THEORETICAL AND EXPERIMENTAL INVESTIGATION OF ATTITUDE CONTROL FOR UNDERACTUATED SPACECRAFT

Rajika Pati Arambage

A thesis submitted to Faculty of Graduate Studies

in partial fulfillment of the requirements for the degree of

Master of Applied Science

Graduate Program in Mechanical Engineering

York University

Toronto, Ontario

December 2020

ABSTRACT

This thesis investigates the detumbling and attitude control capabilities of an underactuated 3U-CubeSat used for ESSENCE mission developed by York University. Three different types of actuators—external thrusters, magnetorquers, and reaction wheels—are considered in the nonlinear rigid-body control problem. According to Brockett's theorem, a smooth or continuous feedback law does not exist in the attempt to stabilize the attitude dynamics, hence robust discontinuous control algorithms such as sliding mode control (SMC) and model predictive control (MPC) are considered. First, two thrusters are used to study the ability to detumble the spacecraft using SMC and nonlinear model predictive control (NMPC). Second, a comparison between the B-dot controller and NMPC in detumbling a spacecraft equipped with two magnetorquers is explored. SEET library in STK is utilized to obtain an accurate geomagnetic field model to model the space environment. Then, the use of two reaction wheels along the principal axes of the spacecraft to perform an attitude correction maneuver is studied using quaternion-based nonlinear control (QBNC) and NMPC. Finally, an air-bearing testbed with a floating satellite simulator that closely mimics the dynamics of a spacecraft is used to validate the use of NMPC for attitude control through experimentation. Air thrusters and a reaction wheel is used to perform two separate experiments to observe the performance of NMPC with each actuator. Design recommendations and challenges posed by underactuated configuration of the ESSENCE 3U-CubeSat are presented as a result of this thesis.

ACKNOWLEDGEMENTS

I would like to extend my greatest gratitude to my supervisor, Professor Zheng Hong (George) Zhu, without whom this thesis would not have been possible. I am honored and forever thankful for giving me the opportunity to be his student, through which I gained valuable knowledge. His continuous guidance and support, along with exceptional knowledge in space technology lead to the successful completion of this thesis. His accomplishments in academia are a true reflection of his remarkable determination and work ethic, which has been a true inspiration to me and many others.

I would also like to thank Mike Alger form Ryerson University for his continuous support throughout this thesis, although having never met in person. I would also like to give many thanks to my colleagues Gangqiang Li, Junjie Kang, Udai Bindra, Latheepan Murugathasan, and Lucas Santaguida for their outpouring support during this thesis.

Finally, I would like to thank the two most important figures in my life, my mom and dad, for their unconditional love and support throughout my academic life. I dedicate this thesis to you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

ADCS	Attitude Determination and Control System
CCW	Counter-Clockwise
CLF	Control-Lyapunov Function
CMG	Control Moment Gyro
COM	Center of Mass
DOF	Degrees of Freedom
ECR	Equal Concern for Relaxation
EKF	Extended Kalman Filter
IGRF	International Geomagnetic Reference Field
ISS	International Space Station
MIMO	Multi-Input-Multi-Output
MPC	Model Predictive Control
MV	Manipulated Variable
NMPC	Nonlinear Model Predictive Control
NRCSD	NanoRacks CubeSat Deployer
OBC	On-Board Computer
OV	Output Variable
PDB	Power Distribution Board
PID	Proportional-Integral-Derivative

RK4	Runge-Kutta 4th Order Method
RW	Reaction Wheel
SEET	Space Environment and Effects Tool
SMC	Sliding Mode Control
SSM	Satellite Simulator Module
STK	Systems Tool Kit software
STLC	Small-Time-Locally Controllable
STSMC	Super Twisting Sliding Mode Control

Chapter 1 INTRODUCTION AND JUSTIFICATION

1.1 Motivation and Justification

Attitude control of an underactuated spacecraft has become a topic of increasing interest in recent years. A spacecraft is considered underactuated when its degrees of freedom are higher than the degrees of actuation. Often a spacecraft become underactuated through an unexpected failure of its actuators. To avoid this situation, most spacecraft are equipped with redundant actuators so they can be utilized in an event of actuator failure. However, multiple actuator failures do occur, which is a common phenomenon. This was the case in FUSE, Dawn, Kepler Space Telescope and Hayabusa spacecraft [1]. The operations of these spacecraft were stopped prematurely before their full missions were achieved. Spacecraft can become underactuated not only by failure but also by design to lower mass and cost. The latter is the case in many small spacecraft, such as CubeSats due to their limited mass, volume and power budget. Hence, the objective of this study is to explore the detumbling and attitude stabilization capabilities of small underactuated spacecraft and to develop associated control strategies. The proposed control strategies are used in the development of the ESSENCE[1] 3U-CubeSat if it was either designed to be underactuated or becomes underactuated through actuator failure.

[1] ESSENCE refers to Educational Space Science and Engineering Cubesat Experiment mission funded by Canadian Space Agency. http://www.yorku.ca/cubesat/index

Losing control of a spacecraft can present many challenges to complete its mission. If control of the spacecraft is not regained, it will render the spacecraft unusable, becoming part of the ever-growing space junk that orbit the Earth. Therefore, it is beneficial to find ways to control the attitude of an underactuated spacecraft. Similar to any rigid body in space, a spacecraft has six degrees of freedom (DOF)—three translational and three rotational. Since attitude control is the focus of this thesis, only rotational DOF are of concern. Therefore, three actuators along each principal axis of the spacecraft are required for the spacecraft to be considered fully actuated. An underactuated spacecraft within the context of this report refers to a spacecraft having two or less actuators.

The research question to be explored in this thesis can be stated as "To which extent can a CubeSat's attitude be manipulated to meet its requirements when the attitude control system is underactuated?". To answer this question, three different scenarios have been explored in this thesis:

- Detumbling of a spacecraft equipped with two bi-directional thrusters placed along two principal axes of the spacecraft.
- Detumbling of a spacecraft equipped with two magnetorquers placed along two principal axes of the spacecraft.
- Controlling of a spacecraft's attitude with two reaction wheels (RWs) placed along two principal axes of the spacecraft.

Many control strategies have been developed over the years to tackle nonlinear control problems. Each control strategy however has their advantages and disadvantages, some of which are outlined in the next section. This thesis presents the application of model predictive control, sliding mode control, B-dot control, and quaternion-based nonlinear control to the nonlinear underactuated attitude control problem using different actuator types. Model predictive control is utilized within each actuator type and chosen as the benchmark controller, against which other control strategies are compared. This is because MPC is known to have exceptional performance and robustness due to its unique systematic approach to computing optimal control inputs. Robustness in control theory refers to the ability of the controller to meet stability and performance requirements in the presence of model and/or disturbance uncertainty [2]. The reason for utilizing sliding mode control, B-dot control, and quaternion-based control is that they are commonly used in literature to address the underactuated attitude control problem as seen in the next section.

1.2 Literature Review

The study of attitude control for underactuated spacecraft was first published in 1984 by P. Crouch [3]. This paper explored the attitude control capabilities of a spacecraft using two thrusters or two momentum exchange devices such as RWs. In the case of thruster actuation, the study found that two pairs of thrusters generating two independent torques or one thruster pair generating torque along a non-principal axis of the spacecraft allowed global controllability and small-time local controllability (STLC). The study also

concluded that using less than three reaction wheels made the system uncontrollable since the dynamics of the system were inaccessible. Later, a study by Byrnes and Isidori [4] in 1991, focused on thruster actuation confirmed Crouch's conclusion and added that these underactuated spacecraft systems cannot be locally asymptotically stabilized using a smooth feedback law due to the violation of Brockett's necessary conditions [5]. Another study in 1994 conducted by Krishnan et al [6] added that a spacecraft system is in fact STLC using two reaction wheels if the total angular momentum of the system is initially zero but cannot be stabilized using a smooth continuous feedback law. The study explained that if the system has zero total angular momentum, then a reduced set of attitude dynamics can be utilized to control the system. However, the practicality of obtaining zero angular momentum in a space environment is under scrutiny in literature. Additionally, C. Peterson in his recent study [7] further explained that it is not only impractical to obtain this condition but also undesirable. The study explained that reaction wheels will experience decreased accuracy, increased coefficient of friction, and decreased operational life since they must spin down during inertial pointing and operate in the zero-crossing region to obtain zero total angular momentum case. The study also explained that control-moment gyros (CMG) will also enter a singular configuration at equilibrium as a result of zero total angular momentum.

A paper published by R.W. Brocket in 1983 [5] outlined the conditions necessary for the existence of a smooth or continuous time-invariant feedback law to stabilize the dynamics of a system. Here, a smooth or continuous function is defined as a function that is

differentiable everywhere and belongs to class C^0. The challenge of underactuated attitude control problem relates directly to that study since the necessary conditions outlined by Brockett are not met. This has paved way for robust discontinuous state-feedback control and variable structure control methods to be utilized by underactuated spacecraft to perform attitude control maneuvers. The two most common control strategies implemented in literature are sliding mode control (SMC) and model predictive control (MPC).

Successful application of SMC for attitude control of an underactuated spacecraft is outlined in many publications such as in [8], [9], [10]. SMC is a variable structure control method with high frequency switching that results in discontinuous control inputs. Therefore, it is able to overcome the Brockett's necessary conditions, and become capable of stabilizing the attitude dynamics of an underactuated spacecraft in finite time. SMC is a well-known robust controller that can withstand external disturbances much better compared to traditional controllers like PID. SMC begins with the design of a sliding surface in state space, which includes the states of the system to be controlled. The time derivative of the sliding variable must contain the control input u, and the states of the system must converge to zero as the sliding surface converges to zero. The goal is to then design the control input u such that it drives the tracking error and its derivative to zero from any initial state in finite time. As states of the system slide along the sliding surface towards zero, high frequency switching control inputs are observed that result in the undesirable chattering phenomenon in SMC [11]. Many disadvantages of chattering in real-life electro-mechanical systems have been studied and discussed in the literature. These

include degradation and/or failure of actuator and other moving mechanical parts, heat losses in power circuits, low control accuracy, excited unmodeled dynamics in the control loop leading to undesirable oscillatory motion, introduction of vibration to the system, time lags in the process preventing high frequency switching leading to unfeasible application [12] [13] [14] [15]. However, this chattering phenomenon is a heavily studied research area and many workarounds have been proposed such as in [13]. It has also been proposed that chattering is most persistent in first order conventional sliding mode controllers, thus it can be reduced or eliminated by using higher order sliding mode controllers such as super-twisting sliding mode control (STSMC) [15] [16]. The use of STSMC in performing attitude control on a spacecraft will be further discussed in following chapters.

A more common control theory used in the underactuated attitude control problem is MPC as outlined in various studies like [7], [10], [17], [18], [19]. The reason for its popularity and its biggest advantage is the ability to implement constraints of the system within the controller design. Most states in a physical system are constrained to limits such as maximum/minimum: torque, thrust, turn angle, linear velocity, angular velocity, just to name a few. Therefore, it is beneficial to be able to integrate such constraints to the controller so the controller can calculate optimal control inputs based on the constraints. Optimal control inputs are computed systematically by MPC using a mathematical model of the plant and by solving an online optimization problem. At each time step, MPC solves the mathematical model of the system for a finite time in the future, called the prediction horizon, calculating a control input at each subsequent time step within the prediction

horizon. This set of control inputs at each time step is computed at current time step, and repeated *n* times to find the optimal set of control inputs that will produce the desired system output at the lowest cost. MPC then chooses only the first control input from the set and applies it to the system and discards all other calculated inputs. This process is then repeated at each time interval henceforth. Therefore, a large number of computed control inputs are discarded at each time step, which improves the robustness of the controller. However, the process makes MPC drastically more computationally expensive than other control algorithms. This can be noted as MPC's biggest drawback as it requires substantial amount of computer memory and higher processing power compared to other control algorithms [20] [21]. There are several ways in which the computational load can be reduced when utilizing MPC. Some of which can be outlined as model order reduction, smaller prediction horizon, smaller control horizon, reduced number of constraints. Another strategy is to employ a sub-optimal MPC. In the sub-optimal MPC, the number of times the controller computes subsequent control inputs at current time interval (*n*) to find the optimal set of control inputs is limited, i.e., a sub-optimal control input is computed to decrease computational load [21]. The use of sub-optimal MPC in the attitude control of a spacecraft will be further discussed in the following chapters. However, these strategies to minimize computational load come at the cost of controller performance. The control inputs from MPC are also discretized in time, therefore discontinuous, as can be seen on Figure 1.1. There are many variations of MPC such as linear MPC, adaptive MPC, gain-scheduled MPC, and nonlinear MPC. Since the attitude dynamics of a rigid body are non-linear, a

nonlinear model predictive controller (NMPC) will be utilized within the context of this thesis.

Various other methods have also been implemented in the literature for the attitude control of an underactuated spacecraft [7], such as the use of open-loop methods in [22], Lyapunov based feedback control in [23], [24], and time-varying feedback techniques in [25]. The control approach presented in [23] utilizes two reaction wheels placed along two principal axes of the spacecraft. It has been validated and flight tested on the UoSAT-12 spacecraft [26], which showed promising results. However, when it was implemented again in UK-DMC1 spacecraft [27], the controller showed poor resistance to external disturbances. This control algorithms will be further explored and applied to the ESSENCE 3U CubeSat in Chapter 5.

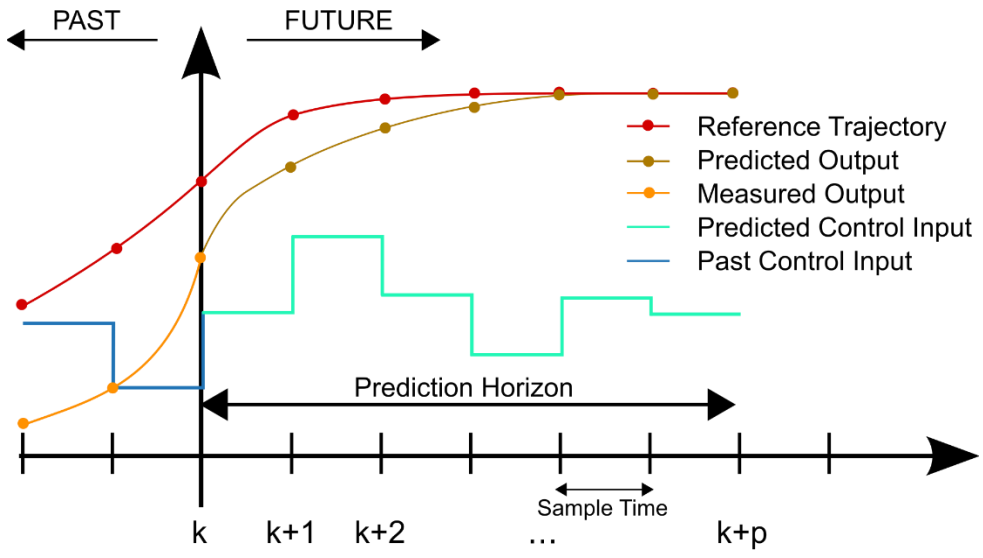

Figure 1.1 - Schematic outlining the workings of MPC [48].

1.3 Research Objectives

This thesis investigates the attitude control capabilities of a 3U-CubeSat under development at York University. The research objectives of this thesis can be outlined as:

1. Construct an accurate spacecraft attitude dynamics model.
    a. Develop a MATLAB-Simulink model to simulate attitude dynamics of a spacecraft.
    b. Model space environment dynamics and disturbances such as gravity gradient torque.
    c. Characterize the attitude of a spacecraft by quaternion representation to avoid the singularity of Euler angle representation (i.e., gimbal lock).
2. Study the underactuated spacecraft attitude control using different control algorithms with each actuator type.
    a. Study the effectiveness of SMC and NMPC with thruster actuation to detumble a spacecraft.
    b. Study the effectiveness of B-dot and NMPC with magnetorquer actuation to detumble a spacecraft.
    c. Study the effectiveness of QBNC and NMPC with reaction wheel actuation to perform attitude control maneuver.
3. Analyze controllability and stability of the spacecraft through numerical simulations.
4. Validate the use of NMPC on a physical spacecraft system on a microgravity testbed.

a. Utilize the air-bearing satellite simulator to validate the use of NMPC for attitude control.
b. Employ the cold air-thrusters on the spacecraft simulator along with NMPC to detumble the satellite simulator.
c. Utilize the reaction wheel on the spacecraft simulator along with NMPC to perform and attitude control maneuver.

1.4 Layout of Thesis

This document consists of eight chapters that outline the work that has been conducted as part of thesis requirements. Chapter 1 begins with an introduction and justification which outlines the motivation and research question, literature review, research objectives, and a background of the spacecraft system in question. In chapter 2, the equations of motion are defined including the coordinate frames, kinematics, attitude dynamics, and external disturbances and gravity gradient torque. The three subsequent chapters analyze the application of different control strategies to the attitude control of an underactuated spacecraft categorized by actuator type. Chapter 3 begins with the use of thruster actuation to detumble an underactuated spacecraft using STSMC and NMPC. Chapter 4 analyzes the use of magnetorquers along with B-dot and NMPC to detumble a spacecraft. Chapter 5 studies the use of reaction wheels to perform an attitude pointing maneuver using QBNC and NMPC. In chapter 6, the NMPC algorithm is validated through experimentation by performing attitude control maneuvers on the microgravity testbed with an air-bearing

satellite simulator. Two separate experiments are performed, one using air-thrusters and one using a reaction wheel mounted on the satellite simulator module. A conclusion, contribution of this work, and future work is outlined in chapter 7. Finally, a list of references used in this document is presented in the last chapter.

1.5 Background

1.5.1 Spacecraft Bus

As previously mentioned, the spacecraft bus in consideration is a 3U-CubeSat. A CubeSat is a class of pico-satellites that has dimensions of 10 cm × 10 cm × 10 cm per unit. Therefore, a 3U CubeSat has the dimensions 10 cm × 10 cm × 30 cm. A preliminary design schematic of the ESSENCE 3U-CubeSat is shown in Figure 1.2.

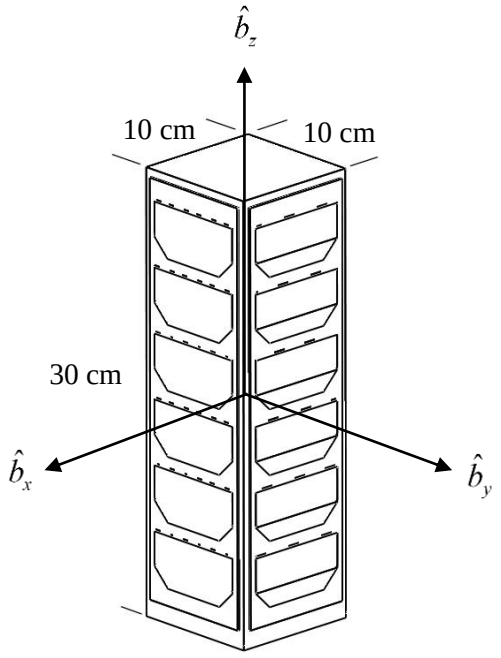

Figure 1.2 - 3D model drawing of the ESSENCE CubeSat.

According to preliminary calculations, the spacecraft's total mass is estimated to be around 4.8 kg. This along with the dimensions of the spacecraft mentioned earlier gives the moment of inertia matrix as,

$$\mathbf{J} = \text{diag}\{J_{xx}, J_{yy}, J_{zz}\}$$

$$\mathbf{J} = \begin{bmatrix} 4.88\times10^{-2} & 0 & 0 \\ 0 & 4.88\times10^{-2} & 0 \\ 0 & 0 & 1.56\times10^{-2} \end{bmatrix} kg \cdot m^2 \tag{1.1}$$

Since the spacecraft bus is a rigid-body and $\mathbf{J}$ is expressed in body-fixed frame $\mathcal{B}$, the following assumption is made.

$$\frac{d}{dt}(\mathbf{J}) = 0 \tag{1.2}$$

This assumption implies that the total mass and the mass distribution of the spacecraft system remains constant overtime. Therefore in Chapter 3 , where thruster actuation is employed to detumble a spacecraft, change in mass due to expended propellent is ignored.

1.5.2 Orbit and Deployment

The orbit of concern in the context of this thesis pertains to the orbit the ESSENCE CubeSat will operate in. ESSENCE CubeSat is to be deployed from NanoRacks CubeSat Deployer (NRCSD) onboard the ISS. Therefore, the orbit will have identical orbital parameters to the orbit of ISS as shown in Table 1.1. Due to the small eccentricity and small difference between perigee and apogee heights, the orbit can be generalized as a circular orbit.

Table 1.1 - Orbital parameters of the ISS [28].

Orbital Element	Value
Perigee Height	417 km
Apogee Height	419 km
Eccentricity	0.00015
Inclination	51.6413°
Right Ascension of the Ascending Node	157.858°
Argument of perigee	295.900°

NanoRacks have deployed over 180 CubeSats to date through its ISS CubeSat Deployment program. Each NRCSD platform can hold six CubeSat units making it possible to launch 1U, 2U, 3U, 4U, 5U and 6U CubeSats. When deploying CubeSats, the platform is carefully moved outside through the Kibo Module's Airlock [29]. A diagram of NanoRacks NRCSD platform is shown in Figure 1.3.

Figure 1.3 - CubeSats being deployed from the NRCSD onboard the ISS [28].

After deployment from the NRCSD platform, it is expected that the CubeSat will experience some initial tumbling due to asymmetric deployment forces applied to eject the CubeSat from the platform as can be seen on Figure 1.3. An approximation of this tumbling rate can be concluded by referring to NRCSD's Interface Definition Document (IDD) [30]. The requirement 4.1.6 2) states that,

- *The CubeSat shall be capable of withstanding up to five (5) deg/sec/axis tip-off rate.*

However, there have been instances where CubeSats have experienced much higher than expected initial tumbling rates, as in the case of QB50P1 and QB50P2, which experienced tumbling rates of ~30°/s [31].

1.5.3 Attitude Determination and Control System (ADCS)

The ADCS subsystem is an integral part of a spacecraft system that allows the spacecraft to determine and control its orientation in space. Sensors, actuators, and control laws are employed to enable the ADCS to achieve this task. Various types of sensors can be used to determine the attitude of a spacecraft, such as, gyroscopes, sun-sensors, star trackers etc. Using these sensory data as feedback, control laws can be developed to apply control torques via onboard actuators to change the orientation of a spacecraft to a desired target.

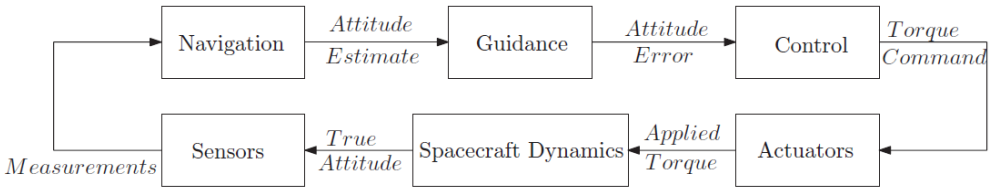

Figure 1.4 – Schematic of a spacecraft attitude control system [47].

Some of the most common actuators used for attitude control include thrusters, reaction wheels, magnetorquers, and CMGs. A generalization of such a control system is shown in Figure 1.4. As the ADCS is activated, it goes through various modes of operations. These modes include detumbling mode, safe mode, idle mode, and attitude pointing modes (nadir pointing, inertial pointing, target pointing). When a spacecraft is first deployed either from a launch vehicle, or from a platform in space, it experiences an initial tumbling, hence detumbling mode is activated to detumble the spacecraft to a desired angular velocity along each axis. Once the spacecraft is detumbled it may enter any of the other modes depending on the circumstances of the mission. If a spacecraft enters the attitude pointing mode, it may choose to follow one or more of the following attitude pointing strategies, nadir pointing, target pointing, and inertial pointing, as shown in Figure 1.5.

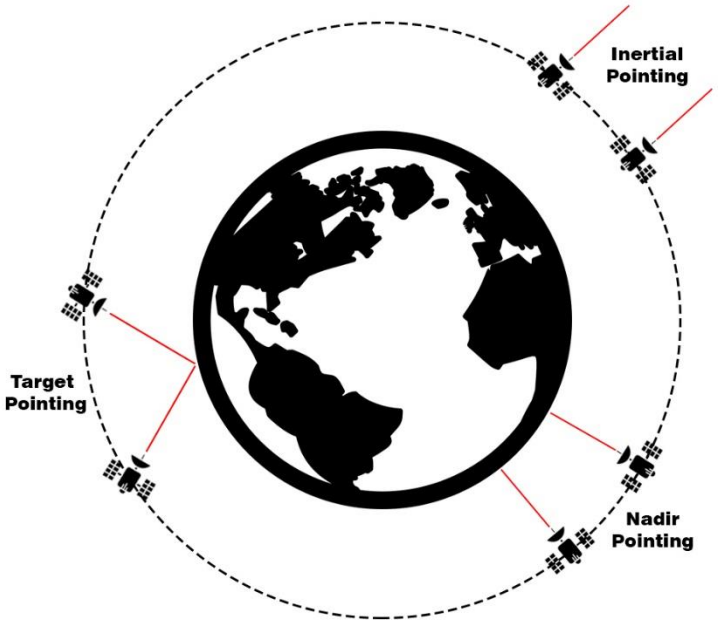

Figure 1.5 - Different attitude pointing modes of a spacecraft.

Chapter 2 EQUATIONS OF MOTION

2.1 Definition of Coordinate Frames

In this thesis, three main reference frames are used to characterize the motion of a spacecraft, as shown in Figure 2.1:

- The inertial frame $\mathcal{I}$ is fixed in space and has the orthogonal components $\hat{i}_x, \hat{i}_y, \hat{i}_z$.
- The body-fixed frame $\mathcal{B}$ is attached to the COM of spacecraft and has the orthogonal components $\hat{b}_x, \hat{b}_y, \hat{b}_z$ aligned with the principal axes of the spacecraft.
- The Local-Vertical-Local-Horizontal (LVLH) frame $\mathcal{A}$ is attached to the COM of spacecraft and has the orthogonal components $\hat{a}_x$, $\hat{a}_y$, $\hat{a}_z$, where $\hat{a}_x$ is aligned with spacecraft velocity vector, $\hat{a}_z$ is aligned in nadir direction, and $\hat{a}_y$ completes the right-handed coordinate system. Frame $\mathcal{A}$ has an angular velocity with respect to $\mathcal{I}$ given as:

$$\boldsymbol{\omega}_{\mathcal{A}/\mathcal{I}} = -n\hat{a}_y \tag{2.1}$$

where *n* is the mean motion of a spacecraft in circular orbit characterized by,

$$n = \sqrt{\frac{\mu}{a^3}} = \frac{2\pi}{P} \tag{2.2}$$

Here, μ is the standard gravitational parameter of earth, a is the semi-major axis of the orbit, and P is the orbital period of the spacecraft. Without loss of generality it is assumed that the orientation of frame $\mathcal{I}$ coincides with frame $\mathcal{A}$ at time t = 0,

$$(\hat{i}_x, \hat{i}_y, \hat{i}_z) = (\hat{a}_x, \hat{a}_y, \hat{a}_z)\big|_{t=0} \tag{2.3}$$

It is assumed without loss of generality that the inertial frame $\mathcal{I}$ is the desired attitude of the spacecraft, and the task in hand is to align $\mathcal{B}$ with $\mathcal{I}$.

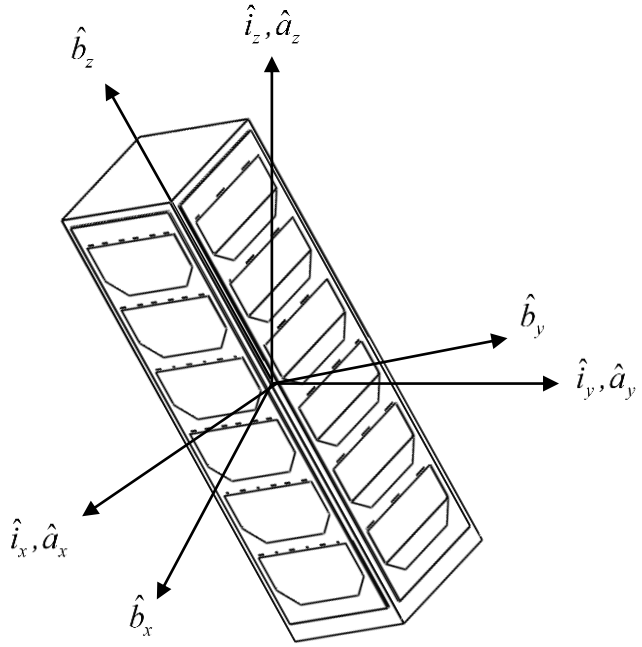

Figure 2.1 – The orientation of reference frames with respect to each other at time t = 0.

2.2 Kinematics

A quaternion formulation is used to represent the kinematics of the spacecraft system instead of classical Euler angle representation, since attitude representation using Euler angles contains a singularity (referred to as gimbal lock) as a consequence of using only three parameters to represent rotation in 3D space.

A quaternion is a 4-dimensional vector analogous to complex numbers. It consists of a scalar (similar to real part of a complex number) and a 3-dimensional vector (similar to imaginary part of a complex number). The challenge in working with quaternions comes from the fact that some basic mathematical operations like commutative multiplication do not apply. It is also less intuitive to imagine a rotation using quaternions compared to Euler angles. There exist many conventions of representing a quaternion. The convention used in this paper is shown below.

$$\boldsymbol{q}_{\mathrm{I}} = (q_1, q_2, q_3, q_0)^T \tag{2.4}$$

$$\boldsymbol{q}_{\mathrm{I}} = \begin{bmatrix} \boldsymbol{\epsilon}_{\mathrm{I}} \\ \eta_{\mathrm{I}} \end{bmatrix} \tag{2.5}$$

where q_0 and η_{I} denotes the scalar component and q_1, q_2, q_3 and $\boldsymbol{\epsilon}_{\mathrm{I}}$ denotes the vector components. The time derivative of the kinematics pertaining to the attitude dynamics using listed convention above can be given as,

$$\dot{\boldsymbol{\epsilon}}_{\mathrm{I}} = \frac{1}{2}\boldsymbol{\Gamma}(\boldsymbol{q}_{\mathrm{I}})\boldsymbol{\omega}$$

$$\dot{\eta}_{\mathrm{I}} = -\frac{1}{2}\boldsymbol{\epsilon}_{\mathrm{I}}{}^{T}\boldsymbol{\omega} \tag{2.6}$$

$$\boldsymbol{\Gamma}(\boldsymbol{q}_{\mathrm{I}}) = \eta_{\mathrm{I}}\mathbf{I} + \boldsymbol{\epsilon}_{\mathrm{I}}{}^{\times}$$

Kinematic equations outlined in (2.6) represent the orientation of the spacecraft's body fixed frame $\mathcal{B}$ relative to inertial frame $\mathcal{I}$.

2.3 Attitude Dynamics

The dynamics related to the area of research in this thesis are the dynamics concerning rotational motion of a rigid body. The equations are a set of nonlinear first-order ordinary differential equations (ODEs) formulated by Leonhard Euler, which characterize the rotational behaviour of a rigid body subjected to external moments. Equation (2.7) shows the general form of Euler's rotational equation [32].

$$\begin{gathered}\mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega}\times(\mathbf{J}\boldsymbol{\omega}) = \boldsymbol{\tau} \\ \dot{\boldsymbol{\omega}} = \mathbf{J}^{-1}\left[-\boldsymbol{\omega}\times(\mathbf{J}\boldsymbol{\omega}) + \boldsymbol{\tau}\right]\end{gathered} \tag{2.7}$$

where $\mathbf{J}$ is the moment of inertia matrix of the rigid body expressed in $\mathcal{B}$, $\boldsymbol{\omega}$ is the angular velocity of $\mathcal{B}$ relative to $\mathcal{I}$ expressed in $\mathcal{B}$, and $\boldsymbol{\tau}$ is the external torque vector acting on the rigid body expressed in $\mathcal{B}$.

The torque vector can be further expanded to include disturbances and control input as,

$$\boldsymbol{\tau} = \boldsymbol{\tau}_d + \boldsymbol{\tau}_g + \boldsymbol{\tau}_u \tag{2.8}$$

where, $\boldsymbol{\tau}_d$ is the disturbance torque vector, $\boldsymbol{\tau}_g$ is the gravity gradient torque vector, and $\boldsymbol{\tau}_u$ is the control input signal.

The attitude dynamics expressed in equation (2.7) can be further expanded to obtain,

$$\dot{\omega}_x = \frac{J_{yy}\omega_y\omega_z - J_{zz}\omega_y\omega_z + \tau_x}{J_{xx}} \tag{2.9}$$

$$\dot{\omega}_y = \frac{J_{xx}\omega_x\omega_z - J_{zz}\omega_x\omega_z + \tau_y}{J_{yy}} \tag{2.10}$$

$$\dot{\omega}_z = \frac{J_{xx}\omega_x\omega_y - J_{yy}\omega_x\omega_y + \tau_z}{J_{zz}} \tag{2.11}$$

Since the spacecraft is a 3U CubeSat as seen in Figure 1.2, it has two sides with identical dimensions. Therefore, its moment of inertia matrix has the property,

$$J_{xx} \simeq J_{yy} \tag{2.12}$$

By studying equations (2.9) to (2.11), it can be seen that for a 3U CubeSat satisfying equation (2.12), the equation (2.11) reduces to,

$$\dot{\omega}_z = \frac{\tau_z}{J_{zz}} \tag{2.13}$$

Hence, in the case of underactuated attitude control of a 3U CubeSat satisfying (2.12), ω_z becomes uncontrollable using actuation about $\hat{b}_x$ and $\hat{b}_y$ axes (XY actuation) alone. Here controllability means the existence of an admissible control input *u* that can drive the states of the system from an initial value to a desired final value in some finite time. It can be seen from equation (2.13), when τ_z is zero, the state $\dot{\omega}_z$ becomes uncontrollable, hence the system becomes uncontrollable. Therefore, in the upcoming chapters in which underactuated attitude control is analyzed, XY actuation is not considered since attitude dynamics does not allow controllability over ω_z due to spacecraft symmetry. Therefore, only XZ and YZ actuation will be considered.

The dynamics and kinematic equations outlined in (2.6) and (2.7) do not have analytic solutions, therefore, can only be solved numerically. Numerical methods such as the Runge-Kutta 4th order (RK4) method can be utilized to solve these first order ordinary differential equations. However, in this thesis, this process is accomplished using MATLAB-Simulink, which will solve the equations internally within the spherical joint block. In order to verify that the values computed by MATLAB-Simulink are accurate, a separate MATLAB script is developed that utilizes the RK4 method. A comparison of the data produced by self-developed RK4 method and the built-in MATLAB-Simulink solver are shown in Figure 2.2 and Figure 2.3. The two figures show the change in angular velocities and quaternions over time. An initial angular velocity of [0.1 0.2 –0.1] rad/s and an initial attitude of [0 0 0 1] with no external torques are used in the simulation.

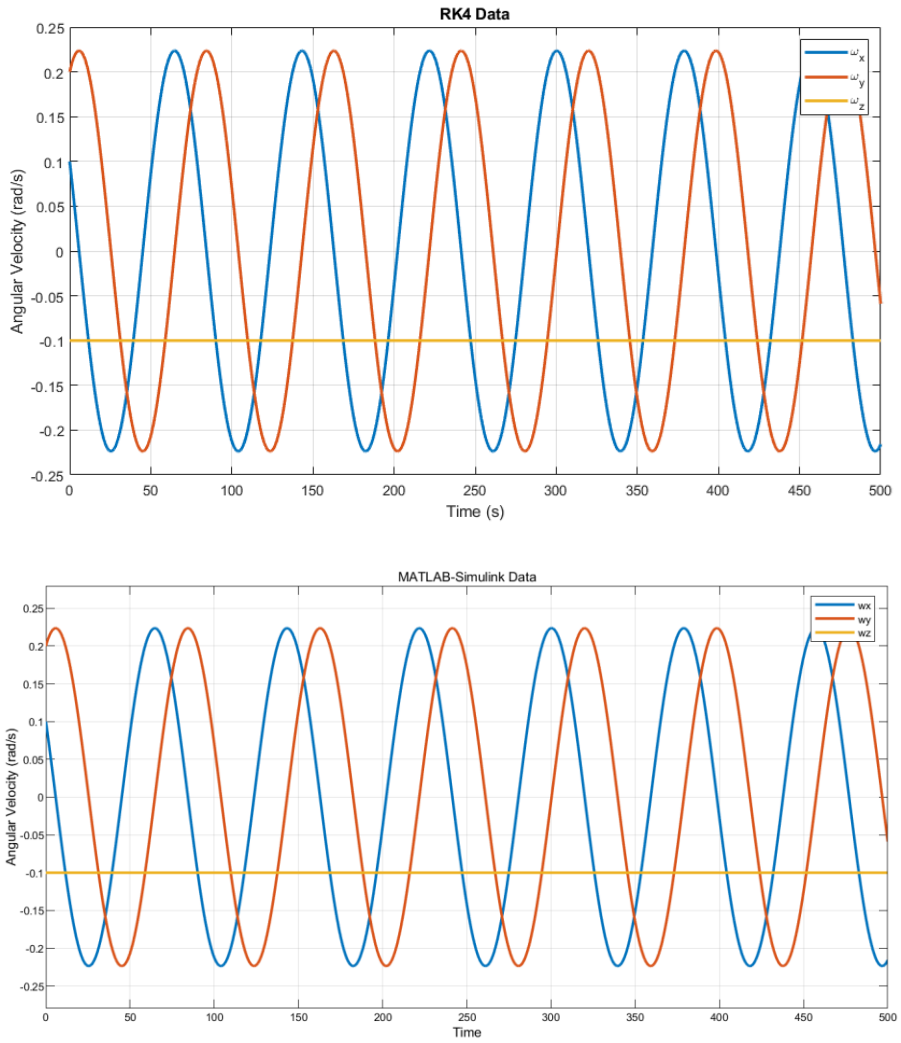

Figure 2.2 - Angular velocities resolved using RK4 Method (top) and MATLAB-Simulink (bottom).

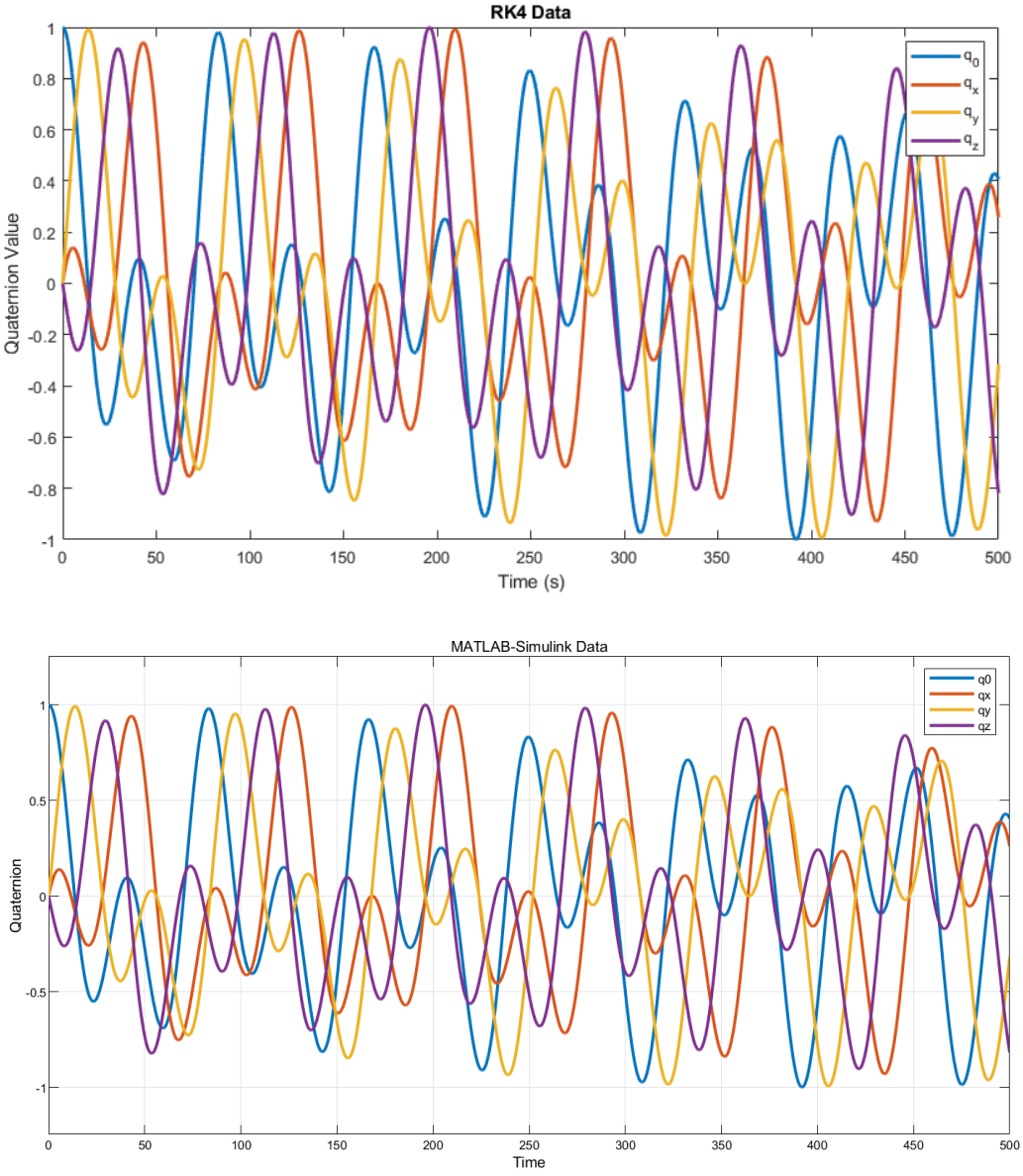

Figure 2.3 – Quaternions resolved using RK4 method (top) and MATLAB-Simulink (bottom).

The plots shown in Figure 2.2 and Figure 2.3 exhibit identical behaviour. Therefore, in the following sections, the built-in MATLAB-Simulink's spherical joint will be used to solve the attitude dynamics and kinematic equations for simplicity.

2.4 Disturbance and Gravity Gradient Torque

A spacecraft will be subjected to different forms of external torques as a result of the space environment. Equation (2.8) briefly characterized some of these torques. The two important external torques considered in this thesis are the disturbance torque and the gravity gradient torque. A disturbance torque is an unknown torque acting on the spacecraft. This disturbance torque is utilized to study the robustness of a controller in the following chapters. A time-varying periodic disturbance torque is modeled as,

$$\boldsymbol{\tau}_d = 5\times10^{-8}\sin(\boldsymbol{\omega}t) \tag{2.14}$$

The amplitude of the disturbance torque is chosen so that its effect is comparable to that of the gravity gradient torque (in the order of 10^{-8}) shown in equation (2.15). The gravity gradient torque is then modelled as,

$$\boldsymbol{\tau}_g = 3n^2\,\hat{a}_z\big|_{\mathcal{B}}\times\mathbf{J}\,\hat{a}_z\big|_{\mathcal{B}} \tag{2.15}$$

where, $\hat{a}_z\big|_{\mathcal{B}}$ is the unit vector in the direction of Earth's center resolved in frame $\mathcal{B}$.

This can be rewritten as,

$$\boldsymbol{\tau}_g = 3n^2(\mathbf{C}_{\mathcal{BA}}\,\hat{a}_z\big|_{\mathcal{A}})\times\mathbf{J}\mathbf{C}_{\mathcal{BA}}\,\hat{a}_z\big|_{\mathcal{A}} \tag{2.16}$$

where $\mathbf{C}_{\mathcal{BA}}$ describes the rotational transformation from orbital frame $\mathcal{A}$ to body frame $\mathcal{B}$ given as,

$$\mathbf{C}_{\mathcal{BA}} = (\eta_o{}^2 - \boldsymbol{\epsilon}_o{}^T\boldsymbol{\epsilon}_o)\mathbf{I} + 2\boldsymbol{\epsilon}_o\boldsymbol{\epsilon}_o{}^T - 2\eta_o\boldsymbol{\epsilon}_o{}^\times \tag{2.17}$$

Note that the quaternions η_o and $\boldsymbol{\epsilon}_o$ represent the orientation of orbital frame $\mathcal{A}$ relative to body-fixed frame $\mathcal{B}$.

Chapter 3 DETUMBLING USING THRUSTERS

This chapter explores the use of external thruster actuation to detumble a spacecraft using STMC and NMPC. External moment actuation is a common way of controlling both attitude and position of a spacecraft. However, in the case of CubeSats, the use of thrusters for attitude control purposes is almost non-existent. This is because thrusters require fuel which takes up space, and space is extremely limited on a CubeSat class spacecraft. Therefore, this study is done for purely academic reasons to explore how the system behaves under such circumstances. In this thesis, six thrusters are combined to create three pairs of bi-directional thrusters, each pair containing two thrusters providing variable thrust forces in opposite directions. Each thruster pair can generate a maximum thrust force of 350μN in each direction. The thrusters used in this chapter are hypothetical based on currently available thrusters such as MicroSpace's nanosatellite micro propulsion system [33] and ENPULSION's IFM nano thruster for CubeSats [34]. Thruster pairs 1, 2 and 3 are responsible for generating torques about $\hat{b}_x$, $\hat{b}_y$, and $\hat{b}_z$ axes respectively, as shown in Figure 3.2.

The simulation of the attitude control problem and the space environment is achieved using MATLAB-Simulink's Simscape Multibody toolbox. A 3D-model of the 3U-CubeSat is converted to .STEP format and imported to the Simulink workspace. Simulink resolves the moment of inertia matrix based on geometry once the mass of the object is provided. Then,

an unconstrained spherical joint is placed at the COM of the spacecraft model. This spherical joint is free to rotate in each of the three axes which mimics rotational motion of the spacecraft. The spherical joint module on Simulink takes control torques as inputs, then outputs angular velocity and quaternion orientation by solving Euler's equation given in equation (2.7). A schematic outlining how MATLAB-Simulink is employed is shown in Figure 3.1. The schematic shows only the basic concept of how the system works, not the original code. It has intentionally left out how gravity gradient torque and other components are modeled for easier understanding.

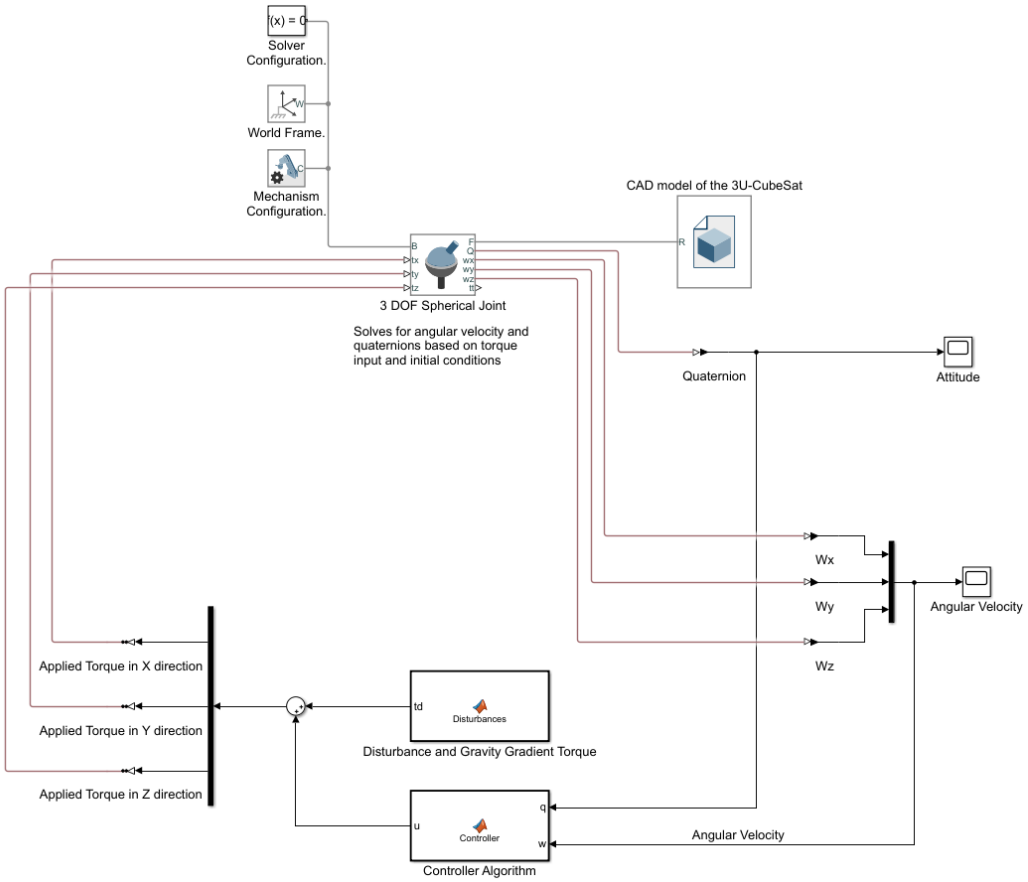

Figure 3.1 - Schematic outlining how MATLAB-Simulink is employed in the simulation of thruster actuation.

3.1 Super Twisting Sliding Mode Control

Sliding mode control is a robust non-linear control technique that utilizes a discontinuous control signal to control the dynamics of a system. It drives the system towards convergence

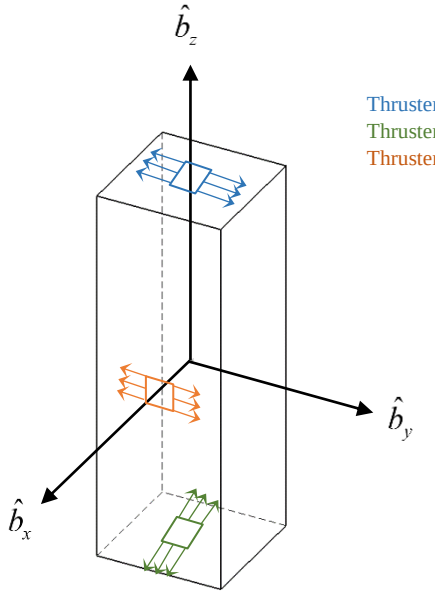

Figure 3.2 - Thruster configuration of spacecraft.

in finite time while maintaining robustness under bounded disturbance. A drawback of traditional first order sliding mode controllers is that it causes high frequency chattering. In a spacecraft system, this chattering phenomenon can introduce vibrations which can become problematic for the sensors, actuators, and other onboard components. Super twisting sliding mode control is a second order SMC that can eliminate some of the chattering. In this section, it is assumed that actuation over $\hat{b}_x$ axis is absent either through failure or through design. Provided that thrusters are used primarily in this thesis for detumbling the spacecraft, the system states can be stated as,

$$\boldsymbol{x} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{3.1}$$

In the sliding mode design, it is a common practice when dealing with underactuated spacecraft to separate the unactuated variables from the actuated variables as seen in [10]. Therefore, the states of the system are divided into unactuated and actuated states x_1 and $\boldsymbol{x}_2$ respectively,

$$\begin{aligned} x_1 &= \omega_x \\ \boldsymbol{x}_2 &= [\omega_y \quad \omega_z]^T \end{aligned} \tag{3.2}$$

Since the spacecraft is unactuated along $\hat{b}_x$ axis, the control input torques can be written as,

$$\boldsymbol{\tau}_u = \begin{bmatrix} u_y \\ u_z \end{bmatrix} \tag{3.3}$$

Then the sliding manifold is then constructed as,

$$\boldsymbol{s} = \boldsymbol{x}_2 + \mathbf{K}x_1 \tag{3.4}$$

where $\mathbf{K}$ is $[0 \quad 1]^T$.

The Lyapunov function to be analyzed for stability criterion is chosen as,

$$\mathrm{V} = \frac{1}{2}\boldsymbol{s}^T\boldsymbol{s} \tag{3.5}$$

The time derivative of (3.5) can be written as,

$$\dot{\mathrm{V}} = \boldsymbol{s}^T \dot{\boldsymbol{s}}$$

$$\dot{\mathrm{V}} = \boldsymbol{s}^T (\dot{\boldsymbol{x}}_2 + \mathbf{K}\dot{x}_1) \tag{3.6}$$

$$\dot{\mathrm{V}} = \boldsymbol{s}^T \begin{bmatrix} \dfrac{-J_{xx}\omega_x\omega_z + J_{zz}\omega_x\omega_z + u_y}{J_{yy}} \\ \dfrac{J_{xx}\omega_x\omega_y - J_{yy}\omega_x\omega_y + u_z}{J_{zz}} + \dfrac{J_{yy}\omega_y\omega_z - J_{zz}\omega_y\omega_z}{J_{xx}} \end{bmatrix} \leq 0$$

The control input $\boldsymbol{u}$ is then divided in to two parts,

$$\boldsymbol{u} = \boldsymbol{u}_{eq} + \boldsymbol{u}_d \tag{3.7}$$

The purpose of $\boldsymbol{u}_{eq}$ is to eliminate the non-linear terms seen on equation (3.6). This can be outlined as,

$$\boldsymbol{u}_{eq} = \begin{bmatrix} J_{xx}\omega_x\omega_z - J_{zz}\omega_x\omega_z \\ -J_{xx}\omega_x\omega_y + J_{yy}\omega_x\omega_y \end{bmatrix} \tag{3.8}$$

$\boldsymbol{u}_d$ is constructed in accordance with super-twisting algorithm shown in [35],

$$\begin{aligned} \boldsymbol{u}_d &= -\lambda_1 \frac{\boldsymbol{s}}{\sqrt{\|\boldsymbol{s}\|}} + \boldsymbol{v} - \lambda_2 \boldsymbol{s} \\ \dot{\boldsymbol{v}} &= -\lambda_3 \frac{\boldsymbol{s}}{\|\boldsymbol{s}\|} - \lambda_4 \boldsymbol{s} \end{aligned} \tag{3.9}$$

Proof of reachability and sliding on the sliding manifold requirements for the proposed control algorithm are outlined in equation (3.9) is provided in [35]. The constants $\lambda_1, \lambda_2, \lambda_3$ and λ_4 are computed by substituting equation (3.7) in to equation (3.6) and solving the inequality in equation (3.6).

3.2 Simulation Results

Table 3.1 provides the parameters used for simulation of the proposed super-twisting control algorithm in section 3.1. Due to the symmetrical characteristics of a CubeSat, it is sensible to assume that XZ actuation and YZ actuation will produce almost identical results. Therefore, only YZ actuation is studied in this section. Here, XZ actuation refers to thrusters being mounted such that they produce a net torque about $\hat{b}_x$ and $\hat{b}_z$ axes and vice versa.

Table 3.1 - Simulation parameters for STSMC with two thrusters.

Parameter	Value
K	$\begin{bmatrix}0 & 1\end{bmatrix}^T$
$(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$	$(2, 4\times10^5, 6, 100)$
$\boldsymbol{\omega}(0)$	$\begin{bmatrix}0.1 & 0.2 & -0.1\end{bmatrix}^T$ rad/s
$\boldsymbol{q}_{\mathrm{I}}(0)$	$\begin{bmatrix}0 & 0 & 0 & 1\end{bmatrix}^T$
$\max\left(\lvert\boldsymbol{u}\rvert\right)$	$1\times10^{-5}\cdot\begin{bmatrix}0 & 5.25 & 1.25\end{bmatrix}^T$ Nm

When detumbling a spacecraft, it is difficult to achieve a detumbled angular velocity of exactly zero in a space environment due to lack of damping in space. It is possible that a

small residual tumbling rate will always be present after detumbling phase is complete. Therefore, the following definition is employed to compare controller performance.

- *The spacecraft can be considered detumbled when,*

$$|\boldsymbol{\omega}| < 10^{-3} rad/s \tag{3.10}$$

It is a common practice in attitude detumbling literature to choose a threshold angular velocity, below which the spacecraft is considered detumbled. In [36], this threshold is taken as 0.1 deg/s (0.002 rad/s), and in [37] this threshold is taken as 0.3 deg/s (0.005 rad/s). Although these thresholds slightly vary from one paper to another, the order of magnitude remains the same.

Using the definition provided in equation (3.10), the time it took to detumble the spacecraft using STSMC with two thrusters generating torques about $\hat{b}_y$ and $\hat{b}_z$ is calculated as 3379.67 seconds (0.61 orbits). The simulation results are shown in Figure 3.3. Studying the attitude plot reveals that the controller was not able to completely detumble the spacecraft. However, looking at the angular velocities plot, it can be said that the controller performed well enough to get angular velocities as close as possible to zero. The thrust plot in Figure 3.3 exhibits the chattering phenomenon discussed earlier. High frequency switching between thrust values can be observed even after detumbling the spacecraft. This high frequency chattering can be harmful to the actuators and can introduced additional vibration

to the spacecraft system. Furthermore, any effects of external disturbance forces cannot be seen on angular velocity plot. This is a testament to the robustness of the STSMC controller.

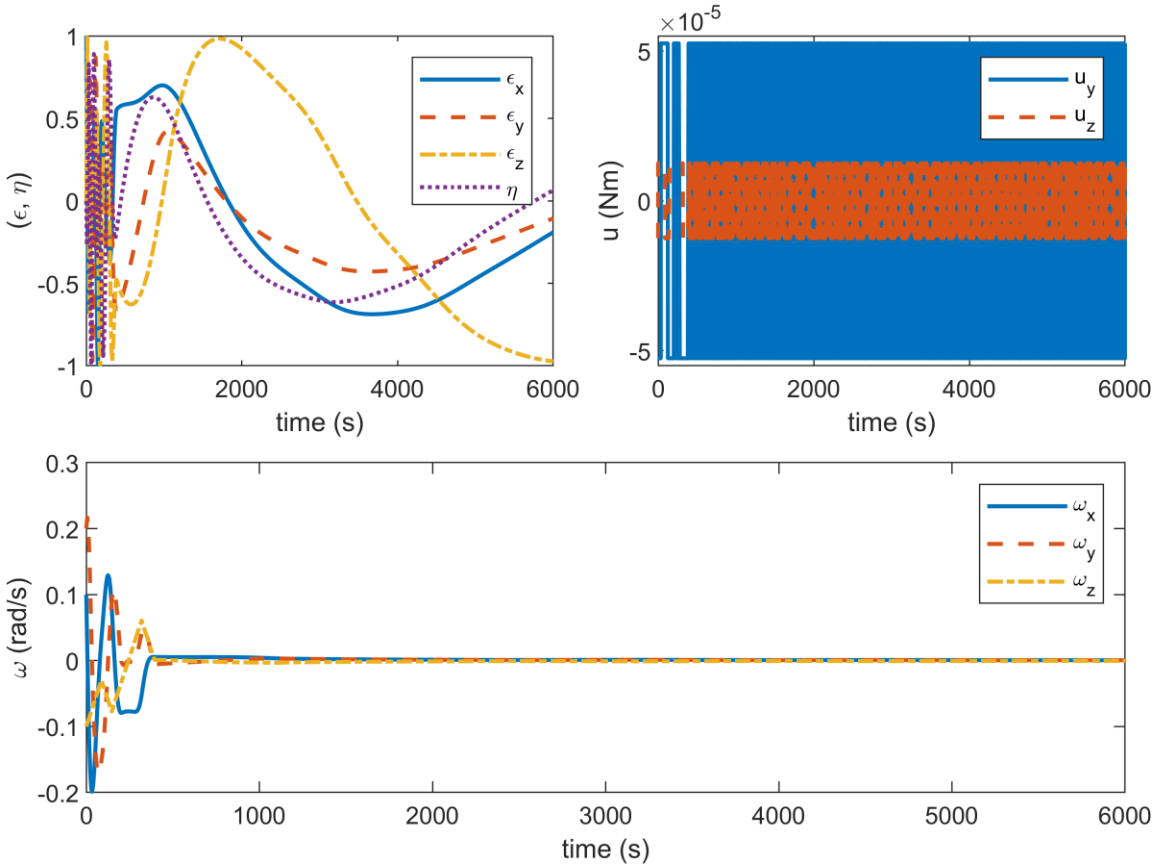

Figure 3.3 - Simulation results for STSMC using thruster actuation.

3.3 Sub-optimal NMPC

This chapter studies the use of suboptimal non-linear model predictive control to detumble a spacecraft from an initial tumbling rate using two external bi-directional variable thrusters. MPC utilizes a mathematical model of the plant to systematically compute optimal control inputs (manipulated variables) by solving an online optimization problem

to achieve a desired output from the system. MPC's biggest advantage is that it can implement constraints on system inputs, sates, and outputs such as thrust force, angular velocity, and attitude. Another advantage MPC provides is its ability to compute control inputs at current time step based on future behaviour of the system. To achieve this, MPC utilizes an Extended Kalman Filter (EKF) for state estimation. In this simulation, MATLAB-Simulink's EKF module is utilized.

A scalar cost function is utilized to handle the optimization that takes into consideration the constraints imposed, current system states and control inputs. Control inputs to the system and system response to the inputs are calculated *n* times at each time step over the prediction horizon. The sequence of inputs that minimizes the cost function is then chosen by the optimizer. However, only the first control input in the sequence is applied to the system and the rest are discarded by the algorithm [20] [21]. Because of this, the drawback of using MPC is its large computational load requirements—one that is especially unattractive to spacecraft. It requires a fast processor and large amounts of memory to execute the algorithm, both of which are not available in abundance on a spacecraft's on-board computer (OBC). There are however workarounds to this issue such as the use of sub-optimal MPC, which limits *n*, the number of iterations considered by the optimizer to find the solution that minimizes the cost function. This becomes a trade-off between controller performance and computing power. Considering the limited computing power available on a spacecraft, and that the dynamics of the system are nonlinear, a suboptimal NMPC is employed that restricts the number of optimizer iterations to a maximum of five iterations.

MATLAB's built-in MPC toolbox is utilized to simulate the scenario. The cost function employed by this toolbox outlined in [38] is,

$$J(z_k) = J_y(z_k) + J_u(z_k) + J_{\Delta u}(z_k) + J_\varepsilon(z_k) \tag{3.11}$$

where,

$$J_y(z_k) = \sum_{j=1}^{n_y}\sum_{i=1}^{p}\left\{\frac{w_{i,j}^y}{s_j^y}\left[r_j(k+i\,|\,k) - y_j(k+i\,|\,k)\right]\right\}^2 \tag{3.12}$$

$$J_u(z_k) = \sum_{j=1}^{n_u}\sum_{i=1}^{p-1}\left\{\frac{w_{i,j}^u}{s_j^u}\left[u_j(k+i\,|\,k) - u_{j,target}(k+i\,|\,k)\right]\right\}^2 \tag{3.13}$$

$$J_{\Delta u}(z_k) = \sum_{j=1}^{n_u}\sum_{i=1}^{p-1}\left\{\frac{w_{i,j}^{\Delta u}}{s_j^u}\left[u_j(k+i\,|\,k) - u_j(k+i-1\,|\,k)\right]\right\}^2 \tag{3.14}$$

$$J_\varepsilon(z_k) = \rho_\varepsilon \varepsilon_k^2 \tag{3.15}$$

where,

z_k — decision made by QP algorithm, given by:

$$z_k^T = \left[u(k\,|\,k)^T \quad u(k+1\,|\,k)^T \quad \ldots \quad u(k+p-1\,|\,k)^T \quad \varepsilon_k\right]$$

n_y — number of plant output variables.

p — prediction horizon.

$w_{i,j}^y$ — weight assigned for j[th] plant output at i[th] prediction horizon step.

s_j^y — scale factor for j[th] plant output.

$r_j(k+i \mid k)$ — reference value for j[th] plant output at i[th] prediction horizon step.

$y_j(k+i \mid k)$ — predicted value of j[th] plant output at i[th] prediction horizon step.

k — current control interval.

n_u — number of manipulated variables.

$w_{i,j}^{u}$ — tuning weight for j[th] MV at ith prediction horizon step.

s_j^u — scale factor for j[th] MV.

$u_j(k+i \mid k)$ — value for j[th] MV at i[th] prediction horizon step.

$u_{j,target}(k+i \mid k)$ — Target value for j[th] MV at i[th] prediction horizon step.

$w_{i,j}^{\Delta u}$ — tuning weight for j[th] MV movement at i[th] prediction horizon step.

ε_k — Slack variable at control interval *k*.

ρ_ε — Constraint violation penalty weight.

Equation (3.12) is the output reference tracking function J_y. It penalizes the controller when the desired output is different from the system output. The optimizer will attempt to minimize the penalty ensuring convergence of the system output towards the desired output.

J_u is the manipulated variable (MV) tracking function given in equation (3.13). This function is employed when it is required to keep the MVs at or near a specified target value. This is not utilized in this scenario.

$$J_u(z_k) = 0 \tag{3.16}$$

$J_{\Delta u}$ is the MV move suppression function stated in equation (3.14). In most cases limiting the change in control inputs is paramount. This is also the case in spacecraft detumbling since a sudden large change in control inputs, similar to chattering, may introduce shock vibrations to the spacecraft system and/or cause actuator failure, both of which can be undesirable.

Equation (3.15) is the constraint violation function J_ε. It takes into consideration the constraints imposed on the system inputs, outputs, and states. System constraints are characterized by its hardness. A hard constraint cannot be violated at any cost, while soft constraints can be violated to a certain degree specified by the control designer. The bounds on plant outputs, manipulated variables, and manipulated variable rates are given as,

$$\frac{y_{j,min}(i)}{s_j^y} - \varepsilon_k E_{j,min}^y(i) \le \frac{y_j(k+i\,|\,k)}{s_j^y} \le \frac{y_{j,max}(i)}{s_j^y} + \varepsilon_k E_{j,max}^y(i) \tag{3.17}$$

$$i = 1, 2, 3, \ldots, p \qquad j = 1, 2, 3, \ldots, n_y$$

$$\frac{u_{j,min}(i)}{s_j^u} - \varepsilon_k E_{j,min}^u(i) \le \frac{u_j(k+i\,|\,k)}{s_j^u} \le \frac{u_{j,max}(i)}{s_j^u} + \varepsilon_k E_{j,max}^u(i) \tag{3.18}$$

$$i = 1, 2, 3, \ldots, p \qquad j = 1, 2, 3, \ldots, n_u$$

$$\frac{\Delta u_{j,min}(i)}{s_j^u} - \varepsilon_k E_{j,min}^{\Delta u}(i) \le \frac{\Delta u_j(k+i\,|\,k)}{s_j^u} \le \frac{\Delta u_{j,max}(i)}{s_j^u} + \varepsilon_k E_{j,max}^{\Delta u}(i) \tag{3.19}$$

$$i = 1, 2, 3, \ldots, p \qquad j = 1, 2, 3, \ldots, n_u$$

where,

$(\cdot)_{j,min}(i)$ — lower bound for j[th] corresponding variable at i[th] prediction horizon step.

$(\cdot)_{j,max}(i)$ — upper bound for j[th] corresponding variable at i[th] prediction horizon step.

$s_j^{(\cdot)}$ — scale factor for j[th] corresponding variable.

ε_k — scalar QP slack variable (dimensionless) used for constraint softening.

$E_{j,min}^{(\cdot)}$ — Equal concern for relaxation (ECR) parameter determining the hardness of the lower bound constraint for the j[th] corresponding variable at i[th] prediction horizon step.

$E_{j,max}^{(\cdot)}$ — Equal concern for relaxation parameter determining the hardness of the upper bound constraint for the j[th] corresponding variable at i[th] prediction horizon step.

where following ECR values determines the consequent level of hardness of the constraint,

0 — No violation allowed (hard constraint)

0.05 — Very small violation allowed (nearly hard)

0.2 — Small violation allowed (quite hard)

1 — average softness

5 — greater-than-average violation allowed (quite soft)

20 — large violation allowed (very soft)

Since constraints are not imposed on the system outputs, equations (3.17) has infinite bounds. The constraints on manipulated variables are applied as hard constraints since the maximum thrust force applied by the thruster cannot be violated. This simplifies equation (3.18) to,

$$\frac{u_{j,min}(i)}{s_j^u} \leq \frac{u_j(k+i\,|\,k)}{s_j^u} \leq \frac{u_{j,max}(i)}{s_j^u} \tag{3.20}$$

3.4 Simulation Results

The relevant parameters for equations (3.12) to (3.15) are shown in Table 3.2. The simulation results are shown in Figure 3.4. The states of the system are identified as,

$$\boldsymbol{x} = \begin{bmatrix} \omega_x & \omega_y & \omega_z & \epsilon_x & \epsilon_y & \epsilon_z & \eta \end{bmatrix} \tag{3.21}$$

The states pertaining to attitude are also taken as system states so they can be observed in detail. Similar to section 3.1, two control torques produced by thrusters about $\hat{b}_y$ and $\hat{b}_z$ axes are considered. The output variables are the states to be controlled, hence identified as

the angular velocity states in equation (3.21), since the problem at hand is to detumble the spacecraft. Another thing to note is that when choosing weights for the output variables, a significantly higher value is chosen for the underactuated state. This ensures that higher priority is given by the controller to detumble the unactuated state ω_x.

Table 3.2 - Simulation parameters for NMPC with two thrusters.

Parameter	Value
Number of OVs (n_y)	3
Number of MVs (n_u)	2
Number of states (n_x)	7
Prediction horizon (p)	25 s
Control horizon (c)	5 s
Simulation time (T)	1000 s
Time step (t_s)	1 s
OV weights ($w_{i,j}^{y}$)	$[10 \quad 1 \quad 1] \qquad i = 1, 2, 3, ..., p$
OV scale factors (s_j^y)	[1 1 1]
OV reference ($r_j(k+i \mid k)$)	$[0 \quad 0 \quad 0] \qquad i = 1, 2, 3, ..., p$
MV rate weights ($w_{i,j}^{\Delta u}$)	$[1 \quad 1] \qquad i = 1, 2, 3, ..., p$
u_y constraints ($[u_{1,min} \quad u_{1,max}]$)	$1\times10^{-5} \cdot [-5.25 \quad 5.25]$ Nm
u_z constraints ($[u_{2,min} \quad u_{2,max}]$)	$1\times10^{-5} \cdot [-1.75 \quad 1.75]$ Nm
Initial angular velocity ($\boldsymbol{\omega}(0)$)	$[0.1 \quad 0.2 \quad -0.1]^T$ rad/s
Initial attitude ($\boldsymbol{q}_1(0)$)	$[0 \quad 0 \quad 0 \quad 1]^T$

It is immediately apparent from the simulation data that the NMPC performs much better compared to STSMC. The detumbling time for angular velocities $\boldsymbol{\omega}$ is much lower for NMPC in comparison to STSMC. This is highly expected since NMPC has the ability to

predict the future behaviour of the system, which STSMC does not. The spacecraft was detumbled to levels specified in equation (3.10) by NMPC in 430.98 seconds (0.078 orbits). This is an almost eight-fold improvement to 3379.67 seconds (0.612 orbits) by STSMC. However, the simulation run time and the computation resources required by NMPC exceeds the requirements of STSMC by a considerable amount, an amount that can be undesirable by spacecraft with limited computing power. Upon further inspection, it can be seen that NMPC was able to detumble the spacecraft much closer to zero compared to STSMC, indicated by horizontal lines on the attitude plot. Also, high frequency switching in control inputs are non-existent in NMPC. Therefore, when choosing between the two controllers, if the spacecraft system can withstand high computational demand, NMPC should be chosen to detumble an underactuated spacecraft using two thrusters.

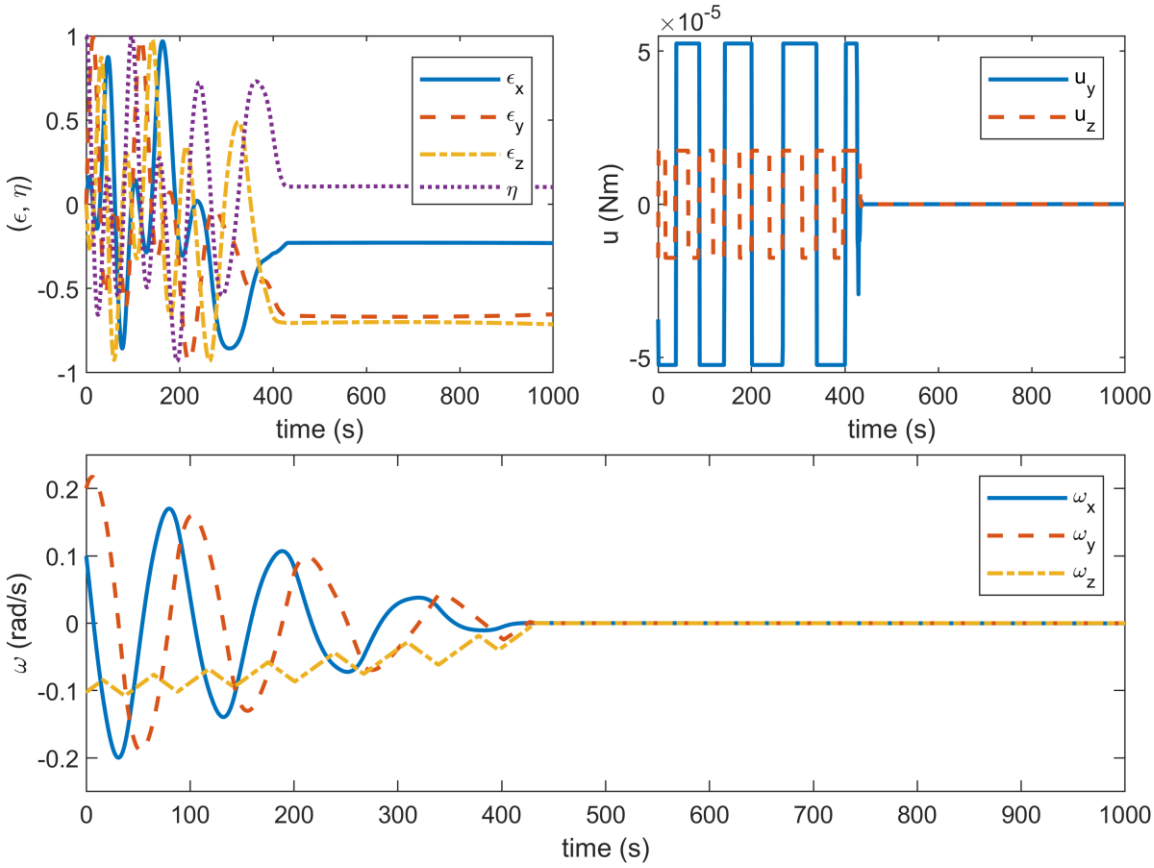

Figure 3.4 - Simulation results for NMPC using thruster actuation.

Chapter 4 DETUMBLING USING MAGNETORQUERS

Magnetorquers are commonly used actuators during the detumbling phase of a space mission that exploits Earth's geomagnetic field. In this chapter, the study of using two magnetorquers to detumble a spacecraft under external disturbances is explored. A novel method of simulating this scenario is presented, which utilizes STK and MATLAB-Simulink. STK allows the simulation to take use of its built-in magnetic field as well as orbital decaying properties to model the system dynamics more accurately. First, the magnetic field vector $\boldsymbol{B}^{mI} = [b_x^{mI} \quad b_y^{mI} \quad b_z^{mI}]^T$ in body-fixed frame of a spacecraft with an inertially fixed attitude is obtained from STK along the orbit of interest at time steps t_s. The vector $\boldsymbol{B^{mI}}$ is then transformed to vector $\boldsymbol{B}^{mb} = [b_x^{mb} \quad b_y^{mb} \quad b_z^{mb}]^T$, which gives the magnetic field vector with respect to the current attitude of the spacecraft $\boldsymbol{q}_I$ in body-fixed frame through equation (4.1). The quaternion $\boldsymbol{q}_I^*$ is the conjugate of $\boldsymbol{q}_I$.

$$\boldsymbol{B}^{mb} = \boldsymbol{q}_I \boldsymbol{B}^{mI} \boldsymbol{q}_I^* \tag{4.1}$$

An illustration characterizing this methodology is shown in Figure 4.1. The magnetic field simulated by STK uses IGRF (International Geomagnetic Reference Field) as the main field, which contains over 100 coefficients in the multi-pole spherical harmonic expansion [39]. The external field utilized in this simulation is the Olson-Pfitzer model. This simulation also employs the built-in SGP4 (Simplified General Perturbations) propagator which takes into account orbital variations due to: earth's oblateness, gravitational effects

from the moon and sun, gravitational resonance effects, and orbital decay [40]. The combination of using STK's complex magnetic field model along with its propagator will accurately mimic the magnetic field environment to be experienced by the ESSENCE 3U-CubeSat.

The magnetorquers used to be used on the ESSENCE CubeSat mission are used for this

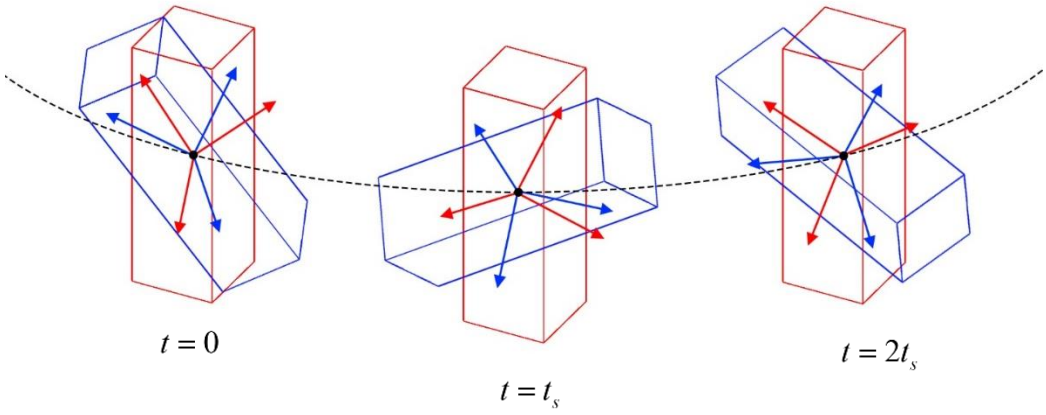

Figure 4.1 - Transformation of magnetic field vectors.

simulation. Three GOMSpace P110 solar panels with built-in magnetorquers are to be placed along the $+\hat{b}_x$ and $+\hat{b}_y$ faces of the CubeSat, and one GOMSpace NanoTorque Z-axis Internal magnetorquer is placed along the $+\hat{b}_z$ axis. In the following simulations,

magnetorquers placed on either $\hat{b}_x$ axis or $\hat{b}_y$ axis will be assumed to have failed. Referring to the datasheets, the magnetorquers' maximum dipole moments are listed on Table 4.1.

Table 4.1 - Maximum dipole moments generated along each axis.

Parameter	Value (Am^2)
$\lvert m_{x\_max} \rvert$	0.129
$\lvert m_{y\_max} \rvert$	0.129
$\lvert m_{z\_max} \rvert$	0.139

The placement of the magnetorquers on the ESSENCE 3U-CubeSat are shown in Figure 4.2. In the following sections, either the magnetorquer placed along the $\hat{b}_x$ axis or the $\hat{b}_y$ axis will be considered to have failed or does not exist by design.

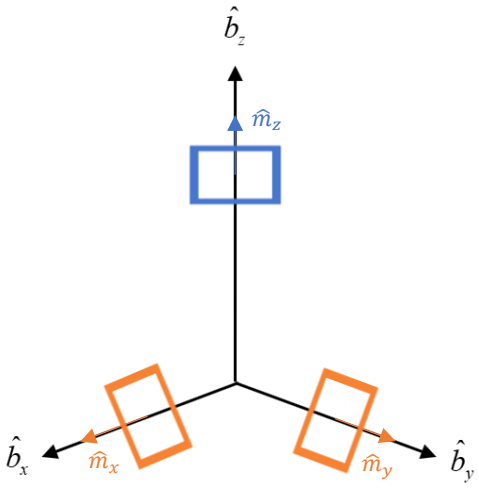

Figure 4.2 - Configuration of magnetorquers on the ESSENCE 3U-CubeSat.

Magnetorquers employ the well-known interaction of a magnetic dipole moment placed in a magnetic field. This relationship is shown in equation (4.2).

$$\boldsymbol{\tau} = NiA \times \boldsymbol{B}^{mb} \tag{4.2}$$

where, N is the number of loops of wire, i is the current, and $\boldsymbol{A}$ is the vector area of the coil in body-fixed frame. These terms can be grouped to be denoted as $\boldsymbol{m}$—the magnetic dipole moment as shown in equation (4.3). When this magnetic dipole moment interacts with a magnetic field $\boldsymbol{B}$, a torque $\boldsymbol{\tau}$ is generated orthogonal to $\boldsymbol{m}$ and $\boldsymbol{B}$.

$$\boldsymbol{\tau} = \boldsymbol{m} \times \boldsymbol{B}^{mb} \tag{4.3}$$

The magnetorquers are placed such that the $[m_x \quad m_y \quad m_z]$ vector components are parallel and coincident with the body fixed frame components $[\hat{b}_x \quad \hat{b}_y \quad \hat{b}_z]$. The challenge in using magnetorquers is the fact that they cannot completely detumble a spacecraft. This is because once $\boldsymbol{m}$ vector becomes aligned with the local geomagnetic field, torques cannot be generated any further.

4.1 B-dot Controller

The purpose of the B-dot controller then is to find values of $\boldsymbol{m}$, so that it generates an appropriate control torque to detumble the spacecraft. The problem formulation given in (2.7) can be rewritten for this scenario using the appropriate terms as,

$$\dot{\boldsymbol{\omega}} = \mathbf{J}^{-1}\left[-\boldsymbol{\omega} \times (\mathbf{J}\boldsymbol{\omega}) + \boldsymbol{\tau}_g + \boldsymbol{\tau}_d + (\boldsymbol{u} \times \boldsymbol{B}^{mb})\right] \tag{4.4}$$

where $\boldsymbol{m}$ is replaced with control input variable $\boldsymbol{u}$. Two variations of the B-dot controller—conventional B-dot and enhanced B-dot controllers are shown in equations (4.5) and (4.6) respectively.

$$\boldsymbol{m} = \boldsymbol{u} = -k \cdot \dot{\boldsymbol{B}}^{mb} \tag{4.5}$$

$$\boldsymbol{m} = \boldsymbol{u} = \frac{-k_{\omega}}{\left\| \boldsymbol{B}^{mb} \right\|^2} (\boldsymbol{B}^{mb} \times \boldsymbol{\omega}) \tag{4.6}$$

The gain value k_{ω} in equation (4.6) is calculated according to [41] as,

$$k_{\omega} = 2n(1+\sin\xi)J_{\min} \tag{4.7}$$

Where n is the mean motion given in (2.2) and ξ is the is the inclination of the orbital plane with respect to the geomagnetic equator and $J_{\min}$ is the minimum moment of inertia of the spacecraft.

4.2 Simulation Results

Since the magnetic field vector vary over the course of the orbit, XZ actuation and YZ actuation no longer exhibit identical behaviour due to spacecraft symmetry. Therefore, for each controller, both XZ actuation and YZ actuation is performed separately to study any discrepancies between the two. Here, XZ actuation refers to two magnetorquers being placed along $\hat{b}_x$ and $\hat{b}_z$ axes such that m_x and m_z vectors are parallel and coincident with said axes. First, XZ actuation using the conventional B-dot algorithm proposed in equation (4.5) is simulated using the parameters shown in Table 4.2.

Table 4.2 - Simulation parameters for conventional B-dot controller.

Parameter	Value
Time step (t_s)	0.01 s
Conventional B-dot gain (k)	2.5×10^5
$\max(\lvert\boldsymbol{m}\rvert) = \max(\lvert\boldsymbol{u}\rvert)$	$\begin{bmatrix}0.129 & 0.129 & 0.139\end{bmatrix}^T$ Am^2
Initial angular velocity ($\boldsymbol{\omega}(0)$)	$\begin{bmatrix}0.1 & 0.2 & -0.1\end{bmatrix}^T$ rad/s
Initial attitude ($\boldsymbol{q}_1(0)$)	$\begin{bmatrix}0 & 0 & 0 & 1\end{bmatrix}^T$

By observing the results in Figure 4.3 and Figure 4.4 it is apparent that XZ actuation and YZ actuation results are different as expected. It is also apparent that the conventional B-dot as it stands has great difficulty in detumbling the spacecraft. After some gain tuning, the B-dot controller exhibited similar behaviour, hence concluding chosen gain value was not at fault for poor behaviour.

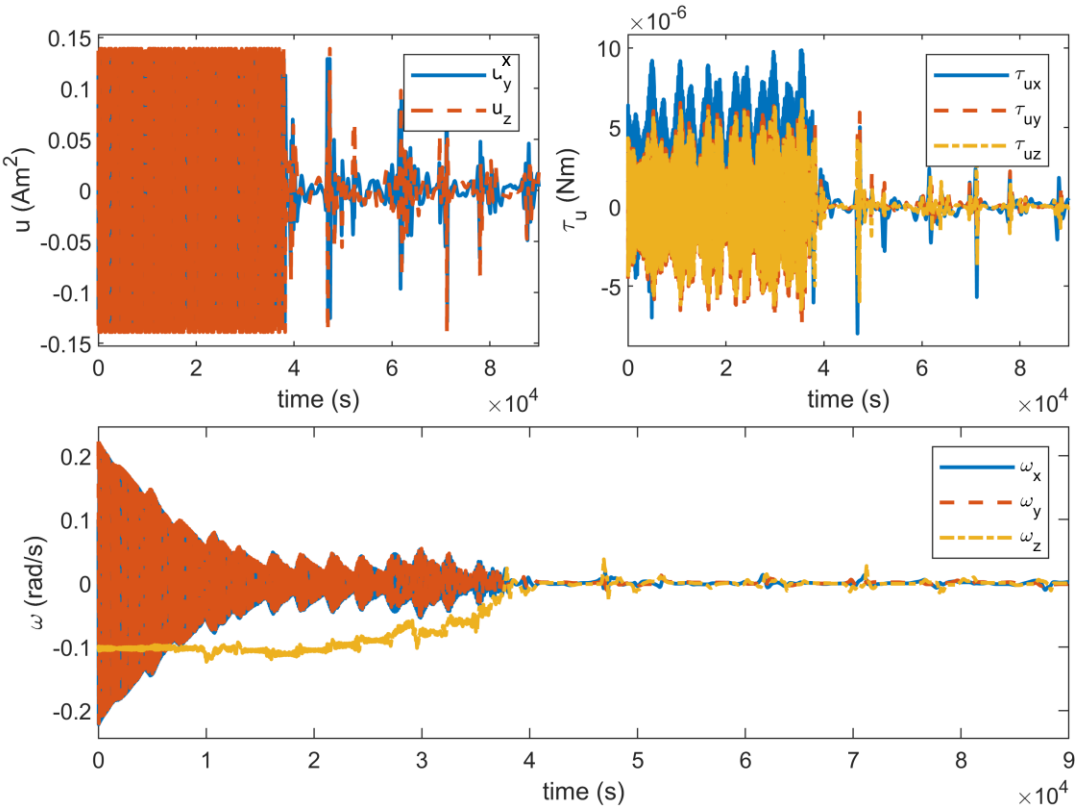

Figure 4.3 - Detumbling spacecraft using conventional B-dot algorithm and XZ actuation.

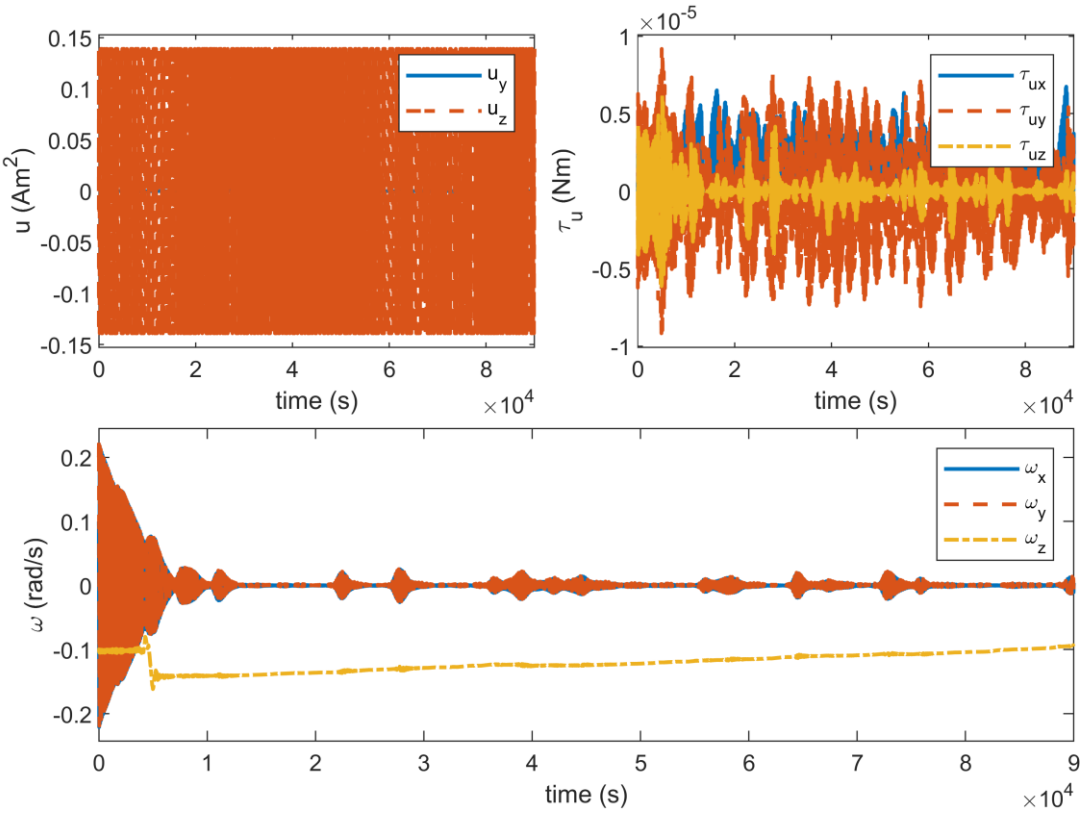

Figure 4.4 - Detumbling spacecraft using conventional B-dot algorithm and YZ actuation.

The simulation was performed again using magnetorquers with larger magnetic dipole moments ***m*** to observe its effect. The magnetic dipole moments were changed such that,

$$\max((|\boldsymbol{m}|) = \max(|\boldsymbol{u}|) = [1.0 \quad 1.0 \quad 2.0]^T \tag{4.8}$$

The gains were also slightly adjusted to,

$$k = 1\times 10^4 \cdot [2.5 \quad 2.5 \quad 5.0]^T \tag{4.9}$$

After implementation of these changes, it can be observed that the results have improved significantly as seen on Figure 4.5 and Figure 4.6. XZ actuation has improved the most with good convergence to a neighbourhood of zero within 15716.09 seconds (2.8 orbits). However, even with these changes, the YZ actuation could not converge the angular velocities. From these results it can be concluded that although the conventional B-dot does a good job at getting the angular velocities to a neighbourhood of zero, it is not effective in detumbling the spacecraft fully. This could be due variations in the magnetic field vector that is preventing the magnetorquer placed on $\hat{b}_y$ axis from exerting the necessary torque to detumble the spacecraft effectively.

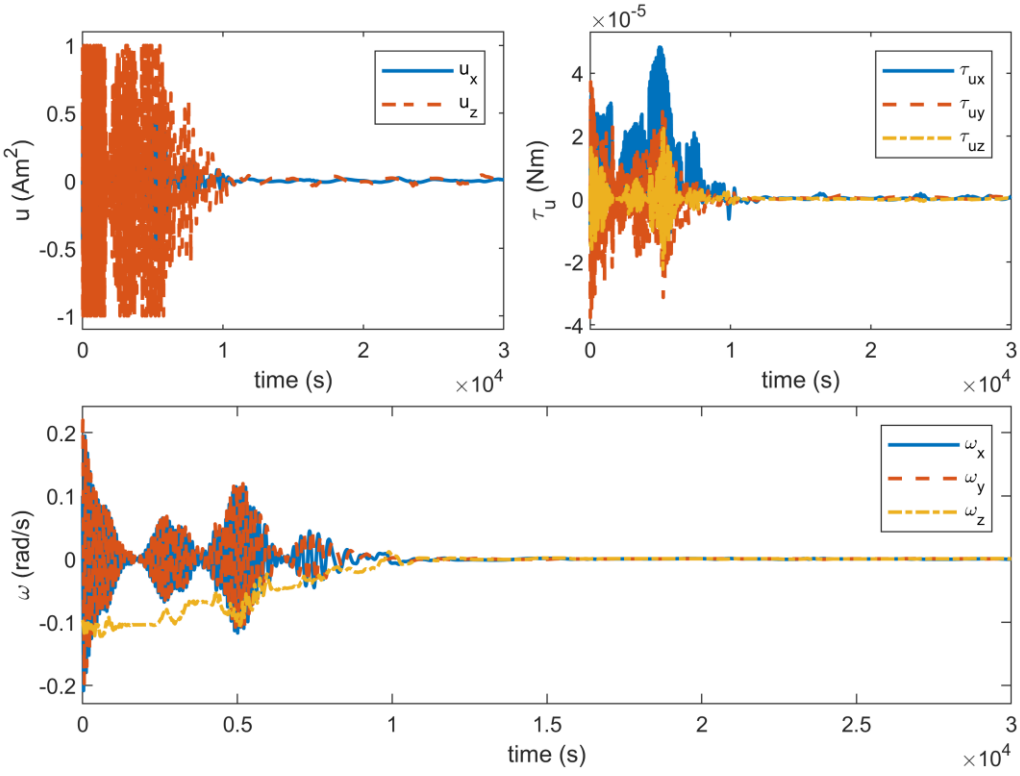

Figure 4.5 - Detumbling spacecraft using conventional B-dot algorithm and XZ magnetorquer actuation with increased magnetic dipole m.

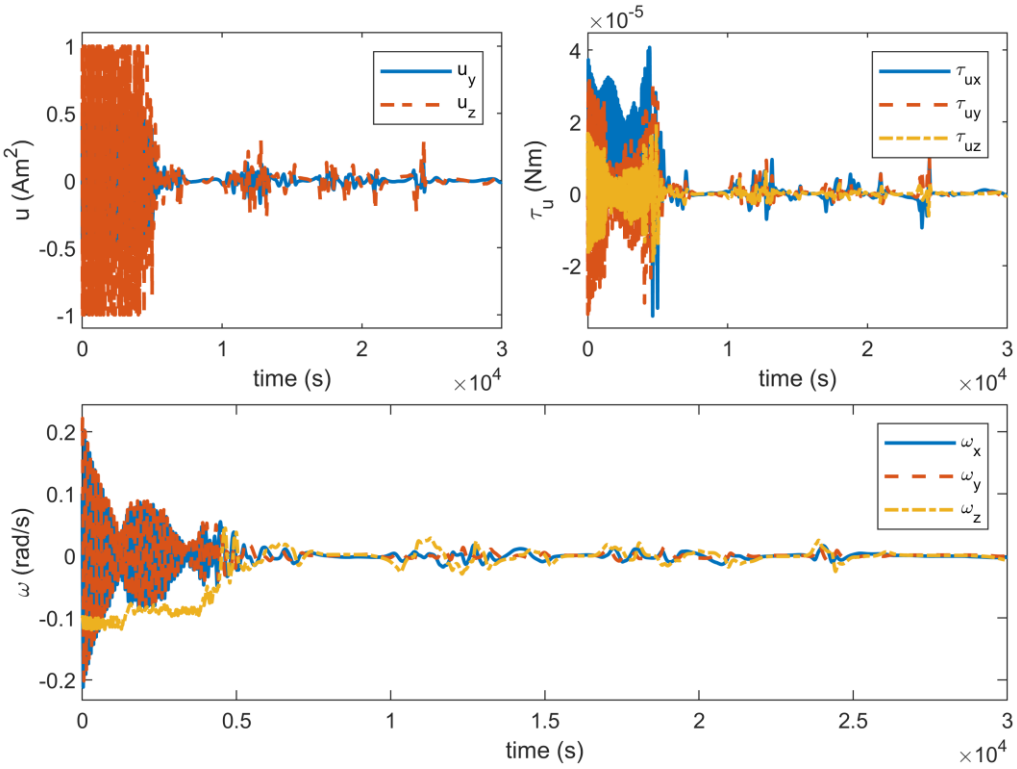

Figure 4.6 - Detumbling spacecraft using conventional B-dot algorithm and YZ magnetorquer actuation with increased magnetic dipole m.

Next, the use of enhanced B-dot algorithm given in equation (4.6) with two magnetorquers is explored. Similar to the conventional B-dot algorithm, the enhanced B-dot algorithm is run twice with XZ actuation and YZ actuation. One advantage of this algorithm is that computation of the time derivative of $\boldsymbol{B}$ is no longer required, which can reduce the onboard computations required. The parameters used for this simulation are shown in Table 4.3.

Table 4.3 - Simulation parameters for enhanced B-dot controller.

Parameter	Value
Time step (t_s)	0.01 s
Enhanced B-dot gain (k_ω)	4.2254×10^{-5}
$\max((\lvert\boldsymbol{m}\rvert) = \max(\lvert\boldsymbol{u}\rvert)$	$\begin{bmatrix}0.129 & 0.129 & 0.139\end{bmatrix}^T$
Initial angular velocity ($\boldsymbol{\omega}(0)$)	$\begin{bmatrix}0.1 & 0.2 & -0.1\end{bmatrix}^T$
Initial attitude ($\boldsymbol{q}_I(0)$)	$\begin{bmatrix}0 & 0 & 0 & 1\end{bmatrix}^T$

The simulation results are shown in Figure 4.7 and Figure 4.8 for the enhanced B-dot controller. In comparison to the conventional B-dot algorithm, the enhanced B-dot algorithm seem to display increased robustness and better performance. The convergence of angular velocities in enhanced B-dot is improved over the conventional B-dot, in that both XZ and YZ actuations drive the system to a neighbourhood of zero within similar time. Both XZ and YZ actuation exhibit identical behaviour that differences in the magnetic field vector $\boldsymbol{B}^{mb}$ with respect to $\boldsymbol{m}$ does not seem to have a significant effect, unlike in the conventional B-dot algorithm. The XZ actuation detumbled the spacecraft as per equation (3.10) in 14306.99 seconds (2.59 orbits), whereas YZ actuation detumbled the spacecraft in 13887.36 seconds (2.52 orbits). It is important to note that in the enhanced B-dot algorithm, stronger magnetic dipole moments were not required to achieve good results. Therefore, changes to the initial ESSENCE CubeSat design will not be needed if enhanced B-dot algorithm is utilized. Considering the results from this simulation it is highly recommended that if a B-dot algorithm is to be used to detumble the ESSENCE CubeSat,

the enhanced B-dot algorithm should be used, as it is more robust and can perform better with two magnetorquers than the conventional B-dot algorithm.

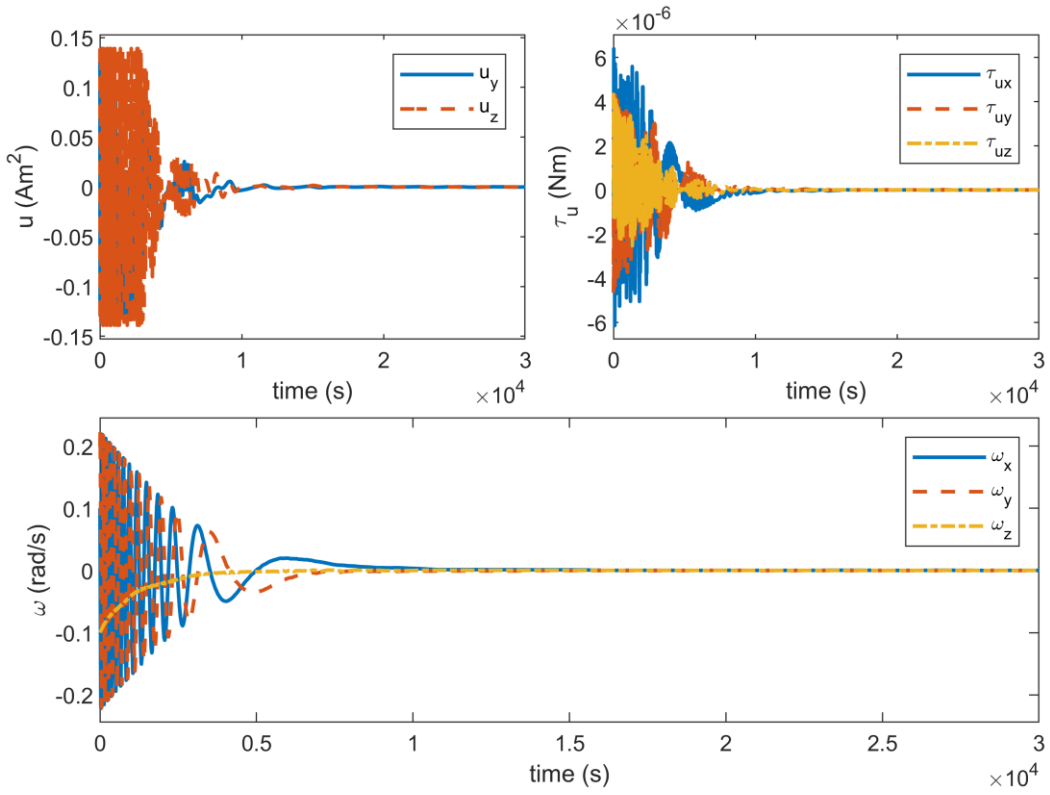

Figure 4.7 - Detumbling spacecraft using enhanced B-dot algorithm and YZ actuation.

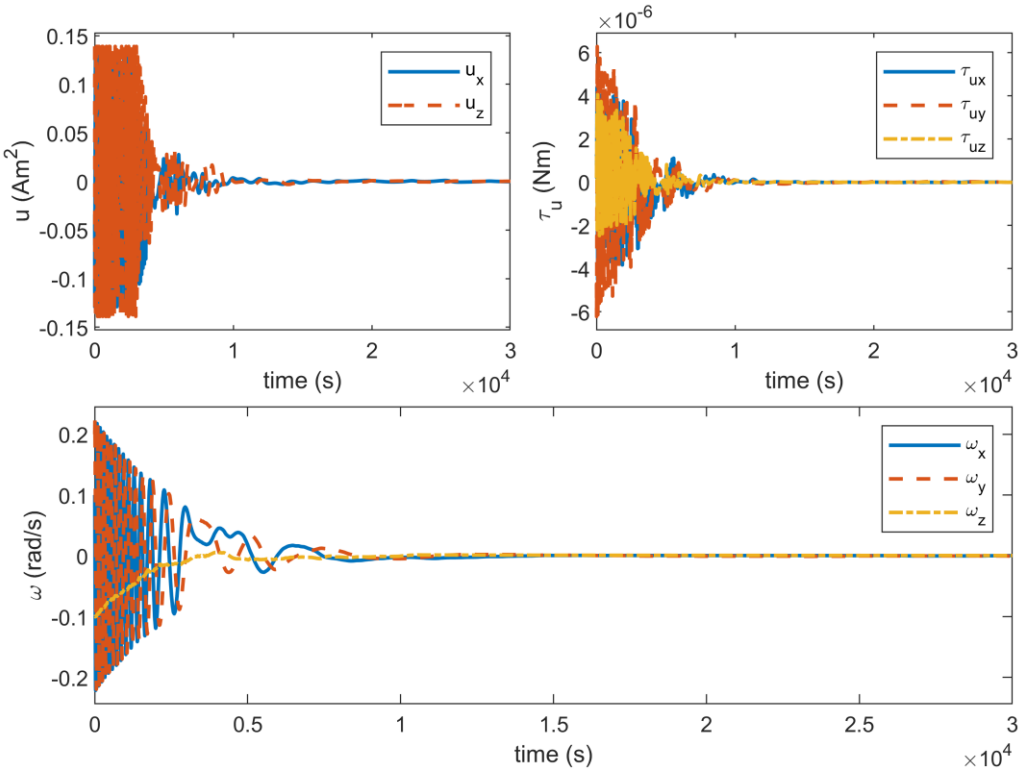

Figure 4.8 - Detumbling spacecraft using enhanced B-dot algorithm and XZ actuation.

4.3 Sub-optimal NMPC

The study of using two magnetorquers to detumble a spacecraft using a sub-optimal NMPC controller is explored in this section. Both XZ and YZ actuation is considered to observe any differences. The dynamics of the system are expressed in equation (4.4). To increase the running time and off load some of the computational load from spacecraft's OBC, the NMPC will restrict the number of iterations considered by the optimizer to five. The cost function utilized in the system is identical to the cost function given in equation (3.11). The system states are given as,

$$\boldsymbol{x} = \begin{bmatrix} \omega_x & \omega_y & \omega_z & \epsilon_x & \epsilon_y & \epsilon_z & \eta \end{bmatrix} \tag{4.10}$$

When detumbling a spacecraft, the states to be controlled are the angular velocities of the spacecraft. Hence, the system outputs are identified as the first three states in equation (4.10). The control inputs or manipulated variables generated by NMPC are in the form of magnetic dipole moments ***m***. The control inputs ***m*** are then converted to torques using (4.3) before being applied to the system. The maximum dipole moments capable of being generated by the magnetorquers according to datasheets are imposed as hard constraints using,

$$\frac{u_{j,min}(i)}{s_j^u} \leq \frac{u_j(k+i \mid k)}{s_j^u} \leq \frac{u_{j,max}(i)}{s_j^u}$$
$$i = 1, 2, 3, \ldots, p \qquad j = 1, 2 \tag{4.11}$$

4.4 Simulation Results

The parameters used in the simulation are shown in Table 4.4. The simulation results for the XZ actuation and YZ actuation are shown in Figure 4.9 and Figure 4.10 respectively.

Table 4.4 - Simulation parameters for NMPC using two magnetorquers.

Parameter	Value
Number of OVs (n_y)	3
Number of MVs (n_u)	2
Number of States (n_x)	7

Prediction horizon (p)	25 s
Control Horizon (c)	5 s
Simulation Time (T)	15000 s
Time Step (t_s)	1 s
OV weights ($w_{i,j}^{y}$)	[6 6 6] $i = 1, 2, 3, ..., p$
OV scale factors (s_j^y)	[1 1 1]
OV reference ($r_j(k+i \mid k)$)	[0 0 0] $i = 1, 2, 3, ..., p$
MV rate weights ($w_{i,j}^{\Delta u}$)	[1 1] $i = 1, 2, 3, ..., p$
u_x / u_y constraints ($[u_{1,min} \quad u_{1,max}]$)	[–0.129 0.129] Am^2
u_z constraints ($[u_{2,min} \quad u_{2,max}]$)	[–0.139 0.139] Am^2
Initial angular velocity ($\boldsymbol{\omega}(0)$)	$\begin{bmatrix} 0.1 & 0.2 & -0.1 \end{bmatrix}^T$ rad/s
Initial attitude ($q_I(0)$)	$\begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T$

The XZ actuation detumbled the spacecraft as per (3.10) in 13737 seconds (2.49 orbits). YZ actuation detumbled the spacecraft in 9538 seconds (1.72 orbits). This in comparison to the enhanced B-dot algorithm with a XZ detumbling time of 14306.99 seconds (2.59 orbits) and a YZ detumbling time of 13887.36 seconds (2.52 orbits) is a slight improvement. However, this slight improvement does not justify the increased computational load inflicted on spacecraft's OBC. Therefore, in a two-magnetorquer spacecraft system with limited computational power, such as in the ESSENCE 3U-CubeSat, it is more sensible to employ the enhanced B-dot algorithm over NMPC to detumble the spacecraft.

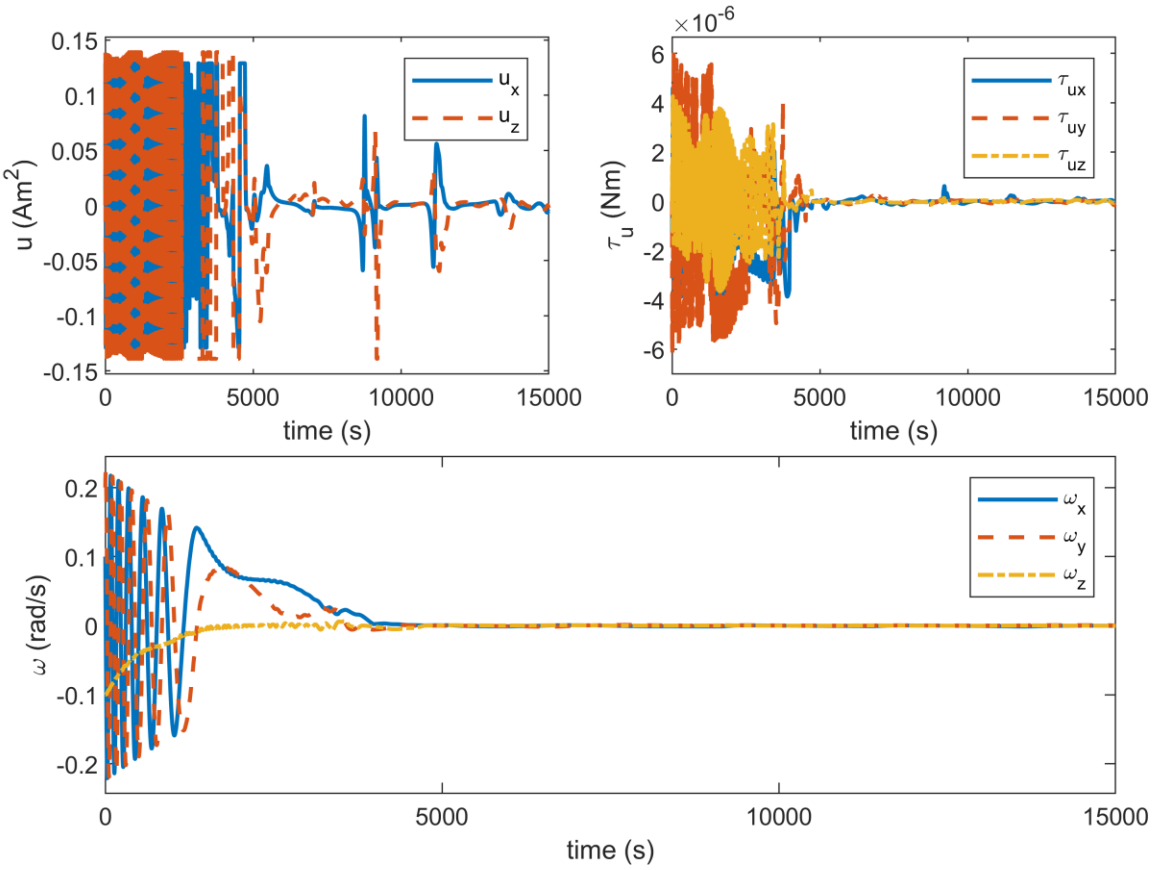

Figure 4.9 - Simulation results for detumbling spacecraft using NMPC and XZ magnetorquer actuation.

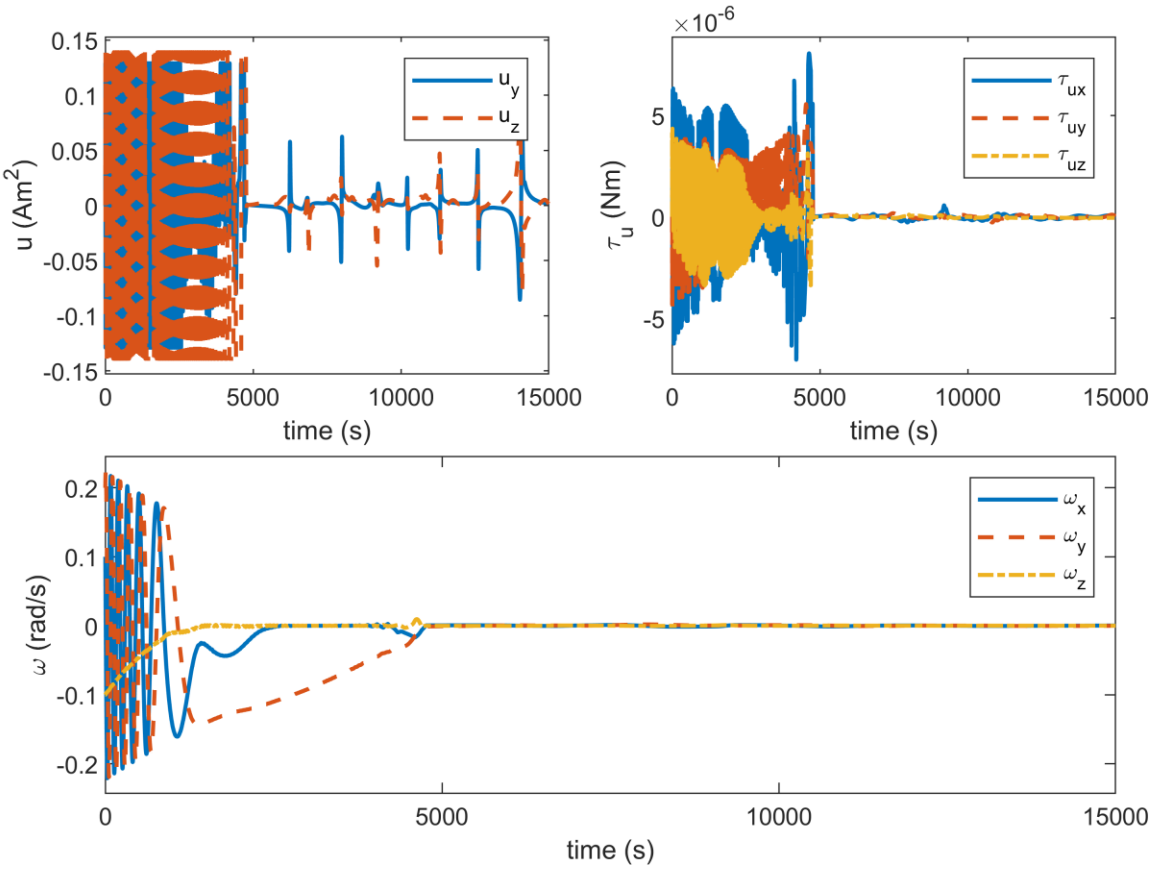

Figure 4.10 - Simulation results for detumbling spacecraft using NMPC and YZ magnetorquer actuation.

Chapter 5 ATTITUDE CONTROL USING REACTION WHEELS

Once detumbling has been successfully accomplished, the next phase of the ADCS is to perform an attitude correction maneuver to point itself towards a target. In this chapter, two attitude control algorithms are utilized to adjust the orientation of an underactuated spacecraft equipped with two reaction wheels placed along principal axes of the spacecraft. Since reaction wheels are momentum exchanging devices, the attitude control problem from equation (2.7) needs to be modified to,

$$\dot{\boldsymbol{\omega}} = \mathbf{J}^{-1}\left[-\boldsymbol{\omega}\times(\mathbf{J}\boldsymbol{\omega}+\boldsymbol{h}_{RW})+\boldsymbol{\tau}_g+\boldsymbol{\tau}_d+\boldsymbol{u}\right]$$
$$\boldsymbol{u} = -\dot{\boldsymbol{h}}_{RW} \tag{5.1}$$

where $\boldsymbol{h}_{RW}$ is the angular momentum vector of each reaction wheel in the system. The reaction wheels to be used on the ESSENCE 3U-CubeSat is used to perform this simulation. RW-0.003 reaction wheels from Sinclair Interplanetary are used to control the CubeSat's attitude. Some important parameters of the reaction wheel are listed on Table 5.1 obtained from [42].

Table 5.1 - Reaction wheel parameters for the RW-0.003.

Parameter		**Value**
Momentum	Nominal	0.003 Nms
	Peak	0.005 Nms
Maximum torque		±1 mNm
Reaction wheel MOI along spin axis		3.37×10^{-6} kg m^2
Reaction wheel saturation limit (nominal)		~ ±890 rad/s

The reaction wheels are placed orthogonal to each other along $\hat{b}_y$ and $\hat{b}_z$ axes of the CubeSat. Therefore, in the following sections, it is assumed that the spacecraft is unactuated along the $\hat{b}_x$ axis. The configuration of the reaction wheels is shown in Figure 5.1, with the reaction wheel highlighted in red is assumed to have either failed or does not exist by design.

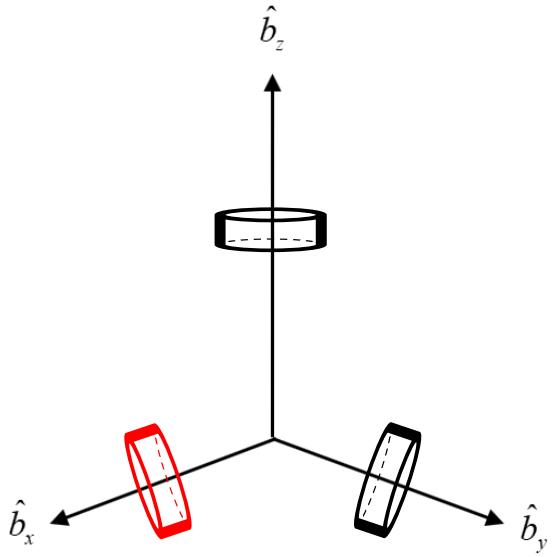

Figure 5.1 - Configuration of reaction wheels on the ESSENCE 3U-CubeSat.

Once again, following simulations are accomplished by using MATLAB-Simulink's Multibody Toolbox. A 3D model of each reaction wheel is modelled within Simulink to specifications. Then each reaction wheel is connected to CubeSat's principal axes using a rigid transformation module and a revolute joint module. The revolute joint provides one rotational degree of freedom, which allows each reaction wheel to rotate along the axis it

was placed in. Revolute joints also give the ability to control its motion via torque inputs, hence only torque control mode in each reaction wheel will be explored in following chapters. A schematic showing how MATLAB-Simulink is employed in the simulation of reaction wheel actuation is shown in Figure 5.2. Once again, this schematic does not represent the original code, rather a generalization of the basic concept of how the program is utilized.

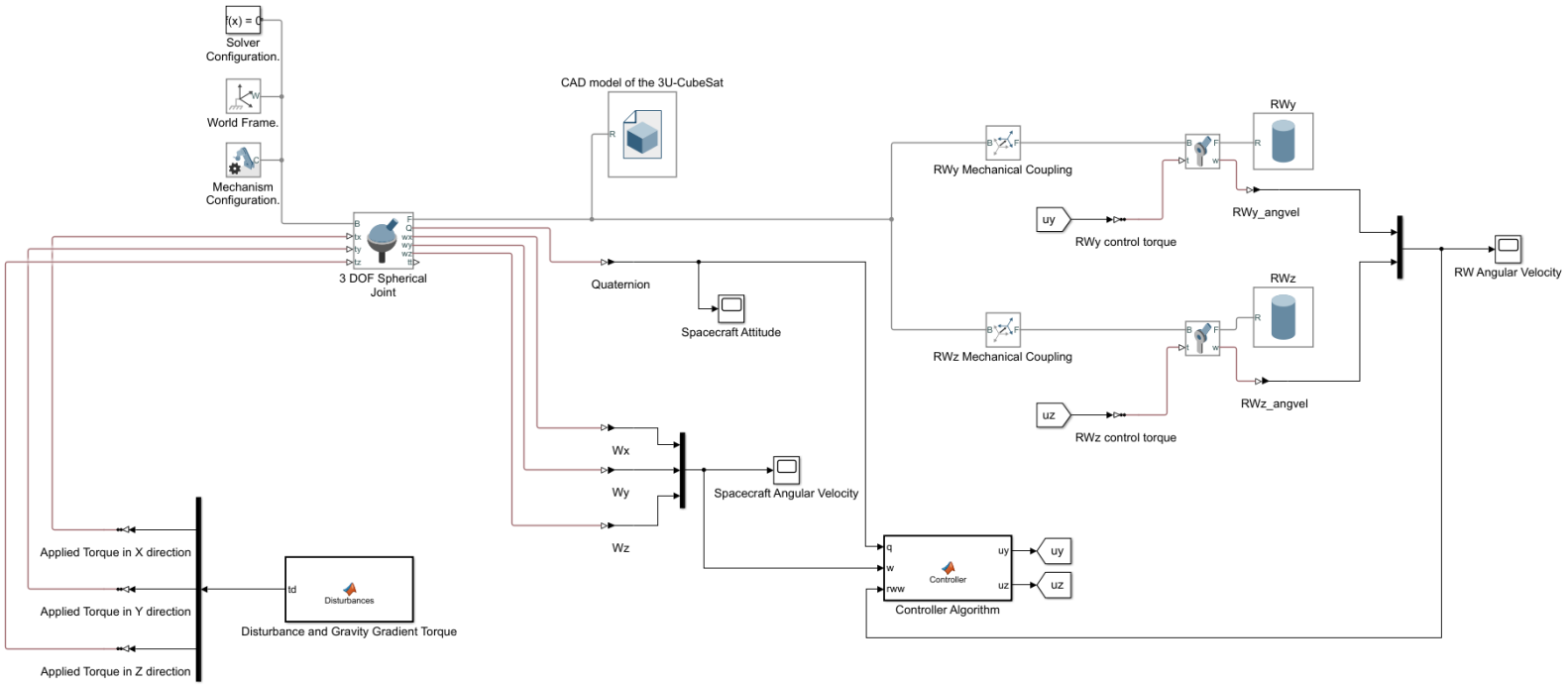

Figure 5.2 - Schematic outlining how MATLAB-Simulink is employed in the simulation of RW actuation.

5.1 Quaternion-Based Nonlinear Controller

In this section the use of quaternion-based nonlinear controller (QBNC) to perform an attitude correction maneuver using two reaction wheels is studied. This control algorithm proposed in [24] is a slight modification to the singular nonlinear controller (SNLC) proposed in [23] to achieve better time response. The control law is proposed as a speed controller for the reaction wheels in order to achieve the desired attitude. Suppose that the spacecraft system is unactuated along the $\hat{b}_x$, that is, only actuation along $\hat{b}_y$ and $\hat{b}_z$ is available. Then, the speed control laws according to [24] is given as,

$$\omega_{RWy} = -\gamma\epsilon_y + \sigma\frac{\epsilon_x\epsilon_z}{\epsilon_y^2+\epsilon_z^2} - k_D\frac{\dot{\epsilon}_x\epsilon_y}{\epsilon_y^2+\epsilon_z^2} \tag{5.2}$$

$$\omega_{RWz} = -\gamma\epsilon_z - \sigma\frac{\epsilon_x\epsilon_y}{\epsilon_y^2+\epsilon_z^2} - k_D\frac{\dot{\epsilon}_x\epsilon_z}{\epsilon_y^2+\epsilon_z^2} \tag{5.3}$$

This is a slight improvement from the speed control laws given in [23] since the new control laws contain terms regarding to the time derivative of the unactuated axis. To avoid the singularities of the proposed control laws, a saturated version of the control laws are shown in equations (5.4) and (5.5), where a_1 and a_2 are strictly positive constant saturation limits. The quaternions used here describe the orientation of $\mathcal{B}$ with respect to $\mathcal{I}$.

$$\omega_{RWy} = -\gamma\epsilon_y + \sigma\,\mathrm{sat}\left(\frac{\epsilon_x\epsilon_z}{\epsilon_y^2+\epsilon_z^2}, a_1\right) - k_D\,\mathrm{sat}\left(\frac{\dot{\epsilon}_x\epsilon_y}{\epsilon_y^2+\epsilon_z^2}, a_2\right) \tag{5.4}$$

$$\omega_{RWz} = -\gamma\epsilon_z - \sigma\,\mathrm{sat}\left(\frac{\epsilon_x\epsilon_y}{\epsilon_y^2+\epsilon_z^2}, a_2\right) - k_D\,\mathrm{sat}\left(\frac{\dot{\epsilon}_x\epsilon_z}{\epsilon_y^2+\epsilon_z^2}, a_1\right) \tag{5.5}$$

where γ and σ are controller gains, and k_D is defined as,

$$k_D = k_D^{'} \operatorname{sgn}(\epsilon_x) \operatorname{sgn}(\eta) \text{ and } k_D^{'} > 0 \tag{5.6}$$

The speed control laws defined in equations (5.4) and (5.5) are then converted to torque control laws using,

$$u_{QBNCi} = J_{ii}(-K_i(\omega_i - \omega_{RWi}) + \dot{\omega}_{RWi}), \quad i = y, z \tag{5.7}$$

where K_i is the controller gain, ω_i is the spacecraft angular velocity, and ω_{RWi} are obtained from equations (5.4) and (5.5). An in-depth analysis of the stability of this controller using Lyapunov method is outlined in [23].

The control law outlined in equation (5.7) is further modified to enhance its efficiency using inverse optimal control theory in [23]. For this controller, disturbances and gravity gradient torque is exempted from the dynamics and the system is redefined as,

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{h}) + g(\boldsymbol{x})\boldsymbol{u} \tag{5.8}$$

where,

$$\begin{aligned} \boldsymbol{x} &= [\boldsymbol{q}_{\mathrm{I}} \quad \boldsymbol{\omega}]^T & f(\boldsymbol{x}, \boldsymbol{h}) &= \left[\dot{\boldsymbol{q}}_{\mathrm{I}} \quad \dot{\boldsymbol{\omega}}\right]_{\boldsymbol{u}=\mathbf{0}_{3\times1}}^T \\ g(\boldsymbol{x}) &= \begin{bmatrix} \mathbf{0}_{4\times3} \\ \mathbf{J}_{3\times3}^{-1} \end{bmatrix} & \boldsymbol{u} &= [0 \quad N_y \quad N_z]^T \end{aligned} \tag{5.9}$$

Then, the optimal controller known as a minimum-norm controller is constructed as,

$$\boldsymbol{u}_{opt} = \begin{cases} \dfrac{(L_g V \boldsymbol{u}_{qbnc})(L_g V)^T}{L_g V (L_g V)^T} & \text{if } L_g V \boldsymbol{u}_{qbnc} < 0 \\ 0 & \text{if } L_g V \boldsymbol{u}_{qbnc} \geq 0 \end{cases} \tag{5.10}$$

where $L_g V$ is the Lie derivative derived from inverse optimal control theory. The inverse optimal control theory states that there exists a control input $\boldsymbol{u}$ satisfying the condition,

$$L_f V(\boldsymbol{x}, \boldsymbol{h}) + L_g V(\boldsymbol{x}) \boldsymbol{u}(\boldsymbol{x}) < 0 \tag{5.11}$$

Where,

$$\begin{aligned} L_f V &= \frac{\partial V^T}{\partial \boldsymbol{x}} f(\boldsymbol{x}, \boldsymbol{h}) \\ L_g V &= \frac{\partial V^T}{\partial \boldsymbol{x}} g(\boldsymbol{x}) \end{aligned} \tag{5.12}$$

Where V is the scalar control-Lyapunov function (CLF).

The CLF is derived in [23] as,

$$V = k(\boldsymbol{\epsilon}_1^T \boldsymbol{\epsilon}_1 + (1-\eta_1)^2) + \frac{1}{2}(\boldsymbol{\omega} - \boldsymbol{\omega}_{RW})^T (\boldsymbol{\omega} - \boldsymbol{\omega}_{RW}) \tag{5.13}$$

where $k > 0$. The partial derivative $\partial V^T / \partial \boldsymbol{x}$ is solved to be,

$$\frac{\partial V^T}{\partial \boldsymbol{x}} = \begin{bmatrix} 2k\epsilon_x \\ 2k\epsilon_y \\ 2k\epsilon_z \\ 4k\eta - 2k \\ \omega_x \\ \omega_y - \omega_{dy} \\ \omega_z - \omega_{dz} \end{bmatrix}^T \tag{5.14}$$

5.2 Simulation Results

The control algorithm outlined in the previous section is simulated using the parameters shown in Table 5.2. The simulation is conducted using only YZ actuation due to geometric symmetry of the CubeSat.

Table 5.2 - Simulation parameters for QBNC controller using two reaction wheels.

Parameter	Value
γ	0.02
σ	0.08
$k_D^{'}$	1
$K_y = K_z$	10
k	0.04
$\boldsymbol{\omega}(0)$	$\begin{bmatrix}0 & 0 & 0\end{bmatrix}^T$ rad/s
$\boldsymbol{q}_I(0)$	$\begin{bmatrix}0.2121 & 0.3536 & -0.2828 & 0.8660\end{bmatrix}^T$
$\max\left(\lvert\boldsymbol{u}\rvert\right)$	$1\times10^{-3}\cdot\begin{bmatrix}1 & 1 & 1\end{bmatrix}^T$ Nm

Figure 5.3 shows the simulation results for the minimum-norm QBNC controller. Although the simulation results look ideal, the environment in which the simulation was performed is not realistic. The simulation was performed under the zero-total angular momentum case in a disturbance free environment. In other words,

$$\boldsymbol{\omega}(0) = \boldsymbol{\omega}_{RW}(0) = [0 \quad 0 \quad 0]^T$$
$$\boldsymbol{\tau}_g = \boldsymbol{\tau}_d = [0 \quad 0 \quad 0]^T \tag{5.15}$$

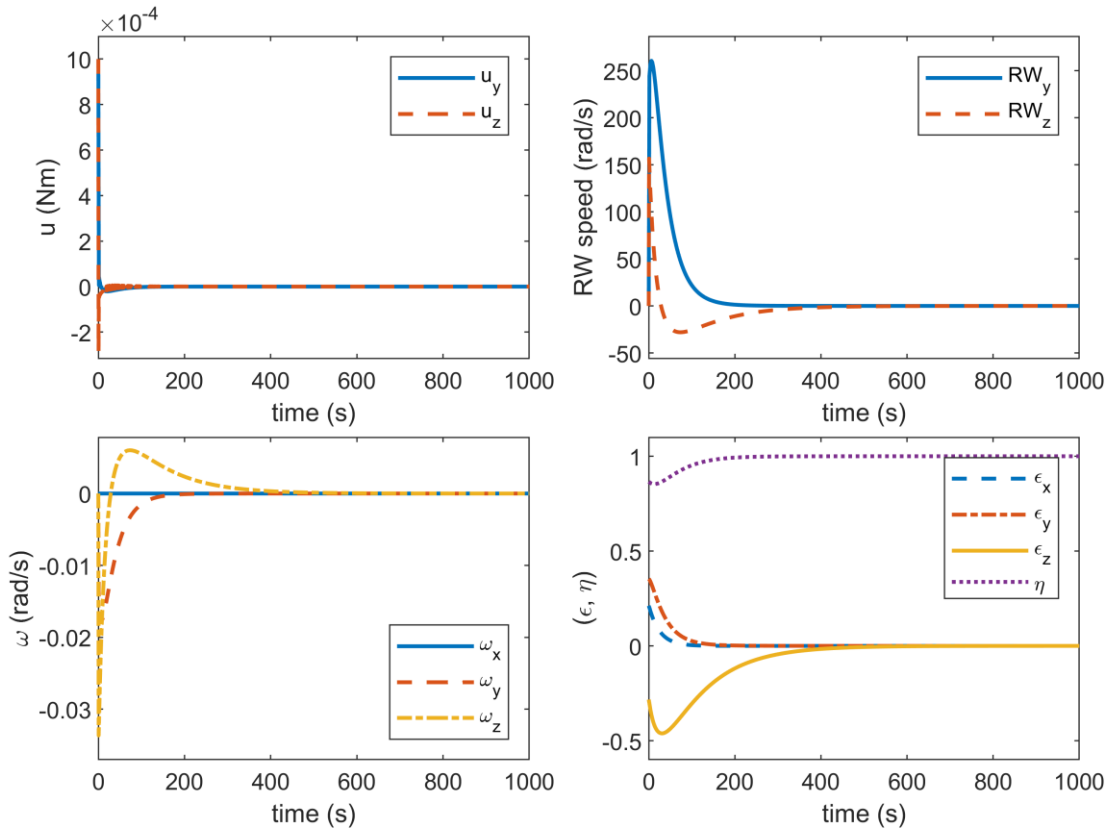

Figure 5.3 - Simulation results for minimum-norm QBNC using RW actuation.

The robustness of the controller is then investigated by performing the simulation with gravity gradient torque and disturbance torque applied. The simulation results for this scenario are shown in Figure 5.4. In the presence of disturbance torques, a lack of robustness can be observed from the QBNC controller. Furthermore, the angular velocity of the spacecraft seems to increase over time along with the reaction wheel speeds. This constant increase in reaction wheel speed will eventually result in the reaction wheel reaching its saturation limit, beyond which torques will no longer be generated in that

specific direction. In the reaction wheel speed plot, frequent zero-region crossing can also be observed, which can be harmful to the reaction wheels causing the reaction wheels to degrade over time. The attitude quaternions also exhibit transient oscillations with increasing frequency. The lack of robustness of this controller makes it difficult to implement in real-life situations.

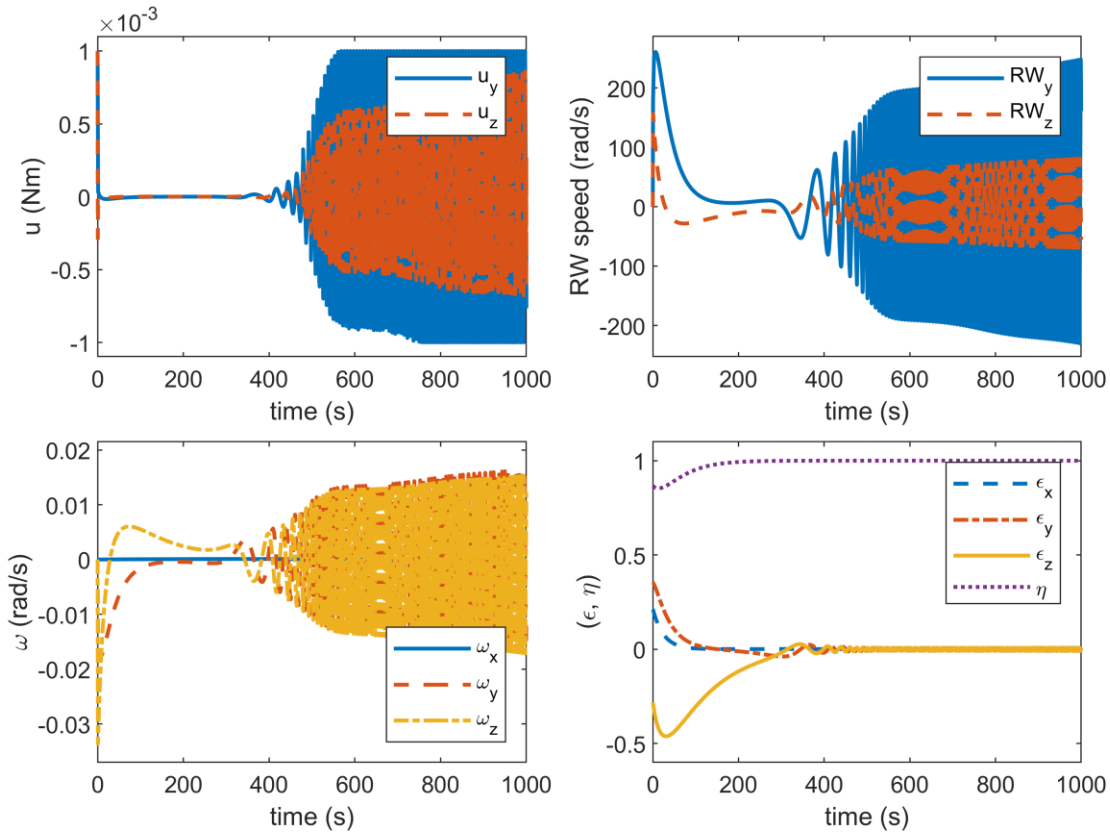

Figure 5.4 - Simulation results for minimum-norm QBNC using RW actuation subjected to disturbances.

5.3 Sub-optimal NMPC

This section explores the use of sub-optimal nonlinear model predictive control to perform an attitude adjustment maneuver using two reaction wheels placed along $\hat{b}_y$ and $\hat{b}_z$ axes of the spacecraft. Due to the robustness of MPC, it is expected to perform better than the controller presented in section Quaternion-Based Nonlinear Controller5.1 under the influence of external disturbances. The intention of utilizing a sub-optimal NMPC is to lower the computational load on the onboard computer of the spacecraft. The number of iterations considered by the optimizer to find a solution that minimizes the cost function has been restricted to 20. The cost function used in the modelling of the controller is identical to the cost function expressed in (3.11). The NMPC is designed as a torque-controller for the reaction wheels. The availability of state constraints is used to advantage to impose reaction wheel saturation limits. The system states can be described as,

$$\boldsymbol{x} = \begin{bmatrix} \omega_x & \omega_y & \omega_z & \epsilon_x & \epsilon_y & \epsilon_z & \eta & \omega_{RWx} & \omega_{RWy} & \omega_{RWz} \end{bmatrix} \tag{5.16}$$

Hence, the reaction wheel saturation limits are imposed as hard constraints,

$$\frac{x_{j,min}(i)}{s_j^x} \leq \frac{x_j(k+i\,|\,k)}{s_j^x} \leq \frac{x_{j,max}(i)}{s_j^x}$$
$$i = 1,2,3,\ldots,p \qquad j = 8,9,10 \tag{5.17}$$

The output variables of the system are the states to be controlled towards a desired target value. These states are identified as the first seven states disregarding the reaction wheel

speeds from (5.16). The manipulated variables of the system are the control torque inputs applied to each reaction wheel given as,

$$\boldsymbol{u} = \begin{bmatrix} u_y & u_z \end{bmatrix} \tag{5.18}$$

5.4 Simulation Results

The parameters used in the simulation are shown in Table 5.3. The system is subjected to gravity gradient torque and disturbance expressed in (2.14) and (2.15). When selecting the weights for the output variables, larger weights were imposed on ω_z and ϵ_x. The reasoning for using a larger weight for ω_z is that ω_x and ω_y cannot influence ω_z as discussed in section 2.3. Therefore, using a larger weight ensures that ω_z is converged to zero at a higher priority. The unactuated state ϵ_x is also given a larger weight since actuation along $\hat{b}_x$ axis is absent. The simulation results for the attitude correction maneuver are shown in Figure 5.5.

Table 5.3- Simulation parameters for NMPC with two RWs.

Parameter	Value
Number of OVs (n_y)	7
Number of MVs (n_u)	2
Number of states (n_x)	10
Prediction horizon (p)	25 s
Control horizon (c)	5 s
Simulation time (T)	200 s
Time step (t_s)	1 s

Simulation solver time step (t_{ss})	variable
OV weights ($w_{i,j}^{y}$)	[1 1 3 8 1 1 1] $i=1,2,3,\ldots,p$
OV scale factors (s_j^{y})	[1 1 1 1 1 1 1]
OV reference ($r_j(k+i\,|\,k)$)	[0 0 0 0 0 0 1] $i=1,2,3,\ldots,p$
MV rate weights ($w_{i,j}^{\Delta u}$)	[1 1] $i=1,2,3,\ldots,p$
u_y constraints ($[u_{1,min} \quad u_{1,max}]$)	$1\times10^{-3}\cdot[-1 \quad 1]$ Nm
u_z constraints ($[u_{2,min} \quad u_{2,max}]$)	$1\times10^{-3}\cdot[-1 \quad 1]$ Nm
Initial angular velocity ($\boldsymbol{\omega}(0)$)	$[0 \quad 0 \quad 0]^T$ rad/s
Initial attitude ($\boldsymbol{q}_I(0)$)	$[0.2121 \quad 0.3535 \quad -0.2828 \quad 0.8660]^T$

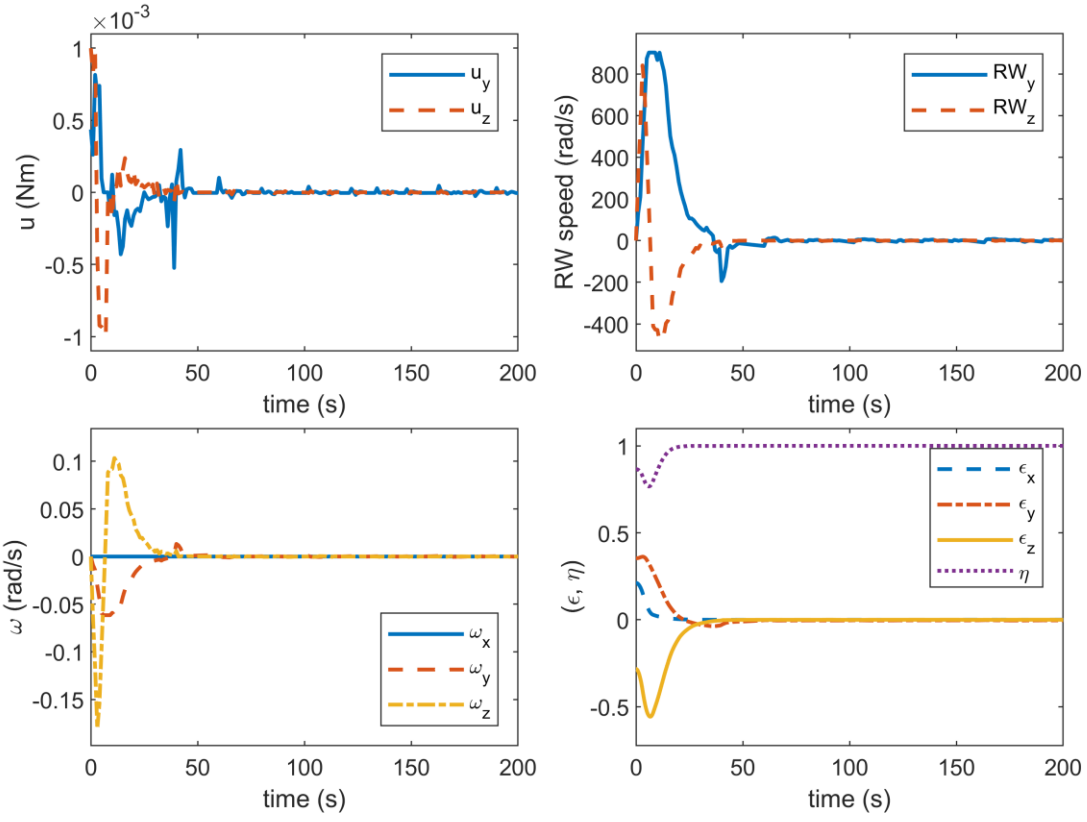

Figure 5.5 - Simulation results for sub-optimal NMPC using RW actuation.

It is evident from the simulation results that NMPC was able to carry out the attitude correction maneuver without difficulty. In the control torque plot, the controller can be seen to apply small torques over time to counteract the disturbance forces. It should also be noted that settling time of NMPC is also much smaller compared to the settling time of QBNC in the disturbance free environment. It can also be concluded that this settling time could be further reduced if an optimal NMPC is utilized instead of a sub-optimal NMPC. Therefore, in the case of ESSENCE CubeSat, it is highly recommended that if the computational power allows, NMPC is utilized when performing attitude control maneuvers using two reaction wheels.

Chapter 6 EXPERIMENT

In previous chapters, the superior performance of NMPC compared to other control algorithms, in the sense of settling times and robustness, was established. This chapter explores the use of the NMPC controller developed in previous chapters on a physical satellite simulator. The satellite simulator setup simulates the dynamics of a spacecraft as closely as possible on ground. The experiment employs a large granite table along with an air-bearing Spacecraft Simulator Module (SSM) that contains various sensors and actuators commonly found on spacecraft. The satellite simulator module and the granite table were originally developed by Tsinghua University. Further modifications and improvements were made to the experiment setup by various students over the years at York University. Vising professor Ning Chen, PhD student Sat Li, MSc students Joshua Cookson and Lucas Santaguida, all of whom contributed various modifications and improvements.

6.1 Experiment Setup

6.1.1 Granite Table

A granite table is used to hold the SSM in place on a two-dimensional plane. A major drawback of this setup is that the motion of the spacecraft is strictly planar, hence reducing the motion to three DOF. Motion along the x and y directions provide two translational DOF, while rotation about the z axis provide one rotational motion. Translational motion along z axis and rotational motion about x and y axis are restricted. The 4m × 2m × 0.5m

table is lifted using three supports and four legs. The four legs can be adjusted to level the table. Ideally when the air-bearings are turned on, the SSM should levitate just above the granite table fixed in place motionless. However, imperfections on the surface of the granite table, accumulated dust particles, and imperfections in levelling, all have contributed to introduction of unwanted motion to the SSM. Due to the settling of the floor and other imperfections, the table has a 0.2° inclination. This introduces an acceleration of 3.08 mm/s^2 to the SSM [43]. An inertial right-handed coordinate frame denoted $\mathcal{W}$ is fixed to the granite table as shown in Figure 6.1. The (0,0) point is located at the bottom left corner when viewed from top. A rotation in the CCW direction about z-axis is taken as positive.

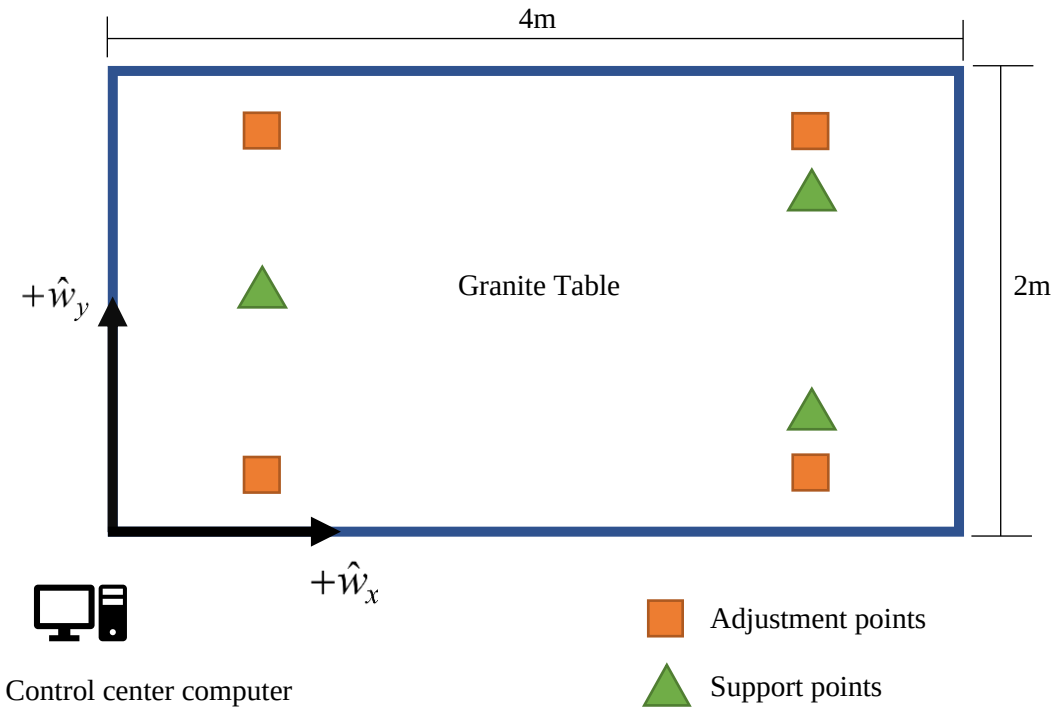

Figure 6.1 - Top-down look of the granite table setup and its coordinate frame $\mathcal{W}$.

6.1.2 Satellite Simulator Module (SSM)

The SSM is an autonomous system composed of actuators, sensors, a battery, and on board-computer that mimic a physical satellite. Three air-bearings mounted at the bottom of the SSM provides near frictionless motion as it allows the SSM to traverse the granite table with 3 DOF. The SSM has dimensions 0.37m×0.37m×0.28m, and a mass of 18kg. This results in a moment of inertia matrix of,

$$\boldsymbol{J} = \begin{bmatrix} 0.32295 & 0 & 0 \\ 0 & 0.32295 & 0 \\ 0 & 0 & 0.4107 \end{bmatrix} kg \cdot m^2 \tag{5.19}$$

The body-fixed coordinate frame of the SSM denoted $\mathcal{B}^{\text{SSM}}$ is shown in Figure 6.2. The components that make up the SSM along with each face are shown in Figure 6.3.

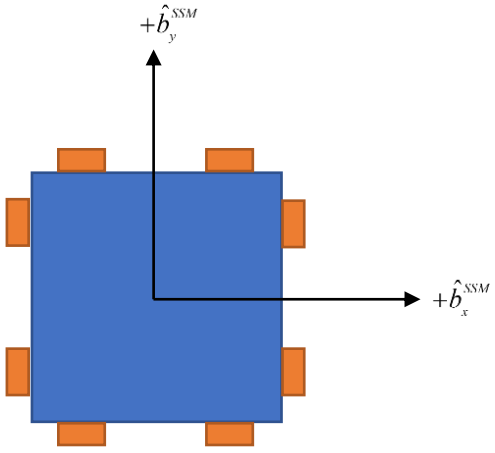

Figure 6.2 - Body-fixed frame of the SSM as viewed from top.

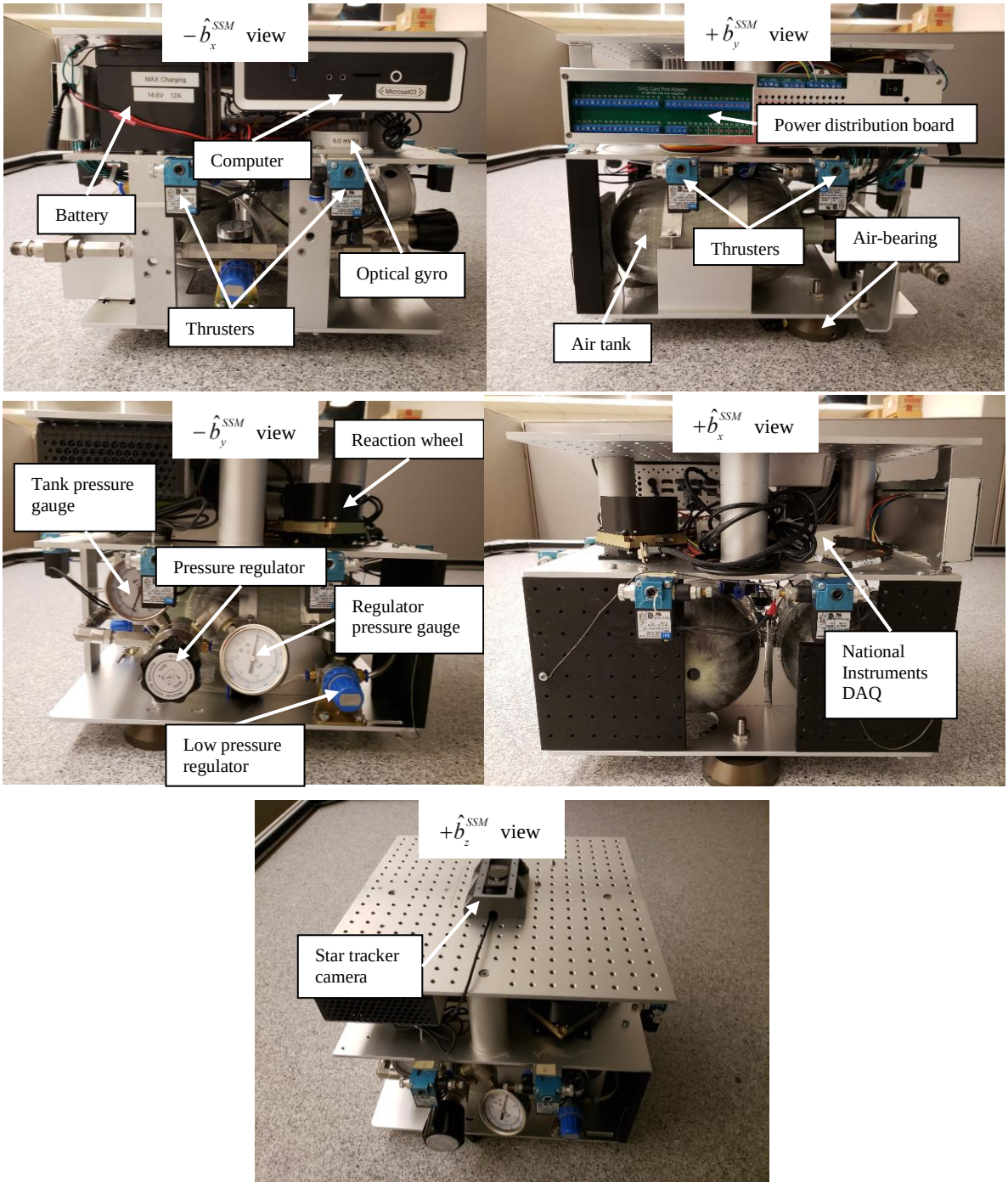

Figure 6.3 - Main components of the SSM.

Electrical components on the SSM are powered using a 12V $LiFePO_4$ battery along with a power distribution board (PDB). The power distribution board supplies 5V, 12V and 24V outputs. The OBC located on the SSM is used to interface with onboard sensors and actuators using LabVIEW since the computer runs on Windows 10 operating system. Additionally, the OBC is equipped with an i7-8550U processor, 16 GB of RAM, and a solid-state drive. The OBC has no fans or any other moving components such as a traditional hard-disk drive to eliminate unwanted moments in the system.

The SSM consists of an optical gyro and a star tracker interfaced via LabVIEW to obtain system state information. The optical gyro provides angular velocities, while the star tracker provides absolute position and orientation of the SSM. The optical gyro is connected to the

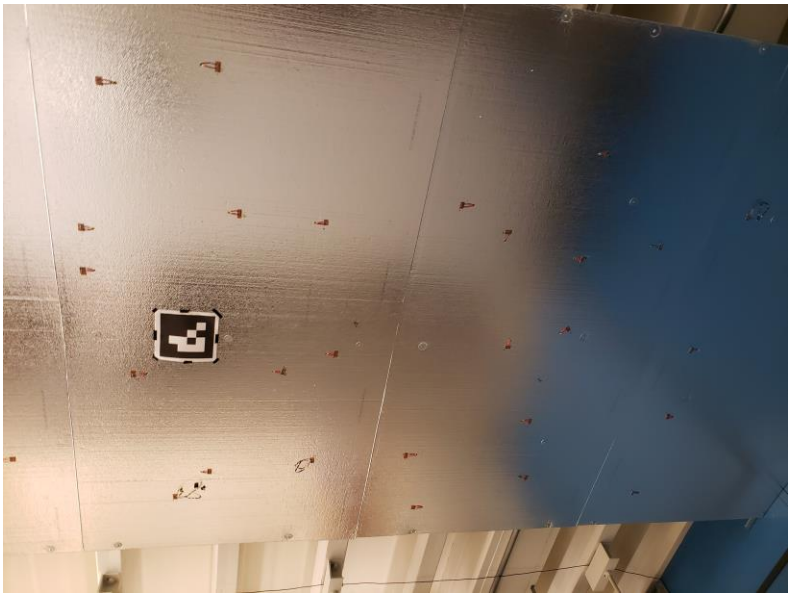

Figure 6.4 - IR star pattern used for position and attitude tracking of SSM.

OBC via a National Instruments NI USB-6212 Data Acquisition Card. LabVIEW code developed by a previous master's student is used to obtain the gyro readings as an angular velocity in rad/s. The star tracker utilizes an infrared (IR) camera mounted on top of the SSM, and IR LED lights that simulate star patterns fixated to the ceiling as seen on Figure 6.4. The stars are placed such that at least five stars are visible to the star tracker camera from any point on the granite table. A catalog of each star and their geometric relationships to four closest neighbouring stars is used to obtain position and orientation of the SSM. The star tracker system allows the SSM to resolve its position and attitude without the need of an external tracking system, much like a real spacecraft [43] [44].

The SSM is also equipped with 8 unidirectional air thrusters and one reaction wheel for actuation. Two thrusters are located on each side of the SSM providing thrust in opposite directions. To simplify the problem, the 8 uni-directional thrusters are combined to create 4 bidirectional thrusters as shown in Figure 6.5. Thrust forces are generated using

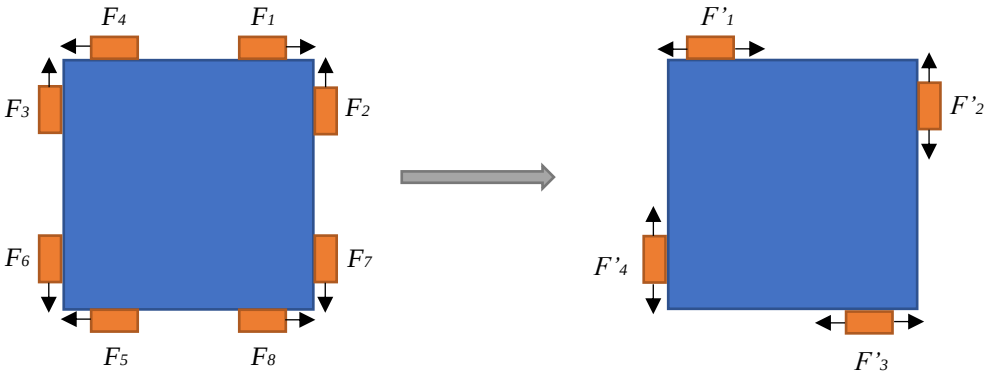

Figure 6.5 - SSM thruster location and grouping.

compressed air stored in air tanks located on the SSM. Each thruster produces 0.065N force in each direction when turned on. The air tanks can store 4L of air at a maximum pressure of 20 MPa. Air is distributed from the tanks using tubes to the 8 thrusters and three air bearing feet that float the SSM above the granite table.

A new reaction wheel—VECTRONIC Aerospace VRW, was installed to replace the old reaction wheel as part of this thesis. An MDM9 connector is used to connect the reaction wheel to a USB port of the OBC using RS422 interface. A LabVIEW code was also developed to communicate with the reaction wheel using HEX inputs as described on the Interface Control Document (ICD) provided by the manufacturer. The moment of inertia of the rotor about the spin axis is 0.000342 kgm^2. The reaction wheel has a maximum angular momentum of 0.21 Nms, providing a RW saturation speed of 628.3 rad/s [45].

6.1.3 Communication

A control center is utilized to communicate with the SSM. A computer located at the control center has wireless remote access to the SSM's OBC over WiFi. This allows the direct control of the OBC and its associated components. As previously mentioned, LabVIEW run on the OBC is used to interface with the sensors and actuators to control the system. A schematic of the power and data connections of the system are shown in Figure 6.6.

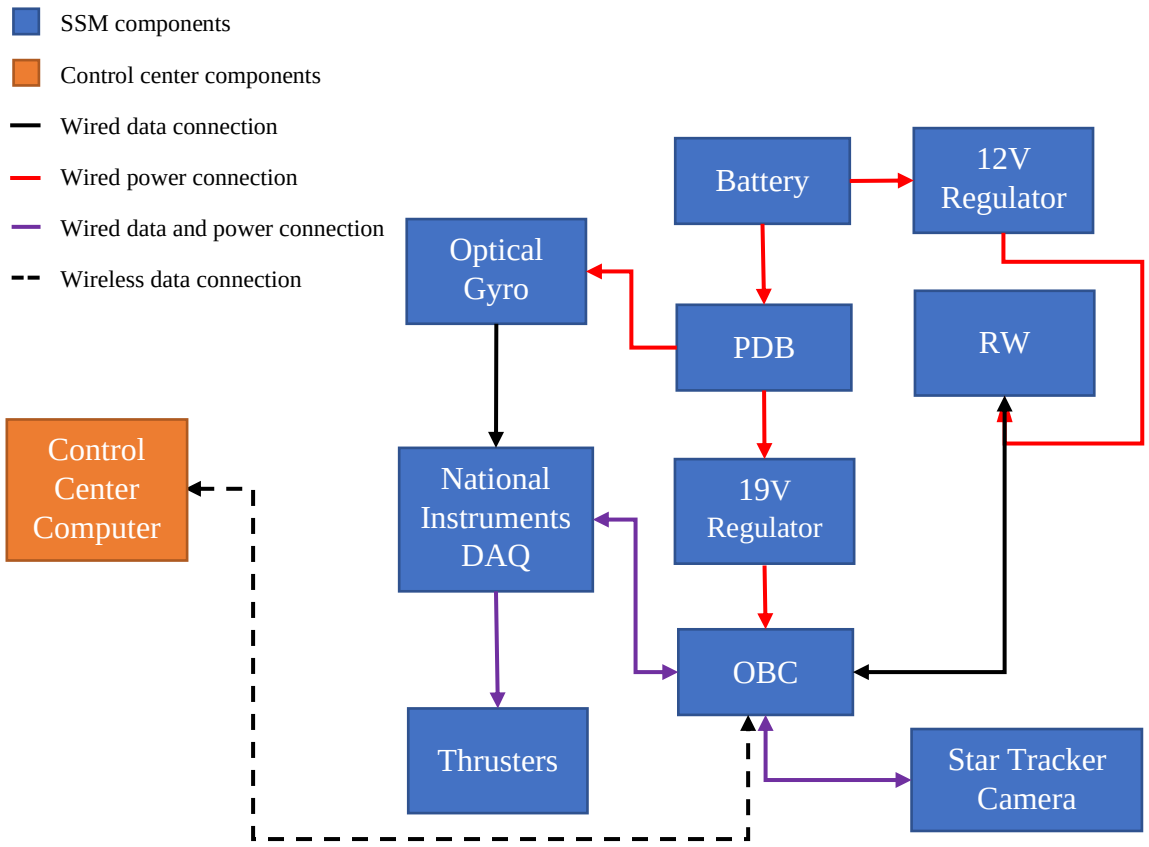

Figure 6.6 - Power and data connection schematic.

6.2 SSM Detumbling using NMPC with Air-thrusters

This section explores the use of NMPC to detumble the SSM using its air-thrusters. The torque generated by each thruster can be evaluated using SSM's physical dimensions and thrust produced by each thruster of 0.065N. The torque experienced by the SSM is calculated as,

$$\tau_i = r_i \times F_i^{'} \qquad i = 1,2,3,4 \qquad (5.20)$$

Considering the dimensions of the SSM, this gives a torque of

$$\tau_i = u_{zi} = \pm 0.0120\,\text{Nm} \qquad i = 1,2,3,4 \tag{5.21}$$

The torque calculated in equation (5.21) is imposed as hard constraints within NMPC using equation (3.20). However, The NMPC controller needs to be modified since the thrusters on the SSM exhibit on/off behaviour rather than variable thrust. Due to the grouping of thrusters shown in Figure 6.5, state of each thruster can be one of three -1, 0 or 1 representing -0.120 Nm, 0 Nm, +0.0120 Nm respectively. The control input calculated by the NMPC controller is forced to one of these states before it is applied to the system. The thrust range is divided equally in to 3 sections as shown.

Forced system input:	-0.0120 Nm	0 Nm	0.0120 Nm
NMPC control input:	[-0.0120, -0.0040) Nm	[-0.0040, 0.0040] Nm	(0.0040, 0.0120] Nm

Figure 6.7 - Control input thrust range division for system input.

For instance, if the control input calculated by NMPC is greater than or equal to -0.0120 N and less than -0.0040 N, the force input to the system is -0.0120 N. Due to MPC's robustness and its ability to predict the future, this method seemed to work well. The schematic of this control technique is shown in Figure 6.8.

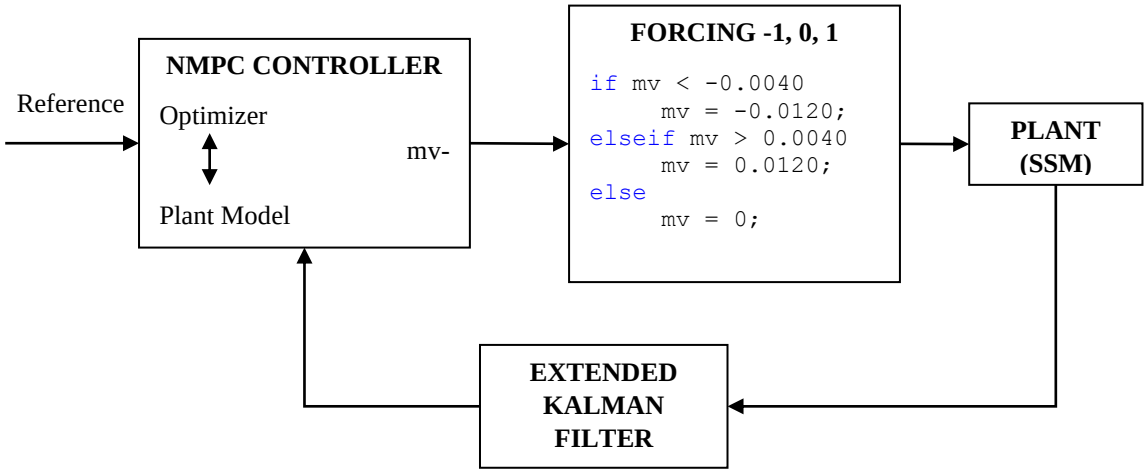

Figure 6.8 - NMPC controller schematic for SSM thruster control.

Restriction in DOF presented by planar motion of the SSM gives rise to a reduced attitude dynamics equation. This reduces the attitude dynamics in equation (2.7) to a linear system given by,

$$\dot{\omega}_z = \frac{\tau_z}{J_{zz}} = \frac{\tau_{dz} + u_z}{J_{zz}} \tag{5.22}$$

However, the complete non-linear attitude dynamics equation along with nonlinear MPC will be utilized in the experiment as to develop a general case of the control problem, and to study the performance of the controller. The disturbance torques in the system are caused by imperfections in the granite table as discussed in section 6.1.1. The states of the system are identified as,

$$\boldsymbol{x} = [\omega_x \quad \omega_y \quad \omega_z \quad \epsilon_x \quad \epsilon_y \quad \epsilon_z \quad \eta] \tag{5.23}$$

Although a reduced number of states can be used, knowing the motion of the system is planar, the full system states are used as not to change the originally developed code and to study the performance of the controller with a higher computational load. Furthermore, the manipulated variables of the system are the four thruster inputs which can be expressed as,

$$\dot{\omega}_z = \frac{\tau_z}{J_{zz}} = \frac{\tau_{dz} + \sum_{i=1}^{4} u_{zi}}{J_{zz}} \tag{5.24}$$

Since the purpose of this controller is to detumble the SSM, output variables of the system are identified as,

$$\boldsymbol{y} = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix} \tag{5.25}$$

6.3 Simulation Results

The parameters used in the experiment are shown in Table 6.1. Initially, the SSM is set to float by activating its air-bearing feet. Then a random initial angular velocity is induced on the SSM by manually spinning it. While the SSM is tumbling, the controller is activated. As previously mentioned, LabVIEW is used to interface with the SSM's actuators and sensors. However, NMPC controller script is developed on MATLAB, hence the MATLAB script is run within LabVIEW. LabVIEW provides current system states, angular velocity, and attitude obtained via its sensors to the MATLAB script. MATLAB script then outputs control inputs computed by NMPC, which is fed into the thrusters through LabVIEW. This process then repeats itself with each loop iteration executed at

every one second. This relatively small frequency was needed to ensure that MATLAB script had enough time to compute accurate control inputs. At higher frequencies, the system behaved unexpectedly, due to lack of control inputs from the MATLAB script.

Table 6.1 - Simulation parameters for detumbling SSM using NMPC and air-thrusters.

Parameter	**Value**
Number of OVs (n_y)	3
Number of MVs (n_u)	4
Number of states (n_x)	7
Prediction horizon (p)	8 s
Control horizon (c)	1 s
Simulation time (T)	variable
Time step (t_s)	1 s
OV weights ($w_{i,j}^{y}$)	$[1 \quad 1 \quad 20]$ $i = 1, 2, 3, \ldots, p$
OV scale factors (s_j^{y})	$[1 \quad 1 \quad 1]$
OV reference ($r_j(k+i \mid k)$)	$[0 \quad 0 \quad 0]$ $i = 1, 2, 3, \ldots, p$
MV rate weights ($w_{i,j}^{\Delta u}$)	$[1 \quad 1 \quad 1 \quad 1]$ $i = 1, 2, 3, \ldots, p$
u_{zi} constraints ($[u_{zi,min} \quad u_{zi,max}]$)	$1 \times 10^{-2} \cdot [-1.2 \quad 1.2] Nm$ $i = 1, 2, 3, 4$
Initial angular velocity ($\boldsymbol{\omega}(0)$)	variable
Initial attitude ($\boldsymbol{q}_1(0)$)	variable

The simulation results for an experiment attempt is shown in Figure 6.9. The initial angular velocity is recorded to be -1.2680 rad/s. It is evident that from the plot that NMPC was able to detumble the SSM to near zero angular velocity. It can also be observed that the thrusters work independently of each other. Thruster 4 can be seen to apply the final thrust required to slow down the SSM without overshooting.

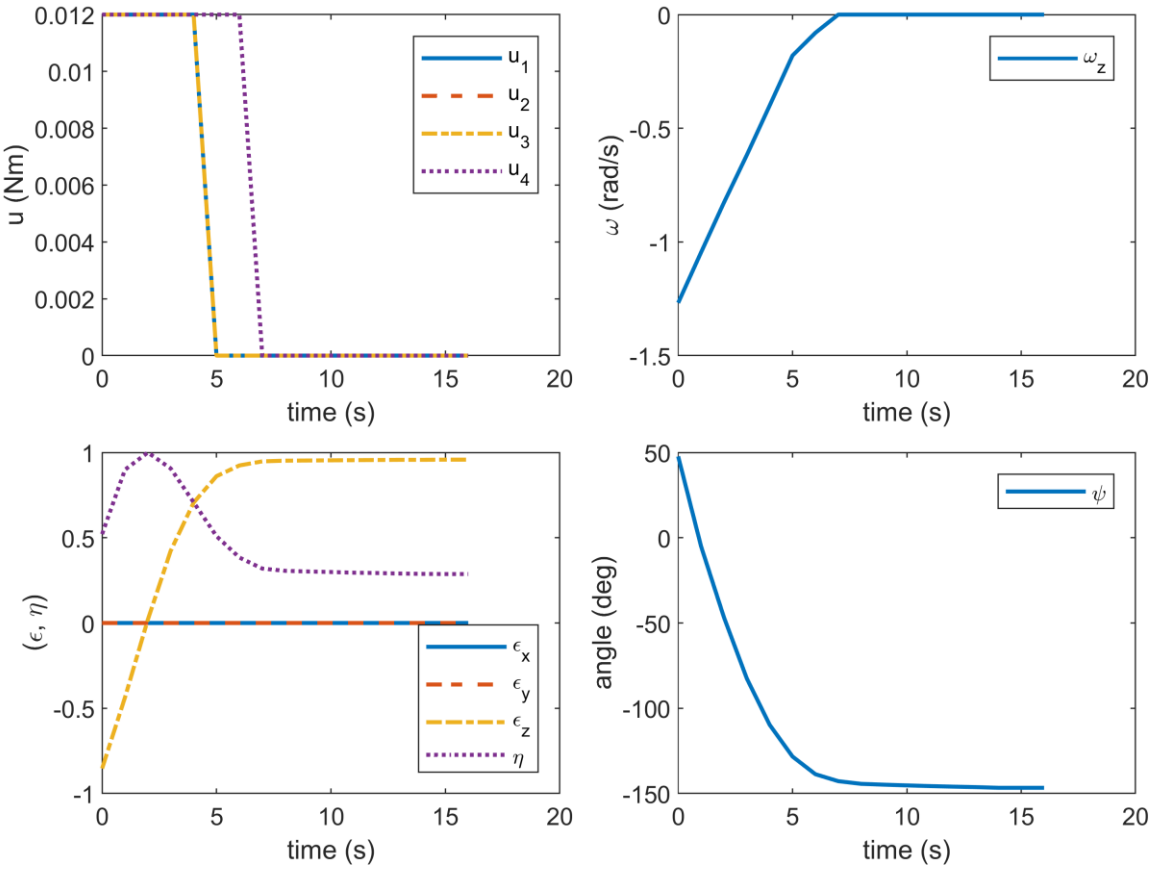

Figure 6.9 - Simulation results for detumbling SSM using NMPC and air-thrusters.

In total, four attempts were performed to study the reliability of the controller with varying initial angular velocities. The simulation results for the experiments are shown in Figure 6.10. It can be seen that except for experiment 3, the rest had overall satisfactory performance. From the plot it can be seen that in experiment 3 the controller is unable to drive the remaining angular velocity to zero and stays constant over time. The reason for this phenomenon can be traced back to the thruster mapping system explained in Figure 6.7 and Figure 6.8. A thruster 'dead-zone' is introduced in the mapping between -0.0040 N and 0.0040N. Which means that if the required thrust control input is between these two values

the controller does not produce any thrust. This can be concluded as the reason for the unexpected performance in experiment 3. To avoid this, a smaller dead-zone in the mapping process or techniques such as pulse width modulation can be utilized.

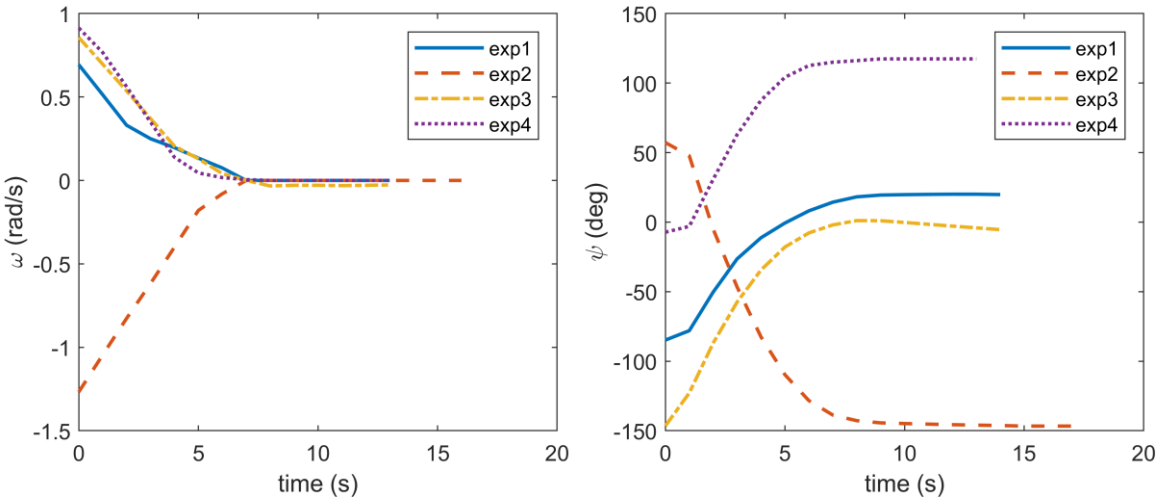

Figure 6.10 - Simulation results for NMPC detumbling experiments using air thrusters.

6.4 SSM Attitude Control using NMPC with Reaction Wheel

The use of NMPC to perform an attitude correction maneuver using the RW onboard the SSM is studied in this section. The reaction wheel onboard is the VECTRONIC Aerospace VRW. Some important parameters of the RW are shown in Table 6.2.

Table 6.2 - Reaction wheel parameters of the VECTRONIC Aerospace VRW.

Parameter		**Value**
Length × width × height		70 mm × 70 mm × 48 mm
Mass		1.0 kg
Momentum	@ 523.6 rad/s	0.18 Nms
	@628.3 rad/s	0.21 Nms

Maximum torque	±10 mNm
Reaction wheel rotor MOI along spin axis	3.42×10^{-4} kg m^2
Reaction wheel saturation limit	±628.3 rad/s

The attitude of the SSM is considered as the attitude of body-fixed frame $\mathcal{B}^{SSM}$ with respect to granite table frame $\mathcal{W}$. Without loss of generality, it is considered that $\mathcal{W}$ is the desired attitude that the SSM wishes to attain. Therefore, the objective of this experiment is to perform an attitude correction maneuver to align the $\mathcal{B}^{SSM}$ frame with $\mathcal{W}$ frame starting from a random initial attitude. The attitude is computed using Euler angles roll (φ), pitch (θ), and yaw (ψ), which describes rotations about x, y, and z axes, respectively. The angle from $\hat{w}_x$ axis to $\hat{b}_x^{SSM}$ axis is described by yaw angle with CCW rotation taken as positive as seen on Figure 6.11.

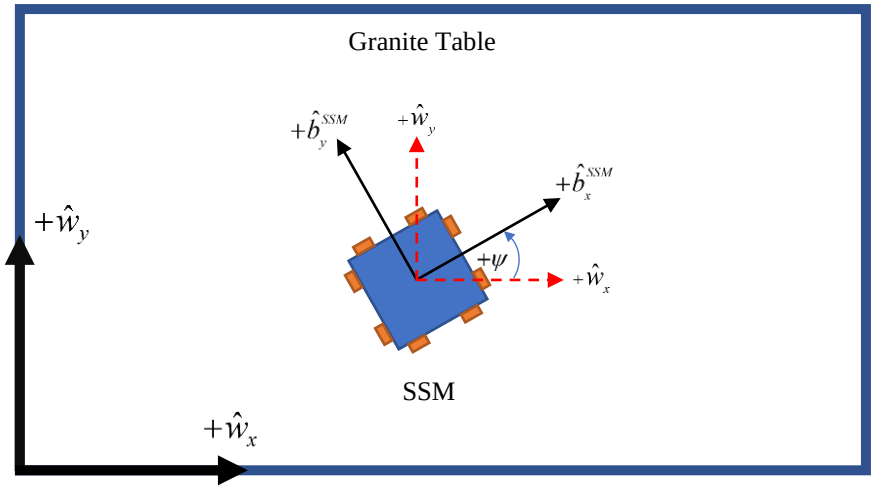

Figure 6.11 - Schematic of frame $\mathcal{B}^{SSM}$ with respect to frame $\mathcal{W}$.

The RW is situated on the SSM such that its spin axis is parallel to the $+\hat{b}_z^{SSM}$. This allows the RW to apply a torque about the $+\hat{b}_z^{SSM}$ axis via momentum transfer. The torque applied is in the opposite direction of the RW spin. If the RW accelerates in the CCW direction, a CW torque is experienced by the SSM and vice versa. Since the originally developed NMPC code runs using quaternions, a simple conversion from Euler angles to quaternions is performed before control inputs are calculated [46]. This conversion is performed as,

$$\begin{aligned} \eta &= ECA + FDB \\ \epsilon_x &= FCA - EDB \\ \epsilon_y &= EDA + FCB \\ \epsilon_z &= ECB - FDA \end{aligned} \tag{5.26}$$

where,

$$\begin{aligned} A &= \cos(0.5\psi) \\ B &= \sin(0.5\psi) \\ C &= \cos(0.5\theta) \\ D &= \sin(0.5\theta) \\ E &= \cos(0.5\phi) \\ F &= \sin(0.5\phi) \end{aligned} \tag{5.27}$$

Since the motion of the SSM is planar, a reduced set of attitude dynamics give rise to the linear dynamics,

$$\begin{aligned} \dot{\omega}_z &= \frac{\omega_y(J_{xx}\omega_x + h_{RWx}) - \omega_x(J_{yy}\omega_y + h_{RWy}) - \dot{h}_{RWz} + \tau_d}{J_{zz}} \\ \dot{\omega}_z &= \frac{-\dot{h}_{RWz} + \tau_d}{J_{zz}} = \frac{u + \tau_d}{J_{zz}} \end{aligned} \tag{5.28}$$

However, the complete non-linear attitude dynamics equation along with nonlinear MPC will be utilized as to develop a general case of the control problem, and to study the performance of the controller.

When applying NMPC to perform the attitude maneuver, maximum torque and reaction wheel saturation speeds are imposed as hard constraints. Although the RW can have a maximum rate of 628.3 rad/s, the speed it limited using NMPC constraint to 600 rad/s as not to over stress the RW. In total, ten system states are identified as,

$$\boldsymbol{x} = [\omega_x \quad \omega_y \quad \omega_z \quad \epsilon_x \quad \epsilon_y \quad \epsilon_z \quad \eta \quad \omega_{RWx} \quad \omega_{RWy} \quad \omega_{RWz}] \tag{5.29}$$

System states ω_{RWx} and ω_{RWy} are zero, since there are no reaction wheels mounted along $\hat{b}_x^{SSM}$ and $\hat{b}_y^{SSM}$ axes. State ω_{RWz} is resolved by solving the differential equation,

$$\dot{\omega}_{RWz} = \frac{\dot{h}_{RWz}}{J_{RWz}} = \frac{-u}{J_{RWz}} \tag{5.30}$$

The reaction wheels are controlled using torque control rather than speed control. Therefore, the manipulated variables of the system are the torque inputs to the reaction wheel. The system output variables are identified as the first seven states in equation (5.29) as,

$$\boldsymbol{y} = [\omega_x \quad \omega_y \quad \omega_z \quad \epsilon_x \quad \epsilon_y \quad \epsilon_z \quad \eta] \tag{5.31}$$

6.5 Simulation Results

The parameters used in the experiment of attitude control maneuver using NMPC are shown in Table 6.3. The SSM is initially at rest pointed in a direction with an arbitrary yaw angle as shown in Figure 6.11. The LabVIEW code is then executed which turns on the air-bearing feet and the controller simultaneously. Similar to the previous section, the NMPC controller is developed using MATLAB. Angular velocity and attitude angle are obtained via LabVIEW through the optical gyro and star tracker, respectively. This feedback information is sent to the MATLAB code. NMPC then computes the control torques necessary to reduce the yaw angle to zero. LabVIEW obtains the computed control torque from MATLAB and converts into a HEX command which is then sent to the RW.

Table 6.3 - Simulation parameters for attitude change maneuver using NMPC and RW.

Parameter	Value
Number of OVs (n_y)	7
Number of MVs (n_u)	3
Number of states (n_x)	10
Prediction horizon (p)	8 s
Control horizon (c)	1 s
Simulation time (T)	variable
Time step (t_s)	1 s
OV weights ($w_{i,j}^{y}$)	[1 1 3 1 1 20 20] $i = 1, 2, 3, ..., p$
OV scale factors (s_j^{y})	[1 1 1]
OV reference ($r_j(k+i \mid k)$)	[0 0 0 0 0 0 1] $i = 1, 2, 3, ..., p$
MV rate weights ($w_{i,j}^{\Delta u}$)	[1 1 1] $i = 1, 2, 3, ..., p$
u_z constraints ($[u_{z,min} \quad u_{z,max}]$)	$1 \times 10^{-2} \cdot [-1.0 \quad 1.0]$ Nm

ω_{RWz} constraints ($[\omega_{RWz,min} \quad \omega_{RWx,max}]$	[–600 600] rad/s
Initial angular velocity ($\boldsymbol{\omega}(0)$)	variable
Initial attitude ($\boldsymbol{q}_I(0)$)	variable

The simulation results for one experiment are shown in Figure 6.12. In this experiment the initial yaw angle was ~178°, which corresponds to a quaternion of (0, 0, 1, 0.17). A small overshooting can be observed as ϵ_z goes past zero axis. However, the angular velocity remains at a constant zero, which indicates that the optical gyro was not able to capture the small angular velocity that is clearly present in the SSM due to lack of sensitivity. The poor

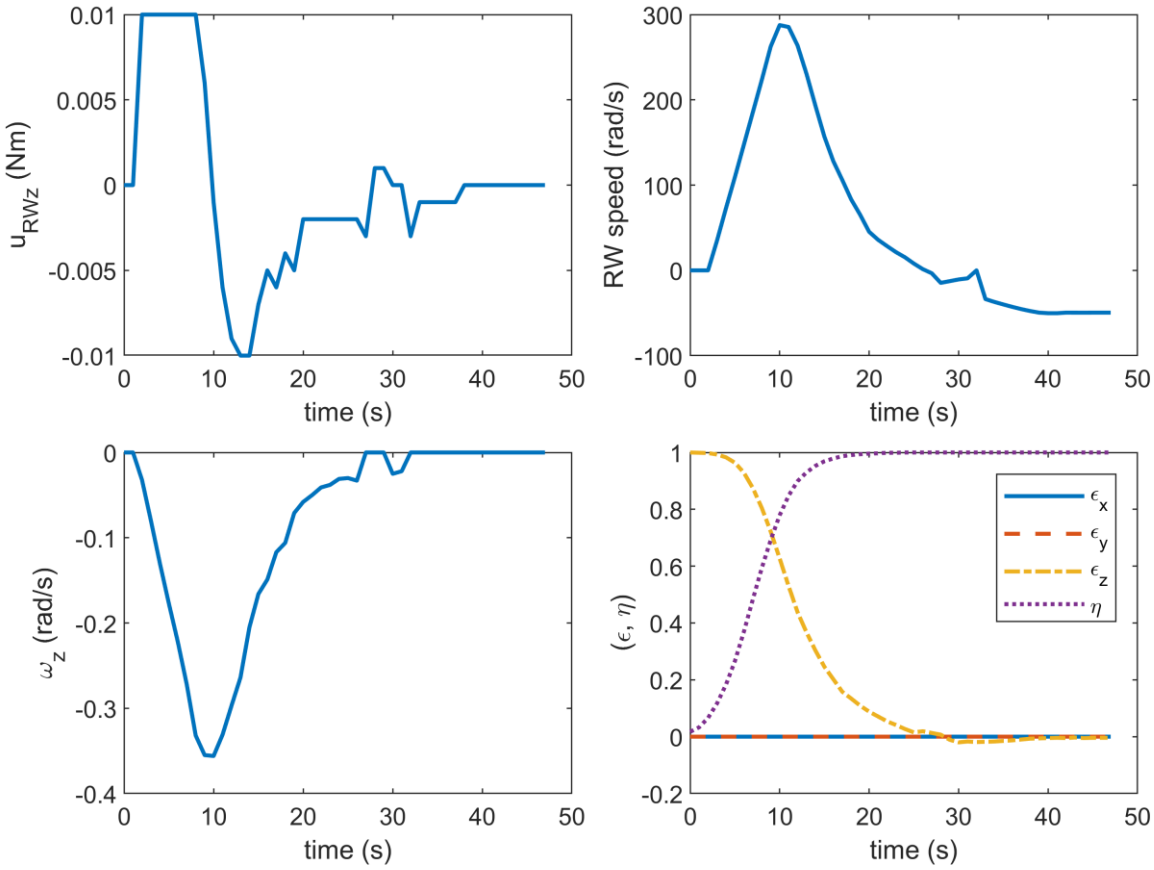

Figure 6.12 - Simulation results for attitude change maneuver using NMPC and RW.

sensitivity of the optical gyro along with other instrumental errors could have been the cause of the minor overshooting observed. Overall, the controller exhibits great performance as it was able to accurately correct the attitude of the SSM. At the end of the maneuver, a residual reaction wheel speed can also be observed. This indicates the existence of disturbances acting on the SSM and showcases the robustness of the controller.

It is important to further investigate potential causes of the overshooting observed earlier. Upon inspection of the control torques calculated by NMPC at each time interval, and the control torques applied by the RW at each time interval, small discrepancies can be observed. A comparison of these two controls are shown in Figure 6.13. A time lag between

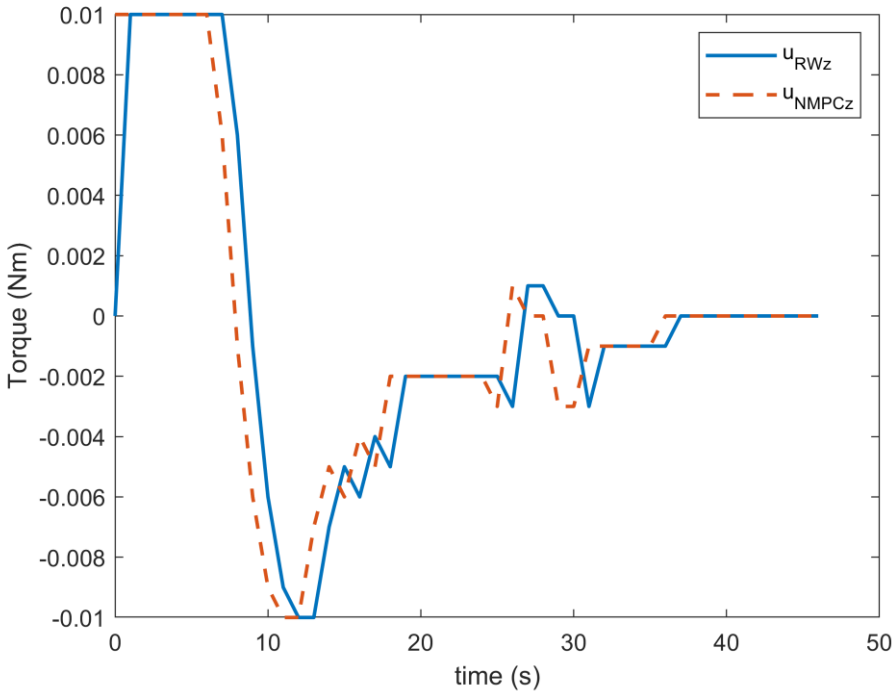

Figure 6.13 - A comparison of the NMPC control torque inputs vs the RW control torque inputs.

the two control torques can be seen. NMPC initially calculates a control torque input of 0.01 Nm, which is not achieved by the RW until one time step later, which introduces a one second lag throughout the experiment. Another discrepancy can be seen at around 30 second time interval. The change in NMPC control torque exhibits smooth behaviour, whereas the change in RW control torque seems to be sharp.

This experiment was repeated numerous times with arbitrary initial yaw angles to observe the reliability of the controller. The simulation results for these experiments are shown in Figure 6.14. The experiments show consistently good performance by the controller. The controller is capable of reducing the initial yaw angle to zero from different starting points.

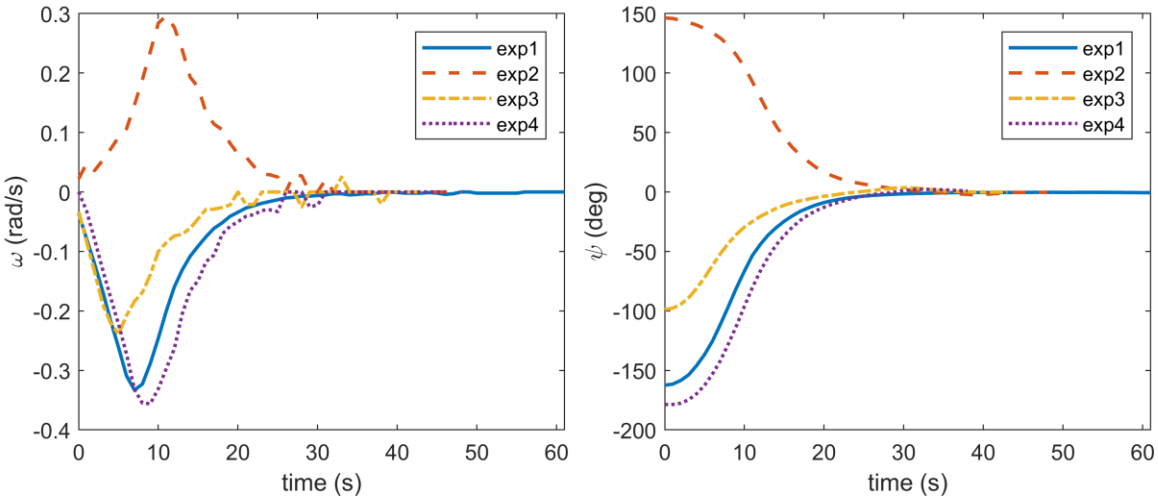

Figure 6.14 - Simulation results for NMPC attitude control experiments using RW.

It is important to note that this simulation was executed with a time step of one second. It also has a 8 second prediction horizon, and one second control horizon. A time step of one

second is a relatively large time step. The reason for choosing such a large time step was that when the simulation was initially executed with a 0.1 second time step, the system exhibited unstable behaviour with periodic oscillations. The reason for this unstable behaviour was concluded by analyzing the time it takes to run each loop iteration in LabVIEW. Each loop iteration consists of obtaining angular velocity and attitude information from sensory inputs, feeding the inputs to NMPC MATLAB code, NMPC calculating optimal control input by solving dynamics equation over prediction horizon, and finally outputting the control torque input to the RW. Due to the time lags in each system and the computational time required by NMPC, it was measured that each loop iteration took longer than 0.1 seconds to execute. The fact that this entire system is run on windows operating system and LabVIEW, further slowed down the process. However, if the program was run on a more efficient operating system such as a Linux based operating system, and a more efficient programming language such as C++, it can be expected to perform these calculations much faster. To be certain, this experiment should be performed on such devices to ensure that NMPC is viable and can be used on a real spacecraft at smaller time steps.

Chapter 7 CONCLUSION

7.1 Conclusion

This thesis analyzed and compared various control algorithms available for the underactuated attitude control problem of a spacecraft using three different actuator types—thrusters, magnetorquers, and reaction wheels. Two actuators of each type were placed along two principal axes of the CubeSat to study the behaviour of the system. It was proven in Section 2.3 that one actuator must be placed along the long axis $\hat{b}_z$ in order to access the dynamics of ω_z. The other actuator can be placed along either $\hat{b}_x$ or $\hat{b}_y$, the two short axes. Two external disturbance forces, gravity gradient torque and a time-varying sinusoidal disturbance, were added to all simulations.

Chapter 3 began with the application of super-twisting sliding mode control and nonlinear model-predictive control to detumble a 3U-CubeSat using two thrusters placed along $\hat{b}_y$ and $\hat{b}_z$ principal axes of the 3U CubeSat. The detumbling time of the CubeSat as per (3.10) was determined to be 3379.67 seconds (0.612 orbits). Next, a sub-optimal nonlinear MPC was applied to perform the same detumbling maneuver using two thrusters, in which the number of iterations to find an optimal control input was limited to five. The NMPC controller even with a sub-optimal configuration was able to detumble the spacecraft in 430.98 seconds (0.078 orbits), which is an eight-fold improvement over the STSMC.

Therefore, it can be concluded that NMPC has superior performance compared to STSMC, while exhibiting robustness.

Chapter 4 studied the use of B-dot controller along with sub-optimal NMPC to detumble a 3U-CubeSat using magnetorquers. The conventional B-dot controller along with the enhanced B-dot controller were studied. It was immediately apparent that enhanced B-dot controller outperformed the conventional B-dot. Initially the conventional B-dot algorithm was not able to detumble the CubeSat, but when stronger magnetic dipole moments were introduced it showed significant improvement in performance. When the enhanced B-dot controller was implemented, it was not required to increase the magnetic dipole moment of the magnetorquers. It was able to perform without difficulty exhibiting great resistance to external disturbances. The enhanced B-dot was able to detumble the CubeSat as per (3.10) with XZ actuation in 14307 seconds (2.59 orbits) and with YZ actuation in 13887 seconds (2.52 orbits). Then a sub-optimal NMPC controller was implemented, which also showed great performance. It was able to detumble the spacecraft and withstand external disturbances. The NMPC controller was able to detumble the spacecraft with XZ actuation in 13737 seconds (2.49 orbits), and with YZ actuation in 9538 seconds (1.73 orbits). This in comparison to enhanced B-dot experiment is only about a ~4% improvement XZ actuation and ~31% improvement in YZ actuation. Considering the complexity of MPC and simplicity of Enhanced B-dot algorithm, and the increased computational load, it can be concluded that it is more sensible to implement Enhanced B-dot algorithm over NMPC when detumbling the ESSENCE 3U CubeSat using magnetorquers.

Chapter 5 studied the use of quaternion-based nonlinear control (QBNC) and NMPC along with two reaction wheels placed orthogonally along two principal axes of the CubeSat to perform an attitude correction maneuver under zero total angular momentum assumption. Although the QBNC controller performed exceptionally under disturbance free environment, however, it had great difficulty converging when disturbances were present. This showed the lack of robustness in QBNC. Once again, MPC was able to perform much better with disturbances. It performed the attitude correction maneuver without a difficulty while maintaining constraints imposed on maximum torque and reaction wheel speed.

The application of nonlinear MPC in a real-life physical system in the form of an experiment was studied in chapter 6. An air-bearing table and a levitating satellite simulator module was utilized to study the attitude dynamics of a spacecraft system. Two types of actuators onboard the SSM were used to perform two different attitude control maneuvers. First, the air-thrusters on the SSM were used to perform a detumbling experiment using NMPC. The NMPC controller was successful in detumbling the spacecraft from various initial angular velocities. Then, the RW onboard was used to perform an attitude correction maneuver using NMPC. The SSM was initially turned in a random direction with an arbitrary yaw angle. Using the RW, the SSM was able to self-correct its attitude to by reducing the yaw angle to zero. This experiment was performed multiple times with different initial yaw angles, and the controller performed consistently each time in correcting the attitude error. This experiment validated the use of NMPC on a real spacecraft. However, it was concluded that NMPC required great amounts of onboard

memory and processing power. It was suggested that using more efficient programming languages such as C++ and Linux based operating systems can reduce the lags in the system in comparison to LABVIEW and windows operating system that was used in the experiment.

7.2 Contribution

There are several contributions that were made in the study of underactuated spacecraft attitude control.

First, an increased confidence in the ADCS subsystem of the ESSENCE 3U-CubeSat is established as a result of the rigorous analysis of attitude control capabilities presented in this thesis. Several recommendations and adjustments to the ADCS subsystem is proposed in this thesis to increase the robustness and to ensure that the ESSENCE CubeSat can operate in underactuated configuration.

Second, a novel way to model the magnetic field when analyzing magnetorquer performance was also presented in Chapter 4 . This method implements STK's built-in magnetic field model that utilizes IGRF as the main field containing over 100 coefficients, and the Olson-Pfitzer model as the external field. The built-in SGP4 propagator on STK was also utilized, which considers secular and periodic variations due to Earth oblateness, solar and lunar gravitational effects, gravitational resonance effects, and orbital decay using

a drag model. This provides an extremely accurate representation of the magnetic field experienced by the ESSENCE 3U CubeSat during orbit.

Third, an improvement to the experimental setup discussed in Chapter 6 was also made. A new RW was installed on the SSM along with a software interface to control the RW using LabVIEW. This process saw the development of a 3D printed adapter to hold the RW in place in the SSM. New electric wiring was also required. A new voltage regulator was connected to the battery to ensure that a constant voltage was applied to the RW. The software interface consisted of developing a LabVIEW code to access telemetry data of the RW such as (RW speed, RW torque). The code also consisted of a way to adjust the reaction wheel speed or torque to whatever we wanted. This process consisted of converting values and data strings to HEX values.

7.3 Future Work

Various studies can be conducted towards further progression of research in the area of underactuated spacecraft attitude control. First, more research can be done to study the non-zero angular momentum case. As previously mentioned, zero angular momentum assumption hinders the practicality of the control algorithms that employ it. This is because obtaining zero total angular momentum is unlikely in a space environment and also undesirable as explained by C. Peterson in [7]. More research needs to be done to improve this area of research if these control strategies are to be utilized on real spacecraft systems with underactuated configurations.

Recently, the use of artificial intelligence has become an increasingly studied area of research. Specifically, in control systems engineering, the use of reinforcement learning has taken over by storm. The application of reinforcement learning in controlling underactuated spacecraft attitude however is less explored. With various advancements being made in machine learning techniques, it is interesting to see how reinforcement learning can improve the underactuated attitude control problem. As mentioned in [47], development of control algorithms such as NMPC require considerable expertise of the control problem by the control engineer. Even then adjusting gains and parameters to develop an optimal controller can be extremely difficult. These complex control systems can be replaced by reinforcement learning techniques, which through machine learning, will self teach itself to compute optimal control inputs without any input from a control engineer. Over time through training, this type of controller has the potential to outperform other systematic control approaches. Furthermore, this type of controller is only computationally expensive during the training process. Once the reinforcement learning policy is trained, it can be deployed in a computationally efficient way. In other words, there is a potential in machine learning techniques that it can perform better than a conventional NMPC at reduced computational load [47]. Therefore, future studies in the underactuated attitude control problem should capitalize on this paradigm shift towards AI and machine learning and apply machine learning techniques to this control problem. With the existence of toolboxes such as Reinforcement Learning Toolbox on MATLAB, it has become easier than ever to implement such control techniques.

REFERENCES

[1] L. Markley and J. L. Crassidis, Fundamentals of Spacecraft Attitude Determination and Control, New York: Springer, 2014.

[2] MathWorks, "Robustness and Worst-Case Analysis," MathWorks, [Online]. Available: https://www.mathworks.com/help/robust/worst-case-analysis.html#:~:text=A%20robust%20control%20system%20meets,possible%20values%20of%20uncertain%20parameters.&text=You%20can%20also%20calculate%20robustness,maintaining%20stability%20or%20desired%20performan. [Accessed 09 December 2020].

[3] P. E. Crouch, "Spacecraft Attitude Control and Stabilization: Applications of Geometric Control Theory to Rigid Body Models," IEEE Transactions on Automatic Control, vol. 29, no. 4, pp. 321-331, 1984.

[4] C. I. Byrnes and A. Isidori, "On the Attitude Stabilization of Rigid Spacecraft," Automatica, vol. 27, no. 1, pp. 87-95, 1991.

[5] R. Brockett, "Asymptotic Stability and Feedback Stabilization," Differential Geometric Control Theory, pp. 181-208, 1983.

[6] H. Krishnan, N. H. McClamroch and M. Reyhanoglu, "Attitude Stabilization of a Rigid Spacecraft Using Two Momentum Wheel Actuators," Journal of Guidance, Control, and Dynamics, vol. 18, no. 2, pp. 256-263, 1995.

[7] C. Petersen, "Advances in Underactuated Spacecraft Control," University of Michigan, Michigan, 2016.

[8] B. Zhang and H. Wu, "Sliding Mode Control for Detumbling Rigid Spacecraft with Underactuated Configuration," 14th World Congress of IFAC, Beijing, 1999.

[9] S. Mobayen, "Adaptive global sliding mode control of underactuated systems using a super-twisting scheme: an experimental study," Journal of Vibration and Control, vol. 25, no. 16, pp. 2215-2224, 2019.

[10] A. Frias, "Nonlinear Attitude Control of Underactuated Spacecraft," Ryerson University, Toronto, 2018.

[11] J.-J. E. Slotine and W. Li, Applied Nonlinear Control, New Jersey: Prentice-Hall Inc., 1991.

[12] M. FURAT and I. EKER, "Experimental Evaluation of Sliding-Mode Control Techniques," Çukurova University, Adana, 2012.

[13] J. Guldne and V. I. Utkin, "The chattering problem in sliding mode systems".

[14] E. Abdulhamitbilal and E. M. Jafarov, "Design of Sliding Mode Attitude Control for Communication Spacecraft," ResearchGate, Istanbul, 2012.

[15] J. R. Dominguez, J. R. Panduro and L. Garcia, "Super-Twisting Sliding Mode in Motion Control Systems," ResearchGate, 2011.

[16] E. S. Elyoussef, E. R. D. Pieri, U. F. Moreno and M. Jungers, "Super-Twisting Sliding Modes Tracking Control of a Nonholonomic Wheeled Mobile Robot," in 10th IFAC Symposium on Robot Control, Dubrovnik, 2012.

[17] C. D. Petersen, F. Leve and I. Kolmanovsky, "Model Predictive Control of an Underactuated Spacecraft with Two Reaction Wheels," Journal of Guidance, Control, and Dynamics, vol. 40, no. 2, pp. 320-332, 2017.

[18] K. Kondo, Y. Yoshimura, M. Bando and S. Nagasaki, "Model Predictive Approach for Detumbling an Underactuated Satellite," ResearchGate, 2020.

[19] M.Mirshams and M.Khosrojerdi, "Attitude control of an underactuated spacecraft using tube-based MPC approach," K.N. Toosi University of Technology, 2015.

[20] J. B. Rawlings and D. Q. Mayne, Model Predictive Control: Theory, Computation, and Design, Santa Barbara: Nob Hill Publishing, 2019.

[21] MathWorks, "Understanding Model Predictive Control," MathWorks, [Online]. Available: https://www.mathworks.com/videos/series/understanding-model-predictive-control.html. [Accessed 15 02 2020].

[22] G. C. Walsh, R. Montgomery and S. S. Sastry, "Orientation Control Of the Dynamic Satellite," in American Control Conference, Baltimore, 1994.

[23] N. M. Horri and P. Palmer, "Practical Implementation of Attitude-Control Algorithms for an Underactuated Satellite," Journal of Guidance, Control, and Dynamics, vol. 35, no. 1, pp. 40-45, 2012.

[24] J. R. Chaurais, H. C. Ferreira, J. Y. Ishiharal and R. A. Borges§, "Attitude Control of an Underactuated Satellite Using Two Reaction Wheels," Journal of Guidance, Control, and Dynamics, vol. 38, no. 10, 2015.

[25] J.-M. Godhavn and O. Egeland, "Attitude Control of an Underactuated Satellite," Proceedings of 1995 34th IEEE Conference on Decision and Control, vol. 4, pp. 3986-3987, 1995.

[26] European Space Agency, "UoSAT-12," European Space Agency, [Online]. Available: https://earth.esa.int/web/eoportal/satellite-missions/u/uosat-12. [Accessed 14 September 2020].

[27] Wikipedia, "UK-DMC," Wikipedia, 21 August 2019. [Online]. Available: https://en.wikipedia.org/wiki/UK-DMC. [Accessed 15 September 2020].

[28] C. Peat, "Heavens Above," 2020. [Online]. Available: https://heavens-above.com/orbit.aspx?satid=25544. [Accessed 2020].

[29] Nanoracks LLC, "ISS CubeSat Deployment," 2020. [Online]. Available: https://nanoracks.com/products/iss-cubesat-deployment/. [Accessed 2020].

[30] T. Prejean and M. Lewis, "NanoRacks CubeSat Deployer (NRCSD) Interface Definition Document (IDD)," 04 06 2018. [Online]. Available: https://nanoracks.com/wp-content/uploads/NanoRacks-CubeSat-Deployer-NRCSD-Interface-Definition-Document.pdf. [Accessed 2020].

[31] eoPotal Directory, "ISS Utilization: NanoRacks-QB50," 2020. [Online]. Available: https://directory.eoportal.org/web/eoportal/satellite-missions/i/iss-nanoracks-qb50. [Accessed 2020].

[32] Wikipedia, "Euler's equations (rigid body dynamics)," 26 June 2020. [Online]. Available: https://en.wikipedia.org/wiki/Euler%27s_equations_(rigid_body_dynamics). [Accessed 08 September 2020].

[33] CubeSatShop, "Nanosatellite Micropropulsion System," [Online]. Available: https://www.cubesatshop.com/product/nanosatellite-micropropulsion-system/. [Accessed 03 September 2020].

[34] CubeSatShop, "IFM Nano Thruster for CubeSats," [Online]. Available: https://www.cubesatshop.com/product/ifm-nano-thruster/. [Accessed 03 September 2020].

[35] I. Nagesh and C. Edwards, "A multivariable super-twisting sliding mode approach," Automatica, vol. 50, no. 3, 2013.

[36] E. Dawson, "Attitude Determination and Control System Design for a CubeSat Mission," Worcester Polytechnic Institute, 2012.

[37] F. Reichel, "Attitude Control System of UWE-3: Design, Testing and Verification," Julius-Maximilians-University, Wurzburg, 2012.

[38] MathWorks, "Optimization Problem," [Online]. Available: https://www.mathworks.com/help/mpc/ug/optimization-problem.html. [Accessed June 2020].

[39] AGI, "SEET: Space Environment and Effects Tool for AGI's Systems Tool Kit (STK)," [Online]. Available: https://help.agi.com/stk/LinkedDocuments/SEET_manual.pdf. [Accessed 17 July 2020].

[40] AGI, "Orbit Propagators for Satellites," September 2016. [Online]. Available: https://help.agi.com/stk/11.0.1/Content/stk/vehSat_orbitProp_choose.htm. [Accessed 12 August 2020].

[41] G. Avanzini and F. Giulietti, "Magnetic Detumbling of a Rigid Spacecraft," Journal of Guidance, Control, and Dynamics, vol. 35, no. 4, 2012.

[42] Sinclair Interplanetary, "Picosatellite Reaction Wheels (RW-0.003)," 2018. [Online]. Available: https://78462f86-a-744dbb28-s-sites.googlegroups.com/a/sinclairinterplanetary.com/www/reactionwheels/3%20mNm-sec%20wheel%202018a.pdf?attachauth=ANoY7cpem9QnWPJ0m5d2BFIN7ADblxeIuemKSm2Q11Hy08KK8px9HEbMGSZE25zScHKQnoPaQluMxbdcMn4IBSSmGe5Uuq1yOo7F6LGM_TE9Qiy. [Accessed 21 August 2020].

[43] L. Santaguida, "Study of Autonomous Capture and Detumble of Non-Cooperative Target by a Free-Flying Space Manipulator using an Air-bearing Platform," York University, Toronto, 2020.

[44] J. Cookson, "Experimental Investigation of Spacecraft Rendezvous and Docking by Development of a 3 Degree Of Freedom Satellite Simulator Testbed," York University, 2019.

[45] VECTRONIC Aerospace, "Interface Control Document," VECTRONIC Aerospace, 2017.

[46] Wikipedia, "Conversion between quaternions and Euler angles," Wikipedia, 22 September 2020. [Online]. Available: https://en.wikipedia.org/wiki/Conversion_between_quaternions_and_Euler_angles. [Accessed 23 September 2020].

[47] MathWorks, "Reinforcement Learning for Control Systems Applications," MathWorks, [Online]. Available: https://www.mathworks.com/help/reinforcement-learning/ug/reinforcement-learning-for-control-systems-applications.html. [Accessed 15 September 2020].

[48] A. H. d. Ruiter, C. J. Damaren and J. R. Forbes, Spacecraft Dynamics and Control: An Introduction, West Sussex: John Wiley & Sons Ltd, 2013.

[49] Wikipedia, "Model predictive control," Wikipedia, 09 August 2020. [Online]. Available: https://en.wikipedia.org/wiki/Model_predictive_control. [Accessed 15 September 2020].