Attitude Dynamics and Control of a Nano-Satellite Orbiting Mars

Joseph Miceli*
University of Colorado, Boulder
Spring 2023

This report summarizes the capstone project of the course ASEN5010 - Spacecraft Attitude Dynamics and Control. The project involves creating a model for an attitude determination and control system (ADCS) and simulating the system on a satellite orbiting Mars. The focus of this project is on the algorithms driving the system (rather than its hardware components) and demonstrating the principles of dynamics used to develop them. The project is structured in a way that incrementally builds each algorithm of the system by first analytically deriving the functions that will result in the desired behavior of the spacecraft and then testing them in simulation. This report begins with a summary of the project. It then describes each individual task, their corresponding solutions, and how they were achieved. The report concludes with a full simulation of the ADCS model and a discussion of the results.

I. Project Overview

THIS project considers a small satellite orbiting Mars at a low altitude gathering science data. The small satellite needs to transfer data to a larger mother satellite orbiting at a higher altitude. Further, the satellite must periodically point its solar panels at the sun to keep the batteries from depleting. Thus, there are three mission goals for the satellite:

1. Point the sensor platform straight down at Mars
2. Point the communication platform at the mother satellite
3. Point the solar arrays at the sun

The high level goal of this project is to design a thruster-based attitude controller to achieve these attitude scenarios. To satisfy the mission requirements, the spacecraft's body frame B must be driven towards various reference body frames R that correspond to the desired attitude. The reference attitudes are computed from a combination of the spacecrafts position and velocity about Mars, the mother spacecraft's motion, and the sun's heading. In all scenarios, the small spacecraft and mother spacecraft are in simple circular orbits whose motion is completely known. Once the reference is derived for each scenario, a control law is derived that drives the current attitude (represented using Modified Rodrigues Parameters (MRP)) and angular velocity toward their reference values. The scope of this project encompasses reference frame generation, attitude characterization, and feedback control.

A. Mission Description

The nano-satellite is orbiting in a circular low Mars orbit (LMO) and is intended to observe the non-sunlit surface of Mars. The mother spacecraft is on a circular geosynchronous Mars orbit (GMO). The nano-satellite is to either point a sensor at the surface (science mode), point the solar panels in the sun direction (power mode), or point the communication antenna at the mother spacecraft (communication mode). For simplicity, this project assumes that both spacecraft are in simple two-body circular orbits. Figure 1 shows these two orbits.

*Graduate Student

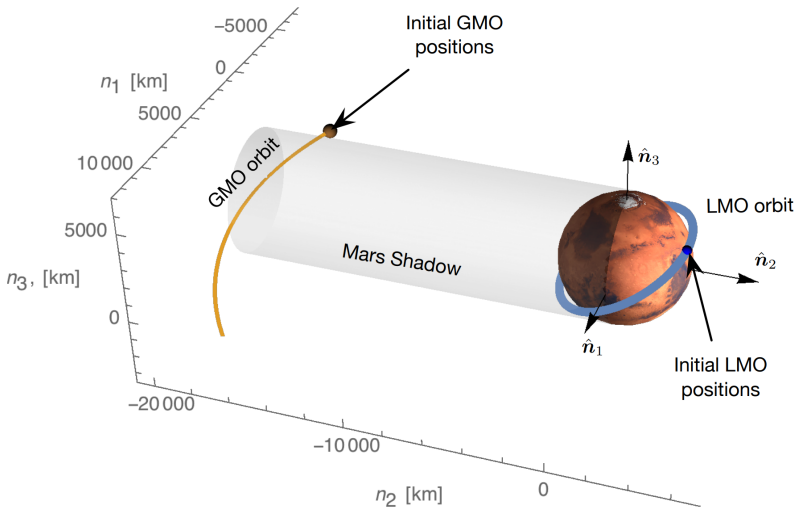

Figure 1: Depiction of LMO and GMO orbits

These two orbits can be described using the Hill (H) frame. Figure 2 depicts the Hill frame with respect to the inertial frame. Note that the orientation of the Hill frame with respect to the inertial frame is given by the $(3-1-3)$ Euler angle set (Ω, i, θ). To make this simulated mission more realistic, it is assumed the LMO spacecraft is equipped

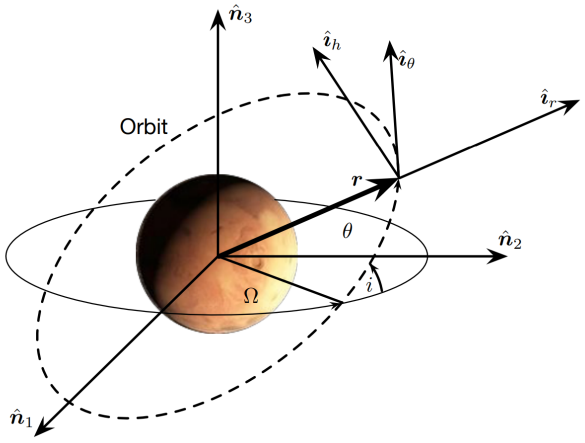

Figure 2: Inertial frame $N = [\hat{n}_1, \hat{n}_2, \hat{n}_3]$ and Hill frame $H = [\hat{h}_1, \hat{h}_2, \hat{h}_3]$ basis vectors

with a communication antenna along its $-\hat{b}_1$ axis. The sensor that will be used to observe the surface is assumed to be aligned with the $+\hat{b}_1$. Finally, the solar panels are assumed to be mounted such that their normal vector points in the $+\hat{b}_3$ direction. Figure 3 depicts the spacecraft component directions.

B. Mission Scenario Definitions

1. Spacecraft Attitude States

In all simulations, the initial attitude and angular velocity of the LMO spacecraft with respect to the inertial frame is defined as

$$\boldsymbol{\sigma}_{B/N}(t_0) = \begin{bmatrix} 0.3 \\ -0.4 \\ 0.5 \end{bmatrix} \quad {}^{B}\boldsymbol{\omega}_{B/N}(t_0) = \begin{bmatrix} 1.00 \\ 1.75 \\ -2.20 \end{bmatrix} \text{ deg/s} \tag{1}$$

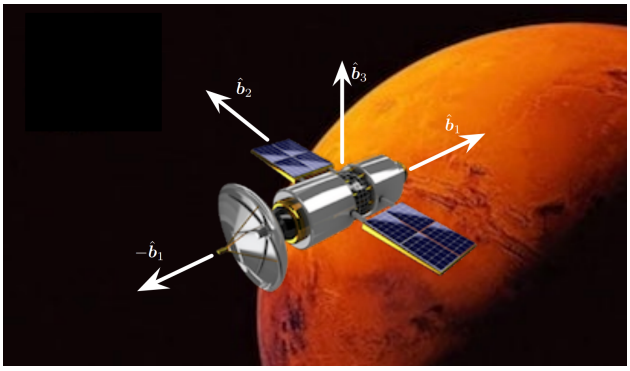

Figure 3: Spacecraft component directions

The inertia tensor of the LMO spacecraft is given as

$$^{B}[I] = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 7.5 \end{bmatrix} \text{kg m}^2 \tag{2}$$

2. Mission Pointing Scenarios

As shown in Fig. 1, the sun is assumed to be in the $\hat{n}_2$ direction and is infinitely far away. During the mission, the controller must detumble the satellite and either point the antenna ($-\hat{b}_1$) towards the GMO spacecraft, the sensor ($\hat{b}_1$) nadir (i.e. the $-r$ direction), or the solar panels ($\hat{b}_3$) at the sun. It is assumed that the LMO spacecraft must point at the sun whenever it is on the sunlit side of Mars. When the spacecraft is on the shaded side of Mars it must either be in communication or science mode. The communication mode requires that the LMO and GMO spacecraft position vectors have an angular difference less than 35 degrees. Table 2. summarizes the various pointing scenarios for the LMO spacecraft.

Table 1: Summary of spacecraft pointing scenarios

Orbital Situation	Primary Pointing Goals	Pointing Mode
SC on sunlit side of Mars	Point solar panels ($\hat{b}_3$) at sun	Power
SC not on sunlit side of Mars and GMO visible	Point antenna ($-\hat{b}_1$) at GMO	Communication
SC not on sunlit side of Mars and GMO not visible	Point sensor ($\hat{b}_1$) at surface of Mars	Science

3. Mission Orbit Overview

The radius of Mars is assumed to be $R_{\male} = 3396.19$[km] while the gravitational constant is $\mu_{\male} = 42828.3[\text{km}^3/\text{s}^2]$. Table 3. depicts the initial orbit conditions for both spacecraft. Note that both spacecraft have constant orbit rates $\dot{\theta}$.

Table 2: Initial orbit conditions for the LMO and GMO spacecraft

Parameter	Description	Value for LMO spacecraft	Value for GMO Spacecraft
h [km]	Altitude	400.0	17028.01
Ω [deg]	Right ascension	20.0	0.0
i [deg]	Inclination	30.0	0.0
θ [deg]	True latitude	60.0	250.0
$\dot{\theta}$ [rad/s]	Orbit rate	0.000884797	0.0000709003

4. Attitude Control Law Overview

In this project, a simple proportional-derivative (PD) control law is used:

$$^{B}\boldsymbol{u} = -K\boldsymbol{\sigma}_{B/R} - P^{B}\boldsymbol{\omega}_{B/R} \tag{3}$$

This equation is used to drive the body frame B and angular velocity to their corresponding references R. Note that K and P are both scalars

II. Project Tasks

This section contains the individual tasks from the project. Each task is summarized and the analytical portions of their corresponding solutions are provided. Validation and software-specific tasks and solutions are not included.

A. Task 1: Orbit Simulation

To derive the inertial frame expressions for the velocity and position of the LMO spacecraft, consider the general orbit frame (as illustrated in Fig. 2):

$$O : \{\hat{i}_r, \hat{i}_\theta, \hat{i}_h\} \tag{4}$$

To derive the inertial velocity vector, the transport theorem[1] must be used to relate the orbit frame relative time derivative of position to the inertial frame relative time derivative of position:

$$\dot{\boldsymbol{r}}_{LMO} = \frac{{}^N d\boldsymbol{r}_{LMO}}{dt} = \frac{{}^O d\boldsymbol{r}_{LMO}}{dt} + \boldsymbol{\omega}_{O/N} \times \boldsymbol{r}_{LMO} \tag{5}$$

Using the definition of the orbit frame, the position of the LMO spacecraft can be written as

$$\boldsymbol{r}_{LMO} = r\hat{i}_r \tag{6}$$

Note that although an orbit frame basis vector is being used, this vector can be expressed in any frame. The time rate of change of $\boldsymbol{r}_{LMO}$ as seen by the orbit frame is then 0.

$$\frac{{}^O d\boldsymbol{r}_{LMO}}{dt} = \boldsymbol{0} \tag{7}$$

The angular velocity of the LMO spacecraft's orbit frame with respect to the inertial frame is simply the orbit rate of the spacecraft about the $\hat{i}_h$ axis.

$$\boldsymbol{\omega}_{O/N} = \dot{\theta}_{LMO}\hat{i}_h = 0.000884797\hat{i}_h \text{rad/s} \tag{8}$$

Equation 5 can then be simplified to:

$$\dot{\boldsymbol{r}}_{LMO} = r_{LMO}\dot{\theta}_{LMO}\hat{i}_\theta \tag{9}$$

Note that this expression is also reference frame independent; the position and velocity of the spacecraft with respect to the inertial frame could be expressed in any frame. To express $\boldsymbol{r}_{LMO}$ and $\dot{\boldsymbol{r}}_{LMO}$ in the inertial frame, it makes sense to fist express them in the orbit frame. Given a radius r_{LMO} and orbit rate $\dot{\theta}_{LMO}$ of the LMO spacecraft, the orbit frame expression of the LMO spacecraft's position and velocity are:

$${}^O\boldsymbol{r}_{LMO} = \begin{bmatrix} r_{LMO} \\ 0 \\ 0 \end{bmatrix} \tag{10}$$

$${}^O\dot{\boldsymbol{r}}_{LMO} = \begin{bmatrix} 0 \\ r_{LMO}\dot{\theta}_{LMO} \\ 0 \end{bmatrix} \tag{11}$$

The provided (3-1-3) Euler angles can be used to calculate the direction cosine matrix (DCM) describing the rotation from the inertial frame to the orbit frame, $[ON(\Omega, i, \theta)]$. The DCM can then be used with $\boldsymbol{r}_{LMO}$ and $\dot{\boldsymbol{r}}_{LMO}$ to express the position and velocity in the inertial frame:

$${}^N\boldsymbol{r}_{LMO} = [NO(\Omega, i, \theta)]{}^O\boldsymbol{r}_{LMO} \tag{12}$$

$${}^N\dot{\boldsymbol{r}}_{LMO} = [NO(\Omega, i, \theta)]{}^O\boldsymbol{r}_{LMO} \tag{13}$$

where $[NO(\Omega, i, \theta)] = [ON(\Omega, i, \theta)]^T$. This same approach can be used to express the inertial position and velocity of the GMO spacecraft in the inertial frame ${}^N\boldsymbol{r}_{GMO}$ and ${}^N\dot{\boldsymbol{r}}_{GMO}$.

B. Task 2: Orbit Frame Orientation

In this mission, the Hill frame $H = \{\hat{i}_r, \hat{i}_\theta, \hat{i}_h\}$ is the orbit frame of the LMO satellite. These base vectors are generally defined as:

$$\hat{i}_r = \frac{\boldsymbol{r}_{LMO}}{|\boldsymbol{r}_{LMO}|}, \quad \hat{i}_\theta = \hat{i}_h \times \hat{i}_r, \quad \hat{i}_h = \frac{\boldsymbol{r}_{LMO} \times \dot{\boldsymbol{r}}_{LMO}}{|\boldsymbol{r}_{LMO} \times \dot{\boldsymbol{r}}_{LMO}|} \tag{14}$$

The DCM describing the orientation from the Hill frame relative to the inertial frame $[NH]$ can be composed from expressing the Hill frame basis vectors in the inertial frame:[1]

$$[NH] = [{}^N\hat{i}_r, {}^N\hat{i}_\theta, {}^N\hat{i}_h] \tag{15}$$

$[HN]$ is then simply $[NH]^T$. To define this calculation as a function of time, each Hill frame basis vector simply needs to be evaluated at the required time, t:

$$[HN(t)] = [NH(t)]^T = \{{}^N\hat{i}_r(t), {}^N\hat{i}_\theta(t), {}^N\hat{i}_h(t)\}^T \tag{16}$$

where

$${}^N\hat{i}_r(t) = \frac{{}^N\boldsymbol{r}_{LMO}(t)}{|{}^N\boldsymbol{r}_{LMO}(t)|}, \quad {}^N\hat{i}_\theta(t) = {}^N\hat{i}_h(t) \times {}^N\hat{i}_r(t), \quad {}^N\hat{i}_h(t) = \frac{{}^N\boldsymbol{r}_{LMO}(t) \times {}^N\dot{\boldsymbol{r}}_{LMO}(t)}{|{}^N\boldsymbol{r}_{LMO}(t) \times {}^N\dot{\boldsymbol{r}}_{LMO}(t)|} \tag{17}$$

Note the expressions for ${}^N\boldsymbol{r}_{LM}(t)$ and ${}^N\dot{\boldsymbol{r}}_{LM}(t)$ where derived in Section A.. Also note that in the case of the LMO satellite, the angles Ω and i are constant but θ varies with time according to a constant orbit rate $\dot{\theta}$. Therefore when calculating the DCM $[HN]$, the angle θ must be propagated to its value at the current time according to

$$\theta(t) = \theta_0 + t\dot{\theta} \tag{18}$$

This same approach is used to propagate the position of the GMO spacecraft as well.

C. Task 3: Sun-Pointing Reference Frame Orientation

To point the spacecraft solar panels axis ($\hat{b}_3$) at the sun, the reference frame R_s must be chosen such that $\hat{r}_3$ axis points in the sun direction ($\hat{n}_2$ in this scenario). Further, is assumed the first axis $\hat{r}_1$ points in $-\hat{n}_1$ direction. Like in Section B., DCM $[NR_s]$ can be composed from the basis vectors of the R_s frame expressed in the inertial, N frame.

$$[NRs] = [{}^N\hat{r}_1, {}^N\hat{r}_2, {}^N\hat{r}_3] \tag{19}$$

Using the provided definition of the R_s frame, the basis vectors of the R_s frame can be written as:

$$\hat{r}_3 = \hat{n}_2, \quad \hat{r}_1 = -\hat{n}_1, \quad \hat{r}_2 = \hat{r}_3 \times \hat{r}_1 \tag{20}$$

where $\hat{n}_i$ are the basis vectors of the N frame. By inspection, these vectors can be expressed in the inertial frame:

$${}^N\hat{r}_3 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad {}^N\hat{r}_1 = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}, \quad {}^N\hat{r}_2 = {}^N\hat{r}_3 \times {}^N\hat{r}_1 \tag{21}$$

DCM $[R_sN]$ is then simply calculated as $[R_sN] = [NR_s]^T$. Because frame R_s maintains a constant orientation with respect to the N frame, the angular velocity of the R_s frame with respect to the N frame expressed in inertial frame components is 0.

$${}^N\boldsymbol{\omega}_{R_s/N} = \mathbf{0} \tag{22}$$

D. Task 4: Nadir-Pointing Reference Frame Orientation

To point the spacecraft sensor platform axis ($\hat{b}_1$) towards the center of Mars (i.e the nadir direction), the reference frame Rn must be chosen such that $\hat{r}_1$ axis points towards the planet. Further, it is assumed the second axis $\hat{r}_1$ points in the velocity direction $\hat{i}_\theta$. Using this definition, the basis vectors of the R_n frame can be defined in relation to the Hill frame basis vectors as

$$\hat{r}_1 = -\hat{i}_r, \quad \hat{r}_2 = \hat{i}_\theta, \quad \hat{r}_3 = \hat{r}_1 \times \hat{r}_2 \tag{23}$$

Similarly to Section B., the DCM $[R_nN]$ can be expressed as $[R_nN] = [NR_n]^T$ where $[NR_n]$ is composed from the basis vectors of the R_n frame expressed in the inertial frame.

$$[NRn] = [{}^N\hat{r}_1, {}^N\hat{r}_2, {}^N\hat{r}_3] \tag{24}$$

Equation 23 can be expressed in the inertial frame using the Hill frame expression of the Hill frame basis vectors and converting them to the inertial frame using the DCM $[NH]$.

$$ {}^{N}\hat{r}_1 = [NH](-{}^{H}\hat{i}_r) = [NH]\begin{bmatrix}-1\\0\\0\end{bmatrix}, \quad {}^{N}\hat{r}_2 = [NH]{}^{H}\hat{i}_\theta = [NH]\begin{bmatrix}0\\1\\0\end{bmatrix}, \quad {}^{N}\hat{r}_3 = {}^{N}\hat{r}_1 \times {}^{N}\hat{r}_2 \tag{25} $$

${}^{H}\hat{i}_r$ and ${}^{H}\hat{i}_\theta$ are time-invariant in the definition of the R_n frame so to make Eq. 25 a function of time, the DCM $[NH]$ just needs to be computed at the required time which can be accomplished with Eq. 16.

The angular velocity of the R_n frame with respect to the N frame can also be defined using the Hill frame basis vector $\hat{i}_h$

$$ \boldsymbol{\omega}_{R_n/N} = \dot{\theta}_{LMO}\hat{i}_h \tag{26} $$

where $\dot{\theta}_{LMO}$ is the orbit rate of the LMO spacecraft. To express $\boldsymbol{\omega}_{R_n/N}$ in inertial frame components, the DCM $[NH]$ can be used again.

$$ {}^{N}\boldsymbol{\omega}_{R_n/N} = [NH]{}^{H}\hat{i}_h = [NH]\begin{bmatrix}0\\0\\\dot{\theta}_{LMO}\end{bmatrix} \tag{27} $$

E. Task 5: GMO-Pointing Reference Frame Orientation

To point the nano-satellite communication platform axis ($-\hat{b}_1$) towards the GMO mother spacecraft in communication mode, the communication reference frame R_c must be chosen such that $-\hat{r}_1$ axis points towards the GMO satellite location. Define $\Delta\boldsymbol{r} = \boldsymbol{r}_{GMO} - \boldsymbol{r}_{LMO}$. Further, to fully define a three-dimensional reference frame, it is assumed the second axis is defined as

$$ \hat{r}_2 = \frac{\Delta\boldsymbol{r} \times \hat{n}_3}{|\Delta\boldsymbol{r} \times \hat{n}_3|} \tag{28} $$

while the third is then $\hat{r}_3 = \hat{r}_1 \times \hat{r}_2$.

Deriving the analytical expression for $[R_cN]$ begins in the same way as in the previous sections. $[R_cN] = [NR_c]^T$ where $[NR_c]$ is composed from the basis vectors of the R_c frame expressed in the N frame.

$$ [NRc] = \{{}^{N}\hat{r}_1, {}^{N}\hat{r}_2, {}^{N}\hat{r}_3\} \tag{29} $$

In this case, $-\hat{r}_1$ points toward the GMO satellite. Using the definition of $\Delta\boldsymbol{r}$, $\hat{r}_1$ can be written as:

$$ \hat{r}_1 = \frac{-\Delta\boldsymbol{r}}{|\Delta\boldsymbol{r}|} \tag{30} $$

Therefore, expressing each basis vector $\hat{r}_i$ in inertial frame components:

$$ {}^{N}\hat{r}_1 = \frac{-{}^{N}\Delta\boldsymbol{r}}{|{}^{N}\Delta\boldsymbol{r}|}, \quad {}^{N}\hat{r}_2 = \frac{{}^{N}\Delta\boldsymbol{r} \times {}^{N}\hat{n}_3}{|{}^{N}\Delta\boldsymbol{r} \times {}^{N}\hat{n}_3|}, \quad {}^{N}\hat{r}_3 = {}^{N}\hat{r}_1 \times {}^{N}\hat{r}_2 \tag{31} $$

Where ${}^{N}\Delta\boldsymbol{r}$ is calculated from the position vectors ${}^{N}\boldsymbol{r}_{GMO}$ and ${}^{N}\boldsymbol{r}_{LMO}$ at the desired time using Eq. 12 and

$$ {}^{N}\hat{n}_3 = \begin{bmatrix}0\\0\\1\end{bmatrix} \tag{32} $$

To calculate the angular velocity of the R_c frame with respect to the N frame, the kinematic relationship between the DCM $[R_cN]$ and the angular velocity $\boldsymbol{\omega}_{R_c/N}$ can be numerically approximated. An analytical solution for $\boldsymbol{\omega}_{R_c/N}$ is challenging to derive but a numerical approximation will give sufficient results. The kinematic differential equation that relates the rate of change of a DCM to angular velocity is given as:[1]

$$ [\dot{NR_c}] = -[{}^{N}\tilde{\boldsymbol{\omega}}_{N/R_c}][NR_c] \tag{33} $$

A finite difference method can be used with a small time step Δt to approximate $[NR_c]$.[2]

$$ [\dot{NR_c}] \approx \frac{[NR_c]_t - [NR_c]_{t-\Delta t}}{\Delta t} \tag{34} $$

Substituting this into Eq. 33 and solving for $[{}^N\tilde{\boldsymbol{\omega}}_{R_c/N}]$ gives:

$$[{}^N\tilde{\boldsymbol{\omega}}_{N/R_c}] = -\left(\frac{[NR_c]_t - [NR_c]_{t-\Delta t}}{\Delta t}\right)[NR_c]_t^T = [{}^N\tilde{\boldsymbol{\omega}}_{R_c/N}]^T \tag{35}$$

Note that ${}^N\boldsymbol{\omega}_{R_c/N}$ is the desired vector but $[{}^N\tilde{\boldsymbol{\omega}}_{R_c/N}] = [{}^N\tilde{\boldsymbol{\omega}}_{N/R_c}]^T$ and the definition of the skew-symmetric matrix operator is given as:

$$[\tilde{\boldsymbol{\omega}}] = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \tag{36}$$

Equation 35 can then be used to calculate the angular velocity of the R_c frame with respect to the N frame expressed in inertial components as:

$${}^N\boldsymbol{\omega}_{R_c/N} = \begin{bmatrix} -[{}^N\tilde{\boldsymbol{\omega}}_{R_c/N}]_{23} \\ [{}^N\tilde{\boldsymbol{\omega}}_{R_c/N}]_{13} \\ -[{}^N\tilde{\boldsymbol{\omega}}_{R_c/N}]_{12} \end{bmatrix} \tag{37}$$

Also note that using this numerical approach requires that when evaluating $[NR_c]$, the argument t must be greater than the time step Δt.

F. Task 6: Attitude Error Evaluation

Any attitude feedback control algorithm requires the ability to compute the attitude and angular velocity tracking errors of the current body frame B relative to the reference frame R. An expression for attitude error will be derived first. Note that the MRP set $\boldsymbol{\sigma}_{B/R}$ represents the orientation of the B frame with respect to the R frame. This can also be expressed as the DCM $[BR]$. To calculate $[BR]$, DCM "addition" can be used:

$$[BR] = [BN][NR] \tag{38}$$

Where $[NR] = [RN]^T$. The DCMs $[BN]$ and $[RN]$ can be calculated from the provided MRPs $\boldsymbol{\sigma}_{B/N}$ and $\boldsymbol{\sigma}_{R/N}$ respectively via:[1]

$$[C] = [I_{3\times 3}] + \frac{8[\tilde{\boldsymbol{\sigma}}]^2 - 4(1-\boldsymbol{\sigma}^2)[\tilde{\boldsymbol{\sigma}}]}{(1+\boldsymbol{\sigma}^2)^2} \tag{39}$$

After calculating the attitude tracking error in DCM form, $[BR]$, it can be converted to its corresponding MRP set using the direct mapping:[1]

$$[\tilde{\boldsymbol{\sigma}}_{B/R}] = \frac{[BR]^T - [BR]}{\zeta(\zeta+2)} \tag{40}$$

where $\zeta = \sqrt{\text{trace}([BR]) + 1}$.

To calculate the angular velocity tracking error, $\boldsymbol{\omega}_{B/R}$, vector subtraction can be used:

$$\boldsymbol{\omega}_{B/R} = \boldsymbol{\omega}_{B/N} - \boldsymbol{\omega}_{R/N} \tag{41}$$

where $\boldsymbol{\omega}_{B/N}$ is provided in body frame components and $\boldsymbol{\omega}_{R/N}$ is provided in inertial frame components. $\boldsymbol{\omega}_{B/R}$ can easily be expressed in the B frame using the DCM $[BN]$ which was calculated using Eq. 39.

$${}^B\boldsymbol{\omega}_{B/R} = {}^B\boldsymbol{\omega}_{B/N} - [BN]{}^N\boldsymbol{\omega}_{R/N} \tag{42}$$

G. Task 7: Numerical Attitude Simulator

This section discusses the numerical integration of the rotational motion of the LMO spacecraft. The numerical integration of position of the LMO satellite was discussed earlier in section A.. Let the propagated attitude state X be:

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{\sigma}_{B/N} \\ {}^B\boldsymbol{\omega}_{B/N} \end{bmatrix} \tag{43}$$

Next, it is assumed that the spacecraft is rigid and its dynamics obey the following equations of motion:

$$[I]\dot{\boldsymbol{\omega}}_{B/N} = -[\tilde{\boldsymbol{\omega}}_{B/N}][I]\boldsymbol{\omega}_{B/N} + \boldsymbol{u} \tag{44}$$

where $\boldsymbol{u}$ is the external control torque vector.

To integrate Eq. 44, a fourth order Runge-Kutta (RK4) algorithm was implemented[3] in Python. The algorithm follows the general functional flow shown in Fig. 4. A time step of $\Delta t = 1$ second was chosen to ensure that small enough such that numerical integrator errors were not visible in your simulation result plots. Further, for all of the following tasks, the control vector $\boldsymbol{u}$ was held piece-wise constant between integration steps. When necessary, $\boldsymbol{u}$ was updated in between integration steps.

$\boldsymbol{X}_0 = \begin{bmatrix} \boldsymbol{\sigma}_{B/N}(t_0) \\ {}^{\mathcal{B}}\boldsymbol{\omega}_{B/N}(t_0) \end{bmatrix}$;
$t_{\text{max}} = \cdots$;
$\Delta t = \cdots$;
t_n = 0.0 ;
while $t_n < t_{max}$ **do**
  if *new control required* **then**
    Evaluate current reference frame states $[RN(t)]$, ${}^{\mathcal{N}}\boldsymbol{\omega}_{R/N}(t)$, etc.;
    Determine control tracking errors $\boldsymbol{\sigma}_{B/R}$ and ${}^{\mathcal{B}}\boldsymbol{\omega}_{B/R}$;
    Determine control solution $\boldsymbol{u}$;
  end
  $k_1 = \Delta t \boldsymbol{f}(\boldsymbol{X}_n, t_n, \boldsymbol{u})$;
  $k_2 = \Delta t \boldsymbol{f}(\boldsymbol{X}_n + \frac{k_1}{2}, t_n + \frac{\Delta t}{2}, \boldsymbol{u})$;
  $k_3 = \Delta t \boldsymbol{f}(\boldsymbol{X}_n + \frac{k_2}{2}, t_n + \frac{\Delta t}{2}, \boldsymbol{u})$;
  $k_4 = \Delta t \boldsymbol{f}(\boldsymbol{X}_n + k_3, t_n + \Delta t, \boldsymbol{u})$;
  $\boldsymbol{X}_{n+1} = \boldsymbol{X}_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$;
  if $|\boldsymbol{\sigma}_{B/N}| > 1$ **then**
    map $\boldsymbol{\sigma}_{B/N}$ to shadow set;
  end
  $t_{n+1} = t_n + \Delta t$;
  save spacecraft states $\boldsymbol{X}$ and $\boldsymbol{u}$;
end

Figure 4: Runge-Kutta algorithm sample provided in project

The calculation of the reference attitude $[RN(t)]$ and angular velocity ${}^{N}\boldsymbol{\omega}_{R/N}(t)$ were discussed in the sections C., D., and E.. The function $\boldsymbol{f}(\boldsymbol{X}, t, \boldsymbol{u})$ is the differential equation defined by Eq. 44. The calculation of tracking errors $\boldsymbol{\sigma}_{B/R}$ and ${}^{B}\boldsymbol{\omega}_{B/R}$ was discussed in section F.. Finally, sections H., I., and J. discuss the calculation of the control vector $\boldsymbol{u}$ from tracking errors $\boldsymbol{\sigma}_{B/R}$ and ${}^{B}\boldsymbol{\omega}_{B/R}$. To support various types of control solutions, the RK4 scheme was implemented such that the control function can be passed into the integrator as an argument rather than the vector $\boldsymbol{u}$ itself. Finally, a complete simulation loop can be created using the following functional flow:

1. Calculate the reference attitude $[RN]$ and angular velocity
2. Calculate the current attitude $[BN]$ and angular velocity $\boldsymbol{\omega}_{B/N}$
3. Calculate reference tracking errors $\boldsymbol{\sigma}_{B/R}$ and $\boldsymbol{\omega}_{B/R}$
4. Calculate the desired control torque $\boldsymbol{u}$
5. "Apply" the control torque by integrating the rotational equations of motion one time step

These steps are repeated each time step.

Figure 5 illustrates the evolution of states with zero control applied (i.e. ${}^{B}\boldsymbol{u} = \boldsymbol{0}$). Similarly, Fig. 6 illustrates the evolution of states with constant control (i.e. ${}^{B}\boldsymbol{u} = [0.01, -0.01, 0.02]^T$). In both cases, the initial conditions specified in Eq. 1 were used.

H. Task 8: Sun Pointing Control

In the remaining sections, the individual control pointing modes are first developed and tested independently. Then, they are combined into a full mission scenario simulation in section K.. In this section, sun pointing control is derived and simulated using the initial spacecraft attitude and orbit conditions in Eq. 1 and Table 3. and the assumption the spacecraft is to engage directly into a sun-pointing mode starting at t_0.

The attitude control law considered in this project is the simple PD control shown in Eq. 3. It is desired to select the scalar K and P feedback gains such that the slowest decay response time (i.e. time for tracking errors to be $1/e$ the

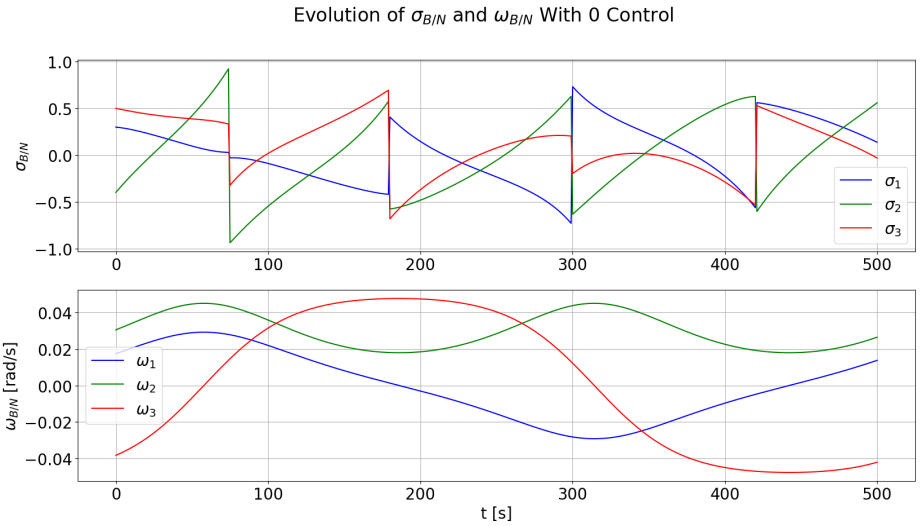

Figure 5: Integration of rotational motion with zero applied control torque

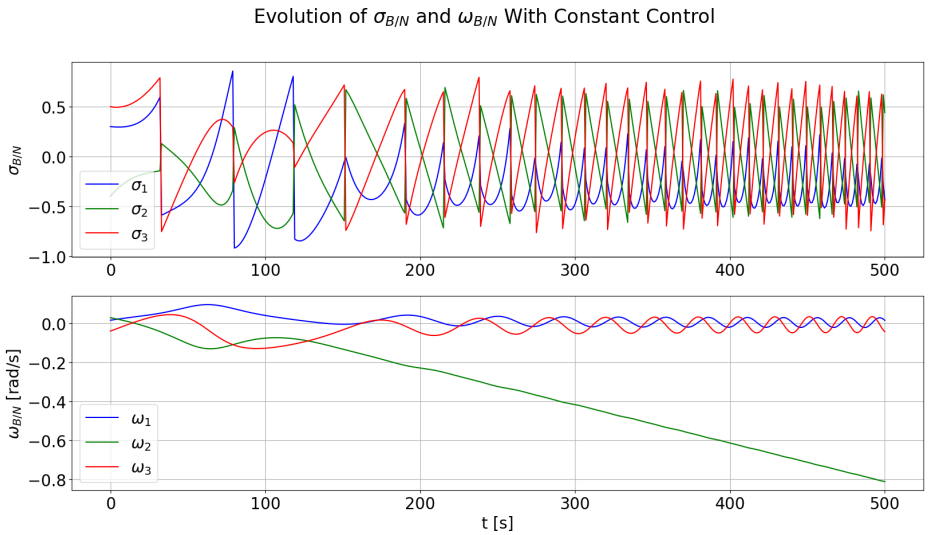

Figure 6: Integration of rotational motion with constant applied control torque

original size) is 120 seconds. This means all decay time constants should be 120 seconds or less. Further, the closed loop response for all $\boldsymbol{\sigma}_{B/N}$ components should be either critically damped or under-damped. Thus at least one mode must be critically damped with $\xi = 1$, while the other modes will have $\xi \leq 1$. To find the proportional feedback gain K and angular velocity feedback gain P, first consider the linearized closed loop attitude dynamics of the spacecraft:[1]

$$\dot{\boldsymbol{X}} = [A]\boldsymbol{X} = \begin{bmatrix} [0]_{3\times3} & \frac{1}{4}[I]_{3\times3} \\ -K[I]^{-1} & -[I]^{-1}[P] \end{bmatrix} \boldsymbol{X} \tag{45}$$

where $\boldsymbol{X}$ was defined in Eq. 43. Analyzing the eigenvalues of $[A]$ allows for the gains to be selected according to the desired damping behavior. To summarize these relationships, equations 8.118 and 8.119 in Schaub & Junkins[1] can be used. For reference, these equations are respectively:

$$\xi_i = \frac{P_i}{\sqrt{KI_i}} \tag{46}$$

$$T_i = \frac{2I_i}{P_i} \tag{47}$$

Knowing that $T_i \leq T_{max} = 120$ seconds, P_i can be calculated for each angular velocity component. In this scenario, P is treated as a scalar (or a diagonal matrix with repeated values). Therefore, P is calculated as

$$P = \frac{2\max(I_i)}{T_{max}} \tag{48}$$

P can now be used to calculate the feedback gain K. Again K is treated as a scalar where all $P_i = P$ and all $\xi_i = \xi = 1$. K is then calculated as

$$K = \frac{P^2}{\xi^2 \min(I_i)} \tag{49}$$

This control function can then be implemented where the sun pointing reference frame $[R_s N]$ and angular velocity $\boldsymbol{\omega}_{Rs/N}$ are used to calculate the reference tracking errors $\boldsymbol{\sigma_{B/R}}$ and $\boldsymbol{\omega}_{B/R}$. This control function can then be simulated using the initial states defined in Eq. 1 and Table 3. and the RK4 method previously developed. The evolution of tracking error magnitudes using this controller are shown in Fig. 7.

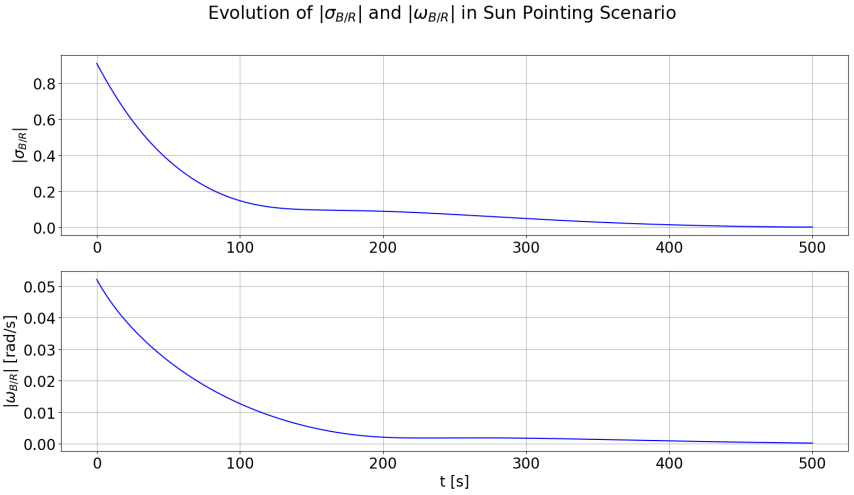

Figure 7: Magnitude of attitude and angular velocity tracking errors in sun pointing scenario

I. Task 9: Nadir Pointing Control

A nadir pointing control function can be implemented in a similar manner to the sun pointing function. The same PD control law and gains K and P that were used for sun-pointing are used here. In this case though, the nadir pointing reference frame $[R_n N]$ and reference angular velocity $\boldsymbol{\omega}_{Rn/N}$ are used to determine the reference tracking errors $\boldsymbol{\sigma_{B/R}}$ and $\boldsymbol{\omega}_{B/R}$. Again, this controller can then be simulated using the RK4 integrator and the initial states defined in Eq. 1 and Table 3.. For this simulation, it is assumed that at t_0 nadir pointing is required, even though the satellite is in the sunlight at this time. The evolution of tracking error magnitudes using this controller are shown in Fig. 8.

J. Task 10: GMO Pointing Control

Next, the GMO pointing attitude mode is created and tested. The same PD control law and gains K and P that were used previously are used here. In this case though, the GMO pointing reference frame $[R_c N]$ and reference angular velocity $\boldsymbol{\omega}_{Rc/N}$ are used to determine the reference tracking errors $\boldsymbol{\sigma_{B/R}}$ and $\boldsymbol{\omega}_{B/R}$. Again, this controller can be simulated using the RK4 integrator and the initial states defined in Eq. 1 and Table 3.. The assumption this time is that the GMO pointing is required at t_0. The evolution of tracking error magnitudes using this controller are shown in Fig. 9.

K. Task 11: Mission Scenario Simulation

For the final task, the full mission scenario will be simulated. The initial spacecraft attitude and orbit conditions defined in Eq. 1 and Table 3. will be used and propagated 6500 seconds to demonstrate the attitude pointing performance of the nano-satellite as it enters different control modes as described in section 2. To implement the necessary logic to switch between pointing modes, the three previously defined pointing control functions can be combined into

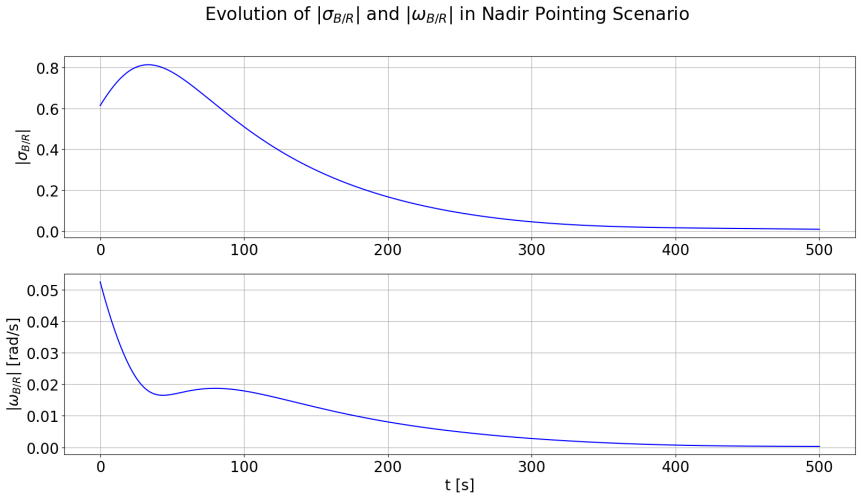

Figure 8: Magnitude of attitude and angular velocity tracking errors in nadir pointing scenario

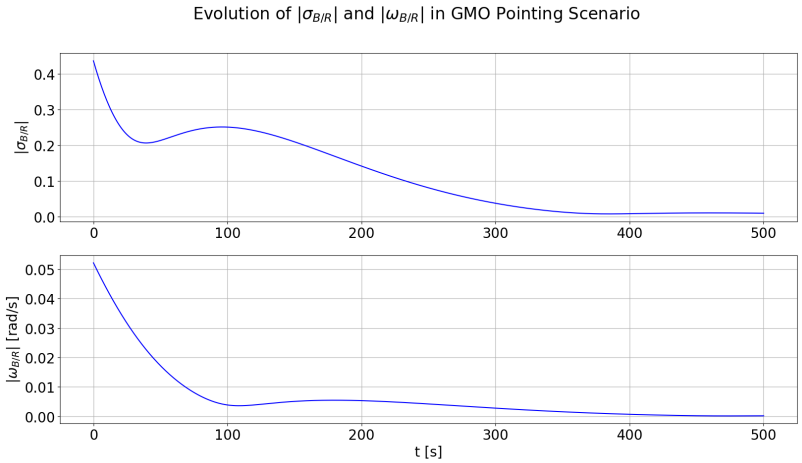

Figure 9: Magnitude of attitude and angular velocity tracking errors in GMO pointing scenario

a single "mission pointing" control function. Internally, the mission pointing control function determines which pointing control function (i.e. sun pointing, nadir pointing, or GMO pointing) to utilize based on logic derived from the pointing scenario summary defined in section 2. This logic can be summarized as follows:

1. Given ${}^N\boldsymbol{r}_{LMO} = [r_1, r_2, r_3]^T$, If $r_2 > 0$ then the sun is visible so use sun pointing
2. Else if $\text{acos}\left(\frac{{}^N\boldsymbol{r}_{LMO} \cdot {}^N\boldsymbol{r}_{GMO}}{|{}^N\boldsymbol{r}_{LMO}||{}^N\boldsymbol{r}_{GMO}|}\right) < 35.0$ deg then the GMO spacecraft is visible so use GMO pointing
3. Else neither the sun nor the GMO spacecraft are visible so use nadir pointing

This mission pointing control function can then be simulated in similar manner to the individual pointing modes. Figure 10 depicts the evolution of states during the full 6500 second mission simulation. The abrupt jumps in MRP components between 0 and approximately 1800 seconds correspond to switches to the shadow set. Near the 1800 second mark, a mode transition occurs causing a change to the reference attitude and angular velocity. This happens again around the 3100 second mark, the 4100 second mark and the 5500 second mark. These transitions are more clearly shown in Fig. 11. Further analysis shows that the attitudes and angular velocities do indeed converge to their reference values in each mode during the simulation. Figure 12 shows the tracking error magnitudes for both attitude and angular velocity. Here the mode transitions (and changes to the reference values) again cause noticeable changes to the tracking errors. Finally, the computed control torque components for the full mission simulation are shown in Fig. 13.

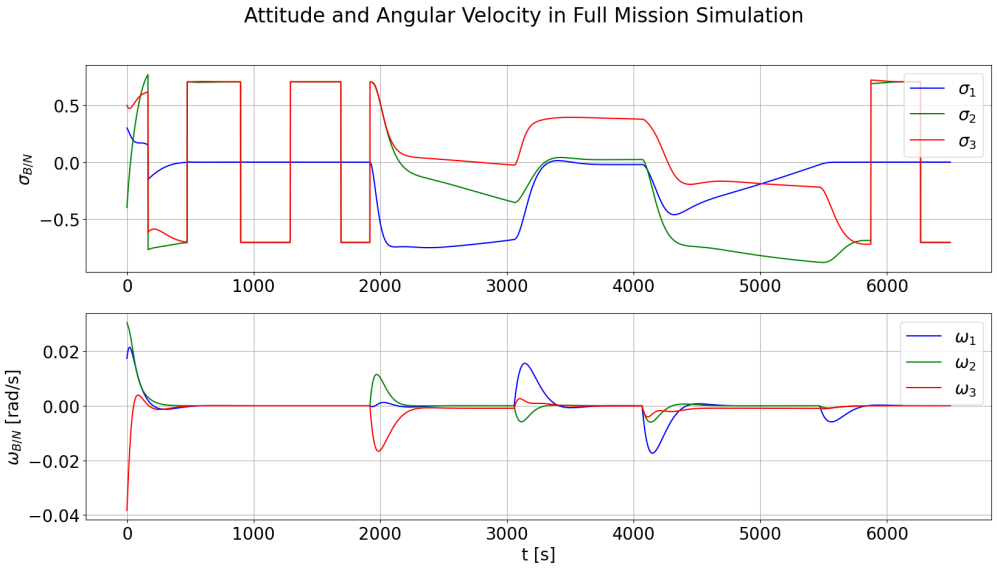

Figure 10: Attitude and angular velocity components in full mission simulation

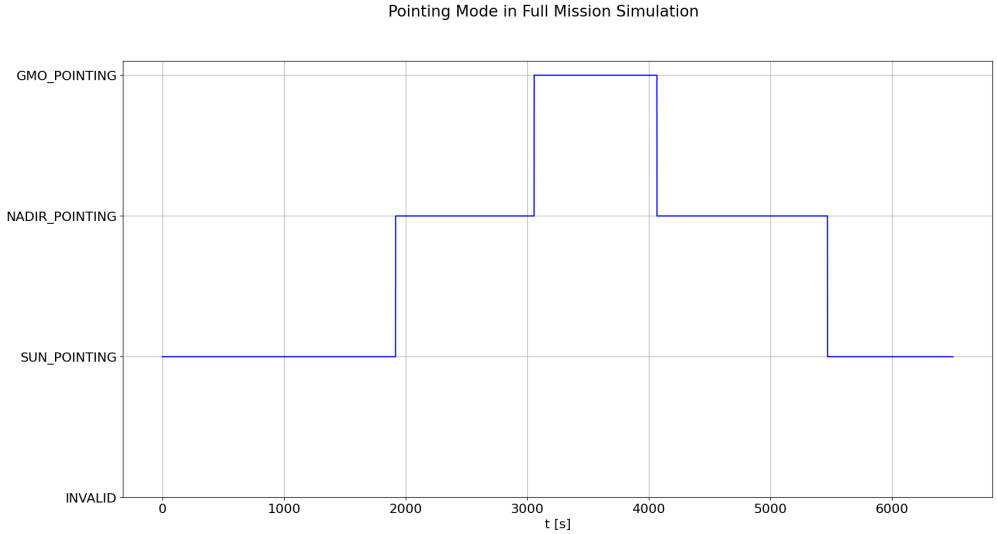

Figure 11: Pointing modes in full mission simulation

III. Conclusion

The focus of this project was on the attitude dynamics underlying reference frame generation, characterizing reference frame tracking errors, and creating a feedback control loop to drive states to their reference values. To that end, this project did not involve any component models in any simulations. Instead, it was assumed that the spacecraft had perfect knowledge of its current state and did not have any constraints on the torque it was able to apply. It was assumed that torque could be applied perfectly in any magnitude and direction. In practice, components are imperfect and limited in their capabilities so additional functionality would need to be added to account for uncertainty and constraints.

Additionally, the control law used in this project assumed that no external disturbance torques were acting on the spacecraft. In reality, an additional term $\boldsymbol{L}$, would be present in the equations of motion defined in Eq. 44 that would define "modeled" disturbance torques acting on the nano-satellite. Further, if this term was not completely accurate and there were some unknown disturbance torques acting on the nano-satellite or uncertainties in the satellite's mass properties, an additional integral term would need to be added to the PD controller to drive reference tracking errors to zero.

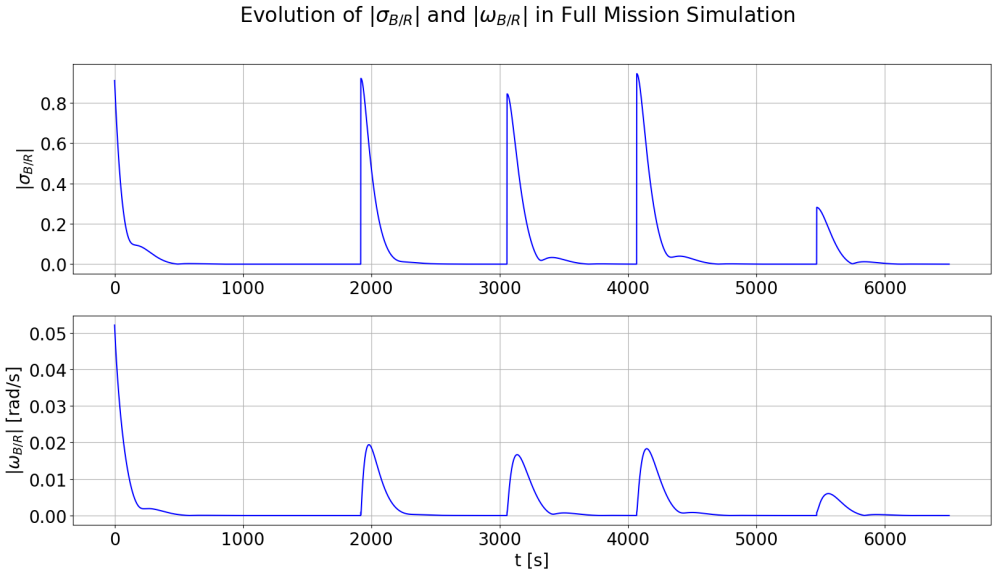

Figure 12: Magnitude of attitude and angular velocity tracking errors in full mission simulation

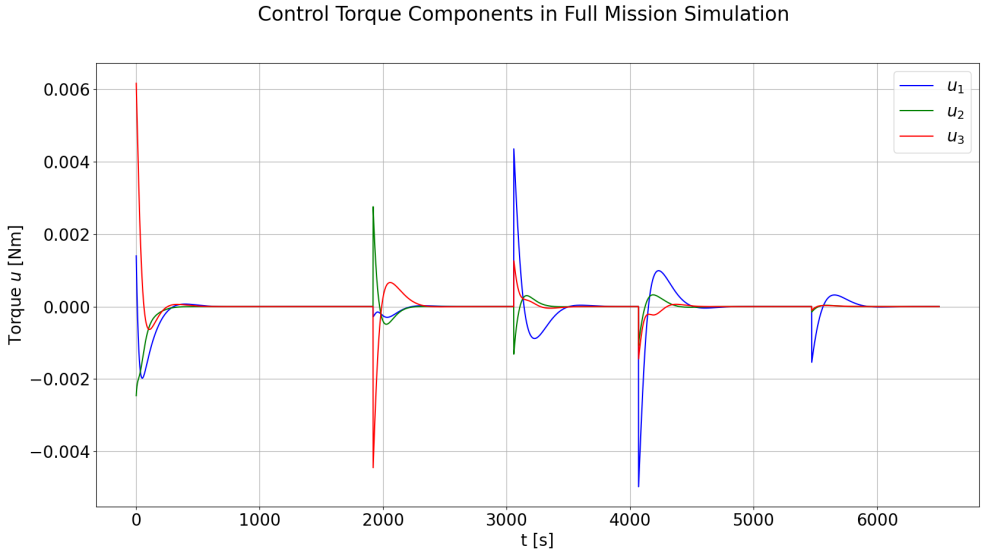

Figure 13: Computed control torques in full mission simulation

Despite these assumptions, this project successfully developed and demonstrated attitude guidance and control functions for a nano-satellite orbiting Mars in a low altitude orbit. Three different pointing ”modes” were developed. One for pointing at the sun to generate power, one for pointing at the surface of Mars to gather science data, and one for pointing at a mother spacecraft orbiting in a geosynchronous Mars orbit for communication. Each pointing case defined a desired reference frame that was used to derive reference attitudes and angular velocities that the nano-satellite needed to achieve in order to satisfy the pointing goals of that mode. In each case, a PD controller was used to drive the nano-satellite's attitude and angular velocity to their corresponding reference values. In the final mission simulation, additional logic was added to enable the satellite to automatically switch between modes to ensure that high level mission objectives were met.

Acknowledgments

The author would like to thank Dr. Cody Allard and Jesse Greaves for a challenging and academically rewarding semester.

References

[1]Schaub, H. and Junkins, J. L., *Analytical Mechanics of Space Systems*, AIAA Education Series, Reston, VA, 2nd ed., October 2009.

[2]Kong, Q., Siauw, T., and Bayen, A., *Python Programming and Numerical Methods: A Guide for Engineers and Scientists*, Elsevier Science, 2020.

[3]Guckenheimer, J., "Chapter 8 - Numerical Analysis of Dynamical Systems* *This work was partially supported by grants from the Department of Energy, Air Force Office of Scientific Research and the National Science Foundation." *Handbook of Dynamical Systems*, edited by B. Fiedler, Vol. 2 of *Handbook of Dynamical Systems*, Elsevier Science, 2002, pp. 345–390.