron Adler§, Behçet Açikmeşe¶ and Mehran Mesbahi¶

University of Washington, Seattle, WA 98195-2400

This paper presents the development of a generic Guidance, Navigation & Control (GNC) system for small satellites at the University of Washington that serves as a testbed for algorithm development and supports small satellite mission design. Our ultimate aim is to advance the capabilities of autonomous satellite operations by demonstrating advanced on-board control and estimation techniques, though the work is equally applicable to earth-imaging, pointing and technology demonstration missions. The work discusses the development of flight software and requisite sensor, actuator and environmental models needed to validate flight software functionality. The first mission application is the HuskySat-1 mission, undertaken by a student-led group at the University of Washington. Primarily a technology demonstration, HuskySat-1 is a 3U CubeSat equipped with a pulsed plasma thruster and a high-frequency communication antenna, equipped with a 3-axis active attitude control subsystem. We demonstrate through software and hardware-based testing that our system supports both scientific payloads. We outline our approach to algorithm development, verification and validation, and briefly hardware-in-the-loop testing.

I. Introduction

Maturing space technology has driven academic institutions to develop increasingly complex space missions. Academia has found that CubeSat programs are an exceptional tool to develop skills in collaborative engineering, technology and product development, team-building, and project management. As such, university design teams focused on space engineering are increasingly popular across the world. With many successful missions to-date [1, 2, 3, 4, 5, 6, 7] this trend is likely to continue.

Although Washington state is a leader in the aerospace industry, it has been nearly two decades since the last large satellite engineering mission was developed by the University of Washington [7, 8]. The University of Washington (UW) has, in recent years, thus begun the re-development of such capabilities. This paper outlines the development of a generic Guidance, Navigation and Control (GNC) system for small satellites, as well as models of related hardware and the space environment. The objective is to produce a baseline set of tools with which to design, build, and test GNC systems for future satellite missions. Through a modular approach to flight software design, we are able to select a subset of the software to rapidly design a GNC system for a variety of missions. As a demonstration this paper discusses the first UW mission, HuskySat-1.

The ultimate goal of UWs satellite program, from a GNC perspective, is to demonstrate the use of advanced control methods for small satellites. There have been several previous projects in which the GNC system is a primary focus, if not a primary mission payload. Most notably these have occurred in the area of formation flight, with leading-edge projects at the Massachusettes Institute of Technology [9, 1, 10, 11] and the University of Toronto [12, 13, 14]. The UW also developed a single satellite as part of a larger formation for the ION-F mission [7, 8]. The work presented here represents the next steps taken at UW to gain the ability to demonstrate our theoretical work in the same areas [15, 16, 17, 18, 19].

As a complement to formation flight capabilities, the generic GNC system will serve as a baseline for the development of optimization based control strategies for small satellites. While classical linear control design through

*Doctoral student, W. E. Boeing Department of Aeronautics & Astronautics, AIAA student member. Corresponding author: tpr6@uw.edu.
†GN&C Engineer, Blue Origin, AIAA member.
‡Doctoral student, W. E. Boeing Department of Aeronautics & Astronautics.
§Undergraduate student, Department of Physics.
¶Professor, W. E. Boeing Department of Aeronautics & Astronautics, AIAA member.

loop-shaping represents a mature control method for spacecraft, convex optimization based algorithms for space systems are not at the same readiness. Even for single-satellite missions, convex optimization can be used as a means to guarantee system-level requirements, such as sensor/payload field of view restrictions, are met [20, 16, 17, 19]. These same concepts may also be applied to formation flying maneuvers [16, 21, 22].

Recent advances in on-board solution methods using convex optimization offer a unique opportunity to advance the state of the art [23, 24, 25, 26, 27]. These works range from on-board trajectory generation for powered descent guidance and agile quad rotor maneuvers to generic convex optimization solvers for embedded systems. The GNC system developed herein aims to support future work in this area for small satellites, and serve as a catalyst to its development.

Finally, our GNC system may be used for more immediate mission architectures. Among these, on-demand precise pointing for earth-imaging and technology demonstration offer a glimpse of the capabilities provided. The HuskySat-1 mission is the first mission where these GNC tools may be tested on a real spacecraft, and represents a technology demonstration mission.

A. HuskySat-1

HuskySat-1 is a 3U nanosatellite being developed by an interdisciplinary team of both graduate and undergraduate students at the UW. HuskySat-1 aims to demonstrate new technologies being developed at the UW, expanding the current capabilities of CubeSats as a whole. Two key scientific payloads are being flown on the satellite; a pulsed plasma thruster (PPT) and a high frequency K-band antenna. The PPT is a robust solid sulfur propellant electric propulsion engine that generates a specific thrust of 45 mN/kW and operates continuously for an extended period of time [28]. The primary objective of the PPT system on board is to fire successfully in space, with a secondary objective of imparting a measurable ΔV of 50 m/s. The high frequency antenna uses K-band and is capable of downlinking data at a rate of up to 1 Mb/s when flying over the UW ground station. The antenna is located at one end of the satellite's long axis, opposite to the PPT. The satellite also includes a low gain antenna that permits lower data rate uplink and downlink capabilities from and to the UW ground station. Since the high-frequency antenna represents a primary scientific payload, the low-gain antenna also serves as a low risk back-up communication system.

In order to demonstrate these new technologies, active 3-axis control of the satellite bus is required. One of the principal drivers is the high-frequency antenna's requirement to be pointed at the UW ground station within 20 degrees to successfully downlink data. At the same time, active 3-axis control was selected to gain institutional knowledge and experience in designing such systems.

The active on-board GNC system is comprised of the attitude determination and attitude control subsystems, while the translational motion is not directly controlled. That is, the PPT is not directly used as an actuator by any on-board control methods; the mission objectives and success criteria do not necessitate so. The attitude control subsystem uses a set of three magnetorquers to actuate the spacecraft with the required accuracy. The attitude determination subsystem uses a combination of digital sun sensor, three three-axis magnetometers, and a single three-axis MEMS gyroscope to achieve the required attitude knowledge. We discuss the details of each hardware component in depth in §III, while the flight software is formed using a subset of our generic GNC software described in the next section.

B. Organization and Notation

The paper is organized into two principal sections; §II describes the main components of the generic GNC system, while §III describes the specific application of these components to the HuskySat-1 mission. For the generic GNC system, §II.A describes the equations of motion, §II.B outlines the disturbance forces and torques acting on the spacecraft, §II.C outlines models for on-board sensors and actuators, and finally §II.D describes all of the flight software components designed for on-board use. For HuskySat-1, §III provides an outline of the mission and the GNC concept of operations, while §III.A and §III.B detail the hardware components used. Finally, §III.C discusses the preliminary hardware-in-the-loop validation of the flight software designed for the mission.

Our notation for vectors will be lowercase, bold-face letters. Matrices are upper case, regular font letters. Each vector may have several meanings; the vector $\boldsymbol{a}$ represents an object with magnitude and direction, whereas $\boldsymbol{a}_{\mathcal{F}}$ represents the coordinates of the vector $\boldsymbol{a}$ in the coordinate frame implied by the subscript $\mathcal{F}$. The same vector may be represented in different coordinate frames, and this is noted using different calligraphic subscripts. Vectors that are measured or estimated, either by sensors or state estimation routines, are adorned with a $\hat{\cdot}$. Quaternions may be thought of as four dimensional vectors and thus appear as lowercase bold-face letters as well. We use $\mathbb{R}$ to denote the set of real numbers, and $\mathbb{R}^n$ to denote the vector space of real n dimensional vectors and $\mathbb{R}^{n\times m}$ to denote the set of $n\times m$ real matrices. The set $\mathbb{S}^n_{++}$ represents the $n\times n$ symmetric positive definite matrices. We use $\mathcal{N}(\mu,\sigma)$ to denote a Gaussian white noise process with mean μ and standard deviation σ.

II. Guidance, Navigation, and Control System

The design of a generic GNC system was broken down into five main components to guide modularity. As shown in Figure 1, these are the *environmental modeling*, *equations of motion* (dynamics), *sensors*, *actuators*, and *flight software*. These five components represent the key ingredients for the GNC system design at the highest level. In what follows, we will highlight each of these libraries and discuss the assumptions, models, and algorithms included in our system. We will use the term "library" to refer to functional pieces of software with well-defined inputs and outputs, such as those seen in Figure 1.

We have chosen to develop the GNC system using Simulink from MathWorks. Simulink provides a natural control system design environment as well as an auto-coding feature to rapidly generate embeddable flight code. The use of Simulink libraries maximizes flexibility and re-usability while ensuring a clean design and debug process. As much as possible, libraries are atomic units referenced by higher-level libraries. The result is that a primary model file contains only five libraries, seen in Figure 1. Each of these five libraries contain numerous sub-libraries, the functionality of which are described in the remainder of this section.

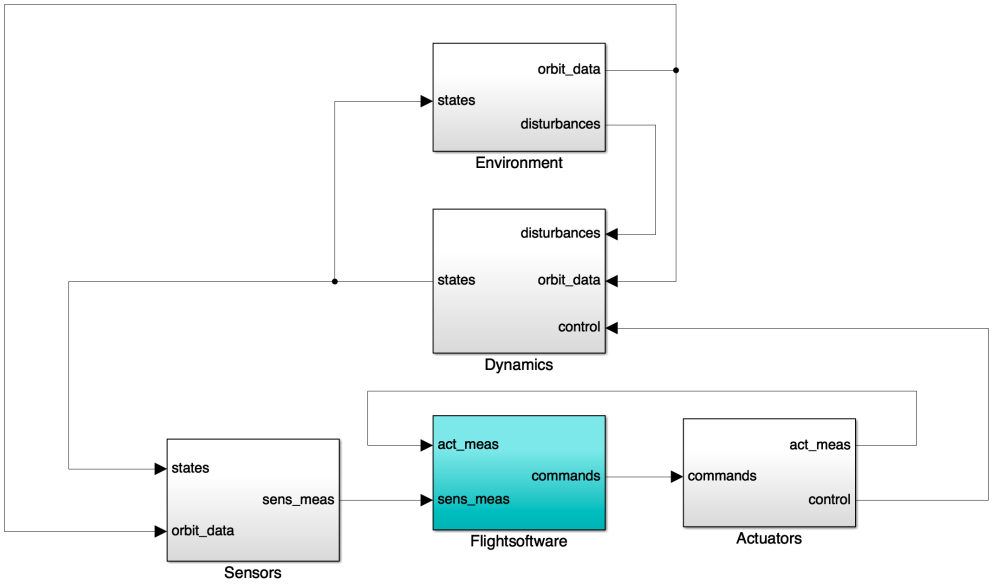


Figure 1. The top level components of a generic GNC system. Not shown are telecommands and telemetry, handled separately and inside the flight software block.

A. Equations of Motion

The rotational motion of the spacecraft is assumed to be completely decoupled from the orbital motion. For motion planning schemes that involve both types of motion one can plan attitude maneuvers that satisfy constraints imposed by the required translational motion, or vise-versa. Making this assumption greatly simplifies the development of the primary GNC system.

To facilitate the use of each library in testing scenarios, we found it beneficial to separate the equations of motion into three main components. These are the rigid body dynamics, the kinematics, and the orbital motion. The decoupling of attitude dynamics and kinematics ensures that our software is agnostic to the chosen parameterization of attitude; the rigid body dynamics model is equally applicable to the use of quaternions, Euler angles, Rodrigues parameters, and other parameterizations. The rigid body dynamics library takes in all external torques acting on the spacecraft and integrates Euler's equations to propagate the angular velocity of the spacecraft. We assume that the rotational motion is governed by

$$J\dot{\boldsymbol{\omega}}(t) = \boldsymbol{\tau}_{\mathrm{c}}(t) + \boldsymbol{\tau}_{\mathrm{d}}(t) - \boldsymbol{\omega}(t)^{\times}\left(J\boldsymbol{\omega}(t) + \boldsymbol{h}_{w}(t)\right), \tag{1}$$

where $\boldsymbol{\tau}_{\mathrm{c}}$ and $\boldsymbol{\tau}_{\mathrm{d}}$ are the control and disturbance torques observed in the spacecraft body fixed frame, $J \in \mathbb{S}^{3}_{++}$ is the positive definite 3×3 inertia matrix of the satellite and $\boldsymbol{h}_{w}$ is the momentum of any on-board momentum storage devices (such as reaction wheels). We use $\cdot^{\times}$ to denote the 3×3 skew-symmetric cross product matrix. Our disturbance torques include contributions from gravity gradients, aerodynamic effects, solar pressure, and residual/parasitic

magnetic interactions. See §II.B for details on the disturbance torques. The control torques are generated from one of several controllers outlined in §II.D.4.

The attitude kinematics are propagated according to a right-handed, Hamiltonian quaternion formalism where the scalar portion occupies the first position. We define the attitude quaternion to map vectors from an earth-centered inertial frame to the spacecraft's body frame. The attitude kinematics library takes in the current angular velocity (the output of the rigid body dynamics library) and integrates the quaternion kinematic equation to propagate the attitude of the spacecraft. The model uses

$$\dot{\boldsymbol{q}}(t) = \frac{1}{2}\boldsymbol{q}(t) \otimes \boldsymbol{\omega}(t), \tag{2}$$

to propagate the attitude quaternion. We use the symbol $\otimes$ to represent the product of two quaternions, expressed in their components as

$$\boldsymbol{q} = \begin{bmatrix} q_0 \\ \boldsymbol{q}_v \end{bmatrix}, \quad \boldsymbol{p} = \begin{bmatrix} p_0 \\ \boldsymbol{p}_v \end{bmatrix} \quad \Rightarrow \quad \boldsymbol{q} \otimes \boldsymbol{p} = \begin{bmatrix} q_0 p_0 - \boldsymbol{q}_v^T \boldsymbol{p}_v \\ p_0 \boldsymbol{q}_v + q_0 \boldsymbol{p}_v + \boldsymbol{p}_v^\times \boldsymbol{q}_v \end{bmatrix}.$$

We refer to q_0 as the scalar part of the quaternion and $\boldsymbol{q}_v$ as the vector part [29]. Note that equation (2) requires that $\boldsymbol{\omega}$ is treated as a *pure quaternion*, or one that has a zero scalar entry. The direction cosine matrix is related to the attitude quaternion via

$$C_{\mathcal{B}\leftarrow\mathcal{I}} = (2q_0^2 - 1)I_{3\times 3} - 2q_0\boldsymbol{q}_v^\times + 2\boldsymbol{q}_v\boldsymbol{q}_v^T \tag{3}$$

By mapping all forces to an earth-centered inertial frame (see §II.B.1), we are able to integrate the position and velocity states directly as a double integrator system. Therefore we model the evolution of the translational states according to

$$\ddot{\boldsymbol{r}}(t) = \boldsymbol{a}_g(t) + \boldsymbol{a}_c(t) + \boldsymbol{a}_d(t), \tag{4}$$

where $\boldsymbol{a}_g$ is the acceleration due to gravity, $\boldsymbol{a}_c$ is the acceleration applied by on-board thrusters, and $\boldsymbol{a}_d$ represents disturbing accelerations. The disturbing accelerations are assumed to come from aerodynamic drag and solar pressure, see §II.B.2.

Note that for all subsequent equations, we will drop the argument of time. When necessary, time dependence will be stated explicitly.

B. Environment Simulation

To validate the flight software that is being developed, it is crucial to simulate its behavior in a realistic space environment. Doing so allows us to ensure that the system as a whole is robust to the disturbances encountered in space, at least to the accuracy of the models used. In this section, we will detail the models that we have designed to emulate the satellite's orbital environment. Ultimately, we may compare actual on orbit data to both assess spacecraft performance and improve the accuracy of the models discussed here.

1. Time & Coordinate Systems

The first and arguably most critical component of environmental modeling (and indeed the flight software itself) is the correct use and application of time and coordinate systems. Each of the models we use rely in some capacity on either time or a coordinate rotation matrix. Incorrect determination of either can therefore lead to incorrect forces, torques, and interpretation of satellite data.

The handling of time on-board the spacecraft is an important design decision. We assume that the satellite's power system is the manager of the estimated mission elapsed time (MET). This is done because the power system is nearly always the system which is turned on first, therefore may begin the MET count before the GNC system receives power. For consistency, we have assumed a universal spacecraft time of Julian seconds since J2000 in the terrestrial time (TT) frame [30]. Other time frames can be used, however we find that this choice gives us the fidelity required without the need to insert leap seconds[a]. The use of Julian seconds in the terrestrial time frame also makes the handling of on-board time agnostic to the inclusion of a GPS sensor, which can itself provide a time value. The GPS time frame is different from Julian seconds in TT and a conversion is needed. If a GPS sensor is not included in a given mission, our system does not need to be redesigned.

While we use Julian seconds in TT as the base satellite time, various algorithms require other time frames. We will highlight these as we go, and only provide a list of the frames used here. Currently, our system supports conversion

[a]Handling of leap seconds can be done in our system by remotely updating a single parameter.

to Universal Coordinated Time (UTC), UT1, and Atomic Time (AT). Most of the on-board algorithms make use of either UTC or UT1 and the difference between these two time frames is updated weekly using IERS data. The format for any time frame may be in YMD-HMS form, Julian date, or Julian century. All Julian dates are referenced to the J2000 epoch. The algorithms for converting between time frames were sourced from [30].

Next, we use the time information to compute the rotation matrices between several key frames. We assume the Earth-centered inertial (ECI) frame as the base frame in which Newton's laws are valid (i.e. for equation (4)). We will use $\{\boldsymbol{I}, \boldsymbol{J}, \boldsymbol{K}\}$ as coordinate vectors to reference the ECI frame, and assume henceforth that $\boldsymbol{K}$ is in the direction of the earth's spin axis, $\boldsymbol{I}$ points toward the Vernal equinox, and $\boldsymbol{J}$ completes a right-handed coordinate system. Vectors resolved in the inertial frame will carry a subscript $\cdot_{\mathcal{I}}$. We use a second inertial frame, the True Equator Mean Equinox (TEME) frame in flight software orbit propagation routines [31]. This frame is not discussed explicitly in [31], however it is traditionally understood that the original work made use of this coordinate frame. Use of these orbit propagation techniques is the only reason the TEME frame is necessary.

Finally, an Earth-centered Earth-fixed (ECEF) frame completes the set of orbital frames[b]. The transformation between ECI and ECEF follows the IAU-76/FK5 convention and takes into account the sidereal time rotation, nutation and precession effects, but ignores polar motion [30]. Velocities are mapped using a pseudo-fixed frame to handle the rotation of the ECEF frame with respect to the ECI frame. The ECEF frame is used primarily for satellite observations and is the frame in which GPS measurements are taken. Vectors resolved in this frame carry a subscript $\cdot_{\mathcal{F}}$.

2. *Environmental Modeling*

The disturbance models use both time and coordinate system information to generate force and torque information. We assume the presence of parasitic magnetic disturbance torque, aerodynamic drag (force and torque), solar pressure (force and torque), gravity force, and gravity gradient torque.

MAGNETIC FIELD MODEL: Magnetic field vectors are computed using the World Magnetic Model (WMM) 2015 computed as a function of ECEF position and UT1 time. The magnetic field is mapped to the ECI frame and output both in units of Tesla and as a normalized unit vector. We have found that the internal Simulink WMM library can render large scale simulations slow. For low fidelity simulations we may therefore replace the WMM model with an approximate model based on a least squares (LS) fit. Using sinusoidal basis functions, we approximated the WMM over a period of two orbits [32]. The result is a library function that uses only a time input to obtain an inertial magnetic field vector and permits faster simulations. The basis functions assumed are

$$\{1,\, \cos\Omega_o t,\, \sin\Omega_o t,\, \cos 2\Omega_o t,\, \sin 2\Omega_o t,\, \cos 3\Omega_o t,\, \sin 3\Omega_o t\}, \tag{5}$$

where Ω_o is the orbital frequency and t is the time in Julian seconds in the UT1 frame. A least squares fit to data generated over a period of two orbits allows us to write the magnetic field vector as

$$\boldsymbol{b}(t) = \hat{\boldsymbol{b}}_0 + \cos\Omega_o t\,\hat{\boldsymbol{b}}_1 + \sin\Omega_o t\,\hat{\boldsymbol{b}}_2 + \cos 2\Omega_o t\,\hat{\boldsymbol{b}}_3 + \sin 2\Omega_o t\,\hat{\boldsymbol{b}}_4 + \cos 3\Omega_o t\,\hat{\boldsymbol{b}}_5 + \sin 3\Omega_o t\,\hat{\boldsymbol{b}_6}, \tag{6}$$

where $\{\hat{\boldsymbol{b}}_i\}_{i=0}^{6}$ are 3×1 real vectors of constant coefficients estimated from the least squares fit. This process allows for faster computation for proof of concept simulations and initial algorithm development. Note that the coefficients $\hat{\boldsymbol{b}}_i$ are orbit specific and must be re-computed for different orbits.

Figure 2 shows a comparison of the WMM 2015 magnetic field model and the results of the least squares procedure over one orbit. For this demonstration, we used a nearly circular orbit with an inclination of 98.5 degrees. One can see that there is good agreement between the two models over this period of time. We note that due to secular variations in the earth's magnetic field, the least squares approximation is only valid for relatively short duration tests.

AERODYNAMIC DRAG: We use the 2001 United States Naval Research Laboratory Mass Spectrometer and Incoherent Scatter Radar Exosphere atmospheric model that is native to Simulink. This library provides the atmospheric density at a given time and position that is then used to compute the aerodynamic force and torque acting on the satellite, assuming some offset between center of mass and center of pressure. The equations used to compute the force and torque, which can be found in [33], assume that all momentum from atmospheric molecules is transfered to the spacecraft.

The satellite is modeled as a nominally non-rotating parallelepiped with six faces. The drag force is computed for each face and then summed together. The drag force on the ith face is computed according to

$$\boldsymbol{f}_{aero,i} = \begin{cases} -\rho \left\|\boldsymbol{v}_{\mathcal{B}}\right\|_2 \boldsymbol{v}_{\mathcal{B}}(\boldsymbol{n}_i \cdot \boldsymbol{v}_{\mathcal{B}}), & \boldsymbol{n}_i \cdot \boldsymbol{v}_{\mathcal{B}} > 0, \\ 0, & \boldsymbol{n}_i \cdot \boldsymbol{v}_{\mathcal{B}} \leq 0, \end{cases} \tag{7}$$

[b]This list excludes the true and estimated satellite body frames, as well as the commanded orientation.

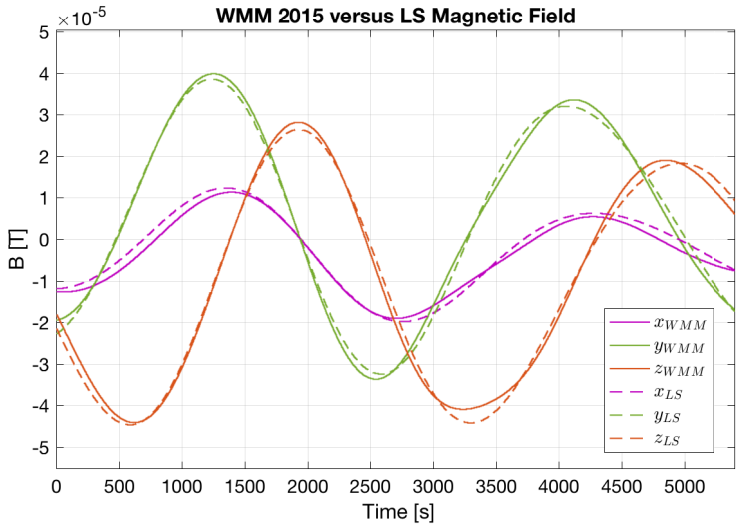


Figure 2. Comparison of the World Magnetic Model 2015 and a least squares fit to data obtained over two representative orbits. The WMM values are depicted by solid lines, whereas the least squares (LS) values are depicted with dashed lines.

where $\boldsymbol{v}_{\mathcal{B}}$ is the velocity resolved in the spacecraft's body frame, ρ is the atmospheric density (a function of position and time), and $\boldsymbol{n}_i$ is the ith face's normal vector.

If $\boldsymbol{r}_{cop}$ is the vector pointing from the spacecraft's center of mass to the center of pressure, then the aerodynamic torque is computed as

$$\boldsymbol{\tau}_{aero} = \boldsymbol{r}_{cop}^{\times} \sum_{i=1}^{6} \boldsymbol{f}_{aero,i} = \boldsymbol{r}_{cop}^{\times} \boldsymbol{f}_{aero}. \tag{8}$$

Note that $\boldsymbol{a}_d$ in equation (4) is partially composed of $\boldsymbol{f}_{aero}/m$, where m is the mass of the spacecraft, while $\boldsymbol{\tau}_d$ in equation (1) is partially composed of $\boldsymbol{\tau}_{aero}$.

SOLAR PRESSURE: The force and torque due to solar pressure on the spacecraft can be computed using equations similar to those for aerodynamic drag. Here, the atmospheric density and velocity vector are replaced by the solar-radiation pressure and relative position vector to the sun. Let $\boldsymbol{r}_{\mathcal{B},\odot}$ be the vector pointing from the spacecraft to the sun, resolved in the spacecraft's body frame. The force due to solar radiation pressure is then

$$\boldsymbol{f}_{srp,i} = \begin{cases} -p_{srp} c_{R,i} \frac{\boldsymbol{r}_{\mathcal{B},\odot}}{\|\boldsymbol{r}_{\mathcal{B},\odot}\|} (\boldsymbol{n}_i \cdot \boldsymbol{r}_{\mathcal{B},\odot}), & \boldsymbol{n}_i \cdot \boldsymbol{r}_{\mathcal{B},\odot} > 0, \\ 0, & \boldsymbol{n}_i \cdot \boldsymbol{r}_{\mathcal{B},\odot} \leq 0, \end{cases} \tag{9}$$

where $c_{R,i} \in [0, 2]$ is the reflectivity of the ith face [30]. These constant coefficients are used to model spacecraft that may have solar panels on some faces, but not on others. The constant $p_{srp} = 4.57 \times 10^{-6} N/m^2$ is the solar pressure force per unit area. The spacecraft-to-sun vector is obtained first in the inertial frame using the method described in §3. The transformation to the spacecraft body frame is then achieved by first summing with the spacecraft-to-earth vector using the state information from equation (4) and then mapping the result to the body frame via the attitude quaternion.

By assuming that the solar pressure acts at the center of pressure, like the aerodynamic drag, we can write the resulting torque as

$$\boldsymbol{\tau}_{srp} = \boldsymbol{r}_{cop}^{\times} \sum_{i=1}^{6} \boldsymbol{f}_{srp,i} = \boldsymbol{r}_{cop}^{\times} \boldsymbol{f}_{srp}. \tag{10}$$

The acceleration $\boldsymbol{f}_{srp}/m$ contributes to $\boldsymbol{a}_d$ in equation (4), while $\boldsymbol{\tau}_{srp}$ contributes to $\boldsymbol{\tau}_d$ in equation (1).

GRAVITY FORCE AND GRADIENTS: The final disturbance model we include is that of gravity gradients, and the associated gravity force. The force of gravity is approximated using the spherical harmonic expansion EGM2008 model that is internal to Simulink. We have found that a 40th degree order model offers a good balance of simulation speed and fidelity. This model emits directly the acceleration due to gravity $\boldsymbol{a}_g$ that is used in equation (4).

We assume a model for gravity gradient torque based on a simplified first order gravity model [33]. The equation for the torque arising from a misalignment between the local gravity vector and principal body axes is, in the body frame,

$$\boldsymbol{\tau}_{gg} = \frac{3\mu}{\|\boldsymbol{r}_{\mathcal{B}}\|^5} \boldsymbol{r}_{\mathcal{B}}^{\times} J \boldsymbol{r}_{\mathcal{B}}, \tag{11}$$

where $\boldsymbol{r}_{\mathcal{B}}$ is the spacecraft position vector in the body frame, μ is the gravitational parameter of the earth, and J is the spacecrafts inertia matrix. The torque $\boldsymbol{\tau}_{gg}$ contributes to $\boldsymbol{\tau}_d$ in equation (1).

C. Sensing & Actuation

Sensor and actuator models can greatly impact the accuracy of GNC simulations. Time delays, noise, and bias are properties inherent to all hardware components, and the GNC system must be designed with these in mind to realize robust performance goals. All sensor readings are quantized to a precision appropriate to the sensor. In this section, we detail the models and assumptions made for each hardware component in our generic GNC testbed.

1. Sun Sensors

We assume that the on-board sun sensors supply two angles, α and β, that combined represent a unit vector pointing from the spacecraft to the sun. The bounds on these angles are the sensor's field of view – unique to each sensor but commonly in the (plus/minus) $60 - 70$ degree range.

Our model assumes that the inertial spacecraft-to-sun vector, $\boldsymbol{s}_{\mathcal{I}}$, is measured in the sun sensor's local frame as $\boldsymbol{s}_{ss} = [s_{ss,x} \;\; s_{ss,y} \;\; s_{ss,z}]^T \in \mathbb{R}^3$. We find it most convenient to compute the latter vector, in simulation, using the spacecraft's time-varying attitude quaternion and a constant direction cosine matrix relating the body frame to the sensor frame. This leads to

$$\boldsymbol{s}_{ss} = C_{ss\leftarrow\mathcal{B}} \boldsymbol{s}_{\mathcal{B}} = C_{ss\leftarrow\mathcal{B}} \, C_{\mathcal{B}\leftarrow\mathcal{I}}(\boldsymbol{q}) \boldsymbol{s}_{\mathcal{I}}.$$

This vector is then turned into the two angles according to

$$\alpha = \tan^{-1} \frac{s_{ss,x}}{s_{ss,z}} \qquad \beta = \tan^{-1} \frac{s_{ss,y}}{s_{ss,z}}$$

where $\tan^{-1}$ is the two-argument arctangent function. Each measurement channel is then corrupted by zero-mean Gaussian white noise, so that the final sensor model is given by

$$\hat{\alpha} = \alpha + \epsilon_{\alpha}, \qquad \hat{\beta} = \beta + \epsilon_{\beta}, \tag{12}$$

where $(\epsilon_{\alpha}, \epsilon_{\beta})$ are the noise terms, with standard deviation drawn from appropriate data sheets.

2. Gyroscopes

The gyroscope model uses the concepts of angle and rate random walk to compute the angular velocity measurements. The angle random walk (ARW) is a measure of the accumulated error (noise) in the sensor at zero bias, whereas the rate random walk (RRW) represents the accumulated error in the bias stability itself. Each of these are typically given in the sensors data sheet, though an experimental Allen Variance plot should be generated to characterize each sensor used in engineering and flight units. We let $\epsilon_a \sim \mathcal{N}(0, \sigma_a)$ represent ARW, with units of radians per second, and $\epsilon_r \sim \mathcal{N}(0, \sigma_r)$ represent RRW, with units of radians per second per second.

The gyroscope model is assumed to be

$$\hat{\boldsymbol{\omega}} = \boldsymbol{\omega} + \epsilon_a + \chi, \tag{13a}$$

$$\dot{\chi} = \epsilon_r, \tag{13b}$$

where χ represents the gyroscope bias and $\boldsymbol{\omega}$ the true angular velocity of the spacecraft bus. It is assumed that the gyroscope's axis is aligned with the principal axis.

3. Magnetometers

We assume that the magnetometers are corrupted by both soft and hard iron biases. Hard iron biases result in a measurement that has a constant offset from the truth value and can be caused by satellite components that create magnetic fields themselves. We shall use $\boldsymbol{\epsilon}_h \in \mathbb{R}^3$ to represent the hard iron bias. Soft iron biases result in a distorted measurement and can be attributed to material characteristics that cause local magnetic field lines to condense. We shall use the direction cosine matrix $C_s \in \mathbb{R}^{3\times3}$ to represent the soft iron bias. While their numerical values are unique to each spacecraft, each of these biases may be considered constant over the mission lifetime and should be determined experimentally for each engineering and flight unit.

Let $\boldsymbol{b}_\mathcal{I}$ be the inertial magnetic field vector at a given time, at which point the spacecraft's attitude quaternion is $\boldsymbol{q}$. The measured magnetic field vector is then given by

$$\hat{\boldsymbol{b}}_{mag} = C_{mag\leftarrow\mathcal{B}}\left(C_s C_{\mathcal{B}\leftarrow\mathcal{I}}(\boldsymbol{q})\boldsymbol{b}_\mathcal{I} + \boldsymbol{\epsilon}_h\right) + \boldsymbol{\epsilon}_B, \tag{14}$$

where $C_{mag\leftarrow\mathcal{B}}$ is the constant direction cosine matrix relating the spacecraft's body frame to the magnetometer sensor's frame and $\boldsymbol{\epsilon}_B \sim \mathcal{N}(0, \sigma_B)$. The standard deviation of the magnetometer noise may be drawn from relevant data sheets, or experimentally obtained via Allen Variance plots.

4. Global Position System

The GPS sensor model provides three outputs: position, velocity, and time. The first two vector measurements are given in the ECEF frame, while the time is assumed to be in the GPS time frame using the two digit week and second-of-week format. The epoch of the GPS frame is January 6, 1980 at midnight UTC. The sensor model therefore takes in the inertial position and velocity, $(\boldsymbol{r}_\mathcal{I}, \boldsymbol{v}_\mathcal{I})$, obtained from the translational dynamic model in equation (4) and the MET in the TT time frame. Currently, there is an offset of 51.184s between the GPS and TT time frames, a number that increases by one with the introduction of each new leap second [30]. This offset is used to transform the Julian seconds since J2000 into the GPS time frame and format. The GPS model outputs position and velocity as

$$\boldsymbol{r}_\mathcal{F} = C_{\mathcal{F}\leftarrow\mathcal{I}}\boldsymbol{r}_\mathcal{I}, \tag{15a}$$

$$\boldsymbol{v}_\mathcal{F} = C_{\mathcal{F}\leftarrow\mathcal{I}}\left(\boldsymbol{v}_\mathcal{I} - \boldsymbol{\omega}_\oplus^\times \boldsymbol{r}_\mathcal{F}\right), \tag{15b}$$

where $\boldsymbol{\omega}_\oplus = \omega_\oplus \boldsymbol{K}$ is the spin rate of the earth and $C_{\mathcal{F}\leftarrow\mathcal{I}}$ is the direction cosine matrix mapping the inertial ECI frame to the earth-fixed ECEF frame discussed in §II.B.1.

5. Reaction Wheels

Reaction wheels are momentum exchange devices used primarily to perform precise large angle reorientations. Since the inertial momentum of the system (reaction wheels and spacecraft bus) is conserved in the absence of external torques, the reaction wheels must increase their momentum (i.e. rotate) in the opposite direction of the desired spacecraft bus motion. Typically, brushless DC motors are paired with a rotating flywheel, whose angular speed is controlled by on-board software. Three reaction wheels are usually mounted along each principal body axis, while four wheels can provide redundancy to the system. Our GNC system assumes three reaction wheels each of which is modeled the same way. The model developed here is based on [34].

The torque provided by each motor is assumed proportional to the current in the armature. The torque is thus

$$\boldsymbol{\tau}_{c,w} = KI = J_w\dot{\Omega} + b\Omega, \tag{16}$$

where Ω is the angular velocity of the reaction wheel, J_w is the motor (with flywheel) inertia, b is a coefficient of viscous friction, K is the motor torque constant, and I is the current in the armature. We assume that the input to the system is the armature voltage. A commanded torque may be mapped to a current, which is then mapped to the input voltage to the motor. The current, as a function of this voltage, is given by

$$\frac{dI}{dt} = \frac{1}{L}\left(V - RI - K\Omega\right), \tag{17}$$

where R and L are the resistance and inductance of the armature circuit, respectively, and V is the input voltage. We

may combine equations (16) and (17) into the state space model

$$\frac{d}{dt}\begin{bmatrix}\Omega\\ I\end{bmatrix} = \begin{bmatrix}-\frac{b}{J_w} & \frac{K}{J_w}\\ -\frac{K}{L} & -\frac{R}{L}\end{bmatrix}\begin{bmatrix}\Omega\\ I\end{bmatrix} + \begin{bmatrix}0\\ \frac{1}{L}\end{bmatrix} V, \tag{18a}$$

$$\boldsymbol{y} = \begin{bmatrix}1 & 0\\ 0 & K\end{bmatrix}\begin{bmatrix}\Omega\\ I\end{bmatrix}. \tag{18b}$$

The output of the model is therefore both the angular velocity of the wheel and the torque. A simplified version of this model can be found in [34]. Note that the continuous time model developed from electro-mechanical principals is discretized in implementation. This facilitates feedback control design that may be applied to the actual system, as we discuss next.

The model given in equation (18) does not precisely track reference inputs commanded by flight software. As a result, a feedback controller is necessary. These are present in all physical reaction wheels and the gain values may be chosen by the GNC designer. As a result, this is a valuable modeling tool to develop during this stage, as it allows us to develop an adequate feedback controller. We assume that the feedback controller is of PID type. The controller is developed so that the closed loop system achieves the characteristics given in Table 1. To achieve these specifications, we find PI control[c] to be sufficient for most plant models as given in equation (18). We assume that the commanded rotation rates are tracked to the ability of the feedback controller and do not impose any further uncertainties. However, reaction wheels suffer poor performance around zero RPM. To-date, our model does not take this physical attribute into account and is the subject of future work.

Table 1. Reaction wheel feedback controller specifications.

Specification	Value
Settling time	0.2s
Overshoot	5%
Gain Margin	3dB
Phase Margin	45°

6. *Magnetic Torquers*

Magnetic torquers, or magnetorquers, are electromagnetic devices that produce torque by creating a local magnetic field that interacts with the ambient earth field. To do so, a current carrying wire is wound around either a core of ferrous material, or a non-conductive support material (air core). When a current is passed through the wire, a magnetic dipole is created that causes a torque on the spacecraft according to

$$\boldsymbol{\tau}_{c,m} = \boldsymbol{m}_{c,m} \times \boldsymbol{b}_{\mathcal{B}}, \tag{19}$$

where $\boldsymbol{m}_{c,m}$ is the magnetic dipole moment created by the magnetorquers, and $\boldsymbol{b}_{\mathcal{B}}$ is the earth's magnetic field in the body frame. Once on-board control algorithms determine the required magnetic moment, these commands are used to alter the magnetic dipole by modulating the current through the wire. The relationship between current and dipole is given by $\boldsymbol{m}_{c,m} = NIA\hat{\boldsymbol{n}}$, where N is the number of loops in the wire, I is the current, A is the loop area and $\hat{\boldsymbol{n}}$ is a normal vector.

In our system, magnetic dipole commands are mapped to a numerical range of integers in the interval $[-100, 100]$. This is done to support using Pulse Width Modulation (PWM) techniques to drive the actual actuators.

Often, magnetorquers can be modeled as a first order electrical system. However, the time constant of these models is quite high and thus lag time is nearly negligible. As such, we have adopted a straightforward model that simply maps the PWM commands back to a dipole, ignoring delays. Uncertainty in the realized dipole may be added to accurately model physical hardware components.

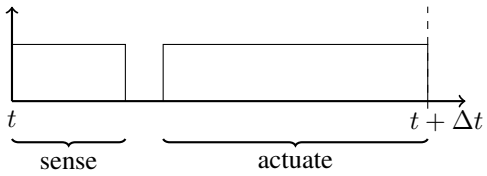
Remark 1. Since the reaction wheels are momentum exchange devices and the magnetorquers create an external torque on the system, we must choose the signs of $\boldsymbol{\tau}_{c,w}$ and $\boldsymbol{\tau}_{c,m}$ in equation (1) appropriately. Euler's equation and the conservation of angular momentum dictate that

$$\boldsymbol{\tau}_c = \boldsymbol{\tau}_{c,m} - \boldsymbol{\tau}_{c,w}. \tag{20}$$

[c]Or, the derivative gain term is very small compared to the proportional and integral.

Remark 2. The use of magnetorquers to generate a magnetic field can significantly affect a magnetometer's ability to measure the earth's magnetic field. We use a high-level duty cycle to manage this interaction by splitting each period of Δt seconds into separate sensing and actuation times. As a result, magnetic actuation and/or electric propulsion does not interfere as much with sensing capabilities. Figure 3 shows a graphical depiction of this duty cycle.

Figure 3. Graphical depiction of duty cycling used to ensure actuation does not interfere with our ability to obtain good sensor measurements.



7. *Propulsion*

We have selected a model of on-board thrusters that is amenable to both continuous and impulsive actuators. Impulsive thrust commands are assumed to occur periodically, with a period much less than the orbital period. This is consistent with pulsed thrusters, such as the PPT developed for the HuskySat-1 mission. This assumption allows us to approximate the impulsive thrust as a continuous one, averaged over a given time interval. For now, we have only assumed thrusters with the ability to produce a constant thrust and our flight software will turn the thrusters on and off to accelerate the spacecraft.

Let T_{imp} be the magnitude of the thrust per impulsive fire and $\hat{\boldsymbol{n}}_{thrust}$ be the boresight vector of the thruster in the spacecraft body frame. Over the time period Δt, we approximate the thrust as

$$\boldsymbol{a}_c = \frac{T_{imp}}{\Delta t}\hat{\boldsymbol{n}}_{thrust}. \tag{21}$$

D. Flight Software

The core flight software is being developed to meet a generic set of mission level requirements. Fundamentally, the role of flight software is to manage the momentum of the satellite. This requires the ability to estimate and control both the linear and angular momentum. This often leads to requirements to point the spacecraft with a specified accuracy, maintain accurate knowledge of the spacecraft's inertial position and the ability to compute and execute desired trajectories in both translational and rotational space. As a research platform, there is also the desire for the software to be able to interface with external experimental software components. We have therefore developed a core set of tools that together comprise a basic flight software toolbox for satellite operations.

The flight software was created as a modular entity so that it could be specialized for certain missions that may have access to either limited computation or a different set of sensors and actuators. All signal routing is done through well defined bus signals; giving each signal a name, data type, and size. Inputs and outputs can thus be exchanged and replaced easily, while retaining a rigorous definition for each signal. This allows for fast development and integration of new libraries. For the remainder of this section, we describe the key flight software components in greater detail.

1. *Input Conditioning*

By input conditioning, we refer to a library that receives raw (or only coarsely processed) data from on-board sensors and maps this to the information required as input to flight software. The easiest way to modularize this library is to include one sub-library for each sensor type.

Most sensors have straightforward conditioning; removal of known or measured sensor biases, such as hard or soft iron biases in magnetometers, filtering of high sample rate or noisy sensors, averaging of multiple homogeneous sensors, etc. Possible non-straightforward computations relate to the use of GPS sensors. Since measurements are received in the ECEF frame, care must be taken to map this data to an inertial frame (used generally as inputs to flight software routines). We have adopted a two-step procedure for this. First, we map the position and velocity readings to an inertial frame[d], accounting for rotation of the planet, and second we map the position-velocity to a set of orbital elements and ultimately a two-line element set (TLE). The purpose of this latter conversion is discussed further in §3.

[d]This can be either the ECI, TEME, or any other inertial frame, depending on the orbit propagation technique used on-board.

Finally, we have found it very useful to add an additional dimension to all vector measurements that serves as an indication of measurement validity. This extra piece of information, stored always in the last entry, tells downstream routines how best to use this measurement. We have adopted a boolean-like $0-1-2$ scheme, where 0 indicates invalid, 1 indicates valid, and 2 indicates a valid reading that was recently invalid but has not crossed a hysteresis threshold. The use of 2 allows us to build in hysteresis to downstream functions through a single computation. Our use of sun sensors provide a prime example. When this sensor returns an invalid reading (due to not having the sun in its field of view), the GNC system will respond by entering a sun seeking control mode. While it does so, the estimation routines do not have sufficient information to compute an attitude and thus the attitude estimate is propagated using gyroscopes only. Once the sun comes into the field of view, the validity bit switches to 2. This allows the spacecraft to now estimate and converge to an attitude state, since the measurements are valid, while remaining in the sun seeking control mode. Only once the sun is sufficiently centered in the field of view will the control mode be changed from sun seeking. This ensures that subsequent attitude maneuvers do not begin very near the sun sensor's field of view boundary. Moreover, since estimation routines have had sufficient information for some time, this also ensures a more accurate initial attitude state estimate.

2. *System Mode Management*

The GNC mode management system collects information from across the satellite to make high level decisions about what the satellite should be doing. We have built in a fundamental set of operating modes for the GNC system, described in Table 2. Each of these modes is assigned an integer value and the spacecraft must always assume one of them.

The default mode can be selected as appropriate, though we find that detumble is a suitable default state. The reason for this is that the detumble mode ensures stability of the spacecraft and implies that control action taken as a result of a mode switch always begins from an angular state of rest. We note also that the removal of reaction wheel momentum can be performed continuously during other operating modes. We intend the GNC mode in Table 2 to mean a dedicated desaturation event whereby all other control objectives are abandoned.

The system is easily augmented to include other modes. We find it practical to add additional system modes when experimental routines are being used, so that we know exactly how the system is producing actuator commands. For example, if both a standard and experimental controller are included for bus pointing, we may assign two different integer values for them. Therefore we have two modes that perform the same overall function, but do so in two different ways. This philosophy ensures that all experimental routines can be compared directly to a standard, or more mature, means to the same end.

Mode transitions are handled either autonomously or by direct ground station command. The autonomous mode selection is carried out by establishing a mode hierarchy, such that we may systematically check the ingress conditions from highest priority to lowest. Once a mode is entered, it is not left until specific egress conditions are met. The mode hierarchy is likely mission specific, so we will not comment on specifics here. However, our generic system uses detumble, low power mode, desaturation, sun seeking, bus pointing as a rough hierarchical structure.

Table 2. Fundamental set of high level operating modes for a generic satellite GNC system.

Mode	GNC Objective
Detumble	Decrease the angular momentum of the spacecraft bus to near zero.
Reaction Wheel Desaturation	Decrease the angular momentum of the spacecraft's reaction wheel system.
Sun Seeking	Reorient the spacecraft so that the sun is in the field of view of the sun sensor(s).
Bus Pointing	Point a specified boresight vector towards a target, while ensuring the sun is inside the field of view of the sun sensor(s).
Low Power Mode	Idle mode; the spacecraft may estimate a state vector but does not exert any control effort.

3. *Environment Models*

Many GNC objectives require specific inertial information to be successful. For example, a velocity pointing maneuver requires on-board knowledge of the spacecraft's inertial velocity vector. Ground station pointing for directed communication requires knowledge of the inertial ground station vector. Sun seeking maneuvers require knowledge of

the sun's position relative to the spacecraft. Each of these imply the need to design on-board instances of some of the environmental models described in §II.B.2. In addition, a suitable on-board orbit propagation technique is required.

To address the first point, the GNC flight software will require environmental models commensurate with the desired pointing objectives. In nearly all cases, however, the spacecraft is required to compute either its inertial position or velocity, i.e., estimating the state from equation (4). There are several methods to do so, the most common of which are Keplerian propagation, Kozai or Brouwer's method, or Simplified General Perturbation (SGP) methods [30, 31]. We have chosen to implement a version of the SGP-4 routine from [31] that is compatible with standard NORAD two-line element (TLE) sets.

Our solution revolves around continually updating an on-board TLE. The on-board TLE serves as an initial condition for the SGP-4 orbit propagator and is stored in persistent memory. The TLE is updated either by a GPS measurement (converted appropriately) or by ground station uplink of a NORAD TLE. The accuracy of the SGP-4 propagator was validated using the STK software with representative low earth orbits.

Using the inertial state information from the orbit propagation routine, we can subsequently compute several key pieces of information. First, attitude state estimation routines that use sun and magnetic field information require two suitable inertial models. These provide both the vector from the spacecraft to the sun and the local magnetic field vector in the ECI frame. Other estimation techniques may require star catalogs for use with star trackers, though we consider this beyond the scope of the current work.

The sun model uses the current time in Julian days since the J2000 epoch in the UT1 frame. This is used to compute the mean anomaly of the sun, $M_\odot$, the ecliptic longitude, λ_{ecl}, the obliquity of the ecliptic, ϵ, and the position magnitude, $\|\boldsymbol{s}_\mathcal{I}\|$, using Algorithm 29 from [30]. The vector from the earth to the sun is computed as

$$\boldsymbol{s}_\mathcal{I} = \begin{bmatrix} \|\boldsymbol{s}_\mathcal{I}\| \cos\lambda_{ecl} \\ \|\boldsymbol{s}_\mathcal{I}\| \cos(\epsilon) \sin\lambda_{ecl} \\ \|\boldsymbol{s}_\mathcal{I}\| \sin(\epsilon) \sin\lambda_{ecl} \end{bmatrix}, \tag{22}$$

where $\boldsymbol{s}_\mathcal{I}$ has units of AU and is in the Mean-of-Date (MOD) coordinate frame. We approximate the MOD frame with the ECI frame after applying a rotation to account for precession only[e] [30]. We may then compute the spacecraft-to-sun vector, $\boldsymbol{r}_\odot$, by summing the ECI position vector from equation (4) with $\boldsymbol{s}_\mathcal{I}$.

The computation of the sun vector allows us to deduce whether the satellite is in the earth's shadow – an indication that information regarding the sun's position sensed by the spacecraft is invalid. The use of a boolean variable to indicate portions of the orbit in Umbra allows the GNC system to function properly when unable to see the sun. For example, sun seeking mode will not be activated if the spacecraft is in Umbra; this would lead to a futile control effort.

Finally, many mission objectives require communication and interaction with terrestrial stations. Therefore, knowledge of the spacecraft's inertial position is used by the GNC system to autonomously determine its line of sight to a ground station. For flexibility we have included two indicators: one that the spacecraft is simply above the ground station's horizon, and a second indicating that the spacecraft is within a cone of a given half-angle, centered at the ground station. The use of both of these allows the GNC system to be amenable to both omni-directional and directional antennas with limited pointing ability. Computation of the spacecraft-to-ground-station vector allows us to compute the attitude required to meet communication requirements and maneuver accordingly. Knowledge of the inertial position also gives the spacecraft time to perform maneuvers required for communication. If the trajectory will pass through the field of view cone (the green cone in Figure 4), a bus pointing command is issued to reorient the spacecraft and point a directional communication antenna at the ground station. A trajectory with both the local horizon and a conic region centered in Seattle, WA, can be seen in Figure 4. In this figure, the red area depicts the local horizon above Seattle, while the green area represents the cone centered in Seattle. This design choice was inspired by the dual communication antenna present for the HuskySat-1 mission.

4. Control

Based on the current operating GNC mode, we autonomously compute a desired quaternion, $\boldsymbol{q}_d$, and angular velocity vector, $\boldsymbol{\omega}_d$, to use as set points for controlling the spacecraft. A desired attitude state is computed regardless of mode, even though at times the control action may not use one or both states.

DETUMBLE: The objective of the detumble mode is to remove momentum from the spacecraft bus. There are two common actuation methods to achieve this goal, either through magnetorquers or reaction wheels. Reaction wheels exchange momentum with the bus, while magnetorquers transfer the momentum to the earth. Since the use of reaction

[e]The error of this approximation is about 0.0002 AU, which is accurate enough for our purposes

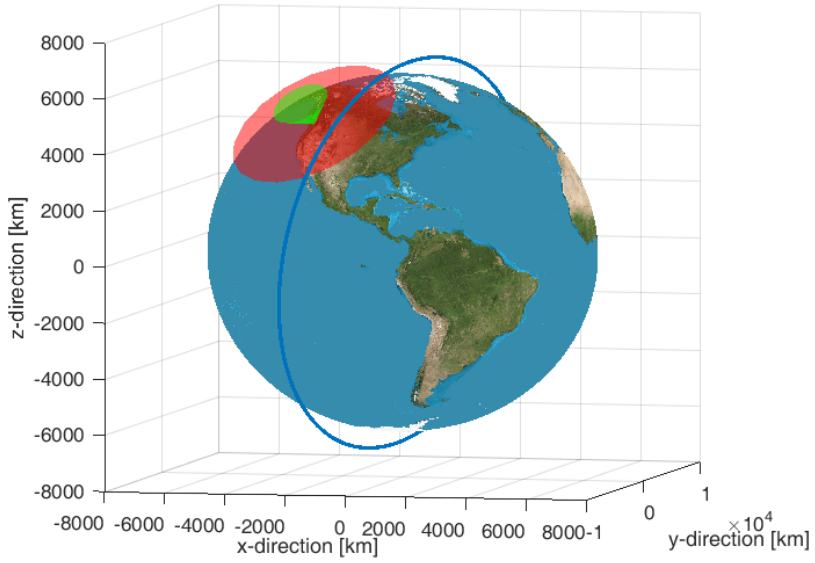


Figure 4. Example of satellite trajectory in which the spacecraft is above the horizon, but not within a 45 degree field of view cone about Seattle, WA.

wheels for this task may employ the same control law described later for the Bus Pointing mode, we focus on solely magnetic control laws here.

Traditionally, magnetic detumbling is achieved through the "B-dot" algorithm [35]. Since Euler's equations (1) are derived by taking a derivative of momentum in the rotating body frame, we can use the spacecraft's angular velocity as a surrogate for momentum. Assuming that the magnetic field lines are constant (locally) over small time periods in the inertial frame, we may say that

$$\dot{\boldsymbol{b}} \propto \boldsymbol{b}^{\times}\boldsymbol{\omega}. \tag{23}$$

This allows us to use the sensed change in the magnetic field vector to approximate the momentum in the system. Magnetometer readings are taken, smoothed, and used to compute a numerical approximation to $\dot{\boldsymbol{b}}$. The control law then uses this information to compute a desired magnetic moment for the magnetorquers according to

$$\boldsymbol{m}_{c,m} = -K\dot{\boldsymbol{b}}, \tag{24}$$

where $K \in \mathbb{S}^3_{++}$ is a positive definite matrix of control gains and $\boldsymbol{m}_{c,m}$ is the commanded magnetic dipole of the magnetorquers. The torque on the spacecraft will then be generated according to equation (19).

The resulting control action can be shown to be asymptotically stabilizing and was originally proposed in [35]. Using physical parameters taken from the HuskySat-1 mission (see §III) the B-dot controller was validated in a full simulation test. The initial tumble rate was $\boldsymbol{\omega}(t_0) = [0.2\ 0.2\ 0.2]^T$ rad/s and the angular velocity over two orbits is given in Figure 5. One can see that the angular velocity (and hence angular momentum) is nulled in roughly 7000 seconds from this initial rate. We believe that an initial tumble rate of 0.36 rad/s RSS is a realistic upper bound for deployment scenarios. Hence, this test offers evidence that the GNC system is capable of detumbling the satellite.

SUN SEEKING: The sun seeking controller uses an eigenaxis rotation to point the sun sensor directly at the sun's estimated position in the body frame. Since this controller may be called upon when the sun is not already in the field of view of the sun sensor, it is necessary to have other methods of sensing the sun. A possible solution is photodiodes, which can be placed on all sides of the spacecraft's body. These can provide a coarse estimate of the sun's position, should it be shining on (at least) one side of the spacecraft. Photodiode data may be used as a binary indication of sun incidence on each face that does not have a sun sensor. Subsequent logic then selects the side of the spacecraft whose photodiodes register the highest voltage, yielding a single unit normal as the estimated sun direction. This method, as opposed to blending all readings, may help alleviate false-positive photodiode voltage caused by earth's albedo.

To compute the control action, we first compute the vector part of the error quaternion, $\delta\boldsymbol{q}$. Using standard quaternion composition rules, we can express the vector portion by the Euler axis $\hat{\boldsymbol{n}}$ and angle θ, so that $\delta\boldsymbol{q}_v = \hat{\boldsymbol{n}} \sin\frac{\theta}{2}$,

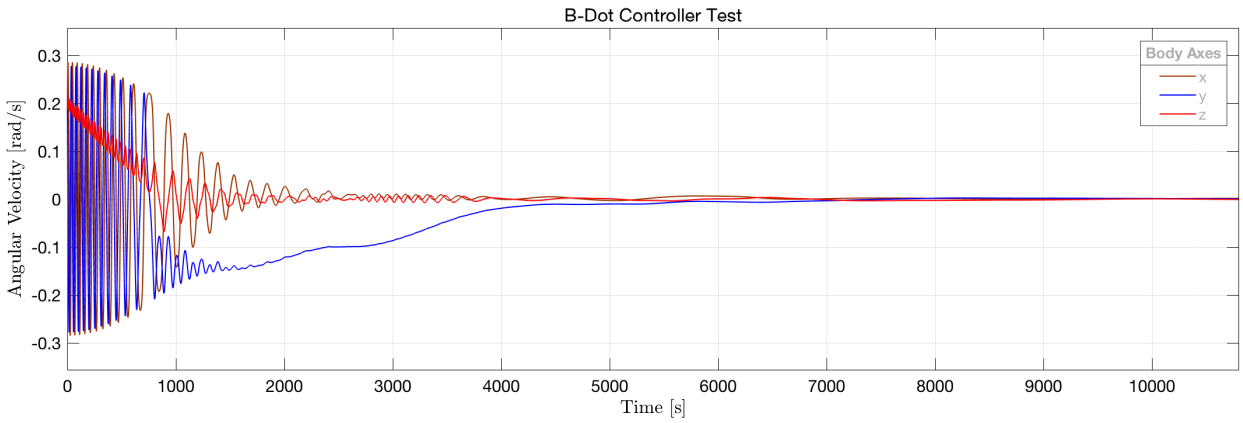


Figure 5. Angular velocity time history for a detumble maneuver using the B-dot controller (24).

where

$$\hat{\boldsymbol{n}} = \hat{\boldsymbol{s}}_{\mathcal{B}} \times \hat{\boldsymbol{p}}_{\mathcal{B}}, \quad \theta = \cos^{-1}(\hat{\boldsymbol{s}}_{\mathcal{B}} \cdot \hat{\boldsymbol{p}}_{\mathcal{B}}),$$

where $\hat{\boldsymbol{s}}_{\mathcal{B}}$ is the unit vector pointing to the sun (estimated from photodiode measurements) in the spacecraft body frame and $\hat{\boldsymbol{p}}_{\mathcal{B}}$ is the constant body boresight vector of the sun sensor. The control torque is then taken to be

$$\boldsymbol{\tau}_{c,w} = K_p \delta \boldsymbol{q}_v + K_d \boldsymbol{\omega}, \tag{25}$$

where $K_p, K_d \in \mathbb{S}^3_{++}$ are positive definite gain matrices. The second term in the control law aims to null angular rates for a rest-to-rest maneuver. Stability of the closed-loop system can be established using LaSalle's theorem [33], or the Lyapunov technique described in the next section. As a result, the control law regulates the attitude of the spacecraft to the orientation where $\hat{\boldsymbol{s}}_{\mathcal{B}}$ is inline with $\hat{\boldsymbol{p}}_{\mathcal{B}}$ and the sun is centered in the field of view of the sun sensor. Per the sun sensor model introduced in §II.C, this implies that the angles α and β are ideally zero. Figure 6 shows a time history of these angles for a given maneuver using the HuskySat-1 satellite parameters. The satellite begins at an orientation where the sun sensor does not have the sun in its 60 degree half-angle field of view. The output of the sensor is saturated to $\alpha = 60$ degrees during this time. The satellite is then reoriented and the control objective is achieved in 1-2 minutes. For this example, the GNC system is forced to remain in sun seeking mode for the duration of the test.

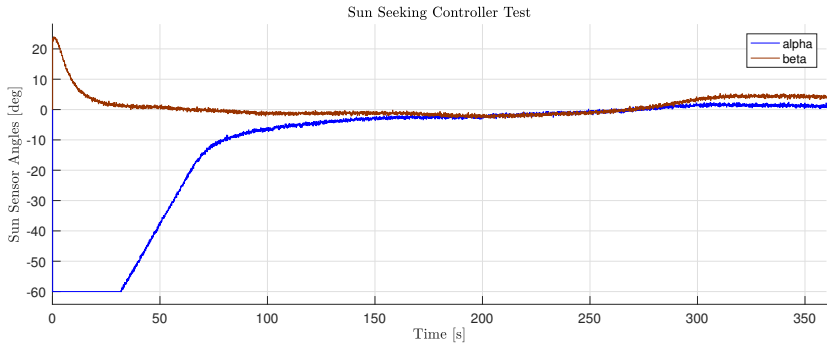


Figure 6. Measured sun sensor angles over time as the sun seeking controller (25) reorients the satellite so that the sun sensor is pointed directly towards the sun.

BUS POINTING: The control law used for basic reorientation of the spacecraft bus assumes a similar proportional-derivative form as equation (25). We assume access to full state information in deriving the error quaternion for this case. This allows us to embed both a primary and secondary pointing objective into the resulting motion; consistent with the fact that upon alignment of one body and inertial vector, there remains a single degree of freedom about the common axis with which to achieve some secondary objective.

Given an arbitrary boresight vector in the body frame, $\boldsymbol{n}_{\mathcal{B}}$, and an inertial vector ,$\boldsymbol{p}_{\mathcal{I}}$, the primary objective is to align these vectors. We take as secondary objective the minimization of the angle between the sun vector in the body frame, $\boldsymbol{s}_{\mathcal{B}}$, and the inertial frame, $\boldsymbol{s}_{\mathcal{I}}$. We assume henceforth that each of these vectors are unit vectors.

The error quaternion is computed by first creating an intermediate reference frame and then projecting the components of this frame in the body and inertial frames [33]. The difference between these projections is defined as the error state. The unit vectors that define these projections and the orthogonal matrix relating the two are

$$\boldsymbol{x}'_{\mathcal{B}} = \boldsymbol{n}_{\mathcal{B}}, \quad \boldsymbol{y}'_{\mathcal{B}} = \boldsymbol{n}_{\mathcal{B}} \times \boldsymbol{s}_{\mathcal{B}}, \quad \boldsymbol{z}'_{\mathcal{B}} = \boldsymbol{n}_{\mathcal{B}} \times \boldsymbol{y}'_{\mathcal{B}}, \tag{26a}$$

$$\boldsymbol{x}'_{\mathcal{I}} = \boldsymbol{p}_{\mathcal{I}}, \quad \boldsymbol{y}'_{\mathcal{I}} = \boldsymbol{p}_{\mathcal{I}} \times \boldsymbol{s}_{\mathcal{I}}, \quad \boldsymbol{z}'_{\mathcal{I}} = \boldsymbol{p}_{\mathcal{I}} \times \boldsymbol{y}'_{\mathcal{I}}, \tag{26b}$$

$$\delta C = \begin{bmatrix} \boldsymbol{x}'_{\mathcal{B}} & \boldsymbol{y}'_{\mathcal{B}} & \boldsymbol{z}'_{\mathcal{B}} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}'^{T}_{\mathcal{I}} \\ \boldsymbol{y}'^{T}_{\mathcal{I}} \\ \boldsymbol{z}'^{T}_{\mathcal{I}} \end{bmatrix}. \tag{26c}$$

The error quaternion, $\delta\boldsymbol{q}$, is then computed from the error direction cosine matrix δC using standard conversions [29, 33, 36]. Note that at least one of the four vectors used in the derivation is time-varying.

The reorientation controller uses a PD-control law augmented with gyroscopic decoupling feedback term [33, 37] and has the form

$$\boldsymbol{\tau}_{c,w} = K_p \delta\boldsymbol{q}_v + K_d\left(\boldsymbol{\omega} - \boldsymbol{\omega}_d\right) + \boldsymbol{\omega}^{\times} J_w \Omega, \tag{27}$$

where $K_p, K_d \in \mathbb{S}^3_{++}$ are positive definite gain matrices to be selected so that the closed loop system has desired properties. The third term in the control law in equation (27) cancels out the same term in the attitude dynamics in equation (1), to the accuracy of the estimated $\boldsymbol{\omega}$. The stability of the closed-loop system in equation (27), assuming no disturbances and perfect state estimation, can be established using similar analysis to [37]. We summarize the steps here, but refer the reader to [37] for complete details. Consider the Lyapunov function

$$V = \frac{1}{2}\boldsymbol{\omega}^T K_p^{-1} J \boldsymbol{\omega} + 2(1 - q_0), \tag{28}$$

which is positive definite so long as K_p^{-1} exists and $K_p^{-1}J$ is positive definite and is asymptotically unbounded in the angular velocity[f]. Along trajectories of the system, we find using equations (1) and (2) that

$$\dot{V} = -\boldsymbol{\omega}^T K_p^{-1} K_d \boldsymbol{\omega}$$

and therefore global stability is guaranteed if $K_p^{-1}K_d$ is a positive definite matrix. Now, since the inertia matrix for all real spacecraft is positive definite, a natural choice for the gain matrices to guarantee this condition is $K_p = aJ$ and $K_d = bJ$, where $a, b > 0$ are real scalar coefficients to be designed.

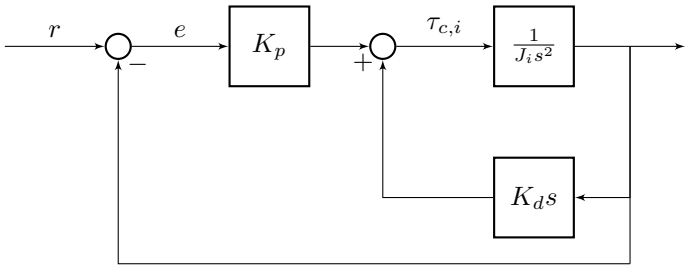


Figure 7. Linearized block diagram for the closed-loop attitude dynamics under control law (27).

To select appropriate coefficients a and b, we turn to linear system analysis. Moreover, we assume that $J = \mathbf{diag}\{J_1, J_2, J_3\}$. A linearized model of the closed loop system (assuming the gyroscopic terms, noise, and disturbances are negligible) is given in Figure 7. Under these assumptions, the attitude dynamics become a set of decoupled, double integrator systems, and the closed-loop transfer function for each axis is

$$T(s) = \frac{a}{s^2 + bs + a}, \tag{29}$$

since we've assumed $K_p = aJ$ and $K_d = bJ$. Comparing this equation to the standard second order system form [38] we can see that

$$\frac{a}{s^2 + bs + a} \iff \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad \Rightarrow \quad a = \omega_n^2,\ b = 2\zeta\omega_n,$$

[f]Recall that the scalar entry in the unit quaternion is constrained to lie in $[-1, 1]$ so that the second term in equation (28) is always nonnegative.

where ω_n is the undamped natural frequency and ζ is the damping ratio of the second order system. Therefore, we see that by choosing the gain coefficients a, b, we can effectively design the response of the linearized system. The gain matrices are therefore chosen as

$$K_p = \omega_n^2 J \quad \text{and} \quad K_d = 2\omega_n \zeta J.$$

We choose a critical damping ratio of $\zeta = 1$ and a small undamped natural frequency $\omega_n \in [0.01, 0.1]$ rad/s. The damping ratio ensures that the overshoot to a step response is null for the ideal linear system. Small choices for ω_n yields an equally small closed-loop bandwidth, allowing the system to respond well to low-frequency command signals while attenuating high-frequency noise.

A complete robustness analysis with regards to noise and disturbances is beyond the scope of the current paper and will be addressed in future work. To validate this controller through simulation, we can expand on the previous example for the sun seeking controller. Here, we start from the same attitude so that the sun is initially not in the field of view of the sun sensor. Hence the GNC system must first reorient the satellite and then change operating modes to achieve the pointing objective. For this example we have selected a velocity pointing maneuver whereby the $\boldsymbol{z}_{\mathcal{B}}$ axis is aligned with the inertial velocity vector, $\boldsymbol{v}_{\mathcal{I}}$, of the spacecraft. To do so, we generate the desired time-varying attitude quaternion using equation (26) with $\boldsymbol{z}_{\mathcal{B}}$ and $\boldsymbol{v}_{\mathcal{I}}$ as the primary alignment vectors. The desired angular velocity is assumed to be zero, since the stability analysis is valid only for rest-to-rest maneuvers.

Figure 8 shows both the commanded and actual attitude quaternions for this maneuver. Note that the dashed lines – the commanded values – represent the constant identity quaternion until such time that the reorientation controller in equation (27) takes over, which occurs at roughly the 60 second mark. The reason for this is that during this time the GNC system is using equation (25) to point the sun sensor at the sun, exactly as seen in Figure 6 in the previous section. The final angle error along the eigenaxis is 2.5 degrees. The impact of disturbance torques that have not yet been taken into account in the control design, as well as attitude knowledge error are two possible reasons for the discrepancy. Future work to increase robustness to noise and disturbances will address this shortcoming.

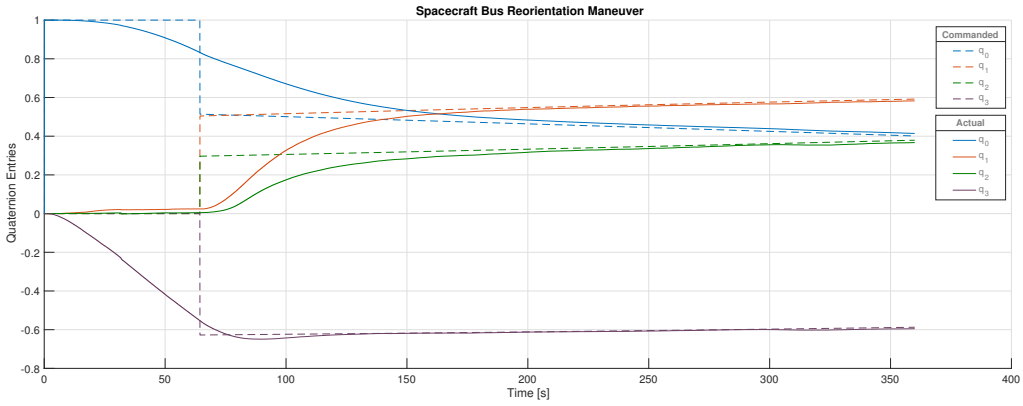


Figure 8. True versus commanded attitude quaternion for a reorientation maneuver. The true quaternion entries are represented by solid lines, while the commanded entries are dashed.

MOMENTUM DUMPING: One of the primary drawbacks to using momentum exchange devices, such as reaction wheels, is that they cannot be used to remove momentum from the combined system. Since disturbance torques – such as those discussed in §II.B – cause the momentum of the combined wheel-spacecraft system to accumulate, momentum dumping is necessary. Without it, the disturbance torques will either cause reaction wheels to saturate and lose their ability to control the spacecraft's orientation, or cause the spacecraft bus to tumble. Momentum dumping is thus important to GNC systems that use reaction wheels.

Fortunately, magnetorquers have the ability to generate an external torque on the system that can be leveraged for momentum dumping. We adopt the convention that momentum dumping can be performed in parallel with the attitude pointing maneuvers discussed in the previous section. Moreover, to avoid forcing the reaction wheels to operate at low RPM, we aim to maintain a set amount of momentum in the system. The objective is thus to maintain a prescribed amount of angular momentum in the spacecraft-wheel system without compromising the ability to meet attitude pointing objectives.

To perform continuous dumping of the reaction wheel momentum, consider again the simplified control system depicted in Figure 7. The torque command, $\boldsymbol{\tau}_{c,w}$, serves as the commanded set point for the reaction wheel and can be integrated to obtain the momentum, $\boldsymbol{h}_w$. Consider the proportional control law

$$\boldsymbol{\tau}_{c,m} = -K_w(\boldsymbol{h}_w - \boldsymbol{h}_{\text{ref}}),$$

where $\boldsymbol{h}_{ref}$ is some non-zero reference momentum and K_w is a positive definite gain matrix to be selected. Note that because we are using magnetorquers to realize this control torque, only the component of the vector orthogonal to the magnetic field will yield useful torque (see equation (19)). Therefore, it is practical to only command torque in this direction to begin with. This results in the control law

$$\boldsymbol{\tau}_{c,m} = -K_w \frac{\boldsymbol{b}_{\mathcal{B}}^{\times}}{||\boldsymbol{b}_{\mathcal{B}}||^2}(\boldsymbol{h}_w - \boldsymbol{h}_{\text{ref}}), \tag{30}$$

where $\boldsymbol{b}_{\mathcal{B}}$ is the magnetic field vector in the body frame. This is effectively a projection of the torque onto the plane orthogonal to $\boldsymbol{b}_{\mathcal{B}}$, and has appeared in similar works [39, 40]. This control law generates a torque in the same direction as the momentum of the reaction wheels, causing them to reduce angular velocity to maintain attitude stability.

Stability of the control law in equation (30) has been established several ways. In [34] the author shows that the steady-state value of the simplified closed-loop system shown in Figure 7 subject to a constant disturbance decreases linearly with the magnitude of the entries in K_w. In [39, 40], the authors use Lyapunov techniques to establish stability of the nonlinear system with no disturbances. We remark that [39] presents several other control laws that may be more apt than equation (30) for continuous momentum dumping without sacrificing attitude pointing accuracy, and these will be explored in future work.

Figure 9 provides a comparison of reaction wheel momentum (magnitude only) during the reorientation maneuver shown in Figure 8. At first, the maneuver is performed without using the control law in equation (30), i.e. $K_w = 0$. Next the maneuver is performed with $K_w = 10I_{3\times3}$ and results in less momentum stored in the reaction wheels for the same maneuver.

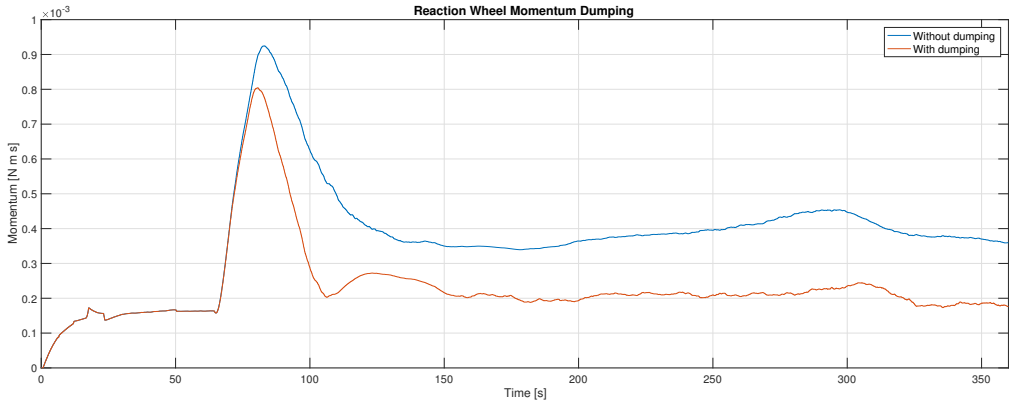


Figure 9. Reaction wheel momentum with and without continuous desaturation during a reorientation maneuver.

5. *Estimation*

Accurate attitude control cannot be achieved without proper attitude knowledge. In fact, each of the control laws in §4 relied on some degree of attitude state feedback. The need to accurately determine our attitude during nominal operation has conscripted us to examine several estimation routines. We have adopted two methods that differ in estimation accuracy and computational requirements. A multiplicative extended Kalman filter (MEKF) for attitude and gyroscope bias estimation provides precise attitude knowledge. A memory-less, low-precision TRIAD algorithm provides a fast and computationally inexpensive estimation routine to compliment the MEKF and add robustness to the system. This allows us, in the case of gyroscope failure in-flight, to maintain enough attitude knowledge to meet reduced pointing requirements and maintain functionality. While there are several other methods for attitude estimation, such as the q-method, Wahba's method and unscented filters, we have found that these two offer an acceptable trade of accuracy, computation, flight heritage and ease of implementation.

In this section, we provide an outline of the two algorithms, the sensor measurements required for each, and close with a summary of the on-board logic that switches between the two attitude algorithms.

The MEKF seeks to estimate a full attitude state vector, which we denote by $[\hat{\boldsymbol{q}}\ \hat{\boldsymbol{\omega}}]^T \in \mathbb{R}^7$. The filter uses a first order approximation to the error quaternion dynamics, which effectively allows us to decrease the order of the filter by one by assuming the scalar entry of the error quaternion is near unity [36]. We will use $\delta\hat{\boldsymbol{\alpha}}$ to denote the vector part of the error quaternion, used as a state in the MEKF in place of $\hat{\boldsymbol{q}}$. The MEKF estimates the gyroscope bias, rather than the angular velocity of the spacecraft directly. Consistent with equation (13), we use $\hat{\boldsymbol{\chi}}$ to denote the estimated gyroscope bias. To ease notational burden, all of the following quantities are assumed to be calculated at the discrete time 'k'. Vectors with a superscript $^-$ denote pre-update quantities, while those with a superscript $^+$ denote the corresponding post-update quantity.

The *full error model* is given in continuous time by

$$\Delta\dot{\boldsymbol{x}} = F\Delta\boldsymbol{x} + G\boldsymbol{w},$$

where

$$\Delta\boldsymbol{x} = \begin{bmatrix} \delta\hat{\boldsymbol{\alpha}} \\ \delta\hat{\boldsymbol{\chi}} \end{bmatrix},\ \boldsymbol{w} = \begin{bmatrix} \boldsymbol{\epsilon}_a \\ \boldsymbol{\epsilon}_r \end{bmatrix},\ F = \begin{bmatrix} -\hat{\boldsymbol{\omega}}^\times & -I_{3\times3} \\ 0_{3\times3} & 0_{3\times3} \end{bmatrix},\ G = \begin{bmatrix} -I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I_{3\times3} \end{bmatrix},\ Q = \begin{bmatrix} \sigma_a^2 I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & \sigma_r^2 I_{3\times3} \end{bmatrix}.$$

where $(\boldsymbol{\epsilon}_a, \boldsymbol{\epsilon}_r, \sigma_a, \sigma_r)$ are the same parameters used to model ARW and RRW in our gyroscope model in equation (13).

The measurement model for the MEKF is given by

$$\tilde{\boldsymbol{y}} = \begin{bmatrix} C(\hat{\boldsymbol{q}}^-)\boldsymbol{b} \\ C(\hat{\boldsymbol{q}}^-)\boldsymbol{s} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\nu}_1 \\ \boldsymbol{\nu}_2 \end{bmatrix} := \boldsymbol{h}(\hat{\boldsymbol{x}}^-) + \boldsymbol{v},$$

where $C(\boldsymbol{q})$ denotes the attitude matrix corresponding to quaternion $\boldsymbol{q}$, $\boldsymbol{b}$ is the magnetic field vector and $\boldsymbol{s}$ is the sun vector. The vector $\boldsymbol{v}$ is used to model sensor noise and is assumed to be a zero-mean Gaussian white noise process. The sensitivity matrix is then given by

$$H(\hat{\boldsymbol{x}}^-) = \begin{bmatrix} (C(\hat{\boldsymbol{q}}^-)\boldsymbol{b})^\times & 0_{3\times3} \\ (C(\hat{\boldsymbol{q}}^-)\boldsymbol{s})^\times & 0_{3\times3} \end{bmatrix}.$$

The state of the MEKF is updated in discrete time according to the equations

$$K = P^- H^T(\hat{\boldsymbol{x}}^-)\left[H(\hat{\boldsymbol{x}}^-)P^- H^T(\hat{\boldsymbol{x}}^-) + R\right]^{-1} \tag{31a}$$

$$P^+ = \left(I - KH(\hat{\boldsymbol{x}}^-)\right)P^- \tag{31b}$$

$$\Delta\hat{\boldsymbol{x}}^+ = K\left(\tilde{\boldsymbol{y}} - h(\hat{\boldsymbol{x}}^-)\right) \tag{31c}$$

$$\hat{\boldsymbol{q}}^+ = \left(\hat{\boldsymbol{q}}^- + \Xi(\hat{\boldsymbol{q}}^-)\delta\hat{\boldsymbol{\alpha}}^+\right)/\|\hat{\boldsymbol{q}}^- + \Xi(\hat{\boldsymbol{q}}^-)\delta\hat{\boldsymbol{\alpha}}^+\| \tag{31d}$$

$$\hat{\boldsymbol{\chi}}^+ = \hat{\boldsymbol{\chi}}^- + \delta\hat{\boldsymbol{\chi}}^+, \tag{31e}$$

where K is the Kalman gain, P is the covariance matrix and,

$$\Xi(\boldsymbol{q}) = \begin{bmatrix} q_0 I_{3\times3} + \boldsymbol{q}_v^\times \\ -\boldsymbol{q}_v^T \end{bmatrix}.$$

Lastly, these quantities are propagated forward to the next time-step (see §7.1.2 of [36]). The left hand side of each of these equations denote pre-update quantities at discrete time instance '$k+1$'

$$P^- = \Phi P^+ \Phi^T + \Gamma Q \Gamma^T, \tag{32a}$$

$$\hat{\boldsymbol{\omega}}^+ = \bar{\boldsymbol{\omega}} - \hat{\boldsymbol{\chi}}^+, \tag{32b}$$

$$\hat{\boldsymbol{q}}^- = \Omega(\hat{\boldsymbol{\omega}}^+)\hat{\boldsymbol{q}}^+, \tag{32c}$$

where the matrices Φ, Γ, Q, Ω are all given in the Appendix, and $\bar{\boldsymbol{\omega}}$ is the measured angular velocity from the gyroscope.

The TRIAD algorithm computes the orthogonal matrix that rotates vectors from the spacecraft body frame to the ECI frame. It is equivalent to the attitude quaternion through equation (3). Unlike the MEKF, the TRIAD algorithm does not update a previous quaternion estimate, nor does it propagate the quaternion estimate forward it time. Rather, it computes a new quaternion at every time-step from four non-parallel vectors; two inertial and two body frame.

Let the linearly independent (unit) vectors in the inertial frame be $\boldsymbol{b}_{\mathcal{I}}$ and $\boldsymbol{s}_{\mathcal{I}}$, and let the linearly independent unit vectors in the body frame be $\boldsymbol{b}_{\mathcal{B}}$ and $\boldsymbol{s}_{\mathcal{B}}$. The attitude matrix relating the two frames is estimated by

$$\hat{C}^T = \left[\begin{array}{c|c|c} \boldsymbol{b}_{\mathcal{I}} & \boldsymbol{s}_{\mathcal{I}} & \boldsymbol{b}_{\mathcal{I}} \times \boldsymbol{s}_{\mathcal{I}} \end{array} \right] \left[\begin{array}{c|c|c} \boldsymbol{b}_{\mathcal{B}} & \boldsymbol{s}_{\mathcal{B}} & \boldsymbol{b}_{\mathcal{B}} \times \boldsymbol{s}_{\mathcal{B}} \end{array} \right]^T. \tag{33}$$

Like the MEKF, the TRIAD algorithm uses the on-board magnetometer and sun sensor to measure the reference vectors in the spacecraft body frame. Vectors in the spacecraft's inertial frame are computed using models of the Earth's magnetic field and spacecraft-to-sun vector as described in §II.D.3.

The fundamental difference between the MEKF and the TRIAD algorithms is that the MEKF contains a step predicting future attitudes by utilizing knowledge about the spacecraft rotation rates. Thus, the gyroscopes provide the final sensor measurement used in the MEKF, yielding the spacecraft's angular velocity in the body frame.

At various points during spacecraft operation, either the MEKF or TRIAD may be better suited for use in obtaining state information. The accuracy of the MEKF dictates that it be used during precise pointing modes, with TRIAD able to serve as a back-up in the event of gyroscope failure. During periods of eclipse, the TRIAD algorithm will not have sufficient information to function, and thus we rely on the propagation of the attitude quaternion from gyroscopic measurements using the MEKF during these times. To this end, a state estimation library was created that runs both the MEKF and TRIAD algorithms in tandem, and then chooses the quaternion estimate based on the flight mode and sensor states. Predominantly during nominal operation, the TRIAD estimate is used during the detumble mode, while the MEKF is used otherwise.

III. HuskySat-1

The first mission to which the generic GNC system has been applied to is the HuskySat-1 mission. The principal requirements of the GNC system were to manage the momentum of the spacecraft and stabilize the bus about an attitude appropriate to demonstrate two scientific payloads. The orientation of the satellite was required to point the high-frequency antenna towards the UW ground station during passes over Seattle with an accuracy of 20 degrees. At the same time, the GNC system shall maintain this attitude within the same bound so that the PPT could be fired as much as possible in a consistent direction.

The sensor, actuator and micro-controller (MCU) suite used for this mission is described in Table 3. Given that actuation is only available from three magnetic torque rods, the HuskySat-1 GNC system is a purely magnetic control system similar to those described in [2, 41, 42]. This represents a special case of the generic GNC software described in §II, and indeed uses only a subset of the libraries presented there.

Table 3. Available GNC hardware for the final HuskySat-1 mission design.

Type	Model	Quantity
Magnetometer	Honeywell HMC 5983	3 (each three-axis)
Gyroscope (MEMS)	STMicroelectronics LSM6DSM	1
Sun Sensor (digital)	nanoSSOC-D60	1
Magnetorquers	Manufactured at UW	3 (one per axis)
Flight computer	TI MSP 430-FR5994	5

An exploded view of the satellite components is shown in Figure 12. The body $\boldsymbol{z}_{\mathcal{B}}$ axis points towards the high frequency antenna, the body $\boldsymbol{y}_{\mathcal{B}}$ axis points outwards from the central solar panel (there are some on the $\pm\, \boldsymbol{x}_{\mathcal{B}}$ faces), and the $\boldsymbol{x}_{\mathcal{B}}$ axis completes a right handed triad.

The GNC system is comprised of three dedicated stack boards and several distributed sensors and actuators. One stack board (top in Figure 12) receives low update rate sensors and pre-processes the readings before sending them to a second stack board (4th from the top) that houses the flight computers running our flight software. The flight software module computes torque commands for the magnetorquers, which are housed on the third stack board (2nd from the bottom). High update rate sensors are located on the same stack board as the flight computers to maximize data throughput. These modules use the on-board controller area network (CAN) bus to transfer data on the spacecraft.

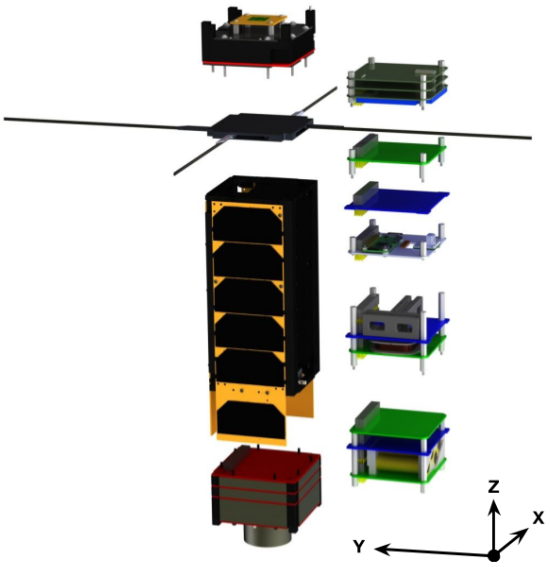


Figure 10. HuskySat-1 satellite overview.

Since there is no GPS sensor on-board, the GNC system is designed to propagate its inertial position using TLEs. The TLE stored in the spacecraft's persistent memory will be updated periodically to maintain reasonably accurate state information. The uncertainties and error in position and velocity that follow from this method are acceptable within the requirements of the mission.

To increase the robustness of the system, and due to very limited computational resources, the GNC system is split across four MCUs. The first contains the B-dot algorithm as a standalone entity that can process raw sensor readings and compute controls according to equation (24). The B-dot algorithm is given a dedicated magnetometer, but is also able to pull readings from the CAN bus should that sensor fail. The second MCU contains all sensor processing libraries and received data from two magnetometers, the gyroscope and the sun sensor. The third and fourth MCUs contain a portion of the core flight software each. One contains all libraries discussed in §II.D.3, while the other contains the remaining functions described in §II.D necessary for the mission. The reason for splitting the flight software in this way is due to limited computational resources. Splitting the tasks was made simple by the modular nature of the Simulink libraries.

When HuskySat-1 is released into orbit, the default detumble mode is engaged after 15 minutes, at which time the spacecraft regulates its angular velocity below a prescribed 0.25 deg/s within 1-2 orbital periods.

Once detumbling is achieved, the spacecraft will wait (in low power mode) for a ground station command before entering subsequent operational modes. This allows us to establish communication (via the low gain antenna) and run checks to ensure the health of the spacecraft prior to operation of any components beyond those required for detumble. Next, the spacecraft may enter a sun seeking mode if the sun is not in the field of view of the sun sensor. Otherwise the system will seek to point the spacecraft so that the sun sensor's boresight vector has maximal incidence to the sun and the angle of the high-frequency antenna's boresight ($z_{\mathcal{B}}$) points nadir. The GNC system shall maintain less than a 20° angle between the antenna's boresight and the nadir direction. Note that the HuskySat-1 system prioritizes the sun vector over nadir pointing, in contrast to equation (26). The fundamental concept of operations described above is shown graphically in Figure 11.

A. Attitude Control Hardware

The actuators used for attitude control were chosen for their ability to provide sufficient control authority to the satellite, in addition to low cost, simplicity, and reliability. To achieve the required pointing, a set of three magnetorquers were chosen and fabricated at UW.

The magnetorquer array is made of three aircore magnetorquers, oriented along the principal axes of the spacecraft. All magnetorquers consist of a 3D-printed spool manufactured in-house and are composed of PETG and wound with

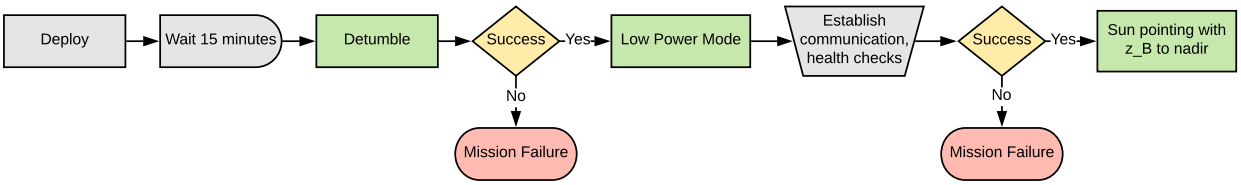


Figure 11. Concept of operations for the HuskySat-1 GNC system.

enameled copper magnet wire. Engineering models used to develop and test our algorithms are shown in Figure 12. The $x_{\mathcal{B}}$ and $y_{\mathcal{B}}$ axis magnetorquers must occupy less space than the $z_{\mathcal{B}}$ magnetorquer, and thus generate a smaller magnetic dipole. The $x_{\mathcal{B}}$ and $y_{\mathcal{B}}$ magnetorquers are composed of 230 turns of 31 gauge wire, and measure 60mm by 21mm. Each provides a dipole moment of 0.08 Am^2 and are shown on the left of Figure 12. The $z_{\mathcal{B}}$ axis magnetorquer consists of 300 turns of 29 gauge wire, measuring 63mm by 45 mm. The $z_{\mathcal{B}}$ axis magnetorquer produces a dipole moment of 0.22 Am^2 and is shown on the right of Figure 12. Each magnetorquer is driven by one Allegro A3903 H-bridge using 4.5 volts, and consumes 350 milliamps of current. The H-bridges are driven with PWM from an MCU, making sure to use a low driving frequency in order to avoid excessive back electromotive force. The driving MCU receives command packets from the core flight software through the CAN bus.

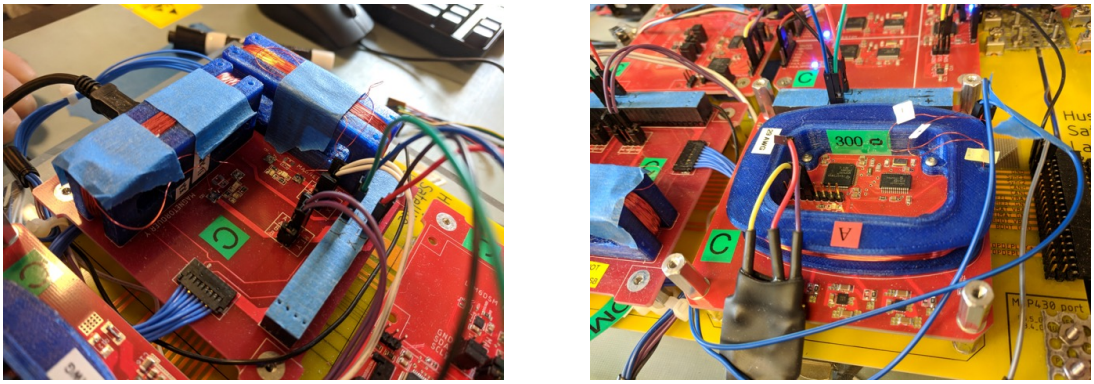


Figure 12. HuskySat-1's magnetorquer array, as pictured on the test bench. To the left is the board containing the $x_{\mathcal{B}}$ and $y_{\mathcal{B}}$ axis magnetorquers, and to the right is the board containing the $z_{\mathcal{B}}$ axis magnetorquer, the H-bridges, the voltage converters, and the magnetorquer's sole MCU.

B. Attitude Estimation Hardware

An array of sensors were chosen favoring commercial off-the-shelf components due to their low cost, extensive development resources, and immediate availability. The attitude determination requirements drove our choices for a magnetometer (capable of 2° absolute magnetic field vector determination) and inertial measurement unit (resolution of 0.22 arcminutes per second). The choice of sun sensor was driven by precision requirements over cost; we chose the nanoSSOC-D60 sun sensor due to its flight heritage and excellent precision. The sensor's field of view is 60 degrees, with an absolute precision of 0.1 degrees. All of these sensors communicate via either I^{2}C or UART, and are attached to a parent MCU for processing. The magnetometers are the most mission critical sensor and the most prone to failure, based on our environmental testing results. Therefore redundancy was favored in the design and three identical magnetometers were chosen. The three magnetometers are connected to two separate MCUs, but may share information through the CAN bus. Because of this design choice if a single sensor fails no software component loses functionality.

C. GNC System Validation

In order to validate HuskySat-1's GNC system, several test routines were developed to test both hardware and software. Each component was examined electrically at a set of common test points to affirm correct voltages. Environmental and mechanical properties were verified through thermal, vacuum and vibrational testing.

Once each component was verified on its own, additional tests were used to verify the entire system's functionality and to identify bugs. A hardware-in-the-loop (HIL) system was devised with the intent of testing specifically the B-dot

controller. The libraries devised to model the satellite's hardware were replaced by the actual sensors and actuators. This was achieved by taking magnetic field outputs from our environmental model module and injecting these values as I^2C packets to the magnetometer on the B-dot MCU using a Total Phase Aardvark. From there, the B-dot flight software processes the magnetometer readings, determines command dipole moments, and relays these over the CAN bus. The magnetorquer MCU receives these packets and executes them as coil currents, and a National Instruments DAQ reads the coil voltages, applies a known scale factor from voltage to dipole moment, and returns this value to the Simulink model via another custom defined block. The dynamics in equation (1) are then simulated to ensure proper functionality of the B-dot controller. This allows flight-configuration spacecraft hardware to be simulated as if it were subject to the same magnetic field as the space environment, and allows the spacecraft to control its orientation in the simulation, mirroring the space environment as closely as possible on the test bench. Further HIL testing is an ongoing process and will be reported on in future work.

IV. Conclusion

In this paper, we have presented the development of a generic set of tools for GNC flight software and simulation for small satellite missions. The software is being developed in Matlab/Simulink, and uses the autocode generation tool to obtain flight code. the flight software we have developed is supported by a cast of environmental models, as well as realistic sensor and actuator models to emulate the spacecraft's orbital environment. Initial testing reveals that the GNC system performs well and is able to meet a generic set of mission requirements. The work will support future spacecraft design missions at the University of Washington, and this paper examines in particular the inaugural mission, HuskySat-1. A subset of available GNC tools were used to design the flight software for the HuskySat-1 mission; meeting requirements within the limitations of the selected hardware.

A. Future Work

The generic GNC system is undergoing continued development to be used as a tool to support future advanced control and estimation algorithm development. Moreover, our system can be specialized for earth-imaging and technology demonstration missions that require little further GNC development. The objective is to use recent theoretical results obtained in convex optimization based attitude control to design a practical spacecraft GNC system. Historically, attitude constraints have not been handled using convex optimization for on-board systems. Instead, algorithms that exploit the geometry of the quaternion manifold via feasible waypoints [43], constraint monitoring algorithms [44, 45] or potential function algorithms [20, 46] have been used primarily.

The advantage of convex optimization based methods is that system level constraints can be more readily guaranteed by an autonomous GNC system. Further, advances in on-board computation has made their use feasible for small satellite GNC systems. Examples of constraints include, among others, ensuring sensitive payloads are not exposed directly to sunlight, or that certain regions remain in the field of view of a sensor.

Current and future work involves the integration of convex optimization based algorithms that leverage recent theoretical advances in attitude control. Integrating these algorithms into our simulation environment serves as a next step towards implementation on subsequent missions, while helping expand the capabilities of small satellite GNC systems.

V. Acknowledgements

The authors acknowledge the support of Dr. Robert Winglee and members of the Advanced Propulsion Lab who led the development of both the PPT and the high-frequency antenna, while serving as project managers for the HuskySat-1 mission.

Appendix

The matrices in equation (32) are given below, where Δt is the sample rate of the gyroscope. Recall that our notation for $\hat{\omega}^+$ describes the post-update estimated angular velocity at the kth time step. Moreover, the parameters (σ_a, σ_r) are the ARW/RRW values from the gyroscope model in equation (13).

$$\Phi = \begin{bmatrix} \Phi_{11} & \Phi_{12} \\ 0_{3\times3} & I_{3\times3} \end{bmatrix}, \quad \Gamma = \begin{bmatrix} -I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I_{3\times3} \end{bmatrix}, \quad Q = \begin{bmatrix} \left(\sigma_a^2\Delta t + \frac{1}{3}\sigma_r^2\Delta t^3\right) I_{3\times3} & \left(\frac{1}{2}\sigma_r^2\Delta t^2\right) I_{3\times3} \\ \left(\frac{1}{2}\sigma_r^2\Delta t^2\right) I_{3\times3} & (\sigma_r^2\Delta t) I_{3\times3} \end{bmatrix}$$

$$\Phi_{11} = I_{3\times3} - (\hat{\boldsymbol{\omega}}^+)^\times \frac{\sin(\|\hat{\boldsymbol{\omega}}^+\|\Delta t)}{\|\hat{\boldsymbol{\omega}}^+\|} + (\hat{\boldsymbol{\omega}}^+)^\times(\hat{\boldsymbol{\omega}}^+)^\times \frac{1-\cos(\|\hat{\boldsymbol{\omega}}^+\Delta t)}{\|\hat{\boldsymbol{\omega}}^+\|^2}$$

$$\Phi_{12} = -I_{3\times3}\Delta t + (\hat{\boldsymbol{\omega}}^+)^\times \frac{1-\cos(\|\hat{\boldsymbol{\omega}}^+\Delta t)}{\|\hat{\boldsymbol{\omega}}^+\|^2} - (\hat{\boldsymbol{\omega}}^+)^\times(\hat{\boldsymbol{\omega}}^+)^\times \frac{\|\hat{\boldsymbol{\omega}}^+\|\Delta t - \sin(\|\hat{\boldsymbol{\omega}}^+\|\Delta t)}{\|\hat{\boldsymbol{\omega}}^+\|^3}$$

$$\Omega(\hat{\boldsymbol{\omega}}^+) = \begin{bmatrix} \cos\left(\frac{1}{2}\|\hat{\boldsymbol{\omega}}^+\|\Delta t\right) I_3 - \hat{\varphi}^\times & \hat{\varphi} \\ -\hat{\varphi}^T & \cos\left(\frac{1}{2}\|\hat{\boldsymbol{\omega}}^+\|\Delta t\right) \end{bmatrix}, \quad \hat{\varphi} = \frac{1}{\|\hat{\boldsymbol{\omega}}^+\|}\sin\left(\frac{1}{2}\|\hat{\boldsymbol{\omega}}^+\Delta t\right)\hat{\boldsymbol{\omega}}^+$$

References

[1] Miller, D., Saenz-Otero, A., Wertz, J., Chen, A., Berkowski, G., Brodel, C., Carlson, S., Carpenter, D., Chen, S., Cheng, S., Feller, D., Jackson, S., Pitts, B., Perez, F., and Szuminski, J., "Spheres: A testbed for long duration satellite formation flying in micro-gravity conditions," *Advances in the Astronautical Sciences*, Vol. 105 I, 2000, pp. 167–179.

[2] Leonard, S. B., "NPSAT1 Magnetic Attitude Control System," *16th Annual AIAA/USU Conference on Small Satellites*, 2002, pp. 7.

[3] McGrogan, D., *Hardware and High Data Speeds on the CINEMA CubeSat*, Master's thesis, EECS Department, University of California, Berkeley, May 2010.

[4] Senatore, P., Klesh, A., Zurbuchen, T. H., McKague, D., and Cutler, J., "Concept, Design, and Prototyping of XSAS: A High Power Extendable Solar Array for CubeSat Applications," *40th Aerospace Mechanisms Symposium*, 2010, pp. 431–444.

[5] Alminde, L., Bisgaard, M., Vinther, D., Viscor, T., and Østergard, K., "Educational value and lessons learned from the AAU-Cubesat project," *RAST 2003 - Proceedings of International Conference on Recent Advances in Space Technologies*, 2003, pp. 57–62.

[6] Cooper, C., Walker, C., and Patel, J., "The CubeSat Ground Station at the University of Arizona," *Proceedings of the 16th AIAA/USU Conference on Small Satellites*, 2002, pp. 1–9.

[7] Cassady, R., Hoskins, W., Campbell, M., and Rayburn, C., "A Micro Pulsed Plasma Thruster (PPT) for the "Dawgstar" Spacecraft," *2000 IEEE Aerospace Conference. Proceedings (Cat. No.00TH8484)*, Vol. 4, 2000.

[8] Campbell, M., Ewig, R., Beltran, E., Carlson, E., Carpenter, K., Davenport, B., Halligan, J. K., Hammack, J., Kaltenbach, S., Lai, G., and Shell, W. M., "Uw Dawgstar : One Third of Ion-F," *AIAA/USU Conference on Small Satellites*, Logan, Utah, 1999, pp. 1–13.

[9] Miller, D. W. and Mallory, G. J., "Control testbeds and flight demonstrations: Transitioning theory to application," *Proceedings of the American Control Conference*, Vol. 2, 1998, pp. 873–878.

[10] Chen, A., Saenz-otero, A., Hilstad, M., and Miller, D., "Development of Formation Flight and Docking Algorithms Using the SPHERES Testbed," *Utah State University Conference on Small Satellites*, Logan, Utah, 2001, pp. 1–14.

[11] Otero, A. S., Miller, D. W., and Hilstad, M., "SPHERES : a Laboratory for Formation Flight and Docking," *DCSSS Conference*, Cambridge, UK, 2002, pp. 1–59–1–73.

[12] Orr, N. G., Eyer, J. K., Larouche, B. P., and Zee, R. E., "Precision formation flight: The CanX-4 and CanX-5 dual nanosatellite mission," *European Space Agency, (Special Publication) ESA SP*, 2008.

[13] Bonin, G., Orr, N., Armitage, S., Roth, N., Risi, B., and Zee, R. E., "The CanX-4&5 Mission: Achieving Precise Formation Flight at the Nanosatellite Scale," *64th International Astronautical Congress*, 2013, pp. 1–6.

[14] Roth, N. H., Risi, B., Grant, C. C., and Zee, R. E., "Flight Results from the CanX-4 and CanX-5 Formation Flying Mission," *4S Symposium*, 2016, pp. 1–15.

[15] Mesbahi, M. and Hadaegh, F., "Formation Flying Control of Multiple Spacecraft: Graph Theoretic properties and switching schemes," *AIAA Guidance, Navigation, and Control (GNC) Conference*, Portland, 1999, pp. 1585–1595.

[16] Mesbahi, M. and Hadaegh, F. Y., "Formation Flying Control of Multiple Spacecraft via Graphs, Matrix Inequalities, and Switching," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 2, 2001, pp. 369–377.

[17] Kim, Y. and Mesbahi, M., "Quadratically Constrained Attitude Control via Semidefinite Programming," *42nd IEEE Conference on Decision and Control*, Maui, HI, 2003, pp. 731–735.

[18] Kim, Y., Mesbahi, M., and Hadaegh, F. Y., "Dual-Spacecraft Formation Flying in Deep Space: Optimal Collision-Free Reconfigurations," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 2, 2003, pp. 375–379.

[19] Kim, Y., Mesbahi, M., Singh, G., and Hadaegh, F. Y., "On the Convex Parametrization of Spacecraft Orientation in Presence of Constraints and its Applications," *IEEE transactions on Aerospace and Electronic Systems*, Vol. 46, No. May 2014, 2010, pp. 1097 – 1109.

[20] Lee, U. and Mesbahi, M., "Feedback control for spacecraft reorientation under attitude constraints via convex potentials," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 50, No. 4, 2014, pp. 2578–2592.

[21] Scharf, D. P., Murray, E. a., and Hadaegh, F. Y., "A Convex Guidance Algorithm for Formation Reconfiguration," *Jet Propulsion*, 2006.

[22] Morgan, D., Chung, S. J., and Hadaegh, F. Y., "Decentralized model predictive control of swarms of spacecraft using sequential convex programming," *Advances in the Astronautical Sciences*, Vol. 148, No. 6, 2013, pp. 3835–3854.

[23] Dueri, D., Açkmee, B., Scharf, D. P., and Harris, M. W., "Customized Real-Time Interior-Point Methods for Onboard Powered-Descent Guidance," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 2, aug 2016, pp. 197–212.

[24] Szmuk, M., Pascucci, C. A., Dueri, D., and Açikmee, B., "Convexification and real-time on-board optimization for agile quad-rotor maneuvering and obstacle avoidance," *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 4862–4868.

[25] Domahidi, A., Chu, E., and Boyd, S., "ECOS: An SOCP solver for embedded systems," *European Control Conference*, 2013, pp. 3071—-3076.

[26] Mattingley, J. and Boyd, S., "CVXGEN: A code generator for embedded convex optimization," *Optimization and Engineering*, Vol. 13, No. 1, 2012, pp. 1–27.

[27] Jerez, J. L., Goulart, P. J., Richter, S., Constantinides, G. a., Kerrigan, E. C., and Morari, M., "Embedded Online Optimization for Model Predictive Control at Megahertz Rates," *Automatic Control, IEEE Transactions on*, Vol. 59, No. 12, 2014, pp. 3238–3251.

[28] Northway, P., Aubuchon, C., Mellema, H., and Winglee, R., "Pulsed Plasma Thruster Gains in Specific Thrust for CubeSat Propulsion," *AIAA/SAE/ASEE Joint Propulsion Conference*, 2017, p. 5040.

[29] Hansen, A., *Visualizing Quaternions*, Elsevier Inc., 2006.

[30] Vallado, D., *Fundamentals of Astrodynamics & Applications*, Springer Science & Business Media, 4th ed., 2014.

[31] Kelso, T., Hoots, F., and Roehrich, R., "Spacetrack Report no. 3 - Models for Propagation of NORAD Element Sets," Tech. rep., NASA, 1988.

[32] Silani, E. and Lovera, M., "Magnetic spacecraft attitude control: A survey and some new results," *Control Engineering Practice*, Vol. 13, No. 3, 2005, pp. 357–371.

[33] de Ruiter, A. H. J., Damaren, C., and Forbes, J., *Spacecraft Dynamics and Control*, John Wiley & Sons Ltd., Chichester, United Kingdom, 2013.

[34] Sidi, M. J., *Spacecraft Dynamics and Control*, Cambridge University Press, 1997.

[35] Stickler, A. C. and Alfriend, K. T., "Elementary Magnetic Attitude Control System," *Journal of Spacecraft and Rockets*, Vol. 13, No. 5, may 1976, pp. 282–287.

[36] Crassidis, J. L. and Junkins, J. L., *Optimal Estimation of Dynamic Systems*, CRC Press, Boca Raton, FL, 2012.

[37] Wie, B., Weiss, H., and Arapostathis, A., "Quarternion feedback regulator for spacecraft eigenaxis rotations," *Journal of Guidance, Control, and Dynamics*, Vol. 12, No. 3, may 1989, pp. 375–380.

[38] Antsaklis, P. and Michel, A., *A Linear Systems Primer*, Birkhauser, Boston, 2007.

[39] Tregouet, J.-F., Arzelier, D., Peaucelle, D., Pittet, C., and Zaccarian, L., "Reaction Wheels Desaturation Using Magnetorquers and Static Input Allocation," *IEEE Transactions on Control Systems Technology*, Vol. 23, No. 2, mar 2015, pp. 525–539.

[40] Giulietti, F., Quarta, A. A., and Tortora, P., "Optimal Control Laws for Momentum-Wheel Desaturation Using Magnetorquers," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 6, 2006, pp. 1464–1468.

[41] Martel, F. and Psiaki, M., "Active Magnetic Control System for Gravity Gradient Stabilized Spacecraft," *2nd Annual AIAA/USU Conf. on Small Satellites*, Logan, Utah, 1988, p. unnumbered.

[42] Si Mohammed, A. M., Benyettou, M., Chouraqui, S., Boudjemai, A., and Hashida, Y., "Magnetorquer control for orbital manoeuvre of low earth orbit microsatellite," *WSEAS Transactions on Communications*, Vol. 5, No. 5, 2006, pp. 944–947.

[43] Hablani, H. B., "Attitude Commands Avoiding Bright Objects and Maintaining Communication with Ground Station," *Journal of Guidance, Control, and Dynamics*, Vol. 22, No. 6, nov 1999, pp. 759–767.

[44] Ahmed, A., Alexander, J., Boussalis, D., Breckenridge, W., Macala, G., Mesbahi, M., San Martin, M., Singh, G., and Wong, E., "Cassini Control Analysis Book," Tech. rep., Jet Propulsion Lab, 1997.

[45] Singh, G., Macala, G., Wong, E., and Rasmussen, R., "A Constraint Monitor Algorithm for the Cassini Spacecraft," Tech. rep., Jet Propulsion Lab, 1997.

[46] Spindler, K., "New Methods in On-board Attitude Control," Tech. rep., NASA, 1998.